



Universidad de Valladolid

FACULTAD DE CIENCIAS

GRADO EN ESTADÍSTICA

**Influencia de factores meteorológicos en la
calidad de la señal eléctrica de generación
fotovoltaica en el hospital clínico de Valladolid.**

Alumno: Ismael Carbajo Valor

Tutor: Miguel Alejandro Fernández Temprano

La obsesión vence al talento.

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutor, Miguel, por su dedicación, apoyo y guía constante durante todo el desarrollo del trabajo.

Asimismo, deseo agradecer especialmente a los profesores Óscar Duque Pérez y Ángel Luis Zorita Lamadrid, del Departamento de Ingeniería Eléctrica, por facilitarme los datos utilizados en este estudio y por su colaboración desinteresada.

A mi familia, por estar siempre a mi lado, por su paciencia y por su apoyo incondicional en cada etapa del camino. Y a mis amigos, por su compañía, por los ánimos y por todos los momentos compartidos a lo largo de estos años.

Resumen

Este Trabajo de Fin de Grado presenta un análisis estadístico exhaustivo de diversas magnitudes que miden la señal de la calidad eléctrica registradas en dos centros de transformación pertenecientes a instalaciones fotovoltaicas del Hospital Clínico de Valladolid. Los datos fueron proporcionados por profesores del Departamento de Ingeniería Eléctrica de la Universidad de Valladolid. El objetivo principal ha sido estudiar la relación entre dichas magnitudes y dos variables ambientales clave: irradiancia solar y temperatura, mediante la construcción y comparación de distintos modelos estadísticos.

En una primera etapa, se emplearon modelos lineales simples y con interacción, que mostraron una capacidad explicativa limitada en la mayoría de los casos, especialmente en uno de los centros. Ante esta limitación, se optó por aplicar modelos aditivos generalizados (GAM), que permiten capturar relaciones no lineales complejas. Esta metodología logró mejorar de forma significativa el ajuste en múltiples indicadores eléctricos, destacando especialmente la relevancia de la interacción entre irradiancia y temperatura.

Se abordaron distintos retos asociados al manejo de datos reales: la organización de grandes volúmenes de registros temporales, la integración de múltiples archivos, el preprocesamiento de la información, la comprobación de supuestos estadísticos y la validación de los modelos mediante técnicas de remuestreo.

Palabras clave: modelos lineales, modelos aditivos generalizados (GAM), irradiancia, temperatura, energía solar, centros de transformación.

Abstract

This Bachelor's Thesis presents a comprehensive statistical analysis of various electrical magnitudes that measure electric power quality recorded at two transformer centers belonging to photovoltaic installations at the Hospital Clínico in Valladolid. The data were provided by professors from the Department of Electrical Engineering at the University of Valladolid. The main objective was to evaluate the relationship between these electrical variables and two key environmental factors: solar irradiance and temperature, through the construction and comparison of different statistical models.

In a first stage, simple linear models and models with interaction terms were employed, which exhibited limited explanatory capacity in most cases, particularly in one of the centers. Due to this limitation, generalized additive models (GAM) were subsequently applied, allowing the modeling of complex nonlinear relationships. This approach significantly improved the fit for multiple electrical indicators, especially highlighting the importance of the nonlinear interaction between irradiance and temperature.

Several challenges associated with real-world data processing were addressed: the organization of large volumes of time-stamped records, the integration of multiple data files, preprocessing procedures, verification of statistical assumptions, and model validation using resampling techniques.

Keywords: linear models, generalized additive models (GAM), irradiance, temperature, solar energy, transformer centers.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIII
Lista de tablas	XV
1. Introducción	1
1.1. Contexto social y científico	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Asignaturas relacionadas	3
1.5. Conocimientos adquiridos	3
1.6. Estructura de la memoria	4
2. Metodología	5
2.1. Interpolación de datos ausentes	5
2.1.1. Interpolación spline cúbica natural	5
2.1.2. Implementación en R	7

2.2. Modelado estadístico	8
2.2.1. Modelos lineales (LM)	8
2.2.2. Modelos aditivos generalizados (GAM)	11
3. Conjuntos de datos	15
3.1. Origen y recopilación de los datos	15
3.2. Estructura de los datos	17
3.3. Descripción de las variables	18
3.3.1. Calidad de la señal eléctrica	18
3.3.2. Ambiental	20
3.4. Procesado y preparación de datos	20
3.4.1. Problemas detectados en la exploración inicial	20
3.4.2. Reducción de dimensionalidad mediante análisis de correlaciones . . .	21
3.4.3. Tratamiento del conjunto total de datos	22
4. Análisis de datos	29
4.1. CT Bombas	29
4.1.1. Análisis exploratorio de correlaciones	29
4.1.2. Modelado lineal con regresión múltiple	31
4.1.3. Modelado no lineal con modelos aditivos generalizados (GAM)	31
4.2. CT Nuevo	39
4.2.1. Análisis exploratorio de correlaciones	39
4.2.2. Modelado lineal con regresión múltiple	41
4.2.3. Modelado no lineal con modelos aditivos generalizados (GAM)	44
5. Conclusiones y trabajo futuro	49
5.1. Líneas de trabajo futuras	50
Bibliografía	52

A. Descripción de variables	53
B. Resultados ampliados	55
B.1. Modelo GAM Desequilibrio.Vn.Med en CT Bombas	55
B.2. Modelo GAM Desequilibrio.Vz.Min en CT Bombas	57
B.3. Modelo lineal Desequilibrio.Vn.Min en CT Nuevo	58
B.4. Modelo lineal Tensión.de.Medio.Ciclo.V.RMS.NG.Med en CT Nuevo	59
B.5. Modelo lineal Corriente.L1.Min en CT Nuevo	60
B.6. Modelo GAM Corriente.L1.Min en CT Nuevo	61
C. Código fuente en R Markdown	63
C.0.1. Análisis exploratorio inicial	63
C.0.2. Nombres de variables constantes	68
C.0.3. Correlaciones mayores a 0.99 en CT Bombas	70
C.0.4. Correlaciones mayores a 0.99 en CT Nuevo	72
C.0.5. Informe inicial	74
C.0.6. Primeros modelos	87
C.0.7. Chequeo final datos iniciales	92
C.0.8. Modelos bombas	100
C.0.9. Modelos nuevo	108
C.0.10. Top5 bombas	122
C.0.11. Procesado	137
C.0.12. Análisis completo	180
C.0.13. Filtrado de datos	193
C.0.14. Estudio de incidencias	202
C.0.15. Estudio de incidencias modificado	209
C.0.16. Procesado modificado	216

C.0.17. Análisis completo modificado 261

C.0.18. Análisis problemas 269

C.0.19. Análisis final 280

Lista de Figuras

2.1. Ilustración de las condiciones de continuidad requeridas en los splines cúbicos. Adaptado de [7].	6
2.2. Ilustración del concepto de interpolación spline cúbica de [14]. Los puntos azules indican los valores observados, la línea continua azul representa el spline interpolado y los asteriscos rojos los valores estimados.	7
2.3. Representación conceptual de un modelo lineal múltiple con dos predictores. .	10
2.4. Ejemplo de superficie estimada por un GAM con interacción bivariada entre dos predictores (aparecerá en el análisis).	12
3.1. Equipos empleados para la medición de calidad eléctrica e irradiancia en el hospital, fotos proporcionadas por el Departamento de Ingeniería Informática de la UVa.	16
3.2. Esquema general del sistema fotovoltaico del hospital proporcionado por el Departamento de Ingeniería Informática de la UVa.	17
3.3. Cobertura temporal con los archivos disponibles para cada uno de los CT. . .	18
3.4. Matrices de correlaciones entre las variables con varianza mayor que 0 en los dos CT.	22
4.1. Efecto no lineal de la irradiancia.	33
4.2. Efecto no lineal de la temperatura.	34
4.3. Interacción no lineal entre temperatura e irradiancia.	34
4.4. Contornos del predictor lineal del modelo GAM en el espacio irradiancia-temperatura.	35
4.5. Superficie 3D del predictor lineal total estimado por el modelo GAM.	35

4.6. Efecto no lineal de la irradiancia. 37

4.7. Efecto no lineal de la temperatura. 37

4.8. Interacción no lineal entre temperatura e irradiancia. 38

4.9. Representación de contornos del término de interacción no lineal entre irradiancia y temperatura. 38

4.10. Superficie 3D del efecto combinado de irradiancia y temperatura en el predictor lineal. 39

4.11. Gráficos de diagnóstico del modelo lineal para *Desequilibrio.Vn.Min.* 42

4.12. Gráficos de diagnóstico del modelo lineal para *Tensión.de.Medio.Ciclo.V.RMS.NG.Med.* 43

4.13. Gráficos de diagnóstico del modelo lineal para *Corriente.L1.Min.* 44

4.14. Efecto no lineal de la irradiancia. 46

4.15. Efecto no lineal de la temperatura. 46

4.16. Interacción no lineal entre temperatura e irradiancia. 47

4.17. Contornos del predictor lineal del modelo GAM. 47

4.18. Superficie 3D del predictor lineal total estimado por el modelo. 48

Lista de Tablas

3.1. Presencia de datos para cada bloque en los distintos archivos. 17

3.2. Resumen de la estructura de variables tras la exploración inicial. 21

3.3. Resumen de variables del conjunto CT Bombas. 23

3.4. Resumen de variables del conjunto CT Nuevo. 23

3.5. Resumen de variables tras el filtrado en CT Bombas. 24

3.6. Resumen de variables tras el filtrado en CT Nuevo. 24

4.1. Top 10 de variables con mayor correlación (absoluta) con la irradiancia. . . . 30

4.2. Top 10 de variables con mayor correlación (absoluta) con la temperatura. . . 30

4.3. Top 10 variables con mayor capacidad explicativa según el modelo lineal múltiple con interacción. 31

4.4. Top 10 variables con mayor capacidad explicativa según el modelo GAM con interacción no lineal. Se muestran también las métricas correspondientes al modelo aditivo sin interacción. 32

4.5. Top 10 de variables con mayor correlación (absoluta) con la irradiancia en el CT Nuevo. 40

4.6. Top 10 de variables con mayor correlación (absoluta) con la temperatura en el CT Nuevo. 40

4.7. Top 10 variables con mayor capacidad explicativa según el modelo lineal múltiple con interacción (CT Nuevo). 41

4.8. Top 10 variables con mayor capacidad explicativa según el modelo GAM con interacción no lineal. Se incluyen también las métricas del modelo aditivo sin interacción. 45

Capítulo 1

Introducción

1.1. Contexto social y científico

El cambio hacia un modelo energético sostenible es uno de los grandes retos del siglo XXI. En este proceso, las energías renovables, y en particular la solar fotovoltaica, han adquirido gran importancia gracias a su bajo impacto ambiental, su modularidad y su facilidad de instalación en entornos urbanos y rurales. España se ha consolidado como uno de los países líderes en Europa en capacidad solar. En 2024, contaba con aproximadamente 33,7 GW instalados, y la generación fotovoltaica ya representa más del 20 % de la electricidad nacional. La energía solar se ha convertido en la principal fuente eléctrica del país, incluso superando a la eólica en algunos períodos [8, 12].

En instalaciones sanitarias críticas, como son los hospitales, la implementación de energía solar ayuda tanto a la sostenibilidad medioambiental como a la necesidad de garantizar un suministro eléctrico continuo y fiable. La posibilidad de generar energía *in situ* permite reducir la dependencia de la red externa y mejorar la autonomía, asegurando el suministro continuo ante posibles interrupciones. Esto resulta muy relevante en el ámbito hospitalario, donde la fiabilidad eléctrica es vital para garantizar la seguridad del paciente y el funcionamiento de equipos médicos sensibles y muy costosos. Diversos estudios han subrayado la importancia de mantener unos niveles adecuados de calidad eléctrica en entornos hospitalarios, ya que posibles perturbaciones pueden afectar negativamente al equipamiento clínico [11]. Por otro lado, se ha observado que el uso de sistemas fotovoltaicos puede mejorar la resiliencia del suministro en centros sanitarios, siempre que estén correctamente integrados [1].

Sin embargo, a pesar de los avances en su implementación, aún existen incertidumbres respecto al impacto que pueden tener las condiciones ambientales sobre la calidad de la señal eléctrica generada. Diferentes factores pueden influir en el comportamiento de los inversores, en la estabilidad de la red interna o en la aparición de distorsiones armónicas, afectando indirectamente al rendimiento general del sistema. Comprender si estas variaciones ambientales tienen efectos significativos es fundamental para garantizar un suministro eléctrico fiable, especialmente en contextos como este, donde la calidad energética es un requisito esencial.

1.2. Motivación

Este trabajo surge a partir de una colaboración con el Departamento de Ingeniería Eléctrica de la Universidad de Valladolid, en concreto con los profesores Óscar Duque Pérez y Ángel Luis Zorita Lamadrid, con quienes se ha trabajado de forma conjunta en el análisis de datos que ellos mismos recogen y supervisan, en el Hospital Clínico Universitario de Valladolid. El estudio se centra en dos instalaciones fotovoltaicas independientes ubicadas en diferentes zonas del hospital y orientadas en distintas direcciones, lo que permite observar una mayor diversidad en las condiciones de funcionamiento.

El objetivo principal de esta colaboración es investigar si variables ambientales, en concreto la irradiancia solar y la temperatura, influyen en la calidad de la señal eléctrica generada por las placas solares. Mientras que muchos estudios sobre instalaciones fotovoltaicas se han centrado en el rendimiento energético o la eficiencia de producción, este trabajo aborda un tema menos explorado: la calidad de la señal eléctrica. Esta incluye parámetros como la estabilidad, la distorsión o la forma de onda, aspectos especialmente relevantes en este tipo de entornos, donde incluso pequeñas anomalías pueden comprometer la seguridad y el correcto funcionamiento de los equipos.

Desde el punto de vista estadístico, el análisis de este tipo de datos representa un desafío especialmente interesante. Las variables ambientales y eléctricas se registran de forma continua y con alta frecuencia, lo que implica trabajar con grandes volúmenes de información, presencia de ruido, valores atípicos y dependencia temporal. A estas dificultades se suman posibles efectos propios de la instalación, como su antigüedad, estado de mantenimiento o configuración. Todo ello hace necesario un enfoque flexible, basado en la aplicación de técnicas estadísticas que se irán seleccionando y adaptando en función de los datos, los resultados obtenidos y de los objetivos concretos de cada fase del análisis.

1.3. Objetivos

Se plantean una serie de objetivos específicos que orientan el desarrollo del análisis:

- Analizar la calidad de los datos disponibles, ya que se trata del primer estudio en este ámbito realizado con observaciones del Hospital Clínico Universitario de Valladolid. Aunque los equipos de medida instalados en las dos ubicaciones son de la misma marca, existen indicios de que la calidad de los datos recogidos no es homogénea entre ambas instalaciones.
- Estudiar la relación entre las variables ambientales (irradiancia y temperatura) y distintos indicadores de calidad de la señal eléctrica, para determinar, en su caso, cuáles de ellos y en qué medida se ven afectados por las condiciones ambientales.
- Comparar el comportamiento eléctrico de las dos instalaciones fotovoltaicas orientadas en direcciones diferentes dentro del hospital, con el fin de saber si se comportan de manera homogénea o existen diferencias de interés en lo que se refiere a la influencia de las variables ambientales en la calidad de la señal.

En resumen, el objetivo global del trabajo es generar conocimiento que pueda servir de apoyo para la toma de decisiones en cuanto al diseño, monitorización y mantenimiento de instalaciones solares en entornos críticos como el hospitalario.

1.4. Asignaturas relacionadas

Para la realización de este trabajo se han aplicado conocimientos adquiridos en asignaturas como:

- Estadística Descriptiva
- Inferencia Estadística I
- Inferencia Estadística II
- Computación Estadística
- Regresión y ANOVA
- Análisis de Datos
- Modelos Lineales
- Modelos Estadísticos Avanzados
- Métodos Estadísticos de Computación Intensiva

1.5. Conocimientos adquiridos

Durante el desarrollo de este trabajo se ha adquirido una serie de competencias clave tanto desde el punto de vista técnico como metodológico. Una de las principales ha sido aprender a manejar una gran cantidad de variables respuesta frente a un conjunto muy limitado de variables explicativas. Esta situación es poco habitual desde el punto de vista estadístico, lo que ha requerido adaptar el enfoque del trabajo de manera especial.

Asimismo, se ha aprendido a gestionar y procesar volúmenes extensos de datos procedentes de sensores industriales, con estructuras horarias y gran cantidad de inconsistencias. Esto implicó desarrollar criterios claros de selección, limpieza y alineación temporal, así como asegurar la coherencia entre distintas fuentes y dispositivos de medición. En particular, ha exigido una organización rigurosa de los datos por fechas, instalaciones y tipos de variable.

Otro aspecto importante ha sido la aplicación e interpretación de modelos estadísticos más avanzados como los modelos aditivos generalizados (GAM), que han permitido explorar relaciones no lineales e interacciones entre variables ambientales de forma más flexible. Esta herramienta ha resultado especialmente útil en contextos donde los modelos lineales tradicionales no lograban capturar adecuadamente la estructura subyacente de los datos.

Finalmente, el trabajo también ha permitido desarrollar habilidades para sintetizar información técnica compleja y estructurarla en informes claros y reproducibles, incorporando visualizaciones, validación de supuestos y evaluación comparativa entre modelos.

Estos aprendizajes no solo son aplicables al contexto específico del análisis de sistemas fotovoltaicos, sino que constituyen una base sólida para abordar problemas de modelado, análisis y diagnóstico en otras áreas.

1.6. Estructura de la memoria

A continuación se describe la estructura del documento:

Capítulo 1 Introducción: Se presenta el contexto social y técnico del estudio, el problema que se pretende abordar, se justifica su relevancia, y se detalla la motivación personal y académica del trabajo. También se especifican los objetivos del estudio, las asignaturas relacionadas con el proyecto y los conocimientos adquiridos durante su desarrollo.

Capítulo 2 Metodología: Se explican los procedimientos seguidos para el análisis de los datos, detallando las técnicas estadísticas utilizadas y su adecuación al problema planteado. Se justifica la elección de métodos exploratorios, de modelado y de validación.

Capítulo 3 Datos: Se describe el origen de los datos utilizados, correspondientes a dos instalaciones fotovoltaicas en el Hospital Clínico Universitario de Valladolid. Se analiza su estructura, el proceso de obtención, los problemas encontrados y las decisiones tomadas durante la limpieza y el preprocesamiento.

Capítulo 4 Análisis de los datos: Se presentan los distintos tipos de análisis realizados, organizados por bloques o variables de interés. Se muestran los resultados obtenidos y se interpretan en función de su impacto sobre la calidad de la señal eléctrica. Se incluyen gráficos, tablas, modelos y pruebas estadísticas.

Capítulo 5 Conclusiones y trabajo futuro: Se recogen las principales conclusiones del estudio, reflexionando sobre los hallazgos obtenidos y sus implicaciones prácticas. También se señalan las limitaciones del trabajo y se proponen posibles líneas de mejora o continuidad para estudios futuros.

Capítulo 2

Metodología

En este capítulo se presentan las herramientas y técnicas estadísticas empleadas en el análisis posterior, desde un punto de vista puramente teórico. Se describen los métodos utilizados para tratar los valores ausentes en las variables ambientales, así como los modelos estadísticos aplicados para estudiar la relación entre las variables eléctricas y las condiciones ambientales. Esta información servirá como guía para la interpretación de resultados.

2.1. Interpolación de datos ausentes

La interpolación es un procedimiento ampliamente utilizado en el tratamiento de series temporales, especialmente en contextos donde pueden existir vacíos puntuales en los registros. Su objetivo principal es estimar los valores ausentes de una variable continua basándose en la información existente en los puntos vecinos, manteniendo así la coherencia y continuidad de la señal.

2.1.1. Interpolación spline cúbica natural

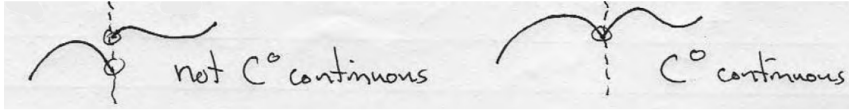
Una *spline* es una función compuesta por varios polinomios definidos por tramos que se ajustan suavemente en determinados puntos denominados *nodos*. En particular, un spline cúbico utiliza polinomios de grado tres y garantiza continuidad no solo en el valor de la función, sino también en su primera y segunda derivada en los puntos de empalme [5, 4].

La forma general de un spline cúbico definido en un intervalo $[a, b]$ con $n + 1$ nodos $\{x_0, x_1, \dots, x_n\}$ consiste en un conjunto de funciones $S_i(x)$, una por cada subintervalo $[x_i, x_{i+1}]$, definidas como:

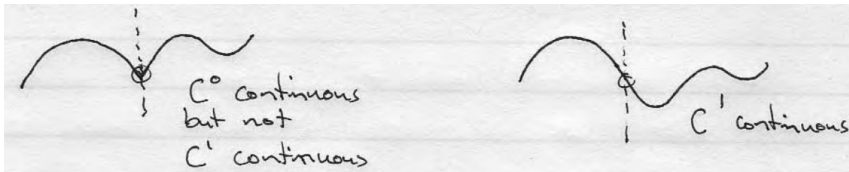
$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad \text{para } x \in [x_i, x_{i+1}]$$

Las condiciones que deben cumplir estos polinomios para formar un spline cúbico son:

- $S_i(x_i) = y_i$ y $S_i(x_{i+1}) = y_{i+1}$ (**Continuidad C^0** : la función pasa por los nodos).
- $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$ (**Continuidad C^1** : continuidad de la primera derivada).
- $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$ (**Continuidad C^2** : continuidad de la segunda derivada).



C^0 : continuidad en el valor de la función.



C^1 : continuidad en la primera derivada.



C^2 : continuidad en la segunda derivada.

Figura 2.1: Ilustración de las condiciones de continuidad requeridas en los splines cúbicos. Adaptado de [7].

En el caso de un spline cúbico natural, se impone además que la segunda derivada en los extremos del dominio sea nula, es decir:

$$S''_0(x_0) = 0, \quad S''_{n-1}(x_n) = 0$$

Estas condiciones adicionales reducen la curvatura en los bordes, generando una transición más suave en los extremos y evitando oscilaciones artificiales [7].

Este tipo de interpolación es particularmente útil en series temporales físicas, como las ambientales, donde se espera un comportamiento suave y continuo a lo largo del tiempo. En comparación con otras técnicas (como la interpolación lineal o por último valor), los splines cúbicos naturales ofrecen una mejor aproximación local sin introducir saltos ni cambios bruscos, y son robustos frente a valores atípicos si se aplican a intervalos bien delimitados.

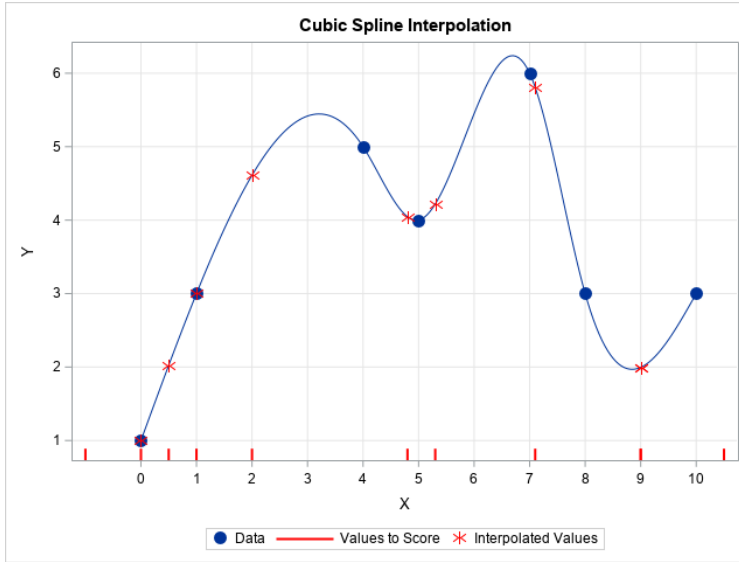


Figura 2.2: Ilustración del concepto de interpolación spline cúbica de [14]. Los puntos azules indican los valores observados, la línea continua azul representa el spline interpolado y los asteriscos rojos los valores estimados.

El sistema de ecuaciones que determina los coeficientes a_i, b_i, c_i, d_i para un spline cúbico natural puede resolverse de forma eficiente mediante métodos matriciales tridiagonales, lo que facilita su aplicación incluso en conjuntos de datos extensos [3].

2.1.2. Implementación en R

La interpolación se llevó a cabo mediante la función `na.spline()` de la librería `zoo`, que permite aplicar interpolación spline sobre series temporales con valores ausentes. A diferencia de otras técnicas como el *forward fill* o el uso de medias móviles, esta aproximación reconstruye la forma de la señal de forma suave y coherente con los patrones de variación observados.

La elección de esta técnica respondió a tres criterios principales:

- Las variables ambientales, como la temperatura o la irradiancia, tienden a evolucionar de forma continua y suave en el tiempo, lo que encaja bien con la naturaleza de los splines cúbicos.
- Los valores ausentes eran poco frecuentes y estaban generalmente aislados (entre 1 y 4 observaciones en la mayoría de los bloques), lo que favorece el uso de una técnica que respete la forma local de la serie.

- La interpolación spline garantiza una transición natural entre observaciones reales, evitando saltos bruscos y artefactos numéricos que pueden aparecer con métodos más simples.

Esta función es ampliamente utilizada en el análisis de series temporales en R. Para más detalles, véase [18].

2.2. Modelado estadístico

2.2.1. Modelos lineales (LM)

Los modelos lineales constituyen una herramienta fundamental en estadística para modelar la relación entre una variable respuesta continua Y y una o varias variables explicativas $X_1, X_2, \dots, X_p$. Su forma general se expresa como:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi} + \varepsilon_i, \quad i = 1, \dots, n$$

donde:

- β_0 es el intercepto (valor medio de Y cuando todos los predictores valen cero),
- β_j representa el efecto marginal de cada predictor X_j sobre Y ,
- $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ son los términos de error aleatorio, independientes e idénticamente distribuidos.

Este modelo se basa en varios supuestos fundamentales [10, 9]:

1. **Linealidad:** la relación entre los predictores y la respuesta es lineal en los parámetros.
2. **Independencia:** los errores son independientes entre sí.
3. **Homoscedasticidad:** la varianza de los errores es constante en todos los niveles de los predictores.
4. **Normalidad:** los errores se distribuyen normalmente, lo que es crucial para los contrastes de hipótesis.

Parámetros e interpretación

Los coeficientes estimados $\hat{\beta}_j$ se obtienen mediante el método de *mínimos cuadrados ordinarios* (OLS), buscando minimizar la suma de los residuos al cuadrado:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \left(Y_i - \beta_0 - \sum_{j=1}^p \beta_j X_{ji} \right)^2$$

Cada coeficiente $\hat{\beta}_j$ representa el cambio esperado en Y ante un aumento de una unidad en X_j , manteniendo constantes el resto de predictores.

Evaluación del modelo

El ajuste del modelo se evalúa principalmente mediante el coeficiente de determinación R^2 , que mide la proporción de variabilidad de Y explicada por el modelo:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

donde SS_{res} es la suma de cuadrados de los residuos, y SS_{tot} la suma total de cuadrados respecto a la media. No obstante, dado que R^2 siempre crece al añadir más predictores, se utiliza el R^2_{ajustado} para penalizar modelos sobreparametrizados:

$$R^2_{\text{ajustado}} = 1 - \left(\frac{n-1}{n-p-1} \right) (1 - R^2)$$

Asimismo, se evalúan los *p-valores* asociados a cada coeficiente para comprobar su significancia estadística mediante contraste de hipótesis:

$$H_0 : \beta_j = 0 \quad \text{vs} \quad H_1 : \beta_j \neq 0$$

Visualización conceptual

En el caso de dos predictores, el modelo se puede representar geoméricamente como un plano en el espacio tridimensional formado por las dos variables explicativas X_1 , X_2 y la variable respuesta Y . La siguiente figura representa este concepto para *Weight*, *Horsepower* y *MPG*, respectivamente:

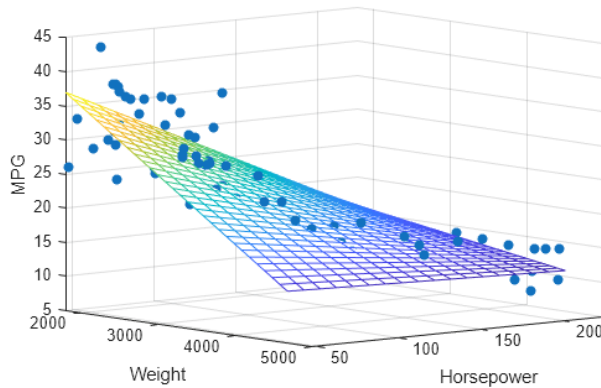


Figura 2.3: Representación conceptual de un modelo lineal múltiple con dos predictores.

Este plano refleja el valor estimado de Y para cada combinación de valores de X_1 e X_2 , asumiendo un efecto lineal y aditivo de los predictores. Este tipo de visualización permite entender de forma intuitiva cómo el modelo ajusta una superficie plana que intenta minimizar la distancia (error) con respecto a los puntos observados.

Implementación en R

El ajuste de modelos lineales en R se realiza mediante la función `lm()`. A modo ilustrativo, un modelo que incluye irradiancia, temperatura y su interacción se especificaría así:

```
modelo <- lm(Y ~ Irradiancia * Temperatura, data = datos)
```

El término `Irradiancia * Temperatura` incluye los efectos principales y su interacción. A partir del modelo ajustado pueden extraerse coeficientes, residuos, R^2 , y realizar tests de hipótesis sobre los parámetros.

Esta función es básica para la aplicación de modelos lineales en R. Para más detalles, véase [13].

Limitaciones

A pesar de su utilidad, los modelos lineales presentan limitaciones importantes:

- No pueden capturar relaciones no lineales a menos que se transformen los predictores.
- Son sensibles a colinealidad entre variables explicativas.

- Pueden resultar inadecuados cuando las condiciones de normalidad u homocedasticidad se violan.

Por estas razones, en muchos casos es conveniente utilizar otros enfoques más flexibles, como el que se describe en la siguiente sección.

2.2.2. Modelos aditivos generalizados (GAM)

Los modelos aditivos generalizados extienden los modelos lineales al permitir relaciones no lineales entre la variable respuesta Y y las variables explicativas, manteniendo la interpretabilidad y estructura aditiva. Fueron introducidos formalmente por Hastie y Tibshirani [6] como una combinación de regresión no paramétrica y modelos lineales generalizados.

La forma general de un GAM es:

$$g(\mathbb{E}[Y]) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$$

donde:

- $g(\cdot)$ es la función de enlace (por ejemplo, la identidad o el logaritmo),
- $f_j(\cdot)$ son funciones suaves estimadas a partir de los datos, generalmente mediante splines,
- β_0 es el intercepto global.

Estas funciones f_j no se restringen a formas paramétricas específicas, sino que se ajustan con técnicas como splines penalizados, lo que permite captar patrones complejos sin sobreajustar los datos [16].

Características principales

- **Flexibilidad no paramétrica:** los GAM permiten capturar relaciones no lineales entre predictores y respuesta sin necesidad de especificar una forma funcional concreta.
- **Estructura aditiva:** cada predictor contribuye con un efecto individual (o combinado, en caso de interacción), lo que facilita la interpretación.
- **Penalización:** para evitar sobreajuste, se penaliza la rugosidad de las funciones f_j , controlando su complejidad.

Comparación con modelos lineales

Mientras que los modelos lineales asumen una relación estrictamente lineal:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \varepsilon$$

los GAM permiten representar relaciones de forma más general:

$$Y = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + \varepsilon$$

De hecho, un GAM se reduce a un modelo lineal cuando todas las funciones suaves f_j corresponden a funciones identidad, es decir, $f_j(X_j) = \beta_j X_j$. Por ello, los modelos lineales pueden considerarse un caso particular dentro del marco más amplio de los modelos aditivos generalizados.

Visualización e interpretación

Cada función $f_j(X_j)$ puede visualizarse como una curva que describe el efecto marginal de esa variable explicativa sobre la respuesta, manteniendo constantes los demás factores. En caso de utilizar términos de interacción bivariada, como $f(X_1, X_2)$, estos pueden representarse mediante superficies o mapas de contorno tridimensionales.

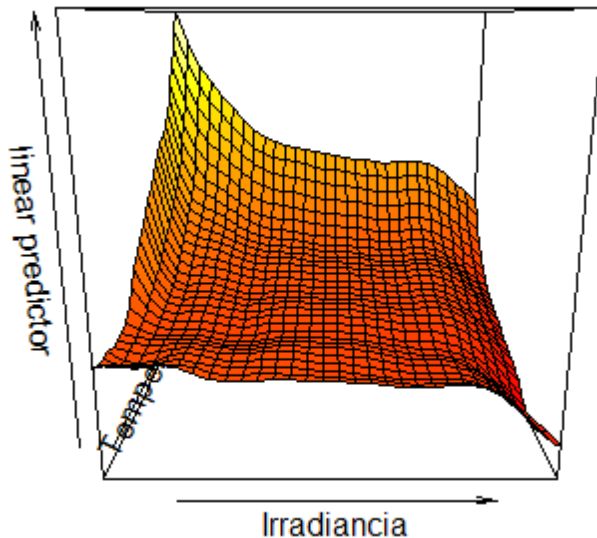


Figura 2.4: Ejemplo de superficie estimada por un GAM con interacción bivariada entre dos predictores (aparecerá en el análisis).

Esta representación permite detectar efectos conjuntos, como zonas de mayor impacto combinando valores de dos predictores.

Evaluación del modelo

Al igual que en los modelos lineales, el ajuste del modelo se evalúa con métricas como:

- El R^2 ajustado.
- La desviación explicada.
- El *Generalized Cross-Validation* (GCV), que mide el compromiso entre ajuste y complejidad del modelo.

El GCV es una estimación del error de predicción del modelo, penalizando la complejidad excesiva. Favorece modelos con buen ajuste y baja complejidad. Cuanto menor sea el valor del GCV, mejor será el equilibrio entre precisión y generalización. Este criterio es especialmente útil para comparar distintos modelos GAM con diferente número de términos suaves o estructuras de interacción.

Implementación en R

En R, el paquete `mgcv` [15] proporciona una implementación robusta de GAM mediante la función `gam()`. Un ejemplo de modelo con términos suavizados sería:

```
library(mgcv)
modelo_gam <- gam(Y ~ s(Irradiancia) + s(Temperatura), data = datos)
```

Además, se pueden incorporar interacciones no lineales entre predictores mediante funciones como `te()` (tensor product) o `ti()` (tensor interaction). Por ejemplo:

```
modelo_gam_interaccion <- gam(Y ~ s(Irradiancia) + s(Temperatura)
+ ti(Irradiancia, Temperatura), data = datos)
```

El término `ti(Irradiancia, Temperatura)` permite modelar una interacción bivariada suavizada entre los dos predictores, lo cual resulta útil cuando se sospecha que su efecto conjunto sobre la variable respuesta no es meramente aditivo.

Esta función es básica para la aplicación de modelos aditivos en R. Para más detalles, véase [17].

Limitaciones

Aunque muy potentes, los GAM también presentan ciertas limitaciones:

- Mayor complejidad computacional que los modelos lineales clásicos.
- Riesgo de sobreajuste si no se controla adecuadamente la penalización.
- Interpretación menos directa en comparación con modelos paramétricos.

Sin embargo, su capacidad para detectar relaciones no lineales de forma flexible y automatizada los convierte en herramientas muy útiles en contextos como el análisis ambiental, médico o energético.

Capítulo 3

Conjuntos de datos

Este capítulo describe el conjunto de datos utilizado en el presente trabajo, incluyendo su origen, la estructura general de los archivos, las variables disponibles y las principales acciones llevadas a cabo para su preparación. A lo largo del mismo, se documentan también los problemas detectados durante el proceso de limpieza, así como las decisiones tomadas para construir una base de datos adecuada para el análisis posterior.

3.1. Origen y recopilación de los datos

Los datos utilizados en este estudio han sido registrados en dos instalaciones fotovoltaicas independientes situadas en el Hospital Clínico Universitario de Valladolid. Aunque ambas pertenecen al mismo complejo y están gestionadas desde un sistema común, cada una cuenta con una estructura propia de captación y medición. Las dos instalaciones utilizan equipos de la misma marca pero distinto modelo, tanto para los inversores como para los sensores de medida, lo cual se espera que garantice cierta homogeneidad en los registros.

En concreto, las mediciones eléctricas se realizaron con dos analizadores trifásicos de calidad de la marca Fluke, modelos 434 (clase S) y 435-II (clase A), ambos diseñados para detectar, analizar y prevenir problemas en entornos industriales. Estos dispositivos, que pueden verse en la Figura 3.1a, permiten la monitorización avanzada de una gran cantidad de parámetros, cumpliendo la norma IEC 61000-4-30. Por su parte, la medida de irradiancia se obtuvo mediante una Célula Calibrada 5V Compensada, del fabricante Atersa, instalada en la azotea del hospital. Esta célula, que aparece en la Figura 3.1b, permite registrar tanto la radiación solar incidente como la temperatura ambiente y la temperatura de la propia célula, con salidas analógicas independientes.



(a) Sistema de análisis de calidad eléctrica (Fluke 434/435-II).



(b) Célula calibrada para medición de irradiancia (Atersa).

Figura 3.1: Equipos empleados para la medición de calidad eléctrica e irradiancia en el hospital, fotos proporcionadas por el Departamento de Ingeniería Informática de la UVA.

A lo largo de este trabajo se hará referencia a estas dos instalaciones como CT Bombas y CT Nuevo. La primera de ellas, CT Bombas, es la más antigua de las dos, mientras que CT Nuevo constituye una incorporación más reciente al sistema fotovoltaico del hospital. Esta diferencia en la antigüedad puede tener implicaciones en el estado de los equipos o en su comportamiento eléctrico.

Ambas instalaciones presentan distinta orientación, tal y como se puede apreciar en el esquema de la Figura 3.2. Esta diferencia proporciona una variedad extra al análisis, ya que permite estudiar el comportamiento del sistema ante exposiciones solares distintas a lo largo del día, lo cual influye en la producción energética y, potencialmente, en la calidad de la señal eléctrica.

Los datos disponibles se dividen en tres bloques principales. Por un lado, se encuentran los datos eléctricos correspondientes a CT Bombas y CT Nuevo, que incluyen diversas variables relacionadas con la calidad de la señal eléctrica. Por otro lado, se dispone de un conjunto adicional que contiene las medidas ambientales, recogidas por una estación meteorológica situada en las proximidades de las instalaciones.

En cuanto a la forma de los datos, se ha trabajado con una frecuencia uniforme de 10 minutos para todos los bloques. Sin embargo, es importante señalar que las mediciones no se registran originalmente con esta frecuencia. En realidad, los valores se capturan a una frecuencia mayor y, posteriormente, se agregan mediante promedios para generar una serie temporal con un intervalo de 10 minutos entre observaciones.

Inicialmente, se dispuso de un conjunto parcial de datos correspondiente a un período de tres semanas del mes de enero, concretamente entre los días 9 y 31. Esta primera entrega permitió comenzar con una fase exploratoria del análisis. Sin embargo, no fue hasta bastante tiempo después cuando se pudo acceder progresivamente al resto de los registros, correspondientes a un rango temporal mucho más amplio. En total, los datos utilizados se encuentran en 16 archivos distintos, correspondientes a los diferentes tramos temporales.

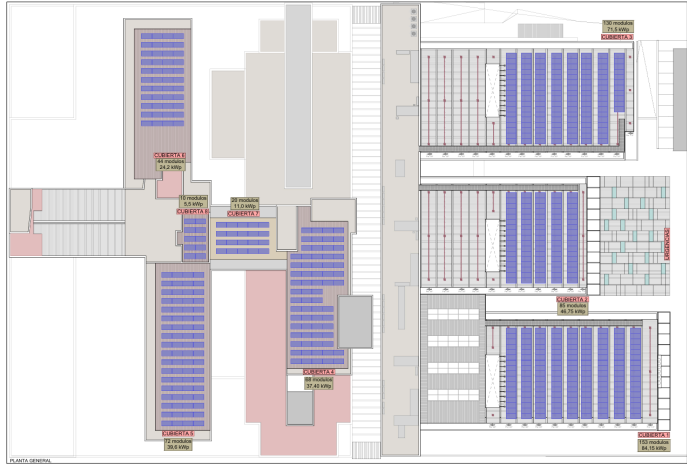


Figura 3.2: Esquema general del sistema fotovoltaico del hospital proporcionado por el Departamento de Ingeniería Informática de la UVa.

3.2. Estructura de los datos

Los 16 archivos de datos se encuentran en formato Excel, cada uno de los cuales contiene hasta tres hojas diferentes: una correspondiente a los datos del CT Bombas, otra al CT Nuevo, y una tercera a las medidas ambientales. Sin embargo, no todos los archivos contienen información de los tres bloques, lo que introduce cierta heterogeneidad estructural que debe tenerse en cuenta en el análisis.

Archivo	CT Bombas	CT Nuevo	Ambiental
Datos1 09-01 a 09-03	Sí	Sí	Sí
Datos2 09-03 a 09-05	Sí	No	Sí
Datos3 09-05 a 09-07	Sí	No	Sí
Datos4 10-01 a 10-03	No	Sí	Sí
Datos5 10-03 a 10-05	No	Sí	Sí
Datos6 10-05 a 10-07	Sí	Sí	Sí
Datos7 11-03 a 11-05	Sí	Sí	Sí
Datos8 11-05 a 11-07	Sí	Sí	Sí
Datos9 11-07 a 11-09	Sí	Sí	Sí
Datos10 11-09 a 11-11	Sí	Sí	Sí
Datos11 11-11 a 11-13	No	Sí	Sí
Datos12 11-13 a 11-15	Sí	Sí	Sí
Datos13 02-04 a 03-04	Sí	Sí	Sí
Datos14 03-04 a 04-04	Sí	Sí	Sí
Datos15 04-04 a 12-04	Sí	No	Sí
Datos16 12-05 a 27-05	Sí	Sí	Sí

Tabla 3.1: Presencia de datos para cada bloque en los distintos archivos.

En la Tabla 3.1 se resume la disponibilidad de cada bloque de datos en los distintos archivos. La información ambiental está presente en todos los casos, mientras que las medidas eléctricas del CT Bombas y del CT Nuevo no siempre aparecen conjuntamente.

En la Figura 3.3 se representa de forma visual la cobertura temporal aproximada de los archivos disponibles, desde finales de agosto de 2024 hasta finales de mayo de 2025. Cada barra indica los períodos en los que se dispone de datos válidos para cada una de las dos instalaciones fotovoltaicas. Como puede observarse, debido a la ausencia de ciertos bloques en algunos archivos, hay intervalos en los que se cuenta con datos de ambas instalaciones, otros en los que solo está disponible una de ellas, y algunos en los que no se dispone de datos de ninguna, dado que los archivos no cubren el rango de manera completa.

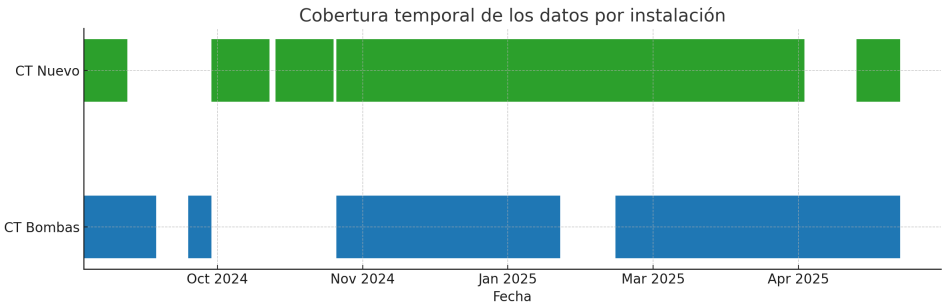


Figura 3.3: Cobertura temporal con los archivos disponibles para cada uno de los CT.

3.3. Descripción de las variables

En esta sección se describen las variables de los diferentes conjuntos. Aunque en el caso de las medidas de calidad eléctrica, no todos los archivos contienen exactamente las mismas variables, la estructura general de los conjuntos es coherente entre ellos. Es decir, las variables comparten un mismo esquema de nombres, tipos de medición y organización, lo que permite hacer una descripción conjunta y estructurada. Las diferencias específicas en la presencia o ausencia de ciertas variables se abordarán más adelante, en el apartado de procesado y preparación de los datos.

3.3.1. Calidad de la señal eléctrica

Las instalaciones fotovoltaicas analizadas se encuentran conectadas a un sistema eléctrico trifásico, una configuración comúnmente empleada en entornos industriales o en instalaciones con alta demanda energética, como es el caso de los hospitales. Este tipo de sistema está compuesto por tres fases (denominadas como **L1**, **L2** y **L3**) y un conductor neutro (**N**), lo que permite una transmisión de energía más eficiente y equilibrada.

A diferencia de los sistemas monofásicos, que utilizan únicamente una fase y el neutro,

el sistema trifásico permite repartir la carga eléctrica entre tres conductores activos. Esto conlleva una mayor estabilidad en el suministro, una reducción en las caídas de tensión y la capacidad de alimentar equipos de alta potencia sin sobrecargar las líneas. Además, posibilita la generación de campos magnéticos rotativos, esenciales para el funcionamiento de motores eléctricos, sistemas de climatización y otros equipos habituales en instalaciones hospitalarias [2].

El conjunto de variables eléctricas registradas en los datos es extenso y recoge múltiples magnitudes relevantes para evaluar la calidad del suministro. Estas variables se pueden agrupar según:

- **La magnitud medida:** entre ellas se encuentran la tensión, la corriente, la potencia activa, el factor de potencia, la distorsión armónica total (THD), la frecuencia, los armónicos individuales y los índices de parpadeo (*flicker*).
- **El punto de medición:** la mayoría de las variables se registran por separado en cada una de las fases (L1, L2, L3), en el conductor neutro (N), o como diferencia entre una fase y el neutro (por ejemplo, L1N). También se encuentran medidas entre el neutro y tierra (NG).
- **El tipo de estadístico calculado:** para cada variable, se dispone de tres estadísticos básicos calculados en cada intervalo de 10 minutos: el valor mínimo, el máximo y la media. Esto permite no solo obtener una visión general del comportamiento medio del sistema, sino también identificar inestabilidades, picos o caídas que puedan afectar a la calidad eléctrica.

Por ejemplo, para la tensión, existen mediciones de *valor eficaz (RMS)*, *pico*, *fundamental* y *factor de cresta*, todas registradas tanto entre fase y neutro (L1N, L2N, L3N) como entre neutro y tierra (NG). De manera análoga, para la corriente se recogen datos en las tres fases y en el neutro, incluyendo componentes fundamentales y armónicos hasta el orden 13.

Las potencias (activa, aparente y reactiva), así como los factores de potencia (tanto clásico como desplazado), se miden principalmente entre cada fase y el neutro. También se encuentran indicadores de desequilibrio, que permiten analizar la simetría entre fases, y múltiples componentes armónicas que dan información sobre la distorsión de la señal.

Se incluyen además indicadores globales de calidad como el THD (Total Harmonic Distortion) para tensión y corriente, tanto en fases como en neutro, así como medidas de flicker a corto y largo plazo (*Pst*, *Plt* y *Pst1min*), especialmente relevantes en la evaluación de la estabilidad percibida del suministro eléctrico.

En conjunto, esta rica batería de variables proporciona una base sólida para caracterizar la señal eléctrica generada por cada instalación y estudiar su relación con las condiciones ambientales, con el fin de detectar posibles problemas de calidad que puedan afectar al rendimiento de los aparatos del hospital.

Si se desea, se puede consultar en el Anexo A un resumen estructurado de las variables eléctricas disponibles, agrupadas según magnitud, punto de medición y estadístico asociado para cada una de las instalaciones.

3.3.2. Ambiental

El conjunto de medidas ambientales se compone de las siguientes variables:

- **Temperatura:** temperatura ambiente medida en una estación meteorológica situada en las inmediaciones de las instalaciones fotovoltaicas. Se encuentra expresada en grados Celsius ($^{\circ}\text{C}$) y, dado que ambas instalaciones se encuentran en el mismo lugar, esta variable es común a ambas.
- **Irradiancia:** irradiancia solar medida en vatios por metro cuadrado (W/m^2). Representa la densidad de potencia solar que incide sobre la superficie de los paneles. En este caso, se dispone de dos series diferenciadas correspondientes a cada instalación, lo que permite capturar diferencias derivadas de su orientación.

Estas variables ambientales jugarán un papel clave en el análisis posterior, ya que constituirán el punto de partida como variables explicativas principales. Se investigará su posible relación con distintos indicadores de calidad de la señal eléctrica generada por los sistemas fotovoltaicos.

3.4. Procesado y preparación de datos

Este proceso comenzó a partir del primer conjunto recibido, correspondiente al mes de enero de 2025. Este bloque inicial permitió realizar una exploración inicial que ya evidenció varios aspectos importantes que se repetirían posteriormente en el resto de archivos del proyecto.

3.4.1. Problemas detectados en la exploración inicial

En primer lugar, se identificaron numerosas variables con varianza nula, es decir, constantes a lo largo de todo el periodo, lo que las hacía irrelevantes para el análisis. También se detectaron zonas con valores ausentes, particularmente en el conjunto del CT Nuevo, aunque en menor medida también en las variables ambientales. Asimismo, se observó que las variables disponibles no coincidían exactamente entre las dos instalaciones: había medidas presentes en uno de los bloques que no aparecían en el otro, y viceversa. Incluso entre las variables comunes, su comportamiento no siempre era paralelo: algunas eran constantes en una instalación, pero no en la otra, o tenían diferentes proporciones de valores ausentes.

Otro hallazgo importante fue que se identificó una descoordinación temporal entre las hojas de datos (eléctricas y ambientales), de modo que no siempre coincidían exactamente los instantes de medida al comienzo y final. Este desfase fue posteriormente corregido en la medida de lo posible durante la fase de sincronización temporal.

	CT Bombas	CT Nuevo
Total de variables registradas	1365	518
Variables con varianza nula	1146	7
Variables con varianza positiva	219	511
Variables comunes (en ambos conjuntos)		126
Variables comunes con varianza positiva		86

Tabla 3.2: Resumen de la estructura de variables tras la exploración inicial.

3.4.2. Reducción de dimensionalidad mediante análisis de correlaciones

Con el objetivo de reducir la dimensión del conjunto de datos y evitar redundancias, se realizó un análisis de correlaciones sobre las variables con varianza mayor que cero. Este análisis se centró inicialmente en las variables correspondientes a la instalación del CT Bombas, y posteriormente se confirmó el mismo patrón en el CT Nuevo.

En ambos casos, se observó que las variables correspondientes a las fases L2 y L3 presentaban una correlación extremadamente alta con las de la fase L1, superando sistemáticamente el umbral de 0.95, y en la mayoría de los casos incluso 0.99. Esta fuerte correlación se debe a la naturaleza equilibrada del sistema trifásico, lo que puede justificar su eliminación sin pérdida relevante de información.

En las Figuras 3.4a y 3.4b se muestra también la distribución del número de pares de variables según su nivel de correlación. Como puede observarse, existe un número considerable de relaciones con correlaciones superiores a 0.95 en valor absoluto, lo que refuerza la decisión tomada de eliminar por completo las variables correspondientes a las fases L2 y L3 para el análisis posterior.

En la parte inferior de cada figura se presenta un resumen de la cantidad de pares de variables que superan distintos umbrales de correlación: 0.5, 0.6, 0.7, 0.8, 0.9, 0.95 y 0.99, todos ellos considerados en valor absoluto. Esta visualización ayuda a cuantificar la redundancia presente en los datos.

Esta decisión se confirmó en reuniones mantenidas con el Departamento de Ingeniería Eléctrica, quienes comentaron que, al tratarse de un sistema trifásico equilibrado, resulta técnicamente válido trabajar únicamente con las medidas de una fase para extraer conclusiones generales. Por tanto, se optó definitivamente por mantener exclusivamente las variables asociadas a la fase L1 en aquellos casos en los que existían versiones equivalentes para L1, L2 y L3, eliminando las otras dos. El resto de variables no asociadas a fases se conservaron íntegramente.

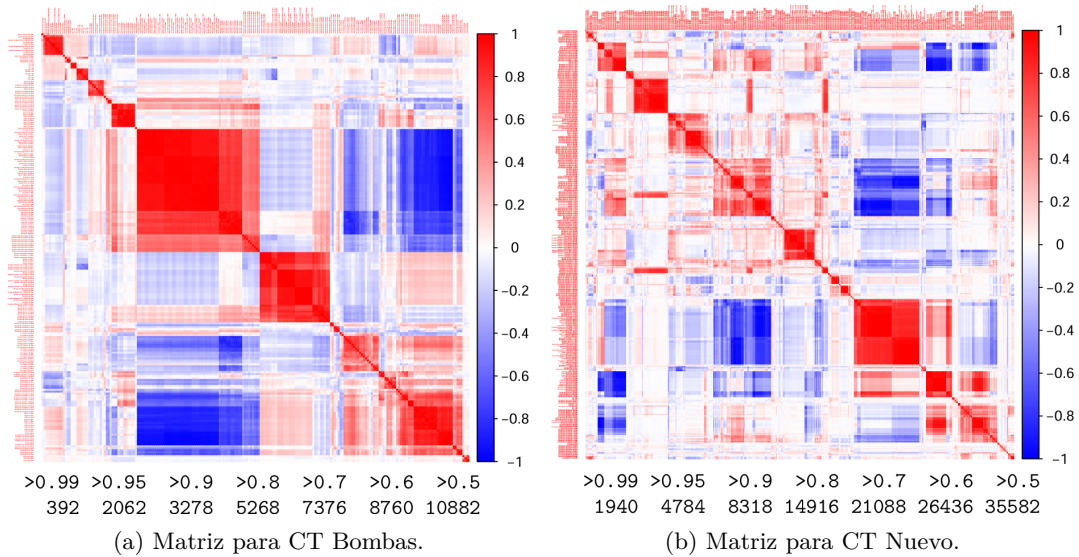


Figura 3.4: Matrices de correlaciones entre las variables con varianza mayor que 0 en los dos CT.

3.4.3. Tratamiento del conjunto total de datos

Una vez se dispone de todos los archivos correspondientes al período de estudio y tras haber eliminado las variables asociadas a las fases L2 y L3, se procede al tratamiento completo del total de datos disponibles.

Como se observa en la Tabla 3.1, los nombres originales de los archivos eran largos y poco prácticos. A partir de este punto, se adoptará una nomenclatura simplificada, como `d12_bombas` para referirse al archivo número 12 del CT Bombas o `d4_nuevo` para el cuarto del CT Nuevo.

Estudio inicial de las variables

El primer análisis conjunto reveló múltiples problemas relacionados con las variables de cada conjunto. En las Tablas 3.3 y 3.4 se presenta un resumen de las variables disponibles en cada archivo antes de aplicar ningún ajuste.

Durante esta etapa se detectó que la presencia de valores ausentes en algunos archivos se debía a errores en la denominación de las variables, derivados directamente del sistema de medición. En algunos casos, estos errores hacían que casi toda la columna apareciera como vacía, exceptuando las primeras 20-30 horas, o incluso que solo figurara el nombre sin ningún dato.

Dataset	Total	Var >0	Var = 0	Con NA
d1_bombas	304	55	35	214
d2_bombas	160	160	0	0
d3_bombas	166	158	8	0
d6_bombas	166	159	7	0
d7_bombas	166	159	7	0
d8_bombas	166	97	69	0
d9_bombas	166	96	70	0
d10_bombas	166	96	70	0
d12_bombas	166	96	70	0
d13_bombas	166	96	70	0
d14_bombas	166	95	71	0
d15_bombas	166	96	70	0
d16_bombas	166	96	70	0

Tabla 3.3: Resumen de variables del conjunto CT Bombas.

Dataset	Total	Var >0	Var = 0	Con NA
d1_nuevo	304	272	32	0
d4_nuevo	168	97	65	6
d5_nuevo	162	96	66	0
d6_nuevo	168	97	65	6
d7_nuevo	168	96	66	6
d8_nuevo	166	159	7	0
d9_nuevo	166	158	8	0
d10_nuevo	166	158	8	0
d11_nuevo	166	158	8	0
d12_nuevo	166	158	8	0
d13_nuevo	166	157	9	0
d14_nuevo	166	161	5	0
d16_nuevo	166	135	6	25

Tabla 3.4: Resumen de variables del conjunto CT Nuevo.

Este problema fue comentado con los profesores del Departamento de Ingeniería Eléctrica, quienes proporcionaron una versión corregida de los archivos, ya depurada. Sin embargo, esta versión no estaba completamente estandarizada. Comentaron la existencia de inconsistencias en la nomenclatura de ciertas variables, que podían aparecer con nombres distintos para una misma magnitud. A partir de las indicaciones de los ingenieros, se aplicaron las siguientes equivalencias para unificar correctamente las columnas:

- `Vrms.ph.n` → Tensión
- `VA.full.clásico` → `Potencia.Aparente`
- `VAR.Clás.` → `Potencia.Reactiva`
- `PF.Clásico` → `Factor.de.Potencia`
- `DPF.Clásico` → `Cos.Phi`

Una vez integradas las correcciones, el número de variables y la calidad de los datos mejoraron notablemente. Las Tablas 3.5 y 3.6 muestran el nuevo resumen de variables tras la depuración y el filtrado de constantes y errores.

Dataset	Total	Var >0	Var = 0	Con NA
d1_bombas	167	98	69	0
d2_bombas	160	160	0	0
d3_bombas	166	158	8	0
d6_bombas	166	159	7	0
d7_bombas	166	159	7	0
d8_bombas	167	98	69	0
d9_bombas	167	97	70	0
d10_bombas	167	97	70	0
d12_bombas	167	97	70	0
d13_bombas	167	97	70	0
d14_bombas	167	96	71	0
d15_bombas	167	97	70	0
d16_bombas	167	97	70	0

Tabla 3.5: Resumen de variables tras el filtrado en CT Bombas.

Dataset	Total	Var >0	Var = 0	Con NA
d1_nuevo	304	272	32	0
d4_nuevo	162	97	65	0
d5_nuevo	162	96	66	0
d6_nuevo	162	97	65	0
d7_nuevo	162	96	66	0
d8_nuevo	166	159	7	0
d9_nuevo	166	158	8	0
d10_nuevo	166	158	8	0
d11_nuevo	166	158	8	0
d12_nuevo	166	158	8	0
d13_nuevo	166	157	9	0
d14_nuevo	166	161	5	0
d16_nuevo	141	135	6	0

Tabla 3.6: Resumen de variables tras el filtrado en CT Nuevo.

Tratamiento de valores ausentes en datos ambientales

Aunque los conjuntos de calidad de la señal ya no presentan valores ausentes, los datos ambientales sí contienen algunos casos. Estos no son valores dispersos en distintas columnas, sino registros completos (filas enteras) ausentes.

En general, los registros incompletos en las variables ambientales son escasos, con entre 0 y 4 filas faltantes por archivo. Para mantener la continuidad temporal de la señal, estos valores ausentes se imputaron mediante interpolación por splines (véase apartado 2.1), una técnica que permite una reconstrucción suave y continua basada en los valores vecinos disponibles.

No obstante, se detectó un caso particular en el que la ausencia de datos se extendía durante casi 24 horas consecutivas. Dada la longitud del hueco, aplicar interpolación en este tramo habría generado una estimación poco fiable. Por ello, se optó por eliminar completamente dicho intervalo, tanto del conjunto de datos ambientales como de los eléctricos correspondientes (CT Bombas y CT Nuevo), garantizando así la coherencia temporal en los datos finales empleados para el análisis.

Sincronización temporal

Otra de las dificultades encontradas al trabajar con los distintos archivos fue la falta de coincidencia exacta en los instantes de muestreo entre las diferentes hojas (CT Bombas, CT Nuevo y ambiental). Aunque todos los bloques presentan una frecuencia de 10 minutos, las marcas temporales no están alineadas entre ellos. Por ejemplo, mientras que un archivo puede tener registros a las 12:00, 12:10, 12:20, etc., otro correspondiente al mismo periodo puede contener observaciones a las 11:08, 11:18, 11:28, y así sucesivamente.

Dado que no se puede modificar la base temporal original de los conjuntos eléctricos (es decir, no se realiza interpolación ni reajuste sobre ellos), el enfoque adoptado consiste en identificar, dentro de cada archivo, los registros que son más cercanos temporalmente entre bloques. Para cada observación en cada uno de los conjuntos eléctricos (CT Bombas y CT Nuevo), se busca la marca temporal más próxima en el conjunto ambiental, de tal manera que cada CT queda asociado a su propia versión de variables ambientales sincronizadas, pues los instantes de muestreo no son iguales entre las dos instalaciones. Este procedimiento permite maximizar la cobertura temporal conjunta sin introducir cambios artificiales en el alineamiento.

También se detectaron algunas anomalías específicas. En determinados archivos, la frecuencia de muestreo cambia de forma puntual a mitad del período, pasando de un ritmo regular de 10 minutos (por ejemplo, 14:00, 14:10, 14:20...) a un único salto irregular, como 14:33, para luego continuar nuevamente con intervalos regulares de 10 minutos (14:43, 14:53, etc.). Este comportamiento sugiere una interrupción momentánea en el sistema de medida, posiblemente provocada por fallos temporales o reinicios puntuales del mismo.

Otra complicación importante surge por la gestión del cambio horario. A lo largo del periodo analizado se producen dos transiciones entre horario estándar y horario de verano: una en octubre de 2024 y otra en marzo de 2025. Aunque todos los *timestamps* fueron

convertidos a UTC durante el procesamiento en R para mantener la coherencia temporal, se observaron comportamientos distintos en los archivos que abarcan estas fechas. En algunos casos, la transición horaria aparece reflejada de forma explícita como un salto temporal (por ejemplo, un registro a las 01:58 seguido directamente de otro a las 03:08). En otros, el cambio se anticipa una hora antes de lo esperado, y en algunos no queda reflejado en absoluto.

Esta inconsistencia en la representación del cambio de hora, limitada únicamente a los archivos que cubren esas fechas críticas, supone un riesgo de desalineación temporal si no se trata adecuadamente. Por ello, se realizó una validación manual de los *timestamps* en estos casos, aplicando los ajustes necesarios.

Consistencia de variables entre archivos

Antes de proceder a la unificación de los distintos bloques de datos, fue necesario analizar con mayor detalle la consistencia del conjunto de variables disponibles en los archivos. Si bien ya se había observado previamente que el número de variables no coincidía exactamente entre todos los conjuntos, se esperaba que estas diferencias fueran menores. Sin embargo, este análisis reveló una variabilidad estructural más amplia de lo previsto.

Para cada uno de los dos centros de transformación (CT Bombas y CT Nuevo), se realizó un estudio detallado de la presencia de variables comunes entre sus respectivos archivos. En el caso de CT Bombas, se encontraron 155 variables comunes presentes en todos los conjuntos, mientras que en CT Nuevo se identificaron 166 variables comunes.

Sin embargo, al comparar los nombres entre ambos conjuntos (una vez unificadas las nomenclaturas y eliminadas las duplicidades), el número final de variables coincidentes entre ambos CT se redujo a 135. Esto implica que hay variables que, si bien aparecen de forma consistente en uno de los centros, están ausentes en el otro, o directamente no están disponibles en todos los archivos de su propio bloque.

Por ejemplo, en CT Bombas los archivos `d2_bombas`, `d3_bombas` y `d6_bombas` presentaban hasta 12, 6 y 6 variables ausentes, respectivamente, relacionadas principalmente con componentes armónicas de orden bajo y factores de potencia. En CT Nuevo, el archivo `d16_nuevo` era el que presentaba más ausencias, con hasta 25 variables faltantes, incluyendo medidas fundamentales de tensión y corriente, así como factores de cresta.

Ante esta variabilidad estructural, y dado que el análisis posterior se realizará de forma independiente para cada uno de los centros de transformación, se decidió conservar las variables que están presentes en todos los archivos de cada CT por separado. Esto permite preservar la mayor cantidad posible de información dentro de cada instalación sin introducir huecos estructurales. En caso de que fuese necesario realizar un análisis conjunto entre ambas instalaciones, se utilizaría entonces el subconjunto de 134 variables comunes a ambos CT.

En resumen, el conjunto final de variables utilizadas para el análisis de cada centro está formado por 155 variables en CT Bombas y 134 variables en CT Nuevo. Para análisis comparativos entre centros, se dispone además de un subconjunto común de 134 variables eléctricas.

Estudio de incidencias

Tras la selección de variables comunes en cada centro, se llevó a cabo un estudio de incidencias una vez combinados todos los bloques por separado. Cabe recordar que, aunque ya se había observado previamente la existencia de variables con varianza nula en varios archivos individuales (como se mostró en las Tablas 3.3 y 3.4), hasta este punto no se había procedido a su tratamiento. Por tanto, esta etapa incluía también la identificación definitiva y el descarte de aquellas variables que seguían siendo constantes a lo largo de todo el conjunto combinado.

En primer lugar, se evaluó la varianza de cada variable a lo largo de todo el conjunto unificado. Aunque muchas variables ya se habían identificado como constantes en los bloques individuales, esta revisión global permitió observar algunos comportamientos interesantes. En particular, se identificaron variables que eran constantes en la mayoría de los archivos, pero que mostraban variabilidad únicamente en uno o dos de ellos, haciendo que ahora no lo fuesen. Este patrón sugiere que en realidad se trata de variables no fiables, que podrían haber sido mal registradas o activadas temporalmente de forma errónea.

Asimismo, se identificaron variables que, aunque formalmente eran numéricas, presentaban una variabilidad muy reducida. En particular, se analizaron aquellas variables que tomaban menos de 100 valores distintos a lo largo de todo el período disponible. Dado este comportamiento, se pudieron considerar candidatas tanto a ser excluidas del análisis como a considerarlas categóricas, analizándolas por tanto de manera diferente.

Con todos estos elementos, se elaboró un informe de incidencias que fue presentado y discutido con los responsables del Departamento de Ingeniería Eléctrica. Como resultado de esta evaluación conjunta, se adoptaron varias decisiones:

- Se constató que la mayoría de variables asociadas al neutro en el CT Bombas correspondían con las de baja variabilidad y un comportamiento errático o físicamente poco coherente. Estas variables representaban la mayoría de las candidatas a ser descartadas, por lo que se acordó suprimir completamente todas las variables del neutro en CT Bombas.
- Las que permanecían presentaban una distribución razonable de valores (todas por encima de 20 valores distintos), por lo que se optó por conservarlas sin transformaciones adicionales.
- Se eliminaron 14 variables adicionales con varianza cero, que seguían sin aportar información tras la combinación de bloques.
- Se decidió conservar aquellas variables que, si bien presentaban baja variabilidad, seguían estando correctamente definidas. Aunque no se espera que tengan un impacto relevante en el análisis, su posible utilidad en tareas supervisadas justificaba su mantenimiento inicial.

En cuanto a CT Nuevo, si bien no se identificaron variables constantes tras la combinación, sí se observaron casos similares a los descritos para Bombas: variables que eran constantes en

todos los bloques excepto en uno, donde presentaban cierto nivel de variación. Aunque este comportamiento indica una baja fiabilidad de dichas variables, no se procedió a su eliminación en esta fase. En su lugar, se decidió conservarlas inicialmente, bajo el supuesto de que su bajo rendimiento en los análisis las descartará de forma automática en las etapas posteriores. En este caso, además, las variables del neutro no mostraron ningún problema aparente, lo que explica en parte la diferencia final en el número de variables seleccionadas entre ambos centros.

Como resultado de este proceso de depuración final, el número de variables eléctricas seleccionadas quedó fijado en:

- 48 variables para CT Bombas.
- 134 variables para CT Nuevo (donde no se identificó ninguna variable con varianza cero tras la combinación).

Estos nuevos conjuntos servirán como base para el análisis del trabajo sobre la calidad de la señal y su relación con las variables ambientales.

Capítulo 4

Análisis de datos

Este capítulo presenta el análisis detallado de la calidad de la señal eléctrica registrada en cada una de las instalaciones, así como su posible relación con las condiciones ambientales. A partir del conjunto de datos ya depurado y estructurado descrito en el capítulo anterior, se aplican diversas técnicas estadísticas y de visualización orientadas a explicar el comportamiento de las variables eléctricas y explorar su dependencia respecto a factores externos relacionados con los sistemas fotovoltaicos.

Dado que la estructura y comportamiento de los dos centros de transformación presentan particularidades propias, el análisis se organiza de manera diferenciada para cada uno de ellos. Además, debido al gran número de variables eléctricas, se sigue una estrategia organizada basada en la aplicación homogénea de modelos sobre todas ellas. A partir de estos resultados, se construyen rankings que permiten identificar las variables más representativas, centrando así el estudio en las variables más importantes.

4.1. CT Bombas

4.1.1. Análisis exploratorio de correlaciones

Como primer paso del análisis se calcularon las correlaciones de todas las variables eléctricas disponibles frente a las principales variables ambientales: irradiancia y temperatura. De esta forma, se pueden identificar posibles variables eléctricas que puedan estar mayormente influenciadas. En la Tabla 4.1 y la Tabla 4.2 se presentan los diez coeficientes de correlación más altos (en valor absoluto) con respecto a irradiancia y temperatura, respectivamente.

Variable	Correlación con Irradiancia
Corriente.Fundamental.L1.Min	-0.1164
Corriente.L1.Min	-0.1161
Armónicos.Corriente13.L1.Med	0.0912
Armónicos.Corriente3.L1.Med	0.0756
Corriente.Fundamental.L1.Med	-0.0747
Corriente.L1.Med	-0.0745
Desequilibrio.Vn.Min	0.0629
Armónicos.Corriente7.L1.Med	0.0603
Frecuencia.Max	0.0473
Armónicos.Corriente11.L1.Min	0.0463

Tabla 4.1: Top 10 de variables con mayor correlación (absoluta) con la irradiancia.

Variable	Correlación con Temperatura
Armónicos.Corriente13.L1.Med	0.2083
Desequilibrio.Vn.Min	0.1806
Desequilibrio.Vz.Min	0.1567
Armónicos.Corriente11.L1.Min	0.1310
Armónicos.Corriente7.L1.Min	0.1214
Armónicos.Corriente7.L1.Max	0.1137
Corriente.Fundamental.L1.Min	-0.1101
Corriente.L1.Min	-0.1066
Armónicos.Corriente11.L1.Max	0.1041
Armónicos.Corriente5.L1.Max	0.1029

Tabla 4.2: Top 10 de variables con mayor correlación (absoluta) con la temperatura.

Se observa que, en general, los valores de correlación son relativamente bajos, lo que puede indicar que el impacto directo de las condiciones ambientales sobre las variables eléctricas no es especialmente alto cuando se considera el conjunto completo. No obstante, se aprecian algunas relaciones consistentes, como la presencia repetida de variables armónicas de corriente (como los órdenes 7, 11 y 13) o la corriente fundamental de la fase L1.

Se aprecia una mayor correlación con la temperatura que con la irradiancia, lo cual puede sugerir una influencia térmica más elevada sobre ciertos parámetros eléctricos, frente al efecto más leve y variable de la radiación solar.

Adicionalmente, la correlación directa entre las dos variables ambientales principales, irradiancia y temperatura, alcanza un valor de 0.7961, lo cual es esperable dado su vínculo físico, especialmente en un entorno como el del Hospital Clínico, donde las placas solares están instaladas en la azotea de un edificio alto con exposición solar directa. Este alto grado de asociación podría implicar cierta redundancia entre ambas variables, por lo que conviene tenerlo en cuenta.

4.1.2. Modelado lineal con regresión múltiple

Para cada una de las variables de calidad se ajustaron cuatro modelos lineales distintos. Estos modelos incluyeron dos versiones simples (irradiancia sola y temperatura sola como variable explicativa) y dos versiones más complejas: un modelo múltiple con ambos predictores y otro que, además, incorpora un término de interacción entre ellos.

Los resultados mostraron que, en la mayoría de los casos, los modelos múltiples ofrecían un mejor ajuste (mayor R^2) que los modelos simples, y que la inclusión del término de interacción permitía capturar efectos combinados relevantes entre temperatura e irradiancia.

Variable	R2Mult	pIrr	pTemp	R2MultTnt	pInt
Desequilibrio.Vz.Min	0.0698	0	0	0.0933	0
Armónicos.Corriente13.L1.Med	0.0585	0	0	0.0603	0
Desequilibrio.Vn.Min	0.0504	0	0	0.0504	0.5721
Armónicos.Corriente11.L1.Min	0.0263	0	0	0.0297	0
Armónicos.Corriente7.L1.Min	0.0273	0	0	0.0283	0
Armónicos.Corriente5.L1.Min	0.0243	0	0	0.0265	0
Armónicos.Corriente7.L1.Max	0.0250	0	0	0.0262	0
Armónicos.Corriente5.L1.Max	0.0236	0	0	0.0259	0
Armónicos.Corriente11.L1.Max	0.0187	0	0	0.0215	0
Desequilibrio.An.Min	0.0147	0	0	0.0179	0

Tabla 4.3: Top 10 variables con mayor capacidad explicativa según el modelo lineal múltiple con interacción.

La Tabla 4.3 presenta las diez variables eléctricas que obtuvieron mayor capacidad explicativa según el modelo múltiple con interacción. Se incluyen el coeficiente de determinación **ajustado** (R^2 para mayor simplicidad) para el modelo sin interacción (**R2Mult**) y con interacción (**R2MultInt**), así como los p -valores (indicados como **p.variable**) correspondientes a cada predictor individual y al término de interacción de los respectivos modelos.

Se observa, por ejemplo, que la variable **Desequilibrio.Vz.Min** presenta una mejora notable del ajuste al incluir la interacción (de 0.0698 a 0.0933 en R^2), con un p -valor muy bajo asociado al término de la interacción, lo que confirma su relevancia. Como excepción, se encuentra **Desequilibrio.Vn.Min**, que no mejora el ajuste, de ahí que se muestre un p -valor elevado (0.5721), indicando que la interacción no es significativa en este caso concreto.

4.1.3. Modelado no lineal con modelos aditivos generalizados (GAM)

Dado que los modelos lineales aplicados en la etapa anterior mostraron una capacidad explicativa bastante limitada en la mayoría de las variables, se planteó la posibilidad de que la relación entre las condiciones ambientales y las magnitudes eléctricas no fuese estrictamente lineal. Para capturar posibles patrones más complejos, se optó por emplear *modelos aditivos*

generalizados (GAM), que permiten incorporar relaciones no lineales mediante funciones suavizadas.

En concreto, se ajustaron dos modelos para cada variable de salida:

- Un modelo GAM aditivo, que incluye términos suavizados independientes para la irradiancia y la temperatura.
- Un modelo GAM con interacción no lineal, donde además se incorpora una superficie bivariada que representa la interacción conjunta de ambas variables explicativas.

La Tabla 4.4 muestra las diez variables con mejor desempeño según el modelo con interacción no lineal, ordenadas por el coeficiente de determinación ajustado (R^2). Se incluyen también la desviación explicada en % (Dev.) y el estadístico GCV (*Generalized Cross-Validation*) tanto para el modelo simple como para el modelo con interacción.

Variable	R^2	Dev.	GCV	R^2_{int}	Dev.int	GCV _{int}
Desequilibrio.Vn.Med	0.0021	0.25	3.8051	0.4837	48.43	1.9700
Desequilibrio.Vz.Med	0.0021	0.25	3.8031	0.4832	48.38	1.9714
Desequilibrio.Vz.Max	0.0021	0.24	3.8088	0.4824	48.30	1.9771
Frecuencia.Max	0.0035	0.37	0.0027	0.4007	40.14	0.0016
Desequilibrio.Vn.Max	0.0011	0.13	5.1310	0.3594	36.02	3.2936
Desequilibrio.Vz.Min	0.1501	15.06	0.0003	0.1965	19.74	0.0003
Corriente.Fundamental.L1.Min	0.1312	13.18	16060.87	0.1693	17.03	15364.73
Corriente.L1.Min	0.1280	12.85	15853.67	0.1662	16.71	15167.46
Corriente.Fundamental.L1.Med	0.1190	11.95	18484.94	0.1567	15.77	17702.81
Corriente.L1.Med	0.1162	11.67	18274.36	0.1539	15.49	17502.48

Tabla 4.4: Top 10 variables con mayor capacidad explicativa según el modelo GAM con interacción no lineal. Se muestran también las métricas correspondientes al modelo aditivo sin interacción.

Una observación importante es que los valores de R^2 ajustados correspondientes a los modelos sin interacción son prácticamente nulos en los primeros casos de la tabla, indicando que la variabilidad de estas variables no se explica apenas con efectos individuales de temperatura o irradiancia. Sin embargo, al introducir una interacción no lineal, el ajuste mejora de forma drástica, superando el 48 % de varianza explicada en los mejores casos. Esto sugiere que la estructura conjunta de ambas variables ambientales es clave para entender las variaciones de ciertas magnitudes eléctricas.

De hecho, al analizar la tabla ordenada por el R^2 del modelo sin interacción, se confirmó que las variables que allí destacaban eran precisamente aquellas que aquí también presentan valores intermedios (por ejemplo, las corrientes fundamentales en L1), pero ninguna superaba el 15 % de varianza explicada. Esto refuerza la idea de que las mejoras obtenidas en los modelos complejos son atribuibles casi exclusivamente a la incorporación de la interacción no lineal.

Por otro lado, el estadístico GCV, que actúa como medida de error predictivo penalizando la complejidad del modelo, también sugiere que los seis primeros modelos de la tabla (hasta **Desequilibrio.Vz.Min**) presentan mejoras relevantes en su capacidad de generalización. Una reducción notable del GCV en comparación con su versión sin interacción indica que no solo mejora el ajuste, sino también la estabilidad del modelo frente a sobreajustes.

En conjunto, estos resultados justifican la elección de modelos GAM con interacción para el análisis detallado de las variables más prometedoras, dado que son capaces de capturar patrones complejos que los enfoques lineales no detectan.

Desequilibrio.Vn.Med

Este fue el modelo con mejor rendimiento según el análisis exploratorio con modelos GAM con interacción no lineal. La variable **Desequilibrio.Vn.Med** presentó un R^2_{adj} de 0.484 y una desviación explicada del 48.4%, valores notablemente superiores a los alcanzados por modelos lineales simples o múltiples sin interacción.

El estadístico GCV fue de 1.97, uno de los más bajos del conjunto, lo que sugiere que el modelo no solo ajusta bien los datos, sino que también tiene buena capacidad de generalización sin llegar al sobreajuste.

En cuanto a los términos suaves, todos resultaron altamente significativos ($p < 0.001$), incluyendo la interacción no lineal entre irradiancia y temperatura, que aporta una complejidad considerable ($EDF > 15$), pero justificada por la ganancia explicativa observada.

A continuación, se presentan los gráficos que ilustran los efectos estimados:

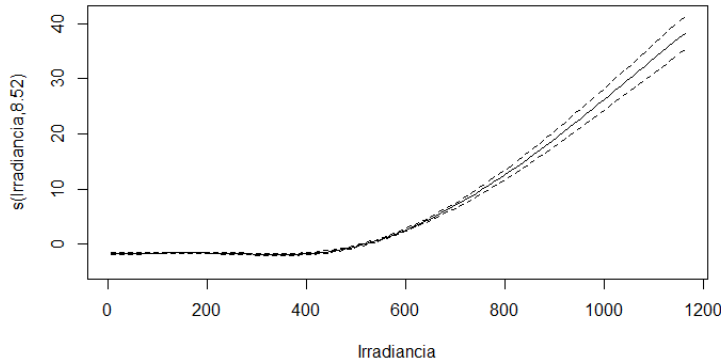


Figura 4.1: Efecto no lineal de la irradiancia.

La función suave correspondiente a la irradiancia muestra una respuesta claramente creciente a partir de valores en torno a 500–600 W/m², lo que indica que el desequilibrio en tensión tiende a aumentar bajo condiciones de alta exposición solar. En niveles bajos de irradiancia, el efecto estimado se mantiene cercano a cero, sugiriendo un umbral del fenómeno en los valores comentados.

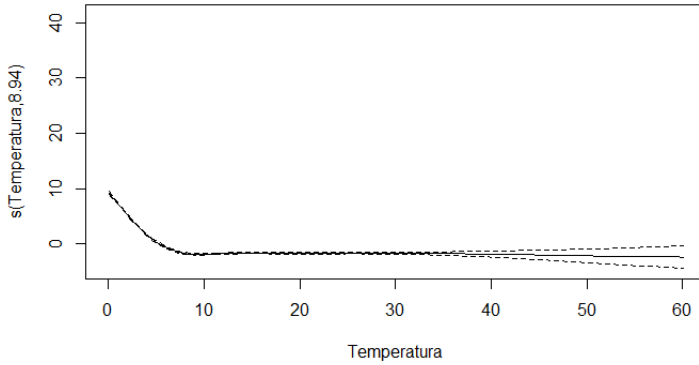


Figura 4.2: Efecto no lineal de la temperatura.

En el caso de la temperatura, se observa un comportamiento inverso: el efecto decrece rápidamente en valores bajos y se aplanan en torno a los 15–20 °C. Esta estabilización sugiere que la temperatura deja de tener un impacto significativo sobre el desequilibrio a partir de cierto umbral térmico.

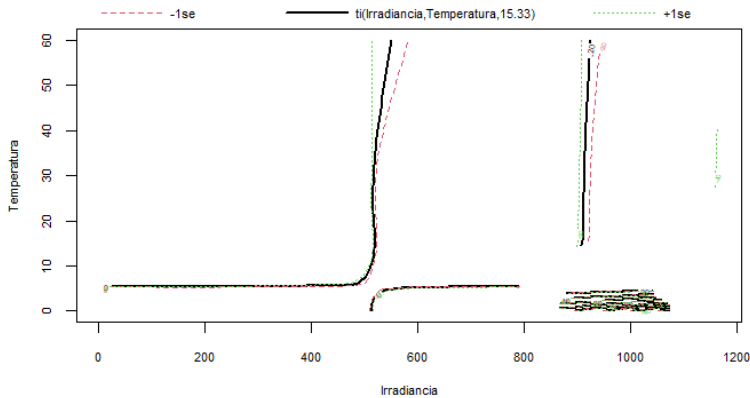


Figura 4.3: Interacción no lineal entre temperatura e irradiancia.

Por último, la superficie de interacción estimada mediante el término $ti(Irradiancia, Temperatura)$ revela un efecto conjunto complejo y no aditivo. Se identifican regiones en las que el impacto de la irradiancia sobre el desequilibrio se ve intensificado por temperaturas bajas, mientras que en condiciones de temperatura elevada el efecto marginal de la irradiancia disminuye. Este comportamiento justifica plenamente la necesidad de incluir una interacción no lineal en el modelo.

Para complementar la interpretación del modelo y ofrecer una visión más integrada del efecto conjunto de ambas variables explicativas, se presentan dos visualizaciones generadas a partir del predictor lineal total del modelo GAM.

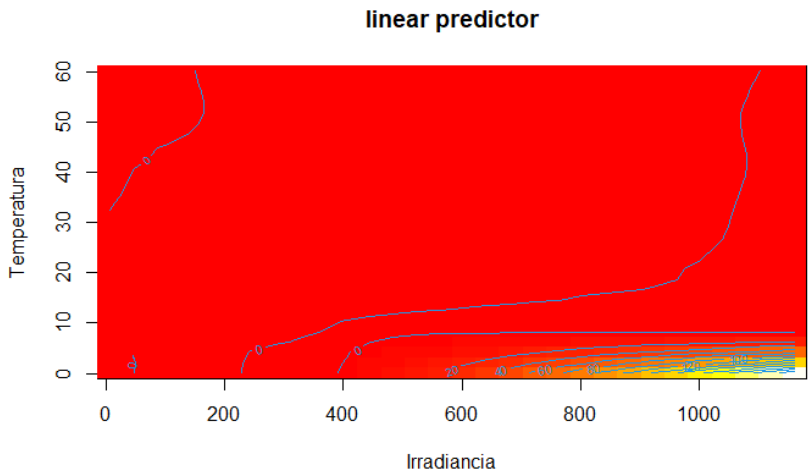


Figura 4.4: Contornos del predictor lineal del modelo GAM en el espacio irradiancia-temperatura.

La Figura 4.4 muestra un mapa de contornos que representa la contribución total del modelo (tanto de los efectos individuales como del término de interacción) en función de la irradiancia y la temperatura. Se aprecian zonas de respuesta diferenciadas, donde la variable de salida (desequilibrio) aumenta bruscamente a partir de ciertos umbrales ambientales, confirmando que los efectos de irradiancia y temperatura no actúan de forma independiente.

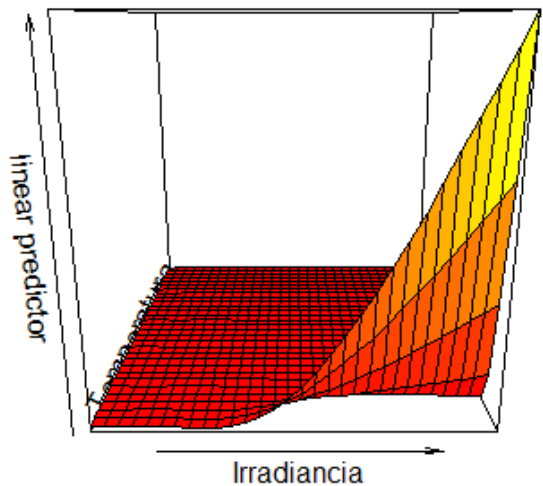


Figura 4.5: Superficie 3D del predictor lineal total estimado por el modelo GAM.

Por su parte, la Figura 4.5 representa la misma información en forma de superficie tridimensional, donde se evidencia la naturaleza no lineal del ajuste. En ella se observa claramente cómo el efecto del modelo se amplifica en regiones donde coinciden irradiancia alta y temperaturas bajas o moderadas, lo que refuerza las conclusiones anteriores respecto a la relevancia de considerar interacciones complejas en el modelado.

Estas visualizaciones, aunque no separan explícitamente el efecto de la interacción, sí permiten apreciar el comportamiento general del modelo en el dominio conjunto de las variables ambientales, aportando una perspectiva complementaria a la interpretación basada en los términos individuales del modelo.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.1.

Los siguientes cuatro modelos en el ranking de desempeño (correspondientes a las variables **Desequilibrio.Vz.Med**, **Desequilibrio.Vz.Max**, **Frecuencia.Max** y **Desequilibrio.Vn.Max**) presentan resultados visuales y estructurales casi idénticos al caso previamente analizado. Tanto las funciones suaves de irradiancia y temperatura como la superficie de interacción no lineal muestran patrones muy similares, lo cual sugiere que la misma dinámica conjunta entre variables ambientales es la principal responsable del comportamiento de estas magnitudes eléctricas.

Dado que las diferencias se encuentran únicamente en los valores absolutos de métricas como el R^2 ajustado, la desviación explicada o el GCV, y no en la forma funcional ni en la interpretación de los efectos, se omite su representación gráfica individual para evitar redundancias.

A continuación, se analiza de forma detallada el siguiente modelo que sí presenta un comportamiento diferenciado.

Desequilibrio.Vz.Min

Este modelo ocupa el sexto lugar en términos de rendimiento según el ranking basado en los modelos GAM con interacción no lineal. Aunque su capacidad explicativa es sensiblemente menor que la de los primeros modelos, sigue mostrando mejoras importantes frente a los modelos sin interacción.

En concreto, la variable **Desequilibrio.Vz.Min** presenta un R^2 ajustado de 0.197 y una desviación explicada del 19.7 %, lo que indica un nivel de ajuste moderado pero claramente significativo. El estadístico GCV es muy bajo (0.0003), lo que refuerza la idea de que el modelo es capaz de generalizar sin sobreajustar los datos.

Todos los términos suaves resultaron significativos ($p < 0.001$), incluyendo nuevamente la interacción no lineal entre irradiancia y temperatura, aunque con una complejidad algo menor ($EDF \approx 12.5$) respecto al caso anterior.

A continuación, se muestran los gráficos correspondientes al análisis de efectos individuales y de interacción:

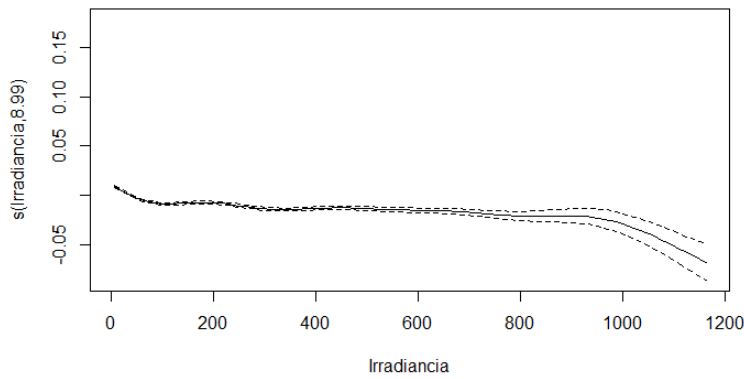


Figura 4.6: Efecto no lineal de la irradiancia.

En este caso, el efecto de la irradiancia se mantiene prácticamente plano en la mayor parte del rango observado, lo que indica que, por sí sola, no influye de forma significativa en el comportamiento de esta variable de desequilibrio. No obstante, a partir de valores cercanos a los 1000 W/m², se aprecia un descenso más acusado, lo que sugiere una posible reducción del desequilibrio bajo condiciones de irradiancia extrema.

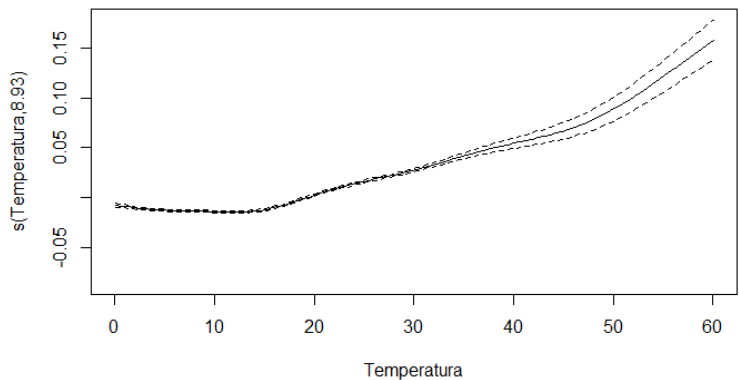


Figura 4.7: Efecto no lineal de la temperatura.

La temperatura, por el contrario, muestra un efecto creciente a partir de los 20–25 °C, indicando que el desequilibrio en tensión aumenta de forma significativa bajo condiciones térmicas elevadas. Esta relación no lineal parece tener mayor relevancia que la irradiancia por separado.

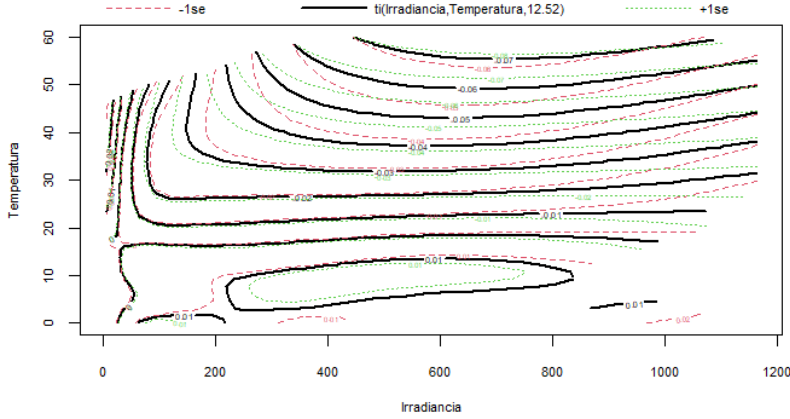


Figura 4.8: Interacción no lineal entre temperatura e irradiancia.

La superficie de interacción muestra un comportamiento estructurado con zonas de curvatura principalmente a bajas temperaturas (por debajo de 10 °C) y niveles de irradiancia bajos a moderados (hasta unos 400 W/m²), donde se aprecian algunas depresiones y elevaciones locales del valor estimado del predictor. Estas zonas sugieren la presencia de una interacción no lineal localizada bajo condiciones ambientales extremas (frías y con poca radiación solar), aunque de magnitud reducida.

En la mayor parte del espacio predictor (especialmente para temperaturas medias y altas) las curvas de nivel se mantienen bastante paralelas y espaciadas, lo que indica que el efecto conjunto de temperatura e irradiancia es más estable y menos relevante en esas condiciones. En conjunto, la interacción modelada no genera variaciones abruptas, sino ajustes suaves, lo que confirma que su aportación al modelo es moderada.

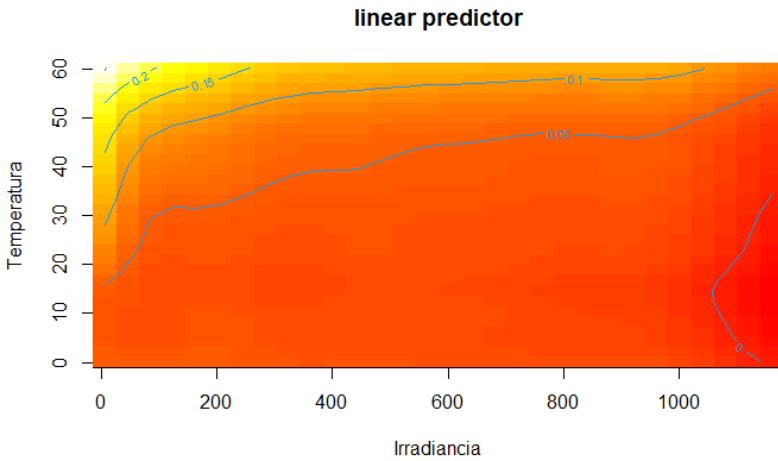


Figura 4.9: Representación de contornos del término de interacción no lineal entre irradiancia y temperatura.

Si se pasa ahora a las representaciones del predictor lineal, en la Figura 4.9, las curvas de nivel revelan zonas con leves incrementos en el valor del predictor lineal a medida que se combinan altas temperaturas con irradiancias medias, aunque la intensidad de este efecto es limitada.

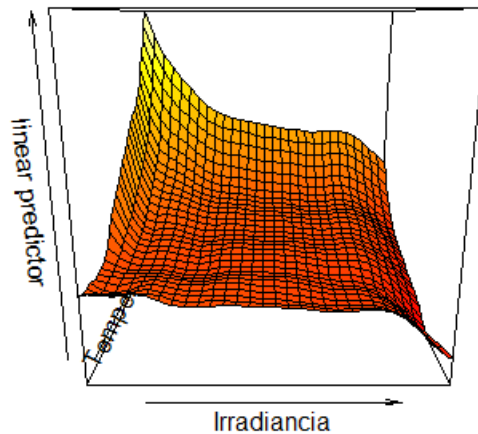


Figura 4.10: Superficie 3D del efecto combinado de irradiancia y temperatura en el predictor lineal.

Finalmente, la Figura 4.10 muestra la forma tridimensional de la superficie ajustada, en la que se observa una elevación progresiva pero moderada del efecto combinado, especialmente en presencia de temperaturas elevadas. Aparte de este comportamiento más progresivo, en comparación con los otros modelos, este desequilibrio parece aumentar de manera inversa, tomando sus valores más altos con altas temperaturas pero poca irradiancia.

Este modelo ilustra un patrón distinto al observado en los cinco primeros casos analizados, donde la interacción no lineal tenía un rol dominante. Aquí, la contribución principal proviene del efecto térmico individual, mientras que la interacción desempeña un papel más complementario.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.2.

4.2. CT Nuevo

4.2.1. Análisis exploratorio de correlaciones

En este caso, se vuelve a calcular la correlación entre todas las variables eléctricas disponibles y las variables ambientales clave: irradiancia y temperatura. A diferencia del CT Bombas, aquí sí se observan correlaciones mucho más elevadas, lo que indica una relación más fuerte entre condiciones ambientales y comportamiento eléctrico.

En las Tablas 4.5 y 4.6 se muestran los diez coeficientes de correlación más altos (en valor absoluto) para irradiancia y temperatura, respectivamente.

Variable	Correlación con Irradiancia
Desequilibrio.Vn.Min	-0.6201
Desequilibrio.Vn.Med	-0.5963
Armónicos.Tensión5.NG.Med	-0.3189
Armónicos.Tensión5.L1N.Med	0.2627
Armónicos.Tensión3.L1N.Med	-0.2567
Armónicos.Tensión5.L1N.Max	0.1913
Desequilibrio.Vz.Min	0.1811
Armónicos.Tensión5.L1N.Min	0.1807
THD.V.L1N.Med	0.1779
Potencia.Aparente.L1N.Min	0.1764

Tabla 4.5: Top 10 de variables con mayor correlación (absoluta) con la irradiancia en el CT Nuevo.

Variable	Correlación con Temperatura
Desequilibrio.Vn.Min	-0.6347
Desequilibrio.Vn.Med	-0.6201
Tensión.NG.Min	-0.4637
Tensión.de.Medio.Ciclo.V.RMS.NG.Med	-0.4500
Tensión.NG.Med	-0.4499
Corriente.L1.Min	0.4478
Tensión.de.Medio.Ciclo.V.RMS.NG.Min	-0.4425
Desequilibrio.Vz.Min	0.4409
Potencia.Activa.L1N.Min	0.4382
Potencia.Aparente.L1N.Min	0.4381

Tabla 4.6: Top 10 de variables con mayor correlación (absoluta) con la temperatura en el CT Nuevo.

En este transformador se observan correlaciones significativamente más elevadas que en el caso de CT Bombas. Destacan particularmente los indicadores de desequilibrio en tensión (*Desequilibrio.Vn.Min* y *Desequilibrio.Vn.Med*), que muestran correlaciones superiores al 60 % en valor absoluto tanto con irradiancia como con temperatura.

Asimismo, aparecen de forma consistente indicadores de calidad de tensión, como armónicos de voltaje y potencias, lo que sugiere una mayor sensibilidad de la instalación ante las condiciones ambientales.

Es especialmente relevante que la correlación entre irradiancia y temperatura alcanza aquí un valor de 0.8558, aún más alto que en el caso anterior. Esto refuerza la idea de que

ambas variables presentan una relación física fuerte y que su efecto conjunto podría estar amplificando el comportamiento eléctrico observado en el sistema.

En conjunto, estos resultados indican que el CT Nuevo tiene una mayor respuesta estructural ante cambios ambientales, lo cual motiva el análisis posterior mediante modelos multivariantes y no lineales.

4.2.2. Modelado lineal con regresión múltiple

Al igual que en el caso anterior, para cada una de las variables eléctricas se ajustaron cuatro modelos lineales: dos simples (con irradiancia o temperatura como único predictor) y dos más complejos, incluyendo un modelo múltiple con ambos predictores y otro que incorpora un término de interacción lineal entre ellos.

A diferencia del CT Bombas, donde los modelos lineales ofrecían un rendimiento muy limitado, en el CT Nuevo se observa una mejora considerable. En varias variables, el modelo múltiple con interacción alcanza valores de R^2 ajustados superiores al 40 %, lo que justifica su análisis detallado y su posterior comparación con modelos no lineales (GAM).

Variable	R2	pIrr	pTemp	R2Int	pInt
Desequilibrio.Vn.Min	0.4249	0	0	0.4663	0
Tensión.de.Medio.Ciclo.V.RMS.NG.Med	0.4576	0	0	0.4651	0
Tensión.NG.Med	0.4575	0	0	0.4649	0
Tensión.NG.Min	0.4582	0	0	0.4631	0
Corriente.L1.Min	0.3633	0	0	0.4371	0
Tensión.de.Pico.NG.Min	0.4304	0	0	0.4369	0
Desequilibrio.Vn.Med	0.4006	0	0	0.4325	0
Tensión.de.Medio.Ciclo.V.RMS.NG.Min	0.4257	0	0	0.4313	0
Tensión.de.Pico.NG.Med	0.4190	0	0	0.4294	0
Potencia.Activa.L1N.Min	0.3508	0	0	0.4272	0

Tabla 4.7: Top 10 variables con mayor capacidad explicativa según el modelo lineal múltiple con interacción (CT Nuevo).

Como puede verse en la Tabla 4.7, todas las variables del top presentan p -valores muy bajos tanto para los efectos individuales como para el término de interacción, lo que indica que los tres componentes del modelo (irradiancia, temperatura e interacción) contribuyen significativamente a explicar la variabilidad observada.

Los valores de R^2 del modelo con interacción ($R2_{Multint}$) superan el 46 % en varios casos, destacando, por ejemplo, *Desequilibrio.Vn.Min*, *Tensión.NG.Min* y *Tensión.NG.Med*. Esto contrasta notablemente con el caso del CT Bombas, donde los modelos lineales apenas alcanzaban valores del 8–10 %.

Este buen rendimiento de los modelos lineales sugiere que, al menos para este centro, la

relación entre las variables ambientales y ciertos indicadores eléctricos puede ser razonablemente aproximada mediante combinaciones lineales, incluyendo efectos cruzados. Aun así, se procederá posteriormente a un ajuste con modelos aditivos generalizados (GAM), que permitirán explorar posibles no linealidades y relaciones más complejas que puedan estar tapadas en la estructura lineal.

Desequilibrio.Vn.Min

En el caso del modelo lineal ajustado para la variable **Desequilibrio.Vn.Min**, se obtuvo un rendimiento destacado, con un coeficiente de determinación ajustado $R_{adj}^2 = 0,4663$, el mayor entre las variables analizadas en el CT Nuevo. El modelo incluye los efectos principales de irradiancia y temperatura, así como su interacción, siendo todos los términos estadísticamente significativos ($p < 2 \times 10^{-16}$). Los coeficientes estimados indican que, de forma individual, tanto la irradiancia como la temperatura se asocian a una reducción del desequilibrio de tensión, mientras que el término de interacción introduce un efecto ligeramente compensatorio en determinadas combinaciones de ambas.

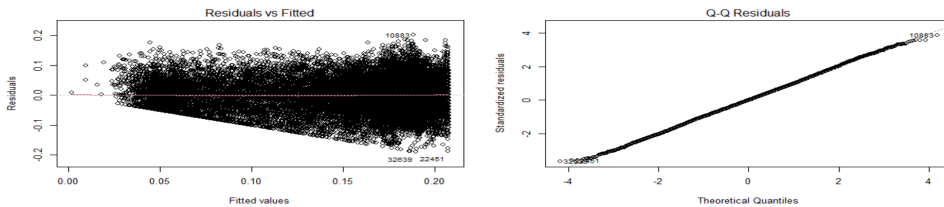


Figura 4.11: Gráficos de diagnóstico del modelo lineal para **Desequilibrio.Vn.Min**.

El análisis gráfico de los residuos revela una ligera heterocedasticidad en los valores más altos del ajuste. Aunque los tests formales no confirman la normalidad, la elevada cantidad de datos permite asumir razonablemente esta aproximación sin que afecte significativamente a la validez del modelo. Además, no se detectan observaciones influyentes destacadas. Las principales asunciones han sido verificadas mediante técnicas de remuestreo tipo *bootstrap*, lo que refuerza la robustez del modelo. Este procedimiento se ha aplicado de manera sistemática a todos los modelos lineales considerados en este estudio.

En conjunto, el modelo resulta adecuado para describir el comportamiento de la variable bajo condiciones ambientales variables, mostrando que la inclusión del término de interacción mejora sustancialmente el ajuste respecto a los efectos lineales simples.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.3.

Tensión.de.Medio.Ciclo.V.RMS.NG.Med

El modelo ajustado para la variable **Tensión.de.Medio.Ciclo.V.RMS.NG.Med** muestra un rendimiento prácticamente idéntico al del caso anterior, con un R_{adj}^2 de 0,4651. Los co-

eficientes para irradiancia, temperatura y su interacción son todos estadísticamente significativos ($p < 2 \times 10^{-16}$), aunque con signos contrarios a los observados en el primer modelo: la irradiancia tiene ahora un efecto positivo y la temperatura uno negativo, mientras que la interacción muestra un efecto suavemente negativo.

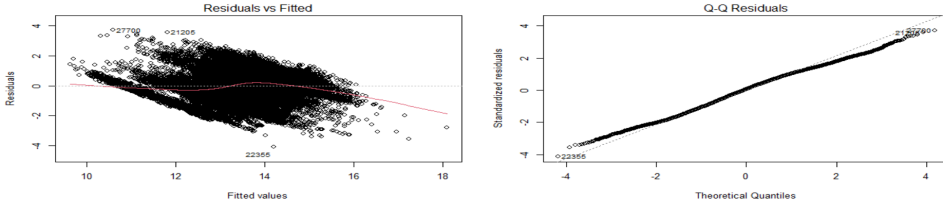


Figura 4.12: Gráficos de diagnóstico del modelo lineal para **Tensión.de.Medio.Ciclo.V.RMS.NG.Med.**

El análisis gráfico de residuos revela patrones similares a los del modelo anterior, con indicios de estructura posiblemente vinculada a la resolución del sensor y una ligera heterocedasticidad.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.4.

Tensión.NG.Med y Tensión.NG.Min

Los modelos ajustados para las variables **Tensión.NG.Med** y **Tensión.NG.Min** presentan gráficos de diagnóstico prácticamente idénticos a los del modelo anterior, así como valores similares de $R^2_{ajustado}$ (0,4649 y 0,4631 respectivamente). Los coeficientes son también altamente significativos y mantienen la estructura general de efectos cruzados entre temperatura e irradiancia.

Dado el elevado paralelismo entre estos casos, se omite la repetición del análisis gráfico detallado.

Corriente.L1.Min

En el caso de la variable **Corriente.L1.Min**, el modelo muestra un R^2_{adj} de 0,4371. A diferencia de los anteriores, los efectos individuales de irradiancia y temperatura, así como su interacción, tienen mayor magnitud, lo que sugiere una sensibilidad más acusada de esta variable a las condiciones ambientales. La irradiancia aparece con un efecto negativo, mientras que la temperatura y su interacción presentan signos positivos.

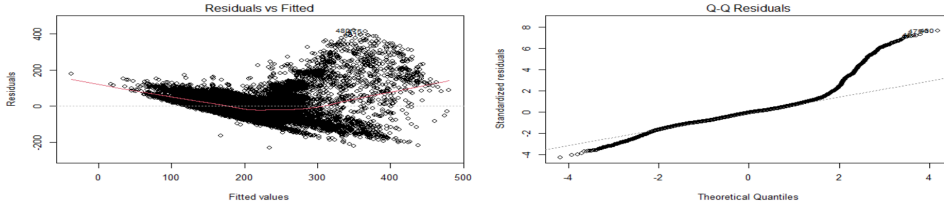


Figura 4.13: Gráficos de diagnóstico del modelo lineal para `Corriente.L1.Min`.

Desde el punto de vista gráfico, este modelo presenta desviaciones más marcadas respecto a los supuestos clásicos de la regresión. El gráfico *Residuals vs Fitted* muestra una clara estructura en los residuos, mientras que el Q-Q plot evidencia importantes desviaciones de la normalidad, especialmente en las colas. Asimismo, se aprecia una heterocedasticidad significativa, con una dispersión creciente de los residuos a medida que aumentan los valores ajustados.

En conjunto, estos indicios sugieren que el modelo podría no ser apropiado en su forma actual, independientemente de la gran cantidad de datos utilizados. Sería recomendable explorar alternativas para modelar esta variable, como modelos no paramétricos o transformaciones de la respuesta.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.5.

4.2.3. Modelado no lineal con modelos aditivos generalizados (GAM)

Tras analizar los resultados obtenidos con modelos lineales en la etapa previa, se constató que, si bien algunas variables mostraban cierta capacidad explicativa, muchas otras apenas eran capturadas por este tipo de relaciones. Ante esta limitación, se planteó la hipótesis de que los vínculos entre las condiciones ambientales (irradiancia y temperatura) y las variables eléctricas pudieran ser de naturaleza no lineal.

Con el objetivo de explorar estas posibles dependencias complejas, se emplearon *modelos aditivos generalizados (GAM)*, que permiten incorporar términos suavizados para cada variable explicativa y, opcionalmente, una interacción no paramétrica entre ellas.

Para cada variable de salida, se ajustaron dos variantes del modelo:

- Un modelo GAM aditivo, que incluye funciones suaves independientes para la irradiancia y la temperatura.
- Un modelo GAM con interacción no lineal, que incorpora además una superficie bivariable para modelar la interacción conjunta entre ambas variables.

La Tabla 4.8 presenta las diez variables con mejor rendimiento según el modelo con interacción, ordenadas por el coeficiente de determinación ajustado (R^2). Se incluyen también

la desviación explicada en porcentaje (Dev.) y el valor del estadístico de validación cruzada generalizada (GCV), tanto para el modelo aditivo como para el modelo con interacción.

Variable	R^2	Dev.	GCV	R^2_{int}	Dev.int	GCV _{int}
Corriente.L1.Min	0.5295	52.98	2497.22	0.6575	65.78	1818.97
Potencia.Activa.L1N.Min	0.5221	52.23	1.46e08	0.6517	65.20	1.06e08
Potencia.Aparente.L1N.Min	0.5146	51.49	1.44e08	0.6506	65.10	1.04e08
Corriente.L1.Med	0.5001	50.03	2897.49	0.6373	63.77	2102.98
Potencia.Aparente.L1N.Med	0.4879	48.82	1.66e08	0.6317	63.20	1.20e08
Potencia.Activa.L1N.Med	0.4929	49.32	1.69e08	0.6306	63.10	1.23e08
Desequilibrio.Vz.Min	0.4926	49.29	0.0002	0.6184	61.88	0.0001
Desequilibrio.Vz.Med	0.4870	48.72	0.0002	0.6099	61.03	0.0001
Tensión.de.Medio.Ciclo.V.RMS.NG.Med	0.5056	50.58	0.913	0.5213	52.17	0.884
Tensión.NG.Med	0.5053	50.55	0.915	0.5210	52.14	0.886

Tabla 4.8: Top 10 variables con mayor capacidad explicativa según el modelo GAM con interacción no lineal. Se incluyen también las métricas del modelo aditivo sin interacción.

A diferencia del caso anterior (modelo lineal), aquí se observan valores de R^2 significativamente elevados desde el primer modelo, incluso en su versión aditiva. No obstante, la inclusión de la interacción no lineal mejora sistemáticamente el ajuste en todos los casos, con incrementos notables en el porcentaje de desviación explicada y reducciones en el error estimado por GCV.

La diferencia más marcada se da en los primeros seis modelos, donde el ajuste con interacción supera el 63 % de varianza explicada, indicando que la relación entre las condiciones ambientales y las variables eléctricas es altamente dependiente del contexto conjunto de temperatura e irradiancia.

Asimismo, el estadístico GCV se reduce de forma consistente, lo que respalda la robustez de los modelos sin incurrir en sobreajuste. Estos resultados avalan la elección de modelos GAM con interacción para analizar con mayor detalle las variables de mayor interés.

Corriente.L1.Min

Este modelo fue uno de los que presentó mejor rendimiento en el conjunto de modelos GAM con interacción no lineal aplicados a los datos del CT Nuevo. La variable **Corriente.L1.Min** alcanzó un R^2_{adj} de 0.658 y una desviación explicada del 65.8 %, lo que indica un ajuste robusto y claramente superior a los obtenidos mediante modelos lineales.

El estadístico de validación cruzada generalizada (GCV) fue de 1818.97, lo que sugiere una adecuada capacidad de generalización del modelo sin evidencia de sobreajuste si se compara con su versión sin interacción. Todos los términos suaves fueron altamente significativos ($p < 0,001$), incluyendo la interacción no lineal entre irradiancia y temperatura, que presentó un grado efectivo de libertad (EDF) cercano a 16, reflejando una relación compleja pero justificada por el rendimiento del modelo.

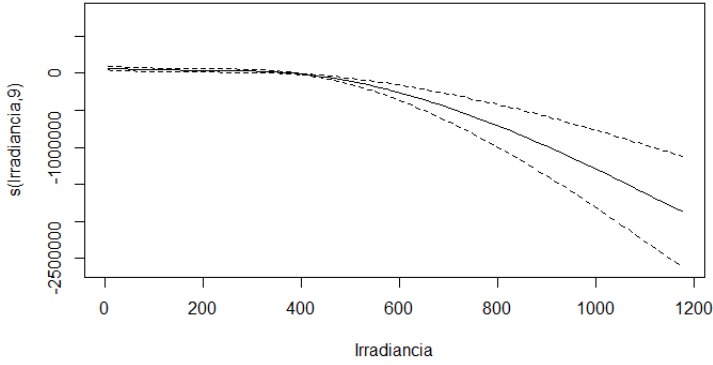


Figura 4.14: Efecto no lineal de la irradiancia.

La función suave correspondiente a la irradiancia muestra un patrón decreciente a lo largo del rango observado, especialmente a partir de los 400–500 W/m^2 , lo que indica que el aumento de irradiancia tiende a reducir los valores estimados de potencia aparente, con una curvatura notable en las zonas de mayor intensidad solar.

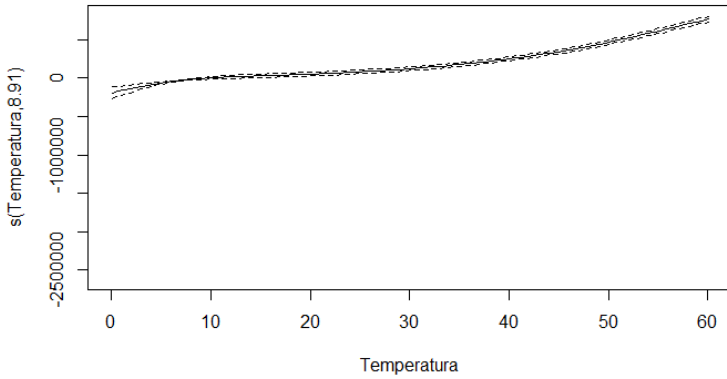


Figura 4.15: Efecto no lineal de la temperatura.

El efecto de la temperatura muestra una tendencia creciente a medida que se superan los 20–25 $^{\circ}C$, indicando que temperaturas elevadas tienden a aumentar la potencia aparente medida. En los valores más bajos, el efecto es prácticamente nulo o incluso levemente negativo.

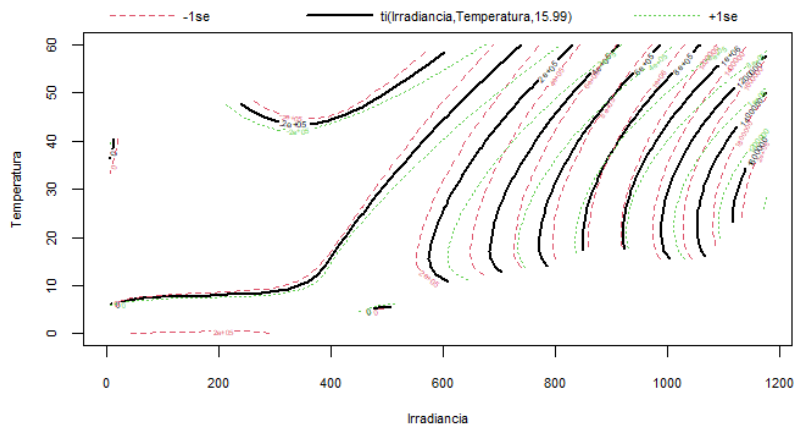


Figura 4.16: Interacción no lineal entre temperatura e irradiancia.

La interacción no lineal revela un comportamiento estructurado en el que se observan combinaciones específicas de irradiancia media y temperatura alta que maximizan el efecto estimado del modelo. En cambio, cuando ambas variables se encuentran en niveles bajos, el efecto conjunto tiende a disminuir, mostrando regiones de mínimo relativo.

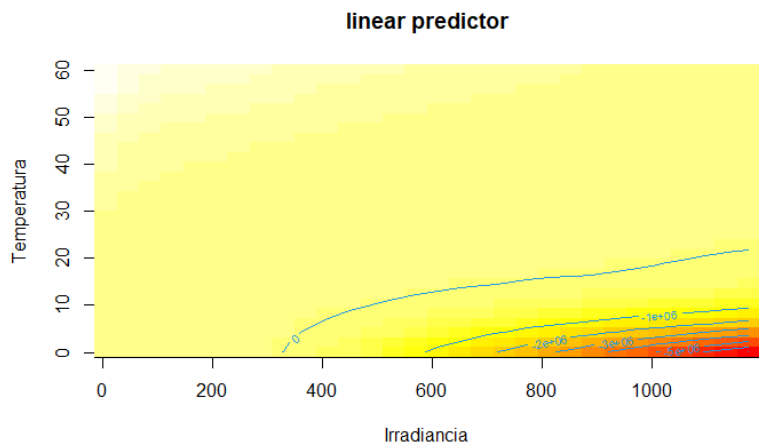


Figura 4.17: Contornos del predictor lineal del modelo GAM.

El mapa de contornos del predictor lineal refuerza la interpretación anterior, mostrando cómo los mayores valores estimados se asocian con temperaturas elevadas en combinación con irradiancia moderada o baja, mientras que el modelo predice valores más bajos bajo irradiancia extrema.

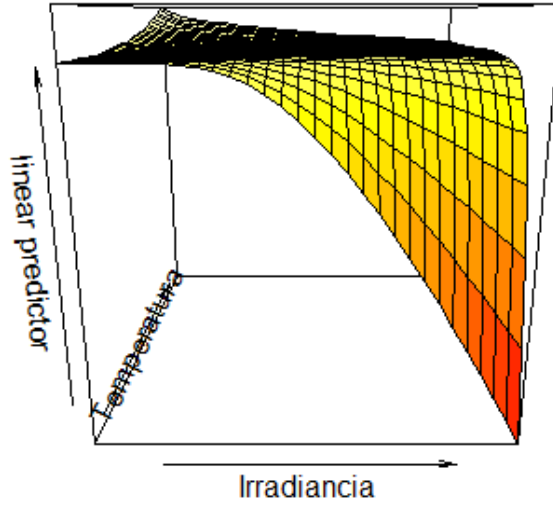


Figura 4.18: Superficie 3D del predictor lineal total estimado por el modelo.

La superficie tridimensional del predictor total del modelo permite visualizar con claridad la respuesta no lineal del sistema: el efecto combinado se acentúa en zonas específicas del espacio predictor, lo que pone de manifiesto la importancia de incluir interacciones no lineales para comprender adecuadamente el comportamiento de esta variable eléctrica.

Para consultar los resultados estadísticos completos del modelo, véase el Apéndice B.6.

Los siguientes cinco modelos con mejor rendimiento (*Potencia.Activa.L1N.Min*, *Potencia.Aparente.L1N.Min*, *Corriente.L1.Med*, *Potencia.Aparente.L1N.Med* y *Potencia.Activa.L1N.Med*) presentan resultados métricos algo distintos, pero sus representaciones gráficas son prácticamente idénticas a las del modelo anterior. Las funciones suaves de irradiancia y temperatura muestran las mismas tendencias generales, y las superficies de interacción no lineal reflejan patrones equivalentes en cuanto a curvatura, regiones de máximo efecto y comportamiento combinado.

Dado que la interpretación estructural no varía significativamente, se omite su análisis gráfico individual, remitiéndose al modelo previamente detallado como representativo de este grupo.

Capítulo 5

Conclusiones y trabajo futuro

A lo largo de este Trabajo de Fin de Grado se ha llevado a cabo un análisis estadístico exhaustivo de distintas magnitudes eléctricas medidas en dos centros de transformación, con el objetivo de estudiar su relación con dos variables ambientales clave: la irradiancia solar y la temperatura. Este enfoque ha permitido aplicar numerosas técnicas y metodologías propias de la estadística aplicada, abarcando desde modelos lineales clásicos hasta enfoques más avanzados como los modelos aditivos generalizados (GAM).

Desde el punto de vista técnico, el trabajo ha puesto de manifiesto la complejidad del análisis de sistemas eléctricos en entornos reales, caracterizados por una elevada dimensionalidad, presencia de ruido y patrones no lineales. En particular, se ha trabajado con un conjunto amplio de variables respuesta, muchas de las cuales presentan relaciones débiles o inexistentes con las variables ambientales consideradas. No obstante, también se han identificado múltiples casos en los que sí se logra capturar una parte significativa de la variabilidad mediante modelos adecuados, especialmente cuando se incorporan interacciones no lineales.

Se han identificado diferencias notables entre ambos centros. Los modelos ajustados en el CT Nuevo han mostrado una capacidad notablemente mayor que los propios de CT Bombas. Esto puede deberse no solo a la antigüedad de la instalación, sino también a la propia calidad en la recolección de los datos, ya que una gran cantidad de variables no han podido ser analizadas en este último. Este estudio, por tanto, permite determinar qué variables asociadas a la calidad de la señal eléctrica generada mediante energía fotovoltaica están más o menos influenciadas por las condiciones meteorológicas bajo las cuales dicha energía ha sido producida.

En algunos casos, como en las variables relacionadas con la corriente o la potencia, los modelos del CT Nuevo han alcanzado coeficientes de determinación ajustados (R^2_{adj}) superiores al 65 %, lo que permite interpretar sus resultados con un alto grado de fiabilidad.

El uso de modelos GAM ha resultado especialmente valioso para capturar relaciones no lineales complejas, particularmente en el CT Nuevo. En varias variables, la introducción de una interacción no lineal entre irradiancia y temperatura ha permitido mejorar sustan-

cialmente la calidad del ajuste, en comparación con los modelos lineales tradicionales. Este patrón se ha observado de forma sistemática, lo que confirma que los efectos ambientales no actúan de manera independiente, sino que su combinación genera dinámicas relevantes que sólo pueden detectarse mediante enfoques flexibles.

Una de las primeras conclusiones que se saca de este trabajo es la necesidad de realizar un proceso riguroso de depuración de datos y estandarización de variables en el análisis de calidad de señal eléctrica en instalaciones de generación fotovoltaica. El tratamiento de datos reales en este contexto ha puesto de manifiesto la presencia frecuente de registros erróneos, formatos inconsistentes y escalas heterogéneas, que pueden comprometer seriamente la fiabilidad de los modelos si no se abordan adecuadamente. Además, el desarrollo del trabajo ha permitido consolidar competencias clave en estas tareas, así como en la aplicación de procedimientos de validación estadística, incluyendo el análisis de residuos, la detección de heterocedasticidad y la evaluación mediante técnicas de remuestreo como el bootstrap.

En conjunto, este proyecto ha permitido no solo poner en práctica herramientas estadísticas avanzadas, sino también desarrollar una capacidad crítica frente a los resultados, diferenciando cuándo un modelo es realmente útil y cuándo sus conclusiones pueden resultar engañosas. Estas capacidades son de gran importancia en el contexto del análisis de datos energéticos, donde las decisiones se toman en base a modelos que deben ser interpretables.

5.1. Líneas de trabajo futuras

- **Incorporación de la variable meses:** Aunque si se ha trabajado con esta variable explicativa, la extensión del documento no ha permitido incluirla en el análisis final. Sin embargo, sus resultados han sido muy prometedores, especialmente en el CT Bombas, donde su inclusión en los modelos lineales generaba mejoras muy significativas. En general, la variable **meses** potencia el rendimiento de todos los modelos, lineales y no lineales, y debería ser considerada como predictor clave en estudios futuros.
- **Análisis multivariable con más predictores:** Ampliar el conjunto de variables explicativas, incorporando otras medidas ambientales (humedad, viento, nubosidad) o de carga, podría mejorar notablemente la capacidad explicativa de los modelos, especialmente en el CT Bombas, donde los actuales resultados han sido más limitados.
- **Aplicación de métodos basados en árboles:** En lugar de modelos lineales o GAM, se podrían explorar modelos basados en árboles como los random forests o los boosted trees, que permiten capturar interacciones y no linealidades de forma automática sin necesidad de especificarlas explícitamente.
- **Ampliación a otros centros:** Finalmente, sería interesante replicar el estudio en otros centros con características intermedias o diferentes, lo que permitiría validar si las conclusiones extraídas aquí son generalizables o dependen del tipo de instalación.

En definitiva, este trabajo constituye un punto de partida para el análisis de datos energéticos en instalaciones industriales de generación renovable, abriendo múltiples posibilidades para su aplicación tanto en investigación como en contextos prácticos.

Bibliografía

- [1] Omar F Al-Rawi, Yusuf Bicer, and Sami G Al-Ghamdi. Sustainable solutions for health-care facilities: examining the viability of solar energy systems. *Frontiers in Energy Research*, 11:1220293, 2023.
- [2] Stephen J. Chapman. *Electric Machinery Fundamentals*. McGraw-Hill Education, 5th edition, 2012.
- [3] John D. Cook. Solving tridiagonal systems, 2018. Último acceso: 10 de julio de 2025.
- [4] Carl de Boor. *A Practical Guide to Splines*. Springer, revised edition edition, 2001.
- [5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.
- [6] Trevor J Hastie and Robert J Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–318, 1986.
- [7] David H. House. Cubic splines lecture notes, 2006. Lecture notes, Clemson University.
- [8] International Energy Agency. Renewables 2024 – executive summary. <https://www.iea.org/reports/renewables-2024/executive-summary>, 2024.
- [9] Michael H. Kutner, Christopher J. Nachtsheim, John Neter, and William Li. *Applied Linear Statistical Models*. McGraw-Hill/Irwin, 5th edition, 2004.
- [10] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to Linear Regression Analysis*. Wiley, 5th edition, 2012.
- [11] I. Made Suartika. The power quality at an electrical power station of the hospital. Available on ResearchGate, 2019. https://www.researchgate.net/publication/333496803_The_Power_Quality_at_an_Electrical_Power_Station_of_the_Hospital.
- [12] Pilar Sánchez Molina. Solar becomes spain’s top power source in 2024. *pv Magazine*, 2025. <https://www.pv-magazine.com/2025/03/26/solar-becomes-spains-top-power-source-in-2024/>.
- [13] R Core Team. lm function - r documentation, 2020.
- [14] Rick Wicklin. Cubic spline interpolation: How to use splines to fill in missing values, 2020. Publicado en el blog The DO Loop de SAS Institute Inc.

- [15] Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B*, 73(1):3–36, 2011.
- [16] Simon N Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2017.
- [17] Simon N. Wood. *gam: Generalized Additive Models*. R Foundation for Statistical Computing, Vienna, Austria, 2024. R package version 1.9-3.
- [18] Achim Zeileis and Gabor Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005.

Apéndice A

Descripción de variables

A continuación, se presenta un listado detallado de las variables eléctricas registradas en cada uno de los dos centros de transformación fotovoltaicos analizados en este estudio: CT Bombas y CT Nuevo. Las mediciones se realizan cada 10 minutos, y los sufijos **.Min**, **.Med** y **.Max** indican, respectivamente, el valor mínimo, medio o máximo observado dentro de cada intervalo temporal.

Centro de Transformación Bombas

- **Fecha:** Fecha y hora exacta del registro, clave para la integración temporal de datos.
- **Corriente.L1.Min/Med/Max:** Corriente total en la fase L1. Refleja el flujo de corriente eléctrica, con sus valores extremos y promedio durante cada intervalo.
- **Corriente.Fundamental.L1.Min/Med/Max:** Componente de la corriente en la frecuencia fundamental (50 Hz). Su análisis permite identificar alteraciones en la forma de onda base.
- **Frecuencia.Med / Frecuencia.Max:** Medidas relacionadas con la frecuencia del sistema. La estabilidad de la frecuencia es esencial para un funcionamiento seguro de los equipos eléctricos.
- **Desequilibrio.Vn/Vz/An/Az.Min/Med/Max:** Medidas de desequilibrio de tensión en diferentes componentes (Vn: neutro, Vz: secuencia cero, An/Az: componentes angulares), que indican asimetrías perjudiciales en el sistema trifásico.
- **THD.A.L1.Min/Med/Max:** Distorsión armónica total de la corriente en L1. Cuantifica la presencia de armónicos y la calidad de la señal.
- **Armónicos.CorrienteX.L1.Min/Med/Max:** Magnitud del armónico de orden X (donde X = 2,3,5,7,9,11,13) en la corriente de L1. Útiles para análisis de compatibilidad electromagnética y detección de distorsiones específicas.

- **Pst..L1N / Plt..L1N / Pst1min..L1N**: Medidas de parpadeo lumínico (flicker) a corto (Pst), largo plazo (Plt), y en ventana de 1 minuto (Pst1min). Evalúan el impacto de fluctuaciones de tensión sobre la percepción humana.

Centro de Transformación Nuevo

- **Fecha**: Marca temporal del registro, esencial para la comparación sincronizada entre variables.
- **Tensión.L1N / NG.Min/Med/Max**: Tensión entre L1 y neutro o entre neutro y tierra (NG). Indicadores clave para evaluar desviaciones de tensión en la red.
- **Tensión.de.Medio.Ciclo.V.RMS.L1N/NG.Min/Med/Max**: Valor eficaz (RMS) de la tensión calculado en medio ciclo de onda, lo que proporciona una medida más precisa de fluctuaciones rápidas.
- **Tensión.de.Pico.L1N/NG.Min/Med/Max**: Valor de tensión instantánea máxima alcanzada en cada intervalo. Puede ser indicativa de sobretensiones transitorias.
- **Corriente.L1 / N.Min/Med/Max**: Corriente medida en la fase L1 o en el conductor neutro. Las diferencias entre ambas pueden revelar desequilibrios de carga.
- **Desequilibrio.Vn/Vz/An/Az.Min/Med/Max**: Mismos indicadores que en CT Bombas, permiten comparar el grado de asimetría entre centros.
- **Potencia.Activa / Aparente / Reactiva.L1N.Min/Med/Max**: Medidas clave del flujo energético. La potencia activa representa energía útil, la reactiva indica almacenamiento temporal de energía, y la aparente es la combinación vectorial de ambas.
- **Factor.de.Potencia.L1N.Min/Med/Max**: Mide la eficiencia con que se utiliza la energía. Valores bajos indican un exceso de potencia reactiva.
- **Cos.Phi.L1N.Med / Max**: Coseno del ángulo de desfase entre corriente y tensión. Relacionado directamente con el factor de potencia.
- **THD.V.L1N / NG.Min/Med/Max**: Distorsión armónica total en la tensión, crucial para evaluar la calidad de suministro.
- **Armónicos.TensiónX.Y.Min/Med/Max**: Amplitud de armónicos específicos (2º, 3º, 5º, etc.) medidos entre distintas fases o respecto al neutro/tierra. Permiten evaluar impactos de cargas no lineales.
- **THD.A.L1 / N.Min/Med/Max**: Distorsión armónica en corriente, extendida aquí también al neutro.
- **Armónicos.CorrienteX.L1/N.Min/Med/Max**: Amplitudes de armónicos de corriente, tanto en L1 como en neutro. Proporcionan diagnóstico detallado de fuentes de distorsión.
- **Pst..L1N / Plt..L1N / Pst1min..L1N**: Medidas de parpadeo lumínico (flicker) a corto (Pst), largo plazo (Plt), y en ventana de 1 minuto (Pst1min). Evalúan el impacto de fluctuaciones de tensión sobre la percepción humana.

Apéndice B

Resultados ampliados

En este apéndice se presentan los resultados ampliados de los modelos específicos analizados.

B.1. Modelo GAM Desequilibrio.Vn.Med en CT Bombas

```
----- RESUMEN DEL MODELO -----

Family: gaussian
Link function: identity

Formula:
Desequilibrio.Vn.Med ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia,
    Temperatura)

Parametric coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.78603    0.07472   23.9  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
              edf Ref.df      F p-value
s(Irradiancia)    8.524  8.926  104.3 <2e-16 ***
s(Temperatura)    8.943  8.997 1403.0 <2e-16 ***
ti(Irradiancia,Temperatura) 15.329 15.614 1684.2 <2e-16 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.484 Deviance explained = 48.4%

GCV = 1.97 Scale est. = 1.9677 n = 28174

----- VALIDACION DEL MODELO (gam.check) -----

Method: GCV Optimizer: magic

Smoothing parameter selection converged after 18 iterations.

The RMS GCV score gradient at convergence was 1.313668e-06 .

The Hessian was positive definite.

Model rank = 35 / 35

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

	k'	edf	k-index	p-value
s(Irradiancia)	9.00	8.52	1.0	0.095 .
s(Temperatura)	9.00	8.94	0.5	<2e-16 ***
ti(Irradiancia,Temperatura)	16.00	15.33	0.5	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

NULL

----- ANOVA DEL MODELO -----

Family: gaussian

Link function: identity

Formula:

Desequilibrio.Vn.Med ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia, Temperatura)

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Irradiancia)	8.524	8.926	104.3	<2e-16
s(Temperatura)	8.943	8.997	1403.0	<2e-16
ti(Irradiancia,Temperatura)	15.329	15.614	1684.2	<2e-16

B.2. Modelo GAM Desequilibrio.Vz.Min en CT Bombas

----- RESUMEN DEL MODELO -----

Family: gaussian
Link function: identity

Formula:
Desequilibrio.Vz.Min ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia, Temperatura)

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.0518687	0.0005517	94.02	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Irradiancia)	8.992	9.000	83.72	<2e-16 ***
s(Temperatura)	8.930	8.995	241.20	<2e-16 ***
ti(Irradiancia,Temperatura)	12.518	13.529	115.97	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.197 Deviance explained = 19.7%
GCV = 0.00029921 Scale est. = 0.00029887 n = 28174

----- VALIDACION DEL MODELO (gam.check) -----

Method: GCV Optimizer: magic
Smoothing parameter selection converged after 9 iterations.
The RMS GCV score gradient at convergence was 4.159369e-08 .
The Hessian was positive definite.
Model rank = 35 / 35

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

	k'	edf	k-index	p-value
s(Irradiancia)	9.00	8.99	0.98	0.09 .
s(Temperatura)	9.00	8.93	0.99	0.34
ti(Irradiancia,Temperatura)	16.00	12.52	0.96	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
NULL

----- ANOVA DEL MODELO -----

Family: gaussian

Link function: identity

Formula:

Desequilibrio.Vz.Min ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia,
Temperatura)

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Irradiancia)	8.992	9.000	83.72	<2e-16
s(Temperatura)	8.930	8.995	241.20	<2e-16
ti(Irradiancia,Temperatura)	12.518	13.529	115.97	<2e-16

B.3. Modelo lineal Desequilibrio.Vn.Min en CT Nuevo

----- RESUMEN DEL MODELO LINEAL -----

Call:

lm(formula = Desequilibrio.Vn.Min ~ Irradiancia * Temperatura,
data = nuevo_modelos)

Residuals:

Min	1Q	Median	3Q	Max
-0.189630	-0.035132	-0.000087	0.034454	0.201493

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.096e-01	5.379e-04	389.59	<2e-16 ***
Irradiancia	-2.422e-04	3.857e-06	-62.81	<2e-16 ***
Temperatura	-3.435e-03	5.268e-05	-65.19	<2e-16 ***
Irradiancia:Temperatura	5.159e-06	1.015e-07	50.82	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05216 on 33263 degrees of freedom

Multiple R-squared: 0.4664, Adjusted R-squared: 0.4663

F-statistic: 9691 on 3 and 33263 DF, p-value: < 2.2e-16

----- ANOVA DEL MODELO -----

Analysis of Variance Table

Response: Desequilibrio.Vn.Min

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Irradiancia	1	65.200	65.200	23966.2	< 2.2e-16 ***
Temperatura	1	6.862	6.862	2522.5	< 2.2e-16 ***
Irradiancia:Temperatura	1	7.027	7.027	2583.0	< 2.2e-16 ***
Residuals	33263	90.493	0.003		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

B.4. Modelo lineal Tensión.de.Medio.Ciclo.V.RMS.NG.Med en CT Nuevo

----- RESUMEN DEL MODELO LINEAL -----

Call:

lm(formula = Tensión.de.Medio.Ciclo.V.RMS.NG.Med ~ Irradiancia *
Temperatura, data = nuevo_modelos)

Residuals:

Min	1Q	Median	3Q	Max
-4.0939	-0.6991	0.0607	0.7383	3.7097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.480e+01	1.025e-02	1444.22	<2e-16 ***
Irradiancia	7.141e-03	7.347e-05	97.19	<2e-16 ***
Temperatura	-1.474e-01	1.004e-03	-146.86	<2e-16 ***
Irradiancia:Temperatura	-4.179e-05	1.933e-06	-21.62	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9935 on 33263 degrees of freedom

Multiple R-squared: 0.4652, Adjusted R-squared: 0.4651

F-statistic: 9644 on 3 and 33263 DF, p-value: < 2.2e-16

----- ANOVA DEL MODELO -----

Analysis of Variance Table

Response: Tensión.de.Medio.Ciclo.V.RMS.NG.Med

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Irradiancia	1	940	940.2	952.43	< 2.2e-16 ***
Temperatura	1	27157	27157.1	27511.07	< 2.2e-16 ***
Irradiancia:Temperatura	1	461	461.3	467.27	< 2.2e-16 ***
Residuals	33263	32835	1.0		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

B.5. Modelo lineal Corriente.L1.Min en CT Nuevo

----- RESUMEN DEL MODELO LINEAL -----

Call:

lm(formula = Corriente.L1.Min ~ Irradiancia * Temperatura, data = nuevo_modelos)

Residuals:

Min	1Q	Median	3Q	Max
-232.01	-33.05	-1.56	22.61	420.19

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.585e+02	5.636e-01	281.17	<2e-16 ***
Irradiancia	-4.586e-01	4.041e-03	-113.48	<2e-16 ***
Temperatura	5.904e+00	5.520e-02	106.95	<2e-16 ***
Irradiancia:Temperatura	7.020e-03	1.063e-04	66.01	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.65 on 33263 degrees of freedom

Multiple R-squared: 0.4371, Adjusted R-squared: 0.4371

F-statistic: 8610 on 3 and 33263 DF, p-value: < 2.2e-16

----- ANOVA DEL MODELO -----

Analysis of Variance Table

Response: Corriente.L1.Min

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Irradiancia	1	5372340	5372340	1798.9	< 2.2e-16 ***
Temperatura	1	58754806	58754806	19674.0	< 2.2e-16 ***
Irradiancia:Temperatura	1	13012088	13012088	4357.1	< 2.2e-16 ***
Residuals	33263	99337496	2986		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

B.6. Modelo GAM Corriente.L1.Min en CT Nuevo

----- RESUMEN DEL MODELO -----

Family: gaussian

Link function: identity

Formula:

Corriente.L1.Min ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia,
Temperatura)

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	75.10	48.66	1.543	0.123

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Irradiancia)	9.000	9.000	160.2	<2e-16 ***
s(Temperatura)	8.951	8.997	1079.8	<2e-16 ***
ti(Irradiancia,Temperatura)	15.974	15.999	777.0	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.657 Deviance explained = 65.8%

GCV = 1819 Scale est. = 1817.1 n = 33267

----- VALIDACION DEL MODELO (gam.check) -----

Method: GCV Optimizer: magic
Smoothing parameter selection converged after 29 iterations.
The RMS GCV score gradient at convergence was 0.0001469221 .
The Hessian was positive definite.
Model rank = 35 / 35

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

	k'	edf	k-index	p-value
s(Irradiancia)	9.00	9.00	1.01	0.72
s(Temperatura)	9.00	8.95	0.95	<2e-16 ***
ti(Irradiancia,Temperatura)	16.00	15.97	0.99	0.38

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
NULL

----- ANOVA DEL MODELO -----

Family: gaussian
Link function: identity

Formula:
Corriente.L1.Min ~ s(Irradiancia) + s(Temperatura) + ti(Irradiancia,
Temperatura)

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Irradiancia)	9.000	9.000	160.2	<2e-16
s(Temperatura)	8.951	8.997	1079.8	<2e-16
ti(Irradiancia,Temperatura)	15.974	15.999	777.0	<2e-16

Apéndice C

Código fuente en R Markdown

En este apéndice se recopilan todos los archivos de código desarrollados en lenguaje R Markdown durante el proceso de análisis. Los scripts incluidos corresponden a los distintos bloques de trabajo realizados a lo largo del proyecto, tanto los compartidos con el tutor como aquellos intercambiados con los profesores del Departamento de Ingeniería Eléctrica de la Universidad de Valladolid.

C.0.1. Análisis exploratorio inicial

Primer contacto con los datos iniciales pertenecientes a las tres semanas de enero.

```
1 ---
2 title: "TFG-Estadística"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output:
6   html_document: default
7   pdf_document: default
8 ---
9
10 # Exploracion de los datos
11
12 ### Librerías necesarias
13 '{{{r , warning=FALSE, message=FALSE}}
14 library(readxl)
15 library(factoextra)
16 library(dplyr)
17 library(lubridate)
18 ''
19
20
21
```

```

22
23 ### Datos de CT Bombas
24
25 Leo los datos, paso el formato de la hora al correcto y creo una
    variable con fecha y hora por si puede ser util mas adelante.
26 ```{r}
27 bombas <- read_excel("C:/Users/ismae/Desktop/TFG EST/data/Datos CT
    Bombas/Datos de 09-01 a 30-01 (CT Bombas).xlsx")
28 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
29 bombas$FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora),
    format="%Y-%m-%d %H:%M:%S")
30 bombas <- bombas[, c(1, 2, ncol(bombas), 3:(ncol(bombas) - 1))]
31 head(bombas)
32 ```
33
34
35 Compruebo si existen datos que sean nulos. Se llega a ver que al
    parecer existe en el excel una columna sin datos al final del mismo
    . Se elimina dicha columna de los datos
36 ```{r}
37 colSums(is.na(bombas))[colSums(is.na(bombas)) > 0] # N mero de NA's
38 bombas <- bombas[, -which(names(bombas) == "Column1")]
39 anyNA(bombas)
40 ```
41
42 Paso a dataframe y creo una copia sin las fechas
43 ```{r}
44 bombas <- as.data.frame(bombas)
45 bombas_sin_fecha<-bombas[,-c(1,2,3)]
46 ```
47
48 Se intenta un pca, pero da error por divisi n por 0. Parece que
    algunas variables no tienen varianza.
49 ```{r , error=TRUE}
50 pca_result <- prcomp(bombas_sin_fecha, center = TRUE, scale. = TRUE)
51 ```
52
53
54 Debido al problema encontrado con las componentes principales, busco
    cuantas variables son de varianza 0.
55 ```{r}
56 sum(apply(bombas_sin_fecha, 2, var)==0)
57 ```
58
59
60 Quito todas las variables con varianza 0.
61 ```{r}
62 cols_cero_var <- names(bombas_sin_fecha)[apply(bombas_sin_fecha, 2,
    var, na.rm = TRUE) == 0]
63 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
    cols_cero_var]

```

```

64  '''
65
66
67
68  Intento reaizar de nuevo el analisis de componentes principales
69  '''{r}
70  pca_result <- prcomp(bombas_filtrado, center = TRUE, scale. = TRUE)
71  fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 100))
72  '''
73
74  Peque o analisis de correlaciones de los datos
75  '''{r}
76  cor_matrix<-cor(bombas_filtrado)
77  high_cor<-which(abs(cor_matrix) > 0.90, arr.ind = TRUE)
78  high_cor<-high_cor[high_cor[,1] != high_cor[,2],]
79  dim(high_cor)
80  head(high_cor)
81  '''
82
83
84  '''{r}
85  low_cor<-which(abs(cor_matrix) < 0.30, arr.ind = TRUE)
86  dim(low_cor)
87  head(low_cor)
88  '''
89
90
91
92
93  ### Datos de CT Nuevo
94  Leo los datos, paso el formato de la hora al correcto y creo una
    variable con fecha y hora por si puede ser util mas adelante.
95  '''{r}
96  nuevo <- read_excel("C:/Users/ismae/Desktop/TFG EST/data/Datos CT
    Nuevo/Datos de 09-01 a 30-01 (CT Nuevo).xlsx")
97  nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
98  nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="
    %Y-%m-%d %H:%M:%S")
99  nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
100 head(nuevo)
101 '''
102
103
104
105 Compruebo si existen datos que sean nulos. Se llega a ver que al
    parecer existen en el excel dos columnas sin datos al final del
    mismo. Se eliminan ambas columnas.
106 '''{r}
107 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
108 colMeans(is.na(nuevo))[colSums(is.na(nuevo)) > 0] * 100 # Porcentaje
    de NA's

```

```

109
110 nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
111 nuevo <- nuevo[, -which(names(nuevo) == "_1")]
112 anyNA(nuevo)
113 ```
114
115 Veo que todos los NA son de esas 4 observaciones.
116 ```{r}
117 which(is.na(nuevo$'Arm nicos Corriente15 N Max'))
118 col_na_579<-which(is.na(nuevo[579,]))
119 col_na_1954<-which(is.na(nuevo[1954,]))
120 col_na_2595<-which(is.na(nuevo[2595,]))
121 col_na_2609<-which(is.na(nuevo[2609,]))
122 colnames(nuevo)[col_na_579]
123 colnames(nuevo)[col_na_1954]
124 colnames(nuevo)[col_na_2595]
125 colnames(nuevo)[col_na_2609]
126 ```
127
128
129
130
131
132 Al igual que antes paso a dataframe y creo una copia sin las columnas
    de las fechas.
133 ```{r}
134 nuevo<-as.data.frame(nuevo)
135 nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
136 sum(apply(nuevo_sin_fecha, 2, var, na.rm = TRUE)==0)
137 ```
138
139
140 Busco las 7 variables sin varianza y las elimino
141 ```{r}
142 cols_cero_var <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha, 2, var,
    na.rm = TRUE) == 0]
143 cols_cero_var
144 nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
    cols_cero_var]
145 ```
146
147
148
149 Todavia no se puede realizar el pca debido a los valores ausentes.
150 ```{r , error=TRUE}
151 pca_result <- prcomp(nuevo_filtrado, center = TRUE, scale. = TRUE)
152 library(factoextra)
153 fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 100))
154 ```
155
156

```

```

157 |
158 |
159 | ### Procesado de los datos de Irradiancia
160 |
161 | ```{r}
162 | irradiancia <- read_excel("C:/Users/ismae/Desktop/TFG EST/data/Datos
163 |   Irradiancia/Irradiancia 09-01 a 30-01.xlsx")
164 | names(irradiancia)[1]<-"FechaHora"
165 | head(irradiancia)
166 | ```
167 |
168 |
169 | Busco NA's y encuentro 4 observaciones
170 | ```{r}
171 | colSums(is.na(irradiancia))[colSums(is.na(irradiancia)) > 0]
172 |
173 | which(is.na(irradiancia[,2]))
174 | which(is.na(irradiancia[,3]))
175 | which(is.na(irradiancia[,4]))
176 | ```
177 |
178 |
179 | Prueba para ver como quedarían las agrupaciones de los datos en
180 |   intervalos de 10 min. El agrupamiento realizado comienza en la hora
181 |   a la que comienzan los de CT Bombas, pues los datasets no tienen
182 |   exactamente los mismos rangos horarios.
183 | ```{r}
184 | inicio <- as.POSIXct("2025-01-09 10:23:45") # Punto de inicio de los
185 |   intervalos
186 |
187 | datos_agrupados <- irradiancia %>%
188 |   mutate(Intervalo = cut(FechaHora, breaks = seq(inicio, max(FechaHora
189 |     ) + 600, by = "10 min"), right = FALSE)) %>%
190 |   group_by(Intervalo) %>%
191 |   summarise(
192 |     Suma_Var1 = sum('Temperatura Prom. (C)'),
193 |     Media_Var1 = mean('Temperatura Prom. (C)'),
194 |     Min_Var1 = min('Temperatura Prom. (C)'),
195 |     Max_Var1 = max('Temperatura Prom. (C)'),
196 |
197 |     Suma_Var2 = sum('Irradiancia (CT Bombas) Prom. (Wm2)'),
198 |     Media_Var2 = mean('Irradiancia (CT Bombas) Prom. (Wm2)'),
199 |     Min_Var2 = min('Irradiancia (CT Bombas) Prom. (Wm2)'),
200 |     Max_Var2 = max('Irradiancia (CT Bombas) Prom. (Wm2)'),
201 |
202 |     Suma_Var3 = sum('Irradiancia CT Nuevo) Prom. (Wm2)'),
203 |     Media_Var3 = mean('Irradiancia CT Nuevo) Prom. (Wm2)'),
204 |     Min_Var3 = min('Irradiancia CT Nuevo) Prom. (Wm2)'),
205 |     Max_Var3 = max('Irradiancia CT Nuevo) Prom. (Wm2)')
206 |   )

```

```

202
203 print(datos_agrupados)
204 '''
205
206
207
208
209 La ltima observaci n de los datos de irradiancia no tiene fecha (
    alomejor es por el agrupamiento realizado, mirarlo)
210 '''{r}
211 tail(datos_agrupados)
212 '''

```

Listing C.1: Análisis exploratorio inicial

C.0.2. Nombres de variables constantes

Generación de informe sobre las variables sin varianza de los datos iniciales.

```

1 ---
2 title: "Nombres de las constantes"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"
5 output: pdf_document
6 ---
7
8
9 '''{r , warning=FALSE, message=FALSE, echo=FALSE}
10 library(readxl)
11 library(factoextra)
12 library(dplyr)
13 library(knitr)
14 library(lubridate)
15 library(data.table)
16
17 # No hace falta poner '/' al final, ya se tiene en cuenta luego
18 ruta_datos <- "E:/TFG EST/data"
19 '''
20
21
22
23 '''{r, echo = FALSE, results = "hide"}
24 ruta_bombas <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Bombas).
    xlsx", sep="")
25 bombas <- read_excel(ruta_bombas)
26 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
27 bombas$FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora),
    format="%Y-%m-%d %H:%M:%S")
28 bombas <- bombas[, c(1, 2, ncol(bombas), 3:(ncol(bombas) - 1))]
29 head(bombas)

```

```

30  '''
31
32
33  '''{r, echo = FALSE, results = "hide"}
34  colSums(is.na(bombas))[colSums(is.na(bombas)) > 0] # N mero de NA's
35  bombas <- bombas[, -which(names(bombas) == "Column1")]
36  anyNA(bombas)
37  '''
38
39  '''{r, echo = FALSE, results = "hide"}
40  bombas <- as.data.frame(bombas)
41  bombas_sin_fecha<-bombas[,-c(1,2,3)]
42  '''
43
44
45  '''{r, echo = FALSE, results = "hide"}
46  ruta_nuevo <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Nuevo).
47    xls",sep="")
48  nuevo <- read_excel(ruta_nuevo)
49  nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
50  nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="
51    %Y-%m-%d %H:%M:%S")
52  nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
53  head(nuevo)
54  '''
55
56  '''{r, echo = FALSE, results = "hide"}
57  colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
58  colMeans(is.na(nuevo))[colSums(is.na(nuevo)) > 0] * 100 # Porcentaje
59    de NA's
60
61  nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
62  nuevo <- nuevo[, -which(names(nuevo) == "_1")]
63  anyNA(nuevo)
64  '''
65
66  '''{r, echo = FALSE, results = "hide"}
67  nuevo<-as.data.frame(nuevo)
68  nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
69  '''
70
71  # Tablas con los nombres de las variables que en realidad son
72    constantes
73
74  '''{r, echo = FALSE, results = "hide"}
75  cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
76    2, var, na.rm = TRUE) == 0]

```

```

76 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
77   , 2, var, na.rm = TRUE) == 0]
78
79
80 ```{r, echo=FALSE}
81 kable(data.frame(Nombres = cols_cero_var_nuevo), caption = "Variables
    que son constantes de 'CT nuevo'")
82 ```
83
84
85 ```{r, echo=FALSE}
86 kable(data.frame(Nombres = cols_cero_var_bombas), caption = "Variables
    que son constantes de 'CT bombas'")
87 ```

```

Listing C.2: Nombres de variables constantes

C.0.3. Correlaciones mayores a 0.99 en CT Bombas

Generación de nombres de las parejas con una correlación mayor a 0.99 de los datos iniciales.

```

1 ---
2 title: "Correlaciones > 0.99 CT bombas"
3 author: "Ismael Carbajo Valor"
4 date: "2025-03-06"
5 output: pdf_document
6 ---
7
8 ```{r , warning=FALSE, message=FALSE, echo=FALSE}
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15
16 # No hace falta poner '/' al final, ya se tiene en cuenta luego
17 ruta_datos <- "E:/TFG EST/data"
18 ```
19
20
21
22 ```{r, echo = FALSE, results = "hide"}
23 ruta_bombas <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Bombas).
    xlsx", sep="")
24 bombas <- read_excel(ruta_bombas)
25 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")

```

```

26 bombas$FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora),
    format="%Y-%m-%d %H:%M:%S")
27 bombas <- bombas[, c(1, 2, ncol(bombas), 3:(ncol(bombas) - 1))]
28 head(bombas)
29 '''
30
31
32 '''{r, echo = FALSE, results = "hide"}
33 colSums(is.na(bombas))[colSums(is.na(bombas)) > 0] # N mero de NA's
34 bombas <- bombas[, -which(names(bombas) == "Column1")]
35 anyNA(bombas)
36 '''
37
38 '''{r, echo = FALSE, results = "hide"}
39 bombas <- as.data.frame(bombas)
40 bombas_sin_fecha<-bombas[, -c(1,2,3)]
41 '''
42
43
44 '''{r, echo = FALSE, results = "hide"}
45 ruta_nuevo <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Nuevo).
    xlsx", sep="")
46 nuevo <- read_excel(ruta_nuevo)
47 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
48 nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="%
    %Y-%m-%d %H:%M:%S")
49 nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
50 head(nuevo)
51 '''
52
53
54 '''{r, echo = FALSE, results = "hide"}
55 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
56 colMeans(is.na(nuevo))[colSums(is.na(nuevo)) > 0] * 100 # Porcentaje
    de NA's
57
58 nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
59 nuevo <- nuevo[, -which(names(nuevo) == "_1")]
60 anyNA(nuevo)
61 '''
62
63
64 '''{r, echo = FALSE, results = "hide"}
65 nuevo<-as.data.frame(nuevo)
66 nuevo_sin_fecha<-nuevo[, -c(1,2,3)]
67 '''
68
69
70
71 '''{r, echo = FALSE, results = "hide"}
72 cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,

```

```

73     2, var, na.rm = TRUE) == 0]
74 nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
    cols_cero_var_nuevo]
75 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
    , 2, var, na.rm = TRUE) == 0]
76 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
    cols_cero_var_bombas]
77
78
79
80 ````{r, echo=FALSE}
81 library(dplyr)
82 library(tidyr)
83
84 cor_matrix <- cor(bombas_filtrado, use = "complete.obs")
85
86 cor_long <- as.data.frame(as.table(cor_matrix))
87 cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
88 cor_long$abs_cor <- abs(cor_long$Freq)
89
90 cor_greater_than_099 <- cor_long %>%
91   filter(abs_cor > 0.99)
92
93 kable(cor_greater_than_099, caption = "Variables con correlaci n
    mayor a 0.99")
94 ````

```

Listing C.3: Correlaciones mayores a 0.99 en CT Bombas

C.0.4. Correlaciones mayores a 0.99 en CT Nuevo

Generación de nombres de las parejas con una correlación mayor a 0.99 de los datos iniciales.

```

1 ---
2 title: "Correlaciones > 0.99 CT nuevo"
3 author: "Ismael Carbajo Valor"
4 date: "2025-03-06"
5 output: pdf_document
6 ---
7
8 ````{r , warning=FALSE, message=FALSE, echo=FALSE}
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)

```

```

15
16 # No hace falta poner '/' al final, ya se tiene en cuenta luego
17 ruta_datos <- "E:/TFG EST/data"
18 '''
19
20
21
22 '{{{r, echo = FALSE, results = "hide"}}
23 ruta_nuevo <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Nuevo).
      xlsx", sep="")
24 nuevo <- read_excel(ruta_nuevo)
25 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
26 nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="
      %Y-%m-%d %H:%M:%S")
27 nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
28 head(nuevo)
29 '''
30
31
32 '{{{r, echo = FALSE, results = "hide"}}
33 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
34 nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
35 anyNA(nuevo)
36 '''
37
38 '{{{r, echo = FALSE, results = "hide"}}
39 nuevo <- as.data.frame(nuevo)
40 nuevo_sin_fecha <- nuevo[, -c(1, 2, 3)]
41 '''
42
43
44 '{{{r, echo = FALSE, results = "hide"}}
45 ruta_nuevo <- paste(ruta_datos, "/Datos de 09-01 a 30-01 (CT Nuevo).
      xlsx", sep="")
46 nuevo <- read_excel(ruta_nuevo)
47 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
48 nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="
      %Y-%m-%d %H:%M:%S")
49 nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
50 head(nuevo)
51 '''
52
53
54 '{{{r, echo = FALSE, results = "hide"}}
55 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
56 colMeans(is.na(nuevo))[colSums(is.na(nuevo)) > 0] * 100 # Porcentaje
      de NA's
57
58 nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
59 nuevo <- nuevo[, -which(names(nuevo) == "_1")]
60 anyNA(nuevo)

```

```

61  ""
62
63
64  ""{r, echo = FALSE, results = "hide"}
65  nuevo<-as.data.frame(nuevo)
66  nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
67  ""
68
69
70
71  ""{r, echo = FALSE, results = "hide"}
72  cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
73    2, var, na.rm = TRUE) == 0]
74  nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
75    cols_cero_var_nuevo]
76  cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
77    2, var, na.rm = TRUE) == 0]
78  nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
79    cols_cero_var_nuevo]
80  ""
81
82  ""{r, echo=FALSE}
83  library(dplyr)
84  library(tidyr)
85
86  cor_matrix <- cor(nuevo_filtrado, use = "complete.obs")
87
88  cor_long <- as.data.frame(as.table(cor_matrix))
89  cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
90  cor_long$abs_cor <- abs(cor_long$Freq)
91
92  cor_greater_than_099 <- cor_long %>%
93    filter(abs_cor > 0.99)
94
95  kable(cor_greater_than_099, caption = "Variables con correlaci n
96    mayor a 0.99")
97  ""

```

Listing C.4: Correlaciones mayores a 0.99 en CT Nuevo

C.0.5. Informe inicial

Generación del informe descriptivo, con el análisis de correlaciones entre otros, de los datos iniciales.

```

1  ---
2  title: "Informe Inicial"

```

```

3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output: pdf_document
6 ---
7
8 '{r , warning=FALSE, message=FALSE, echo=FALSE}'
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15
16 # No hace falta poner '/' al final, ya se tiene en cuenta luego
17 ruta_datos <- "H:/TFG EST/data"
18 ''
19
20
21
22 # Datos de CT Bombas
23
24 '{r, echo = FALSE, results = "hide"}'
25 ruta_bombas <- paste(ruta_datos, "Datos CT Bombas/Datos de 09-01 a
26 30-01 (CT Bombas).xlsx", sep="")
27 bombas <- read_excel(ruta_bombas)
28 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
29 bombas$FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora),
30 format="%Y-%m-%d %H:%M:%S")
31 bombas <- bombas[, c(1, 2, ncol(bombas), 3:(ncol(bombas) - 1))]
32 head(bombas)
33 ''
34
35 '{r, echo = FALSE, results = "hide"}'
36 colSums(is.na(bombas))[colSums(is.na(bombas)) > 0] # N mero de NA's
37 bombas <- bombas[, -which(names(bombas) == "Column1")]
38 anyNA(bombas)
39 ''
40
41 '{r, echo = FALSE, results = "hide"}'
42 bombas <- as.data.frame(bombas)
43 bombas_sin_fecha <- bombas[, -c(1,2,3)]
44 ''
45
46 '{r, echo=FALSE}'
47 cat("Fecha y hora inicio de los datos: ", format(bombas$FechaHora[1], "
48 %Y-%m-%d %H:%M:%S"), "\n")
49 cat("Fecha y hora final de los datos: ", format(bombas$FechaHora[length
50 (bombas$FechaHora)], "%Y-%m-%d %H:%M:%S"), "\n")
51 cat("\n", "

```

```

    ", "\n\n")
49 cat("N mero de variables recogidas: ", dim(bombas_sin_fecha)[2], "\n"
    )
50 cat("N mero de variables que en realidad son constantes: ", sum(apply
    (bombas_sin_fecha, 2, var)==0), "\n")
51 cat("N mero de verdaderas variables: ", sum(apply(bombas_sin_fecha,
    2, var)!=0), "\n")
52 cat("\n", "
    -----
    ", "\n\n")
53 cat("No hay valores ausentes", "\n")
54 '''
55
56
57 # Datos de CT Nuevo
58
59 '''{r, echo = FALSE, results = "hide"}
60 ruta_nuevo <- paste(ruta_datos, "Datos CT Nuevo/Datos de 09-01 a 30-01
    (CT Nuevo).xlsx", sep="")
61 nuevo <- read_excel(ruta_nuevo)
62 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
63 nuevo$FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="
    %Y-%m-%d %H:%M:%S")
64 nuevo <- nuevo[, c(1, 2, ncol(nuevo), 3:(ncol(nuevo) - 1))]
65 head(nuevo)
66 '''
67
68
69 '''{r, echo = FALSE, results = "hide"}
70 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
71 colMeans(is.na(nuevo))[colSums(is.na(nuevo)) > 0] * 100 # Porcentaje
    de NA's
72
73 nuevo <- nuevo[, -which(names(nuevo) == "Column1")]
74 nuevo <- nuevo[, -which(names(nuevo) == "_1")]
75 anyNA(nuevo)
76 '''
77
78 '''{r, echo = FALSE, results = "hide"}
79 which(is.na(nuevo$'Arm nicos Corriente15 N Max'))
80 col_na_579<-which(is.na(nuevo[579,]))
81 col_na_1954<-which(is.na(nuevo[1954,]))
82 col_na_2595<-which(is.na(nuevo[2595,]))
83 col_na_2609<-which(is.na(nuevo[2609,]))
84 colnames(nuevo)[col_na_579]
85 colnames(nuevo)[col_na_1954]
86 colnames(nuevo)[col_na_2595]
87 colnames(nuevo)[col_na_2609]
88 '''
89
90 '''{r, echo = FALSE, results = "hide"}

```

```

91 nuevo<-as.data.frame(nuevo)
92 nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
93 sum(apply(nuevo_sin_fecha, 2, var, na.rm = TRUE)==0)
94 '''
95
96 '{{{r, echo=FALSE}
97 cat("Fecha y hora inicio de los datos: ",format(nuevo$FechaHora[1], "%
98 Y-%m-%d %H:%M:%S"), "\n")
99 cat("Fecha y hora final de los datos: ",format(nuevo$FechaHora[length(
100 nuevo$FechaHora)], "%Y-%m-%d %H:%M:%S"), "\n")
101 cat("\n", "
102 -----
103 ", "\n\n")
104 cat("N mero de variables recogidas: ", dim(nuevo_sin_fecha)[2], "\n")
105 cat("N mero de variables que en realidad son constantes: ", sum(apply
106 (nuevo_sin_fecha, 2, var, na.rm = TRUE)==0), "\n")
107 cat("N mero de verdaderas variables: ", sum(apply(nuevo_sin_fecha, 2,
108 var, na.rm = TRUE)!=0), "\n")
109 cat("\n", "
110 -----
111 ", "\n\n")
112 cat("4 valores ausentes", "\n\n")
113 cat("Las observaciones:",which(is.na(nuevo$'Arm nicos Corriente15 N
114 Max')), "\n")
115 cat("\nCorrespondientes a los tiempos: \n")
116 nuevo$FechaHora[c(which(is.na(nuevo$'Arm nicos Corriente15 N Max')))]
117 cat("\nLos NA se encuentran en las mismas variables en las 4
118 observaciones:", "\n")
119 colnames(nuevo)[col_na_579]
120 '''
121
122 # Estudio de las variables que son constantes
123
124 '{{{r, echo = FALSE, results = "hide"}
125 cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
126 2, var, na.rm = TRUE) == 0]
127 nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
128 cols_cero_var_nuevo]
129 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
130 , 2, var, na.rm = TRUE) == 0]
131 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
132 cols_cero_var_bombas]
133 '''
134
135 ## Cuales son esas variables
136 Para este caso, solo se ense an las variables que son constantes del
137 dataset 'CT nuevo', pues 'CT bombas' cuenta con una gran cantidad
138 las cuales no se pueden ense ar de manera compacta en este pdf.
139 Igualmente, por si fuese de inter s, ambas tablas se encuentran en
140 el archivo 'nombres-constantes.pdf'.

```

```

124
125   ```{r, echo=FALSE}
126   kable(data.frame(Nombres = cols_cero_var_nuevo), caption = "Variables
127   que son constantes de 'CT nuevo'")
128   ```
129   ## Las constantes de 'CT nuevo' estan en 'CT bombas'?
130
131   Vemos que las 7 constantes de 'nuevo' no son constantes en 'bombas'
132   debido al resultado nulo:
133   ```{r, echo=FALSE}
134   intersect(cols_cero_var_nuevo, cols_cero_var_bombas)
135   ```
136   Veamos ahora si alguna de las 7 constantes si est n en 'CT bombas'
137   pero no son constantes en este:
138   ```{r, echo=FALSE}
139   intersect(cols_cero_var_nuevo, colnames(bombas))
140   ```
141   De las 7 solo 1 es com n a ambos datasets, siendo constante para 'CT
142   nuevo' pero siendo variable para 'CT bombas'.
143
144   ```{r, echo=FALSE}
145   kable(data.frame(Nombres= intersect(cols_cero_var_nuevo, colnames(
146   bombas))), caption = "Constantes de 'CT nuevo' que son variables en
147   'CT bombas'")
148   ```
149   ## Las constantes de 'CT bombas' estan en 'CT nuevo'?
150
151   Ya hemos visto que ninguna constante es com n a los dos datasets, por
152   tanto se busca si las constantes de 'CT bombas' aparecen como
153   variables en 'CT nuevo':
154   ```{r, echo=FALSE}
155   intersecciones_cte_bombas_con_nuevo<-intersect(cols_cero_var_bombas,
156   colnames(nuevo))
157   cat("Contamos con ",length(intersecciones_cte_bombas_con_nuevo), " que
158   si aparecen")
159   ```
160
161   ```{r, echo=FALSE}
162   kable(data.frame(Nombres= intersecciones_cte_bombas_con_nuevo),
163   caption = "Constantes de 'CT bombas' que son variables en 'CT nuevo'")
164   ```
165
166   # Cuantas variables estan en ambos

```

```

163
164 ## Si consideramos todas las variables que contiene el dataset
165 ```{r, echo=FALSE}
166 coincidencias<-intersect(colnames(bombas_sin_fecha),colnames(
167     nuevo_sin_fecha))
168 cat("Contamos con un total de ", length(coincidencias), " variables
169     que aparecen en ambos datasets." )
170
171
172 ## Si consideramos solo aquellas que verdaderamente son variables (
173     varianza distinta de 0)
174 ```{r, echo=FALSE}
175 coincidencias<-intersect(colnames(bombas_filtrado),colnames(
176     nuevo_filtrado))
177 cat("Contamos con un total de ", length(coincidencias), " variables
178     que aparecen en ambos datasets." )
179
180
181 ```{r, echo=FALSE}
182 kable(coincidencias, caption = "Variables verdaderas comunes a ambos
183     datasets")
184
185
186
187 # Conclusiones sobre las variables y constantes
188
189 Vemos que 126 variables son las que se recogen en ambos centros, de
190     las cuales 86 son variables de verdad, el resto son constantes. Las
191     40 restantes, como ya hemos visto, son constantes en uno de ellos
192     y variable en el otro.
193
194 Aunque en 'CT bombas' se hayan recogido muchas m s variables, si
195     quitamos las que son constantes tenemos menos que en 'CT nuevo'.
196
197 Los rangos de fechas en los que se encuentran son pr cticamente los
198     mismos, pero no iguales, uno empieza 1 hora mas tarde y acaba unas
199     23 horas m s tarde.
200
201 Los nicos valores ausentes, al ser tan pocos no parecen que vayan a
202     originar ning n problema. Igualmente, dependiendo del analisis,
203     ser necesario tratarlos para no originar problemas.
204
205 Las constantes de cada dataset, si se encuentran en el otro, lo hacen
206     como variables, no como constantes.

```

```

199
200 ```{r, echo = FALSE}
201 resumen<-data.frame(
202   "CT Bombas" = c(sum(apply(bombas_sin_fecha, 2, var)==0),length(
      bombas_filtrado), length(coincidencias), length(bombas_filtrado)-
      length(coincidencias)),
203   "CT Nuevo" = c(sum(apply(nuevo_sin_fecha, 2, var, na.rm = TRUE)!=0),
      length(nuevo_filtrado),length(coincidencias), length(
      nuevo_filtrado)-length(coincidencias)))
204 rownames(resumen) <- c("Num constantes", "Num verdaderasVariables", "
      Num variables comunes", "Num variables unicas")
205 kable(resumen, caption = "Resumen sobre verdaderas variables")
206 ```
207
208
209 # Conclusiones seg n la naturaleza de los datos
210
211 Seg n se habl en la reuni n en la escuela de ingenieros, los
      nicos arm nicos que son importantes son los impares, los pares
      ni se utilizan. Esto es debido a la simetr a impar de la onda,
      pues si reflejas la onda respecto al eje horizontal y desplazas
      medio ciclo, la forma se mantiene.
212
213 Esto hace que en la serie de fourier, no solo valores altos de
      arm nicos sean 0, sino tambi n todos los pares.
214
215 Esto es importante, porque vemos que los arm nicos que son variables
      y est n presentes en ambos datasets son 3, 5, 7, 9, 11. Mientras
      que los pares y mayores a 11 son constantes, pero solo en 'CT
      bombas', pues en 'CT nuevo' s son variables, cosa que habr a que
      investigar.
216
217 Esto da cierta explicaci n, pero habr a que analizar m s a fondo,
      pues esto no se cumple con todas las variables.
218
219 Otra cosa a destacar son que solo var an las medias de estos
      arm nicos, todos los m nimos y m ximos no han variado nada.
220
221
222 # Correlaciones de las variables en CT bombas
223
224 Debido a que hay muchas m s variables en 'CT nuevo', primero vemos un
      mapa de calor de 'CT bombas', en el cual ya podemos ver que una
      gran cantidad de variables dicen lo mismo.
225 ```{r, echo=FALSE, warning=FALSE}
226 library(ggplot2)
227 library(GGally)
228 library(corrplot)
229
230 cor_matrix <- cor(bombas_filtrado, use = "complete.obs")
231

```

```

232 corrrplot(cor_matrix, method = "color",
233           tl.cex = 0.1,
234           addCoef.col = NULL,
235           col = colorRampPalette(c("blue", "white", "red"))(200),
236           order = "hclust")
237
238 ' ' '
239
240 '{r, echo=FALSE, warning=FALSE}
241 library(ggcorrplot)
242
243 ggcorrplot(cor_matrix,
244            hc.order = TRUE, # Ordenar por clustering jerarquico
245            type = "lower", # Mostrar solo la mitad inferior (sin
                duplicados)
246            lab = FALSE, # No mostrar los n meros en las celdas
247            tl.col = "white", # Color de las etiquetas (blanco hace
                que sean invisibles)
248            tl.cex = 0, # Hacer que las etiquetas sean
                invisibles al poner el tamaño a 0
249            colors = c("blue", "white", "red")) # Paleta de colores
250
251 ' ' '
252
253
254
255 # Cuantas variables tienen correlacion alta en 'CT bombas'
256
257
258 Aqu se puede ver el n mero de pares de variables que tienen una
    correlacion mayor a cierto nivel.
259 '{r, echo=FALSE}
260 library(dplyr)
261 library(tidyr)
262
263 cor_matrix <- cor(bombas_filtrado, use = "complete.obs")
264
265 cor_long <- as.data.frame(as.table(cor_matrix))
266 cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
267 cor_long$abs_cor <- abs(cor_long$Freq)
268
269 umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
270 resultado <- sapply(umbral_list, function(threshold) {
271   sum(cor_long$abs_cor > threshold)
272 })
273
274 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
    = " ")
275 resultado
276 ' ' '
277

```

```

278 Se adjuntar un pdf con la tabla de los pares de correlaciones
    mayores a 0.99, por si se pueden ver ciertas relaciones.
279
280
281
282
283 # Diferencias entre las 3 ondas en 'CT bombas'?
284
285 Lo primero que he hecho ha sido comprobar que todas las variables
    conforman un tr o con las 3 fases, de esta manera sabemos que no
    hay alguna variable medida en solo alguna de ellas.
286
287 Luego, si solo vemos aquellas que tienen varianza, exceptuando una de
    ellas la cual en L2 es constante ('Cos Phi Cl sico L2N M x.'),
    las dem s si se pueden agrupar con sus semejantes de todas las
    fases.
288
289 Para no ocasionar futuros problemas en los siguientes c lculos , se
    mete en la tabla con las dem s .
290
291 ```{r, echo=FALSE}
292 fase1_cols <- grep("L1", colnames(bombas_filtrado), value = TRUE)
293 fase2_cols <- grep("L2", colnames(bombas_filtrado), value = TRUE)
294 fase3_cols <- grep("L3", colnames(bombas_filtrado), value = TRUE)
295
296 fase2_filled <- append(fase2_cols, "Cos Phi Cl sico L2N M x.", after
    = 35)
297
298 grupo_fases <- list(fase1 = fase1_cols, fase2 = fase2_filled, fase3 =
    fase3_cols)
299
300 grupo_fases_df <- data.frame(
301   Fase_L1 = grupo_fases$fase1,
302   Fase_L2 = grupo_fases$fase2,
303   Fase_L3 = grupo_fases$fase3
304 )
305 kable(grupo_fases_df, caption = "Grupos de variables por fase (L1, L2,
    L3)")
306 ```
307
308 Vemos ahora la tabla de cuales son las correlaciones entre los
    diferentes pares de cada grupo de variables en las fases.
309
310 La mayor a de las variables su correlaci n es muy alta y puede ser
    que lo mejor sea combinarlas, aunque algunas no se parecen entre
    las diferentes fases.
311
312 ```{r, echo=FALSE, warning=FALSE}
313 grupo_fases_df_cor <- data.frame()
314
315 for(i in 1:length(grupo_fases$fase1)) {

```

```

316 # Tomamos las columnas correspondientes a la misma variable en L1,
    L2 y L3
317 variable_L1 <- grupo_fases$fase1[i]
318 variable_L2 <- grupo_fases$fase2[i]
319 variable_L3 <- grupo_fases$fase3[i]
320
321 # Seleccionamos las columnas de las tres fases
322 columnas_grupo <- bombas[, c(variable_L1, variable_L2, variable_L3)]
323
324 # Calcular la correlación entre las tres columnas
325 cor_value <- cor(columnas_grupo, use = "complete.obs")
326
327 # Aadir los resultados a un dataframe de correlaciones por grupo
328 grupo_fases_df_cor <- rbind(grupo_fases_df_cor, data.frame(Variable
    = variable_L1,
329
    Correlacion_L1_L2
    =
    cor_value
    [1,2],
330
    Correlacion_L1_L3
    =
    cor_value
    [1,3],
331
    Correlacion_L2_L3
    =
    cor_value
    [2,3]))
332 }
333
334 # Mostrar la tabla de correlaciones entre las fases L1, L2 y L3
335 kable(grupo_fases_df_cor, caption = "Correlaciones entre las mismas
    variables en L1, L2, L3")
336
337 ' ' '
338
339
340 # Correlaciones de las variables en CT nuevo
341 Por el momento, hasta no sacar conclusiones efectivas por esta parte,
    se omite realizar exactamente el mismo estudio de correlaciones en
    'CT nuevo' debido al gran tamaño de las tablas. Pero si se realiza
    un mapa de calor para poder ver un poco la estructura y número de
    parejas con correlaciones altas.
342
343
344 Vemos al igual que antes una gran cantidad de zonas con mucha
    correlación.
345
346 '{r, echo=FALSE, warning=FALSE}
347 library(ggplot2)
348 library(GGally)
349 library(corrplot)

```

```

350
351 cor_matrix <- cor(nuevo_filtrado, use = "complete.obs")
352
353 corrplot(cor_matrix, method = "color",
354           tl.cex = 0.1,
355           addCoef.col = NULL,
356           col = colorRampPalette(c("blue", "white", "red"))(200),
357           order = "hclust")
358
359 '''
360
361
362
363 Viendo ahora el n mero de parejas con correlaciones altas.
364
365 ```{r, echo=FALSE}
366 library(dplyr)
367 library(tidyr)
368
369 cor_matrix <- cor(nuevo_filtrado, use = "complete.obs")
370
371 cor_long <- as.data.frame(as.table(cor_matrix))
372 cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
373 cor_long$abs_cor <- abs(cor_long$Freq)
374
375 umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
376 resultado <- sapply(umbral_list, function(threshold) {
377   sum(cor_long$abs_cor > threshold)
378 })
379
380 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
381   = "")
382 resultado
383 '''
384
385 # Analisis de correlaciones con nicamente las 86 variables comunes
386
387 ## En 'CT bombas'
388 ```{r, echo=FALSE, warning=FALSE}
389 library(ggplot2)
390 library(GGally)
391 library(corrplot)
392
393 cor_matrix <- cor(bombas[,coincidencias], use = "complete.obs")
394
395 corrplot(cor_matrix, method = "color",
396           tl.cex = 0.1,
397           addCoef.col = NULL,
398           col = colorRampPalette(c("blue", "white", "red"))(200),
399           order = "hclust")

```

```

400  ‘‘‘
401
402  ‘‘{r, echo=FALSE}
403  library(dplyr)
404  library(tidyr)
405
406  cor_matrix <- cor(bombas[,coincidencias], use = "complete.obs")
407
408  cor_long <- as.data.frame(as.table(cor_matrix))
409  cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
410  cor_long$abs_cor <- abs(cor_long$Freq)
411
412  umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
413  resultado <- sapply(umbral_list, function(threshold) {
414    sum(cor_long$abs_cor > threshold)
415  })
416
417  names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
418    = "")
419  resultado
420  ‘‘‘
421  ## En 'CT nuevo'
422  ‘‘{r, echo=FALSE, warning=FALSE}
423  library(ggplot2)
424  library(GGally)
425  library(corrplot)
426
427  cor_matrix <- cor(nuevo[,coincidencias], use = "complete.obs")
428
429  corrplot(cor_matrix, method = "color",
430    tl.cex = 0.1,
431    addCoef.col = NULL,
432    col = colorRampPalette(c("blue", "white", "red"))(200),
433    order = "hclust")
434
435  ‘‘‘
436
437  ‘‘{r, echo=FALSE}
438  library(dplyr)
439  library(tidyr)
440
441  cor_matrix <- cor(nuevo[,coincidencias], use = "complete.obs")
442
443  cor_long <- as.data.frame(as.table(cor_matrix))
444  cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
445  cor_long$abs_cor <- abs(cor_long$Freq)
446
447  umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
448  resultado <- sapply(umbral_list, function(threshold) {
449    sum(cor_long$abs_cor > threshold)

```

```

450 })
451
452 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
    =")
453 resultado
454 ""
455
456 # Conclusiones finales y dudas
457
458 ## De este documento y viendo los resultados
459 Claramente, todas las constantes tienen que ser quitadas de los datos
    pues no aportan nada a la hora de la modelizaci n.
460
461 Una vez hecho esto, hay que tratar el c mo se va a efectuar el
    analisis con variables que no est n en ambos datasets.
462
463 Una ventaja que tienen las 86 variables que coinciden en ambos
    datasets es que los tr os est n completos, es decir, las
    variables que se miden de manera separada en cada fase se
    encuentran las 3.
464
465 Viendo ya la parte de las correlaciones, hay muchas variables con alta
    correlaci n. Lo que no estoy seguro es la manera de proceder,
    pues hay muchas parejas posibles. La elecci n del umbral a partir
    del cual se considera una correlaci n lo suficientemente alta es
    crucial. Adem s, puede que la mejor opci n sea primero juntar los
    grupos de 3 variables que son pr cticamente iguales en las 3
    fases y luego proceder a realizar el analisis de correlaciones
    general para seguir reduciendo el n mero de variables.
466
467
468 ## De las tablas hechas, tanto las de este documento como las de los
    otros
469
470 Hay variables que son la suma de las 3 fases que claramente van a
    tener gran correlacion entre ellas.
471
472 Pr cticamente la mayor a de constantes de 'CT bombas' corresponden
    con arm nicos, pues se miden hasta 50. Siendo los nicos que no
    lo son los comentados antes, medias de los primeros impares.
473
474 Viendo las tablas de las correlaciones >0.99, algunas corresponden a
    los pares de variables que son las mismas en las diferentes fases.
    Cabe decir que no hay una estrucutra clara sobre como son estas
    parejas. Depende de cada variable la estrucutra es distinta. Por
    ejemplo:
475
476 Si vamos a 'Var2' de la tabla de 'CT bombas', si miramos 'Corriente',
    las correlaciones altas son de la siguiente manera: dividimos en
    grupos por fase (L1, L2 o L3)y tipo de medida (min, max o med).
    Estos grupos se relacionan con las mismas variables de su

```

correspondiente fase y tipo de medida, pero siempre son las mismas 6 variables. Esto ocurre de esa variable en adelante, pero no en las anteriores en la tabla.

Listing C.5: Informe inicial

C.0.6. Primeros modelos

Generación del informe de los primeros modelos lineales de los datos iniciales.

```

1 ---
2 title: "Informe Inicial"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output: pdf_document
6 ---
7
8 '{r , warning=FALSE, message=FALSE, echo=FALSE}'
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15
16 # No hace falta poner '/' al final, ya se tiene en cuenta luego
17 ruta_datos <- "C:/Users/ismae/Desktop/TFG EST/data/modificados"
18 ''
19
20 # Datos de CT Bombas
21
22 '{r, echo = FALSE, results = "hide"}'
23 ruta_bombas <- paste(ruta_datos, "/CT-bombas.xlsx", sep="")
24 bombas <- read_excel(ruta_bombas)
25 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
26 FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora), format="%Y-%
    m-%d %H:%M:%S")
27 bombas<-cbind(FechaHora, bombas)
28 head(bombas)
29 ''
30
31 # Datos de CT Nuevo
32
33 '{r, echo = FALSE, results = "hide"}'
34 ruta_nuevo <- paste(ruta_datos, "/CT-nuevo.xlsx", sep="")
35 nuevo <- read_excel(ruta_nuevo)
36 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
37 FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="%Y-%m
    -%d %H:%M:%S")
38 nuevo<-cbind(FechaHora, nuevo)

```

```

39 head(nuevo)
40 ""
41
42
43 ""{r}
44 bombas <- as.data.frame(bombas)
45 bombas_sin_fecha<-bombas[,-c(1,2,3)]
46 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
47   , 2, var, na.rm = TRUE) == 0]
48 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
49   cols_cero_var_bombas]
50 ""
51
52 # Datos irradiancia
53 ""{r, echo = FALSE, results = "hide"}
54 ruta_irradiancia <- paste(ruta_datos, "/irradiancia.xlsx", sep="")
55 irradiancia <- read_excel(ruta_irradiancia)
56 head(irradiancia)
57 ""
58
59
60 # Comienzo de analisis con CT bombas
61
62 ## Corelaciones
63 ""{r}
64 correlaciones <- sapply(bombas_filtrado, function(x) cor(x,
65   irradiancia$Irradiancia_CTBombas, use = "complete.obs"))
66 corr_irr <- data.frame(correlacion = correlaciones) %>% arrange(desc(
67   abs(correlacion)))
68 corr_irr
69 ""
70
71 ""{r}
72 correlaciones <- sapply(bombas_filtrado, function(x) cor(x,
73   irradiancia$Temperatura, use = "complete.obs"))
74 corr_temp <- data.frame(correlacion = correlaciones) %>% arrange(desc(
75   abs(correlacion)))
76 corr_temp
77 ""
78
79 ""{r}
80 cor_irr_temp <- cor(irradiancia$Irradiancia_CTBombas,
81   irradiancia$Temperatura, use = "complete.obs")
82 cor_irr_temp
83 ""
84
85 ## Dataframe conjunto para los modelos
86

```

```

83  '''{r}
84  bombas_modelos <- cbind(irradiancia[,c(3,4)], bombas_filtrado)
85  head(bombas_modelos)
86  '''
87
88
89
90  ## Algunos de los modelos
91
92
93  '''{r}
94  modelo1 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas,
95               data = bombas_modelos)
96  modelo2 <- lm('Arm nicos Tensi n5 L1N Med' ~ Temperatura, data =
97               bombas_modelos)
98  modelo3 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas +
99               Temperatura, data = bombas_modelos)
100  modelo4 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas *
101               Temperatura, data = bombas_modelos)
102  '''
103
104
105  '''{r}
106  anova(modelo1, modelo3, modelo4)
107  '''
108
109
110
111
112
113
114
115
116
117
118  # Comienzo de analisis con CT nuevo
119
120  '''{r}
121  nuevo_sin_na <- nuevo_filtrado
122
123  nuevo_sin_na[] <- lapply(nuevo_sin_na, function(col) {
124    if (is.numeric(col)) {
125      col[is.na(col)] <- mean(col, na.rm = TRUE)
126    }
127    return(col)
128  })
129  '''

```

```

130
131
132 ## Corelaciones
133 ```{r}
134 correlaciones <- sapply(nuevo_sin_na, function(x) cor(x,
135   irradiancia$Irradiancia_CTNuevo, use = "complete.obs"))
136 corr_irr <- data.frame(correlacion = correlaciones) %>% arrange(desc(
137   abs(correlacion)))
138
139
140 ```{r}
141 correlaciones <- sapply(nuevo_sin_na, function(x) cor(x,
142   irradiancia$Temperatura, use = "complete.obs"))
143 corr_temp <- data.frame(correlacion = correlaciones) %>% arrange(desc(
144   abs(correlacion)))
145
146
147 ```{r}
148 cor_irr_temp <- cor(irradiancia$Irradiancia_CTNuevo,
149   irradiancia$Temperatura, use = "complete.obs")
150 cor_irr_temp
151
152 ## Dataframe conjunto para los modelos
153
154 ```{r}
155 nuevo_modelos <- cbind(irradiancia[,c(2,4)], nuevo_sin_na)
156 head(nuevo_modelos)
157
158
159 ## Algunos de los modelos
160
161
162 ```{r}
163 modelo1 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo, data =
164   nuevo_modelos)
165 modelo2 <- lm('Desequilibrio Vn Min' ~ Temperatura, data =
166   nuevo_modelos)
167 modelo3 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo +
168   Temperatura, data = nuevo_modelos)
169 modelo4 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo *
170   Temperatura, data = nuevo_modelos)
171
172 ```{r}
173 anova(modelo1, modelo3, modelo4)

```

```

172  ‘‘‘
173
174  ‘‘‘{r}
175  anova(modelo2, modelo3, modelo4)
176  ‘‘‘
177
178
179
180  ‘‘‘{r}
181  # Nos aseguramos de tener solo columnas num ricas
182  indicadores_num <- nuevo_sin_na[sapply(nuevo_sin_na, is.numeric)]
183
184  # Variables explicativas
185  irr <- irradiancia$Irradiancia_CTNuevo
186  temp <- irradiancia$Temperatura
187
188  # Creamos una lista para almacenar resultados
189  resultados <- data.frame(
190    Indicador = character(),
191    R2_Irr = numeric(),
192    p_Irr = numeric(),
193    R2_Temp = numeric(),
194    p_Temp = numeric(),
195    R2_Mult = numeric(),
196    p_Irr_Mult = numeric(),
197    p_Temp_Mult = numeric(),
198    stringsAsFactors = FALSE
199  )
200
201  # Bucle por cada indicador
202  for (var in names(indicadores_num)) {
203    y <- indicadores_num[[var]]
204
205    # Regresi n simple: Irradiancia
206    mod_irr <- lm(y ~ irr)
207    summ_irr <- summary(mod_irr)
208
209    # Regresi n simple: Temperatura
210    mod_temp <- lm(y ~ temp)
211    summ_temp <- summary(mod_temp)
212
213    # Regresi n m ltiple
214    mod_mult <- lm(y ~ irr + temp)
215    summ_mult <- summary(mod_mult)
216
217    # Guardamos los resultados
218    resultados <- rbind(resultados, data.frame(
219      Indicador = var,
220      R2_Irr = summ_irr$r.squared,
221      p_Irr = coef(summ_irr)[2, 4],
222      R2_Temp = summ_temp$r.squared,

```

```

223     p_Temp = coef(summ_temp)[2, 4],
224     R2_Mult = summ_mult$r.squared,
225     p_Irr_Mult = coef(summ_mult)["irr", 4],
226     p_Temp_Mult = coef(summ_mult)["temp", 4]
227 ))
228 }
229
230 # Ordenamos por R    m ltiple descendente
231 resultados <- resultados[order(-resultados$R2_Mult), ]
232
233 # Mostramos la tabla
234 print(resultados)
235
236 ‘ ‘ ‘

```

Listing C.6: Primeros modelos

C.0.7. Chequeo final datos iniciales

Generación del informe de los datos tras ciertas decisiones.

```

1  ---
2  title: "Informe Inicial"
3  author: "Ismael Carbajo"
4  date: "r Sys.Date()"
5  output: pdf_document
6  ---
7
8  ‘ ‘ {r , warning=FALSE, message=FALSE, echo=FALSE}
9  library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15
16 # No hace falta poner '/' al final, ya se tiene en cuenta luego
17 #ruta_datos <-"C:/Users/ismae/Desktop/TFG EST/data/modificados"
18 ruta_datos <-"H:/TFG EST/data/modificados"
19 ‘ ‘ ‘
20
21
22
23 # Datos de CT Bombas
24
25 ‘ ‘ {r, echo = FALSE, results = "hide"}
26 ruta_bombas <- paste(ruta_datos, "/CT-bombas.xlsx", sep="")
27 bombas <- read_excel(ruta_bombas)
28 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")

```

```

29 FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora), format="%Y-%
    m-%d %H:%M:%S")
30 bombas<-cbind(FechaHora, bombas)
31 head(bombas)
32 '''
33
34
35 '''{r, echo = FALSE, results = "hide"}
36 colSums(is.na(bombas))[colSums(is.na(bombas)) > 0] # N mero de NA's
37 anyNA(bombas)
38 '''
39
40 '''{r, echo = FALSE, results = "hide"}
41 bombas <- as.data.frame(bombas)
42 bombas_sin_fecha<-bombas[,-c(1,2,3)]
43 '''
44
45
46 '''{r, echo=FALSE}
47 cat("Fecha y hora inicio de los datos: ",format(bombas$FechaHora[1], "
    %Y-%m-%d %H:%M:%S"), "\n")
48 cat("Fecha y hora final de los datos: ",format(bombas$FechaHora[length
    (bombas$FechaHora)], "%Y-%m-%d %H:%M:%S"), "\n")
49 cat("\n", "
    -----
    ", "\n\n")
50 cat("N mero de variables recogidas: ", dim(bombas_sin_fecha)[2], "\n"
    )
51 cat("N mero de variables que en realidad son constantes: ", sum(apply
    (bombas_sin_fecha, 2, var)==0), "\n")
52 cat("N mero de verdaderas variables: ", sum(apply(bombas_sin_fecha,
    2, var)!=0), "\n")
53 cat("\n", "
    -----
    ", "\n\n")
54 cat("No hay valores ausentes", "\n")
55 '''
56
57
58 # Datos de CT Nuevo
59
60 '''{r, echo = FALSE, results = "hide"}
61 ruta_nuevo <- paste(ruta_datos, "/CT-nuevo.xlsx", sep="")
62 nuevo <- read_excel(ruta_nuevo)
63 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
64 FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="%Y-%m
    -%d %H:%M:%S")
65 nuevo<-cbind(FechaHora, nuevo)
66 head(nuevo)
67 '''
68

```

```

69
70 ```{r, echo = FALSE, results = "hide"}
71 colSums(is.na(nuevo))[colSums(is.na(nuevo)) > 0] # N mero de NA's
72 anyNA(nuevo)
73 ```
74
75 ```{r, echo = FALSE, results = "hide"}
76 which(is.na(nuevo$'Arm nicos Corriente15 N Max'))
77 col_na_579<-which(is.na(nuevo[579,]))
78 col_na_1954<-which(is.na(nuevo[1954,]))
79 col_na_2595<-which(is.na(nuevo[2595,]))
80 col_na_2609<-which(is.na(nuevo[2609,]))
81 colnames(nuevo)[col_na_579]
82 colnames(nuevo)[col_na_1954]
83 colnames(nuevo)[col_na_2595]
84 colnames(nuevo)[col_na_2609]
85 ```
86
87 ```{r, echo = FALSE, results = "hide"}
88 nuevo<-as.data.frame(nuevo)
89 nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
90 sum(apply(nuevo_sin_fecha, 2, var, na.rm = TRUE)==0)
91 ```
92
93 ```{r, echo=FALSE}
94 cat("Fecha y hora inicio de los datos: ",format(nuevo$FechaHora[1], "%
95 Y-%m-%d %H:%M:%S"), "\n")
96 cat("Fecha y hora final de los datos: ",format(nuevo$FechaHora[length(
97 nuevo$FechaHora)], "%Y-%m-%d %H:%M:%S"), "\n")
98 cat("\n", "
99 -----
100 ", "\n\n")
101 cat("N mero de variables recogidas: ", dim(nuevo_sin_fecha)[2], "\n")
102 cat("N mero de variables que en realidad son constantes: ", sum(apply
103 (nuevo_sin_fecha, 2, var, na.rm = TRUE)==0), "\n")
104 cat("N mero de verdaderas variables: ", sum(apply(nuevo_sin_fecha, 2,
105 var, na.rm = TRUE)!=0), "\n")
106 cat("\n", "
107 -----
108 ", "\n\n")
109 cat("4 valores ausentes", "\n\n")
110 cat("Las observaciones:",which(is.na(nuevo$'Pst L1N')), "\n")
111 cat("\nCorrespondientes a los tiempos: \n")
112 nuevo$FechaHora[c(which(is.na(nuevo$'Pst L1N')))]
113 cat("\nLos NA se encuentran en las mismas variables en las 4
114 observaciones:", "\n")
115 colnames(nuevo)[col_na_579]
116 ```
117
118
119 # Correcci n de dos nombres de variables para las coincidencias
120

```

```

111 Al ejecutar el código, se tiene esta salida, se cambia a mano dentro
    de los archivos para no volver a tener el problema de nuevo:
112 [1] "Arm nicos Corriente13 L1 M n." "Arm nicos Corriente13 L1 M x."
    " "Arm nicos Corriente13 L1 Min"
113 [4] "Arm nicos Corriente13 L1 Max"
114
115 ```{r}
116 no_coinciden <- union(
117   setdiff(colnames(bombas_sin_fecha), colnames(nuevo_sin_fecha)),
118   setdiff(colnames(nuevo_sin_fecha), colnames(bombas_sin_fecha))
119 )
120 no_coinciden
121 ```
122
123
124
125 # Estudio de las variables que son constantes
126
127 ```{r, echo = FALSE, results = "hide"}
128 cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
129   2, var, na.rm = TRUE) == 0]
129 nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
130   cols_cero_var_nuevo]
130 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
131   , 2, var, na.rm = TRUE) == 0]
131 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
132   cols_cero_var_bombas]
132 ```
133
134
135 ## Cuales son esas variables
136
137 ```{r, echo=FALSE}
138 kable(data.frame(Nombres = cols_cero_var_nuevo), caption = "Variables
139   que son constantes de 'CT nuevo'")
139 ```
140
141 ```{r, echo=FALSE}
142 kable(data.frame(Nombres = cols_cero_var_bombas), caption = "Variables
143   que son constantes de 'CT bombas'")
143 ```
144
145 ## Las constantes de 'CT nuevo' estan en 'CT bombas'?
146
147
148 Veamos ahora que las 3 constantes si estan en 'CT bombas' pero no son
    constantes en este:
149 ```{r, echo=FALSE}
150 intersect(cols_cero_var_nuevo, colnames(bombas))
151 ```
152

```

```

153 ## Las constantes de 'CT bombas' estan en 'CT nuevo'?
154
155 Ya hemos visto que ninguna constante es com n a los dos datasets, por
    tanto se busca si las constantes de 'CT bombas' aparecen como
    variables en 'CT nuevo':
156 ```{r, echo=FALSE}
157 intersecciones_cte_bombas_con_nuevo<-intersect(cols_cero_var_bombas,
    colnames(nuevo))
158 cat("Contamos con ",length(intersecciones_cte_bombas_con_nuevo), " que
    si aparecen")
159 ```
160
161
162 ```{r, echo=FALSE}
163 kable(data.frame(Nombres= intersecciones_cte_bombas_con_nuevo),
    caption = "Constantes de 'CT bombas' que son variables en 'CT nuevo
    ")
164 ```
165
166
167 # Cuantas variables estan en ambos
168
169
170 ## Si consideramos todas las variables que contiene el dataset
171 ```{r, echo=FALSE}
172 coincidencias<-intersect(colnames(bombas_sin_fecha),colnames(
    nuevo_sin_fecha))
173 cat("Contamos con un total de ", length(coincidencias), " variables
    que aparecen en ambos datasets." )
174 ```
175
176
177
178 ## Si consideramos solo aquellas que verdaderamente son variables (
    varianza distinta de 0)
179 ```{r, echo=FALSE}
180 coincidencias<-intersect(colnames(bombas_filtrado),colnames(
    nuevo_filtrado))
181 cat("Contamos con un total de ", length(coincidencias), " variables
    que aparecen en ambos datasets." )
182 ```
183
184
185 ```{r, echo=FALSE}
186 kable(coincidencias, caption = "Variables verdaderas comunes a ambos
    datasets")
187 ```
188
189
190
191

```

```

192 |
193 |
194 | # Tabla resumen
195 | ```{r, echo = FALSE}
196 | resumen<-data.frame(
197 |   "CT Bombas" = c(sum(apply(bombas_sin_fecha, 2, var)==0),length(
198 |     bombas_filtrado), length(coincidencias), length(bombas_filtrado)-
199 |     length(coincidencias)),
200 |   "CT Nuevo" = c(sum(apply(nuevo_sin_fecha, 2, var, na.rm = TRUE)==0),
201 |     length(nuevo_filtrado),length(coincidencias), length(
202 |     nuevo_filtrado)-length(coincidencias)))
203 | rownames(resumen) <- c("Num constantes", "Num verdaderasVariables", "
204 |   Num variables comunes", "Num variables unicas")
205 | kable(resumen, caption = "Resumen sobre verdaderas variables")
206 | ```
207 |
208 | # Correlaciones de las variables en CT bombas
209 |
210 | ```{r, echo=FALSE, warning=FALSE}
211 | library(ggplot2)
212 | library(GGally)
213 | library(corrplot)
214 |
215 | cor_matrix <- cor(bombas_filtrado, use = "complete.obs")
216 |
217 | corrplot(cor_matrix, method = "color",
218 |   tl.cex = 0.1,
219 |   addCoef.col = NULL,
220 |   col = colorRampPalette(c("blue", "white", "red"))(200),
221 |   order = "hclust")
222 |
223 | ```
224 |
225 | ```{r}
226 | cor_matrix <- cor(nuevo_filtrado, use = "complete.obs")
227 |
228 | corrplot(cor_matrix, method = "color",
229 |   tl.cex = 0.1,
230 |   addCoef.col = NULL,
231 |   col = colorRampPalette(c("blue", "white", "red"))(200),
232 |   order = "hclust")
233 |
234 | ```
235 |
236 | # Correlaciones de las variables en CT bombas
237 |
238 | ```{r, echo=FALSE}
239 | library(dplyr)
240 | library(tidyr)
241 |
242 | cor_matrix <- cor(bombas_filtrado, use = "complete.obs")

```

```

238 cor_long <- as.data.frame(as.table(cor_matrix))
239 cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
240 cor_long$abs_cor <- abs(cor_long$Freq)
241
242 umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
243 resultado <- sapply(umbral_list, function(threshold) {
244   sum(cor_long$abs_cor > threshold)
245 })
246
247 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
   = "")
248 resultado
249 ""
250
251
252 # Correlaciones de las variables en CT nuevo
253
254 "{r, echo=FALSE}
255 library(dplyr)
256 library(tidyr)
257
258 cor_matrix <- cor(nuevo_filtrado, use = "complete.obs")
259
260 cor_long <- as.data.frame(as.table(cor_matrix))
261 cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
262 cor_long$abs_cor <- abs(cor_long$Freq)
263
264 umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
265 resultado <- sapply(umbral_list, function(threshold) {
266   sum(cor_long$abs_cor > threshold)
267 })
268
269 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
   = "")
270 resultado
271 ""
272
273 # Analisis de correlaciones con nicamente las 95 variables comunes
274
275 ## En 'CT bombas'
276 "{r, echo=FALSE, warning=FALSE}
277 library(ggplot2)
278 library(GGally)
279 library(corrplot)
280
281 cor_matrix <- cor(bombas[,coincidencias], use = "complete.obs")
282
283 corrplot(cor_matrix, method = "color",
284   tl.cex = 0.1,
285   addCoef.col = NULL,
286   col = colorRampPalette(c("blue", "white", "red"))(200),

```

```
287     order = "hclust")
288
289   ' ' '
290
291   '{r, echo=FALSE}
292   library(dplyr)
293   library(tidyr)
294
295   cor_matrix <- cor(bombas[,coincidencias], use = "complete.obs")
296
297   cor_long <- as.data.frame(as.table(cor_matrix))
298   cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
299   cor_long$abs_cor <- abs(cor_long$Freq)
300
301   umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
302   resultado <- sapply(umbral_list, function(threshold) {
303     sum(cor_long$abs_cor > threshold)
304   })
305
306   names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
307     = " ")
308   resultado
309   ' ' '
310
311   ## En 'CT nuevo'
312   '{r, echo=FALSE, warning=FALSE}
313   library(ggplot2)
314   library(GGally)
315   library(corrplot)
316
317   cor_matrix <- cor(nuevo[,coincidencias], use = "complete.obs")
318
319   corrplot(cor_matrix, method = "color",
320     tl.cex = 0.1,
321     addCoef.col = NULL,
322     col = colorRampPalette(c("blue", "white", "red"))(200),
323     order = "hclust")
324
325   ' ' '
326
327   '{r, echo=FALSE}
328   library(dplyr)
329   library(tidyr)
330
331   cor_matrix <- cor(nuevo[,coincidencias], use = "complete.obs")
332
333   cor_long <- as.data.frame(as.table(cor_matrix))
334   cor_long <- cor_long[cor_long$Var1 != cor_long$Var2, ]
335   cor_long$abs_cor <- abs(cor_long$Freq)
336
337   umbral_list <- c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5)
```

```

337 resultado <- sapply(umbral_list, function(threshold) {
338   sum(cor_long$abs_cor > threshold)
339 })
340
341 names(resultado)<-paste(">",c(0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5),sep
342   ="")
343 resultado
344
345
346
347 ## Conclusiones
348
349 Hay una variable que todav a era distinta por un par de tildes, por
350   ello, los resultados de ahora no coinciden si se ejecuta el c digo
351   con los datasets subidos al teams.
352
353 El n mero de variables que existen entre las diferentes estaciones ha
354   aumentado, aunque sigue habiendo algunas que son propias de ambos.
355
356 La mayor a de constantes han sido eliminadas pero todavia quedan
357   algunas.
358
359 Se trabajar con estos datasets quitando las constantes en cada uno
360   de ellos como primera opci n.

```

Listing C.7: Chequeo final datos iniciales

C.0.8. Modelos bombas

Generación del informe con varios modelos de bombas de los datos iniciales, primeros GAM introducidos.

```

1 ---
2 title: "Análisis CT bombas"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output: pdf_document
6 ---
7
8 '{r , warning=FALSE, message=FALSE, echo=FALSE}'
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15 library(car)
16

```

```

17 # No hace falta poner '/' al final, ya se tiene en cuenta luego
18 #ruta_datos <-"C:/Users/ismae/Desktop/TFG EST/data/modificados"
19 ruta_datos <-"H:/TFG EST/data/modificados"
20 '''
21
22
23 # Datos irradiancia
24 ```{r, results = "hide"}
25 ruta_irradiancia <- paste(ruta_datos, "/irradiancia.xlsx", sep="")
26 irradiancia <- read_excel(ruta_irradiancia)
27 head(irradiancia)
28 '''
29
30
31 # Datos CT bombas
32
33 ```{r, results = "hide"}
34 ruta_bombas <- paste(ruta_datos, "/CT-bombas.xlsx", sep="")
35 bombas <- read_excel(ruta_bombas)
36 bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
37 FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora), format="%Y-%
    m-%d %H:%M:%S")
38 bombas<-cbind(FechaHora, bombas)
39 head(bombas)
40 '''
41
42
43 ```{r}
44 bombas <- as.data.frame(bombas)
45 bombas_sin_fecha<-bombas[,-c(1,2,3)]
46 cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
    , 2, var, na.rm = TRUE) == 0]
47 bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
    cols_cero_var_bombas]
48 '''
49
50
51 # Analisis
52
53 ## Corelaciones
54
55 ### Correlaciones con la irradiancia
56 ```{r}
57 correlaciones <- sapply(bombas_filtrado, function(x) cor(x,
    irradiancia$Irradiancia_CTBombas, use = "complete.obs"))
58 corr_irr <- data.frame(correlacion = correlaciones) %>% arrange(desc(
    abs(correlacion)))
59 corr_irr
60 '''
61
62

```

```

63 ### Correlaciones con la temperatura
64 ```{r}
65 correlaciones <- sapply(bombas_filtrado, function(x) cor(x,
66   irradiancia$Temperatura, use = "complete.obs"))
67 corr_temp <- data.frame(correlacion = correlaciones) %>% arrange(desc(
68   abs(correlacion)))
69 corr_temp
70 ```
71
72 ### Correlaci n entre irradiancia y temperatura
73
74 Parece que si tienen cierta correlaci n, igualmente, se realizar n
75   modelos con ambas para ver su comportamiento.
76 ```{r}
77 cor_irr_temp <- cor(irradiancia$Irradiancia_CTBombas,
78   irradiancia$Temperatura, use = "complete.obs")
79 cor_irr_temp
80 ```
81
82 ## Dataframe conjunto para los modelos
83
84 Se mete la irradiancia de CT bombas y la temperatura en los datos para
85   realizar los modelos de manera m s facil.
86 ```{r}
87 bombas_modelos <- cbind(irradiancia[,c(3,4)], bombas_filtrado)
88 ```
89
90 ## Diferentes modelos para la variable con mayor correlaci n en
91   irradiancia
92
93 Probando a realizar los 4 modelos posibles con las 2 variables
94   explicativas disponibles, poniendo como explicativa la variable con
95   mayor correlaci n.
96 ```{r}
97 modelo1 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas,
98   data = bombas_modelos)
99 modelo2 <- lm('Arm nicos Tensi n5 L1N Med' ~ Temperatura, data =
100   bombas_modelos)
101 modelo3 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas +
102   Temperatura, data = bombas_modelos)
103 modelo4 <- lm('Arm nicos Tensi n5 L1N Med' ~ Irradiancia_CTBombas *
104   Temperatura, data = bombas_modelos)
105 ```
106
107 Parece ser que el mejor modelo corresponde al m s complejo, ambas
108   variables y su interacci n. Aunque tengan una correlaci n

```

```

moderadamente alta como es 0.66 ambas juntas trabajan bien.
101
102   ```{r}
103   anova(modelo1, modelo3, modelo4)
104   ```
105
106   ```{r}
107   anova(modelo2, modelo3, modelo4)
108   ```
109
110   ### An lisis de la multicolinealidad
111
112   Evaluando la presencia de la multicolinealidad, irradiancia y
       sobretodo la interacci n indican una colinealidad moderada, pues
       su valor de la inflaci n de la varianza se encuentra en 5-10. Al
       no ser una colinealidad muy severa de momento se considera tener en
       cuenta el modelo completo.
113   ```{r}
114   vif(modelo3)
115   vif(modelo4)
116   ```
117
118
119   ## Bucle con todos los modelos posibles
120
121   Se realiza ahora un bucle para hacer estos 4 modelos para todas las
       variables y guardar los resultados que nos interesan.
122
123   ```{r}
124   bombas_bucle <- bombas_modelos[,-c(1,2)]
125   indicadores_num <- bombas_bucle[sapply(bombas_bucle, is.numeric)]
126
127   irr <- irradiancia$Irradiancia_CTBombas
128   temp <- irradiancia$Temperatura
129
130   resultados <- c()
131
132   for (nombre_var in names(indicadores_num)) {
133     y <- indicadores_num[[nombre_var]]
134
135     # Irradiancia
136     mod_irr <- lm(y ~ irr)
137     summ_irr <- summary(mod_irr)
138
139     # Temperatura
140     mod_temp <- lm(y ~ temp)
141     summ_temp <- summary(mod_temp)
142
143     # Multiple
144     mod_mult <- lm(y ~ irr + temp)
145     summ_mult <- summary(mod_mult)

```

```

146
147 # Multiple con interaccion
148 mod_mult_int <- lm(y ~ irr * temp)
149 summ_mult_int <- summary(mod_mult_int)
150
151
152 resultados <- rbind(resultados, data.frame(
153   Indicador = nombre_var,
154   R2Irr = summ_irr$adj.r.squared,
155   pIrr = coef(summ_irr)[2, 4],
156   R2Temp = summ_temp$adj.r.squared,
157   pTemp = coef(summ_temp)[2, 4],
158   R2Mult = summ_mult$adj.r.squared,
159   pIrrMult = coef(summ_mult)["irr", 4],
160   pTempMult = coef(summ_mult)["temp", 4],
161   R2Multint = summ_mult_int$adj.r.squared,
162   pInt = coef(summ_mult_int)["irr:temp", 4]
163 ))
164 }
165
166 resultados <- resultados[order(-resultados$R2Multint), ]
167
168
169 Viendo los resultados, si bien es verdad que los R2 ajustados de los
    modelos no son muy grandes, muchos modelos parecen significativos y
    se pueden llegar a sacar ciertas conclusiones teniendo en cuenta
    el significado de las variables como tal. Se presentan los
    resultados (10 primeros) en dos tablas ordenadas por el R2 ajustado
    del modelo con interacci n:
170
171 ```{r}
172 knitr::kable(head(resultados[,c(1,2,3,4,5)],10), digits = 4, caption =
    "Resultados de las regresiones modelos simples")
173 ```
174
175 ```{r}
176 knitr::kable(head(resultados[, -c(2,3,4,5)],10), digits = 4, caption =
    "Resultados de las regresiones modelos complejos")
177 ```
178
179
180
181
182 ### Primeras conclusiones acerca de los resultados
183 Aunque ya se ha visto que hay correlacion entre 'irr' y 'temp', parece
    ser que el modelo con ambas mejora mucho la variabilidad explicada
    , e incluir la interacci n tambi n mejora pero en menor medida.
184
185 Puede que 'Tensi n de Medio Ciclo V RMS NG' y 'Vrms ph-n NG' sean las
    magnitudes m s afectadas, pues sus 3 variables asociadas de
    m nimo , m ximo y media se encuentran en el top 10 de los modelos

```

```

186     de interacci n.
187 Parece que solo el arm nico 5 es afectado en gran medida, pues es el
188     nico que aparece, pero solo su variable de media.
189
190
191 ## An lisis espec ficos para los primeros modelos
192
193
194 ### Arm nicos Tensi n5 L1N Med
195
196
197 ##### Modelo lineal con interacci n
198
199 El modelo considera intercept, ambas variables e interacci n como muy
200     significativas, adem s, el modelo en s tambi n es
201     significativo.
202
203 Viendo los gr ficos de residuos, lo primero que se ve es la gran
204     cantidad de datos acumulada en valores bajos.
205
206 Tambi n se puede llegar a apreciar cierta curvatura, lo que podr a
207     implicar no linealidad residual.
208
209 Parece que el comportamiento de los residuos es diferente desde ese
210     valor predicho modal en adelante, donde hay muchos menos datos, que
211     hacia atr s, donde se encuentran la mayor a de ellos.
212
213 ```{r}
214 modelo_AT5_med <- lm('Arm nicos Tensi n5 L1N Med' ~
215     Irradiancia_CTBombas * Temperatura, data = bombas_modelos)
216 summary(modelo_AT5_med)
217 plot(modelo_AT5_med)
218 ```
219
220 ##### Modelo GAM
221
222 Como se ha visto que la linealidad podr a llegar a ser un problema,
223     se modelan relaciones no lineales a partir de funciones suaves de
224     los predictores.
225
226 Viendo los resultados del summary:
227
228 R2 ajustado aumenta en gran medida, llegando a m s del 50%.
229
230 Deviance explicada tambi n mayor al 50%.
231
232 Tanto irradiancia como temperatura son significativos.
233
234 GCV bajo, indicando que no hay sobreajuste claramente.

```

```

226
227 Viendo las gráficas:
228
229 -irr
230 Al principio, el valor de los armónicos aumentan rápidamente con
    irradiancia. Luego, la pendiente se reduce, y vuelve a aumentar al
    final. Esto podría estar indicando efectos umbral o saturación,
    es decir, que solo a partir de cierto valor de la irradiancia hay
    un efecto o que lo hay hasta cierto valor.
231
232 -temp
233 La forma ondulada podría reflejar que la temperatura afecta en
    ciertos rangos, pero no de forma constante. Puede que solo afecte
    cuando la temperatura es muy alta, cosa que puede llegar a tener
    sentido.
234
235
236
237 ```{r}
238 library(mgcv)
239 bombas_AT5<-bombas_modelos
240 names(bombas_AT5)[names(bombas_modelos) == "Armónicos Tensión L1N
    Med"] <- "AT5_L1N_Med"
241 modelo_gam_AT5 <- gam(AT5_L1N_Med ~ s(irr) + s(temp), data =
    bombas_AT5)
242 summary(modelo_gam_AT5)
243 plot(modelo_gam_AT5, pages = 1)
244 ```
245
246 Aadiendo la interacción lineal, se ve que no es significativa y que
    no aumenta la explicabilidad del modelo.
247 ```{r}
248 modelo_gam_AT5_int <- gam(AT5_L1N_Med ~ s(irr) + s(temp) + irr:temp,
    data = bombas_AT5)
249 summary(modelo_gam_AT5_int)
250 ```
251
252 Pero, si se añade una interacción no lineal, si es significativa y
    el modelo parece ajustar un poco mejor.
253 GCV aumenta de manera muy leve, por tanto seguimos sin problemas.
254 ```{r}
255 modelo_gam_AT5_int <- gam(AT5_L1N_Med ~ s(irr) + s(temp) + ti(irr,
    temp), data = bombas_AT5)
256 summary(modelo_gam_AT5_int)
257 ```
258
259
260 Las representaciones de la interacción indican que tiene su mayor
    efecto cuando hay mucha temperatura pero en cambio la irradiancia
    es prácticamente nula (de noche).
261 ```{r, warning=FALSE}

```

```

262 vis.gam(modelo_gam_AT5_int, view = c("irr", "temp"), plot.type = "
    persp")
263 vis.gam(modelo_gam_AT5_int, view = c("irr", "temp"), plot.type = "
    contour")
264
265
266
267
268
269 ## GAM para todas las variables
270
271 Aunque se puede seguir analizando las siguientes variables en el top,
    puede que tenga m s sentido aplicar este nuevo modelo que parece
    funcionar mejor a todas las variables del dataset y volver a
    realizar un nuevo ranking.
272
273
274 ```{r}
275 resultados_gam <- c()
276
277 for (nombre_var in names(indicadores_num)) {
278   y <- indicadores_num[[nombre_var]]
279
280   gam_mod <- gam(y ~ s(irr) + s(temp))
281   gam_mod_int <- gam(y ~ s(irr) + s(temp) + ti(irr, temp))
282   summ_gam <- summary(gam_mod)
283   summ_gam_int <- summary(gam_mod_int)
284
285   r2_adj <- summ_gam$r.sq
286   dev_exp <- summ_gam$dev.expl
287   gcv_val <- summ_gam$sp.criterion
288   r2_adj_int <- summ_gam_int$r.sq
289   dev_exp_int <- summ_gam_int$dev.expl
290   gcv_val_int <- summ_gam_int$sp.criterion
291
292   resultados_gam <- rbind(resultados_gam, data.frame(
293     Indicador = nombre_var,
294     R2Adj = round(r2_adj, 4),
295     DevExp = round(dev_exp * 100, 2),
296     GCV = round(gcv_val, 4),
297     R2AdjInt = round(r2_adj_int, 4),
298     DevExpInt = round(dev_exp_int * 100, 2),
299     GCVInt = round(gcv_val_int, 4)
300   ))
301 }
302
303 ```
304
305
306 Los mejores 10 viendo el R2 del modelo sin interacci n.
307 ```{r}

```

```

308 resultados_gam_sin <- resultados_gam[order(-resultados_gam$R2Adj), ]
309 head(resultados_gam_sin, 10)
310 '''
311
312
313 Los mejores 10 viendo el R2 del modelo con interacci n.
314 '''{r}
315 resultados_gam_con <- resultados_gam[order(-resultados_gam$R2AdjInt),
316 ]
317 head(resultados_gam_con, 10)
318 '''
319
320
321 # Tabla de nombres
322
323 Tras todo el analisis realizado hasta el momento, se pueden sacar los
324   top 10 nombres de variables con mayores correlaciones con
325   irradiancia y temperatura, mejores resultados de los modelos
326   lineales (con interacci n) y los resultados de los modelos GAM (
327   sin y con interacci n).
328
329 '''{r}
330 nombres<-cbind(row.names(corr_irr),row.names(corr_temp),
331   resultados$Indicador,resultados_gam_sin$Indicador,
332   resultados_gam_con$Indicador)
333 colnames(nombres)<-c("Corr irr","Corr temp","LM","GAM sin","GAM con")
334 knitr::kable(head(nombres,10), caption = "TOP 10 nombres de diferentes
335   resultados")
336 '''
337
338
339 De esto podemos ver que los arm nicos de orden 5 y las corrientes de
340   neutro, son especialmente sensibles a las condiciones ambientales,
341   es decir, son las mejores explicadas por la temperatura y la
342   irradiancia. La incorporaci n de modelos GAM revela relaciones no
343   lineales e interacciones entre irradiancia y temperatura que no
344   eran evidentes en modelos lineales o correlaciones simples,
345   haciendo que variables que no aparecen con las m s correlacionadas
346   luego si aparezcan en los modelos y viceversa.

```

Listing C.8: Modelos bombas

C.0.9. Modelos nuevo

Generaci3n del informe con varios modelos de nuevo de los datos iniciales, primeros GAM introducidos.

```

1 ---
2 title: "Análisis CT nuevo"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output: pdf_document
6 ---
7
8
9 '{r , warning=FALSE, message=FALSE, echo=FALSE}
10 library(readxl)
11 library(factoextra)
12 library(dplyr)
13 library(knitr)
14 library(lubridate)
15 library(data.table)
16 library(car)
17
18 # No hace falta poner '/' al final, ya se tiene en cuenta luego
19 #ruta_datos <-"C:/Users/ismael/Desktop/TFG EST/data/modificados"
20 ruta_datos <-"H:/TFG EST/data/modificados"
21 ''
22
23
24 # Datos irradiancia
25 '{r, results = "hide"}
26 ruta_irradiancia <- paste(ruta_datos, "/irradiancia.xlsx", sep="")
27 irradiancia <- read_excel(ruta_irradiancia)
28 head(irradiancia)
29 ''
30
31
32 # Datos CT nuevo
33 '{r, results = "hide"}
34 ruta_nuevo <- paste(ruta_datos, "/CT-nuevo.xlsx", sep="")
35 nuevo <- read_excel(ruta_nuevo)
36 nuevo$Hora <- format(nuevo$Hora, "%H:%M:%S")
37 FechaHora <- as.POSIXct(paste(nuevo$Fecha, nuevo$Hora), format="%Y-%m
    -%d %H:%M:%S")
38 nuevo<-cbind(FechaHora, nuevo)
39 head(nuevo)
40 ''
41
42
43
44 '{r}
45 nuevo<-as.data.frame(nuevo)
46 nuevo_sin_fecha<-nuevo[,-c(1,2,3)]
47 cols_cero_var_nuevo <- names(nuevo_sin_fecha)[apply(nuevo_sin_fecha,
    2, var, na.rm = TRUE) == 0]
48 nuevo_filtrado <- nuevo_sin_fecha[, !names(nuevo_sin_fecha) %in%
    cols_cero_var_nuevo]

```

```

49  '''
50
51
52  Vemos las 3 variables con los 4 valores ausentes.
53  '''{r}
54  names(which(colSums(is.na(nuevo_filtrado)) > 0))
55  '''
56
57  Interpolamos los 4 valores ausentes a traves de un spline.
58  '''{r , warning=FALSE}
59  library(zoo)
60  nuevo_filtrado <- as.data.frame(lapply(nuevo_filtrado, na.spline),
61    check.names = FALSE)
62  '''
63
64  Ya no tenemos valores ausentes.
65  '''{r}
66  names(which(colSums(is.na(nuevo_filtrado)) > 0))
67  '''
68
69  ## Corelaciones
70  '''{r}
71  correlaciones <- sapply(nuevo_filtrado, function(x) cor(x,
72    irradiancia$Irradiancia_CTNuevo, use = "complete.obs"))
73  corr_irr <- data.frame(correlacion = correlaciones) %>% arrange(desc(
74    abs(correlacion)))
75  corr_irr
76  '''
77  '''{r}
78  correlaciones <- sapply(nuevo_filtrado, function(x) cor(x,
79    irradiancia$Temperatura, use = "complete.obs"))
80  corr_temp <- data.frame(correlacion = correlaciones) %>% arrange(desc(
81    abs(correlacion)))
82  corr_temp
83  '''
84  Vemos que la correlacion en este CT es mayor incluso que en bombas (
85    que ten a un 0.67 aprox).
86  '''{r}
87  cor_irr_temp <- cor(irradiancia$Irradiancia_CTNuevo,
88    irradiancia$Temperatura, use = "complete.obs")
89  cor_irr_temp
90  '''
91  ## Dataframe conjunto para los modelos
92
93  Se mete la irradiancia de CT nuevo y la temperatura en los datos para
94    realizar los modelos de manera m s facil.

```

```

92  '''{r}
93  nuevo_modelos <- cbind(irradiancia[,c(2,4)], nuevo_filtrado)
94  '''
95
96
97  ## Algunos de los modelos
98
99
100  Probando a realizar los 4 modelos posibles con las 2 variables
      explicativas disponibles, poniendo como explicativa la variable con
      mayor correlaci n.
101
102  '''{r}
103  modelo1 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo, data =
      nuevo_modelos)
104  modelo2 <- lm('Desequilibrio Vn Min' ~ Temperatura, data =
      nuevo_modelos)
105  modelo3 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo +
      Temperatura, data = nuevo_modelos)
106  modelo4 <- lm('Desequilibrio Vn Min' ~ Irradiancia_CTNuevo *
      Temperatura, data = nuevo_modelos)
107  '''
108
109
110  Parece ser que el mejor modelo corresponde al m s complejo, ambas
      variables y su interacci n. Aunque tengan una correlaci n alta
      como es 0.76 ambas juntas trabajan bien.
111
112  '''{r}
113  anova(modelo1, modelo3, modelo4)
114  '''
115
116  '''{r}
117  anova(modelo2, modelo3, modelo4)
118  '''
119
120  ### An lisis de la multicolinealidad
121
122  Evaluando la presencia de la multicolinealidad, irradiancia y
      sobretodo la interacci n indican una colinealidad alta, pues su
      valor de la inflaci n de la varianza se encuentra en >10. Al
      contrario que en bombas, ahora si tenemos una colinealidad bastante
      mayor y con un valor problem tico. Se va a seguir realizando
      igualmente el an lisis paralelo, pero hay que tener muy en cuenta
      esto m s adelante.
123  '''{r}
124  vif(modelo3)
125  vif(modelo4)
126  '''
127
128

```

```

129 ## Bucle con todos los modelos posibles
130
131 Se realiza ahora un bucle para hacer estos 4 modelos para todas las
    variables y guardar los resultados que nos interesan.
132
133
134 ““{r}
135 indicadores_num <- nuevo_filtrado[sapply(nuevo_filtrado, is.numeric)]
136
137 irr <- irradiancia$Irradiancia_CTNuevo
138 temp <- irradiancia$Temperatura
139
140 resultados <- c()
141
142 for (nombre_var in names(indicadores_num)) {
143   y <- indicadores_num[[nombre_var]]
144
145   # Irradiancia
146   mod_irr <- lm(y ~ irr)
147   summ_irr <- summary(mod_irr)
148
149   # Temperatura
150   mod_temp <- lm(y ~ temp)
151   summ_temp <- summary(mod_temp)
152
153   # Multiple
154   mod_mult <- lm(y ~ irr + temp)
155   summ_mult <- summary(mod_mult)
156
157   # Multiple con interaccion
158   mod_mult_int <- lm(y ~ irr * temp)
159   summ_mult_int <- summary(mod_mult_int)
160
161
162   resultados <- rbind(resultados, data.frame(
163     Indicador = nombre_var,
164     R2Irr = summ_irr$adj.r.squared,
165     pIrr = coef(summ_irr)[2, 4],
166     R2Temp = summ_temp$adj.r.squared,
167     pTemp = coef(summ_temp)[2, 4],
168     R2Mult = summ_mult$adj.r.squared,
169     pIrrMult = coef(summ_mult)["irr", 4],
170     pTempMult = coef(summ_mult)["temp", 4],
171     R2Multint = summ_mult_int$adj.r.squared,
172     pInt = coef(summ_mult_int)["irr:temp", 4]
173   ))
174 }
175
176 resultados <- resultados[order(-resultados$R2Multint), ]
177 ““
178

```

```

179
180 Viendo los resultados, si bien es verdad que los R2 ajustados de los
    modelos no son muy grandes, muchos modelos parecen significativos y
    se pueden llegar a sacar ciertas conclusiones teniendo en cuenta
    el significado de las variables como tal. Pero tenemos el problema
    de la multicolinealidad. Tambi n se ve que los R2 son ligeramente
    superiores que en bombas.
181
182 Se presentan los resultados (10 primeros) en dos tablas ordenadas por
    el R2 ajustado del modelo con interacci n:
183
184 ``{r}
185 knitr::kable(head(resultados[,c(1,2,3,4,5)],10), digits = 4, caption =
    "Resultados de las regresiones modelos simples")
186 ```
187
188 ``{r}
189 knitr::kable(head(resultados[, -c(2,3,4,5)],10), digits = 4, caption =
    "Resultados de las regresiones modelos complejos")
190 ```
191
192
193
194 ### Primeras conclusiones acerca de los resultados
195 Aunque ya se ha visto que hay correlacion entre 'irr' y 'temp', parece
    ser que el modelo con ambas mejora mucho la variabilidad explicada
    , e incluir la interacci n tambi n mejora pero en menor medida.
196
197 Puede que 'Arm nicos Tensi n5 L1N' y 'THD V L1N ' son las magnitudes
    m s afectadas, pues sus 3 variables asociadas de m nimo , m ximo
    y media se encuentran en el top 10 de los modelos de interacci n.
    No solo eso sino que las de la primera se encuentran como 1,2 y 4,
    siendo la m s afectada. Tambi n cabe decir que esta mimsa es la
    que mayores correlaciones y mejores resultados parece dar en el
    otro CT.
198
199
200
201 ## An lisis espec ficos para los primeros modelos
202
203
204
205 ### Arm nicos Tensi n5 L1N Med
206
207 Utilizando la variable que se utiliz en bombas (top 2 aqu ), para
    comparaci n.
208
209 ##### Modelo lineal con interacci n
210
211 Mismas conclusiones en esta parte, aunque R2 algo mayor:
212 El modelo considera intercept, ambas variables e interacci n como muy

```

significativas, adem s , el modelo en s tambi n es significativo.

Viendo los gr ficos de residuos, lo primero que se ve es la gran cantidad de datos acumulada en valores bajos.

Se vuelve a apreciar la misma curvatura que ocurria en el modelo espec fico del otro CT.

Parece que el comportamiento de los residuos es diferente desde ese valor predicho modal en adelante, donde hay muchos menos datos, que hacia atr s , donde se encuentran la mayor a de ellos.

```
““{r}
modelo_AT5_med <- lm('Arm nicos Tensi n5 L1N Med' ~
  Irradiancia_CTNuevo * Temperatura, data = nuevo_modelos)
summary(modelo_AT5_med)
plot(modelo_AT5_med)
““
```

Modelo GAM

Los resultados son pr cticamente iguales al modelo creado en CT bombas, exceptuando los gr ficos , que en este caso tenemos alg n cambio.

Viendo los resultados del summary:

R2 ajustado aumenta en gran medida, llegando a m s del 50%.

Deviance explicada tambi n mayor al 50%.

Tanto irradiancia como temperatura son significativos.

GCV bajo, indicando que no hay sobreajuste claramente.

Viendo las gr ficas:

-irr

Al principio, el valor de los arm nicos aumentan r pidamente con irradiancia. Luego, la pendiente se reduce, y vuelve a aumentar al final. Esto podr a estar indicando efectos umbral o saturaci n , es decir, que solo a partir de cierto valor de la irradiancia hay un efecto o que lo hay hasta cierto valor.

```

251 En comparaci n con bombas , ahora en vez de aumentar y luego
    permanecer relativamente estable va aumentando poco a poco de forma
    sinusoidal hasta los 600 donde empieza a decaer , pero esta parte
    final es m s dif cil de evaluar por sus intervalos m s anchos.
252
253 -temp
254 La forma ondulada podr a reflejar que la temperatura afecta en
    ciertos rangos , pero no de forma constante. Puede que solo afecte
    cuando la temperatura es muy alta , cosa que puede llegar a tener
    sentido.
255
256 En comparaci n con bombas , es bastante igual exceptuando que en los
    15 grados no tiene una peque a subida antes de seguir bajando.
257
258
259
260 ``{r}
261 library(mgcv)
262 nuevo_AT5<-nuevo_modelos
263 names(nuevo_AT5)[names(nuevo_modelos) == "Arm nicos Tensi n5 L1N Med
    "] <- "AT5_med"
264 modelo_gam_AT5_med <- gam(AT5_med ~ s(irr) + s(temp), data = nuevo_AT5
    )
265 summary(modelo_gam_AT5_med)
266 plot(modelo_gam_AT5_med, pages = 1)
267 ``
268
269
270
271 En este caso la interacci n lineal si es significativa . Pero como
    tiene una VIF alto seguramente se deba a eso , y no le veo el
    sentido tampoco tener que modelar en este caso este tipo de
    interacci n , sobretodo cuando ahora se ve que la interacci n no
    lineal da mejores resultados y en los gr ficos claramente se ve la
    no linealidad.
272 ``{r}
273 modelo_gam_AT5_med_int_lin <- gam(AT5_med ~ s(irr) + s(temp) + irr:
    temp, data = nuevo_AT5)
274 summary(modelo_gam_AT5_med_int_lin)
275 ``
276
277
278
279
280 A aadiendo una interacci n no lineal tambi n es significativa , m s
    que la lineal. Tambi n mejora la deviance explicada y el R2 ,
    incluso baja el GCv , indicando que este modelo puede ser mejor que
    el de la interacci n normal. Creo que tiene sentido , pues no usar
    un modelo lineal pero con una interacci n lineal no acaba de
    cuadrarme , por no hablar de la clara no linealidad de la
    interacci n que se ver ahora.

```

```

281  ““{r}
282  modelo_gam_AT5_med_int <- gam(AT5_med ~ s(irr) + s(temp) + ti(irr,temp
    ), data = nuevo_AT5)
283  summary(modelo_gam_AT5_med_int)
284  ““
285
286
287
288  Las representaciones de la interacci n indican que tiene su mayor
    efecto cuando hay mucha temperatura pero en cambio la irradiancia
    es pr cticamente nula (de noche). Adem s , al contrario que en CT
    bombas, cuando la irradiancia es alta y la temperatura baja, el
    sistema responde con valores bajos de esta variable.
289
290
291  ““{r, warning=FALSE}
292  vis.gam(modelo_gam_AT5_med_int, view = c("irr", "temp"), plot.type = "
    persp")
293  vis.gam(modelo_gam_AT5_med_int, view = c("irr", "temp"), plot.type = "
    contour")
294  ““
295
296
297
298
299
300  ### Arm nicos Tensi n5 L1N Max
301
302  Utilizando la variable top1 de los lineales. Da los mismos resultados
    que la media de esta misma medida que acabamos de ver. Puede que
    algun grafico algo diferente, pero no lo veo significativamente
    diferente.
303
304  #### Modelo lineal con interacci n
305
306  ““{r}
307  modelo_AT5_max <- lm(‘Arm nicos Tensi n5 L1N Max’ ~
    Irradiancia_CTNuevo * Temperatura, data = nuevo_modelos)
308  summary(modelo_AT5_max)
309  plot(modelo_AT5_max)
310  ““
311
312
313
314
315
316  #### Modelo GAM
317
318
319
320  ““{r}

```

```

321 library(mgcv)
322 nuevo_AT5<-nuevo_modelos
323 names(nuevo_AT5)[names(nuevo_modelos) == "Arm nicos Tensi n5 L1N Max
    "] <- "AT5_max"
324 modelo_gam_AT5_max <- gam(AT5_max ~ s(irr) + s(temp), data = nuevo_AT5
    )
325 summary(modelo_gam_AT5_max)
326 plot(modelo_gam_AT5_max, pages = 1)
327 '''
328
329
330 {{r}}
331 modelo_gam_AT5_max_int_lin <- gam(AT5_max ~ s(irr) + s(temp) + irr:
    temp, data = nuevo_AT5)
332 summary(modelo_gam_AT5_max_int_lin)
333 '''
334
335
336
337
338 {{r}}
339 modelo_gam_AT5_max_int <- gam(AT5_max ~ s(irr) + s(temp) + ti(irr,temp
    ), data = nuevo_AT5)
340 summary(modelo_gam_AT5_max_int)
341 '''
342
343
344 {{r, warning=FALSE}}
345 vis.gam(modelo_gam_AT5_max_int, view = c("irr", "temp"), plot.type = "
    persp")
346 vis.gam(modelo_gam_AT5_max_int, view = c("irr", "temp"), plot.type = "
    contour")
347 '''
348
349
350
351
352 ### THD V L1N Max
353
354 Resultados similares a las otras dos.
355
356 #### Modelo lineal con interacci n
357 {{r}}
358 modelo_THD_V_max <- lm('THD V L1N Max' ~ Irradiancia_CTNuevo *
    Temperatura, data = nuevo_modelos)
359 summary(modelo_THD_V_max)
360 plot(modelo_THD_V_max)
361 '''
362
363
364

```

```

365
366
367 ##### Modelo GAM
368
369
370 ‘‘{r}
371 library(mgcv)
372 nuevo_THD<-nuevo_modelos
373 names(nuevo_THD)[names(nuevo_modelos) == "THD V L1N Max"] <- "
  THD_V_L1N_Max"
374 modelo_gam_THD <- gam(THD_V_L1N_Max ~ s(irr) + s(temp), data =
  nuevo_THD)
375 summary(modelo_gam_THD)
376 plot(modelo_gam_THD, pages = 1)
377 ‘‘‘
378
379
380
381 ‘‘{r}
382 modelo_gam_THD_int_lin <- gam(THD_V_L1N_Max ~ s(irr) + s(temp) + irr:
  temp, data = nuevo_THD)
383 summary(modelo_gam_THD_int_lin)
384 ‘‘‘
385
386
387
388
389 ‘‘{r}
390 modelo_gam_THD_int <- gam(THD_V_L1N_Max ~ s(irr) + s(temp) + ti(irr,
  temp), data = nuevo_THD)
391 summary(modelo_gam_THD_int)
392 ‘‘‘
393
394
395
396 ‘‘{r, warning=FALSE}
397 vis.gam(modelo_gam_THD_int, view = c("irr", "temp"), plot.type = "
  persp")
398 vis.gam(modelo_gam_THD_int, view = c("irr", "temp"), plot.type = "
  contour")
399 ‘‘‘
400
401
402
403 ## GAM para todas las variables
404
405 Aunque se puede seguir analizando las siguientes variables en el top,
  puede que tenga m s sentido aplicar este nuevo modelo que parece
  funcionar mejor a todas las variables del dataset y volver a
  realizar un nuevo ranking.
406

```

```

407
408   '{r}
409 resultados_gam <- c()
410
411 for (nombre_var in names(indicadores_num)) {
412   y <- indicadores_num[[nombre_var]]
413
414   gam_mod <- gam(y ~ s(irr) + s(temp))
415   gam_mod_int <- gam(y ~ s(irr) + s(temp) + ti(irr, temp))
416   summ_gam <- summary(gam_mod)
417   summ_gam_int <- summary(gam_mod_int)
418
419   r2_adj <- summ_gam$r.sq
420   dev_exp <- summ_gam$dev.expl
421   gcv_val <- summ_gam$sp.criterion
422   r2_adj_int <- summ_gam_int$r.sq
423   dev_exp_int <- summ_gam_int$dev.expl
424   gcv_val_int <- summ_gam_int$sp.criterion
425
426   resultados_gam <- rbind(resultados_gam, data.frame(
427     Indicador = nombre_var,
428     R2Adj = round(r2_adj, 4),
429     DevExp = round(dev_exp * 100, 2),
430     GCV = round(gcv_val, 4),
431     R2AdjInt = round(r2_adj_int, 4),
432     DevExpInt = round(dev_exp_int * 100, 2),
433     GCVInt = round(gcv_val_int, 4)
434   ))
435 }
436
437   ''
438
439 Las tres variables de Arm nicos Tensi n5 L1N parecen ser las mejores
    explicadas, seguidas por THD V L1N Max, la cual tambi n estaba en
    el top de los lineales.
440
441
442 Los mejores 10 viendo el R2 del modelo sin interacci n.
443   '{r}
444 resultados_gam_sin <- resultados_gam[order(-resultados_gam$R2Adj), ]
445 head(resultados_gam_sin, 10)
446   ''
447
448
449 Los mejores 10 viendo el R2 del modelo con interacci n.
450   '{r}
451 resultados_gam_con <- resultados_gam[order(-resultados_gam$R2AdjInt),
    ]
452 head(resultados_gam_con, 10)
453   ''
454

```

```

455
456
457 # Tabla de nombres
458
459 Tras todo el analisis realizado hasta el momento, se pueden sacar los
    top 10 nombres de variables con mayores correlaciones con
    irradiancia y temperatura, mejores resultados de los modelos
    lineales (con interacci n) y los resultados de los modelos GAM (
    sin y con interacci n).
460
461 ```{r}
462 nombres<-cbind(row.names(corr_irr),row.names(corr_temp),
    resultados$Indicador,resultados_gam_sin$Indicador,
    resultados_gam_con$Indicador)
463 colnames(nombres)<-c("Corr irr","Corr temp","LM","GAM sin","GAM con")
464
465 knitr::kable(head(nombres,10), caption = "TOP 10 nombres de diferentes
    resultados")
466 ```
467
468 Tenemos conclusiones paralelas a CT bombas:
469
470
471 De esto podemos ver que los arm nicos de orden 5 y THD V L1N, son
    especialmente sensibles a las condiciones ambientales, es decir,
    son las mejores explicadas por la temperatura y la irradiancia. La
    incorporaci n de modelos GAM revela relaciones no lineales e
    interacciones entre irradiancia y temperatura que no eran evidentes
    en modelos lineales o correlaciones simples, haciendo que
    variables que no aparecen con las m s correlacionadas luego si
    aparezcan en los modelos y viceversa.
472
473
474
475
476
477 ## Agrupaci n de resultados por variables f sicas
478
479 Agrupamos los resultados segun la variable fisica (falta revision a
    mano, pues esto esta hecho de manera automatica y hay variables que
    faltan).
480
481 ```{r}
482 # Creamos la columna "Grupo" seg n el contenido del nombre del
    indicador
483 resultados_gam_con$Grupo <- case_when(
    grepl("Arm nicos", resultados_gam_con$Indicador, ignore.case = TRUE
    ) ~ "Arm nicos",
484 grepl("THD", resultados_gam_con$Indicador, ignore.case = TRUE) ~ "
    THD",
485 grepl("Tensi n", resultados_gam_con$Indicador, ignore.case = TRUE)
    ~ "Tensi n",

```

```

486   grepl("Corriente", resultados_gam_con$Indicador, ignore.case = TRUE)
487     ~ "Corriente",
488   grepl("Potencia", resultados_gam_con$Indicador, ignore.case = TRUE)
489     ~ "Potencia",
490   grepl("Desequilibrio", resultados_gam_con$Indicador, ignore.case =
491     TRUE) ~ "Desequilibrio",
492   grepl("Cos Phi|Factor de Potencia", resultados_gam_con$Indicador,
493     ignore.case = TRUE) ~ "Factor Potencia",
494   grepl("Frecuencia", resultados_gam_con$Indicador, ignore.case = TRUE
495     ) ~ "Frecuencia",
496   grepl("Pst|Plt", resultados_gam_con$Indicador, ignore.case = TRUE) ~
497     "Flicker",
498   TRUE ~ "Otros"
499 )
500 '''
501
502   {{r}}
503   resumen_grupos <- resultados_gam_con %>%
504     group_by(Grupo) %>%
505     summarise(
506       R2_GAM_con_int = mean(R2AdjInt),
507       R2_GAM_sin_int = mean(R2Adj),
508       .groups = "drop"
509     ) %>%
510     arrange(desc(R2_GAM_con_int))
511
512   knitr::kable(resumen_grupos, caption = "Resumen por variables fisicas"
513     )
514   '''
515
516   ## Agrupaci n de resultados por cada trio max, med, min
517
518   {{r}}
519   # Crear nueva columna con nombre base sin Min, Med o Max
520   resultados_gam_con$VariableBase <- resultados_gam_con$Indicador |>
521     gsub(" Max$", "", x = _, ignore.case = TRUE) |>
522     gsub(" Med$", "", x = _, ignore.case = TRUE) |>
523     gsub(" Min$", "", x = _, ignore.case = TRUE)
524   '''
525
526
527
528
529

```

```

530  ““{r}
531
532  agrupados <- resultados_gam_con %>%
533    group_by(VariableBase) %>%
534    summarise(
535      R2_GAM_con_int = mean(R2AdjInt),
536      R2_GAM_sin_int = mean(R2Adj)
537    ) %>%
538    arrange(desc(R2_GAM_con_int))
539
540  knitr::kable(agrupados, caption = "Resumen por trio de variables")
541  ““

```

Listing C.9: Modelos nuevo

C.0.10. Top5 bombas

Generación del informe de modelos específicos de modelos bombas.

```

1  ---
2  title: "Top 5 bombas"
3  author: "Ismael Carbajo Valor"
4  date: "2025-05-20"
5  output: pdf_document
6  ---
7
8  ““{r , warning=FALSE, message=FALSE, echo=FALSE}
9  library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15 library(car)
16
17 # No hace falta poner '/' al final, ya se tiene en cuenta luego
18 #ruta_datos <-"C:/Users/ismae/Desktop/TFG EST/data/modificados"
19 ruta_datos <-"H:/TFG EST/data/modificados"
20 ““
21
22
23 # Datos irradiancia
24 ““{r, results = "hide"}
25 ruta_irradiancia <- paste(ruta_datos, "/irradiancia.xlsx", sep="")
26 irradiancia <- read_excel(ruta_irradiancia)
27 head(irradiancia)
28 ““
29
30 ““{r}
31 irradiancia[!complete.cases(irradiancia), ]

```

```
32  '''
33
34  Interpolamos el valor ausente a traves de un spline.
35  '''{r , warning=FALSE}
36  library(zoo)
37  irradiancia <- as.data.frame(lapply(irradiancia, na.spline),check.
    names = FALSE)
38  '''
39
40  '''{r}
41  irradiancia[!complete.cases(irradiancia), ]
42  '''
43
44
45  # Datos CT bombas
46
47  '''{r, results = "hide"}
48  ruta_bombas <- paste(ruta_datos, "/CT-bombas.xlsx",sep="")
49  bombas <- read_excel(ruta_bombas)
50  bombas$Hora <- format(bombas$Hora, "%H:%M:%S")
51  FechaHora <- as.POSIXct(paste(bombas$Fecha, bombas$Hora), format="%Y-%
    m-%d %H:%M:%S")
52  bombas<-cbind(FechaHora, bombas)
53  head(bombas)
54  '''
55
56
57  '''{r}
58  bombas <- as.data.frame(bombas)
59  bombas_sin_fecha<-bombas[,-c(1,2,3)]
60  cols_cero_var_bombas <- names(bombas_sin_fecha)[apply(bombas_sin_fecha
    , 2, var, na.rm = TRUE) == 0]
61  bombas_filtrado <- bombas_sin_fecha[, !names(bombas_sin_fecha) %in%
    cols_cero_var_bombas]
62  bombas_modelos <- cbind(irradiancia[,c(3,4)], bombas_filtrado)
63  '''
64
65
66  '''{r}
67  names(bombas_modelos)[names(bombas_modelos) == "Irradiancia_CTBombas"]
    <- "irr"
68  names(bombas_modelos)[names(bombas_modelos) == "Temperatura"] <- "temp
    "
69  '''
70
71
72  '''{r}
73  irr<-bombas_modelos$irr
74  temp<-bombas_modelos$temp
75  '''
76
```

```

77  "{r}
78  dia_semana <- as.factor(weekdays(FechaHora))
79  table(dia_semana)
80  ""
81
82
83
84  # Mejores modelos lineales con interacci n
85
86  La multicolinealidad no se considera excesiva ([5,10]) por tanto se
    pueden ver los modelos lineales.
87  "{r}
88  modelo_para_vif <- lm('Pst1min L1N' ~ irr * temp, data =
    bombas_modelos)
89  vif(modelo_para_vif)
90  ""
91
92
93  ## Arm nicos Tensi n5 L1N Med
94
95  "{r}
96  names(bombas_modelos)[names(bombas_modelos) == "Arm nicos Tensi n5
    L1N Med"] <- "AT5_Med"
97  ""
98
99
100  Grafico para ver un poco su relaci n , se utiliza el logaritmo de la
    irradiancia para que los valores sean parecidos y se analize mejor.
101  "{r}
102  plot(bombas_modelos$AT5_Med~FechaHora, type="l", col="blue")
103  lines(log10(bombas_modelos$irr)~FechaHora, col="red")
104  ""
105
106
107  El modelo lineal indica que todos los parametros son significativos.
    Se cuenta con un R2 ajustado de 0.3878 y el modelo en s tambi n
    es significativo.
108  "{r}
109  mod_lin_AT5_med <- lm(AT5_Med ~ irr * temp, data = bombas_modelos)
110  summary(mod_lin_AT5_med)
111  ""
112
113  Como ya se analiz antes, los residuos podr an llegar a ser no
    lineales, pero lo m s destacable aqu es el aparente umbral que
    hay a partir de cierto valor en el cual el comportamiento de los
    errores cambia.
114  "{r}
115  plot(mod_lin_AT5_med)
116  ""
117
118  Dado que dicho umbral parece ser aproximadamente el valor del dato

```

```

    792, se selecciona dicho valor, pues intentando encontrar una
    posible moda, los valores cuentan con una gran cantidad de
    decimales y por tanto no es posible encontrar un valor de esta
    forma.
119  '''{r}
120  umbral<-mod_lin_AT5_med$fitted.values[792]
121  umbral
122  '''
123
124  Si se realiza ahora un plot de residuales dibujando tambi n dicho
    umbral, vemos que aproximadamente coincide.
125  '''{r}
126  plot(mod_lin_AT5_med$fitted.values,mod_lin_AT5_med$residuals)
127  abline(v=umbral, lwd=2, col="red")
128  '''
129
130
131
132
133  '''{r}
134  plot(bombas_modelos$AT5_Med~FechaHora, type="l", col="blue", ylab = "
    AT5_Med")
135  abline(h=umbral, lwd=2, col="red")
136  '''
137
138  '''{r}
139  plot(mod_lin_AT5_med$fitted.values~FechaHora, type="l", col="lightblue
    ", ylab = "Fitted")
140  abline(h=umbral, lwd=2, col="red")
141  '''
142
143
144  Se crea una nueva variable que modele el umbral, o este cambio, para
    ver como se comporta.
145  '''{r}
146  varumbral <- ifelse(bombas_modelos$AT5_Med > umbral, 1, 0)
147  '''
148
149
150  Al realizar el nuevo modelo, se ve como la nueva variable que modela
    el umbral es muy significativa y mejora el modelo en gran medida.
    Ahora se llega a un R2 ajustado de 0.75 lo cual es una gran mejora.
    El modelo tambi n sigue siendo significativo
151
152
153
154  '''{r}
155  mod_umbral <- lm(AT5_Med ~ irr * temp + varumbral + irr:varumbral +
    temp:varumbral, data=bombas_modelos)
156  summary(mod_umbral)
157  '''

```

```

158
159 Adem s , se ve que no tiene problemas de colinealidad esta variable
    nueva.
160 ```{r}
161 vif(mod_umbral)
162 ```
163
164
165 Ahora los residuos si parecen ser mas lineales
166 ```{r}
167 plot(mod_umbral)
168 ```
169
170
171 ```{r}
172 library(nortest)
173 ad.test(resid(mod_umbral))
174 ad.test(resid(mod_umbral)[varumbral == 0])
175 ad.test(resid(mod_umbral)[varumbral == 1])
176
177 shapiro.test(resid(mod_umbral))
178 shapiro.test(resid(mod_umbral)[varumbral == 0])
179 shapiro.test(resid(mod_umbral)[varumbral == 1])
180 ```
181
182 ```{r}
183 plot(mod_umbral$fitted.values~FechaHora, type="l", col="lightblue",
    ylab = "Fitted")
184 #lines(bombas_modelos$AT5_Med~FechaHora, type="l", col="blue")
185 abline(h=umbral, lwd=2, col="red")
186 ```
187
188
189
190 ```{r}
191 mod_dias <- lm(AT5_Med ~ irr * temp + dia_semana, data=bombas_modelos)
192 summary(mod_dias)
193 ```
194
195
196 ```{r}
197 mod_dias_umbral <- lm(AT5_Med ~ irr * temp + dia_semana + varumbral,
    data=bombas_modelos)
198 summary(mod_dias_umbral)
199 ```
200
201
202 Bootstrap para poder calcular resultados mas robustos debido a la
    normalidad de los residuos.
203 ```{r}
204 library(boot)

```

```

205
206 boot_fn <- function(data, indices) {
207   d <- bombas_modelos[indices, ]
208   coef(lm(AT5_Med ~ irr * temp + varumbral, data = d))
209 }
210
211 set.seed(123)
212 boot_results <- boot(data = bombas_modelos, statistic = boot_fn, R =
  1000)
213 cat("Para intercept: ")
214 boot.ci(boot_results, type = "perc", index = 1)
215 cat("Para irr: ")
216 boot.ci(boot_results, type = "perc", index = 2)
217 cat("Para temp: ")
218 boot.ci(boot_results, type = "perc", index = 3)
219 cat("Para varumbral: ")
220 boot.ci(boot_results, type = "perc", index = 4)
221 ''
222
223
224
225 Se intenta boxcox tambien a ver que sale
226 ''{r}
227 library(MASS)
228 boxcox(mod_umbral)
229 ''
230
231
232
233
234 ## Corriente N Med
235
236 ''{r}
237 names(bombas_modelos)[names(bombas_modelos) == "Corriente N Med"] <- "
  Corr_N_Med"
238 ''
239
240 Curiosamente, esta variable parece tener un comportamiento escalonado.
241 ''{r}
242 plot(bombas_modelos$Corr_N_Med~FechaHora, type="l", col="blue")
243 ''
244
245
246 Aunque los resultados fuesen los segundos mejores segun la tabla del
  anterior informe, vemos los comportamientos tan extra os de los
  residuos.
247 ''{r}
248 mod_lin_Corr_N_Med <- lm(Corr_N_Med ~ irr * temp, data =
  bombas_modelos)
249 summary(mod_lin_Corr_N_Med)
250 plot(mod_lin_Corr_N_Med)

```

```

251  ““
252
253
254  Este comportamiento tiene sentido, pues el modelo lineal no es capaz
      de ajustar una funcion escalonada.
255  ““{r}
256  plot(mod_lin_Corr_N_Med$fitted.values, type="l")
257  lines(bombas_modelos$Corr_N_Med, col ="blue")
258  ““
259
260  Introduciendo ahora la variable de los dias de la semana.
261  ““{r}
262  mod_lin_Corr_N_Med_dias <- lm(Corr_N_Med ~ irr * temp + dia_semana,
      data = bombas_modelos)
263  summary(mod_lin_Corr_N_Med_dias)
264  plot(mod_lin_Corr_N_Med_dias)
265  ““
266
267
268  Como sigue siendo un modelo lineal, no va a modelar bien los escalones
      , aunque mejora en cierta manera.
269  ““{r}
270  plot(mod_lin_Corr_N_Med_dias$fitted.values, type="l")
271  lines(bombas_modelos$Corr_N_Med, col ="blue")
272  ““
273
274
275  Vemos que la variable solo toma 4 valores posibles.
276  ““{r}
277  table(bombas_modelos$Corr_N_Med)
278  ““
279
280  Se realiza ahora un modelo multinomial, considerando la variable como
      una categorica.
281  ““{r}
282  library(nnet)
283  Corr_Factor <- as.factor(bombas_modelos$Corr_N_Med)
284  mod_clas <- multinom(Corr_Factor ~ irr * temp)
285  summary(mod_clas)
286  ““
287
288
289  ““{r}
290  par(mfrow=c(1,2))
291  plot(as.numeric(predict(mod_clas, type = "class")), type="l")
292  #lines(bombas_modelos$Corr_N_Med, col ="blue")
293  plot(bombas_modelos$Corr_N_Med, col ="blue", type="l")
294  ““
295
296
297  ““{r}

```

```

298 par(mfrow=c(1,1))
299 ''
300
301
302
303
304 ## Corriente N Max
305
306 ''{r}
307 names(bombas_modelos)[names(bombas_modelos) == "Corriente N Max"] <- "
    Corr_N_Max"
308 ''
309
310 Si su media tiene un comportamiento escalonado, era claro que el
    m ximo tambi n. Por tanto ya podemos anticipar que los resultados
    ser n similares a los encontrados con la variable de media.
311 ''{r}
312 plot(bombas_modelos$Corr_N_Max~FechaHora, type="l", col="blue")
313 ''
314
315
316 ''{r}
317 mod_lin_Corr_N_Max <- lm(Corr_N_Max ~ irr * temp, data =
    bombas_modelos)
318 summary(mod_lin_Corr_N_Max)
319 plot(mod_lin_Corr_N_Max)
320 ''
321
322
323
324
325 Introduciendo ahora la variable de los d as de la semana.
326 ''{r}
327 mod_lin_Corr_N_Max_dias <- lm(Corr_N_Max ~ irr * temp + dia_semana,
    data = bombas_modelos)
328 summary(mod_lin_Corr_N_Max_dias)
329 plot(mod_lin_Corr_N_Max_dias)
330 ''
331
332
333 Al igual que con la media, como sigue siendo un modelo lineal, no va a
    modela bien los escalones, aunque mejora en cierta manera.
334 ''{r}
335 plot(mod_lin_Corr_N_Med_dias$fitted.values, type="l")
336 lines(bombas_modelos$Corr_N_Med, col ="blue")
337 ''
338
339
340
341 Se prueba el multinomial
342 ''{r}

```

```

343 library(nnet)
344 Corr_Factor <- as.factor(bombas_modelos$Corr_N_Max)
345 mod_clas_2 <- multinom(Corr_Factor ~ irr * temp)
346 summary(mod_clas_2)
347 ""
348
349
350 ""{r}
351 par(mfrow=c(1,2))
352 plot(as.numeric(predict(mod_clas_2, type = "class")), type="l")
353 #lines(bombas_modelos$Corr_N_Max, col ="blue")
354 plot(bombas_modelos$Corr_N_Max, col ="blue", type="l")
355 ""
356
357
358 ""{r}
359 par(mfrow=c(1,1))
360 ""
361
362
363
364
365 ## Tensi n de Medio Ciclo V RMS NG Min
366
367 ""{r}
368 names(bombas_modelos)[names(bombas_modelos) == "Tensi n de Medio
    Ciclo V RMS NG Min"] <- "TensionMC_V_RMS_Min"
369 ""
370
371 Volvemos a tener una funci n escalonada, aunque en este caso parece
    tener mayor n mero de valores posibles.
372 ""{r}
373 plot(bombas_modelos$TensionMC_V_RMS_Min~FechaHora, type="l", col="blue
    ")
374 ""
375
376 Seguimos con el comportamiento de las dos variables anteriores.
377 ""{r}
378 mod_lin_Ten_RMS_Min <- lm(TensionMC_V_RMS_Min ~ irr * temp, data =
    bombas_modelos)
379 plot(mod_lin_Ten_RMS_Min)
380 summary(mod_lin_Ten_RMS_Min)
381 ""
382
383 ""{r}
384 plot(mod_lin_Ten_RMS_Min$fitted.values, type="l")
385 lines(bombas_modelos$TensionMC_V_RMS_Min, col ="blue")
386 ""
387
388
389

```

```

390 Introduciendo ahora la variable de los días de la semana.
391 ```{r}
392 mod_lin_Ten_RMS_Min_dias <- lm(TensionMC_V_RMS_Min ~ irr * temp +
    dia_semana, data = bombas_modelos)
393 summary(mod_lin_Ten_RMS_Min_dias)
394 plot(mod_lin_Ten_RMS_Min_dias)
395 ```
396
397
398 Al igual que con las otras dos variables, como sigue siendo un modelo
    lineal, no va a modelar bien los escalones, aunque mejora en cierta
    manera.
399 ```{r}
400 plot(mod_lin_Ten_RMS_Min_dias$fitted.values, type="l")
401 lines(bombas_modelos$TensionMC_V_RMS_Min, col = "blue")
402 ```
403
404
405 Se prueba el multinomial
406 ```{r}
407 library(nnet)
408 Tens_Factor <- as.factor(bombas_modelos$TensionMC_V_RMS_Min)
409 mod_clas_3 <- multinom(Tens_Factor ~ irr * temp)
410 summary(mod_clas_3)
411 ```
412
413
414
415 ```{r}
416 par(mfrow=c(1,2))
417 plot(as.numeric(predict(mod_clas_3, type = "class")), type="l")
418 #lines(bombas_modelos$TensionMC_V_RMS_Min, col = "blue")
419 plot(bombas_modelos$TensionMC_V_RMS_Min, col = "blue", type="l")
420 ```
421
422
423 ```{r}
424 par(mfrow=c(1,1))
425 ```
426
427
428
429 ## Vrms ph-n NG Min
430
431
432 ```{r}
433 names(bombas_modelos)[names(bombas_modelos) == "Vrms ph-n NG Min"] <-
    "Vrms_NG_Min"
434 ```
435
436 Seguimos con el escalonado. Se ve ahora que los resultados son

```

```

similares a las ltimas 3 variables.
437 ```{r}
438 plot(bombas_modelos$Vrms_NG_Min~FechaHora, type="l", col="blue")
439 ```
440
441 Mismo comportamiento.
442 ```{r}
443 mod_lin_Vrms_NG_Min <- lm(Vrms_NG_Min ~ irr * temp, data =
444   bombas_modelos)
445 plot(mod_lin_Vrms_NG_Min)
446 summary(mod_lin_Vrms_NG_Min)
447 ```
448
449 ```{r}
450 plot(mod_lin_Vrms_NG_Min$fitted.values, type="l")
451 lines(bombas_modelos$Vrms_NG_Min, col ="blue")
452 ```
453
454
455 Introduciendo ahora la variable de los d as de la semana.
456 ```{r}
457 mod_lin_Vrms_NG_Min_dias <- lm(Vrms_NG_Min ~ irr * temp + dia_semana,
458   data = bombas_modelos)
459 summary(mod_lin_Vrms_NG_Min_dias)
460 plot(mod_lin_Vrms_NG_Min_dias)
461 ```
462
463
464 Al igual que con las otras dos variables, como sigue siendo un modelo
465   lineal, no va a modela bien los escalones, aunque mejora en cierta
466   manera.
467 ```{r}
468 plot(mod_lin_Vrms_NG_Min_dias$fitted.values, type="l")
469 lines(bombas_modelos$Vrms_NG_Min, col ="blue")
470 ```
471
472 Se prueba el multinomial
473 ```{r}
474 library(nnet)
475 Vrms_Factor <- as.factor(bombas_modelos$Vrms_NG_Min)
476 mod_clas_4 <- multinom(Vrms_Factor ~ irr * temp)
477 summary(Vrms_Factor)
478 ```
479
480
481 ```{r}
482 plot(as.numeric(as.character(predict(mod_clas_4, type = "class"))),

```

```

    type="1", ylim=c(0.038,0.08))
483 lines(bombas_modelos$Vrms_NG_Min, col ="blue")
484 '""
485
486
487 '{r}
488 par(mfrow=c(1,1))
489 '""
490
491
492
493
494
495 # Mejores modelos GAM
496
497
498 ## Arm nicos Tensi n5 L1N Med
499 Ya se ha visto la utilidad y la creaci n de estos modelos antes, se
    pasa directamente al modelo completo con interacci n no lineal.
500
501 Al parecer, si bien es verdad que es significativo, en el gr fico se
    puede ver como la irradiancia tiene un aporte pr cticamente lineal
    , pero en el modelo sin interacci n se hab a visto que no era
    as . Esto se puede deber a que la interacci n no lineal ha
    absorbido ese efecto.
502 '{r}
503 library(mgcv)
504 mod_gam_AT5_int <- gam(AT5_Med ~ s(irr) + s(temp) + ti(irr, temp),
    data = bombas_modelos)
505 summary(mod_gam_AT5_int)
506 plot(mod_gam_AT5_int)
507 '""
508
509
510
511
512 El efecto que hacia antes irradiancia en el modelo sin interacci n ya
    se modela perfectamente en la interacci n del modelo completo.
513 '{r}
514 mod_gam_AT5 <- gam(AT5_Med ~ s(irr) + s(temp), data = bombas_modelos)
515 plot(mod_gam_AT5, select = 1)
516 '""
517
518
519 '{r}
520 mod_gam_AT5_irrlin <- gam(AT5_Med ~ irr + s(temp) + ti(irr, temp),
    data = bombas_modelos)
521 summary(mod_gam_AT5_irrlin)
522 plot(mod_gam_AT5_irrlin)
523 '""
524

```

```

525
526 Probando ahora a a adir la variable umbral. Los resultados son mucho
    mejores.
527 ```{r}
528 mod_gam_AT5_umbral <- gam(AT5_Med ~ s(irr) + s(temp) + varumbral + ti(
    irr, temp), data = bombas_modelos)
529 summary(mod_gam_AT5_umbral)
530 plot(mod_gam_AT5_umbral)
531 ```
532
533
534 En el gr fico se ve claramente la mejora de a adir el umbral.
535 ```{r}
536 plot(mod_gam_AT5_umbral$fitted.values, type="l")
537 lines(bombas_modelos$AT5_Med, col="blue")
538 ```
539
540
541
542 ```{r, warning=FALSE}
543 vis.gam(mod_gam_AT5_int, view = c("irr", "temp"), plot.type = "persp")
544 vis.gam(mod_gam_AT5_int, view = c("irr", "temp"), plot.type = "contour"
    ")
545 ```
546
547
548
549
550
551 ## Arm nicos Corriente5 L1N Med
552
553 ```{r}
554
555 ```
556
557
558 ## Corriente N Med
559
560 Aunque ahora tenemos resultados ciertamente diferentes, seguimos
    teniendo que la no linealidad del efecto de la irradiancia se lo
    vuelve a llevar la interacci n.
561 ```{r}
562 mod_gam_Corr_N_Med_int <- gam(Corr_N_Med ~ s(irr) + s(temp) + ti(irr,
    temp), data = bombas_modelos)
563 summary(mod_gam_Corr_N_Med_int)
564 plot(mod_gam_Corr_N_Med_int)
565 ```
566
567 Aqu se ve el efecto de la irradiancia si no se introduce la
    interacci n en el modelo.
568 ```{r}

```

```

569 mod_gam_Corr_N_Med <- gam(Corr_N_Med ~ s(irr) + s(temp), data =
    bombas_modelos)
570 plot(mod_gam_Corr_N_Med, select = 1)
571 '''
572
573
574 Viendo las predicciones del modelo, si bien los modelos GAM tampoco
    modelan bien las escalonadas, parece hacerlo mejor que el modelo
    lineal.
575 '''{r}
576 plot(mod_gam_Corr_N_Med_int$fitted.values, type="l")
577 lines(bombas_modelos$Corr_N_Med, col="blue")
578 '''
579
580
581 Introduciendo dia de la semana mejorar levemente la varianza que se
    explica.
582 '''{r}
583 mod_gam_Corr_N_Med_dias <- gam(Corr_N_Med ~ s(irr) + s(temp) +
    dia_semana + ti(irr, temp), data = bombas_modelos)
584 summary(mod_gam_Corr_N_Med_dias)
585 plot(mod_gam_Corr_N_Med_dias)
586 '''
587
588
589 Parece que no hay suficiente variabilidad para ponerlo como efecto
    aleatorio.
590 '''{r , error=TRUE}
591 mod_gamm <- gamm(Corr_N_Med ~ s(irr) + s(temp) + ti(irr, temp),
592                 random = list(dia_semana = ~1), data = bombas_modelos
                    )
593 summary(mod_gamm)
594 '''
595
596
597 Se intenta un vglm utilizando el valor como si fuese un factor.
598 '''{r}
599 library(VGAM)
600 mod_vglm <- vglm(as.factor(Corr_N_Med) ~ s(irr) + s(temp), family =
    multinomial, data = bombas_modelos)
601 summary(mod_vglm)
602 '''
603
604 '''{r}
605 pred_vglm <- predict(mod_vglm, type = "response")
606 pred_class <- apply(pred_vglm, 1, which.max)
607 '''
608
609 '''{r}
610 tabla_confusion <- table(Predicho = pred_class, Real = as.numeric(as.
    factor(bombas_modelos$Corr_N_Med)))

```

```

611 accuracy <- mean(pred_class == as.numeric(as.factor(
        bombas_modelos$Corr_N_Med)))
612 accuracy
613 '''
614
615
616 ```{r}
617 # Suponiendo que irr y temp var an en estos rangos
618 nuevo_grid <- expand.grid(
619   irr = seq(min(bombas_modelos$irr), max(bombas_modelos$irr), length.
        out = 100),
620   temp = median(bombas_modelos$temp)
621 )
622
623 prob_pred <- predict(mod_vglm, newdata = nuevo_grid, type = "response"
        )
624 matplot(nuevo_grid$irr, prob_pred, type = "l", lty = 1, col = 1:4,
        ylab = "Probabilidad", xlab = "Irradiancia")
625 legend("topright", legend = c("0.2", "0.21", "0.22", "0.23"), col =
        1:4, lty = 1)
626
627 '''
628
629
630 ```{r}
631 # predicciones de clase m s probable
632 pred_probs <- predict(mod_vglm, type = "response") # matriz de
        probabilidades
633 pred_class <- apply(pred_probs, 1, which.max)
634
635 # Convertir ndices a valores num ricos reales
636 niveles <- levels(as.factor(bombas_modelos$Corr_N_Med))
637 pred_clase_real <- as.numeric(as.character(niveles[pred_class]))
638
639 '''
640
641
642 ```{r}
643 plot(pred_clase_real, type="l")
644 lines(bombas_modelos$Corr_N_Med, col="blue")
645 '''
646
647
648
649
650
651 ## Desequilibrio An Med
652
653 ## Tensi n de Medio Ciclo V RMS NG Min

```

Listing C.10: Modelos nuevo

C.0.11. Procesado

Tras recibir todos los demás archivos, proceso que se encarga de depurar todos los archivos y realizar los *merge* necesarios.

```
1 ---
2 title: "Procesado datos"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()'"
5 output: pdf_document
6 ---
7
8 '{r , warning=FALSE, message=FALSE, echo=FALSE}'
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15 library(car)
16 library(zoo)
17
18 # No hace falta poner '/' al final, ya se tiene en cuenta luego
19 #ruta_datos <-"C:/Users/ismae/Desktop/TFG EST/data/modificados"
20 ruta_datos <-"H:/TFG EST/General"
21 ''
22
23 Todos los NA's que se sustituyen son trav s de una interpolaci n con
24 splines (librer a 'zoo').
25
26 Grandes cantidades de NA's no se pueden interpolar, por tanto esas
27 partes de tiempo son borradas.
28
29
30 # Extracci n de cada dataset y comprobaciones
31
32 ## 1. Datos de 19-08 a 03-09
33
34 Este rango de datos contiene las dos CT.
35
36
37 ### Irradiancia
38
39 '{r}'
40 ruta_especifica_d1<-paste(ruta_datos, "/Datos1 de 19-08 a 03-09.xlsx",
41 sep="")
42 d1_irr <- read_excel(ruta_especifica_d1, sheet = "Irradiancia")
43 head(d1_irr)
44 ''
```

```

44 | Contiene una fila de NA's
45 | ```{r, echo = FALSE, results = "hide"}
46 | anyNA(d1_irr)
47 | na_cols <- colSums(is.na(d1_irr))
48 | na_cols <- na_cols[na_cols > 0]
49 |
50 | which_rows_na <- which(rowSums(is.na(d1_irr[names(na_cols)]))) > 0)
51 |
52 | d1_irr[which_rows_na, ]
53 | ```
54 |
55 | ```{r}
56 | d1_irr <- as.data.frame(lapply(d1_irr, na.spline), check.names = FALSE)
57 | anyNA(d1_irr)
58 | ```
59 |
60 |
61 | ### CT Bombas
62 |
63 |
64 | ```{r , message = FALSE}
65 | d1_bombas <- read_excel(ruta_especifica_d1, sheet = "CT Bombas")
66 | #names(d1_bombas)
67 |
68 | # Eliminar columnas desde "Column1" en adelante
69 | columnas_borrar <- which(names(d1_bombas) == "Column1")
70 | if(length(columnas_borrar) > 0) {
71 |   d1_bombas <- d1_bombas[, 1:(columnas_borrar - 1)]
72 | }
73 |
74 | # Eliminar columnas que contienen "L2" o "L3"
75 | d1_bombas <- d1_bombas[, !grepl("L2|L3", names(d1_bombas))]
76 |
77 | #names(d1_bombas)
78 | head(d1_bombas[,c(1,2,3,4)],3)
79 | d1_bombas$Hora <- format(d1_bombas$Hora, "%H:%M:%S")
80 | Fecha <- as.POSIXct(paste(d1_bombas$Fecha, d1_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz="UTC")
81 | d1_bombas<-cbind(Fecha, d1_bombas[, -c(1,2)])
82 | head(d1_bombas[,c(1,2,3,4)],3)
83 | ```
84 |
85 | Muchas variables son practicamente enteras Na's, mirar a ver.
86 | ```{r, echo = FALSE, results = "hide"}
87 | anyNA(d1_bombas)
88 | na_cols <- colSums(is.na(d1_bombas))
89 | na_cols <- na_cols[na_cols > 0]
90 | na_cols
91 | #which_rows_na <- which(rowSums(is.na(d1_bombas[names(na_cols)]))) > 0)
92 |
93 | #d1_bombas[which_rows_na, names(na_cols)]

```

```

94  '''
95
96
97  ### CT Nuevo
98
99
100  '''{r}
101  d1_nuevo <- read_excel(ruta_especifica_d1, sheet = "CT Nuevo")
102  #names(d1_nuevo)
103
104  d1_nuevo <- d1_nuevo[, !grepl("L2|L3", names(d1_nuevo))]
105  d1_nuevo <- d1_nuevo[, -ncol(d1_nuevo)]
106  #names(d1_nuevo)
107
108  head(d1_nuevo[,c(1,2,3,4)],3)
109  d1_nuevo$Hora <- format(d1_nuevo$Hora, "%H:%M:%S")
110  Fecha <- as.POSIXct(paste(d1_nuevo$Fecha, d1_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz="UTC")
111  d1_nuevo<-cbind(Fecha, d1_nuevo[, -c(1,2)])
112  head(d1_nuevo[,c(1,2,3,4)],3)
113  '''
114
115
116  '''{r, echo = FALSE, results = "hide"}
117  anyNA(d1_nuevo)
118  '''
119
120
121  ### Variables de corrección de rangos de fechas
122
123  '''{r}
124  cat("Para irr \n")
125  as.POSIXct(d1_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
126  as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
127  cat("\nPara bombas \n")
128  as.POSIXct(d1_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
129  as.POSIXct(d1_bombas$Fecha[length(d1_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
130  cat("\nPara nuevo \n")
131  as.POSIXct(d1_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
132  as.POSIXct(d1_nuevo$Fecha[length(d1_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
133  '''
134
135  '''{r}
136  cat("Para irr de bombas\n")
137  as.POSIXct(d1_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
138  as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
139  cat("\nPara bombas \n")

```

```

140 as.POSIXct(d1_bombas$Fecha[3], origin = "1970-01-01", tz = "UTC")
141 as.POSIXct(d1_bombas$Fecha[length(d1_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
142 cat("\nPara irr de nuevo\n")
143 as.POSIXct(d1_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
144 as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
145 cat("\nPara nuevo\n")
146 as.POSIXct(d1_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
147 as.POSIXct(d1_nuevo$Fecha[length(d1_nuevo$Fecha)-1], origin = "
    1970-01-01", tz = "UTC")
148 '''
149
150
151 '{{{r}
152 # Convertir fechas a POSIXct en UTC
153 f_irr    <- as.POSIXct(d1_irr$Fecha, origin = "1970-01-01", tz = "UTC")
154 f_bomb   <- as.POSIXct(d1_bombas$Fecha, origin = "1970-01-01", tz = "
    UTC")
155 f_nuev   <- as.POSIXct(d1_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
    ")
156
157 # Recortes aplicados seg n lo indicado
158 d1_irr_bombas <- d1_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1), ]
159 d1_bombas    <- d1_bombas[f_bomb >= f_bomb[3] & f_bomb <= tail(f_bomb
    , 1), ]
160 d1_irr_nuevo <- d1_irr[f_irr >= f_irr[2] & f_irr <= tail(f_irr, 1), ]
161 d1_nuevo     <- d1_nuevo[f_nuev >= f_nuev[1] & f_nuev <= f_nuev[
    length(f_nuev) - 1], ]
162
163 # Verificaci n
164 cat(sprintf("d1_bombas: %d | d1_nuevo: %d | d1_irr_bombas: %d |
    d1_irr_nuevo: %d\n",
165             nrow(d1_bombas), nrow(d1_nuevo), nrow(d1_irr_bombas), nrow
    (d1_irr_nuevo)))
166 '''
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181 ## 2. Datos de 03-09 a 13-09

```

```

182
183 Este rango de datos solo contiene CT bombas.
184
185
186 ### Irradiancia
187
188 ```{r}
189 ruta_especifica_d2<-paste(ruta_datos, "/Datos2 de 03-09 a 13-09.xlsx",
190   sep="")
191 d2_irr <- read_excel(ruta_especifica_d2, sheet = "Irradiancia")
192 head(d2_irr)
193 ```
194
195 Contiene dos filas seguidas de NA's
196 ```{r, echo = FALSE, results = "hide"}
197 anyNA(d2_irr)
198 na_cols <- colSums(is.na(d2_irr))
199 na_cols <- na_cols[na_cols > 0]
200
201 which_rows_na <- which(rowSums(is.na(d2_irr[names(na_cols)])) > 0)
202
203 d2_irr[which_rows_na, ]
204 ```
205
206 ```{r}
207 d2_irr <- as.data.frame(lapply(d2_irr, na.spline),check.names = FALSE)
208 anyNA(d2_irr)
209 ```
210
211 ### CT Bombas
212
213
214 ```{r}
215 d2_bombas <- read_excel(ruta_especifica_d2, sheet = "CT Bombas")
216 #names(d2_bombas)
217
218 # Eliminar columnas desde "Column1" en adelante
219 columnas_borrar <- which(names(d2_bombas) == "Column1")
220 if(length(columnas_borrar) > 0) {
221   d2_bombas <- d2_bombas[, 1:(columnas_borrar - 1)]
222 }
223
224 # Eliminar columnas que contienen "L2" o "L3"
225 d2_bombas <- d2_bombas[, !grepl("L2|L3", names(d2_bombas))]
226
227 #names(d2_bombas)
228 head(d2_bombas[,c(1,2,3,4)],3)
229 d2_bombas$Hora <- format(d2_bombas$Hora, "%H:%M:%S")
230 Fecha <- as.POSIXct(paste(d2_bombas$Fecha, d2_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")

```

```

231 d2_bombas<-cbind(Fecha, d2_bombas[,-c(1,2)])
232 head(d2_bombas[,c(1,2,3,4)],3)
233 '''
234
235 '''{r, echo = FALSE, results = "hide"}
236 anyNA(d2_bombas)
237 '''
238
239
240 ### Variables de correcci n de rangos de fechas
241
242 '''{r}
243 cat("Para irr \n")
244 as.POSIXct(d2_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
245 as.POSIXct(d2_irr$Fecha[length(d2_irr$Fecha)], origin = "1970-01-01",
246           tz = "UTC")
247 cat("\nPara bombas \n")
248 as.POSIXct(d2_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
249 as.POSIXct(d2_bombas$Fecha[length(d2_bombas$Fecha)], origin = "
250           1970-01-01", tz = "UTC")
251 '''
252
253
254
255
256
257
258
259 ## 3. Datos de 24-09 a 02-10
260
261 Este rango de datos solo contiene CT bombas.
262
263
264
265 ### Irradiancia
266
267 '''{r}
268 ruta_especifica_d3<-paste(ruta_datos, "/Datos3 de 24-09 a 02-10.xlsx",
269                           sep="")
270 d3_irr <- read_excel(ruta_especifica_d3, sheet = "Irradiancia")
271 head(d3_irr)
272
273
274 '''{r, echo = FALSE, results = "hide"}
275 anyNA(d3_irr)
276 '''
277
278 ### CT Bombas

```

```

279 |
280 |
281 | ‘‘{r}
282 | d3_bombas <- read_excel(ruta_especifica_d3, sheet = "CT Bombas")
283 | #names(d3_bombas)
284 |
285 | # Eliminar columnas desde "Column1" en adelante
286 | columnas_borrar <- which(names(d3_bombas) == "Column1")
287 | if(length(columnas_borrar) > 0) {
288 |   d3_bombas <- d3_bombas[, 1:(columnas_borrar - 1)]
289 | }
290 |
291 | # Eliminar columnas que contienen "L2" o "L3"
292 | d3_bombas <- d3_bombas[, !grepl("L2|L3", names(d3_bombas))]
293 |
294 | #names(d3_bombas)
295 | head(d3_bombas[,c(1,2,3,4)],3)
296 | d3_bombas$Hora <- format(d3_bombas$Hora, "%H:%M:%S")
297 | Fecha <- as.POSIXct(paste(d3_bombas$Fecha, d3_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
298 | d3_bombas<-cbind(Fecha, d3_bombas[, -c(1,2)])
299 | head(d3_bombas[,c(1,2,3,4)],3)
300 | ‘‘‘
301 |
302 |
303 | Muchas variables son practicamente enteras Na's, mirar a ver. Vuelve a
    pasar lo mismo, aqui ahora hay otro problema, hay variables que
    aparecen con ...XXX siendo XXX numeros, esto al parecer ocurre
    porque hay variables con nombres repetidos.
304 | ‘‘{r, echo = FALSE, results = "hide"}
305 | anyNA(d3_bombas)
306 | na_cols <- colSums(is.na(d3_bombas))
307 | na_cols <- na_cols[na_cols > 0]
308 | na_cols
309 | #which_rows_na <- which(rowSums(is.na(d3_bombas[names(na_cols)])) > 0)
310 |
311 | #d3_bombas[which_rows_na, names(na_cols)]
312 | ‘‘‘
313 |
314 |
315 |
316 |
317 |
318 |
319 |
320 | ### Variables de correccion de rangos de fechas
321 |
322 |
323 | Debido a que los intervalos de las fechas cambian, hay que dividirlo
    en dos y luego volver a juntar supongo?
324 | ‘‘{r}

```

```

325 cat("Para irr \n")
326 as.POSIXct(d3_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
327 as.POSIXct(d3_irr$Fecha[length(d3_irr$Fecha)], origin = "1970-01-01",
328           tz = "UTC")
329 cat("\nPara bombas \n")
330 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
331 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
332           1970-01-01", tz = "UTC")
333 '''
334 cat("\nPara irr \n")
335 as.POSIXct(d3_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
336 d3_irr$Fecha[900]
337 d3_irr$Fecha[901]
338 as.POSIXct(d3_irr$Fecha[length(d3_irr$Fecha)], origin = "1970-01-01",
339           tz = "UTC")
340 cat("\nPara bombas \n")
341 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
342 as.POSIXct(d3_bombas$Fecha[899], origin = "1970-01-01", tz = "UTC")
343 as.POSIXct(d3_bombas$Fecha[901], origin = "1970-01-01", tz = "UTC")
344 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
345           1970-01-01", tz = "UTC")
346 '''
347 cat("\nPara bombas \n")
348 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
349 as.POSIXct(d3_bombas$Fecha[899], origin = "1970-01-01", tz = "UTC")
350 as.POSIXct(d3_bombas$Fecha[901], origin = "1970-01-01", tz = "UTC")
351 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
352           1970-01-01", tz = "UTC")
353 '''
354 cat("\nPara bombas \n")
355 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
356 as.POSIXct(d3_bombas$Fecha[899], origin = "1970-01-01", tz = "UTC")
357 as.POSIXct(d3_bombas$Fecha[901], origin = "1970-01-01", tz = "UTC")
358 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
359           1970-01-01", tz = "UTC")
360 '''
361 cat("\nPara bombas \n")
362 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
363 as.POSIXct(d3_bombas$Fecha[899], origin = "1970-01-01", tz = "UTC")
364 as.POSIXct(d3_bombas$Fecha[901], origin = "1970-01-01", tz = "UTC")
365 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
366           1970-01-01", tz = "UTC")
367 '''

```

```

368
369
370
371
372
373
374
375
376 ## 4. Datos de 02-10 a 22-10
377
378 Este rango de datos solo contiene CT nuevo.
379
380
381
382 ### Irradiancia
383
384 ```{r}
385 ruta_especifica_d4<-paste(ruta_datos, "/Datos4 de 02-10 a 22-10.xlsx",
386   sep="")
387 d4_irr <- read_excel(ruta_especifica_d4, sheet = "Irradiancia")
388 head(d4_irr)
389
390 Contiene dos fila de NA's
391 ```{r, echo = FALSE, results = "hide"}
392 anyNA(d4_irr)
393 na_cols <- colSums(is.na(d4_irr))
394 na_cols <- na_cols[na_cols > 0]
395
396 which_rows_na <- which(rowSums(is.na(d4_irr[names(na_cols)])) > 0)
397
398 d4_irr[which_rows_na, ]
399 ```
400
401 ```{r}
402 d4_irr <- as.data.frame(lapply(d4_irr, na.spline),check.names = FALSE)
403 anyNA(d4_irr)
404 ```
405
406
407
408 ### CT Nuevo
409
410
411 ```{r}
412 d4_nuevo <- read_excel(ruta_especifica_d4, sheet = "CT Nuevo")
413 #names(d4_nuevo)
414
415 head(d4_nuevo[,c(1,2,3,4)],3)
416 d4_nuevo$Hora <- format(d4_nuevo$Hora, "%H:%M:%S")
417 Fecha <- as.POSIXct(paste(d4_nuevo$Fecha, d4_nuevo$Hora), format="%Y-%

```

```

    m-%d %H:%M:%S", tz = "UTC")
418 d4_nuevo<-cbind(Fecha, d4_nuevo[,-c(1,2)])
419 head(d4_nuevo[,c(1,2,3,4)],3)
420 ""
421
422 6 variables con una gran cantidad de NA's
423 "{r, echo = FALSE, results = "hide"}
424 anyNA(d4_nuevo)
425 na_cols <- colSums(is.na(d4_nuevo))
426 na_cols <- na_cols[na_cols > 0]
427 na_cols
428 #which_rows_na <- which(rowSums(is.na(d4_nuevo[names(na_cols)])) > 0)
429
430 #d4_nuevo[which_rows_na, names(na_cols)]
431 ""
432
433
434 ### Variables de correcci n de rangos de fechas
435
436
437 "{r}
438 cat("Para irr \n")
439 as.POSIXct(d4_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
440 as.POSIXct(d4_irr$Fecha[length(d4_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
441 cat("\nPara nuevo \n")
442 as.POSIXct(d4_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
443 as.POSIXct(d4_nuevo$Fecha[length(d4_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
444 ""
445
446 "{r}
447 cat("Para irr \n")
448 as.POSIXct(d4_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
449 as.POSIXct(d4_irr$Fecha[length(d4_irr$Fecha)-1], origin = "1970-01-01"
    , tz = "UTC")
450 cat("\nPara nuevo \n")
451 as.POSIXct(d4_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
452 as.POSIXct(d4_nuevo$Fecha[length(d4_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
453 ""
454
455
456
457 "{r}
458 # Fechas como POSIXct en UTC
459 f_irr <- as.POSIXct(d4_irr$Fecha, origin = "1970-01-01", tz = "UTC")
460 f_nuev <- as.POSIXct(d4_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
    ")
461
462 # Recortes basados en inspecci n

```

```

463 d4_irr_nuevo    <- d4_irr[1:(nrow(d4_irr) - 1), ]
464 d4_nuevo <- d4_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1),
465   ]
466 # Verificaci n de filas
467 cat(sprintf("d4_irr_nuevo: %d | d4_nuevo: %d\n", nrow(d4_irr_nuevo),
468   nrow(d4_nuevo)))
469
470
471
472
473
474
475
476
477
478
479
480 ## 5. Datos de 24-10 a 13-11
481
482 Este rango de datos solo contiene CT nuevo.
483
484
485
486 ### Irradiancia
487
488 ```{r}
489 ruta_especifica_d5<-paste(ruta_datos, "/Datos5 de 24-10 a 13-11.xlsx",
490   sep="")
491 d5_irr <- read_excel(ruta_especifica_d5, sheet = "Irradiancia")
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508

```

```

509  ““{r}
510  d5_irr <- as.data.frame(lapply(d5_irr, na.spline),check.names = FALSE)
511  anyNA(d5_irr)
512  ““
513
514
515
516  ### CT Nuevo
517
518
519  ““{r}
520  d5_nuevo <- read_excel(ruta_especifica_d5, sheet = "CT Nuevo")
521  #names(d5_nuevo)
522
523  head(d5_nuevo[,c(1,2,3,4)],3)
524  d5_nuevo$Hora <- format(d5_nuevo$Hora, "%H:%M:%S")
525  Fecha <- as.POSIXct(paste(d5_nuevo$Fecha, d5_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz = "UTC")
526  d5_nuevo<-cbind(Fecha, d5_nuevo[, -c(1,2)])
527  head(d5_nuevo[,c(1,2,3,4)],3)
528  ““
529
530
531  ““{r, echo = FALSE, results = "hide"}
532  anyNA(d5_nuevo)
533  ““
534
535  ### Variables de correcci n de rangos de fechas
536
537  ““{r}
538  cat("Para irr \n")
539  as.POSIXct(d5_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
540  as.POSIXct(d5_irr$Fecha[length(d5_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
541  cat("\nPara nuevo \n")
542  as.POSIXct(d5_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
543  as.POSIXct(d5_nuevo$Fecha[length(d5_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
544  ““
545
546
547
548  ““{r}
549  cat("Para irr \n")
550  as.POSIXct(d5_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
551  as.POSIXct(d5_irr$Fecha[length(d5_irr$Fecha)-6], origin = "1970-01-01"
    , tz = "UTC")
552  cat("\nPara nuevo \n")
553  as.POSIXct(d5_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
554  as.POSIXct(d5_nuevo$Fecha[length(d5_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")

```

```

555  '''
556
557
558  '{{{r}
559  # Convertir fechas a POSIXct en UTC
560  f_irr <- as.POSIXct(d5_irr$Fecha, origin = "1970-01-01", tz = "UTC")
561  f_nuev <- as.POSIXct(d5_nuevo$Fecha, origin = "1970-01-01", tz = "UTC"
562    )
563
564  # Aplicar recortes
565  d5_irr_nuevo <- d5_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
566    f_irr) - 6], ]
567  d5_nuevo <- d5_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1),
568    ]
569
570  # Verificaci n
571  cat(sprintf("d5_irr_nuevo: %d | d5_nuevo: %d\n", nrow(d5_irr_nuevo),
572    nrow(d5_nuevo)))
573
574  '''
575
576  ## 6. Datos de 14-11 a 27-11
577
578  Este rango de datos contiene las dos CT.
579
580
581  ### Irradiancia
582
583  '{{{r}
584  ruta_especifica_d6<-paste(ruta_datos, "/Datos de 14-11 a 27-11.xlsx",
585    sep="")
586  d6_irr <- read_excel(ruta_especifica_d6, sheet = "Irradiancia")
587  head(d6_irr)
588  '''
589
590  '{{{r, echo = FALSE, results = "hide"}
591  anyNA(d6_irr)
592  '''
593
594  ### CT Bombas
595
596  '{{{r}
597  d6_bombas <- read_excel(ruta_especifica_d6, sheet = "CT Bombas")
598  #names(d6_bombas)
599
600  head(d6_bombas[,c(1,2,3,4)],3)

```

```

601 d6_bombas$Hora <- format(d6_bombas$Hora, "%H:%M:%S")
602 Fecha <- as.POSIXct(paste(d6_bombas$Fecha, d6_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
603 d6_bombas<-cbind(Fecha, d6_bombas[, -c(1,2)])
604 head(d6_bombas[, c(1,2,3,4)], 3)
605 ```
606
607 ```{r, echo = FALSE, results = "hide"}
608 anyNA(d6_bombas)
609 ```
610
611
612 ### CT Nuevo
613
614
615 ```{r}
616 d6_nuevo <- read_excel(ruta_especifica_d6, sheet = "CT Nuevo")
617 #names(d6_nuevo)
618
619 head(d6_nuevo[, c(1,2,3,4)], 3)
620 d6_nuevo$Hora <- format(d6_nuevo$Hora, "%H:%M:%S")
621 Fecha <- as.POSIXct(paste(d6_nuevo$Fecha, d6_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz = "UTC")
622 d6_nuevo<-cbind(Fecha, d6_nuevo[, -c(1,2)])
623 head(d6_nuevo[, c(1,2,3,4)], 3)
624 ```
625
626 6 variables con una gran cantidad de NA's
627 ```{r, echo = FALSE, results = "hide"}
628 anyNA(d6_nuevo)
629 na_cols <- colSums(is.na(d6_nuevo))
630 na_cols <- na_cols[na_cols > 0]
631 na_cols
632 #which_rows_na <- which(rowSums(is.na(d6_nuevo[names(na_cols)])) > 0)
633
634 #d6_nuevo[which_rows_na, names(na_cols)]
635 ```
636
637
638 ### Variables de correcci n de rangos de fechas
639
640 ```{r}
641 cat("Para irr \n")
642 as.POSIXct(d6_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
643 as.POSIXct(d6_irr$Fecha[length(d6_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
644 cat("\nPara bombas \n")
645 as.POSIXct(d6_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
646 as.POSIXct(d6_bombas$Fecha[length(d6_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
647 cat("\nPara nuevo \n")

```

```

648 as.POSIXct(d6_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
649 as.POSIXct(d6_nuevo$Fecha[length(d6_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
650
651
652 ```{r}
653 d6_irr_bombas <- d6_irr
654 d6_irr_nuevo <- d6_irr
655 ```
656
657 ```{r}
658 cat(sprintf("d6_bombas: %d | d6_nuevo: %d | d6_irr_bombas: %d |
    d6_irr_nuevo: %d\n",
659           nrow(d6_bombas), nrow(d6_nuevo), nrow(d6_irr_bombas), nrow
        (d6_irr_nuevo)))
660
661
662
663
664
665
666
667
668 ## 7. Datos de 27-11 a 03-12
669
670 Este rango de datos contiene las dos CT.
671
672
673
674 ### Irradiancia
675
676 ```{r}
677 ruta_especifica_d7<-paste(ruta_datos, "/Datos7 de 27-11 a 03-12.xlsx",
    sep="")
678 d7_irr <- read_excel(ruta_especifica_d7, sheet = "Irradiancia")
679 head(d7_irr)
680 ```
681
682
683 ```{r, echo = FALSE, results = "hide"}
684 anyNA(d7_irr)
685 ```
686
687
688 ### CT Bombas
689
690
691 ```{r}
692 d7_bombas <- read_excel(ruta_especifica_d7, sheet = "CT Bombas")
693 #names(d7_bombas)
694

```

```

695 head(d7_bombas[,c(1,2,3,4)],3)
696 d7_bombas$Hora <- format(d7_bombas$Hora, "%H:%M:%S")
697 Fecha <- as.POSIXct(paste(d7_bombas$Fecha, d7_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz="UTC")
698 d7_bombas<-cbind(Fecha, d7_bombas[,c(1,2)])
699 head(d7_bombas[,c(1,2,3,4)],3)
700 ```
701
702 ```{r, echo = FALSE, results = "hide"}
703 anyNA(d7_bombas)
704 ```
705
706
707 ### CT Nuevo
708
709
710 ```{r}
711 d7_nuevo <- read_excel(ruta_especifica_d7, sheet = "CT Nuevo")
712 #names(d7_nuevo)
713
714 head(d7_nuevo[,c(1,2,3,4)],3)
715 d7_nuevo$Hora <- format(d7_nuevo$Hora, "%H:%M:%S")
716 Fecha <- as.POSIXct(paste(d7_nuevo$Fecha, d7_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz="UTC")
717 d7_nuevo<-cbind(Fecha, d7_nuevo[,c(1,2)])
718 head(d7_nuevo[,c(1,2,3,4)],3)
719 ```
720
721
722 6 variables con una gran cantidad de NA's
723 ```{r, echo = FALSE, results = "hide"}
724 anyNA(d7_nuevo)
725 na_cols <- colSums(is.na(d7_nuevo))
726 na_cols <- na_cols[na_cols > 0]
727 na_cols
728 #which_rows_na <- which(rowSums(is.na(d7_nuevo[names(na_cols)])) > 0)
729
730 #d7_nuevo[which_rows_na, names(na_cols)]
731 ```
732
733 ### Variables de corrección de rangos de fechas
734
735 ```{r}
736 cat("Para irr \n")
737 as.POSIXct(d7_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
738 as.POSIXct(d7_irr$Fecha[length(d7_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
739 cat("\nPara bombas \n")
740 as.POSIXct(d7_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
741 as.POSIXct(d7_bombas$Fecha[length(d7_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")

```

```

742 cat("\nPara nuevo \n")
743 as.POSIXct(d7_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
744 as.POSIXct(d7_nuevo$Fecha[length(d7_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
745 ' ' '
746
747 '{r}
748 d7_irr_bombas <- d7_irr
749 d7_irr_nuevo <- d7_irr
750
751 # Fila resumen
752 cat(sprintf("d7_bombas: %d | d7_nuevo: %d | d7_irr_bombas: %d |
    d7_irr_nuevo: %d\n",
753             nrow(d7_bombas), nrow(d7_nuevo), nrow(d7_irr_bombas), nrow
              (d7_irr_nuevo)))
754 ' ' '
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769 ## 8. Datos de 03-12 a 20-12
770
771 Este rango de datos contiene las dos CT.
772
773
774
775 ### Irradiancia
776
777 '{r}
778 ruta_especifica_d8<-paste(ruta_datos, "/Datos8 de 03-12 a 20-12.xlsx",
    sep="")
779 d8_irr <- read_excel(ruta_especifica_d8, sheet = "Irradiancia")
780 head(d8_irr)
781 ' ' '
782
783 Contiene una fila de NA's
784 '{r, echo = FALSE, results = "hide"}
785 anyNA(d8_irr)
786 na_cols <- colSums(is.na(d8_irr))
787 na_cols <- na_cols[na_cols > 0]
788

```

```

789 which_rows_na <- which(rowSums(is.na(d8_irr[names(na_cols)])) > 0)
790
791 d8_irr[which_rows_na, ]
792 ```
793
794 ```{r}
795 d8_irr <- as.data.frame(lapply(d8_irr, na.spline), check.names = FALSE)
796 anyNA(d8_irr)
797 ```
798
799
800 ### CT Bombas
801
802
803 ```{r}
804 d8_bombas <- read_excel(ruta_especifica_d8, sheet = "CT Bombas")
805 #names(d8_bombas)
806
807 d8_bombas <- d8_bombas[, -ncol(d8_bombas)]
808 #names(d8_bombas)
809
810 head(d8_bombas[, c(1,2,3,4)], 3)
811 d8_bombas$Hora <- format(d8_bombas$Hora, "%H:%M:%S")
812 Fecha <- as.POSIXct(paste(d8_bombas$Fecha, d8_bombas$Hora), format="%Y-
      -%m-%d %H:%M:%S", tz = "UTC")
813 d8_bombas <- cbind(Fecha, d8_bombas[, -c(1,2)])
814 head(d8_bombas[, c(1,2,3,4)], 3)
815 ```
816
817
818 ```{r, echo = FALSE, results = "hide"}
819 anyNA(d8_bombas)
820 ```
821
822
823 ### CT Nuevo
824
825
826 ```{r}
827 d8_nuevo <- read_excel(ruta_especifica_d8, sheet = "CT Nuevo")
828 #names(d8_nuevo)
829
830 head(d8_nuevo[, c(1,2,3,4)], 3)
831 d8_nuevo$Hora <- format(d8_nuevo$Hora, "%H:%M:%S")
832 Fecha <- as.POSIXct(paste(d8_nuevo$Fecha, d8_nuevo$Hora), format="%Y-%
      m-%d %H:%M:%S", tz = "UTC")
833 d8_nuevo <- cbind(Fecha, d8_nuevo[, -c(1,2)])
834 head(d8_nuevo[, c(1,2,3,4)], 3)
835 ```
836
837

```

```

838   "'{r, echo = FALSE, results = "hide"}
839   anyNA(d8_nuevo)
840   "'
841
842   ### Variables de corrección de rangos de fechas
843
844   "'{r}
845   cat("Para irr \n")
846   as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
847   as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)], origin = "1970-01-01",
848     tz = "UTC")
849   cat("\nPara bombas \n")
850   as.POSIXct(d8_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
851   as.POSIXct(d8_bombas$Fecha[length(d8_bombas$Fecha)], origin = "
852     1970-01-01", tz = "UTC")
853   cat("\nPara nuevo \n")
854   as.POSIXct(d8_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
855   as.POSIXct(d8_nuevo$Fecha[length(d8_nuevo$Fecha)], origin = "
856     1970-01-01", tz = "UTC")
857   "'
858
859   "'{r}
860   cat("Para irr de bombas\n")
861   as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
862   as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)-1], origin = "1970-01-01"
863     , tz = "UTC")
864   cat("\nPara bombas \n")
865   as.POSIXct(d8_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
866   as.POSIXct(d8_bombas$Fecha[length(d8_bombas$Fecha)], origin = "
867     1970-01-01", tz = "UTC")
868   cat("\nPara irr de nuevo\n")
869   as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
870   as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)], origin = "1970-01-01",
871     tz = "UTC")
872   cat("\nPara nuevo \n")
873   as.POSIXct(d8_nuevo$Fecha[2], origin = "1970-01-01", tz = "UTC")
874   as.POSIXct(d8_nuevo$Fecha[length(d8_nuevo$Fecha)-2], origin = "
875     1970-01-01", tz = "UTC")
876   "'
877
878   "'{r}
879   # Convertir fechas a POSIXct en UTC
880   f_irr    <- as.POSIXct(d8_irr$Fecha, origin = "1970-01-01", tz = "UTC")
881   f_bomb    <- as.POSIXct(d8_bombas$Fecha, origin = "1970-01-01", tz = "
882     UTC")
883   f_nuev    <- as.POSIXct(d8_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
884     ")
885
886   # Recortes

```

```

880 d8_irr_bombas <- d8_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
      f_irr) - 1], ]
881 d8_bombas      <- d8_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(f_bomb
      , 1), ]
882 d8_irr_nuevo   <- d8_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1), ]
883 d8_nuevo        <- d8_nuevo[f_nuev >= f_nuev[2] & f_nuev <= f_nuev[
      length(f_nuev) - 2], ]
884
885 # Verificaci n de dimensiones
886 cat(sprintf("d8_bombas: %d | d8_nuevo: %d | d8_irr_bombas: %d |
      d8_irr_nuevo: %d\n",
887           nrow(d8_bombas), nrow(d8_nuevo), nrow(d8_irr_bombas), nrow
      (d8_irr_nuevo)))
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904 ## 9. Datos de 20-12 a 09-01
905
906 Este rango de datos contiene las dos CT.
907
908
909
910 ### Irradiancia
911
912 ```{r}
913 ruta_especifica_d9<-paste(ruta_datos, "/Datos9 de 20-12 a 09-01.xlsx",
      sep="")
914 d9_irr <- read_excel(ruta_especifica_d9, sheet = "Irradiancia")
915 head(d9_irr)
916 ```
917
918 Contiene dos filas de NA's y la columna de irradiancia de CT bombas
      contiene algo m s de 24 horas seguidas de NA's, mirar a ver que
      hacer con ello, pues interpolar aqui no le veo el sentido.
919 ```{r, echo = FALSE, results = "hide"}
920 anyNA(d9_irr)
921 na_cols <- colSums(is.na(d9_irr))
922 na_cols <- na_cols[na_cols > 0]

```

```

923
924 which_rows_na <- which(rowSums(is.na(d9_irr[names(na_cols)])) > 0)
925
926 d9_irr[which_rows_na, ]
927 ```
928
929
930
931 Habria que ejecutar el codigo este aunque sea para las 2 filas de NA's
932
933 ```{r}
934 #d9_irr <- as.data.frame(lapply(d9_irr, na.spline),check.names = FALSE
935 )
936 #anyNA(d9_irr)
937 ```
938
939 ### CT Bombas
940
941
942 ```{r , message = FALSE}
943 d9_bombas <- read_excel(ruta_especifica_d9, sheet = "CT Bombas")
944 #names(d9_bombas)
945
946 # Eliminar columnas desde "Column1" en adelante
947 columnas_borrar <- which(names(d9_bombas) == "...169")
948 if(length(columnas_borrar) > 0) {
949   d9_bombas <- d9_bombas[, 1:(columnas_borrar - 1)]
950 }
951
952 #names(d9_bombas)
953 head(d9_bombas[,c(1,2,3,4)],3)
954 d9_bombas$Hora <- format(d9_bombas$Hora, "%H:%M:%S")
955 Fecha <- as.POSIXct(paste(d9_bombas$Fecha, d9_bombas$Hora), format="%Y
956   -%m-%d %H:%M:%S", tz = "UTC")
957 d9_bombas<-cbind(Fecha, d9_bombas[,c(1,2)])
958 head(d9_bombas[,c(1,2,3,4)],3)
959 ```
960
961 ```{r, echo = FALSE, results = "hide"}
962 anyNA(d9_bombas)
963 ```
964
965 ### CT Nuevo
966
967
968 ```{r}
969 d9_nuevo <- read_excel(ruta_especifica_d9, sheet = "CT Nuevo")
970 #names(d9_nuevo)

```

```

971 head(d9_nuevo[,c(1,2,3,4)],3)
972 d9_nuevo$Hora <- format(d9_nuevo$Hora, "%H:%M:%S")
973 Fecha <- as.POSIXct(paste(d9_nuevo$Fecha, d9_nuevo$Hora), format="%Y-%
974 m-%d %H:%M:%S", tz = "UTC")
975 d9_nuevo<-cbind(Fecha, d9_nuevo[, -c(1,2)])
976 head(d9_nuevo[,c(1,2,3,4)],3)
977 ' ' '
978
979
980 '{r, echo = FALSE, results = "hide"}
981 anyNA(d9_nuevo)
982 ' ' '
983
984
985
986 ### Variables de corrección de rangos de fechas
987
988 '{r}
989 cat("Para irr \n")
990 as.POSIXct(d9_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
991 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)], origin = "1970-01-01",
992           tz = "UTC")
993 cat("\nPara bombas \n")
994 as.POSIXct(d9_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
995 as.POSIXct(d9_bombas$Fecha[length(d9_bombas$Fecha)], origin = "
996           1970-01-01", tz = "UTC")
997 cat("\nPara nuevo \n")
998 as.POSIXct(d9_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
999 as.POSIXct(d9_nuevo$Fecha[length(d9_nuevo$Fecha)], origin = "
1000           1970-01-01", tz = "UTC")
1001 ' ' '
1002
1003 '{r}
1004 cat("Para irr de bombas\n")
1005 as.POSIXct(d9_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
1006 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)-7], origin = "1970-01-01"
1007           , tz = "UTC")
1008 cat("\nPara bombas \n")
1009 as.POSIXct(d9_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1010 as.POSIXct(d9_bombas$Fecha[length(d9_bombas$Fecha)], origin = "
1011           1970-01-01", tz = "UTC")
1012 cat("\nPara irr de nuevo\n")
1013 as.POSIXct(d9_irr$Fecha[5], origin = "1970-01-01", tz = "UTC")
1014 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)], origin = "1970-01-01",
1015           tz = "UTC")
1016 cat("\nPara nuevo \n")
1017 as.POSIXct(d9_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1018 as.POSIXct(d9_nuevo$Fecha[length(d9_nuevo$Fecha)], origin = "
1019           1970-01-01", tz = "UTC")
1020 ' ' '

```

```

1014 |
1015 |
1016 |   ```{r}
1017 |   # Convertir fechas a POSIXct en UTC
1018 |   f_irr    <- as.POSIXct(d9_irr$Fecha, origin = "1970-01-01", tz = "UTC")
1019 |   f_bomb   <- as.POSIXct(d9_bombas$Fecha, origin = "1970-01-01", tz = "
1020 |   UTC")
1021 |   f_nuev   <- as.POSIXct(d9_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
1022 |   ")
1023 |
1024 |   # Recortes aplicados
1025 |   d9_irr_bombas <- d9_irr[f_irr >= f_irr[2] & f_irr <= f_irr[length(
1026 |   f_irr) - 7], ]
1027 |   d9_bombas     <- d9_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(f_bomb
1028 |   , 1), ]
1029 |   d9_irr_nuevo  <- d9_irr[f_irr >= f_irr[5] & f_irr <= tail(f_irr, 1), ]
1030 |   d9_nuevo      <- d9_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev,
1031 |   1), ]
1032 |
1033 |   # Verificaci n
1034 |   cat(sprintf("d9_bombas: %d | d9_nuevo: %d | d9_irr_bombas: %d |
1035 |   d9_irr_nuevo: %d\n",
1036 |   nrow(d9_bombas), nrow(d9_nuevo), nrow(d9_irr_bombas), nrow
1037 |   (d9_irr_nuevo)))
1038 |
1039 |   ```
1040 |
1041 |
1042 |
1043 |   Una vez hecho todo esto, si se pueden eliminar bien las filas
1044 |   ```{r}
1045 |   filas_na <- which(is.na(d9_irr_bombas$Irradiancia_CTBombas))
1046 |   filas_eliminar <- filas_na[1:(length(filas_na) - 2)]
1047 |   filas_eliminar
1048 |   ```
1049 |
1050 |   ```{r}
1051 |   d9_irr_bombas <- d9_irr_bombas[-filas_eliminar, ]
1052 |   d9_bombas <- d9_bombas[-filas_eliminar, ]
1053 |   ```
1054 |
1055 |   ```{r}
1056 |   d9_irr_bombas <- as.data.frame(lapply(d9_irr_bombas, na.spline), check.
1057 |   names = FALSE)
1058 |   anyNA(d9_irr_bombas)
1059 |   ```
1060 |
1061 |   ```{r}
1062 |   d9_irr_nuevo <- d9_irr_nuevo[, !names(d9_irr_nuevo) %in% "
1063 |   Irradiancia_CTBombas"]
1064 |   ```

```

```

1056   ```{r}
1057   anyNA(d9_irr_nuevo)
1058   d9_irr_nuevo <- as.data.frame(lapply(d9_irr_nuevo, na.spline), check.
1059     names = FALSE)
1060   anyNA(d9_irr_nuevo)
1061   ```
1062
1063   ```{r}
1064   cat(sprintf("d9_bombas: %d | d9_nuevo: %d | d9_irr_bombas: %d |
1065     d9_irr_nuevo: %d\n",
1066       nrow(d9_bombas), nrow(d9_nuevo), nrow(d9_irr_bombas), nrow
1067         (d9_irr_nuevo)))
1068   ```
1069
1070
1071
1072
1073   ## 10. Datos de 09-01 a 30-01
1074
1075   Este rango de datos contiene las dos CT.
1076
1077
1078
1079   ### Irradiancia
1080
1081   ```{r}
1082   ruta_especifica_d10<-paste(ruta_datos, "/Datos10 de 09-01 a 30-01.xlsx
1083     ", sep="")
1084   d10_irr <- read_excel(ruta_especifica_d10, sheet = "Irradiancia")
1085   head(d10_irr)
1086   ```
1087
1088   Contiene una fila de NA's
1089   ```{r, echo = FALSE, results = "hide"}
1090   anyNA(d10_irr)
1091   na_cols <- colSums(is.na(d10_irr))
1092   na_cols <- na_cols[na_cols > 0]
1093
1094   which_rows_na <- which(rowSums(is.na(d10_irr[na_cols]))) > 0)
1095   d10_irr[which_rows_na, ]
1096   ```
1097
1098   ```{r}
1099   d10_irr <- as.data.frame(lapply(d10_irr, na.spline), check.names =
1100     FALSE)
1101   anyNA(d10_irr)
1102   ```

```

```

1102 |
1103 |
1104 | ### CT Bombas
1105 |
1106 |
1107 | ```{r}
1108 | d10_bombas <- read_excel(ruta_especifica_d10, sheet = "CT Bombas")
1109 | #names(d10_bombas)
1110 |
1111 | # Eliminar columnas desde "Column1" en adelante
1112 | columnas_borrar <- which(names(d10_bombas) == "...169")
1113 | if(length(columnas_borrar) > 0) {
1114 |   d10_bombas <- d10_bombas[, 1:(columnas_borrar - 1)]
1115 | }
1116 |
1117 | #names(d10_bombas)
1118 | head(d10_bombas[,c(1,2,3,4)],3)
1119 | d10_bombas$Hora <- format(d10_bombas$Hora, "%H:%M:%S")
1120 | Fecha <- as.POSIXct(paste(d10_bombas$Fecha, d10_bombas$Hora), format="
1121 |   %Y-%m-%d %H:%M:%S", tz = "UTC")
1122 | d10_bombas<-cbind(Fecha, d10_bombas[, -c(1,2)])
1123 | head(d10_bombas[,c(1,2,3,4)],3)
1124 | ```
1125 | ```{r, echo = FALSE, results = "hide"}
1126 | anyNA(d10_bombas)
1127 | ```
1128 |
1129 |
1130 | ### CT Nuevo
1131 |
1132 |
1133 | ```{r}
1134 | d10_nuevo <- read_excel(ruta_especifica_d10, sheet = "CT Nuevo")
1135 | #names(d10_nuevo)
1136 |
1137 | head(d10_nuevo[,c(1,2,3,4)],3)
1138 | d10_nuevo$Hora <- format(d10_nuevo$Hora, "%H:%M:%S")
1139 | Fecha <- as.POSIXct(paste(d10_nuevo$Fecha, d10_nuevo$Hora), format="%Y
1140 |   -%m-%d %H:%M:%S", tz = "UTC")
1141 | d10_nuevo<-cbind(Fecha, d10_nuevo[, -c(1,2)])
1142 | head(d10_nuevo[,c(1,2,3,4)],3)
1143 | ```
1144 |
1145 | ```{r, echo = FALSE, results = "hide"}
1146 | anyNA(d10_nuevo)
1147 | ```
1148 |
1149 |
1150 | ### Variables de corrección de rangos de fechas

```

```

1151   "'{r}
1152   cat("Para irr \n")
1153   as.POSIXct(d10_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1154   as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)], origin = "1970-01-01"
1155     , tz = "UTC")
1156   cat("\nPara bombas \n")
1157   as.POSIXct(d10_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1158   as.POSIXct(d10_bombas$Fecha[length(d10_bombas$Fecha)], origin = "
1159     1970-01-01", tz = "UTC")
1160   cat("\nPara nuevo \n")
1161   as.POSIXct(d10_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1162   as.POSIXct(d10_nuevo$Fecha[length(d10_nuevo$Fecha)], origin = "
1163     1970-01-01", tz = "UTC")
1164   "'{r}
1165   cat("Para irr de bombas\n")
1166   as.POSIXct(d10_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1167   as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)-134], origin = "
1168     1970-01-01", tz = "UTC")
1169   cat("\nPara bombas \n")
1170   as.POSIXct(d10_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1171   as.POSIXct(d10_bombas$Fecha[length(d10_bombas$Fecha)], origin = "
1172     1970-01-01", tz = "UTC")
1173   cat("\nPara irr de nuevo\n")
1174   as.POSIXct(d10_irr$Fecha[7], origin = "1970-01-01", tz = "UTC")
1175   as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)-1], origin = "
1176     1970-01-01", tz = "UTC")
1177   cat("\nPara nuevo \n")
1178   as.POSIXct(d10_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1179   as.POSIXct(d10_nuevo$Fecha[length(d10_nuevo$Fecha)], origin = "
1180     1970-01-01", tz = "UTC")
1181   "'
1182   # Convertir fechas a POSIXct en UTC
1183   f_irr    <- as.POSIXct(d10_irr$Fecha, origin = "1970-01-01", tz = "UTC"
1184     )
1185   f_bomb   <- as.POSIXct(d10_bombas$Fecha, origin = "1970-01-01", tz = "
1186     UTC")
1187   f_nuev   <- as.POSIXct(d10_nuevo$Fecha, origin = "1970-01-01", tz = "
1188     UTC")
1189   # Recortes
1190   d10_irr_bombas <- d10_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
1191     f_irr) - 134], ]
1192   d10_bombas     <- d10_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
1193     f_bomb, 1), ]
1194   d10_irr_nuevo  <- d10_irr[f_irr >= f_irr[7] & f_irr <= f_irr[length(

```

```

    f_irr) - 1], ]
1190 d10_nuevo      <- d10_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
    f_nuev, 1), ]
1191
1192 # Verificaci n
1193 cat(sprintf("d10_bombas: %d | d10_nuevo: %d | d10_irr_bombas: %d |
    d10_irr_nuevo: %d\n",
1194           nrow(d10_bombas), nrow(d10_nuevo), nrow(d10_irr_bombas),
    nrow(d10_irr_nuevo)))
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208 ## 11. Datos de 30-01 a 18-02
1209
1210 Este rango de datos solo contiene CT nuevo.
1211
1212
1213
1214 ### Irradiancia
1215
1216 ```{r}
1217 ruta_especifica_d11<-paste(ruta_datos, "/Datos11 de 30-01 a 18-02.xlsx
    ",sep="")
1218 d11_irr <- read_excel(ruta_especifica_d11, sheet = "Irradiancia")
1219 head(d11_irr)
1220
1221
1222 Contiene una fila de NA's
1223 ```{r, echo = FALSE, results = "hide"}
1224 anyNA(d11_irr)
1225 na_cols <- colSums(is.na(d11_irr))
1226 na_cols <- na_cols[na_cols > 0]
1227
1228 which_rows_na <- which(rowSums(is.na(d11_irr[names(na_cols)])) > 0)
1229
1230 d11_irr[which_rows_na, ]
1231
1232
1233 ```{r}
1234 d11_irr <- as.data.frame(lapply(d11_irr, na.spline),check.names =
    FALSE)

```

```

1235 anyNA(d11_irr)
1236 '""
1237
1238
1239 ### CT Nuevo
1240
1241
1242 ""{r}
1243 d11_nuevo <- read_excel(ruta_especifica_d11, sheet = "CT Nuevo")
1244 #names(d11_nuevo)
1245
1246 head(d11_nuevo[,c(1,2,3,4)],3)
1247 d11_nuevo$Hora <- format(d11_nuevo$Hora, "%H:%M:%S")
1248 Fecha <- as.POSIXct(paste(d11_nuevo$Fecha, d11_nuevo$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
1249 d11_nuevo<-cbind(Fecha, d11_nuevo[, -c(1,2)])
1250 head(d11_nuevo[,c(1,2,3,4)],3)
1251 '""
1252
1253
1254 ""{r, echo = FALSE, results = "hide"}
1255 anyNA(d11_nuevo)
1256 '""
1257
1258
1259 ### Variables de correcci n de rangos de fechas
1260
1261 Hay descuadre pero e cambio da igual en este caso, pues el que
    corresponde sigue siendo el m s pr ximo.
1262 ""{r}
1263 cat("Para irr \n")
1264 as.POSIXct(d11_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1265 as.POSIXct(d11_irr$Fecha[length(d11_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1266 cat("\nPara nuevo \n")
1267 as.POSIXct(d11_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1268 as.POSIXct(d11_nuevo$Fecha[length(d11_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1269 '""
1270
1271
1272 ""{r}
1273 cat("Para irr \n")
1274 as.POSIXct(d11_irr$Fecha[12], origin = "1970-01-01", tz = "UTC")
1275 as.POSIXct(d11_irr$Fecha[length(d11_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1276 cat("\nPara nuevo \n")
1277 as.POSIXct(d11_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1278 as.POSIXct(d11_nuevo$Fecha[length(d11_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1279 '""

```

```

1280 |
1281 |
1282 |   '{r}
1283 | # Convertir fechas a POSIXct en UTC
1284 | f_irr  <- as.POSIXct(d11_irr$Fecha, origin = "1970-01-01", tz = "UTC")
1285 | f_nuev <- as.POSIXct(d11_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
1286 |   ")
1287 |
1288 | # Recortes
1289 | d11_irr_nuevo  <- d11_irr[f_irr >= f_irr[12] & f_irr <= tail(f_irr,
1290 |   1), ]
1291 | d11_nuevo <- d11_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1)
1292 |   , ]
1293 |
1294 | # Verificaci n
1295 | cat(sprintf("d11_irr_nuevo: %d | d11_nuevo: %d\n", nrow(d11_irr_nuevo)
1296 |   , nrow(d11_nuevo)))
1297 |
1298 |
1299 |
1300 |
1301 |
1302 |
1303 |
1304 |
1305 | ## 12. Datos de 18-02 a 12-03
1306 |
1307 | Este rango de datos contiene las dos CT.
1308 |
1309 |
1310 | ### Irradiancia
1311 |
1312 |   '{r}
1313 | ruta_especifica_d12<-paste(ruta_datos, "/Datos12 de 18-02 a 12-03.xlsx
1314 |   ",sep="")
1315 | d12_irr <- read_excel(ruta_especifica_d12, sheet = "Irradiancia")
1316 | head(d12_irr)
1317 |
1318 |
1319 | Contiene tres filas seguidas de NA's
1320 |   '{r, echo = FALSE, results = "hide"}
1321 | anyNA(d12_irr)
1322 | na_cols <- colSums(is.na(d12_irr))
1323 | na_cols <- na_cols[na_cols > 0]
1324 |
1325 | which_rows_na <- which(rowSums(is.na(d12_irr[names(na_cols)]))) > 0)

```

```

1326 d12_irr[which_rows_na, ]
1327 ""
1328
1329 ""{r}
1330 d12_irr <- as.data.frame(lapply(d12_irr, na.spline), check.names =
      FALSE)
1331 anyNA(d12_irr)
1332 ""
1333
1334
1335 ### CT Bombas
1336
1337
1338 ""{r , message = FALSE}
1339 d12_bombas <- read_excel(ruta_especifica_d12, sheet = "CT Bombas")
1340 #names(d12_bombas)
1341
1342 # Eliminar columnas desde "Column1" en adelante
1343 columnas_borrar <- which(names(d12_bombas) == "...169")
1344 if(length(columnas_borrar) > 0) {
1345   d12_bombas <- d12_bombas[, 1:(columnas_borrar - 1)]
1346 }
1347
1348 #names(d12_bombas)
1349 head(d12_bombas[,c(1,2,3,4)],3)
1350 d12_bombas$Hora <- format(d12_bombas$Hora, "%H:%M:%S")
1351 Fecha <- as.POSIXct(paste(d12_bombas$Fecha, d12_bombas$Hora), format="
      %Y-%m-%d %H:%M:%S", tz = "UTC")
1352 d12_bombas<-cbind(Fecha, d12_bombas[, -c(1,2)])
1353 head(d12_bombas[,c(1,2,3,4)],3)
1354 ""
1355
1356
1357 ""{r, echo = FALSE, results = "hide"}
1358 anyNA(d12_bombas)
1359 ""
1360
1361
1362 ### CT Nuevo
1363
1364
1365 ""{r}
1366 d12_nuevo <- read_excel(ruta_especifica_d12, sheet = "CT Nuevo")
1367 #names(d12_nuevo)
1368
1369 head(d12_nuevo[,c(1,2,3,4)],3)
1370 d12_nuevo$Hora <- format(d12_nuevo$Hora, "%H:%M:%S")
1371 Fecha <- as.POSIXct(paste(d12_nuevo$Fecha, d12_nuevo$Hora), format="%Y
      -%m-%d %H:%M:%S", tz = "UTC")
1372 d12_nuevo<-cbind(Fecha, d12_nuevo[, -c(1,2)])
1373 head(d12_nuevo[,c(1,2,3,4)],3)

```

```

1374  ' ' '
1375
1376
1377  '{r, echo = FALSE, results = "hide"}
1378  anyNA(d12_nuevo)
1379  ' ' '
1380
1381
1382
1383  ### Variables de corrección de rangos de fechas
1384
1385  '{r}
1386  cat("Para irr \n")
1387  as.POSIXct(d12_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1388  as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)], origin = "1970-01-01"
1389    , tz = "UTC")
1389  cat("\nPara bombas \n")
1390  as.POSIXct(d12_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1391  as.POSIXct(d12_bombas$Fecha[length(d12_bombas$Fecha)], origin = "
1392    1970-01-01", tz = "UTC")
1392  cat("\nPara nuevo \n")
1393  as.POSIXct(d12_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1394  as.POSIXct(d12_nuevo$Fecha[length(d12_nuevo$Fecha)], origin = "
1395    1970-01-01", tz = "UTC")
1395  ' ' '
1396
1397  '{r}
1398  cat("Para irr de bombas\n")
1399  as.POSIXct(d12_irr$Fecha[1008], origin = "1970-01-01", tz = "UTC")
1400  as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)-4], origin = "
1401    1970-01-01", tz = "UTC")
1401  cat("\nPara bombas \n")
1402  as.POSIXct(d12_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1403  as.POSIXct(d12_bombas$Fecha[length(d12_bombas$Fecha)], origin = "
1404    1970-01-01", tz = "UTC")
1404  cat("\nPara irr de nuevo\n")
1405  as.POSIXct(d12_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1406  as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)], origin = "1970-01-01"
1407    , tz = "UTC")
1407  cat("\nPara nuevo \n")
1408  as.POSIXct(d12_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1409  as.POSIXct(d12_nuevo$Fecha[length(d12_nuevo$Fecha)], origin = "
1410    1970-01-01", tz = "UTC")
1410  ' ' '
1411
1412
1413  '{r}
1414  # Convertir fechas a POSIXct en UTC
1415  f_irr    <- as.POSIXct(d12_irr$Fecha, origin = "1970-01-01", tz = "UTC"
1416    )
1416  f_bomb   <- as.POSIXct(d12_bombas$Fecha, origin = "1970-01-01", tz = "

```

```

1417     UTC")
1418
1419 # Recortes
1420 d12_irr_bombas <- d12_irr[f_irr >= f_irr[1008] & f_irr <= f_irr[length
    (f_irr) - 4], ]
1421 d12_bombas <- d12_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
    f_bomb, 1), ]
1422 d12_irr_nuevo <- d12_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1),
    ]
1423 d12_nuevo <- d12_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
    f_nuev, 1), ]
1424
1425 # Verificaci n
1426 cat(sprintf("d12_bombas: %d | d12_nuevo: %d | d12_irr_bombas: %d |
    d12_irr_nuevo: %d\n",
1427             nrow(d12_bombas), nrow(d12_nuevo), nrow(d12_irr_bombas),
                nrow(d12_irr_nuevo)))
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440 ## 13. Datos de 12-03 a 03-04
1441
1442 Este rango de datos contiene las dos CT.
1443
1444
1445 ### Irradiancia
1446
1447 ```{r}
1448 ruta_especifica_d13<-paste(ruta_datos, "/Datos13 de 12-03 a 03-04.xlsx
    ",sep="")
1449 d13_irr <- read_excel(ruta_especifica_d13, sheet = "Irradiancia")
1450 head(d13_irr)
1451 ```
1452
1453
1454 ```{r, echo = FALSE, results = "hide"}
1455 anyNA(d13_irr)
1456 ```
1457
1458 ### CT Bombas

```

```

1459 |
1460 |
1461 | ‘‘{r , message = FALSE}
1462 | d13_bombas <- read_excel(ruta_especifica_d13, sheet = "CT Bombas")
1463 | #names(d13_bombas)
1464 | fila_inicial <- 2541 # Cambia este valor seg n d nde quieras
1465 |     aplicar el ajuste
1466 |
1467 | # Sumar 60 minutos a partir de la fila indicada
1468 | d13_bombas$Hora[fila_inicial:nrow(d13_bombas)] <- d13_bombas$Hora[
1469 |     fila_inicial:nrow(d13_bombas)] + 60*60 # 60 minutos en segundos
1470 | # Eliminar columnas desde "Column1" en adelante
1471 | columnas_borrar <- which(names(d13_bombas) == "...169")
1472 | if(length(columnas_borrar) > 0) {
1473 |     d13_bombas <- d13_bombas[, 1:(columnas_borrar - 1)]
1474 | }
1475 | #names(d13_bombas)
1476 | head(d13_bombas[,c(1,2,3,4)],3)
1477 | d13_bombas$Hora <- format(d13_bombas$Hora, "%H:%M:%S")
1478 | Fecha <- as.POSIXct(paste(d13_bombas$Fecha, d13_bombas$Hora), format="
1479 |     %Y-%m-%d %H:%M:%S", tz="UTC")
1480 | d13_bombas<-cbind(Fecha, d13_bombas[, -c(1,2)])
1481 | head(d13_bombas[,c(1,2,3,4)],3)
1482 | ‘‘‘
1483 |
1484 | Problema interesante a solucionar, 6 horas dan NA, esto se debe justo
1485 |     al cambio horario, que el formato respeta que esas horas como tal
1486 |     no existieron y por tanto de ahi el NA. Esto puede ser un problema
1487 |     que se alargue a todos los dem s si es que la hora registrada por
1488 |     las estaciones no se tiene en cuenta el cambio horario.
1489 | ‘‘{r, echo = FALSE, results = "hide"}
1490 | anyNA(d13_bombas)
1491 | na_cols <- colSums(is.na(d13_bombas))
1492 | na_cols <- na_cols[na_cols > 0]
1493 | na_cols
1494 | which_rows_na <- which(rowSums(is.na(d13_bombas[names(na_cols)])) > 0)
1495 | d13_bombas[which_rows_na, names(na_cols)]
1496 | ‘‘‘
1497 |
1498 | ### CT Nuevo
1499 |
1500 | ‘‘{r}
1501 | d13_nuevo <- read_excel(ruta_especifica_d13, sheet = "CT Nuevo")
1502 | #names(d13_nuevo)
1503 | fila_inicial <- 2538 # Cambia este valor seg n d nde quieras
1504 |     aplicar el ajuste
1505 |

```

```

1502 # Sumar 60 minutos a partir de la fila indicada
1503 d13_nuevo$Hora[fila_inicial:nrow(d13_nuevo)] <- d13_nuevo$Hora[
      fila_inicial:nrow(d13_nuevo)] + 60*60 # 60 minutos en segundos
1504
1505 head(d13_nuevo[,c(1,2,3,4)],3)
1506 d13_nuevo$Hora <- format(d13_nuevo$Hora, "%H:%M:%S")
1507 Fecha <- as.POSIXct(paste(d13_nuevo$Fecha, d13_nuevo$Hora), format="%Y
      -%m-%d %H:%M:%S", tz="UTC")
1508 d13_nuevo<-cbind(Fecha, d13_nuevo[, -c(1,2)])
1509 head(d13_nuevo[,c(1,2,3,4)],3)
1510 ```
1511
1512 Lo mismo de nuevo.
1513 ```{r, echo = FALSE, results = "hide"}
1514 anyNA(d13_nuevo)
1515 na_cols <- colSums(is.na(d13_nuevo))
1516 na_cols <- na_cols[na_cols > 0]
1517 na_cols
1518 which_rows_na <- which(rowSums(is.na(d13_nuevo[names(na_cols)])) > 0)
1519
1520 d13_nuevo[which_rows_na, names(na_cols)]
1521 ```
1522
1523
1524 ### Variables de correcci n de rangos de fechas
1525
1526 ```{r}
1527 cat("Para irr \n")
1528 as.POSIXct(d13_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1529 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)], origin = "1970-01-01"
      , tz = "UTC")
1530 cat("\nPara bombas \n")
1531 as.POSIXct(d13_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1532 as.POSIXct(d13_bombas$Fecha[length(d13_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1533 cat("\nPara nuevo \n")
1534 as.POSIXct(d13_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1535 as.POSIXct(d13_nuevo$Fecha[length(d13_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1536 ```
1537
1538 ```{r}
1539 cat("Para irr de bombas\n")
1540 as.POSIXct(d13_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1541 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)-306], origin = "
      1970-01-01", tz = "UTC")
1542 cat("\nPara bombas \n")
1543 as.POSIXct(d13_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1544 as.POSIXct(d13_bombas$Fecha[length(d13_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1545 cat("\nPara irr de nuevo\n")

```

```

1546 as.POSIXct(d13_irr$Fecha[4], origin = "1970-01-01", tz = "UTC")
1547 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1548 cat("\nPara nuevo \n")
1549 as.POSIXct(d13_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1550 as.POSIXct(d13_nuevo$Fecha[length(d13_nuevo$Fecha)-5], origin = "
    1970-01-01", tz = "UTC")
1551 '""
1552
1553
1554 ''{r}
1555 # Convertir fechas
1556 f_irr    <- as.POSIXct(d13_irr$Fecha, origin = "1970-01-01", tz = "UTC"
    )
1557 f_bomba  <- as.POSIXct(d13_bombas$Fecha, origin = "1970-01-01", tz = "
    UTC")
1558 f_nuevo  <- as.POSIXct(d13_nuevo$Fecha, origin = "1970-01-01", tz = "
    UTC")
1559
1560 # Recortes
1561 d13_nuevo    <- d13_nuevo[f_nuevo <= f_nuevo[length(f_nuevo)-5], ]
1562 d13_irr_bombas <- d13_irr[f_irr <= f_irr[length(f_irr)-306], ]
1563 d13_irr_nuevo  <- d13_irr[f_irr >= f_irr[4], ]
1564
1565 # Imprimir n mero de filas
1566 cat(sprintf("d13_bombas: %d | d13_irr_bombas: %d | d13_nuevo: %d |
    d13_irr_nuevo: %d\n",
1567             nrow(d13_bombas), nrow(d13_irr_bombas), nrow(d13_nuevo),
    nrow(d13_irr_nuevo)))
1568 '""
1569
1570
1571
1572
1573
1574
1575 ## 14. Datos de 03-04 a 24-04
1576
1577 Este rango de datos contiene las dos CT.
1578
1579
1580 ### Irradiancia
1581
1582 ''{r}
1583 ruta_especifica_d14<-paste(ruta_datos, "/Datos14 de 03-04 a 24-04.xlsx
    ",sep="")
1584 d14_irr <- read_excel(ruta_especifica_d14, sheet = "Irradiancia")
1585 head(d14_irr)
1586 '""
1587
1588

```

```

1589   '{r, echo = FALSE, results = "hide"}
1590 anyNA(d14_irr)
1591   '
1592
1593
1594   ### CT Bombas
1595
1596
1597   '{r , message = FALSE}
1598   d14_bombas <- read_excel(ruta_especifica_d14, sheet = "CT Bombas")
1599   #names(d14_bombas)
1600
1601   # Eliminar columnas desde "Column1" en adelante
1602   columnas_borrar <- which(names(d14_bombas) == "...169")
1603   if(length(columnas_borrar) > 0) {
1604     d14_bombas <- d14_bombas[, 1:(columnas_borrar - 1)]
1605   }
1606
1607   #names(d14_bombas)
1608   head(d14_bombas[,c(1,2,3,4)],3)
1609   d14_bombas$Hora <- format(d14_bombas$Hora, "%H:%M:%S")
1610   Fecha <- as.POSIXct(paste(d14_bombas$Fecha, d14_bombas$Hora), format="
    %Y-%m-%d %H:%M:%S", tz = "UTC")
1611   d14_bombas<-cbind(Fecha, d14_bombas[, -c(1,2)])
1612   head(d14_bombas[,c(1,2,3,4)],3)
1613   '
1614
1615
1616   '{r, echo = FALSE, results = "hide"}
1617   anyNA(d14_bombas)
1618   '
1619
1620
1621   ### CT Nuevo
1622
1623
1624   '{r}
1625   d14_nuevo <- read_excel(ruta_especifica_d14, sheet = "CT Nuevo")
1626   #names(d14_nuevo)
1627
1628   head(d14_nuevo[,c(1,2,3,4)],3)
1629   d14_nuevo$Hora <- format(d14_nuevo$Hora, "%H:%M:%S")
1630   Fecha <- as.POSIXct(paste(d14_nuevo$Fecha, d14_nuevo$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
1631   d14_nuevo<-cbind(Fecha, d14_nuevo[, -c(1,2)])
1632   head(d14_nuevo[,c(1,2,3,4)],3)
1633   '
1634
1635
1636   '{r, echo = FALSE, results = "hide"}
1637   anyNA(d14_nuevo)

```

```

1638  ' ' '
1639
1640
1641  ### Variables de corrección de rangos de fechas
1642
1643  ' ' '{r}
1644  cat("Para irr \n")
1645  as.POSIXct(d14_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1646  as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1647  cat("\nPara bombas \n")
1648  as.POSIXct(d14_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1649  as.POSIXct(d14_bombas$Fecha[length(d14_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1650  cat("\nPara nuevo \n")
1651  as.POSIXct(d14_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1652  as.POSIXct(d14_nuevo$Fecha[length(d14_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1653  ' ' '
1654
1655  ' ' '{r}
1656  cat("Para irr de bombas\n")
1657  as.POSIXct(d14_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1658  as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)-169], origin = "
    1970-01-01", tz = "UTC")
1659  cat("\nPara bombas \n")
1660  as.POSIXct(d14_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1661  as.POSIXct(d14_bombas$Fecha[length(d14_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1662  cat("\nPara irr de nuevo\n")
1663  as.POSIXct(d14_irr$Fecha[5], origin = "1970-01-01", tz = "UTC")
1664  as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)-1], origin = "
    1970-01-01", tz = "UTC")
1665  cat("\nPara nuevo \n")
1666  as.POSIXct(d14_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1667  as.POSIXct(d14_nuevo$Fecha[length(d14_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1668  ' ' '
1669
1670  ' ' '{r}
1671  # Convertir fechas a POSIXct en UTC
1672  f_irr    <- as.POSIXct(d14_irr$Fecha, origin = "1970-01-01", tz = "UTC"
    )
1673  f_bomb   <- as.POSIXct(d14_bombas$Fecha, origin = "1970-01-01", tz = "
    UTC")
1674  f_nuev   <- as.POSIXct(d14_nuevo$Fecha, origin = "1970-01-01", tz = "
    UTC")
1675
1676  # Recortes
1677  d14_irr_bombas <- d14_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
    f_irr) - 169], ]

```

```

1678 d14_bombas      <- d14_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
      f_bomb, 1), ]
1679 d14_irr_nuevo  <- d14_irr[f_irr >= f_irr[5] & f_irr <= f_irr[length(
      f_irr) - 1], ]
1680 d14_nuevo      <- d14_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
      f_nuev, 1), ]
1681
1682 # Verificaci n
1683 cat(sprintf("d14_bombas: %d | d14_nuevo: %d | d14_irr_bombas: %d |
      d14_irr_nuevo: %d\n",
1684           nrow(d14_bombas), nrow(d14_nuevo), nrow(d14_irr_bombas),
           nrow(d14_irr_nuevo)))
1685
1686 “ “ “
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703 ## 15. Datos de 24-04 a 12-05
1704
1705 Este rango de datos solo contiene CT bombas.
1706
1707
1708 ### Irradiancia
1709
1710 “ “ {r}
1711 ruta_especifica_d15<-paste(ruta_datos, "/Datos15 de 24-04 a 12-05.xlsx
      ", sep=" ")
1712 d15_irr <- read_excel(ruta_especifica_d15, sheet = "Irradiancia")
1713 head(d15_irr)
1714 “ “ “
1715
1716 Contiene 4 horas y media de NA's
1717 “ “ {r, echo = FALSE, results = "hide"}
1718 anyNA(d15_irr)
1719 na_cols <- colSums(is.na(d15_irr))
1720 na_cols <- na_cols[na_cols > 0]
1721
1722 which_rows_na <- which(rowSums(is.na(d15_irr[names(na_cols)])) > 0)

```

```

1723
1724 d15_irr[which_rows_na, ]
1725 '""
1726
1727 Son demasiadas filas para hacer interpolaci n yo creo.
1728 '""{r}
1729 #d15_irr <- as.data.frame(lapply(d15_irr, na.spline),check.names =
      FALSE)
1730 #anyNA(d15_irr)
1731 '""
1732
1733
1734
1735 ### CT Bombas
1736
1737
1738 '""{r , message = FALSE}
1739 d15_bombas <- read_excel(ruta_especifica_d15, sheet = "CT Bombas")
1740 #names(d15_bombas)
1741
1742 # Eliminar columnas desde "Column1" en adelante
1743 columnas_borrar <- which(names(d15_bombas) == "...169")
1744 if(length(columnas_borrar) > 0) {
1745   d15_bombas <- d15_bombas[, 1:(columnas_borrar - 1)]
1746 }
1747
1748 #names(d15_bombas)
1749 head(d15_bombas[,c(1,2,3,4)],3)
1750 d15_bombas$Hora <- format(d15_bombas$Hora, "%H:%M:%S")
1751 Fecha <- as.POSIXct(paste(d15_bombas$Fecha, d15_bombas$Hora), format="
      %Y-%m-%d %H:%M:%S", tz = "UTC")
1752 d15_bombas<-cbind(Fecha, d15_bombas[, -c(1,2)])
1753 head(d15_bombas[,c(1,2,3,4)],3)
1754 '""
1755
1756
1757 '""{r, echo = FALSE, results = "hide"}
1758 anyNA(d15_bombas)
1759 '""
1760
1761
1762 ### Variables de correcci n de rangos de fechas
1763
1764 '""{r}
1765 cat("Para irr \n")
1766 as.POSIXct(d15_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1767 as.POSIXct(d15_irr$Fecha[length(d15_irr$Fecha)], origin = "1970-01-01"
      , tz = "UTC")
1768 cat("\nPara bombas \n")
1769 as.POSIXct(d15_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1770 as.POSIXct(d15_bombas$Fecha[length(d15_bombas$Fecha)], origin = "

```

```

1771     1970-01-01", tz = "UTC")
1772   ""
1773
1774   ""{r}
1775   d15_irr_bombas <- d15_irr
1776
1777   cat(sprintf("d15_bombas: %d | d15_irr_bombas: %d\n",
1778               nrow(d15_bombas), nrow(d15_irr_bombas)))
1779   ""
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793   ## 16. Datos de 12-05 a 27-05
1794
1795   Este rango de datos contiene las dos CT.
1796
1797
1798   ### Irradiancia
1799
1800   ""{r}
1801   ruta_especifica_d16<-paste(ruta_datos, "/Datos16 de 12-05 a 27-05.xlsx",
1802                               sep="")
1803   d16_irr <- read_excel(ruta_especifica_d16, sheet = "Irradiancia")
1804   head(d16_irr)
1805   ""
1806
1807   Contiene una fila de NA's
1808   ""{r, echo = FALSE, results = "hide"}
1809   anyNA(d16_irr)
1810   na_cols <- colSums(is.na(d16_irr))
1811   na_cols <- na_cols[na_cols > 0]
1812
1813   which_rows_na <- which(rowSums(is.na(d16_irr[names(na_cols)])) > 0)
1814
1815   d16_irr[which_rows_na, ]
1816   ""
1817
1818   ""{r}
1819   d16_irr <- as.data.frame(lapply(d16_irr, na.spline), check.names =
1820                               FALSE)

```

```

1819 anyNA(d16_irr)
1820 ' ' '
1821
1822
1823 ### CT Bombas
1824
1825
1826 '{r}
1827 d16_bombas <- read_excel(ruta_especifica_d16, sheet = "CT Bombas")
1828 #names(d16_bombas)
1829
1830 # Eliminar columnas desde "Column1" en adelante
1831 columnas_borrar <- which(names(d16_bombas) == "...169")
1832 if(length(columnas_borrar) > 0) {
1833   d16_bombas <- d16_bombas[, 1:(columnas_borrar - 1)]
1834 }
1835
1836 #names(d16_bombas)
1837 head(d16_bombas[,c(1,2,3,4)],3)
1838 d16_bombas$Hora <- format(d16_bombas$Hora, "%H:%M:%S")
1839 Fecha <- as.POSIXct(paste(d16_bombas$Fecha, d16_bombas$Hora), format="
  %Y-%m-%d %H:%M:%S", tz = "UTC")
1840 d16_bombas<-cbind(Fecha, d16_bombas[, -c(1,2)])
1841 head(d16_bombas[,c(1,2,3,4)],3)
1842 ' ' '
1843
1844
1845 '{r, echo = FALSE, results = "hide"}
1846 anyNA(d16_bombas)
1847 ' ' '
1848
1849
1850 ### CT Nuevo
1851
1852
1853 '{r}
1854 d16_nuevo <- read_excel(ruta_especifica_d16, sheet = "CT Nuevo")
1855 #names(d16_nuevo)
1856
1857 head(d16_nuevo[,c(1,2,3,4)],3)
1858 d16_nuevo$Hora <- format(d16_nuevo$Hora, "%H:%M:%S")
1859 Fecha <- as.POSIXct(paste(d16_nuevo$Fecha, d16_nuevo$Hora), format="%Y
  -%m-%d %H:%M:%S", tz = "UTC")
1860 d16_nuevo<-cbind(Fecha, d16_nuevo[, -c(1,2)])
1861 head(d16_nuevo[,c(1,2,3,4)],3)
1862 ' ' '
1863
1864 Unas cuantas variables que en el excel solo aparece su nombre, ningun
  dato.
1865 '{r, echo = FALSE, results = "hide"}
1866 anyNA(d16_nuevo)

```

```

1867 na_cols <- colSums(is.na(d16_nuevo))
1868 na_cols <- na_cols[na_cols > 0]
1869 na_cols
1870 #which_rows_na <- which(rowSums(is.na(d16_nuevo[names(na_cols)])) > 0)
1871
1872 #d16_nuevo[which_rows_na, names(na_cols)]
1873 ""
1874
1875
1876
1877
1878 ### Variables de correcci n de rangos de fechas
1879
1880 ""{r}
1881 cat("Para irr \n")
1882 as.POSIXct(d16_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1883 as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1884 , tz = "UTC")
1885 cat("\nPara bombas \n")
1886 as.POSIXct(d16_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1887 as.POSIXct(d16_bombas$Fecha[length(d16_bombas$Fecha)], origin = "
1888 1970-01-01", tz = "UTC")
1889 cat("\nPara nuevo \n")
1890 as.POSIXct(d16_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1891 as.POSIXct(d16_nuevo$Fecha[length(d16_nuevo$Fecha)], origin = "
1892 1970-01-01", tz = "UTC")
1893 ""
1894 ""{r}
1895 cat("Para irr de bombas\n")
1896 as.POSIXct(d16_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1897 as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1898 , tz = "UTC")
1899 cat("\nPara bombas \n")
1900 as.POSIXct(d16_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1901 as.POSIXct(d16_bombas$Fecha[length(d16_bombas$Fecha)], origin = "
1902 1970-01-01", tz = "UTC")
1903 cat("\nPara irr de nuevo\n")
1904 as.POSIXct(d16_irr$Fecha[7], origin = "1970-01-01", tz = "UTC")
1905 as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1906 , tz = "UTC")
1907 cat("\nPara nuevo \n")
1908 as.POSIXct(d16_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1909 as.POSIXct(d16_nuevo$Fecha[length(d16_nuevo$Fecha)], origin = "
1910 1970-01-01", tz = "UTC")
1911 ""
1912 ""{r}
1913 # Convertir fechas a POSIXct en UTC
1914 f_irr <- as.POSIXct(d16_irr$Fecha, origin = "1970-01-01", tz = "UTC"

```

```

1911   )
1912   f_bomb  <- as.POSIXct(d16_bombas$Fecha, origin = "1970-01-01", tz = "
         UTC")
1913   f_nuev  <- as.POSIXct(d16_nuevo$Fecha, origin = "1970-01-01", tz = "
         UTC")
1914   # Recortes
1915   d16_irr_bombas <- d16_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1),
         ]
1916   d16_bombas    <- d16_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
         f_bomb, 1), ]
1917   d16_irr_nuevo <- d16_irr[f_irr >= f_irr[7] & f_irr <= tail(f_irr, 1),
         ]
1918   d16_nuevo     <- d16_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
         f_nuev, 1), ]
1919   # Verificaci n
1920   cat(sprintf("d16_bombas: %d | d16_nuevo: %d | d16_irr_bombas: %d |
         d16_irr_nuevo: %d\n",
1921             nrow(d16_bombas), nrow(d16_nuevo), nrow(d16_irr_bombas),
         nrow(d16_irr_nuevo)))
1922   ' ' '
1923   ' ' '
1924   ' ' '
1925   ' ' '
1926   ' ' '
1927   ' ' '
1928   ' ' '
1929   ' ' '
1930   ' ' '
1931   ' ' '
1932   ' ' '
1933   ' ' '
1934   ' ' '
1935   ' ' '
1936   ' ' '
1937   # Guardados a csv
1938   ' ' '{r}
1939   # Guardar bombas
1940   bombas_datasets <- paste0("d", setdiff(1:16, c(4, 5, 11)), "_bombas")
1941   for (nombre in bombas_datasets) {
1942     df <- get(nombre)
1943     write.csv(df, file = file.path("data_csv", paste0(nombre, ".csv")),
         row.names = FALSE)
1944   }
1945   # Guardar nuevos
1946   nuevo_datasets <- paste0("d", setdiff(1:16, c(2, 3, 15)), "_nuevo")
1947   for (nombre in nuevo_datasets) {
1948     df <- get(nombre)
1949   }
1950
1951

```

```

1952 write.csv(df, file = file.path("data_csv", paste0(nombre, ".csv")),
1953         row.names = FALSE)
1954 }
1955 '''
1956
1957 '{{{r}
1958 irr_bombas_datasets <- sub("_bombas", "_irr_bombas", bombas_datasets)
1959 irr_nuevo_datasets <- sub("_nuevo", "_irr_nuevo", nuevo_datasets)
1960 for (nombre in irr_bombas_datasets) {
1961   df <- get(nombre)
1962   write.csv(df, file = file.path("data_csv", paste0(nombre, ".csv")),
1963           row.names = FALSE)
1964 }
1965 for (nombre in irr_nuevo_datasets) {
1966   df <- get(nombre)
1967   write.csv(df, file = file.path("data_csv", paste0(nombre, ".csv")),
1968           row.names = FALSE)
1969 }
1970 }
1971 '''

```

Listing C.11: Procesado

C.0.12. Análisis completo

Un primer análisis completo con todos los datos.

```

1 ---
2 title: "Análisis completo"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"
5 output: pdf_document
6 ---
7
8 '{{{r}
9 archivos_csv <- list.files("data_csv", pattern = "\\*.csv$", full.names
10 = TRUE)
11
12 for (archivo in archivos_csv) {
13   nombre_objeto <- tools::file_path_sans_ext(basename(archivo))
14
15   df <- read.csv(archivo, stringsAsFactors = FALSE)
16
17   if ("Fecha" %in% names(df)) {
18     df$Fecha <- as.POSIXct(df$Fecha, tz = "UTC")
19   }
20
21   assign(nombre_objeto, df, envir = .GlobalEnv)
22 }
23 rm(df)

```

```

23  '''
24
25
26
27
28  # Que rangos tienen datos de CT?
29
30  Hay que tener en cuenta que los rangos son orientativos, pues al crear
    nuevos archivos para sincronizar las filas los rango entre CT's
    difieren.
31  '{r, echo=FALSE}'
32  # Nombres de los archivos
33  archivos <- c(
34    "Datos d1 09-01 a 09-03.xlsx",
35    "Datos d2 09-03 a 09-05.xlsx",
36    "Datos d3 09-05 a 09-07.xlsx",
37    "Datos d4 10-01 a 10-03.xlsx",
38    "Datos d5 10-03 a 10-05.xlsx",
39    "Datos d6 10-05 a 10-07.xlsx",
40    "Datos d7 11-03 a 11-05.xlsx",
41    "Datos d8 11-05 a 11-07.xlsx",
42    "Datos d9 11-07 a 11-09.xlsx",
43    "Datos d10 11-09 a 11-11.xlsx",
44    "Datos d11 11-11 a 11-13.xlsx",
45    "Datos d12 11-13 a 11-15.xlsx",
46    "Datos d13 02-04 a 03-04.xlsx",
47    "Datos d14 03-04 a 04-04.xlsx",
48    "Datos d15 04-04 a 12-04.xlsx",
49    "Datos d16 12-05 a 27-05.xlsx"
50  )
51
52  # Crear vectores lógicos con TRUE por defecto
53  ct_bombas <- rep(TRUE, length(archivos))
54  ct_nuevo <- rep(TRUE, length(archivos))
55
56  # Aplicar excepciones
57  ct_bombas[c(4, 5, 11)] <- FALSE # d4, d5, d11
58  ct_nuevo[c(2, 3, 15)] <- FALSE # d2, d3, d15
59
60  # Crear el data frame final
61  tabla_datos <- data.frame(
62    Archivo = archivos,
63    CT_bombas = ct_bombas,
64    CT_nuevo = ct_nuevo,
65    stringsAsFactors = FALSE
66  )
67
68  # Ver resultado
69  tabla_datos
70
71  '''

```

```

72 |
73 |
74 |
75 |
76 |
77 | # Como son las variables
78 |
79 |
80 | ## CT Bombas
81 | ```{r}
82 | # Lista de nombres de datasets v lidos
83 | bombas_datasets <- paste0("d", setdiff(1:16, c(4, 5, 11))), "_bombas")
84 |
85 | # Funci n para analizar cada dataset
86 | analiza_varianza <- function(nombre) {
87 |   df <- get(nombre)[, -1] # Excluye la columna de fecha
88 |   total <- ncol(df)
89 |   con_na <- sapply(df, function(x) any(is.na(x)))
90 |   sin_na <- df[, !con_na, drop = FALSE]
91 |
92 |   varianzas <- sapply(sin_na, var)
93 |   mayores_0 <- sum(varianzas > 0)
94 |   iguales_0 <- sum(varianzas == 0)
95 |   con_na_count <- sum(con_na)
96 |
97 |   data.frame(
98 |     Dataset = nombre,
99 |     Total = total,
100 |     'Var > 0' = mayores_0,
101 |     'Var = 0' = iguales_0,
102 |     'Con NA' = con_na_count,
103 |     check.names = FALSE
104 |   )
105 | }
106 |
107 | # Aplica la funci n y combina resultados
108 | tabla_bombas <- do.call(rbind, lapply(bombas_datasets,
109 |   analiza_varianza))
110 |
111 | ```
112 |
113 |
114 | ## CT Nuevo
115 | ```{r}
116 | # Lista de nombres de datasets v lidos
117 | nuevo_datasets <- paste0("d", setdiff(1:16, c(2, 3, 15))), "_nuevo")
118 |
119 | # Reutiliza la misma funci n de antes
120 | tabla_nuevo <- do.call(rbind, lapply(nuevo_datasets, analiza_varianza)
121 |   )

```

```
121 tabla_nuevo
122 ' ' '
123
124
125
126
127
128
129
130 # Rangos de fechas
131
132
133 ## CT Bombas
134 Inicio y final de los datasets en orden de bombas.
135
136
137 '{r , echo=FALSE}
138 bombas_validos <- setdiff(1:16, c(4, 5, 11))
139
140 for (nombre in bombas_validos) {
141   dataset <- get(paste0("d", nombre, "_bombas"))
142   cat("\n", strrep("-", 75), "\n\n")
143   cat("Fecha y hora inicio de los datos: ", format(dataset$Fecha[1], "
144     %Y-%m-%d %H:%M:%S"), "\n")
145   cat("Fecha y hora final de los datos: ", format(dataset$Fecha[length
146     (dataset$Fecha)], "%Y-%m-%d %H:%M:%S"), "\n")
147 }
148 cat("\n", strrep("-", 75), "\n\n")
149 ' ' '
150
151
152 ## CT Nuevo
153 Inicio y final de los datasets en orden de nuevo.
154
155
156 '{r , echo=FALSE}
157 nuevo_validos <- setdiff(1:16, c(2, 3, 15))
158
159 for (nombre in nuevo_validos) {
160   dataset <- get(paste0("d", nombre, "_nuevo"))
161   cat("\n", strrep("-", 75), "\n\n")
162   cat("Fecha y hora inicio de los datos: ", format(dataset$Fecha[1], "
163     %Y-%m-%d %H:%M:%S"), "\n")
164   cat("Fecha y hora final de los datos: ", format(dataset$Fecha[length
165     (dataset$Fecha)], "%Y-%m-%d %H:%M:%S"), "\n")
166 }
167 cat("\n", strrep("-", 75), "\n\n")
168 ' ' '
169
170
171
```

```

168
169
170
171
172
173
174
175
176
177
178 # ATENCI N, ESTA PRIMERA PARTE DEL ESTUDIO DE LOS NOMBRES DE
    VARIABLES COMUNES SE HACE SIN TENER EN CUENTA SI TIENEN O NO
    VARIABILIDAD
179
180
181 # Estudio de variables comunes en bombas
182
183 ## Soluci n problemas nombres repetidos en d3_bombas
184 Se seleccionan los 167 primeros, pues coincidir n con los nombres de
    la mayor a de los datasets
185
186 ‘‘{r}
187 # Selecciona por posici n y luego limpia los nombres duplicados
188 d3_bombas <- d3_bombas[, 1:167]
189
190 # Si quieres conservar los nombres originales sin los ...n
191 clean_names <- sub("\\.\\.\\.\\.\\. [0-9]+$", "", colnames(d3_bombas))
192 colnames(d3_bombas) <- clean_names
193 ‘‘
194
195 ## Nombres comunes exceptuando 'd2_bombas'
196
197 ‘‘{r}
198 # Obtener solo los nombres de columnas (sin la primera, que es la
    fecha)
199 prueba <- bombas_datasets[-2]
200 columnas_sin_fecha <- lapply(prueba, function(nombre) {
201   df <- get(nombre)
202   colnames(df)[-1] # Excluir primera columna
203 })
204
205 # Intersecci n de todas las listas de nombres
206 comunes_bombas_sin_d2 <- Reduce(intersect, columnas_sin_fecha)
207
208 # Resultado
209 cat("N mero de variables comunes sin d2: ", length(
    comunes_bombas_sin_d2), "\n\n")
210 cat("Nombres de variables comunes sin d2:\n")
211 print(comunes_bombas_sin_d2)
212 ‘‘
213

```

```

214
215
216
217
218
219
220
221
222
223
224
225 Estas son las 6 variables que faltan en d2_bombas que est n en todos
    los dem s bombas.
226 ```{r, echo=FALSE}
227 # Variables sin la fecha para d2
228 cols_d2 <- colnames(d2_bombas)[-1]
229
230 # Comparaci n
231 faltan_en_d2 <- setdiff(comunes_bombas_sin_d2, cols_d2)
232
233 # Resultados
234 cat("Variables que faltan en d2_bombas:\n")
235 print(faltan_en_d2)
236 cat("\nTotal faltantes en d2_bombas:", length(faltan_en_d2), "\n\n")
237 ```
238
239 Debido a esos 6, se procede a utilizar las 160 variables quitando esos
    6.
240 ```{r}
241 prueba <- bombas_datasets
242 columnas_sin_fecha <- lapply(prueba, function(nombre) {
243   df <- get(nombre)
244   colnames(df)[-1] # Excluir primera columna
245 })
246
247 # Intersecci n de todas las listas de nombres
248 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)
249
250 # Resultado
251 cat("N mero de variables comunes de todos los bombas: ", length(
    comunes_bombas), "\n\n")
252 ```
253
254
255 A adimos la fecha a los nombres de las variables.
256 ```{r}
257 nombres_bombas <- c("Fecha", comunes_bombas)
258 ```
259
260
261

```

```

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277 # Estudio de variables comunes en nuevo
278
279 ## Nombres comunes exceptuando d5_nuevo
280
281 ```{r}
282 # Obtener solo los nombres de columnas (sin la primera, que es la
    fecha)
283 prueba <- nuevo_datasets[-3]
284 columnas_sin_fecha <- lapply(prueba, function(nombre) {
285   df <- get(nombre)
286   colnames(df)[-1] # Excluir primera columna
287 })
288
289 # Intersección de todas las listas de nombres
290 comunes_nuevo_sin_d5 <- Reduce(intersect, columnas_sin_fecha)
291
292 # Resultado
293 cat("Número de variables comunes sin d5: ", length(
    comunes_nuevo_sin_d5), "\n\n")
294 ```
295
296
297
298 Nombres de d5_nuevo:
299 ```{r}
300 names(d5_nuevo)
301 ```
302
303
304
305
306 Viendo los nombres de d5_nuevo y cuales son las variables que faltan,
    se puede ver que el problema es que los nombres de mínimo y
    máximo todavía contienen las tildes.
307 ```{r}
308 cols_d5 <- colnames(d5_nuevo)[-1]

```

```

309 |
310 | # Comparaci n
311 | faltan_en_d5 <- setdiff(comunes_nuevo_sin_d5, cols_d5)
312 |
313 | # Resultados
314 | cat("Variables que faltan en d5_nuevo:\n")
315 | print(faltan_en_d5)
316 | cat("\nTotal faltantes en d5_nuevo:", length(faltan_en_d5), "\n\n")
317 | '''
318 |
319 | Se cambian las tildes a los nombres:
320 | '''{r}
321 | nombres <- names(d5_nuevo)
322 | nombres <- gsub("M n.", "Min", nombres)
323 | nombres <- gsub("M x.", "Max", nombres)
324 | names(d5_nuevo) <- nombres
325 | '''
326 |
327 |
328 |
329 | '''{r}
330 | # Obtener solo los nombres de columnas (sin la primera, que es la
      fecha)
331 | prueba <- nuevo_datasets
332 | columnas_sin_fecha <- lapply(prueba, function(nombre) {
333 |   df <- get(nombre)
334 |   colnames(df)[-1] # Excluir primera columna
335 | })
336 |
337 | # Intersecci n de todas las listas de nombres
338 | comunes_nuevo <- Reduce(intersect, columnas_sin_fecha)
339 |
340 | # Resultado
341 | cat("N mero de variables comunes: ", length(comunes_nuevo), "\n\n")
342 | '''
343 | Variables que falta en d5.
344 | '''{r,echo=FALSE}
345 | cols_d5 <- colnames(d5_nuevo)[-1]
346 |
347 | # Comparaci n
348 | faltan_en_d5 <- setdiff(comunes_nuevo_sin_d5, cols_d5)
349 |
350 | # Resultados
351 | cat("Variables que faltan en d5_nuevo:\n")
352 | print(faltan_en_d5)
353 | cat("\nTotal faltantes en d5_nuevo:", length(faltan_en_d5), "\n\n")
354 | '''
355 |
356 |
357 | A adimos la fecha a los nombres de las variables.
358 | '''{r}

```

```

359 nombres_nuevo <- c("Fecha", comunes_nuevo)
360 ""
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376 # Analisis de nombres comunes entre ambos datasets
377
378
379 En caso de no considerar los dos datasets problematicos el n mero de
    variables es el mismo.
380 ""{r}
381 diferencia <- setdiff(comunes_bombas_sin_d2, comunes_nuevo_sin_d5)
382
383 # Resultados
384 cat("Numero variables bombas sin d2: ", length(comunes_bombas_sin_d2),
    "\n\n")
385 cat("Numero variables nuevo sin d5: ", length(comunes_nuevo_sin_d5), "\n\n")
386 cat("Variables que no estan en ambos CT:\n")
387 print(diferencia)
388 cat("\nTotal faltantes:", length(diferencia), "\n\n")
389 ""
390
391
392
393 Si consideramos los dos datasets problematicos dentro de los nombres
    en ambos CT:
394 ""{r}
395 diferencia <- setdiff(nombres_bombas, nombres_nuevo)
396
397 # Resultados
398 cat("Numero variables bombas: ", length(nombres_bombas), "\n\n")
399 cat("Numero variables nuevo: ", length(nombres_nuevo), "\n\n")
400 cat("Variables que no estan en ambos CT:\n")
401 print(diferencia)
402 cat("\nTotal faltantes:", length(diferencia), "\n\n")
403 ""
404
405

```

```

406
407
408
409 # Que variables son continuas y cuales discretas en bombas?
410
411
412 Ejemplo de numero de valores que toman cada una de las variables (no
    aparecen las que sean un nico valor).
413 ```{r}
414 for (nombre in nombres_bombas) {
415   if (nombre %in% names(d1_bombas)) {
416     valores_unicos <- length(table(d1_bombas[[nombre]]))
417     if (valores_unicos > 1) {
418       cat(sprintf("%s: %d\n", nombre, valores_unicos))
419     }
420   } else {
421     cat(sprintf("%s: [NO ENCONTRADA EN d1_bombas]\n", nombre))
422   }
423 }
424 ```
425
426
427
428 Dado que hay una gran cantidad de datasets se realiza la comprobaci n
    de si todos los datasets tienen las mismas variable con menos de
    50 valores. Aunque no se imprima, al parecer las variables son
    totalmente distintas.
429 ```{r}
430 variables_menos_50 <- list()
431
432 for (nombre in bombas_datasets) {
433   df <- get(nombre)
434   vars_menos_50 <- c()
435   for (col in names(df)) {
436     if (length(table(df[[col]])) < 50) {
437       vars_menos_50 <- c(vars_menos_50, col)
438     }
439   }
440   variables_menos_50[[nombre]] <- sort(vars_menos_50)
441 }
442
443 nombres <- names(variables_menos_50)
444 referencia <- variables_menos_50[[1]]
445 diferencias <- list()
446
447 for (nombre in nombres) {
448   if (!identical(variables_menos_50[[nombre]], referencia)) {
449     diferencias[[nombre]] <- setdiff(union(variables_menos_50[[nombre]]
450                                           intersect(variables_menos_50[[
    nombre]], referencia))

```

```

451 }
452 }
453
454 # Resultado
455 if (length(diferencias) == 0) {
456   cat("Todas las variables con menos de 50 valores nicos son iguales
457       en todos los datasets.\n")
458 } else {
459   cat("Hay diferencias en las siguientes datasets:\n")
460   for (nombre in names(diferencias)) {
461     cat(sprintf("- %s\n", nombre))
462   }
463 }
464
465
466
467
468 Dado esto, se juntan los datasets a trav s de las variables comunes
469     (161, pues utilizamos d2 tambi n), y se hace un analisis completo.
470
471 ""{r}
472 # Lista para almacenar los subconjuntos
473 lista_bombas <- list()
474
475 # Recorrer y extraer solo columnas de inter s
476 for (nombre in bombas_datasets) {
477   if (exists(nombre)) {
478     df <- get(nombre)
479     comunes <- intersect(nombres_bombas, names(df)) # Por si falta
480     alguna
481     lista_bombas[[nombre]] <- df[, comunes, drop = FALSE]
482   } else {
483     warning(sprintf("Dataset %s no existe", nombre))
484   }
485 }
486
487 # Unir todos en uno solo
488 CT_bombas <- do.call(rbind, lista_bombas)
489
490
491
492
493 ""{r}
494 ### N mero de valores que toma cada variable en el dataset conjunto
495     de bombas.
496 conteos <- sapply(nombres_bombas, function(nombre) {
497   length(table(CT_bombas[[nombre]]))
498 })

```

```

498 conteos_filtrados <- conteos[conteos > 1]
499
500 conteos_ordenados <- sort(conteos_filtrados)
501
502 for (nombre in names(conteos_ordenados)) {
503   cat(sprintf("%s: %d\n", nombre, conteos_ordenados[[nombre]]))
504 }
505
506 '''
507
508
509
510
511
512
513 # Que variables son continuas y cuales discretas en nuevo?
514
515 Dado que hay una gran cantidad de datasets se realiza la comprobaci n
    de si todos los datasets tienen las mismas variable con menos de
    50 valores. Aunque no se imprima, al parecer las variables son
    totalmente distintas.
516 '''{r}
517 variables_menos_50 <- list()
518
519 for (nombre in nuevo_datasets) {
520   df <- get(nombre)
521   vars_menos_50 <- c()
522   for (col in names(df)) {
523     if (length(table(df[[col]])) < 50) {
524       vars_menos_50 <- c(vars_menos_50, col)
525     }
526   }
527   variables_menos_50[[nombre]] <- sort(vars_menos_50)
528 }
529
530 nombres <- names(variables_menos_50)
531 referencia <- variables_menos_50[[1]]
532 diferencias <- list()
533
534 for (nombre in nombres) {
535   if (!identical(variables_menos_50[[nombre]], referencia)) {
536     diferencias[[nombre]] <- setdiff(union(variables_menos_50[[nombre]
537                                           ], referencia),
538                                     intersect(variables_menos_50[[
539                                           nombre]], referencia))
540   }
541 }
542
543 # Resultado
544 if (length(diferencias) == 0) {
545   cat("Todas las variables con menos de 50 valores nicos son iguales

```

```

        en todos los datasets.\n")
544 } else {
545   cat("Hay diferencias en las siguientes datasets:\n")
546   for (nombre in names(diferencias)) {
547     cat(sprintf("- %s\n", nombre))
548   }
549 }
550
551 ""
552
553
554
555 Dado esto, se juntan los datasets a trav s de las variables comunes
    (140, pues utilizamos d5 tambi n), y se hace un analisis completo.
556
557 ""{r}
558 # Lista para almacenar los subconjuntos
559 lista_bombas <- list()
560
561 # Recorrer y extraer solo columnas de inter s
562 for (nombre in nuevo_datasets) {
563   if (exists(nombre)) {
564     df <- get(nombre)
565     comunes <- intersect(nombres_nuevo, names(df)) # Por si falta
        alguna
566     lista_bombas[[nombre]] <- df[, comunes, drop = FALSE]
567   } else {
568     warning(sprintf("Dataset %s no existe", nombre))
569   }
570 }
571
572 # Unir todos en uno solo
573 CT_nuevo <- do.call(rbind, lista_bombas)
574
575 ""
576
577
578
579 ### N mero de valores que toma cada variable en el dataset conjunto
    de nuevo.
580 ""{r}
581 conteos <- sapply(nombres_nuevo, function(nombre) {
582   length(table(CT_nuevo[[nombre]]))
583 })
584
585 conteos_filtrados <- conteos[conteos > 1]
586
587 conteos_ordenados <- sort(conteos_filtrados)
588
589 for (nombre in names(conteos_ordenados)) {
590   cat(sprintf("%s: %d\n", nombre, conteos_ordenados[[nombre]]))

```

```
591 }
592
593 ' ' '

```

Listing C.12: Análisis completo

C.0.13. Filtrado de datos

Filtrado de diferentes variables en cada archivo.

```

1  ---
2  title: "Filtrado de datos"
3  author: "Ismael Carbajo"
4  date: "r Sys.Date()"
5  output: pdf_document
6  ---
7
8
9  # Obtencion de conjuntos de datos
10 ' ' '{r}
11 archivos_csv <- list.files("data_csv", pattern = "\\..csv$", full.names
    = TRUE)
12
13 for (archivo in archivos_csv) {
14   nombre_objeto <- tools::file_path_sans_ext(basename(archivo))
15
16   df <- read.csv(archivo, stringsAsFactors = FALSE)
17
18   if ("Fecha" %in% names(df)) {
19     df$Fecha <- as.POSIXct(df$Fecha, tz = "UTC")
20   }
21
22   assign(nombre_objeto, df, envir = .GlobalEnv)
23 }
24 rm(df)
25 ' ' '
26
27
28 ' ' '{r}
29 # Selecciona por posici n y luego limpia los nombres duplicados
30 d3_bombas <- d3_bombas[, 1:167]
31
32 # Si quieres conservar los nombres originales sin los ...n
33 clean_names <- sub("\\..\\.\\.\\.\\. [0-9]+$", "", colnames(d3_bombas))
34 colnames(d3_bombas) <- clean_names
35 ' ' '
36
37
38
39 # CT bombas

```

```

40
41 ## Juntando bombas desde d6
42
43 ```{r}
44 bombas_datasets <- paste0("d", setdiff(6:16, c(11)), "_bombas")
45 bombas_datasets
46 ```
47
48
49 ```{r}
50 columnas_sin_fecha <- lapply(bombas_datasets, function(nombre) {
51   df <- get(nombre)
52   colnames(df)[-1]
53 })
54
55 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)
56
57 cat("N mero de variables comunes: ", length(comunes_bombas), "\n\n")
58 ```
59
60
61 ```{r}
62 nombres_bombas <- c("Fecha", comunes_bombas)
63 ```
64
65
66 ```{r}
67 # Lista para almacenar los subconjuntos
68 lista_bombas <- list()
69
70 # Recorrer y extraer solo columnas de inter s
71 for (nombre in bombas_datasets) {
72   if (exists(nombre)) {
73     df <- get(nombre)
74     comunes <- intersect(nombres_bombas, names(df)) # Por si falta
75     alguna
76     lista_bombas[[nombre]] <- df[, comunes, drop = FALSE]
77   } else {
78     warning(sprintf("Dataset %s no existe", nombre))
79   }
80 }
81
82 # Unir todos en uno solo
83 CT_bombas <- do.call(rbind, lista_bombas)
84
85
86
87 No contiene valores ausentes
88 ```{r}
89 anyNA(CT_bombas)

```

```

90  '''
91
92
93  Contiene nicamente 5 variables con varianza 0.
94  '''{r}
95  # Obtener nombres de columnas con varianza 0
96  colnames(CT_bombas)[sapply(CT_bombas, function(x) var(x) == 0)]
97  '''
98
99
100
101  ## Reconocimiento de variables categoricas
102
103
104  Nombres de las top 10 variables m s importantes seg n la tabla de
    an lisis anteriores.
105  '''{r}
106  Corr_irr_bombas <- c(
107    "Arm nicos.Tensi n5.L1N.Med",
108    "Cos.ph.L1N.Min",
109    "Factor.de.Potencia.L1N.Med",
110    "Cos.ph.L1N.Max",
111    "Potencia.Reactiva.L1N.Med",
112    "Potencia.Reactiva.L1N.Min",
113    "Desequilibrio.An.Min",
114    "Corriente.N.Med",
115    "Corriente.N.Max",
116    "Vrms.ph.n.NG.Med"
117  )
118
119  Corr_temp_bombas <- c(
120    "Factor.de.Potencia.L1N.Med",
121    "Cos.ph.L1N.Min",
122    "Arm nicos.Corriente.L1.Med",
123    "Corriente.N.Med",
124    "Corriente.N.Max",
125    "Vrms.ph.n.NG.Med",
126    "Vrms.ph.n.NG.Max",
127    "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
128    "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
129    "THD.A.L1.Med"
130  )
131
132  LM_bombas <- c(
133    "Arm nicos.Tensi n5.L1N.Med",
134    "Corriente.N.Med",
135    "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
136    "Vrms.ph.n.NG.Med",
137    "Vrms.ph.n.NG.Max",
138    "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
139    "Cos.ph.L1N.Min",

```

```

140 "Cos.ph.L1N.Max",
141 "THD.A.L1.Med",
142 "Potencia.Reactiva.L1N.Med"
143 )
144
145 GAM_sin_bombas <- c(
146 "Armnicos.Tensi n5.L1N.Med",
147 "Corriente.N.Med",
148 "Corriente.N.Max",
149 "Desequilibrio.An.Med",
150 "THD.A.L1.Med",
151 "Potencia.Reactiva.L1N.Med",
152 "Potencia.Reactiva.L1N.Min",
153 "Vrms.ph.n.NG.Med",
154 "Vrms.ph.n.NG.Max",
155 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med"
156 )
157
158 GAM_con_bombas <- c(
159 "Desequilibrio.An.Med",
160 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
161 "THD.A.L1.Med",
162 "Potencia.Reactiva.L1N.Med",
163 "Corriente.N.Max",
164 "Vrms.ph.n.NG.Max",
165 "Corriente.N.Med",
166 "Armnicos.Tensi n5.L1N.Med",
167 "THD.V.L1N.Max",
168 "THD.V.NG.Med"
169 )
170 ' ' '
171
172
173 Variables con menos de 100 valores.
174 ' '{r}
175 conteos <- sapply(nombres_bombas, function(nombre) {
176   length(table(CT_bombas[[nombre]]))
177 })
178
179 conteos_filtrados <- conteos[conteos > 1 & conteos < 100]
180
181 conteos_ordenados <- sort(conteos_filtrados)
182
183 pocos_valores_bombas <- names(conteos_ordenados)
184
185 print(pocos_valores_bombas)
186 ' ' '
187
188
189 Se verifica que variables estan en el top.
190 ' '{r}

```

```

191 # Intersecciones con cada vector
192 en_Corr_irr    <- intersect(pocos_valores_bombas, Corr_irr_bombas)
193 en_Corr_temp   <- intersect(pocos_valores_bombas, Corr_temp_bombas)
194 en_LM          <- intersect(pocos_valores_bombas, LM_bombas)
195 en_GAM_sin     <- intersect(pocos_valores_bombas, GAM_sin_bombas)
196 en_GAM_con     <- intersect(pocos_valores_bombas, GAM_con_bombas)
197
198 # Mostrar resultados
199 cat("En Corr_irr:\n"); print(en_Corr_irr)
200 cat("\nEn Corr_temp:\n"); print(en_Corr_temp)
201 cat("\nEn LM:\n"); print(en_LM)
202 cat("\nEn GAM_sin:\n"); print(en_GAM_sin)
203 cat("\nEn GAM_con:\n"); print(en_GAM_con)
204 ''
205
206
207
208
209
210 Haciendo lo mismo pero con menos de 50 valores tenemos las mismas
    variables.
211 ''{r}
212 conteos_filtrados <- conteos[conteos > 1 & conteos < 50]
213
214 conteos_ordenados <- sort(conteos_filtrados)
215
216 pocos_valores_bombas_50 <- names(conteos_ordenados)
217
218 # Intersecciones con cada vector
219 en_Corr_irr    <- intersect(pocos_valores_bombas_50, Corr_irr_bombas)
220 en_Corr_temp   <- intersect(pocos_valores_bombas_50, Corr_temp_bombas)
221 en_LM          <- intersect(pocos_valores_bombas_50, LM_bombas)
222 en_GAM_sin     <- intersect(pocos_valores_bombas_50, GAM_sin_bombas)
223 en_GAM_con     <- intersect(pocos_valores_bombas_50, GAM_con_bombas)
224
225 # Mostrar resultados
226 cat("En Corr_irr:\n"); print(en_Corr_irr)
227 cat("\nEn Corr_temp:\n"); print(en_Corr_temp)
228 cat("\nEn LM:\n"); print(en_LM)
229 cat("\nEn GAM_sin:\n"); print(en_GAM_sin)
230 cat("\nEn GAM_con:\n"); print(en_GAM_con)
231 ''
232
233
234 Las variables se repiten, por tanto se ve cuantos valores toman:
235 ''{r}
236 vars_repetidas <- c(en_Corr_irr, en_Corr_temp, en_LM, en_GAM_sin,
    en_GAM_con)
237
238 vars_unicas <- unique(vars_repetidas)
239

```

```

240 conteos_valores <- sapply(vars_unicas, function(var) {
241   length(unique(CT_bombas[[var]]))
242 })
243
244 print(conteos_valores)
245
246 valores_unicos <- lapply(vars_unicas, function(var) {
247   sort(unique(CT_bombas[[var]]))
248 })
249 names(valores_unicos) <- vars_unicas
250
251
252 ""
253
254
255 ""{r}
256 write.csv(CT_bombas,"data_csv/CT_bombas.csv")
257 ""
258
259
260
261 En total son 46 variables a eliminar, pero 7 de ellas pertenecen a los
    tops. Cuidado, porque no todos los tops pueden llegar a ser
    tiles , en caso de solo tener en cuenta los tops de modelos
    lineales y modelos GAM con interacci n quedan 6 variables:
262 ""{r}
263 unique(c(en_LM, en_GAM_con))
264 ""
265
266
267
268
269 # CT nuevo
270
271 ## Juntando nuevo desde d6
272
273 ""{r}
274 nuevo_datasets <- paste0("d", setdiff(6:16, c(15)), "_nuevo")
275 nuevo_datasets
276 ""
277
278
279 ""{r}
280 columnas_sin_fecha <- lapply(nuevo_datasets, function(nombre) {
281   df <- get(nombre)
282   colnames(df)[-1]
283 })
284
285 comunes_nuevo <- Reduce(intersect, columnas_sin_fecha)
286
287 cat("N mero de variables comunes: ", length(comunes_nuevo), "\n\n")

```

```

288  '''
289
290
291  '''{r}
292  nombres_nuevo <- c("Fecha", comunes_nuevo)
293  '''
294
295
296  '''{r}
297  # Lista para almacenar los subconjuntos
298  lista_nuevo <- list()
299
300  # Recorrer y extraer solo columnas de inter s
301  for (nombre in nuevo_datasets) {
302    if (exists(nombre)) {
303      df <- get(nombre)
304      comunes <- intersect(nombres_nuevo, names(df)) # Por si falta
305      lista_nuevo[[nombre]] <- df[, comunes, drop = FALSE]
306    } else {
307      warning(sprintf("Dataset %s no existe", nombre))
308    }
309  }
310
311  # Unir todos en uno solo
312  CT_nuevo <- do.call(rbind, lista_nuevo)
313
314  '''
315
316
317  Contiene valores ausentes.
318  '''{r}
319  anyNA(CT_nuevo)
320  '''
321
322  Se trantan de un total de 25 variables con una gran cantidad.
323  Proviene de d6, d7 y d16 si se ve la tabla del anterior informe.
324  '''{r}
325  nas_por_variable <- sapply(CT_nuevo, function(x) sum(is.na(x)))
326
327  nas_con_datos <- nas_por_variable[nas_por_variable > 0]
328
329  print(nas_con_datos)
330
331  '''
332
333  Todas las variables tienen varianza.
334  '''{r}
335  colnames(CT_nuevo)[sapply(CT_nuevo, function(x) var(x, na.rm = T) ==
336    0)]
337  '''

```

```

336 ## Reconocimiento de variables categoricas
337
338
339 Nombres de las top 10 variables m s importantes seg n la tabla de
    analisis anteriores.
340 ``{r}
341 Corr_irr_nuevo <- c(
342   "Desequilibrio.Vn.Min",
343   "Desequilibrio.Vn.Med",
344   "Arm nicos.Tensi n5.NG.Med",
345   "Arm nicos.Tensi n5.L1N.Max",
346   "THD.V.L1N.Max",
347   "Arm nicos.Tensi n5.L1N.Med",
348   "Arm nicos.Tensi n5.NG.Min",
349   "Tensi n.de.Pico.NG.Med",
350   "Arm nicos.Tensi n5.L1N.Min",
351   "THD.V.L1N.Med"
352 )
353
354 Corr_temp_nuevo <- c(
355   "Desequilibrio.Vn.Min",
356   "Arm nicos.Tensi n5.NG.Med",
357   "Desequilibrio.Vn.Med",
358   "Arm nicos.Tensi n5.NG.Min",
359   "THD.A.N.Med",
360   "Arm nicos.Corriente1.L1.Med",
361   "Arm nicos.Corriente1.L1.Max",
362   "Tensi n.Factor.de.Cresta.NG.Med",
363   "Corriente.Fundamental.N.Med",
364   "THD.V.L1N.Med"
365 )
366
367 LM_nuevo <- c(
368   "Arm nicos.Tensi n5.L1N.Med",
369   "Arm nicos.Tensi n5.L1N.Max",
370   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
371   "Vrms.ph.n.NG.Max",
372   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
373   "Corriente.N.Min",
374   "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
375   "THD.V.L1N.Med",
376   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
377   "Corriente.N.Med"
378 )
379
380 GAM_sin_nuevo <- c(
381   "Arm nicos.Tensi n5.L1N.Med",
382   "Arm nicos.Tensi n5.L1N.Min",
383   "THD.V.L1N.Max",
384   "Vrms.ph.n.NG.Med",
385   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",

```

```

386   "Corriente.N.Med",
387   "THD.V.L1N.Med",
388   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
389   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
390   "Corriente.N.Min"
391 )
392
393 GAM_con_nuevo <- c(
394   "Arm nicos.Tensi n5.L1N.Med",
395   "Arm nicos.Tensi n5.L1N.Min",
396   "THD.V.L1N.Max",
397   "Vrms.ph.n.NG.Max",
398   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
399   "THD.V.L1N.Med",
400   "Corriente.N.Med",
401   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
402   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
403   "Corriente.N.Min"
404 )
405 ' ' '
406
407
408
409
410 Variables con menos de 100 valores.
411 '{r}
412 conteos <- sapply(nombres_nuevo, function(nombre) {
413   length(table(CT_nuevo[[nombre]]))
414 })
415
416 conteos_filtrados <- conteos[conteos > 1 & conteos < 100]
417
418 conteos_ordenados <- sort(conteos_filtrados)
419
420 pocos_valores_nuevo <- names(conteos_ordenados)
421
422 print(pocos_valores_nuevo)
423 ' ' '
424
425
426
427 Se verifica que variables estan en el top.
428 En este caso, parece que no hay variables categoricas en los tops 10
   importantes.
429 '{r}
430 # Intersecciones con cada vector
431 en_Corr_irr <- intersect(pocos_valores_nuevo, Corr_irr_nuevo)
432 en_Corr_temp <- intersect(pocos_valores_nuevo, Corr_temp_nuevo)
433 en_LM <- intersect(pocos_valores_nuevo, LM_nuevo)
434 en_GAM_sin <- intersect(pocos_valores_nuevo, GAM_sin_nuevo)
435 en_GAM_con <- intersect(pocos_valores_nuevo, GAM_con_nuevo)

```

```

436
437 # Mostrar resultados
438 cat("En Corr_irr:\n"); print(en_Corr_irr)
439 cat("\nEn Corr_temp:\n"); print(en_Corr_temp)
440 cat("\nEn LM:\n"); print(en_LM)
441 cat("\nEn GAM_sin:\n"); print(en_GAM_sin)
442 cat("\nEn GAM_con:\n"); print(en_GAM_con)
443 ' '
444
445
446 '{r}
447 write.csv(CT_nuevo, "data_csv/CT_nuevo.csv")
448 ' '
449
450
451
452
453
454
455
456
457 # Revisi n en conjunto
458
459 '{r}
460 unique(c(pocos_valores_bombas, pocos_valores_nuevo))
461 ' '

```

Listing C.13: Filtrado de datos

C.0.14. Estudio de incidencias

Generación de informe del estudio de incidencias para el Departamento de Ingeniería Eléctrica.

```

1 ---
2 title: "Estudio de incidencias"
3 author: "Ismael Carbajo"
4 date: "'r Sys.Date()"
5 output:
6   pdf_document:
7     latex_engine: pdflatex
8     keep_tex: true
9
10 ---
11
12
13 '{r, echo=FALSE}
14 CT_nuevo <- read.csv("data_csv/CT_nuevo.csv", row.names = 1,
15   stringsAsFactors = FALSE)

```

```

15 CT_nuevo$Fecha <- as.POSIXct(CT_nuevo$Fecha, tz = "UTC")
16
17 CT_bombas <- read.csv("data_csv/CT_bombas.csv", row.names = 1,
18   stringsAsFactors = FALSE)
19 CT_bombas$Fecha <- as.POSIXct(CT_bombas$Fecha, tz = "UTC")
20
21
22 ```{r, echo=FALSE}
23 Corr_irr_bombas <- c(
24   "Arm nicos.Tensi n5.L1N.Med",
25   "Cos.ph.L1N.Min",
26   "Factor.de.Potencia.L1N.Med",
27   "Cos.ph.L1N.Max",
28   "Potencia.Reactiva.L1N.Med",
29   "Potencia.Reactiva.L1N.Min",
30   "Desequilibrio.An.Min",
31   "Corriente.N.Med",
32   "Corriente.N.Max",
33   "Vrms.ph.n.NG.Med"
34 )
35
36 Corr_temp_bombas <- c(
37   "Factor.de.Potencia.L1N.Med",
38   "Cos.ph.L1N.Min",
39   "Arm nicos.Corriente.L1.Med",
40   "Corriente.N.Med",
41   "Corriente.N.Max",
42   "Vrms.ph.n.NG.Med",
43   "Vrms.ph.n.NG.Max",
44   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
45   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
46   "THD.A.L1.Med"
47 )
48
49 LM_bombas <- c(
50   "Arm nicos.Tensi n5.L1N.Med",
51   "Corriente.N.Med",
52   "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
53   "Vrms.ph.n.NG.Med",
54   "Vrms.ph.n.NG.Max",
55   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
56   "Cos.ph.L1N.Min",
57   "Cos.ph.L1N.Max",
58   "THD.A.L1.Med",
59   "Potencia.Reactiva.L1N.Med"
60 )
61
62 GAM_sin_bombas <- c(
63   "Arm nicos.Tensi n5.L1N.Med",
64   "Corriente.N.Med",

```

```

65 "Corriente.N.Max",
66 "Desequilibrio.An.Med",
67 "THD.A.L1.Med",
68 "Potencia.Reactiva.L1N.Med",
69 "Potencia.Reactiva.L1N.Min",
70 "Vrms.ph.n.NG.Med",
71 "Vrms.ph.n.NG.Max",
72 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med"
73 )
74
75 GAM_con_bombas <- c(
76 "Desequilibrio.An.Med",
77 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
78 "THD.A.L1.Med",
79 "Potencia.Reactiva.L1N.Med",
80 "Corriente.N.Max",
81 "Vrms.ph.n.NG.Max",
82 "Corriente.N.Med",
83 "Arm nicos.Tensi n5.L1N.Med",
84 "THD.V.L1N.Max",
85 "THD.V.NG.Med"
86 )
87 '""
88
89 '{r, echo=FALSE}
90
91 Corr_irr_nuevo <- c(
92 "Desequilibrio.Vn.Min",
93 "Desequilibrio.Vn.Med",
94 "Arm nicos.Tensi n5.NG.Med",
95 "Arm nicos.Tensi n5.L1N.Max",
96 "THD.V.L1N.Max",
97 "Arm nicos.Tensi n5.L1N.Med",
98 "Arm nicos.Tensi n5.NG.Min",
99 "Tensi n.de.Pico.NG.Med",
100 "Arm nicos.Tensi n5.L1N.Min",
101 "THD.V.L1N.Med"
102 )
103
104 Corr_temp_nuevo <- c(
105 "Desequilibrio.Vn.Min",
106 "Arm nicos.Tensi n5.NG.Med",
107 "Desequilibrio.Vn.Med",
108 "Arm nicos.Tensi n5.NG.Min",
109 "THD.A.N.Med",
110 "Arm nicos.Corriente1.L1.Med",
111 "Arm nicos.Corriente1.L1.Max",
112 "Tensi n.Factor.de.Cresta.NG.Med",
113 "Corriente.Fundamental.N.Med",
114 "THD.V.L1N.Med"
115 )

```

```

116
117 LM_nuevo <- c(
118   "Arm nicos.Tensi n5.L1N.Med",
119   "Arm nicos.Tensi n5.L1N.Max",
120   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
121   "Vrms.ph.n.NG.Max",
122   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
123   "Corriente.N.Min",
124   "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
125   "THD.V.L1N.Med",
126   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
127   "Corriente.N.Med"
128 )
129
130 GAM_sin_nuevo <- c(
131   "Arm nicos.Tensi n5.L1N.Med",
132   "Arm nicos.Tensi n5.L1N.Min",
133   "THD.V.L1N.Max",
134   "Vrms.ph.n.NG.Med",
135   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
136   "Corriente.N.Med",
137   "THD.V.L1N.Med",
138   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
139   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
140   "Corriente.N.Min"
141 )
142
143 GAM_con_nuevo <- c(
144   "Arm nicos.Tensi n5.L1N.Med",
145   "Arm nicos.Tensi n5.L1N.Min",
146   "THD.V.L1N.Max",
147   "Vrms.ph.n.NG.Max",
148   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
149   "THD.V.L1N.Med",
150   "Corriente.N.Med",
151   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
152   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
153   "Corriente.N.Min"
154 )
155 ' ' '
156
157 ' ' '{r, echo=FALSE}
158
159 anotar_top <- function(varnames,
160                        LM_b = LM_bombas, GAM_b = GAM_con_bombas,
161                        LM_n = LM_nuevo, GAM_n = GAM_con_nuevo) {
162
163   etiquetas <- sapply(varnames, function(v) {
164     tags <- c()
165
166     if (v %in% LM_b) tags <- c(tags, paste0("B-LM-", which(LM_b == v))

```

```

167     )
168     if (v %in% GAM_b) tags <- c(tags, paste0("B-GAM-", which(GAM_b ==
169     v)))
170     if (v %in% LM_n) tags <- c(tags, paste0("N-LM-", which(LM_n == v))
171     )
172     if (v %in% GAM_n) tags <- c(tags, paste0("N-GAM-", which(GAM_n ==
173     v)))
174
175     if (length(tags) > 0) paste(v, "[", paste(tags, collapse = ", "),
176     "]" ) else v
177 })
178
179 return(etiquetas)
180 }
181 ' ' '
182
183 ### Explicaci n tabla
184 Si aparece 'X valores', se trata de una variable que toma menos de 100
185 valores en todo el conjunto, en concreto toma X valores.
186
187 Si aparece 'NA's: X', significa que contiene X valores ausentes.
188
189 Si aparece 'var 0', significa que esa variable no tiene varianza (
190 siempre toma el mismo valor).
191
192 Si una variable para un dataset espec fico contiene varias
193 incidencias, aparecen todas separadas por '|'.
194
195 Adem s , se ha incorporado una **anotaci n junto al nombre de cada
196 variable** que indica si esta forma parte del conjunto de variables
197 m s importantes en alguno de los modelos seleccionados (modelos
198 lineales o modelos GAM con interacci n). La notaci n utilizada es
199 la siguiente:
200
201 - 'B-LM-X': variable en la posici n **X** del top 10 del modelo
202 lineal para 'CT_bombas'.
203 - 'B-GAM-X': posici n **X** en el top del modelo GAM con interacci n
204 para 'CT_bombas'.
205 - 'N-LM-X': posici n **X** en el modelo lineal de 'CT_nuevo'.
206 - 'N-GAM-X': posici n **X** en el modelo GAM con interacci n para '
207 CT_nuevo'.
208
209 Despu s de la tabla principal se encuentra otra tabla, que expone los
210 rangos de fechas donde ocurren los valores ausentes.
211
212 ' ' '{r, warning=FALSE, echo=FALSE, message=FALSE}
213 library(knitr)
214 library(kableExtra)
215 library(dplyr)

```

```

202 # Funci n para detectar incidencias
203 detectar_incidencias <- function(df) {
204   incidencias <- list()
205
206   for (col in names(df)) {
207     x <- df[[col]]
208
209     texto <- NULL
210     if (is.numeric(x)) {
211       if (var(x, na.rm = TRUE) == 0) {
212         texto <- "var 0"
213       }
214       if (sum(is.na(x)) > 0) {
215         texto <- c(texto, paste0("NA's: ", sum(is.na(x))))
216       }
217       n_vals <- length(unique(na.omit(x)))
218       if (n_vals < 100) {
219         texto <- c(texto, paste0(n_vals, " valores"))
220       }
221     } else {
222       if (sum(is.na(x)) > 0) {
223         texto <- paste0("NA's: ", sum(is.na(x)))
224       }
225     }
226
227     if (!is.null(texto)) {
228       incidencias[[col]] <- paste(texto, collapse = " | ")
229     }
230   }
231
232   return(incidencias)
233 }
234
235 # Aplicar funci n a ambos dataframes
236 inc_bombas <- detectar_incidencias(CT_bombas)
237 inc_nuevo <- detectar_incidencias(CT_nuevo)
238
239 # Variables con incidencias
240 todas_vars <- union(names(inc_bombas), names(inc_nuevo))
241
242 # Construir data.frame
243 tabla_incidencias <- data.frame(
244   'CT_bombas' = sapply(todas_vars, function(v) ifelse(!is.null(
245     inc_bombas[[v]]), inc_bombas[[v]], "")),
245   'CT_nuevo' = sapply(todas_vars, function(v) ifelse(!is.null(
246     inc_nuevo[[v]]), inc_nuevo[[v]], ""))
246 )
247
248 rownames(tabla_incidencias) <- anotar_top(rownames(tabla_incidencias))
249
250 # Mostrar tabla

```

```

251 kable(tabla_incidencias, format = "latex", booktabs = TRUE, longtable
      = TRUE,
252       caption = "Tabla de incidencias por variable") %>%
253 kable_styling(latex_options = c("hold_position", "repeat_header"),
254              full_width = FALSE, position = "center") %>%
255 row_spec(seq(1, nrow(tabla_incidencias), 2), background = "#D9D9D9")
256 '""
257
258
259
260 Los valores ausentes provienen de hasta 3 archivos diferentes, d6, d7
    y d16:
261
262 - 'Datos6 de 12-03 a 27-03.xlsx'
263 - 'Datos7 de 24-03 a 05-04.xlsx'
264 - 'Datos16 de 12-03 a 27-05.xlsx'
265
266 Los rangos donde se encuentran los valores ausentes son:
267
268
269 '{r, echo=FALSE}
270 d6_nuevo <- read.csv("data_csv/d6_nuevo.csv", stringsAsFactors = FALSE
    )
271 d6_nuevo$Fecha <- as.POSIXct(d6_nuevo$Fecha, tz = "UTC")
272
273 d7_nuevo <- read.csv("data_csv/d7_nuevo.csv", stringsAsFactors = FALSE
    )
274 d7_nuevo$Fecha <- as.POSIXct(d7_nuevo$Fecha, tz = "UTC")
275
276 d16_nuevo <- read.csv("data_csv/d16_nuevo.csv", stringsAsFactors =
    FALSE)
277 d16_nuevo$Fecha <- as.POSIXct(d16_nuevo$Fecha, tz = "UTC")
278 '""
279
280
281
282 '{r, echo=FALSE}
283 # Funci n que detecta para cada variable los rangos de fechas con NA
    (seguidos)
284 rangos_na_archivo <- function(df) {
285   res <- list()
286
287   for (col in setdiff(names(df), "Fecha")) {
288     if (any(is.na(df[[col]]))) {
289       fechas_na <- df$Fecha[is.na(df[[col]])]
290       res[[col]] <- paste0(format(min(fechas_na), "%Y-%m-%d"), " - ",
        format(max(fechas_na), "%Y-%m-%d"))
291     }
292   }
293
294   return(res)

```

```
295 }
296
297 # Aplicar a cada archivo
298 r6 <- rangos_na_archivo(d6_nuevo)
299 r7 <- rangos_na_archivo(d7_nuevo)
300 r16 <- rangos_na_archivo(d16_nuevo)
301
302 # Combinar en una tabla con origen
303 crear_tabla <- function(lista, origen) {
304   if (length(lista) == 0) return(NULL)
305   data.frame(
306     Variable = names(lista),
307     Rango = unname(unlist(lista)),
308     Archivo = origen,
309     stringsAsFactors = FALSE
310   )
311 }
312
313 tabla_rangos <- bind_rows(
314   crear_tabla(r6, "d6"),
315   crear_tabla(r7, "d7"),
316   crear_tabla(r16, "d16")
317 )
318
319
320 # Mostrar tabla
321 kable(tabla_rangos, format = "latex", booktabs = TRUE, longtable =
322   TRUE,
323   caption = "Rangos de fechas con NA por variable y archivo") %>%
324   kable_styling(latex_options = c("hold_position", "repeat_header"),
325     full_width = FALSE, position = "center") %>%
326   row_spec(seq(1, nrow(tabla_incidenicias), 2), background = "#D9D9D9")
327 , , ,
```

Listing C.14: Estudio de incidencias

C.0.15. Estudio de incidencias modificado

Generación de informe del estudio de incidencias para el Departamento de Ingeniería Eléctrica tras haber comentado el primer estudio con ellos y haber recibido los nuevos datos con los problemas corregidos.

```
1 ---
2 title: "Estudio de incidencias"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"
5 output:
6   pdf_document:
7     latex_engine: pdflatex
```

```

8      keep_tex: true
9
10 ---
11
12
13   '{r, echo=FALSE}
14 CT_nuevo <- read.csv("data_csv_modif/CT_nuevo.csv", stringsAsFactors =
15   FALSE)
16 CT_nuevo$Fecha <- as.POSIXct(CT_nuevo$Fecha, tz = "UTC")
17
18 CT_bombas <- read.csv("data_csv_modif/CT_bombas.csv", stringsAsFactors
19   = FALSE)
20 CT_bombas$Fecha <- as.POSIXct(CT_bombas$Fecha, tz = "UTC")
21 ''
22
23   '{r, echo=FALSE}
24 Corr_irr_bombas <- c(
25   "Arm nicos.Tensi n5.L1N.Med",
26   "Cos.ph.L1N.Min",
27   "Factor.de.Potencia.L1N.Med",
28   "Cos.ph.L1N.Max",
29   "Potencia.Reactiva.L1N.Med",
30   "Potencia.Reactiva.L1N.Min",
31   "Desequilibrio.An.Min",
32   "Corriente.N.Med",
33   "Corriente.N.Max",
34   "Tensi n.NG.Med"
35 )
36
37 Corr_temp_bombas <- c(
38   "Factor.de.Potencia.L1N.Med",
39   "Cos.ph.L1N.Min",
40   "Arm nicos.Corriente.L1.Med",
41   "Corriente.N.Med",
42   "Corriente.N.Max",
43   "Tensi n.NG.Med",
44   "Tensi n.NG.Max",
45   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
46   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
47   "THD.A.L1.Med"
48 )
49
50 LM_bombas <- c(
51   "Arm nicos.Tensi n5.L1N.Med",
52   "Corriente.N.Med",
53   "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
54   "Tensi n.NG.Med",
55   "Tensi n.NG.Max",
56   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
57   "Cos.ph.L1N.Min",

```

```

57   "Cos.ph.L1N.Max",
58   "THD.A.L1.Med",
59   "Potencia.Reactiva.L1N.Med"
60 )
61
62 GAM_sin_bombas <- c(
63   "Arm nicos.Tensi n5.L1N.Med",
64   "Corriente.N.Med",
65   "Corriente.N.Max",
66   "Desequilibrio.An.Med",
67   "THD.A.L1.Med",
68   "Potencia.Reactiva.L1N.Med",
69   "Potencia.Reactiva.L1N.Min",
70   "Tensi n.NG.Med",
71   "Tensi n.NG.Max",
72   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med"
73 )
74
75 GAM_con_bombas <- c(
76   "Desequilibrio.An.Med",
77   "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
78   "THD.A.L1.Med",
79   "Potencia.Reactiva.L1N.Med",
80   "Corriente.N.Max",
81   "Tensi n.NG.Max",
82   "Corriente.N.Med",
83   "Arm nicos.Tensi n5.L1N.Med",
84   "THD.V.L1N.Max",
85   "THD.V.NG.Med"
86 )
87 '""
88
89
90 '{r, echo=FALSE}
91 Corr_irr_nuevo <- c(
92   "Desequilibrio.Vn.Min",
93   "Desequilibrio.Vn.Med",
94   "Arm nicos.Tensi n5.NG.Med",
95   "Arm nicos.Tensi n5.L1N.Max",
96   "THD.V.L1N.Max",
97   "Arm nicos.Tensi n5.L1N.Med",
98   "Arm nicos.Tensi n5.NG.Min",
99   "Tensi n.de.Pico.NG.Med",
100  "Arm nicos.Tensi n5.L1N.Min",
101  "THD.V.L1N.Med"
102 )
103
104 Corr_temp_nuevo <- c(
105   "Desequilibrio.Vn.Min",
106   "Arm nicos.Tensi n5.NG.Med",
107   "Desequilibrio.Vn.Med",

```

```

108 "Arm nicos.Tensi n5.NG.Min",
109 "THD.A.N.Med",
110 "Arm nicos.Corriente1.L1.Med",
111 "Arm nicos.Corriente1.L1.Max",
112 "Tensi n.Factor.de.Cresta.NG.Med",
113 "Corriente.Fundamental.N.Med",
114 "THD.V.L1N.Med"
115 )
116
117 LM_nuevo <- c(
118 "Arm nicos.Tensi n5.L1N.Med",
119 "Arm nicos.Tensi n5.L1N.Max",
120 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
121 "Tensi n.NG.Max",
122 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
123 "Corriente.N.Min",
124 "Tensi n.de.Medio.Ciclo.V.RMS.NG.M n",
125 "THD.V.L1N.Med",
126 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
127 "Corriente.N.Med"
128 )
129
130 GAM_sin_nuevo <- c(
131 "Arm nicos.Tensi n5.L1N.Med",
132 "Arm nicos.Tensi n5.L1N.Min",
133 "THD.V.L1N.Max",
134 "Tensi n.NG.Med",
135 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
136 "Corriente.N.Med",
137 "THD.V.L1N.Med",
138 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
139 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
140 "Corriente.N.Min"
141 )
142
143 GAM_con_nuevo <- c(
144 "Arm nicos.Tensi n5.L1N.Med",
145 "Arm nicos.Tensi n5.L1N.Min",
146 "THD.V.L1N.Max",
147 "Tensi n.NG.Max",
148 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Max",
149 "THD.V.L1N.Med",
150 "Corriente.N.Med",
151 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Med",
152 "Tensi n.de.Medio.Ciclo.V.RMS.NG.Min",
153 "Corriente.N.Min"
154 )
155 " "
156
157 " "{r, echo=FALSE}
158

```

```

159 anotar_top <- function(varnames,
160                        LM_b = LM_bombas, GAM_b = GAM_con_bombas,
161                        LM_n = LM_nuevo, GAM_n = GAM_con_nuevo) {
162
163   etiquetas <- sapply(varnames, function(v) {
164     tags <- c()
165
166     if (v %in% LM_b) tags <- c(tags, paste0("B-LM-", which(LM_b == v))
167     )
168     if (v %in% GAM_b) tags <- c(tags, paste0("B-GAM-", which(GAM_b ==
169     v)))
170     if (v %in% LM_n) tags <- c(tags, paste0("N-LM-", which(LM_n == v))
171     )
172     if (v %in% GAM_n) tags <- c(tags, paste0("N-GAM-", which(GAM_n ==
173     v)))
174
175     if (length(tags) > 0) paste(v, "[", paste(tags, collapse = ", "),
176     "]" ) else v
177   })
178
179   return(etiquetas)
180 }
181 '''
182
183 ### Explicaci n tabla
184 Si aparece 'X valores', se trata de una variable que toma menos de 100
185 valores en todo el conjunto, en concreto toma X valores.
186
187 Si aparece 'NA's: X', significa que contiene X valores ausentes.
188
189 Si aparece 'var 0', significa que esa variable no tiene varianza (
190 siempre toma el mismo valor).
191
192 Si aparece 'NO LA TIENE' esa variable no se encuentra en el dataset.
193
194 Si una variable para un dataset espec fico contiene varias
195 incidencias, aparecen todas separadas por '|'.
196
197 Adem s , se ha incorporado una **anotaci n junto al nombre de cada
198 variable** que indica si esta forma parte del conjunto de variables
199 m s importantes en alguno de los modelos seleccionados (modelos
200 lineales o modelos GAM con interacci n). La notaci n utilizada es
201 la siguiente:
202
203 - 'B-LM-X': variable en la posici n **X** del top 10 del modelo
204 lineal para 'CT_bombas'.
205 - 'B-GAM-X': posici n **X** en el top del modelo GAM con interacci n
206 para 'CT_bombas'.
207 - 'N-LM-X': posici n **X** en el modelo lineal de 'CT_nuevo'.
208 - 'N-GAM-X': posici n **X** en el modelo GAM con interacci n para '
209 CT_nuevo'.

```

```

195
196 Despu s de la tabla principal se encuentra otra tabla, que expone los
      rangos de fechas donde ocurren los valores ausentes.
197
198
199 ‘‘{r, warning=FALSE, echo=FALSE, message=FALSE}
200 library(knitr)
201 library(kableExtra)
202 library(dplyr)
203
204 # Funci n para detectar incidencias
205 detectar_incidencias <- function(df) {
206   incidencias <- list()
207
208   for (col in names(df)) {
209     x <- df[[col]]
210
211     texto <- NULL
212     if (is.numeric(x)) {
213       if (var(x, na.rm = TRUE) == 0) {
214         texto <- "var 0"
215       }
216       if (sum(is.na(x)) > 0) {
217         texto <- c(texto, paste0("NA's: ", sum(is.na(x))))
218       }
219       n_vals <- length(unique(na.omit(x)))
220       if (n_vals < 100) {
221         texto <- c(texto, paste0(n_vals, " valores"))
222       }
223     } else {
224       if (sum(is.na(x)) > 0) {
225         texto <- paste0("NA's: ", sum(is.na(x)))
226       }
227     }
228
229     if (!is.null(texto)) {
230       incidencias[[col]] <- paste(texto, collapse = " | ")
231     }
232   }
233
234   return(incidencias)
235 }
236
237 # Aplicar funci n a ambos dataframes
238 inc_bombas <- detectar_incidencias(CT_bombas)
239 inc_nuevo <- detectar_incidencias(CT_nuevo)
240
241 # Variables con incidencias
242 todas_vars <- union(names(inc_bombas), names(inc_nuevo))
243
244 # Construir data.frame

```

```

245 tabla_incidencias <- data.frame(
246   'CT_bombas' = sapply(todas_vars, function(v) {
247     if (!v %in% names(CT_bombas)) "NO LA TIENE"
248     else if (!is.null(inc_bombas[[v]])) inc_bombas[[v]]
249     else ""
250   }),
251   'CT_nuevo' = sapply(todas_vars, function(v) {
252     if (!v %in% names(CT_nuevo)) "NO LA TIENE"
253     else if (!is.null(inc_nuevo[[v]])) inc_nuevo[[v]]
254     else ""
255   })
256 )
257 rownames(tabla_incidencias) <- anotar_top(rownames(tabla_incidencias))
258
259 # Mostrar tabla
260 kable(tabla_incidencias, format = "latex", booktabs = TRUE, longtable
261       = TRUE,
262       caption = "Tabla de incidencias por variable") %>%
263       kable_styling(latex_options = c("hold_position", "repeat_header"),
264                     full_width = FALSE, position = "center") %>%
265       row_spec(seq(1, nrow(tabla_incidencias), 2), background = "#D9D9D9")
266
267
268
269
270 # Si se quitan todas las variables pertenecientes a neutro en CT
271   bombas, queda lo siguiente:
272
273 ''{r, echo=FALSE}
274 CT_bombas_sinN <- CT_bombas[ , !grepl("N\\.|NG", names(CT_bombas))]
275 ''
276
277 ''{r, warning=FALSE, echo=FALSE, message=FALSE}
278 library(knitr)
279 library(kableExtra)
280 library(dplyr)
281
282 # Funci n para detectar incidencias
283 detectar_incidencias <- function(df) {
284   incidencias <- list()
285
286   for (col in names(df)) {
287     x <- df[[col]]
288
289     texto <- NULL
290     if (is.numeric(x)) {
291       if (var(x, na.rm = TRUE) == 0) {
292         texto <- "var 0"
293       }

```

```

294     if (sum(is.na(x)) > 0) {
295         texto <- c(texto, paste0("NA's: ", sum(is.na(x))))
296     }
297     n_vals <- length(unique(na.omit(x)))
298     if (n_vals < 100) {
299         texto <- c(texto, paste0(n_vals, " valores"))
300     }
301 } else {
302     if (sum(is.na(x)) > 0) {
303         texto <- paste0("NA's: ", sum(is.na(x)))
304     }
305 }
306
307 if (!is.null(texto)) {
308     incidencias[[col]] <- paste(texto, collapse = " | ")
309 }
310 }
311
312 return(incidencias)
313 }
314
315 # Aplicar funci n a CT_bombas
316 incidencias_bombas <- detectar_incidencias(CT_bombas_sinN)
317
318 # Crear tabla
319 tabla_incidencias <- data.frame(
320     Incidencia = unlist(incidencias_bombas)
321 )
322
323 # Mostrar tabla
324 kable(tabla_incidencias, format = "latex", booktabs = TRUE, longtable
      = TRUE,
325       caption = "Incidencias detectadas en CT\\_bombas") %>%
326 kable_styling(latex_options = c("hold_position", "repeat_header"),
327               full_width = FALSE, position = "center") %>%
328 row_spec(seq(1, nrow(tabla_incidencias), 2), background = "#D9D9D9")
329 ' ' '

```

Listing C.15: Estudio de incidencias modificado

C.0.16. Procesado modificado

Tras recibir todos los archivos modificados, proceso que se encarga de depurar todos los archivos y realizar los *merge* necesarios.

```

1 ---
2 title: "Procesado datos"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"

```

```
5 output: pdf_document
6 ---
7
8 ‘‘{r , warning=FALSE, message=FALSE, echo=FALSE}
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(data.table)
15 library(car)
16 library(zoo)
17
18 # No hace falta poner '/' al final, ya se tiene en cuenta luego
19 ruta_datos <- "H:/TFG EST/General_modif"
20 ‘‘‘
21
22 Todos los NA's que se sustituyen son trav s de una interpolaci n con
23   splines (librer a 'zoo').
24
25 Grandes cantidades de NA's no se pueden interpolar, por tanto esas
26   partes de tiempo son borradas.
27
28
29 # Procesados de los datos modificados
30
31 ## 1. Datos de 19-08 a 03-09
32
33 Este rango de datos contiene las dos CT.
34
35
36 ### Irradiancia
37
38 ‘‘{r}
39 ruta_especifica_d1<-paste(ruta_datos, "/Datos1 de 19-08 a 03-09.xlsx",
40   sep="")
41 d1_irr <- read_excel(ruta_especifica_d1, sheet = "Irradiancia")
42 head(d1_irr)
43 ‘‘‘
44
45 Contiene una fila de NA's
46 ‘‘{r, echo = FALSE, results = "hide"}
47 anyNA(d1_irr)
48 na_cols <- colSums(is.na(d1_irr))
49 na_cols <- na_cols[na_cols > 0]
50
51 which_rows_na <- which(rowSums(is.na(d1_irr[names(na_cols)])) > 0)
52
53 d1_irr[which_rows_na, ]
54 ‘‘‘
```

```

53  "'{r}
54  d1_irr <- as.data.frame(lapply(d1_irr, na.spline), check.names = FALSE)
55  anyNA(d1_irr)
56  "'
57
58
59
60  ### CT Bombas
61
62
63  "'{r , message = FALSE}
64  d1_bombas <- read_excel(ruta_especifica_d1, sheet = "CT Bombas")
65  #names(d1_bombas)
66
67  nombres <- names(d1_bombas)
68  nombres <- gsub("M n .", "Min", nombres)
69  nombres <- gsub("M x .", "Max", nombres)
70  names(d1_bombas) <- nombres
71
72  #names(d1_bombas)
73  head(d1_bombas[,c(1,2,3,4)],3)
74  d1_bombas$Hora <- format(d1_bombas$Hora, "%H:%M:%S")
75  Fecha <- as.POSIXct(paste(d1_bombas$Fecha, d1_bombas$Hora), format="%Y-
    -%m-%d %H:%M:%S", tz="UTC")
76  d1_bombas<-cbind(Fecha, d1_bombas[, -c(1,2)])
77  head(d1_bombas[,c(1,2,3,4)],3)
78  "'
79
80  "'{r, echo = FALSE, results = "hide"}
81  anyNA(d1_bombas)
82  "'
83
84
85  ### CT Nuevo
86
87
88  "'{r}
89  d1_nuevo <- read_excel(ruta_especifica_d1, sheet = "CT Nuevo")
90  #names(d1_nuevo)
91
92  d1_nuevo <- d1_nuevo[, !grepl("L2|L3", names(d1_nuevo))]
93  #names(d1_nuevo)
94
95  head(d1_nuevo[,c(1,2,3,4)],3)
96  d1_nuevo$Hora <- format(d1_nuevo$Hora, "%H:%M:%S")
97  Fecha <- as.POSIXct(paste(d1_nuevo$Fecha, d1_nuevo$Hora), format="%Y-
    -m-%d %H:%M:%S", tz="UTC")
98  d1_nuevo<-cbind(Fecha, d1_nuevo[, -c(1,2)])
99  head(d1_nuevo[,c(1,2,3,4)],3)
100  "'
101

```

```

102
103   ‘‘‘{r, echo = FALSE, results = "hide"}
104 anyNA(d1_nuevo)
105   ‘‘‘
106
107
108 ### Variables de corrección de rangos de fechas
109
110   ‘‘‘{r}
111   cat("Para irr \n")
112   as.POSIXct(d1_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
113   as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
114             tz = "UTC")
114   cat("\nPara bombas \n")
115   as.POSIXct(d1_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
116   as.POSIXct(d1_bombas$Fecha[length(d1_bombas$Fecha)], origin = "
117             1970-01-01", tz = "UTC")
117   cat("\nPara nuevo \n")
118   as.POSIXct(d1_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
119   as.POSIXct(d1_nuevo$Fecha[length(d1_nuevo$Fecha)], origin = "
120             1970-01-01", tz = "UTC")
121   ‘‘‘
122
123   ‘‘‘{r}
124   cat("Para irr de bombas\n")
125   as.POSIXct(d1_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
126   as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
127             tz = "UTC")
127   cat("\nPara bombas \n")
128   as.POSIXct(d1_bombas$Fecha[3], origin = "1970-01-01", tz = "UTC")
129   as.POSIXct(d1_bombas$Fecha[length(d1_bombas$Fecha)-1], origin = "
130             1970-01-01", tz = "UTC")
131   cat("\nPara irr de nuevo\n")
132   as.POSIXct(d1_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
133   as.POSIXct(d1_irr$Fecha[length(d1_irr$Fecha)], origin = "1970-01-01",
134             tz = "UTC")
134   cat("\nPara nuevo \n")
135   as.POSIXct(d1_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
136   as.POSIXct(d1_nuevo$Fecha[length(d1_nuevo$Fecha)-1], origin = "
137             1970-01-01", tz = "UTC")
138   ‘‘‘
139
140   ‘‘‘{r}
141   # Convertir fechas a POSIXct en UTC
142   f_irr    <- as.POSIXct(d1_irr$Fecha, origin = "1970-01-01", tz = "UTC")
143   f_bomb   <- as.POSIXct(d1_bombas$Fecha, origin = "1970-01-01", tz = "
144             UTC")
145   f_nuev   <- as.POSIXct(d1_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
146             ")

```

```

144 # Recortes aplicados seg n lo indicado
145 d1_irr_bombas <- d1_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1), ]
146 d1_bombas      <- d1_bombas[f_bomb >= f_bomb[3] & f_bomb <= f_bomb[
147     length(f_bomb) - 1], ]
148 d1_irr_nuevo   <- d1_irr[f_irr >= f_irr[2] & f_irr <= tail(f_irr, 1), ]
149 d1_nuevo       <- d1_nuevo[f_nuev >= f_nuev[1] & f_nuev <= f_nuev[
150     length(f_nuev) - 1], ]
151 # Verificaci n
152 cat(sprintf("d1_bombas: %d | d1_nuevo: %d | d1_irr_bombas: %d |
153     d1_irr_nuevo: %d\n",
154     nrow(d1_bombas), nrow(d1_nuevo), nrow(d1_irr_bombas), nrow
155     (d1_irr_nuevo)))
156
157
158
159
160
161
162
163
164
165
166
167
168 ## 2. Datos de 03-09 a 13-09
169
170 Este rango de datos solo contiene CT bombas.
171
172
173 ### Irradiancia
174
175 ```{r}
176 ruta_especifica_d2<-paste(ruta_datos, "/Datos2 de 03-09 a 13-09.xlsx",
177     sep="")
178 d2_irr <- read_excel(ruta_especifica_d2, sheet = "Irradiancia")
179 head(d2_irr)
180
181
182 Contiene dos filas seguidas de NA's
183 ```{r, echo = FALSE, results = "hide"}
184 anyNA(d2_irr)
185 na_cols <- colSums(is.na(d2_irr))
186 na_cols <- na_cols[na_cols > 0]
187
188 which_rows_na <- which(rowSums(is.na(d2_irr[names(na_cols)])) > 0)
189
190 d2_irr[which_rows_na, ]

```

```

190  '''
191
192  '''{r}
193  d2_irr <- as.data.frame(lapply(d2_irr, na.spline),check.names = FALSE)
194  anyNA(d2_irr)
195  '''
196
197
198  ### CT Bombas
199
200
201  '''{r}
202  d2_bombas <- read_excel(ruta_especifica_d2, sheet = "CT Bombas")
203  #names(d2_bombas)
204
205  head(d2_bombas[,c(1,2,3,4)],3)
206  d2_bombas$Hora <- format(d2_bombas$Hora, "%H:%M:%S")
207  Fecha <- as.POSIXct(paste(d2_bombas$Fecha, d2_bombas$Hora), format="%Y
      -%m-%d %H:%M:%S", tz = "UTC")
208  d2_bombas<-cbind(Fecha, d2_bombas[, -c(1,2)])
209  head(d2_bombas[,c(1,2,3,4)],3)
210  '''
211
212  '''{r, echo = FALSE, results = "hide"}
213  anyNA(d2_bombas)
214  '''
215
216
217  ### Variables de corrección de rangos de fechas
218
219  '''{r}
220  cat("Para irr \n")
221  as.POSIXct(d2_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
222  as.POSIXct(d2_irr$Fecha[length(d2_irr$Fecha)], origin = "1970-01-01",
      tz = "UTC")
223  cat("\nPara bombas \n")
224  as.POSIXct(d2_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
225  as.POSIXct(d2_bombas$Fecha[length(d2_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
226  '''
227  '''{r}
228  d2_irr_bombas<-d2_irr
229  '''
230
231
232
233
234
235
236  ## 3. Datos de 24-09 a 02-10
237

```

```

238 Este rango de datos solo contiene CT bombas.
239
240
241
242 ### Irradiancia
243
244 ```{r}
245 ruta_especifica_d3<-paste(ruta_datos, "/Datos3 de 24-09 a 02-10.xlsx",
246   sep="")
247 d3_irr <- read_excel(ruta_especifica_d3, sheet = "Irradiancia")
248 head(d3_irr)
249 ```
250
251 ```{r, echo = FALSE, results = "hide"}
252 anyNA(d3_irr)
253 ```
254
255 ### CT Bombas
256
257
258 ```{r}
259 d3_bombas <- read_excel(ruta_especifica_d3, sheet = "CT Bombas")
260 #names(d3_bombas)
261
262 # Eliminar columnas desde "Column1" en adelante
263 columnas_borrar <- which(names(d3_bombas) == "Column1")
264 if(length(columnas_borrar) > 0) {
265   d3_bombas <- d3_bombas[, 1:(columnas_borrar - 1)]
266 }
267
268 # Eliminar columnas que contienen "L2" o "L3"
269 d3_bombas <- d3_bombas[, !grepl("L2|L3", names(d3_bombas))]
270
271 #names(d3_bombas)
272 head(d3_bombas[,c(1,2,3,4)],3)
273 d3_bombas$Hora <- format(d3_bombas$Hora, "%H:%M:%S")
274 Fecha <- as.POSIXct(paste(d3_bombas$Fecha, d3_bombas$Hora), format="%Y
275   -%m-%d %H:%M:%S", tz = "UTC")
276 d3_bombas<-cbind(Fecha, d3_bombas[, -c(1,2)])
277 head(d3_bombas[,c(1,2,3,4)],3)
278 ```
279
280 Muchas variables son practicamente enteras Na's, mirar a ver. Vuelve a
281   pasar lo mismo, aqui ahora hay otro problema, hay variables que
282   aparecen con ...XXX siendo XXX numeros, esto al parecer ocurre
283   porque hay variables con nombres repetidos.
284
285 ```{r, echo = FALSE, results = "hide"}
286 anyNA(d3_bombas)
287 na_cols <- colSums(is.na(d3_bombas))

```

```

284 na_cols <- na_cols[na_cols > 0]
285 na_cols
286 #which_rows_na <- which(rowSums(is.na(d3_bombas[names(na_cols)])) > 0)
287
288 #d3_bombas[which_rows_na, names(na_cols)]
289 ' ' '
290
291
292
293
294
295
296
297 ### Variables de corrección de rangos de fechas
298
299
300 Debido a que los intervalos de las fechas cambian, hay que dividirlo
301     en dos y luego volver a juntar supongo?
302 ' ' {r}
303 cat("Para irr \n")
304 as.POSIXct(d3_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
305 as.POSIXct(d3_irr$Fecha[length(d3_irr$Fecha)], origin = "1970-01-01",
306     tz = "UTC")
307 cat("\nPara bombas \n")
308 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
309 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
310     1970-01-01", tz = "UTC")
311 ' ' '
312
313 ' ' {r}
314 cat("Para irr \n")
315 as.POSIXct(d3_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
316 d3_irr$Fecha[900]
317 d3_irr$Fecha[901]
318 as.POSIXct(d3_irr$Fecha[length(d3_irr$Fecha)], origin = "1970-01-01",
319     tz = "UTC")
320
321 cat("\nPara bombas \n")
322 as.POSIXct(d3_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
323 as.POSIXct(d3_bombas$Fecha[899], origin = "1970-01-01", tz = "UTC")
324 as.POSIXct(d3_bombas$Fecha[901], origin = "1970-01-01", tz = "UTC")
325 as.POSIXct(d3_bombas$Fecha[length(d3_bombas$Fecha)], origin = "
326     1970-01-01", tz = "UTC")
327 ' ' '
328
329 ' ' {r}
330 # Convertir a POSIXct
331 f_irr <- as.POSIXct(d3_irr$Fecha, origin = "1970-01-01", tz = "UTC")
332 f_bomb <- as.POSIXct(d3_bombas$Fecha, origin = "1970-01-01", tz = "
333     UTC")

```

```

329 # Aplicar l gica de recorte sincronizada
330 d3_irr_bombas <- d3_irr[2:nrow(d3_irr), ] # Saltar
    primera fila
331 d3_bombas <- d3_bombas[-900, ] # Eliminar fila
    900
332
333 # Validar dimensiones y alineaci n
334 cat(sprintf("d3_bombas: %d | d3_irr_bombas: %d\n", nrow(d3_bombas),
    nrow(d3_irr_bombas)))
335
336
337 ' ' '
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353 ## 4. Datos de 02-10 a 22-10
354
355 Este rango de datos solo contiene CT nuevo.
356
357
358
359 ### Irradiancia
360
361 '{r}
362 ruta_especifica_d4<-paste(ruta_datos, "/Datos4 de 02-10 a 22-10.xlsx",
    sep="")
363 d4_irr <- read_excel(ruta_especifica_d4, sheet = "Irradiancia")
364 head(d4_irr)
365 ' ' '
366
367 Contiene dos fila de NA's
368 '{r, echo = FALSE, results = "hide"}
369 anyNA(d4_irr)
370 na_cols <- colSums(is.na(d4_irr))
371 na_cols <- na_cols[na_cols > 0]
372
373 which_rows_na <- which(rowSums(is.na(d4_irr[names(na_cols)])) > 0)
374
375 d4_irr[which_rows_na, ]

```

```

376  '
377
378  '{r}
379  d4_irr <- as.data.frame(lapply(d4_irr, na.spline),check.names = FALSE)
380  anyNA(d4_irr)
381  '
382
383
384
385  ### CT Nuevo
386
387
388  '{r}
389  d4_nuevo <- read_excel(ruta_especifica_d4, sheet = "CT Nuevo")
390  #names(d4_nuevo)
391
392  nombres <- names(d4_nuevo)
393  nombres <- gsub(" M n.", "Min", nombres)
394  nombres <- gsub(" M x.", "Max", nombres)
395  names(d4_nuevo) <- nombres
396
397  head(d4_nuevo[,c(1,2,3,4)],3)
398  d4_nuevo$Hora <- format(d4_nuevo$Hora, "%H:%M:%S")
399  Fecha <- as.POSIXct(paste(d4_nuevo$Fecha, d4_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz = "UTC")
400  d4_nuevo<-cbind(Fecha, d4_nuevo[, -c(1,2)])
401  head(d4_nuevo[,c(1,2,3,4)],3)
402  '
403
404  '{r, echo = FALSE, results = "hide"}
405  anyNA(d4_nuevo)
406  '
407
408  ### Variables de corrección de rangos de fechas
409
410
411  '{r}
412  cat("Para irr \n")
413  as.POSIXct(d4_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
414  as.POSIXct(d4_irr$Fecha[length(d4_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
415  cat("\nPara nuevo \n")
416  as.POSIXct(d4_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
417  as.POSIXct(d4_nuevo$Fecha[length(d4_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
418  '
419
420  '{r}
421  cat("Para irr \n")
422  as.POSIXct(d4_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
423  as.POSIXct(d4_irr$Fecha[length(d4_irr$Fecha)-1], origin = "1970-01-01"

```

```

    , tz = "UTC")
424 cat("\nPara nuevo \n")
425 as.POSIXct(d4_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
426 as.POSIXct(d4_nuevo$Fecha[length(d4_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
427 ' ' '
428
429
430
431 '{r}
432 # Fechas como POSIXct en UTC
433 f_irr <- as.POSIXct(d4_irr$Fecha, origin = "1970-01-01", tz = "UTC")
434 f_nuev <- as.POSIXct(d4_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
    ")
435
436 # Recortes basados en inspecci n
437 d4_irr_nuevo <- d4_irr[1:(nrow(d4_irr) - 1), ]
438 d4_nuevo <- d4_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1),
    ]
439
440 # Verificaci n de filas
441 cat(sprintf("d4_irr_nuevo: %d | d4_nuevo: %d\n", nrow(d4_irr_nuevo),
    nrow(d4_nuevo)))
442
443 ' ' '
444
445
446
447
448
449
450
451
452
453
454 ## 5. Datos de 24-10 a 13-11
455
456 Este rango de datos solo contiene CT nuevo.
457
458
459
460 ### Irradiancia
461
462 '{r}
463 ruta_especifica_d5<-paste(ruta_datos, "/Datos5 de 24-10 a 13-11.xlsx",
    sep="")
464 d5_irr <- read_excel(ruta_especifica_d5, sheet = "Irradiancia")
465
466 fila_inicial <- 375 # Cambia este valor seg n d nde quieras aplicar
    el ajuste
467

```

```

468 # Sumar 60 minutos a partir de la fila indicada
469 d5_irr$Fecha[fila_inicial:nrow(d5_irr)] <- d5_irr$Fecha[fila_inicial:
      nrow(d5_irr)] + 60*60 # 60
470 head(d5_irr)
471 '''
472
473 Contiene una fila de NA's
474 ```{r, echo = FALSE, results = "hide"}
475 anyNA(d5_irr)
476 na_cols <- colSums(is.na(d5_irr))
477 na_cols <- na_cols[na_cols > 0]
478
479 which_rows_na <- which(rowSums(is.na(d5_irr[names(na_cols)])) > 0)
480
481 d5_irr[which_rows_na, ]
482 '''
483
484 ```{r}
485 d5_irr <- as.data.frame(lapply(d5_irr, na.spline), check.names = FALSE)
486 anyNA(d5_irr)
487 '''
488
489
490 ### CT Nuevo
491
492
493 ```{r}
494 d5_nuevo <- read_excel(ruta_especifica_d5, sheet = "CT Nuevo")
495 #names(d5_nuevo)
496
497 nombres <- names(d5_nuevo)
498 nombres <- gsub("M n.", "Min", nombres)
499 nombres <- gsub("M x.", "Max", nombres)
500 names(d5_nuevo) <- nombres
501
502 head(d5_nuevo[,c(1,2,3,4)],3)
503 d5_nuevo$Hora <- format(d5_nuevo$Hora, "%H:%M:%S")
504 Fecha <- as.POSIXct(paste(d5_nuevo$Fecha, d5_nuevo$Hora), format="%Y-%
      m-%d %H:%M:%S", tz = "UTC")
505 d5_nuevo <- cbind(Fecha, d5_nuevo[, -c(1,2)])
506 head(d5_nuevo[,c(1,2,3,4)],3)
507 '''
508
509
510 ```{r, echo = FALSE, results = "hide"}
511 anyNA(d5_nuevo)
512 '''
513
514 ### Variables de corrección de rangos de fechas
515
516 ```{r}

```

```

517 cat("Para irr \n")
518 as.POSIXct(d5_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
519 as.POSIXct(d5_irr$Fecha[length(d5_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
520 cat("\nPara nuevo \n")
521 as.POSIXct(d5_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
522 as.POSIXct(d5_nuevo$Fecha[length(d5_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
523 ' ' '
524
525
526
527 ' ' '{r}
528 cat("Para irr \n")
529 as.POSIXct(d5_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
530 as.POSIXct(d5_irr$Fecha[length(d5_irr$Fecha)-6], origin = "1970-01-01"
    , tz = "UTC")
531 cat("\nPara nuevo \n")
532 as.POSIXct(d5_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
533 as.POSIXct(d5_nuevo$Fecha[length(d5_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
534 ' ' '
535
536
537 ' ' '{r}
538 # Convertir fechas a POSIXct en UTC
539 f_irr <- as.POSIXct(d5_irr$Fecha, origin = "1970-01-01", tz = "UTC")
540 f_nuev <- as.POSIXct(d5_nuevo$Fecha, origin = "1970-01-01", tz = "UTC"
    )
541
542 # Aplicar recortes
543 d5_irr_nuevo <- d5_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
    f_irr) - 6], ]
544 d5_nuevo <- d5_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1),
    ]
545
546 # Verificaci n
547 cat(sprintf("d5_irr_nuevo: %d | d5_nuevo: %d\n", nrow(d5_irr_nuevo),
    nrow(d5_nuevo)))
548
549 ' ' '
550
551
552
553
554 ## 6. Datos de 14-11 a 27-11
555
556 Este rango de datos contiene las dos CT.
557
558
559

```

```

560 ### Irradiancia
561
562 ```{r}
563 ruta_especifica_d6<-paste(ruta_datos, "/Datos6 de 14-11 a 27-11.xlsx",
564   sep="")
565 d6_irr <- read_excel(ruta_especifica_d6, sheet = "Irradiancia")
566 head(d6_irr)
567
568 ```{r, echo = FALSE, results = "hide"}
569 anyNA(d6_irr)
570 ```
571
572 ### CT Bombas
573
574
575 ```{r}
576 d6_bombas <- read_excel(ruta_especifica_d6, sheet = "CT Bombas")
577 #names(d6_bombas)
578
579 head(d6_bombas[,c(1,2,3,4)],3)
580 d6_bombas$Hora <- format(d6_bombas$Hora, "%H:%M:%S")
581 Fecha <- as.POSIXct(paste(d6_bombas$Fecha, d6_bombas$Hora), format="%Y-
582   -%m-%d %H:%M:%S", tz = "UTC")
583 d6_bombas<-cbind(Fecha, d6_bombas[, -c(1,2)])
584 head(d6_bombas[,c(1,2,3,4)],3)
585
586 ```{r, echo = FALSE, results = "hide"}
587 anyNA(d6_bombas)
588 ```
589
590
591 ### CT Nuevo
592
593
594 ```{r}
595 d6_nuevo <- read_excel(ruta_especifica_d6, sheet = "CT Nuevo")
596 #names(d6_nuevo)
597
598 nombres <- names(d6_nuevo)
599 nombres <- gsub(" M n .", "Min", nombres)
600 nombres <- gsub(" M x .", "Max", nombres)
601 names(d6_nuevo) <- nombres
602
603 head(d6_nuevo[,c(1,2,3,4)],3)
604 d6_nuevo$Hora <- format(d6_nuevo$Hora, "%H:%M:%S")
605 Fecha <- as.POSIXct(paste(d6_nuevo$Fecha, d6_nuevo$Hora), format="%Y-%
606   m-%d %H:%M:%S", tz = "UTC")
607 d6_nuevo<-cbind(Fecha, d6_nuevo[, -c(1,2)])
608 head(d6_nuevo[,c(1,2,3,4)],3)

```

```

608  ' ' '
609
610  '{r, echo = FALSE, results = "hide"}
611  anyNA(d6_nuevo)
612  ' ' '
613
614
615  ### Variables de correcci n de rangos de fechas
616
617  '{r}
618  cat("Para irr \n")
619  as.POSIXct(d6_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
620  as.POSIXct(d6_irr$Fecha[length(d6_irr$Fecha)], origin = "1970-01-01",
621    tz = "UTC")
622  cat("\nPara bombas \n")
623  as.POSIXct(d6_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
624  as.POSIXct(d6_bombas$Fecha[length(d6_bombas$Fecha)], origin = "
625    1970-01-01", tz = "UTC")
626  cat("\nPara nuevo \n")
627  as.POSIXct(d6_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
628  as.POSIXct(d6_nuevo$Fecha[length(d6_nuevo$Fecha)-4], origin = "
629    1970-01-01", tz = "UTC")
630  ' ' '
631
632  '{r}
633  d6_irr_bombas <- d6_irr
634  d6_irr_nuevo <- d6_irr
635  ' ' '
636
637  '{r}
638  f_nuev <- as.POSIXct(d6_nuevo$Fecha, origin = "1970-01-01", tz = "UTC"
639    )
640
641  # Aplicar recortes
642  d6_nuevo <- d6_nuevo[f_nuev <= f_nuev[length(f_nuev) - 4], ]
643
644  cat(sprintf("d6_bombas: %d | d6_nuevo: %d | d6_irr_bombas: %d |
645    d6_irr_nuevo: %d\n",
646      nrow(d6_bombas), nrow(d6_nuevo), nrow(d6_irr_bombas), nrow
647        (d6_irr_nuevo)))
648  ' ' '
649
650  ## 7. Datos de 27-11 a 03-12
651
652  Este rango de datos contiene las dos CT.

```

```

653
654
655
656 ### Irradiancia
657
658 ```{r}
659 ruta_especifica_d7<-paste(ruta_datos, "/Datos7 de 27-11 a 03-12.xlsx",
660   sep="")
661 d7_irr <- read_excel(ruta_especifica_d7, sheet = "Irradiancia")
662 head(d7_irr)
663
664
665 ```{r, echo = FALSE, results = "hide"}
666 anyNA(d7_irr)
667
668
669
670 ### CT Bombas
671
672
673 ```{r}
674 d7_bombas <- read_excel(ruta_especifica_d7, sheet = "CT Bombas")
675 #names(d7_bombas)
676
677 head(d7_bombas[,c(1,2,3,4)],3)
678 d7_bombas$Hora <- format(d7_bombas$Hora, "%H:%M:%S")
679 Fecha <- as.POSIXct(paste(d7_bombas$Fecha, d7_bombas$Hora), format="%Y
680   -%m-%d %H:%M:%S", tz="UTC")
681 d7_bombas<-cbind(Fecha, d7_bombas[,c(1,2)])
682 head(d7_bombas[,c(1,2,3,4)],3)
683
684
685 ```{r, echo = FALSE, results = "hide"}
686 anyNA(d7_bombas)
687
688
689
690 ### CT Nuevo
691
692
693 ```{r}
694 d7_nuevo <- read_excel(ruta_especifica_d7, sheet = "CT Nuevo")
695 #names(d7_nuevo)
696
697 nombres <- names(d7_nuevo)
698 nombres <- gsub("M n.", "Min", nombres)
699 nombres <- gsub("M x.", "Max", nombres)
700 names(d7_nuevo) <- nombres
701
702 head(d7_nuevo[,c(1,2,3,4)],3)

```

```

702 d7_nuevo$Hora <- format(d7_nuevo$Hora, "%H:%M:%S")
703 Fecha <- as.POSIXct(paste(d7_nuevo$Fecha, d7_nuevo$Hora), format="%Y-%
      m-%d %H:%M:%S", tz="UTC")
704 d7_nuevo<-cbind(Fecha, d7_nuevo[, -c(1,2)])
705 head(d7_nuevo[, c(1,2,3,4)], 3)
706 ```
707
708
709 ```{r, echo = FALSE, results = "hide"}
710 anyNA(d7_nuevo)
711 ```
712
713 ### Variables de correcci n de rangos de fechas
714
715 ```{r}
716 cat("Para irr \n")
717 as.POSIXct(d7_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
718 as.POSIXct(d7_irr$Fecha[length(d7_irr$Fecha)], origin = "1970-01-01",
      tz = "UTC")
719 cat("\nPara bombas \n")
720 as.POSIXct(d7_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
721 as.POSIXct(d7_bombas$Fecha[length(d7_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
722 cat("\nPara nuevo \n")
723 as.POSIXct(d7_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
724 as.POSIXct(d7_nuevo$Fecha[length(d7_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
725 ```
726
727 ```{r}
728 d7_irr_bombas <- d7_irr
729 d7_irr_nuevo <- d7_irr
730
731 # Fila resumen
732 cat(sprintf("d7_bombas: %d | d7_nuevo: %d | d7_irr_bombas: %d |
      d7_irr_nuevo: %d\n",
733             nrow(d7_bombas), nrow(d7_nuevo), nrow(d7_irr_bombas), nrow
      (d7_irr_nuevo)))
734 ```
735
736
737
738
739
740
741
742
743
744
745
746

```

```

747
748
749 ## 8. Datos de 03-12 a 20-12
750
751 Este rango de datos contiene las dos CT.
752
753
754
755 ### Irradiancia
756
757 ```{r}
758 ruta_especifica_d8<-paste(ruta_datos, "/Datos8 de 03-12 a 20-12.xlsx",
759   sep="")
760 d8_irr <- read_excel(ruta_especifica_d8, sheet = "Irradiancia")
761 head(d8_irr)
762
763 Contiene una fila de NA's
764 ```{r, echo = FALSE, results = "hide"}
765 anyNA(d8_irr)
766 na_cols <- colSums(is.na(d8_irr))
767 na_cols <- na_cols[na_cols > 0]
768
769 which_rows_na <- which(rowSums(is.na(d8_irr[names(na_cols)])) > 0)
770
771 d8_irr[which_rows_na, ]
772 ```
773
774 ```{r}
775 d8_irr <- as.data.frame(lapply(d8_irr, na.spline),check.names = FALSE)
776 anyNA(d8_irr)
777 ```
778
779
780 ### CT Bombas
781
782
783 ```{r}
784 d8_bombas <- read_excel(ruta_especifica_d8, sheet = "CT Bombas")
785 #names(d8_bombas)
786
787 nombres <- names(d8_bombas)
788 nombres <- gsub("M n.", "Min", nombres)
789 nombres <- gsub("M x.", "Max", nombres)
790 names(d8_bombas) <- nombres
791
792 head(d8_bombas[,c(1,2,3,4)],3)
793 d8_bombas$Hora <- format(d8_bombas$Hora, "%H:%M:%S")
794 Fecha <- as.POSIXct(paste(d8_bombas$Fecha, d8_bombas$Hora), format="%Y
795   -%m-%d %H:%M:%S", tz = "UTC")
796 d8_bombas<-cbind(Fecha, d8_bombas[, -c(1,2)])

```

```

796 head(d8_bombas[,c(1,2,3,4)],3)
797 ""
798
799
800 "{r, echo = FALSE, results = "hide"}
801 anyNA(d8_bombas)
802 ""
803
804
805 ### CT Nuevo
806
807
808 "{r}
809 d8_nuevo <- read_excel(ruta_especifica_d8, sheet = "CT Nuevo")
810 #names(d8_nuevo)
811
812 head(d8_nuevo[,c(1,2,3,4)],3)
813 d8_nuevo$Hora <- format(d8_nuevo$Hora, "%H:%M:%S")
814 Fecha <- as.POSIXct(paste(d8_nuevo$Fecha, d8_nuevo$Hora), format="%Y-%
      m-%d %H:%M:%S", tz = "UTC")
815 d8_nuevo<-cbind(Fecha, d8_nuevo[, -c(1,2)])
816 head(d8_nuevo[,c(1,2,3,4)],3)
817 ""
818
819
820 "{r, echo = FALSE, results = "hide"}
821 anyNA(d8_nuevo)
822 ""
823
824 ### Variables de correcci n de rangos de fechas
825
826 "{r}
827 cat("Para irr \n")
828 as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
829 as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)], origin = "1970-01-01",
      tz = "UTC")
830 cat("\nPara bombas \n")
831 as.POSIXct(d8_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
832 as.POSIXct(d8_bombas$Fecha[length(d8_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
833 cat("\nPara nuevo \n")
834 as.POSIXct(d8_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
835 as.POSIXct(d8_nuevo$Fecha[length(d8_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
836 ""
837
838 "{r}
839 cat("Para irr de bombas\n")
840 as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
841 as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)-1], origin = "1970-01-01"
      , tz = "UTC")

```

```

842 cat("\nPara bombas \n")
843 as.POSIXct(d8_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
844 as.POSIXct(d8_bombas$Fecha[length(d8_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
845 cat("\nPara irr de nuevo\n")
846 as.POSIXct(d8_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
847 as.POSIXct(d8_irr$Fecha[length(d8_irr$Fecha)], origin = "1970-01-01",
      tz = "UTC")
848 cat("\nPara nuevo \n")
849 as.POSIXct(d8_nuevo$Fecha[2], origin = "1970-01-01", tz = "UTC")
850 as.POSIXct(d8_nuevo$Fecha[length(d8_nuevo$Fecha)-2], origin = "
      1970-01-01", tz = "UTC")
851 ' ' '
852
853
854
855 '{r}
856 # Convertir fechas a POSIXct en UTC
857 f_irr    <- as.POSIXct(d8_irr$Fecha, origin = "1970-01-01", tz = "UTC")
858 f_bomb   <- as.POSIXct(d8_bombas$Fecha, origin = "1970-01-01", tz = "
      UTC")
859 f_nuev   <- as.POSIXct(d8_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
      ")
860
861 # Recortes
862 d8_irr_bombas <- d8_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
      f_irr) - 1], ]
863 d8_bombas    <- d8_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(f_bomb
      , 1), ]
864 d8_irr_nuevo  <- d8_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1), ]
865 d8_nuevo      <- d8_nuevo[f_nuev >= f_nuev[2] & f_nuev <= f_nuev[
      length(f_nuev) - 2], ]
866
867 # Verificaci n de dimensiones
868 cat(sprintf("d8_bombas: %d | d8_nuevo: %d | d8_irr_bombas: %d |
      d8_irr_nuevo: %d\n",
869             nrow(d8_bombas), nrow(d8_nuevo), nrow(d8_irr_bombas), nrow
      (d8_irr_nuevo)))
870 ' ' '
871
872
873
874
875
876
877
878
879
880
881
882

```

```

883
884
885
886 ## 9. Datos de 20-12 a 09-01
887
888 Este rango de datos contiene las dos CT.
889
890
891
892 ### Irradiancia
893
894 ```{r}
895 ruta_especifica_d9<-paste(ruta_datos, "/Datos9 de 20-12 a 09-01.xlsx",
896   sep="")
897 d9_irr <- read_excel(ruta_especifica_d9, sheet = "Irradiancia")
898 head(d9_irr)
899 ```
900 Contiene dos filas de NA's y la columna de irradiancia de CT bombas
901   contiene algo m s de 24 horas seguidas de NA's, mirar a ver que
902   hacer con ello, pues interpolar aqui no le veo el sentido.
903 ```{r, echo = FALSE, results = "hide"}
904 anyNA(d9_irr)
905 na_cols <- colSums(is.na(d9_irr))
906 na_cols <- na_cols[na_cols > 0]
907
908 which_rows_na <- which(rowSums(is.na(d9_irr[names(na_cols)])) > 0)
909
910 d9_irr[which_rows_na, ]
911 ```
912
913 Habria que ejecutar el codigo este aunque sea para las 2 filas de NA's
914 .
915 ```{r}
916 #d9_irr <- as.data.frame(lapply(d9_irr, na.spline),check.names = FALSE
917   )
918 #anyNA(d9_irr)
919 ```
920
921 ### CT Bombas
922
923
924 ```{r , message = FALSE}
925 d9_bombas <- read_excel(ruta_especifica_d9, sheet = "CT Bombas")
926 #names(d9_bombas)
927
928 nombres <- names(d9_bombas)

```

```

929 nombres <- gsub(" M n .", "Min", nombres)
930 nombres <- gsub(" M x .", "Max", nombres)
931 names(d9_bombas) <- nombres
932
933
934 #names(d9_bombas)
935 head(d9_bombas[,c(1,2,3,4)],3)
936 d9_bombas$Hora <- format(d9_bombas$Hora, "%H:%M:%S")
937 Fecha <- as.POSIXct(paste(d9_bombas$Fecha, d9_bombas$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
938 d9_bombas<-cbind(Fecha, d9_bombas[, -c(1,2)])
939 head(d9_bombas[,c(1,2,3,4)],3)
940 '''
941
942 '''{r, echo = FALSE, results = "hide"}
943 anyNA(d9_bombas)
944 '''
945
946
947 ### CT Nuevo
948
949
950 '''{r}
951 d9_nuevo <- read_excel(ruta_especifica_d9, sheet = "CT Nuevo")
952 #names(d9_nuevo)
953
954 head(d9_nuevo[,c(1,2,3,4)],3)
955 d9_nuevo$Hora <- format(d9_nuevo$Hora, "%H:%M:%S")
956 Fecha <- as.POSIXct(paste(d9_nuevo$Fecha, d9_nuevo$Hora), format="%Y-%
    m-%d %H:%M:%S", tz = "UTC")
957 d9_nuevo<-cbind(Fecha, d9_nuevo[, -c(1,2)])
958 head(d9_nuevo[,c(1,2,3,4)],3)
959 '''
960
961
962 '''{r, echo = FALSE, results = "hide"}
963 anyNA(d9_nuevo)
964 '''
965
966
967
968 ### Variables de corrección de rangos de fechas
969
970 '''{r}
971 cat("Para irr \n")
972 as.POSIXct(d9_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
973 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)], origin = "1970-01-01",
    tz = "UTC")
974 cat("\nPara bombas \n")
975 as.POSIXct(d9_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
976 as.POSIXct(d9_bombas$Fecha[length(d9_bombas$Fecha)], origin = "

```

```

1970-01-01", tz = "UTC")
977 cat("\nPara nuevo \n")
978 as.POSIXct(d9_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
979 as.POSIXct(d9_nuevo$Fecha[length(d9_nuevo$Fecha)], origin = "
1970-01-01", tz = "UTC")
980 ' ' '
981
982 ' ' '{r}
983 cat("\nPara irr de bombas\n")
984 as.POSIXct(d9_irr$Fecha[2], origin = "1970-01-01", tz = "UTC")
985 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)-7], origin = "1970-01-01"
, tz = "UTC")
986 cat("\nPara bombas \n")
987 as.POSIXct(d9_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
988 as.POSIXct(d9_bombas$Fecha[length(d9_bombas$Fecha)], origin = "
1970-01-01", tz = "UTC")
989 cat("\nPara irr de nuevo\n")
990 as.POSIXct(d9_irr$Fecha[5], origin = "1970-01-01", tz = "UTC")
991 as.POSIXct(d9_irr$Fecha[length(d9_irr$Fecha)], origin = "1970-01-01",
tz = "UTC")
992 cat("\nPara nuevo \n")
993 as.POSIXct(d9_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
994 as.POSIXct(d9_nuevo$Fecha[length(d9_nuevo$Fecha)], origin = "
1970-01-01", tz = "UTC")
995 ' ' '
996
997
998 ' ' '{r}
999 # Convertir fechas a POSIXct en UTC
1000 f_irr <- as.POSIXct(d9_irr$Fecha, origin = "1970-01-01", tz = "UTC")
1001 f_bomb <- as.POSIXct(d9_bombas$Fecha, origin = "1970-01-01", tz = "
UTC")
1002 f_nuev <- as.POSIXct(d9_nuevo$Fecha, origin = "1970-01-01", tz = "UTC
")
1003
1004 # Recortes aplicados
1005 d9_irr_bombas <- d9_irr[f_irr >= f_irr[2] & f_irr <= f_irr[length(
f_irr) - 7], ]
1006 d9_bombas <- d9_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(f_bomb
, 1), ]
1007 d9_irr_nuevo <- d9_irr[f_irr >= f_irr[5] & f_irr <= tail(f_irr, 1), ]
1008 d9_nuevo <- d9_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev,
1), ]
1009
1010 # Verificaci n
1011 cat(sprintf("d9_bombas: %d | d9_nuevo: %d | d9_irr_bombas: %d |
d9_irr_nuevo: %d\n",
1012 nrow(d9_bombas), nrow(d9_nuevo), nrow(d9_irr_bombas), nrow
(d9_irr_nuevo)))
1013 ' ' '
1014

```

```

1015
1016
1017 Una vez hecho todo esto, si se pueden eliminar bien las filas
1018 ```{r}
1019 filas_na <- which(is.na(d9_irr_bombas$Irradiancia_CTBombas))
1020 filas_eliminar <- filas_na[1:(length(filas_na) - 2)]
1021 filas_eliminar
1022 ```
1023
1024
1025 ```{r}
1026 d9_irr_bombas <- d9_irr_bombas[-filas_eliminar, ]
1027 d9_bombas <- d9_bombas[-filas_eliminar, ]
1028 ```
1029
1030 ```{r}
1031 d9_irr_bombas <- as.data.frame(lapply(d9_irr_bombas, na.spline), check.
      names = FALSE)
1032 anyNA(d9_irr_bombas)
1033 ```
1034
1035 ```{r}
1036 d9_irr_nuevo <- d9_irr_nuevo[, !names(d9_irr_nuevo) %in% "
      Irradiancia_CTBombas"]
1037 ```
1038
1039 ```{r}
1040 anyNA(d9_irr_nuevo)
1041 d9_irr_nuevo <- as.data.frame(lapply(d9_irr_nuevo, na.spline), check.
      names = FALSE)
1042 anyNA(d9_irr_nuevo)
1043 ```
1044
1045 ```{r}
1046 cat(sprintf("d9_bombas: %d | d9_nuevo: %d | d9_irr_bombas: %d |
      d9_irr_nuevo: %d\n",
1047           nrow(d9_bombas), nrow(d9_nuevo), nrow(d9_irr_bombas), nrow
      (d9_irr_nuevo)))
1048 ```
1049
1050
1051
1052
1053
1054
1055 ## 10. Datos de 09-01 a 30-01
1056
1057 Este rango de datos contiene las dos CT.
1058
1059
1060

```

```

1061 ### Irradiancia
1062
1063 ```{r}
1064 ruta_especifica_d10<-paste(ruta_datos, "/Datos10 de 09-01 a 30-01.xlsx", sep="")
1065 d10_irr <- read_excel(ruta_especifica_d10, sheet = "Irradiancia")
1066 head(d10_irr)
1067 ```
1068
1069 Contiene una fila de NA's
1070 ```{r, echo = FALSE, results = "hide"}
1071 anyNA(d10_irr)
1072 na_cols <- colSums(is.na(d10_irr))
1073 na_cols <- na_cols[na_cols > 0]
1074
1075 which_rows_na <- which(rowSums(is.na(d10_irr[names(na_cols)])) > 0)
1076
1077 d10_irr[which_rows_na, ]
1078 ```
1079
1080 ```{r}
1081 d10_irr <- as.data.frame(lapply(d10_irr, na.spline), check.names =
1082 FALSE)
1083 anyNA(d10_irr)
1084 ```
1085
1086 ### CT Bombas
1087
1088
1089 ```{r}
1090 d10_bombas <- read_excel(ruta_especifica_d10, sheet = "CT Bombas")
1091 #names(d10_bombas)
1092
1093 nombres <- names(d10_bombas)
1094 nombres <- gsub("M n .", "Min", nombres)
1095 nombres <- gsub("M x .", "Max", nombres)
1096 names(d10_bombas) <- nombres
1097
1098 #names(d10_bombas)
1099 head(d10_bombas[,c(1,2,3,4)],3)
1100 d10_bombas$Hora <- format(d10_bombas$Hora, "%H:%M:%S")
1101 Fecha <- as.POSIXct(paste(d10_bombas$Fecha, d10_bombas$Hora), format="
1102 %Y-%m-%d %H:%M:%S", tz = "UTC")
1103 d10_bombas<-cbind(Fecha, d10_bombas[, -c(1,2)])
1104 head(d10_bombas[,c(1,2,3,4)],3)
1105 ```
1106
1107 ```{r, echo = FALSE, results = "hide"}
1108 anyNA(d10_bombas)
1109 ```

```

```

1109
1110
1111 ### CT Nuevo
1112
1113
1114 ```{r}
1115 d10_nuevo <- read_excel(ruta_especifica_d10, sheet = "CT Nuevo")
1116 #names(d10_nuevo)
1117
1118 head(d10_nuevo[,c(1,2,3,4)],3)
1119 d10_nuevo$Hora <- format(d10_nuevo$Hora, "%H:%M:%S")
1120 Fecha <- as.POSIXct(paste(d10_nuevo$Fecha, d10_nuevo$Hora), format="%Y
      -%m-%d %H:%M:%S", tz = "UTC")
1121 d10_nuevo<-cbind(Fecha, d10_nuevo[, -c(1,2)])
1122 head(d10_nuevo[,c(1,2,3,4)],3)
1123 ```
1124
1125
1126 ```{r, echo = FALSE, results = "hide"}
1127 anyNA(d10_nuevo)
1128 ```
1129
1130
1131 ### Variables de corrección de rangos de fechas
1132
1133 ```{r}
1134 cat("Para irr \n")
1135 as.POSIXct(d10_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1136 as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)], origin = "1970-01-01"
      , tz = "UTC")
1137 cat("\nPara bombas \n")
1138 as.POSIXct(d10_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1139 as.POSIXct(d10_bombas$Fecha[length(d10_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1140 cat("\nPara nuevo \n")
1141 as.POSIXct(d10_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1142 as.POSIXct(d10_nuevo$Fecha[length(d10_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1143 ```
1144
1145 ```{r}
1146 cat("Para irr de bombas\n")
1147 as.POSIXct(d10_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1148 as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)-134], origin = "
      1970-01-01", tz = "UTC")
1149 cat("\nPara bombas \n")
1150 as.POSIXct(d10_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1151 as.POSIXct(d10_bombas$Fecha[length(d10_bombas$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1152 cat("\nPara irr de nuevo\n")
1153 as.POSIXct(d10_irr$Fecha[7], origin = "1970-01-01", tz = "UTC")

```

```

1154 as.POSIXct(d10_irr$Fecha[length(d10_irr$Fecha)-1], origin = "
      1970-01-01", tz = "UTC")
1155 cat("\nPara nuevo \n")
1156 as.POSIXct(d10_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1157 as.POSIXct(d10_nuevo$Fecha[length(d10_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1158 ' ' '
1159
1160
1161 '{r}
1162 # Convertir fechas a POSIXct en UTC
1163 f_irr <- as.POSIXct(d10_irr$Fecha, origin = "1970-01-01", tz = "UTC"
      )
1164 f_bomb <- as.POSIXct(d10_bombas$Fecha, origin = "1970-01-01", tz = "
      UTC")
1165 f_nuev <- as.POSIXct(d10_nuevo$Fecha, origin = "1970-01-01", tz = "
      UTC")
1166
1167 # Recortes
1168 d10_irr_bombas <- d10_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
      f_irr) - 134], ]
1169 d10_bombas <- d10_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
      f_bomb, 1), ]
1170 d10_irr_nuevo <- d10_irr[f_irr >= f_irr[7] & f_irr <= f_irr[length(
      f_irr) - 1], ]
1171 d10_nuevo <- d10_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
      f_nuev, 1), ]
1172
1173 # Verificaci n
1174 cat(sprintf("d10_bombas: %d | d10_nuevo: %d | d10_irr_bombas: %d |
      d10_irr_nuevo: %d\n",
1175           nrow(d10_bombas), nrow(d10_nuevo), nrow(d10_irr_bombas),
      nrow(d10_irr_nuevo)))
1176 ' ' '
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189 ## 11. Datos de 30-01 a 18-02
1190
1191 Este rango de datos solo contiene CT nuevo.
1192
1193

```

```

1194 |
1195 | ### Irradiancia
1196 |
1197 | ```{r}
1198 | ruta_especifica_d11<-paste(ruta_datos, "/Datos11 de 30-01 a 18-02.xlsx
1199 |                               ",sep="")
1200 | d11_irr <- read_excel(ruta_especifica_d11, sheet = "Irradiancia")
1201 | head(d11_irr)
1202 | ```
1203 | Contiene una fila de NA's
1204 | ```{r, echo = FALSE, results = "hide"}
1205 | anyNA(d11_irr)
1206 | na_cols <- colSums(is.na(d11_irr))
1207 | na_cols <- na_cols[na_cols > 0]
1208 |
1209 | which_rows_na <- which(rowSums(is.na(d11_irr[names(na_cols)])) > 0)
1210 |
1211 | d11_irr[which_rows_na, ]
1212 | ```
1213 |
1214 | ```{r}
1215 | d11_irr <- as.data.frame(lapply(d11_irr, na.spline),check.names =
1216 |                             FALSE)
1217 | anyNA(d11_irr)
1218 | ```
1219 |
1220 | ### CT Nuevo
1221 |
1222 |
1223 | ```{r}
1224 | d11_nuevo <- read_excel(ruta_especifica_d11, sheet = "CT Nuevo")
1225 | #names(d11_nuevo)
1226 |
1227 | head(d11_nuevo[,c(1,2,3,4)],3)
1228 | d11_nuevo$Hora <- format(d11_nuevo$Hora, "%H:%M:%S")
1229 | Fecha <- as.POSIXct(paste(d11_nuevo$Fecha, d11_nuevo$Hora), format="%Y
1230 |                       -%m-%d %H:%M:%S", tz = "UTC")
1231 | d11_nuevo<-cbind(Fecha, d11_nuevo[, -c(1,2)])
1232 | head(d11_nuevo[,c(1,2,3,4)],3)
1233 | ```
1234 |
1235 | ```{r, echo = FALSE, results = "hide"}
1236 | anyNA(d11_nuevo)
1237 | ```
1238 |
1239 |
1240 | ### Variables de corrección de rangos de fechas
1241 |

```

```

1242 Hay descuadre pero e cambio da igual en este caso, pues el que
      corresponde sigue siendo el m s pr ximo.
1243 ```{r}
1244 cat("Para irr \n")
1245 as.POSIXct(d11_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1246 as.POSIXct(d11_irr$Fecha[length(d11_irr$Fecha)], origin = "1970-01-01"
      , tz = "UTC")
1247 cat("\nPara nuevo \n")
1248 as.POSIXct(d11_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1249 as.POSIXct(d11_nuevo$Fecha[length(d11_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1250 ```
1251
1252 ```{r}
1253 cat("Para irr \n")
1254 as.POSIXct(d11_irr$Fecha[12], origin = "1970-01-01", tz = "UTC")
1255 as.POSIXct(d11_irr$Fecha[length(d11_irr$Fecha)], origin = "1970-01-01"
      , tz = "UTC")
1256 cat("\nPara nuevo \n")
1257 as.POSIXct(d11_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1258 as.POSIXct(d11_nuevo$Fecha[length(d11_nuevo$Fecha)], origin = "
      1970-01-01", tz = "UTC")
1259 ```
1260
1261
1262
1263 ```{r}
1264 # Convertir fechas a POSIXct en UTC
1265 f_irr <- as.POSIXct(d11_irr$Fecha, origin = "1970-01-01", tz = "UTC")
1266 f_nuev <- as.POSIXct(d11_nuevo$Fecha, origin = "1970-01-01", tz = "UTC"
      ")
1267
1268 # Recortes
1269 d11_irr_nuevo <- d11_irr[f_irr >= f_irr[12] & f_irr <= tail(f_irr,
      1), ]
1270 d11_nuevo <- d11_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(f_nuev, 1)
      , ]
1271
1272 # Verificaci n
1273 cat(sprintf("d11_irr_nuevo: %d | d11_nuevo: %d\n", nrow(d11_irr_nuevo)
      , nrow(d11_nuevo)))
1274 ```
1275
1276
1277
1278
1279
1280
1281
1282
1283

```

```

1284
1285
1286 ## 12. Datos de 18-02 a 12-03
1287
1288 Este rango de datos contiene las dos CT.
1289
1290
1291 ### Irradiancia
1292
1293 ```{r}
1294 ruta_especifica_d12<-paste(ruta_datos, "/Datos12 de 18-02 a 12-03.xlsx
1295 ",sep="")
1296 d12_irr <- read_excel(ruta_especifica_d12, sheet = "Irradiancia")
1297 head(d12_irr)
1298 ```
1299
1300 Contiene tres filas seguidas de NA's
1301 ```{r, echo = FALSE, results = "hide"}
1302 anyNA(d12_irr)
1303 na_cols <- colSums(is.na(d12_irr))
1304 na_cols <- na_cols[na_cols > 0]
1305
1306 which_rows_na <- which(rowSums(is.na(d12_irr[names(na_cols)])) > 0)
1307
1308 d12_irr[which_rows_na, ]
1309 ```
1310
1311 ```{r}
1312 d12_irr <- as.data.frame(lapply(d12_irr, na.spline),check.names =
1313 FALSE)
1314 anyNA(d12_irr)
1315 ```
1316
1317 ### CT Bombas
1318
1319 ```{r , message = FALSE}
1320 d12_bombas <- read_excel(ruta_especifica_d12, sheet = "CT Bombas")
1321 #names(d12_bombas)
1322
1323 nombres <- names(d12_bombas)
1324 nombres <- gsub(" M n .", "Min", nombres)
1325 nombres <- gsub(" M x .", "Max", nombres)
1326 names(d12_bombas) <- nombres
1327
1328 #names(d12_bombas)
1329 head(d12_bombas[,c(1,2,3,4)],3)
1330 d12_bombas$Hora <- format(d12_bombas$Hora, "%H:%M:%S")
1331 Fecha <- as.POSIXct(paste(d12_bombas$Fecha, d12_bombas$Hora), format="
1332 %Y-%m-%d %H:%M:%S", tz = "UTC")

```

```

1332 d12_bombas<-cbind(Fecha, d12_bombas[,-c(1,2)])
1333 head(d12_bombas[,c(1,2,3,4)],3)
1334 '''
1335
1336
1337 '{{{r, echo = FALSE, results = "hide"}}
1338 anyNA(d12_bombas)
1339 '''
1340
1341
1342 ### CT Nuevo
1343
1344
1345 '{{{r}
1346 d12_nuevo <- read_excel(ruta_especifica_d12, sheet = "CT Nuevo")
1347 #names(d12_nuevo)
1348
1349 head(d12_nuevo[,c(1,2,3,4)],3)
1350 d12_nuevo$Hora <- format(d12_nuevo$Hora, "%H:%M:%S")
1351 Fecha <- as.POSIXct(paste(d12_nuevo$Fecha, d12_nuevo$Hora), format="%Y
    -%m-%d %H:%M:%S", tz = "UTC")
1352 d12_nuevo<-cbind(Fecha, d12_nuevo[,-c(1,2)])
1353 head(d12_nuevo[,c(1,2,3,4)],3)
1354 '''
1355
1356
1357 '{{{r, echo = FALSE, results = "hide"}}
1358 anyNA(d12_nuevo)
1359 '''
1360
1361
1362
1363 ### Variables de correcci n de rangos de fechas
1364
1365 '{{{r}
1366 cat("Para irr \n")
1367 as.POSIXct(d12_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1368 as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1369 cat("\nPara bombas \n")
1370 as.POSIXct(d12_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1371 as.POSIXct(d12_bombas$Fecha[length(d12_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1372 cat("\nPara nuevo \n")
1373 as.POSIXct(d12_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1374 as.POSIXct(d12_nuevo$Fecha[length(d12_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1375 '''
1376
1377 '{{{r}
1378 cat("Para irr de bombas\n")

```

```

1379 as.POSIXct(d12_irr$Fecha[1008], origin = "1970-01-01", tz = "UTC")
1380 as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)-4], origin = "
1381   1970-01-01", tz = "UTC")
1382 cat("\nPara bombas \n")
1383 as.POSIXct(d12_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1384 as.POSIXct(d12_bombas$Fecha[length(d12_bombas$Fecha)], origin = "
1385   1970-01-01", tz = "UTC")
1386 cat("\nPara irr de nuevo\n")
1387 as.POSIXct(d12_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1388 as.POSIXct(d12_irr$Fecha[length(d12_irr$Fecha)], origin = "1970-01-01"
1389   , tz = "UTC")
1390 cat("\nPara nuevo \n")
1391 as.POSIXct(d12_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1392 as.POSIXct(d12_nuevo$Fecha[length(d12_nuevo$Fecha)], origin = "
1393   1970-01-01", tz = "UTC")
1394 ' ' '
1395 ' ' {r}
1396 # Convertir fechas a POSIXct en UTC
1397 f_irr    <- as.POSIXct(d12_irr$Fecha, origin = "1970-01-01", tz = "UTC"
1398   )
1399 f_bomb    <- as.POSIXct(d12_bombas$Fecha, origin = "1970-01-01", tz = "
1400   UTC")
1401 f_nuev    <- as.POSIXct(d12_nuevo$Fecha, origin = "1970-01-01", tz = "
1402   UTC")
1403 # Recortes
1404 d12_irr_bombas <- d12_irr[f_irr >= f_irr[1008] & f_irr <= f_irr[length
1405   (f_irr) - 4], ]
1406 d12_bombas    <- d12_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
1407   f_bomb, 1), ]
1408 d12_irr_nuevo <- d12_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1),
1409   ]
1410 d12_nuevo     <- d12_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
1411   f_nuev, 1), ]
1412 # Verificaci n
1413 cat(sprintf("d12_bombas: %d | d12_nuevo: %d | d12_irr_bombas: %d |
1414   d12_irr_nuevo: %d\n",
1415     nrow(d12_bombas), nrow(d12_nuevo), nrow(d12_irr_bombas),
1416     nrow(d12_irr_nuevo)))
1417 ' ' '

```

```

1417
1418
1419
1420 ## 13. Datos de 12-03 a 03-04
1421
1422 Este rango de datos contiene las dos CT.
1423
1424
1425 ### Irradiancia
1426
1427 ```{r}
1428 ruta_especifica_d13<-paste(ruta_datos, "/Datos13 de 12-03 a 03-04.xlsx
      ",sep="")
1429 d13_irr <- read_excel(ruta_especifica_d13, sheet = "Irradiancia")
1430 head(d13_irr)
1431 ```
1432
1433
1434 ```{r, echo = FALSE, results = "hide"}
1435 anyNA(d13_irr)
1436 ```
1437
1438 ### CT Bombas
1439
1440
1441 ```{r , message = FALSE}
1442 d13_bombas <- read_excel(ruta_especifica_d13, sheet = "CT Bombas")
1443 #names(d13_bombas)
1444
1445 nombres <- names(d13_bombas)
1446 nombres <- gsub("M n.", "Min", nombres)
1447 nombres <- gsub("M x.", "Max", nombres)
1448 names(d13_bombas) <- nombres
1449
1450 fila_inicial <- 2541 # Cambia este valor seg n d nde quieras
      aplicar el ajuste
1451
1452 # Sumar 60 minutos a partir de la fila indicada
1453 d13_bombas$Hora[fila_inicial:nrow(d13_bombas)] <- d13_bombas$Hora[
      fila_inicial:nrow(d13_bombas)] + 60*60 # 60
1454
1455 #names(d13_bombas)
1456 head(d13_bombas[,c(1,2,3,4)],3)
1457 d13_bombas$Hora <- format(d13_bombas$Hora, "%H:%M:%S")
1458 Fecha <- as.POSIXct(paste(d13_bombas$Fecha, d13_bombas$Hora), format="
      %Y-%m-%d %H:%M:%S", tz="UTC")
1459 d13_bombas<-cbind(Fecha, d13_bombas[, -c(1,2)])
1460 head(d13_bombas[,c(1,2,3,4)],3)
1461 ```
1462
1463 ```{r, echo = FALSE, results = "hide"}

```

```

1464 anyNA(d13_bombas)
1465 ' ' '
1466
1467
1468 ### CT Nuevo
1469
1470
1471 '{r}
1472 d13_nuevo <- read_excel(ruta_especifica_d13, sheet = "CT Nuevo")
1473 #names(d13_nuevo)
1474
1475 fila_inicial <- 2538 # Cambia este valor seg n d nde quieras
1476     aplicar el ajuste
1477
1478 # Sumar 60 minutos a partir de la fila indicada
1479 d13_nuevo$Hora[fila_inicial:nrow(d13_nuevo)] <- d13_nuevo$Hora[
1480     fila_inicial:nrow(d13_nuevo)] + 60*60 # 6
1481
1482 head(d13_nuevo[,c(1,2,3,4)],3)
1483 d13_nuevo$Hora <- format(d13_nuevo$Hora, "%H:%M:%S")
1484 Fecha <- as.POSIXct(paste(d13_nuevo$Fecha, d13_nuevo$Hora), format="%Y
1485     -%m-%d %H:%M:%S", tz="UTC")
1486
1487 d13_nuevo<-cbind(Fecha, d13_nuevo[, -c(1,2)])
1488 head(d13_nuevo[,c(1,2,3,4)],3)
1489 ' ' '
1490
1491 Lo mismo de nuevo.
1492 '{r, echo = FALSE, results = "hide"}
1493 anyNA(d13_nuevo)
1494 ' ' '
1495
1496 ### Variables de correcci n de rangos de fechas
1497
1498 '{r}
1499 cat("Para irr \n")
1500 as.POSIXct(d13_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1501 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)], origin = "1970-01-01"
1502     , tz = "UTC")
1503 cat("\nPara bombas \n")
1504 as.POSIXct(d13_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1505 as.POSIXct(d13_bombas$Fecha[length(d13_bombas$Fecha)], origin = "
1506     1970-01-01", tz = "UTC")
1507 cat("\nPara nuevo \n")
1508 as.POSIXct(d13_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1509 as.POSIXct(d13_nuevo$Fecha[length(d13_nuevo$Fecha)], origin = "
1510     1970-01-01", tz = "UTC")
1511 ' ' '
1512
1513 '{r}
1514 cat("Para irr de bombas\n")

```

```

1509 as.POSIXct(d13_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1510 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)-306], origin = "
1511 1970-01-01", tz = "UTC")
1512 cat("\nPara bombas \n")
1513 as.POSIXct(d13_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1514 as.POSIXct(d13_bombas$Fecha[length(d13_bombas$Fecha)], origin = "
1515 1970-01-01", tz = "UTC")
1516 cat("\nPara irr de nuevo\n")
1517 as.POSIXct(d13_irr$Fecha[4], origin = "1970-01-01", tz = "UTC")
1518 as.POSIXct(d13_irr$Fecha[length(d13_irr$Fecha)-1], origin = "
1519 1970-01-01", tz = "UTC")
1520 cat("\nPara nuevo \n")
1521 as.POSIXct(d13_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1522 as.POSIXct(d13_nuevo$Fecha[length(d13_nuevo$Fecha)-6], origin = "
1523 1970-01-01", tz = "UTC")
1524 ' ' '
1525 ' ' '{r}
1526 # Convertir fechas
1527 f_irr <- as.POSIXct(d13_irr$Fecha, origin = "1970-01-01", tz = "UTC"
1528 )
1529 f_bomba <- as.POSIXct(d13_bombas$Fecha, origin = "1970-01-01", tz = "
1530 UTC")
1531 f_nuevo <- as.POSIXct(d13_nuevo$Fecha, origin = "1970-01-01", tz = "
1532 UTC")
1533 # Recortes
1534 d13_irr_bombas <- d13_irr[f_irr <= f_irr[length(f_irr)-306], ]
1535 d13_nuevo <- d13_nuevo[f_nuevo <= f_nuevo[length(f_nuevo)-6], ]
1536 d13_irr_nuevo <- d13_irr[f_irr >= f_irr[4] & f_irr <= f_irr[length(
1537 f_irr)-1], ]
1538 # Imprimir n mero de filas
1539 cat(sprintf("d13_bombas: %d | d13_irr_bombas: %d | d13_nuevo: %d |
1540 d13_irr_nuevo: %d\n",
1541 nrow(d13_bombas), nrow(d13_irr_bombas), nrow(d13_nuevo),
1542 nrow(d13_irr_nuevo)))
1543 ' ' '
1544 ## 14. Datos de 03-04 a 24-04
1545
1546 Este rango de datos contiene las dos CT.
1547
1548
1549 ### Irradiancia

```

```

1550
1551   ```{r}
1552 ruta_especifica_d14<-paste(ruta_datos , "/Datos14 de 03-04 a 24-04.xlsx
      ",sep="")
1553 d14_irr <- read_excel(ruta_especifica_d14, sheet = "Irradiancia")
1554 head(d14_irr)
1555   ```
1556
1557
1558   ```{r, echo = FALSE, results = "hide"}
1559 anyNA(d14_irr)
1560   ```
1561
1562
1563 ### CT Bombas
1564
1565
1566   ```{r , message = FALSE}
1567 d14_bombas <- read_excel(ruta_especifica_d14, sheet = "CT Bombas")
1568 #names(d14_bombas)
1569
1570 nombres <- names(d14_bombas)
1571 nombres <- gsub(" M n .", "Min", nombres)
1572 nombres <- gsub(" M x .", "Max", nombres)
1573 names(d14_bombas) <- nombres
1574
1575 #names(d14_bombas)
1576 head(d14_bombas[,c(1,2,3,4)],3)
1577 d14_bombas$Hora <- format(d14_bombas$Hora , "%H:%M:%S")
1578 Fecha <- as.POSIXct(paste(d14_bombas$Fecha, d14_bombas$Hora), format="
      %Y-%m-%d %H:%M:%S", tz = "UTC")
1579 d14_bombas<-cbind(Fecha, d14_bombas[, -c(1,2)])
1580 head(d14_bombas[,c(1,2,3,4)],3)
1581   ```
1582
1583
1584   ```{r, echo = FALSE, results = "hide"}
1585 anyNA(d14_bombas)
1586   ```
1587
1588
1589 ### CT Nuevo
1590
1591
1592   ```{r}
1593 d14_nuevo <- read_excel(ruta_especifica_d14, sheet = "CT Nuevo")
1594 #names(d14_nuevo)
1595
1596 head(d14_nuevo[,c(1,2,3,4)],3)
1597 d14_nuevo$Hora <- format(d14_nuevo$Hora , "%H:%M:%S")
1598 Fecha <- as.POSIXct(paste(d14_nuevo$Fecha, d14_nuevo$Hora), format="%Y

```

```

    -%m-%d %H:%M:%S", tz = "UTC")
1599 d14_nuevo<-cbind(Fecha, d14_nuevo[, -c(1,2)])
1600 head(d14_nuevo[, c(1,2,3,4)], 3)
1601 ""
1602
1603
1604 ““{r, echo = FALSE, results = "hide"}
1605 anyNA(d14_nuevo)
1606 ““
1607
1608
1609 ### Variables de correcci n de rangos de fechas
1610
1611 ““{r}
1612 cat("Para irr \n")
1613 as.POSIXct(d14_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1614 as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)], origin = "1970-01-01"
    , tz = "UTC")
1615 cat("\nPara bombas \n")
1616 as.POSIXct(d14_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1617 as.POSIXct(d14_bombas$Fecha[length(d14_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1618 cat("\nPara nuevo \n")
1619 as.POSIXct(d14_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1620 as.POSIXct(d14_nuevo$Fecha[length(d14_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1621 ““
1622
1623 ““{r}
1624 cat("Para irr de bombas\n")
1625 as.POSIXct(d14_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1626 as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)-169], origin = "
    1970-01-01", tz = "UTC")
1627 cat("\nPara bombas \n")
1628 as.POSIXct(d14_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1629 as.POSIXct(d14_bombas$Fecha[length(d14_bombas$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1630 cat("\nPara irr de nuevo\n")
1631 as.POSIXct(d14_irr$Fecha[5], origin = "1970-01-01", tz = "UTC")
1632 as.POSIXct(d14_irr$Fecha[length(d14_irr$Fecha)-1], origin = "
    1970-01-01", tz = "UTC")
1633 cat("\nPara nuevo \n")
1634 as.POSIXct(d14_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1635 as.POSIXct(d14_nuevo$Fecha[length(d14_nuevo$Fecha)], origin = "
    1970-01-01", tz = "UTC")
1636 ““
1637
1638 ““{r}
1639 # Convertir fechas a POSIXct en UTC
1640 f_irr    <- as.POSIXct(d14_irr$Fecha, origin = "1970-01-01", tz = "UTC"
    )

```

```

1641 f_bomb   <- as.POSIXct(d14_bombas$Fecha, origin = "1970-01-01", tz = "
      UTC")
1642 f_nuev   <- as.POSIXct(d14_nuevo$Fecha, origin = "1970-01-01", tz = "
      UTC")
1643
1644 # Recortes
1645 d14_irr_bombas <- d14_irr[f_irr >= f_irr[1] & f_irr <= f_irr[length(
      f_irr) - 169], ]
1646 d14_bombas    <- d14_bombas[f_bomb >= f_bomb[1] & f_bomb <= tail(
      f_bomb, 1), ]
1647 d14_irr_nuevo <- d14_irr[f_irr >= f_irr[5] & f_irr <= f_irr[length(
      f_irr) - 1], ]
1648 d14_nuevo     <- d14_nuevo[f_nuev >= f_nuev[1] & f_nuev <= tail(
      f_nuev, 1), ]
1649
1650 # Verificaci n
1651 cat(sprintf("d14_bombas: %d | d14_nuevo: %d | d14_irr_bombas: %d |
      d14_irr_nuevo: %d\n",
1652             nrow(d14_bombas), nrow(d14_nuevo), nrow(d14_irr_bombas),
              nrow(d14_irr_nuevo)))
1653
1654 ' ' '
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671 ## 15. Datos de 24-04 a 12-05
1672
1673 Este rango de datos solo contiene CT bombas.
1674
1675
1676 ### Irradiancia
1677
1678 '{r}
1679 ruta_especifica_d15<-paste(ruta_datos, "/Datos15 de 24-04 a 12-05.xlsx
      ", sep="")
1680 d15_irr <- read_excel(ruta_especifica_d15, sheet = "Irradiancia")
1681 head(d15_irr)
1682 ' ' '

```

```

1683
1684 Contiene 4 horas y media de NA's
1685 ```{r, echo = FALSE, results = "hide"}
1686 anyNA(d15_irr)
1687 na_cols <- colSums(is.na(d15_irr))
1688 na_cols <- na_cols[na_cols > 0]
1689
1690 which_rows_na <- which(rowSums(is.na(d15_irr[names(na_cols)])) > 0)
1691
1692 d15_irr[which_rows_na, ]
1693 ```
1694
1695 Son demasiadas filas para hacer interpolaci n yo creo.
1696 ```{r}
1697 #d15_irr <- as.data.frame(lapply(d15_irr, na.spline),check.names =
1698   FALSE)
1699 #anyNA(d15_irr)
1700 ```
1701
1702
1703 ### CT Bombas
1704
1705
1706 ```{r , message = FALSE}
1707 d15_bombas <- read_excel(ruta_especifica_d15, sheet = "CT Bombas")
1708 #names(d15_bombas)
1709
1710 nombres <- names(d15_bombas)
1711 nombres <- gsub("M n .", "Min", nombres)
1712 nombres <- gsub("M x .", "Max", nombres)
1713 names(d15_bombas) <- nombres
1714
1715 #names(d15_bombas)
1716 head(d15_bombas[,c(1,2,3,4)],3)
1717 d15_bombas$Hora <- format(d15_bombas$Hora, "%H:%M:%S")
1718 Fecha <- as.POSIXct(paste(d15_bombas$Fecha, d15_bombas$Hora), format="
1719   %Y-%m-%d %H:%M:%S", tz = "UTC")
1720 d15_bombas<-cbind(Fecha, d15_bombas[, -c(1,2)])
1721 head(d15_bombas[,c(1,2,3,4)],3)
1722 ```
1723
1724 ```{r, echo = FALSE, results = "hide"}
1725 anyNA(d15_bombas)
1726 ```
1727
1728
1729 ### Variables de correcci n de rangos de fechas
1730
1731 ```{r}

```

```

1732 cat("Para irr \n")
1733 as.POSIXct(d15_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1734 as.POSIXct(d15_irr$Fecha[length(d15_irr$Fecha)], origin = "1970-01-01"
, tz = "UTC")
1735 cat("\nPara bombas \n")
1736 as.POSIXct(d15_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1737 as.POSIXct(d15_bombas$Fecha[length(d15_bombas$Fecha)], origin = "
1970-01-01", tz = "UTC")
1738 ' ' '
1739
1740
1741 '{r}
1742 d15_irr_bombas <- d15_irr
1743
1744 cat(sprintf("d15_bombas: %d | d15_irr_bombas: %d\n",
1745 nrow(d15_bombas), nrow(d15_irr_bombas)))
1746 ' ' '
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760 ## 16. Datos de 12-05 a 27-05
1761
1762 Este rango de datos contiene las dos CT.
1763
1764
1765 ### Irradiancia
1766
1767 '{r}
1768 ruta_especifica_d16<-paste(ruta_datos, "/Datos16 de 12-05 a 27-05.xlsx
",sep="")
1769 d16_irr <- read_excel(ruta_especifica_d16, sheet = "Irradiancia")
1770 head(d16_irr)
1771 ' ' '
1772
1773 Contiene una fila de NA's
1774 '{r, echo = FALSE, results = "hide"}
1775 anyNA(d16_irr)
1776 na_cols <- colSums(is.na(d16_irr))
1777 na_cols <- na_cols[na_cols > 0]
1778
1779 which_rows_na <- which(rowSums(is.na(d16_irr[names(na_cols)])) > 0)

```

```

1780
1781 d16_irr[which_rows_na, ]
1782 ""
1783
1784 ""{r}
1785 d16_irr <- as.data.frame(lapply(d16_irr, na.spline), check.names =
1786 FALSE)
1787 anyNA(d16_irr)
1788 ""
1789
1790 ### CT Bombas
1791
1792 ""{r}
1793 d16_bombas <- read_excel(ruta_especifica_d16, sheet = "CT Bombas")
1794 #names(d16_bombas)
1795
1796 nombres <- names(d16_bombas)
1797 nombres <- gsub("M n.", "Min", nombres)
1798 nombres <- gsub("M x.", "Max", nombres)
1799 names(d16_bombas) <- nombres
1800
1801 #names(d16_bombas)
1802 head(d16_bombas[,c(1,2,3,4)],3)
1803 d16_bombas$Hora <- format(d16_bombas$Hora, "%H:%M:%S")
1804 Fecha <- as.POSIXct(paste(d16_bombas$Fecha, d16_bombas$Hora), format="
1805 %Y-%m-%d %H:%M:%S", tz = "UTC")
1806 d16_bombas<-cbind(Fecha, d16_bombas[, -c(1,2)])
1807 head(d16_bombas[,c(1,2,3,4)],3)
1808 ""
1809
1810 ""{r, echo = FALSE, results = "hide"}
1811 anyNA(d16_bombas)
1812 ""
1813
1814
1815
1816 ### CT Nuevo
1817
1818 ""{r}
1819 d16_nuevo <- read_excel(ruta_especifica_d16, sheet = "CT Nuevo")
1820 #names(d16_nuevo)
1821
1822 head(d16_nuevo[,c(1,2,3,4)],3)
1823 d16_nuevo$Hora <- format(d16_nuevo$Hora, "%H:%M:%S")
1824 Fecha <- as.POSIXct(paste(d16_nuevo$Fecha, d16_nuevo$Hora), format="%Y
1825 -%m-%d %H:%M:%S", tz = "UTC")
1826 d16_nuevo<-cbind(Fecha, d16_nuevo[, -c(1,2)])
1827 head(d16_nuevo[,c(1,2,3,4)],3)

```

```

1828   '
1829
1830   '{r, echo = FALSE, results = "hide"}
1831   anyNA(d16_nuevo)
1832   '
1833
1834
1835
1836
1837   ### Variables de corrección de rangos de fechas
1838
1839   '{r}
1840   cat("Para irr \n")
1841   as.POSIXct(d16_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1842   as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1843     , tz = "UTC")
1844   cat("\nPara bombas \n")
1845   as.POSIXct(d16_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1846   as.POSIXct(d16_bombas$Fecha[length(d16_bombas$Fecha)], origin = "
1847     1970-01-01", tz = "UTC")
1848   cat("\nPara nuevo \n")
1849   as.POSIXct(d16_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1850   as.POSIXct(d16_nuevo$Fecha[length(d16_nuevo$Fecha)], origin = "
1851     1970-01-01", tz = "UTC")
1852   '
1853
1854   '{r}
1855   cat("Para irr de bombas\n")
1856   as.POSIXct(d16_irr$Fecha[1], origin = "1970-01-01", tz = "UTC")
1857   as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1858     , tz = "UTC")
1859   cat("\nPara bombas \n")
1860   as.POSIXct(d16_bombas$Fecha[1], origin = "1970-01-01", tz = "UTC")
1861   as.POSIXct(d16_bombas$Fecha[length(d16_bombas$Fecha)-414], origin = "
1862     1970-01-01", tz = "UTC")
1863   cat("\nPara irr de nuevo\n")
1864   as.POSIXct(d16_irr$Fecha[7], origin = "1970-01-01", tz = "UTC")
1865   as.POSIXct(d16_irr$Fecha[length(d16_irr$Fecha)], origin = "1970-01-01"
1866     , tz = "UTC")
1867   cat("\nPara nuevo \n")
1868   as.POSIXct(d16_nuevo$Fecha[1], origin = "1970-01-01", tz = "UTC")
1869   as.POSIXct(d16_nuevo$Fecha[length(d16_nuevo$Fecha)-413], origin = "
1870     1970-01-01", tz = "UTC")
1871   '
1872
1873
1874   '{r}
1875   # Convertir fechas a POSIXct en UTC
1876   f_irr    <- as.POSIXct(d16_irr$Fecha, origin = "1970-01-01", tz = "UTC"
1877     )
1878   f_bomb   <- as.POSIXct(d16_bombas$Fecha, origin = "1970-01-01", tz = "

```

```

    UTC")
1871 f_nuev <- as.POSIXct(d16_nuevo$Fecha, origin = "1970-01-01", tz = "
    UTC")
1872
1873 # Recortes
1874 d16_irr_bombas <- d16_irr[f_irr >= f_irr[1] & f_irr <= tail(f_irr, 1),
    ]
1875 d16_bombas <- d16_bombas[f_bomb >= f_bomb[1] & f_bomb <= f_bomb[
    length(f_bomb) - 414], ]
1876 d16_irr_nuevo <- d16_irr[f_irr >= f_irr[7] & f_irr <= tail(f_irr, 1),
    ]
1877 d16_nuevo <- d16_nuevo[f_nuev >= f_nuev[1] & f_nuev <= f_nuev[
    length(f_nuev) - 413], ]
1878
1879 # Verificaci n
1880 cat(sprintf("d16_bombas: %d | d16_nuevo: %d | d16_irr_bombas: %d |
    d16_irr_nuevo: %d\n",
1881             nrow(d16_bombas), nrow(d16_nuevo), nrow(d16_irr_bombas),
                nrow(d16_irr_nuevo)))
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902 # Guardados a csv
1903
1904
1905 # Guardar bombas
1906 for (nombre in bombas_datasets) {
1907     df <- get(nombre)
1908     write.csv(df, file = file.path("data_csv_modif", paste0(nombre, ".
        csv")), row.names = FALSE)
1909 }
1910
1911
1912 # Guardar nuevos

```

```

1913 for (nombre in nuevo_datasets) {
1914   df <- get(nombre)
1915   write.csv(df, file = file.path("data_csv_modif", paste0(nombre, ".
      csv")), row.names = FALSE)
1916 }
1917 '''
1918
1919
1920 '''{r}
1921 for (nombre in irr_bombas_datasets) {
1922   df <- get(nombre)
1923   write.csv(df, file = file.path("data_csv_modif", paste0(nombre, ".
      csv")), row.names = FALSE)
1924 }
1925 for (nombre in irr_nuevo_datasets) {
1926   df <- get(nombre)
1927   write.csv(df, file = file.path("data_csv_modif", paste0(nombre, ".
      csv")), row.names = FALSE)
1928 }
1929 '''
1930
1931
1932
1933 # Irradiancias completas
1934
1935
1936
1937 '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
1938 # Lista para almacenar los subconjuntos de bombas
1939 lista_bombas <- list()
1940 irr_bombas_datasets <- sub("_bombas", "_irr_bombas", bombas_datasets)
1941
1942 for (nombre in irr_bombas_datasets) {
1943   if (exists(nombre)) {
1944     df <- get(nombre)
1945
1946     # Seleccionar columnas disponibles entre: Fecha, Temperatura,
      Irradiancia_CTBombas
1947     columnas_ok <- intersect(c("Fecha", "Temperatura", "
      Irradiancia_CTBombas"), names(df))
1948     df <- df[, columnas_ok]
1949
1950     # Renombrar irradiancia para unificar nombre
1951     if ("Irradiancia_CTBombas" %in% names(df)) {
1952       names(df)[names(df) == "Irradiancia_CTBombas"] <- "Irradiancia"
1953     }
1954
1955     lista_bombas[[nombre]] <- df
1956   } else {
1957     warning(sprintf("Dataset %s no existe", nombre))
1958   }

```

```

1959 }
1960
1961 # Unir todos los de bombas
1962 irradiancia_bombas <- do.call(rbind, lista_bombas)
1963 ""
1964
1965
1966
1967 ""{r}
1968 write.csv(irradiancia_bombas, file = file.path("data_csv_modif",
1969 paste0("irr_bombas", ".csv")), row.names = FALSE)
1970 ""
1971 ""{r}
1972 # Lista para almacenar los subconjuntos de nuevo
1973 lista_nuevo <- list()
1974 irr_nuevo_datasets <- sub("_nuevo", "_irr_nuevo", nuevo_datasets)
1975
1976 for (nombre in irr_nuevo_datasets) {
1977   if (exists(nombre)) {
1978     df <- get(nombre)
1979
1980     # Seleccionar columnas disponibles entre: Fecha, Temperatura,
1981       Irradiancia_CTNuevo
1982     columnas_ok <- intersect(c("Fecha", "Temperatura", "
1983       Irradiancia_CTNuevo"), names(df))
1984     df <- df[, columnas_ok]
1985
1986     # Renombrar irradiancia para unificar nombre
1987     if ("Irradiancia_CTNuevo" %in% names(df)) {
1988       names(df)[names(df) == "Irradiancia_CTNuevo"] <- "Irradiancia"
1989     }
1990
1991     lista_nuevo[[nombre]] <- df
1992   } else {
1993     warning(sprintf("Dataset %s no existe", nombre))
1994   }
1995 }
1996
1997 # Unir todos los de nuevo
1998 irradiancia_nuevo <- do.call(rbind, lista_nuevo)
1999 ""
2000
2001 ""{r}
2002 write.csv(irradiancia_nuevo, file = file.path("data_csv_modif", paste0
2003 ("irr_nuevo", ".csv")), row.names = FALSE)
2004 ""
2005 ""{r}
2006 # Comparar longitudes entre datasets principales e irradiancia

```

```
2006
2007 # Revisar bombas
2008 for (nombre in bombas_datasets) {
2009   nombre_irr <- sub("_bombas", "_irr_bombas", nombre)
2010
2011   if (exists(nombre) && exists(nombre_irr)) {
2012     df <- get(nombre)
2013     df_irr <- get(nombre_irr)
2014
2015     if (nrow(df) != nrow(df_irr)) {
2016       cat(sprintf("      %s y %s tienen distinta longitud: %d vs %d\n",
2017                 nombre, nombre_irr, nrow(df), nrow(df_irr)))
2018     }
2019   } else {
2020     warning(sprintf("Falta dataset: %s o %s", nombre, nombre_irr))
2021   }
2022 }
2023
2024 # Revisar nuevo
2025 for (nombre in nuevo_datasets) {
2026   nombre_irr <- sub("_nuevo", "_irr_nuevo", nombre)
2027
2028   if (exists(nombre) && exists(nombre_irr)) {
2029     df <- get(nombre)
2030     df_irr <- get(nombre_irr)
2031
2032     if (nrow(df) != nrow(df_irr)) {
2033       cat(sprintf("      %s y %s tienen distinta longitud: %d vs %d\n",
2034                 nombre, nombre_irr, nrow(df), nrow(df_irr)))
2035     }
2036   } else {
2037     warning(sprintf("Falta dataset: %s o %s", nombre, nombre_irr))
2038   }
2039 }
2040
2041 ' ' '

```

Listing C.16: Procesado modificado

C.0.17. Análisis completo modificado

Un primer análisis completo con todos los datos modificados.

```
1 ---
2 title: "Análisis completo"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"
5 output: pdf_document
6 ---
7

```

```

8  '{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
9  archivos_csv <- list.files("data_csv_modif", pattern = "\\*.csv$", full
   .names = TRUE)
10
11  for (archivo in archivos_csv) {
12    nombre_objeto <- tools::file_path_sans_ext(basename(archivo))
13
14    df <- read.csv(archivo, stringsAsFactors = FALSE)
15
16    if ("Fecha" %in% names(df)) {
17      df$Fecha <- as.POSIXct(df$Fecha, tz = "UTC")
18    }
19
20    assign(nombre_objeto, df, envir = .GlobalEnv)
21  }
22  rm(df)
23  ''
24
25  '{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
26  # Selecciona por posici n y luego limpia los nombres duplicados
27  d3_bombas <- d3_bombas[, 1:167]
28
29  # Si quieres conservar los nombres originales sin los ...n
30  clean_names <- sub("\\.\\.\\.\\.\\. [0-9]+$", "", colnames(d3_bombas))
31  colnames(d3_bombas) <- clean_names
32  ''
33
34
35  # Estudio de las variables tras las modificaciones
36
37
38  ## CT Bombas
39  '{r, echo=FALSE, message=FALSE, warnings=FALSE}
40  # Lista de nombres de datasets v lidos
41  bombas_datasets <- paste0("d", setdiff(1:16, c(4, 5, 11)), "_bombas")
42
43  # Funci n para analizar cada dataset
44  analiza_varianza <- function(nombre) {
45    df <- get(nombre)[, -1] # Excluye la columna de fecha
46    total <- ncol(df)
47    con_na <- sapply(df, function(x) any(is.na(x)))
48    sin_na <- df[, !con_na, drop = FALSE]
49
50    varianzas <- sapply(sin_na, var)
51    mayores_0 <- sum(varianzas > 0)
52    iguales_0 <- sum(varianzas == 0)
53    con_na_count <- sum(con_na)
54
55    data.frame(
56      Dataset = nombre,
57      Total = total,

```

```
58   'Var > 0' = mayores_0,
59   'Var = 0' = iguales_0,
60   'Con NA' = con_na_count,
61   check.names = FALSE
62 )
63 }
64
65 # Aplica la funci n y combina resultados
66 tabla_bombas <- do.call(rbind, lapply(bombas_datasets,
67   analiza_varianza))
68
69 '''
70
71
72 ## CT Nuevo
73 '{r, echo=FALSE, message=FALSE, warnings=FALSE}'
74 # Lista de nombres de datasets v lidos
75 nuevo_datasets <- paste0("d", setdiff(1:16, c(2, 3, 15)), "_nuevo")
76
77 # Reutiliza la misma funci n de antes
78 tabla_nuevo <- do.call(rbind, lapply(nuevo_datasets, analiza_varianza)
79 )
80
81
82
83
84 # Estudio de variables comunes en bombas
85
86 '{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}'
87 # Lista de reemplazos original
88 replacements <- c(
89   "Vrms.ph.n"      = "Tensi n",
90   "VA.full.cl sico" = "Potencia.Aparente",
91   "VAR.Cl s ."     = "Potencia.Reactiva",
92   "PF.Cl sico"     = "Factor.de.Potencia",
93   "DPF.Cl sico"    = "Cos.Phi"
94   # "L2" = "L1"
95 )
96
97 # Ordenar los reemplazos por longitud del patr n (clave), de mayor a
98   menor
99 replacements <- replacements[order(nchar(names(replacements)),
100   decreasing = TRUE)]
101
102 # Aplicar los reemplazos correctamente
103 for (df_name in bombas_datasets) {
104   df <- get(df_name)
105
106   new_names <- names(df)
```

```

105   for (old in names(replacements)) {
106     new_names <- gsub(old, replacements[[old]], new_names, fixed =
      TRUE)
107   }
108
109   names(df) <- new_names
110
111   #df <- df[ , !grepl("N", names(df))]
112
113   assign(df_name, df)
114 }
115 '''
116
117 '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
118 #neutro_pattern <- "N\\.|NG"
119 #names(d14_bombas[ 1, grepl("N" , names(d14_bombas))])
120 '''
121
122
123
124
125 ## Nombres comunes
126
127 '''{r, echo=FALSE, message=FALSE, warnings=FALSE}
128 # Obtener solo los nombres de columnas (sin la primera, que es la
      fecha)
129 prueba <- bombas_datasets
130 columnas_sin_fecha <- lapply(prueba, function(nombre) {
131   df <- get(nombre)
132   colnames(df)[-1] # Excluir primera columna
133 })
134
135 # Intersección de todas las listas de nombres
136 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)
137
138 # Resultado
139 cat("N mero de variables comunes: ", length(comunes_bombas), "\n\n")
140 '''
141
142
143
144 ## Cuales son las variables que faltan y en que dataset
145
146 '''{r, echo=FALSE, message=FALSE, warnings=FALSE}
147 comunes_bombas <- names(d1_bombas[,-1])
148 # Recorremos todos los dataframes de bombas
149 for (df_name in bombas_datasets) {
150   df <- get(df_name)
151
152   # Excluimos la primera columna (suponemos que es 'fecha' o similar)
153   cols_sin_fecha <- colnames(df)[-1]

```

```
154
155 # Comparamos con las comunes
156 faltantes <- setdiff(comunes_bombas, cols_sin_fecha)
157
158 # Mostramos resultados
159 cat("Variables que faltan en", df_name, ":\n")
160 print(faltantes)
161 cat("Total faltantes:", length(faltantes), "\n\n")
162 }
163
164 '''
165
166 '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
167 # Obtener solo los nombres de columnas (sin la primera, que es la
168   fecha)
169 prueba <- bombas_datasets
170 columnas_sin_fecha <- lapply(prueba, function(nombre) {
171   df <- get(nombre)
172   colnames(df)[-1] # Excluir primera columna
173 })
174
175 # Intersección de todas las listas de nombres
176 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)
177
178 '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
179 nombres_bombas <- c("Fecha", comunes_bombas)
180
181
182
183
184
185
186
187
188 '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
189 # Lista para almacenar los subconjuntos
190 lista_bombas <- list()
191
192 # Recorrer y extraer solo columnas de interés
193 for (nombre in bombas_datasets) {
194   if (exists(nombre)) {
195     df <- get(nombre)
196     comunes <- intersect(nombres_bombas, names(df)) # Por si falta
197       alguna
198     lista_bombas[[nombre]] <- df[, comunes, drop = FALSE]
199   } else {
200     warning(sprintf("Dataset %s no existe", nombre))
201   }
202 }
```

```

203 # Unir todos en uno solo
204 CT_bombas <- do.call(rbind, lista_bombas)
205
206 ```
207
208
209
210 ```{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
211 #write.csv(CT_bombas, file = file.path("data_csv_modif", paste0("
    CT_bombas", ".csv")), row.names = FALSE)
212 ```
213
214
215
216
217
218
219
220
221
222
223
224
225
226 # Estudio de variables comunes en nuevo
227
228 ```{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
229 # Lista de reemplazos original
230 replacements <- c(
231   "Vrms.ph.n"      = "Tensi n",
232   "VA.full.cl sico" = "Potencia.Aparente",
233   "VAR.Cl s."      = "Potencia.Reactiva",
234   "PF.Cl sico"     = "Factor.de.Potencia",
235   "DPF.Cl sico"    = "Cos.Phi"
236   # "L2" = "L1"
237 )
238
239 # Ordenar los reemplazos por longitud del patr n (clave), de mayor a
    menor
240 replacements <- replacements[order(nchar(names(replacements)),
    decreasing = TRUE)]
241
242 # Aplicar los reemplazos correctamente
243 for (df_name in nuevo_datasets) {
244   df <- get(df_name)
245
246   new_names <- names(df)
247   for (old in names(replacements)) {
248     new_names <- gsub(old, replacements[[old]], new_names, fixed =
        TRUE)
249   }

```

```

250
251   names(df) <- new_names
252
253   #df <- df[ , !grepl("N", names(df))]
254
255   assign(df_name, df)
256 }
257 '""
258
259 ## Nombres comunes
260
261 '""{r, echo=FALSE, message=FALSE, warnings=FALSE}
262 # Obtener solo los nombres de columnas (sin la primera, que es la
    fecha)
263 prueba <- nuevo_datasets[c(6,8,9,10,11,12)]
264 columnas_sin_fecha <- lapply(prueba, function(nombre) {
265   df <- get(nombre)
266   colnames(df)[-1] # Excluir primera columna
267 })
268
269 # Intersección de todas las listas de nombres
270 comunes_nuevo <- Reduce(intersect, columnas_sin_fecha)
271
272 # Resultado
273 cat("Número de variables comunes sin d5: ", length(comunes_nuevo), "\n\n")
274 '""
275
276 ## Cuáles son las variables que faltan y en qué dataset
277
278 '""{r, echo=FALSE, message=FALSE, warnings=FALSE}
279 comunes_nuevo <- names(d14_nuevo[,-1])
280 # Recorremos todos los dataframes de bombas
281 for (df_name in nuevo_datasets) {
282   df <- get(df_name)
283
284   # Excluimos la primera columna (suponemos que es 'fecha' o similar)
285   cols_sin_fecha <- colnames(df)[-1]
286
287   # Comparamos con las comunes
288   faltantes <- setdiff(comunes_nuevo, cols_sin_fecha)
289
290   # Mostramos resultados
291   cat("Variables que faltan en", df_name, ":\n")
292   print(faltantes)
293   cat("Total faltantes:", length(faltantes), "\n\n")
294 }
295
296 '""
297
298 '""{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}

```

```

299 # Obtener solo los nombres de columnas (sin la primera, que es la
      fecha)
300 prueba <- nuevo_datasets
301 columnas_sin_fecha <- lapply(prueba, function(nombre) {
302   df <- get(nombre)
303   colnames(df)[-1] # Excluir primera columna
304 })
305
306 # Intersección de todas las listas de nombres
307 comunes_nuevo <- Reduce(intersect, columnas_sin_fecha)
308
309 # Resultado
310 cat("Número de variables comunes sin d5: ", length(comunes_nuevo), "\n\n")
311
312
313
314 ```{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
315 nombres_nuevo <- c("Fecha", comunes_nuevo)
316 ```
317
318
319
320
321
322
323
324
325
326 ```{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
327 # Lista para almacenar los subconjuntos
328 lista_nuevo <- list()
329
330 # Recorrer y extraer solo columnas de interés
331 for (nombre in nuevo_datasets) {
332   if (exists(nombre)) {
333     df <- get(nombre)
334     comunes <- intersect(nombres_nuevo, names(df)) # Por si falta alguna
335     lista_nuevo[[nombre]] <- df[, comunes, drop = FALSE]
336   } else {
337     warning(sprintf("Dataset %s no existe", nombre))
338   }
339 }
340
341 # Unir todos en uno solo
342 CT_nuevo <- do.call(rbind, lista_nuevo)
343
344 ```
345
346

```

```

347
348 ‘‘‘{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
349 #write.csv(CT_nuevo, file = file.path("data_csv_modif", paste0("
      CT_nuevo", ".csv")), row.names = FALSE)
350 ‘‘‘
351
352
353
354
355
356 # Analisis de nombres comunes entre ambos datasets
357
358 ‘‘‘{r, echo=FALSE, message=FALSE, warnings=FALSE}
359 diferencia <- setdiff(nombres_bombas, nombres_nuevo)
360
361 # Resultados
362 cat("Numero variables bombas: ", length(nombres_bombas), "\n\n")
363 cat("Numero variables nuevo: ", length(nombres_nuevo), "\n\n")
364 cat("Variables que no estan en ambos CT:\n")
365 print(diferencia)
366 cat("\nTotal faltantes:", length(diferencia), "\n\n")
367 ‘‘‘

```

Listing C.17: Análisis completo modificado

C.0.18. Análisis problemas

Análisis de problemas encontrados.

```

1 ---
2 title: "Analizando problemas"
3 author: "Ismael Carbajo"
4 date: "r Sys.Date()"
5 output: pdf_document
6 ---
7
8 ‘‘‘{r , warning=FALSE, message=FALSE, echo=FALSE}
9 library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(ggplot2)
15 library(data.table)
16 library(car)
17 library(mgcv)
18 library(purrr)
19 ‘‘‘
20
21 ‘‘‘{r, echo=FALSE, results='hide', message=FALSE, warnings=FALSE}

```

```

22 archivos_csv <- list.files("data_csv_modif", pattern = "\\\\.csv$", full
    .names = TRUE)
23
24 for (archivo in archivos_csv) {
25     nombre_objeto <- tools::file_path_sans_ext(basename(archivo))
26
27     df <- read.csv(archivo, stringsAsFactors = FALSE)
28
29     if ("Fecha" %in% names(df)) {
30         df$Fecha <- as.POSIXct(df$Fecha, tz = "UTC")
31     }
32
33     assign(nombre_objeto, df, envir = .GlobalEnv)
34 }
35 rm(df)
36 '''
37
38 '''{r, echo=FALSE, results='hide', message=FALSE, warnings=FALSE}
39 # Selecciona por posici n y luego limpia los nombres duplicados
40 d3_bombas <- d3_bombas[, 1:167]
41
42 # Si quieres conservar los nombres originales sin los ...n
43 clean_names <- sub("\\\\.\\.\\.\\.\\. [0-9]+$", "", colnames(d3_bombas))
44 colnames(d3_bombas) <- clean_names
45 '''
46
47
48
49
50
51
52
53 # Por qu hay tantas variables con var=0 de manera individual pero
    no en conjunto?
54
55
56 ## CT Bombas
57 '''{r, echo=FALSE, message=FALSE, warnings=FALSE}
58 # Lista de nombres de datasets v lidos
59 bombas_datasets <- paste0("d", setdiff(1:16, c(4, 5, 11)), "_bombas")
60
61 # Funci n para analizar cada dataset
62 analiza_varianza <- function(nombre) {
63     df <- get(nombre)[, -1] # Excluye la columna de fecha
64     total <- ncol(df)
65     con_na <- sapply(df, function(x) any(is.na(x)))
66     sin_na <- df[, !con_na, drop = FALSE]
67
68     varianzas <- sapply(sin_na, var)
69     mayores_0 <- sum(varianzas > 0)
70     iguales_0 <- sum(varianzas == 0)

```

```

71   con_na_count <- sum(con_na)
72
73   data.frame(
74     Dataset = nombre,
75     Total = total,
76     'Var > 0' = mayores_0,
77     'Var = 0' = iguales_0,
78     'Con NA' = con_na_count,
79     check.names = FALSE
80   )
81 }
82
83 # Aplica la funci n y combina resultados
84 tabla_bombas <- do.call(rbind, lapply(bombas_datasets,
85   analiza_varianza))
86
87   ' ' '
88
89
90
91
92   ' ' '{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
93 # Lista de reemplazos original
94 replacements <- c(
95   "Vrms.ph.n"      = "Tensi n",
96   "VA.full.cl sico" = "Potencia.Aparente",
97   "VAR.Cl s ."     = "Potencia.Reactiva",
98   "PF.Cl sico"     = "Factor.de.Potencia",
99   "DPF.Cl sico"    = "Cos.Phi"
100   # "L2" = "L1"
101 )
102
103 # Ordenar los reemplazos por longitud del patr n (clave), de mayor a
104   menor
105 replacements <- replacements[order(nchar(names(replacements)),
106   decreasing = TRUE)]
107
108 # Aplicar los reemplazos correctamente
109 for (df_name in bombas_datasets) {
110   df <- get(df_name)
111
112   new_names <- names(df)
113   for (old in names(replacements)) {
114     new_names <- gsub(old, replacements[[old]], new_names, fixed =
115       TRUE)
116   }
117
118   names(df) <- new_names
119
120   #df <- df[ , !grepl("N", names(df))]

```

```

118
119   assign(df_name, df)
120 }
121 '''
122
123   '{r, echo=FALSE, message=FALSE, warnings=FALSE}
124 # Obtener solo los nombres de columnas (sin la primera, que es la
    fecha)
125 prueba <- bombas_datasets
126 columnas_sin_fecha <- lapply(prueba, function(nombre) {
127   df <- get(nombre)
128   colnames(df)[-1] # Excluir primera columna
129 })
130
131 # Intersección de todas las listas de nombres
132 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)
133
134 # Resultado
135 cat("N mero de variables comunes: ", length(comunes_bombas), "\n\n")
136 '''
137
138   '{r, echo=FALSE, message=FALSE, warnings=FALSE}
139 comunes_bombas <- names(d1_bombas[,-1])
140 # Recorremos todos los dataframes de bombas
141 for (df_name in bombas_datasets) {
142   df <- get(df_name)
143
144   # Excluimos la primera columna (suponemos que es 'fecha' o similar)
145   cols_sin_fecha <- colnames(df)[-1]
146
147   # Comparamos con las comunes
148   faltantes <- setdiff(comunes_bombas, cols_sin_fecha)
149
150   # Mostramos resultados
151   cat("Variables que faltan en", df_name, ":\n")
152   print(faltantes)
153   cat("Total faltantes:", length(faltantes), "\n\n")
154 }
155
156 '''
157
158   '{r, echo=FALSE, results='hide', message=FALSE, warnings=FALSE}
159 # Obtener solo los nombres de columnas (sin la primera, que es la
    fecha)
160 prueba <- bombas_datasets
161 columnas_sin_fecha <- lapply(prueba, function(nombre) {
162   df <- get(nombre)
163   colnames(df)[-1] # Excluir primera columna
164 })
165
166 comunes_bombas <- Reduce(intersect, columnas_sin_fecha)

```

```

167  '''
168
169  '''{r, echo=FALSE, results=FALSE, message=FALSE, warnings=FALSE}
170  nombres_bombas <- c("Fecha", comunes_bombas)
171  nombres_bombas_sinN <- nombres_bombas[!grepl("N\\.|NG", nombres_bombas
172    )]
173  '''
174
175  '''{r}
176  # Lista de nombres de datasets v lidos
177  bombas_datasets <- paste0("d", setdiff(1:16, c(4, 5, 11)), "_bombas")
178
179  # Funci n modificada
180  analiza_varianza_filtrada <- function(nombre) {
181    df <- get(nombre)[, -1] # Excluye la columna de fecha
182
183    # Filtra solo las variables de inter s
184    df <- df[, intersect(colnames(df), nombres_bombas_sinN), drop =
185      FALSE]
186
187    total <- ncol(df)
188    con_na <- sapply(df, function(x) any(is.na(x)))
189    sin_na <- df[, !con_na, drop = FALSE]
190
191    varianzas <- sapply(sin_na, var)
192    mayores_0 <- sum(varianzas > 0)
193    iguales_0 <- sum(varianzas == 0)
194    con_na_count <- sum(con_na)
195
196    data.frame(
197      Dataset = nombre,
198      Total = total,
199      'Var > 0' = mayores_0,
200      'Var = 0' = iguales_0,
201      'Con NA' = con_na_count,
202      check.names = FALSE
203    )
204  }
205
206  # Aplica la funci n y combina resultados
207  tabla_bombas_filtrada <- do.call(rbind, lapply(bombas_datasets,
208    analiza_varianza_filtrada))
209  tabla_bombas_filtrada
210  '''
211
212  '''{r, echo=FALSE}
213  # Lista para guardar resultados
214  vars_var0_por_dataset <- list()

```

```

215
216 # Itera sobre cada dataset
217 for (nombre in bombas_datasets) {
218   df <- get(nombre)[, -1] # Excluye la columna de fecha si es la
      primera
219   df <- df[, intersect(colnames(df), nombres_bombas_sinN), drop =
      FALSE]
220
221   # Elimina columnas con NA para evitar errores en var()
222   df <- df[, colSums(is.na(df)) == 0, drop = FALSE]
223
224   # Calcular varianzas
225   varianzas <- sapply(df, var)
226
227   # Extraer nombres de variables con varianza == 0
228   vars_0 <- names(varianzas[varianzas == 0])
229
230   # Guardar en la lista
231   vars_var0_por_dataset[[nombre]] <- vars_0
232 }
233
234 # Puedes ver el resultado para un dataset espec fico as :
235 #vars_var0_por_dataset[["d1_bombas"]]
236
237 # O todos con un resumen
238 #lapply(vars_var0_por_dataset, length) # cu ntas vars con var=0 por
      dataset
239 vars_var0_por_dataset
240 ' '
241
242
243
244
245 '{r, echo=FALSE}
246 # Paso 1: recolectar los valores constantes (tablas de var=0)
247 valores_constantes <- list()
248
249 for (nombre in bombas_datasets) {
250   df <- get(nombre)[, -1]
251   df <- df[, intersect(colnames(df), nombres_bombas_sinN), drop =
      FALSE]
252   df <- df[, colSums(is.na(df)) == 0, drop = FALSE]
253   varianzas <- map_dbl(df, ~ var(.x, na.rm = TRUE))
254   vars_0 <- names(varianzas[varianzas == 0])
255
256   if (length(vars_0) > 0) {
257     valores <- sapply(vars_0, function(var) unique(df[[var]]),
      simplify = TRUE)
258     valores_constantes[[nombre]] <- valores
259   }
260 }

```

```

261 |
262 | # Paso 2: obtener todas las variables nicas con varianza 0 en al
      | menos un dataset
263 | todas_las_vars <- unique(unlist(lapply(valores_constantes, names)))
264 |
265 | # Paso 3: construir data.frame final
266 | tabla_resultado <- data.frame(variable = todas_las_vars,
      | stringsAsFactors = FALSE)
267 |
268 | for (nombre in bombas_datasets) {
269 |   columna <- sapply(todas_las_vars, function(var) {
270 |     if (!is.null(valores_constantes[[nombre]]) && var %in% names(
      | valores_constantes[[nombre]])) {
271 |       return(valores_constantes[[nombre]][[var]])
272 |     } else {
273 |       return(NA)
274 |     }
275 |   })
276 |   tabla_resultado[[nombre]] <- columna
277 | }
278 |
279 | # Mostrar la tabla
280 | tabla_resultado
281 | '""
282 |
283 |
284 | '""{r, echo=FALSE}
285 | # Datasets objetivo
286 | datasets_objetivo <- c("d2_bombas", "d3_bombas", "d6_bombas", "
      | d7_bombas")
287 |
288 | # Variables con varianza 0 en d8_bombas
289 | vars_d8 <- vars_var0_por_dataset[["d8_bombas"]]
290 |
291 | # Vector para guardar resultados
292 | resumen_vector <- c()
293 |
294 | # Loop
295 | for (dataset in datasets_objetivo) {
296 |   df <- get(dataset)
297 |
298 |   for (var in vars_d8) {
299 |     if (var %in% names(df)) {
300 |       n_valores <- length(unique(df[[var]]))
301 |       resumen_vector <- c(resumen_vector,
302 |                           paste0(var, " en ", dataset, ": ", n_valores
      | , " valores distintos"))
303 |     } else {
304 |       resumen_vector <- c(resumen_vector,
305 |                           paste0(var, " en ", dataset, ": NO PRESENTE"
      | ))

```

```

306     }
307   }
308 }
309
310 # Ver resultado
311 resumen_vector
312
313
314 '''
315
316 Viendo alguna, se ve que no son valores cercanos a 0 (que es el valor
317   que toma en los datasets que su var=0).
318 '''{r}
319 summary(d7_bombas$Armnicos.Corriente11.L1.Min)
320
321
322
323
324
325 # Comparaci n explicita de modelos Desequilibrio.vz.min (mejor
326   variable que aparece en ambos tops de GAM)
327
328 '''{r}
329 CT_nuevo_orig <- read.csv("data_csv_modif/CT_nuevo.csv",
330   stringsAsFactors = FALSE)
331 CT_nuevo_orig$Fecha <- as.POSIXct(CT_nuevo_orig$Fecha, tz = "UTC")
332
333 CT_bombas_orig <- read.csv("data_csv_modif/CT_bombas.csv",
334   stringsAsFactors = FALSE)
335 CT_bombas_orig$Fecha <- as.POSIXct(CT_bombas_orig$Fecha, tz = "UTC")
336
337 irr_bombas <- read.csv("data_csv_modif/irr_bombas.csv",
338   stringsAsFactors = FALSE)
339 irr_bombas$Fecha <- as.POSIXct(irr_bombas$Fecha, tz = "UTC")
340
341 irr_nuevo <- read.csv("data_csv_modif/irr_nuevo.csv", stringsAsFactors
342   = FALSE)
343 irr_nuevo$Fecha <- as.POSIXct(irr_nuevo$Fecha, tz = "UTC")
344
345 '''
346
347 '''{r}
348 # ---- Para CT_bombas ----
349
350 # Detectar columnas con varianza 0
351 varianzas_bombas <- sapply(CT_bombas_orig, function(x) if (is.numeric(
352   x)) var(x, na.rm = TRUE) else NA)
353 cols_var0_bombas <- names(varianzas_bombas[!is.na(varianzas_bombas) &
354   varianzas_bombas == 0])
355
356 # Quitar variables con "N." o "NG"

```

```

349 CT_bombas <- CT_bombas_orig[ , !grepl("N\\.|NG", names(CT_bombas_orig)
    )]
350
351
352 # ---- Para CT_nuevo ----
353
354 # Detectar columnas con varianza 0
355 varianzas_nuevo <- sapply(CT_nuevo_orig, function(x) if (is.numeric(x)
    ) var(x, na.rm = TRUE) else NA)
356 cols_var0_nuevo <- names(varianzas_nuevo[!is.na(varianzas_nuevo) &
    varianzas_nuevo == 0])
357
358 # Crear CT_nuevo_var>0
359 CT_nuevo <- CT_nuevo_orig[ , !(names(CT_nuevo_orig) %in%
    cols_var0_nuevo)]
360
361 '''
362
363 ```{r, echo=FALSE}
364 bombas_modelos<-cbind(irr_bombas[, -1], CT_bombas[, -1])
365 nuevo_modelos<-cbind(irr_nuevo[, -1], CT_nuevo[, -1])
366 '''
367
368 ```{r}
369 modelo_bombas <- gam(Desequilibrio.Vz.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = bombas_modelos)
370 summary(modelo_bombas)
371 plot(modelo_bombas)
372 '''
373
374 ```{r, echo=FALSE, warning=FALSE}
375 vis.gam(modelo_bombas, view = c("Irradiancia", "Temperatura"), plot.
    type = "persp")
376 vis.gam(modelo_bombas, view = c("Irradiancia", "Temperatura"), plot.
    type = "contour")
377 '''
378
379
380 ```{r}
381 modelo_nuevo <- gam(Desequilibrio.Vz.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
382 summary(modelo_nuevo)
383 plot(modelo_nuevo)
384 '''
385
386
387 ```{r, echo=FALSE, warning=FALSE}
388 vis.gam(modelo_nuevo, view = c("Irradiancia", "Temperatura"), plot.
    type = "persp")
389 vis.gam(modelo_nuevo, view = c("Irradiancia", "Temperatura"), plot.
    type = "contour")

```

```

390  ' ' '
391
392
393
394
395
396
397
398
399
400
401 # Comparaci n explicita de nuevo pero ahora a adiando la variable
    meses
402
403  '{r}
404 CT_nuevo_orig <- read.csv("data_csv_modif/CT_nuevo.csv",
    stringsAsFactors = FALSE)
405 CT_nuevo_orig$Fecha <- as.POSIXct(CT_nuevo_orig$Fecha, tz = "UTC")
406
407 CT_bombas_orig <- read.csv("data_csv_modif/CT_bombas.csv",
    stringsAsFactors = FALSE)
408 CT_bombas_orig$Fecha <- as.POSIXct(CT_bombas_orig$Fecha, tz = "UTC")
409
410 irr_bombas <- read.csv("data_csv_modif/irr_bombas.csv",
    stringsAsFactors = FALSE)
411 irr_bombas$Fecha <- as.POSIXct(irr_bombas$Fecha, tz = "UTC")
412
413 irr_nuevo <- read.csv("data_csv_modif/irr_nuevo.csv", stringsAsFactors
    = FALSE)
414 irr_nuevo$Fecha <- as.POSIXct(irr_nuevo$Fecha, tz = "UTC")
415  ' ' '
416
417  '{r}
418 # ---- Para CT_bombas ----
419
420 # Detectar columnas con varianza 0
421 varianzas_bombas <- sapply(CT_bombas_orig, function(x) if (is.numeric(
    x)) var(x, na.rm = TRUE) else NA)
422 cols_var0_bombas <- names(varianzas_bombas[!is.na(varianzas_bombas) &
    varianzas_bombas == 0])
423
424 # Quitar variables con "N." o "NG"
425 CT_bombas <- CT_bombas_orig[ , !grepl("N\\.|NG", names(CT_bombas_orig)
    )]
426
427
428 # ---- Para CT_nuevo ----
429
430 # Detectar columnas con varianza 0
431 varianzas_nuevo <- sapply(CT_nuevo_orig, function(x) if (is.numeric(x)
    ) var(x, na.rm = TRUE) else NA)

```

```

432 cols_var0_nuevo <- names(varianzas_nuevo[!is.na(varianzas_nuevo) &
    varianzas_nuevo == 0])
433
434 # Crear CT_nuevo_var>0
435 CT_nuevo <- CT_nuevo_orig[ , !(names(CT_nuevo_orig) %in%
    cols_var0_nuevo)]
436
437 '''
438
439 ```{r, echo=FALSE}
440 bombas_modelos<-cbind(irr_bombas[,-1],CT_bombas[,-1])
441 nuevo_modelos<-cbind(irr_nuevo[,-1],CT_nuevo[,-1])
442 ```
443
444 ```{r}
445 modelo_bombas <- gam(Desequilibrio.Vz.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = bombas_modelos)
446 summary(modelo_bombas)
447 plot(modelo_bombas)
448 ```
449
450 ```{r, echo=FALSE, warning=FALSE}
451 vis.gam(modelo_bombas, view = c("Irradiancia", "Temperatura"), plot.
    type = "persp")
452 vis.gam(modelo_bombas, view = c("Irradiancia", "Temperatura"), plot.
    type = "contour")
453 ```
454
455
456 ```{r}
457 modelo_nuevo <- gam(Desequilibrio.Vz.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
458 summary(modelo_nuevo)
459 plot(modelo_nuevo)
460 ```
461
462
463 ```{r, echo=FALSE, warning=FALSE}
464 vis.gam(modelo_nuevo, view = c("Irradiancia", "Temperatura"), plot.
    type = "persp")
465 vis.gam(modelo_nuevo, view = c("Irradiancia", "Temperatura"), plot.
    type = "contour")
466 ```
467
468
469
470 # Comparacion ahora de sus modelos lineales
471 ```{r, echo=FALSE}
472 modelo <- lm(Desequilibrio.Vn.Min ~ Irradiancia * Temperatura, data =
    nuevo_modelos)
473 summary(modelo)

```

```

474 plot(modelo)
475 ' ' '
476
477
478 '{r, echo=FALSE}
479 modelo <- lm(Desequilibrio.Vn.Min ~ Irradiancia * Temperatura, data =
      bombas_modelos)
480 summary(modelo)
481 plot(modelo)
482 ' ' '

```

Listing C.18: Análisis problemas

C.0.19. Análisis final

Análisis final de todos los datos.

```

1  ---
2  title: "Análisis completo"
3  author: "Ismael Carbajo"
4  date: "r Sys.Date()"
5  output: pdf_document
6  ---
7
8  '{r , warning=FALSE, message=FALSE, echo=FALSE}
9  library(readxl)
10 library(factoextra)
11 library(dplyr)
12 library(knitr)
13 library(lubridate)
14 library(ggplot2)
15 library(data.table)
16 library(car)
17 library(mgcv)
18 ' ' '
19
20 '{r, echo=FALSE}
21 CT_nuevo_orig <- read.csv("data_csv_modif/CT_nuevo.csv",
      stringsAsFactors = FALSE)
22 CT_nuevo_orig$Fecha <- as.POSIXct(CT_nuevo_orig$Fecha, tz = "UTC")
23
24 CT_bombas_orig <- read.csv("data_csv_modif/CT_bombas.csv",
      stringsAsFactors = FALSE)
25 CT_bombas_orig$Fecha <- as.POSIXct(CT_bombas_orig$Fecha, tz = "UTC")
26
27 irr_bombas <- read.csv("data_csv_modif/irr_bombas.csv",
      stringsAsFactors = FALSE)
28 irr_bombas$Fecha <- as.POSIXct(irr_bombas$Fecha, tz = "UTC")
29

```

```

30 irr_nuevo <- read.csv("data_csv_modif/irr_nuevo.csv", stringsAsFactors
    = FALSE)
31 irr_nuevo$Fecha <- as.POSIXct(irr_nuevo$Fecha, tz = "UTC")
32
33 prueba<-read.csv("data_csv_modif/d1_nuevo.csv", stringsAsFactors =
    FALSE)
34 '''
35
36 Primero, se quitan variables con varianza = 0, parece que al juntar
    los datos, casi todas las variables tienen varianza.
37 Tambi n se quitan las variables de neutro de CT bombas.
38 '''{r}
39 # ---- Para CT_bombas ----
40
41 # Detectar columnas con varianza 0
42 varianzas_bombas <- sapply(CT_bombas_orig, function(x) if (is.numeric(
    x)) var(x, na.rm = TRUE) else NA)
43 cols_var0_bombas <- names(varianzas_bombas[!is.na(varianzas_bombas) &
    varianzas_bombas == 0])
44
45 # Quitar variables con "N." o "NG"
46 CT_bombas <- CT_bombas_orig[ , !grepl("N\\.|NG", names(CT_bombas_orig)
    )]
47
48
49 # ---- Para CT_nuevo ----
50
51 # Detectar columnas con varianza 0
52 varianzas_nuevo <- sapply(CT_nuevo_orig, function(x) if (is.numeric(x)
    ) var(x, na.rm = TRUE) else NA)
53 cols_var0_nuevo <- names(varianzas_nuevo[!is.na(varianzas_nuevo) &
    varianzas_nuevo == 0])
54
55 # Crear CT_nuevo_var>0
56 CT_nuevo <- CT_nuevo_orig[ , !(names(CT_nuevo_orig) %in%
    cols_var0_nuevo)]
57
58 '''
59
60
61
62 # Analisis para bombas (sin neutro)
63
64
65 ## Correlaciones
66 Al parecer ahora hay pocas variables con correlacion alta cuando se
    considera todo el conjunto.
67 '''{r, echo=FALSE}
68 correlaciones <- sapply(CT_bombas[,-1], function(x) cor(x,
    irr_bombas$Irradiancia, use = "complete.obs"))
69 corr_irr <- data.frame(top10_correlacion_irr = correlaciones) %>%

```

```

    arrange(desc(abs(top10_correlacion_irr)))
70 head(corr_irr,10)
71 correlaciones <- sapply(CT_bombas[,-1], function(x) cor(x,
    irr_bombas$Temperatura, use = "complete.obs"))
72 corr_irr <- data.frame(top10_correlacion_temp = correlaciones) %>%
    arrange(desc(abs(top10_correlacion_temp)))
73 head(corr_irr,10)
74 ""
75
76
77 Correlaci n bastante alta entre temperatura e irradiancia.
78 ""{r, echo=FALSE}
79 cor_irr_temp<-cor(irr_bombas$Irradiancia,irr_bombas$Temperatura,use="
    complete.obs")
80 cor_irr_temp
81 ""
82
83
84 ""{r, echo=FALSE}
85 bombas_modelos<-cbind(irr_bombas[,-1],CT_bombas[,-1])
86 ""
87
88
89
90 ## Modelos lineales
91
92 ""{r, echo=FALSE}
93 bombas_bucle <- bombas_modelos[,-c(1,2)]
94 indicadores_num <- bombas_bucle[sapply(bombas_bucle, is.numeric)]
95
96 irr <- irr_bombas$Irradiancia
97 temp <- irr_bombas$Temperatura
98
99 resultados <- c()
100
101 for (nombre_var in names(indicadores_num)) {
102   y <- indicadores_num[[nombre_var]]
103
104   # Irradiancia
105   mod_irr <- lm(y ~ irr)
106   summ_irr <- summary(mod_irr)
107
108   # Temperatura
109   mod_temp <- lm(y ~ temp)
110   summ_temp <- summary(mod_temp)
111
112   # Multiple
113   mod_mult <- lm(y ~ irr + temp)
114   summ_mult <- summary(mod_mult)
115
116   # Multiple con interaccion

```

```

117   mod_mult_int <- lm(y ~ irr * temp)
118   summ_mult_int <- summary(mod_mult_int)
119
120
121   resultados <- rbind(resultados, data.frame(
122     Indicador = nombre_var,
123     R2Irr = summ_irr$adj.r.squared,
124     pIrr = coef(summ_irr)[2, 4],
125     R2Temp = summ_temp$adj.r.squared,
126     pTemp = coef(summ_temp)[2, 4],
127     R2Mult = summ_mult$adj.r.squared,
128     pIrrMult = coef(summ_mult)["irr", 4],
129     pTempMult = coef(summ_mult)["temp", 4],
130     R2Multint = summ_mult_int$adj.r.squared,
131     pInt = coef(summ_mult_int)["irr:temp", 4]
132   ))
133 }
134
135
136 ‘‘‘
137
138
139 ‘‘{r, echo=FALSE}
140 resultados <- resultados[order(-resultados$R2Multint), ]
141 ‘‘‘
142
143
144 Ahora los resultados no son nada prometedores en los modelos lineales.
145 ‘‘{r, echo=FALSE}
146 knitr::kable(head(resultados[,c(1,2,3,4,5)],10), digits = 4, caption =
147   "Resultados de las regresiones modelos simples")
148 ‘‘‘
149 ‘‘{r, echo=FALSE}
150 knitr::kable(head(resultados[,c(2,3,4,5)],10), digits = 4, caption =
151   "Resultados de las regresiones modelos complejos")
152 ‘‘‘
153
154
155
156
157 ## Modelos GAM
158
159
160 ‘‘{r, echo=FALSE}
161 # resultados_gam <- c()
162 #
163 # for (nombre_var in names(indicadores_num)) {
164 #   y <- indicadores_num[[nombre_var]]
165 #

```

```

166 # gam_mod <- gam(y ~ s(irr) + s(temp))
167 # gam_mod_int <- gam(y ~ s(irr) + s(temp) + ti(irr, temp))
168 # summ_gam <- summary(gam_mod)
169 # summ_gam_int <- summary(gam_mod_int)
170 #
171 # r2_adj <- summ_gam$r.sq
172 # dev_exp <- summ_gam$dev.expl
173 # gcv_val <- summ_gam$sp.criterion
174 # r2_adj_int <- summ_gam_int$r.sq
175 # dev_exp_int <- summ_gam_int$dev.expl
176 # gcv_val_int <- summ_gam_int$sp.criterion
177 #
178 # resultados_gam <- rbind(resultados_gam, data.frame(
179 #   Indicador = nombre_var,
180 #   R2Adj = round(r2_adj, 4),
181 #   DevExp = round(dev_exp * 100, 2),
182 #   GCV = round(gcv_val, 4),
183 #   R2AdjInt = round(r2_adj_int, 4),
184 #   DevExpInt = round(dev_exp_int * 100, 2),
185 #   GCVInt = round(gcv_val_int, 4)
186 # ))
187 # }
188 ""
189
190 "{r, echo=FALSE}
191 #write.csv(resultados_gam, file = file.path("data_csv_modif", paste0("
192   result_gam_bombas", ".csv")), row.names = FALSE)
193 ""
194
195 "{r, echo=FALSE}
196 resultados_gam<-read.csv(file = file.path("data_csv_modif", paste0("
197   result_gam_bombas", ".csv")))
198 ""
199
200 "{r, echo=FALSE}
201 resultados_gam_con <- resultados_gam[order(-resultados_gam$R2AdjInt),
202   ]
203 head(resultados_gam_con, 10)
204 ""
205
206
207
208
209 # Analisis para nuevo (con neutro)
210
211
212 ## Correlaciones
213 En este casi si que tenemos correlaciones altas.

```

```

214  ““{r, echo=FALSE}
215  correlaciones <- sapply(CT_nuevo[, -1], function(x) cor(x,
    irr_nuevo$Irradiancia, use = "complete.obs"))
216  corr_irr <- data.frame(top10_correlacion_irr = correlaciones) %>%
    arrange(desc(abs(top10_correlacion_irr)))
217  head(corr_irr, 10)
218  correlaciones <- sapply(CT_nuevo[, -1], function(x) cor(x,
    irr_nuevo$Temperatura, use = "complete.obs"))
219  corr_irr <- data.frame(top10_correlacion_temp = correlaciones) %>%
    arrange(desc(abs(top10_correlacion_temp)))
220  head(corr_irr, 10)
221  ““
222
223
224
225
226  Correlaci n entre irradiancia y temperatura m s alta que en bombas
    incluso.
227  ““{r, echo=FALSE}
228  cor_irr_temp<-cor(irr_nuevo$Irradiancia, irr_nuevo$Temperatura, use="
    complete.obs")
229  cor_irr_temp
230  ““
231
232
233  ““{r, echo=FALSE}
234  nuevo_modelos<-cbind(irr_nuevo[, -1], CT_nuevo[, -1])
235  ““
236
237
238
239  ## Modelos lineales
240
241  ““{r, echo=FALSE}
242  nuevo_bucle <- nuevo_modelos[, -c(1,2)]
243  indicadores_num <- nuevo_bucle[sapply(nuevo_bucle, is.numeric)]
244
245  irr <- irr_nuevo$Irradiancia
246  temp <- irr_nuevo$Temperatura
247
248  resultados <- c()
249
250  for (nombre_var in names(indicadores_num)) {
251    y <- indicadores_num[[nombre_var]]
252
253    # Irradiancia
254    mod_irr <- lm(y ~ irr)
255    summ_irr <- summary(mod_irr)
256
257    # Temperatura
258    mod_temp <- lm(y ~ temp)

```

```

259 summ_temp <- summary(mod_temp)
260
261 # Multiple
262 mod_mult <- lm(y ~ irr + temp)
263 summ_mult <- summary(mod_mult)
264
265 # Multiple con interaccion
266 mod_mult_int <- lm(y ~ irr * temp)
267 summ_mult_int <- summary(mod_mult_int)
268
269
270 resultados <- rbind(resultados, data.frame(
271   Indicador = nombre_var,
272   R2Irr = summ_irr$adj.r.squared,
273   pIrr = coef(summ_irr)[2, 4],
274   R2Temp = summ_temp$adj.r.squared,
275   pTemp = coef(summ_temp)[2, 4],
276   R2Mult = summ_mult$adj.r.squared,
277   pIrrMult = coef(summ_mult)["irr", 4],
278   pTempMult = coef(summ_mult)["temp", 4],
279   R2Multint = summ_mult_int$adj.r.squared,
280   pInt = coef(summ_mult_int)["irr:temp", 4]
281 ))
282 }
283
284 resultados <- resultados[order(-resultados$R2Multint), ]
285 ```
286
287 Los resultados ahora si que indican posibles modelos tiles .
288
289 ```{r, echo=FALSE}
290 knitr::kable(head(resultados[,c(1,2,3,4,5)],10), digits = 4, caption =
291   "Resultados de las regresiones modelos simples")
292 ```
293
294 ```{r, echo=FALSE}
295 knitr::kable(head(resultados[,-c(2,3,4,5)],10), digits = 4, caption =
296   "Resultados de las regresiones modelos complejos")
297 ```
298
299
300
301
302
303
304
305 ## Modelos GAM
306
307

```

```

308 | '{r, echo=FALSE}
309 | # resultados_gam <- c()
310 | #
311 | # for (nombre_var in names(indicadores_num)) {
312 | #   y <- indicadores_num[[nombre_var]]
313 | #
314 | #   gam_mod <- gam(y ~ s(irr) + s(temp))
315 | #   gam_mod_int <- gam(y ~ s(irr) + s(temp) + ti(irr, temp))
316 | #   summ_gam <- summary(gam_mod)
317 | #   summ_gam_int <- summary(gam_mod_int)
318 | #
319 | #   r2_adj <- summ_gam$r.sq
320 | #   dev_exp <- summ_gam$dev.expl
321 | #   gcv_val <- summ_gam$sp.criterion
322 | #   r2_adj_int <- summ_gam_int$r.sq
323 | #   dev_exp_int <- summ_gam_int$dev.expl
324 | #   gcv_val_int <- summ_gam_int$sp.criterion
325 | #
326 | #   resultados_gam <- rbind(resultados_gam, data.frame(
327 | #     Indicador = nombre_var,
328 | #     R2Adj = round(r2_adj, 4),
329 | #     DevExp = round(dev_exp * 100, 2),
330 | #     GCV = round(gcv_val, 4),
331 | #     R2AdjInt = round(r2_adj_int, 4),
332 | #     DevExpInt = round(dev_exp_int * 100, 2),
333 | #     GCVInt = round(gcv_val_int, 4)
334 | #   ))
335 | # }
336 | ''
337 |
338 |
339 | '{r, echo=FALSE}
340 | resultados_gam_nuevo<-read.csv(file = file.path("data_csv_modif",
341 | paste0("result_gam_nuevo", ".csv")))
342 | ''
343 |
344 | '{r, echo=FALSE}
345 | #write.csv(resultados_gam_nuevo, file = file.path("data_csv_modif",
346 | paste0("result_gam_nuevo", ".csv")), row.names = FALSE)
347 | ''
348 |
349 | '{r, echo=FALSE}
350 | resultados_gam_con_nuevo <- resultados_gam_nuevo[order(-
351 |   resultados_gam_nuevo$R2AdjInt), ]
352 | head(resultados_gam_con_nuevo, 10)
353 | ''
354 |
355 |

```

```

356
357
358
359
360
361
362
363
364 # Analisis de modelos individuales
365 Se analizan ahora ciertos modelos de manera individual.
366
367
368
369 # Modelos lineales CT bombas
370 Aunque los resultados son bastante malos en comparaci n con el
    estudio inicial de las 3 semanas de enero, parece ser que las
    variables son todas significativas, aunue claro, estamos con una
    gran cantidad de observaciones. Adem s, como ya se hab a visto,
    no se llega ni al 10% de explicaci n.
371
372 ``{r}
373 modelo <- lm(Desequilibrio.Vz.Min ~ Irradiancia * Temperatura, data =
    bombas_modelos)
374 summary(modelo)
375 plot(modelo)
376 ``
377
378 Aun teniendo unos malos resultados, se a ade la variable mes para ver
    que tal se comporta.
379 ``{r}
380 mes_bombas = lubridate::month(CT_bombas$Fecha, label = TRUE, abbr =
    TRUE)
381 ``
382
383
384 Ahora si que se explica bastante m s, y curiosmaente la temperatura
    deja de ser importante, pero no se puede quitar, pues la
    interacci n sigue siendo importante.
385 ``{r}
386 modelo <- lm(Desequilibrio.Vz.Min ~ Irradiancia * Temperatura +
    mes_bombas, data = bombas_modelos)
387 summary(modelo)
388 ``
389
390 # Modelos GAM CT bombas
391
392 Volvemos a tener todo muy significativo, comportamiento curioso, al
    contrario de como pasaba con los dem s modelos GAM del primer
    analisis, ahora parece que temperatura aun teniendo la interacci n
    conserva cierta curvatura.
393 La interacci n tiene un comportamiento complejo, hay que estudiarlo

```

```

    bien.
394
395 ```{r}
396 modelo <- gam(Desequilibrio.Vn.Med ~ s(Irradiancia) + s(Temperatura) +
    ti(Irradiancia, Temperatura), data = bombas_modelos)
397 summary(modelo)
398 plot(modelo)
399 ```
400
401
402
403 Ahora la representacion cambia totalmente, parece que los días muy
    soleados tienen gran implicación en este cambio.
404
405 ```{r, echo=FALSE, warning=FALSE}
406 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
407 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
408 ```
409
410
411 Todos estos siguientes modelos tienen un comportamiento muy parejo al
    primero, tanto en gráficas como en resultados. No pongo los
    resultados para no alargar el pdf.
412
413 Comportamiento parejo.
414
415 ```{r, results='hide', fig.show = 'hide'}
416 modelo <- gam(Desequilibrio.Vz.Med ~ s(Irradiancia) + s(Temperatura) +
    ti(Irradiancia, Temperatura), data = bombas_modelos)
417 summary(modelo)
418 plot(modelo)
419 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
420 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
421 ```
422
423
424 Comportamiento parejo.
425
426 ```{r, results='hide', fig.show = 'hide'}
427 modelo <- gam(Desequilibrio.Vz.Max ~ s(Irradiancia) + s(Temperatura) +
    ti(Irradiancia, Temperatura), data = bombas_modelos)
428 summary(modelo)
429 plot(modelo)
430 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
431 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")

```

```

432  ' '
433
434
435
436 Comportamiento parejo.
437
438 '{r, results='hide', fig.show = 'hide'}
439 modelo <- gam(Frecuencia.Max ~ s(Irradiancia) + s(Temperatura) + ti(
    Irradiancia, Temperatura) + mes_bombas, data = bombas_modelos)
440 summary(modelo)
441 plot(modelo)
442 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
443 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
444  ' '
445
446
447
448 Comportamiento parejo.
449
450
451 '{r, results='hide', fig.show = 'hide'}
452 modelo <- gam(Desequilibrio.Vn.Max ~ s(Irradiancia) + s(Temperatura) +
    ti(Irradiancia, Temperatura), data = bombas_modelos)
453 summary(modelo)
454 plot(modelo)
455 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
456 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
457  ' '
458
459 '{r, results='hide', fig.show = 'hide'}
460 modelo <- gam(Desequilibrio.Vz.Min ~ s(Irradiancia) + s(Temperatura) +
    ti(Irradiancia, Temperatura), data = bombas_modelos)
461 summary(modelo)
462 plot(modelo)
463 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
464 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
465  ' '
466
467 -----A partir de aqui estan los que son como de la otra manera
468
469
470
471 '{r, results='hide', fig.show = 'hide'}
472 modelo <- gam(Corriente.Fundamental.L1.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = bombas_modelos)

```

```

473 summary(modelo)
474 plot(modelo)
475 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
476 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
477 ' ' '
478
479
480 '{r, results='hide', fig.show = 'hide'}
481 modelo <- gam(Corriente.L1.Min ~ s(Irradiancia) + s(Temperatura) + ti(
    Irradiancia, Temperatura), data = bombas_modelos)
482 summary(modelo)
483 plot(modelo)
484 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
485 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
486 ' ' '
487
488
489 '{r, results='hide', fig.show = 'hide'}
490 modelo <- gam(Corriente.Fundamental.L1.Med ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = bombas_modelos)
491 summary(modelo)
492 plot(modelo)
493 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
494 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
495 ' ' '
496
497
498 '{r, results='hide', fig.show = 'hide'}
499 modelo <- gam(Corriente.L1.Med ~ s(Irradiancia) + s(Temperatura) + ti(
    Irradiancia, Temperatura), data = bombas_modelos)
500 summary(modelo)
501 plot(modelo)
502 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    persp")
503 vis.gam(modelo, view = c("Irradiancia", "Temperatura"), plot.type = "
    contour")
504 ' ' '
505
506
507
508 ### Se prueba ahora el modelo a añadiendo la variable mes
509
510 Si se a ade como factor normal sin suavizar, parece que no a ade
    nada til al modelo.
511

```

```

512  "{r}
513 modelo <- gam(Desequilibrio.Vn.Med ~ s(Irradiancia) + s(Temperatura) +
514   ti(Irradiancia, Temperatura) + mes_bombas, data = bombas_modelos)
515  ""
516
517 A adiendo mes como efecto aleatorio parece que tampoco es til para
518 el modelo.
519
520 "{r}
521 modelo <- gam(Desequilibrio.Vn.Med ~ s(Irradiancia) + s(Temperatura) +
522   ti(Irradiancia, Temperatura) + s(mes_bombas, bs = "re"), data =
523   bombas_modelos)
524 summary(modelo)
525 #plot(modelo)
526 ""
527
528 si se mira esto para todos los modelos que se consideran parejos, los
529 resultados son iguales exceptuando Frecuencia.Max:
530 - Si a ades mes sin suavizar el Q es *** el L es . y ^4 es *, pero
531 la mejora que realiza es insignificante.
532 - Si a ades mes suavizado sale como muy significativo, pero la
533 mejora vuelve a ser insignificante.
534
535 En cuanto a insignificante hablamos de una mejora en R cudrado de
536 0.001~0.002
537
538
539
540
541
542
543
544
545
546
547
548
549 # Modelos lineales CT nuevo
550
551
552 ### Desequilibrio.Vn.Min
553
554

```

```

555   '{r, echo=FALSE}
556 modelo <- lm(Desequilibrio.Vn.Min ~ Irradiancia * Temperatura, data =
      nuevo_modelos)
557 summary(modelo)
558 par(mfrow=c(2,2))
559 plot(modelo)
560 par(mfrow=c(1,1))
561 ''
562
563
564   '{r, echo=FALSE}
565 mes_nuevo = lubridate::month(CT_nuevo$Fecha, label = TRUE, abbr = TRUE
      )
566 df_residuos <- data.frame(
567   fitted = modelo$fitted.values,
568   residuals = modelo$residuals,
569   Mes = mes_nuevo
570 )
571 # Graficar residuos coloreados por mes
572 ggplot(df_residuos, aes(x = fitted, y = residuals, color = Mes)) +
573   geom_point() +
574   labs(title = "Residuos del modelo por mes",
575        x = "Fecha",
576        y = "Residuo") +
577   theme_minimal() +
578   scale_color_brewer(palette = "Set3")
579 ''
580
581
582 Se hace ahora el modelo con mes, hay una mejora de 0.46 a 0.49 en el
      Rcuadrado y semejante para deviance explicada. Aunque no aparezcan
      los graficos, los graficos de los errores son similares.
583
584 Tiene sentido tener poca mejora, seg n se ven en el gr fico de
      errores no parece haber uan gran diferencia entre los diferentes
      meses.
585
586   '{r}
587 modelo <- lm(Desequilibrio.Vn.Min ~ Irradiancia * Temperatura +
      mes_nuevo, data = nuevo_modelos)
588 summary(modelo)
589 ''
590
591
592
593
594 ### Tensi n.de.Medio.Ciclo.V.RMS.NG.Med
595
596
597   '{r}
598 modelo <- lm(Tensi n.de.Medio.Ciclo.V.RMS.NG.Med ~ Irradiancia *

```

```

    Temperatura, data = nuevo_modelos)
599 summary(modelo)
600 par(mfrow=c(2,2))
601 plot(modelo)
602 par(mfrow=c(1,1))
603 '''
604
605
606
607 En este caso el mes si parece tener cierto efecto.
608
609 ```{r, echo=FALSE}
610 df_residuos <- data.frame(
611   fitted = modelo$fitted.values,
612   residuals = modelo$residuals,
613   Mes = mes_nuevo
614 )
615 ggplot(df_residuos, aes(x = fitted, y = residuals, color = Mes)) +
616   geom_point() +
617   labs(title = "Residuos del modelo por mes",
618        x = "Fecha",
619        y = "Residuo") +
620   theme_minimal() +
621   scale_color_brewer(palette = "Set3")
622 ```
623
624
625
626 Haciendo el modelo con los meses llegamos a un modelo con un R
    cuadrado gigantesco en comparaci n con lo que se ha estado viendo
    a lo largo de todo el estudio, llegando a casi 95% de deviance
    explicada.
627
628 ```{r, echo=FALSE}
629 modelo <- lm(Tensi n.de.Medio.Ciclo.V.RMS.NG.Med ~ Irradiancia *
    Temperatura + mes_nuevo, data = nuevo_modelos)
630 summary(modelo)
631 plot(modelo)
632 ```
633
634
635 Dado estos resultados tan buenos, realizo unos graficos para poder ver
    la precisi n del modelo con los fitted values.
636
637 ```{r}
638 par(mfrow=c(2,1))
639 plot(modelo$fitted.values,type="l")
640 plot(CT_nuevo$Tensi n.de.Medio.Ciclo.V.RMS.NG.Med,type="l")
641 par(mfrow=c(1,1))
642 ```
643

```

```

644 Se ve como se explica mucho mejor utilizando el mes, pues se tiene
      cierta curvatura, teniendo mayor en ciertos meses. Imagino que ese
      outlier es por el dato recogido en el momento del apag n como
      dijeron en el despacho.
645
646 ```{r}
647 plot(modelo$fitted.values, type = "l", col = "blue",
648       ylab = "Valor", xlab = " ndice ", lwd = 2,
649       main = "Valores ajustados vs Tensi n RMS NG Med")
650 lines(CT_nuevo$Tensi n.de.Medio.Ciclo.V.RMS.NG.Med, col = "red", lwd
        = 2)
651 legend("topleft", legend = c("Ajustado", "Tensi n RMS NG Med"),
652       col = c("blue", "red"), lty = 1, lwd = 2)
653 ```
654
655
656
657 ### Tensi n.NG.Med
658
659
660 ```{r, echo=FALSE}
661 modelo <- lm(Tensi n.NG.Med ~ Irradiancia * Temperatura, data =
        nuevo_modelos)
662 summary(modelo)
663 plot(modelo)
664 ```
665
666
667 En este caso el mes si parece tener cierto efecto.
668
669 ```{r, echo=FALSE}
670 df_residuos <- data.frame(
671   fitted = modelo$fitted.values,
672   residuals = modelo$residuals,
673   Mes = mes_nuevo
674 )
675 ggplot(df_residuos, aes(x = fitted, y = residuals, color = Mes)) +
676   geom_point() +
677   labs(title = "Residuos del modelo por mes",
678        x = "Fecha",
679        y = "Residuo") +
680   theme_minimal() +
681   scale_color_brewer(palette = "Set3")
682 ```
683
684
685 Se vuelve a tener efecto parecido a antes, se vuelve a probar modelo
      con el mes y se vuelven a obtener resultados tan buenos como en el
      anterior.
686
687 ```{r, echo=FALSE}

```

```

688 modelo <- lm(Tensi n.NG.Med ~ Irradiancia * Temperatura + mes_nuevo,
689 data = nuevo_modelos)
689 summary(modelo)
690 plot(modelo)
691 '''
692
693
694 Dado estos resultados tan buenos, realizo unos graficos para poder ver
695 la precisi n del modelo con los fitted values.
696
697 '''{r}
697 par(mfrow=c(2,1))
698 plot(modelo$fitted.values,type="l")
699 plot(CT_nuevo$Tensi n.NG.Med,type="l")
700 par(mfrow=c(1,1))
701 '''
702
703 Se ve como se explica mucho mejor utilizando el mes, pues se tiene
704 cierta curvatura, teniendo mayor en ciertos meses. Imagino que ese
705 outlier es por el dato recogido en el momento del apag n como
706 dijeron en el despacho.
707
708 '''{r}
709 plot(modelo$fitted.values, type = "l", col = "blue",
710 ylab = "Valor", xlab = " ndice ", lwd = 2,
711 main = "Valores ajustados vs Tensi n NG Med")
712 lines(CT_nuevo$Tensi n.NG.Med, col = "red", lwd = 2)
713 legend("topleft", legend = c("Ajustado", "Tensi n NG Med"),
714 col = c("blue", "red"), lty = 1, lwd = 2)
715
716
717
718
719 ### Tensi n.NG.Min
720
721 No se ponen los resultados, modelos practicamente iguales, mismo
722 efecto del mes en los residuos (gr fico pr cticamente igual).
723
724 '''{r, results='hide', fig.show = 'hide'}
725 modelo <- lm(Tensi n.NG.Min ~ Irradiancia * Temperatura, data =
726 nuevo_modelos)
727 summary(modelo)
728 plot(modelo)
729 '''
730
731 '''{r, results='hide', fig.show = 'hide'}
732 df_residuos <- data.frame(
733 fitted = modelo$fitted.values,
734 residuals = modelo$residuals,
735 Mes = mes_nuevo

```

```

732 )
733 ggplot(df_residuos, aes(x = fitted, y = residuals, color = Mes)) +
734   geom_point() +
735   labs(title = "Residuos del modelo por mes",
736         x = "Fecha",
737         y = "Residuo") +
738   theme_minimal() +
739   scale_color_brewer(palette = "Set3")
740 '""
741
742 Volvemos a tener R cuadrado muy alto.
743
744 '""{r, echo=FALSE}
745 modelo <- lm(Tensi n.NG.Min ~ Irradiancia * Temperatura + mes_nuevo,
746             data = nuevo_modelos)
747 summary(modelo)
748 plot(modelo)
749 '""
750
751 Dado estos resultados tan buenos, realizo unos graficos para poder ver
752   la precisi n del modelo con los fitted values.
753
754 '""{r}
755 par(mfrow=c(2,1))
756 plot(modelo$fitted.values, type="l")
757 plot(CT_nuevo$Tensi n.NG.Min, type="l")
758 par(mfrow=c(1,1))
759 '""
760 Se ve como se explica mucho mejor utilizando el mes, pues se tiene
761   cierta curvatura, teniendo mayor en ciertos meses. Imagino que ese
762   outlier es por el dato recogido en el momento del apag n como
763   dijeron en el despacho.
764
765 '""{r}
766 plot(modelo$fitted.values, type = "l", col = "blue",
767       ylab = "Valor", xlab = " ndice ", lwd = 2,
768       main = "Valores ajustados vs Tensi n NG Min")
769 lines(CT_nuevo$Tensi n.NG.Min, col = "red", lwd = 2)
770 legend("topleft", legend = c("Ajustado", "Tensi n NG Min"),
771       col = c("blue", "red"), lty = 1, lwd = 2)
772 '""
773
774 ### Corriente.L1.Min
775
776
777 Modelo con unos residuos muy malos, ns yo que decir de este.

```

```

778   "{r, echo=FALSE}
779   modelo <- lm(Corriente.L1.Min ~ Irradiancia * Temperatura, data =
780     nuevo_modelos)
781   summary(modelo)
782   par(mfrow=c(2,2))
783   plot(modelo)
784   par(mfrow=c(1,1))
785   ""
786
787   Parece ser que agosto es el principal causante del problema con los
788     residuos.
789
790   "{r, echo=FALSE}
791   df_residuos <- data.frame(
792     fitted = modelo$fitted.values,
793     residuals = modelo$residuals,
794     Mes = mes_nuevo
795   )
796   ggplot(df_residuos, aes(x = fitted, y = residuals, color = Mes)) +
797     geom_point() +
798     labs(title = "Residuos del modelo por mes",
799           x = "Fecha",
800           y = "Residuo") +
801     theme_minimal() +
802     scale_color_brewer(palette = "Set3")
803   ""
804
805
806   Aunque no se tenga una estructura de 'bandas' como se tenia en los
807     otros modelos, debido a la influencia de agosto y septiembre se
808     modela con el mes como variable para ver si se pueden mejorar los
809     residuos.
810
811   Los residuos siguen con un comportamiento extra o pero parece mejorar
812     el modelo.
813
814   "{r, echo=FALSE}
815   modelo <- lm(Corriente.L1.Min ~ Irradiancia * Temperatura + mes_nuevo,
816     data = nuevo_modelos)
817   summary(modelo)
818   plot(modelo)
819   ""
820
821   Dado estos resultados tan buenos, realizo unos graficos para poder ver
822     la precisi n del modelo con los fitted values.
823   Vemos que el comportamiento de la variable en si es extra o y dificil

```

```

    de explicar, pues hay un gran escalon al principio.
821   '{r}
822 plot(modelo$fitted.values,type="l")
823   '
824
825 Se ve como se explica mucho mejor utilizando el mes, pues se tiene
    cierta curvatura, teniendo mayor en ciertos meses. Imagino que ese
    outlier es por el dato recogido en el momento del apag n como
    dijeron en el despacho.
826
827   '{r}
828 plot(modelo$fitted.values, type = "l", col = "blue",
829       ylab = "Valor", xlab = " ndice ", lwd = 2,
830       main = "Valores ajustados vs Corriente L1 Min")
831 lines(CT_nuevo$Corriente.L1.Min, col = "red", lwd = 2)
832 legend("topleft", legend = c("Ajustado", "Corriente L1 Min"),
833       col = c("blue", "red"), lty = 1, lwd = 2)
834   '
835
836
837
838 Faltar an otros 5 modelos, pero demomento he hecho los 5 primeros.
839
840
841 # Modelos GAM CT nuevo
842
843
844   '{r, results='hide', fig.show = 'hide'}
845 modelo <- gam(Corriente.L1.Min ~ s(Irradiancia) + s(Temperatura) + ti(
    Irradiancia, Temperatura), data = nuevo_modelos)
846 summary(modelo)
847 plot(modelo)
848   '
849
850
851
852 El comportamiento es totalmente diferente a bombas, aunque estemos
    comparando variables diferentes, en el anterior analisi ocurria que
    los modelos en ambos CT eran parecidos. En cambio, aqui vemos que
    su estrucutura es diferente, adem s de explicar un 20% m s de
    variabilidad, tambi n se tiene distinto comportamiento para todas
    las variables.
853 Aqui se puede ver una comparacion de todas las opciones de modelos,
    nos quedamos con el mas completo como ya sabiamos.
854
855   '{r}
856 # Modelo completo
857 modelo_completo <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
858
859 # Modelo doble

```

```

860 modelo_doble <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia) + s(
    Temperatura), data = nuevo_modelos)
861
862 # Modelo sin irradiancia
863 modelo_sin_irr <- gam(Potencia.Aparente.L1N.Med ~ s(Temperatura), data
    = nuevo_modelos)
864
865 # Modelo sin temperatura
866 modelo_sin_temp <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia),
    data = nuevo_modelos)
867
868 # R ajustado
869 summary(modelo_completo)$r.sq
870 summary(modelo_doble)$r.sq
871 summary(modelo_sin_irr)$r.sq
872 summary(modelo_sin_temp)$r.sq
873
874 # Deviance explicada
875 summary(modelo_completo)$dev.expl
876 summary(modelo_doble)$dev.expl
877 summary(modelo_sin_irr)$dev.expl
878 summary(modelo_sin_temp)$dev.expl
879
880 ‘‘‘
881
882
883
884 Esto es una prueba de unas permutaciones para ver importancia, tampoco
    lo veo del todo interesante pero bueno.
885
886
887 ‘‘‘{r}
888 set.seed(123)
889
890 # Datos originales
891 original_preds <- predict(modelo_completo, newdata = nuevo_modelos)
892
893 # Permutar temperatura
894 nuevo_perm_temp <- nuevo_modelos
895 nuevo_perm_temp$Temperatura <- sample(nuevo_modelos$Temperatura)
896
897 perm_temp_preds <- predict(modelo_completo, newdata = nuevo_perm_temp)
898
899 # Permutar irradiancia
900 nuevo_perm_irr <- nuevo_modelos
901 nuevo_perm_irr$Irradiancia <- sample(nuevo_modelos$Irradiancia)
902
903 perm_irr_preds <- predict(modelo_completo, newdata = nuevo_perm_irr)
904
905 # Calcular RMSE
906 rmse <- function(y, yhat) sqrt(mean((y - yhat)^2))

```

```

907
908 original_rmse <- rmse(nuevo_modelos$Potencia.Aparente.L1N.Med,
    original_preds)
909 temp_rmse <- rmse(nuevo_modelos$Potencia.Aparente.L1N.Med,
    perm_temp_preds)
910 irr_rmse <- rmse(nuevo_modelos$Potencia.Aparente.L1N.Med,
    perm_irr_preds)
911
912 # Ver aumento de error
913 c(
914   original = original_rmse,
915   perm_temp = temp_rmse,
916   perm_irr = irr_rmse
917 )
918
919 '''
920
921
922
923 Resumen del modelo con interaccion.
924
925 '''{r, echo=FALSE}
926 summary(modelo_completo)
927 plot(modelo_completo)
928 '''
929
930
931
932 No solo eso, tambi n vemos que en comportamiento general, ahora esta
    justo al rev s , el valor baja cuando irradiancia alta y
    temperatura baja.
933
934 '''{r, echo=FALSE, warning=FALSE}
935 vis.gam(modelo_completo, view = c("Irradiancia", "Temperatura"), plot.
    type = "persp")
936 vis.gam(modelo_completo, view = c("Irradiancia", "Temperatura"), plot.
    type = "contour")
937 '''
938
939
940
941 Igual que antes, resultados pr cticamente iguales, mismas graficas y
    resumentes. No se pone para no alargar el pdf.
942
943 '''{r, results='hide', fig.show = 'hide'}
944 modelo <- gam(Potencia.Activa.L1N.Min ~ s(Irradiancia) + s(Temperatura
    ) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
945 summary(modelo)
946 plot(modelo)
947 '''
948

```

```

949  ““{r, results='hide', fig.show = 'hide'}
950  modelo <- gam(Potencia.Aparente.L1N.Min ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
951  summary(modelo)
952  plot(modelo)
953  ““
954
955  ““{r, results='hide', fig.show = 'hide'}
956  modelo <- gam(Corriente.L1.Med ~ s(Irradiancia) + s(Temperatura) + ti(
    Irradiancia, Temperatura), data = nuevo_modelos)
957  summary(modelo)
958  plot(modelo)
959  ““
960
961  ““{r, results='hide', fig.show = 'hide'}
962  modelo <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura), data = nuevo_modelos)
963  summary(modelo)
964  plot(modelo)
965  ““
966
967
968
969
970  ### Se pureba ahora a introducir el mes a los modelos GAM
971
972  En este caso, parece que a adir el mes sin suavizado si induce cierta
    mejora.
973
974  ““{r}
975  modelo <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura) + mes_nuevo, data =
    nuevo_modelos)
976  summary(modelo)
977  plot(modelo)
978  ““
979
980  Utilizar el mes con suavizado tambi n parece mejorar el modelo.
981
982
983  ““{r}
984  modelo <- gam(Potencia.Aparente.L1N.Med ~ s(Irradiancia) + s(
    Temperatura) + ti(Irradiancia, Temperatura) + s(mes_nuevo, bs = "re
    "), data = nuevo_modelos)
985  summary(modelo)
986  plot(modelo)
987  ““
988
989
990  Los modelos que se han considerado con resultados semejanes, tambi n
    se comportan parecido al a adir la variable mes de las dos maneras

```

. Mejoran hasta 0.75 aproximadamente y los gráficos son prácticamente iguales. Como mucho el gráfico de la interacción en alguna zona concreta es algo más diferente, pero ahora prefiero no alargar el pdf y dejarlo para cuando vaya más en detalle en cada cosa.

```
# Pruebas de clustering
```

```
## Búsqueda de mejor número de clusters
```

```
Grafico de codo.
```

```
```{r, echo=FALSE}
calidad_nuevo_esc <- scale(nuevo_bucle)
wss <- sapply(1:10, function(k){
 kmeans(calidad_nuevo_esc, centers = k, nstart = 10)$tot.withinss
})

plot(1:10, wss, type = "b", pch = 19,
 xlab = "Número de clusters", ylab = "Suma de cuadrados intra-
 cluster")
```
```

```
## Con 3 clusters
```

```
```{r, echo=FALSE}
#set.seed(123)
num_clusters <- 3

clustering <- kmeans(calidad_nuevo_esc, centers = num_clusters)
nuevo_modelos$cluster_calidad <- as.factor(clustering$cluster)
```

```{r, echo=FALSE}
resumen_clusters <- nuevo_modelos %>%
 group_by(cluster_calidad) %>%
 summarise(
 media_irradiancia = mean(Irradiancia, na.rm = TRUE),
```

```

1036 sd_irradiancia = sd(Irradiancia, na.rm = TRUE),
1037 media_temperatura = mean(Temperatura, na.rm = TRUE),
1038 sd_temperatura = sd(Temperatura, na.rm = TRUE),
1039 n = n()
1040)
1041 print(resumen_clusters)
1042 ```
1043
1044
1045
1046 ```{r, echo=FALSE}
1047 # Dispersi n Irradiancia vs Temperatura por cl ster
1048 ggplot(nuevo_modelos, aes(x = Irradiancia, y = Temperatura, color =
1049 cluster_calidad)) +
1050 geom_point(alpha = 0.5) +
1051 theme_minimal()
1052
1053 # Boxplot de temperatura por cl ster
1054 ggplot(nuevo_modelos, aes(x = cluster_calidad, y = Temperatura, fill =
1055 cluster_calidad)) +
1056 geom_boxplot() +
1057 theme_minimal()
1058
1059 # Igual para irradiancia
1060 ggplot(nuevo_modelos, aes(x = cluster_calidad, y = Irradiancia, fill =
1061 cluster_calidad)) +
1062 geom_boxplot() +
1063 theme_minimal()
1064 ```
1065
1066 ## Se prueba ahora en vez de todas las variables, las mejores por
1067 parte de los modelos GAM con interaccion
1068
1069 ```{r, echo=FALSE}
1070 top_vars_10 <- resultados_gam_con_nuevo$Indicador[1:10]
1071 top_vars_15 <- resultados_gam_con_nuevo$Indicador[1:15]
1072 datos_top <- scale(nuevo_modelos[, top_vars_10])
1073 ```
1074
1075
1076 Parece que sigue siendo 3 clusters lo mejor.
1077
1078 ```{r, echo=FALSE}
1079 wss <- sapply(1:10, function(k){
1080 kmeans(datos_top, centers = k, nstart = 10)$tot.withinss
1081 })
1082

```

```

1083 plot(1:10, wss, type = "b", pch = 19,
1084 xlab = "N mero de clusters", ylab = "Suma de cuadrados intra-
 cluster")
1085 ' ' '
1086
1087
1088 ## Con 3 clusters
1089
1090
1091 ' ' '{r, echo=FALSE}
1092 #set.seed(123)
1093 num_clusters <- 3
1094
1095 clustering <- kmeans(datos_top, centers = num_clusters)
1096 nuevo_modelos$cluster_calidad <- as.factor(clustering$cluster)
1097 ' ' '
1098
1099
1100
1101
1102 ' ' '{r, echo=FALSE}
1103 resumen_clusters <- nuevo_modelos %>%
1104 group_by(cluster_calidad) %>%
1105 summarise(
1106 media_irradiancia = mean(Irradiancia, na.rm = TRUE),
1107 sd_irradiancia = sd(Irradiancia, na.rm = TRUE),
1108 media_temperatura = mean(Temperatura, na.rm = TRUE),
1109 sd_temperatura = sd(Temperatura, na.rm = TRUE),
1110 n = n()
1111)
1112 print(resumen_clusters)
1113 ' ' '
1114
1115
1116
1117
1118
1119 De aqui alomejor si se puede llegar a sacar algo.
1120
1121 ' ' '{r, echo=FALSE}
1122 # Dispersi n Irradiancia vs Temperatura por cl ster
1123 ggplot(nuevo_modelos, aes(x = Irradiancia, y = Temperatura, color =
 cluster_calidad)) +
1124 geom_point(alpha = 0.5) +
1125 theme_minimal()
1126
1127 # Boxplot de temperatura por cl ster
1128 ggplot(nuevo_modelos, aes(x = cluster_calidad, y = Temperatura, fill =
 cluster_calidad)) +
1129 geom_boxplot() +
1130 theme_minimal()

```

---

```

1131
1132 # Igual para irradiancia
1133 ggplot(nuevo_modelos, aes(x = cluster_calidad, y = Irradiancia, fill =
 cluster_calidad)) +
1134 geom_boxplot() +
1135 theme_minimal()
1136 ' ' '
1137
1138
1139
1140 Si se hace con 4 clusters, la estrucutra es parecida, contiene
 tambi n esas "bandas" de colores en el gr fico, y en los de caja
 sigue destacando solo un grupo, el m s alto.

```

Listing C.19: Análisis final