Clasificación de paquetes de series temporales en R. Ejemplo de modelización ARIMA con datos de contaminación de Madrid

Trabajo Fin de Grado de Estadística

Facultad de Ciencias
Universidad de Valladolid

28 de junio de 2020

Universidad de Valladolid

Autor:
Martín Mateos, Miguel
Tutora:
Rodríguez del Tío,María del Pilar

Índice

Resumen

Este trabajo fin de grado presenta una pequeña clasificación de los paquetes y funciones de R más recomendables para tratar con series temporales y en particular con modelos ARIMA.

Como ejemplo para el uso de R en la modelación de una serie temporal, se utilizan datos horarios y diarios de contaminación de NO_2 de una zona concreta de Madrid, tanto para el análisis descriptivo como para crear modelos predictivos. Previamente, se realiza un preprocesamiento de los datos.
A nivel descriptivo se realizan comparaciones anuales,mensuales,diarias y horarias utilizando la herramienta Tableau.

Además, se trabaja con datos de Marzo y Abril de 2020 en los que se produjo el estado de alarma por la enfermedad del Covid19, lo cual se verá reflejado en una disminución notable de la contaminación.

Para los modelos predictivos se aplica un enfoque de Box-Jenkins a dos series,una mensual y otra horaria, esta última se modela con doble estacionalidad.

Palabras clave

Series temporales, ARIMA, R, Tableau, multiestacionalidad, contaminación Madrid, dióxido de nitrógeno, filtrado datos.

Abstract

This Final Degree Project presents a small classification of the most recommended R packages and functions for dealing with time series and in particular with ARIMA models.

As an example for the use of R in the modeling of a time series, hourly and daily pollution data of NO_2 are used from a specific area in Madrid, both for descriptive analysis and for creating predictive models. Previously, the data is pre-processed.
At a descriptive level, annual, monthly, daily and hourly comparisons are made using the Tableau tool.

In addition, we work with data from March and April 2020, when the state of alarm for Covid disease19 occurred,, which will be reflected in a significant decrease in the contamination levels.

For predictive models, a Box-Jenkins methodology is applied to two series, one monthly and the other one hourly. The latter is modeled with double seasonality.

1. Introducción

1.1. Objetivos

Este trabajo de fin de grado se podría dividir en dos partes. La primera dedicada a un estudio de paquetes del lenguaje de programación R relacionados con el análisis de series temporales. En gran parte la búsqueda va enfocada a comandos necesarios o útiles para modelos ARIMA. No obstante, habrá comandos y paquetes que serán útiles para otros modelos.

En cuanto a la segunda parte de este trabajo, se analizan dos series como ejemplo con todo detalle aplicando el enfoque de Box-Jenkins. Se analizarán datos de la calidad del aire de Madrid tanto a nivel mensual como horario. Se seleccionarán en ambos casos unos datos de entrenamiento y con el modelo resultante se intentará minimizar el error entre la predicción y los datos reales.

Además, se incluye una pequeña sección derivada de los acontecimientos ocurridos por el virus Covid-19. Son interesantes los resultados extraídos sobre como ha afectado este confinamiento a la contaminación en Madrid, concretamente en la estación de estudio de mi investigación que es la Plaza Castilla. Comparamos las reducciones de NO2 de manera mensual en Marzo y Abril en relación a estos meses en años anteriores, observando que, tras el confinamiento, se produjo un descenso muy pronunciado del gas nocivo, que se transformó en un aire más limpio.

1.2. Motivación

Hoy en día, el análisis de series temporales se realiza en casi todos los ámbitos, al igual que realizar preprocesamientos y limpiezas a datos en bruto(raw Data).

La elección de unos datos abiertos sin procesamiento me ha supuesto una motivación de cara al futuro laboral. La decisión de utilizar los datos de contaminación de Madrid me pareció un tema interesante a tratar, ya que todos los días se habla de ello en los medios informativos.

Según el periódico el País [1], la contaminación de NO2 a mediados de 2019 descendió significativamente en 14 de 24 estaciones de Madrid, provocado por las medidas otorgadas en Madrid Central. En efecto, como veremos más adelante en el análisis descriptivo, 2019 sufrió un descenso general de NO2 que provocará que la predicción a nivel mensual no sea tan precisa como se desearía. Además, con la irrupción del virus Covid19 en Marzo y Abril de 2020, de manera muy destacada en Madrid con la parada de la mayoría de la ciudad en el sector industrial o de automoción, se traduciría en una bajada destacada de la contaminación en esta ciudad como veremos más adelante en la sección 3.3.

La primera parte de este trabajo basada en recopilar información organizada sobre los paquetes del lenguaje R para el análisis de una serie temporal, está motivada por la importancia que el uso de R está tomando en la actualidad.
En la asignatura de Análisis de Series Temporales se trabajaba con los programas SAS y Statgraphics, por lo que añadir esta guía para utilizar R en el análisis de Series Temporales nos parece oportuno como tema del presente trabajo.

El libro **Time Series Analysis with Applications in R** [2], me ha servido como base previa para realizar un análisis sobre distintos paquetes en R a utilizar para analizar una serie temporal, así como sus comandos implementados. No obstante, se plantea como una 'ayuda' o 'guía' para

personas/estudiantes que quieran familiarizarse con los comandos que R ofrece para un análisis univariante detallado de series temporales.

1.3. Implementación

Para llevar a cabo este trabajo se va a hacer uso del siguiente software estadístico: SAS,R y Tableau.

Se usará el lenguaje R para la lectura y tratamiento de los datos, además de la posterior aplicación de los modelos ARIMA a los datos.También se utilizará la exportación de datos con R para el análisis con otros programas.

La herramienta de visualización gráfica Tableau se usará para, una vez exportados los datos filtrados en R, poder visualizarlos de manera gráfica. Esta herramienta nos permite obtener gráficos horarios,diarios,mensuales o anuales interactivos y con variedad de filtros que permiten dar distintos enfoques a la serie de manera sencilla. En definitiva, en el apartado de análisis descriptivo de los datos [3.3], se expondrán los gráficos y tablas realizados con esta herramienta.

Por último, se lleva a cabo la comparación de la modelación de las series utilizando ARIMA con SAS y con R. Además el uso de SAS tiene una ventaja adicional, ya que se pueden aplicar más de una estacionalidad a una serie en modelos ARIMA, lo cual no está operativo a día de hoy en R , al menos desde el enfoque Box-Jenkins sin tener que acudir a enfoques más generales como el de los modelos de Espacio de Estados(State Space)[3].

2. Análisis de paquetes de series temporales en R

Tras investigar algunos de los diferentes paquetes que existen en R, he escogido dos de ellos, **forecast** y **TSA**. La primera porque realizando búsquedas sobre ciertas funciones específicas para analizar la serie, como ajustarla o predecirla, en la mayoría de mis búsquedas era la más usada e implementada y con un mayor rango de aplicación en sus diferentes temáticas. El segundo paquete elegido, es el paquete más utilizado en el libro inicial que me tomé como referencia **Time Series Analysis with Applications in R**[2].

En resumen, ambos paquetes disponen de un conjunto de funciones bastante completo y son de uso frecuente por los investigadores, razones por las que las he elegido para explicarlas con más énfasis. En concreto, se añade más información del paquete forecast con el fin de no repetir comandos que se utilizan para el mismo fin.

No obstante, quiero aclarar que el objetivo de esta parte no será detallar todos los procedimientos para el análisis de una serie temporal con estas librerías. No se pretende explicar minuciosamente cada una de las funciones de las librerías, sino las más útiles y que no posean otras librerías, o proporcionen información conjunta con una única orden.Se adjuntará la fuente asociada a cada método de la librería específica para explicar más detallado su funcionamiento. Además irá mayormente enfocado a su uso para modelos ARIMA y series univariantes, aunque muchas de las funciones indicadas valdrán para más modelos.

De manera menos detallada se informará sobre métodos o funciones de algunos paquetes también de series temporales que aportan funcionalidades que otras librerías no dan.
Al principio se detallarán algunos de los comandos más básicos y necesarios para el análisis de cualquier serie temporal.Todos ellos se encuentran en la librería *stats* [4]. A continuación se expondrán los diferentes paquetes estadísticos(se utilizará el término paquete/librería indistantemente) con los comandos que se ha considerado más interesantes.
Por último, con motivo de modelizar una serie con más de una estacionalidad, en el presente trabajo se detallan algunas formas de llevarlo a cabo en R aunque para modelos ARIMA no se haya encontrado una forma de realizarlo.

2.1. Comandos básicos y necesarios para series temporales

- ts : Se utiliza para crear series temporales. Se le pueden pasar un vector o una matriz de datos. Puedes incorporar las fechas desde cuando datan hasta cuando finalizan. Además se le añade la frecuencia de los datos(12 si es mensual, 24 horarias, 7 diarias..). A continuación se añade la sentencia asociada a unos datos desde enero de 2018 hasta enero de 2020 con frecuencia mensual [5]:

  ts(data = x, start=c(2018,1), end = c(2020,1), frequency = 12)

- diff: Su uso se lleva a cabo para diferenciar la serie, normalmente, debido a una tendencia clara. Se puede tanto diferenciar de manera regular (con la suborden *differences=x* indicando el número de diferenciaciones a realizar) como diferenciar estacionalmente (añadiendo el número del periodo de la serie en la suborden *lag=x* indicando dicho periodo). A continuación se añade un ejemplo de diferenciar dos veces estacionalmente con periodo 12 [6]:

diff(x, lag = 12, differences = 2)

- acf: Este comando realiza de manera gráfica la función de autocorrelación. Gracias a este gráfico podemos detectar si una serie es estacionaria o no, ya que describe las autocorrelaciones, es decir, determinar la correlacion entre observaciones consecutivas o entre observaciones del periodo en concreto. Importante la suborden lag.max para indicar el número máximo de lags que se quieren utilizar. Por defecto suelen poner muy pocos lo que puede llevar a interpretar de manera errónea dicha gráfica.La orden x se corresponderá con una serie temporal. A continuación se expone un ejemplo[7]:

  acf(x, lag.max = 10,...)

- log: Se encarga de transformar la serie original en una serie logarítmica con el objetivo de reducir la varianza. En concreto sería:

  log(serie)

2.2. Librería forecast

[8]

El paquete forecast escrito por *Rob J Hyndman*, contiene métodos y herramientas para mostrar y analizar pronósticos de series de tiempo univariantes, que incluyen suavizado exponencial a través de modelos State Space y modelado automático **ARIMA**.

Uno de los objetivos de este trabajo es modelar una serie temporal que pueda trabajar con más de una estacionalidad(es decir si por ejemplo es horaria, poder modelizarla incluyendo periodos correspondientes a un día, una semana, un mes y un año). Por tanto habría que buscar un paquete que permitiera aplicar más de una estacionalidad a una serie dada. Claramente, esto tiene interés si se detecta que estos periodos son necesarios para explicar el comportamiento de la serie.

Para obtener una serie con más de una estacionalidad en R se utiliza el comando msts() que se detallará a continuación. No obstante, hasta el momento no existe una función (o al menos no se ha encontrado) en todo el CRAN, que permita modelizar una serie con más de una estacionalidad utilizando la metodología de Box-Jenkins.

2.2.1. msts()

A modo de resumen se expone la estructura de ésta, que es:

msts(data, seasonal.periods, ts.frequency = floor(max(seasonal.periods));

siendo data, el dataset a tratar, *seasonal.periods()* los distintos periodos de la serie y *ts.frequency()* que es la frecuencia de partida que se utiliza.

Si tenemos una serie horaria, y queremos modelarla además diariamente y semanalmente se realizaría del siguiente modo:

msts(data,seasonal.periods=c(24,168)).[9]

No obstante, no he encontrado un método ARIMA en R que acepte una serie temporal de múltiple estacionalidad por lo que de momento se realizará con una única estacionalidad. Por tanto, esta parte se implementará en SAS como veremos más adelante en nuestro ejemplo.

2.2.2. Arima()

Este comando ajusta un modelo ARIMA de una serie univariante. La principal diferencia con el comando básico de arima de stats es que se puede incluir la constante para determinar si influye de manera positiva o negativa en el modelo. Como comentamos anteriormente, solo se puede incluir una serie de tipo univariante para ajustar modelos ARIMA(es decir objeto ts). A modo de ejemplo se añade la sentencia asociada a un comando ARIMA(1,1,0)(0,0,1)12 correspondiente con una diferenciación regular, un parámetro autorregresivo y parámetro de media móvil en parte estacional.

El comando es el siguiente:

Arima(y,order=c(1,1,0),seasonal=list(order=c(0,0,1),period=12,include.constant=TRUE)[10],

donde *y* es la serie temporal univariante, *order* corresponde a los parámetros por la parte regular, concretamente,autoregresivos en la primera posición, número de diferenciaciones regulares en el segundo y número de parámetros de media móvil.
Del mismo modo, pero para la parte estacional de la serie se realiza en *seasonal*. En el atributo *period* añadimos el periodo de la serie(12 Mensual, 7 Diario..etc) y por último, el atributo diferenciador que hemos descrito, la opción de incluir la constante para observar si es influyente en el modelo y en qué medida. Además, por defecto ,la estimación se realiza utilizando máxima verosimilitud('ML'). Si queremos realizar mínimos cuadrados es 'CSS'.

2.2.3. auto.arima()

Otro comando interesante a la hora de automatizar la búsqueda de mejores modelos de Box-Jenkins es *auto.arima()*.

Proporciona una opción rápida para construir estimaciones de las series temporales, evaluando entre todos los posibles modelos teniendo en cuenta diversos criterios como la estacionalidad, las diferencias de la serie..etc. No obstante, no tiene en cuenta la validación de los residuos de los distintos modelos. Tiene muchas opciones para restringir la salida como puede ser un valor dado de p del modelo autoregresivo, el máximo número de autoregresores estacionales(P), así como otros aspectos. Para que te informe de los mejores modelos posibles hay que añadir la sentencia *trace=TRUE*. Con stepwise a TRUE nos saca todos los modelos que encuentra pero esto tiene como consecuencia que tiene mucho tiempo de cómputo. Además es interesante que en este comando se puede elegir el método de estimación ya sea por mínimos cuadrados o máxima verosimilitud mediante la opcion *method*.[11]

La estructura este comando es la siguiente:

Comando: auto.arima(data, trace=TRUE).

Quizás este comando sea útil para llevarte una ligera idea para que modelo ARIMA puede acercarse más a la serie dada. No obstante, no es recomendable guiarse por el resultado de ésta sin realizar un análisis más exhaustivo, ya que tiende a recomendarte los modelos con mejor AIC o BIC y hay que tener en cuenta otros factores como el sobreajuste, la significación de cada parámetro, o la validación de residuos.

2.2.4. tsdisplay() y checkresiduals()

Otro comando útil para la obtención de los residuales de manera gráfica es con tsdisplay(). Nos muestra el gráfico de residuales, acf y pacf en un mismo dashboard. Se puede añadir el número de retardos máximo a mostrar con *lag.max=x*. De nuevo, solo acepta series temporales de una única estacionalidad.[12]

tsdisplay(residuals(modeloarima), lag.max=10)

Este comando puede ser de gran utilidad para la parte de validación de un modelo, observando como se comportan los residuales de la serie con una única sentencia.

Otro comando muy parecido es *checkresiduals()*. La diferencia radica en que este proporciona un histograma para ver si se distribuye los residuales como una normal y *tsdisplay* proporcionaba PACF. Tanto el gráfico de residuales como el ACF, lo proporcionan ambas sentencias.[13]

checkresiduals(objeto).

siendo objeto el modelo aplicado a los datos.

2.2.5. accuracy()

Este comando se encarga de obtener el error producido entre los datos predichos y los datos reales. Se puede utilizar distintas medidas del error como RMSE,MSE,MAE...[14] Un ejemplo es:

accuracy(objeto)

Donde objeto puede ser de tipo Arima, ets(suavizado exponencial) o lm(modelo lineal), además que sea de tipo forecast el objeto.
Para evaluar el error en modelos ARIMA se suele utilizar tanto el AIC como el BIC, pero la aportación de otros errores como el RMSE puede ayudar a solventar dudas.

2.2.6. forecast()

Este comando se utiliza para predecir la serie temporal de modelos dados. Se le pasa un objeto Arima y se le indica el numero de pasos adelante, así como el nivel de confianza. En este caso podemos encontrar en predict una funcionalidad muy semejante que no hace uso de ninguna librería. Ambos comandos solo aceptan series temporales con una estacionalidad.[15]

forecast(objeto Arima,h=10,level=c(99.5)).

Destacar que esta sentencia solo funciona si has aplicado modelos ARIMA con el comando predeterminado de la librería forecast *Arima* [2.2.2].

2.3. Libreria TSA

Esta segunda librería [16] me ha parecido de las más completas junto a la ya explicada. En el libro **Time Series Analysis with applications in R**[2] hace un seguimiento por todas fases para analizar una serie temporal, desde la estimación a la predicción.
Además, cuenta con funciones para modelos multivariantes que no se describen aquí puesto que no son objeto de este trabajo.

2.3.1. LB.test()

Otro comando útil en la etapa de validación es **LB.test** que realiza el test de Ljung-Box a los residuos. Ésta, es una evolución del método Box.test en el que además de examinar la hipótesis nula de independencia de la serie, también comprueba si los residuos pueden ser ruido blanco o no. Hay que pasarle un objeto de tipo arima. Añadiendo *lag=X*, se puede determinar hasta que retardo acumulado se realiza el test.[17]

LB.test(objetoArima,type="Ljung-Box",lag=X)

2.3.2. armasubsets()

Esta función encuentra el mejor modelo ARMA. La funcionalidad es similar a auto.arima() del paquete forecast. No obstante , esta función es más sencilla, ya que se realiza para modelos ARMA y no ARIMA, es decir, no contemplan la diferenciación de la serie.[18]

armasubsets(serietemporal,nar,nma,ar.method)

Los parámetros nar,nma y ar.method significan el orden máximo de parámetros autoregresivos, el orden máximo de parámetros de media móvil y el método en le que basarse para obtener el mejor modelo. Por defecto es 'ols' que usa el AIC.

2.4. Libreria tseries

La información sobre dicha librería se puede encontrar en [19].

2.4.1. adf.test()

La Prueba de Dickey-Fuller busca determinar la existencia o no de raíces unitarias en una serie de tiempo.[20]. La hipótesis nula de esta prueba es que existe una raíz unitaria en la serie.

adf.test(serietemporal, alternative = "stationary")

Para conseguir que la serie tratada sea estacionaria, el test deberá ofrecer un pvalor muy pequeño para determinar que la serie es estacionaria con una confianza alta. Otra opción es utilizar adfTest() de la librería fUnitRoots [21] que lleva a cabo el mismo test.

2.4.2. kpss.test()

El test de Kwiatkowski-Phillips Schmidt-Shin también es usado para contrastar en la hipótesis nula que una serie temporal es estacionaria frente a la alternativa de una raíz unitaria. La finalidad del test es la misma que Dickey-Fuller pero en este caso se pretende 'aceptar' la hipótesis nula para determinar que la serie sea estacionaria [22]. El comando se expone a continuación:

kpss.test(x,null=c("Trend"))

Es necesario añadir 'Trend' para que la hipótesis nula se rija por la tendencia estacionaria y no el nivel.

2.5. Librería lmtest

La información relacionada con esta librería puede encontrarse en [23]. Uno de los comandos más interesantes de esta librería es coeftest().Este comando aporta información complementaria a la que aporta el comando Arima().Muestra toda la información sobre los coeficientes y errores estándar(lo cual proporcionaba el comando Arima()) pero además realiza un test de Wald para cada parámetro del modelo para obtener la significancia de este en el modelo con el pvalor asociado [24]. Se describe a continuación:

coeftest(objetoArima)

También nos proporciona los intervalos basados en la verosimilitud perfil de cada parámetro del modelo con *confint(objeto)*.

2.6. Libreria caschrono

Las funciones o comandos de esta librería pueden encontrarse en [25]. Una de las formas más importantes para detectar si existe sobreajuste en el modelo es partir de la información que nos proporciona la matriz de correlaciones de los estimadores de los parámetros. Una forma de obtenerla es con el comando cor.arma. Con la ayuda de este comando nos podrá indicar si existen demasiados parámetros para ajustar el modelo o no. [26]

cor.arma(objeto)

En este caso objeto se identifica con un modelo arima realizado en el que se realiza la matriz de correlaciones para todos los parámetros implicados en dicho modelo arima.

2.7. Comandos para modelos con más de una estacionalidad

Uno de los objetivos al inicio del TFG era la aplicación de un modelo ARIMA con más de una estacionalidad pero los comandos de Arima solo aceptan series temporales con una única estacionalidad(es decir, no acepta una serie de tipo msts, solo ts).A continuación se citan algunos comandos que aceptan más de una estacionalidad y a qué procedimiento se refieren:

- msarima: Modelo ARIMA del state space [27]

- bats: Utiliza suavizado exponencial del state space + transformación de box-cox y errores arima.[28]
- tbats: Utiliza suavizado exponencial del state space + transformación box-cox + errores arima + estacionalidad trigonométrica[29].

3. Origen de los datos

Los datos a tratar en este trabajo provienen de la página de datos abiertos del Ayuntamiento de Madrid. En lo que concierne a este trabajo se van a utilizar datos de contaminación de Madrid.

En esta página de datos abiertos de Madrid [30], se pueden encontrar datos horarios desde 2001 hasta 2020 en distintas zonas de Madrid. Del mismo modo, en una página adyacente de la misma fuente [31] se pueden encontrar datos diarios desde 2001 hasta 2020. Los datasets a utilizar tendrán el formato CSV.

La interpretación de los datos que se va a realizar a continuación se puede encontrar en un pdf llamado interpretación datos de calidad del aire [31]. Un pequeño ejemplo del formato de los datos a utilizar es:

PROVINCIA	MUNICIPIO	ESTACIÓN	MAGNITUD	PUNTO_MUESTREO	AÑO	MES	D01	V01	...	D31	V31
28	79	4	1	28079004_1_38	2019	1	23	V		17	N

Haciendo referencia a los datos de la tabla de ejemplo, los campos correspondientes a D01 indican el Día 1 como D31 el día 31. La variable Mes nos indica el mes en el que se encuentra calculado el dato observado. Para cada día habrá un código de validación para indicar si los datos son válidos o no, habiendo una 'V' si es válido ó 'N' si es No válido.

Como se observa en las imágenes [1] y [2] podemos ver el punto de muestreo asociado a Plaza Castilla. Este código está compuesto por los campos provincia,municipio,estación y magnitud. Los códigos asociados a estos campos y a magnitud son (la técnica no hace falta ya que solo hay una opción) :

- Provincia: 28
- Municipio: 79
- Estación: 50
- Magnitud: 8

07	Monóxido de Nitrógeno	NO	µg/m³	08	Quimioluminiscencia
08	Dióxido de Nitrógeno	NO_2	µg/m³	08	Id.

Figura 1: Código Magnitud NO2

28079050	Pza. Castilla	Alta.- 08/02/2010 (00:00 h.)

Figura 2: Codigos Plaza Castilla

En este trabajo se va a escoger una zona específica de Madrid,concretamente la Plaza Castilla, que es una zona céntrica que se encuentra en el Paseo de la Castellana y que pienso que puede ser de interés.
La magnitud con la que se ha medido dicha calidad del aire es Dióxido de Nitrógeno(NO2) y la

técnica de muestreo es Quimioluminiscencia. A continuación se describe de manera resumida los detalles de la variable que se va a estudiar, dióxido de Nitrógeno(NO2).

El dióxido de nitrógeno(NO2) forma parte de un grupo de contaminantes gaseosos que se producen como consecuencia del tráfico rodado y la producción de energia. Los estudios realizados sobre la población humana indican que la exposición a largo plazo al NO2 puede provocar disminución de la función pulmonar y aumentar el riesgo de aparición de síntomas respiratorios como bronquitis aguda, tos y flema. La exposición individual al NO2 depende principalmente de las concentraciones exteriores locales. Sin embargo, también puede verse afectada por fuentes contaminantes de interiores como el humo del tabaco y las cocinas de gas o los aparatos de calefacción de gas sin ventilación [32].

Para los datos horarios sólo existen pequeñas diferencias. Una de ellas es que hay una variable que nos indica el día en el que están calculados los datos. En este caso hay una variable H01 indicando la 1ª hora del dia así como H24 para dar la última hora del dia. De nuevo, se utiliza una variable que nos indica por cada hora si dicha observación es válida('V') o no ('N').

PROVINCIA	MUNICIPIO	ESTACIÓN	MAGNITUD	PUNTO_MUESTREO	AÑO	MES	DIA	H01	V01	...	H24	V24
28	79	4	1	28079004_1_38	2019	1	1	23	V		17	N

3.1. Objetivo del Análisis

3.1.1. Datos Mensuales 2011-2020

Con los datos mensuales pretendemos entrenar un modelo estadístico a partir de los años 2011 a principios de 2018 con el objetivo de predecir la contaminación desde Marzo de 2018 hasta Febrero de 2020(no se incluye Marzo y Abril en los datos de prueba ya que como veremos en el análisis descriptivo los datos resultantes son muy anómalos debido al Covid19).

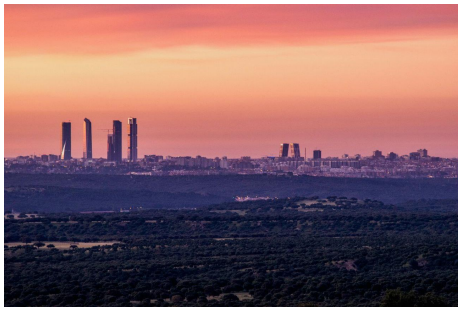
3.1.2. Datos horarios 2020 Covid19

La situación de emergencia sanitaria ocasionada por la COVID-19 junto a la pandemia internacional por la Organización Mundial de la Salud (OMS) el pasado 11 de marzo, crisis mundial sin precedentes en la historia, con amplias repercusiones en la movilidad y en la actividad económica.[33]

'Fruto de la declaración del estado de alarma por Real Decreto 463/2020, de 14 de marzo, hasta el 12 de abril por Real Decreto 476/2020, de 27 de marzo, se han adoptado una serie de medidas de limitación de la libertad de circulación de las personas que han derivado en una reducción drástica y generalizada del transporte y, en menor medida, de la actividad industrial y la generación de electricidad, fuentes principales de la emisión de los contaminantes a la atmósfera.' [33]

Teniendo en cuenta que el tráfico es el principal factor que influye en la calidad del aire urbano,sobretodo en Madrid, es evidente que un descenso tan influyente de la circulación y de sus emisiones a la atmósfera está traduciéndose en una mejora muy significativa de la calidad del aire. [33]

Una de las imágenes que nos deja esta situación de emergencia, es la diferencia de contaminación en el aire de un día de 2019 y un día del 2020 en el estado de alarma. En ambas imágenes además, se 'observa' la Plaza Castilla, el lugar de donde extraemos los datos:



(a) **Madrid Sin Contaminación** Fuente: [34]

(b) **Madrid con Contaminación** Fuente: [35]

Teniendo en cuenta los sucesos acontecidos, se ha decidido considerar la información sobre la contaminación de Madrid en Plaza Castilla de NO2 durante los 4 primeros meses de 2020, con un periodo horario.

3.2. Limpieza y Preprocesado

De partida se tienen dos tipos de archivos, uno horario con información extraída del año 2019 y 2020, y diarios de los años 2011 a 2020. El principal objetivo es convertir los datos en crudo(raw data) en información en una única columna(univariante) para realizar el estudio. Ejemplo de cómo son los datos en crudo tanto para datos horarios como diarios se observa en las tablas [48] y [49]. Además, con la existencia de valores no válidos en los conjuntos de datos, se tiene que realizar una serie de transformaciones para dar valor a estas observaciones de un modo adecuado.

Inicialmente, es unido en un único dataset los 10 años de 2011 a 2020(los archivos diarios vienen por años) con el objetivo de filtrar como hemos comentado, por la Plaza Castilla y con magnitud Dióxido de Nitrógeno.

Para cada año era necesario buscar los datos no válidos tal y como nos muestra el informe oficial de donde se extraen los datos. Estos datos corresponden con los **datos diarios** que servirán para obtener los mensuales promediando. A continuación se indican el número de observaciones no válidas asociada al año:

2011: 9N; 2012: 7N; 2013: 10N; 2014: 8N; 2015: 15N ; 2016: 6N; 2017: 9N ;2018: 8N ; 2019: 7N ; 2020: 3N

Hay que destacar que de los **No Válidos** de cada año no bisiesto, hay 6N provocados porque venían con 31 días todos los meses, y 7N en los años bisiestos por el día 29 de Febrero(son 2012,2016 y 2020).
El resto de observaciones no válidas son reemplazadas los valores por NA's para poder aplicar la media con el resto de datos. Por tanto, se obtiene la media de cada mes y con ello se forma un dataset con 112 observaciones mensuales desde Enero de 2011 hasta Abril de 2020 que más adelante

será modelado con la metodología Box-Jenkins. Una descripción de este dataset se observa en la tabla [4].

	Años	Meses	No2
1	2011	1	53.96667
2	2011	2	75.21429
3	2011	3	48.93548
4	2011	4	44.44828
5	2011	5	44.87097
6	2011	6	49.50000

Figura 4: **Dataset final mensual**

Con la obtención de los datos mensuales, el objetivo es procesar los datos diarios. Solo hace falta reemplazar los valores no válidos por los valores medios del mes correspondiente. Adicionalmente, son creados variables asociadas al día,mes y año correspondiente a la observación medida tras trasponer los datos de la variable explicativa NO2 para tener una serie univariante. Este proceso se realiza para poder llevar a cabo el análisis descriptivo de modo más sencillo y con la mayor información posible. Un ejemplo de ello se encuentra en la tabla [5].

	Anio	Mes	Dia	NO2
61	2020	3	1	14
62	2020	3	2	18
63	2020	3	3	26
64	2020	3	4	24
65	2020	3	5	22
66	2020	3	6	28

Figura 5: **Formato Dataset final diario**

Para la obtención de los datos horarios, hay que llevar a cabo una búsqueda sobre los valores no válidos. En este caso, la fuente [30], nos proporciona conjuntos de datos distintos para cada mes, por lo que se buscan los no válidos para cada uno de ellos. Los 4 primeros meses de 2020 se analizaron aparte. El número de valores no válidos encontrados son tres, correspondientes a las 10 del 9 de Enero, a las 3 del día 29 Marzo y a las 12 del día 12 Febrero. No obstante, para todo el año de 2019 fueron hallados 24(N) correspondientes con valores no válidos. En concreto para cada mes fueron:

Enero : 6N ;Febrero: 1N ;Marzo: 2N;Abril: 0N;Mayo: 3N;Junio: 5N;Julio:1N;Agosto: 1N;Septiembre: 0N;Octubre: 4N;Noviembre: 1N;Diciembre: 0N

Para todas las horas no válidas encontradas, los valores de estas fueron transformadas al valor medio de contaminación de la hora en cuestión. Por ejemplo, si la hora no válida es las 3 de la tarde, será reemplazado por el valor medio de contaminación de las 3 de la tarde del mes en el que era no válida. El formato horario del dataset se observa en la tabla [6].

	Año	Mes	Dia	Hora	N02
2185	2020	4	1	0	13
2186	2020	4	1	1	16
2187	2020	4	1	2	13
2188	2020	4	1	3	15
2189	2020	4	1	4	12
2190	2020	4	1	5	12

Figura 6: **Formato Dataset final horario**

En anexos [7.2] se adjunta código R a modo de ejemplo sobre cómo se ha realizado la limpieza de los datos ya explicada.

3.3. Análisis Descriptivo de los datos

En esta sección se van a explicar diferentes visualizaciones de los datos con distintas estacionalidades, con objetivo de entender de la mejor manera posible los datos que vamos a tratar más adelante.
Detectar patrones de comportamiento, tendencias, outliers u otras informaciones nos pueden ayudar a explicar las predicciones del modelo. A continuación se diferencian 4 secciones en las que se han aplicado distintos filtros para detectar posibles anomalías en distintas estacionalidades.

3.3.1. Serie Anual y Mensual

Los datos filtrados por Dióxido de Nitrógeno en Plaza Castilla datan de 2011 a 2020, teniendo 112 observaciones desde Enero 2011 a Abril 2020. En la gráfica adjunta [7] se observa una ligera tendencia decreciente con el transcurso de los años. Hay un claro efecto estacional. Con la ayuda de las etiquetas de los meses podemos observar a que meses corresponden los mínimos y máximos.

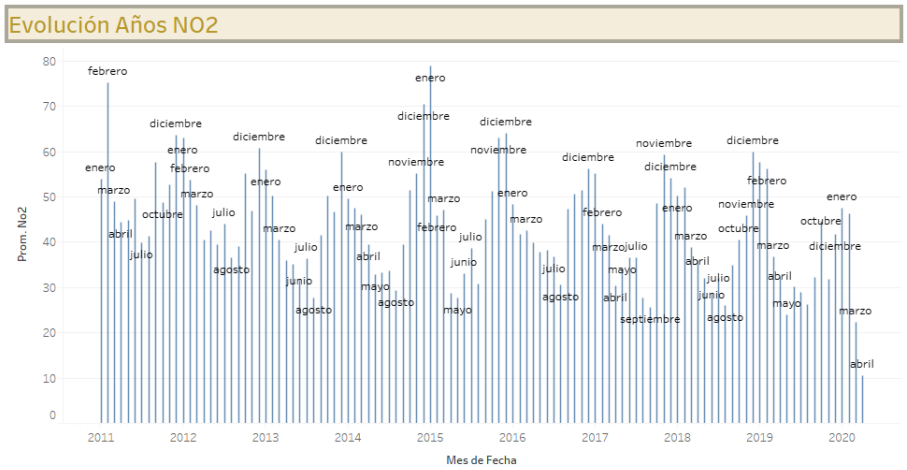


Figura 7: Gráfico Serie 2011 a 2020

Podemos observar que los meses que presentan mayor media de consumo de Dióxido de Nitrógeno representado en μ/cm^3 son los meses entre Noviembre y Febrero (de manera interactiva podríamos clickar sobre cada punto y obtener la información a la que corresponde dicha observación). En

cambio, los meses con menor cantidad de dióxido de Nitrógeno corresponden a los meses de verano.
Esto puede ser ocasionado porque la mayoría de trabajadores o estudiantes se van a sus ciudades natales o a segundas residencias. Pero seguramente la explicación más razonable, además de la demográfica, es que la mayoría de industrias en Agosto cierran o realizan servicios mínimos para poder dar descanso a los trabajadores. Por tanto, la estacionalidad es clara.

Para observar la variabilidad mensual, se adjunta un boxplot [8] asociado a cada mes que nos proporciona información sobre la cantidad media/mediana de dióxido de nitrógeno en dicho mes y su dispersión. Para los outliers, máximos y mínimos se indica el año al que esta asociada esa observación. Destacan sobre todo el año 2019 por los mínimos en los meses de Mayo,Julio,Noviembre o Diciembre. También destaca el mínimo global en Marzo y Abril de 2020 ocasionado por el estado de alarma acordado el día 15 de Marzo. Destacan outliers mensuales sobre todo en los años 2011,2015,2019 o 2020.

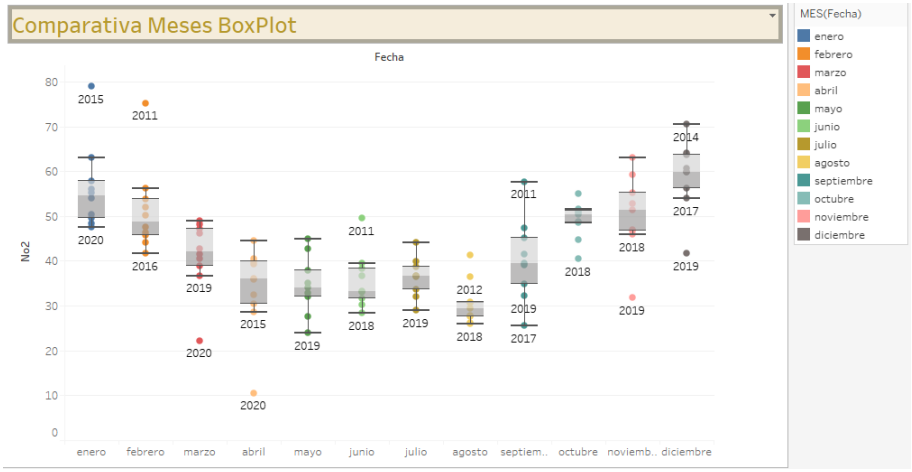


Figura 8: Gráfico Serie 2011 a 2020

Con objeto de destacar el descenso de dióxido de nitrógeno, se observa en la siguiente tabla resumen [9] la cantidad media de dióxido de nitrógeno por año con un mínimo en 2019 de 36.92. De hecho, si hacemos una media de la contaminación de los 8 años anteriores obtenemos un valor de 44.5 μ/cm^3 de NO2. Esto quiere decir que se ha reducido entre un 17 y un 18 % la cantidad de dióxido de nitrógeno específicamente en la zona que estamos evaluando(Plaza Castilla). Estos datos corroboran lo afirmado en un artículo de el País [36], aunque en aquel estudio no se incluía en el análisis el último mes de Diciembre de 2019, ni se refería concretamente a la Plaza Castilla.

Fecha									
2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
51,74	47,49	42,65	44,01	46,18	43,47	41,12	39,72	36,92	31,65

Figura 9: Tabla Resumen NO2 Promedio de 2011 a 2020

3.3.2. Serie Diaria

Para analizar los datos diarios se han filtrado los años 2015 a 2020, que datan desde el 1 de enero de 2015 al 30 de Abril de 2020.

En la figura [10] nos encontramos diagramas de caja para cada día de la semana con las observaciones a lo largo de los 5 años. Se puede observar que muchos de los máximos diarios se encuentran a principios del año 2015.

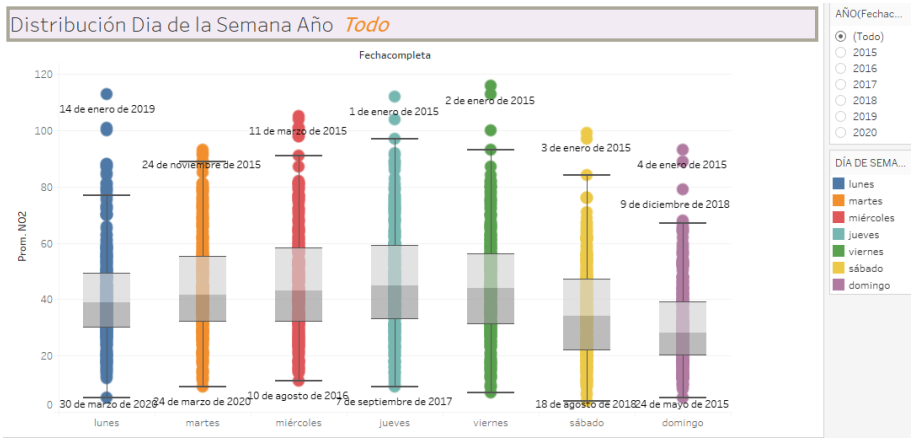


Figura 10: Distribución Semanal NO2 2015-2020

Esta información se vio reflejada en el periódico la Vanguardia [37].En este artículo, la asociación **Ecologistas en Acción** apuntaba: 'la situación es tal que en los doce primeros días del 2015 seis estaciones de Madrid superan ya el límite legal de contaminación del aire para todo el 2015".

De manera más genérica sorprende una diferencia del 10 % de dióxido de nitrógeno entre los lunes y los miércoles,jueves o viernes, los cuales son los días con mayor contaminación media de NO2 por encima de 44 μ/cm^3.
Como era previsible, los sábados tienen de media un 20 % menos que la media de los días laborables y los domingos un 31 % de media menos que aquellos. Podemos concluirlo observando la tabla resumen media semanal [11].

Fechacompleta

lunes	martes	miércoles	jueves	viernes	sábado	domingo
41,13	43,85	45,21	45,92	44,59	34,99	30,22

Figura 11: Contaminación media por día de 2015 a 2020

3.3.3. Serie Horaria

Para visualizar los datos horarios se han promediado los datos del año 2019, contando con 8744 observaciones. En la figura [12] se presenta un gráfico de barras con la media para cada hora del día. Los mínimos se encuentran en las horas nocturnas de 2 a 7.

No obstante, destacar que hay una gran diferencia entre las 12 y las 4 de la tarde comparando con respecto a la madrugada o la noche. En concreto una diferencia de más del 40 % entre esta

franja horaria y las 10 de la noche con una cantidad media de casi 57 μ/cm^3 .

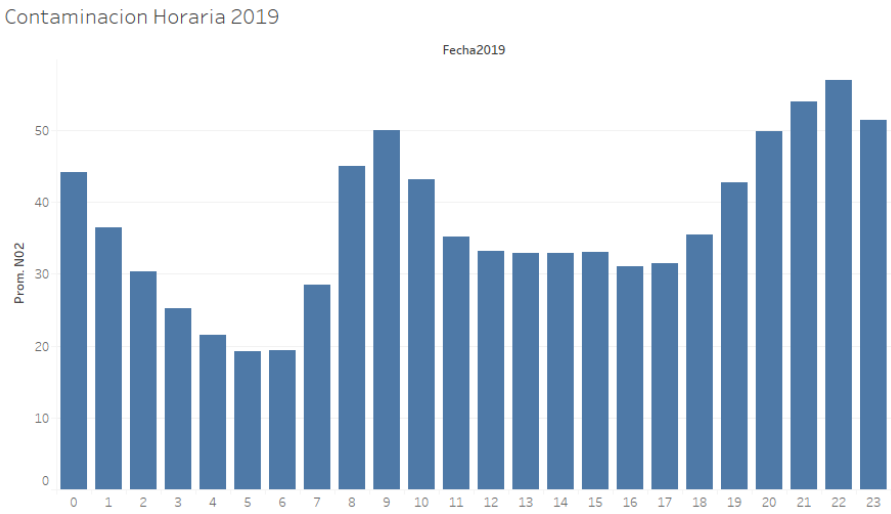


Figura 12: Distribución Horaria 2019

Los datos de la tabla [13] recogen las medias representadas en el diagrama de barras para comparar exactamente diferencias entre cada franja horaria.

Contaminacion Horaria 2019

Fecha2019											
0	1	2	3	4	5	6	7	8	9	10	11
44,15	36,42	30,30	25,15	21,54	19,16	19,44	28,48	45,04	49,98	43,10	35,16
12	13	14	15	16	17	18	19	20	21	22	23
33,15	32,90	32,95	33,06	31,10	31,42	35,47	42,69	49,84	53,92	56,96	51,43

Figura 13: Tabla Resumen Horaria 2019

3.3.4. Marzo y Abril 2020

Esta sección se incluye motivada por los acontecimientos que han tenido lugar derivados de la pandemia provocada por el virus Covid19 y el estado de alarma declarado a mediados de marzo. Una de las pocas noticias positivas de este virus es el descenso de dióxido de nitrógeno expuesto en ciudades como Madrid.

Es reseñable como los días anteriores a establecer el estado de alarma, se registraron 60-70 μ/cm^3. Sin embargo con el cierre de los colegios el día 12 de marzo ya descendió a 50 μ/cm^3 y el viernes día 13 de marzo descendió más de 10 puntos. Si hacemos una comparación de los días 10 y 11 de Marzo frente a días de confinamiento el descenso es mayor del 80 %.

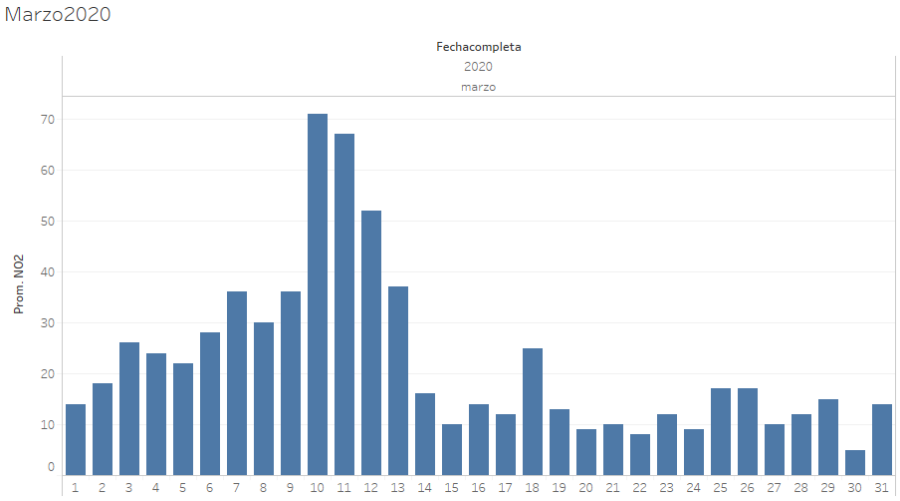
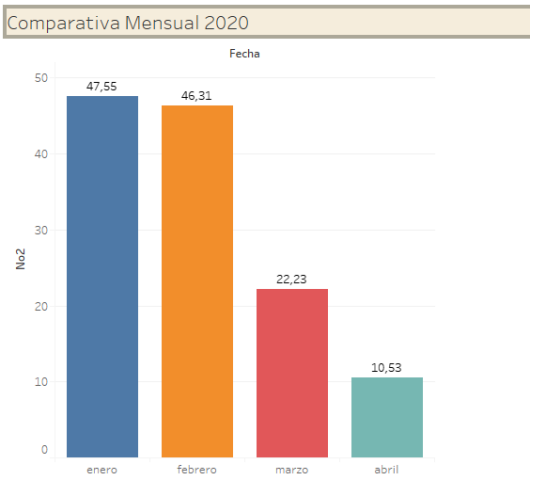


Figura 14: Contaminación NO2 diaria marzo 2020

Aunque el confinamiento afectó solo a 15 días de Marzo, hubo una reducción del 50 % de NO2 en ese mes (gráfica [15a]), lo que dejaba imágenes de Madrid 'antes no vistas' producidas por la contaminación.
Más reducción hubo en Abril, en concreto del 50 % más que en Marzo, lo que lleva a una media de 10.53 μ/cm^3, cifras nunca vistas en los últimos 10 años de estudio investigadas en este trabajo



(a) **Contaminación Mensual 2020**

(b) **Contaminación NO2 Abril 2011-2020**

Figura 15: **Contaminación comparativa Estado de Alarma**

Por último, se observa una diferencia muy sustancial en la gráfica [15b] dónde los valores de NO2 se redujeron entre el 60 % y el 80 % entre los años 2011 y 2019 con respecto a 2020.

4. Modelo autorregresivo integrado de media móvil(ARIMA)

Este tipo de modelos que caracterizan las series por sumas o diferencias de las series resultantes fue propuesto por Yule y Slutzsky en la década de los 20. Esto sería la base para los procesos autorregresivos y de media móvil que tuvieron un mayor impacto con la publicación en 1970 del libro de Box-Jenkins sobre modelos ARIMA [38].

El objetivo principal de los modelos ARIMA es realizar predicciones en base a la información que se obtiene de los datos del pasado. Es decir, estos modelos tienen en cuenta la dependencia temporal de los datos, de modo que cada observación depende de los valores anteriores.[39]
Permiten describir un valor de la serie como una función lineal de los datos anteriores y de los errores aleatorios. Box-Jenkins recomendaba más de 50 observaciones de la serie temporal, lo que cumplen las dos series de este trabajo.

En concreto, los modelos ARIMA se aplican a series univariantes, dónde el objetivo es encontrar un modelo que represente las características de la serie de un modo preciso.Antes de entrar a explicar los modelos ARIMA, se comenzará por caracterizar los procesos por los que están formados, los modelos autoregresivos y modelos de medias móviles [38].

Por último se detalla una forma compacta para formular los modelos ARIMA de aquí en adelante. Se define un operador de retardos, como B, que asocia cada variable X_t con la variable en el instante anterior X_{t-1} tal que : $BX_t = X_{t-1}$

4.1. Modelos Autoregresivos(AR)

En estos modelos cada observación se puede expresar como una regresión sobre observaciones anteriores. Estos modelos se basan en la idea de que el valor actual de la serie puede explicarse o predecirse en función de valores pasados más un término de error. El orden del modelo expresa el número de observaciones atrasadas de la serie. A continuación se detalla un ejemplo sobre un AR(1) o un AR(p):

AR(1)

- Fórmula estándar:

$$X_t = \phi X_{t-1} + a_t \tag{1}$$

- Formula con retardos:

$$(1 - \phi \mathbf{B}) X_t = \mathbf{a}_t \tag{2}$$

AR(P)

- Fórmula estándar:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + a_t \tag{3}$$

- Formula con retardos:

$$\left(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p\right) X_t = a_t \tag{4}$$

4.2. Modelos de Medias Móviles(MA)

Los modelos de medias móviles son aquellos que explican el valor de una determinada variable en un tiempo t en función de un término independiente y una sucesión de errores correspondientes a periodos precedentes.A continuación se detalla un ejemplo sobre MA(1) o MA(q):

MA(1)

- Fórmula estándar:

$$X_t = a_t - \theta a_{t-1} \tag{5}$$

- Fórmula con retardos:

$$X_t = (1 - \theta B)a_t \tag{6}$$

MA(q)

- Fórmula estándar:

$$X_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots - \theta_q a_{t-q} \tag{7}$$

- Fórmula con retardos:

$$X_t = \theta(B)a_t \tag{8}$$

4.3. Modelos ARMA y ARIMA

Existen dos modos generales de enfocar las series temporales comenzando con un punto de vista determinista como puede ser el *suavizado exponencial* , y desde un punto de vista estocástico como los modelos que vamos a explicar a continuación (ARMA y ARIMA).

Una extensión natural de los modelos AR(p) y MA(q) es un tipo de modelos que incluyen tanto términos autorregresivos como de medias móviles y se definen como ARMA(p,q).

La formulación de los modelos ARMA(p,q) y ARIMA(p,d,q) se diferencian únicamente en la diferenciación (componente integrada) utilizada para conseguir que una serie sea estacionaria y así poder modelizarla mediante un modelo ARMA. La formulación de los modelos ARMA y ARIMA es la siguiente:

ARMA(p,q)

- Formulación estándar:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q} \tag{9}$$

- Formulación con retardos:

$$\left(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p\right) X_t = \left(1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q\right) a_t \tag{10}$$

$$\phi(B)X_t = \theta(B)a_t \tag{11}$$

ARIMA(p,d,q)

Estos modelos se les ha añadido una componente integrada(I) formando los modelos ARIMA(Autoregressive Integrated Moving Average). Éste está compuesto del componente autoregresivo (AR),la componente integrada(I) y del componente de media móvil (MA).

- Fórmula con retardos:

$$\phi(B)(1-B^d)X_t = \theta(B)a_t \tag{12}$$

Tanto en el modelo ARMA como el ARIMA , ϕ corresponde al polinomio en B de la parte autorregresiva y θ corresponde al parámetro de la media móvil. La **d** corresponderá al número de diferencias que se aplica a la serie.

4.4. Modelos ARIMA estacionales

Este tipo de modelos se caracterizan por la existencia de un patrón estacional en los datos. Las series con tendencia y variaciones cíclicas pueden representarse como modelos SARIMA o ARIMA estacionales: SARIMA(p,d,q)(P,D,Q)s . Primer paréntesis se refiere a la parte regular y el segundo a la parte estacional de la serie temporal junto con la s que determina el periodo de dicha serie. A continuación se expone la fórmula con el operador de retardos:

$$\phi_p(B)\Phi_P\left(B^s\right)(1-B)^d\left(1-B^s\right)^D X_t = \theta_q(B)\Theta_Q\left(B^S\right)a_t \tag{13}$$

Las componentes del modelo se detallan a continuación:

- $$\phi_p(B) = 1-\phi_1 B-\phi_2 B^2-\ldots-\phi_p B^p \tag{14}$$
- $$\Phi_P\left(B^s\right) = 1-\Phi_1 B^s-\Phi_2 B^{2s}-\ldots-\Phi_P B^{sP} \tag{15}$$
- $$\theta_q(B) = 1-\theta_1 B-\theta_2 B^2-\ldots-\theta_q B^q \tag{16}$$
- $$\Theta_Q\left(B^S\right) = 1-\Theta_1 B^s-\Theta_2 B^{2s}-\ldots-\Theta_Q B^{sQ} \tag{17}$$

Donde x_t es la serie observada y a_t variables de ruido blanco.Los subíndices p,P,q,Q indican el grado del polinomio bien en B o en B^s. Por último el superíndice d indica el grado de diferenciación simple y D el estacional.

5. Enfoque Box-Jenkins para datos mensuales y horarios

La metodología de Box-Jenkins fue nombrada en honor a los estadísticos George E.P.Box y Gwilym Jenkins. Ésta es aplicada a los modelos autorregresivos de media movil (ARMA) o a los modelos autorregresivos integrados de media móvil(ARIMA) o en su defecto estacionales(SARIMA), con el objetivo de encontrar el mejor ajuste a unos datos de modelo temporal univariante.

Esta metodología utiliza una serie de fases iterativas, las cuales se describen a continuación, detallando como las llevaremos a cabo con nuestros datos:

- Identificación
  - Detectar Serie estacionaria o no estacionaria y actuar en consecuencia.
  - Estacionariedad en Media y Varianza
  - Detectar Tendencia.
  - Detectar Estacionalidad.
  - Identificar modelos con ACF y PACF
  - Test Estadisticos para estacionariedad
- Estimación: Estimar los parámetros de los modelos escogidos en base a los resultados en la etapa de identificación. Se puede realizar por máxima verosimilitud o mínimos cuadrados. Tener en cuenta la sobreparametrización. Se realizarán constrastes de hipotesis sobre los parámetros estimados.
- Validación : Evaluar si los modelos estimados se ajustan bien a los datos antes de realizar predicción
  - Análisis Residuos
  - Media y Varianza constante en el tiempo
  - Realización ACF residual
  - Test Ljung-Box
- Comparación de modelos
- Predicción: Conocido el modelo de la serie temporal escogido, predecir valores futuros.

5.1. Modelización ARIMA datos mensuales contaminación en Madrid

Los datos que se van a utilizar en la práctica son:

- Datos de entrenamiento: Enero 2011 a Febrero 2018; 87 observaciones
- Datos de test: Marzo 2018 a Febrero 2020; 24 observaciones(2 periodos completos)

Uno de los objetivos iniciales es observar gráficamente la serie y poder detectar la tendencia de la serie. Para ello se utilizarán unos datos de entrenamiento,para posteriormente, intentar predecir con los datos de test utilizando el mejor modelo encontrado.

Observando la serie, es necesario detectar si es **estacionaria**. Para ello, hay que detectar si es necesario una transformación de Box-Cox (o logarítmica) que permita estabilizar la serie debido a la varianza no es constante. De acuerdo a la gráfica de la serie [16] no parece que sea necesario una transformación.

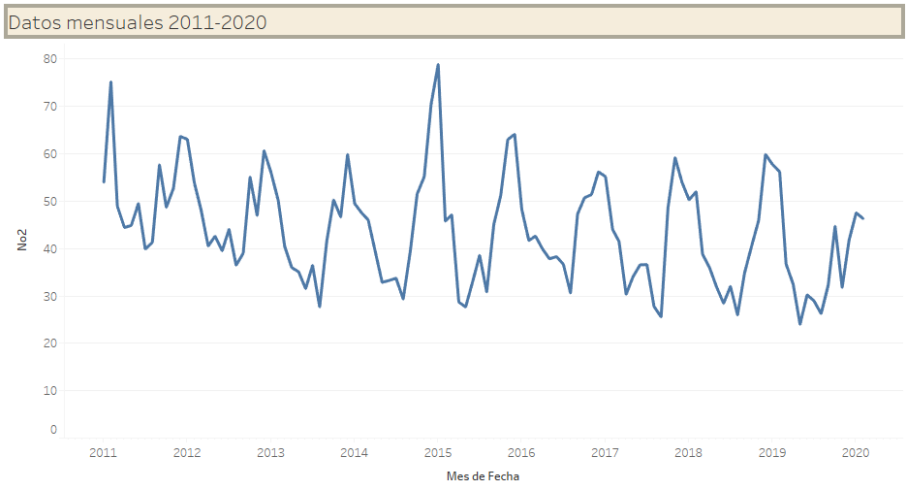
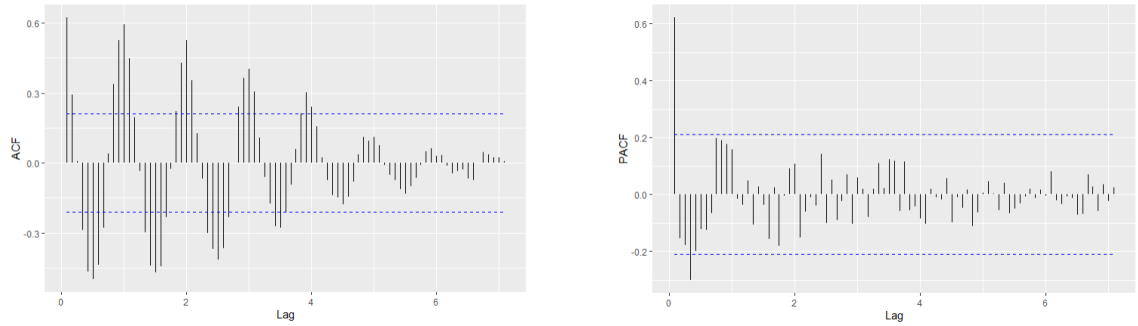


Figura 16: Gráfica Serie Enero 2011 - Febrero 2020

Con la información aportada en [16] se concluye que existe una correlación estacional clara, así como una tendencia decreciente pronunciada de 2011 a 2015 y posterior decrecimiento menos brusco hasta 2018. *Con esta tendencia marcada se declara que la serie no es estacionaria y necesitará de diferenciación.*

Para cercionarnos de esta conclusión, es necesario la realización de las funciones de autocorrelación muestral(ACF) y las funciones de autocorrelación parcial muestral(PACF). Estas nos sirven para examinar la dependencia de la serie y para detectar qué valores del pasado tienen correlación con la observación actual.

Para ello, se utilizan los ACF y PACF [17a] de la serie para decidir el mínimo orden de diferenciación que haga que la serie sea estacionaria.



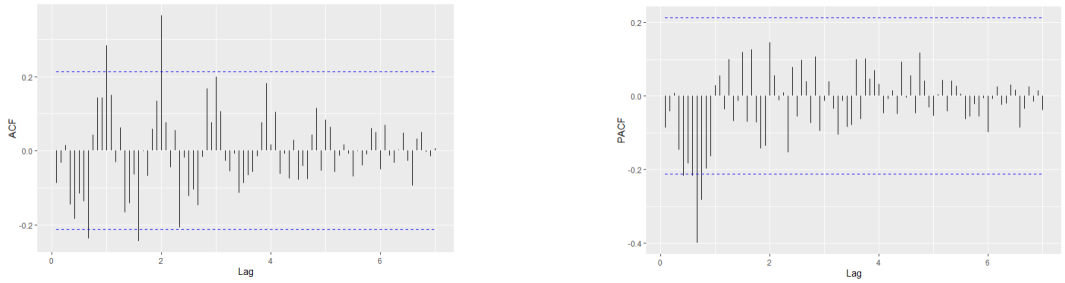
(a) **ACF Serie Original** (b) **PACF Serie Original**

Figura 17: ACF y PACF serie original

Advertimos que en el eje X donde observamos 1,2,3,4.. realmente representan los retardos 12,24,36,48... porque para crear una serie temporal en R inicialmente se indica el periodo de la serie y lo 'comprime entre el periodo estacional'.

El primer indicio al observar el ACF es que la parte estacional decrece muy lentamente, por lo que indica que la serie no es estacionaria y necesitará una diferenciación estacional. Al ser una diferenciación estacional de periodo 12, se diferencia no con la observación anterior sino con 'la del mismo mes de otro año'.

En cuánto a la parte regular parece estacionaria, por lo que será necesario únicamente una diferenciación estacional para convertir la serie en estacionaria. Se diferencia la serie estacionalmente obteniendo los dos gráficos de autocorrelación siguientes [18a] y [18b]:

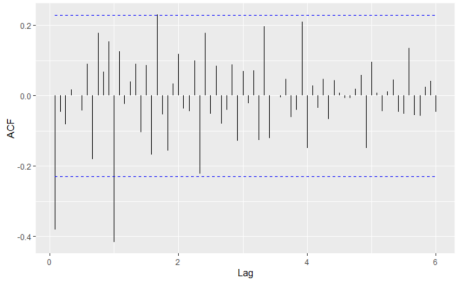


(a) **ACF Serie Diferenciada Estacionalmente** (b) **PACF Serie Diferenciada Estacionalmente**

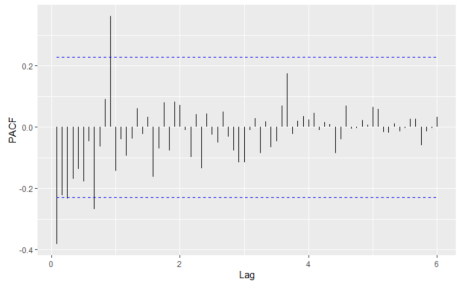
Figura 18: ACF y PACF serie diferenciada estacionalmente

Con la diferenciación estacional se observa que el decrecimiento lento estacional ha pasado a un decrecimiento exponencial, por lo que la serie parece ser estacionaria a priori.

No obstante, se aplica una diferenciación regular para comprobar si la serie tiene un comportamiento mejor en la parte regular. Los gráficos resultantes son [19a] y [19b]:



(a) **ACF Serie Diferenciada Regular y Estacionalmente**



(b) **PACF Serie Diferenciada Regular y Estacionalmente**

Figura 19: ACF y PACF serie diferenciada regular y estacionalmente

A la vista de los resultados no parece haber una mejora significativa. A continuación, se comparan varianzas para terminar de discutir el número de diferenciaciones a aplicar.

```
     Serie Inicial Serie Dif Regular Serie Dif Estacional Serie Dif Regular y Estac
[1,]      122.1209          91.98825             69.32941                  116.3666
```

Figura 20: Comparación de Varianzas

Con los datos resultantes, se concluye como la mejor solución,la diferenciación estacional, ya que tiene el 50 % menos de varianza que si se aplica una diferencia regular adicional.

Para obtener resultados partiendo de test estadísticos, se realiza uno específico encargado de medir si la serie es estacionaria o no. Este es el de Dickey-Fuller Ampliado(ADF).
La hipótesis nula asociada de este test es que la serie no es estacionaria, por tanto será beneficioso obtener un pvalor pequeño para rechazar la hipótesis nula con un nivel de significación alto.

Los resultados del test con la serie inicial[50a] y la serie resultante tras diferenciar [50b] corroboran que la serie inicialmente no era estacionaria, obteniendo un pvalor de 0.4, pero con las diferenciaciones pertinentes, se consigue rechazar la hipótesis nula con un nivel de significación alto, al ser el pvalor de 0.01.
Con la serie estacionaria, el objetivo se basa en determinar que posibles modelos ARIMA pueden corresponderse con los resultados de las ACF y PACF con la serie diferenciada estacionalmente [18a] y [18b].

Tras observar la función de autocorrelación muestral se observa que desde el primer retardo estacional decrece exponencialmente los siguientes retardos estacionales. Si nos fijamos en la PACF el decrecimiento también es llamativo pero los siguientes retardos estacionales no son tan cercanos a cero sino que van decreciendo algo más lento. Se podría decir que la parte estacional corresponde con una media móvil de orden 1, es decir MA(12). Sin embargo no se desecha la posibilidad de que sea un modelo autorregresivo de orden 1 o ambos.
Para la parte regular es mas complejo determinarlo. Los autocorrelaciones del ACF están más cerca del 0 que en la PACF por lo que podría ser un modelo de media móvil de orden 1. Sin embargo, de nuevo no se pueden sacar conclusiones certeras más allá de la diferenciación estacional realizada anteriormente.

Los modelos iniciales probados en la etapa de estimación son:

- **ARIMA(0,0,1)(1,1,0)12**

$$(1 - \Phi_1 B^{12})(1 - B^{12})X_t = (1 - \theta_1 B)a_t \tag{18}$$

- **ARIMA(1,0,0)(1,1,0)12**

$$(1 - \phi_1 B)(1 - \Phi_1 B^{12})(1 - B^{12})X_t = a_t \tag{19}$$

- **ARIMA(1,0,0)(0,1,1)12**

$$(1 - \phi_1 B)(1 - B^{12})X_t = (1 - \Theta_1 B^{12})a_t \tag{20}$$

- **ARIMA(0,0,1)(0,1,1)12**

$$(1 - B^{12})X_t = (1 - \theta_1 B)(1 - \Theta_1 B^{12})a_t \tag{21}$$

Los parámetros son estimados por el método de **máxima verosimilitud**. En general los estimadores máximo verosímiles son más precisos que los mínimos cuadráticos aunque requieren más cálculo. Para poder comparar cada estimador con su error estándar se realiza el cociente de ambos que sigue una distribución t de student con n-(p+q) grados de libertad bajo la hipótesis de que el parámetro sea cero.

Además, con la ayuda de la matriz de correlaciones se podrá determinar si existe sobreparametrización cuando haya una correlación alta entre dos parámetros. A continuación se realiza un breve análisis sobre el ajuste/estimación de los distintos modelos.

Estimaciones máximo verosímiles de los modelos

```
Series: serie
ARIMA(0,0,1)(1,1,0)[12]

Coefficients:
         ma1     sar1
      0.2335  -0.4698
s.e.  0.1038   0.1143

sigma^2 estimated as 56.29:  log likelihood=-254.64
AIC=515.29   AICc=515.63   BIC=522.2

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ma1   0.23349    0.10379  2.2497   0.02447 *
sar1 -0.46981    0.11428 -4.1111 3.938e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) **Resultados ARIMA(0,0,1)(1,1,0)12**

```
Series: serie
ARIMA(1,0,0)(1,1,0)[12]

Coefficients:
         ar1     sar1
      0.2982  -0.5006
s.e.  0.1157   0.1152

sigma^2 estimated as 54.79:  log likelihood=-253.89
AIC=513.79   AICc=514.13   BIC=520.7

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1   0.29818    0.11572  2.5768  0.009972 **
sar1 -0.50065    0.11518 -4.3465 1.383e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(b) **Resultados ARIMA(1,0,0)(1,1,0)12**

```
Series: serie
ARIMA(1,0,0)(0,1,1)[12]

Coefficients:
         ar1     sma1
      0.2708  -0.6049
s.e.  0.1135   0.1481

sigma^2 estimated as 52.3:  log likelihood=-253.16
AIC=512.33   AICc=512.67   BIC=519.24

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1   0.27082    0.11352  2.3855   0.01705 *
sma1 -0.60491    0.14809 -4.0848 4.411e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) **Resultados ARIMA(1,0,0)(0,1,1)12**

```
Series: serie
ARIMA(0,0,1)(0,1,1)[12]

Coefficients:
         ma1     sma1
      0.2315  -0.5899
s.e.  0.1078   0.1463

sigma^2 estimated as 53.23:  log likelihood=-253.64
AIC=513.28   AICc=513.62   BIC=520.19

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ma1   0.23153    0.10779   2.148   0.03171 *
sma1 -0.58991    0.14627  -4.033 5.506e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(b) **Resultados ARIMA(0,0,1)(0,1,1)12**

El resultado aportado de los cuatro modelos indican un buen ajuste del modelo observando la significación de los parámetros. Del mismo modo, en los resultados correspondientes a las matrices de correlación: [51a],[51b],[52a] y [52b] no se encuentra en ninguno de los cuatro modelos una matriz de correlaciones con valores altos.

Estimación ARIMA automático librería forecast

$$(1 - B^{12})X_t = (1 - \Theta_1 B^{12})a_t \tag{22}$$

Se ha realizado un análisis automático con una función(auto.arima() de la libreria forecast) que encuentra el mejor modelo ARIMA para los datos otorgados. No obstante, este tiende a buscar en función del AIC más bajo y sin tener en cuenta los residuos , por lo que es posible que dicho modelo no sea el mejor para los datos.

El ajuste de este modelo,sugiere que con la estimación del parámetro de media móvil por la parte estacional con la constante del modelo es más que suficiente. Los resultados son satisfactorios tanto en el test de Wald [23a] como en la matriz de correlaciones [23b].

```
Series: serie
ARIMA(0,0,0)(0,1,1)[12] with drift

Coefficients:
         sma1    drift
      -0.7961  -0.1062
s.e.   0.2416   0.0295

sigma^2 estimated as 45.75:  log likelihood=-251.23
AIC=508.46   AICc=508.81   BIC=515.38

z test of coefficients:

       Estimate Std. Error z value  Pr(>|z|)
sma1  -0.796088   0.241585 -3.2953 0.0009833 ***
drift -0.106170   0.029486 -3.6007 0.0003173 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) **Resultados ARIMA(0,0,0)(0,1,1)12 con constante**

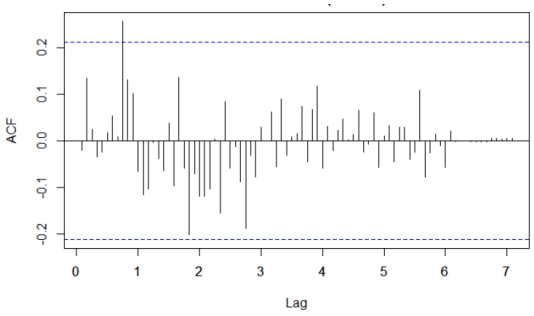
	sma1	drift
sma1	1.00000000	-0.08065356
drift	-0.08065356	1.00000000

(b) **Matriz de correlaciones ARIMA(0,0,0)(0,1,1)12**

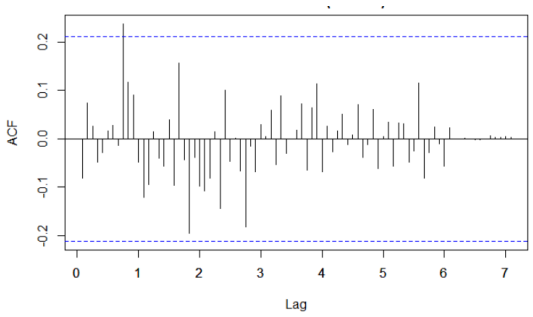
Figura 23: Ajuste del modelo ARIMA automático

A la vista de los resultados se concluye que no hay una elección clara sobre cual es el mejor modelo en la estimación ya que los resultados eran bastante similares. Por tanto, se procede a la siguiente etapa de validación de los residuos con los modelos estimados.

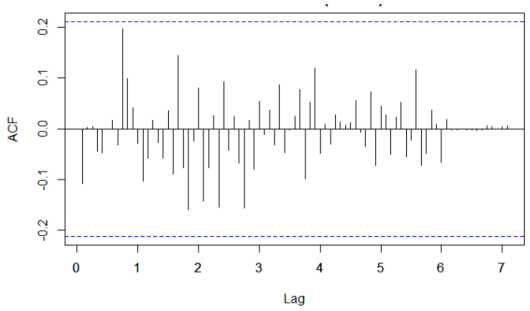
Uno de los métodos más eficientes para comprobar como se comportan los residuales es calcular ACF de los residuales y observar que valor obtienen las primeras autocorrelaciones así como las autocorrelaciones del periodo. A continuación se muestran los ACF residuales para los 4 modelos iniciales:



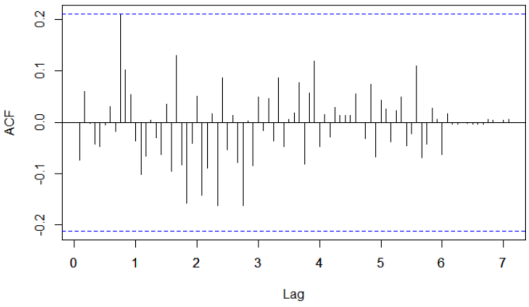
(a) **ACF Residuales ARIMA(0,0,1)(1,1,0)12**



(b) **ACF Residuales ARIMA(1,0,0)(1,1,0)12**



(a) **ACF Residuales ARIMA(1,0,0)(0,1,1)12**



(b) **ACF Residuales ARIMA(0,0,1)(0,1,1)12**

A la vista de los resultados del ACF de residuales, sólo destaca el valor de la segunda autocorrelación del modelo ARIMA(0,0,1)(1,1,0)12.

A continuación se adjunta el ACF de residuales del modelo ofrecido por auto.arima():

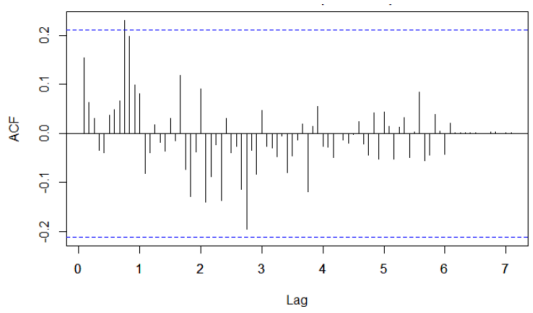


Figura 26: ACF Residuales ARIMA(0,0,0)(0,1,1)12

A la vista de los resultados de los ACF de residuales, se desestima el último modelo ARIMA(0,0,0)(0,1,1) por la fuerte autocorrelación del primer retardo. Del mismo modo, tampoco se procede a seguir usando el primer modelo ARIMA(0,0,1)(1,1,0)12 debido a la alta autocorrelación del segundo retardo.

Para determinar si los residuales son o no incorrelados se usa el test de Ljung-Box.

En las imágenes siguientes se adjuntan las salidas para el test utilizando los residuales hasta el retardo 70 para los 3 modelos restantes: [53a], [53b] y [53c]. Los resultados de los tres modelos son el no rechazo de la hipótesis nula, lo que indica que los residuos del modelo no están correlados.

Para comparar el grado de ajuste de los tres modelos se utiliza el AIC para selección del modelo más adecuado. Se denominan modelo 2: ARIMA(1,0,0)(1,1,0)12; modelo 3: ARIMA(1,0,0)(0,1,1)12 ; modelo 4: ARIMA(0,0,1)(0,1,1)12. Los resultados se ofrecen en la siguiente imagen:

Aic Modelo2	Aic Modelo3	AIC Modelo4
513.7894	512.3272	513.2816

Figura 27: Comparación AIC

Se determina que los dos mejores modelos que se ajustan a los datos son los modelos 3 y 4. Por tanto se realiza la predicción para ambos modelos. Es recomendable comparar la predicción de los 24 meses siguientes y compararlos con los datos reservados para certificar que el modelo se ajusta a los datos. Para ello se hace uso de la suma de cuadrados del error(SSE).

Para observar cómo se comportan los datos de prueba, se representan los datos reales de contaminación de los meses de Marzo de 2018 a Febrero de 2020.

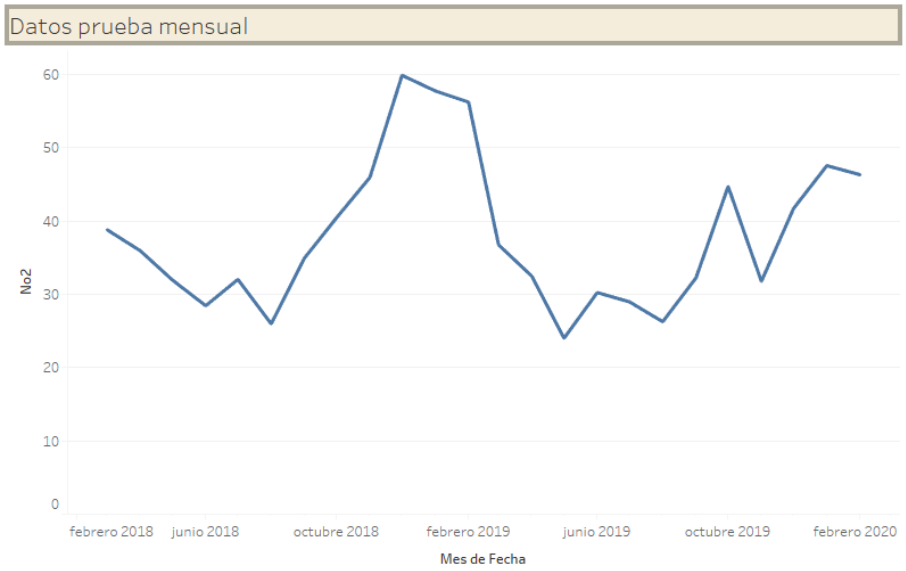


Figura 28: Datos de prueba marzo 2018-febrero 2020

Se ha realizado una comparación de la predicción de los dos modelos con respecto a los datos reales. Los puntos representan los datos reales mientras las otros tres gráficas (con trazo lineal) los distintos modelos, siendo:

- arima3: ARIMA(1,0,0)(0,1,1)12
- arima4: ARIMA(0,0,1)(0,1,1)12

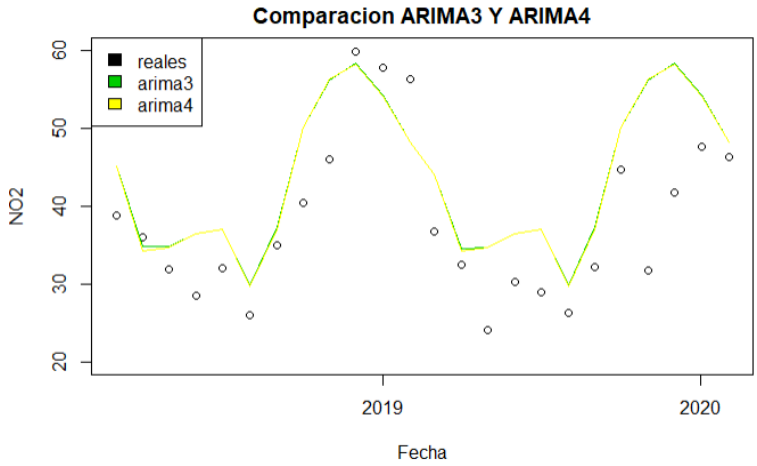


Figura 29: Comparación Predicciones Test con Reales

Se puede observar cómo la predicción de los modelos arima3 y arima4 son prácticamente similares.Para comprobar cual tiene menor error en la predicción se calcula el *SSE* de los dos modelos:

	ARIMA(1,0,0)(0,1,1)12	**ARIMA(0,0,1)(0,1,1)12**
SSE	1698.002	1688.106

Se determina por una diferencia mínima que el modelo ARIMA(0,0,1)(0,1,1)12 tiene mejor capacidad de predicción. Por ello, se ha realizado para este modelo una comparativa con los valores predichos y sus intervalos de confianza en relación a los datos reales:

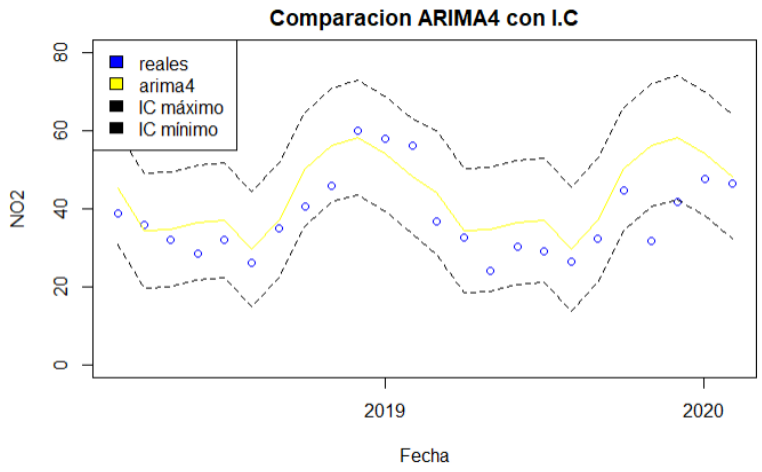


Figura 30: Comparación predicciones modelo final con intervalos

En la figura [30] se observa como casi todos los valores reales se encuentran dentro de la banda de confianza. Los valores de NO_2 de Noviembre y Diciembre de 2019 fueron 'anómalos' ya que son los meses del año con más contaminación pero en este caso se redujo sustancialmente cómo se ve en la gráfica [28]. No obstante, salvo esos dos meses atípicos, la predicción fue buena.

En la figura [31], se observa cómo el modelo fue capaz de ajustarse al comportamiento de la serie y los 24 meses siguiente los predijo de manera similar a los datos de entrenamiento.

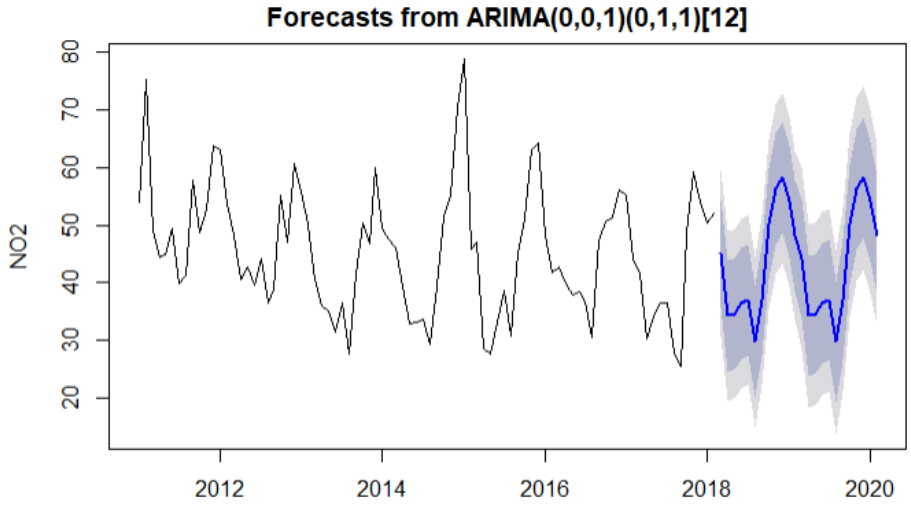


Figura 31: Comparación predicciones modelo final con intervalos

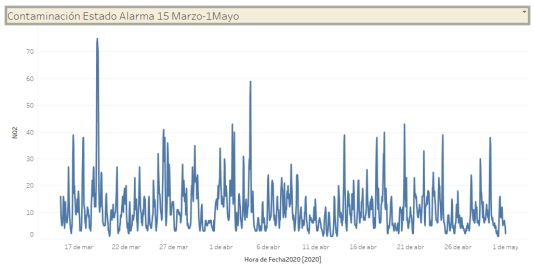
5.2. Modelización ARIMA datos horarios contaminación en Madrid

Las observaciones asociadas a NO_2 de la plaza castilla de manera horaria durante parte del estado de alarma se han distribuido en datos de entrenamiento y de test tal que:

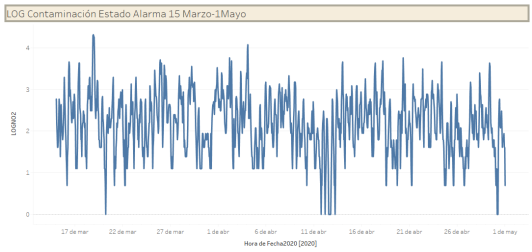
- Datos de entrenamiento: 15 Marzo- 22 Abril: 936 observaciones
- Datos de Prueba: 23 Abril- 29 Abril: 168 observaciones

Al igual que en el análisis mensual, se va a tratar de observar las características de la serie. Observando la gráfica [32a] parece haber una tendencia estable con el tiempo en la que en algunas horas hay picos de consumo. También son reseñables dos outliers con un valor de contaminación de NO2 muy alto.

Antes de entrar a realizar las funciones de autocorrelación, es necesario comprobar si es necesario la realización de una transformación logarítmica a los datos. Viendo la gráfica de la serie parece que la varianza no es constante a lo largo del tiempo por lo que se realiza la transformación, viendo el resultado en la gráfica [32b].



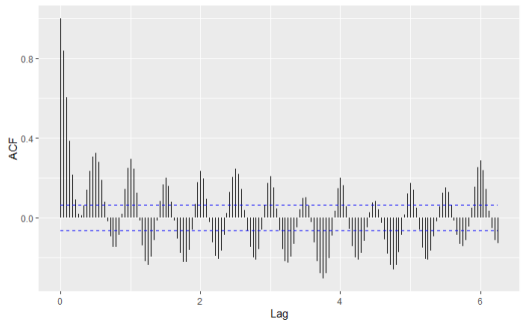
(a) Serie 15 marzo - 1 mayo



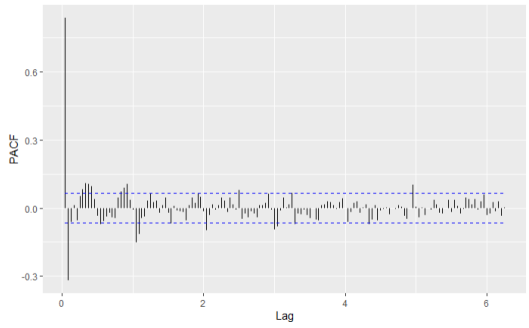
(b) Logaritmo Serie 15 marzo - 1 mayo

Figura 32: Serie original y logarítmica datos horarios

Además, observando la serie, se necesita aplicar una diferenciación. Para ello hay que observar el decrecimiento de las autocorrelaciones en la ACF.



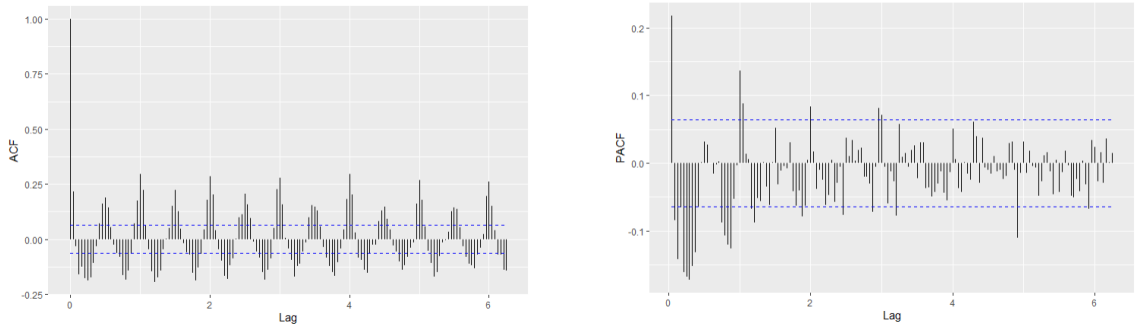
(a) **ACF serie original**



(b) **PACF serie original**

Figura 33: Serie horaria original y logarítmica

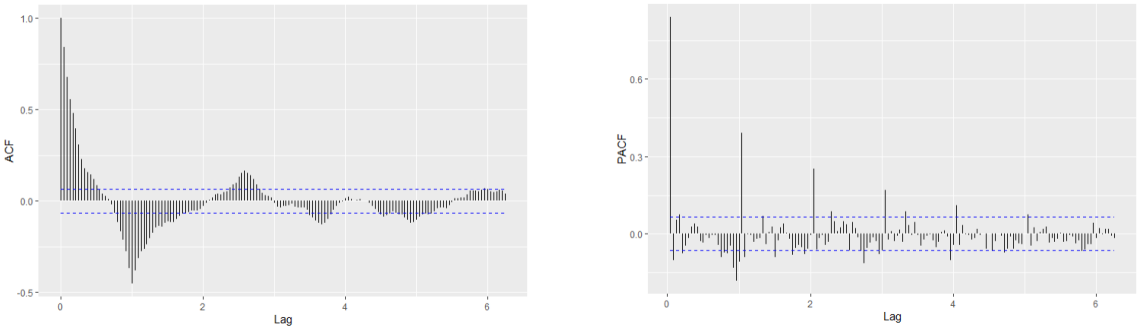
Como se ve en la figura [33a] el decrecimiento por la parte regular y por la parte estacional es muy lento. Por ello, se estima oportuno realizar una diferenciación regular.



(a) **ACF diferenciada regularmente** (b) **PACF diferenciada regularmente**

Figura 34: ACF y PÀCF diferenciada regularmente

Si observamos la gráfica [34a] se observa un decrecimiento por ondas por la parte regular. Sin embargo la parte estacional no parece haber sido alterada, por lo que sigue sin ser estacionaria la serie. Por tanto, se realiza una diferenciación estacional de 24 retardos para ver como se comporta la serie, observando las gráficas [35a] y [35b].

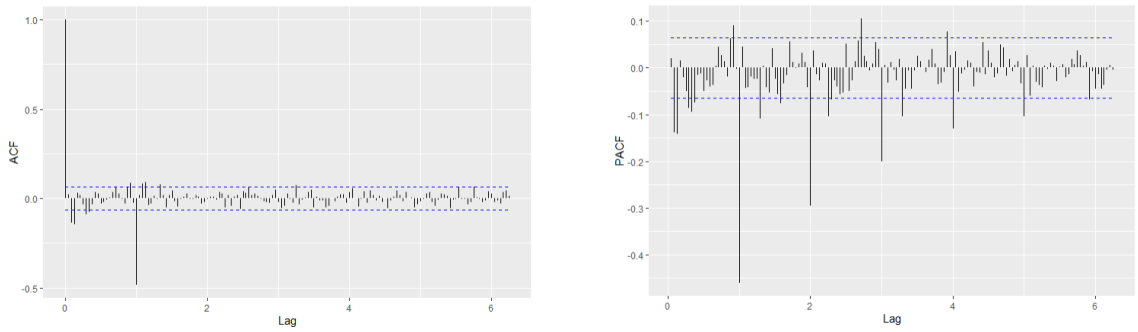


(a) **ACF diferenciada estacionalmente 24 retardos** (b) **PACF diferenciada estacionalmente 24 retardos**

Figura 35: ACF y PACF diferenciada estacionalmente

En este caso, la diferencia estacional ha dado como resultado un decrecimiento exponencial en la parte estacional mientras que la parte regular decrece lento.

Con los resultados obtenidos hasta el momento, no se ha podido determinar que la serie sea estacionaria. Por ello, se realiza una diferenciación regular y estacional, observando las gráficas resultantes en las figuras [36a] y [36b].



(a) **ACF** (b) **PACF**

Figura 36: ACF y PACF diferenciados regularmente (1) y estacionalmente(24)

Con la realización de las diferenciaciones regular y estacional, se observa en la figura [36a] que existe un decrecimiento más rápido y por tanto se considera la serie estacionaria.

Con el objetivo de hacer un análisis global de las diferenciaciones realizadas se adjuntan las varianzas asociadas a cada una de las series [37]. La menor varianza es proporcionada por la serie diferenciada regularmente. Sin embargo, observamos que el decrecimiento en la parte estacional era muy lento y por tanto no se podía considerar como serie estacionaria. La mejor opción parece realizar una diferenciación regular y estacional, que además tiene una varianza menor.

	Serie Inicial	Serie Dif Regular	Serie Dif Estacional	Serie Dif Regular y Estac
[1,]	0.5072583	0.1647882	0.7195425	0.2288644

Figura 37: Varianzas de cada serie

Para intentar decidir que modelos pueden ajustarse a los datos,se grafican los ACF y PACF resultantes de las diferenciaciones anteriores [36a] y [36b].

Se ha realizado el test de Dickey Fuller para la serie inicial [54a],la serie diferenciada estacionalmente [54b] y la serie final [54c], tras realizar la diferenciación regular y estacional para convertir la serie estacionaria. Se observa cómo inicialmente sin diferenciaciones y tras la diferenciación estacional, no se rechaza la hipótesis nula de que la serie es no estacionaria, por lo que 'acepta' su no estacionariedad. Sin embargo, tras aplicar las diferenciaciones, el pvalor resultante muy significativo, por lo que se rechaza la hipótesis nula, indicando la estacionariedad de la serie resultante.

El método de estimación que se va a utilizar es el método de máxima verosimilitud. Para ello, se utilizarán las funciones de autocorrelación (ACF) [36a] y funciones de autocorrelación parcial (PACF)[36b] obtenidas de diferenciar regularmente y estacionalmente la serie inicial. Por último, se probarán en SAS modelos con más de una estacionalidad observando si tienen una mayor capacidad de predicción.

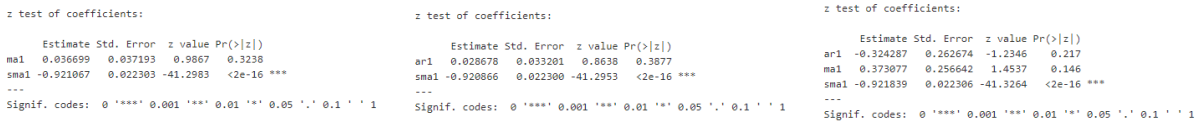
Observando las gráficas de ACF y PACF parece indicar el ajuste de un parámetro de media móvil en la parte estacional ya que en la ACF únicamente hay una autocorrelación alta correspondiente al retardo 1(en realidad retardo 24). Además, existe un decrecimiento lineal en la PACF en los retardos estacionales(24,48,72..). Por ello, se prueba de acuerdo a esta característica distintos

modelos modificando parámetros en la parte regular, 'siendo en la parte estacional siempre un modelo ARIMA(0,1,1)24 '.

Inicialmente, han sido probados los modelos más básicos posibles como son:

- ARIMA(0,1,1)(0,1,1)24
- ARIMA(1,1,0)(0,1,1)24
- ARIMA(1,1,1)(0,1,1)24

En los resultados siguientes podemos observar cómo ninguno de los tres modelos parece adecuarse a los datos. Los parámetros de orden 1 tanto autorregresivos como de media móvil no son significativos a un nivel de confianza alto. Únicamente es significativo el parámetro estacional asociado a la media móvil que se decidió añadir en el modelo de forma predeterminada.



(a) ARIMA(1,1,0)(0,1,1)24 (b) ARIMA(0,1,1)(0,1,1)24 (c) ARIMA(1,1,1)(0,1,1)24

Figura 38: Estimación de modelos ARIMA

A la vista de los resultados, se prueban dos modelos, ambos añadiendo un parámetro adicional al último modelo ARIMA probado: ARIMA(1,1,1)(0,1,1)24.

Estimación ARIMA(1,1,2)(0,1,1)24

$$(1-\phi_1 B)(1-B)(1-B^{24})X_t = (1-\theta_1 B-\theta_2 B^2)(1-\Theta_1 B^{24})a_t \tag{23}$$

En los resultados siguientes se puede explicar la estimación del modelo así como la matriz de correlaciones para los distintos parámetros.

```
Series: serie
ARIMA(1,1,2)(0,1,1)[24]

Coefficients:
          ar1      ma1      ma2     sma1
      0.8110  -0.8640  -0.1360  -0.9236
s.e.  0.0229   0.0388   0.0382   0.0227

sigma^2 estimated as 0.1128:  log likelihood=-323.95
AIC=657.91   AICc=657.97   BIC=681.98

z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ar1  0.810997   0.022909  35.4005 < 2.2e-16 ***
ma1 -0.864043   0.038767 -22.2879 < 2.2e-16 ***
ma2 -0.135954   0.038238  -3.5554 0.0003773 ***
sma1 -0.923642   0.022670 -40.7427 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(a) **Ajuste del modelo**

	ar1	ma1	ma2	sma1
ar1	1.00000000	-0.51225126	0.51930292	0.05174838
ma1	-0.51225126	1.00000000	-0.98204371	-0.05078344
ma2	0.51930292	-0.98204371	1.00000000	0.05146667
sma1	0.05174838	-0.05078344	0.05146667	1.00000000

(b) **Matriz de correlaciones**

Figura 39: Estimación de modelo ARIMA(1,1,2)(0,1,1)24

Se observa un modelo en el que los 4 parámetros asociados al modelo son muy significativos. Sin embargo, la matriz de correlaciones parece indicar una correlación muy alta de 0.98 entre los

parámetros de orden 1 y 2 de media móvil. Por tanto no se continua con la utilización de este modelo.

Estimación ARIMA(2,1,1)(0,1,1)24

La ecuación asociada al modelo estimado es:

$$(1-\phi_1 B-\phi_2 B^2)(1-B)(1-B^{24})X_t=(1-\theta_1 B)1-\Theta_1 B^{24})a_t \tag{24}$$

A continuación se adjuntan las estimaciones realizadas a los modelos comentados:

```
Series: serie
ARIMA(2,1,1)(0,1,1)[24]

Coefficients:
         ar1      ar2      ma1     sma1
      0.944  -0.1125  -1.0000  -0.9233
s.e.  0.033   0.0330   0.0066   0.0226

sigma^2 estimated as 0.1128:  log likelihood=-324.2
AIC=658.4   AICc=658.47   BIC=682.47

z test of coefficients:

       Estimate Std. Error   z value  Pr(>|z|)
ar1   0.9439634  0.0330001   28.6049 < 2.2e-16 ***
ar2  -0.1124661  0.0330435   -3.4036 0.0006651 ***
ma1  -0.9999978  0.0066103 -151.2778 < 2.2e-16 ***
sma1 -0.9232670  0.0226437  -40.7736 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) **Ajuste del modelo**

	ar1	ar2	ma1	sma1
ar1	1.000000e+00	-8.459799e-01	-4.986593e-05	-0.018048352
ar2	-8.459799e-01	1.000000e+00	-5.250679e-05	0.042288473
ma1	-4.986593e-05	-5.250679e-05	1.000000e+00	-0.000094666
sma1	-1.804835e-02	4.228847e-02	-9.466600e-05	1.000000000

(b) **Matriz de correlaciones**

Figura 40: Estimación de modelos ARIMA(2,1,1)(0,1,1)24

Todos los parámetros del modelo rechazan la hipótesis nula, indicando con los pvalores la alta significación de los parámetros del modelo. Sin embargo, la matriz de correlaciones vuelve a inducir una correlación alta, en este caso entre los parámetros autorregresivos de orden 1 y 2. No obstante, debido a la dificultad para encontrar mejores modelos, se utilizará este modelo para comprobar los residuales.

La adicción de parámetros de media móvil o autorregresivos no indagaban en una mejora de los modelos propuestos, por ello no han sido añadidos al trabajo en cuestión.

Estimación ARIMA(0,1,0)(2,1,0)24

$$(1-\Phi_1 B^{24}-\Phi_2 B^{48})(1-B)(1-B^{24})X_t=a_t \tag{25}$$

Por último, tal y como comentamos en el capítulo 2, existe un comando en el paquete *forecast* que nos indica el mejor modelo arima que encuentra dicha función (auto.arima()). Se han obtenido los siguientes resultados:

```
Series: serie
ARIMA(0,1,0)(2,1,0)[24]

Coefficients:
         sar1     sar2
      -0.6704  -0.3442
s.e.   0.0318   0.0323

sigma^2 estimated as 0.1526:  log likelihood=-441.8
AIC=889.6   AICc=889.62   BIC=904.04

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)
sar1 -0.670447   0.031826 -21.066 < 2.2e-16 ***
sar2 -0.344248   0.032284 -10.663 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) **Ajuste del modelo**

```
z test of coefficients:

     Estimate Std. Error  z value Pr(>|z|)
ar1   0.028678   0.033201   0.8638   0.3877
sma1 -0.920866   0.022300 -41.2953   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b) **Matriz de correlaciones**

Figura 41: Estimación de modelos ARIMA(0,1,0)(2,1,0)24

Los resultados que ofrece este modelo son satisfactorios. La alta significancia de los parámetros así como una matriz de correlaciones con correlacion pequeña indaga en un buen ajuste del modelo. No obstante, la no adicción de parámetros en la parte regular del modelo así como la utilización de parámetros autorregresivos en la parte estacional en vez de parámetros de media móvil, ofrece dudas sobre los residuos que tiene este modelo.

De acuerdo a las estimaciones de los modelos ARIMA realizados a los datos, sólo han sido encontrados dos modelos con buen ajuste a los datos. Sin embargo, es necesario que estos modelos también tengan unos residuos compatibles para que el modelo sea bueno. Para ello, se ha aplicado uno de los test más influyentes en la validación de los residuos como es el test de LjungBox.

El resultado del test para el modelo ARIMA(2,1,1)(0,1,1)24 se observa en [55b], mientras que para el modelo ARIMA(0,1,0)(2,1,0)24 en [55a],ambos realizados comparando autocorrelaciones hasta el retardo 168. En el primer caso no se rechaza la hipótesis nula(igualdad de los residuos), por lo que el resultado del test es satisfactorio. Sin embargo, se rechaza fuertemente la hipótesis nula para el modelo estimado por *auto.arima()*. Por tanto, sólo se trabajará de aquí en adelante con el único modelo válido encontrado en la estimación : ARIMA(2,1,1)(0,1,1)24.

Otros procedimientos útiles para comprobar los residuales, es observar las autocorrelaciones en la ACF así como comprobar la normalidad de los residuos. En el cuadro de figuras siguiente podemos observar los resultados de estas funciones:

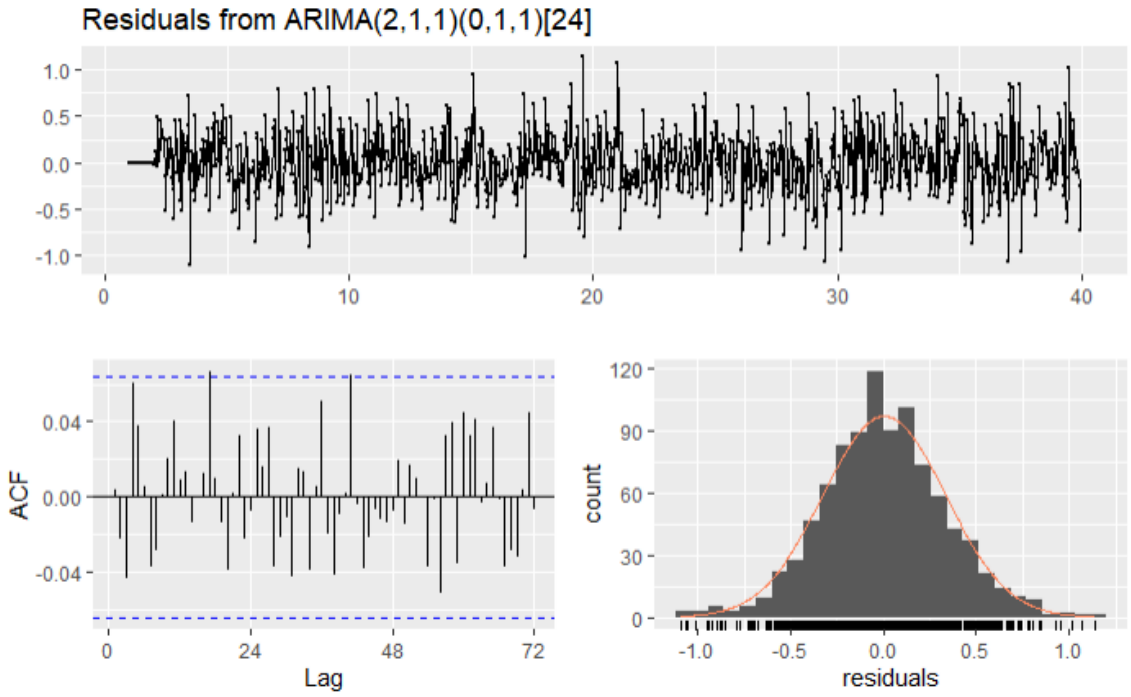


Figura 42: **Validación residuos del modelo**

En la primera imagen, se observa que los residuales se encuentran en torno al cero con una varianza relativamente constante por lo que es un resultado bueno. En cuanto a la ACF no se observan en los retardos más influyentes(lo estacionales o los primeros de la función) tengan una

autocorrelación alta. Por último, se detecta con un histograma cómo se distribuyen los residuos que vimos en la primera gráfica. Los residuos parecen distribuirse como una normal pero existe una desviación con valores algo por debajo de 0 que pueden hacer dudar sobre la normalidad de los residuos.

Debido a las dudas que ofrece el histograma, se realiza el test de Shapiro-Wilks observando el pvalor del test [43], el cual ofrece más dudas sobre la normalidad de los residuos. Se podría decir que con un nivel de significación del 98 % podemos no rechazar el test y por tanto corroborar la normalidad de los residuos.

```
        Shapiro-Wilk normality test

data:  residuals(arima5)
W = 0.9962, p-value = 0.02238
```

Figura 43: **Test Shapiro-Wilks**

Con la validación de los residuos del modelos realizada, se lleva a cabo la predicción. De acuerdo a las 168 observaciones reservadas para test, desde el 23 de Abril hasta el 29 de Abril, se han utilizado estos días para comparar los valores de NO_2 de esos días con la predicción de los datos de entrenamiento de 168 pasos hacia adelante con el modelo ajustado ARIMA(2,1,1)(0,1,1)24 . En la gráfica siguiente se muestra la evolución de la serie con los datos de entrenamiento seguido de la predicción.

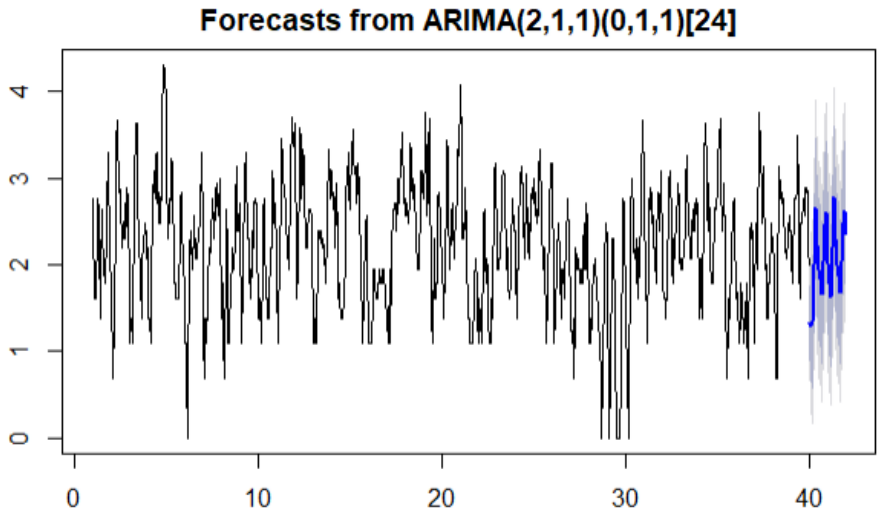


Figura 44: **Predicción total datos de prueba**

Se muestra como ha detectado la tendencia y el patrón de la serie. Con ayuda de los intervalos de confianza, es posible detectar los valores de NO2 con mayor contaminación correspondientes a horas puntas durante algún día.

Para observar el comportamiento únicamente de los datos reservados para test, se adjunta un gráfico con la evolución, ya con el logaritmo aplicado:

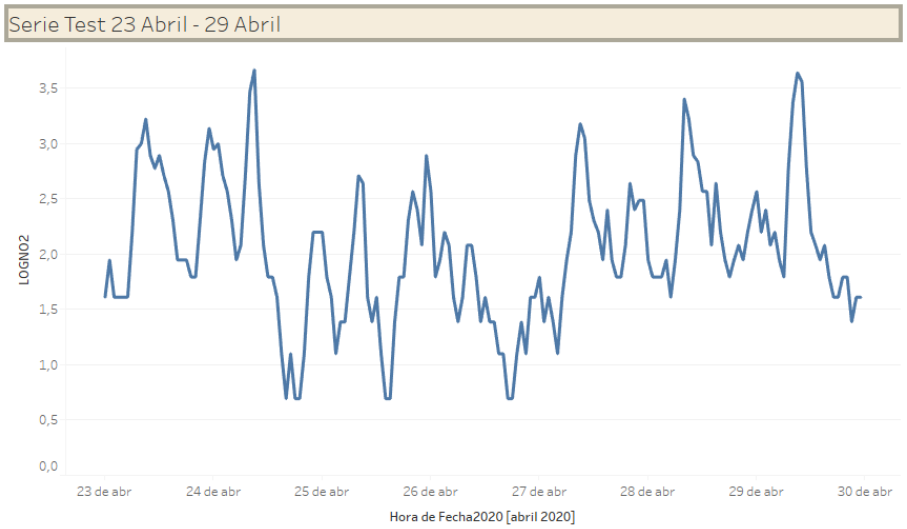


Figura 45: **Serie Test Horaria 168 observaciones**

Para observar de forma mas concreta una comparación de los datos reales con los datos predichos, se adjunta una gráfica [44] con dicha comparación con unos intervalos de predicción para la contaminación.

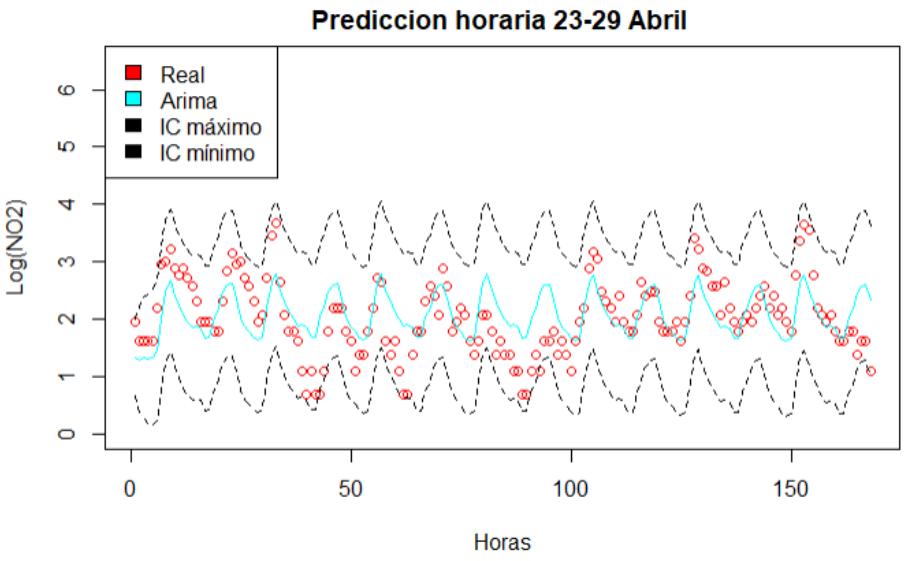


Figura 46: **Predicción 168 horas con intervalos de confianza**

Para comparar la capacidad de predicción de este modelo con el modelo con múltiple estacionalidad que se realiza en la siguiente sección, se ha calculado la suma de cuadrados del error reservando 168 observaciones y realizando la predicción calculando 168 pasos hacia adelante. El valor resultante es 52.689.

5.2.1. Modelos ARIMA con múltiple estacionalidad

En esta parte del trabajo, se va a proceder a buscar modelos con más de una estacionalidad que se ajusten bien a los datos. Como ha sido comentado al inicio de este trabajo, no ha sido posible

la aplicación de este tipo de modelos utilizando paquetes específicos de modelado ARIMA en R con estimación de parámetros de máxima verosimilitud. Es por ello, que esta última parte de este trabajo se lleva a cabo con SAS.

La ventaja que proporciona el *PROC ARIMA* de SAS es que ajusta modelos con múltiples periodos y modelos con restricciones en los coeficientes (en la que muchos coeficientes se hacen 0).

En este sentido, se han encontrado dos modelos ARIMA concretos con múltiple estacionalidad que pueden considerarse aceptables para nuestros datos. Se compara la capacidad de predicción de los mismos con respecto al mejor modelo encontrado con una única estacionalidad en R. Para evitar las diferencias que pueda haber en el cálculo del AIC en R y SAS se ha utilizado como medida de comparación la suma de cuadrados del error(SSE). Los modelos son (en Anexos más detallado [56] y [57]):

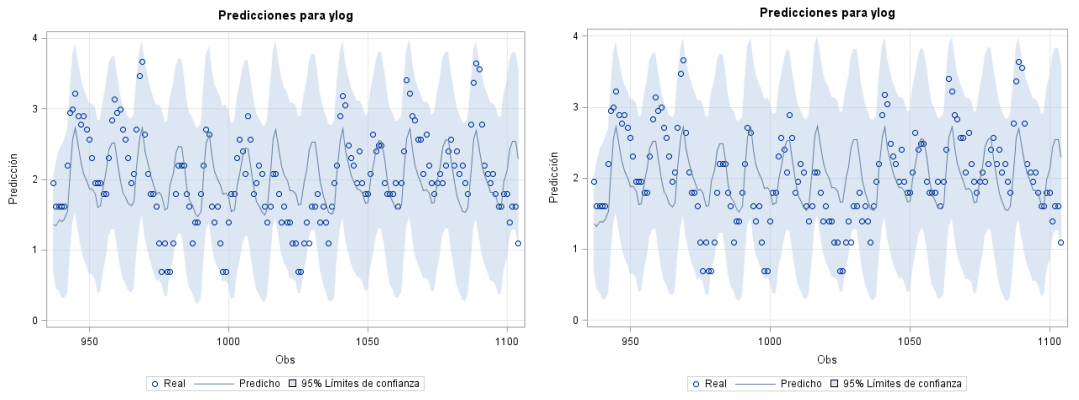
Modelo 1

$$(1-\phi_1 B-\phi_2 B^2)(1-B)(1-B^{24})X_t = (1-\theta_1 B-\theta_{24}B^{24}-\theta_{25}B^{25}-\theta_{96}B^{96}-\theta_{97}B^{97} \\ -\theta_{144}B^{144}-\theta_{145}B^{145}-\theta_{168}B^{168}-\theta_{169}B^{169})a_t \tag{26}$$

Modelo 2

$$(1-\phi_1 B-\phi_2 B^2)(1-B)(1-B^{24})X_t = (1-\theta_1 B-\theta_{24}B^{24}-\theta_{25}B^{25}-\theta_{72}B^{72}-\theta_{73}B^{73} \\ -\theta_{168}B^{168}-\theta_{169}B^{169})a_t \tag{27}$$

Las predicciones 168 pasos hacia adelante junto con los datos reales de ambos modelos, se adjuntan a continuación:



(a) **Predicción 168 observaciones modelo 1** (b) **Predicción 168 observaciones modelo 2**

Figura 47: Predicción con modelos múltiple estacionalidad

Las predicciones en las figuras [47a] y [47b] son muy similares por lo que se compara la predicción con el SSE para determinar que modelo obtiene una mejor capacidad predictiva. Los resultados se detallan en la tabla [58]. El primer modelo obtiene un valor ligeramente menor, por tanto con una mejor capacidad predictiva. Para comparar la capacidad predictiva con el modelo implementado

en R, se añade una tabla comparativa del valor SSE. Se observa como, de nuevo, el mejor modelo es el modelo 1 con un valor de 49.548.

	Modelo Arima(2,1,1)(0,1,1)24	**Modelo 1 múltiple estacionalidad**	**Modelo 2 múltiple estacionalidad**
SSE	52.689	49.548	50.8916

El código SAS asociado para obtener el SSE de los dos modelos con múltiple estacionalidad se encuentra en Anexos [7.4.3].

6. Conclusiones y lineas futuras

En este trabajo se han utilizado distintas herramientas para modelar mediante ARIMA los datos de contaminación de Madrid. Se han utilizado modelos con distintas estacionalidades, mensual y horaria, utilizando datos desde 2011 hasta 2020.

La primera parte de este trabajo se basó en realizar una pequeña guía de paquetes y funciones en R que sirvieran como ayuda para poder trabajar con series temporales. En este sentido, puede ser de ayuda para estudiantes que se inicien en la práctica al manejo de series. Personalmente, al haberme iniciado del mismo modo a manejar series temporales en el lenguaje R, pienso que la ayuda de esta pequeña guía podría ahorrar tiempo de búsqueda de comandos concretos.

Una parte interesante del presente trabajo es el análisis descriptivo sobre la contaminación producida en Madrid en una de las zonas más dañadas por la contaminación como es la *Plaza Castilla*. Además, el aprendizaje de una herramienta gráfica como Tableau llevó a realizar unos gráficos más interesantes y con más contenido que los realizados en R o en SAS.

Una mayor contaminación en los años 2015 o 2016 con una disminución notable en 2019 reflejan el interés de los últimos años en reducir la contaminación en una zona como Madrid donde viven millones de personas y pueden haberse visto afectadas por la abundante contaminación existente. También es reseñable como el miércoles o el jueves existe de media un 10 % más de contaminación que otro día laborable como el lunes.
Por otra parte, es destacable una gran diferencia de contaminación en las horas puntas como son las 8 de la mañana o las 9 de la noche en la que la contaminación duplica los valores medios diarios, seguramente provocado por el tráfico generado diariamente.

Por último, se dedicó una pequeña sección a los sucesos acontecidos debido a la enfermedad del Covid-19. Se filtraron datos provenientes de los meses de Marzo y Abril de 2020 para observar si el confinamiento provocado por el estado de alarma había provocado un descenso de la contaminación. Concretamente, en la zona de estudio, se redujo en Marzo un 50 % del gas NO2, habiendo estado únicamente 15 días de confinamiento en este mes. Sin embargo en Abril la reducción con respecto a otros años fue mayor del 70 %, lo que se vió reducido en imágenes de Madrid hasta hace mucho no vistas debido a la contaminación en el aire. Por tanto, ante esta catástrofe mundial debida a la enfermedad del coronavirus, existe alguna buena noticia como la reducción drástica del NO2 para contribuir a mejorar la salud de las personas.

Por último, fue aplicado el enfoque de Box-Jenkins aplicando modelos ARIMA a dos series con motivo de encontrar un modelo que se ajustara a los datos y predijera datos futuros del mejor modo posible. Para ello se analizó inicialmente una serie mensual de los años 2011 a 2020, en la que se encontró una predicción bastante buena con pequeñas estimaciones erróneas, salvo los meses de Noviembre y Diciembre de 2019, que cambiaron la tendencia positiva de la serie.

Sin embargo, la serie horaria con una mayor cantidad de información a modelar (casi 1000 observaciones), fue más difícil encontrar modelos que se adecuaran a los datos. Además, debido a una varianza no constante de la serie, fue necesario inicialmente realizar una transformación logarítmica de la serie con motivo de reducir esa varianza. La validación de los residuos fue buena y la predicción, teniendo en cuenta que se estaba trabajando con datos durante el estado de

alarma, no fueron malos. Hay que tener en cuenta que no existía cierta garantía de saber cómo iba a evolucionar la contaminación con esta nueva normalidad. No obstante, el resultado predicho no parece ser malo aunque los valores de contaminación de algún fin de semana eran demasiado bajos y se alejaban de los valores predichos por el modelo.

Para poder mejorar el modelo predictivo encontrado, se buscaron modelos similares que mejoraran la capacidad de predicción añadiendo un periodo a la estacionalidad del modelo. Debido al alcance de este trabajo, se realizó únicamente una comparación de la capacidad predictiva del modelo implementado en R con respecto a los modelos implementados con PROC ARIMA [40], obteniendo una mejor capacidad de predicción en los modelos con múltiple estacionalidad. Como lineas futuras sería interesante implementar en R este tipo de modelos con estimación por máxima verosimilitud con múltiple estacionalidad.

Referencias

[1] P. León, "La contaminación baja a "niveles históricos" debido a madrid central," *El Pais*, Junio 2019.

[2] K. sik Chan and J. Cryer, *Time Series Analysis with Applications in R*. Springer, Marzo 2008.

[3] R. Hyndmann, "Two seasonal periods in arima using r." `URL:https://stats.stackexchange.com/questions/47729/two-seasonal-periods-in-arima-using-r`, 2015.

[4] R. C. Team and contributors worldwide, "**stats-package:** the r stats package." https://rdrr.io/r/stats/stats-package.html.

[5] RDocumentation, "Comando ts." `URL:https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ts`.

[6] RDocumentation, "Comando diff." `URL:https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/diff`.

[7] RDocumentation, "Comando acf." `URL:https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/acf`.

[8] R. Hyndman, "**forecast:** forecasting functions for time series and linear models." https://cran.r-project.org/web/packages/forecast/forecast.pdf, (2020). R package version 8.12.

[9] S. Razbash and R. J. Hyndman, "msts: Multi-seasonal time series." https://rdrr.io/cran/forecast/man/msts.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[10] R. J. Hyndman, "Arima: Fit arima model to univariate time series." https://rdrr.io/cran/forecast/man/Arima.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[11] R. J. Hyndman, "auto.arima: Fit best arima model to univariate time series." https://rdrr.io/cran/forecast/man/auto.arima.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[12] R. J. Hyndman, "tsdisplay: Time series display." https://rdrr.io/cran/forecast/man/tsdisplay.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[13] R. J. Hyndman, "checkresiduals: Check that residuals from a time series model look like white...." checkresiduals: Check that residuals from a time series model look like white... **In forecast**: Forecasting Functions for Time Series and Linear Models.

[14] R. J. Hyndman, "accuracy: Accuracy measures for a forecast model." https://rdrr.io/cran/forecast/man/accuracy.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[15] R. J. Hyndman, "forecast: Forecasting time series." https://rdrr.io/cran/forecast/man/forecast.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[16] K.-S. Chan and B. Ripley, "**TSA:** time series analysis." https://rdrr.io/cran/TSA/. Contains R functions and datasets detailed in the book. 'Time Series Analysis with Applications in R (second edition)'.

[17] K.-S. C. based on A. Trapletti's work on the Box.test function in the stats package, "Lb.test: Portmanteau tests for fitted arima models." https://rdrr.io/cran/TSA/man/LB.test.html. **In TSA**: Time Series Analysis.

[18] K. S. Chan, "armasubsets: Selection of subset arma models." https://rdrr.io/cran/TSA/man/armasubsets.html. **In TSA**: Time Series Analysis.

[19] A. Trapletti and K. Hornik, "**tseries:** time series analysis and computational finance." https://cran.r-project.org/web/packages/tseries/tseries.pdf, (2019). R package version 0.10-47.

[20] A. Trapletti, "adf.test: Augmented dickey-fuller test." https://rdrr.io/cran/tseries/man/adf.test.html. **In tseries**: Time Series Analysis and Computational Finance.

[21] D. Wuertz, "**fUnitRoots**: Rmetrics - modelling trends and unit roots." https://rdrr.io/rforge/fUnitRoots/man/. R package version 3042.79.

[22] D. Qiu, "kpss.test: Kwiatkowski-phillips-schmidt-shin test." https://rdrr.io/cran/aTSA/man/kpss.test.html. In aTSA: Alternative Time Series Analysis.

[23] T. Hothorn, A. Zeileis, and R. W. Farebrother, "**lmtest**: Testing linear regression models." https://cran.r-project.org/web/packages/lmtest/lmtest.pdf, (2019). R package version 0.9-37.

[24] "Coeftest: Inference for estimated coefficients." https://rdrr.io/cran/lmtest/man/coeftest.html. In lmtest: Testing Linear Regression Models.

[25] Y. Aragon, "**caschrono-package**: Series temporelles avec r." https://rdrr.io/cran/caschrono/man/caschrono-package.html, (2016). R package version 2.0.

[26] Y. Aragon, "cor.arma: Correlation matrix of the parameters for an arima model." https://rdrr.io/cran/caschrono/man/cor.arma.html. In caschrono: Series Temporelles Avec R.

[27] I. Svetunkov, "msarima: Multiple seasonal arima." https://rdrr.io/cran/smooth/man/msarima.html. **In smooth**: Forecasting Using State Space Models.

[28] S. Razbash and R. J. Hyndman, "bats: Bats model (exponential smoothing state space model." https://rdrr.io/cran/forecast/man/bats.html. **In forecast**: Forecasting Functions for Time Series and Linear Models.

[29] Y.-T. Chou, "Comando bats vs comando tbats." `URL:https://yintingchou.com/posts/bats-and-tbats-model/`.

[30] "Calidad del aire. datos horarios años 2001 a 2020." `URL:https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=f3c0f7d512273410VgnVCM2000000c205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default`. Datos abiertos gobierno de Madrid.

[31] "Calidad del aire. datos diarios años 2001 a 2020." URL:https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=aecb88a7e2b73410VgnVCM2000000c205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default. Datos abiertos gobierno de Madrid.

[32] GreenFacts, "Contaminación del aire dióxido de nitrógeno," *GreenFacts*, Abril 2006.

[33] E. en Accion, "Efectos de la crisis de la covid-19 en la calidad del aire urbano en españa," *Creative Commons*, Abril 2020.

[34] B. G. Núñez, "Madrid sin contaminación." Twitter, Mayo 2020.

[35] B. Samper, "La boina de contaminación sobre madrid," *El Mundo*, 2019.

[36] M. A. Medina, "Madrid central reduce la contaminación un 20 % en el primer año," *El Pais*, Noviembre 2019.

[37] A. C. C. López, "Madrid supera ya el límite de la ue por contaminación del aire para todo el 2015," *La Vanguardia*, Enero 2015.

[38] M. P. G. Casimiro, *Análisis de series temporales: Modelos ARIMA*. Universidad de País Vasco, Departamento de Economía Aplicada III, Abril 2009.

[39] S. de la Fuente Fernández, *Series Temporales: Modelo Arima*. Universidad Autónoma de Madrid, No hay fecha.

[40] SAS, *SAS/ETS® 13.2 User's Guide The ARIMA Procedure*. SAS Institute Inc., Cary, NC, USA, 2014.

7. Anexos

7.1. Tablas conjunto de datos

7.1.1. Ejemplo de conjunto de datos inicial diario

	PROVINCIA	MUNICIPIO	ESTACION	MAGNITUD	PUNTO_MUESTREO	ANO	MES	D01	V01	D02	V02
8614	28	79	50	8	28079050_8_8	2015	11	23	V	43	V
8615	28	79	50	8	28079050_8_8	2015	12	80	V	101	V
8616	28	79	50	9	28079050_9_47	2015	1	0	N	0	N
8617	28	79	50	9	28079050_9_47	2015	2	4	V	8	V
8618	28	79	50	9	28079050_9_47	2015	3	6	V	12	V
8619	28	79	50	9	28079050_9_47	2015	4	5	V	4	V
8620	28	79	50	9	28079050_9_47	2015	5	7	V	8	V

Figura 48: **Formato Dataset inicial diario**

7.1.2. Ejemplo de conjunto de datos inicial horario

	PROVINCIA	MUNICIPIO	ESTACION	MAGNITUD	PUNTO_MUESTREO	ANO	MES	DIA	H01	V01	H02	V02
3131	28	79	50	8	28079050_8_8	2020	4	4	53.00	V	59.00	V
3132	28	79	50	8	28079050_8_8	2020	4	5	3.00	V	5.00	V
3133	28	79	50	8	28079050_8_8	2020	4	6	12.00	V	7.00	V
3134	28	79	50	8	28079050_8_8	2020	4	7	11.00	V	7.00	V
3135	28	79	50	8	28079050_8_8	2020	4	8	20.00	V	15.00	V
3136	28	79	50	8	28079050_8_8	2020	4	9	24.00	V	21.00	V
3137	28	79	50	8	28079050_8_8	2020	4	10	6.00	V	4.00	V
3138	28	79	50	8	28079050_8_8	2020	4	11	5.00	V	3.00	V
3139	28	79	50	8	28079050_8_8	2020	4	12	10.00	V	4.00	V

Figura 49: **Formato Dataset inicial horario**

7.2. Procesado de los datos código en R

7.2.1. Filtrado de los datos y transformación de los valores 'N' a NA. Ejemplo para 2011

```
datos # Es la union de todos los anios de 2011 a 2020
indices<-which(datos$ESTACION=="50")
datosestacion<-datos[indices,]
indices<-which(datosestacion$MAGNITUD=="8")
datoscastilla<-datosestacion[indices,]
## Nos quedamos con la informacion relevante
datosfiltrados<-datoscastilla[c(-1,-2,-3,-4,-5)]

## Datos 2011
indices11<-which(datosfiltrados$ANO==2011)
datosfiltrados11<-datosfiltrados[indices11,]
## Obtener los indices de los no validos
indicesN11<-which(datosfiltrados11=="N")#9N

## Eliminar las columnas de V o N ya sabiendo la posicion de los N
datosfiltrados11V<-datosfiltrados11[,-sequencia]

# 2011: 9N (7 son porque estan 31 dias para todos los meses)
# Se observa de que dias son

datosfiltrados11V[1,27]<-NA # Dia 25 Enero
datosfiltrados11V[4,11]<-NA # Dia 9 Abril
```

7.2.2. Transformacion de los datos a columnas y reemplazar valores NA por la media del mes asociado

```
diames<-rep(seq(1,31,1),12)

datosanioconjnum<-as.numeric()
datosmesconjnum<-as.numeric()
datosconjnum<-as.numeric()

datosfiltrar<-function(x){

  for (i in 1:nrow(x)){
    anio <-x[i,1]
    mes <-x[i,2]
    dianum<-x[i,3:33]
    anionum<-t(anio)
    anionum<-rep(anionum,31)
    mesnum<-t(mes)
    mesnum<-rep(mesnum,31)
    datosTnum<-t(dianum)
    datosanioconjnum<-append(datosanioconjnum,anionum)
    datosmesconjnum<-append(datosmesconjnum,mesnum)
    datosconjnum<-append(datosconjnum,datosTnum)
  }
  return(data.frame(datosanioconjnum,datosmesconjnum,diames,datosconjnum))
}

datosconjuntos11<-datosfiltrar(datosfiltrados11V)
names(datosconjuntos11)<-c("Anio","Mes","Dia","NO2")

# Correspondientes a dias que no tiene el mes. Ej 30 Febrero
eliminar11<-which(datosconjuntos11$NO2=="0")
datosconjuntos11N<-datosconjuntos11[-eliminar11,]

# Detectar valores del mes donde hay NA
indice11_1<-which(datosconjuntos11N$Mes=="1")
indice11_2<-which(datosconjuntos11N$Mes=="4")

# Obtener la media del mes
media11_1<-mean(datosconjuntos11N[indice11_1,4],na.rm=TRUE)
media11_2<-mean(datosconjuntos11N[indice11_2,4],na.rm=TRUE)

# Remplazar por la media los valores no validos sabiendo el dia
datosconjuntos11N[25,4]<-media11_1
datosconjuntos11N[99,4]<-media11_2
```

7.3. Serie Mensual

7.3.1. Resultados test de Dickey Fuller

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 10
  STATISTIC:
    Dickey-Fuller: -2.4261
  P VALUE:
    0.401
```

(a) **Test de Dickey Fuller**

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 10
  STATISTIC:
    Dickey-Fuller: -6.918
  P VALUE:
    0.01
```

(b) **Test de Dickey Fuller**

Figura 50: Test de Dickey Fuller

7.3.2. Matrices de correlación

	ma1	sar1
ma1	1.0000000	-0.1612371
sar1	-0.1612371	1.0000000

(a) **ARIMA(0,0,1)(1,1,0)12**

	ar1	sar1
ar1	1.0000000	-0.2675273
sar1	-0.2675273	1.0000000

(b) **ARIMA(1,0,0)(1,1,0)12**

Figura 51: Matriz de correlaciones

	ar1	sma1
ar1	1.0000000	-0.1696703
sma1	-0.1696703	1.0000000

(a) **ARIMA(1,0,0)(0,1,1)12**

	ma1	sma1
ma1	1.0000000	-0.1417944
sma1	-0.1417944	1.0000000

(b) **ARIMA(0,0,1)(0,1,1)12**

Figura 52: Matriz de correlaciones

7.3.3. Test de LjungBox

```
Box-Ljung test

data:  residuals from  arima2
X-squared = 68.815, df = 68, p-value = 0.4496
```

(a) **ARIMA(1,0,0)(1,1,0)12**

```
Box-Ljung test

data:  residuals from  arima3
X-squared = 68.587, df = 68, p-value = 0.4573
```

(b) **ARIMA(1,0,0)(0,1,1)12**

```
Box-Ljung test

data:  residuals from  arima4
X-squared = 63.105, df = 68, p-value = 0.6453
```

(c) **ARIMA(0,0,1)(0,1,1)12**

Figura 53: Test de LjungBox

7.4. Serie horaria

7.4.1. Resultados test de Dickey Fuller

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 200
  STATISTIC:
    Dickey-Fuller: -1.8478
  P VALUE:
    0.6427
```

(a) **Test de Dickey Fuller serie inicial horaria**

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 200
  STATISTIC:
    Dickey-Fuller: -2.7475
  P VALUE:
    0.2619
```

(b) **Test de Dickey Fuller serie horaria diferenciada estacionalmente**

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 200
  STATISTIC:
    Dickey-Fuller: -4.4749
  P VALUE:
    0.01
```

(c) **Test de Dickey Fuller serie horaria diferenciada regularmente y estacionalmente**

Figura 54: Test de Dickey Fuller

7.4.2. Test de LjungBox

```
        Box-Ljung test

data:  residuals from  arimaaut
X-squared = 303.64, df = 166, p-value = 3.876e-10
```

(a) **ARIMA(0,1,0)(2,1,0)24**

```
        Box-Ljung test

data:  residuals from  arima5
X-squared = 140.67, df = 164, p-value = 0.9062
```

(b) **ARIMA(2,1,1)(0,1,1)24**

Figura 55: Test de Ljung Box

7.4.3. Código SAS Ajuste,Validación y predicción del modelo con doble estacionalidad

```
DATA logcovid19;
        SET covid19total;
        ylog=log(contam20);
RUN;


DATA modif; SET logcovid19; IF  _N_< 937;
RUN;

PROC ARIMA DATA=logcovid19;
        IDENTIFY var=ylog(1,24) nlag=168;
        RUN;
                ESTIMATE p=2 q=(1,24,25,96,97,144,145,168,169)
                METHOD=ml noconstant;
        RUN;
        FORECAST lead=168 back=168 out=A;
RUN;

PROC ARIMA DATA=logcovid19;
        IDENTIFY var=ylog(1,24) nlag=168;
        RUN;
        ESTIMATE p=2 q=(1,24,25,72,73,168,169) METHOD=ml ;
        RUN;
        FORECAST lead=168 back=168 out=B;
RUN;

DATA modelo; SET logcovid19; KEEP ylog; if _N_>936; RUN;

DATA modelo1;SET A;KEEP FORECAST; IF _N_>936;
RENAME FORECAST=FORECASTSARIMA1;
RUN;

data modelo2;SET B;keep FORECAST; IF _N_>936;
RENAME FORECAST=FORECASTSARIMA2;
RUN;

DATA x;MERGE modelo modelo1 modelo2;res1=ylog-FORECASTSARIMA1;
res2=ylog-FORECASTSARIMA2;
ssemodelo1+res1*res1;
ssemodelo2+res2*res2;

DATA sse; SET x;
```

```
IF _n_=168;
RUN;

PROC PRINT DATA=sse;RUN;
```

7.4.4. Modelo 1 con valores de los parámetros

Factores autoregresivos	
Factor 1:	1 - 0.91815 B**(1) + 0.10706 B**(2)

Factores de la media móvil	
Factor 1:	1 - 0.98801 B**(1) - 0.93278 B**(24) + 0.91493 B**(25) + 0.00932 B**(96) - 0.00033 B**(97) + 0.013 B**(144) - 0.01358 B**(145) - 0.03649 B**(168) + 0.0344 B**(169)

Figura 56: **Valor de los parámetros del Modelo 1**

7.4.5. Modelo 2 con valores de los parámetros

Factores autoregresivos	
Factor 1:	1 - 0.92003 B**(1) + 0.10933 B**(2)

Factores de la media móvil	
Factor 1:	1 - 0.9855 B**(1) - 0.9346 B**(24) + 0.91228 B**(25) + 0.01211 B**(72) - 0.00287 B**(73) - 0.02221 B**(168) + 0.02126 B**(169)

Figura 57: **Valor de los parámetros del Modelo 2**

7.4.6. SSE modelos PROC ARIMA

Obs	ylog	FORECASTSARIMA1	FORECASTSARIMA2	res1	res2	ssemodelo1	ssemodelo2
1	1.09861	2.28854	2.29542	-1.18992	-1.19681	49.5480	50.8916

Figura 58: **SSE modelos ARIMA múltiple estacionalidad**