Universidad de Valladolid

Facultad de Ciencias

Grado en Estadística

Análisis de series multiestacionales mediante modelos de Espacio de Estados

---

Autor: Víctor Vaquero Martínez

Tutora: M. Pilar Rodríguez del Tío

Análisis de series multiestacionales mediante modelos de Espacio de Estados

Víctor Vaquero Martínez

Índice general

Índice de figuras

Índice de cuadros

Dedicado a
mi familia y amigos

Agradecimientos

Muchas gracias a mi familia por apoyarme durante toda la carrera y a mi tutora, por ser una increíble profesora y guía.

Resumen

Los modelos de Espacios de Estados son una invención relativamente moderna que ofrecen una mayor versatilidad para el análisis de series temporales en comparación con los más comunes modelos ARIMA. En este proyecto analizamos y presentamos las diferentes variantes que existen actualmente de estos modelos, además de mostrar sus similitudes y diferencias. Posteriormente desarrollamos su aplicación practica en un análisis de R sobre datos multi-estacionales de una bolsa energética europea.

Palabras claves: multiestacionalidad, estacionalidad, modelos de espacio de estados, series temporales, R

Abstract

State Space Models are a relatively new invention that offer more versatility in time series analysis in comparison to the more common ARIMA models. In this project, we will analyze and display the range of variants that currently exist for these models and show all their similarities and differences. Afterwards, we will develop their practical application with the R language on multiseasonal data of an European energy market.

Keywords: multiseasonality, seasonality, state space models, time series, R

Parte I

Memoria del Proyecto

Capítulo 1

Descripción del proyecto

A continuación vamos a dar una breve introducción del objetivo de este estudio. En primer lugar daremos una primera impresión sobre las series temporales en general y los diferentes modelos comúnmente usados para su análisis. Después comentaremos cual es la situación actual de la técnica estadística, así como el porque de la realización de este proyecto, cómo se enmarca dentro de la carrera y su utilidad en general. Por último, mencionaremos los objetivos específicos que se desea alcanzar y los datos sobre los que se aplicara un análisis practico de las ideas desarrolladas previamente.

1.1. Introducción

Una serie temporal es «una sucesión de datos medidos en determinados momentos y ordenados cronológicamente» [Ser]; son básicamente cualquier colección de observaciones hechas secuencialmente en el tiempo, de naturaleza estocástica, sobre las que la estadística se provee para estudiar sus características. La propiedad principal que las diferencia del resto de conjuntos de datos es que cada observación esta correlaccionada con el resto (el patrón exacto depende de la serie) y por tanto la típica hipótesis de independencia no se mantiene. Las más comúnmente estudiadas son aquellas con observaciones equi-espaciadas y en tiempo discreto (foco de nuestra atención en este proyecto) aunque no son necesariamente las únicas de interés. Un ejemplo común de este primer tipo de datos son los índices de la bolsa, e.g. una sucesión de precios de las acciones de una empresa en un conjunto de días dados (Imagen 1.1).

Una vez se obtienen los datos es necesario, para estudiarlos, establecer un modelo probabilístico como base del análisis. Específicamente, un modelo de serie temporal sobre un conjunto de observaciones es una especificación de la distribución conjunta de una secuencia de variables aleatorias correlaccionadas entre sí [BD02].

Figura 1.1: Serie temporal de las acciones del Banco Santander durante el periodo del corona-virus (1 Junio- 4 Mayo) - Fuente Yahoo Finance

$$X_1, X_2 ... X_n$$
$$P(X_1 \leq x_1 ... X_n \leq x_n) \ \ \forall n = 1, 2, ...$$

Dado que una especificación completa de la distribución es, en la practica, inviable, se han desarrollado un número de modelos que, partiendo de simplificaciones (como estacionaridad, errores de tipo gausiano, linearidad, etc), son capaces de modelar y predecir complejos sistemas de muy diversas naturalezas con una alta fiabilidad.

Ejemplos clásicos de estos modelos estadísticos son los ARIMA[1] y los SSM[2], siendo el estudio de estos últimos, para la modelación de datos multiestacionales, el objetivo principal de este proyecto.

1.2. Pasado y situación actual

En general, el estudio estadístico de series temporales es relativamente reciente, pues si bien hay investigaciones previas, tan solo en los años $30s \sim 40s$ Herman Wold describió de manera unificada los ahora famosos modelos ARMA[3] y no fue hasta los años $70s$ que George Edward Pelham Box y Wilym Meirion Jenkin establecieron por completo

[1] Auto Regresive Integrated Moving Average o Autoregresión integrada de media móvil.
[2] State Space Models o Modelos de Espacio de Estados.
[3] Auto Regresive Moving Average o Autoregresión de media móvil

los modelos ARIMA y el método de Box-Jenkins, ahora un enfoque muy común para encontrar y estimar dichos modelos dada una serie de observaciones [NW13].

Los SSM (conocidos usualmente como modelos de espacio de estados o modelos dinámicos) son incluso más recientes, pues no fue desarrollado el primer método de estimación, el filtro de Kalman, hasta 1960 por Rudolf E. Kálmán de donde toma su nombre, aunque Thorvald Nicolai Thiele y Peter Swerling habrían desarrollado, al mismo tiempo, un algoritmo muy similar, al igual que Ruslan Leontevich Stratonovich, que desarrolló por aquel entonces un método más general en la entonces Unión Soviética [GA10] [Kal].

Desde entonces la investigación ha tomado dos ramas diferentes; por un lado han continuado siendo desarrolladas versiones más complejas de estos métodos, ahora clásicos, para paliar sus debilidades, e.g. versiones no lineales, a tiempo continuo, etc. Por otro lado, dado el reciente desarrollo y puesta en practica del Deep Learning, han aparecido numerosos modelos automáticos capaces de competir con los métodos más "tradicionales", como CNN (Convolutional Neural Network o redes neuronales convoluciones) o como las RNN (Recurrent Neural Networks o redes neuronales recurrentes) [Faw+18].

En el terreno de los SSM, el filtro de Kalman en particular ha observado una serie de continuos desarrollos para sobrepasar las diversas limitaciones del algoritmo original. Por ejemplo, el EKF[4] es una generalización muy usada que permite la estimación de modelos no lineales [JU04]; también tenemos el filtro de información que es una reformulación algebraica que, entre otras cosas, permite tratar de manera directa series no estacionarias (con varianza infinita/prior difusa) [Hyn+08].

Estos nuevos modelos/algoritmos, junto con otras mejoras prácticas en el terreno del computo/optimización numérica, ha provocado que estos métodos sean actualmente increíblemente usados en todas sus variantes.

1.3. Motivación

Dado el actual interés sobre la predicción de series temporales y los nuevos desarrollos que se están llevando a cabo en este momento, deseamos analizar las relativas ventajas de los SSM. En particular, deseamos realizar la comparación con los modelos ARIMA estudiados en la carrera. Además nos interesa especialmente investigar los beneficios de estos modelos en el caso de datos con múltiples estacionalidades (E.g. diarias, semanales, anuales) tanto en el modelado como en la estimación de parámetros. Por último, queremos mostrar las diferencias practicas de implementación de estos modelos en el lenguaje R, mostrar la disponibilidad de las librerías estadísticas y analizar su rendimiento relativo.

[4]Extended Kalman Filter o filtro de Kalman extendido.

1.4. Objetivos del trabajo

A continuación, mostramos una lista de los objetivos primarios y secundarios que intentaremos cubrir en este proyecto:

- Presentar el fundamento teórico de los modelos ARIMA y SSM
- Analizar las similitudes y diferencias entre ambos modelos
- Mostrar las variantes de modelado SSM
- Presentar las similitudes y diferencias entre el filtro de información y el de Kalman
- Analizar datos multi-estacionales con R
- Explorar beneficios y limitaciones de los múltiples modelos

1.5. Datos

Para mostrar la aplicación práctica de estos modelos hemos elegido trabajar sobre datos reales de bolsa. La razón principal es que son un ejemplo típico de series temporales, con un relativamente sencillo acceso público. Además tomamos un conjunto de observaciones correspondientes a los precios por MWh en un mercado energético, concretamente la bolsa europea NordPool, [Nor] basándonos en la hipótesis de que dicha serie poseerá una estructura estacional correspondiendo a los diferentes patrones de gasto energético provocados por las diferentes horas de día y los diferentes días de la semana.

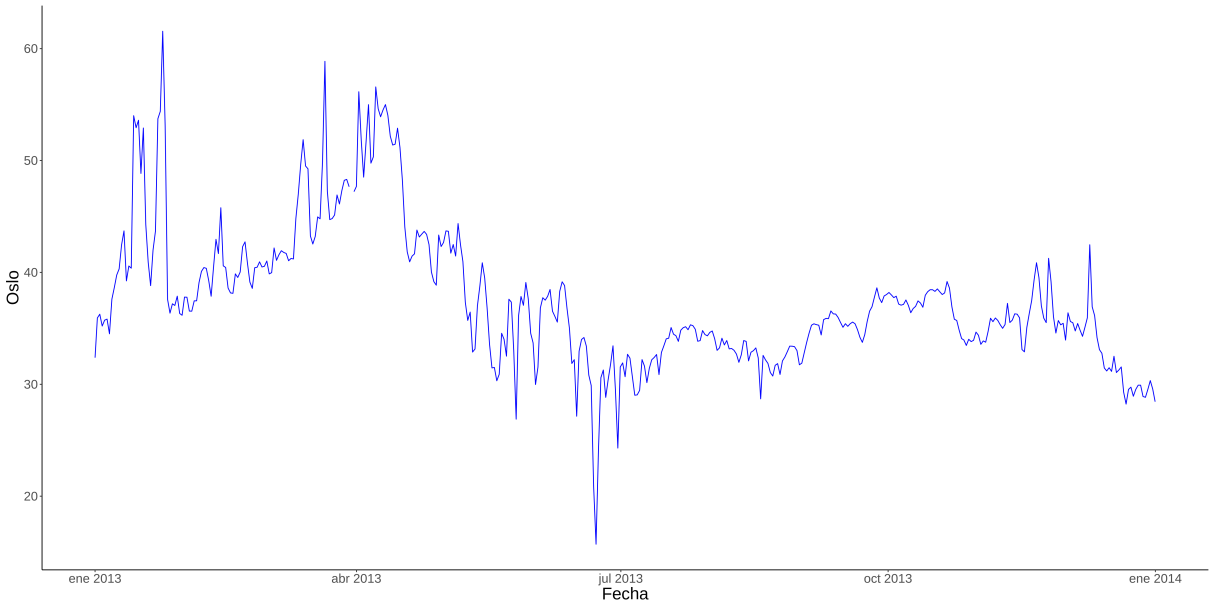


Figura 1.2: Serie temporal Oslo año 2013

En estos datos nos son facilitados precios (euros) por MWh intradía (horarios) sobre un total de 8 países y 16 áreas; podemos obtener mínimos, máximos, media y último valor para cada hora. Hay además información sobre volumen, flujo entre áreas, capacidades, etc; en este caso nos centramos prioritariamente en la serie de precios por MWh para la zona de Oslo.

Capítulo 2

Fundamento Teórico

A continuación, vamos a dar unas nociones básicas sobre uno de los más sencillos métodos de predicción, el suavizado exponencial; luego hablaremos sobre el tratamiento matemático y estadístico tanto de los modelos ARIMA como de los SSM, con especial atención a las variantes estacionales. Después, haremos una comparativa entre ambos modelos, con sus ventajas e inconvenientes, y por último, mencionaremos otras posibilidades de análisis estadístico y que quedan fuera del alcance de este proyecto.

2.1. Suavizado Exponencial

Este algoritmo es el primer intento de predicción de series temporales; no es un modelo estadístico en sí, sino que es tan solo un método para obtener predicciones puntuales [Hyn+08]. Por esta razón no es posible obtener distribuciones, intervalos de confianza, etc pues desde un principio este método no trata con variables aleatorias [1]

Si bien hay muchas variedades, la idea principal de estos algoritmos es el dar un peso exponencialmente decreciente a la información alejada en el tiempo; de esta manera actualizamos a cada paso los parámetros del modelo dando más importancia, en la estimaciones de los parámetros, a las observaciones más recientes. El efecto que esto produce es de un "suavizado" de la serie original, característica origen de su nombre.

Un ejemplo moderadamente complejo de estos métodos es el suavizado estacional aditivo de Winters; cuyo método, resumiendo, consiste en la descomposición de la serie en una componente de nivel (μ_i), otra de tendencia (β_i) y otra estacional (s_i) que se actualizan con cada nueva observación:

[1] Aun así, es posible obtener/adaptar modelos estadísticos a cada tipo de suavizado exponencial con la mismas predicciones puntuales [Hyn+08].

$$X_{n+j} = (\mu_n + j\beta_n) + s_{n+j} + z_{n+j}$$
$$\sum_{i=1}^{s} s_i = 0$$

Los valores de la descomposición se inicializarían[2] y a partir de ahí se estimarían con el método de mínimos cuadrados ponderados o, en la práctica, con una versión recursiva de este.

2.2. ARIMA

Los modelos ARIMA o Autoregressive Integrated Moving Average son una generalización de los más sencillos modelos ARMA para tratar series temporales no estacionarias; añaden a la representación ARMA (explicada brevemente a continuación) un coeficiente (o varios) de diferenciación ($\Delta X = X_t - X_{t-1}$) para conseguir la deseada estacionaridad.

Son uno de los modelos más usados para la predicción de series temporales y existe toda una metodología (el método de Box-Jenkins) para seleccionar y estimar el modelo ARIMA apropiado.

2.2.1. ARMA

Los modelos ARMA o Autoregressive Moving Average describen una serie estacionaria a través de dos polinomios en B distintos, donde B es el operador de retardos; por un lado el de autoregresión (AR), que expresa una variable en términos de sus valores previos; y por otro el de media móvil (MA), que expresa dicha variable en término de una combinación lineal de los errores previos.

$$X_t = c + \sum_{i=1}^{p} \gamma_i X_{t-i} + \epsilon_t$$
$$X_t = \mu + \sum_{i=1}^{q} \lambda_i \epsilon_{t-i} + \epsilon_t$$

Por lo tanto la combinación de ambos es lo que se conoce como el modelo ARMA(p,q), con p y q los coeficientes de los polinomios en el operador de retardos.

$$X_t = c + \sum_{i=1}^{p} \gamma_i X_{t-i} + \sum_{i=1}^{q} \lambda_i \epsilon_{t-i} + \epsilon_t$$

[2]La inicialización de los parámetros es un problema compartido con los SSM con "inicio finito" y su tratamiento para los SSM se comentará más adelante.

2.3. Modelos de Espacio de Estados

Estos son una clase de modelos que describen un conjunto de variables aleatorias descomponiéndolas en dos partes; por un lado las variables latentes, ocultas o de estado y por el otro las variables visibles que representa las observaciones o medidas de la serie. De esta manera podemos separar el calculo de ambas; tanto el estado del modelo como la observación de la serie solo dependen del valor del estado durante el momento previo junto con un error aleatorio .

Ejemplo de un SSM con tendencia local:

$$y_t = l_{t-1} + b_{t-1} + \epsilon_t \tag{2.1}$$
$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t \tag{2.2}$$
$$b_t = b_{t-1} + \beta\epsilon_t \tag{2.3}$$

Aquí es importante separar la descripción de las variables de la estimación de estas, los SSM son modelos estructurales que simplemente representan unas variables observables a través de otras ocultas; otro ejemplo de esto mismo serían las cadenas ocultas de Markov; lo importante radica en qué subconjunto específico de estos modelos nos centremos y de qué manera llevamos a cabo la estimación de los parámetros y las predicciones.

Así pues, nosotros restringimos nuestro estudio a aquellos modelos que poseen las siguientes restricciones:

- Variables de observación y estado continuas
- Variables con tiempo discreto
- Observación univariante
- Se cumple la propiedad de Markov
  - La distribución del estado t+1 solo depende del estado t
  - La distribución de la observación t+1 solo depende del estado t
- La distribución de la observación solo depende del estado
- Sin variables predictoras/independientes extra

Además, en general trataremos con la versión lineal (aunque también mostraremos la generalización al modelo no lineal, por ejemplo útil para modelos con error multiplicativo) y nos limitaremos a la estimación y generación de predicciones a través del filtro de Kalman y sus inmediatos derivados.

2.3.1. Fundamento

En esta sección damos un tratamiento más riguroso a la definición de modelos de Espacios de Estados. En total tenemos tres secuencias diferentes de variables aleatorias: y_t corresponde a las observaciones de la serie, x_t corresponde al estado y ϵ_t es nuestro ruido, todas y cada una para el momento/tiempo t.

A continuación mostramos el modelo general lineal con la formulación de innovaciones:

$$y_t = \mathbf{w^t x_{t-1}} + \epsilon_\mathbf{t} \tag{2.4}$$
$$\mathbf{x_t} = \mathbf{F x_{t-1}} + \mathbf{g}\epsilon_\mathbf{t} \tag{2.5}$$

La primera ecuación se suele conocer como ecuación de medidas o de observación y la segunda como ecuación de transición o de estado. Además, esta es la que llamamos formulación de innovaciones o de SSOE,[3] como se observa por la existencia de un solo error para ambas ecuaciones (ϵ_t). En el siguiente apartado comentaremos las similaridades y diferencias con otra formulación, la de MSOE[4].

SSOE vs MSOE

Estas son las dos variantes usadas en los SSM, primero tenemos la formulación SSOE; se llama así porque en las ecuaciones tenemos como origen de error ϵ_t y $\mathbf{g}\epsilon_\mathbf{t}$ las cuales están perfectamente correlaccionadas.

En cambio en la formulación MSOE, actualmente más usada, tenemos dos series de errores diferentes, ϵ y α, con la restricción añadida de que sean perfectamente independientes (matriz de varianzas-covarianzas diagonal).

$$y_t = \mathbf{w^t x_{t-1}} + \epsilon_\mathbf{t}$$
$$\mathbf{x_t} = \mathbf{F x_{t-1}} + \alpha_\mathbf{t}$$

Si bien de un primer vistazo parece que esta segunda versión tiene una mayor capacidad de modelado por ser más genérica (dos errores en vez de uno), estas son en verdad equivalentes[5][Hyn+08].

Ambas formulaciones tienen sus ventajas y desventajas, en nuestro caso trataremos siempre con la versión SSOE, a menos que se diga lo contrario.

[3] Single Source of Error o fuente única de error.

[4] Multiple Sources Of Error o múltiples fuentes de error.

[5] Solo en el caso de errores gaussianos, no es así para otras distribuciones pero esta es la más comúnmente usada

Generalización no lineal

Para trabajar sobre modelos multiplicativos (por ejemplo $y_t = l_{t-1}(1 + \epsilon_t)$) no es suficiente el anterior modelo y es necesario utilizar una versión más general del modelo:

$$y_t = w(\mathbf{x_{t-1}}) + \mathbf{r}(\mathbf{x_{t-1}})\epsilon_\mathbf{t}$$
$$\mathbf{x_t} = \mathbf{f}(\mathbf{x_{t-1}}) + \mathbf{g}(\mathbf{x_{t-1}})\alpha_\mathbf{t}$$

Como se ve, son paralelas a las previas, sustituyendo las matrices por funciones arbitrarias. Por desgracia el filtro de Kalman básico no es capaz de trabajar con esta versión y es necesario una modificación del algoritmo original para trabajar con estos modelos no lineales.

Inicio finito o Infinito

Los valores de los estados han de ser inicializados antes de poder aplicar el filtro de Kalman y existen dos maneras de modelar esta información a priori. La más simple es la usada en modelo de tipo ARMA o ARIMA, se considera que la serie comienza en ese momento (justo al inicio de los datos) y por tanto tan solo debemos establecer la semilla/el valor inicial de los estados; aquí llegamos al problema de establecer cual debe ser dicho valor para cada estado. Podemos usar estrategias heurísticas, estimar usando máxima verosimilitud, etc. Esta es una metodología útil y con suficientes observaciones el valor dado pierde importancia, aun así se puede mejorar.

$$\mathcal{L}^*(\theta, \mathbf{x_0}) = \mathbf{n}\log(\sum_{\mathbf{t=1}}^{\mathbf{n}} \epsilon^\mathbf{2}) + \mathbf{2}\sum_{\mathbf{t=1}}^{\mathbf{n}} \log|\mathbf{r}(\mathbf{x_{t-1}})|$$

Ecuación 1: Verosimilitud del modelo para estimar
los estados iniciales o cualquier otro parámetro desconocido

Por otro lado, podemos modelar dichos estados iniciales como si la serie hubiera comenzado en un pasado indefinido y por tanto dichos valores son variables aleatorias. Esto provoca que en vez de tener una predicción puntual de los estados, que es en la práctica una combinación lineal de los estados iniciales y de las observaciones, tenemos unas variables con una distribución desconocida que es necesario estimar. Esto conlleva la necesidad de métodos como el filtro de Kalman, que estiman recursivamente la distribución (los momentos de una distribución gausiana en realidad) de la variable/s estado/s.

2.3.2. Estimación de parámetros desconocidos

Usualmente el modelo creado tiene uno o varios parámetros desconocidos, e.g. las varianzas de los errores, y es necesario estimarlos antes de comenzar las predicciones de la serie, dado que sí bien es posible calcular las distribuciones a posteri, directamente con dichos parámetros desconocidos, llevando a cabo análisis bayesiano, en general es más común estimar primero los valores óptimos a través de estimadores de máxima verosimilitud o algún otro método similar.

En un principio esta etapa es sencilla, pues generalmente podemos obtener la verosimilitud del modelo, pero en la práctica encontramos problemas técnicos en la optimización, por la relativa "tosca" superficie que lleva en muchos casos a encontrar máximos locales o porque decidimos usar alguna otra función de coste en la optimización, e.g. la suma de residuos al cuadrado.

2.3.3. Filtro de Kalman

Una vez tenemos preparado el modelo completo que vamos a usar con todos los parámetros calculados, necesitamos llevar a cabo la estimación de las distribuciones de la serie; tenemos que establecer, a partir de las observaciones y de los estados iniciales, cual es la distribución del estado a cada momento t del tiempo. Tenemos además que poder predecir, dado un estado y toda la serie previamente observada, la distribución predicha de la siguiente observación (la del momento $t + 1$).

Para realizar esto se creó el filtro de Kalman, el cual itera a cada instante t calculando a cada paso los momentos (esperanza y varianza) de las variables de estado a partir de los valores anteriores. Decimos los momentos y no la distribución completa pues se simplifica así el calculo; además, si la distribución es Gaussiana, con tan solo la media y la varianza tenemos definida la distribución completa. Si quisiéramos estimar distribuciones arbitrarias deberíamos recurrir al filtro bayesiano (el cual bajo errores y prior gaussiana es equivalente al de Kalman).

Filtrado vs Suavizado

Estos términos se refieren a que tipo de predicciones de la serie realiza el filtro de Kalman. En concreto, sobre qué variables se está condicionando la distribución de la serie en el momento $t + 1$; sí solo se condiciona sobre las observaciones hasta el momento t, es decir, solo sobre la información actual, se llama filtrado. Si, en cambio, condicionamos sobre la totalidad de las observaciones, esto no vale ya para realizar predicciones (pues estamos usando información futura), y es conocido como suavizado de Kalman.

$$P(x_t|y_t, ..., y_1) \quad \text{vs} \quad P(x_t|y_n, ..., y_t, ..., y_1)$$

Fundamento Teórico

La construcción del filtro es muy similar a un proceso de condicionamiento recursivo bayesiano. Se basa en dos etapas (además de la etapa de inicialización durante la primera iteración) que se llevan a cabo a cada paso de algoritmo. Primero una etapa de predicción en la cual partimos con los momentos del vector de estado $x_{t-1|t-1}$[6] y a partir de este y usando técnicas estándar de estadística, obtenemos los momentos de $x_{t|t-1}$ con los que conseguimos predecir la siguiente observación.

$$\mathbf{m}_{t|t-1} = \mathbf{F}\mathbf{m}_{t-1|t-1}$$
$$\mathbf{V}_{t|t-1} = \mathbf{F}\mathbf{V}_{t-1|t-1}\mathbf{F}^t + \sigma^2\mathbf{g}\mathbf{g}^t$$

A continuación, en el en el momento t en el que observamos el valor de la serie, pasamos a la etapa de revisión en la cual actualizamos, con dicha observación, nuestras estimaciones de los momentos. Esto se lleva a cabo siguiendo las reglas generales de probabilidad condicionada a otra variable aleatoria.

$$\mathbf{m}_{t|t} = \mathbf{F}\mathbf{m}_{t-1|t-1} + \mathbf{k}_t(y_t - \mu_{t|t-1})$$
$$\mathbf{V}_{t|t} = \mathbf{V}_{t|t-1} - v_{t|t-1}\mathbf{k}_t\mathbf{k}_t^t$$
$$\mathbf{k}_t = \varsigma_{t|t-1}v_{t|t-1}^{-1}$$

Sin entrar en demasiados detalles, aquí es donde aparece la famosa ganancia de Kalman, que no es sino el valor α que solemos usar en el suavizado exponencial pero teniendo en cuenta información extra[7]

Filtro de Información

El filtro de kalman tiene varias limitaciones y una de ellas es que es necesario establecer la distribución a priori de los estados. En el caso gausiano esto significa establecer la media y la matriz de varianzas-covarianzas. No es así posible, en principio, inicializar los estados con una prior difusa, i.e. con una varianza infinita. Además tiene similares problemas al tratar con series no estacionarias.

Como solución se plantea el filtro de información. De manera resumida, en vez de trabajar directamente con los momentos modela una transformación lineal en la cual la matriz de varianzas-covarianzas es diagonal, es decir, una transformación en la cual tengamos variables independientes [8].

[6] Trataremos $x_{a|b}$ como $P(x_a|y_b, ..., y_1)$

[7] Se puede demostrar que este valor, al infinito y bajo ciertas condiciones, tiende a un valor fijo k [Hyn+08]; esto nos dice que el filtro de Kalman tendería al suavizado exponencial.

[8] Esto se basa en descomposición de matrices, triangulación de matrices estocásticas, transformaciones rápidas de Givens...

2.3.4. Extensiones

Para sobrepasar los problemas básicos del filtro de kalman se han ido desarrollando diversas modificaciones al algoritmo original; a continuación describimos brevemente aquellas que más nos interesan para este trabajo.

Inicio difuso exacto

Si bien el algoritmo original no puede tratar con prior difusas/no informativas, sí existen varias modificaciones centradas en superar dicho problema. Una de ellas es el inicio difuso exacto que, simplificando, divide el filtrado en dos fases. La primera, de inicialización, aumenta la matriz de covarianzas con otra "difusa"; con esta toma suficientes periodos como para realizar la primera estimación de los momentos difusos; a partir de ahí, en la segunda etapa se desarrolla el filtro de Kalman ya explicado anteriormente [Sko11].

2.4. Comparativa

Todas estas representaciones de las que hemos estado hablando hasta ahora están muy relacionadas. Todo ARIMA puede ser trasformado a un SSM y todo SSM (bajo ciertas condiciones) converge a un modelo de suavizado exponencial. En cambio, no todo SSM puede ser expresado como un modelo ARIMA; es así pues, una generalización que nos da, en teoría, una mayor flexibilidad al crear nuestro modelo.

Otra diferencia entre los modelos ARIMA y SSM es que estos últimos son más explanatorios, es decir, al modelar directamente los estados es posible analizar la construcción del modelo, interpretar los coeficientes, cambiar módulos/subconjuntos del modelo sin tocar otros, etc. Los modelos ARIMA en cambio son cajas negras sobre las que perdemos varios grados de control.

Por otro lado, a través del suavizado exponencial podemos trabajar con un gran conjunto de los modelos estructurales; pero, a diferencia de los SSM, no permite manejar otras distribuciones ni modelos con matrices arbitrarias.

2.5. Otras posibilidades de los SSM

Tan solo hemos tocado la superficie de estos modelos, limitándonos a una versión muy especifica con claras restricciones. Si, por ejemplo, permitimos que la variable de tiempo sea continua, tenemos las ecuaciones diferenciales estocásticas; y si al revés, limitamos todo a variables discretas, tenemos las cadenas ocultas de Markov. Incluso dentro de los modelos estructurales estimables a través del filtro de Kalman podemos encontrar muchas variaciones.

De otra manera, podemos trabajar con el mismo subconjunto de modelos pero con

otro enfoque; e.g. en vez de estimar parámetros y calcular distribuciones con el filtro de Kalman, estimamos directamente distribuciones a posteriori arbitrarias a través de un filtro bayesiano.

Como se observa, el estudio de series temporales y de los modelos de espacios de estados es increíblemente amplio y se requeriría un proyecto de mucho más alcance para si quiera comenzar a estudiar todas las posibilidades.

Parte II

Análisis librerías de R

Capítulo 3

Librerías

A continuación vamos a dar una breve introducción a R y a las librerías más usadas en el estudio de SSM. Además vamos a dar una explicación en profundidad de tres de ellas con ejemplos de su uso práctico. Posteriormente compararemos, con distintas valoraciones e índices, dichas librerías y daremos finalmente una conclusión resumiendo los resultados.

3.1. Introducción

Durante todo este proceso vamos a trabajar sobre el lenguaje estadístico R; actualmente es uno de los lenguajes más usados junto con Python para estadística e inteligencia artificial. Es un lenguaje maduro, pues ha estado en uso desde 1991, cuando fue creado como un dialecto de S[1] por Ross Ihaka y Robert Gentleman[Pen19]. Desde entonces, y gracias a su estatus como código abierto, cientos de personas han contribuido con librerías para incrementar las capacidades del lenguaje.

Si bien el estudio de los SSM no es excesivamente común, sí que existen diversas librerías que trabajan con ellos desde diferentes puntos de vista. Esto es a la vez una ventaja y un inconveniente, pues aunque la variedad facilita el encontrar una que se ajuste a nuestras necesidades, dada la relativa novedad de este tipo de modelos, estas librarías aun no están lo suficientemente estandarizadas y no hay ninguna que ofrezca todas las posibilidades/características.

Paquetes que realicen diversas tareas relativas a estos modelos son **KFAS**, centrado alrededor del filtro de Kalman; **dlm**, centrado en análisis bayesiano pero con funciones de verosimilitud y filtro de Kalman; **sspir**, totalmente implementado en R y con interfaz tipo glm pero, por desgracia, abandonado actualmente; **dse**, el más antiguo de todos con funciones para ARIMA y SSM, con filtro de Kalman y estimación de máxima verosimilitud; **MARSS**, bastante reciente y que plantea la forma más general del modelo estructural; y **FKF**, centrado en ser la implementación más rápida del filtro de Kalman.

[1] Lenguaje previo desarrollado hacia 1976 por Bell Labs.

	dse 2009.10-2	**sspir** 0.2.8	**dlm** 1.1-1	**FKF** 0.1.0	**KFAS** 0.6.0
Coded in:	R + Fortran	R	R + C	R + C	R + Fortran
Class model (if any)	S3	S3	S3	S3	
Algorithm	CF	CF	SRCF	CF	CF
Sequential processing					√
Exact diffuse initialization					√
Missing values in $\boldsymbol{y_t}$ allowed			√	√	√
Time varying matrices		√	√	√	√
Simulator	√		√		
Smoother	√	√	√		√
Simulation smoother		√	√		√
Disturbance smoother					√
MLE routine	√	√	√		
Non-Gaussian models		√			√

Figura 3.1: Comparación de varias librerías de R - Fuente "Kalman Filtering in R"[Tus11]

Como se ve, existen una multitud de librerías cada una con un objetivo ligeramente distinto, con distintas formulaciones de los modelos, distintas implementaciones del filtro, distintos lenguajes de implementación, con distintas capacidades básicas (inicializado difuso, suavizado+filtrado, estimación de máxima verosimilitud, etc)...

Teniendo en cuenta las características básicas, la complejidad y las diferentes capacidades de los paquetes, hemos elegido tres que describiremos en profundidad y que compararemos entre sí.

3.2. KFAS

KFAS[Hel17] o Kalman Filtering And Smoothing[2] es un paquete que permite no solo trabajar con modelos de espacios de estados lineales generales, sino que también es posible usar otras familias exponenciales, como la Poisson, la Binomial Negativa... Como su nombre indica, esta centrado alrededor del filtro de Kalman

Esta escrito totalmente en Fortran (excepto la interfaz), lo que lo hace relativamente veloz; usa el modelo de múltiples errores o MSOE; permite realizar inicialización difusa y usar matrices que varíen con el tiempo.

Funciona de la siguiente manera, primero se necesita crear un modelo con la función **SSModel**, la cual puede tomar fórmulas sencillas, uniendo componentes polinómicas –**SSMtrend** –, componentes estacionales –**SSMseasonal**–; o directamente construyendo las matrices de manera manual con **SSMcustom**. Si además se necesita crear modelos

[2]Suavizado y filtrado de filtro de Kalman.

cuyas matrices varíen en el tiempo, es posible añadiendo, de manera manual, una tercera dimensión a las matrices del sistema.

A partir de aquí, si es necesario estimar algún parámetro desconocido del modelo, se puede optimizar directamente sobre la verosimilitud con **logLik** o, para modelos sencillos, aprovechar la función otorgada por la librería **fitSSM**.

Por último, se calculan las predicciones a un paso (filtrado) con la función **KFS**. Se pueden también obtener las predicciones de los estados además del suavizado de ambos. Si se desea obtener las predicciones a diferentes retardos, 2,4,24,etc; es necesaria la función **predict**.

Un ejemplo del uso de la librería para un modelo simple con tendencia y estacionalidad diaria para datos horarios es el siguiente:

```
1  variances <- c(NA, NA, NA)
2  # SSModel crea el modelo
3  model <- SSModel(Oslo ~
4              SSMtrend(2, Q=list(matrix(variances[1]), matrix(variances[2])))+
5              SSMseasonal(24, Q=matrix(variances[3]), sea.type = "dummy"), data=data)
6  # fitSSM usa optim para encontrar los parametros NA
7  model <- fitSSM(model, c(1,1,1))$model
8  # KFS calcula las predicciones con el filtro de kalman
9  estimates <- KFS(model, filtering = c("state", "signal"), smoothing= "none")
10
11 predictions<-predict(model, n.ahead=24)
```

Se observa como se crea un modelo con intercept y tendencia (polinomio de grado 2) en la linea 4; y una estacionalidad diaria con un ciclo de 24 horas, con 3 varianzas en total a estimar (NAs en la linea 1). Se estiman las varianzas (linea 7) y se obtienen las estimaciones o filtros, tanto del estado como de las predicciones o respuesta (linea 9). Por último en la linea 11 se obtienen las últimas predicciones para todo un periodo completo a partir de la última observación.

3.3. dlm

Dlm[Pet10] o Dynamical Lineal Models[3] es una librería centrada en el análisis bayesiano de estos modelos, con herramientas para filtrar y suavizar con el filtro de Kalman además de herramientas de Markov Chain Montecarlo (MCMC) para la estimación de distribuciones posteriores. Por último, aunque se centra en el estilo bayesiano, también facilita la estimación de parámetros desconocidos a través de estimadores de máxima verosimilitud.

Como KFAS, está escrito casi totalmente en otro lenguaje, en este caso C, para acelerar el cómputo. Usa similarmente el modelo MSOE; no permite inicialización difusa ni modelos no gausianos, pero si matrices que varíen con el tiempo.

[3]Modelos dinámicos lineales, otro nombre para los modelos de espacio de estados.

Su funcionamiento básico es el siguiente, primero se genera un modelo con la función **dlm**, de manera manual con matrices, o con las funciones **dlmModPoly**, **dlmModSeas**, etc, que permite unir componentes diferentes en un modelo más grande. Para crear matrices que varíen en el tiempo es necesario realizarlo manualmente usando las matrices extra **JGG**, **X**...

Ahora, si se desea estimar algún parámetro se puede usar la función **dlmMLE** sobre un modelo cualquiera, siendo una simple cubierta de la función **optim**.

Por último, se pueden calcular los diferentes filtros y suavizados requeridos, con las funciones **dlmFilter** y **dlmSmooth**.

A continuación mostramos un ejemplo de uso de la librería para implementar un modelo con tendencia y estacionalidad diaria para datos horarios:

```
1  # Creamos el modelo sumando componentes
2  model <-  dlmModPoly(order=2, dV= 1) +
3            dlmModSeas(frequency = 24, dV = 1)
4  # Estimacion de las varianzas, retorna los mejores parametros
5  model_fit <- dlmMLE(data$Oslo, c(1,1), build = function(x){
6            dlmModPoly(order=2, dV= x[1]) +
7            dlmModSeas(frequency = 24, dV = x[2])
8  })
9  # Predicciones a un paso y residuales
10 model_filter <- dlmFilter(data$Oslo, model)
11 model_residuals <- residuals(model_filter)$res
```

Como vemos, la creación del modelo es sencilla, tan solo sumamos las componentes requeridas (válido siempre que el modelo no sea muy complejo); luego estimamos varianzas (lineas 5-7) las cuales usamos para completar la creación del modelo original. Por último, estimamos las predicciones y obtenemos los residuos del modelo (lineas 10-11) que podemos analizar a posteriori.

3.4. dse

Dse[Gil95] o Dynamic System Estimation[4] es una librería de análisis de series temporales con modelos ARMA y State Space, con capacidad para pasar de una formulación a otra. Es un paquete algo teórico, sin excesivas funcionalidades, pero su atractivo es que es capaz de tratar a la vez dos formulaciones de los SSM, SSOE y MSOE.

Como la librería KFAS, está escrito en Fortran. Además tiene rutinas para la estimación de parámetros y para filtrado y suavizado, pero no acepta matrices que varíen en el tiempo, una limitación muy severa en comparación con las otras dos librerías.

Como los otros paquetes sigue un patrón de creación del modelo, estimación de parámetros y predicción. La creación de SSM es exclusivamente manual (otra limitación

[4]Estimación de sistemas dinámicos.

respecto de las otras librerías), por lo que se le deben dar todas las matrices del sistema a la función de creación del modelo para que funcione.

El proceso va así, primero se genera el modelo con la función **SS**, luego se estima con uno de los diversos procedimientos existentes, como **estMaxLik**, **estSSMittnik**, etc. Posteriormente, se usa la función **l** para pasar el filtro de kalman y obtener las predicciones; existen además una batería de métodos para analizar los resultados de dicha función.

Este sería un ejemplo del uso de la librería, las matrices se corresponden con las de un modelo con nivel, tendencia y estacionalidad diaria para datos horarios:

```
# Definicion de las matrices del modelo
alfa<-0.2;beta<-0.2;gamma<-0.8
Fm<-matrix(0,25,25)
Fm[1:2,1:2]<-matrix(c(1,1,0,1),2,2, byrow=TRUE)
Fm[3,3:25]<--1
Fm[4:25,3:24]<-diag(22)
K<-matrix(c(alfa,beta,gamma, rep(0,22)), 25,1)
H<-matrix(c(1,0,1,rep(0,22)),1,25)

# Creacion del modelo
model<-SS(F=Fm,K=K,H=H,G=NULL, names=list(output="oslo"))
# Estimacion
output <- l(model, data)
# Resultados
summary(output)
tfplot(output)
```

El proceso es similar a los anteriores, no entramos en demasiados detalles, pues varias construcciones de modelos han dado diversos problemas y, dadas las diversas restricciones del paquete, no vemos necesario un análisis más profundo.

3.5. Comparativa

A continuación hacemos una breve comparación entre las tres librerías a partir de la experiencia de uso que hemos tenido de cada una de ellas.

3.5.1. Características

De las tres, KFAS es de lejos la más completa. Aunque no facilita el análisis bayesiano como dlm, es más por diseño que por fallo del autor, pues no se centra en dicho tema. Es la única de las tres que facilita trabajar sobre modelos con errores no gaussianos y que permite la modificación del filtro de Kalman para la inicialización difusa.

La más restrictiva de todas es la librería dse, pues ni permite datos ausentes (NAs) ni permite matrices que varíen con el tiempo; esto último restringe mucho el conjunto de modelos que se pueden desarrollar con la herramienta.

Aun así, ninguna de las tres permite la creación de modelos no lineales, necesarios, por ejemplo, para crear modelos multiplicativos. Esto solo es posible yéndonos a paquetes mucho más complejos como dnyr fuera del alcance de este proyecto.

3.5.2. Coste computacional

Ahora, con el modelo presentado en los ejemplos de las librerías (es el mismo en las tres), vamos a estimar y guardar el tiempo que tarda cada una en terminar los cálculos.

Librería	Creación Modelo	Estimación	Filtrado
KFAS	0.011	50.0	0.56
dlm	0.035	135.0	4.47
dse	0.031	[5]	0.01

Cuadro 3.1: Comparativa de tiempo de cómputo (segundos)

Se observa como la librería KFAS es la más rápida, casi 3 veces más rápida que dlm en la estimación y más de 8 veces en el filtrado. En la creación del modelo todas las librerías son prácticamente instantáneas, lo cual es lo esperado pues es tan solo crear varias matrices de pequeño tamaño.

3.5.3. Facilidad de uso

Aquí compiten KFAS y dlm, ambas facilitan tanto la creación manual de modelos a través de las matrices, como la generación de modelos típicos a través de combinaciones de múltiples funciones más simples. Respecto a la claridad de la interfaz y la documentación en general, están de nuevo igualadas excepto quizás en algunos puntos de la interfaz que dlm supera a KFAS.

Dse queda bastante por detrás posiblemente debido a que es una librería bastante menos usada y testeada y que actualmente parece que se encuentra fuera de desarrollo.

3.6. Conclusiones

Comparando las tres, dadas las capacidades prácticamente equivalentes entre KFAS y dlm, con el añadido de que esta primera es capaz de trabajar sobre varias distribuciones y, además, dada la gran diferencia en tiempos de cálculo, nos decantamos por la librería KFAS para futuro trabajo/investigaciones. Es la más completa, flexible y veloz. Aun así, si se deseara trabajar con métodos bayesianos, seria necesario utilizar la librería dlm.

[5] Por imposibilidades técnicas se ve imposible obtener la estimación con dse

Para terminar, dse queda totalmente apartada pues no esta a la altura de las otras dos.

Parte III

Datos

Capítulo 4

Datos

A continuación vamos a dar una descripción en detalle de los datos que vamos a usar, de su formato y del trabajo de pre-procesado que debemos realizar. Después, vamos a comenzar con un análisis general de las características de los datos junto con una serie de gráficos que muestren con claridad dichas propiedades; luego, partiendo de dicho análisis, vamos a seleccionar un conjunto de modelos concordantes sobre los cuales, a posteriori, vamos a realizar una comparativa y así más tarde seleccionar el mejor. Por último daremos unas conclusiones del proceso que hemos llevado a cabo a lo largo del análisis.

4.1. Descripción

Como hemos comentado previamente, nuestros datos son un conjunto de $8761 = 24 * 365 + 1$ observaciones de una serie de precios (euros/MWh) en Oslo. Estos datos son muestras horarias que van desde el 1 de Enero de 2013 a las 00:00 hasta el 31 de Diciembre del mismo año a las 23:00; tenemos pues datos de cada hora y de cada día del año más una observación extra, correspondiente a la última hora del 2012 (el paso al siguiente día).

Los valores de los precios van desde 1.38 hasta 109.55 con una media del 37.56 y una varianza del 51.27; tan solo existe un NA en la fecha 31 de Marzo a las 02:00 (observación número 2139). Por desgracia, no podemos encontrar el motivo detrás de esta falta, no parece ser un fallo, pues la observación está bien construida, pero no tiene ningún valor en ninguna de las zonas; decidimos imputar el valor, será tratado en la sección de pre-procesamiento.

Formato

Obtenemos nuestra base de datos a través de una descarga en la pagina web de Noord-Pool[Nor], esta nos da un fichero llamado "elspot-prices_2013_hourly_eur" con formato html. En concreto, está establecido en forma de tabla estándar html, con <tbody>, <thead>, cada fila con una etiqueta <tr>, etc.

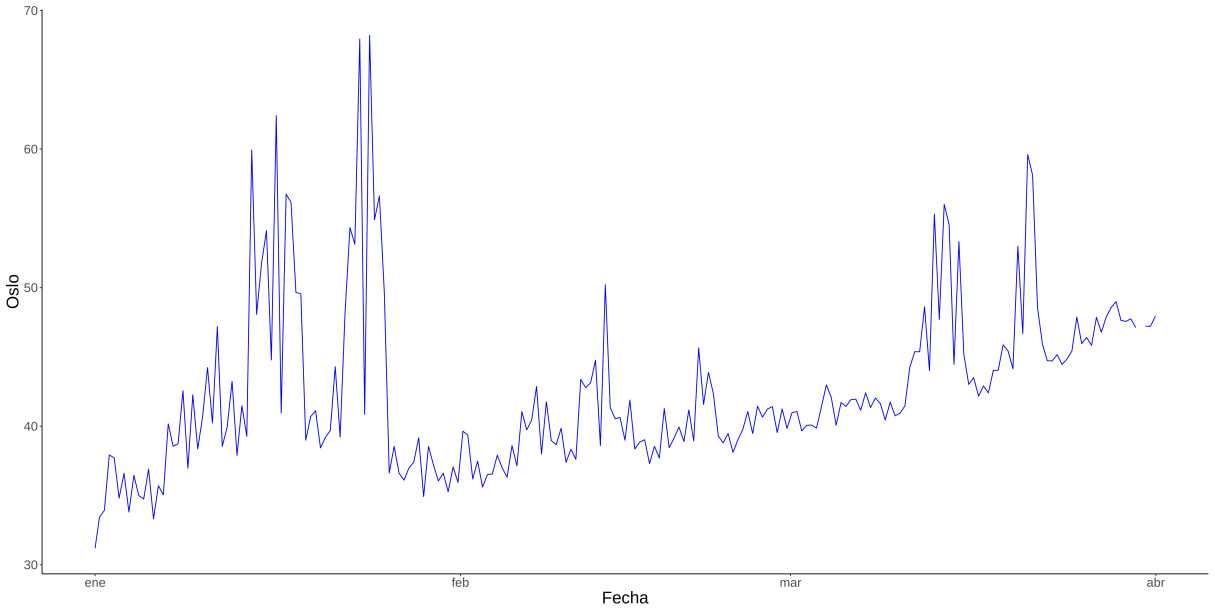


Figura 4.1: Serie temporal Oslo Enero a Marzo del año 2013

Además de las múltiples variables de precios, tenemos una fecha dividida en dos variables; una del día en formato DD-MM-YYYY y otra de hora en formato HH - HH. Así describe de que hora a que hora está calculado el precio medio (pues no tenemos acceso a todas las transacciones que ocurren en dicha hora).

Elspot Prices in EUR/MWh
Data was last updated 12-11-2015

Hours	SYS	SE1	SE2	SE3	SE4	FI	DK1	DK2	Oslo	Kr.sand	Bergen	Molde	Tr.heim	Tromsø	EE	ELE	LV	LT
01-01-2013 00 - 01	31,05	31,04	31,04	31,04	31,04	31,04	14,03	14,03	32,98	32,98	32,98	31,04	31,04	31,04	31,12	31,12		24,42
01-01-2013 01 - 02	30,47	27,51	27,51	27,51	27,51	27,51	11,06	11,06	32,97	32,97	32,97	30,81	30,81	30,81	30,61	30,61		23,62
01-01-2013 02 - 03	28,92	24,44	24,44	24,44	24,44	24,44	8,50	8,50	32,59	32,59	32,59	30,77	30,77	30,77	24,44	24,44		23,93
01-01-2013 03 - 04	27,88	21,81	21,81	21,81	21,81	21,81	0,10	0,10	31,53	31,53	31,53	30,71	30,71	30,71	21,81	21,81		23,85
01-01-2013 04 - 05	26,96	22,37	22,37	22,37	22,37	22,37	2,01	2,01	30,54	30,54	30,54	30,63	30,63	30,63	22,37	22,37		23,26
01-01-2013 05 - 06	27,84	25,51	25,51	25,51	25,51	25,51	22,81	22,81	29,93	29,93	29,93	30,58	30,58	30,58	25,51	25,51		23,54
01-01-2013 06 - 07	28,79	27,93	27,93	27,93	27,93	27,93	27,93	27,93	30,23	30,23	30,23	30,64	30,64	30,64	30,71	30,71		22,47

Figura 4.2: Muestra del fichero de datos html

4.1.1. Pre-procesamiento

El primer paso es trabajar sobre los datos originales para pasarlos a un formato más manejable, usamos la librería **xml2**[WHO20] y las funciones de trabajo sobre ficheros de tipo xml para extraer la tabla en formato *tibble* y guardarlo como *.rds* para futuro uso.

Ahora que podemos trabajar sobre los datos procedemos a tratar las fechas perdidas; como comentamos previamente en la serie Oslo tan solo hay una por lo que podemos pasar a completarla con una regresión sin peligro de introducir sesgo en los datos.

Como la serie es no lineal, una típica regresión lineal estaría muy alejada de la realidad; es por esto que llevamos a cabo una regresión local (loess) que tenga más en cuenta las

últimas observaciones en la predicción de nuestra observación perdida.

Por último, reservamos los 20 días finales ($20*24 = 480$ observaciones) para comprobar el error de generalización de los modelos al final del proyecto.

Ya acabado el pre-procesamiento, pasamos al análisis de los datos.

4.2. Análisis

Analizando las imágenes 1.2 y 4.1 observamos tendencias deterministas probablemente determinadas por los diferentes patrones de gasto energético a lo largo del año. Además, si nos fijamos en las continuas subidas y bajadas cíclicas podemos intuir la existencia de una periodicidad en nuestros datos.

Así pues, partimos con el análisis del correlograma (por limitaciones de página mostramos tan solo las primeras 500 auto-correlaciones). Este gráfico nos muestra primero la marcada no estacionaridad de la serie, pues carece de la estructura de decrecimiento exponencial y todas las correlaciones se salen de las bandas de confianza del 95 %. Vemos además una marcada tendencia así como periodicidad en los retardos múltiplos de 24 (ciclo diario, bandas rojas) y también en los retardos múltiplos de 168 (ciclo semanal, bandas verdes)

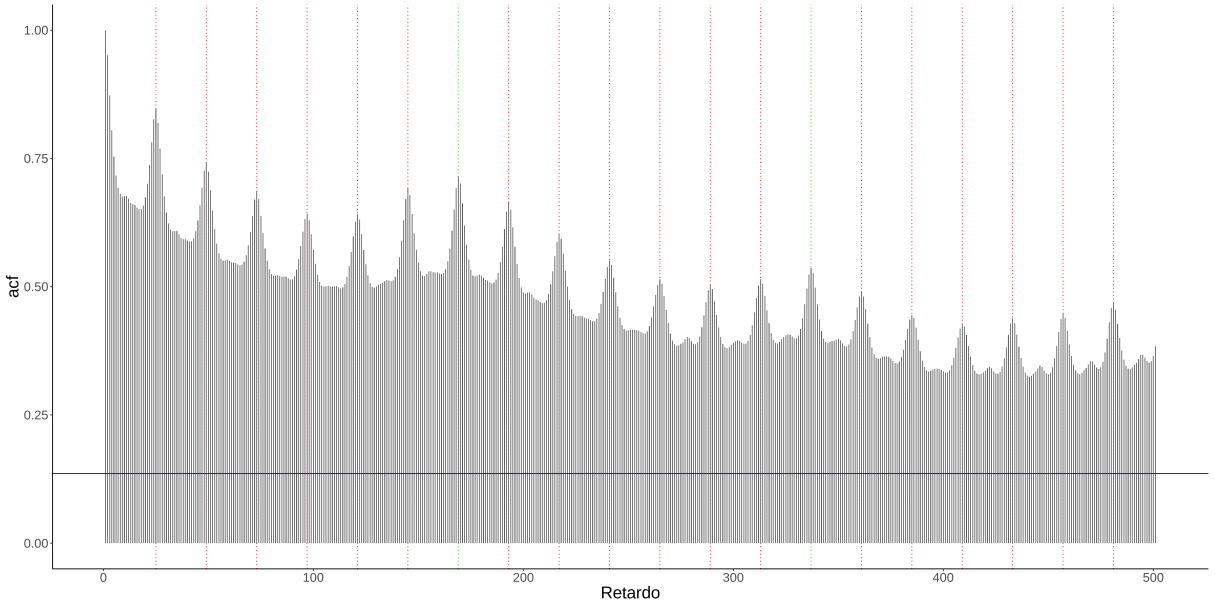
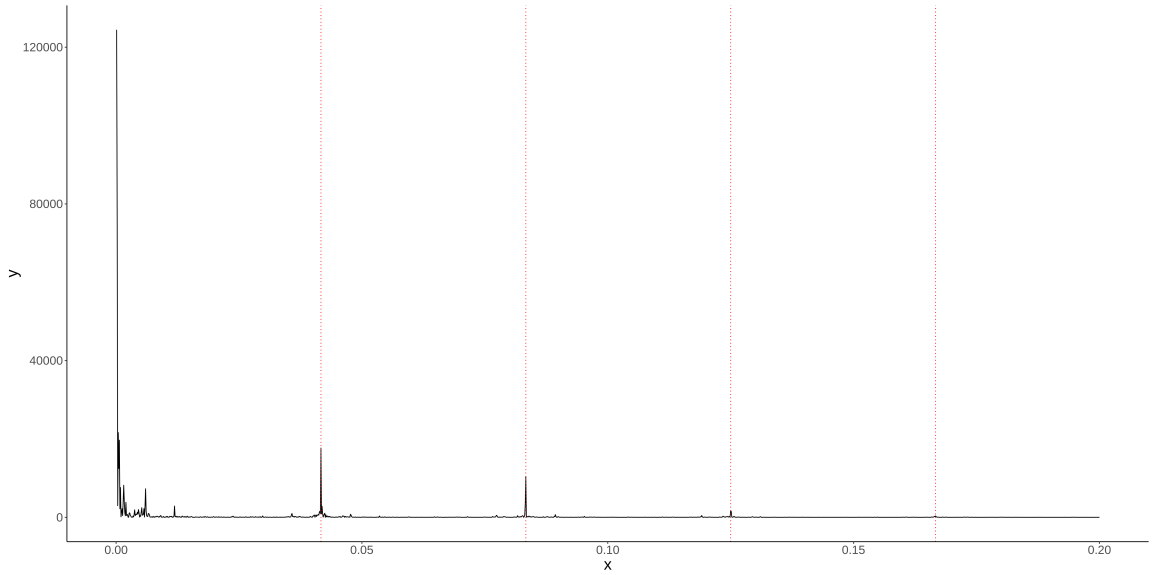


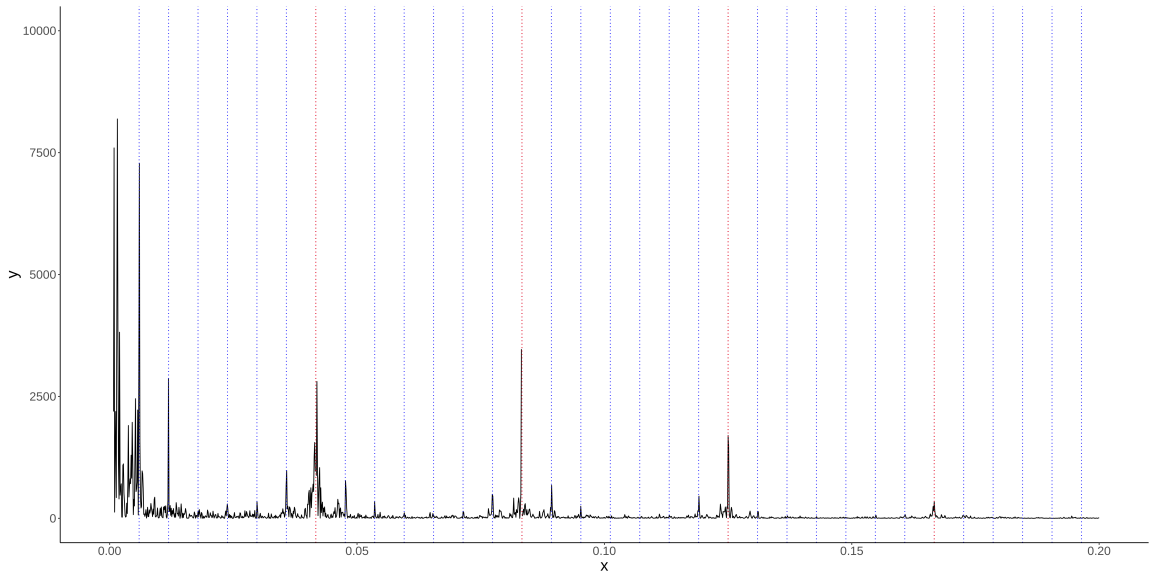
Figura 4.3: Primeras 500 auto-correlaciones de la serie Oslo

Pasamos al periodograma, el cual nos confirma lo anteriormente dicho, vemos como las frecuencias más bajas corresponden con ordenadas increíblemente altas lo que nos confirma la marcada tendencia de la serie; además vemos ordenadas relativamente altas

correspondiendo con los armónicos de las estacionalidades diarias (1/24, banda roja) y semanales (1/168, banda azul).



(a) Gráfico completo

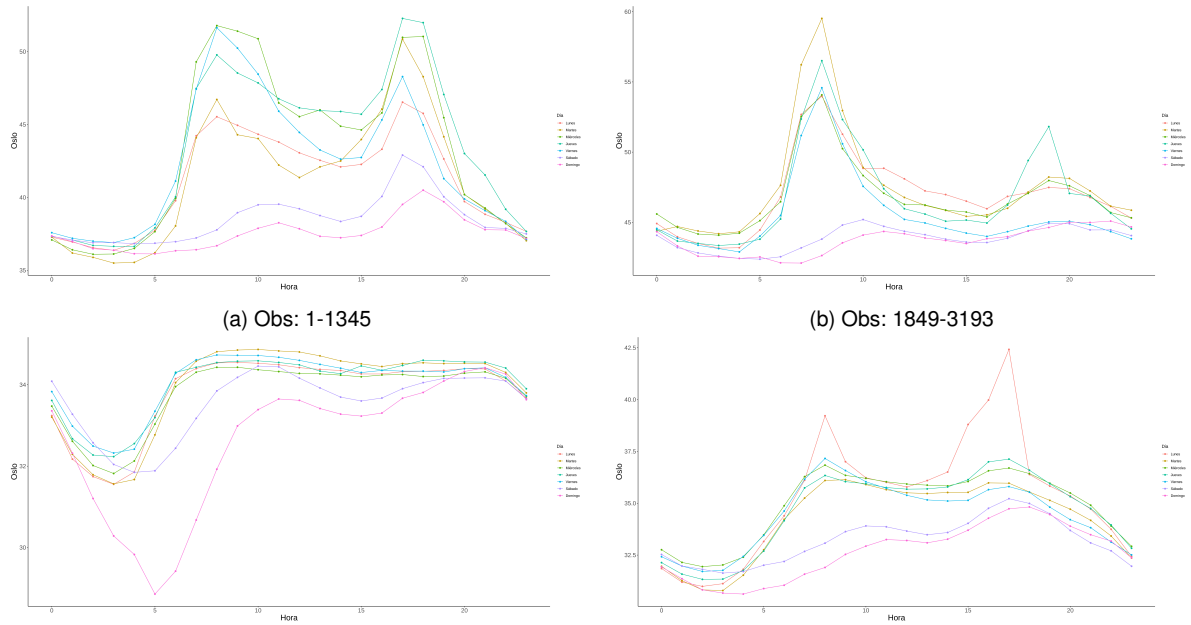


(b) Zoom del periodograma

Figura 4.4: Periodograma de los datos, los armónicos de 24 son bandas azules verticales y los de 168 bandas rojas

Continuamos con el análisis de la estacionalidad, nos interesa conocer exactamente cual es la interacción entre el patrón diario y el semanal, ver si cambia con los días de la semana y si es estable a lo largo del año; para ello creamos un gráfico que nos muestre, con cada día de la semana por separado, cual han sido los precios por hora. Para suavizar

la información llevamos a cabo una media de los precios durante 8 semanas consecutivas (alrededor de dos meses, un total de $8 * 7 * 24 = 1344$ observaciones).



(a) Obs: 1-1345 (b) Obs: 1849-3193

(c) Obs: 4873-6217 (d) Obs: 7417-8761

Figura 4.5: Patrones de precios energéticos horario-semanales

Primero observamos que existe un patrón de uso horario en el cual el precio energético (y por tanto el uso) cae durante la noche y se alza en dos picos hacia las 8 y las 16 horas, coincidiendo con los movimientos de la población (el despertar, horas de la comida, la vuelta del trabajo y las horas de sueño). No se mantiene igual a lo largo del año, observamos como en la tercera figura, coincidiendo con el verano, cambia completamente el patrón horario, se invierte y observamos un precio más constante con tan solo una bajada hacia las horas más profundas de la noche.

También observamos una interacción entre las estacionalidades, el patrón del fin de semana (Sábado y Domingo, morado y rosa respectivamente) es completamente diferente al del resto de los días de la semana con un coste mucho menor durante estos primeros días. Esto se mantiene incluso durante los meses de verano.

Análisis de la varianza

Acabamos el análisis básico de la serie estudiando su varianza. Para ello creamos el gráfico rango-media de la serie completa; los valores se calculan usando periodos completos de 24 observaciones (1 día).

Si bien la mayor parte de los puntos están agrupados en el centro y parecen estar distribuidos de manera uniforme, sí que observamos una tendencia al alza de los últimos datos,

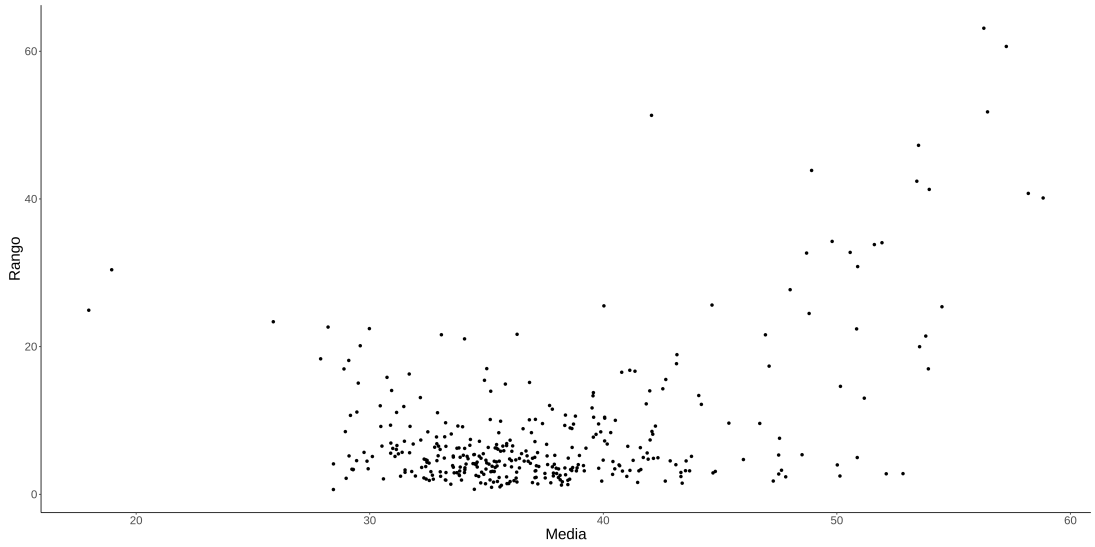


Figura 4.6: Gráfico rango-media de la serie Oslo

lo cual nos muestra una posible heterocedasticidad, en particular una mayor varianza a mayor sea el nivel.

Por esto consideramos realizar, o bien una transformación Box-cox sobre la serie, o usar directamente modelos multiplicativos.

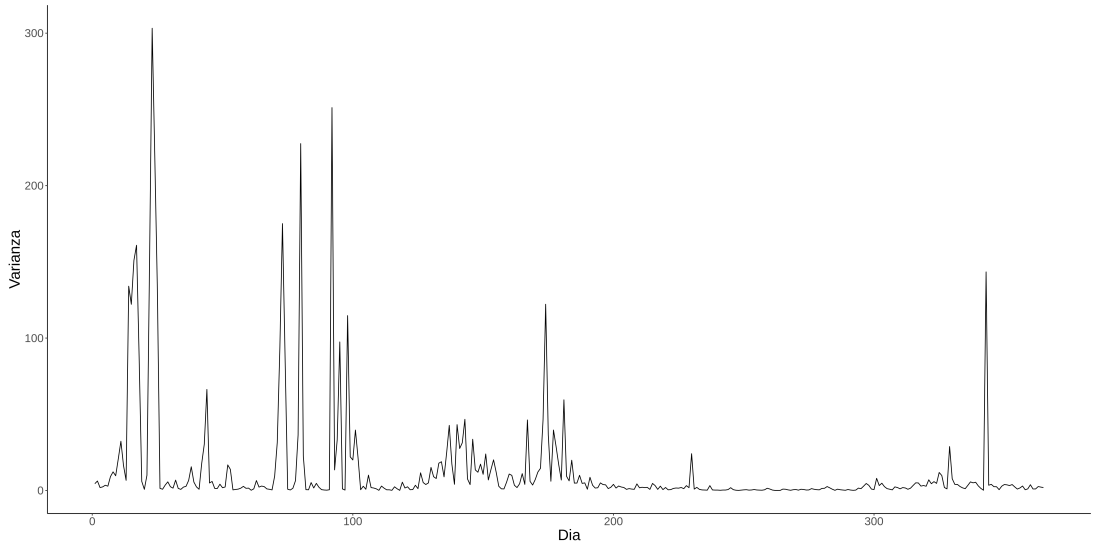


Figura 4.7: Gráfico de la varianza por día de la serie Oslo

Por último, si ponemos en un gráfico la variabilidad de cada día del año, advertimos como no solo hay heterocedasticidad, sino que esta se concentra en rachas sobre todo en las primeras épocas del año. Este fenómeno es muy común en series financieras como esta y se conoce como agrupamiento o clusterización de la volatilidad. En la figura 4.7

encontramos que la varianza no se reparte de manera uniforme en la totalidad de la serie, sino que se agrupa principalmente en tres rachas durante los primeros meses del año y otra más durante los últimos; Esto rompe nuestra suposición de errores gaussianos y requiere de una mayor especificación del modelo.

4.2.1. Selección de Modelos

Ahora que comprendemos mejor la estructura de la serie, pasamos a plantear diversos modelos plausibles para más tarde aplicarlos y ver cuales se adecúan de mejor manera a los datos. A menos que se diga lo contrario, todos los modelos estarán estimados con la librería KFAS y los estados se inicializarán de manera difusa.

Dentro de los SSM vamos a trabajar de manera incremental, comenzando con modelos más sencillos e incrementándolos sucesivamente; partimos de uno con intercept y tendencia y vamos a tomar distintas estacionalidades.

El proceso para cada candidato SSM será el siguiente, primero estableceremos brevemente con ecuaciones el modelo que deseamos crear, lo implementaremos y estimaremos los parámetros necesarios. Por último, analizaremos brevemente los residuos para decidir si es necesario un modelo más complejo.

Pero antes comenzaremos estimando un modelo base ARIMA.

Modelo ARIMA

De manera resumida, para crear el modelo ARIMA (SARIMA, en realidad, pues tenemos estacionalidad) comenzamos con el correlograma de la serie original y vamos diferenciando hasta que obtengamos patrón estacionario. Además, comprobamos la varianza para asegurarnos de no sobre-diferenciar.

El correlograma original se muestra en la figura 4.3 y la varianza de la serie es 48,3; si diferenciamos una vez simplificamos en gran medida el correlograma y obtenemos una varianza de 5.07; y si diferenciamos otra vez de manera estacional y diaria hacemos que casi todas las auto-correlaciones estén por debajo de la banda de confianza y que las primeras, tras las dos del principio, tengan un decrecimiento exponencial.

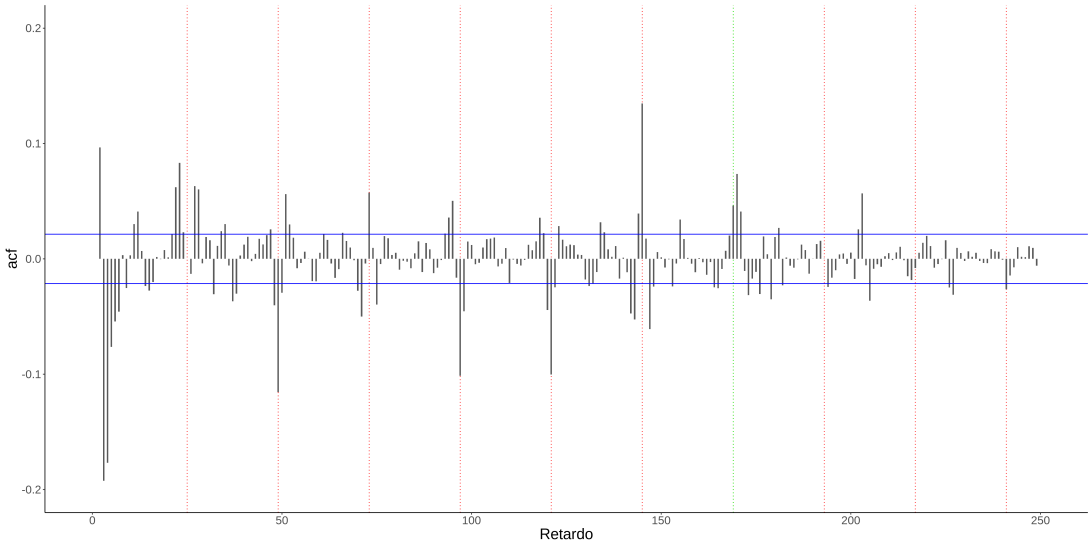


Figura 4.8: Correlograma de la serie diferenciada

La varianza, además, vuelve a disminuir a 4,58 pero en el correlograma siguen destacando las autocorrelaciones en retardos múltiplos de la estacionalidad diaria, que no desaparecen con sucesivas diferenciaciones. Por esta razón, nos detenemos con una diferenciación simple y otra estacional.

Pasamos a la propuesta de modelos, a través del estudio del ACF y PACF buscamos valores de p, q, ps y qs que encajen con la estructura de auto-correlaciones, tras varias pruebas nos decidimos en un SARIMA$(1,1,3)(2,1,0)_{24}$ sin ninguna transformación en los datos (los modelos sobre el logaritmo de la serie que analizamos poseen peor capacidad predictiva) y pasamos a validarlo.

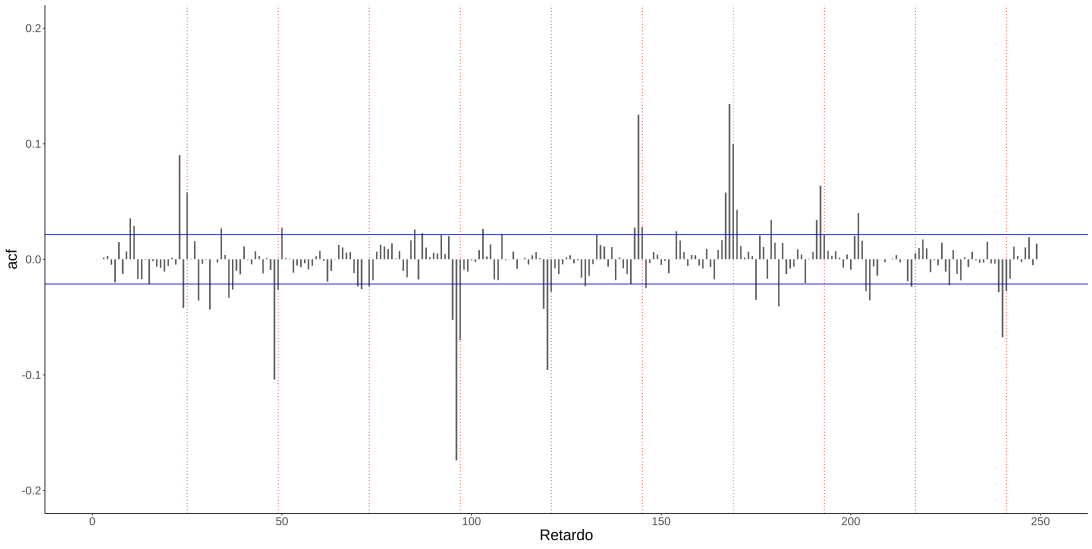


Figura 4.9: Correlograma de los residuos del SARIMA$(1,1,3)(2,1,0)_{24}$

Observamos auto-correlacciones de bajo valor pero aun hay varias que sobrepasan las bandas de confianza, especialmente las de retardos múltiplos de la estacionalidad diaria. Los test de Ljung-Box a partir de retardo 12 rechazan la hipótesis nula (p-valor de 0.004 para el retardo 12 y 0. para el 24, hasta ahí sobrepasan todos el nivel de 0.05).

	Estimación	Std	z value	p-value
ar1	0.571388	0.024058	23.7503	<2.2e-16
ma1	-0.489329	0.025679	-19.0559	<2.2e-16
ma2	-0.287817	0.011534	-24.9530	<2.2e-16
ma3	-0.094707	0.016006	-5.9169	3.28e-09
sar1	-0.490582	0.010340	-47.4433	<2.2e-16
sar2	-0.287873	0.010236	-28.1235	<2.2e-16

Cuadro 4.1: Coficientes del modelo SARIMA

Por otro lado, los parámetros del modelo son todos significativos pero si echamos un vistazo a las correlaciones vemos que hay varios coeficientes que están exageradamente correlacionados.

	ar1	ma1	ma2	ma3	sar1	sar2
ar1	1.000000000	-0.907910025	0.08668054	0.74027949	0.005679868	-0.024379200
ma1	-0.907910025	1.000000000	-0.28120153	-0.76708249	-0.062780569	0.008462644
ma2	0.086680542	-0.281201529	1.00000000	-0.25190532	0.066931989	-0.015573005
ma3	0.740279489	-0.767082488	-0.25190532	1.00000000	0.026230243	0.014984791
sar1	0.005679868	-0.062780569	0.06693199	0.02623024	1.000000000	0.380529048
sar2	-0.024379200	0.008462644	-0.01557300	0.01498479	0.380529048	1.000000000

Cuadro 4.2: Matriz de correlaciones de los parámetros estimados

Finalmente la suma de residuos al cuadrado nos da 7191.

Acabamos aquí el análisis, no es el mejor modelo SARIMA posible (como hemos visto tiene varios problemas) y no hemos, ni siquiera, intentado incorporar la doble estacionalidad de la serie. Esto no es necesario pues aquí solo buscamos un modelo base para pasar al interés principal, que son los modelos de espacio de estados; al fin y al cabo parte del objetivo de este proyecto es esquivar la excesiva complicación de encontrar un buen candidato ARIMA.

Modelo de estacionalidad diaria

Tras trabajar brevemente con los ARIMA pasamos a los SSM, el modelo más simple (relativamente) que podemos crear es aquel que posee tan solo una estacionalidad, la diaria; es decir, con tan solo 24 parámetros estacionales (23 en realidad con una restricción de que la suma valga 0).

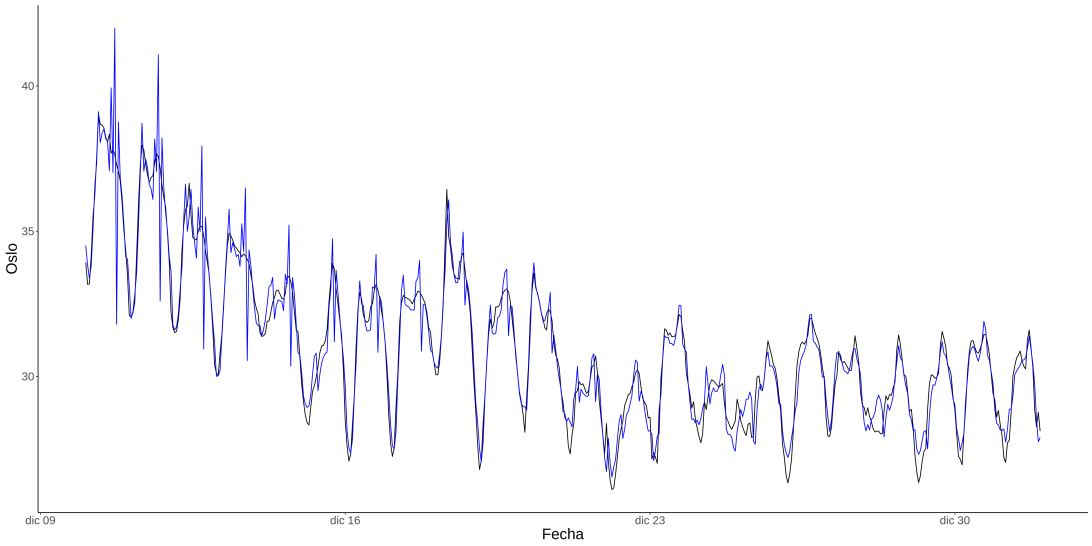


Figura 4.10: Serie (negro) y predicción a un paso del modelo simple para los últimos 20 días

El modelo es el siguiente (como usamos KFAS usamos la formulación MSOE) con $m = 24$:

$$\begin{aligned} y_t &= l_{t-1} + b_{t-1} + s_{t-1} + \epsilon_t \\ l_t &= l_{t-1} + b_{t-1} + \alpha_t^1 \\ b_t &= b_{t-1} + \alpha_t^2 \\ s_t &= s_{t-m} + \alpha_t^3 \end{aligned}$$

Tenemos en total 4 varianzas por estimar, lo llevamos a cabo y obtenemos los siguientes valores: $Var(\epsilon) = 0{,}000001$, $Var(\alpha) = (3{,}6 \quad 0 \quad 0{,}012)$. Parece que la tendencia de este modelo no cambia a lo largo del tiempo, lo estudiaremos más en detalle posteriormente.

Ahora que tenemos el modelo completo, aplicamos el filtro de Kalman y calculamos para cada instante t del tiempo los valores de los estados y las predicciones a un paso. Esto nos permite además obtener los residuos para analizarlos; si los analizamos con el correlograma vemos lo siguiente:

De manera inmediata observamos que el gráfico dista mucho de corresponder a uno de ruido blanco o ni siguiera a una estacionaria, por lo que continuamos con la búsqueda de modelos.

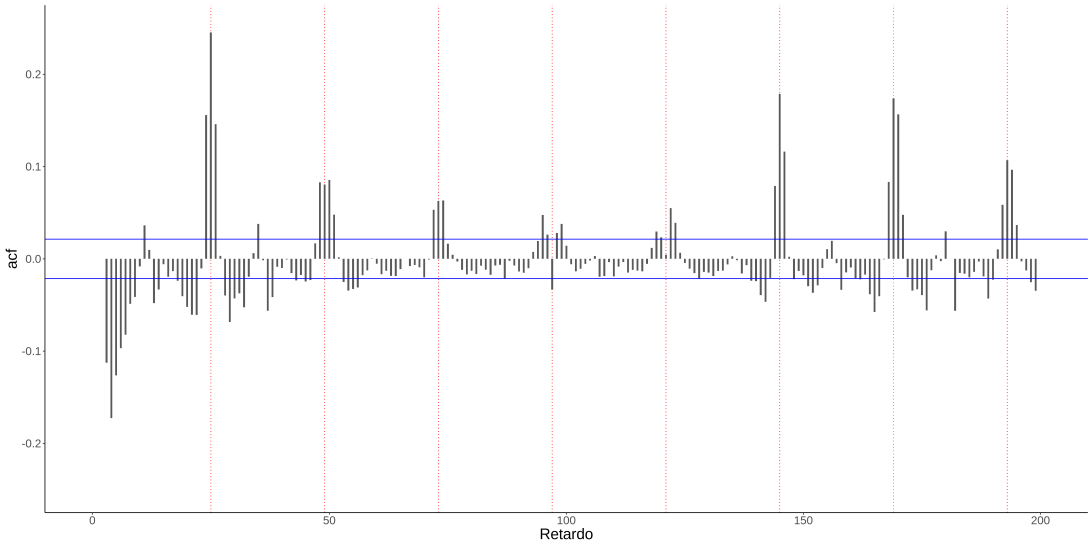


Figura 4.11: Correlograma de los residuos del modelo de estacionalidad diaria (24)

Modelo de doble estacionalidad

Añadimos la estacionalidad semanal al modelo previo, en un principio vamos a la manera más simple (e ineficiente) de implementarla que es trabajando con la totalidad de los posibles parámetros, en total los 168 (167 por las mismas razones que antes); esto hace que el modelo tome cada hora de cada día de la semana como un nivel independiente que ha de estimar, por esto lo llamamos ineficiente pues no usa la relación aparente entre las distintas horas y días de la semana.

El modelo es el mismo que antes pero con la $m = 168$; estimamos varianzas y obtenemos los valores $Var(\epsilon) = 0{,}000001$, $Var(\alpha) = (3{,}6 \quad 0 \quad 0{,}012)$, sorprendentemente los valores son prácticamente idénticos (los valores aquí mostrados están redondeados, los coeficientes difieren en las centésimas). Repetimos la estimación anterior y analizamos el correlograma:

De nuevo el gráfico dista de lo esperado, no hay estacionaridad, hay demasiadas auto-correlacciones que superan el umbral y aun hay una aparente estacionalidad, aunque no parece que exista tendencia (tras las dos primeras auto-correlacciones parece haber un decrecimiento exponencial). Además, si comparamos la suma del error cuadrático de los residuos[1] de este modelo con los del anterior, descubrimos que ha empeorado (el modelo simple tiene un RSSE de 6464 y este último de 8167); aunque este último sea un modelo más complejo, es demasiado ineficiente, tiene demasiados parámetros para tan pocos datos (tan solo un año) y no utiliza gran parte de la información lo que al final acaba perjudicando la predicción. Esto es lo que vamos a intentar arreglar a continuación.

[1]Debido a que los modelos tardan en obtener una buena estimación de los estados, para el calculo eliminamos los primeros datos y solo usamos las predicciones/residuos de la segunda mitad de la serie (las primeras 4032 observaciones)

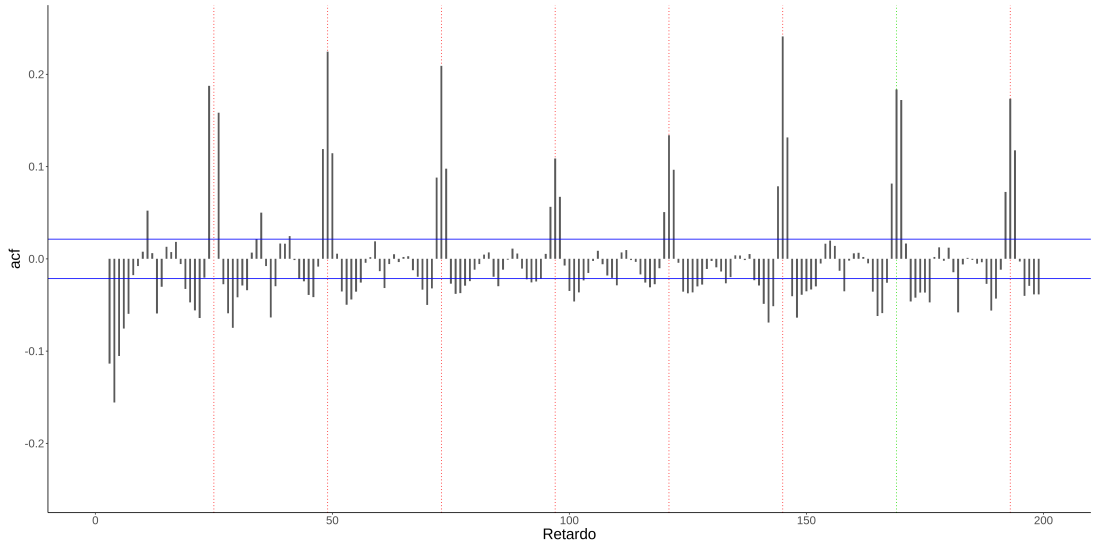


Figura 4.12: Correlograma de los residuos del modelo de estacionalidad diaria+semanal (168)

Modelo de estacionalidades anidadas

El problema del anterior modelo es, primero que tiene demasiados parámetros (las estimaciones de los parámetros tardan mucho en converger) y, además, que estamos tratando cada hora de cada día de la semana de manera independiente sin tener en cuenta que el nivel a una hora específica del día esta correlaccionado con el nivel de otro. Para arreglarlo vamos a crear un nuevo modelo que tenga en cuenta estos problemas; que reduzca el número de patrones horarios (como vimos en la figura 4.5, con solo dos o tres representaríamos casi toda la variación) y que permita actualizar los parámetros más de una vez por ciclo.

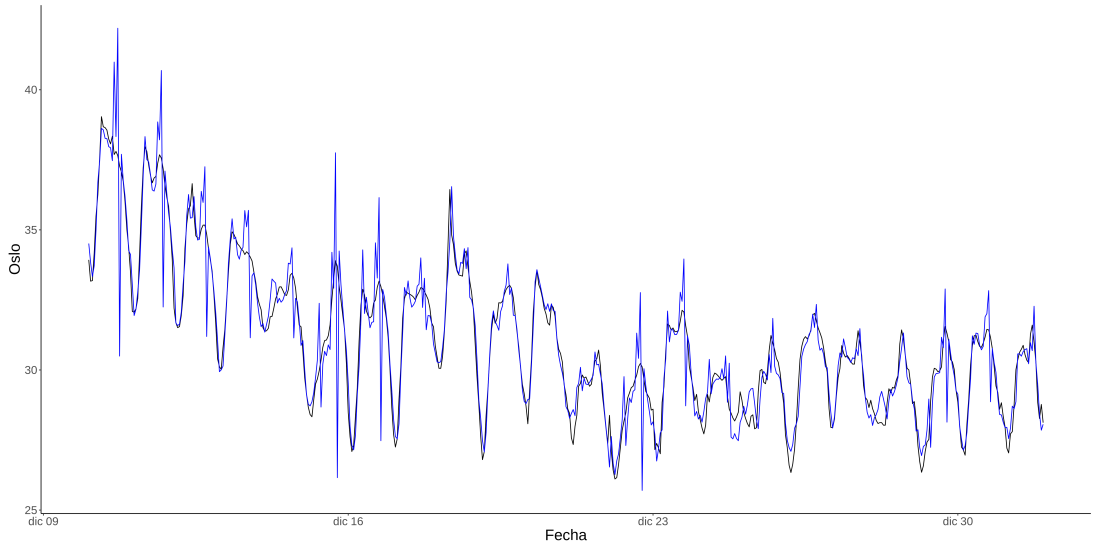


Figura 4.14: Serie (negro) y predicción a un paso del modelo anidado para los últimos 20 días

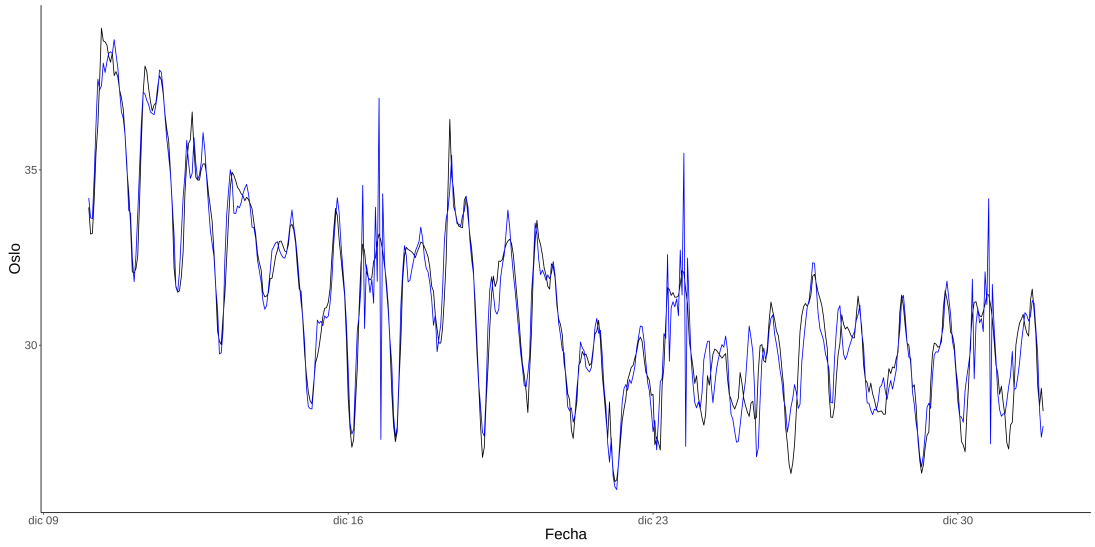


Figura 4.13: Serie (negro) y predicción a un paso del modelo de doble estacionalidad para los últimos 20 días

Para llevar esto a cabo anidamos los periodos teniendo en cuenta los distintos patrones horarios de cada día de la semana (la interacción entre ambas estacionalidades) pero permitiendo reducir estos patrones tanto como queramos (a tan solo dos, tres...). Además añadimos unos nuevos parámetros de suavizado[2] que permiten actualizar las estimaciones de un periodo durante otro distinto.

El modelo que resulta es el siguiente (basado en el de [Hyn+08], con SSOE):

$$
\begin{aligned}
y_t &= l_{t-1} + b_{t-1} + \mathbf{x}^T \mathbf{s}_{t-m_1} + \epsilon_t \\
l_t &= l_{t-1} + b_{t-1} + \alpha \epsilon_t \\
b_t &= b_{t-1} + \beta \epsilon_t \\
\mathbf{s}_t &= \mathbf{s}_{t-m_1} + \mathbf{\Gamma} \mathbf{x}_t \epsilon_t \\
\hat{y}_{t+1|t} &= l_{t-1} + b_{t-1} + \mathbf{x}_t^T \mathbf{s}_{t-m_1}
\end{aligned}
$$

Con $m_1 = 24 \quad m_2 = 24 * 7 = 168$, l nivel, b tendencia, s_i estacionalidades, Γ parámetros de suavizado con γ_{ii} actualizando durante el mismo subciclo y el resto actualizando durante distintos.

Para implementarlo en R necesitamos transformarlo en la forma de primer orden con

[2]Esta es una matriz que va a ser necesaria estimar, por lo que estableceremos restricciones para limitar su complejidad

solo un retardo:

$$y_t = \mathbf{w}_t^T x_{t-1} + \epsilon_t$$
$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}_t \epsilon_t$$
$$\mathbf{a}_t = (l_t, b_t, s_{1t}, ..., s_{1t-m_1+1}, s_{2t}, ..., s_{rt-m_1+1})^T$$
$$\mathbf{w}_t = (1, 1, 0, ..., 0, x_{1t}, 0, ..., x_{rt})$$
$$\mathbf{g}_t = \begin{bmatrix} \alpha \\ \beta \\ 0 \\ \vdots \\ \sum \gamma_{1i} x_{it}{}_{i=1}^{r} \\ 0 \\ \vdots \\ \sum \gamma_{ri} x_{it}{}_{i=1}^{r} \\ \vdots \\ 0 \end{bmatrix}$$

Con las siguientes matrices y $r = 2$ (un patrón horario para entre-semana y otro para el fin de semana):

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_l & \dots & 0 \\ \dots & \vdots & \dots \\ 0 & \dots & \mathbf{F}_s \end{bmatrix}$$
$$\mathbf{F}_l = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$
$$\mathbf{F_s} = \mathbf{I} \otimes \mathbf{F_1}$$
$$\mathbf{F_1} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1. & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

Adaptamos el modelo para usarlo con la formulación MSOE (la usada en la librería KFAS) y estimamos los parámetros necesarios, en particular, las varianzas y la matriz de suavizado; para esta última establecemos la restricción de que todos los valores de la diagonal son iguales, y lo mismo con los de fuera de la diagonal. Así restringimos la matriz a tan solo dos valores a estimar.

Así pues obtenemos, como anteriormente, residuos y predicciones.

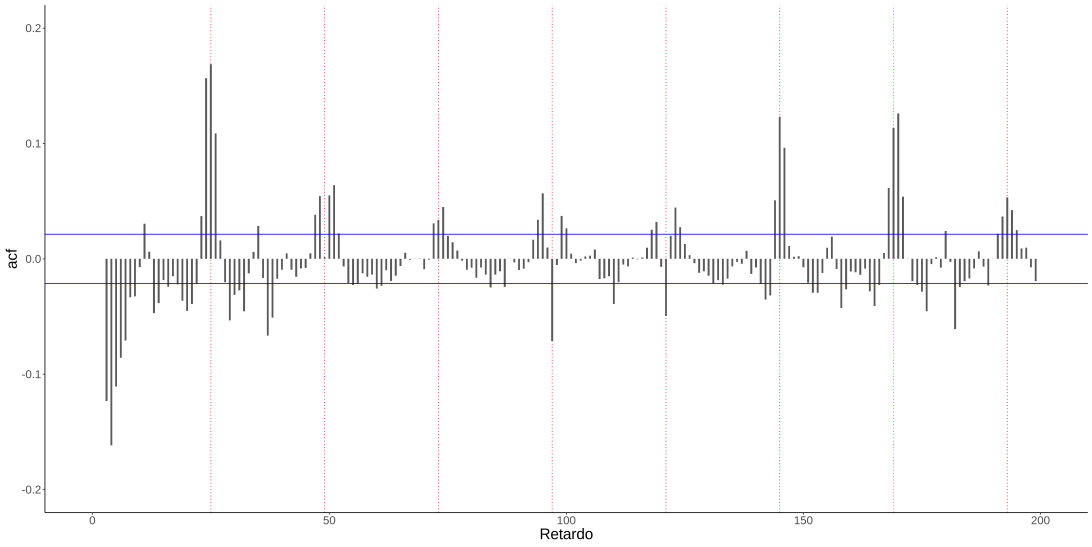


Figura 4.15: Correlograma de los residuos del modelo de estacionalidades anidadas

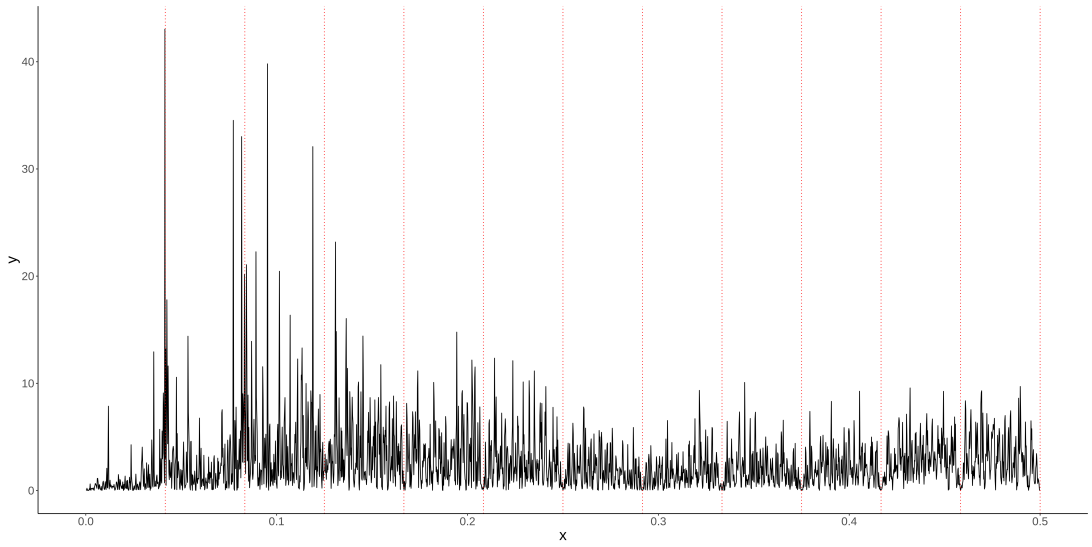


Figura 4.16: Periodograma de los residuos del modelo de estacionalidades anidadas

De nuevo observamos un patrón de auto-correlaciones con estacionalidad pero mucho más reducido; estudiando este gráfico junto con el periodograma vemos que aun queda cierta estacionalidad por explicar y si aplicamos los tests de Ljung-Box para distintos retardos vemos que son todos y cada uno rechazados (con p-valor menor de 0.01). Aun así, si tenemos en cuenta el RSSE, vemos que hemos mejorado respecto al modelo de una sola estacionalidad con un error de 6158.

Modelo estacional+GARCH

Estudiando el gráfico de residuos, observamos clusters de error que coinciden con una mayor variabilidad en la serie original, seguidos de largas rachas de baja variabilidad. Este es un patrón de heterocedasticidad agregada que ya observamos en la primera parte del análisis de los datos. Para paliar este problema – pues hasta ahora hemos asumido una varianza única para cada componente – complementamos nuestro modelo anterior con otro de tipo GARCH[3].

Estos modelos formulan la varianza como un proceso ARMA, lo cual les permite generar una estimación de esta a cada paso del modelo. Para nuestro SSM, mantenemos el modelo de estacionalidades anidadas y usamos un GARCH(1,1) para estimar la varianza. Además, para paliar el problema de las elevadas autocorrelaciones en los primeros retardos, añadimos un ARMA(1,3) basándonos en el previo modelo ARIMA.

Para observar el efecto que tienen ambas soluciones, las incluimos de manera aislada una por una; primero el GARCH:

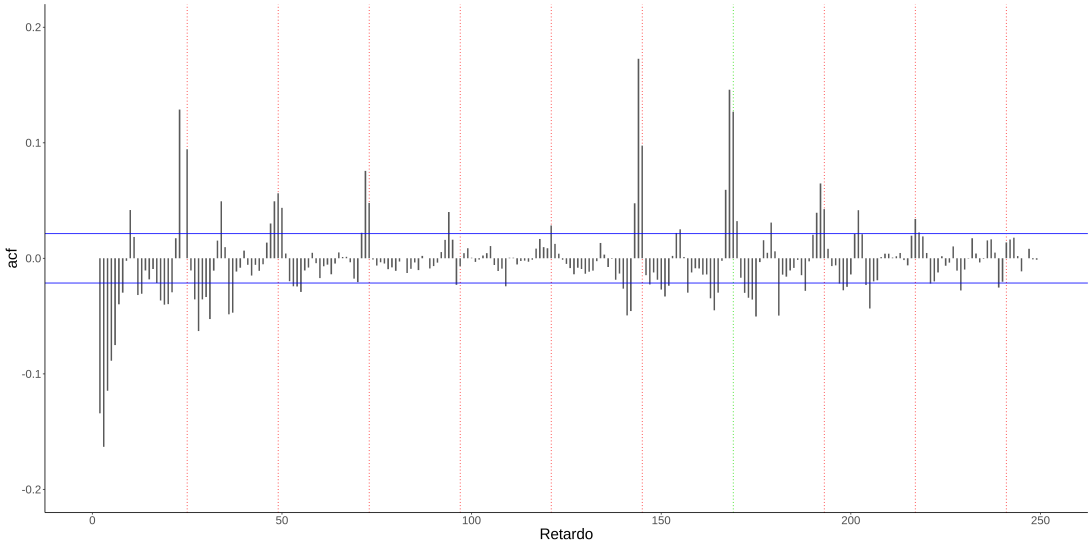


Figura 4.17: Correlograma del modelo de estacionalidades anidadas+GARCH

Observamos una mejora en la capacidad de predicción, con un RSSE de 5779, que nos confirma nuestras sospechas del problema con la hetecedasticidad; aun así el patrón de autocorrelaciones sigue prácticamente igual, lo que nos dice que aún hay problemas sin revisar; continuamos incluyendo el ARMA:

[3]Generalized Autoregressive Conditional Heteroskedasticity o Heterocedasticidad Condicional Autoregresiva Generalizada

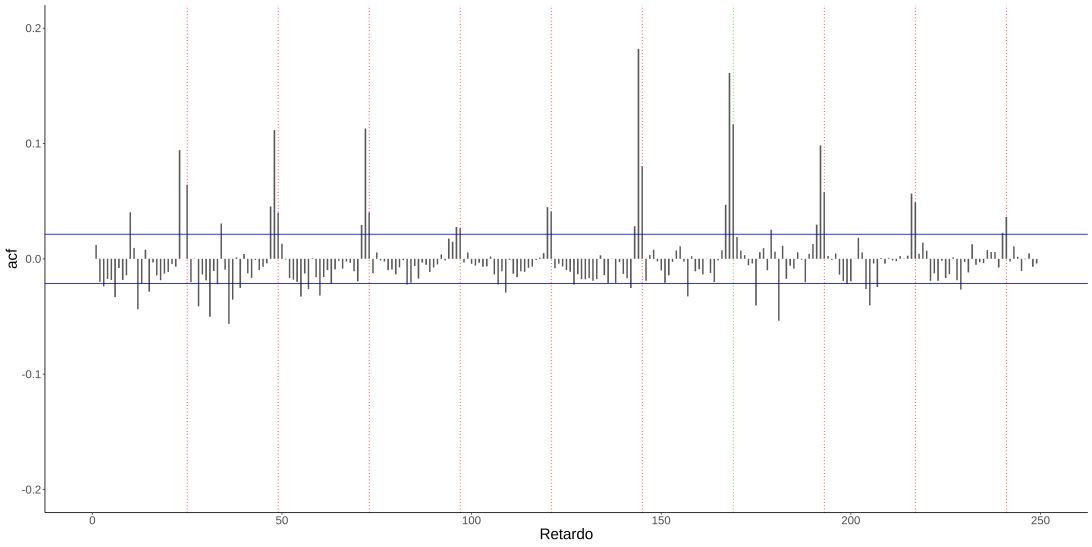


Figura 4.18: Correlograma del modelo de estacionalidades anidadas+GARCH+ARMA

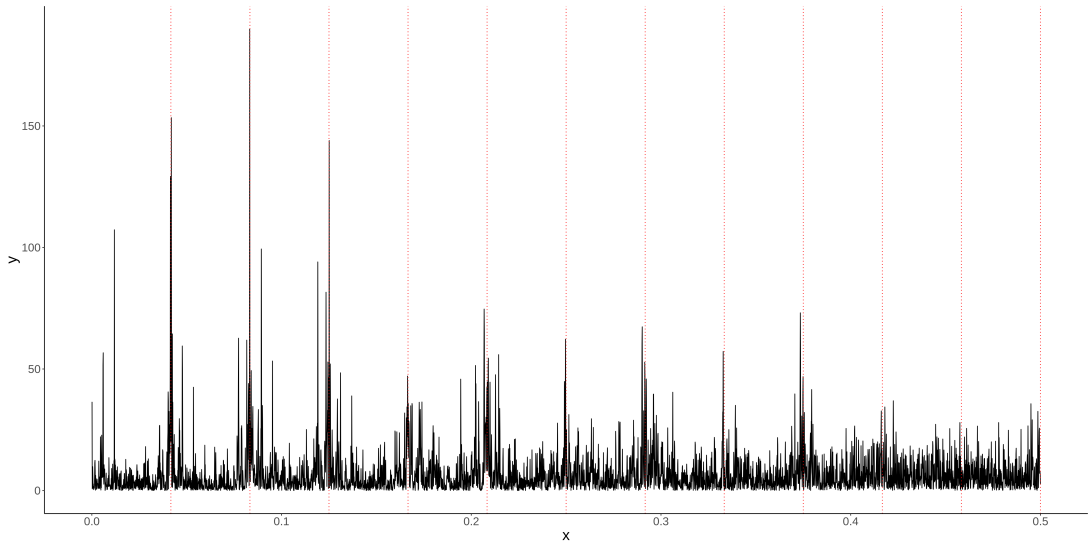


Figura 4.19: Periodograma del modelo de estacionalidades anidadas+GARCH+ARMA

Esta vez el RSSE empeora ligeramente a 5883 pero conseguimos eliminar casi por completo la autocorrelación en los primeros retardos, objetivo principal de la inclusión del ARMA. El periodograma no parece cambiar excesivamente respecto a los anteriores modelos.

Ahora sí, pasamos a analizar y comparar los mejores modelos.

4.2.2. Adecuación de los Modelos

A continuación vamos a analizar en profundidad los modelos planteados hasta el momento; dado que ya hemos mostrado la ACF y el periodograma, pasamos directamente a los tests de Ljung-Box:

Modelo	Retardo 1	2	3	6	12	24	48
Simple	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Doble	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ARIMA	0.99	0.99	0.99	0.70	0.004	0.0	0.0
Anidado	0.26	0.09	0.02	0.0	0.0	0.0	0.0
GARCH	0.0	0.0	0.0	0.0	0.0	0.0	0
GARCH+ARMA	0.26	0.09	0.02	0.0003	0.0	0.0	0.0

Cuadro 4.3: P-valores del test Ljung-Box

Ningún modelo pasa el test para más de 12 retardos, y los dos primeros modelos junto con el anidado+GARCH ni siquiera los pasan para el primer retardo, confirmando de nuevo el problema observado en el correlograma.

Residuos

Pasamos por encima el qqplot ya que todos los modelos, incluso el ARIMA, se desvían significativamente de la normalidad; a continuación mostramos los gráficos de los residuos para el modelo anidado+GARCH y el modelo anidado+GARCH+ARMA:

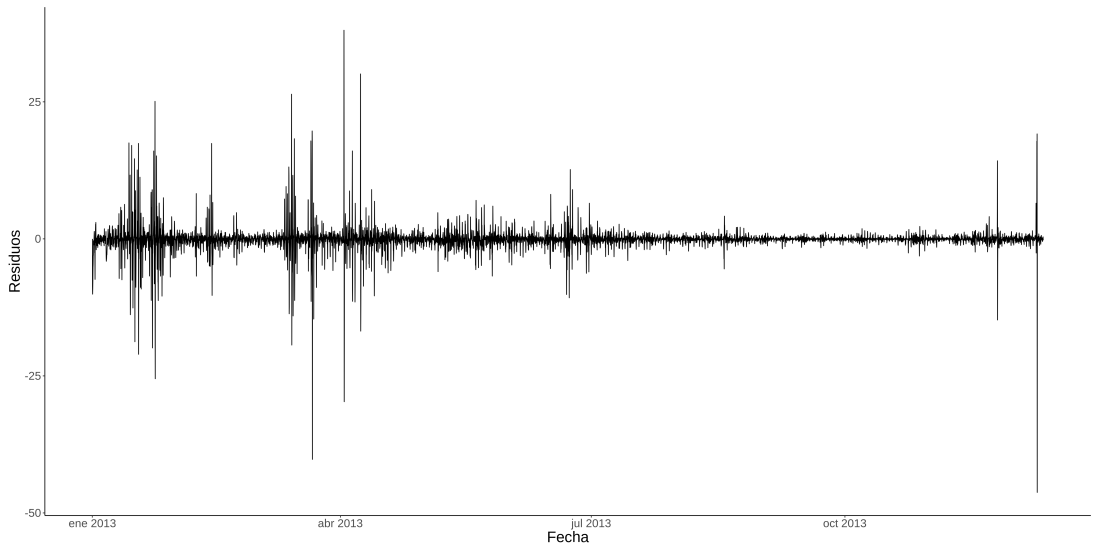


Figura 4.20: Residuos del modelo de estacionalidades anidadas+GARCH

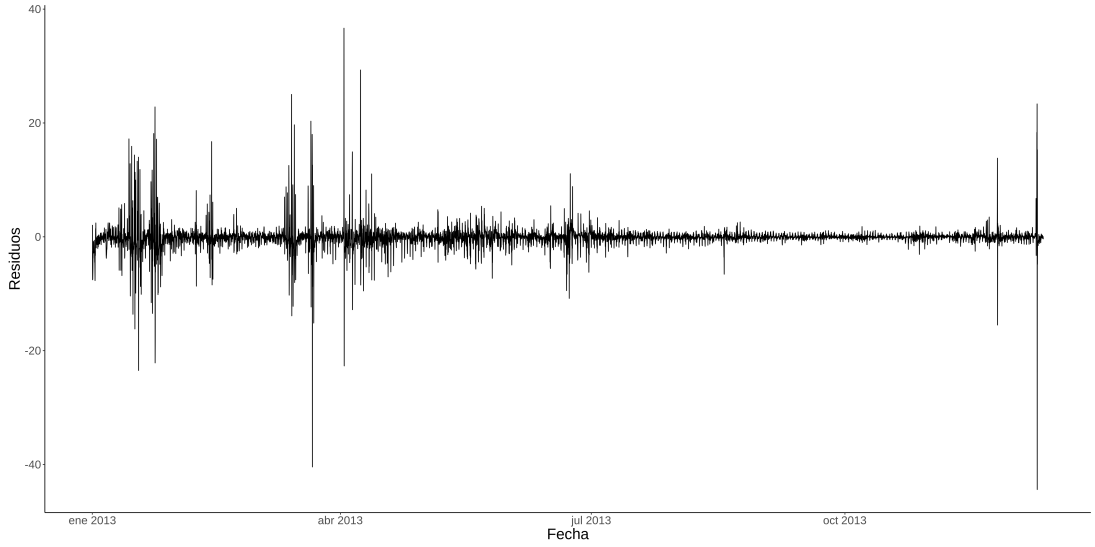


Figura 4.21: Residuos del modelo de estacionalidades anidadas+GARCH+ARMA

Advertimos el mismo patrón encontrado en la serie original; a pesar del GARCH, los modelos pierden capacidad predictiva durante las rachas de alta volatilidad; sin embardo, durante las rachas más estables ambos modelos poseen una tasa de error sustancialmente más baja.

4.2.3. Comparativa

Aquí mostramos una tabla con los índices usados para comparar los modelos; usamos la suma de residuos al cuadrado (total, sin las primeras 4032 observaciones y sobre los últimos 20 días reservados), el AIC[4] y el número total de parámetros:

Modelo	RSSE(tot.)	RSSE(sin primrs.)	RSSE(resrv.)	AIC	Parámetros
ARIMA	31838	7191	334	36112	7
Simple	38334	6464	202	37099	28
Doble	44895	8167	332	37756	172
Anidado	31154	5883	352	39184	53
GARCH	34337	5815	96	28295	54
GARCH+ARMA	31154	5883	89	38767	59

Cuadro 4.4: Tabla comparativa de modelos

Advertimos como en el AIC el mejor es el modelo con estacionalidades anidadas+GARCH, que además tiene el mejor error de todos los modelos (tenemos menos en cuenta el RSSE

[4]Para los modelos con GARCH la verosimilitud esta calculada sin esta componente, al ser la librería KFAS para SSM lineales esta se ha calculado separadamente

total pues los SSM, al inicializar de manera difusa, tienen un coeficiente mayor de error durante las primeras observaciones; respecto a las observaciones reservadas, aunque útiles, debido a la baja volatilidad de los últimos datos de la serie no son del todo representativas de la serie completa).

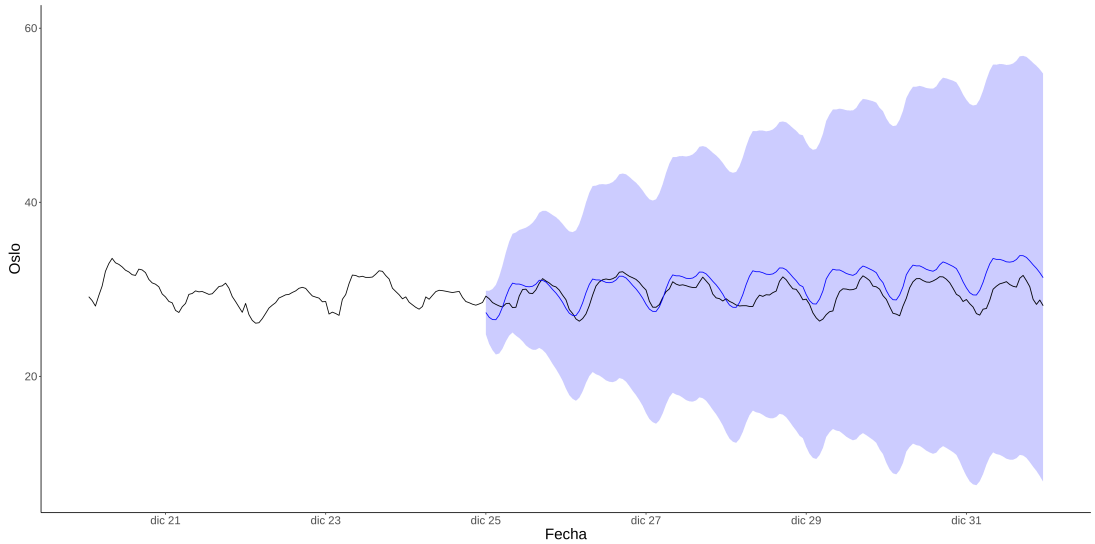


Figura 4.22: Bandas de predicción para la última semana (168 observaciones) - Modelo Anidado+GARCH

Cabe mencionar que el último modelo, en el cual introducimos un ARMA para mejorar la estructura de autocorrelación, tiene el peor AIC de todos despues del anidado; esto proviene de que al optimizar los parámetros de éste tomamos como función de coste la ACF y no la verosimilitud[5], como en el resto de modelos. Esto es una ventaja de los SSM, que permiten optimizar fácilmente sus parámetros sobre otras funciones, pero a su vez introduce complicaciones como la que vemos ahora.

Finalmente, si tuviéramos que seleccionar un solo modelo elegiríamos el modelo con estacionalidades anidadas+GARCH, pues tiene, como hemos comentado previamente, la mayor capacidad predictiva aunque el modelo posea ciertos problemas con los residuales.

4.3. Conclusiones

Hemos visto como los modelos de Espacio de Estados tienen una flexibilidad singular, que permite añadir y quitar componentes de manera casi independiente. También hemos observado como esta mayor flexibilidad nos permite generar modelos más complejos a partir de otros más simples, incrementando así la capacidad predictiva y el ajuste de la serie.

[5]Durante la creación de modelos siempre se estimaron los parámetros a través de la verosimilitud, pero en ningún caso el ARMA mejoro ni la RSSE ni la acf. Posteriormente se paso a usar la acf como función de coste para mostrar las posibilidades de los SSM, con los resultados aquí mostrados.

Además, gracias a la interpretabilidad de los parámetros de los modelos, durante todo el desarrollo hemos podido usarlos para analizar diversos problemas y refinar, de manera reiterada, los modelos.

Es cierto que en este caso hemos encontrado diversos problemas pero es necesario tener en cuenta la dificultad de modelar una serie financiera tan compleja, con múltiples estacionalidades y heterocedasticidad.

Concluimos pues, que esta rama del análisis es increíblemente útil y que es una gran herramienta para cualquier estadístico.

Parte IV

Apéndices

Apéndice A

Anexos

A.1. Código R

A continuación incluimos el código R para generar el modelo de estacionalidades anidadas+GARCH, se incluyen al principio las librerías usadas.

```
# Principales librerias usadas
library(tidyverse)
library(purrr)
library(broom)
library(KFAS)
library(rugarch)

# Parametros del modelo
sp <- 24
sg <- 2
variances <- c(1.601697,0, 0.02848101)
a<-0.4

# Creacion manual de las matrices del modelo
n <- 2+sp*sg
Fm <- matrix(0,n,n)
Fl <- matrix(c(1,1,0,1),2,2,byrow = TRUE)
F1 <- diag(sp)
aux <- F1[sp,]
for(i in seq(sp,2,-1)) F1[i,]<-F1[i-1,]
F1[1, ] <- aux
Fs <- diag(sg) %x% F1
Fm[1:2,1:2] <- Fl
Fm[3:n,3:n] <- Fs

get_ws <- function(t){
  index <- (t-1)%%168
  if(index<sp*5) w_s<-c(1,0)
  else w_s <- c(0,1)
  return(w_s)
}
```

```
w <- sapply(1:nrow(data_energy), function(t) c(1,1,
                unlist(lapply(get_ws(t),
                        function(x) return(c(rep(0, sp-1), x)) ))))
dim(w) <- c(1, dim(w))

k<-length(variances)
Q <- diag(variances)

get_rs <- function(t){
  index <- (t-1)%%168
  if(index<sp*5) w_s<-c(1,a)
  else w_s <- c(a,1)
  return(w_s)
}
R <- matrix(0,nrow(Fm), k)
R[1:k,1:k]<-diag(k)
R<-vapply(1:nrow(data_energy), function(t){
  rs <- get_rs(t)
  r <- R
  r[k,k]<- rs[1]
  r[k+sp-1,k]<-rs[2]
  return(r)
  }, R)
P1inf<-diag(nrow(Fm))

# Preparacion por adelantado del modelo GARCH (variante iGARCH)
gmodel<-ugarchspec(variance.model = list(model="iGARCH", garchOrder=c(1,1)),
                   mean.model = list(armaOrder=c(1,3), include.mean=TRUE, arfima=TRUE))
gfit<-ugarchfit(gmodel, data_energy$Oslo)

Qm <- vapply(1:nrow(data_energy), function(t){
  return(Q*gfit@fit$sigma[t])
}, Q)

# Creacion del modelo
custom_garch <- SSModel(Oslo ~
                          SSMcustom(w, Fm, R=R, Q=Qm, P1inf=P1inf), data=data_energy, H
                              =0.001)
```

Bibliografía

[Gil95] Gilbert, P.D. «Combining VAR Estimation and State Space Model Reduction for Simple Good Predictions». En: *J. of Forecasting: Special Issue on VAR Modelling* 14 (1995), págs. 229-250.

[BD02] Peter Brockwell y Richard Davis. *An Introduction to Time Series and Forecasting*. Vol. 39. Ene. de 2002. DOI: 10.1007/978-1-4757-2526-1.

[JU04] S. J. Julier y J. K. Uhlmann. «Unscented filtering and nonlinear estimation». En: *Proceedings of the IEEE* 92.3 (2004), págs. 401-422.

[Hyn+08] Rob Hyndman y col. «Forecasting with exponential smoothing. The state space approach». En: ene. de 2008. DOI: 10.1007/978-3-540-71918-2.

[GA10] M. S. Grewal y A. P. Andrews. «Applications of Kalman Filtering in Aerospace 1960 to the Present [Historical Perspectives]». En: *IEEE Control Systems Magazine* 30.3 (2010), págs. 69-78.

[Pet10] Giovanni Petris. «An R Package for Dynamic Linear Models». En: *Journal of Statistical Software* 36.12 (2010), págs. 1-16. URL: http://www.jstatsoft.org/v36/i12/.

[Sko11] Boris Skorohod. «Diffuse Initialization of Kalman Filter». En: *Journal of Automation and Information Sciences* 43 (ene. de 2011), págs. 20-34. DOI: 10.1615/JAutomatInfScien.v43.i4.30.

[Tus11] Fernando Tusell. «Kalman Filtering in R». En: *Journal of Statistical Software, Articles* 39.2 (2011), págs. 1-27. ISSN: 1548-7660. DOI: 10.18637/jss.v039.i02. URL: https://www.jstatsoft.org/v039/i02.

[NW13] Shu-yuan Nie y Xin-qian Wu. «A Historical Study About the Developing Process of the Classical Linear Time Series Models». En: *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013*. Ed. por Zhixiang Yin, Linqiang Pan y Xianwen Fang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, págs. 425-433. ISBN: 978-3-642-37502-6.

[Hel17] Jouni Helske. «KFAS: Exponential Family State Space Models in R». En: *Journal of Statistical Software* 78.10 (2017), págs. 1-39. DOI: 10.18637/jss.v078.i10.

[Faw+18] Hassan Ismail Fawaz y col. «Deep learning for time series classification: a review». En: *CoRR* abs/1809.04356 (2018). arXiv: 1809.04356. URL: http://arxiv.org/abs/1809.04356.

[Pen19] Roger D. Peng. *R Programming for Data Science*. 2019. URL: https://bookdown.org/rdpeng/rprogdatascience/.

[WHO20] Hadley Wickham, Jim Hester y Jeroen Ooms. *xml2: Parse XML*. R package version 1.2.5. 2020. URL: https://CRAN.R-project.org/package=xml2.

[Kal] Kalman Filter. *Kalmna Filter — Wikipedia, The Free Encyclopedia*. [Online; acceso 04-Mayo-2020]. URL: https://en.wikipedia.org/wiki/Kalman_filter.

[Nor] Nord Pool. *Nord Pool, leading power market in Europe*. [Online; acceso 05-Mayo-2020]. URL: https://www.nordpoolgroup.com/.

[Ser] Serie temporal. *Serie temporal — Wikipedia, The Free Encyclopedia*. [Online; acceso 04-Mayo-2020]. URL: https://es.wikipedia.org/wiki/Serie_temporal.