Gestión integrada de un banco competitivo de modelos predictivos estacionarios y volátiles

by

Miguel Ángel García Díaz

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

Noviembre 2017

Gestión integrada de un banco competitivo de modelos predictivos estacionarios y volátiles

Miguel Ángel García Díaz

Máster en Economía, Finanzas y Computación

Antonio Aníbal Golpe Moya y Pedro Cadahia
Universidad de Huelva y Universidad Internacional de Andalucía

2017

Abstract

The information and data are the important keys in any business ‘s strategy.

Nowadays, in many companies, the DataBase is a relevant factor because of his great computer supports.

At present, any organization, even the smallest, has to know how to manage its Database in order to get an effective objective.

Our objective is to show you how to deal with a real DataBase, step by step, given no importance to the amount of observations. Therefore we will focus on doing a deep pre-processing’s.

Firstly, we will split up the original dataset of different companies into a single dataset per company.

Secondly, we will form a control system which contains predictions of where a massive dataset will be processed to distinguish between two types of time series known, by the end, as seasonality and volatile series.

Finally, we will estimate the MAE and MSE errors.

Key words: dataset, MAE, MSE, KDD, seasonality, volatile, missing values.

Resumen

La información y los datos son un factor clave en la estrategia de cualquier empresa. Hoy en día, las Bases de Datos tienen una gran relevancia a nivel empresarial, ya se consideran una de las mayores aportaciones que ha dado la informática a las empresas. En la actualidad, cualquier organización que se precie, por pequeña que sea, debe saber cómo gestionarlas para que sea efectiva.

En este trabajo, vamos a demostrar como se debe tratar una base de datos real paso a paso sin importar en número de observaciones que presente. Por tanto, nos centraremos en realizar un buen preprocesamiento. Dividiremos el dataset original de las diferentes empresas, en un dataset por empresa. Seguidamente, construiremos un sistema de control integrado de predicciones en el que un dataset masivo de datos sea tratado para discernir entre dos tipos de series temporales definidas a posteriori como predictivas o volátiles. Finalmente, calcularemos los errores MAE y MSE.

Agradecimientos

En primer lugar agradecer tanto a Pedro Cadahia, Aníbal y a todo el equipo de Minsait el apoyo prestado durante estos meses de duro trabajo. Gracias a ellos, he podido conocer el Big Data desde el punto de vista laboral.

Agradecer a todos los profesores del Máster el trato cercano y la illusión que nos transmitieron desde el primer día y poder darnos la oportunidad de aprender acerca de este maravilloso mundo de los datos.

Dar las gracias una vez más a mi familia, por tener siempre su apoyo constante.

Y por ultimo, a tí abuelo, que aunque no estés con nosotros todo esto va dedicado a tí.

Tabla de Contenidos

Lista de Tablas

Lista de Figuras

1 Introducción

Hoy en día, los datos se han convertido en el activo tangible más valioso y con más potencial de las empresas, ya que tan sólo en el último año se generaron más datos que en 50 siglos de historia y la tendencia es de crecimiento exponencial de los mismos.

La información se ha convertido en un elemento clave en los procesos organizacionales. En los últimos años, la tecnología ha tenido un crecimiento acelerado como herramienta útil y necesaria para facilitar dichos procesos y mejorar la productividad.

En un mercado cada vez más competitivo, el conocimiento cobra mayor relevancia. En este sentido, las empresas buscan la posibilidad de obtener ventajas empresariales, mediante su aplicación en los procesos de toma de decisiones. Entre las técnicas que se utilizan, la minería de datos ha ocupado un lugar primordial en el mundo empresarial, ya que permite encontrar modelos ocultos dentro de las grandes cantidades de datos y generar conocimiento.

El alcance de este proyecto no es conseguir el mejor modelo, ya que consideraremos dos tipos de modelos autorregresivos como son el arima y la regresión lasso, sino construir un sistema de control integrado de predicciones en el que un dataset masivo de datos sea tratado para discernir entre dos tipos de series temporales definidas a posteriori como predictivas o volátiles. De este modo, se construye un datapipe (flujo de datos) en el que se realizará una serie de predicciones con el objetivo de facilitar un primer enfoque fácil y rápido.

Para la realización de este trabajo, he contado con el apoyo de Indra. Indra es una de las principales empresas globales de consultoría y tecnología y el socio tecnológico de los negocios clave de sus clientes en todo el mundo. Desarrolla una oferta de soluciones propias y servicios avanzados y de alto valor añadido en tecnología, que permiten a sus clientes resolver sus asuntos más críticos y mejorar sus procesos, eficiencia, rentabilidad y diferenciación.

Indra es líder mundial en el desarrollo de soluciones tecnológicas integrales en campos como Defensa y Seguridad; Transporte y Tráfico; Energía e Industria; Telecomunicaciones y Media; Servicios financieros; y Administraciones públicas y Sanidad. Y a través de su unidad Minsait, da respuesta a los retos que plantea la transformación digital.

En este trabajo se pretende abordar la importancia que tiene realizar un buen preprocesamiento de una base datos y perder la menor información posible. Por esta razón, cabe destacar, que anteponemos la calidad de los datos ante la cantidad.

2 Base de datos

La base de datos con la que vamos a realizar el proyecto es propiedad de Indra. Dicha base de datos contiene 1.764.824 observaciones y 42 variables, empezando en Enero 2005 y acabando en Febrero 2017. Cabe destacar, que no todos los datos de las diferentes empresas comienzan en Enero del 2005 y acaban en Febrero del 2017, pero este hecho lo abordaremos mas adelante. De las 42 variables solo conocemos la naturaleza de las siguientes:

- DATE: Fecha determinada

- RP_ENTITY_ID: Un ide de activo

- T0_RETURN: Return on the given date.

- T1_RETURN: Next day Return.

- Un set de 36 Indicatores diferentes (Features)

El motivo por el cual desconocemos las demás variables es por su política de privacidad de Indra, ya que no nos pueden proporcionar datos de sus clientes.

3 Preprocesamiento de la base de datos

El preprocesamiento de datos es una etapa esencial del proceso de descubrimiento de información o KDD (Knowledge Discovery in Databases, en inglés). Esta etapa se encarga de la limpieza de datos, su integración, transformación y reducción para la siguiente fase de minería de datos.

Debido a que normalmente el uso de datos de baja calidad implica un proceso de minería de datos con pobres resultados, se hace necesaria la aplicación de técnicas de preprocesamiento.

En el desarrollo teórico de la mayoría de técnicas y modelos que se aplican a una base de datos, es primordial darle importancia a la existencia de datos faltantes, también denominados perdidos o incompletos. Muchas bases de datos contienen valores perdidos o no presentes en las observaciones que la componen (ya sean provocados por mediciones incorrectas, errores, ect.). Estos valores perdidos crean numerosos problemas y hacen difícil el análisis de los datos, por lo tanto hace que sea necesaria una etapa de preprocesado de la base de datos.

Después de la aplicación de la fase de preprocesamiento, el conjunto resultante puede ser visto como una fuente consistente y adecuada de datos de calidad para la aplicación de algoritmos de minería de datos. El preprocesamiento, incluye un rango amplio de técnicas que podemos agrupar en dos áreas: preparación de datos y reducción de datos.

Las técnicas de reducción de datos se orientan a obtener una representación reducida de los datos originales, manteniendo en la mayor medida posible la integridad y la información existente en los datos.

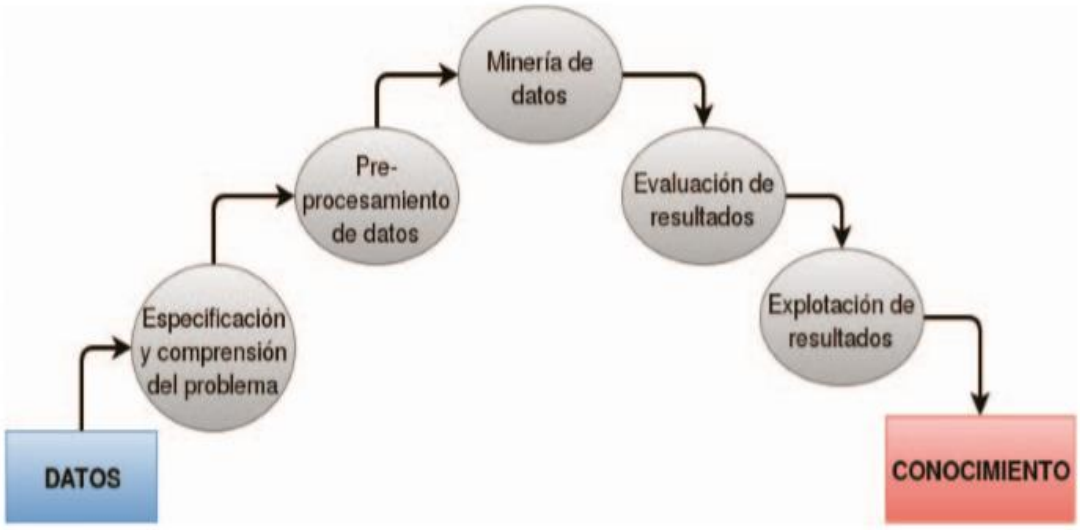

Figura 1 Prepocesamiento de una base de datos. Fuente**:** Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada (España)

3.1 Programa utilizado R

R es un software para el análisis estadístico de datos considerado como uno de los más interesantes. Apoyan esta opinión la vasta variedad de métodos estadísticos que cubre, las

capacidades gráficas que ofrece y, también muy importante, el hecho de ser un software libre, es decir, gratuito.

R es una implementación de software libre del lenguaje S pero con soporte de alcance estático. Se trata de uno de los lenguajes más utilizados en investigación por la comunidad estadística, siendo además muy popular en el campo de la minería de datos, la investigación biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y gráficas

3.2 Limpieza de los datos

No es posible lograr un buen resultado final en un proceso, acorde a los objetos marcados, sino se realiza previamente una buena limpieza de los datos. Sin esta etapa previa no es posible disponer de una base de datos de calidad que permite la toma de decisiones acertadas a nivel estratégico. Esto da una idea de la enorme necesidad de tomarse muy en serio esta etapa, realizándola acorde a unos parámetros correctos.

3.2.1 Representación de los valores perdidos

En primer lugar, es interesante saber la naturaleza de cada una de las variables que componen la base de datos:

```
Classes 'data.table' and 'data.frame':  1764824 obs. of  42 variables:
 $ DATE                : Factor w/ 3099 levels "2005-01-03","2005-01-04",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ RP_ENTITY_ID        : Factor w/ 1768 levels "00067A","003B70",..: 9 9 25 52 52 52 74 75 77 77 ...
 $ GLOBAL_ALL          : num  0.61 0.61 0.01 0.633 0.633 ...
 $ GLOBAL_HEAD         : num  NA NA 0.01 0.633 0.633 ...
 $ GLOBAL_BODY         : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GLOBAL_ALL_SG90     : num  0.61 NA 0.01 0.66 NA 0.66 0.55 0 0.62 NA ...
 $ GLOBAL_HEAD_SG90    : num  NA NA 0.01 0.66 NA 0.66 NA NA 0.62 NA ...
 $ GLOBAL_BODY_SG90    : num  0.61 NA NA NA NA NA 0.55 0 NA NA ...
 $ GLOBAL_ALL_SG365    : num  0.61 NA 0.01 NA NA 0.7 0.55 0 NA NA ...
 $ GLOBAL_HEAD_SG365   : num  NA NA 0.01 NA NA 0.7 NA NA NA NA ...
 $ GLOBAL_BODY_SG365   : num  0.61 NA NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_A_ALL         : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_HEAD        : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_BODY        : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_A_ALL_SG90    : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_HEAD_SG90   : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_BODY_SG90   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ GROUP_A_ALL_SG365   : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_HEAD_SG365  : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_BODY_SG365  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ GROUP_E_ALL         : num  0.61 0.61 NA NA NA 0.7 NA 0 NA NA ...
 $ GROUP_E_HEAD        : num  NA NA NA NA NA 0.7 NA NA NA NA ...
 $ GROUP_E_BODY        : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_E_ALL_SG90    : num  0.61 NA NA NA NA 0.7 NA 0 NA NA ...
 $ GROUP_E_HEAD_SG90   : num  NA NA NA NA NA 0.7 NA NA NA NA ...
 $ GROUP_E_BODY_SG90   : num  0.61 NA NA NA NA NA NA 0 NA NA ...
 $ GROUP_E_ALL_SG365   : num  0.61 NA NA NA NA 0.7 NA 0 NA NA ...
 $ GROUP_E_HEAD_SG365  : num  NA NA NA NA NA 0.7 NA NA NA NA ...
 $ GROUP_E_BODY_SG365  : num  0.61 NA NA NA NA NA NA 0 NA NA ...
 $ GROUP_AM_ALL        : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_HEAD       : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_BODY       : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_AM_ALL_SG90   : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_HEAD_SG90  : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_BODY_SG90  : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_ALL_SG365  : logi  NA NA NA NA NA NA ...
 $ GROUP_AM_HEAD_SG365: logi  NA NA NA NA NA NA ...
 $ GROUP_AM_BODY_SG365: logi  NA NA NA NA NA NA ...
 $ T0_RETURN           : num  0.01852 0.01852 -0.00141 -0.00546 -0.00546 ...
 $ T1_RETURN           : num  -0.03753 -0.03753 0.00774 0.00811 0.00811 ...
 $ ANYO                : num  2005 2005 2005 2005 2005 ...
 $ MES                 : num  1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Figura 2 Tipología de las variables de la base de datos. Fuente: Elaboración propia

Como podemos observar, la mayoría de las variables son numéricas exceptuando 'DATE' y 'RP_ENTITY_ID' que son factores. También observamos, que las variables que contienen 'GROUP_AM' son de carácter lógico.

Realizando un análisis superficial, observamos que tenemos muchos valores perdidos, por lo que sería interesante realizar un estudio detallado para tener mayor conocimiento de ello.

Los datos ausentes son algo habitual, de hecho, rara es la investigación en la que no aparece este tipo de datos.

El primer paso en el tratamiento de datos ausentes consiste en evaluar la magnitud del problema. Por tanto,si existen casos con un alto porcentaje de datos ausentes se deberían excluir del

problema. Así mismo, si existe una variable con un alto porcentaje de este valores perdidos su exclusión dependerá de la importancia teórica de la misma.

En primer lugar, realizaremos un estudio de los valores perdidos por años, para conocer los datos faltantes:

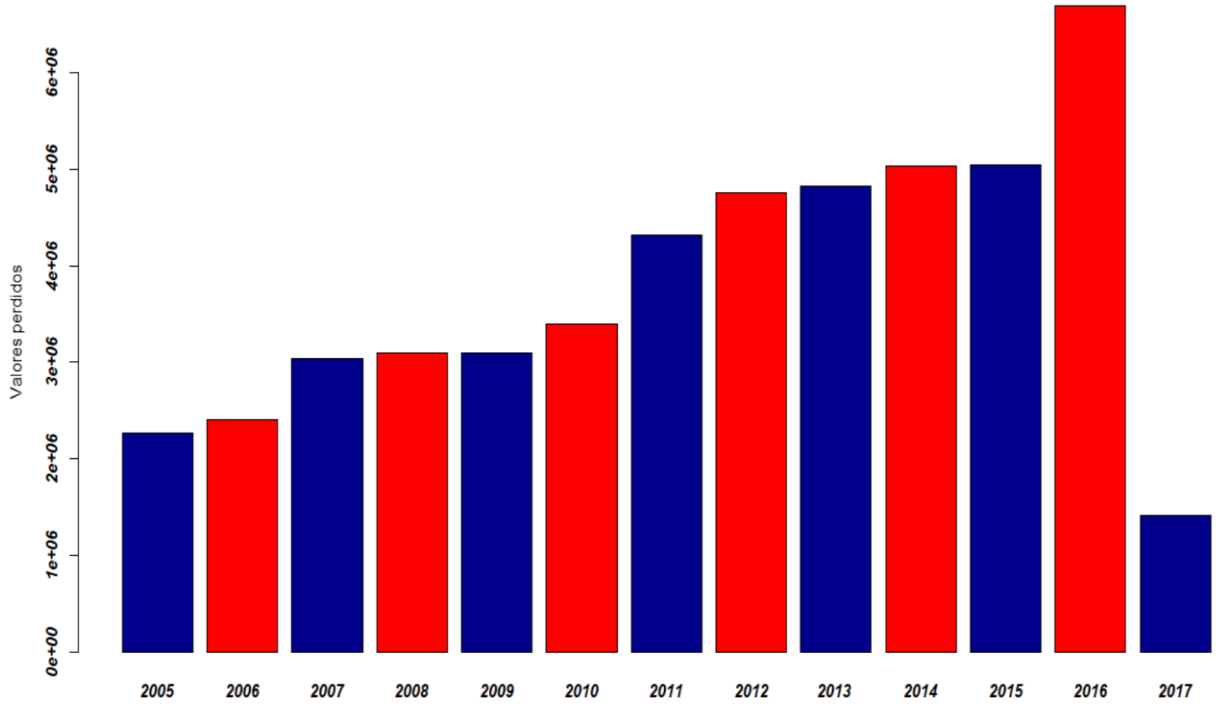

Figura 3 Representación de valores perdidos por año. Fuente: Elaboración propia

Como podemos comprobar, tenemos muchos missing values por año. Concretamente, un total de 49.370.111valores perdidos, lo que supone un 66.60% de datos del dataset. La nota negativa de este gráfico, es que el número de missing values aumenta conforme nos acercamos al año 2017. Cabe destacar, que del año 2017 únicamente tenemos datos hasta Febrero.

Antes de tomar cualquier decisión sobre el dataset, nos centramos en el análisis de missing values por columnas. Es decir, puede ser que en un año tengamos muchos valores perdidos pero estos sean producidos mayoritariamente por varias columnas.

A continuación, representamos los valores perdidos para las diferentes variables del dataset: "GLOBAL_ALL","GLOBAL_HEAD","GLOBAL_BODY","GLOBAL_ALL_SG90","GLOBAL_HEAD_SG90","GLOBAL_BODY_SG90","GLOBAL_ALL_SG365","GLOBAL_HEAD_SG365","GLOBAL_BODY_SG365","GROUP_A_ALL","GROUP_A_HEAD","GROUP_A_BODY", "GROUP_A_ALL_SG90":

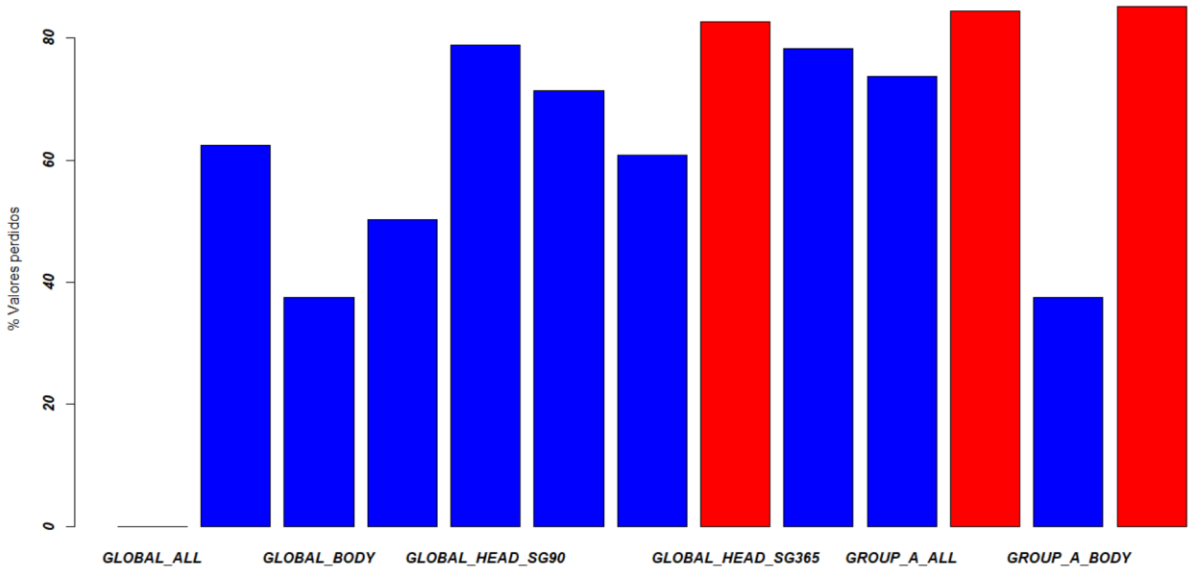

Figura 4 **Representación de valores perdidos por cada variable. Fuente: Elaboración propia**

En azul se muestra aquellas variables que tienen un porcentaje de missing values menor del 80%. En rojo, aparecen aquellas variables que tienen más de un 80% de valores perdidos. Por tanto, podemos concluir que las variables que tienen más de un 80% de valores perdidos son: "GLOBAL_HEAD_SG365","GROUP_A_ALL_SG90","GROUP_A_HEAD".

Continuamos con la representación de las siguientes variables del dataset: "GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_ALL","GROUP_E_HEAD","GROUP_E_BODY","GROUP_E_ALL_SG90","GROUP_E_HEAD_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365","GROUP_E_HEAD_SG365"

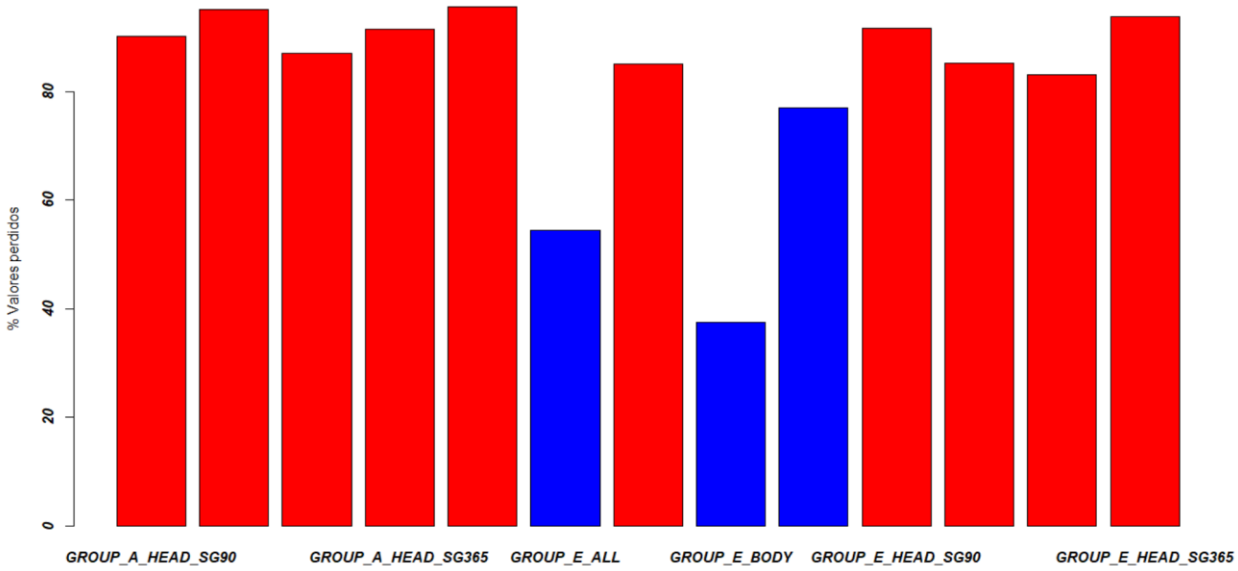

Figura 5 Representación de valores perdidos por cada variable. Fuente: Elaboración propia

En este caso, observamos que tenemos muchas variables con un porcentaje mayor al 80%.Dichas variables son: "GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_HEAD",GROUP_E_HEAD_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365", "GROUP_E_HEAD_SG365".

Y por último, representamos las últimas variables: "GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM_BODY","GROUP_AM_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG90","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365", "GROUP_AM_BODY_SG365", "T0_RETURN","T1_RETURN"

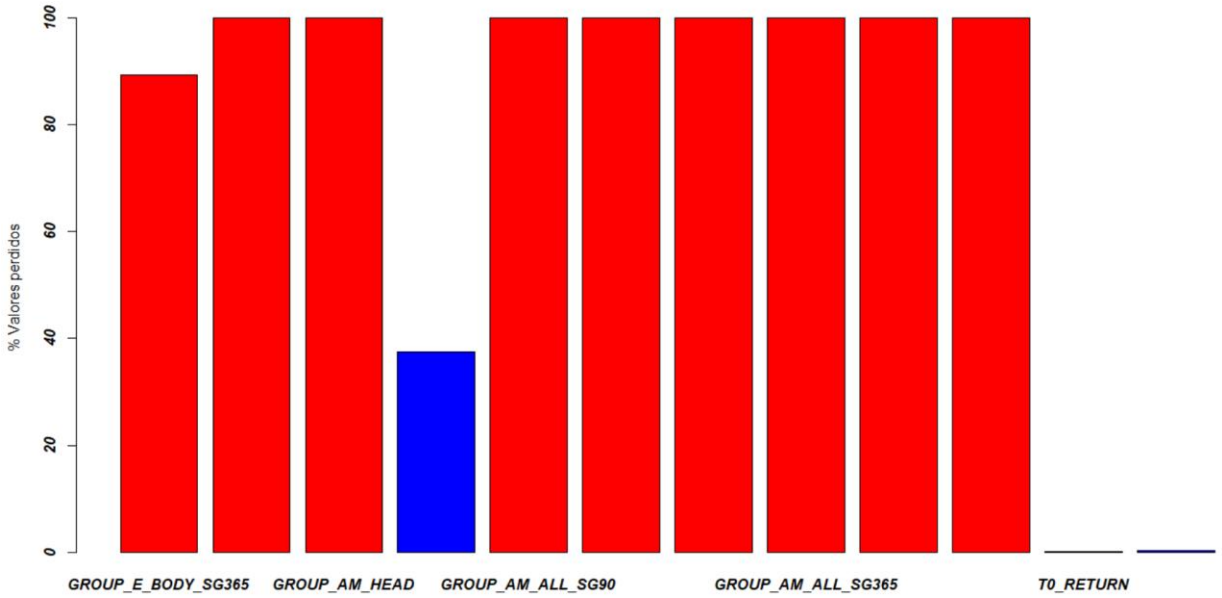

Figura 6 Representación de valores perdidos por cada variable. Fuente: Elaboración propia

Por tanto,podemos concluir que las columnas que están por encima de un 80% de missing values son:"GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG90","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365", "GROUP_AM_BODY_SG365".

Como nota signiticativa, observabamos que tenemos columnas con un 100% de valores perdidos.

Para finalizar, podemos observar el porcentaje de missing values de cada columna:

DATE	RP_ENTITY_ID	GLOBAL_ALL	GLOBAL_HEAD	GLOBAL_BODY
0.0000000	0.0000000	0.0000000	62.4909339	37.5090661
GLOBAL_ALL_SG90	GLOBAL_HEAD_SG90	GLOBAL_BODY_SG90	GLOBAL_ALL_SG365	GLOBAL_HEAD_SG365
50.2278414	78.7908596	71.4369818	60.8873746	82.6216099
GLOBAL_BODY_SG365	GROUP_A_ALL	GROUP_A_HEAD	GROUP_A_BODY	GROUP_A_ALL_SG90
78.2657647	73.6694424	84.4684796	37.5090661	85.1897980
GROUP_A_HEAD_SG90	GROUP_A_BODY_SG90	GROUP_A_ALL_SG365	GROUP_A_HEAD_SG365	GROUP_A_BODY_SG365
90.1185047	95.0712932	87.0302081	91.4044120	95.6257961
GROUP_E_ALL	GROUP_E_HEAD	GROUP_E_BODY	GROUP_E_ALL_SG90	GROUP_E_HEAD_SG90
54.4465624	85.0187894	37.5090661	76.9986696	91.7016654
GROUP_E_BODY_SG90	GROUP_E_ALL_SG365	GROUP_E_HEAD_SG365	GROUP_E_BODY_SG365	GROUP_AM_ALL
85.2970041	83.0820524	93.7054346	89.3766177	100.0000000
GROUP_AM_HEAD	GROUP_AM_BODY	GROUP_AM_ALL_SG90	GROUP_AM_HEAD_SG90	GROUP_AM_BODY_SG90
100.0000000	37.5090661	100.0000000	100.0000000	100.0000000
GROUP_AM_ALL_SG365	GROUP_AM_HEAD_SG365	GROUP_AM_BODY_SG365	T0_RETURN	T1_RETURN
100.0000000	100.0000000	100.0000000	0.1599026	0.3301179
ANYO	MES			
0.0000000	0.0000000			

Figura 7 Porcentaje de missing values en el dataset. Fuente: Elaboración propia

Una vez que conocemos el porcentaje de valores perdidos de cada columna nos hacemos la siguiente pregunta, ¿A partir de qué porcentaje de missing values no recuperaremos dichos valores? Esta pregunta tiene una amplia gama de respuestas por parte de los expertos en este sector, ya que ronda entre el 60% y 90%.

En nuestro caso, recuperaremos todas aquellas variables que tenga como máximo un 80% de valores perdidos.

3.2.2 Eliminación de las variables

Como he comentado anteriormente, eliminaremos aquellas variables que tengan más de un 80% de valores perdidos. Por tanto, las variables eliminadas serán las siguientes:

"GLOBAL_HEAD_SG365","GROUP_A_ALL_SG90","GROUP_A_HEAD","GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_HEAD",GROUP_E_HEAD_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365","GROUP_E_HEAD_SG365","GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG90","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365", "GROUP_AM_BODY_SG365".

Lo que evidentemente supone una reducción de 22 variables, pasando de las 42 que teníamos en un principio a 20 variables.

3.3 Normalización de los datos

La normalización de la base datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

3.3.1 Eliminación de duplicados

Un aspecto muy importante es la eliminación de duplicados. Es decir, puede darse el caso de tener información redundante en la base de datos con filas duplicadas. Afortunadamente, en nuestra base de datos objeto de estudio no tenemos ningún elemento duplicado, por lo que el número de observaciones se mantienen. En R, se calcula fácilmente utilizando el comando duplicated().

3.3.2 Comprobación del tipo de datos de las columnas

Es muy importante que cada columna sea de un único tipo de datos. Como podemos comprobar en nuestro caso se cumple, ya que solo tenemos dos variables que son factores y el resto que son numéricas.

```
Classes ‘data.table’ and 'data.frame':  1764824 obs. of  20 variables:
 $ DATE             : Factor w/ 3099 levels "2005-01-03","2005-01-04",..
 $ RP_ENTITY_ID     : Factor w/ 1768 levels "00067A","003B70",..: 9 9 25
 $ GLOBAL_ALL       : num  0.61 0.61 0.01 0.633 0.633 ...
 $ GLOBAL_HEAD      : num  NA NA 0.01 0.633 0.633 ...
 $ GLOBAL_BODY      : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GLOBAL_ALL_SG90  : num  0.61 NA 0.01 0.66 NA 0.66 0.55 0 0.62 NA ...
 $ GLOBAL_HEAD_SG90 : num  NA NA 0.01 0.66 NA 0.66 NA NA 0.62 NA ...
 $ GLOBAL_BODY_SG90 : num  0.61 NA NA NA NA NA 0.55 0 NA NA ...
 $ GLOBAL_ALL_SG365 : num  0.61 NA 0.01 NA NA 0.7 0.55 0 NA NA ...
 $ GLOBAL_BODY_SG365: num  0.61 NA NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_A_ALL      : num  NA NA 0.01 NA NA NA NA NA NA NA ...
 $ GROUP_A_BODY     : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_E_ALL      : num  0.61 0.61 NA NA NA 0.7 NA 0 NA NA ...
 $ GROUP_E_BODY     : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ GROUP_E_ALL_SG90 : num  0.61 NA NA NA NA 0.7 NA 0 NA NA ...
 $ GROUP_AM_BODY    : num  0.61 0.61 NA NA NA NA 0.55 0 NA NA ...
 $ T0_RETURN        : num  0.01852 0.01852 -0.00141 -0.00546 -0.00546 ..
 $ T1_RETURN        : num  -0.03753 -0.03753 0.00774 0.00811 0.00811 ...
 $ ANYO             : num  2005 2005 2005 2005 2005 ...
 $ MES              : num  1 1 1 1 1 1 1 1 1 1 ...
```

Figura 8 Tipología de las variables de la base de datos. Fuente: Elaboración propia

3.4 Imputación de los datos faltantes

Una vez que hemos eliminados las variables que tenían más de un 80% de valores perdidos, llevaremos la imputación de dichos valores de las variables que forman parte del dataset.

En estadística, la imputación es el proceso de reemplazar los valores perdidos con valores sustitutos. El objetivo de la imputación es rellenar los valores perdidos con estimaciones (realizadas con el método de aprendizaje más apropiado para cada caso) de estos teniendo en cuenta las relaciones posibles entre las observaciones. Existen muchos métodos para la imputación de valores faltantes, desde métodos de imputación simples como pueden ser la media y la interpolación hasta métodos más complejos.

Antes de comenzar la imputación, podemos observar a continuación como se encuentra el dataset antes de la imputación de valores:

	DATE	RP_ENTITY_ID	GLOBAL_ALL	GLOBAL_HEAD	GLOBAL_BODY	GLOBAL_ALL_SG90
1	2005-01-03	0157B1	0.61000000	NA	0.6100000	0.6100000
2	2005-01-03	0157B1	0.61000000	NA	0.6100000	NA
3	2005-01-03	046263	0.01000000	0.0100000	NA	0.0100000
4	2005-01-03	07CA6A	0.63333333	0.6333333	NA	0.6600000
5	2005-01-03	07CA6A	0.63333333	0.6333333	NA	NA
6	2005-01-03	07CA6A	0.63333333	0.6333333	NA	0.6600000
7	2005-01-03	0BC29E	0.55000000	NA	0.5500000	0.5500000
8	2005-01-03	0BC6D8	0.00000000	NA	0.0000000	0.0000000
9	2005-01-03	0BE0AE	0.62000000	0.6200000	NA	0.6200000
10	2005-01-03	0BE0AE	0.62000000	0.6200000	NA	NA
11	2005-01-03	0D920D	0.00000000	NA	0.0000000	0.0000000

Figura 9 Dataset con presencia de missing values. Fuente: Elaboración propia

3.4.1 Filtro de Kalman

Para llevar a cabo la imputación de datos faltantes nos decantaremos por el Filtro de Kalman.

El filtro de Kalman fue creado por Rudolf Emil Kalman en 1958, pero hasta el año 1961 no publicó su trabajo, lo que supuso una revolución en el campo de la estimación

El filtrado de Kalman, también conocido como estimación lineal cuadrática (LQE), es un algoritmo que utiliza una serie de mediciones observadas a lo largo del tiempo, que contiene ruido estadístico y otras imprecisiones, y produce estimaciones de variables desconocidas que tienden a ser más precisas que las basadas en medición individual sola, mediante el uso de inferencia Bayesiana y la estimación de una distribución de probabilidad conjunta sobre las variables para cada intervalo de tiempo.

El filtro de Kalman es un concepto ampliamente utilizado en el análisis de series temporales utilizado en campos como el procesamiento de señales y la econometría.

El algoritmo funciona en un proceso de dos pasos. En el paso de predicción, el filtro de Kalman produce estimaciones de las variables de estado actuales, junto con sus incertidumbres. Una vez

que se observa el resultado de la siguiente medición (con cierta cantidad de error, incluido el ruido aleatorio), estas estimaciones se actualizan utilizando un promedio ponderado, y se da más peso a las estimaciones con mayor certeza. El propósito de las ponderaciones es que los valores con mejor incertidumbre estimada (es decir, más pequeña) sean más "confiables". Los pesos se calculan a partir de la covarianza, una medida de la incertidumbre estimada de la predicción del estado del sistema. El resultado del promedio ponderado es una nueva estimación del estado que se encuentra entre el estado pronosticado y el estado medido, y tiene una incertidumbre estimada mejor que cualquiera de los dos. El algoritmo es recursivo. Puede ejecutarse en tiempo real, utilizando solo las medidas de entrada actuales y el estado previamente calculado y su matriz de incertidumbre; no se requiere información previa adicional.

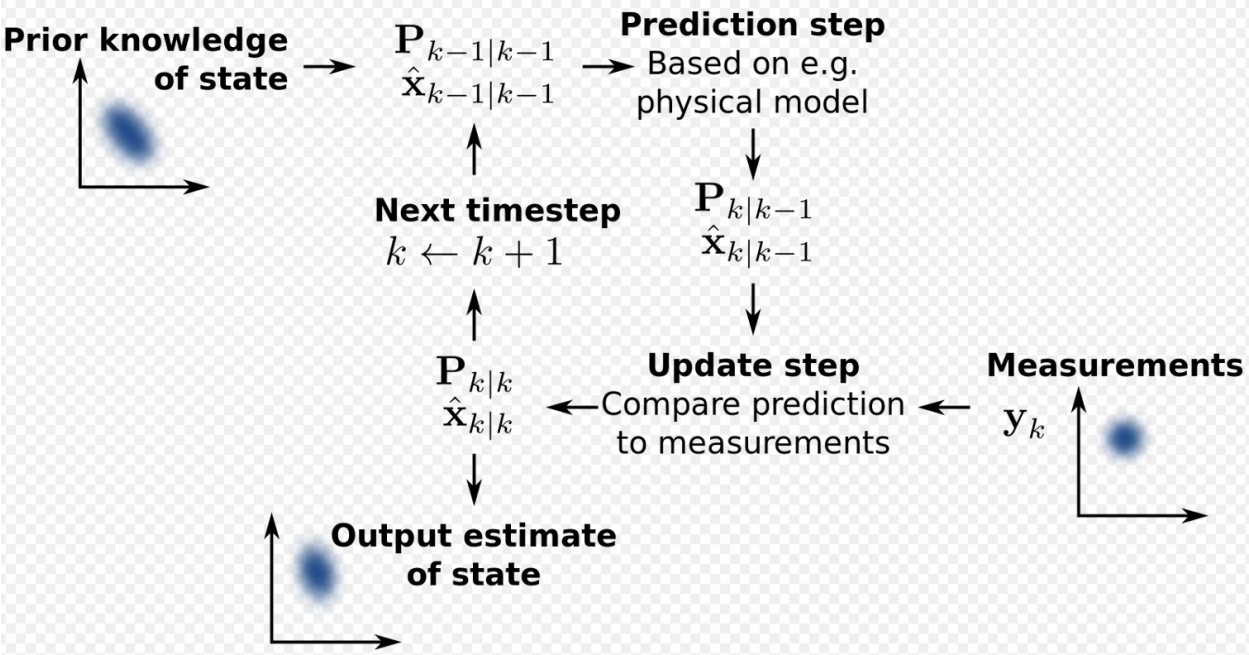

Figura 10 Funcionamiento del filtro de kalman. Fuente: Wikipedia

3.5 Transformación de los datos

Para llevar a cabo la imputación de los datos, hemos reestructurado el dataset original en una lista de dataset a través de la variable RP_ENTITY_ID. Por tanto, para cada empresa tendremos un dataset,y a su vez a cada dataset le hemos imputado los datos faltantes. A través de la variable RP_ENTITY_ID podemos averiguar el número de empresas que forman parte del dataset original. Por tanto, en el dataset objeto de estudio tenemos 1786 empresas, lo que supone 1786 datasets.

Por ejemplo, para la empresa ‘0157B1’ sería el siguiente (omitimos datos porque el dataset es muy grande):

	DATE	RP_ENTITY_ID	GLOBAL_ALL	GLOBAL_HEAD	GLOBAL_BODY	GLOBAL_ALL_SG90	GLOBAL_HEAD_SG90
1	2005-01-03	0157B1	0.61000000	-0.98000000	0.610000000	0.610000000	-0.9574686730
2	2005-01-03	0157B1	0.61000000	-0.98000000	0.610000000	0.182516121	-0.9350444522
183	2005-01-04	0157B1	-0.98000000	-0.98000000	0.143603550	-0.980000000	-0.9800000000
184	2005-01-04	0157B1	-0.98000000	-0.98000000	-0.131021548	-0.980000000	-0.9800000000
185	2005-01-04	0157B1	-0.98000000	-0.98000000	-0.405646645	-0.249469242	-0.8667409120
186	2005-01-04	0157B1	-0.98000000	-0.87000000	-0.980000000	-0.980000000	-0.8435947010
426	2005-01-05	0157B1	-0.98000000	-0.76000000	-0.980000000	-0.980000000	-0.8204484900
427	2005-01-05	0157B1	-0.98000000	-0.65000000	-0.980000000	-0.283818623	-0.7973022790
1303	2005-01-10	0157B1	0.55000000	-0.54000000	0.550000000	-0.237655803	-0.7741560681
2465	2005-01-14	0157B1	0.54000000	-0.43000000	0.540000000	0.540000000	-0.7510098571
4173	2005-01-21	0157B1	-0.67000000	-0.32000000	-0.670000000	-0.670000000	-0.7278636461
4174	2005-01-21	0157B1	-0.67000000	-0.21000000	-0.670000000	-0.187542119	-0.7047174351

Figura 11 Dataset de la empresa ‘0157B1’ con datos imputados. Fuente: Elaboración propia

Como podemos observar, tenemos para un mismo día varias observaciones. Por lo que haremos es agruparlos en una sola observación por día. Por ello, agruparemos haciendo la media. Es destacable, que muchas de estas variables coinciden en las diferentes observaciones.

Para mayor comodidad, nos quedaremos con una sola variable por mes. Tomo esta decisión porque no tengo un número determinado de observaciones por mes, y por tanto es más fácil para llevar a cabo representaciones.

	DATE	ANYO2	GLOBAL_ALL2	GLOBAL_HEAD2	GLOBAL_BODY2	GLOBAL_ALL_SG902	GLOBAL_HEAD_SG902
1	2005-01-03	2005	0.61000000	-0.98000000	0.61000000	0.396258060	-0.946256563
2	2005-01-04	2005	-0.98000000	-0.95250000	-0.34326616	-0.797367310	-0.917583903
3	2005-01-05	2005	-0.98000000	-0.70500000	-0.98000000	-0.631909311	-0.808875385
4	2005-01-10	2005	0.55000000	-0.54000000	0.55000000	-0.237655803	-0.774156068
5	2005-01-14	2005	0.54000000	-0.43000000	0.54000000	0.540000000	-0.751009857
6	2005-01-21	2005	-0.67000000	-0.26500000	-0.67000000	-0.428771060	-0.716290541
7	2005-01-24	2005	-0.67000000	-0.10000000	-0.67000000	-0.165287910	-0.681571224
8	2005-01-25	2005	0.01000000	0.01000000	-0.37918568	0.010000000	0.010000000
9	2005-02-01	2005	0.00000000	0.08985294	0.00000000	-0.134266288	-0.638343049
10	2005-02-02	2005	-0.74000000	0.16970588	-0.74000000	-0.740000000	-0.618261084
11	2005-02-03	2005	0.20000000	0.27598039	0.12566608	0.150191482	-0.263876201

Figura 12 Dataset de la empresa ‘0157B1’ con datos imputados. Fuente: Elaboración propia

3.5.1 Dataset final

Uno de los problemas que hemos observado es que no todos los datasets de cada empresa empiezan en Enero del 2017 y acaban en Febrero de 2017. Por tanto, nos quedamos con aquellas series que tienen datos entre Diciembre del 2016 y Febrero de 2017. Quedándonos finalmente con una lista de 1056 datasets.

4 Clasificación de series temporales mediante índice de predictibilidad

La estacionalidad o variación estacional de una serie temporal es la variación periódica y predecible de la misma con un periodo inferior o igual a un año. Es una de las componentes de las series temporales, y se contrapone a la tendencia (comportamiento a largo plazo) y a la variación cíclica (variación periódica con un periodo superior al año).

En esta parte del trabajo, tratamos de diferenciar si una serie es o no estacional cuando se conoce el período potencial, que en este caso es de años.

Un simple enfoque sería ajustar un modelo que introduce estacionalidad si esta está presente. En nuestro caso, ajustamos con un modelo ETS(Exponential smoothing state space model) usando ets() en R, y si el modelo elegido tiene una componente estacional, entonces los datos son estacionales. Ya que la función ets selecciona un modelo ETS (A, N, A). Es decir, detecta un componente estacional aditivo.

La selección del modelo se basa en el AIC en lugar de cualquier prueba de hipótesis. El criterio de información de Akaike (AIC) es una medida de la calidad relativa de un modelo estadístico, para un conjunto dado de datos. Como tal, el AIC proporciona un medio para la selección del modelo.

Basándonos en el paper de (2017, Hyndman) esta no es una prueba formal de estacionalidad, ya que la selección del modelo se basa en la AIC y no en una prueba de hipótesis. Sin embargo, existe una prueba de probabilidad logarítmica relacionada basada en la diferencia entre el modelo seleccionado y el modelo equivalente con un término estacional adicional añadido. Dos veces la diferencia entre las dos probabilidades logarítmicas tendrá una distribución chi cuadrado según el teorema de Wilks. Los grados de libertad serán la diferencia en el número de parámetros que se estiman en los dos modelos.

La componente estacional mejora la precisión del pronóstico, y eso es precisamente lo que nos dice el AIC. Minimizar el AIC es asintóticamente equivalente a minimizar el MSE de one-step-head out-of-sample. Entonces, un AIC más pequeño significa mejores pronósticos, y eso es lo que generalmente nos importa.

De esta forma, mediante un bucle obtengo una lista en la que indicamos mediante true o false si la variable respuesta de cada dataset es volátil o estacional. Por tanto, almaceno true para las series estacionales y false para las series son volátiles.

5 Conceptos estadísticos y modelados

El contenido incorpora todos los conceptos necesarios para la comprensión de la metodología llevada en este proyecto, tanto estadística como en lo referido a modelos predictivos.

Finalmente, nos quedamos con una lista de 1044 valores que pueden ser true o false. True nos indica que la serie es estacional y false nos indica que la serie es volátil. Este hecho es muy importante, ya que no se puede tratar de igual manera a una serie que sea estacional de la que no lo es.

A la hora de la predicción utilizaremos la función predict(),la cual se encuentra sobrecargada con una versión específica para cada modelo utilizado.

El método predictivo para realizar la medición del error será mediante la predicción ‘One step ahead’, consistente en ir incorporando al modelo nuevos datos a medida que se dispone de ellos, con el objeto de reestimar cada vez el modelo y obtener previsiones para un solo período hacia delante.

Por tanto, independientemente de si la serie es estacional o no, dividiremos nuestros datos en entrenamiento y test basándonos en la variable respuesta T1_RETURN. La parte de entrenamiento estará compuesta de todos los valores de la variable respuesta, exceptuando el último valor que será asignado al test.

Existen varios modelos disponibles para el modelado y el pronóstico de series de tiempo. Estos abarcan desde aproximaciones más tradicionales como los modelos de suavizado exponencial y modelos ARIMA (paquete forecast), hasta modelos no lineales en media como las redes neuronales autoregresivas y modelos de transición de regímenes (paquete tsDyn), y los modelos de varianza condicional

En el caso de nuestras series estacionales las trataremos mediante la función auto.arima de R y en el caso de las volátiles nos basaremos en las cross-validation.

Destacar que disponemos de 17 series estacionales y el resto volátiles.

En el caso de los modelos ARIMA, la función auto.arima() realiza una búsqueda entre un conjunto de posibles modelos y selecciona el mejor de ellos basado en un criterio de información.

Esta es una clara ventaja, ya que la correcta especificación de un modelo ARIMA es una tarea difícil, aún más cuando la dinámica de la serie de tiempo es compleja.

El proceso seguido es el siguiente:

La función auto.arima () en R usa una variación del algoritmo de Hyndman y Khandakar que combina pruebas de raíz unitaria, minimización de AICc y MLE para obtener un modelo ARIMA. El algoritmo sigue estos pasos.

Algoritmo Hyndman-Khandakar para modelado ARIMA automático

1. El número de diferencias d se determina usando tests repetidos de KPSS.
2. Los valores de p y q se eligen minimizando el AICc después de diferenciar los datos d veces. En lugar de considerar todas las combinaciones posibles de p y q, el algoritmo usa una búsqueda por pasos para recorrer el espacio modelo.
a) El mejor modelo (con el AICc más pequeño) se selecciona de entre los cuatro siguientes: ARIMA(2,d,2),

ARIMA(0,d,0),

ARIMA(1,d,0),

ARIMA(0,d,1).
Si d = 0, entonces se incluye la constante c; si d≥1, entonces la constante c se establece en cero. Esto se llama el "modelo actual".
b) Se consideran las variaciones en el modelo actual:
var p y / o q del modelo actual en ± 1
incluir / excluir c del modelo actual.

El mejor modelo considerado hasta ahora (ya sea el modelo actual o una de estas variaciones) se convierte en el nuevo modelo actual.
c) Se repite el Paso 2 b) hasta que no se encuentre AICc inferior.

En el caso de las series volátiles utilizaremos la validación cruzada. La validación cruzada o cross-validation es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. En las series volátiles consideraremos variables exógenas para intentar enriquecer el modelo y así predecir la serie con exógenas. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica. Es una técnica muy utilizada en proyectos de inteligencia artificial para validar modelos generados.

Concretamente utilizaremos la función cv.lasso.1() de R. En Estadística y Aprendizaje Automático, Lasso (least absolute shrinkage and selection operator, por sus siglas en inglés), es

un método de análisis de regresión que realiza selección de variables y regularización para mejorar la exactitud e interpretabilidad del modelo estadístico producido por este.

5.1 Ejemplos de representación de una serie estacionaria y otra volátil

Para llevar a cabo la representación, denotaremos de color rojo para los datos de training, y azul para la predicción.

Para llevar a cabo la representación, utilizamos al vector vectores() donde tengo almacenados los diferentes valores de cada uno de los datasets, concretamente en la posición diez (podría haber elegido cualquier otra posición que contenga TRUE), para representar una serie estacionaria.

Dentro de vectores(), en cada posición encontramos 5 datos. Que indican:

1) Nombre de la empresa.
2) Indica TRUE si la serie es estacional o FALSE si la serie es volátil.
3) MAE
4) MSE
5) Modelo de la serie.

```
[[1]]
[1] "331BD2"

[[2]]
[1] TRUE

[[3]]
[1] 0.004585524

[[4]]
[1] 2.102703e-05

[[5]]
[1] 0 0 0 0 1 0 0
```

Figura 13 Información sobre la posición 10 del vector. Fuente: Elaboración propia

Representación de la variable respuesta, para dicha posición del vector:

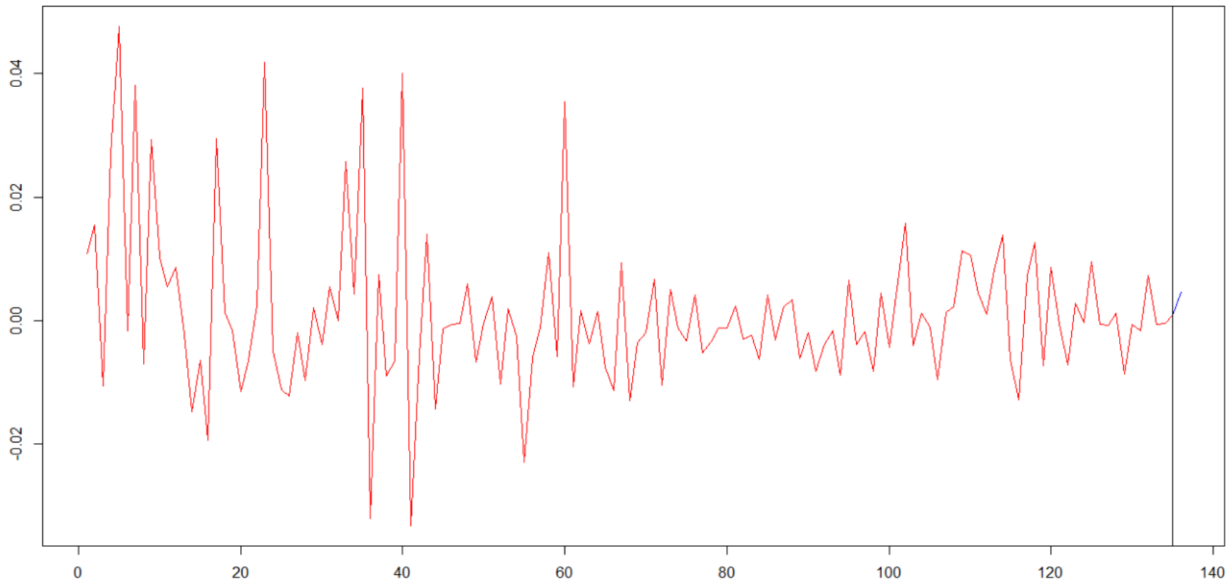

Figura 14 Representación de una serie estacional. Fuente: Elaboración propia

Para llevar a cabo la representación de una serie volátil vamos a la posición uno del vector donde tenemos almacenados los diferentes valores de cada uno de los datasets.

```
[[1]]
[1] "0157B1"

[[2]]
[1] FALSE

[[3]]
[1] 0.0008368336

[[4]]
[1] 7.002905e-07

[[5]]
11 x 1 sparse Matrix of class "dgCMatrix"
                              1
(Intercept)       -0.0003293361
GLOBAL_ALL2        .
GLOBAL_HEAD2       .
GLOBAL_BODY2       .
GLOBAL_ALL_SG902   .
GLOBAL_ALL_SG3652  .
GROUP_A_BODY2      .
GROUP_E_ALL2       .
GROUP_E_BODY2      .
GROUP_AM_BODY2     .
TO_RETURN2         0.6384696933
```

Figura 15 Información sobre la posición 1 del vector. Fuente: Elaboración propia

Las posiciones indican lo siguiente:

1) Nombre de la empresa.

2) Indica TRUE si la serie es estacional o FALSE si la serie es volátil.

3) MAE

4) MSE

5) Peso de las variables exógenas.

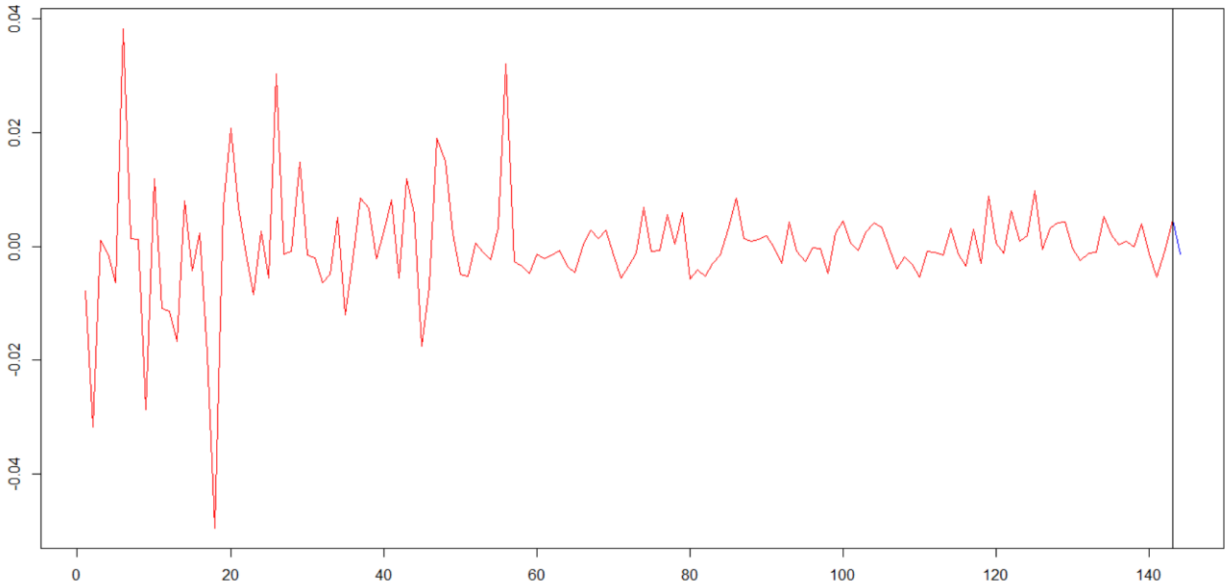

Figura 16 Representación de una serie volátil. Fuente: Elaboración propia

Este apartado lo introducimos a modo de ejemplo, ya que sería impensable representar las 1044 series temporales. Por este motivo, representamos una de cada tipo al azar.

6 Estimación de los errores. MAE y MSE

Para finalizar, calcularemos el MSE y el MAE de cada una de nuestras series temporales.

- ¿Qué es el MSE?

En estadística, el error cuadrático medio (MSE o mean square error en inglés) es una forma de evaluar la diferencia entre un estimador y el valor real de la cantidad que se quiere calcular. El MSE mide el promedio del cuadrado del "error", siendo el error el valor en la que el estimador difiere de la cantidad a ser estimada.

Una forma simple de pensar en el MSE es considerándolo como un criterio para seleccionar un estimador apropiado: en los modelos estadísticos los modeladores deben elegir entre varios estimadores potenciales. En términos prácticos, el MSE equivale a la suma de la varianza y la desviación al cuadrado del estimador. Un estimador es usado para deducir el valor de un parámetro desconocido en un modelo estadístico. La desviación es la diferencia entre el valor esperado del estimador y el valor real del parámetro que se quiere estimar.

En el modelado estadístico, el MSE es usado para determinar la medida en la que el modelo no se ajusta a la información, o si el quitar ciertos términos puede simplificar el modelo de manera beneficiosa. El MSE proporciona una forma para elegir el mejor estimador: un MSE mínimo a menudo, pero no siempre, indica una variación mínima, y por lo tanto indica un buen estimador. Al calcular la raíz cuadrada del MSE se obtiene la raíz cuadrada de la desviación media, que es una buena medida de precisión y también es conocida como la media cuadrática.

Como crítica, el MSE coloca más peso en los errores grandes que en los pequeños (como resultado de elevar al cuadrado cada término), y por lo tanto enfatiza datos atípicos de maneras inconsistentes con la mediana de los datos de la muestra.

- ¿Qué es el MAE?

En estadística, el error absoluto medio (MAE) es una medida de la diferencia entre dos variables continuas. Supongamos que X e Y son variables de observaciones emparejadas que expresan el mismo fenómeno. Los ejemplos de Y frente a X incluyen comparaciones de tiempo predicho frente observado, tiempo posterior frente tiempo inicial, y una técnica de medición frente a una técnica alternativa de medición. Si considerásemos un diagrama de dispersión de n puntos, donde el punto i tiene coordenadas (xi, yi), el error absoluto medio (MAE) sería la distancia vertical promedio entre cada punto y la línea Y = X, que también se conoce como la línea One-to-One. MAE es también la distancia horizontal promedio entre cada punto y la línea Y = X.

Como su nombre lo indica, el error absoluto medio, es un promedio de los errores absolutos $|e_i| = |y_i - x_i|$. Tenga en cuenta que las formulas alternativas pueden incluir frecuencias relativas como factores de peso. El error absoluto medio usa la misma escala que los datos que se miden. Esto se conoce como una medida de precisión dependiente de la escala y, por lo tanto, no se puede usar para hacer comparaciones entre series que usan escalas diferentes. El error absoluto medio es una medida común del error de pronóstico en el análisis de series de tiempo

El error absoluto medio es una de las formas de comparar pronósticos con sus resultados eventuales.

En primer lugar, llevaremos a cabo el tratamiento de las series estacionales. Concretamente, de las 1040 series objeto de estudio, únicamente 17 son estacionales.

Para estas 17 series temporales, hemos comprobado que surgen 10 modelos distintos, por lo que hay algunas series que repiten el mismo modelo. Como hemos comentado anteriormente, a la series estacionales le aplicamos el auto.arima(). Como resultado, cada modelo estará compuesto por 7 números, que significan lo siguiente:

ARIMA(p,d,q)(P,D,Q)m

La primera parte (p,d,q) representa la parte no estacional del modelo, y la segunda a la parte estacional (P,D,Q)

m: Número de periodos por temporada.

d: Número de diferencias regulares.

p: Orden del autorregresivo AR.

q: Orden de la móvil regular MA.

D: Numero de diferencias estacionales.

P: Orden del autorregresivo estacional SAR.

Q: Orden de la media móvil estacional SMA.

Primeramente, representaremos la frecuencia de cada modelo y observaremos cuál es el más común.

Figura 17 Frecuencia de modelos aplicados a las series estacionales. Fuente: Elaboración propia

El modelo más frecuente es el '0000100' y el '0100110', apareciendo el primero de ellos en siete series temporales del dataset y el segundo en dos. El resto de los modelos nos están repetidos.

A continuación, representamos en una tabla tanto el valor del MAE como el MSE para cada uno de los modelos.

Tabla 1 MAE y MSE de cada modelo estacional. Fuente: Elaboración propia

Modelo	MAE	MSE
0 0 0 0 1 0 0	0.00525555	3.67E-05
3 1 0 0 1 1 0	0.000483	2.33E-07
1 2 0 0 1 1 0	0.00418347	1.75E-05
5 4 0 0 1 0 0	0.00137956	1.90E-06
2 1 0 0 1 1 0	0.01041544	0.00010848
4 1 0 0 1 1 0	0.00113125	1.28E-06
0 1 0 0 1 1 0	0.00685688	4.82E-05
2 2 0 0 1 0 0	0.00408103	1.67E-05
1 1 0 0 1 1 0	0.01014961	0.00010301
3 0 0 0 1 1 0	0.00929625	8.64E-05

Para saber cuál es el mejor modelo, debemos encontrar aquel que tenga menor error. Por ello, lo representaremos mediante histogramas, ya que observaremos el menor error de una forma clara y concisa.

Representación del MAE:

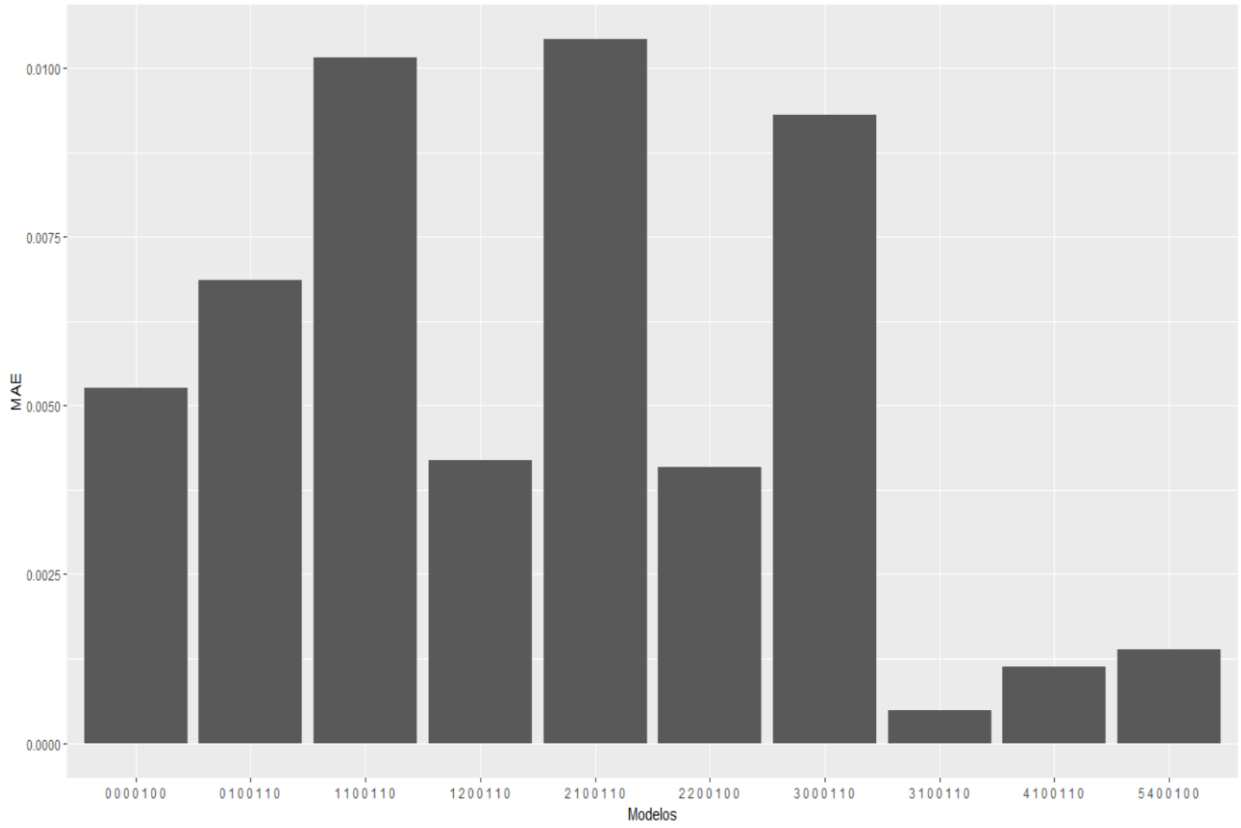

Figura 18 MAE de cada modelo. Fuente: Elaboración propia

En la representación del MAE, observamos claramente que el mejor modelo es el ‘3100110’, teniendo un MAE de 0.000483.

Representación del MSE:

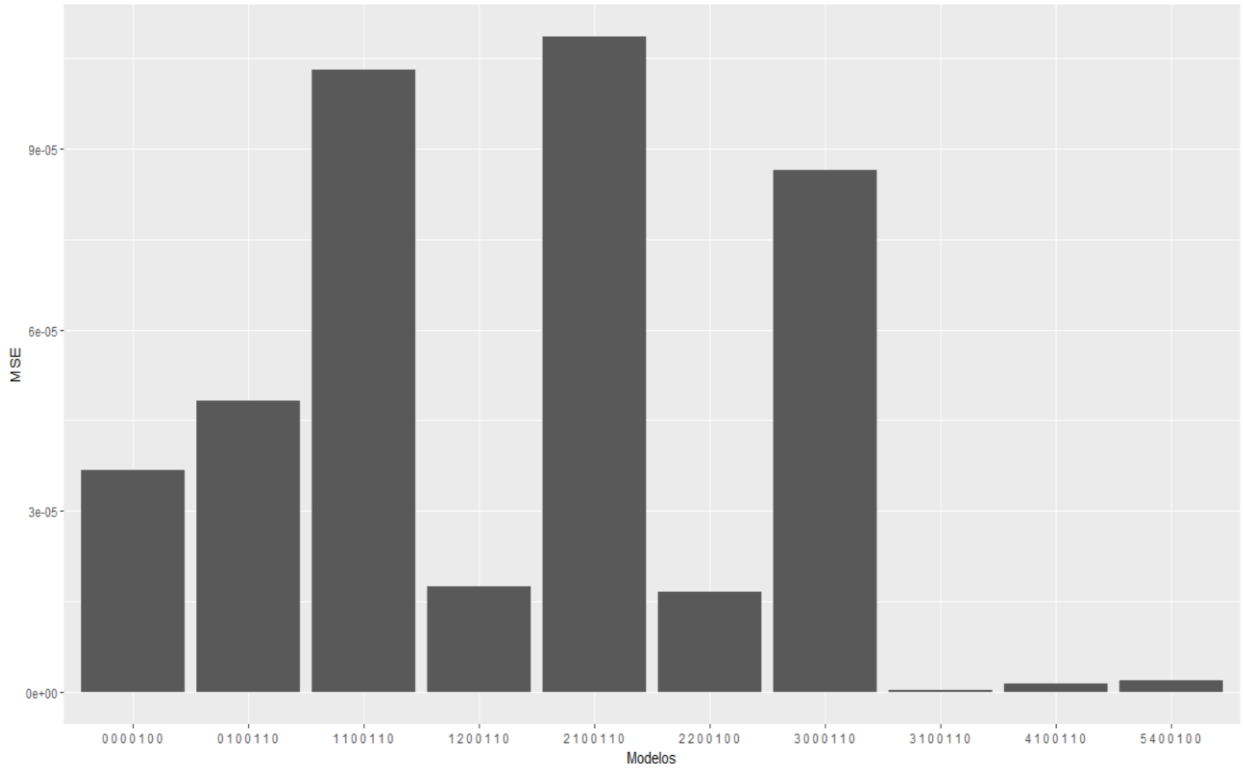

Figura 19 MSE de cada modelo. Fuente: Elaboración propia

En el modelado estadístico, el MSE es usado para determinar la medida en la que el modelo no se ajusta a la información. El MSE proporciona una forma para elegir el mejor estimador: un MSE mínimo a menudo, pero no siempre, indica una variación mínima, y por lo tanto indica un buen estimador. En nuestro caso, el mejor modelo es el '3100110', ya que es el que tiene el menor valor.

Tener un error cuadrático medio de cero es ideal pero no es posible en la mayoría de las situaciones. Un MSE de 0 significa que el estimador predice las observaciones con una precisión perfecta.

En la representación del MSE, observamos claramente que el mejor modelo es el '3100110'.Tanto el MAE como el MSE, nos dan como mejor modelo al '3100110'.

En el caso de las series volátiles, no las podemos agrupar como en el caso de la estacionales, ya que el peso de las variables exógenas van cambiando. Por tanto, lo que vamos a representar son aquellas empresas que tienen un MAE e MSE más bajo.

Tabla 2 MAE y MSE de cada modelo estacional. Fuente: Elaboración propia

RP_ENTITY_ID	MAE	MSE
4D371E	1.08E-05	1.1722E-10
A38B5A	1.10E-05	1.204E-10
2B7A40	1.4686E-05	2.1568E-10
E8846E	2.16E-05	4.6579E-10
14C7B2	2.17E-05	4.6944E-10
F6DCE4	3.68E-05	1.3549E-09
E6E012	4.13E-05	1.7065E-09
EB61C4	6.0417E-05	3.6502E-09
5A6336	6.55E-05	4.2911E-09
F57F6F	6.72E-05	4.5204E-09
FD4E8D	7.14E-05	5.0968E-09

Para saber cuál es el mejor modelo, debemos encontrar aquel que tenga menor error. Por ello, lo representaremos mediante histogramas, ya que observaremos el menor error de una forma clara y concisa.

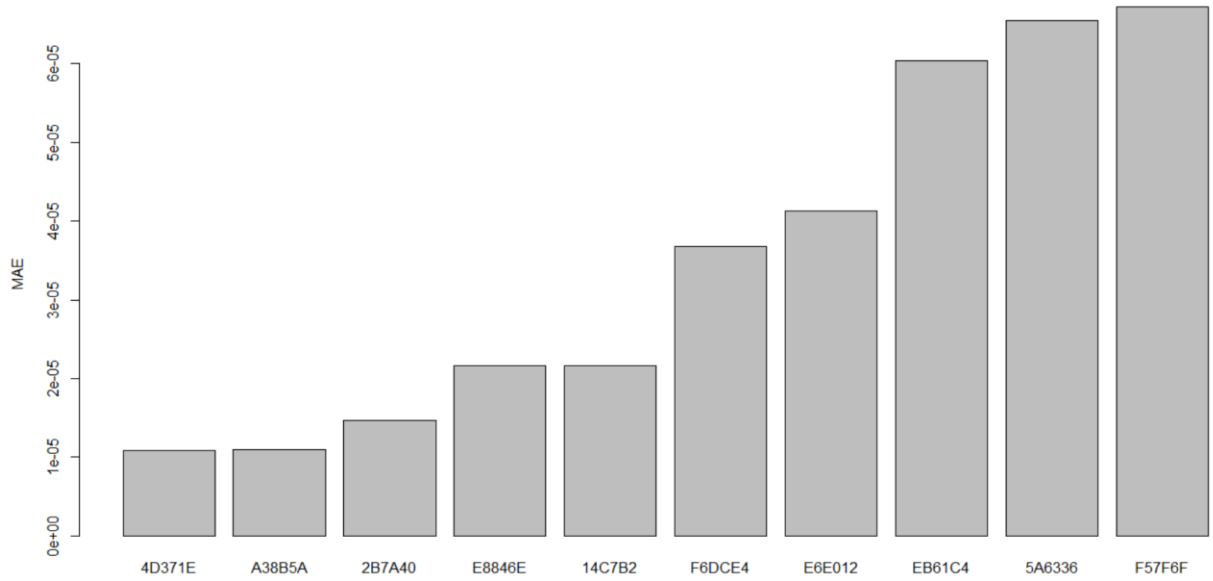

Figura 20 MAE de los 10 mejores modelos volátiles. Fuente: Elaboración propia

La empresa ‘4D371E’ es la que tiene un MAE más pequeño junto con la empresa ‘A38B5A’, teniendo un error de 1,08 x 10^{-5} y 1,10 x 10^{-5} respectivamente.

Seguidamente, representamos el MSE:

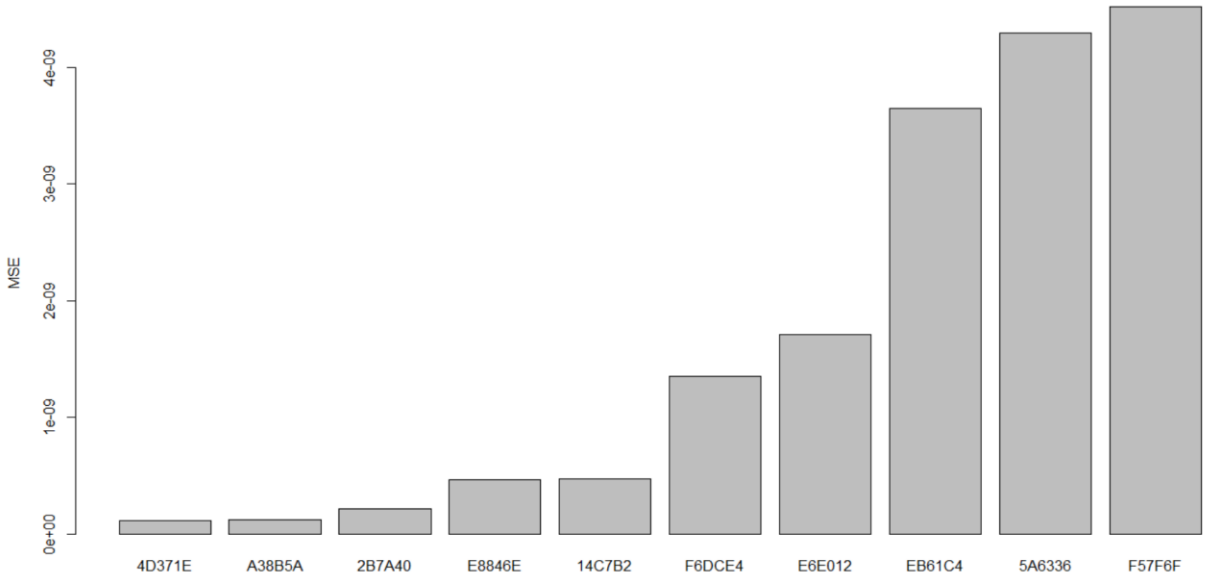

Figura 21 MSE de los 10 mejores modelos volátiles. Fuente: Elaboración propia

Al igual que ocurría en el MAE, los errores más pequeños en el MSE lo tienen las empresas ‘4D371E’ y ‘A38B5A’, teniendo un error de 1,1722 x 10^{-10} y 1,204 x 10^{-10} respectivamente.

A continuación, mostramos el peso que tiene cada una de las variables exógenas.

Tabla 3 Peso de las variables exógenas. Fuente: Elaboración propia

RP_ENTITY_ID	GLOBAL_ALL	GLOBAL_HEAD	GLOBAL_BODY	GLOBAL_ALL_SG90	GLOBAL_ALL_SG365	GROUP_A_BODY	GROUP_E_ALL	GROUP_E_BODY	GROUP_AM_BODY	T0_RETURN
4D371E	0	0	0	0	0	0	0	0	0	0
A38B5A	0	0	0	0	0	0	0	0	0	0
2B7A40	0	0	0	0	0	0	0	0	0	0
E8846E	0	0	0	0	0	0	0	0	0	0.289901466
14C7B2	0	0	0	0	0	0	0	0	0	0
F6DCE4	0	0	0	0	0	0	0	0	0	0.085816485
E6E012	0	0	0	0	0	0	0	0	0	0
EB61C4	0	0	0	0	0	0	0	0	0	0
5A6336	0	0	0	0	0	0	0	0	0	0.23573814
F57F6F	0	0	0	0	0	0	0	0	0	0

7 Conclusiones

El trabajo realizado es realmente útil para tener claro que pasos realizar a la hora de tratar una base de datos real.

El principal reto que me propuse a la hora de realizar este trabajo fue tratar una base de datos grande, para entender la metodología que se llevaba a cabo en el preprocesamiento masivo de datos. En concreto, esta base de datos tenía 1.764.824 observaciones y 42 variables.

La premisa que nos propusimos antes de tratar la base de datos objeto de estudio es que consiguiésemos unos datos de calidad e intentar eliminar la menor cantidad posible de variables, partiendo de una base de datos en el que el 66.60% de los datos eran datos faltantes. Para ello, nos pusimos un límite alto a partir del cual no imputásemos datos y eliminásemos variables, concretamente un 80%. Aun así, nos vimos obligados a eliminar 22 variables, pasando de las 42 variables que teníamos en un principio a 20. Las variables eliminadas fueron las siguientes:"GLOBAL_HEAD_SG365","GROUP_A_ALL_SG90","GROUP_A_HEAD","GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_HEAD",GROUP_E_HEAD_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365","GROUP_E_HEAD_SG365","GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM_ALL_SG90"

,"GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG90","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365", "GROUP_AM_BODY_SG365".

Una vez que habíamos eliminados estas variables, decidimos crear un dataset para cada empresa. Por tanto, pasamos de tener un dataset a tener 1756.

Decidimos imputar por el filtro de kalman, podríamos haber optado por interpolación o por la media. Decidimos optar por este método porque es uno de los métodos que mejor rendimiento esta dando en Indra. Este proceso duró días mientras el ordenador imputaba datos.Concretamente imputamos 13.372.790 millones de datos.

Otra decisión importante fue eliminar aquellos datasets que que no contenían información en el 2016 y 2017. Quedándonos con 1040 datasets en lugar de los 1756 que teníamos al principio.

Basándonos en el paper de (Hyndman, 2017) clasificamos las series en estacionales y volátiles. Concretamente, 17 de estas series eran estacionales y el resto volátiles.

Otro aspecto clave e importante es la normalización de los datos. Esta etapa es sencilla, pero no menos importante que las demás. Siempre hemos procurado tener especial cuidado en el tipo de datos que estabamos tratando.Es muy importante que cada columna sea de un único tipo de datos. Como podemos comprobar en nuestro caso se cumple, ya que solo tenemos dos variables que son factores y el resto que son numéricas.

El alcance de este proyecto no ha sido conseguir el mejor modelo, ya que hemos consideramos dos tipos de modelos autorregresivos como son el arima y la regresión lasso, sino construir un sistema de control integrado de predicciones en el que un dataset masivo de datos son tratados para discernir entre dos tipos de series temporales definidas a posteriori como predictivas o volátiles.

Por último, calculamos una medida del error mediante MAE y MSE para todas las series estacionales y para las diez volátiles con menor índice. En el que observamos que para las series volátiles el mejor modelo es ‘3100110’ y en el caso de las series volátiles la empresa que tenía menor ajuste del error era ‘4D371E’, por lo tanto son ambas de sus respectivos modelos las que estiman mejor.

Referencias

Robjhyndman.com. (2017). Detecting seasonality | Rob J Hyndman. [online] Available at: https://robjhyndman.com/hyndsight/detecting-seasonality/ [Accessed 25 Nov. 2017].

Sci2s.ugr.es. (2017). Citar un sitio web - Cite This For Me. [online] Available at: http://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2133_Nv237-Digital-sramirez.pdf [Accessed 25 Nov. 2017].

Otexts.org. (2017). 8.7 ARIMA modelling in R. [online] Available at: https://www.otexts.org/fpp/8/7 [Accessed 25 Nov. 2017].

Anon, (2017). [online] Available at: http://Hyndman, R.J., Koehler, A.B., Snyder, R.D., and Grose, S. (2002) "A state space framework for automatic forecasting using exponential smoothing methods", International J. Forecasting, 18(3), 439--454. [Accessed 25 Nov. 2017].

Robjhyndman.com. (2017). Citar un sitio web - Cite This For Me. [online] Available at: https://robjhyndman.com/papers/hksg.pdf [Accessed 25 Nov. 2017].

Rdocumentation.org. (2017). ets function | R Documentation. [online] Available at: https://www.rdocumentation.org/packages/forecast/versions/8.1/topics/ets [Accessed 25 Nov. 2017].

Stackoverflow.com. (2017). Stack Overflow - Where Developers Learn, Share, & Build Careers. [online] Available at: https://stackoverflow.com/ [Accessed 25 Nov. 2017].

Minsait. (2017). Data Scientists. [online] Available at: https://www.minsait.com/es/what-we-do/data-scientists [Accessed 25 Nov. 2017].

Anexo

```
#Trabajo Fin de Máster

##En este trabajo trataremos una base de datos de dos millones de elementos, la cúal prácticamente desconocemos que son cada una de sus variables. El objetivo principal, es dada cualquier base de datos (sea cual sea su procedencia) llevar a cabo un proceso de pre-procesing y unas serie de conclusiones.

#Configuración del entorno de trabajo

#	Instalación de paquetes

install.packages("imputeTS")

library(imputeTS)

library(forecast)

library(ggplot2)

#Carga de librerías (library)

library('dplyr')

library('data.table')

#Carga de los datos

fullPath = 'C:/Users/migue/Desktop/TFM ENTREGA/SampleDataSet.csv'

DataSet = data.table(read.csv(file = fullPath))

#PRE-PROCESSING

#Como podemos observar en la base de datos cargada tenemos 1764824 observaciones. A simple vista, observamos un número elevado de missing values.
```

```
#El próposito del preprocesamiento de datos es principalmente corregir las inconsistencias de los datos que serán la base del análisis que llevaremos a cabo.

#Datos incompletos conllevan a ruido y el ruido a incosistencias

#Ordenar las columnas de un dataframe por columnas

DataSet= DataSet[ , order(c(names(DataSet)))]

#Conversión a formato fecha

DataSet$DATE3 <- as.Date(DataSet$DATE, format="%Y-%m-%d")

#Extracción de los años

DataSet$ANYO <- format(DataSet$DATE3,"%Y")

DataSet$ANYO <- as.numeric(as.character(DataSet$ANYO))

#DataSet<-DataSet[,-c(41)]

#Extracción de los meses

DataSet$MES <- as.Date(DataSet$DATE, format="%Y-%m-%d")

DataSet$MES <- format(DataSet$MES,"%m")

DataSet$MES <- as.numeric(as.character(DataSet$MES))

DataSet<-DataSet[,-c(41)]

str(DataSet)

#Calcular el número de NA por años

missing_values_anyo<-
c(sum(is.na(a2005)=='TRUE'),(sum(is.na(a2006)=='TRUE')),(sum(is.na(a2007)=='TRUE')),(sum(is.na(a2008)=='TRUE')),(sum(is.na(a2009)=='TRUE')),(sum(is.na(a2010)=='TRUE')),(sum(is.na(a2011)=='TRUE')),(sum(is.na(a2012)=='TRUE')),(sum(is.na(a2013)=='TRUE')),(sum(is.na(a
```

```
2014)=='TRUE')),(sum(is.na(a2015)=='TRUE')),(sum(is.na(a2016)=='TRUE')),(sum(is.na(a2017
)=='TRUE')))

barplot(missing_values_anyo, font = 4, font.main = 1, ,ylab ="Valores perdidos", names.arg =
c("2005","2006","2007","2008","2009","2010","2011","2012","2013 ",

                                                    "2014", "2015","2016",


"2017"),col=c("darkblue","red","darkblue","red","darkblue","red","darkblue","red","darkblue","r
ed","darkblue","red","darkblue"))

##Como podemos observar tenemos un elevado número de missing en cada uno de los años
comprendidos entre 2005 y 2017.

sum(is.na(DataSet))

mean(is.na(DataSet))

NA_GLOBAL_ALL<-DataSet[,1:3]

NA_GLOBAL_HEAD<-DataSet[,c(1:2,4)]

NA_GLOBAL_BODY<-DataSet[,c(1:2,5)]

NA_GLOBAL_ALL_SG90<-DataSet[,c(1:2,6)]

NA_GLOBAL_HEAD_SG90<-DataSet[,c(1:2,7)]

NA_GLOBAL_BODY_SG90<-DataSet[,c(1:2,8)]

NA_GLOBAL_ALL_SG365<-DataSet[,c(1:2,9)]

NA_GLOBAL_HEAD_SG365<-DataSet[,c(1:2,10)]

NA_GLOBAL_BODY_SG365<-DataSet[,c(1:2,11)]

NA_GROUP_A_ALL<-DataSet[,c(1:2,12)]
```

```
NA_GROUP_A_HEAD<-DataSet[,c(1:2,13)]

NA_GROUP_A_BODY<-DataSet[,c(1:2,14)]

NA_GROUP_A_ALL_SG90<-DataSet[,c(1:2,15)]


NA_GROUP_A_HEAD_SG90<-DataSet[,c(1:2,16)]

NA_GROUP_A_BODY_SG90<-DataSet[,c(1:2,17)]

NA_GROUP_A_ALL_SG365<-DataSet[,c(1:2,18)]

NA_GROUP_A_HEAD_SG365<-DataSet[,c(1:2,19)]

NA_GROUP_A_BODY_SG365<-DataSet[,c(1:2,20)]

NA_GROUP_E_ALL<-DataSet[,c(1:2,21)]

NA_GROUP_E_HEAD<-DataSet[,c(1:2,22)]

NA_GROUP_E_BODY<-DataSet[,c(1:2,23)]

NA_GROUP_E_ALL_SG90<-DataSet[,c(1:2,24)]

NA_GROUP_E_HEAD_SG90<-DataSet[,c(1:2,25)]

NA_GROUP_E_BODY_SG90<-DataSet[,c(1:2,26)]

NA_GROUP_E_ALL_SG365<-DataSet[,c(1:2,27)]

NA_GROUP_E_HEAD_SG365<-DataSet[,c(1:2,28)]


NA_GROUP_E_BODY_SG365<-DataSet[,c(1:2,29)]

NA_GROUP_AM_ALL<-DataSet[,c(1:2,30)]
```

```
NA_GROUP_AM_HEAD<-DataSet[,c(1:2,31)]

NA_GROUP_AM_BODY<-DataSet[,c(1:2,32)]

NA_GROUP_AM_ALL_SG90<-DataSet[,c(1:2,33)]

NA_GROUP_AM_HEAD_SG90<-DataSet[,c(1:2,34)]

NA_GROUP_AM_BODY_SG90<-DataSet[,c(1:2,35)]

NA_GROUP_AM_ALL_SG365<-DataSet[,c(1:2,36)]

NA_GROUP_AM_HEAD_SG365<-DataSet[,c(1:2,37)]

NA_GROUP_AM_BODY_SG365<-DataSet[,c(1:2,38)]

NA_T0_RETURN<-DataSet[,c(1:2,39)]

NA_T1_RETURN<-DataSet[,c(1:2,40)]

#Representacion de los valores perdidos de las variables
:"GLOBAL_ALL","GLOBAL_HEAD","GLOBAL_BODY","GLOBAL_ALL_SG90","GLOBA
L_HEAD_SG90","GLOBAL_BODY_SG90","GLOBAL_ALL_SG365","GLOBAL_HEAD_SG
365","GLOBAL_BODY_SG365 ","GROUP_A_ALL",
"GROUP_A_HEAD","GROUP_A_BODY","GROUP_A_ALL_SG90"

#1-"GLOBAL_ALL".3

#2-"GLOBAL_HEAD".4

#3-"GLOBAL_BODY".5

#4-"GLOBAL_ALL_SG90".6

#5-"GLOBAL_HEAD_SG90".7

#6-"GLOBAL_BODY_SG90".8

#7-"GLOBAL_ALL_SG365".9
```

#8-"GLOB#AL_HEAD_SG365".10

#9-"GLOBAL_BODY_SG365 ".11

#10-"GROUP_A_ALL".12

#11-"GROUP_A_HEAD".13

#12-"GROUP_A_BODY".14

#13-"GROUP_A_ALL_SG90".15

```
missing_values_var_1<-
c(sum(is.na(NA_GLOBAL_ALL)=='TRUE'),(sum(is.na(NA_GLOBAL_HEAD)=='TRUE')),(sum(is.na(NA_GLOBAL_BODY)=='TRUE')),(sum(is.na(NA_GLOBAL_ALL_SG90)=='TRUE')),(sum(is.na(NA_GLOBAL_HEAD_SG90)=='TRUE')),(sum(is.na(NA_GLOBAL_BODY_SG90)=='TRUE')),(sum(is.na(NA_GLOBAL_ALL_SG365)=='TRUE')),(sum(is.na(NA_GLOBAL_HEAD_SG365)=='TRUE')),(sum(is.na(NA_GLOBAL_BODY_SG365)=='TRUE')),(sum(is.na(NA_GROUP_A_ALL)=='TRUE')),(sum(is.na(NA_GROUP_A_HEAD)=='TRUE')),(sum(is.na(NA_GROUP_A_BODY)=='TRUE')),(sum(is.na(NA_GROUP_A_ALL_SG90)=='TRUE')))

missing_values_var_1<-(missing_values_var_1*100)/1764824


barplot(missing_values_var_1, font = 4, font.main = 1, ylab ="% Valores perdidos", names.arg =
c("GLOBAL_ALL","GLOBAL_HEAD","GLOBAL_BODY","GLOBAL_ALL_SG90","GLOBAL_HEAD_SG90","GLOBAL_BODY_SG90","GLOBAL_ALL_SG365","GLOBAL_HEAD_SG365","GLOBAL_BODY_SG365 ",

                                                    "GROUP_A_ALL",
"GROUP_A_HEAD","GROUP_A_BODY",
```

"GROUP_A_ALL_SG90"),col=c("blue","blue","blue","blue","blue","blue","blue","red","blue","blue","red","blue","red"))

#Por tanto podemos observar que las columnas que están por encima de un 80% de missing values son: "GLOBAL_HEAD_SG365","GROUP_A_ALL_SG90","GROUP_A_HEAD"

#Representacion de los valores perdidos de las variables :"GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_ALL","GROUP_E_HEAD","GROUP_E_BODY","GROUP_E_ALL_SG90", "GROUP_E_HEAD_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365","GROUP_E_HEAD_SG365"

#1-"GROUP_A_HEAD_SG90".16

#2-"GROUP_A_BODY_SG90".17

#3-"GROUP_A_ALL_SG36".18

#4-"GROUP_A_HEAD_SG365".19

#5-"GROUP_A_BODY_SG365".20

#6-"GROUP_E_ALL".21

#7-"GROUP_E_HEAD".22

#8-"GROUP_E_BODY".23

#9-"GROUP_E_ALL_SG90".24

#10-"GROUP_E_HEAD_SG90".25

#11-"GROUP_E_BODY_SG90".26

#12-"GROUP_E_ALL_SG365".27

#13-"GROUP_E_HEAD_SG365".28

```
missing_values_var_2<-
c(sum(is.na(NA_GROUP_A_HEAD_SG90)=='TRUE'),(sum(is.na(NA_GROUP_A_BODY_SG
90)=='TRUE')),(sum(is.na(NA_GROUP_A_ALL_SG365)=='TRUE')),(sum(is.na(NA_GROUP_
A_HEAD_SG365)=='TRUE')),(sum(is.na(NA_GROUP_A_BODY_SG365)=='TRUE')),(sum(is.
na(NA_GROUP_E_ALL)=='TRUE')),(sum(is.na(NA_GROUP_E_HEAD)=='TRUE')),(sum(is.n
a(NA_GROUP_E_BODY)=='TRUE')),(sum(is.na(NA_GROUP_E_ALL_SG90)=='TRUE')),(su
m(is.na(NA_GROUP_E_HEAD_SG90)=='TRUE')),(sum(is.na(NA_GROUP_E_BODY_SG90)
=='TRUE')),(sum(is.na(NA_GROUP_E_ALL_SG365)=='TRUE')),(sum(is.na(NA_GROUP_E_
HEAD_SG365)=='TRUE')))

missing_values_var_2<-(missing_values_var_2*100)/1764824

barplot(missing_values_var_2, font = 4, font.main = 1,  ylab ="% Valores perdidos", names.arg
=
c("GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GROUP
_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_ALL","GROUP_E_HEAD","G
ROUP_E_BODY","GROUP_E_ALL_SG90",

                                                    "GROUP_E_HEAD_SG90",
"GROUP_E_BODY_SG90","GROUP_E_ALL_SG365",


"GROUP_E_HEAD_SG365"),col=c("red","red","red","red","red","blue","red","blue","blue","re
d","red","red","red"))


#Por tanto podemos observar que las columnas que están por encima de un 80% de missing
values
son:"GROUP_A_HEAD_SG90","GROUP_A_BODY_SG90","GROUP_A_ALL_SG36","GRO
UP_A_HEAD_SG365","GROUP_A_BODY_SG365","GROUP_E_HEAD",GROUP_E_HEAD
_SG90", "GROUP_E_BODY_SG90","GROUP_E_ALL_SG365",
"GROUP_E_HEAD_SG365".
```

```
#Representacion de los valores perdidos de las
variables:"GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP
_AM_BODY","GROUP_AM_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BO
DY_SG90","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365",
"GROUP_AM_BODY_SG365", "T0_RETURN","T1_RETURN"

#1-"GROUP_E_BODY_SG365".29

#2-"GROUP_AM_ALL".30

#3-"GROUP_AM_HEAD".31

#4-"GROUP_AM_BODY".32

#5-"GROUP_AM_ALL_SG90".33

#6-"GROUP_AM_HEAD_SG90".34

#7-"GROUP_AM_BODY_SG90".35

#8-"GROUP_AM_ALL_SG365".36

#9-"GROUP_AM_HEAD_SG365".37

#10-"GROUP_AM_BODY_SG365".38

#11-"T0_RETURN".39

#12-"T1_RETURN".40

missing_values_var_3<-
c(sum(is.na(NA_GROUP_E_BODY_SG365)=='TRUE'),(sum(is.na(NA_GROUP_AM_ALL)==
'TRUE')),(sum(is.na(NA_GROUP_AM_HEAD)=='TRUE')),(sum(is.na(NA_GROUP_AM_BO
DY)=='TRUE')),(sum(is.na(NA_GROUP_AM_ALL_SG90)=='TRUE')),(sum(is.na(NA_GROU
P_AM_HEAD_SG90)=='TRUE')),(sum(is.na(NA_GROUP_AM_BODY_SG90)=='TRUE')),(su
m(is.na(NA_GROUP_AM_ALL_SG365)=='TRUE')),(sum(is.na(NA_GROUP_AM_HEAD_SG
```

```
365)=='TRUE')),(sum(is.na(NA_GROUP_AM_BODY_SG365)=='TRUE')),(sum(is.na(NA_T0_
RETURN)=='TRUE')),(sum(is.na(NA_T1_RETURN)=='TRUE')))

missing_values_var_3<-(missing_values_var_3*100)/1764824

barplot(missing_values_var_3, font = 4, font.main = 1, ylab ="% Valores perdidos", names.arg =
c("GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM_B
ODY","GROUP_AM_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG9
0","GROUP_AM_ALL_SG365","GROUP_AM_HEAD_SG365",
"GROUP_AM_BODY_SG365",
"T0_RETURN","T1_RETURN"),col=c("red","red","red","blue","red","red","red","red","red","re
d","red","blue","blue"))

#Por tanto podemos observar que las columnas que están por encima de un 80% de missing
values
son:"GROUP_E_BODY_SG365","GROUP_AM_ALL","GROUP_AM_HEAD","GROUP_AM
_ALL_SG90","GROUP_AM_HEAD_SG90","GROUP_AM_BODY_SG90","GROUP_AM_AL
L_SG365","GROUP_AM_HEAD_SG365", "GROUP_AM_BODY_SG365"

#Un resumen del porcentaje de missing values por columnas seria el siguiente:

sapply(DataSet, function(x) ((sum(is.na(x)))*100/length(x)))

#Medias de NA en el DataSet

mean(is.na(DataSet))

#Elimino todas aquellas columnas con un 80% de missing values

New_DataSet<-DataSet[, -which(colMeans(is.na(DataSet)) > 0.8)]

#Elimino los duplicados de la base de datos en el caso de que existan

New_DataSet[!duplicated(New_DataSet), ]

View(New_DataSet)
```

```
#Como puedo observar no existen duplicados, por lo tanto mantengo el numero de observaciones

str(New_DataSet)

#El siguiente paso que realizo es crear una lista que contenga un dataSet para cada una de las empresas que tenemos en la base de datos.

nombres_empresa<-list()

for (id_empresa in unique(New_DataSet$RP_ENTITY_ID)){nombres_empresa[[length(nombres_empresa)+1]]<-subset(New_DataSet,RP_ENTITY_ID==id_empresa)}

#Aplico el filtro de kalman a cada uno de los datasets de las diferentes empresas

data_frame_kalman<-c()

for (i in (1:1768)){x1<-nombres_empresa[[i]];data_frame_kalman[[length(data_frame_kalman)+1]] <-  na.kalman(x1)}

#Para cada dataset de cada empresa, realizamos la media de aquellas variables que coinciden en el mismo dia. Quedandonos por tanto, con una unica observacion por dia

lista_dataframes<-list()

for (i in (1:1768)){x<-data_frame_kalman[[i]];lista_dataframes[[length(lista_dataframes)+1]] <-x %>%
select("DATE","RP_ENTITY_ID","MES","ANYO","GLOBAL_ALL","GLOBAL_HEAD","GLOBAL_BODY","GLOBAL_ALL_SG90","GLOBAL_HEAD_SG90","GLOBAL_BODY_SG90","GLOBAL_ALL_SG365","GLOBAL_BODY_SG365","GROUP_A_ALL","GROUP_A_BODY","GROUP_E_ALL","GROUP_E_BODY","GROUP_E_ALL_SG90","GROUP_AM_BODY","T0_RETURN","T1_RETURN") %>% group_by(DATE)%>% summarise(ANYO2 = mean(ANYO,na.rm=T),ANYO2 = mean(ANYO,na.rm=T),GLOBAL_ALL2 = mean(GLOBAL_ALL,na.rm=T),GLOBAL_HEAD2 = mean(GLOBAL_HEAD,na.rm=T),GLOBAL_BODY2 = mean(GLOBAL_BODY,na.rm=T),GLOBAL_ALL_SG902 =
```

```
mean(GLOBAL_ALL_SG90,na.rm=T),GLOBAL_HEAD_SG902 =
mean(GLOBAL_HEAD_SG90,na.rm=T),GLOBAL_BODY_SG902 =
mean(GLOBAL_BODY_SG90,na.rm=T),GLOBAL_ALL_SG3652 =
mean(GLOBAL_ALL_SG365,na.rm=T),GLOBAL_BODY_SG3652 =
mean(GLOBAL_BODY_SG365,na.rm=T),GROUP_A_ALL2 =
mean(GROUP_A_ALL,na.rm=T),GROUP_A_BODY2 =
mean(GROUP_A_BODY,na.rm=T),GROUP_E_ALL2 =
mean(GROUP_E_ALL,na.rm=T),GROUP_E_BODY2 =
mean(GROUP_E_BODY,na.rm=T),GROUP_E_ALL_SG902 =
mean(GROUP_E_ALL_SG90,na.rm=T),GROUP_AM_BODY2 =
mean(GROUP_AM_BODY,na.rm=T),T0_RETURN2 =
mean(T0_RETURN,na.rm=T),T1_RETURN2 = mean(T1_RETURN,na.rm=T),MES2 =
mean(MES,na.rm=T))}

#Me interesan aquellas series que tienen datos entre Diciembre del 2016 y Febrero del 2017. Es decir, si de una serie solo tengo datos hasta 2014 no me interesa.

#Calculo el máximo año de cada dataset

almacenamiento_anyo_max<-c()

for (i in (1:1768)){data3<-
lista_dataframes[[i]];almacenamiento_anyo_max[[length(almacenamiento_anyo_max)+1]] <-
as.integer(max(data3$ANYO2))}

#Calculo el mes máximo de cada dataset

almacenamiento_mes<-c()

for (i in (1:1768)){if (almacenamiento_anyo_max[i]>=2016){data2<-
lista_dataframes[[i]];almacenamiento_mes[[length(almacenamiento_mes)+1]] <-
as.integer(max(data2$MES2))} else {data2<-
lista_dataframes[[i]];almacenamiento_mes[[length(almacenamiento_mes)+1]] <-
as.integer(min(data2$MES2))}}
```

```
#Nombre de todos los dataset

almacenamiento_ID<-c()

for (id_empresa in
unique(DataSet$RP_ENTITY_ID)){almacenamiento_ID[[length(almacenamiento_ID)+1]]<-
id_empresa}

almacenamiento_ID_CONDICION<-c()

for (i in (1:1768)){if ((almacenamiento_anyo_max[i]>=2016) &
(almacenamiento_mes[i]==12)){almacenamiento_ID_CONDICION[[length(almacenamiento_I
D_CONDICION)+1]]<-almacenamiento_ID[i]}}

#Aquí aplico la condicion de Diciembre 2016 a Febrero 2017

dataframes_cond<-list()

for (i in (1:1768)){if ((almacenamiento_anyo_max[i]>=2016) & (almacenamiento_mes[i]==12))
{y<-lista_dataframes[[i]];dataframes_cond[[length(dataframes_cond)+1]] <-  y %>%
select("DATE","MES2","ANYO2","GLOBAL_ALL2","GLOBAL_HEAD2","GLOBAL_BOD
Y2","GLOBAL_ALL_SG902","GLOBAL_ALL_SG3652","GROUP_A_BODY2","GROUP_E_
ALL2","GROUP_E_BODY2","GROUP_AM_BODY2","T0_RETURN2","T1_RETURN2")
%>% group_by(DATE)%>% summarise(ANYO2 = mean(ANYO2,na.rm=T),GLOBAL_ALL2
= mean(GLOBAL_ALL2,na.rm=T),GLOBAL_HEAD2 =
mean(GLOBAL_HEAD2,na.rm=T),GLOBAL_BODY2 =
mean(GLOBAL_BODY2,na.rm=T),GLOBAL_ALL_SG902 =
mean(GLOBAL_ALL_SG902,na.rm=T),GLOBAL_ALL_SG3652 =
mean(GLOBAL_ALL_SG3652,na.rm=T),GROUP_A_BODY2 =
mean(GROUP_A_BODY2,na.rm=T),GROUP_E_ALL2 =
mean(GROUP_E_ALL2,na.rm=T),GROUP_E_BODY2 =
mean(GROUP_E_BODY2,na.rm=T),GROUP_AM_BODY2 =
mean(GROUP_AM_BODY2,na.rm=T),T0_RETURN2 =
mean(T0_RETURN2,na.rm=T),T1_RETURN2 = mean(T1_RETURN2,na.rm=T),MES2 =
mean(MES2,na.rm=T))}}
```

```
#Agrupo por mes y año, para tener una unica observacion por mes.

DataSet_final<-list()

for (i in (1:1056)){x<-dataframes_cond[[i]];DataSet_final[[length(DataSet_final)+1]] <-  x %>%
select("DATE","MES2","ANYO2","GLOBAL_ALL2","GLOBAL_HEAD2","GLOBAL_BOD
Y2","GLOBAL_ALL_SG902","GLOBAL_ALL_SG3652","GROUP_A_BODY2","GROUP_E_
ALL2","GROUP_E_BODY2","GROUP_AM_BODY2","T0_RETURN2","T1_RETURN2")
%>% group_by(ANYO2,MES2)%>% summarise(GLOBAL_ALL2 =
mean(GLOBAL_ALL2,na.rm=T),GLOBAL_HEAD2 =
mean(GLOBAL_HEAD2,na.rm=T),GLOBAL_BODY2 =
mean(GLOBAL_BODY2,na.rm=T),GLOBAL_ALL_SG902 =
mean(GLOBAL_ALL_SG902,na.rm=T),GLOBAL_ALL_SG3652 =
mean(GLOBAL_ALL_SG3652,na.rm=T),GROUP_A_BODY2 =
mean(GROUP_A_BODY2,na.rm=T),GROUP_E_ALL2 =
mean(GROUP_E_ALL2,na.rm=T),GROUP_E_BODY2 =
mean(GROUP_E_BODY2,na.rm=T),GROUP_AM_BODY2 =
mean(GROUP_AM_BODY2,na.rm=T),T0_RETURN2 =
mean(T0_RETURN2,na.rm=T),T1_RETURN2 = mean(T1_RETURN2,na.rm=T))}

#Funcion que calcula las series temporales mediante índice de predictibilidad

seas_measure <- function(time_series, cutoff=0.05){

  library(fma)

  fit1 <- ets(time_series)

  fit2 <- ets(time_series,model="ANN")

  ans <- c()

  ans$deviance <- 2*c(logLik(fit1) - logLik(fit2))

  ans$df <- attributes(logLik(fit1))$df - attributes(logLik(fit2))$df
```

```
 #P value
 ans$p_value<- 1-pchisq(ans$deviance,ans$df)
 ans$reject_null_hyp <- ans$p_value<=cutoff # True = rechazamos, False = no rechazamos
 ans
}
#Calulo en a traves de la funcion si es TRUE o False
#Creo una lista que contendra valores TRUE o FALSE. TRUE si es una serie estacional y FALSE si es volatil
estudio_t_f<-list()
for (i in (1:1040)){x<-DataSet_final[[i]];estudio_t_f[[length(estudio_t_f)+1]]<-seas_measure(x$T1_RETURN2) }
dataframe_true_false<-list()
for (i in (1:1040)){x1<-estudio_t_f[[i]];dataframe_true_false[[length(dataframe_true_false)+1]]<-(x1$reject_null_hy) }
#Dependiendo si el elemento de la lista dataframe_true_false es TRUE o FALSE se aplicara un auto.arima() o un cv.lasso1. Tanto a uno como otro se almacenara en un vector con respectivo MAE,MSE
require(forecast)

library(glmnet)
alma_ID_False<-list()
vectores=list()
```

```
for (i in (1:length(dataframe_true_false))){

  if (dataframe_true_false[[i]]){

    a<-DataSet_final[[i]]$T1_RETURN2 # a es T1_return2 del dataset por empresa que se recorre en el bucle

    train=a[1:(length(a)-1)]

    test=a[length(a)]

    model=auto.arima(train)

    predicted= predict(model,h=1)$pred


    ID<-almacenamiento_ID_CONDICION[[i]]

    MAE = Metrics::mae(test,predicted)

    MSE = Metrics::mse(test,predicted)

    vectores[[length(vectores)+1]]<-list(ID,TRUE,MAE,MSE,model$arma)

  }else {

    a<-DataSet_final[[i]] # a es T1_return2 del dataset por empresa que se recorre en el bucle,calcula la parte de la volatilidad

    a<-a[,(3:length(a))]


    y.1 <- a$T1_RETURN2[1:(length(a$T1_RETURN2)-1)]

    y.2 <- a$T1_RETURN2[length(a$T1_RETURN2)]

    x.1 <- as.matrix(a[1:(length(a$T1_RETURN2)-1),c(1:10)])
```

```
    x.2 <- as.matrix(a[(length(a$T1_RETURN2)),c(1:10)])


    lasso.1 <- glmnet(y=y.1, x= x.1, family="gaussian")
    cv.lasso.1 <- cv.glmnet(y=y.1, x= x.1, family="gaussian")
    predict.1.1 <- predict(cv.lasso.1, newx=x.1)
    predict.1.2 <- predict(cv.lasso.1, newx=x.2)
    #MSPR.lasso <- mean((y.2 - predict.1.2)^2)


    ID<-almacenamiento_ID_CONDICION[[i]]
    alma_ID_False[[length(alma_ID_False)+1]]<-ID
    MAE = Metrics::mae(y.2 ,predict.1.2)
    MSE = Metrics::mse(y.2,predict.1.2)
    vectores[[length(vectores)+1]]<-list(ID,FALSE,MAE,MSE,coef(cv.lasso.1))


  }
}
#Si el elemto del vector es TRUE, represento la frecuencia en la que se repite cada modelo
x1<-list()
for (i in (1:length(dataframe_true_false))){
  if (dataframe_true_false[[i]]){x1[[length(x1)+1]]<-
do.call(paste,c(as.list(as.character(vectores[[i]][[5]]), sep = "")))}}
```

```
x1_factor<-as.factor(unlist(x1))

hist(table(x1_factor), freq=TRUE, xlab = levels(x1_factor), ylab = "Frequencies")

barplot(table(x1_factor))

barplot(prop.table(table(x1_factor)))

#Almacenamiento de las series volatiles y las estacionales con sus respectivos valores de MAE y MSE

vector_true=c()

for ( i in vectores){

  if(i[[2]]){

    mo <- do.call(paste,c(as.list(as.character(i[[5]]), sep = "")))

    vector_true<- rbind(vector_true,c(mo,i[[3]],i[[4]]))



  }

}

vector_true=as.data.frame(vector_true)

vector_true$V1<-as.character(vector_true$V1)

vector_true$V2<-as.numeric(as.character(vector_true$V2))

vector_true$V3<-as.numeric(as.character(vector_true$V3))



vector_false=c()

for ( i in vectores){
```

```
 if(i[[2]]==FALSE){


   vector_false<- rbind(vector_false,c(i[[3]],i[[4]]))

 }

}

vector_false=as.data.frame(vector_false)

vector_false$V1<-as.numeric(as.character(vector_false$V1))

vector_false$V2<-as.numeric(as.character(vector_false$V2))

#Para cada modelo obtengo la media de su MAE y MSE (es decir de aquellos modelos que surgan en numerosas ocasiones)

almacen_true<-list()

for (id_empresa in unique(l$V1)){almacen_true[[length(almacen_true)+1]]<-subset(l,l$V1==id_empresa)}

resultados<-list()

for (i in (1:length(almacen_true))){x<-almacen_true[[i]];resultados[[length(resultados)+1]] <- x %>%select("V1","V2","V3")%>% group_by(x$V1)%>% summarise(V2 = mean(x$V2,na.rm=T),V3 = mean(x$V3,na.rm=T))}

df_df_true<-data.frame(matrix(unlist(resultados),nrow = 10,byrow = T))

df_true$X2<-as.numeric(as.character(df_true$X2))

df_true$X3<-as.numeric(as.character(df_true$X3))

df_true<-data.frame(df_true)

write.table(df_true,file = "tavol.csv",sep = ";")
```

```
#Representacion del MAE para series estacionarias
ggplot(df_true, aes(x=X1, y=X2)) + geom_bar(stat="identity") +
  labs(x="Modelos", y="MAE")
#Representacion del MSE para series estacionarias
ggplot(df_true, aes(x=X1, y=X3)) + geom_bar(stat="identity") +
  labs(x="Modelos", y="MSE")
#
x111<-list()
x222<-list()
x333<-list()
x444<-list()
x555<-list()
x666<-list()
x777<-list()
x888<-list()
x999<-list()
x101<-list()
x201<-list()
for (i in (1:length(dataframe_true_false))){
  if (dataframe_true_false[[i]]==FALSE){x111[[length(x111)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][1])));x222[[length(x222)+1]]<-
```

```
c(as.list(as.character(vectores[[i]][[5]][2])));x333[[length(x333)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][3])));x444[[length(x444)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][4])));x555[[length(x555)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][5])));x666[[length(x666)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][6])));x777[[length(x777)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][7])));x888[[length(x888)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][8])));x999[[length(x999)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][9])));x101[[length(x101)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][10])));x201[[length(x201)+1]]<-
c(as.list(as.character(vectores[[i]][[5]][11])))}}

x1_factor<-as.factor(unlist(x111))

x2_factor<-as.factor(unlist(x222))

x3_factor<-as.factor(unlist(x333))

x4_factor<-as.factor(unlist(x444))

x5_factor<-as.factor(unlist(x555))

x6_factor<-as.factor(unlist(x666))

x7_factor<-as.factor(unlist(x777))

x8_factor<-as.factor(unlist(x888))

x9_factor<-as.factor(unlist(x999))

x10_factor<-as.factor(unlist(x101))

x11_factor<-as.factor(unlist(x201))

int<-
data.frame(x1_factor,x2_factor,x3_factor,x4_factor,x5_factor,x6_factor,x7_factor,x8_factor,x9_
factor,x10_factor,x11_factor)
```

```
#Representacion del MAE para series volatiles

vector_false$V1<-as.numeric(as.character(vector_false$V1))

vector_false$V2<-as.numeric(as.character(vector_false$V2))

vector_false$names=alma_ID_False

vector_false<-data.frame(vector_false)

newdata <- vector_false[order(vector_false$V1),]

barplot(newdata$V1[1:10], names.arg=newdata$lol[1:10],ylab ="MAE")

#Representacion del MAE para series volatiles

barplot(newdata$V2[1:10], names.arg=newdata$lol[1:10],ylab ="MSE")
```