



Universidad  
Internacional  
Menéndez Pelayo

## MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL

— TRABAJO SERIES TEMPORALES —

### *Datos temporales y complejos*

---

Laura Rodríguez Navas

Diciembre de 2020

[rodrigueznavas@posgrado.uimp.es](mailto:rodrigueznavas@posgrado.uimp.es)

# Índice general

<b>1. Análisis de la serie temporal</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Datos de la serie temporal . . . . .	2
1.3. Visualización de la serie temporal . . . . .	3
1.4. Componentes de la serie temporal . . . . .	5
<b>2. WORKING HERE</b>	<b>8</b>
<b>3. Búsqueda de outliers</b>	<b>9</b>
3.1. Reemplazo de outliers . . . . .	11
3.2. INVENT . . . . .	13
<b>4. Método de predicción</b>	<b>17</b>
<b>5. Aplicación del método de predicción</b>	<b>18</b>
<b>Bibliografía</b>	<b>19</b>

# Capítulo 1

## Análisis de la serie temporal

### 1.1. Introducción

Una empresa tecnológica cuya área de negocio es la inteligencia artificial es contratada por una empresa eléctrica para que diseñe un sistema de recomendación que haga ofertas personalizadas a sus clientes sobre los paquetes energéticos más adecuados a sus consumos. Para ello, la empresa primero debe llevar a cabo un análisis exhaustivo de los consumos energéticos y diseñar una técnica de predicción que sea capaz de predecir dichos consumos con un horizonte temporal dado. La empresa eléctrica suministra a la empresa tecnológica para dicho cometido los consumos eléctricos desde el 01 de enero de 2015 hasta el 31 de diciembre de 2015 medidos con una frecuencia temporal de 10 minutos.

Se pide:

1. Analizar la serie temporal de consumos eléctricos y describir brevemente las principales características de esta.
2. Realizar un estudio para determinar si la serie temporal presenta *outliers*. Describir brevemente el estudio realizado y las conclusiones alcanzadas.
3. Seleccionar un artículo publicado en una revista científica de prestigio internacional en el que se presente un método de predicción. Estudiar de manera detallada el método publicado y describirlo de forma resumida.
4. Aplicar el método seleccionado en el apartado anterior para obtener una predicción del consumo eléctrico con un horizonte temporal de 4 horas para los días de la semana desde el lunes 8 de junio hasta el domingo 14 de junio. Para ello, se debe implementar un método de predicción en el lenguaje de programación que se estime oportuno o bien usar software libre disponible como WEKA, R, KEEL, ... Una vez obtenidas las predicciones del periodo indicado, visualizar los resultados (predicciones, errores, ...).

El trabajo de investigación se realiza utilizando el lenguaje de programación R y todos los datos y el código necesarios para reproducir el estudio se proporcionan dentro de este documento para garantizar que éste sea completamente reproducible.

Se utilizan los siguientes paquetes para el análisis:

```
library(tidyverse)
library(lubridate)
library(xts)
library(forecast)
```

- `tidyverse` es un conjunto de librerías de R diseñadas para la “*Ciencia de datos*”.
- `lubridate` es una librería que permite manipular fechas e intervalos de tiempo.
- `xts` es una librería que permite convertir datos en series temporales.

## 1.2. Datos de la serie temporal

Los datos están disponibles para descargar [aquí](#). Una vez descargados, se cargan en el espacio de trabajo.

```
data <- read.csv("Demanda_2015.csv", header = FALSE, sep = ",")
```

Buscamos valores perdidos.

```
sum(is.na(data))
```

```
## [1] 0
```

Vemos que no existen valores perdidos, por lo tanto no tendremos que eliminarlos.

```
# Establecemos nombres a las columnas
colnames(data) <- c("date", "time", "demand")
head(data, 10)
```

```
##           date time demand
## 1 01/01/2015 0:00 25459
## 2 01/01/2015 0:10 25591
## 3 01/01/2015 0:20 25531
## 4 01/01/2015 0:30 25453
## 5 01/01/2015 0:40 25329
## 6 01/01/2015 0:50 25247
## 7 01/01/2015 1:00 25093
## 8 01/01/2015 1:10 24853
## 9 01/01/2015 1:20 24678
## 10 01/01/2015 1:30 24391
```

```
summary(data)
```

```
##           date              time          demand
##  Length:52560    Length:52560     Min.   :17985
##  Class :character Class :character  1st Qu.:24392
##  Mode   :character Mode   :character Median :28566
##                                         Mean   :28349
##                                         3rd Qu.:31664
##                                         Max.   :40648
```

Tenemos la variable *date* como *character*, la variable *time* como *character* y la variable *demand* como *integer*. Para su análisis posterior, el marco de datos de fechas y horas se formatea. Para ello, primero crearemos una nueva columna *datetime*, donde juntaremos las fechas con las horas usando la función *cbind()*. Luego procederemos a la conversión de tipos *character* a *POSIXct* para la fecha-hora usando la función *parse\_date\_time()*.

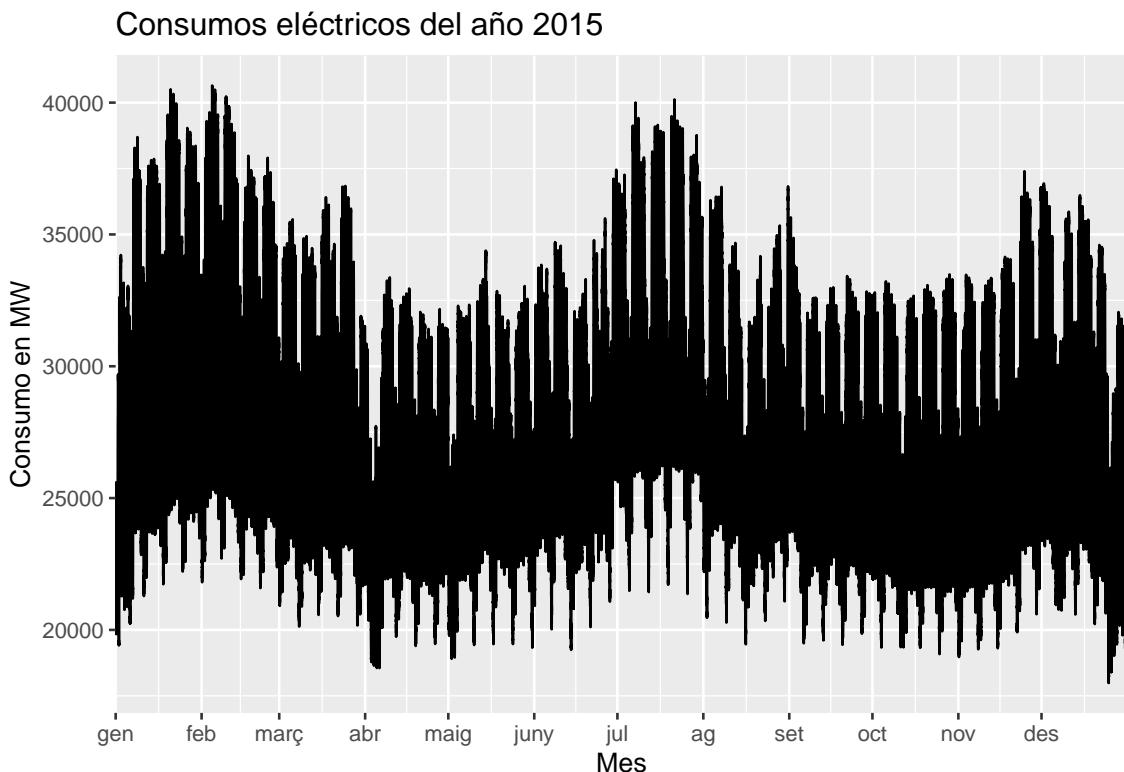
```
data <- cbind(datetime = paste(data$date, data$time), data)
data$datetime <-
  parse_date_time(data$datetime, "dmY HMS", truncated = 3, tz = "UTC")
head(data, 10)
```

	datetime	date	time	demand
## 1	2015-01-01 00:00:00	01/01/2015	0:00	25459
## 2	2015-01-01 00:10:00	01/01/2015	0:10	25591
## 3	2015-01-01 00:20:00	01/01/2015	0:20	25531
## 4	2015-01-01 00:30:00	01/01/2015	0:30	25453
## 5	2015-01-01 00:40:00	01/01/2015	0:40	25329
## 6	2015-01-01 00:50:00	01/01/2015	0:50	25247
## 7	2015-01-01 01:00:00	01/01/2015	1:00	25093
## 8	2015-01-01 01:10:00	01/01/2015	1:10	24853
## 9	2015-01-01 01:20:00	01/01/2015	1:20	24678
## 10	2015-01-01 01:30:00	01/01/2015	1:30	24391

### 1.3. Visualización de la serie temporal

La siguiente figura muestra la distribución del consumo eléctrico a lo largo del 2015.

```
ggplot(data = data, aes(x = datetime, y = demand)) +
  geom_line() +
  scale_x_datetime(date_labels = "%b",
                   breaks = "1 month",
                   expand = c(0, 0)) +
  ggtitle("Consumos eléctricos del año 2015") +
  xlab("Mes") +
  ylab("Consumo en MW")
```



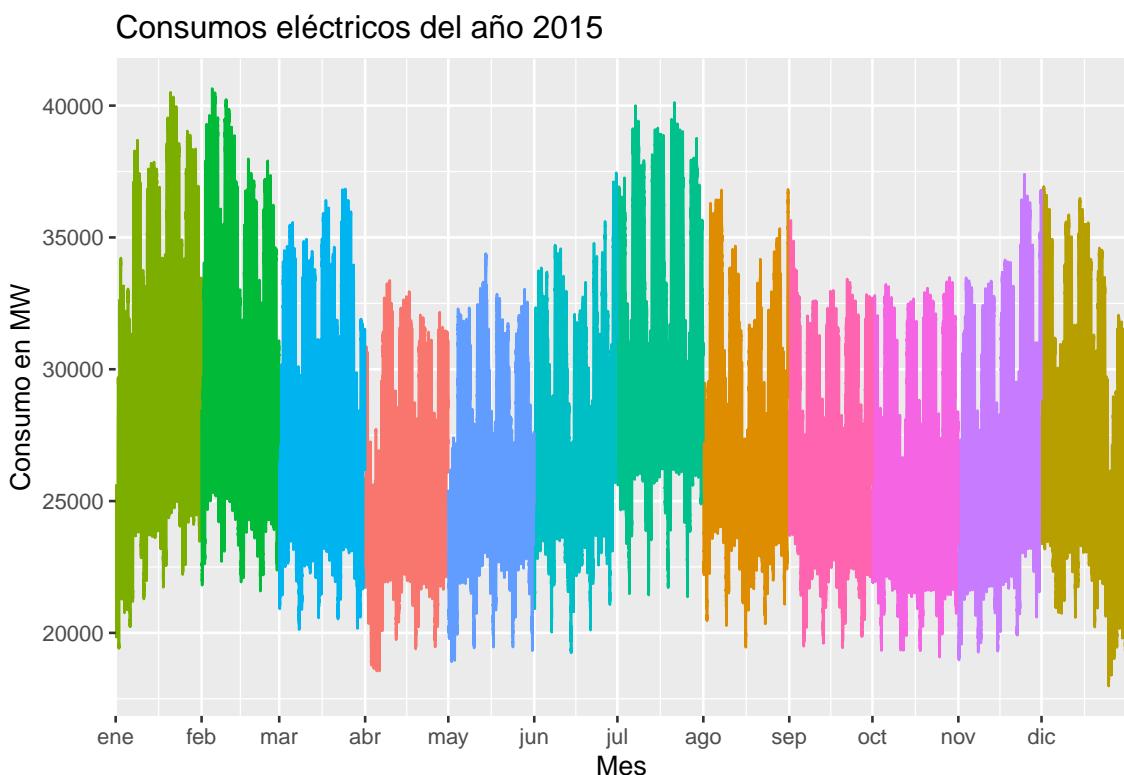
Observando la figura se puede detectar una dependencia estacional del consumo eléctrico, aunque pueden existir otros factores que pueden afectarla, como los días festivos, los fines de semana, ...

Parece que el principal desafío introducido por la serie temporal podría ser su estacionalidad múltiple. En ese caso, las siguientes estacionalidades pueden estar presentes:

- Las personas usan la mayoría de sus electrodomésticos en determinadas horas del día, lo que resulta en una estacionalidad diaria.
- Las personas usan la electricidad de manera diferente los días entre semana y los fines de semana, lo que resulta en una estacionalidad semanal.
- Las personas usan la electricidad de manera diferente en diferentes épocas del año, lo que resulta en una estacionalidad anual.

Para mejorar la visualización de la distribución del consumo eléctrico a lo largo del 2015, la siguiente figura es muy útil.

```
ggplot(data, aes(x = datetime, y = demand)) +
  geom_line(aes(colour = month)) +
  scale_x_datetime(date_labels = "%b",
                   breaks = "1 month",
                   expand = c(0, 0)) +
  ggtitle("Consumos eléctricos del año 2015") +
  xlab("Mes") +
  ylab("Consumo en MW") +
  theme(legend.position = "none")
```



Es importante tener en cuenta cómo aumenta el consumo de la electricidad tanto en los meses de verano, de mayo a septiembre, como en los días de invierno, desde noviembre hasta febrero. Esto se debe a que las personas usan la electricidad para enfriar y calentar sus viviendas. Durante los períodos de invierno y de verano, el consumo eléctrico es de manera clara mayor salvo, probablemente, en los períodos vacacionales. Como se puede observar en el mes de agosto. En resumen, visualmente podemos deducir que los consumos eléctricos siguen un patrón de subidas y bajadas en una tendencia que varía de manera estacional.

## 1.4. Componentes de la serie temporal

Es frecuente analizar las series temporales des del punto de vista de sus componentes estructurales. Pero primero necesitamos que R trate los datos como una serie temporal, así que para ello tendremos que determinar apropiadamente sus características con la función `ts()`. Para definir la serie correctamente escribimos:

```
ts <- ts(data$demand, frequency = 24 * 60 / 10)
# El argumento frequency de la función *ts()* se utiliza para indicar
# la periodicidad de la serie (en este caso es de 10 minutos)
```

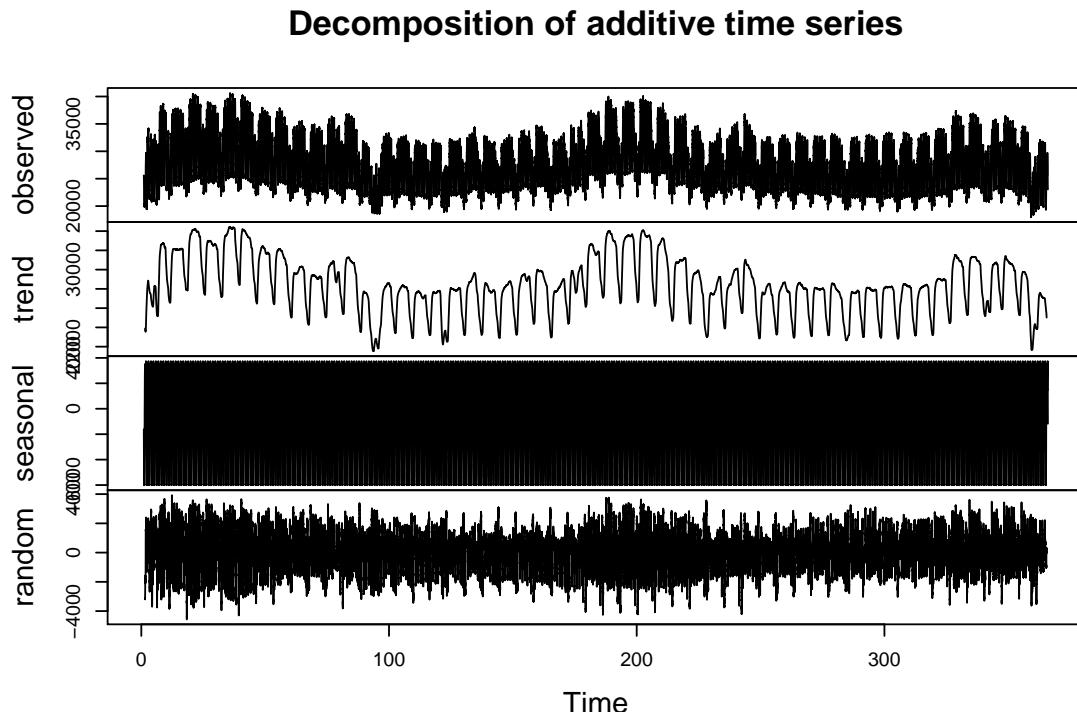
Los componentes de una serie temporal son:

- Tendencia: un aumento o disminución a largo plazo de los datos se denomina tendencia. No tiene por qué ser necesariamente lineal. Es el patrón subyacente en los datos a lo largo del tiempo.
- Estacional o Periódico: cuando una serie está influenciada por factores estacionales. Como en este caso y hemos podido visualizar en el apartado anterior.

- Cíclico: cuando los datos muestran subidas y bajadas que no son de un mismo período.

En R se pueden averiguar los componentes de la serie temporal con la siguiente función:

```
componentes.ts <- decompose(ts)
plot(componentes.ts)
```

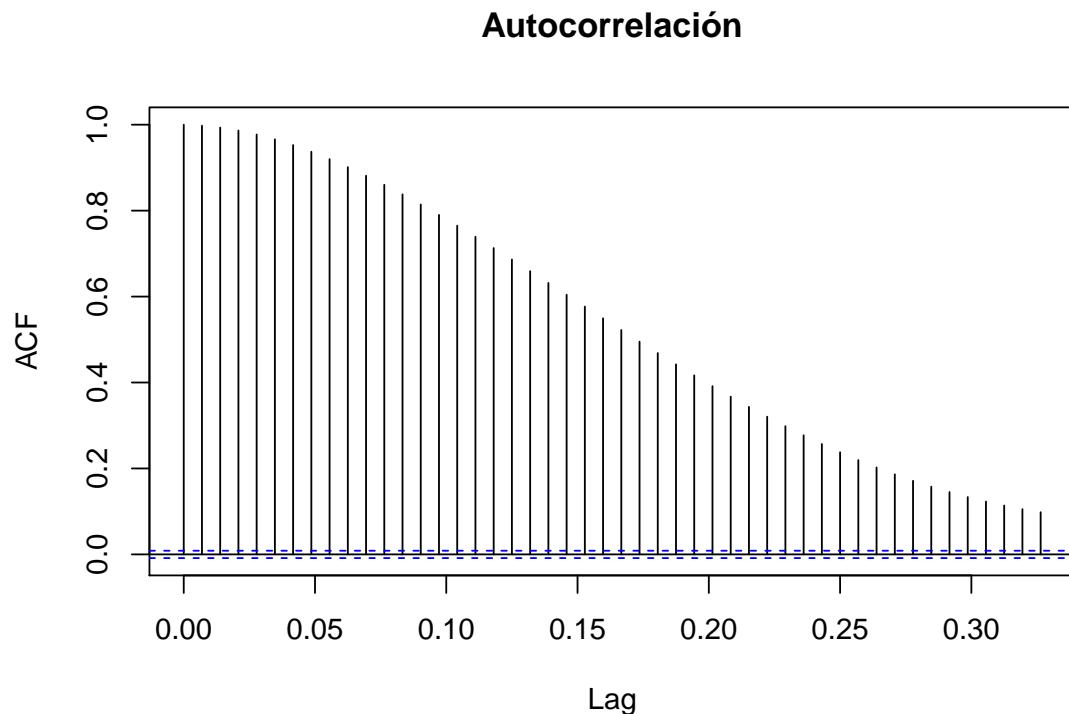


En la figura se observan cuatro sub-figuras:

- observed: los datos actuales.
- trend: el movimiento general hacia arriba o hacia abajo de los datos.
- seasonal: cualquier patrón mensual o anual de los datos.
- random: parte inexplicable de los datos.

A partir de las sub-figuras se puede determinar que la serie temporal no presenta una tendencia. Con esto se puede afirmar que la serie es estacionaria. También se puede observar que la serie es estacionaria en cuanto a la varianza, ya que no se aprecia gran variabilidad. En cambio, sí que presenta una estacionalidad muy marcada (el consumo aumenta durante los meses de invierno y verano). Por ese motivo, para mejorar las futuras predicciones se debe eliminar la estacionalidad de la serie temporal original. Pero antes, para confirmar que la serie es estacionaria trazaremos la gráfica de su función de autocorrelación, usando la función `acf()`.

```
acf(ts, main = "Autocorrelación")
```



Se puede confirmar que la serie es estacionaria, ya que el valor de la función de autocorrelación decae de manera exponencial a medida que aumentan los rezagos en el tiempo. Como es estacionaria será más fácil de predecir.

## **Capítulo 2**

# **WORKING HERE**

## Capítulo 3

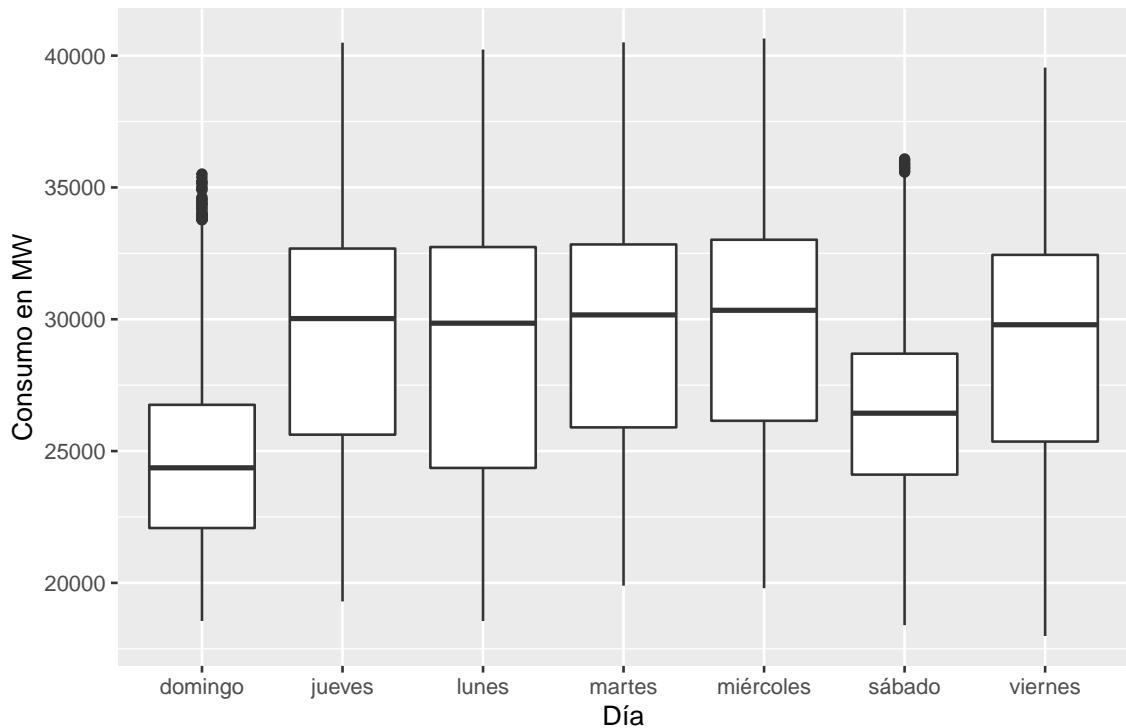
# Búsqueda de outliers

En el capítulo anterior se ha comentado que podrían existir factores que afecten a la serie temporal. Uno de esos factores podría ser que, durante los fines de semana, el consumo eléctrico disminuya considerablemente en comparación con el resto de los días de la semana, y eso provocaría la aparición de valores atípicos en la serie temporal. Otro factor que puede hacer disminuir el consumo eléctrico considerablemente y provocar la aparición de valores atípicos en la serie temporal, son los días festivos. Así, se han considerado estos dos factores para localizar valores atípicos en la serie temporal, si es que tiene.

Primero se analiza el consumo eléctrico durante los fines de semana mediante la siguiente figura. Pero antes se añade a los datos una columna nueva *day*, para indicar los días de la semana.

```
data$day <- format(data$datetime, format = "%A")
ggplot(data, aes(day, demand)) +
  geom_boxplot() +
  ggtitle("Consumos eléctricos por día de la semana del año 2015") +
  xlab("Día") +
  ylab("Consumo en MW")
```

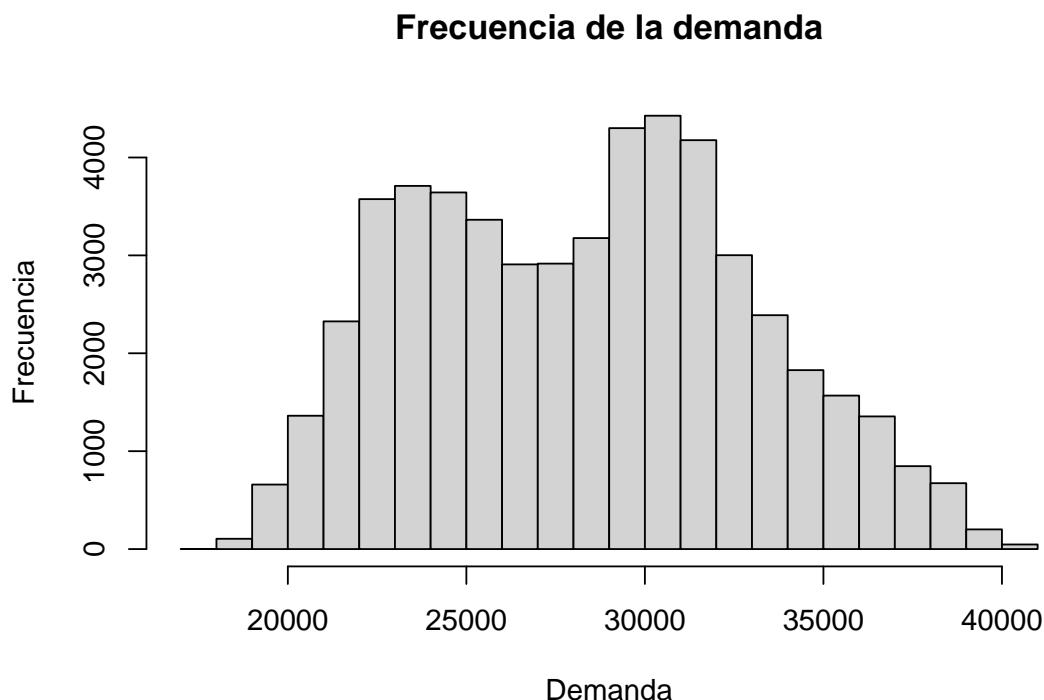
### Consumos eléctricos por día de la semana del año 2015



Como se observa en la figura durante los fines de semana (sábado y domingo), la demanda disminuye considerablemente en comparación con el resto de los días de la semana y aparecen valores atípicos.

Por otro lado, ahora nos fijaremos en qué rangos se mueve la demanda del consumo eléctrico a lo largo del 2015. Se usa un histograma para que aporte esa información con un simple vistazo.

```
hist(data$demand,  
      main = "Frecuencia de la demanda",  
      ylab = "Frecuencia",  
      xlab = "Demanda")
```



La figura anterior muestra una distribución extraña para la demanda. Lo que se entiende es que:

- Hay poca frecuencia de la demanda, cuando la demanda es inferior a 20000 MW y superior a 35000 MW (aproximadamente).
- La frecuencia de la demanda es frecuente entre 20000 MW y 35000 MW.
- La baja demanda es más frecuente de lo normal dentro de conjunto de datos de la serie temporal, parece que predominan bastantes valores atípicos.

### 3.1. Reemplazo de outliers

A continuación, se definirán los umbrales para reemplazar los valores atípicos de la serie temporal. Se opta por el reemplazo y no por la eliminación de los valores atípicos, para no perder datos representativos de la serie. La eliminación de bastantes datos representativos de la serie puede tener un impacto negativo en las futuras predicciones. Los umbrales que se definirán indicarán qué valores de la demanda serán reemplazados.

Se usa la función `summary()`, que nos devolverá información sobre los cuartiles de la serie. Los cuartiles se usarán para calcular los umbrales.

```
summary(data$demand)
```

```
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 17985   24392   28566   28349   31664   40648
```

En este ejemplo que nos traemos entre manos, se puede observar que el primer cuartil ( $Q1$ ) está en 24392 MW y que el tercer cuartil ( $Q3$ ) está en 31664 MW. El uso de los cuartiles se debe a que entre  $Q1$  y  $Q3$  sabemos que se encuentran el 50% de los valores de la serie. A la distancia entre los valores de  $Q1$  y  $Q3$  se le llama rango intercuantílico (*IQR: InterQuantile Range*). En este caso, el valor de *IQR* es:

```
IQR <- 31664 - 24392
IQR
```

```
## [1] 7272
```

Así teniendo en cuenta los cuartiles sobre la demanda, se define como valor atípico leve aquel que dista 1.5 veces el rango intercuantílico por debajo de  $Q1$  o por encima de  $Q3$ .

$$\text{*outlier*} < Q1 - 1.5 * \text{IQR} \text{ o bien } \text{*outlier*} > Q3 + 1.5 * \text{IQR}$$

y valor atípico extremo aquel que dista 3 veces el rango intercantílico por debajo de  $Q1$  o por encima de  $Q3$ .

$$\text{*outlier*} < Q1 - 3 * \text{IQR} \text{ o bien } \text{*outlier*} > Q3 + 3 * \text{IQR}$$

De hecho, con esta información, se calculan los umbrales *highLimit* y *lowLimit*, de la siguiente manera:

```
highLimit <- 24392 + 1.5 * IQR
lowLimit <- 31664 - 1.5 * IQR
highLimit
```

```
## [1] 35300
```

```
lowLimit
```

```
## [1] 20756
```

Teniendo en cuenta los umbrales, todos los valores de la demanda que superen los 35300 MW se consideran *outliers*. Y todos los valores de la demanda que sean inferiores a los 20756 MW también son *outliers*.

Una vez que se han identificado los *outliers*, se procede al reemplazo de estos. Para ello, se define una función propia que recibe como parámetros el conjunto de datos de la serie temporal, el umbral inferior y el umbral superior. En la función se decide reemplazar por la media aquellos valores atípicos que estén por debajo del umbral inferior, y por la mediana aquellos que estén por encima del umbral superior, procedimiento muy utilizado en el procedimiento de reemplazar *outliers*.

La función y el procedimiento del reemplazo se muestra a continuación:

```

outliersReplace <- function(data, lowLimit, highLimit) {
  data[data < lowLimit] <- mean(data)
  data[data > highLimit] <- median(data)
  data
}

data$demand.W0 <- outliersReplace(data$demand, lowLimit, highLimit)

summary(data$demand)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    17985    24392   28566    28349   31664    40648

summary(data$demand.W0)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    20756    24863   28566    27954   30720    35300

```

## 3.2. INVENT

Es normal que la media, mediana y cuartiles hayan cambiado, pues los outliers tenían mucho peso y distorsionaban los datos. Aún así, este muestreo que es un subconjunto del original, tiene pinta que no está bien balanceado y tendrá sus propios outliers. No hay más que ver la distancia que hay por encima del Q3.

```

breakPoints <- c(0, 24392, 31664, Inf)
categories <- c("Low", "Medium", "High")
data$demand.F <- cut(data$demand, breaks = breakPoints, labels = categories)
demand_low <- data[data$demand.F == "Low", ]
demand_low2 <- demand_low[demand_low$day != "dissabte" & demand_low$day != "diumenge", ]
dates <- as.Date(demand_low2$date, format = '%d/%m/%Y')
dates_unique <- unique(dates)
dates_unique

## [1] "2015-01-01" "2015-01-02" "2015-01-03" "2015-01-04" "2015-01-05"
## [6] "2015-01-06" "2015-01-07" "2015-01-08" "2015-01-09" "2015-01-10"
## [11] "2015-01-11" "2015-01-12" "2015-01-13" "2015-01-14" "2015-01-15"
## [16] "2015-01-16" "2015-01-17" "2015-01-18" "2015-01-19" "2015-01-20"
## [21] "2015-01-24" "2015-01-25" "2015-01-26" "2015-01-27" "2015-01-29"
## [26] "2015-01-31" "2015-02-01" "2015-02-02" "2015-02-07" "2015-02-08"
## [31] "2015-02-09" "2015-02-13" "2015-02-14" "2015-02-15" "2015-02-16"
## [36] "2015-02-17" "2015-02-18" "2015-02-19" "2015-02-20" "2015-02-21"
## [41] "2015-02-22" "2015-02-23" "2015-02-24" "2015-02-25" "2015-02-26"
## [46] "2015-02-27" "2015-02-28" "2015-03-01" "2015-03-02" "2015-03-03"
## [51] "2015-03-04" "2015-03-05" "2015-03-06" "2015-03-07" "2015-03-08"
## [56] "2015-03-09" "2015-03-10" "2015-03-11" "2015-03-12" "2015-03-13"
## [61] "2015-03-14" "2015-03-15" "2015-03-16" "2015-03-17" "2015-03-18"
## [66] "2015-03-19" "2015-03-20" "2015-03-21" "2015-03-22" "2015-03-23"

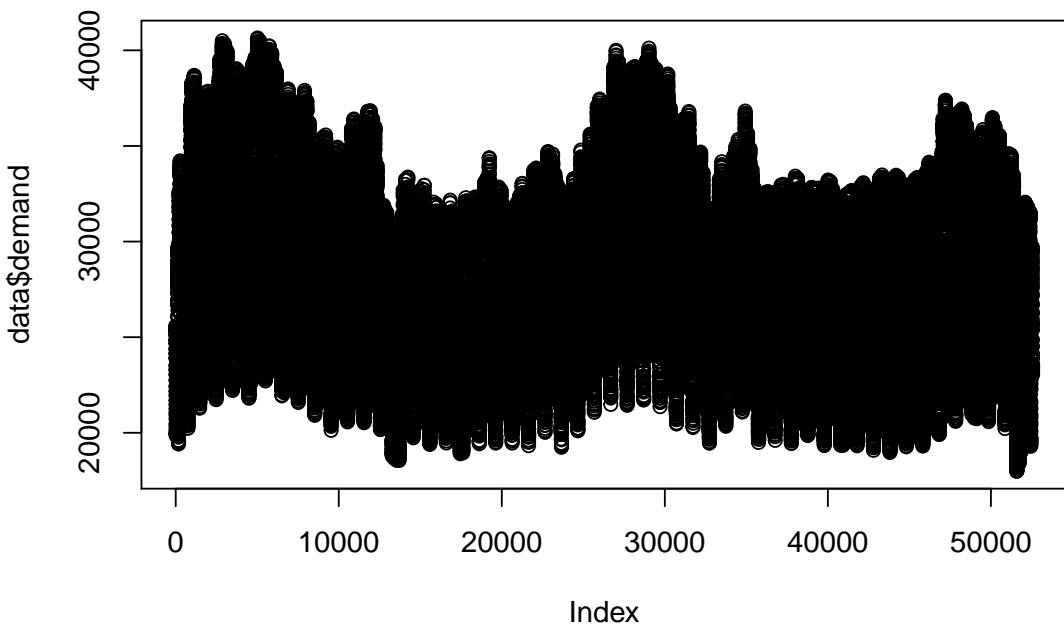
```

```
## [71] "2015-03-24" "2015-03-25" "2015-03-26" "2015-03-27" "2015-03-28"
## [76] "2015-03-29" "2015-03-30" "2015-03-31" "2015-04-01" "2015-04-02"
## [81] "2015-04-03" "2015-04-04" "2015-04-05" "2015-04-06" "2015-04-07"
## [86] "2015-04-08" "2015-04-09" "2015-04-10" "2015-04-11" "2015-04-12"
## [91] "2015-04-13" "2015-04-14" "2015-04-15" "2015-04-16" "2015-04-17"
## [96] "2015-04-18" "2015-04-19" "2015-04-20" "2015-04-21" "2015-04-22"
## [101] "2015-04-23" "2015-04-24" "2015-04-25" "2015-04-26" "2015-04-27"
## [106] "2015-04-28" "2015-04-29" "2015-04-30" "2015-05-01" "2015-05-02"
## [111] "2015-05-03" "2015-05-04" "2015-05-05" "2015-05-06" "2015-05-07"
## [116] "2015-05-08" "2015-05-09" "2015-05-10" "2015-05-11" "2015-05-12"
## [121] "2015-05-13" "2015-05-14" "2015-05-15" "2015-05-16" "2015-05-17"
## [126] "2015-05-18" "2015-05-19" "2015-05-20" "2015-05-21" "2015-05-22"
## [131] "2015-05-23" "2015-05-24" "2015-05-25" "2015-05-26" "2015-05-27"
## [136] "2015-05-28" "2015-05-29" "2015-05-30" "2015-05-31" "2015-06-01"
## [141] "2015-06-02" "2015-06-03" "2015-06-04" "2015-06-05" "2015-06-06"
## [146] "2015-06-07" "2015-06-08" "2015-06-09" "2015-06-10" "2015-06-11"
## [151] "2015-06-12" "2015-06-13" "2015-06-14" "2015-06-15" "2015-06-16"
## [156] "2015-06-17" "2015-06-18" "2015-06-19" "2015-06-20" "2015-06-21"
## [161] "2015-06-22" "2015-06-23" "2015-06-24" "2015-06-25" "2015-06-26"
## [166] "2015-06-27" "2015-06-28" "2015-06-29" "2015-07-04" "2015-07-05"
## [171] "2015-07-06" "2015-07-11" "2015-07-12" "2015-07-13" "2015-07-19"
## [176] "2015-07-20" "2015-07-25" "2015-07-26" "2015-07-27" "2015-08-01"
## [181] "2015-08-02" "2015-08-03" "2015-08-04" "2015-08-05" "2015-08-06"
## [186] "2015-08-07" "2015-08-08" "2015-08-09" "2015-08-10" "2015-08-11"
## [191] "2015-08-12" "2015-08-13" "2015-08-14" "2015-08-15" "2015-08-16"
## [196] "2015-08-17" "2015-08-18" "2015-08-19" "2015-08-20" "2015-08-21"
## [201] "2015-08-22" "2015-08-23" "2015-08-24" "2015-08-25" "2015-08-26"
## [206] "2015-08-27" "2015-08-28" "2015-08-29" "2015-08-30" "2015-08-31"
## [211] "2015-09-01" "2015-09-02" "2015-09-03" "2015-09-04" "2015-09-05"
## [216] "2015-09-06" "2015-09-07" "2015-09-08" "2015-09-09" "2015-09-10"
## [221] "2015-09-11" "2015-09-12" "2015-09-13" "2015-09-14" "2015-09-15"
## [226] "2015-09-16" "2015-09-17" "2015-09-18" "2015-09-19" "2015-09-20"
## [231] "2015-09-21" "2015-09-22" "2015-09-23" "2015-09-24" "2015-09-25"
## [236] "2015-09-26" "2015-09-27" "2015-09-28" "2015-09-29" "2015-09-30"
## [241] "2015-10-01" "2015-10-02" "2015-10-03" "2015-10-04" "2015-10-05"
## [246] "2015-10-06" "2015-10-07" "2015-10-08" "2015-10-09" "2015-10-10"
## [251] "2015-10-11" "2015-10-12" "2015-10-13" "2015-10-14" "2015-10-15"
## [256] "2015-10-16" "2015-10-17" "2015-10-18" "2015-10-19" "2015-10-20"
## [261] "2015-10-21" "2015-10-22" "2015-10-23" "2015-10-24" "2015-10-25"
## [266] "2015-10-26" "2015-10-27" "2015-10-28" "2015-10-29" "2015-10-30"
## [271] "2015-10-31" "2015-11-01" "2015-11-02" "2015-11-03" "2015-11-04"
## [276] "2015-11-05" "2015-11-06" "2015-11-07" "2015-11-08" "2015-11-09"
## [281] "2015-11-10" "2015-11-11" "2015-11-12" "2015-11-13" "2015-11-14"
## [286] "2015-11-15" "2015-11-16" "2015-11-17" "2015-11-18" "2015-11-19"
## [291] "2015-11-20" "2015-11-21" "2015-11-22" "2015-11-23" "2015-11-24"
## [296] "2015-11-25" "2015-11-26" "2015-11-27" "2015-11-28" "2015-11-29"
## [301] "2015-11-30" "2015-12-01" "2015-12-02" "2015-12-03" "2015-12-04"
## [306] "2015-12-05" "2015-12-06" "2015-12-07" "2015-12-08" "2015-12-09"
## [311] "2015-12-10" "2015-12-11" "2015-12-12" "2015-12-13" "2015-12-14"
## [316] "2015-12-15" "2015-12-16" "2015-12-17" "2015-12-18" "2015-12-19"
## [321] "2015-12-20" "2015-12-21" "2015-12-22" "2015-12-23" "2015-12-24"
```

```
## [326] "2015-12-25" "2015-12-26" "2015-12-27" "2015-12-28" "2015-12-29"
## [331] "2015-12-30" "2015-12-31"
```

los high pasamos de explicar que date son. ya hay suficientes ejemplos. explicar quartiles y reemplazo de outliers en R.

```
plot(data$demand)
```



```
summary(data$demand)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
## 17985   24392   28566   28349   31664   40648
```

```
datarm <- data[data$demand.F != "Low",]
summary(datarm)
```

```
##      datetime                  date                  time
##  Min.   :2015-01-01 00:00:00  Length:39419   Length:39419
##  1st Qu.:2015-03-24 07:15:00  Class :character  Class :character
##  Median :2015-07-01 06:00:00  Mode   :character  Mode   :character
##  Mean   :2015-06-28 03:30:23
##  3rd Qu.:2015-09-24 10:55:00
##  Max.   :2015-12-31 22:30:00
##      demand          month           day      demand.W0
##  Min.   :24393  Length:39419  Length:39419  Min.   :24393
##  1st Qu.:27480  Class :character  Class :character  1st Qu.:27480
```

```
## Median :30183   Mode  :character   Mode  :character   Median :29232
## Mean    :30324                           Mean    :29431
## 3rd Qu.:32687                           3rd Qu.:31454
## Max.   :40648                           Max.   :35300
##      demand.F
##  Low   :    0
## Medium:26282
## High  :13137
##
```

## Capítulo 4

# Método de predicción

## **Capítulo 5**

# **Aplicación del método de predicción**

Se aplica el método de predicción descrito en el capítulo anterior para obtener una predicción del consumo eléctrico con un horizonte temporal de 4 horas para los días de la semana desde el lunes 8 de junio hasta el domingo 14 de junio.

# Bibliografía

Fernando Campos. Análisis de series temporales en r. arima. Disponible en <https://www.diegocalvo.es/analisis-de-series-temporales-en-r-arima/>, 2018.

Pedro L. Luque-Calvo. Escribir un trabajo fin de estudios con r markdown. Disponible en <http://destio.us.es/calvo/post/escribir-un-trabajo-fin-de-estudios-con-rmarkdown/>, 2017.