

11. ANEXOS

Para la elaboración del presente Trabajo de Fin de Máster se han utilizado las herramientas de programación R y Python. Los siguientes apartados muestran el código de los distintos archivos donde se ha desarrollado todo el código necesario.

ANEXO I: ARCHIVOS CSV

Los archivos csv utilizados en el TFM son:

- 1.Precio medio diario EE 2012-2022.csv
- 2.Demanda media diaria 2012-2022.csv
- 3.Precio petróleo 2012-2022.csv
- 4.Precio Gas Natural 2012-2022.csv
- 5.Precio Carbon 2012-2022.csv
- 6.Producción por tecnologías 2012-2022.csv
- 7.1.Temp Madrid 2012-2022.csv
- 7.1.Temp Barcelona 2012-2022.csv
- 7.1.Temp Valencia 2012-2022.csv
- 7.1.Temp Sevilla 2012-2022.csv
- 7.1.Temp Zaragoza 2012-2022.csv
- 8.1.Vel viento Valladolid 2012-2022.csv
- 8.2.Vel viento Albacete 2012-2022.csv
- 8.3.Vel viento Zaragoza 2012-2022.csv
- 8.4.Vel viento La Coruña 2012-2022.csv
- 8.5.Vel viento Huelva 2012-2022.csv

- 9.Reservas hidraulicas 2012-2022.csv
- 10.Precio derechos emisión CO2 2012-2022.csv
- 11.Indice Ibex35 2012-2022.csv
- 12.Intercambio de EE paises 2012-2022.csv
- 13.Dataset de datos (df_3).csv
- 14.Correlacion_entre_var v01.xlsx
- 15.Dataset final de datos.csv
- 16.df_def_outliers_elim.csv
- 17.df_def_outliers_sust.csv

ANEXO II: EXPLORACION DE LAS VARIABLES

Exploración de las variables

Iñigo Elorza Barea

2023-07-05

6.5. EXPLORACION DE LAS VARIABLES

Con este código se pretende analizar de forma independientes las variables que se van a emplear en los modelos para la predicción del precio de la EE en España

Las variables a analizar son:

NOMBRE	CODIGO
Precio medio diario de la EE en España	Pre_elec
Demanda media diaria	Dem
Precio del petróleo	Prec_petr
Precio del gas natural	Prec_gas
Precio del carbón	Prec_car
Producción en parques eólicos	Prod_eol
Producción en parques solares	Prod_sol
Producción hidráulica y turbinación por bombeo	Prod_hidr
Producción de otras fuentes renovables	Prod_ofr
Producción en centrales nucleares	Prod_nucl
Producción en centrales convencionales de gasóleo	Prod_pet
Producción en centrales con turbina de gas	Prod_gas
Producción en centrales convencionales de carbón	Prod_carb
Producción en centrales de ciclo combinado	Prod_conv
Producción en centrales de cogeneración	Prod_cog
Producción de otras fuentes no renovables	Prod_noren
Temperatura	Temp
Velocidad del viento	Vel_vien
Reservas hidráulicas	Res_hidr
Precio por Derechos de emisión de CO2	Der_CO2
Índice IBEX-35	Ibex
Intercambio de EE con otros países	Int_ee

Los datos de todas estas variables se encuentran en archivos csv, por lo que se irán cargando según sea necesario y trabajando con ellos. Enumeramos a continuación estos archivos:

1. *Precio medio diario EE 2012-2022.csv*
 2. *Demanda media diaria 2012-2022.csv*
 3. *Precio petróleo 2012-2022.csv*
 4. *Precio Gas Natural 2012-2022.csv*
 5. *Precio Carbon 2012-2022.csv*
 6. *Producción por tecnologías 2012-2022.csv*
 7. *Temperatura 2012-2022.csv*
 8. *Velocidad viento 2012-2022.csv*
 9. *Reservas hidráulicas 2012-2022.csv*
 10. *Precio derechos emisión CO2 2012-2022.csv*
 11. *Índice Ibex35 2012-2022.csv*
 12. *Intercambio de EE países 2012-2022.csv*
-

Antes de nada cargaremos las librerías necesarias:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##   filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Loading required package: timechange
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   date, intersect, setdiff, union
```

6.5.1 La primera variable que analizaremos será "Precio medio diario EE (Pre_elec)"

```
# Cargamos el archivo con los datos de esta variable "Precio medio diario EE (2012-2022).csv"
```

```
Pre_elec <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.  
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv  
y excel/1.Precio medio diario EE 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(Pre_elec)
```

```

##   id          name geoid geoname  value
## 1 10211 Precio medio horario final suma de componentes NA  NA 48.1992
## 2 10211 Precio medio horario final suma de componentes NA  NA 49.7375
## 3 10211 Precio medio horario final suma de componentes NA  NA 57.1317
## 4 10211 Precio medio horario final suma de componentes NA  NA 54.6721
## 5 10211 Precio medio horario final suma de componentes NA  NA 51.5471
## 6 10211 Precio medio horario final suma de componentes NA  NA 52.4642
##           datetime
## 1 2012-01-01T00:00:00+01:00
## 2 2012-01-02T00:00:00+01:00
## 3 2012-01-03T00:00:00+01:00
## 4 2012-01-04T00:00:00+01:00
## 5 2012-01-05T00:00:00+01:00
## 6 2012-01-06T00:00:00+01:00

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Pre_elec)

## 'data.frame': 3652 obs. of 6 variables:
## $ id    : int 10211 10211 10211 10211 10211 10211 10211 10211 10211 ...
## $ name  : chr "Precio medio horario final suma de componentes" ...
## $ geoid : logi NA NA NA NA NA NA ...
## $ geoname : logi NA NA NA NA NA ...
## $ value : num 48.2 49.7 57.1 54.7 51.5 ...
## $ datetime: chr "2012-01-01T00:00:00+01:00" "2012-01-02T00:00:00+01:00" "2012-01-03T00:00:00+01:00" "2012-01-04T00:00:00+01:00" ...

print("-----")
## [1] "-----"

summary(Pre_elec)

##   id      name    geoid    geoname
## Min. :10211  Length:3652  Mode:logical Mode:logical
## 1st Qu.:10211  Class :character NA's:3652  NA's:3652
## Median :10211  Mode :character
## Mean   :10211
## 3rd Qu.:10211
## Max.  :10211
##   value    datetime
## Min. :10.04  Length:3652
## 1st Qu.:51.10  Class :character
## Median :59.18  Mode :character
## Mean   :73.89
## 3rd Qu.:69.06
## Max.  :556.15

# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'value' es:", format(mean(is.na(Pre_elec$value)))))

## [1] "El número de NA's de la columna 'value' es: 0"

```

```
print(paste("El número de NA's de la columna 'datetime' es:", format(mean(is.na(Pre_elec$datetime)))))
```

```
## [1] "El número de NA's de la columna 'datetime' es: 0"
```

Como se puede comprobar tenemos 3652 registros (correspondientes al número de días de los años del periodo estudiado). Las columnas que nos interesan son “value” y “datetime” y en ninguna de las dos existen valores perdidos.

Posteriormente habrá que transformar los datos de la columna “datetime” para que sean del tipo “fecha”.

6.5.2 La segunda variable es “Demanda media diaria (Dem)”

```
# Cargamos el archivo con los datos
```

```
Dem <- read.table("D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/2.Demanda media diaria 2012-2022.csv", header = T, sep = ';', encoding = "UTF-8")
```

```
head(Dem)
```

```
##   id      name geoid geoname   value      datetime
## 1 1293 Demanda real    NA    NA 20888.99 2012-01-01T00:00:00+01:00
## 2 1293 Demanda real    NA    NA 23325.20 2012-01-02T00:00:00+01:00
## 3 1293 Demanda real    NA    NA 26994.32 2012-01-03T00:00:00+01:00
## 4 1293 Demanda real    NA    NA 27934.39 2012-01-04T00:00:00+01:00
## 5 1293 Demanda real    NA    NA 28089.09 2012-01-05T00:00:00+01:00
## 6 1293 Demanda real    NA    NA 24915.97 2012-01-06T00:00:00+01:00
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos

```
str(Dem)
```

```
## 'data.frame': 3651 obs. of 6 variables:
## $ id : int 1293 1293 1293 1293 1293 1293 1293 1293 1293 ...
## $ name : chr "Demanda real" "Demanda real" "Demanda real" "Demanda real" ...
## $ geoid : logi NA NA NA NA NA ...
## $ geoname : logi NA NA NA NA NA ...
## $ value : num 20889 23325 26994 27934 28089 ...
## $ datetime: chr "2012-01-01T00:00:00+01:00" "2012-01-02T00:00:00+01:00" "2012-01-03T00:00:00+01:00" "2012-01-04T00:00:00+01:00" ...
```

```
print("-----")
```

```
## [1] "-----"
```

```
summary(Dem)
```

```
##   id      name      geoid      geoname
## Min. :1293 Length:3651 Mode:logical Mode:logical
## 1st Qu.:1293 Class :character NA's:3651 NA's:3651
## Median :1293 Mode :character
## Mean  :1293
## 3rd Qu.:1293
## Max. :1293
##   value      datetime
## Min. :19122 Length:3651
## 1st Qu.:26233 Class :character
## Median :28193 Mode :character
## Mean  :28045
```

```
## 3rd Qu.:29903
## Max. :35306

# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'value' es:", format(mean(is.na(Dem$value)))))

## [1] "El número de NA's de la columna 'value' es: 0"

print(paste("El número de NA's de la columna 'datetime' es:", format(mean(is.na(Dem$datetime)))))

## [1] "El número de NA's de la columna 'datetime' es: 0"
```

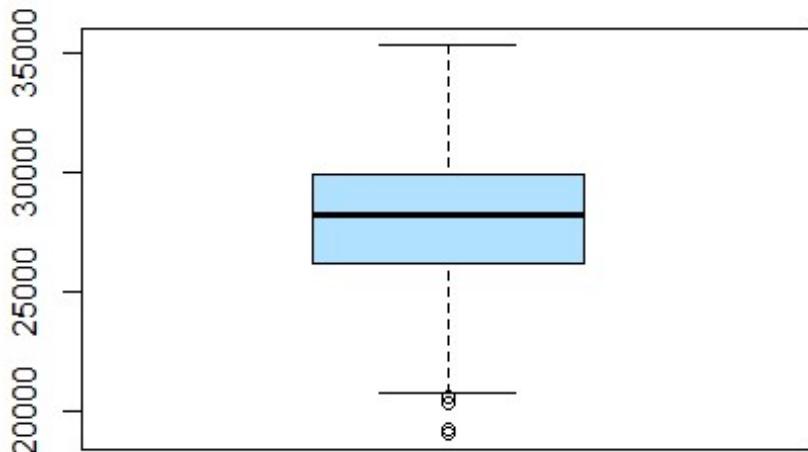
Este caso tiene la misma estructura que la anterior por lo que las columnas que nos interesan serán “value” y “datetime”. Ninguna de las dos columnas posee valores perdidos. Lo que no es igual es el número de registros que tenemos. Esto es debido a que en la pagina de Red Eléctrica Española no tienen datos de esta variable para los años 2012 y 2013.

Más adelante decidiremos que hacer con este problema.

Estudiamos ahora los Outliers

```
boxplot(Dem$value, col = 'lightskyblue1', main="Outliers variable: Demanda")
```

Outliers variable: Demanda



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'value' es:", format(length(boxplot.stats(Dem$value)$out))))

## [1] "El número de Outliers existentes en la columna 'value' es: 5"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Dem$value)$out)/length(Dem$value)*100,3)), "%"))
```

```
## [1] "Y el porcentaje: 0.137 %"
```

Podemos comprobar que no son un problema ya que solo existen 2.

6.5.3 La siguiente variable es “Precio del petróleo (Prec_petr)”

Cargamos el archivo con los datos

```
Prec_petr <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11
.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv
y excel/3.Precio petróleo 2012-2022.csv', header = T, sep = ',', encoding = "UTF-8")
head(Prec_petr)
```

```
##     fecha ultimo apertura maximo minimo volumen variacion
## 1 01/01/2012 110.85 109.55 111.20 109.50 210.93  1.62%
## 2 02/01/2012 108.35 110.85 111.32 108.06 189.77 -2.26%
## 3 03/01/2012 112.13 108.35 112.44 108.35 178.62  4.42%
## 4 04/01/2012 113.70 112.10 113.97 111.27 221.27  1.40%
## 5 05/01/2012 112.74 113.50 114.64 112.10 205.41 -0.84%
## 6 06/01/2012 113.06 112.85 113.68 112.03 190.83  0.28%
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos

```
str(Prec_petr)
```

```
## 'data.frame': 2889 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 111 108 112 114 113 ...
## $ apertura : num 110 111 108 112 114 ...
## $ maximo : num 111 111 112 114 115 ...
## $ minimo : num 110 108 108 111 112 ...
## $ volumen : num 211 190 179 221 205 ...
## $ variacion: chr "1.62%" "-2.26%" "4.42%" "1.40%" ...
```

```
print('-----')
```

```
## [1] "-----"
```

```
summary(Prec_petr)
```

```
##    fecha      ultimo      apertura      maximo
## Length:2889   Min. :19.33   Min. :19.90   Min. :21.29
## Class :character 1st Qu.:53.92  1st Qu.:53.97  1st Qu.:54.69
## Mode  :character Median :68.87  Median :68.81  Median :69.63
##                  Mean  :75.20  Mean  :75.20  Mean  :76.19
##                  3rd Qu.:103.41 3rd Qu.:103.32 3rd Qu.:104.60
##                  Max.  :130.24  Max.  :130.28  Max.  :139.13
##
##    minimo      volumen      variacion
## Min. :15.98  Min. : 0.02  Length:2889
## 1st Qu.:53.02 1st Qu.:173.08  Class :character
## Median :67.73  Median :228.19  Mode :character
## Mean  :74.16  Mean  :225.16
## 3rd Qu.:102.07 3rd Qu.:285.35
## Max.  :125.00  Max.  :779.72
## NA's   :1
```

Estudiamos los valores nulos de las columnas que nos interesan

```
print(paste("El número de NA's de la columna 'fecha' es:", format(mean(is.na(Prec_petr$fecha)))))
```

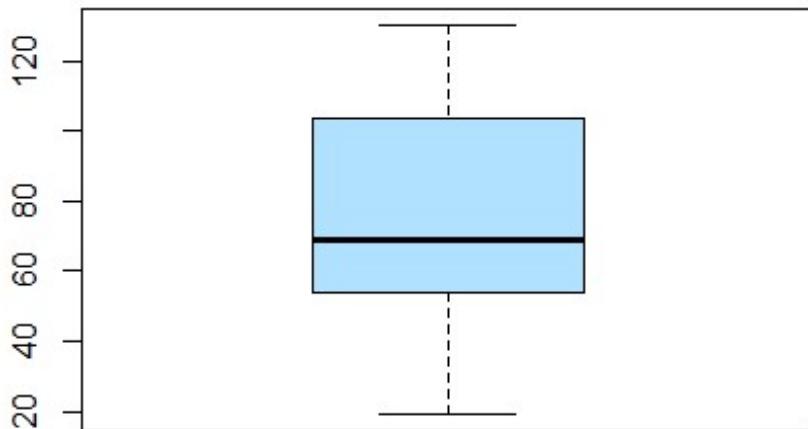
```
## [1] "El número de NA's de la columna 'fecha' es: 0"
print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(Prec_petr$ultimo)))))
## [1] "El número de NA's de la columna 'ultimo' es: 0"
```

En este dataframe existen 2887 registros (765 días menos) y es debido a que esta materia prima se negocia de lunes a sábado. En este caso solo nos interesan las dos primeras columnas ya que son las que indican la fecha y el último valor que tomó el barril de petróleo. Ninguna de las dos columnas presenta valores NA's.

Estudiamos ahora los Outliers

```
boxplot(Prec_petr$ultimo, col = 'lightskyblue1', main='Outliers variable "Precio petróleo"')
```

Outliers variable "Precio petróleo"



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'value' es:", format(length(boxplot.stats(Prec_petr$ultimo)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'value' es: 0"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Prec_petr$ultimo)$out)/length(Prec_petr$ultimo)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0 %"
```

Comprobamos que no existen outliers en esta variable.

6.5.4 La siguiente variable es "Precio del gas natural (Prec_gas)"

Cargamos el archivo con los datos

```
Prec_gas <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv
```

```

y excel/4.Precio Gas Natural 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
head(Prec_gas)

## fecha ultimo apertura maximo minimo volumen variacion
## 1 01/01/2012 55.60 55.23 55.62 55.12 4.95 -0.0150
## 2 02/01/2012 53.75 55.60 55.91 53.96 5.26 -0.0232
## 3 03/01/2012 52.75 53.75 53.75 52.50 5.36 -0.0287
## 4 04/01/2012 53.09 52.76 53.45 52.75 4.96 0.0064
## 5 05/01/2012 52.95 53.60 53.90 52.90 5.23 -0.0026
## 6 06/01/2012 52.87 52.85 53.30 52.80 5.98 -0.0015

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Prec_gas)

## 'data.frame': 2541 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 55.6 53.8 52.8 53.1 53 ...
## $ apertura : num 55.2 55.6 53.8 52.8 53.6 ...
## $ maximo : num 55.6 55.9 53.8 53.5 53.9 ...
## $ minimo : num 55.1 54 52.5 52.8 52.9 ...
## $ volumen : num 4.95 5.26 5.36 4.96 5.23 5.98 9.09 8.26 6.17 2.45 ...
## $ variacion: num -0.015 -0.0232 -0.0287 0.0064 -0.0026 -0.0015 0.0373 -0.0067 0.018 -0.0069 .
..  

print(  
## [1] -----"  

summary(Prec_gas)

## fecha ultimo apertura maximo
## Length:2541 Min. :24.18 Min. :24.49 Min. :24.95
## Class:character 1st Qu.:41.57 1st Qu.:41.60 1st Qu.:42.10
## Mode:character Median :53.86 Median :53.95 Median :54.50
## Mean :77.93 Mean :78.25 Mean :81.17
## 3rd Qu.:66.59 3rd Qu.:66.66 3rd Qu.:67.21
## Max. :640.36 Max. :659.50 Max. :800.00
##
## minimo volumen variacion
## Min. :23.59 Min. :1.11 Min. :-0.299500
## 1st Qu.:41.21 1st Qu.:6.55 1st Qu.:-0.014200
## Median :53.40 Median :9.92 Median :-0.000300
## Mean :75.52 Mean :10.47 Mean :0.001295
## 3rd Qu.:66.11 3rd Qu.:13.41 3rd Qu.:0.015100
## Max. :559.29 Max. :38.61 Max. :0.509300
## NA's :15

# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'fecha' es:", format(mean(is.na(Prec_gas$fecha)))))

## [1] "El número de NA's de la columna 'fecha' es: 0"

print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(Prec_gas$ultimo)))))

## [1] "El número de NA's de la columna 'ultimo' es: 0"

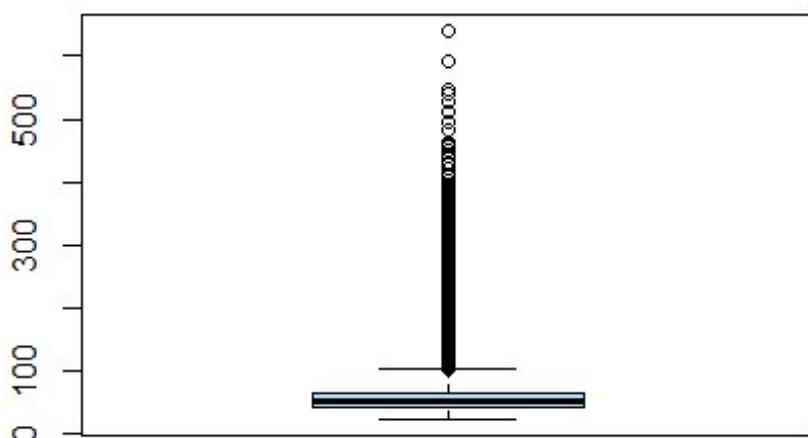
```

Este caso tiene la misma estructura que la anterior por lo que las columnas que nos interesan serán “fecha” y “ultimo”. Ninguna de las dos columnas posee valores perdidos. Para esta variable tenemos 2544 registros correspondientes a los días de lunes a viernes.

Estudiamos ahora los Outliers

```
boxplot(Prec_gas$ultimo, col = 'lightskyblue1', main="Outliers variable Precio gas")
```

Outliers variable Precio gas



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'ultimo' es:", format(length(boxplot.stats(Prec_gas$ultimo)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'ultimo' es: 357"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Prec_gas$ultimo)$out)/length(Prec_gas$ultimo)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 14.05 %"
```

En otra fase del estudio decidiremos que hacemos con estos valores.

6.5.5 La siguiente variable es “Precio del carbón (Prec_carb)”

Cargamos el archivo con los datos

```
Prec_carb <- read.table("D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/5.Precio Carbon 2012-2022.csv", header = T, sep = ';', encoding = "UTF-8")
```

```
head(Prec_carb)
```

```
##     fecha ultimo apertura maximo minimo volumen variacion
## 1 01/01/2012 111.56 110.85 111.98 110.85  0.13   -0.86
## 2 02/01/2012 110.20 111.56 111.56 109.56  0.05   -1.22%
## 3 03/01/2012 109.35 109.35 109.35 109.35  0.12   -0.77%
```

```

## 4 04/01/2012 109.55 109.55 109.55 109.55 0.13 0.18%
## 5 05/01/2012 110.10 110.10 110.10 110.10 0.05 0.50%
## 6 06/01/2012 110.20 110.20 110.20 110.20 0.20 0.09%

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Prec_carb)

## 'data.frame': 2817 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 112 110 109 110 110 ...
## $ apertura : num 111 112 109 110 110 ...
## $ maximo : num 112 112 109 110 110 ...
## $ minimo : num 111 110 109 110 110 ...
## $ volumen : num 0.13 0.05 0.12 0.13 0.05 0.2 0.15 0.02 0.08 0.1 ...
## $ variacion: chr "-0.86" "-1.22%" "-0.77%" "0.18%" ...

print(-----)
## [1] "-----"

summary(Prec_carb)

##   fecha      ultimo     apertura     maximo
##  Length:2817   Min. :38.45   Min. :38.45   Min. :38.45
##  Class :character 1st Qu.:59.05  1st Qu.:59.05  1st Qu.:59.05
##  Mode  :character Median :78.30  Median :78.25  Median :78.30
##                  Mean  :96.68  Mean  :96.64  Mean  :96.95
##                  3rd Qu.:92.45  3rd Qu.:92.50  3rd Qu.:92.50
##                  Max.  :439.00  Max.  :465.00  Max.  :465.00
##
##   minimo     volumen     variacion
##  Min. :38.45  Min. :0.0000  Length:2817
##  1st Qu.:59.00  1st Qu.:0.0100  Class :character
##  Median :78.25  Median :0.0400  Mode  :character
##  Mean  :96.40  Mean  :0.0611
##  3rd Qu.:92.50  3rd Qu.:0.0800
##  Max.  :450.00  Max.  :0.6000
##  NA's    :919

# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'fecha' es:", format(mean(is.na(Prec_carb$fecha)))))

## [1] "El número de NA's de la columna 'fecha' es: 0"

print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(Prec_carb$ultimo)))))

## [1] "El número de NA's de la columna 'ultimo' es: 0"

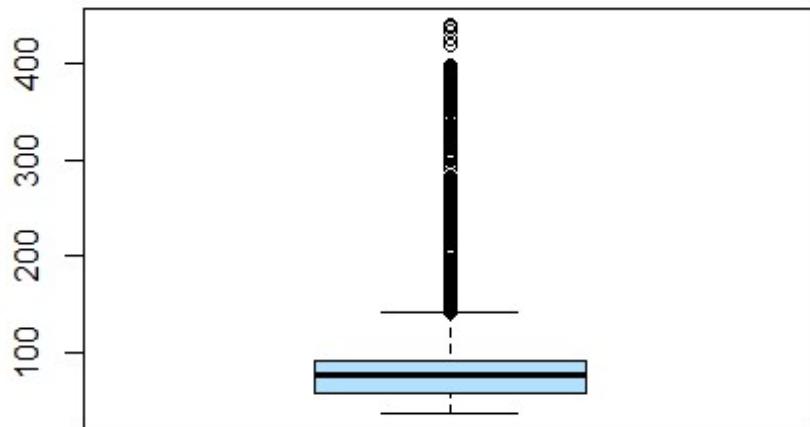
```

Es el mismo caso que el anterior. Lo único que cambia es que existe mayor número de registros debido a que su cotización se produce de lunes a sábado.

Estudiamos ahora los Outliers

```
boxplot(Prec_carb$ultimo, col = 'lightskyblue1', main="Outliers variable Precio carbón")
```

Outliers variable Precio carbón



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'ultimo' es:", format(length(boxplot.stats(Prec_carb$ultimo)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'ultimo' es: 328"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Prec_carb$ultimo)$out)/length(Prec_carb$ultimo)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 11.64 %"
```

En otra fase del estudio decidiremos que hacemos con estos valores.

6.5.6 La siguientes variables corresponden a la producción por tecnología y se encuentran juntas en el mismo archivo

```
# Cargamos el archivo con los datos
Prod_tec <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11. MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/6.Produccion por tecnologias 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
head(Prod_tec)
```

```
##     fecha Eolica Hidroelotica Solar_fotovoltaica Solar_termica Hidraulica
## 1 01/01/2012 155.1421      NA    16.55900   3.342588 44.11561
## 2 02/01/2012 261.0320      NA    12.82085   1.896593 48.18073
## 3 03/01/2012 164.6931      NA    17.66322   4.537361 58.38051
## 4 04/01/2012 164.3039      NA    17.64268   4.962424 63.71691
## 5 05/01/2012 204.1075      NA    17.57371   5.104392 56.62367
## 6 06/01/2012 215.8213      NA    14.44478   3.273886 55.10793
## Turbinacion_bombeo Otras_renovables Residuos_renovables Nuclear
## 1          7.586468    10.28479    2.270296 156.0602
## 2          8.837414    10.25505    2.238878 155.9083
```

```

## 3    12.350879   10.13681   2.265258 155.9070
## 4    13.218982   10.54009   2.282968 155.9415
## 5    14.932920   10.66351   2.327133 155.8380
## 6    9.088064    10.91022   2.318175 156.1161
## Motores_diesel Turbina_de_gas Carbon Ciclo_combinado Cogeneracion Fuel.Gas
## 1    9.099916    1.211399  95.83518   63.99322  61.13340 0.024088
## 2    9.585639    1.852946 118.34087   61.77065  86.04903 0.010099
## 3    9.693961    2.357607 169.17696   99.75440  95.67452 0.007815
## 4    9.460365    2.276651 176.29588   102.35277 96.84716 0.009619
## 5    9.619993    1.970738 140.08747   82.15043  95.93271 0.009536
## 6    9.145990    1.218538 86.14172    54.24204  86.23782 0.024052
## Turbina_de_vapor Residuos_no_renovables Generacion_total
## 1    5.926551     3.558299   636.0949
## 2    7.399333     3.636452   789.7946
## 3    7.737631     3.591249   813.9127
## 4    7.881408     4.013747   831.7278
## 5    8.117390     4.531231   809.5713
## 6    6.464696     4.759207   715.2664

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Prod_tec)

## 'data.frame': 3652 obs. of 19 variables:
## $ fecha      : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ Eolica      : num 155 261 165 164 204 ...
## $ Hidroelica   : num NA NA NA NA NA NA NA NA NA ...
## $ Solar_fotovoltaica : num 16.6 12.8 17.7 17.6 17.6 ...
## $ Solar_termica  : num 3.34 1.9 4.54 4.96 5.1 ...
## $ Hidraulica    : num 44.1 48.2 58.4 63.7 56.6 ...
## $ Turbinacion_bombeo : num 7.59 8.84 12.35 13.22 14.93 ...
## $ Otras_renovables : num 10.3 10.3 10.1 10.5 10.7 ...
## $ Residuos_renovables : num 2.27 2.24 2.27 2.28 2.33 ...
## $ Nuclear       : num 156 156 156 156 156 ...
## $ Motores_diesel  : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Turbina_de_gas  : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Carbon        : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Ciclo_combinado : num 64 61.8 99.8 102.4 82.2 ...
## $ Cogeneracion    : num 61.1 86 95.7 96.8 95.9 ...
## $ Fuel.Gas       : num 0.02409 0.0101 0.00782 0.00962 0.00954 ...
## $ Turbina_de_vapor : num 5.93 7.4 7.74 7.88 8.12 ...
## $ Residuos_no_renovables: num 3.56 3.64 3.59 4.01 4.53 ...
## $ Generacion_total : num 636 790 814 832 810 ...

print('-----')
## [1] "-----"

summary(Prod_tec)

##  fecha      Eolica      Hidroelica  Solar_fotovoltaica
## Length:3652  Min. : 9.688  Min. :0.0000  Min. : 3.739
## Class:character 1st Qu.: 84.011  1st Qu.:0.0128  1st Qu.: 18.479
## Mode:character Median:127.595  Median:0.0425  Median: 26.327
##                  Mean :143.356  Mean :0.0512  Mean : 31.560
##                  3rd Qu.:190.193 3rd Qu.:0.0842  3rd Qu.: 31.889

```

```

##          Max. :430.064  Max. :0.1550  Max. :127.947
##          NA's :911
## Solar_termica    Hidraulica   Turbinacion_bombeo Otras_renovables
## Min. :0.000  Min. :12.74   Min. :0.0121   Min. :5.363
## 1st Qu.: 4.039  1st Qu.: 48.07  1st Qu.: 3.6524  1st Qu.: 9.684
## Median :11.027  Median :67.69   Median :7.0555  Median :10.484
## Mean  :12.821  Mean  :78.46   Mean  :7.7359  Mean  :10.669
## 3rd Qu.:21.210  3rd Qu.: 96.68  3rd Qu.:10.9008  3rd Qu.:11.666
## Max. :33.388  Max. :250.70   Max. :30.4844  Max. :15.495
## NA's :4
## Residuos_renovables Nuclear   Motores_diesel  Turbina_de_gas
## Min. :0.5756  Min. :79.42   Min. :4.579   Min. :0.1438
## 1st Qu.:1.9600 1st Qu.:141.95  1st Qu.: 7.805  1st Qu.:1.4439
## Median :2.3065  Median :156.10   Median :8.655  Median :2.1196
## Mean  :2.1779  Mean  :151.38   Mean  :8.600  Mean  :2.1835
## 3rd Qu.:2.5544  3rd Qu.:168.51  3rd Qu.: 9.527  3rd Qu.:2.8689
## Max. :3.0419  Max. :180.17   Max. :12.026  Max. :5.2203
##
## Carbon      Ciclo_combinado Cogeneracion   Fuel.Gas
## Min. : 0.4657  Min. :22.73   Min. :18.50   Min. :0.0000
## 1st Qu.:27.4792 1st Qu.: 60.56  1st Qu.: 68.57  1st Qu.:0.0040
## Median :84.5868  Median : 86.07  Median : 74.90  Median :0.0051
## Mean  :91.4891  Mean  :105.83   Mean  : 73.72  Mean  :0.0076
## 3rd Qu.:150.2429 3rd Qu.:134.89  3rd Qu.: 81.38  3rd Qu.:0.0120
## Max. :238.1793  Max. :396.45   Max. :103.09  Max. :0.0647
## NA's :2590
## Turbina_de_vapor Residuos_no_renovables Generacion_total
## Min. :0.5636  Min. :2.541   Min. :530.8
## 1st Qu.:4.7191 1st Qu.:4.993   1st Qu.:679.6
## Median :6.2775  Median : 6.007   Median :730.4
## Mean  :5.9172  Mean  : 5.932   Mean  : 731.9
## 3rd Qu.:7.2975 3rd Qu.:6.948   3rd Qu.:781.7
## Max. :9.5210  Max. : 8.821   Max. : 997.2
##

```

Estudiamos los valores nulos de las variables

```
apply(is.na(Prod_tec),2,sum)
```

```

##       fecha      Eolica      Hidroeolica
##          0          0         911
## Solar_fotovoltaica Solar_termica    Hidraulica
##          0          4          0
## Turbinacion_bombeo  Otras_renovables Residuos_renovables
##          0          0          0
##      Nuclear     Motores_diesel  Turbina_de_gas
##          0          0          0
##      Carbon      Ciclo_combinado Cogeneracion
##          0          0          0
##      Fuel.Gas    Turbina_de_vapor Residuos_no_renovables
##      2590          0          0
## Generacion_total
##          0

```

Este dataframe contiene una gran cantidad de variables que se emplearán en los modelos. No existe ninguna sola columna que no nos interese para el estudio. El numero de registros es de 3653 correspondientes al total de días de todo el periodo estudiado. Solo 2 variables contienen valores faltantes. En el caso de la "Solar_termica" son despreciables porque son solo 2 casos. Para la variable "Hidroeléctrica" existen casi 1000 casos debido a que hasta junio del 2014 no entró en funcionamiento la primera instalación de estas características en España.

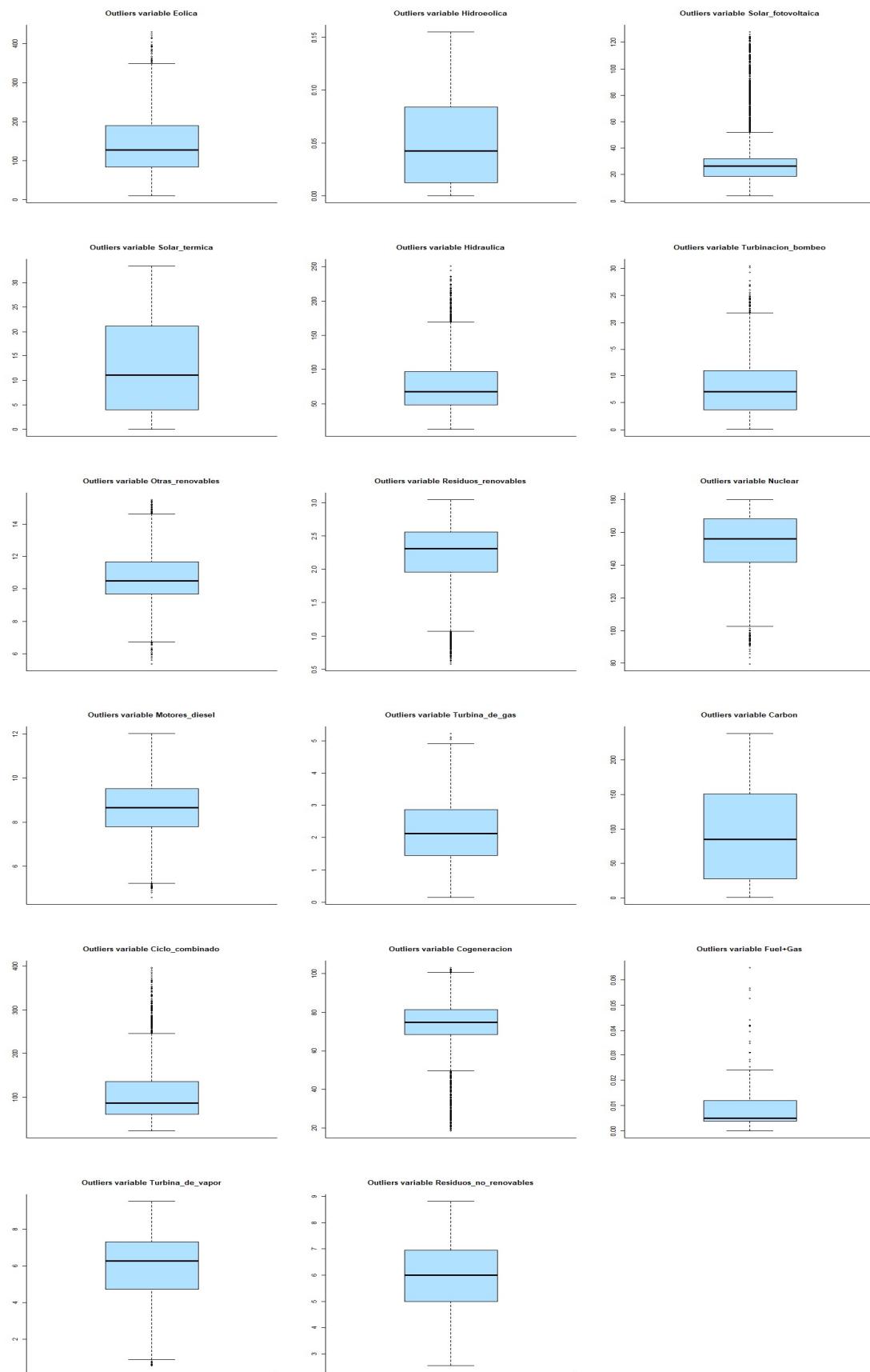
Estudiamos ahora los Outliers

```
par(bty='l', mfrow=c(6,3))
```

```
boxplot(Prod_tec$Eólica, col = 'lightskyblue1', main="Outliers variable Eólica")
boxplot(Prod_tec$Hidroeléctrica, col = 'lightskyblue1', main="Outliers variable Hidroeléctrica")
boxplot(Prod_tec$Solar_fotovoltaica, col = 'lightskyblue1', main="Outliers variable Solar_fotovoltaica")
boxplot(Prod_tec$Solar_termica, col = 'lightskyblue1', main="Outliers variable Solar_termica")
boxplot(Prod_tec$Hidráulica, col = 'lightskyblue1', main="Outliers variable Hidráulica")
boxplot(Prod_tec$Turbinación_bombeo, col = 'lightskyblue1', main="Outliers variable Turbinación_bombeo")
boxplot(Prod_tec$Otras_renovables, col = 'lightskyblue1', main="Outliers variable Otras_renovables")
boxplot(Prod_tec$Residuos_renovables, col = 'lightskyblue1', main="Outliers variable Residuos_renovables")
boxplot(Prod_tec$Nuclear, col = 'lightskyblue1', main="Outliers variable Nuclear")
boxplot(Prod_tec$Motores_diesel, col = 'lightskyblue1', main="Outliers variable Motores_diesel")
boxplot(Prod_tec$Turbina_de_gas, col = 'lightskyblue1', main="Outliers variable Turbina_de_gas")
boxplot(Prod_tec$Carbon, col = 'lightskyblue1', main="Outliers variable Carbon")
boxplot(Prod_tec$Ciclo_combinado, col = 'lightskyblue1', main="Outliers variable Ciclo_combinado")
boxplot(Prod_tec$Cogeneración, col = 'lightskyblue1', main="Outliers variable Cogeneración")
boxplot(Prod_tec$Fuel.Gas, col = 'lightskyblue1', main="Outliers variable Fuel+Gas")
boxplot(Prod_tec$Turbina_de_vapor, col = 'lightskyblue1', main="Outliers variable Turbina_de_vapor")
boxplot(Prod_tec$Residuos_no_renovables, col = 'lightskyblue1', main="Outliers variable Residuos_no_renovables")
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL



```
# Obtenemos el porcentaje de outliers de las variables
print(paste ("El número de Outliers existentes en la columna 'Eolica' es:", format(length(boxplot.stats(Prod_tec$Eolica)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Eolica)$out)/length(Prod_tec$Eolica)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Eolica' es: 33 y el porcentaje: 0.9 %"

print(paste ("El número de Outliers existentes en la columna 'Hidroelica' es:", format(length(boxplot.stats(Prod_tec$Hidroelica)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Hidroelica)$out)/length(Prod_tec$Hidroelica)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Hidroelica' es: 0 y el porcentaje: 0 %"

print(paste ("El número de Outliers existentes en la columna 'Solar_fotovoltaica' es:", format(length(boxplot.stats(Prod_tec$Solar_fotovoltaica)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Solar_fotovoltaica)$out)/length(Prod_tec$Solar_fotovoltaica)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Solar_fotovoltaica' es: 487 y el porcentaje: 13.34 %"

print(paste ("El número de Outliers existentes en la columna 'Solar_termica' es:", format(length(boxplot.stats(Prod_tec$Solar_termica)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Solar_termica)$out)/length(Prod_tec$Solar_termica)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Solar_termica' es: 0 y el porcentaje: 0 %"

print(paste ("El número de Outliers existentes en la columna 'Hidraulica' es:", format(length(boxplot.stats(Prod_tec$Hidraulica)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Hidraulica)$out)/length(Prod_tec$Hidraulica)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Hidraulica' es: 160 y el porcentaje: 4.38 %"

print(paste ("El número de Outliers existentes en la columna 'Turbinacion_bombeo' es:", format(length(boxplot.stats(Prod_tec$Turbinacion_bombeo)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Turbinacion_bombeo)$out)/length(Prod_tec$Turbinacion_bombeo)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Turbinacion_bombeo' es: 45 y el porcentaje: 1.23 %"

print(paste ("El número de Outliers existentes en la columna 'Otras_renovables' es:", format(length(boxplot.stats(Prod_tec$Otras_renovables)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Otras_renovables)$out)/length(Prod_tec$Otras_renovables)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Otras_renovables' es: 68 y el porcentaje: 1.86 %"

print(paste ("El número de Outliers existentes en la columna 'Residuos_renovables' es:", format(length(boxplot.stats(Prod_tec$Residuos_renovables)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Residuos_renovables)$out)/length(Prod_tec$Residuos_renovables)*100,2)), "%"))
## [1] "El número de Outliers existentes en la columna 'Residuos_renovables' es: 203 y el porcentaje: 5.56 %"
```

```

print(paste ("El número de Outliers existentes en la columna 'Nuclear' es:", format(length(boxplot.stats(Prod_tec$Nuclear)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Nuclear)$out)/length(Prod_tec$Nuclear)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Nuclear' es: 67 y el porcentaje: 1.83 %"

print(paste ("El número de Outliers existentes en la columna 'Motores_diesel' es:", format(length(boxplot.stats(Prod_tec$Motores_diesel)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Motores_diesel)$out)/length(Prod_tec$Motores_diesel)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Motores_diesel' es: 20 y el porcentaje: 0.55 %"

print(paste ("El número de Outliers existentes en la columna 'Turbina_de_gas' es:", format(length(boxplot.stats(Prod_tec$Turbina_de_gas)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Turbina_de_gas)$out)/length(Prod_tec$Turbina_de_gas)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Turbina_de_gas' es: 4 y el porcentaje: 0.11 %"

print(paste ("El número de Outliers existentes en la columna 'Carbon' es:", format(length(boxplot.stats(Prod_tec$Carbon)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Carbon)$out)/length(Prod_tec$Carbon)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Carbon' es: 0 y el porcentaje: 0 %"

print(paste ("El número de Outliers existentes en la columna 'Ciclo_combinado' es:", format(length(boxplot.stats(Prod_tec$Ciclo_combinado)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Ciclo_combinado)$out)/length(Prod_tec$Ciclo_combinado)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Ciclo_combinado' es: 160 y el porcentaje: 4.38 %"

print(paste ("El número de Outliers existentes en la columna 'Cogeneracion' es:", format(length(boxplot.stats(Prod_tec$Cogeneracion)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Cogeneracion)$out)/length(Prod_tec$Cogeneracion)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Cogeneracion' es: 248 y el porcentaje: 6.79 %"

print(paste ("El número de Outliers existentes en la columna 'Fuel+Gas' es:", format(length(boxplot.stats(Prod_tec$Fuel.Gas)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Fuel.Gas)$out)/length(Prod_tec$Fuel.Gas)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Fuel+Gas' es: 17 y el porcentaje: 0.47 %"

print(paste ("El número de Outliers existentes en la columna 'Turbina_de_vapor' es:", format(length(boxplot.stats(Prod_tec$Turbina_de_vapor)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Turbina_de_vapor)$out)/length(Prod_tec$Turbina_de_vapor)*100,2)), "%"))

## [1] "El número de Outliers existentes en la columna 'Turbina_de_vapor' es: 14 y el porcentaje: 0.38 %"

print(paste ("El número de Outliers existentes en la columna 'Residuos_no_renovables' es:", format(length(boxplot.stats(Prod_tec$Residuos_no_renovables)$out)), "y el porcentaje:", format(round(length(boxplot.stats(Prod_tec$Residuos_no_renovables)$out)/length(Prod_tec$Residuos_no_renovables)*100,2)), "%"))

```

```
## [1] "El número de Outliers existentes en la columna 'Residuos_no_renovables' es: 0 y el porcentaje: 0 %"
```

Vemos que hay varias variables que presentan valores altos de outliers como son "Fuel+Gas" (23,3%), "Solar_fotovoltaica" (14,89%) y "Cogeneracion" (9,28%). Despues hay una serie de ellas que poseen un porcentaje entre el 5 y el 1% como son "Ciclo_combinado", "Hidraulica", "Residuos_renovables" y "Nuclear". El resto, o bien, no poseen valores anormales, o son inferiores al 1%. Más adelante habrá que tomar una decisión con estos valores en las variables que existen en gran cantidad.

6.5.7 La siguiente variable es "Temperatura (Temp)"

En este tipo de variable, nos interesa tener los datos de la temperatura en varias ciudades diferentes (en este caso las 5 más pobladas), por lo que se contará con varias "subvariables", es decir, trabajaremos con las variables "TempMad", "TempBar", "TempVal", "TempSev" y "TempZar"

Comenzamos con MADRID

Cargamos el archivo con los datos

```
TempMad <- read.table("D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.1.Temp Madrid 2012-2022.csv", header = T, sep = ',', encoding = "UTF-8")
head(TempMad)
```

```
##   fecha indicativo     nombre provincia altitud tmed prec tmin
## 1 01/01/2012      3195 MADRID, RETIRO    MADRID    667 7.0  Ip 2.3
## 2 02/01/2012      3195 MADRID, RETIRO    MADRID    667 8.7 0.1 5.8
## 3 03/01/2012      3195 MADRID, RETIRO    MADRID    667 6.0 0 1.5
## 4 04/01/2012      3195 MADRID, RETIRO    MADRID    667 6.0 0 0.5
## 5 05/01/2012      3195 MADRID, RETIRO    MADRID    667 7.6 0 2.2
## 6 06/01/2012      3195 MADRID, RETIRO    MADRID    667 9.0 0 3.0
##   horatmin tmax horatmax dir velmedia racha horaracha
## 1    7:30 11.6 16:10 25   1.1  7.2 23:59
## 2   23:59 11.6 14:25 25   3.9 12.5 11:40
## 3    7:40 10.5 14:55 23   0.8  3.9 15:10
## 4    7:30 11.6 15:30 12   0.8  5.8  3:40
## 5    7:00 13.0 Varias 19   0.8  4.7  8:10
## 6    7:30 15.0 13:50 11   3.1 11.4 16:10
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
 str(TempMad)

```
## 'data.frame': 3651 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 3195 3195 3195 3195 3195 3195 3195 3195 3195 ...
## $ nombre : chr "MADRID, RETIRO" "MADRID, RETIRO" "MADRID, RETIRO" "MADRID, RETIRO" ...
## $ provincia : chr "MADRID" "MADRID" "MADRID" "MADRID" ...
## $ altitud : int 667 667 667 667 667 667 667 667 667 ...
## $ tmed : num 7.8 7.6 6.7 6.9 8.2 7.3 7.6 3 ...
## $ prec : chr "Ip" "0.1" "0" "0" ...
## $ tmin : num 2.3 5.8 1.5 0.5 2.2 3.3 2.2 1.8 ...
## $ horatmin : chr "7:30" "23:59" "7:40" "7:30" ...
## $ tmax : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ horatmax : chr "16:10" "14:25" "14:55" "15:30" ...
## $ dir : chr "25" "25" "23" "12" ...
## $ velmedia : chr "1.1" "3.9" "0.8" "0.8" ...
```

```

## $ racha : num 7.2 12.5 3.9 5.8 4.7 11.4 5.3 3.6 5.3 6.1 ...
## $ horaracha : chr "23:59" "11:40" "15:10" "3:40" ...

print('-----')
## [1] "-----"

summary(TempMad)

## fecha indicativo nombre provincia
## Length:3651 Min. :3195 Length:3651 Length:3651
## Class :character 1st Qu.:3195 Class :character Class :character
## Mode :character Median :3195 Mode :character Mode :character
## Mean :3195
## 3rd Qu.:3195
## Max. :3195
##
## altitud tmed prec tmin
## Min. :667 Min. :-3.40 Length:3651 Min. :-7.40
## 1st Qu.:667 1st Qu.: 9.40 Class :character 1st Qu.: 5.30
## Median :667 Median :15.00 Mode :character Median :10.30
## Mean :667 Mean :16.04 Mean :10.96
## 3rd Qu.:667 3rd Qu.:22.80 3rd Qu.:16.60
## Max. :667 Max. :33.40 Max. :26.20
## NA's :14 NA's :14
## horatmin tmax horatmax dir
## Length:3651 Min. : 0.30 Length:3651 Length:3651
## Class :character 1st Qu.:13.10 Class :character Class :character
## Mode :character Median :19.80 Mode :character Mode :character
## Mean :21.11
## 3rd Qu.:29.00
## Max. :40.70
## NA's :14
## velmedia racha horaracha
## Length:3651 Min. : 1.9 Length:3651
## Class :character 1st Qu.: 7.2 Class :character
## Mode :character Median : 9.2 Mode :character
## Mean :105.0
## 3rd Qu.:12.2
## Max. :951.1
## NA's :143

# Estudiamos los valores nulos de las variables
apply(is.na(TempMad), 2, sum)

## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 14 0
## tmin horatmin tmax horatmax dir velmedia racha
## 14 0 14 0 0 0 143
## horaracha
## 0

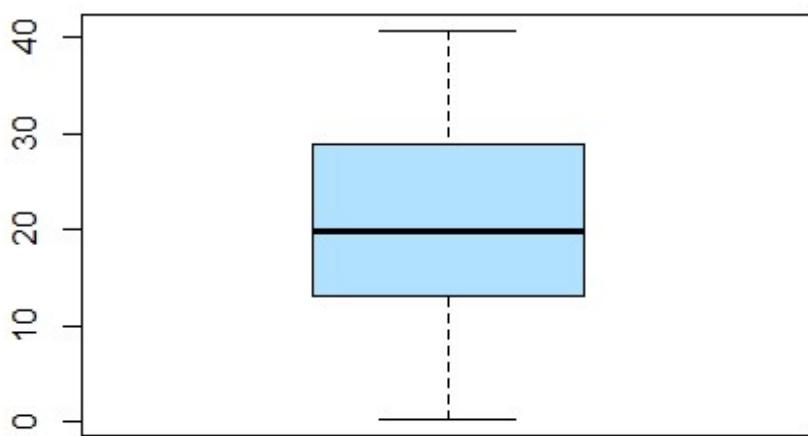
```

Lo primero que hay que destacar es que el dataframe que proporciona la AEMET posee numerosas variables de las cuales solo nos interesa la Temperatura en sus valores mínimo y máximo, la fecha y la localización de la estación donde se han tomado los datos meteorológicos. El número de registros es correcto ya que corresponde al número de días que existen en el intervalo temporal seleccionado para el estudio. Comprobamos que solo existen 14 datos faltantes en estos dos valores.

Estudiamos ahora los Outliers

```
boxplot(TempMad$tmax, col = 'lightskyblue1', main="Outliers variable Temp. Madrid")
```

Outliers variable Temp. Madrid



Obtenemos el porcentaje de outliers de la variable que nos interesa

```
print(paste("El número de Outliers existentes en la columna 'tmax' es:", format(length(boxplot.stats(TempMad$tmax)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'tmax' es: 0"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(TempMad$tmax)$out)/length(TempMad$tmax)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0 %"
```

Comprobamos que no existen outliers en esta variable.

Seguimos con BARCELONA

Cargamos el archivo con los datos

```
TempBar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.2.Temp Barcelona 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(TempBar)
```

```
##   fecha indicativo nombre provincia altitud tmed prec tmin horatmin
## 1 12/01/2012    0201D BARCELONA BARCELONA    6 10.2 0.0 5.7  7:40
## 2 13/01/2012    0201D BARCELONA BARCELONA    6 10.8 0.0 6.0  6:20
## 3 14/01/2012    0201D BARCELONA BARCELONA    6  9.0 0.0 4.5  6:00
## 4 15/01/2012    0201D BARCELONA BARCELONA    6  9.8 0.0 7.1  2:10
## 5 16/01/2012    0201D BARCELONA BARCELONA    6 10.0 0.2 7.8  1:50
## 6 17/01/2012    0201D BARCELONA BARCELONA    6 11.8 0.0 9.9 23:59
##   tmax horatmax dir velmedia racha horaracha
## 1 14.8 15:00 24  3.1  6.7 11:50
## 2 15.6 14:30 20  2.2  7.2 23:20
## 3 13.4 13:40 34  1.4  7.5  3:50
## 4 12.4 14:50 34  2.5  5.8  5:30
## 5 12.2 10:00  6  5.0 15.0 12:30
## 6 13.8 11:30  5  3.9  9.7  1:40
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
`str(TempBar)`

```
## 'data.frame': 3441 obs. of 15 variables:
## $ fecha : chr "12/01/2012" "13/01/2012" "14/01/2012" "15/01/2012" ...
## $ indicativo: chr "0201D" "0201D" "0201D" "0201D" ...
## $ nombre : chr "BARCELONA" "BARCELONA" "BARCELONA" "BARCELONA" ...
## $ provincia : chr "BARCELONA" "BARCELONA" "BARCELONA" "BARCELONA" ...
## $ altitud : int 6 6 6 6 6 6 6 6 ...
## $ tmed : num 10.2 10.8 9.9 8.1 10.1 11.8 10.6 10.8 12.2 12.8 ...
## $ prec : num 0 0 0 0 0.2 0 0 0 0 0 ...
## $ tmin : num 5.7 6 4.5 7.1 7.8 9.9 7.5 6.3 8.9 8.9 ...
## $ horatmin : chr "7:40" "6:20" "6:00" "2:10" ...
## $ tmax : num 14.8 15.6 13.4 12.4 12.2 13.8 13.7 15.4 15.6 16.7 ...
## $ horatmax : chr "15:00" "14:30" "13:40" "14:50" ...
## $ dir : int 24 20 34 34 6 5 33 25 30 33 ...
## $ velmedia : num 3.1 2.2 1.4 2.5 5 3.9 2.5 2.5 1.9 2.5 ...
## $ racha : num 6.7 7.2 7.5 5.8 15.9 7.7 7.5 7.2 8.1 5.6 ...
## $ horaracha : chr "11:50" "23:20" "3:50" "5:30" ...
print('-----')
## [1] "-----"
summary(TempBar)

##   fecha      indicativo     nombre     provincia
##  Length:3441  Length:3441  Length:3441  Length:3441
##  Class :character Class :character Class :character Class :character
##  Mode  :character Mode  :character Mode  :character Mode  :character
## 
## 
## 
##   altitud      tmed      prec      tmin
##  Min. :6 Min. :2.70 Min. :0.000 Min. :-0.20
##  1st Qu.:6 1st Qu.:12.90 1st Qu.: 0.000 1st Qu.: 9.80
##  Median :6 Median :17.10 Median : 0.000 Median :14.20
##  Mean   :6 Mean  :17.64 Mean  : 1.164 Mean  :14.69
##  3rd Qu.:6 3rd Qu.:22.70 3rd Qu.: 0.000 3rd Qu.:20.00
```

```

## Max. :6 Max. :30.00 Max. :83.900 Max. :27.10
##      NA's :215 NA's :22    NA's :215
## horatmin     tmax     horatmax     dir
## Length:3441   Min. :4.60 Length:3441   Min. :1.0
## Class :character 1st Qu.:15.90 Class :character 1st Qu.:11.0
## Mode :character Median :20.10 Mode :character Median :20.0
##          Mean :20.58          Mean :23.9
##          3rd Qu.:25.40         3rd Qu.:27.0
##          Max. :35.20          Max. :99.0
##          NA's :214           NA's :25
## velmedia     racha     horaracha
## Min. :0.0  Min. :3.900 Length:3441
## 1st Qu.: 2.5 1st Qu.: 7.200 Class :character
## Median : 3.3 Median : 8.900 Mode :character
## Mean : 3.5 Mean : 9.652
## 3rd Qu.: 4.2 3rd Qu.:11.400
## Max. :13.9 Max. :26.100
## NA's :14  NA's :25

```

Estudiamos los valores nulos de las variables

```
apply(is.na(TempBar),2,sum)
```

```

## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 215 22
## tmin horatmin tmax horatmax dir velmedia racha
## 215 0 214 0 25 14 25
## horaracha
## 0

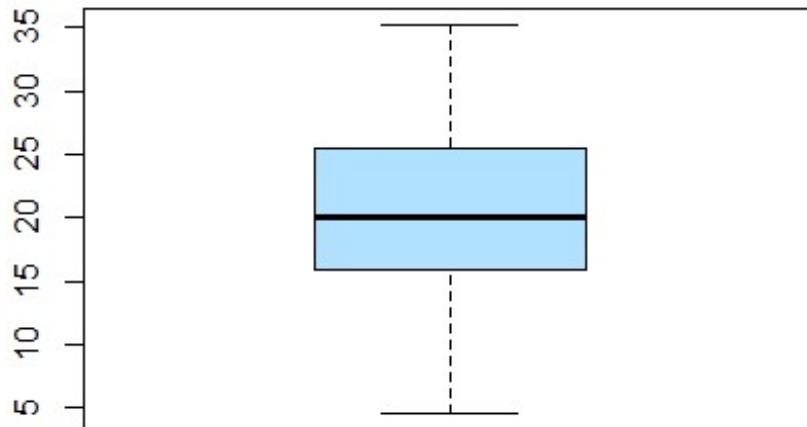
```

En esta ocasión el número de registros es inferior al que debería existir y esto es debido a que AEMET no proporciona datos desde julio a diciembre del 2020. Mas tarde habrá que tomar una solución a este problema. Los datos faltantes en las temperaturas parecen ser que son de esos días mencionados anteriormente.

Estudiamos ahora los Outliers

```
boxplot(TempBar$tmax, col = 'lightskyblue1', main="Outliers variable Temp. Barcelona")
```

Outliers variable Temp. Barcelona



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'tmax' es:", format(length(boxplot.stats(TempBar$tmax)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'tmax' es: 0"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(TempBar$tmax)$out)/length(TempBar$tmax)*100, 2)), "%"))
```

```
## [1] "Y el porcentaje: 0 %"
```

Comprobamos que no existen outliers en esta variable.

La siguiente ciudad es VALENCIA

```
# Cargamos el archivo con los datos
TempVal <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv
y excel/7.3.Temp Valencia 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(TempVal)
```

```
##     fecha nombre provincia altitud tmed prec tmin horatmin tmax horatmax
## 1 01/01/2012 VALENCIA VALENCIA    11 15.5   0 10.0  8:00 21.0 13:50
## 2 02/01/2012 VALENCIA VALENCIA    11 15.4   0 10.3 23:10 20.4 13:30
## 3 03/01/2012 VALENCIA VALENCIA    11 13.4   0  7.2  6:30 19.7 15:00
## 4 04/01/2012 VALENCIA VALENCIA    11 14.6   0  6.7  6:15 22.4 15:30
## 5 05/01/2012 VALENCIA VALENCIA    11 17.4   0  9.2  6:40 25.6 14:20
## 6 06/01/2012 VALENCIA VALENCIA    11 17.2   0 11.4 23:40 23.0 Varias
##     dir velmedia racha horaracha
## 1 24    0.8  7.2  23:00
## 2 26    2.8 13.9 12:40
## 3 14    1.7  6.1 16:50
```

```

## 4 32 1.7 8.3 19:40
## 5 25 2.5 7.8 18:30
## 6 34 3.1 17.8 3:30

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(TempVal)

## 'data.frame': 3652 obs. of 14 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ nombre : chr "VALENCIA" "VALENCIA" "VALENCIA" "VALENCIA" ...
## $ provincia: chr "VALENCIA" "VALENCIA" "VALENCIA" "VALENCIA" ...
## $ altitud : int 11 11 11 11 11 11 11 11 11 11 ...
## $ tmed   : num 15.5 15.4 13.4 14.6 17.4 17.2 13 11.8 11.6 11.7 ...
## $ prec   : chr "0" "0" "0" "0" ...
## $ tmin   : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ horatmin: chr "8:00" "23:10" "6:30" "6:15" ...
## $ tmax   : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ horatmax: chr "13:50" "13:30" "15:00" "15:30" ...
## $ dir    : int 24 26 14 32 25 34 28 11 18 18 ...
## $ velmedia: num 0.8 2.8 1.7 1.7 2.5 3.1 0.8 0.8 0.8 0.8 ...
## $ racha  : num 7.2 13.9 6.1 8.3 7.8 17.8 4.7 5 4.7 7.2 ...
## $ horaracha: chr "23:00" "12:40" "16:50" "19:40" ...

print('-----')
## [1] "-----"

summary(TempVal)

##  fecha      nombre     provincia      altitud
##  Length:3652  Length:3652  Length:3652  Min. :11
##  Class :character  Class :character  Class :character  1st Qu.:11
##  Mode  :character  Mode  :character  Mode  :character  Median :11
##                                         Mean :11
##                                         3rd Qu.:11
##                                         Max. :11
##
##  tmed      prec      tmin      horatmin
##  Min. : 5.30  Length:3652  Min. : 0.00  Length:3652
##  1st Qu.:14.20  Class :character  1st Qu.: 9.70  Class :character
##  Median :18.85  Mode  :character  Median :14.50  Mode  :character
##  Mean   :19.04  Mean   :14.72
##  3rd Qu.:23.90  3rd Qu.:19.80
##  Max.   :32.80  Max.   :27.50
##
##  tmax      horatmax      dir      velmedia
##  Min. : 0.00  Length:3652  Min. : 1.00  Min. :0.600
##  1st Qu.: 7.70  Class :character  1st Qu.:11.00  1st Qu.:1.100
##  Median :11.10  Mode  :character  Median :22.00  Median :1.550
##  Mean   :13.86  Mean   :20.78  Mean   :1.702
##  3rd Qu.:21.20  3rd Qu.:30.00  3rd Qu.:1.975
##  Max.   :42.00  Max.   :35.00  Max.   :3.900
##  NA's   :8       NA's   :3592  NA's   :3592
##  racha    horaracha
##  Min. : 3.600  Length:3652

```

```

## 1st Qu.: 5.225 Class :character
## Median : 6.800 Mode :character
## Mean : 7.920
## 3rd Qu.: 8.675
## Max. :19.700
## NA's :3592

# Estudiamos los valores nulos de las variables
apply(is.na(TempVal),2,sum)

##   fecha nombre provincia altitud tmed prec tmin horatmin
##       0      0        0      0      0     0     0
##   tmax horatmax dir velmedia racha horaracha
##   8      0      3592    3592   3592     0

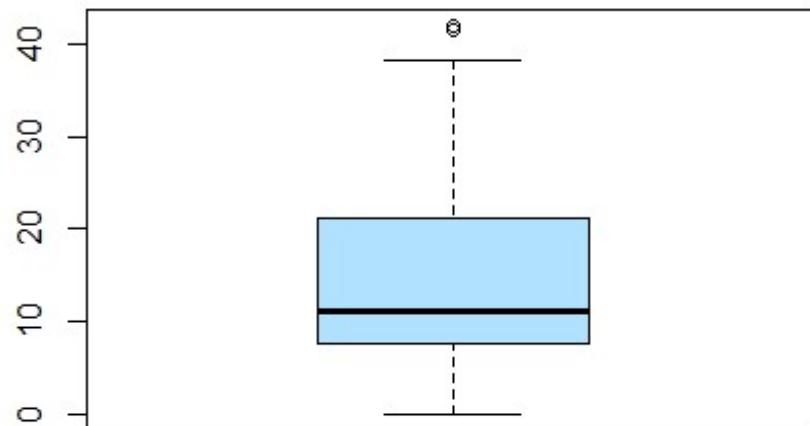
```

En el caso de Valencia el numero de registros es correcto y comprobamos que no existen datos faltantes en las temperaturas.

Estudiamos ahora los Outliers

```
boxplot(TempVal$tmax, col = 'lightskyblue1', main="Outliers variable Temp. Valencia")
```

Outliers variable Temp. Valencia



Obtenemos el porcentaje de outliers de la variable que nos interesa

```
print(paste("El número de Outliers existentes en la columna 'tmax' es:", format(length(boxplot.stats(TempVal$tmax)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'tmax' es: 2"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(TempVal$tmax)$out)/length(TempVal$tmax)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0.05 %"
```

Comprobamos que solo existen 2 outliers en esta variable .

La siguiente ciudad es SEVILLA

Cargamos el archivo con los datos

```
TempSev <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11
.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv
y excel/7.4.Temp Sevilla 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
head(TempSev)
```

```
##   fecha indicativo      nombre provincia altitud tmed prec tmin
## 1 01/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 9.9  0 2.4
## 2 02/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 13.2  0 7.5
## 3 03/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 11.0  0 4.2
## 4 04/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 11.4  0 4.0
## 5 05/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 13.8  0 5.2
## 6 06/01/2012      5783 SEVILLA AEROPUERTO SEVILLA    34 10.2  0 1.4
##   horatmin tmax horatmax dir velmedia racha horaracha
## 1    7:44 17.4 14:35 22   0.8  5.0 Varias
## 2   23:38 19.0 15:22 31   1.4  7.8 16:10
## 3    7:20 17.7 15:38  7   2.2  6.7 14:00
## 4    4:24 18.8 15:31  5   3.3  9.7 12:00
## 5   7:50 22.4 16:00 99   2.2  7.2 Varias
## 6   7:53 19.1 17:04  7   1.1  6.1 10:30
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(TempSev)

```
## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 5783 5783 5783 5783 5783 5783 5783 5783 ...
## $ nombre : chr "SEVILLA AEROPUERTO" "SEVILLA AEROPUERTO" "SEVILLA AEROPUERTO" "SEVILLA AEROPUERTO" ...
## $ provincia : chr "SEVILLA" "SEVILLA" "SEVILLA" "SEVILLA" ...
## $ altitud : int 34 34 34 34 34 34 34 34 34 ...
## $ tmed : num 9.9 13.2 11 11.4 13.8 10.2 13.7 12.8 11.6 11.9 ...
## $ prec : chr "0" "0" "0" "0" ...
## $ tmin : num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ horatmin : chr "7:44" "23:38" "7:20" "4:24" ...
## $ tmax : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ horatmax : chr "14:35" "15:22" "15:38" "15:31" ...
## $ dir : int 22 31 7 5 99 7 7 5 99 5 ...
## $ velmedia : num 0.8 1.4 2.2 3.3 2.2 1.1 5.3 3.6 2.5 2.8 ...
## $ racha : num 5 7.8 6.7 9.7 7.2 6.1 11.4 10.3 6.1 7.2 ...
## $ horaracha : chr "Varias" "16:10" "14:00" "12:00" ...
```

```
print('-----')
```

```
## [1] -----"
```

summary(TempSev)

```
##   fecha      indicativo      nombre      provincia
##  Length:3652  Min. :5783  Length:3652  Length:3652
##  Class :character 1st Qu.:5783  Class :character  Class :character
##  Mode  :character Median :5783  Mode  :character Mode  :character
##                  Mean :5783
```

```

##          3rd Qu.:5783
##      Max. :5783
##
##      altitud     tmed      prec      tmin
##  Min. :34  Min. :4.80  Length:3652  Min. :-2.00
##  1st Qu.:34  1st Qu.:13.80  Class :character  1st Qu.: 8.20
##  Median :34  Median :19.10  Mode  :character  Median :13.50
##  Mean   :34  Mean   :19.58           Mean   :13.16
##  3rd Qu.:34  3rd Qu.:25.40           3rd Qu.:18.10
##  Max. :34  Max. :36.30           Max. :28.10
##      NA's :15           NA's :15
##      horatmin      tmax      horatmax      dir
##  Length:3652  Min. : 7.6  Length:3652  Min. : 1.00
##  Class :character  1st Qu.:18.8  Class :character  1st Qu.:21.00
##  Mode  :character  Median :25.2  Mode  :character  Median :24.00
##          Mean   :26.0           Mean   :45.38
##          3rd Qu.:32.9           3rd Qu.:99.00
##          Max. :45.9           Max. :99.00
##          NA's :13            NA's :32
##      velmedia      racha      horaracha
##  Min. : 0.000  Min. : 3.600  Length:3652
##  1st Qu.: 1.900  1st Qu.: 7.800  Class :character
##  Median : 3.100  Median : 9.700  Mode  :character
##  Mean   : 3.196  Mean   : 9.798
##  3rd Qu.: 4.200  3rd Qu.:11.400
##  Max. :12.800  Max. :28.900
##  NA's :10    NA's :32

```

Estudiamos los valores nulos de las variables

```
apply(is.na(TempSev),2,sum)
```

```

##  fecha indicativo nombre provincia altitud     tmed      prec
##      0      0       0       0       0      15      0
##  tmin horatmin      tmax      horatmax      dir velmedia      racha
##  15      0       13      0       32      10      32
##  horaracha
##      0

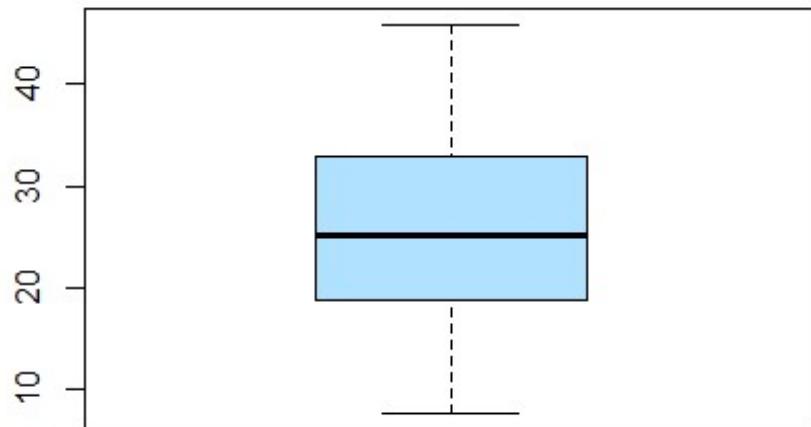
```

El numero de registros es correcto y comprobamos que el numero de datos faltantes en las temperaturas es despreciable.

Estudiamos ahora los Outliers

```
boxplot(TempSev$tmax, col = 'lightskyblue1', main="Outliers variable Temp. Sevilla")
```

Outliers variable Temp. Sevilla



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'tmax' es:", format(length(boxplot.stats(TempSev$tmax)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'tmax' es: 0"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(TempSev$tmax)$out)/length(TempSev$tmax)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0 %"
```

Comprobamos que no existen outliers en esta variable.

La ultima ciudad es ZARAGOZA

```
# Cargamos el archivo con los datos
TempZar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11
.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv
y excel/7.5.Temp Zaragoza 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(TempZar)
```

```
##     fecha indicativo      nombre provincia altitud tmed prec tmin
## 1 01/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 11.8 0 5.4
## 2 02/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 9.6 0.7 5.4
## 3 03/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 7.6 0 2.6
## 4 04/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 9.6 0 3.6
## 5 05/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 10.6 0 5.4
## 6 06/01/2012 9434 ZARAGOZA, AEROPUERTO ZARAGOZA 249 11.9 0 8.4
##     horatmin tmax horatmax dir velmedia racha horaracha
## 1    7:00 18.2 14:40 27   1.7  9.7  0:10
## 2   23:00 13.7 15:00 24   4.4 16.9  0:50
## 3   5:10 12.5 14:45 27   4.4  8.3 19:20
```

```

## 4 5:10 15.5 14:40 29 6.9 15.0 20:20
## 5 8:10 15.7 15:50 30 7.8 19.2 23:25
## 6 23:59 15.4 13:15 29 12.8 20.6 7:00

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(TempZar)

## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 9434 9434 9434 9434 9434 9434 9434 9434 9434 ...
## $ nombre : chr "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" ...
## $ provincia : chr "ZARAGOZA" "ZARAGOZA" "ZARAGOZA" "ZARAGOZA" ...
## $ altitud : int 249 249 249 249 249 249 249 249 249 ...
## $ tmed : num 11.8 9.6 7.6 9.6 10.6 11.9 11.2 10.3 8.4 ...
## $ prec : chr "0" "0.7" "0" "0" ...
## $ tmin : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ horatmin : chr "7:00" "23:00" "5:10" "5:10" ...
## $ tmax : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ horatmax : chr "14:40" "15:00" "14:45" "14:40" ...
## $ dir : int 27 24 27 29 30 29 29 29 25 ...
## $ velmedia : num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ racha : num 9.7 16.9 8.3 15 19.2 20.6 16.9 18.1 12.5 4.7 ...
## $ horaracha : chr "0:10" "0:50" "19:20" "20:20" ...

print('-----')
## [1] "-----"

summary(TempZar)

## fecha indicativo nombre provincia
## Length:3652 Min. :9434 Length:3652 Length:3652
## Class :character 1st Qu.:9434 Class :character Class :character
## Mode :character Median :9434 Mode :character Mode :character
## Mean :9434
## 3rd Qu.:9434
## Max. :9434
##
## altitud tmed prec tmin
## Min. :249 Min. :-0.80 Length:3652 Min. :-5.00
## 1st Qu.:249 1st Qu.:10.30 Class :character 1st Qu.: 5.60
## Median :249 Median :16.00 Mode :character Median :10.60
## Mean :249 Mean :16.54 Mean :10.93
## 3rd Qu.:249 3rd Qu.:22.80 3rd Qu.:16.50
## Max. :249 Max. :33.80 Max. :24.80
##
## horatmin tmax horatmax dir
## Length:3652 Min. :0.40 Length:3652 Min. : 2.00
## Class :character 1st Qu.:14.90 Class :character 1st Qu.:24.00
## Mode :character Median :21.70 Mode :character Median :29.00
## Mean :22.14 Mean :32.27
## 3rd Qu.:29.30 3rd Qu.:31.00
## Max. :44.50 Max. :99.00
## NA's :1

```

```

##  velmedia    racha   horaracha
##  Min. :0.000  Min. :1.9  Length:3652
##  1st Qu.:2.500  1st Qu.:8.9  Class :character
##  Median :3.900  Median :12.2 Mode  :character
##  Mean   :4.644  Mean  :12.5
##  3rd Qu.:6.400  3rd Qu.:15.6
##  Max.  :15.300 Max.  :37.5
##  NA's   :1     NA's  :1

# Estudiamos los valores nulos de las variables
apply(is.na(TempZar),2,sum)

##    fecha indicativo nombre provincia altitud tmed prec
##    0        0         0       0       0       0      0
##    tmin horatmin   tmax horatmax   dir velmedia   racha
##    0        0         0       0       1       1      1
##    horaracha
##    0

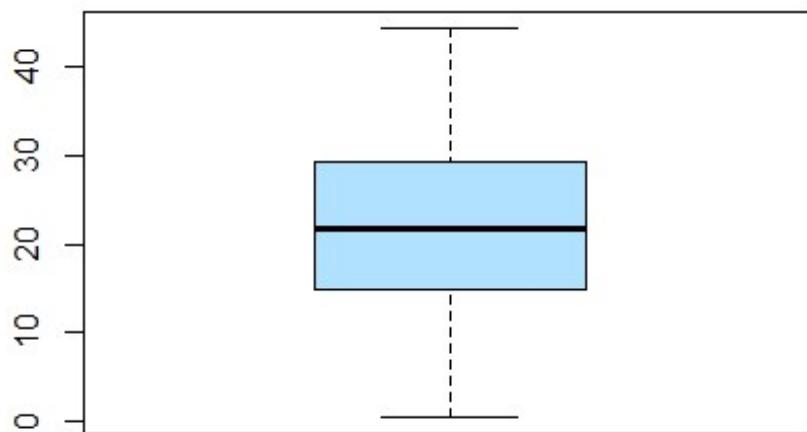
```

Se puede comprobar que todo es correcto.

Estudiamos ahora los Outliers

```
boxplot(TempZar$tmax, col = 'lightskyblue1', main="Outliers variable Temp. Zaragoza")
```

Outliers variable Temp. Zaragoza



Obtenemos el porcentaje de outliers de la variable que nos interesa

```
print(paste("El número de Outliers existentes en la columna 'tmax' es:", format(length(boxplot.stats(TempZar$tmax)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'tmax' es: 0"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(TempZar$tmax)$out)/length(TempZar$tmax)*100,2)), "%"))
## [1] "Y el porcentaje: 0 %"
```

Comprobamos que no existen outliers en esta variable.

6.5.8 La siguiente variable es “Velocidad del viento (Vel_vien)”

Con esta variable sucede lo mismo que con la temperatura, así que tendremos las siguientes subvariables: “Vel_vien_Vall”, “Vel_vien_Alb”, “Vel_vien_Zar”, “Vel_vien_Cor” y “Vel_vien_Hue”

La primera ciudad será VALLADOLID

```
# Cargamos el archivo con los datos
Vel_vien_Vall <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER /11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.1.Vel viento Valladolid 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
head(Vel_vien_Vall)
```

```
##   fecha indicativo nombre provincia altitud tmed prec tmin horatmin
## 1 01/01/2012    2422 VALLADOLID VALLADOLID 735 4.0 0.6 1.1  0:40
## 2 02/01/2012    2422 VALLADOLID VALLADOLID 735 5.7 0 2.7 23:59
## 3 03/01/2012    2422 VALLADOLID VALLADOLID 735 3.2 0 -1.0 8:10
## 4 04/01/2012    2422 VALLADOLID VALLADOLID 735 5.3 0 2.6 Varias
## 5 05/01/2012    2422 VALLADOLID VALLADOLID 735 3.2 0 1.7 19:20
## 6 06/01/2012    2422 VALLADOLID VALLADOLID 735 4.0 0 0.3 8:40
##   tmax horatmax dir velmedia racha horaracha
## 1 7.0 16:30 99 1.1 5.3 Varias
## 2 8.7 13:00 24 5.0 12.5 3:50
## 3 7.3 16:00 18 1.7 5.8 10:40
## 4 8.0 15:10 20 1.4 5.8 5:30
## 5 4.8 16:00 25 1.1 5.3 12:50
## 6 7.7 15:00 19 1.1 3.9 0:40
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
`str(Vel_vien_Vall)`

```
## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 2422 2422 2422 2422 2422 2422 2422 2422 2422 ...
## $ nombre : chr "VALLADOLID" "VALLADOLID" "VALLADOLID" "VALLADOLID" ...
## $ provincia : chr "VALLADOLID" "VALLADOLID" "VALLADOLID" "VALLADOLID" ...
## $ altitud : int 735 735 735 735 735 735 735 735 735 ...
## $ tmed : num 4 5.7 3.2 5.3 3.2 4 5.7 0.2 2 0.8 ...
## $ prec : chr "0.6" "0" "0" "0" ...
## $ tmin : num 1.1 2.7 -1 2.6 1.7 0.3 1.1 -3.4 -0.9 -2.6 ...
## $ horatmin : chr "0:40" "23:59" "8:10" "Varias" ...
## $ tmax : num 7 8.7 7.3 8 4.8 7.7 10.3 3.8 4.8 4.3 ...
## $ horatmax : chr "16:30" "13:00" "16:00" "15:10" ...
## $ dir : int 99 24 18 20 25 19 1 20 6 99 ...
## $ velmedia : num 1.1 5 1.7 1.4 1.1 1.0 8 1.1 1.1 1.1 ...
## $ racha : num 5.3 12.5 5.8 5.8 5.3 3.9 7.5 4.7 3.6 3.3 ...
## $ horaracha : chr "Varias" "3:50" "10:40" "5:30" ...
```

```
print('-----')
```

```
## [1] "-----"
```

```
summary(Vel_vien_Vall)
```

```
## fecha indicativo nombre provincia
## Length:3652 Min. :2422 Length:3652 Length:3652
## Class :character 1st Qu.:2422 Class :character Class :character
## Mode :character Median :2422 Mode :character Mode :character
## Mean :2422
## 3rd Qu.:2422
## Max. :2422
##
## altitud tmed prec tmin
## Min. :735 Min. :-3.10 Length:3652 Min. :-6.700
## 1st Qu.:735 1st Qu.: 7.30 Class :character 1st Qu.: 2.400
## Median :735 Median :12.80 Mode :character Median : 7.300
## Mean :735 Mean :13.44 Mean : 7.324
## 3rd Qu.:735 3rd Qu.:19.40 3rd Qu.:12.400
## Max. :735 Max. :31.90 Max. :23.500
##
## horatmin tmax horatmax dir
## Length:3652 Min. :-1.10 Length:3652 Min. : 1.00
## Class :character 1st Qu.:11.80 Class :character 1st Qu.:10.00
## Mode :character Median :18.90 Mode :character Median :25.00
## Mean :19.56 Mean :24.29
## 3rd Qu.:26.90 3rd Qu.:28.00
## Max. :41.10 Max. :99.00
## NA's :4
## velmedia racha horaracha
## Min. :0.300 Min. : 2.200 Length:3652
## 1st Qu.:1.400 1st Qu.: 6.100 Class :character
## Median :1.700 Median : 8.300 Mode :character
## Mean :2.005 Mean : 8.552
## 3rd Qu.:2.500 3rd Qu.:10.300
## Max. :8.300 Max. :22.800
## NA's :5 NA's :4
```

Estudiamos los valores nulos de las variables

```
apply(is.na(Vel_vien_Vall),2,sum)
```

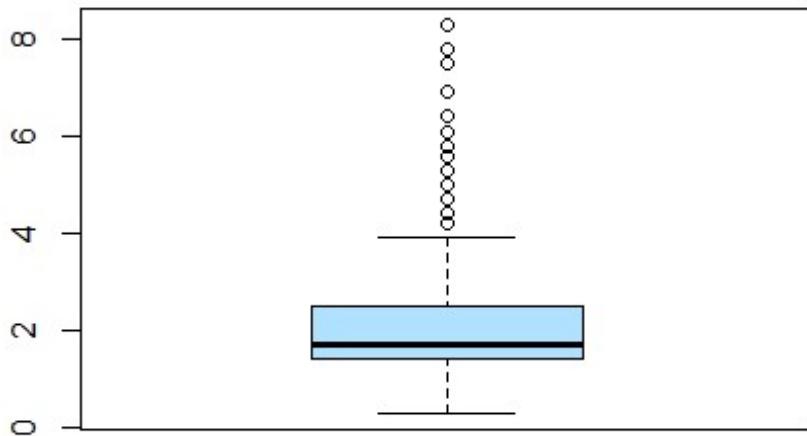
```
## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 0 0
## tmin horatmin tmax horatmax dir velmedia racha
## 0 0 0 0 4 5 4
## horaracha
## 0
```

Sucede lo mismo que con la temperatura ya que los datos se obtienen de la misma fuente (AEMET). Las columnas que nos interesan son la “velmedia”, la fecha y la localización de la estación donde se han tomado los datos meteorológicos. El número de registros es correcto, teniendo uno por día en todo el período estudiado. El número de datos faltantes en la velocidad del viento es despreciable.

Estudiamos ahora los Outliers

```
boxplot(Vel_vien_Vall$velmedia, col = 'lightskyblue1', main="Outliers variable Velocidad viento Valladolid")
```

Outliers variable Velocidad viento Valladolid



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'velmedia' es:", format(length(boxplot.stats(Vel_vien_Vall$velmedia)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'velmedia' es: 196"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Vel_vien_Vall$velmedia)$out)/length(Vel_vien_Vall$velmedia)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 5.37 %"
```

Comprobamos que los outliers son despreciables en esta variable.

La segunda ciudad será ALBACETE

Cargamos el archivo con los datos

```
Vel_vien_Alb <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER /11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.2.Vel viento Albacete 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(Vel_vien_Alb)
```

```
## fecha indicativo nombre provincia altitud tmed prec tmin horatmin tmax
## 1 01/01/2012 8178D ALBACETE ALBACETE 674 10.2 0 2.1 8:00 18.2
## 2 02/01/2012 8178D ALBACETE ALBACETE 674 8.9 0 4.9 23:59 12.9
## 3 03/01/2012 8178D ALBACETE ALBACETE 674 7.2 0 0.2 7:00 14.1
## 4 04/01/2012 8178D ALBACETE ALBACETE 674 7.8 0 0.4 7:00 15.3
## 5 05/01/2012 8178D ALBACETE ALBACETE 674 10.0 0 4.1 Varias 15.8
## 6 06/01/2012 8178D ALBACETE ALBACETE 674 10.9 0 5.0 23:20 16.8
## horatmax dir velmedia racha horaracha
## 1 14:23 29 0.3 6.4 23:59
## 2 13:10 28 3.6 14.2 10:50
## 3 14:00 30 1.4 6.4 12:50
```

```

## 4 14:20 29 1.7 7.8 14:50
## 5 14:50 29 2.8 9.7 13:50
## 6 14:00 29 2.8 10.0 13:10

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Vel_vien_AlB)

## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: chr "8178D" "8178D" "8178D" "8178D" ...
## $ nombre : chr "ALBACETE" "ALBACETE" "ALBACETE" "ALBACETE" ...
## $ provincia : chr "ALBACETE" "ALBACETE" "ALBACETE" "ALBACETE" ...
## $ altitud : int 674 674 674 674 674 674 674 674 674 ...
## $ tmed : num 10.2 8.9 7.2 7.8 10 10.9 7.2 6.6 6.6 8.5 ...
## $ prec : chr "0" "0" "0" "0" ...
## $ tmin : num 2.1 4.9 0.2 0.4 4.1 5 0.3 -2.4 -2.1 3.1 ...
## $ horatmin : chr "8:00" "23:59" "7:00" "7:00" ...
## $ tmax : num 18.2 12.9 14.1 15.3 15.8 16.8 14.2 15.5 15.3 13.9 ...
## $ horatmax : chr "14:23" "13:10" "14:00" "14:20" ...
## $ dir : int 29 28 30 29 29 29 NA 21 4 6 ...
## $ velmedia : num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ racha : num 6.4 14.2 6.4 7.8 9.7 10 NA 3.6 6.1 8.3 ...
## $ horaracha : chr "23:59" "10:50" "12:50" "14:50" ...

print('-----')
## [1] "-----"

summary(Vel_vien_AlB)

## fecha indicativo nombre provincia
## Length:3652 Length:3652 Length:3652 Length:3652
## Class :character Class :character Class :character Class :character
## Mode :character Mode :character Mode :character Mode :character
##
## 
## 
## 
## altitud tmed prec tmin
## Min. :674 Min. :-4.00 Length:3652 Min. :-11.300
## 1st Qu.:674 1st Qu.: 9.40 Class :character 1st Qu.: 3.800
## Median :674 Median :15.20 Mode :character Median : 9.300
## Mean :674 Mean :15.99 Mean : 9.555
## 3rd Qu.:674 3rd Qu.:22.80 3rd Qu.: 15.400
## Max. :674 Max. :34.10 Max. : 26.100
##
## horatmin tmax horatmax dir
## Length:3652 Min. :0.40 Length:3652 Min. : 0.00
## Class :character 1st Qu.:14.70 Class :character 1st Qu.:14.00
## Mode :character Median :21.90 Mode :character Median :26.00
## Mean :22.42 Mean :25.87
## 3rd Qu.:30.20 3rd Qu.:28.00
## Max. :42.70 Max. :99.00
## NA's :19
## velmedia racha horaracha

```

```
## Min. :0.000 Min. :0.000 Length:3652
## 1st Qu.:0.600 1st Qu.: 6.700 Class :character
## Median :1.100 Median : 8.300 Mode :character
## Mean :1.359 Mean : 8.592
## 3rd Qu.:1.900 3rd Qu.:10.000
## Max. :7.200 Max. :26.400
## NA's :16 NA's :19
```

Estudiamos los valores nulos de las variables
`apply(is.na(Vel_vien_Alb),2,sum)`

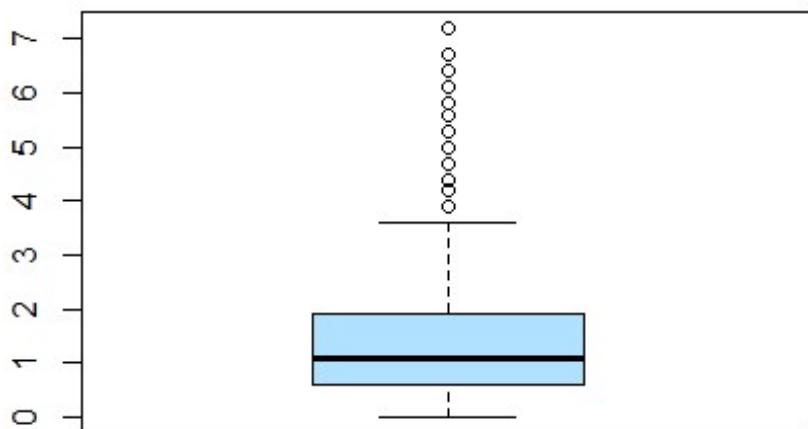
```
## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 0 0
## tmin horatmin tmax horatmax dir velmedia racha
## 0 0 0 0 19 16 19
## horaracha
## 0
```

Sucede lo mismo que con los datos de Valladolid.

Estudiamos ahora los Outliers

`boxplot(Vel_vien_Alb$velmedia, col = 'lightskyblue1', main="Outliers variable Velocidad viento Albacete")`

Outliers variable Velocidad viento Albacete



Obtenemos el porcentaje de outliers de la variable que nos interesa

`print(paste("El número de Outliers existentes en la columna 'velmedia' es:", format(length(boxplot.stats(Vel_vien_Alb$velmedia)$out))))`

[1] "El número de Outliers existentes en la columna 'velmedia' es: 101"

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Vel_vien_Alb$velmedia)$out)/length(Vel_vien_Alb$velmedia)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 2.77 %"
```

Comprobamos que los outliers son despreciables en esta variable.

La siguiente ciudad será ZARAGOZA

```
# Cargamos el archivo con los datos
```

```
Vel_vien_Zar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER /11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.3.Vel viento Zaragoza 2012-2022.csv', header = T, sep = ',', encoding = "UTF-8")
```

```
head(Vel_vien_Zar)
```

```
##   fecha indicativo      nombre provincia altitud tmed prec tmin
## 1 01/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 11.8  0 5.4
## 2 02/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 9.6  0.7 5.4
## 3 03/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 7.6  0 2.6
## 4 04/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 9.6  0 3.6
## 5 05/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 10.6 0 5.4
## 6 06/01/2012    9434 ZARAGOZA, AEROPUERTO ZARAGOZA  249 11.9 0 8.4
##   horatmin tmax horatmax dir velmedia racha horaracha
## 1    7:00 18.2 14:40 27   1.7  9.7  0:10
## 2   23:00 13.7 15:00 24   4.4 16.9  0:50
## 3   5:10 12.5 14:45 27   4.4  8.3 19:20
## 4   5:10 15.5 14:40 29   6.9 15.0 20:20
## 5   8:10 15.7 15:50 30   7.8 19.2 23:25
## 6  23:59 15.4 13:15 29  12.8 20.6  7:00
```

```
# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
```

```
str(Vel_vien_Zar)
```

```
## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 9434 9434 9434 9434 9434 9434 9434 9434 9434 ...
## $ nombre : chr "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" "ZARAGOZA, AEROPUERTO" ...
## $ provincia : chr "ZARAGOZA" "ZARAGOZA" "ZARAGOZA" "ZARAGOZA" ...
## $ altitud : int 249 249 249 249 249 249 249 249 249 ...
## $ tmed : num 11.8 9.6 7.6 9.6 10.6 11.9 11.2 10.3 8.4 ...
## $ prec : chr "0" "0.7" "0" "0" ...
## $ tmin : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ horatmin : chr "7:00" "23:00" "5:10" "5:10" ...
## $ tmax : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ horatmax : chr "14:40" "15:00" "14:45" "14:40" ...
## $ dir : int 27 24 27 29 30 29 29 29 29 25 ...
## $ velmedia : num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ racha : num 9.7 16.9 8.3 15 19.2 20.6 16.9 18.1 12.5 4.7 ...
## $ horaracha : chr "0:10" "0:50" "19:20" "20:20" ...
```

```
print('-----')
```

```
## [1] "-----"
```

```
summary(Vel_vien_Zar)
```

```

## fecha indicativo nombre provincia
## Length:3652 Min. :9434 Length:3652 Length:3652
## Class :character 1st Qu.:9434 Class :character Class :character
## Mode :character Median :9434 Mode :character Mode :character
## Mean :9434
## 3rd Qu.:9434
## Max. :9434
##
## altitud tmed prec tmin
## Min. :249 Min. :-0.80 Length:3652 Min. :-5.00
## 1st Qu.:249 1st Qu.:10.30 Class :character 1st Qu.: 5.60
## Median :249 Median :16.00 Mode :character Median :10.60
## Mean :249 Mean :16.54 Mean :10.93
## 3rd Qu.:249 3rd Qu.:22.80 3rd Qu.:16.50
## Max. :249 Max. :33.80 Max. :24.80
##
## horatmin tmax horatmax dir
## Length:3652 Min. :0.40 Length:3652 Min. : 2.00
## Class :character 1st Qu.:14.90 Class :character 1st Qu.:24.00
## Mode :character Median :21.70 Mode :character Median :29.00
## Mean :22.14 Mean :32.27
## 3rd Qu.:29.30 3rd Qu.:31.00
## Max. :44.50 Max. :99.00
## NA's :1
## velmedia racha horaracha
## Min. : 0.000 Min. : 1.9 Length:3652
## 1st Qu.: 2.500 1st Qu.: 8.9 Class :character
## Median : 3.900 Median :12.2 Mode :character
## Mean : 4.644 Mean :12.5
## 3rd Qu.: 6.400 3rd Qu.:15.6
## Max. :15.300 Max. :37.5
## NA's :1 NA's :1

```

Estudiamos los valores nulos de las variables

```
apply(is.na(Vel_vien_Zar),2,sum)
```

```

## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 0 0
## tmin horatmin tmax horatmax dir velmedia racha
## 0 0 0 0 1 1 1
## horaracha
## 0

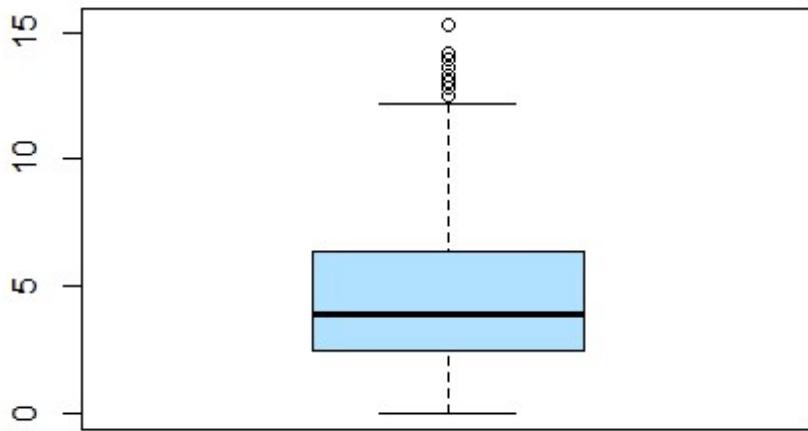
```

Sucede lo mismo que con las anteriores ciudades

Estudiamos ahora los Outliers

```
boxplot(Vel_vien_Zar$velmedia, col = 'lightskyblue1', main="Outliers variable Velocidad viento Zara goza")
```

Outliers variable Velocidad viento Zaragoza



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'velmedia' es:", format(length(boxplot.stats(Vel_vien_Zar$velmedia)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'velmedia' es: 22"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Vel_vien_Zar$velmedia)$out)/length(Vel_vien_Zar$velmedia)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0.6 %"
```

Comprobamos que no existen outliers en esta variable.

La siguiente ciudad será LA CORUÑA

Cargamos el archivo con los datos

```
Vel_vien_Cor <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER /11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.4.Vel viento Coruña 2012-2022.csv', header = T, sep = ',', encoding = "UTF-8")
```

```
head(Vel_vien_Cor)
```

```
## fecha indicativo nombre provincia altitud tmed prec tmin horatmin
## 1 01/01/2012 1387 A CORUNIA A CORUNIA 58 14.1 15.9 12.2 23:59
## 2 02/01/2012 1387 A CORUNIA A CORUNIA 58 11.6 1.9 9.3 8:20
## 3 03/01/2012 1387 A CORUNIA A CORUNIA 58 12.8 2.5 10.5 0:50
## 4 04/01/2012 1387 A CORUNIA A CORUNIA 58 13.4 0.9 12.0 23:59
## 5 05/01/2012 1387 A CORUNIA A CORUNIA 58 12.4 1.6 11.2 5:20
## 6 06/01/2012 1387 A CORUNIA A CORUNIA 58 12.2 1.9 11.2 Varias
## tmax horatmax dir velmedia racha horaracha presMax horaPresMax presMin
## 1 16.0 12:40 3.5 NA 1019.2 0 1009.5
## 2 13.8 14:10 2.4 NA 1022.7 22 1013.4
## 3 15.2 11:40 20 5.0 13.9 17:50 1024.1 24 1021.4
```

```

## 4 14.8 0:00 24 2.2 11.4 2:30 1028.7 21 1024.1
## 5 13.6 13:50 27 3.1 8.9 14:10 1027.9 Varias 1026.5
## 6 13.2 14:10 36 2.5 7.5 8:00 1027.9 11 1025.1
## horaPresMin sol
## 1 15 0.9
## 2 0 4.4
## 3 Varias 1.0
## 4 0 0.0
## 5 15 0.0
## 6 Varias 0.0

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Vel_vien_Cor)

## 'data.frame': 3652 obs. of 20 variables:
## $ fecha    : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo : int 1387 1387 1387 1387 1387 1387 1387 1387 1387 1387 ...
## $ nombre   : chr "A CORUNIA" "A CORUNIA" "A CORUNIA" "A CORUNIA" ...
## $ provincia : chr "A CORUNIA" "A CORUNIA" "A CORUNIA" "A CORUNIA" ...
## $ altitud   : int 58 58 58 58 58 58 58 58 58 58 ...
## $ tmed     : num 14.1 11.6 12.8 13.4 12.4 12.2 12.6 11.4 10.9.1 ...
## $ prec     : chr "15.9" "1.9" "2.5" "0.9" ...
## $ tmin     : num 12.2 9.3 10.5 12 11.2 11.2 10.2 7.9 5.2 5 ...
## $ horatmin : chr "23:59" "8:20" "0:50" "23:59" ...
## $ tmax     : num 16 13.8 15.2 14.8 13.6 13.2 14.9 15 14.9 13.2 ...
## $ horatmax : chr "12:40" "14:10" "11:40" "0:00" ...
## $ dir      : chr "" "" "20" "24" ...
## $ velmedia : num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ racha    : num NA NA 13.9 11.4 8.9 7.5 10.8 7.8 6.4 5.3 ...
## $ horaracha: chr "" "" "17:50" "2:30" ...
## $ presMax  : num 1019 1023 1024 1029 1028 ...
## $ horaPresMax: chr "0" "22" "24" "21" ...
## $ presMin  : chr "1009.5" "1013.4" "1021.4" "1024.1" ...
## $ horaPresMin: chr "15" "0" "Varias" "0" ...
## $ sol      : num 0.9 4.4 1 0 0 0 2.5 8.5 8.6 8.2 ...

print("-----")
## [1] "-----"

summary(Vel_vien_Cor)

## fecha      indicativo     nombre      provincia
## Length:3652  Min. :1387  Length:3652  Length:3652
## Class :character 1st Qu.:1387  Class :character  Class :character
## Mode  :character Median :1387  Mode :character Mode :character
##                  Mean :1387
##                  3rd Qu.:1387
##                  Max. :1387
##
## altitud      tmed      prec      tmin
## Min. :58  Min. :4.40  Length:3652  Min. :1.00
## 1st Qu.:58  1st Qu.:12.10  Class :character 1st Qu.: 9.40
## Median :58  Median :15.20  Mode :character Median :12.30
## Mean  :58  Mean  :15.25           Mean  :12.17

```

```

## 3rd Qu.:58 3rd Qu.:18.40          3rd Qu.:15.40
## Max. :58  Max. :26.40          Max. :20.60
##      NA's :1             NA's :1
## horatmin     tmax    horatmax     dir
## Length:3652   Min. :6.90  Length:3652   Length:3652
## Class :character 1st Qu.:14.80  Class :character  Class :character
## Mode :character Median :18.00  Mode :character Mode :character
##                  Mean :18.32
##                  3rd Qu.:21.60
##                  Max. :33.60
##
## velmedia     racha    horaracha    presMax
## Min. :0.600  Min. : 3.9  Length:3652   Min. : 0.8
## 1st Qu.: 2.200 1st Qu.: 15.6  Class :character 1st Qu.: 2.8
## Median : 3.300  Median:1005.6  Mode :character Median : 4.4
## Mean : 3.611  Mean :709.6        Mean :308.9
## 3rd Qu.: 4.700 3rd Qu.:1011.8        3rd Qu.:1005.8
## Max. :11.100  Max. :1033.0        Max. :1033.7
## NA's :40  NA's :16            NA's :38
## horaPresMax   presMin    horaPresMin    sol
## Length:3652   Length:3652   Length:3652   Min. :0.00
## Class :character  Class :character  Class :character 1st Qu.: 6.00
## Mode :character  Mode :character  Mode :character Median :18.00
##                  Mean :20.03
##                  3rd Qu.:28.00
##                  Max. :99.00
## NA's :48

```

Estudiamos los valores nulos de las variables

```
apply(is.na(Vel_vien_Cor),2,sum)
```

```

## fecha indicativo nombre provincia altitud tméd
##      0      0      0      0      0      1
## prec tmin horatmin     tmax horatmax     dir
##      0      1      0      0      0      0
## velmedia     racha    horaracha    presMax horaPresMax    presMin
##      40      16      0      38      0      0
## horaPresMin    sol
##      0      48

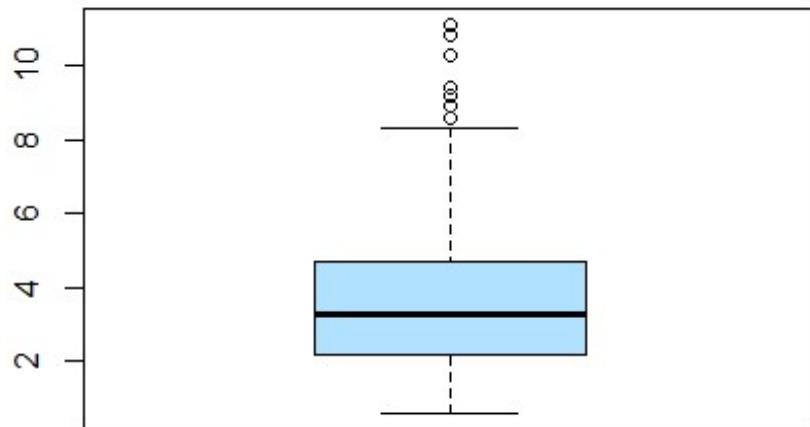
```

Sucede lo mismo que con las anteriores ciudades

Estudiamos ahora los Outliers

```
boxplot(Vel_vien_Cor$velmedia, col = 'lightskyblue1', main="Outliers variable Velocidad viento Coruña")
```

Outliers variable Velocidad viento Coruña



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'velmedia' es:", format(length(boxplot.stats(Vel_vien_Cor$velmedia)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'velmedia' es: 22"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Vel_vien_Cor$velmedia)$out)/length(Vel_vien_Cor$velmedia)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0.6 %"
```

Comprobamos que los outliers son despreciables en esta variable.

La ultima ciudad es HUELVA

Cargamos el archivo con los datos

```
Vel_vien_Hue <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTE
R/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivo
s csv y excel/8.5.Vel viento Huelva 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
head(Vel_vien_Hue)
```

```
## fecha indicativo nombre provincia altitud tmed prec tmin
## 1 01/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 9.6 0 3.1
## 2 02/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 13.1 0 7.9
## 3 03/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 11.6 0 4.9
## 4 04/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 10.8 0 2.1
## 5 05/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 13.4 0 4.9
## 6 06/01/2012 4642 HUELVA, RONDA ESTE HUELVA 18 10.8 0 4.2
## horatmin tmax horatmax dir velmedia racha horaracha
## 1 7:30 16 15:30 29 1.4 7.5 16:00
## 2 23:59 18.3 14:30 36 1.4 7.8 14:10
## 3 6:50 18.3 16:00 13 1.4 5.6 13:40
```

```

## 4 7:50 19.6 15:10 34 1.4 7.2 20:20
## 5 4:30 21.9 14:32 7 1.9 8.3 22:30
## 6 6:00 17.4 16:00 34 1.9 8.1 19:30

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Vel_vien_Hue)

## 'data.frame': 3643 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: num 4642 4642 4642 4642 4642 ...
## $ nombre : chr "HUELVA, RONDA ESTE" "HUELVA, RONDA ESTE" "HUELVA, RONDA ESTE" "H
UELVA, RONDA ESTE" ...
## $ provincia : chr "HUELVA" "HUELVA" "HUELVA" "HUELVA" ...
## $ altitud : int 18 18 18 18 18 18 18 18 18 ...
## $ tmed : chr "9.6" "13.1" "11.6" "10.8" ...
## $ prec : chr "0" "0" "0" "0" ...
## $ tmin : chr "3.1" "7.9" "4.9" "2.1" ...
## $ horatmin : chr "7:30" "23:59" "6:50" "7:50" ...
## $ tmax : chr "16" "18.3" "18.3" "19.6" ...
## $ horatmax : chr "15:30" "14:30" "16:00" "15:10" ...
## $ dir : chr "29" "36" "13" "34" ...
## $ velmedia : num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ racha : chr "7.5" "7.8" "5.6" "7.2" ...
## $ horaracha : chr "16:00" "14:10" "13:40" "20:20" ...

print('-----')
## [1] "-----"

summary(Vel_vien_Hue)

## fecha indicativo nombre provincia
## Length:3643 Min. :4642 Length:3643 Length:3643
## Class :character 1st Qu.:4642 Class :character Class :character
## Mode :character Median :4642 Mode :character Mode :character
## Mean :4642
## 3rd Qu.:4642
## Max. :4642
## altitud tmed prec tmin
## Min. :18 Length:3643 Length:3643 Length:3643
## 1st Qu.:18 Class :character Class :character Class :character
## Median :18 Mode :character Mode :character Mode :character
## Mean :18
## 3rd Qu.:18
## Max. :18
## horatmin tmax horatmax dir
## Length:3643 Length:3643 Length:3643 Length:3643
## Class :character Class :character Class :character Class :character
## Mode :character Mode :character Mode :character Mode :character
## 
## 
## 
## velmedia racha horaracha
## Min. :0.300 Length:3643 Length:3643
## 1st Qu.:2.500 Class :character Class :character

```

```
## Median :3.300 Mode :character Mode :character
## Mean :3.246
## 3rd Qu.:3.900
## Max. :9.400

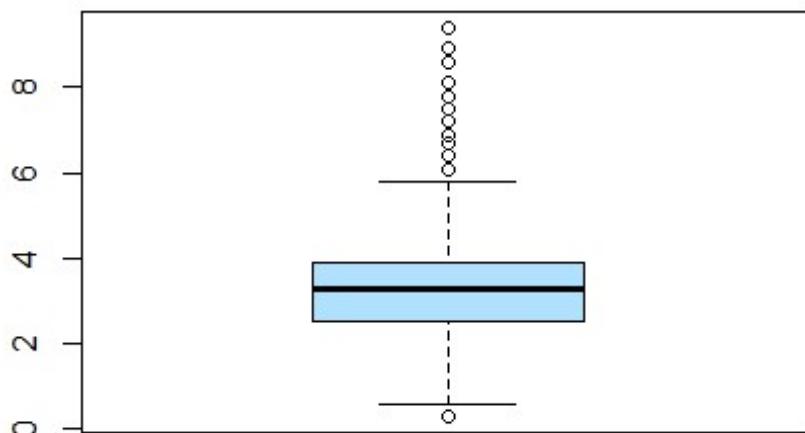
# Estudiamos los valores nulos de las variables
apply(is.na(Vel_vien_Hue),2,sum)

## fecha indicativo nombre provincia altitud tmed prec
## 0 0 0 0 0 0 0
## tmin horatmin tmax horatmax dir velmedia racha
## 0 0 0 0 0 0 0
## horaracha
## 0
```

Sucede lo mismo que con las anteriores ciudades

```
# Estudiamos ahora los Outliers
boxplot(Vel_vien_Hue$velmedia, col = 'lightskyblue1', main="Outliers variable Velocidad viento Huelva")
```

Outliers variable Velocidad viento Huelva



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'velmedia' es:", format(length(boxplot.stats(Vel_vien_Hue$velmedia)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'velmedia' es: 76"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Vel_vien_Hue$velmedia)$out)/length(Vel_vien_Hue$velmedia)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 2.09 %"
```

Comprobamos que los outliers son despreciables en esta variable.

6.5.9 La siguiente variable es “Reservas hidráulicas (Res_hidr)”

Cargamos el archivo con los datos

```
Res_hidr <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.  
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv  
y excel/9.Reservas hidraulicas 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")  
head(Res_hidr)
```

```
##     ambito    embalse   fecha agua_total agua_actual  
## 1 Minio - Sil Albarellos 01/01/2012     91      29  
## 2 Minio - Sil      Bao 01/01/2012    238     169  
## 3 Minio - Sil     Belesar 01/01/2012    655     182  
## 4 Minio - Sil Campaniana, La 01/01/2012     14      10  
## 5 Minio - Sil     Castrelo 01/01/2012     60      54  
## 6 Minio - Sil      Cenza 01/01/2012     40      12  
## agua_almacenada...  
## 1      31.87  
## 2      71.01  
## 3      27.79  
## 4      71.43  
## 5      90.00  
## 6      30.00
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Res_hidr)

```
## 'data.frame': 48205 obs. of 6 variables:  
## $ ambito      : chr "Minio - Sil" "Minio - Sil" "Minio - Sil" "Minio - Sil" ...  
## $ embalse     : chr "Albarellos" "Bao" "Belesar" "Campaniana, La" ...  
## $ fecha       : chr "01/01/2012" "01/01/2012" "01/01/2012" "01/01/2012" ...  
## $ agua_total   : int 91 238 655 14 60 40 61 80 14 44 ...  
## $ agua_actual  : int 29 169 182 10 54 12 17 28 4 41 ...  
## $ agua_almacenada...: num 31.9 71 27.8 71.4 90 ...  
  
print('-----')  
## [1] -----  
  
summary(Res_hidr)  
  
##     ambito    embalse   fecha   agua_total  
##  Length:48205  Length:48205  Length:48205  Min. : 5.0  
##  Class :character  Class :character  Class :character  1st Qu.: 14.0  
##  Mode  :character  Mode :character  Mode :character  Median : 34.0  
##                                         Mean :187.7  
##                                         3rd Qu.:123.0  
##                                         Max. :3160.0  
## agua_actual  agua_almacenada...  
##  Min. : 0.0  Min. : 0.00  
##  1st Qu.: 9.0 1st Qu.: 54.15  
##  Median : 25.0 Median : 78.23  
##  Mean   : 126.5 Mean   : 70.54  
##  3rd Qu.:104.0 3rd Qu.: 90.00  
##  Max. :3127.0 Max. :100.52
```

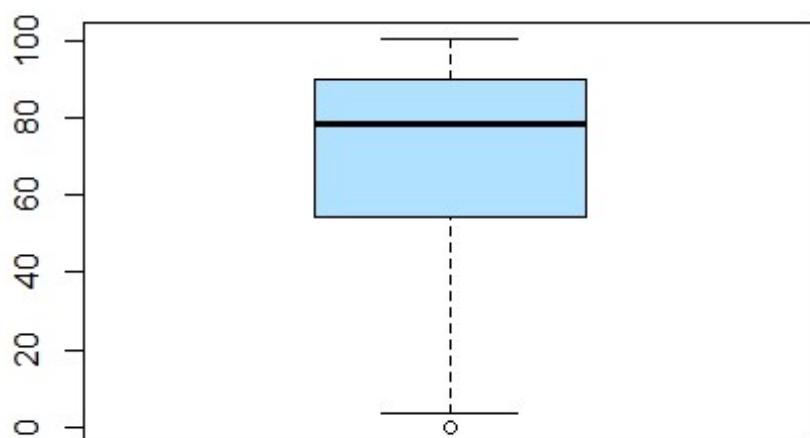
```
# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'agua_almacenada' es:", format(mean(is.na(Res_hidr$agua_almacenada)))))

## [1] "El número de NA's de la columna 'agua_almacenada' es: 0"
```

En esta ocasión aunque tenemos ya el porcentaje de llenada de cada embalse lo que nos interesará será el porcentaje general, por lo que deberemos calcularlo a partir de las columnas de "agua_total" y "agua_actual". Aunque es una variable que está medida de forma temporal cada semana, el número de registros es muy elevado debido a que existe un gran cantidad de embalses productores de EE en nuestro país.

```
# Estudiamos ahora los Outliers
boxplot(Res_hidr$agua_almacenada, col = 'lightskyblue1', main="Outliers variable agua almacenada")
```

Outliers variable agua almacenada



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'Agua_almacenada' es:", format(length(boxplot.stats(Res_hidr$agua_almacenada)$out))))

## [1] "El número de Outliers existentes en la columna 'Agua_almacenada' es: 104"

print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Res_hidr$agua_almacenada)$out)/length(Res_hidr$agua_almacenada)*100,2)), "%"))

## [1] "Y el porcentaje: 0.22 %"
```

Comprobamos que los outliers son despreciables en esta variable.

6.5.10 La siguiente variable es "Precio Derechos de emisión de CO2 (Der_CO2)"

Cargamos el archivo con los datos

```
Der_CO2 <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.  
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv  
y excel/10.Precio derechos emision CO2 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")  
head(Der_CO2)
```

```
## fecha ultimo apertura maximo minimo volumen variacion  
## 1 01/01/2012 7.89 6.66 7.02 6.66 0.40 2.01%  
## 2 02/01/2012 7.14 7.14 7.14 7.14 NA 2.51%  
## 3 03/01/2012 6.28 6.60 6.60 6.37 0.31 -12.11%  
## 4 04/01/2012 6.27 6.15 6.15 6.15 2.20 -0.16%  
## 5 05/01/2012 6.43 6.44 6.50 6.28 0.30 2.55%  
## 6 06/01/2012 6.30 6.22 6.22 6.18 0.66 -2.02%
```

Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(Der_CO2)

```
## 'data.frame': 2589 obs. of 7 variables:  
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...  
## $ ultimo : num 7.89 7.14 6.28 6.27 6.43 6.3 6.51 6.84 6.75 6.92 ...  
## $ apertura : num 6.66 7.14 6.6 6.15 6.44 6.22 6.51 6.85 6.8 7.02 ...  
## $ maximo : num 7.02 7.14 6.66 6.15 6.5 6.22 6.51 7.02 6.89 7.02 ...  
## $ minimo : num 6.66 7.14 6.37 6.15 6.28 6.18 6.51 6.8 6.74 6.95 ...  
## $ volumen : num 0.4 NA 0.31 2.2 0.3 0.66 0.16 0.38 0.16 0.67 ...  
## $ variacion: chr "2.01%" "2.51%" "-12.11%" "-0.16%" ...
```

```
print('-----')
```

```
## [1] -----"
```

summary(Der_CO2)

```
## fecha ultimo apertura maximo  
## Length:2589 Min. :2.70 Min. :2.71 Min. :2.87  
## Class:character 1st Qu.: 5.72 1st Qu.: 5.74 1st Qu.: 5.76  
## Mode:character Median :7.72 Median :7.72 Median :7.78  
## Mean :21.51 Mean :21.49 Mean :21.81  
## 3rd Qu.:26.28 3rd Qu.:26.36 3rd Qu.:26.45  
## Max. :98.01 Max. :98.50 Max. :99.22  
##  
## minimo volumen variacion  
## Min. :2.49 Min. :0.000 Length:2589  
## 1st Qu.: 5.70 1st Qu.: 0.090 Class:character  
## Median : 7.65 Median : 0.590 Mode:character  
## Mean :21.18 Mean :4.317  
## 3rd Qu.:26.18 3rd Qu.: 3.100  
## Max. :95.60 Max. :78.660  
## NA's :28
```

Estudiamos los valores nulos de las columnas que nos interesan

```
print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(Der_CO2$ultimo)))))
```

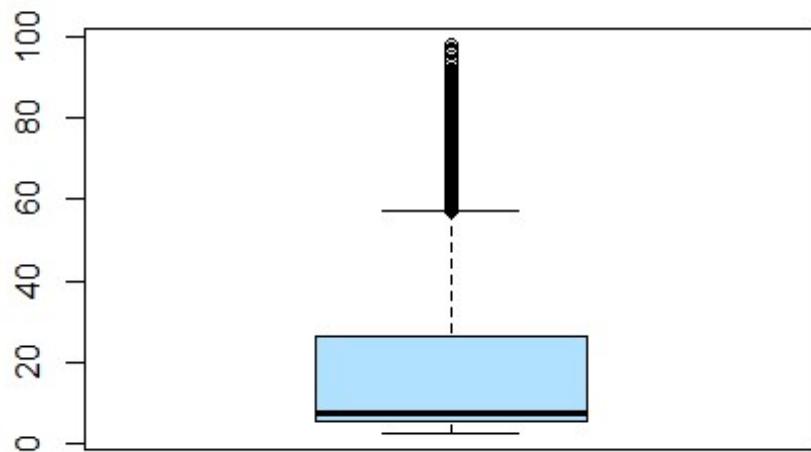
```
## [1] "El número de NA's de la columna 'ultimo' es: 0"
```

De este dataframe nos interesa solo el “precio ultimo” y la “fecha”. Segun el numero de registros vemos que la cotizacion de estos derechos se negocian de lunes a viernes. No tenemos valores faltantes.

Estudiamos ahora los Outliers

```
boxplot(Der_CO2$ultimo, col = 'lightskyblue1', main="Outliers variable Precio Derechos de emision CO2")
```

Outliers variable Precio Derechos de emision CO2



Obtenemos el porcentaje de outliers de la variable que nos interesa

```
print(paste("El número de Outliers existentes en la columna 'ultimo' es:", format(length(boxplot.stats(Der_CO2$ultimo)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'ultimo' es: 357"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Der_CO2$ultimo)$out)/length(Der_CO2$ultimo)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 13.79 %"
```

Comprobamos que los outliers son despreciables en esta variable.

6.5.11 La siguiente variable es “Situación socio-económica del país (Ibex)”

Cargamos el archivo con los datos

```
Ibex <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/11.Indice Ibex35 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
head(Ibex)
```

```
## fecha ultimo apertura maximo minimo volumen variacion
```

```
## 1 01/01/2012 8327.2 8539.4 8581.8 8411.0 231.35 -1.34%
```

```
## 2 02/01/2012 8723.8 8558.0 8724.2 8557.4 102.66 1.84%
```

```

## 3 03/01/2012 8732.4 8739.2 8743.3 8597.1 186.69 0.10%
## 4 04/01/2012 8581.8 8683.4 8701.3 8526.8 243.8 -1.72%
## 5 05/01/2012 8329.6 8598.7 8598.7 8301.2 192.7 -2.94%
## 6 06/01/2012 8289.1 8369.7 8445.9 8233.8 153.76 -0.49%

# Comprobamos la estructura del dataset y obtenemos los valores estadísticos
str(ibex)

## 'data.frame': 2559 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 8327 8724 8732 8582 8330 ...
## $ apertura : num 8539 8558 8739 8683 8599 ...
## $ maximo : num 8582 8724 8743 8701 8599 ...
## $ minimo : num 8411 8557 8597 8527 8301 ...
## $ volumen : chr "231.35" "102.66" "186.69" "243.8" ...
## $ variacion: chr "-1.34%" "1.84%" "0.10%" "-1.72%" ...

print(-----)
## [1] "-----"

summary(ibex)

##   fecha      ultimo     apertura     maximo
##  Length:2559   Min.   :5956   Min.   :5950   Min.   :6093
##  Class :character 1st Qu.:8478   1st Qu.:8484   1st Qu.:8555
##  Mode  :character Median :9114   Median :9113   Median :9163
##                Mean   :9211   Mean   :9215   Mean   :9277
##                3rd Qu.:10092  3rd Qu.:10091  3rd Qu.:10158
##                Max.  :11866  Max.  :11798  Max.  :11885
##   minimo     volumen     variacion
##  Min.   :5905  Length:2559  Length:2559
##  1st Qu.:8410  Class :character Class :character
##  Median :9049  Mode  :character Mode  :character
##  Mean   :9142
##  3rd Qu.:10006
##  Max.  :11761

# Estudiamos los valores nulos de las columnas que nos interesan
print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(ibex$ultimo)))))

## [1] "El número de NA's de la columna 'ultimo' es: 0"

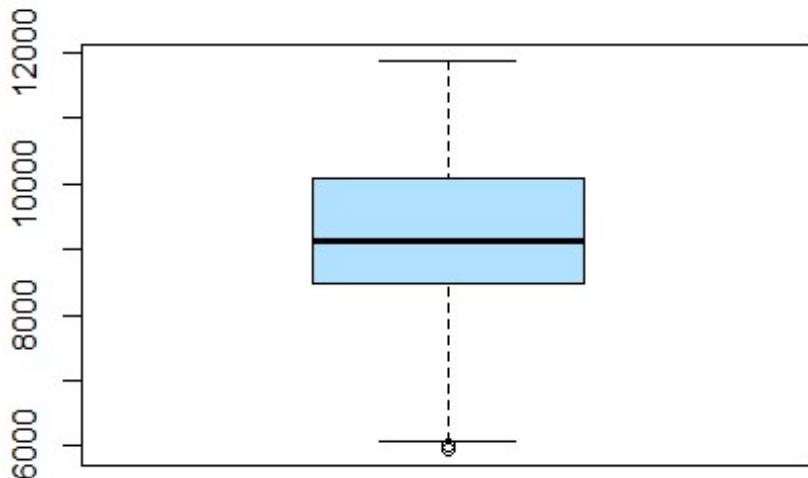
```

Es el mismo caso que el anterior en que la cotización se produce de lunes a viernes y nos interesan solo la “fecha” y el “precio ultimo”. No tenemos valores faltantes.

Estudiamos ahora los Outliers

```
boxplot(ibex$ultimo, col = 'lightskyblue1', main="Outliers variable Indice Ibex35")
```

Outliers variable Índice Ibex35



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'ultimo' es:", format(length(boxplot.stats(Ibex$ultimo)$out))))
```

```
## [1] "El número de Outliers existentes en la columna 'ultimo' es: 2"
```

```
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Ibex$ultimo)$out)/length(Ibex$ultimo)*100,2)), "%"))
```

```
## [1] "Y el porcentaje: 0.08 %"
```

Comprobamos que no existen outliers en esta variable.

6.5.12 La siguiente variable es “Intercambio de EE con otros países (Int_ee)”

```
# Cargamos el archivo con los datos
Int_ee <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/12.Intercambio de EE paises 2012-2022.csv', header = T, sep = ',', encoding = "UTF-8")
head(Int_ee)
```

```
##   fecha exportacion importacion saldo
## 1 01/01/2012 -52979.48  36053.57 -16925.91
## 2 02/01/2012 -57162.03  23083.65 -34078.38
## 3 03/01/2012 -60439.32  33641.79 -26797.53
## 4 04/01/2012 -55014.41  25486.19 -29528.22
## 5 05/01/2012 -60318.75  22250.09 -38068.67
## 6 06/01/2012 -67317.28  11436.08 -55881.20
```

```
# Comprobamos la estructura del dataset y obtenemos los valores estadísticos  
str(Int_ee)
```

```
## 'data.frame': 3652 obs. of 4 variables:  
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...  
## $ exportacion: num -52979 -57162 -60439 -55014 -60319 ...  
## $ importacion: num 36054 23084 33642 25486 22250 ...  
## $ saldo : num -16926 -34078 -26798 -29528 -38069 ...
```

```
print(-----)
```

```
## [1] -----"
```

```
summary(Int_ee)
```

```
## fecha exportacion importacion saldo  
## Length:3652 Min. :-128163 Min. : 346.5 Min. :-127715  
## Class :character 1st Qu.: -54290 1st Qu.: 26212.2 1st Qu.: -25759  
## Mode :character Median : -43085 Median : 38836.8 Median : -3787  
## Mean : -45091 Mean : 43549.9 Mean : -1541  
## 3rd Qu.: -32180 3rd Qu.: 60837.7 3rd Qu.: 26981  
## Max. : -1827 Max. : 124817.0 Max. : 101780
```

```
# Estudiamos los valores nulos de las columnas que nos interesan
```

```
print(paste("El número de NA's de la columna 'ultimo' es:", format(mean(is.na(Int_ee$saldo)))))
```

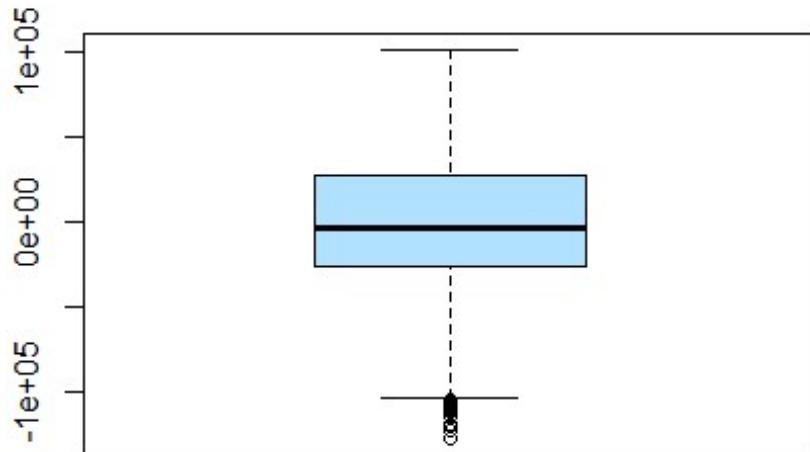
```
## [1] "El número de NA's de la columna 'ultimo' es: 0"
```

En este caso las columnas que nos interesan para el estudio serán “fecha” y “saldo”. Y tenemos registros todos los días de la semana. No tenemos valores faltantes.

```
# Estudiamos ahora los Outliers
```

```
boxplot(Int_ee$saldo, col = 'lightskyblue1', main="Outliers variable Intercambio EE")
```

Outliers variable Intercambio EE



```
# Obtenemos el porcentaje de outliers de la variable que nos interesa
print(paste("El número de Outliers existentes en la columna 'Saldo' es:", format(length(boxplot.stats(Int_ee$saldo)$out))))  
  
## [1] "El número de Outliers existentes en la columna 'Saldo' es: 24"  
  
print(paste("Y el porcentaje:", format(round(length(boxplot.stats(Int_ee$saldo)$out)/length(Int_ee$saldo)*100,2)), "%"))  
  
## [1] "Y el porcentaje: 0.66 %"
```

Comprobamos que los outliers son despreciables en esta variable.

ANEXO III UNION Y TRATAMIENTO DE LOS DATOS

[Formacion del dataset del estudio](#)

Iñigo Elorza Barea

2023-07-10

6.5. TRATAMIENTO DE LOS DATOS

Cargamos las librerías necesarias:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
library(tibble)
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
library(tidyr)
```

```
library(readr)
```

```
library(stringr)
```

Cargamos todos los archivos necesarios

```
Pre_elec <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/1.Precio medio diario EE 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Dem <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/2.Demanda media diaria 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Prec_petr <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/3.Precio petróleo 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Prec_gas <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/4.Precio Gas Natural 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Prec_carb <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/5.Precio Carbon 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Prod_tec <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/6.Producción por tecnologías 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```
Temp_Mad <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.1.Temp Madrid 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
```

```

Temp_Bar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.
1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.2.Temp Barcelona 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Temp_Val <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.
1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.3.Temp Valencia 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Temp_Sev <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.
1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.4.Temp Sevilla 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Temp_Zar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/1.
1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.5.Temp Zaragoza 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Vel_vien_Vall <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.1.Vel viento Valladolid 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Vel_vien_Alb <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.2.Vel viento Albacete 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Vel_vien_Zar <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.3.Vel viento Zaragoza 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Vel_vien_Cor <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.4.Vel viento Coruña 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Vel_vien_Hue <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/8.5.Vel viento Huelva 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Res_hidr <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/9.Reservas hidraulicas 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Der_CO2 <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/10.Precio derechos emision CO2 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Ibex <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/11.Indice Ibex35 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")
Int_ee <- read.table('D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.
MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/12.Intercambio de EE paises 2012-2022.csv', header = T, sep = ';', encoding = "UTF-8")

```

Para formar el dataset final con el que trabajaremos en los modelos de predicción, partiendo del primer dataframe del archivo “Precio medio diario EE 2012-2022” e iremos uniendo de Unimos en Unimos los datos de los demás archivos. Aprovecharemos para hacer las trasformaciones necesarias.

6.5.1 Unimos las variables “fecha”, “Prec_elec”, “dia_semana” y “Dem”

```

# Limpiamos el primer archivo
df <- select(Pre_elec, datetime, value)
df <- rename(df, fecha = datetime, Pre_elec = value)
df$fecha <- as.Date(df$fecha)
head(df)

##     fecha Pre_elec
## 1 2012-01-01 48.1992

```

```

## 2 2012-01-02 49.7375
## 3 2012-01-03 57.1317
## 4 2012-01-04 54.6721
## 5 2012-01-05 51.5471
## 6 2012-01-06 52.4642

df <- mutate(df, dia_sem = weekdays(df$fecha))
df <- df[,c(1, 3, 2)]

df$dia_sem <- str_replace_all(df$dia_sem, c("lunes" = "1", "martes" = "2", "miércoles" = "3", "jueves" = "4", "viernes" = "5", "sábado" = "6", "domingo" = "7"))

# Limpiamos el segundo archivo
Dem <- select(Dem, datetime, value)
Dem <- rename(Dem, fecha = datetime, Dem = value)
Dem$fecha <- as.Date(Dem$fecha)
head(Dem)

##     fecha      Dem
## 1 2012-01-01 20888.99
## 2 2012-01-02 23325.20
## 3 2012-01-03 26994.32
## 4 2012-01-04 27934.39
## 5 2012-01-05 28089.09
## 6 2012-01-06 24915.97

# Unimos los dos dataframes
df <- merge(df, Dem, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 4 variables:
##   $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
##   $ dia_sem : chr "7" "1" "2" "3" ...
##   $ Pre_elec: num 48.2 49.7 57.1 54.7 51.5 ...
##   $ Dem    : num 20889 23325 26994 27934 28089 ...

```

6.5.2 Unimos al dataframe creado la variable “Prec_petr”

```

str(Prec_petr)

## 'data.frame': 2889 obs. of 7 variables:
##   $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
##   $ ultimo : num 111 108 112 114 113 ...
##   $ apertura : num 110 111 108 112 114 ...
##   $ maximo : num 111 111 112 114 115 ...
##   $ minimo : num 110 108 108 111 112 ...
##   $ volumen : num 211 190 179 221 205 ...
##   $ variacion: chr "1.62%" "-2.26%" "4.42%" "1.40%" ...

names(Prec_petr)

## [1] "fecha"   "ultimo"   "apertura" "maximo"   "minimo"   "volumen"
## [7] "variacion"

```

```

# Limpiamos el segundo archivo
Prec_petr <- select(Prec_petr, fecha, ultimo)
Prec_petr <- rename(Prec_petr, Prec_petr = ultimo)
Prec_petr$fecha <- as.Date(Prec_petr$fecha, format = "%d/%m/%Y")
head(Prec_petr)

##    fecha Prec_petr
## 1 2012-01-01 110.85
## 2 2012-01-02 108.35
## 3 2012-01-03 112.13
## 4 2012-01-04 113.70
## 5 2012-01-05 112.74
## 6 2012-01-06 113.06

# Unimos con el dataframe inicial
df <- merge(df, Prec_petr, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 5 variables:
## $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem : chr "7" "1" "2" "3" ...
## $ Pre_elec : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr: num 111 108 112 114 113 ...

head(df)

##    fecha dia_sem Pre_elec Dem Prec_petr
## 1 2012-01-01     7 48.1992 20888.99 110.85
## 2 2012-01-02     1 49.7375 23325.20 108.35
## 3 2012-01-03     2 57.1317 26994.32 112.13
## 4 2012-01-04     3 54.6721 27934.39 113.70
## 5 2012-01-05     4 51.5471 28089.09 112.74
## 6 2012-01-06     5 52.4642 24915.97 113.06

```

6.5.3 Unimos ahora la variable "Prec_gas"

```

str(Prec_gas)

## 'data.frame': 2541 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 55.6 53.8 52.8 53.1 53 ...
## $ apertura : num 55.2 55.6 53.8 52.8 53.6 ...
## $ maximo : num 55.6 55.9 53.8 53.5 53.9 ...
## $ minimo : num 55.1 54 52.5 52.8 52.9 ...
## $ volumen : num 4.95 5.26 5.36 4.96 5.23 5.98 9.09 8.26 6.17 2.45 ...
## $ variacion: num -0.015 -0.0232 -0.0287 0.0064 -0.0026 -0.0015 0.0373 -0.0067 0.018 -0.0069 .
..

names(Prec_gas)

## [1] "fecha"   "ultimo"   "apertura" "maximo"   "minimo"   "volumen"
## [7] "variacion"

```

Limpiamos el archivo

```
Prec_gas <- select(Prec_gas, fecha, ultimo)
Prec_gas <- rename(Prec_gas, Prec_gas = ultimo)
Prec_gas$fecha <- as.Date(Prec_gas$fecha, format = "%d/%m/%Y")
head(Prec_gas)
```

```
##   fecha Prec_gas
## 1 2012-01-01 55.60
## 2 2012-01-02 53.75
## 3 2012-01-03 52.75
## 4 2012-01-04 53.09
## 5 2012-01-05 52.95
## 6 2012-01-06 52.87
```

Unimos con el dataframe inicial

```
df <- merge(df, Prec_gas, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 6 variables:
## $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem : chr "7" "1" "2" "3" ...
## $ Pre_elec : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr: num 111 108 112 114 113 ...
## $ Prec_gas : num 55.6 53.8 52.8 53.1 53 ...
```

```
head(df)
```

```
##   fecha dia_sem Pre_elec Dem Prec_petr Prec_gas
## 1 2012-01-01    7 48.1992 20888.99  110.85 55.60
## 2 2012-01-02    1 49.7375 23325.20  108.35 53.75
## 3 2012-01-03    2 57.1317 26994.32  112.13 52.75
## 4 2012-01-04    3 54.6721 27934.39  113.70 53.09
## 5 2012-01-05    4 51.5471 28089.09  112.74 52.95
## 6 2012-01-06    5 52.4642 24915.97  113.06 52.87
```

6.5.4 Unimos ahora la variable “Prec_carb”

```
str(Prec_carb)
```

```
## 'data.frame': 2817 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 112 110 109 110 110 ...
## $ apertura : num 111 112 109 110 110 ...
## $ maximo : num 112 112 109 110 110 ...
## $ minimo : num 111 110 109 110 110 ...
## $ volumen : num 0.13 0.05 0.12 0.13 0.05 0.2 0.15 0.02 0.08 0.1 ...
## $ variacion: chr "-0.86" "-1.22%" "-0.77%" "0.18%" ...
```

```
names(Prec_carb)
```

```
## [1] "fecha"   "ultimo"   "apertura" "maximo"   "minimo"   "volumen"
## [7] "variacion"
```

Limpiamos el archivo

```
Prec_carb <- select(Prec_carb, fecha, ultimo)
Prec_carb <- rename(Prec_carb, Prec_carb = ultimo)
Prec_carb$fecha <- as.Date(Prec_carb$fecha, format = "%d/%m/%Y")
head(Prec_carb)
```

```
##   fecha Prec_carb
## 1 2012-01-01 111.56
## 2 2012-01-02 110.20
## 3 2012-01-03 109.35
## 4 2012-01-04 109.55
## 5 2012-01-05 110.10
## 6 2012-01-06 110.20
```

Unimos con el dataframe inicial

```
df <- merge(df, Prec_carb, all.x = TRUE)
str(df)
```

```
## 'data.frame': 3652 obs. of 7 variables:
## $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem : chr "7" "1" "2" "3" ...
## $ Pre_elec : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr: num 111 108 112 114 113 ...
## $ Prec_gas : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb: num 112 110 109 110 110 ...
```

```
head(df)
```

```
##   fecha dia_sem Pre_elec Dem Prec_petr Prec_gas Prec_carb
## 1 2012-01-01    7 48.1992 20888.99 110.85  55.60 111.56
## 2 2012-01-02    1 49.7375 23325.20 108.35  53.75 110.20
## 3 2012-01-03    2 57.1317 26994.32 112.13  52.75 109.35
## 4 2012-01-04    3 54.6721 27934.39 113.70  53.09 109.55
## 5 2012-01-05    4 51.5471 28089.09 112.74  52.95 110.10
## 6 2012-01-06    5 52.4642 24915.97 113.06  52.87 110.20
```

6.5.5 Unimos ahora la variable "Prod_tec"

```
str(Prod_tec)
```

```
## 'data.frame': 3652 obs. of 19 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ Eolica : num 155 261 165 164 204 ...
## $ Hidroelotica : num NA NA NA NA NA NA NA NA NA ...
## $ Solar_fotovoltaica : num 16.6 12.8 17.7 17.6 17.6 ...
## $ Solar_termica : num 3.34 1.9 4.54 4.96 5.1 ...
## $ Hidraulica : num 44.1 48.2 58.4 63.7 56.6 ...
## $ Turbinacion_bombeo : num 7.59 8.84 12.35 13.22 14.93 ...
## $ Otras_renovables : num 10.3 10.3 10.1 10.5 10.7 ...
## $ Residuos_renovables : num 2.27 2.24 2.27 2.28 2.33 ...
## $ Nuclear : num 156 156 156 156 156 ...
## $ Motores_diesel : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Turbina_de_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Carbon : num 95.8 118.3 169.2 176.3 140.1 ...
```

```

## $ Ciclo_combinado    : num 64 61.8 99.8 102.4 82.2 ...
## $ Cogeneracion       : num 61.1 86 95.7 96.8 95.9 ...
## $ Fuel.Gas            : num 0.02409 0.0101 0.00782 0.00962 0.00954 ...
## $ Turbina_de_vapor   : num 5.93 7.4 7.74 7.88 8.12 ...
## $ Residuos_no_renovables: num 3.56 3.64 3.59 4.01 4.53 ...
## $ Generacion_total    : num 636 790 814 832 810 ...

names(Prod_tec)

## [1] "fecha"          "Eolica"          "Hidroelica"
## [4] "Solar_fotovoltaica" "Solar_termica"    "Hidraulica"
## [7] "Turbinacion_bombeo"  "Otras_renovables" "Residuos_renovables"
## [10] "Nuclear"         "Motores_diesel"   "Turbina_de_gas"
## [13] "Carbon"          "Ciclo_combinado" "Cogeneracion"
## [16] "Fuel.Gas"         "Turbina_de_vapor" "Residuos_no_renovables"
## [19] "Generacion_total"

# Eliminamos la ultima columna que es la suma de todas las producciones y le damos formato a la columna "fecha" para que la reconozca correctamente
Prod_tec$Generacion_total <- NULL
Prod_tec$fecha <- as.Date(Prod_tec$fecha, format = "%d/%m/%Y")

# Renombramos las variables
Prod_tec <- rename(Prod_tec, Prod_eol = Eolica,
                    Prod_hidroel = Hidroelica,
                    Prod_sol_fot = Solar_fotovoltaica,
                    Prod_sol_ter = Solar_termica,
                    Prod_hidr = Hidraulica,
                    Prod_hidr_tur = Turbinacion_bombeo,
                    Prod_ofr = Otras_renovables,
                    Prod_nucl = Nuclear,
                    Prod_pet = Motores_diesel,
                    Prod_gas = Turbina_de_gas,
                    Prod_carb = Carbon,
                    Prod_comb = Ciclo_combinado,
                    Prod_cog = Cogeneracion,
                    Prod_fuel_gas = Fuel.Gas,
                    Prod_tur_vap = Turbina_de_vapor,
                    Prod_noren = Residuos_no_renovables)
names(Prod_tec)

## [1] "fecha"          "Prod_eol"        "Prod_hidroel"
## [4] "Prod_sol_fot"   "Prod_sol_ter"    "Prod_hidr"
## [7] "Prod_hidr_tur"  "Prod_ofr"        "Residuos_renovables"
## [10] "Prod_nucl"       "Prod_pet"        "Prod_gas"
## [13] "Prod_carb"      "Prod_comb"       "Prod_cog"
## [16] "Prod_fuel_gas"  "Prod_tur_vap"    "Prod_noren"

# Antes de unir todas estas variables vamos a reducir el número sumando algunas cuya tecnología es equiparable
Prod_tec$Prod_eol <- rowSums(Prod_tec[, c('Prod_eol','Prod_hidroel')], na.rm=TRUE)
Prod_tec$Prod_hidroel <- NULL
head(Prod_tec)

```

```

##    fecha Prod_eol Prod_sol_fot Prod_sol_ter Prod_hidr Prod_hidr_tur
## 1 2012-01-01 155.1421 16.55900 3.342588 44.11561 7.586468
## 2 2012-01-02 261.0320 12.82085 1.896593 48.18073 8.837414
## 3 2012-01-03 164.6931 17.66322 4.537361 58.38051 12.350879
## 4 2012-01-04 164.3039 17.64268 4.962424 63.71691 13.218982
## 5 2012-01-05 204.1075 17.57371 5.104392 56.62367 14.932920
## 6 2012-01-06 215.8213 14.44478 3.273886 55.10793 9.088064
## Prod_ofr Residuos_renovables Prod_nucl Prod_pet Prod_gas Prod_carb Prod_comb
## 1 10.28479 2.270296 156.0602 9.099916 1.211399 95.83518 63.99322
## 2 10.25505 2.238878 155.9083 9.585639 1.852946 118.34087 61.77065
## 3 10.13681 2.265258 155.9070 9.693961 2.357607 169.17696 99.75440
## 4 10.54009 2.282968 155.9415 9.460365 2.276651 176.29588 102.35277
## 5 10.66351 2.327133 155.8380 9.619993 1.970738 140.08747 82.15043
## 6 10.91022 2.318175 156.1161 9.145990 1.218538 86.14172 54.24204
## Prod_cog Prod_fuel_gas Prod_tur_vap Prod_noren
## 1 61.13340 0.024088 5.926551 3.558299
## 2 86.04903 0.010099 7.399333 3.636452
## 3 95.67452 0.007815 7.737631 3.591249
## 4 96.84716 0.009619 7.881408 4.013747
## 5 95.93271 0.009536 8.117390 4.531231
## 6 86.23782 0.024052 6.464696 4.759207

Prod_tec$Prod_sol_fot <- rowSums(Prod_tec[, c('Prod_sol_fot','Prod_sol_ter')], na.rm=TRUE)
Prod_tec <- rename(Prod_tec, Prod_sol = Prod_sol_fot)
Prod_tec$Prod_sol_ter <- NULL
head(Prod_tec)

##    fecha Prod_eol Prod_sol Prod_hidr Prod_hidr_tur Prod_ofr
## 1 2012-01-01 155.1421 19.90159 44.11561 7.586468 10.28479
## 2 2012-01-02 261.0320 14.71745 48.18073 8.837414 10.25505
## 3 2012-01-03 164.6931 22.20058 58.38051 12.350879 10.13681
## 4 2012-01-04 164.3039 22.60510 63.71691 13.218982 10.54009
## 5 2012-01-05 204.1075 22.67810 56.62367 14.932920 10.66351
## 6 2012-01-06 215.8213 17.71867 55.10793 9.088064 10.91022
## Residuos_renovables Prod_nucl Prod_pet Prod_gas Prod_carb Prod_comb Prod_cog
## 1 2.270296 156.0602 9.099916 1.211399 95.83518 63.99322 61.13340
## 2 2.238878 155.9083 9.585639 1.852946 118.34087 61.77065 86.04903
## 3 2.265258 155.9070 9.693961 2.357607 169.17696 99.75440 95.67452
## 4 2.282968 155.9415 9.460365 2.276651 176.29588 102.35277 96.84716
## 5 2.327133 155.8380 9.619993 1.970738 140.08747 82.15043 95.93271
## 6 2.318175 156.1161 9.145990 1.218538 86.14172 54.24204 86.23782
## Prod_fuel_gas Prod_tur_vap Prod_noren
## 1 0.024088 5.926551 3.558299
## 2 0.010099 7.399333 3.636452
## 3 0.007815 7.737631 3.591249
## 4 0.009619 7.881408 4.013747
## 5 0.009536 8.117390 4.531231
## 6 0.024052 6.464696 4.759207

Prod_tec$Prod_hidr <- rowSums(Prod_tec[, c('Prod_hidr','Prod_hidr_tur')], na.rm=TRUE)
Prod_tec$Prod_hidr_tur <- NULL
head(Prod_tec)

##    fecha Prod_eol Prod_sol Prod_hidr Prod_ofr Residuos_renovables Prod_nucl
## 1 2012-01-01 155.1421 19.90159 51.70207 10.28479 2.270296 156.0602

```

```

## 2 2012-01-02 261.0320 14.71745 57.01815 10.25505      2.238878 155.9083
## 3 2012-01-03 164.6931 22.20058 70.73139 10.13681      2.265258 155.9070
## 4 2012-01-04 164.3039 22.60510 76.93590 10.54009      2.282968 155.9415
## 5 2012-01-05 204.1075 22.67810 71.55659 10.66351      2.327133 155.8380
## 6 2012-01-06 215.8213 17.71867 64.19600 10.91022      2.318175 156.1161
## Prod_pet Prod_gas Prod_carb Prod_comb Prod_cog Prod_fuel_gas Prod_tur_vap
## 1 9.099916 1.211399 95.83518 63.99322 61.13340     0.024088 5.926551
## 2 9.585639 1.852946 118.34087 61.77065 86.04903     0.010099 7.399333
## 3 9.693961 2.357607 169.17696 99.75440 95.67452     0.007815 7.737631
## 4 9.460365 2.276651 176.29588 102.35277 96.84716     0.009619 7.881408
## 5 9.619993 1.970738 140.08747 82.15043 95.93271     0.009536 8.117390
## 6 9.145990 1.218538 86.14172 54.24204 86.23782     0.024052 6.464696
## Prod_noren
## 1 3.558299
## 2 3.636452
## 3 3.591249
## 4 4.013747
## 5 4.531231
## 6 4.759207

Prod_tec$Prod_ofr <- rowSums(Prod_tec[, c('Prod_ofr','Residuos_renovables')], na.rm=TRUE)
Prod_tec$Residuos_renovables <- NULL
head(Prod_tec)

## fecha Prod_eol Prod_sol Prod_hidr Prod_ofr Prod_nucl Prod_pet Prod_gas
## 1 2012-01-01 155.1421 19.90159 51.70207 12.55508 156.0602 9.099916 1.211399
## 2 2012-01-02 261.0320 14.71745 57.01815 12.49393 155.9083 9.585639 1.852946
## 3 2012-01-03 164.6931 22.20058 70.73139 12.40207 155.9070 9.693961 2.357607
## 4 2012-01-04 164.3039 22.60510 76.93590 12.82306 155.9415 9.460365 2.276651
## 5 2012-01-05 204.1075 22.67810 71.55659 12.99064 155.8380 9.619993 1.970738
## 6 2012-01-06 215.8213 17.71867 64.19600 13.22839 156.1161 9.145990 1.218538
## Prod_carb Prod_comb Prod_cog Prod_fuel_gas Prod_tur_vap Prod_noren
## 1 95.83518 63.99322 61.13340     0.024088 5.926551 3.558299
## 2 118.34087 61.77065 86.04903     0.010099 7.399333 3.636452
## 3 169.17696 99.75440 95.67452     0.007815 7.737631 3.591249
## 4 176.29588 102.35277 96.84716     0.009619 7.881408 4.013747
## 5 140.08747 82.15043 95.93271     0.009536 8.117390 4.531231
## 6 86.14172 54.24204 86.23782     0.024052 6.464696 4.759207

Prod_tec$Prod_fuel_gas <- rowSums(Prod_tec[, c('Prod_fuel_gas','Prod_tur_vap', 'Prod_noren')], na.rm=TRUE)
Prod_tec <- rename(Prod_tec, Prod_no_ren = Prod_fuel_gas)
Prod_tec$Prod_tur_vap <- NULL
Prod_tec$Prod_noren <- NULL
head(Prod_tec)

## fecha Prod_eol Prod_sol Prod_hidr Prod_ofr Prod_nucl Prod_pet Prod_gas
## 1 2012-01-01 155.1421 19.90159 51.70207 12.55508 156.0602 9.099916 1.211399
## 2 2012-01-02 261.0320 14.71745 57.01815 12.49393 155.9083 9.585639 1.852946
## 3 2012-01-03 164.6931 22.20058 70.73139 12.40207 155.9070 9.693961 2.357607
## 4 2012-01-04 164.3039 22.60510 76.93590 12.82306 155.9415 9.460365 2.276651
## 5 2012-01-05 204.1075 22.67810 71.55659 12.99064 155.8380 9.619993 1.970738
## 6 2012-01-06 215.8213 17.71867 64.19600 13.22839 156.1161 9.145990 1.218538
## Prod_carb Prod_comb Prod_cog Prod_no_ren
## 1 95.83518 63.99322 61.13340     9.508938

```

```

## 2 118.34087 61.77065 86.04903 11.045884
## 3 169.17696 99.75440 95.67452 11.336695
## 4 176.29588 102.35277 96.84716 11.904774
## 5 140.08747 82.15043 95.93271 12.658157
## 6 86.14172 54.24204 86.23782 11.247955

# Ahora unimos este dataframe con el principal
df <- merge(df, Prod_tec, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 18 variables:
## $ fecha    : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol  : num 155 261 165 164 204 ...
## $ Prod_sol  : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr  : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet  : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas  : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog  : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren: num 9.51 11.05 11.34 11.9 12.66 ...

```

6.5.6 Unimos ahora las variables de Temperatura

MADRID

```
str(Temp_Mad)
```

```

## 'data.frame': 3651 obs. of 15 variables:
## $ fecha    : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 3195 3195 3195 3195 3195 3195 3195 3195 3195 ...
## $ nombre    : chr "MADRID, RETIRO" "MADRID, RETIRO" "MADRID, RETIRO" "MADRID, RETIRO" ...
## $ provincia : chr "MADRID" "MADRID" "MADRID" "MADRID" ...
## $ altitud   : int 667 667 667 667 667 667 667 667 667 ...
## $ tmed     : num 7 8.7 6.6 7.6 9 8.2 7.3 7 6.3 ...
## $ prec      : chr "Ip" "0.1" "0" "0" ...
## $ tmin     : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ horatmin : chr "7:30" "23:59" "7:40" "7:30" ...
## $ tmax     : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ horatmax : chr "16:10" "14:25" "14:55" "15:30" ...
## $ dir      : chr "25" "25" "23" "12" ...
## $ velmedia : chr "1.1" "3.9" "0.8" "0.8" ...
## $ racha    : num 7.2 12.5 3.9 5.8 4.7 11.4 5.3 3.6 5.3 6.1 ...
## $ horaracha: chr "23:59" "11:40" "15:10" "3:40" ...

```

```
names(Temp_Mad)
```

```

## [1] "fecha"    "indicativo" "nombre"    "provincia" "altitud"
## [6] "tmed"     "prec"      "tmin"      "horatmin"  "tmax"
## [11] "horatmax" "dir"       "velmedia"   "racha"    "horaracha"

# Limpiamos el dataframe
Temp_Mad <- select(Temp_Mad, fecha, tmin, tmax)
Temp_Mad <- rename(Temp_Mad, Temp_min_Mad = tmin, Temp_max_Mad = tmax)
Temp_Mad$fecha <- as.Date(Temp_Mad$fecha, format = "%d/%m/%Y")
head(Temp_Mad)

##    fecha Temp_min_Mad Temp_max_Mad
## 1 2012-01-01    2.3     11.6
## 2 2012-01-02    5.8     11.6
## 3 2012-01-03    1.5     10.5
## 4 2012-01-04    0.5     11.6
## 5 2012-01-05    2.2     13.0
## 6 2012-01-06    3.0     15.0

# Unimos con el dataframe inicial
df <- merge(df, Temp_Mad, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 20 variables:
## $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem : chr "7" "1" "2" "3" ...
## $ Pre_elec : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol : num 155 261 165 164 204 ...
## $ Prod_sol : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...

```

BARCELONA

```

# Limpiamos el dataframe BARCELONA
Temp_Bar <- select(Temp_Bar, fecha, tmin, tmax)
Temp_Bar <- rename(Temp_Bar, Temp_min_Bar = tmin, Temp_max_Bar = tmax)
Temp_Bar$fecha <- as.Date(Temp_Bar$fecha, format = "%d/%m/%Y")
head(Temp_Bar)

##    fecha Temp_min_Bar Temp_max_Bar
## 1 2012-01-12    5.7     14.8
## 2 2012-01-13    6.0     15.6
## 3 2012-01-14    4.5     13.4

```

```

## 4 2012-01-15    7.1    12.4
## 5 2012-01-16    7.8    12.2
## 6 2012-01-17    9.9    13.8

# Unimos con el dataframe inicial
df <- merge(df, Temp_Bar, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 22 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem    : chr "7" "1" "2" "3" ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr  : num 111 108 112 114 113 ...
## $ Prec_gas   : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb  : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren: num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...

```

VALENCIA

```

# Limpiamos el dataframe VALENCIA
Temp_Val <- select(Temp_Val, fecha, tmin, tmax)
Temp_Val <- rename(Temp_Val, Temp_min_Val = tmin, Temp_max_Val = tmax)
Temp_Val$fecha <- as.Date(Temp_Val$fecha, format = "%d/%m/%Y")
head(Temp_Val)

##     fecha Temp_min_Val Temp_max_Val
## 1 2012-01-01    10.0    21.0
## 2 2012-01-02    10.3    20.4
## 3 2012-01-03     7.2    19.7
## 4 2012-01-04     6.7    22.4
## 5 2012-01-05     9.2    25.6
## 6 2012-01-06    11.4    23.0

# Unimos con el dataframe inicial
df <- merge(df, Temp_Val, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 24 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem    : chr "7" "1" "2" "3" ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...

```

```
## $ Dem      : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val: num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val: num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
```

SEVILLA

```
# Limpiamos el dataframe SEVILLA
Temp_Sev <- select(Temp_Sev, fecha, tmin, tmax)
Temp_Sev <- rename(Temp_Sev, Temp_min_Sev = tmin, Temp_max_Sev = tmax)
Temp_Sev$fecha <- as.Date(Temp_Sev$fecha, format = "%d/%m/%Y")
head(Temp_Sev)

##    fecha Temp_min_Sev Temp_max_Sev
## 1 2012-01-01     2.4     17.4
## 2 2012-01-02     7.5     19.0
## 3 2012-01-03     4.2     17.7
## 4 2012-01-04     4.0     18.8
## 5 2012-01-05     5.2     22.4
## 6 2012-01-06     1.4     19.1

# Unimos con el dataframe inicial
df <- merge(df, Temp_Sev, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 26 variables:
## $ fecha    : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
```

```
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val: num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val: num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev: num 2.4 7.5 4.2 4.5 2.1 4.7 4.6 8.3 9.4 4.4 ...
## $ Temp_max_Sev: num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
```

ZARAGOZA*# Limpiamos el dataframe ZARAGOZA*Temp_Zar <- **select**(Temp_Zar, fecha, tmin, tmax)Temp_Zar <- **rename**(Temp_Zar, Temp_min_Zar = tmin, Temp_max_Zar = tmax)Temp_Zar\$fecha <- **as.Date**(Temp_Zar\$fecha, format = "%d/%m/%Y")**head**(Temp_Zar)

```
##   fecha Temp_min_Zar Temp_max_Zar
## 1 2012-01-01      5.4     18.2
## 2 2012-01-02      5.4     13.7
## 3 2012-01-03      2.6     12.5
## 4 2012-01-04      3.6     15.5
## 5 2012-01-05      5.4     15.7
## 6 2012-01-06      8.4     15.4
```

*# Unimos con el dataframe inicial*df <- **merge**(df, Temp_Zar, all.x = TRUE)**str**(df)

```
## 'data.frame': 3652 obs. of 28 variables:
##   $ fecha    : Date, format:"2012-01-01" "2012-01-02" ...
##   $ dia_sem   : chr "7" "1" "2" "3" ...
##   $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
##   $ Dem       : num 20889 23325 26994 27934 28089 ...
##   $ Prec_petr : num 111 108 112 114 113 ...
##   $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
##   $ Prec_carb : num 112 110 109 110 110 ...
##   $ Prod_eol  : num 155 261 165 164 204 ...
##   $ Prod_sol  : num 19.9 14.7 22.2 22.6 22.7 ...
##   $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
##   $ Prod_ofr  : num 12.6 12.5 12.4 12.8 13 ...
##   $ Prod_nucl : num 156 156 156 156 156 ...
##   $ Prod_pet  : num 9.1 9.59 9.69 9.46 9.62 ...
##   $ Prod_gas  : num 1.21 1.85 2.36 2.28 1.97 ...
##   $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
##   $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
##   $ Prod_cog  : num 61.1 86 95.7 96.8 95.9 ...
##   $ Prod_no_ren: num 9.51 11.05 11.34 11.9 12.66 ...
##   $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
##   $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
##   $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
```

```
## $ Temp_max_Bar: num NA ...
## $ Temp_min_Val: num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val: num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev: num 2.4 7.5 4.2 4.5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev: num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar: num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar: num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
```

6.5.7 Unimos ahora las variables de Velocidad del viento**VALLADOLID****str(Vel_vien_Vall)**

```
## 'data.frame': 3652 obs. of 15 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ indicativo: int 2422 2422 2422 2422 2422 2422 2422 2422 2422 ...
## $ nombre : chr "VALLADOLID" "VALLADOLID" "VALLADOLID" "VALLADOLID" ...
## $ provincia : chr "VALLADOLID" "VALLADOLID" "VALLADOLID" "VALLADOLID" ...
## $ altitud : int 735 735 735 735 735 735 735 735 735 ...
## $ tmed : num 4 5.7 3.2 5.3 3.2 4 5.7 0.2 2 0.8 ...
## $ prec : chr "0.6" "0" "0" "0" ...
## $ tmin : num 1.1 2.7 -1 2.6 1.7 0.3 1.1 -3.4 -0.9 -2.6 ...
## $ horatmin : chr "0:40" "23:59" "8:10" "Varias" ...
## $ tmax : num 7 8.7 7.3 8 4.8 7.7 10.3 3.8 4.8 4.3 ...
## $ horatmax : chr "16:30" "13:00" "16:00" "15:10" ...
## $ dir : int 99 24 18 20 25 19 1 20 6 99 ...
## $ velmedia : num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ racha : num 5.3 12.5 5.8 5.8 5.3 3.9 7.5 4.7 3.6 3.3 ...
## $ horaracha : chr "Varias" "3:50" "10:40" "5:30" ...
```

names(Vel_vien_Vall)

```
## [1] "fecha"    "indicativo" "nombre"    "provincia" "altitud"
## [6] "tmed"      "prec"       "tmin"      "horatmin"  "tmax"
## [11] "horatmax"  "dir"        "velmedia"   "racha"     "horaracha"
```

Limpiamos el dataframe VALLADOLID

```
Vel_vien_Vall <- select(Vel_vien_Vall, fecha, velmedia)
Vel_vien_Vall <- rename(Vel_vien_Vall, Vel_media_Val = velmedia)
Vel_vien_Vall$fecha <- as.Date(Vel_vien_Vall$fecha, format = "%d/%m/%Y")
head(Vel_vien_Vall)
```

```
##     fecha Vel_media_Val
## 1 2012-01-01      1.1
## 2 2012-01-02      5.0
## 3 2012-01-03      1.7
## 4 2012-01-04      1.4
## 5 2012-01-05      1.1
## 6 2012-01-06      1.1
```

Unimos con el dataframe inicial

```
df <- merge(df, Vel_vien_Vall, all.x = TRUE)
str(df)
```

```
## 'data.frame': 3652 obs. of 29 variables:
## $ fecha     : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol  : num 155 261 165 164 204 ...
## $ Prod_sol  : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr  : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet  : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas  : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog  : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren: num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val: num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val: num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev: num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev: num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar: num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar: num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
```

ALBACETE

```
# Limpiamos el dataframe ALBACETE
Vel_vien_AlB <- select(Vel_vien_AlB, fecha, velmedia)
Vel_vien_AlB <- rename(Vel_vien_AlB, Vel_media_AlB = velmedia)
Vel_vien_AlB$fecha <- as.Date(Vel_vien_AlB$fecha, format = "%d/%m/%Y")
head(Vel_vien_AlB)
```

```
##     fecha Vel_media_AlB
## 1 2012-01-01      0.3
## 2 2012-01-02      3.6
## 3 2012-01-03      1.4
## 4 2012-01-04      1.7
## 5 2012-01-05      2.8
## 6 2012-01-06      2.8
```

```
# Unimos con el dataframe inicial
df <- merge(df, Vel_vien_AlB, all.x = TRUE)
str(df)
```

```
## 'data.frame': 3652 obs. of 30 variables:
## $ fecha     : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
```

```
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol : num 155 261 165 164 204 ...
## $ Prod_sol : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
```

ZARAGOZA

Limpiamos el dataframe ZARAGOZA

```
Vel_vien_Zar <- select(Vel_vien_Zar, fecha, velmedia)
Vel_vien_Zar <- rename(Vel_vien_Zar, Vel_media_Zar = velmedia)
Vel_vien_Zar$fecha <- as.Date(Vel_vien_Zar$fecha, format = "%d/%m/%Y")
head(Vel_vien_Zar)
```

```
##     fecha Vel_media_Zar
## 1 2012-01-01      1.7
## 2 2012-01-02      4.4
## 3 2012-01-03      4.4
## 4 2012-01-04      6.9
## 5 2012-01-05      7.8
## 6 2012-01-06     12.8
```

Unimos con el dataframe inicial

```
df <- merge(df, Vel_vien_Zar, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 31 variables:
## $ fecha     : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol  : num 155 261 165 164 204 ...
```

```
## $ Prod_sol : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4.5 2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
```

CORUÑA

Limpiamos el dataframe CORUÑA

```
Vel_vien_Cor <- select(Vel_vien_Cor, fecha, velmedia)
Vel_vien_Cor <- rename(Vel_vien_Cor, Vel_media_Cor = velmedia)
Vel_vien_Cor$fecha <- as.Date(Vel_vien_Cor$fecha, format = "%d/%m/%Y")
head(Vel_vien_Cor)
```

```
##     fecha Vel_media_Cor
## 1 2012-01-01      3.5
## 2 2012-01-02      2.4
## 3 2012-01-03      5.0
## 4 2012-01-04      2.2
## 5 2012-01-05      3.1
## 6 2012-01-06      2.5
```

Unimos con el dataframe inicial

```
df <- merge(df, Vel_vien_Cor, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 32 variables:
## $ fecha     : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol  : num 155 261 165 164 204 ...
## $ Prod_sol  : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr  : num 12.6 12.5 12.4 12.8 13 ...
```

```
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
```

HUELVA

```
# Limpiamos el dataframe HUELVA
Vel_vien_Hue <- select(Vel_vien_Hue, fecha, velmedia)
Vel_vien_Hue <- rename(Vel_vien_Hue, Vel_media_Hue = velmedia)
Vel_vien_Hue$fecha <- as.Date(Vel_vien_Hue$fecha, format = "%d/%m/%Y")
head(Vel_vien_Hue)

##     fecha Vel_media_Hue
## 1 2012-01-01      1.4
## 2 2012-01-02      1.4
## 3 2012-01-03      1.4
## 4 2012-01-04      1.4
## 5 2012-01-05      1.9
## 6 2012-01-06      1.9

# Unimos con el dataframe inicial
df <- merge(df, Vel_vien_Hue, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 33 variables:
## $ fecha    : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol  : num 155 261 165 164 204 ...
## $ Prod_sol  : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr  : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet  : num 9.1 9.59 9.69 9.46 9.62 ...
```

```
## $ Prod_gas    : num  1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb   : num  95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num  64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog    : num  61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num  9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num  2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num  11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num  NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num  NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num  10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num  21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num  2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num  17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num  5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num  18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num  1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num  0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num  1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num  3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num  1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
```

6.5.8 Unimos ahora la variable "Res_hidr"

```
str(Res_hidr)
```

```
## 'data.frame': 48205 obs. of 6 variables:
## $ ambito      : chr "Minio - Sil" "Minio - Sil" "Minio - Sil" "Minio - Sil" ...
## $ embalse     : chr "Albarellos" "Bao" "Belesar" "Campaniana, La" ...
## $ fecha       : chr "01/01/2012" "01/01/2012" "01/01/2012" "01/01/2012" ...
## $ agua_total   : int 91 238 655 14 60 40 61 80 14 44 ...
## $ agua_actual  : int 29 169 182 10 54 12 17 28 4 41 ...
## $ agua_almacenada...: num 31.9 71 27.8 71.4 90 ...
```

```
names(Res_hidr)
```

```
## [1] "ambito"      "embalse"      "fecha"
## [4] "agua_total"  "agua_actual"  "agua_almacenada..."
```

Limpiamos el segundo archivo

```
Res_hidr <- select(Res_hidr, fecha, agua_total, agua_actual)
Res_hidr <- rename(Res_hidr, Res_total = agua_total, Res_act = agua_actual)
Res_hidr$fecha <- as.Date(Res_hidr$fecha, format = "%d/%m/%Y")
```

Antes de unir los datos de esta variable, hay que realizar algunas operaciones para unirlo correctamente.

Como tenemos un registro por cada embalse y semana, lo primero que hay que hacer es dejar solo un registro para todos los embalses que sume sus capacidades totales y sus niveles de llenado para cada registro semanal

```
Res_hidr <- Res_hidr %>% group_by(fecha) %>% summarise(Suma_total = sum(Res_total), Suma_a
ct = sum(Res_act))
```

Creamos una columna nueva para obtener el porcentaje de llenado de todos los embalses juntos

```
Res_hidr <- add_column(Res_hidr, Res_hidr = round(Res_hidr$Suma_act/Res_hidr$Suma_total*100,
2))
```

```

Res_hidr <- select(Res_hidr, fecha, Res_hidr)
head(Res_hidr)

## # A tibble: 6 × 2
##   fecha   Res_hidr
##   <date>     <dbl>
## 1 2012-01-01  56.8
## 2 2012-01-10  56.8
## 3 2012-01-17  56.2
## 4 2012-01-24  55.5
## 5 2012-01-31  55.2
## 6 2012-02-07  55.1

# Ahora ya lo podemos unir con el dataframe inicial
df <- merge(df, Res_hidr, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 34 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem    : chr "7" "1" "2" "3" ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr  : num 111 108 112 114 113 ...
## $ Prec_gas   : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb  : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ Res_hidr    : num 56.8 NA NA NA NA ...

```

6.5.9 Unimos ahora la variable "Der_CO2"

```
str(Der_CO2)
```

```
## 'data.frame': 2589 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 7.89 7.14 6.28 6.27 6.43 6.3 6.51 6.84 6.75 6.92 ...
## $ apertura : num 6.66 7.14 6.6 6.15 6.44 6.22 6.51 6.85 6.8 7.02 ...
## $ maximo : num 7.02 7.14 6.6 6.15 6.5 6.22 6.51 7.02 6.89 7.02 ...
## $ minimo : num 6.66 7.14 6.37 6.15 6.28 6.18 6.51 6.8 6.74 6.95 ...
## $ volumen : num 0.4 NA 0.31 2.2 0.3 0.66 0.16 0.38 0.16 0.67 ...
## $ variacion: chr "2.01%" "2.51%" "-12.11%" "-0.16%" ...
```

```
names(Der_CO2)
```

```
## [1] "fecha"   "ultimo"   "apertura" "maximo"   "minimo"   "volumen"
## [7] "variacion"
```

Limpiamos el segundo archivo

```
Der_CO2 <- select(Der_CO2, fecha, ultimo)
Der_CO2 <- rename(Der_CO2, Der_CO2 = ultimo)
Der_CO2$fecha <- as.Date(Der_CO2$fecha, format = "%d/%m/%Y")
head(Der_CO2)
```

```
##     fecha Der_CO2
## 1 2012-01-01  7.89
## 2 2012-01-02  7.14
## 3 2012-01-03  6.28
## 4 2012-01-04  6.27
## 5 2012-01-05  6.43
## 6 2012-01-06  6.30
```

Unimos con el dataframe inicial

```
df <- merge(df, Der_CO2, all.x = TRUE)
str(df)
```

```
## 'data.frame': 3652 obs. of 35 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem    : chr "7" "1" "2" "3" ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr  : num 111 108 112 114 113 ...
## $ Prec_gas   : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb  : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
```

```
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA ...
## $ Temp_max_Bar : num NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ Res_hidr   : num 56.8 NA NA NA NA ...
## $ Der_CO2    : num 7.89 7.14 6.28 6.27 6.43 6.3 NA NA 6.51 6.84 ...
```

6.5.10 Unimos ahora la variable "Ibex"

str(Ibex)

```
## 'data.frame': 2559 obs. of 7 variables:
## $ fecha : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ ultimo : num 8327 8724 8732 8582 8330 ...
## $ apertura : num 8539 8558 8739 8683 8599 ...
## $ maximo : num 8582 8724 8743 8701 8599 ...
## $ minimo : num 8411 8557 8597 8527 8301 ...
## $ volumen : chr "231.35" "102.66" "186.69" "243.8" ...
## $ variacion: chr "-1.34%" "1.84%" "0.10%" "-1.72%" ...
```

names(Ibex)

```
## [1] "fecha"  "ultimo"  "apertura" "maximo"  "minimo"  "volumen"
## [7] "variacion"
```

Limpiamos el segundo archivo

```
Ibex <- select(Ibex, fecha, ultimo)
Ibex <- rename(Ibex, Ibex = ultimo)
Ibex$fecha <- as.Date(Ibex$fecha, format = "%d/%m/%Y")
head(Ibex)
```

```
##     fecha Ibex
## 1 2012-01-01 8327.2
## 2 2012-01-02 8723.8
## 3 2012-01-03 8732.4
## 4 2012-01-04 8581.8
## 5 2012-01-05 8329.6
## 6 2012-01-06 8289.1
```

Unimos con el dataframe inicial

```
df <- merge(df, Ibex, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 36 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
```

```
## $ dia_sem   : chr "7" "1" "2" "3" ...
## $ Pre_elec  : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem       : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr : num 111 108 112 114 113 ...
## $ Prec_gas  : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar : num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4.5 2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ Res_hidr   : num 56.8 NA NA NA NA ...
## $ Der_CO2    : num 7.89 7.14 6.28 6.27 6.43 6.3 NA NA 6.51 6.84 ...
## $ Ibex      : num 8327 8724 8732 8582 8330 ...
```

6.5.11 Unimos ahora la variable "Int_ee"

```
str(Int_ee)
```

```
## 'data.frame': 3652 obs. of 4 variables:
## $ fecha     : chr "01/01/2012" "02/01/2012" "03/01/2012" "04/01/2012" ...
## $ exportacion: num -52979 -57162 -60439 -55014 -60319 ...
## $ importacion: num 36054 23084 33642 25486 22250 ...
## $ saldo      : num -16926 -34078 -26798 -29528 -38069 ...
```

```
names(Int_ee)
```

```
## [1] "fecha"     "exportacion" "importacion" "saldo"
```

Limpiamos el segundo archivo

```
Int_ee <- select(Int_ee, fecha, saldo)
Int_ee <- rename(Int_ee, Int_ee = saldo)
```

```

Int_ee$fecha <- as.Date(Int_ee$fecha, format = "%d/%m/%Y")
head(Int_ee)

##    fecha   Int_ee
## 1 2012-01-01 -16925.91
## 2 2012-01-02 -34078.38
## 3 2012-01-03 -26797.53
## 4 2012-01-04 -29528.22
## 5 2012-01-05 -38068.67
## 6 2012-01-06 -55881.20

# Unimos con el dataframe inicial
df <- merge(df, Int_ee, all.x = TRUE)
str(df)

## 'data.frame': 3652 obs. of 37 variables:
## $ fecha      : Date, format: "2012-01-01" "2012-01-02" ...
## $ dia_sem    : chr "7" "1" "2" "3" ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr  : num 111 108 112 114 113 ...
## $ Prec_gas   : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb  : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
## $ Prod_nucl  : num 156 156 156 156 156 ...
## $ Prod_pet   : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas   : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb  : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb   : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog   : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren: num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad: num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad: num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_min_Val: num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val: num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev: num 2.4 7.5 4.2 4 5.2 1.4 7.4 6.8 3.9 4.4 ...
## $ Temp_max_Sev: num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar: num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar: num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 NA 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ Res_hidr   : num 56.8 NA NA NA NA ...
## $ Der_CO2    : num 7.89 7.14 6.28 6.27 6.43 6.3 NA NA 6.51 6.84 ...
## $ Ibex       : num 8327 8724 8732 8582 8330 ...
## $ Int_ee     : num -16926 -34078 -26798 -29528 -38069 ...

```

6.5.12 Asignación de valores a los datos faltantes (NA's)

```
# Creamos una copia del dataframe para trabajar con ella  
df_2 <- df
```

Obtenemos las variables que poseen valores NA's y el numero de ellos que presentan
`apply(is.na(df_2), 2, sum)`

```
##     fecha    dia_sem  Pre_elec      Dem  Prec_petr  
##      0         0       1      1022  
##  Prec_gas  Prec_carb  Prod_eol  Prod_sol  Prod_hidr  
##  1111      1094       0       0       0  
##  Prod_ofr  Prod_nucl  Prod_pet  Prod_gas  Prod_carb  
##      0         0       0       0       0  
##  Prod_comb  Prod_cog  Prod_no_ren Temp_min_Mad Temp_max_Mad  
##      0         0       0      15      15  
##  Temp_min_Bar Temp_max_Bar Temp_min_Val Temp_max_Val Temp_min_Sev  
##  426        425       0       8      15  
##  Temp_max_Sev Temp_min_Zar Temp_max_Zar Vel_media_Val Vel_media_Alb  
##   13         0       0       5      16  
##  Vel_media_Zar Vel_media_Cor Vel_media_Hue  Res_hidr    Der_CO2  
##      1        40       9     3183      1063  
##  Ibex     Int_ee  
##  1093       0
```

Haciendo un resumen tendremos el siguiente número de NA's, sus motivos y la opción tomada para solucionarlo:

"**Dem**" posee **1**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Prec_petr**" posee **1022** correspondientes a los sábados y los domingos en el periodo de los 10 años estudiado. El motivo es que el mercado de negociación de esta materia prima no se efectúa estos días.

Para solucionar este problema se tomará como valor para estos dos días el del viernes anterior.

"**Prec_gas**" posee **1111**, es el mismo caso que el anterior.

"**Prec_carb**" posee **1094**, es el mismo caso que los dos anteriores

"**Temp_min_Mad**" posee **15** y "**Temp_max_Mad**" **15**, el motivo es desconocido pero al ser cantidades tan pequeñas no influirán en los resultados. **Se ha tomado la decisión de asignar el valor del día anterior.**

"**Temp_min_Bar**" posee **426** y "**Temp_max_Bar**" **425**, el motivo es que la fuente de los datos no los proporciona en tres períodos, el primero corresponden a los 11 primeros valores del año 2012, el segundo entre el 1 de enero y el 13 junio de 2017, y el segundo entre el 1 de julio y el 31 de diciembre del 2020. El motivo de este percance es desconocido. **La solución adoptada es la de tomar para el primer tramo es tomar la media de los valores de los dos años posteriores, para el segundo tramo la media de los valores del año anterior y posterior, y para el ultimo tramo la media de los valores de los dos años anteriores**

"**Temp_max_Val**" posee **8**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Temp_min_Sev**" posee **15** y "**Temp_max_Sev**" **13**, el motivo es desconocido pero al ser cantidades tan pequeñas no influirán en los resultados. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Vel_media_Val**" posee **5**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Vel_media_Alb**" posee **16**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Vel_media_Zar**" posee **1**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Vel_media_Cor**" posee **42**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Vel_media_Hue**" posee **9**, el motivo es desconocido pero al ser una cantidad tan pequeña no influirán en el resultado. **Se ha tomado la decisión de tomar el valor del día anterior.**

"**Res_hidr**" posee **3183**, en este caso existen tantos datos faltantes porque los registros obtenidos son semanales, por lo que solo se tiene 1 valor de cada 7 de cada semana. **La solución adoptada es la de dar a todos los días de esa semana el mismo valor ya que es un índice que varía lentamente.**

"**Der_CO2**" posee **1063**, sucede lo mismo que las variables de precio del petróleo, gas natural y carbono, por lo que la solución será la misma.

"**Ibex**" posee **1093**, sucede lo mismo que las variables de precio del petróleo, gas natural y carbono, por lo que la solución será la misma.

El resto de variables no presentan ningún dato NA.

####6.5.12.1 Primero asignamos valores a las variables que tienen poca cantidad de NA's, a las que corresponden a los sábados y domingo y a la variable "Res_hidr", que simplemente hay que asignar el valor de los martes a todos los días de la semana

```
df_2 <- df_2 %>% fill(Dem, Prec_petr, Prec_gas, Prec_carb, Temp_min_Mad, Temp_max_Mad, Temp_max_Val, Temp_min_Sev, Temp_max_Sev, Vel_media_Val, Vel_media_Al, Vel_media_Zar, Vel_media_Cor, Vel_media_Hue, Res_hidr, Der_CO2, Ibex)
```

```
apply(is.na(df_2), 2, sum)
```

```
##     fecha dia_sem Pre_elec Dem Prec_petr
##     0      0      0      0      0
## Prec_gas Prec_carb Prod_eol Prod_sol Prod_hidr
##     0      0      0      0      0
## Prod_ofr Prod_nucl Prod_pet Prod_gas Prod_carb
##     0      0      0      0      0
## Prod_comb Prod_cog Prod_no_ren Temp_min_Mad Temp_max_Mad
##     0      0      0      0      0
## Temp_min_Bar Temp_max_Bar Temp_min_Val Temp_max_Val Temp_min_Sev
##     426     425      0      0      0
## Temp_max_Sev Temp_min_Zar Temp_max_Zar Vel_media_Val Vel_media_Al
##     0      0      0      0      0
## Vel_media_Zar Vel_media_Cor Vel_media_Hue Res_hidr Der_CO2
##     0      0      0      0      0
##     Ibex   Int_ee
##     0      0
```

6.5.12.2 Ahora pasamos a asignar valores a los NA's de las variables Temp_min_Bar y Temp_max_Bar.

Se hará de la siguiente forma:

Para los **primeros datos del año 2012**, se asignará la media de las temperaturas de los dos años posteriores.

Para los **datos del primer semestre del 2017**, se asignará la media de las temperaturas del año anterior y posterior.

Para los **datos del segundo semestre del 2022**, se asignará la media de las temperaturas de los dos años anteriores.

```
# Cogemos la variable "Prec_elec" porque tiene todas las fechas del periodo de estudio y limpiamos el dataframe para quedarnos con la columna "fecha"
db_prec_elec <- select(Pre_elec, datetime)
db_prec_elec <- rename(db_prec_elec, fecha = datetime)
db_prec_elec$fecha <- as.Date(db_prec_elec$fecha)
```

Vuelvo a cargar el archivo para tener todo el dataframe

```
Temp_Bar <- read.table('D:/0.BIG DATA/0.MASTER Y CURSOS/1.MODULOS MASTER/1.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/7.2.Temp Barcelona 2012-2022.csv', header = T, sep = ',', encoding = "UTF-8")
```

Limpiamos el dataframe para las temperaturas de Barcelona

```
Temp_Bar <- select(Temp_Bar, fecha, tmin, tmax)
Temp_Bar <- rename(Temp_Bar, Temp_min_Bar = tmin, Temp_max_Bar = tmax)
Temp_Bar$fecha <- as.Date(Temp_Bar$fecha, format = "%d/%m/%Y")
```

```

# Unimos los dos dataframes
Temp_Bar <- merge(db_prec_elec, Temp_Bar, all.x = TRUE)
str(Temp_Bar)

## 'data.frame': 3652 obs. of 3 variables:
## $ fecha : Date, format: "2012-01-01" "2012-01-02" ...
## $ Temp_min_Bar: num NA NA NA NA NA NA NA NA NA ...
## $ Temp_max_Bar: num NA NA NA NA NA NA NA NA NA ...

# Compruebo que son las mismas variables con los mismos NA's
apply(is.na(Temp_Bar), 2, sum)

##     fecha Temp_min_Bar Temp_max_Bar
##      0        426        425

```

Comenzamos con los primeros datos del año 2012

```

# Definimos las fechas que marcan el inicio y el final de cada periodo de NA's, y obtenemos su indice
fecha_1.1 <- which(as.Date(c("2012-01-01")) == Temp_Bar$fecha)
fecha_1.2 <- which(as.Date(c("2012-01-11")) == Temp_Bar$fecha)
fecha_1.3 <- which(as.Date(c("2013-01-01")) == Temp_Bar$fecha)
fecha_1.4 <- which(as.Date(c("2013-01-11")) == Temp_Bar$fecha)
fecha_1.5 <- which(as.Date(c("2014-01-01")) == Temp_Bar$fecha)
fecha_1.6 <- which(as.Date(c("2014-01-11")) == Temp_Bar$fecha)

```

```

# Creamos 3 dataframes, uno para cada año
Temp_Bar_1_2012 <- Temp_Bar[(fecha_1.1:fecha_1.2),]
Temp_Bar_1_2012 <- rename(Temp_Bar_1_2012, T_min_Bar_2012 = Temp_min_Bar, T_max_Bar_2012 = Temp_max_Bar)

Temp_Bar_1_2013 <- Temp_Bar[(fecha_1.3:fecha_1.4),]
Temp_Bar_1_2013 <- rename(Temp_Bar_1_2013, f_2013 = fecha, T_min_Bar_2013 = Temp_min_Bar, T_max_Bar_2013 = Temp_max_Bar)

Temp_Bar_1_2014 <- Temp_Bar[(fecha_1.5:fecha_1.6),]
Temp_Bar_1_2014 <- rename(Temp_Bar_1_2014, f_2014 = fecha, T_min_Bar_2014 = Temp_min_Bar, T_max_Bar_2014 = Temp_max_Bar)

```

```

# Unimos los 3 dataframes
Temp_Bar_1 <- cbind(Temp_Bar_1_2012, Temp_Bar_1_2013, Temp_Bar_1_2014)
# Eliminamos las dos columnas de los años 2016 y 2018
Temp_Bar_1 <- select(Temp_Bar_1, -f_2013, -f_2014)

# Ahora es el momento de dale el valor a los días del 2017. Recordamos que le daremos la media de las temperaturas de los dos años posteriores
Temp_Bar_1$T_min_Bar_2012 <- rowMeans(Temp_Bar_1[,c(4, 6)])
Temp_Bar_1$T_max_Bar_2012 <- rowMeans(Temp_Bar_1[,c(5, 7)])
# Nos quedamos solo con las columnas de 2017 y renombramos las variables
Temp_Bar_1 <- select(Temp_Bar_1, fecha, T_min_Bar_2012, T_max_Bar_2012)

df_3 <- df_2
# Unimos el dataframe con los datos asignados para el 2012 al dataframe principal ("df_2")
df_3 <- merge(df_3, Temp_Bar_1, all.x = TRUE)

df_3 <- mutate_at(df_3, c("Temp_min_Bar", "Temp_max_Bar", "T_min_Bar_2012", "T_max_Bar_2012"), ~replace(., is.na(.), 0))

```

```
df_3 <- mutate(df_3, Temp_min_Bar = Temp_min_Bar + T_min_Bar_2012)
df_3 <- mutate(df_3, Temp_max_Bar = Temp_max_Bar + T_max_Bar_2012)

df_3 <- select(df_3, -T_min_Bar_2012, -T_max_Bar_2012)
```

*Seguimos con los datos **datos del primer semestre del 2017***

```
# Definimos las fechas que marcan el inicio y el final de cada periodo de NA's, y obtenemos su indice
fecha_2.1 <- which(as.Date(c("2017-01-01")) == Temp_Bar$fecha)
fecha_2.2 <- which(as.Date(c("2017-06-13")) == Temp_Bar$fecha)
fecha_2.3 <- which(as.Date(c("2016-01-01")) == Temp_Bar$fecha)
fecha_2.4 <- which(as.Date(c("2016-06-13")) == Temp_Bar$fecha)
fecha_2.5 <- which(as.Date(c("2018-01-01")) == Temp_Bar$fecha)
fecha_2.6 <- which(as.Date(c("2018-06-13")) == Temp_Bar$fecha)
```

Creamos 3 dataframes, uno para cada año

```
Temp_Bar_2_2017 <- Temp_Bar[(fecha_2.1:fecha_2.2),]
Temp_Bar_2_2017 <- rename(Temp_Bar_2_2017, T_min_Bar_2017 = Temp_min_Bar, T_max_Bar_2017 = Temp_max_Bar)
```

```
Temp_Bar_2_2016 <- Temp_Bar[(fecha_2.3:fecha_2.4),]
```

```
Temp_Bar_2_2016 <- rename(Temp_Bar_2_2016, f_2016 = fecha, T_min_Bar_2016 = Temp_min_Bar, T_max_Bar_2016 = Temp_max_Bar)
```

```
Temp_Bar_2_2018 <- Temp_Bar[(fecha_2.5:fecha_2.6),]
```

```
Temp_Bar_2_2018 <- rename(Temp_Bar_2_2018, f_2018 = fecha, T_min_Bar_2018 = Temp_min_Bar, T_max_Bar_2018 = Temp_max_Bar)
```

Como el año 2016 es bisiesto hay que eliminar el día 29 de febrero

```
Temp_Bar_2_2016 <- Temp_Bar_2_2016[!(Temp_Bar_2_2016$f_2016 == "2016-02-29"),]
```

Unimos los 3 dataframes

```
Temp_Bar_2 <- cbind(Temp_Bar_2_2017, Temp_Bar_2_2016, Temp_Bar_2_2018)
```

Eliminamos las dos columnas de los años 2016 y 2018

```
Temp_Bar_2 <- select(Temp_Bar_2, -f_2016, -f_2018)
```

Ahora es el momento de dale el valor a los días del 2017. Recordamos que le daremos la media de las temperaturas del año anterior y posterior

```
Temp_Bar_2$T_min_Bar_2017 <- rowMeans(Temp_Bar_2[,c(4, 6)])
```

```
Temp_Bar_2$T_max_Bar_2017 <- rowMeans(Temp_Bar_2[,c(5, 7)])
```

Nos quedamos solo con las columnas de 2017 y renombramos las variables

```
Temp_Bar_2 <- select(Temp_Bar_2, fecha, T_min_Bar_2017, T_max_Bar_2017)
```

Unimos el dataframe con los datos asignados para el 2012 al dataframe principal ("df_3")

```
df_3 <- merge(df_3, Temp_Bar_2, all.x = TRUE)
```

```
df_3 <- mutate_at(df_3, c("Temp_min_Bar", "Temp_max_Bar", "T_min_Bar_2017", "T_max_Bar_2017"),
  ~replace(., is.na(.), 0))
```

```
df_3 <- mutate(df_3, Temp_min_Bar = Temp_min_Bar + T_min_Bar_2017)
```

```
df_3 <- mutate(df_3, Temp_max_Bar = Temp_max_Bar + T_max_Bar_2017)
```

```
df_3 <- select(df_3, -T_min_Bar_2017, -T_max_Bar_2017)
```

*Terminamos con los **datos del segundo semestre del 2022***

```
# Definimos las fechas que marcan el inicio y el final de cada periodo de NA's, y obtenemos su indice
fecha_3.1 <- which(as.Date(c("2022-07-01")) == Temp_Bar$fecha)
```

```

fecha_3.2 <- which (as.Date(c("2022-12-31")) == Temp_Bar$fecha)
fecha_3.3 <- which (as.Date(c("2021-07-01")) == Temp_Bar$fecha)
fecha_3.4 <- which (as.Date(c("2021-12-31")) == Temp_Bar$fecha)
fecha_3.5 <- which (as.Date(c("2019-07-01")) == Temp_Bar$fecha)
fecha_3.6 <- which (as.Date(c("2019-12-31")) == Temp_Bar$fecha)

# Creamos 3 dataframes, uno para cada año
Temp_Bar_3_2022 <- Temp_Bar[(fecha_3.1:fecha_3.2),]
Temp_Bar_3_2022 <- rename(Temp_Bar_3_2022, T_min_Bar_2022 = Temp_min_Bar, T_max_Bar_2022 = Temp_max_Bar)

Temp_Bar_3_2021 <- Temp_Bar[(fecha_3.3:fecha_3.4),]
Temp_Bar_3_2021 <- rename(Temp_Bar_3_2021, f_2021 = fecha, T_min_Bar_2021 = Temp_min_Bar, T_max_Bar_2021 = Temp_max_Bar)

Temp_Bar_3_2019 <- Temp_Bar[(fecha_3.5:fecha_3.6),]
Temp_Bar_3_2019 <- rename(Temp_Bar_3_2019, f_2019 = fecha, T_min_Bar_2019 = Temp_min_Bar, T_max_Bar_2019 = Temp_max_Bar)

# Unimos los 3 dataframes
Temp_Bar_3 <- cbind(Temp_Bar_3_2022, Temp_Bar_3_2021, Temp_Bar_3_2019)
# Eliminamos las dos columnas de los años 2016 y 2018
Temp_Bar_3 <- select(Temp_Bar_3, -f_2021, -f_2019)

# Ahora es el momento de dale el valor a los días del 2017. Recordamos que le daremos la media de las temperaturas del año anterior y posterior
Temp_Bar_3$T_min_Bar_2022 <- rowMeans(Temp_Bar_3[,c(4, 6)])
Temp_Bar_3$T_max_Bar_2022 <- rowMeans(Temp_Bar_3[,c(5, 7)])
# Nos quedamos solo con las columnas de 2017 y renombramos las variables
Temp_Bar_3 <- select(Temp_Bar_3, fecha, T_min_Bar_2022, T_max_Bar_2022)

# Unimos el dataframe con los datos asignados para el 2012 al dataframe principal ("df_3")
df_3 <- merge(df_3, Temp_Bar_3, all.x = TRUE)

df_3 <- mutate_at(df_3, c("Temp_min_Bar", "Temp_max_Bar", "T_min_Bar_2022", "T_max_Bar_2022"),
  ~replace(., is.na(.), 0))

df_3 <- mutate(df_3, Temp_min_Bar = Temp_min_Bar + T_min_Bar_2022)
df_3 <- mutate(df_3, Temp_max_Bar = Temp_max_Bar + T_max_Bar_2022)

df_3 <- select(df_3, -T_min_Bar_2022, -T_max_Bar_2022)
head(df_3)

##    fecha dia_sem Pre_elec Dem Prec_petr Prec_gas Prec_carb Prod_eol
## 1 2012-01-01     7 48.1992 20888.99 110.85  55.60  111.56 155.1421
## 2 2012-01-02     1 49.7375 23325.20 108.35  53.75  110.20 261.0320
## 3 2012-01-03     2 57.1317 26994.32 112.13  52.75  109.35 164.6931
## 4 2012-01-04     3 54.6721 27934.39 113.70  53.09  109.55 164.3039
## 5 2012-01-05     4 51.5471 28089.09 112.74  52.95  110.10 204.1075
## 6 2012-01-06     5 52.4642 24915.97 113.06  52.87  110.20 215.8213
##   Prod_sol Prod_hidr Prod_ofr Prod_nucl Prod_pet Prod_gas Prod_carb Prod_comb
## 1 19.90159 51.70207 12.55508 156.0602 9.099916 1.211399 95.83518 63.99322
## 2 14.71745 57.01815 12.49393 155.9083 9.585639 1.852946 118.34087 61.77065
## 3 22.20058 70.73139 12.40207 155.9070 9.693961 2.357607 169.17696 99.75440
## 4 22.60510 76.93590 12.82306 155.9415 9.460365 2.276651 176.29588 102.35277

```

```

## 5 22.67810 71.55659 12.99064 155.8380 9.619993 1.970738 140.08747 82.15043
## 6 17.71867 64.19600 13.22839 156.1161 9.145990 1.218538 86.14172 54.24204
## Prod_cog Prod_no_ren Temp_min_Mad Temp_max_Mad Temp_min_Bar Temp_max_Bar
## 1 61.13340 9.508938    2.3    11.6    7.95   13.55
## 2 86.04903 11.045884    5.8    11.6    9.70   16.25
## 3 95.67452 11.336695    1.5    10.5    9.80   18.30
## 4 96.84716 11.904774    0.5    11.6    9.80   17.10
## 5 95.93271 12.658157    2.2    13.0    10.60   17.65
## 6 86.23782 11.247955    3.0    15.0    10.95   17.90
## Temp_min_Val Temp_max_Val Temp_min_Sev Temp_max_Sev Temp_min_Zar Temp_max_Zar
## 1    10.0    21.0    2.4    17.4    5.4    18.2
## 2    10.3    20.4    7.5    19.0    5.4    13.7
## 3     7.2    19.7    4.2    17.7    2.6    12.5
## 4     6.7    22.4    4.0    18.8    3.6    15.5
## 5     9.2    25.6    5.2    22.4    5.4    15.7
## 6    11.4    23.0    1.4    19.1    8.4    15.4
## Vel_media_Val Vel_media_Alb Vel_media_Zar Vel_media_Cor Vel_media_Hue
## 1     1.1     0.3     1.7     3.5     1.4
## 2     5.0     3.6     4.4     2.4     1.4
## 3     1.7     1.4     4.4     5.0     1.4
## 4     1.4     1.7     6.9     2.2     1.4
## 5     1.1     2.8     7.8     3.1     1.9
## 6     1.1     2.8    12.8     2.5     1.9
## Res_hidr Der_CO2 Ibex Int_ee
## 1 56.83 7.89 8327.2 -16925.91
## 2 56.83 7.14 8723.8 -34078.38
## 3 56.83 6.28 8732.4 -26797.53
## 4 56.83 6.27 8581.8 -29528.22
## 5 56.83 6.43 8329.6 -38068.67
## 6 56.83 6.30 8289.1 -55881.20

```

6.5.13 Exportación del dataframe obtenido

Exportamos a un archivo el dataframe resultante para emplearlo en el código de los modelos de predicción

Para exportar el dataframe cambiar la ruta y descomentar el código siguiente

```

# setwd("D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO")
# write_csv(df_3, "df_3.csv")

```

ANEXO IV: CORRELACION ENTRE VARIABLES

Correlacion entre variables

Iñigo Elorza Barea

2023-07-13

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.2.3
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
```

6.7 CORRELACION ENTRE VARIABLES

Es importante conocer la relacion que hay entre las diferentes variables.

Una **correlación positiva** indica una **relación directa**, es decir, que la subida de una de las variables, supondrá la subida de la otra.

En una **correlación negativa** la relacion es **Inversa**.

```
# Cargamos el archivo en donde tenemos el dataset del estudio
```

```
df_3 <- read.csv(D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Trabajo Final de Master/1.TFM/5.CODIGO/FICHEROS A ENTREGAR/01 Archivos csv y excel/13.Dataset de datos (df_3).csv', header = T, sep = ',', encoding = "UTF-8")
```

```
# Comprobamos que se han cargado todas las variables y el tipo de datos que son
```

```
str(df_3)
```

```
## 'data.frame': 3652 obs. of 37 variables:
## $ fecha     : chr "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" ...
## $ dia_sem   : int 7 1 2 3 4 5 6 7 1 2 ...
## $ Pre_elec   : num 48.2 49.7 57.1 54.7 51.5 ...
## $ Dem        : num 20889 23325 26994 27934 28089 ...
## $ Prec_petr  : num 111 108 112 114 113 ...
## $ Prec_gas   : num 55.6 53.8 52.8 53.1 53 ...
## $ Prec_carb  : num 112 110 109 110 110 ...
## $ Prod_eol   : num 155 261 165 164 204 ...
## $ Prod_sol   : num 19.9 14.7 22.2 22.6 22.7 ...
## $ Prod_hidr  : num 51.7 57 70.7 76.9 71.6 ...
## $ Prod_ofr   : num 12.6 12.5 12.4 12.8 13 ...
```

```

## $ Prod_nucl : num 156 156 156 156 156 ...
## $ Prod_pet : num 9.1 9.59 9.69 9.46 9.62 ...
## $ Prod_gas : num 1.21 1.85 2.36 2.28 1.97 ...
## $ Prod_carb : num 95.8 118.3 169.2 176.3 140.1 ...
## $ Prod_comb : num 64 61.8 99.8 102.4 82.2 ...
## $ Prod_cog : num 61.1 86 95.7 96.8 95.9 ...
## $ Prod_no_ren : num 9.51 11.05 11.34 11.9 12.66 ...
## $ Temp_min_Mad : num 2.3 5.8 1.5 0.5 2.2 3 3.2 2 1 1.8 ...
## $ Temp_max_Mad : num 11.6 11.6 10.5 11.6 13 15 13.2 12.6 13.1 10.8 ...
## $ Temp_min_Bar : num 7.95 9.7 9.8 9.8 10.6 ...
## $ Temp_max_Bar : num 13.6 16.2 18.3 17.1 17.6 ...
## $ Temp_min_Val : num 10 10.3 7.2 6.7 9.2 11.4 7.6 6.8 5.3 5.5 ...
## $ Temp_max_Val : num 21 20.4 19.7 22.4 25.6 23 18.4 16.8 18 17.9 ...
## $ Temp_min_Sev : num 2.4 7.5 4.2 4.5 2.1 4.7 4.6 8.3 9.4 4.4 ...
## $ Temp_max_Sev : num 17.4 19 17.7 18.8 22.4 19.1 20 18.7 19.4 19.4 ...
## $ Temp_min_Zar : num 5.4 5.4 2.6 3.6 5.4 8.4 7.6 6.6 3.5 -1.6 ...
## $ Temp_max_Zar : num 18.2 13.7 12.5 15.5 15.7 15.4 14.8 14 12.6 9.6 ...
## $ Vel_media_Val: num 1.1 5 1.7 1.4 1.1 1.1 0.8 1.1 1.1 1.1 ...
## $ Vel_media_Alb: num 0.3 3.6 1.4 1.7 2.8 2.8 2.8 0.3 0.6 1.1 ...
## $ Vel_media_Zar: num 1.7 4.4 4.4 6.9 7.8 12.8 9.7 10.3 5 1.1 ...
## $ Vel_media_Cor: num 3.5 2.4 5 2.2 3.1 2.5 5 3.1 2.2 1.7 ...
## $ Vel_media_Hue: num 1.4 1.4 1.4 1.4 1.9 1.9 1.7 1.7 1.1 1.4 ...
## $ Res_hidr : num 56.8 56.8 56.8 56.8 56.8 ...
## $ Der_CO2 : num 7.89 7.14 6.28 6.27 6.43 6.3 6.3 6.3 6.51 6.84 ...
## $ Ibex : num 8327 8724 8732 8582 8330 ...
## $ Int_ee : num -16926 -34078 -26798 -29528 -38069 ...

```

Obtenemos la matriz de correlación redondeada a 2 decimales

```
correlacion <- round(cor(df_3[,2:37]),2)
```

```
correlacion
```

```

##      dia_sem Pre_elec Dem Prec_petr Prec_gas Prec_carb Prod_eol
## dia_sem    1.00 -0.05 -0.35   0.00   0.00   0.00   0.01
## Pre_elec   -0.05  1.00  0.00   0.21   0.79   0.73  -0.04
## Dem       -0.35   0.00  1.00  -0.21  -0.12  -0.13   0.04
## Prec_petr   0.00   0.21 -0.21   1.00   0.31   0.39   0.04
## Prec_gas    0.00   0.79 -0.12   0.31   1.00   0.87   0.12
## Prec_carb   0.00   0.73 -0.13   0.39   0.87   1.00   0.07
## Prod_eol    0.01  -0.04  0.04   0.04   0.12   0.07   1.00
## Prod_sol    0.00   0.43 -0.13   0.16   0.46   0.60  -0.17
## Prod_hidr   -0.13  -0.26  0.16   0.00  -0.22  -0.28   0.08
## Prod_ofr     0.05   0.56  0.00   0.29   0.53   0.54   0.02
## Prod_nucl   -0.02  -0.03  0.11   0.04   0.02   0.03  -0.04
## Prod_pet    -0.08  -0.33  0.23  -0.11  -0.41  -0.37  -0.28
## Prod_gas    -0.28  -0.10  0.27   0.15  -0.11  -0.08  -0.22
## Prod_carb   -0.12  -0.23  0.26   0.02  -0.36  -0.34  -0.41
## Prod_comb   -0.22  0.52  0.24   0.08  0.49   0.47  -0.33
## Prod_cog    -0.19  -0.32  0.25   0.06  -0.54  -0.54  -0.08
## Prod_no_ren -0.10  -0.38  0.32  -0.49  -0.51  -0.50  -0.15
## Temp_min_Mad  0.00   0.04 -0.04   0.02   0.04   0.12  -0.36
## Temp_max_Mad  0.00   0.03 -0.04   0.03   0.00   0.10  -0.42
## Temp_min_Bar  0.01   0.02 -0.06   0.06   0.01   0.09  -0.34
## Temp_max_Bar  0.01   0.03 -0.05   0.04   0.03   0.10  -0.31
## Temp_min_Val  0.01   0.02 -0.08   0.04   0.04   0.11  -0.27
## Temp_max_Val  0.01  -0.22 -0.12   0.44  -0.23  -0.23  -0.17

```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

```
## Temp_min_Sev  0.01  0.02 -0.10  0.02  0.03  0.09 -0.32
## Temp_max_Sev 0.00  0.03 -0.07  0.02  0.00  0.10 -0.40
## Temp_min_Zar 0.00  0.03 -0.06  0.04  0.04  0.13 -0.29
## Temp_max_Zar 0.01  0.00 -0.06  0.06 -0.01  0.10 -0.38
## Vel_media_Val 0.00 -0.20  0.02  0.08 -0.10 -0.09  0.58
## Vel_media_Alb 0.03 -0.23  0.05  0.00 -0.18 -0.16  0.54
## Vel_media_Zar 0.01 -0.12  0.06  0.02 -0.07 -0.05  0.49
## Vel_media_Cor 0.00 -0.15  0.00  0.03 -0.05 -0.06  0.58
## Vel_media_Hue 0.01 -0.06 -0.01 -0.03 -0.03  0.02  0.18
## Res_hidr     0.00 -0.42  0.02 -0.11 -0.48 -0.50 -0.10
## Der_CO2       0.00  0.72 -0.11  0.15  0.80  0.80  0.14
## Ibex         0.00 -0.22  0.10 -0.35 -0.34 -0.34 -0.07
## Int_ee        -0.01 -0.31  0.19 -0.50 -0.49 -0.46 -0.35
## Prod_sol     Prod_hidr Prod_ofr Prod_nucl Prod_pet Prod_gas Prod_carb
## dia_sem      0.00 -0.13  0.05 -0.02 -0.08 -0.28 -0.12
## Pre_elec     0.43 -0.26  0.56 -0.03 -0.33 -0.10 -0.23
## Dem          -0.13  0.16  0.00  0.11  0.23  0.27  0.26
## Prec_petr    0.16  0.00  0.29  0.04 -0.11  0.15  0.02
## Prec_gas     0.46 -0.22  0.53  0.02 -0.41 -0.11 -0.36
## Prec_carb    0.60 -0.28  0.54  0.03 -0.37 -0.08 -0.34
## Prod_eol     -0.17  0.08  0.02 -0.04 -0.28 -0.22 -0.41
## Prod_sol     1.00 -0.25  0.41  0.00 -0.22 -0.07 -0.34
## Prod_hidr    -0.25  1.00 -0.28 -0.04 -0.09 -0.15 -0.17
## Prod_ofr     0.41 -0.28  1.00 -0.06 -0.37 -0.13 -0.22
## Prod_nucl    0.00 -0.04 -0.06  1.00  0.20  0.05 -0.02
## Prod_pet     -0.22 -0.09 -0.37  0.20  1.00  0.45  0.61
## Prod_gas     -0.07 -0.15 -0.13  0.05  0.45  1.00  0.50
## Prod_carb    -0.34 -0.17 -0.22 -0.02  0.61  0.50  1.00
## Prod_comb    0.33 -0.35  0.37 -0.01 -0.14  0.15 -0.09
## Prod_cog     -0.45  0.10 -0.12 -0.02  0.24  0.13  0.35
## Prod_no_ren   -0.46 -0.06 -0.37  0.10  0.58  0.32  0.53
## Temp_min_Mad 0.48 -0.34  0.09  0.00  0.36  0.39  0.11
## Temp_max_Mad 0.59 -0.32  0.06  0.02  0.37  0.37  0.12
## Temp_min_Bar 0.43 -0.32  0.09  0.02  0.36  0.39  0.13
## Temp_max_Bar 0.43 -0.32  0.09  0.03  0.32  0.36  0.11
## Temp_min_Val 0.43 -0.33  0.09 -0.01  0.36  0.41  0.08
## Temp_max_Val -0.01  0.02 -0.16  0.02  0.35  0.36  0.46
## Temp_min_Sev 0.41 -0.32  0.07 -0.02  0.35  0.39  0.09
## Temp_max_Sev 0.60 -0.36  0.07  0.01  0.35  0.36  0.11
## Temp_min_Zar 0.48 -0.33  0.09 -0.03  0.33  0.38  0.07
## Temp_max_Zar 0.54 -0.29  0.04  0.02  0.35  0.36  0.09
## Vel_media_Val -0.11  0.11 -0.15 -0.04  0.04 -0.01 -0.19
## Vel_media_Alb -0.21  0.21 -0.19 -0.03  0.05 -0.08 -0.18
## Vel_media_Zar 0.08  0.00 -0.06  0.00 -0.01 -0.03 -0.14
## Vel_media_Cor -0.11  0.10 -0.10 -0.05 -0.07 -0.07 -0.23
## Vel_media_Hue 0.14  0.00 -0.06 -0.05  0.08  0.08 -0.17
## Res_hidr     -0.07  0.60 -0.38 -0.10  0.23  0.02  0.11
## Der_CO2       0.61 -0.23  0.63 -0.02 -0.63 -0.30 -0.59
## Ibex         -0.17  0.06 -0.30 -0.13  0.16  0.21  0.23
## Int_ee        -0.20  0.11 -0.23 -0.09  0.21  0.04  0.06
## Prod_comb    Prod_cog Prod_no_ren Temp_min_Mad Temp_max_Mad
## dia_sem      -0.22 -0.19  -0.10  0.00  0.00
## Pre_elec     0.52 -0.32  -0.38  0.04  0.03
## Dem          0.24  0.25  0.32 -0.04 -0.04
```

## Prec_petr	0.08	0.06	-0.49	0.02	0.03
## Prec_gas	0.49	-0.54	-0.51	0.04	0.00
## Prec_carb	0.47	-0.54	-0.50	0.12	0.10
## Prod_eol	-0.33	-0.08	-0.15	-0.36	-0.42
## Prod_sol	0.33	-0.45	-0.46	0.48	0.59
## Prod_hidr	-0.35	0.10	-0.06	-0.34	-0.32
## Prod_ofr	0.37	-0.12	-0.37	0.09	0.06
## Prod_nucl	-0.01	-0.02	0.10	0.00	0.02
## Prod_pet	-0.14	0.24	0.58	0.36	0.37
## Prod_gas	0.15	0.13	0.32	0.39	0.37
## Prod_carb	-0.09	0.35	0.53	0.11	0.12
## Prod_comb	1.00	-0.13	-0.21	0.20	0.18
## Prod_cog	-0.13	1.00	0.39	-0.23	-0.20
## Prod_no_ren	-0.21	0.39	1.00	-0.02	-0.01
## Temp_min_Mad	0.20	-0.23	-0.02	1.00	0.94
## Temp_max_Mad	0.18	-0.20	-0.01	0.94	1.00
## Temp_min_Bar	0.16	-0.20	-0.01	0.87	0.83
## Temp_max_Bar	0.17	-0.19	-0.01	0.78	0.77
## Temp_min_Val	0.16	-0.25	-0.03	0.92	0.86
## Temp_max_Val	-0.22	0.10	-0.05	0.27	0.31
## Temp_min_Sev	0.16	-0.22	-0.01	0.93	0.87
## Temp_max_Sev	0.18	-0.19	-0.02	0.91	0.95
## Temp_min_Zar	0.17	-0.25	-0.05	0.94	0.89
## Temp_max_Zar	0.17	-0.20	-0.04	0.93	0.94
## Vel_media_Val	-0.30	0.00	-0.05	0.02	-0.04
## Vel_media_Alb	-0.34	0.07	0.02	-0.10	-0.16
## Vel_media_Zar	-0.23	0.04	-0.02	-0.13	-0.05
## Vel_media_Cor	-0.29	-0.04	-0.06	-0.13	-0.17
## Vel_media_Hue	-0.09	-0.16	-0.07	0.33	0.26
## Res_hidr	-0.51	0.03	0.06	0.12	0.17
## Der_CO2	0.50	-0.49	-0.62	0.06	0.04
## Ibex	-0.25	-0.08	0.33	0.00	0.02
## Int_ee	-0.23	0.28	0.43	0.08	0.09
## Temp_min_Bar Temp_max_Bar Temp_min_Val Temp_max_Val Temp_min_Sev					
## dia_sem	0.01	0.01	0.01	0.01	0.01
## Pre_elec	0.02	0.03	0.02	-0.22	0.02
## Dem	-0.06	-0.05	-0.08	-0.12	-0.10
## Prec_petr	0.06	0.04	0.04	0.44	0.02
## Prec_gas	0.01	0.03	0.04	-0.23	0.03
## Prec_carb	0.09	0.10	0.11	-0.23	0.09
## Prod_eol	-0.34	-0.31	-0.27	-0.17	-0.32
## Prod_sol	0.43	0.43	0.43	-0.01	0.41
## Prod_hidr	-0.32	-0.32	-0.33	0.02	-0.32
## Prod_ofr	0.09	0.09	0.09	-0.16	0.07
## Prod_nucl	0.02	0.03	-0.01	0.02	-0.02
## Prod_pet	0.36	0.32	0.36	0.35	0.35
## Prod_gas	0.39	0.36	0.41	0.36	0.39
## Prod_carb	0.13	0.11	0.08	0.46	0.09
## Prod_comb	0.16	0.17	0.16	-0.22	0.16
## Prod_cog	-0.20	-0.19	-0.25	0.10	-0.22
## Prod_no_ren	-0.01	-0.01	-0.03	-0.05	-0.01
## Temp_min_Mad	0.87	0.78	0.92	0.27	0.93
## Temp_max_Mad	0.83	0.77	0.86	0.31	0.87
## Temp_min_Bar	1.00	0.95	0.88	0.31	0.84

```

## Temp_max_Bar    0.95   1.00   0.79   0.28   0.75
## Temp_min_Val   0.88   0.79   1.00   0.28   0.91
## Temp_max_Val   0.31   0.28   0.28   1.00   0.26
## Temp_min_Sev   0.84   0.75   0.91   0.26   1.00
## Temp_max_Sev   0.81   0.75   0.84   0.30   0.87
## Temp_min_Zar   0.88   0.79   0.93   0.28   0.89
## Temp_max_Zar   0.84   0.78   0.86   0.32   0.86
## Vel_media_Val  0.03   0.01   0.11   0.11   0.03
## Vel_media_Alb  -0.09  -0.10   0.00   0.01  -0.08
## Vel_media_Zar  -0.10  -0.09  -0.06   0.01  -0.13
## Vel_media_Cor  -0.11  -0.12  -0.06   0.01  -0.11
## Vel_media_Hue  0.30   0.25   0.39   0.01   0.37
## Res_hidr       0.12   0.08   0.11   0.34   0.11
## Der_CO2        0.01   0.04   0.04  -0.42   0.03
## Ibex           0.01   0.01  -0.01   0.10  -0.01
## Int_ee         0.06   0.05   0.03  -0.23   0.07
## Temp_max_Sev  Temp_min_Zar Temp_max_Zar Vel_media_Val
## dia_sem        0.00   0.00   0.01   0.00
## Pre_elec       0.03   0.03   0.00  -0.20
## Dem            -0.07  -0.06  -0.06   0.02
## Prec_petr      0.02   0.04   0.06   0.08
## Prec_gas       0.00   0.04  -0.01  -0.10
## Prec_carb     0.10   0.13   0.10  -0.09
## Prod_eol       -0.40  -0.29  -0.38   0.58
## Prod_sol       0.60   0.48   0.54  -0.11
## Prod_hidr     -0.36  -0.33  -0.29   0.11
## Prod_ofr       0.07   0.09   0.04  -0.15
## Prod_nucl      0.01  -0.03   0.02  -0.04
## Prod_pet       0.35   0.33   0.35   0.04
## Prod_gas       0.36   0.38   0.36  -0.01
## Prod_carb     0.11   0.07   0.09  -0.19
## Prod_comb      0.18   0.17   0.17  -0.30
## Prod_cog       -0.19  -0.25  -0.20   0.00
## Prod_no_ren   -0.02  -0.05  -0.04  -0.05
## Temp_min_Mad  0.91   0.94   0.93   0.02
## Temp_max_Mad  0.95   0.89   0.94  -0.04
## Temp_min_Bar  0.81   0.88   0.84   0.03
## Temp_max_Bar  0.75   0.79   0.78   0.01
## Temp_min_Val  0.84   0.93   0.86   0.11
## Temp_max_Val  0.30   0.28   0.32   0.11
## Temp_min_Sev  0.87   0.89   0.86   0.03
## Temp_max_Sev  1.00   0.86   0.89  -0.06
## Temp_min_Zar  0.86   1.00   0.91   0.07
## Temp_max_Zar  0.89   0.91   1.00   0.05
## Vel_media_Val -0.06  -0.07   0.05   1.00
## Vel_media_Alb -0.18  -0.05  -0.10   0.63
## Vel_media_Zar -0.03  -0.02  -0.13   0.17
## Vel_media_Cor -0.17  -0.10  -0.14   0.53
## Vel_media_Hue  0.23   0.36   0.31   0.39
## Res_hidr       0.14   0.13   0.17   0.12
## Der_CO2        0.04   0.07   0.03  -0.11
## Ibex           0.01   0.00   0.00  -0.01
## Int_ee         0.07   0.05   0.08  -0.14
## Vel_media_Alb Vel_media_Zar Vel_media_Cor Vel_media_Hue Res_hidr

```

```

## dia_sem    0.03   0.01   0.00   0.01   0.00
## Pre_elec  -0.23  -0.12  -0.15  -0.06  -0.42
## Dem       0.05   0.06   0.00  -0.01   0.02
## Prec_petr  0.00   0.02   0.03  -0.03  -0.11
## Prec_gas   -0.18  -0.07  -0.05  -0.03  -0.48
## Prec_carb  -0.16  -0.05  -0.06  0.02  -0.50
## Prod_eol   0.54   0.49   0.58   0.18  -0.10
## Prod_sol   -0.21  0.08  -0.11   0.14  -0.07
## Prod_hidr  0.21   0.00   0.10   0.00   0.60
## Prod_ofr   -0.19  -0.06  -0.10  -0.06  -0.38
## Prod_nucl  -0.03   0.00  -0.05  -0.05  -0.10
## Prod_pet   0.05  -0.01  -0.07   0.08   0.23
## Prod_gas   -0.08  -0.03  -0.07   0.08   0.02
## Prod_carb  -0.18  -0.14  -0.23  -0.17   0.11
## Prod_comb   -0.34  -0.23  -0.29  -0.09  -0.51
## Prod_cog   0.07   0.04  -0.04  -0.16   0.03
## Prod_no_ren 0.02  -0.02  -0.06  -0.07   0.06
## Temp_min_Mad -0.10  -0.13  -0.13   0.33   0.12
## Temp_max_Mad -0.16  -0.05  -0.17   0.26   0.17
## Temp_min_Bar -0.09  -0.10  -0.11   0.30   0.12
## Temp_max_Bar -0.10  -0.09  -0.12   0.25   0.08
## Temp_min_Val  0.00  -0.06  -0.06   0.39   0.11
## Temp_max_Val  0.01   0.01   0.01   0.01   0.34
## Temp_min_Sev  -0.08  -0.13  -0.11   0.37   0.11
## Temp_max_Sev  -0.18  -0.03  -0.17   0.23   0.14
## Temp_min_Zar  -0.05  -0.02  -0.10   0.36   0.13
## Temp_max_Zar  -0.10  -0.13  -0.14   0.31   0.17
## Vel_media_Val  0.63   0.17   0.53   0.39   0.12
## Vel_media_Alb  1.00   0.25   0.36   0.34   0.16
## Vel_media_Zar  0.25   1.00   0.21   0.08   0.08
## Vel_media_Cor  0.36   0.21   1.00   0.19   0.08
## Vel_media_Hue  0.34   0.08   0.19   1.00   0.17
## Res_hidr     0.16   0.08   0.08   0.17   1.00
## Der_CO2      -0.17  -0.05  -0.05   0.02  -0.47
## Ibex        0.04   0.02   0.00   0.07   0.32
## Int_ee      -0.07  -0.22  -0.12   0.03   0.25
##             Der_CO2 Ibex Int_ee
## dia_sem    0.00   0.00  -0.01
## Pre_elec   0.72  -0.22  -0.31
## Dem       -0.11  0.10   0.19
## Prec_petr  0.15  -0.35  -0.50
## Prec_gas   0.80  -0.34  -0.49
## Prec_carb  0.80  -0.34  -0.46
## Prod_eol   0.14  -0.07  -0.35
## Prod_sol   0.61  -0.17  -0.20
## Prod_hidr -0.23  0.06   0.11
## Prod_ofr   0.63  -0.30  -0.23
## Prod_nucl  -0.02  -0.13  -0.09
## Prod_pet   -0.63  0.16   0.21
## Prod_gas   -0.30  0.21   0.04
## Prod_carb  -0.59  0.23   0.06
## Prod_comb   0.50  -0.25  -0.23
## Prod_cog   -0.49  -0.08  0.28
## Prod_no_ren -0.62  0.33   0.43

```

```

## Temp_min_Mad 0.06 0.00 0.08
## Temp_max_Mad 0.04 0.02 0.09
## Temp_min_Bar 0.01 0.01 0.06
## Temp_max_Bar 0.04 0.01 0.05
## Temp_min_Val 0.04 -0.01 0.03
## Temp_max_Val -0.42 0.10 -0.23
## Temp_min_Sev 0.03 -0.01 0.07
## Temp_max_Sev 0.04 0.01 0.07
## Temp_min_Zar 0.07 0.00 0.05
## Temp_max_Zar 0.03 0.00 0.08
## Vel_media_Val -0.11 -0.01 -0.14
## Vel_media_Alb -0.17 0.04 -0.07
## Vel_media_Zar -0.05 0.02 -0.22
## Vel_media_Cor -0.05 0.00 -0.12
## Vel_media_Hue 0.02 0.07 0.03
## Res_hidr -0.47 0.32 0.25
## Der_CO2 1.00 -0.33 -0.32
## Ibex -0.33 1.00 0.32
## Int_ee -0.32 0.32 1.00

# Ahora obtenemos la matriz de correlación con los p-values
df_3_p_valor <- df_3
df_3_p_valor <- select(df_3_p_valor, -fecha)

p.mat <- cor_pmat(df_3_p_valor)
head(p.mat[, 1:36])

##          dia_sem Pre_elec Dem Prec_petr Prec_gas
## dia_sem 0.000000e+00 1.712599e-03 3.204494e-107 8.729852e-01 8.394289e-01
## Pre_elec 1.712599e-03 0.000000e+00 9.535909e-01 2.067750e-36 0.000000e+00
## Dem 3.204494e-107 9.535909e-01 0.000000e+00 1.073044e-37 7.341051e-13
## Prec_petr 8.729852e-01 2.067750e-36 1.073044e-37 0.000000e+00 1.219460e-84
## Prec_gas 8.394289e-01 0.000000e+00 7.341051e-13 1.219460e-84 0.000000e+00
## Prec_carb 9.885705e-01 0.000000e+00 2.161140e-15 8.989263e-133 0.000000e+00
##          Prec_carb Prod_eol Prod_sol Prod_hidr Prod_ofr
## dia_sem 9.885705e-01 5.264282e-01 8.412514e-01 7.398600e-16 5.539972e-03
## Pre_elec 0.000000e+00 1.040718e-02 6.974269e-168 2.370609e-55 2.130843e-304
## Dem 2.161140e-15 6.723381e-03 1.778804e-15 6.712384e-22 8.338208e-01
## Prec_petr 8.989263e-133 3.274217e-02 2.774210e-21 9.680039e-01 6.291329e-72
## Prec_gas 0.000000e+00 1.630211e-12 4.882187e-191 6.198769e-42 7.770897e-261
## Prec_carb 0.000000e+00 1.965995e-05 0.000000e+00 2.161979e-66 7.673231e-270
##          Prod_nucl Prod_pet Prod_gas Prod_carb Prod_comb
## dia_sem 3.398439e-01 9.704238e-07 1.981336e-64 5.999496e-13 1.820782e-42
## Pre_elec 8.391289e-02 3.304890e-95 6.371208e-09 3.228570e-46 2.324042e-254
## Dem 3.496470e-12 1.055992e-43 5.690459e-64 2.073751e-56 4.186523e-48
## Prec_petr 2.687722e-02 6.086389e-12 4.198754e-20 1.518482e-01 2.242913e-06
## Prec_gas 1.994777e-01 4.738030e-149 2.542034e-11 1.473820e-114 3.553499e-216
## Prec_carb 7.937180e-02 2.474058e-120 1.799452e-06 7.582775e-100 1.347674e-196
##          Prod_cog Prod_no_ren Temp_min_Mad Temp_max_Mad Temp_min_Bar
## dia_sem 2.684194e-32 1.871221e-09 9.834619e-01 7.995671e-01 5.639370e-01
## Pre_elec 2.001326e-90 1.279795e-126 9.441048e-03 7.464837e-02 2.813199e-01
## Dem 2.344264e-53 5.117111e-86 7.877961e-03 1.126555e-02 1.261760e-04
## Prec_petr 2.780544e-04 3.763791e-222 1.729928e-01 1.024548e-01 8.376241e-04
## Prec_gas 2.518427e-274 3.788552e-242 1.816390e-02 9.009655e-01 4.395331e-01
## Prec_carb 5.217601e-276 5.394803e-231 1.895445e-13 9.046010e-10 9.809251e-09

```

```

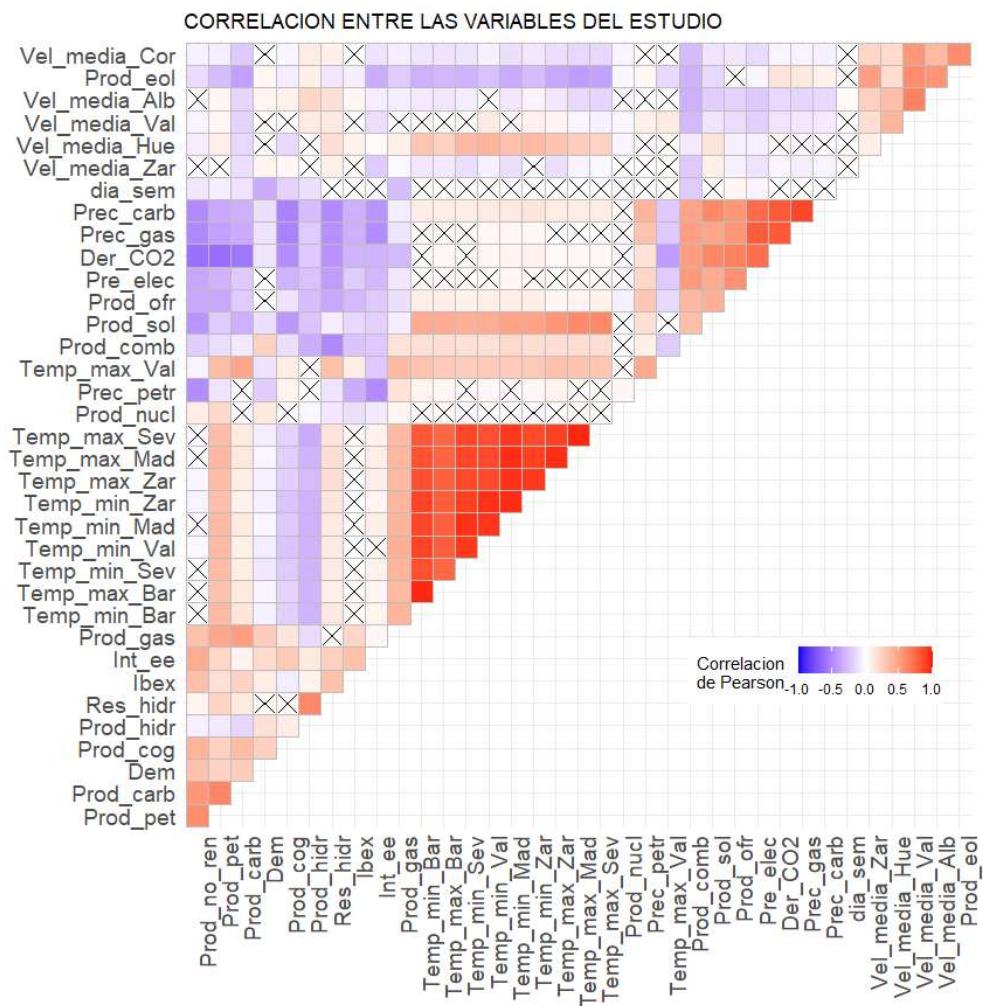
##      Temp_max_Bar Temp_min_Val Temp_max_Val Temp_min_Sev Temp_max_Sev
## dia_sem 4.685177e-01 5.238682e-01 7.258903e-01 5.830649e-01 8.355053e-01
## Pre_elec 7.463746e-02 2.527341e-01 4.616718e-42 2.079737e-01 6.102124e-02
## Dem     1.163211e-03 3.526798e-06 9.736030e-14 6.381616e-09 8.753164e-06
## Prec_petr 1.653705e-02 1.199611e-02 7.900065e-177 3.491451e-01 1.419122e-01
## Prec_gas 9.022105e-02 7.569885e-03 3.007222e-43 9.480825e-02 7.755323e-01
## Prec_carb 6.774405e-10 3.905795e-12 2.790735e-46 1.271580e-08 1.339867e-09
##      Temp_min_Zar Temp_max_Zar Vel_media_Val Vel_media_AlB Vel_media_Zar
## dia_sem 9.669747e-01 6.896612e-01 8.411632e-01 4.366703e-02 5.801100e-01
## Pre_elec 5.864310e-02 9.589161e-01 4.485484e-33 8.877967e-45 1.514708e-12
## Dem     9.467853e-05 3.526388e-04 2.246943e-01 2.198793e-03 5.311382e-04
## Prec_petr 2.359065e-02 7.984137e-04 1.292625e-06 7.678767e-01 1.611358e-01
## Prec_gas 1.647623e-02 6.471513e-01 6.223541e-10 7.979022e-28 1.414822e-05
## Prec_carb 3.595691e-16 4.066453e-10 5.662891e-08 7.045604e-23 2.268582e-03
##      Vel_media_Cor Vel_media_Hue Res_hidr Der_CO2 Ibex
## dia_sem 9.867985e-01 0.475997062 9.970522e-01 9.734632e-01 8.931381e-01
## Pre_elec 2.295160e-19 0.000207114 2.280448e-159 0.000000e+00 7.950135e-41
## Dem     7.897611e-01 0.704217222 2.295585e-01 6.010123e-11 2.068301e-10
## Prec_petr 8.698241e-02 0.092824120 1.351402e-11 1.407869e-20 2.034374e-106
## Prec_gas 2.963192e-03 0.068238120 5.537307e-205 0.000000e+00 1.969502e-98
## Prec_carb 2.694064e-04 0.270588264 1.600041e-226 0.000000e+00 1.248603e-98
##      Int_ee
## dia_sem 3.955868e-01
## Pre_elec 8.016590e-85
## Dem     1.457975e-30
## Prec_petr 3.639674e-229
## Prec_gas 1.091276e-217
## Prec_carb 6.759206e-187

ggcorrplot(correlacion, hc.order = TRUE, type = "upper", tl.srt = 90, title = "CORRELACION ENTRE LAS VARIABLES DEL ESTUDIO", tl.cex = 14, legend.title = "Correlacion \nde Pearson", outline.color = "grey", p.mat = p.mat) +
  theme (legend.position=c (0.8, 0.2), legend.background = element_rect (color="white", fill = "white")) +
  theme (legend.direction = "horizontal")

```

MODELOS PARA LA PREDICCION DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL



```
#library(readr)
#setwd("D:/0.BIG DATA/0.MASTER/1.MASTER Y CURSOS/1.MODULOS MASTER/11.MODULO XI - Tra
bajo Final de Master/1.TFM/5.CODIGO")

#correlacion <- as.data.frame(correlacion)
#p.mat <- as.data.frame(p.mat)

#write_csv(correlacion, "Correlacion_entre_var.csv")
#write_csv(p.mat, "P_valor_entre_var.csv")
```

ANEXO V: PROCESADO OUTLIERS (ELIM)

o

PROCESADO DE LOS OUTLIERS

Como el numero de variables que presentan valores atípicos es bastante elevado se ha creado este apartado para tratar este problema.

Las variables que presentan este tipo de valores son:

- Dem: 5 valores (0.137%)
- Prec_gas: 516 valores (14.129%)
- Prec_car: 468 valores (12.815%)
- Prod_eol: 33 valores (0.904%)
- Prod_sol: 207 valores (5.668%)
- Prod_hidr: 172 valores (4.710%)
- Prod_ofr: 323 valores (8.844%)
- Prod_nucl: 67 valores (1.835%)
- Prod_pet: 20 valores (0.548%)
- Prod_gas: 4 valores (0.110%)
- Prod_comb: 160 valores (4.381%)
- Prod_cog: 248 valores (6.791%)
- Vel_media_Val: 197 valores (5.394%)
- Vel_media_Alb: 101 valores (2.766%)
- Vel_media_Zar: 22 valores (0.602%)
- Vel_media_Cor: 22 valores (0.602%)
- Vel_media_Hue: 76 valores (2.081%)
- Der_CO2: 505 valores (13.828%)
- Ibex: 5 valores (0.137%)
- Int_ee: 24 valores (0.657%)

Cargamos las librerías necesarias

```
In [2]: import pandas as pd
from scipy import stats
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Cargamos los datos

```
In [3]: df_def = pd.read_csv(r'C:\Users\User\1.PYTHON\00.TFM\00.Archivos_utiliz
In [4]: print('Las dimensiones de los datos "df_def" son ' + str(df_def.shape))
Las dimensiones de los datos "df_def" son (3652, 28)
```

Outliers en variable "Dem"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
count      3652.00000
mean      28045.150005
std       2781.353082
min      19122.490000
25%     26232.872500
50%     28194.870000
75%     29903.192500
max      35306.410000
Name: Dem, dtype: float64
```

Para eliminar los valores extremos vamos a emplear el metodo de los percentiles

```
In [297]: # Para obtener el Rango Intercuartilico tenemos que calcular la diferencia
# Luego en base a esto calcularemos los valores mínimos y máximos para que
# serán descartadas.
```

```
In [298]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out = df_def.copy()
df_def_out_Q1 = df_def_out['Dem'].quantile(0.25)
df_def_out_Q3 = df_def_out['Dem'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Dem" en "train" es ' + str(df_def_out_LR))
print ('El valor Maximo para "Dem" en "train" es ' + str(df_def_out_UR))

El valor Minimo para "Dem" en "train" es 20727.3925
El valor Maximo para "Dem" en "train" es 35408.6725
```

```
In [299]: # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Dem > df_def_out_UR) | (df_def_out.Dem < df_def_out_LR)], axis=0, inplace=True)

# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (3647, 28)
```

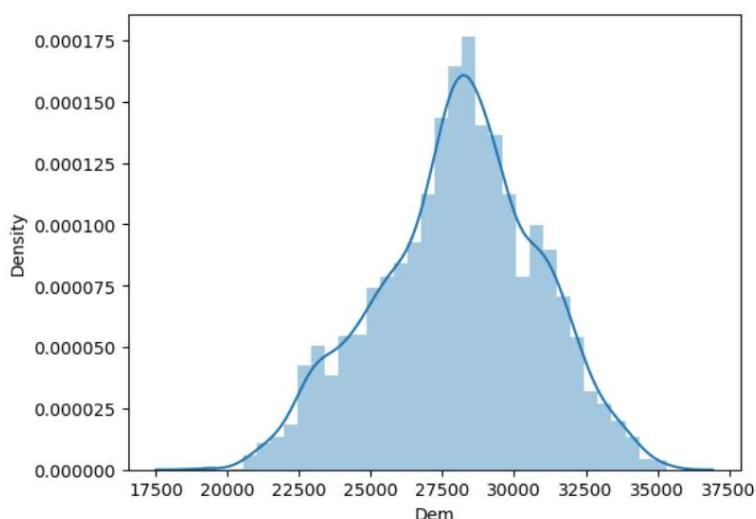
```
In [300]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Dem'])

Out[300]: <AxesSubplot: xlabel='Dem', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

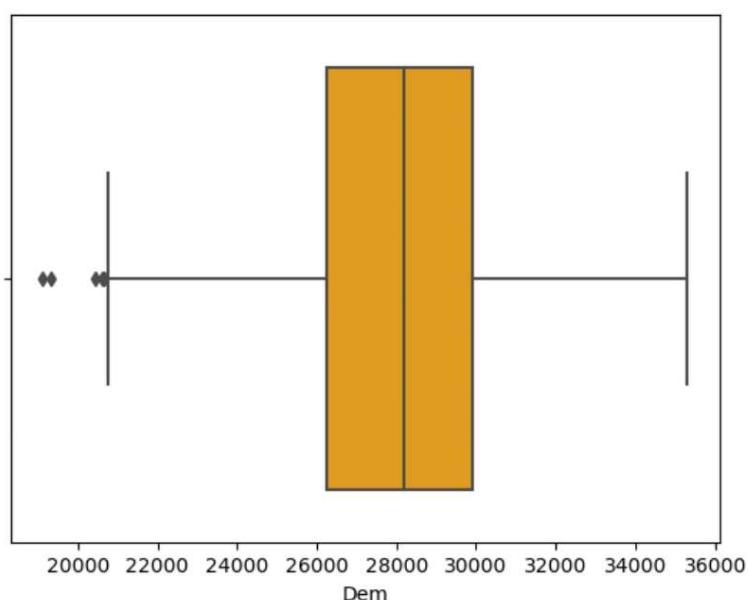
```
In [5]: # Obtenemos la distribución de la variable "Dem" en "train" y en "test"
sns.distplot(df_def['Dem'])
```

```
Out[5]: <AxesSubplot:xlabel='Dem', ylabel='Density'>
```



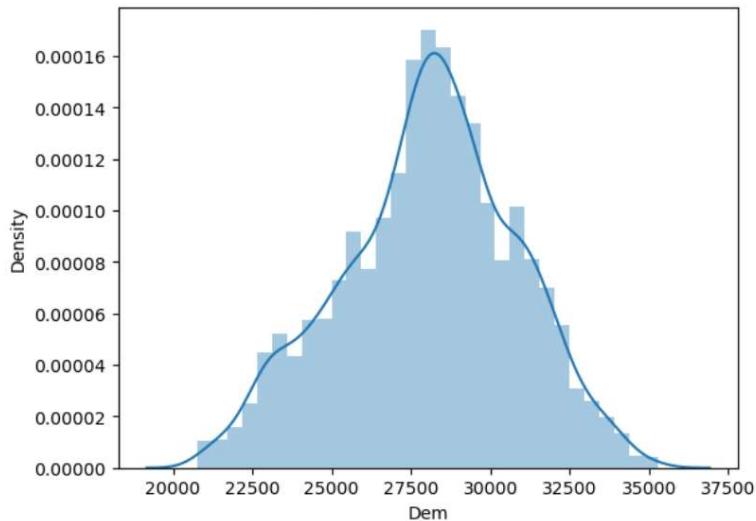
```
In [7]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def['Dem'], color = "orange")
```

```
Out[7]: <AxesSubplot:xlabel='Dem'>
```

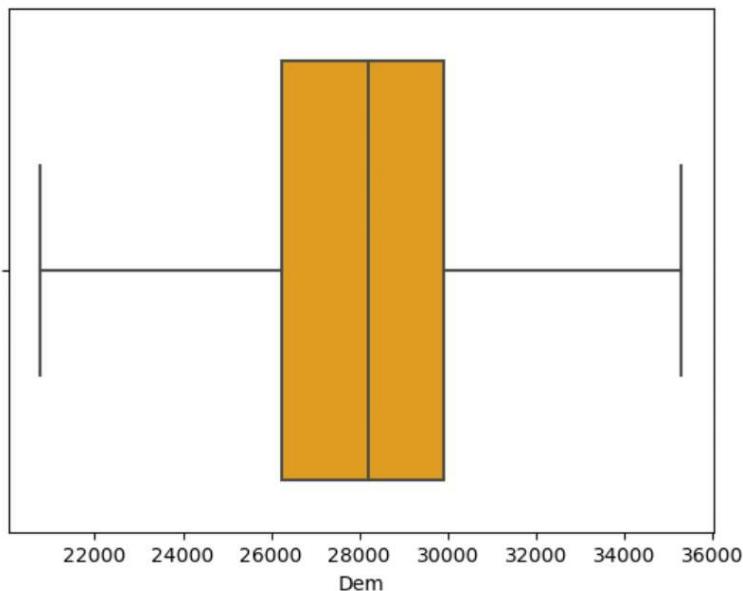


```
In [296...]: # Obtenemos las estadísticas de los datos
print(df_def.Dem.describe())
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL



```
In [301]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Dem'], color = "orange")
Out[301]: <AxesSubplot:xlabel='Dem'>
```

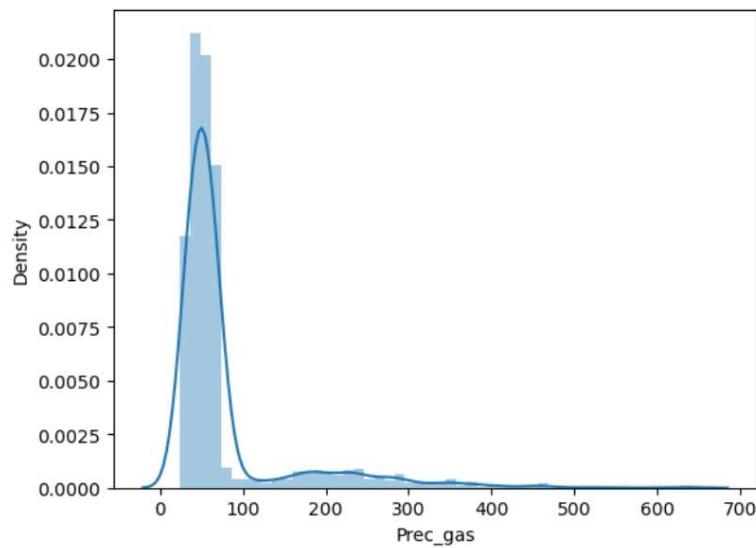


Outliers en variable "Prec_gas"

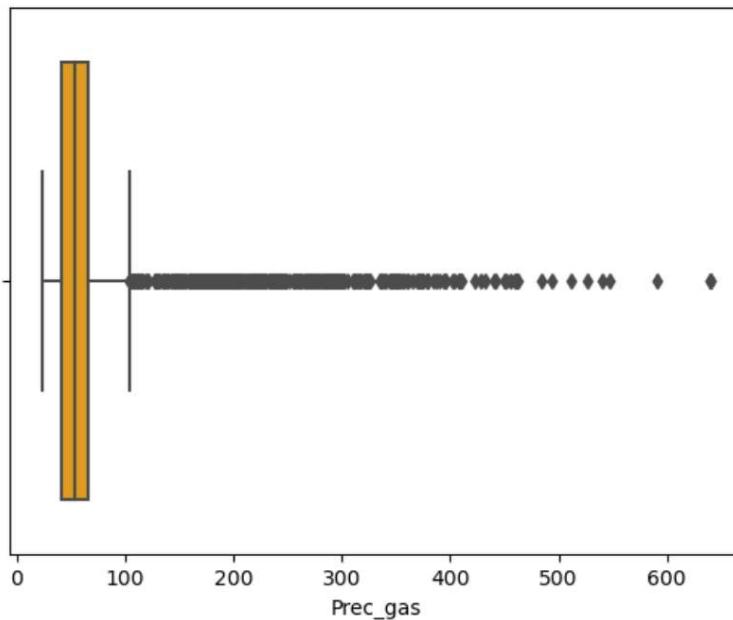
```
In [302]: # Obtenemos la distribución de la variable "Prec_gas" en "train" y en "t
sns.distplot(df_def_out['Prec_gas'])
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[302]: <AxesSubplot:xlabel='Prec_gas', ylabel='Density'>
```



```
In [303...]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prec_gas'], color = "orange")  
Out[303]: <AxesSubplot:xlabel='Prec_gas'>
```



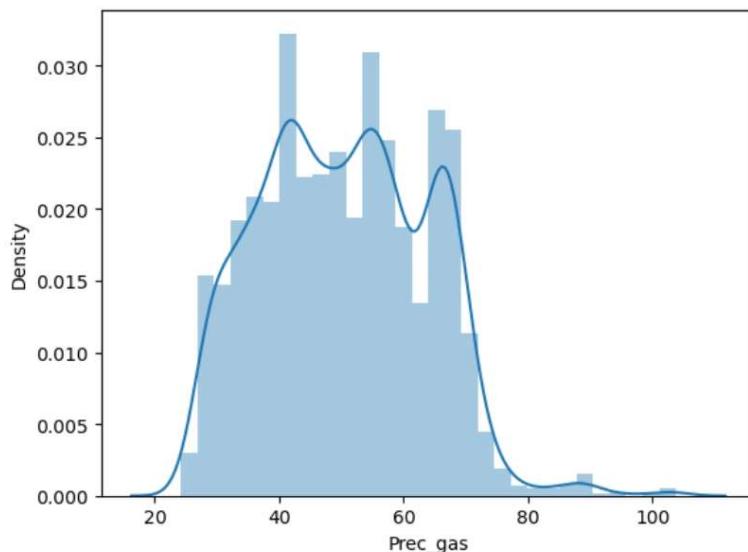
```
In [ ]: # Podríamos llegar a una rápida conclusión, de que los valores de "Prec_  
# valores fuera de rango, esto puede generar "ruido" en nuestro análisis  
# deshacernos de esos valores y volver a graficar.
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

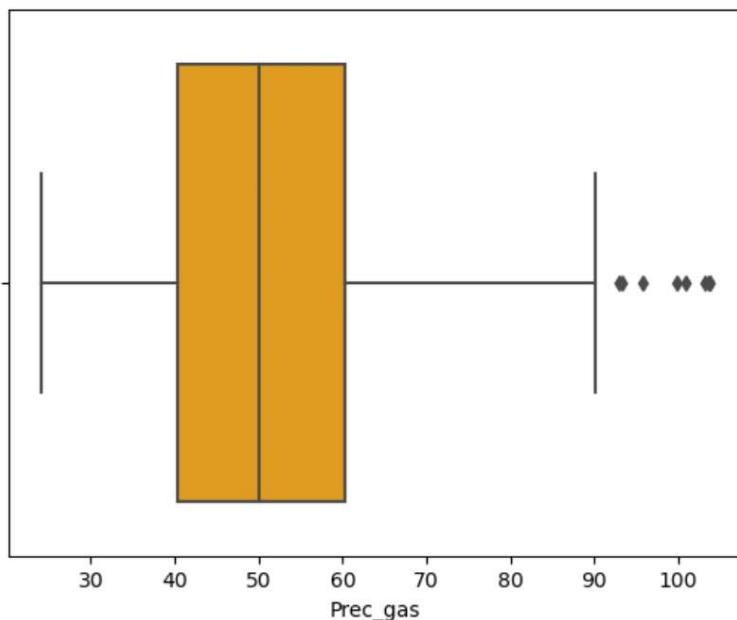
EN EL MERCADO ESPAÑOL

```
# Para eliminar estos valores empleamos el metodo de Los percentiles
In [304...]: # Obtenemos Las estadisticas de Los datos
print(df_def_out.Prec_gas.describe())
count      3647.000000
mean       77.721626
std        77.061969
min       24.180000
25%      41.610000
50%      53.870000
75%      66.650000
max      640.360000
Name: Prec_gas, dtype: float64
In [305...]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prec_gas'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prec_gas'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prec_gas" en "train" es '+ str(df_def_out_
print ('El valor Maximo para "Prec_gas" en "train" es '+ str(df_def_out_
El valor Minimo para "Prec_gas" en "train" es 4.05
El valor Maximo para "Prec_gas" en "train" es 104.21
In [306...]: # Eliminamos Los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prec_gas > df_def_out_UR) | (df_def_out.Prec_gas < df_def_out_LR)], axis=0, inplace=True)
# Obtenemos Las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))
Las nuevas dimensiones de los datos "train" son (3134, 28)
In [307...]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prec_gas'])
Out[307]: <AxesSubplot:xlabel='Prec_gas', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL



```
In [308]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prec_gas'], color = "orange")
Out[308]: <AxesSubplot:xlabel='Prec_gas'>
```



```
In [ ]:
```

Outliers en variable "Prec_carb"

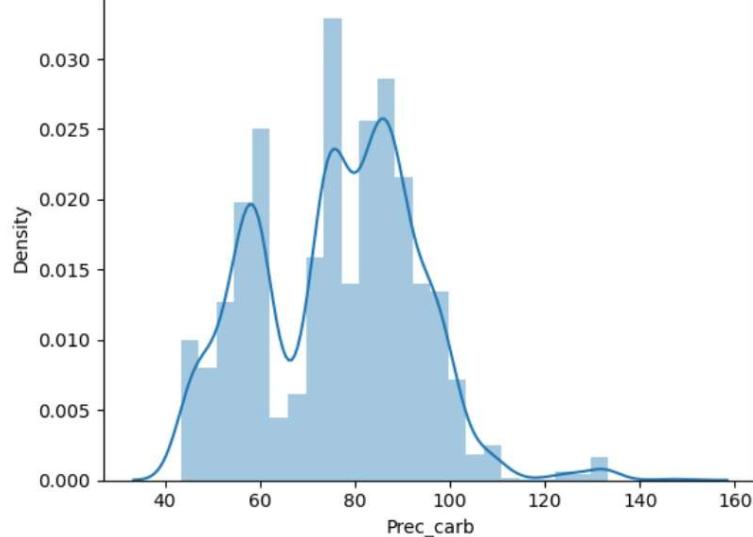
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [309]:
```

```
# Obtenemos la distribución de la variable "Prec_carb" en "train" y en "test"
sns.distplot(df_def_out['Prec_carb'])
```

```
Out[309]:
```

```
<AxesSubplot:xlabel='Prec_carb', ylabel='Density'>
```

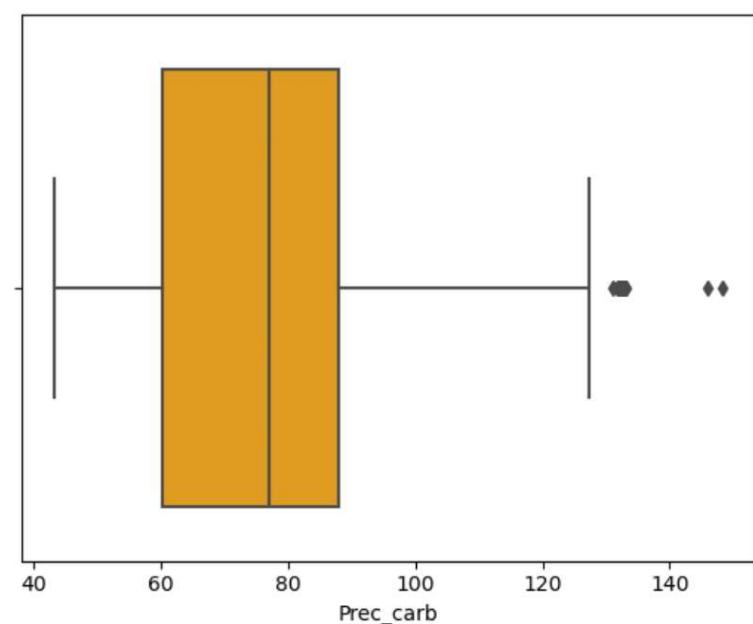


```
In [310]:
```

```
# Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prec_carb'], color = "orange")
```

```
Out[310]:
```

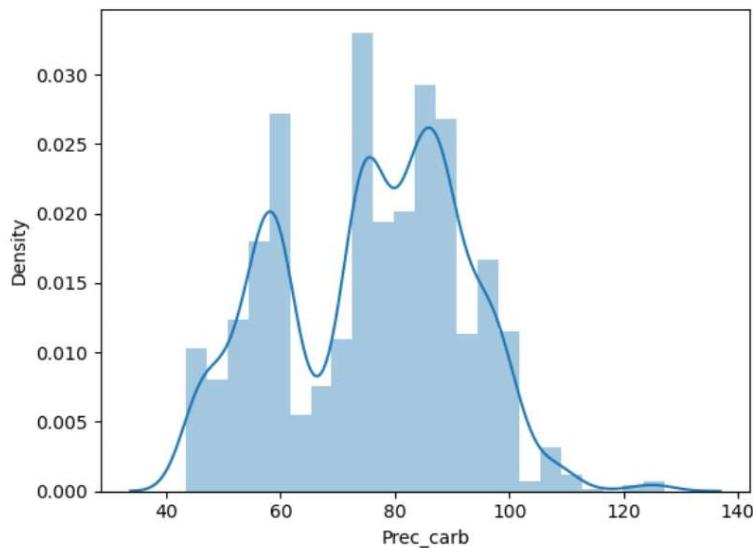
```
<AxesSubplot:xlabel='Prec_carb'>
```



MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [311... # Obtenemos Las estadísticas de Los datos
print(df_def_out.Prec_carb.describe())
count    3134.00000
mean     76.14627
std      16.74241
min      43.40000
25%     60.36250
50%     76.95000
75%     88.00000
max     148.35000
Name: Prec_carb, dtype: float64
In [312... # Realizamos el filtrado intercuartílico en Los datos "train"
df_def_out_Q1 = df_def_out['Prec_carb'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prec_carb'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prec_carb" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prec_carb" en "train" es '+ str(df_def_out_UR))
El valor Minimo para "Prec_carb" en "train" es 18.9062
El valor Maximo para "Prec_carb" en "train" es 129.4562
In [313... # Eliminamos Los Outliers en Los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prec_carb > df_def_out_UR) | (df_
# Obtenemos Las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))
Las nuevas dimensiones de los datos "train" son (3113, 28)
In [314... # Volvemos a graficar La variable para comprobar el cambio
sns.distplot(df_def_out['Prec_carb'])
Out[314]: <AxesSubplot:xlabel='Prec_carb', ylabel='Density'>
```

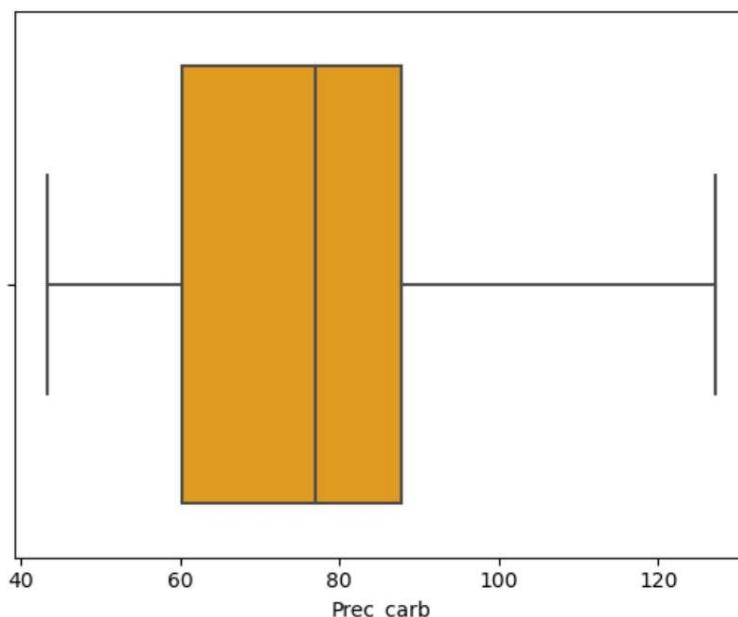


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [315... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers  
sns.boxplot(x = df_def_out['Prec_carb'], color = "orange")
```

```
Out[315]: <AxesSubplot:xlabel='Prec_carb'>
```

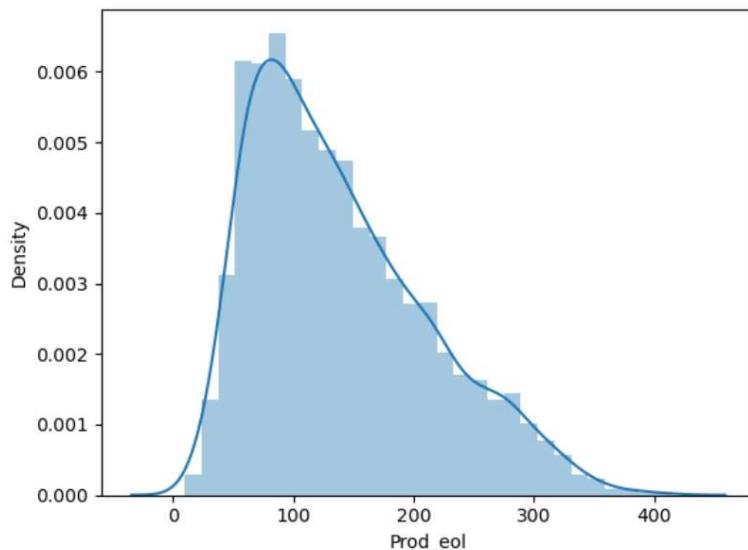


Outliers en variable "Prod_eol"

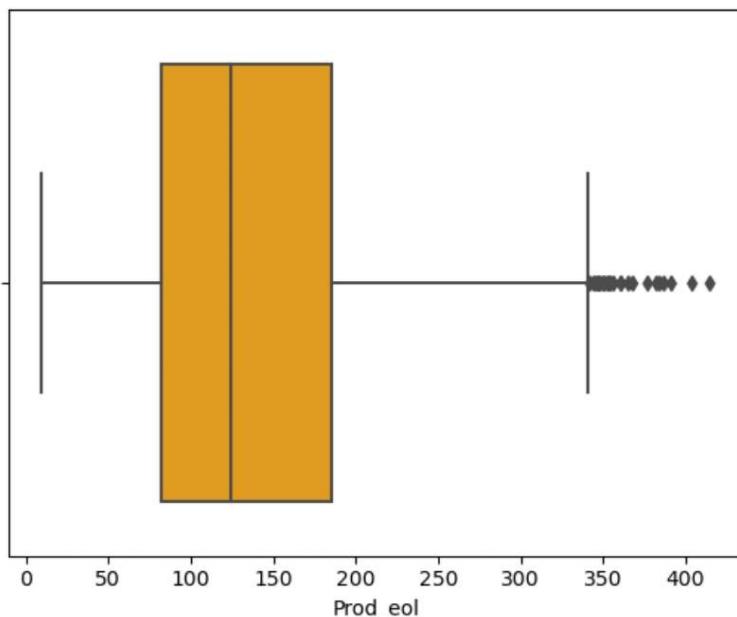
```
In [316... # Obtenemos La distribucion de la variable "Prod_eol" en "train" y en "t  
sns.distplot(df_def_out['Prod_eol'])
```

```
Out[316]: <AxesSubplot:xlabel='Prod_eol', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL



```
In [317]: # Dibujamos el boxplot para Los datos "train"  
sns.boxplot(x = df_def_out['Prod_eol'], color = "orange")  
  
Out[317]: <AxesSubplot:xlabel='Prod_eol'>
```



```
In [318]: # Obtenemos Las estadísticas de Los datos  
print(df_def_out.Prod_eol.describe())
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
count      3113.000000
mean      139.654105
std       73.851818
min       9.687570
25%      81.628354
50%      124.579640
75%      185.386245
max      414.579447
Name: Prod_eol, dtype: float64
```

```
In [319...]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_eol'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_eol'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_eol" en "train" es ' + str(df_def_out_
print ('El valor Maximo para "Prod_eol" en "train" es ' + str(df_def_out_
```

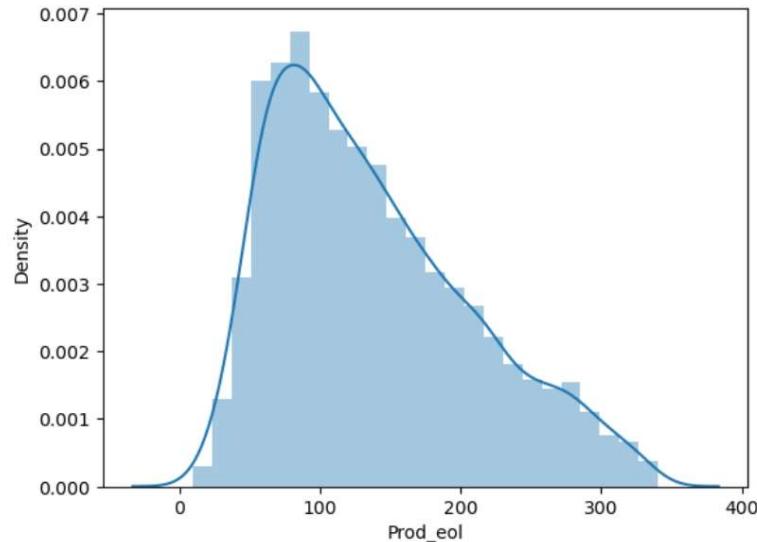
```
El valor Minimo para "Prod_eol" en "train" es -74.0085
El valor Maximo para "Prod_eol" en "train" es 341.0231
```

```
In [320...]: # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_eol > df_def_out.UR) | (df_c
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out
```

```
Las nuevas dimensiones de los datos "train" son (3089, 28)
```

```
In [321...]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_eol'])
```

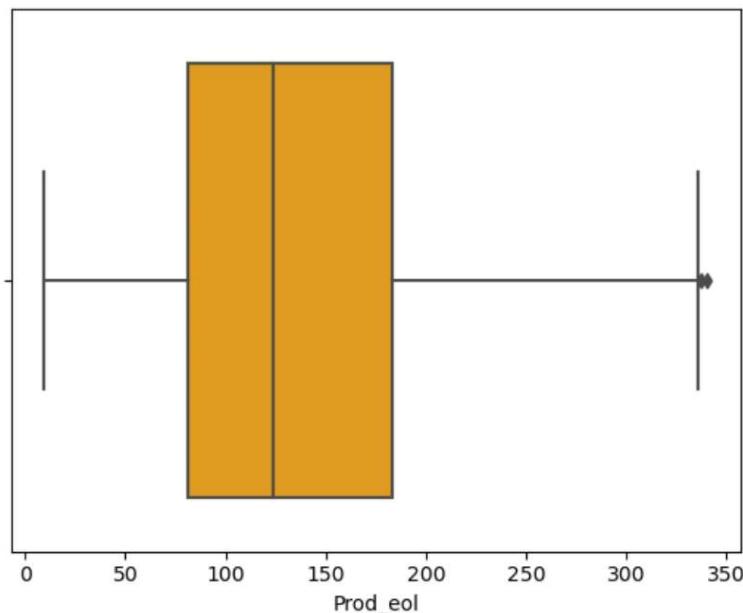
```
Out[321]: <AxesSubplot:xlabel='Prod_eol', ylabel='Density'>
```



```
In [322...]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_eol'], color = "orange")
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

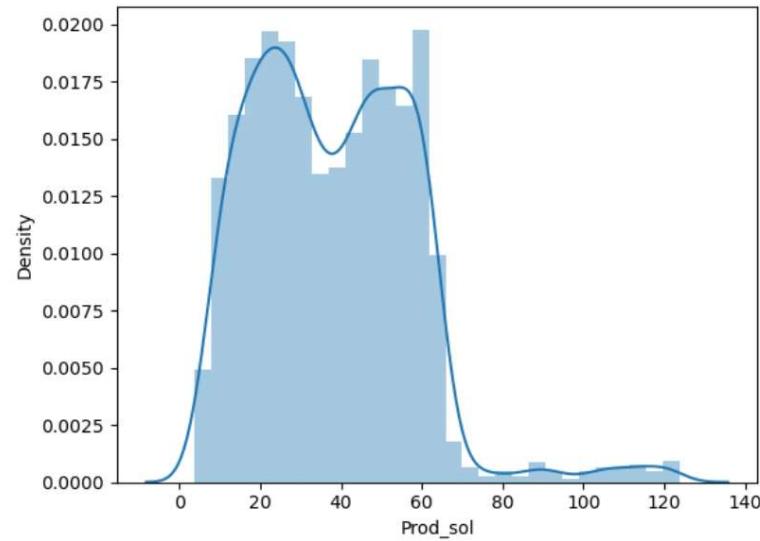
```
Out[322]: <AxesSubplot:xlabel='Prod_eol'>
```



Outliers en variable "Prod_sol"

```
In [323... # Obtenemos La distribucion de La variable "Prod_sol" en "train" y en "t
sns.distplot(df_def_out['Prod_sol'])
```

```
Out[323]: <AxesSubplot:xlabel='Prod_sol', ylabel='Density'>
```

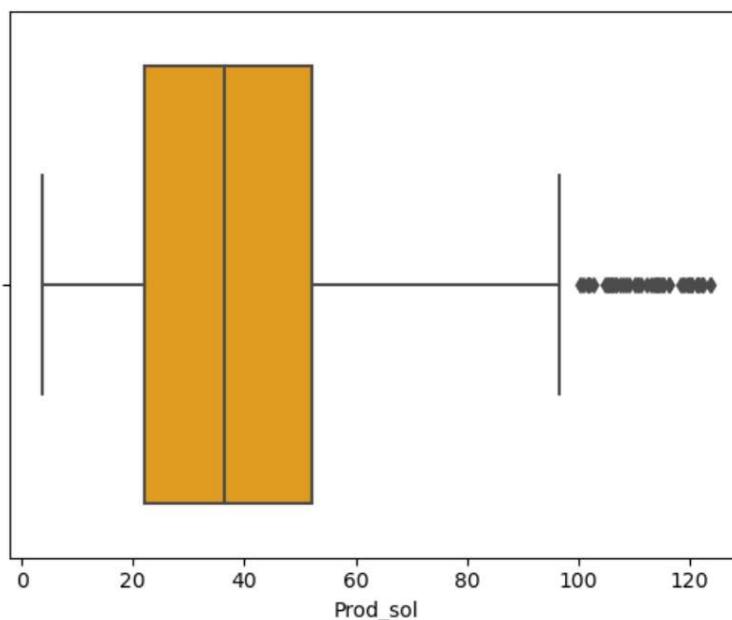


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [324... # Dibujamos el boxplot para Los datos "train"
sns.boxplot(x = df_def_out['Prod_sol'], color = "orange")
```

```
Out[324]: <AxesSubplot:xlabel='Prod_sol'>
```



```
In [325... # Obtenemos Las estadísticas de Los datos
print(df_def_out.Prod_sol.describe())
```

```
count    3089.000000
mean     37.892831
std      19.945902
min      3.794730
25%     21.996246
50%     36.501100
75%     52.193258
max     123.802665
Name: Prod_sol, dtype: float64
```

```
In [326... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_sol'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_sol'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_sol" en "train" es '+ str(df_def_out_
print ('El valor Maximo para "Prod_sol" en "train" es '+ str(df_def_out_
```

El valor Minimo para "Prod_sol" en "train" es -23.2993
El valor Maximo para "Prod_sol" en "train" es 97.4888

```
In [327... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_sol > df_def_out_UR) | (df_c
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

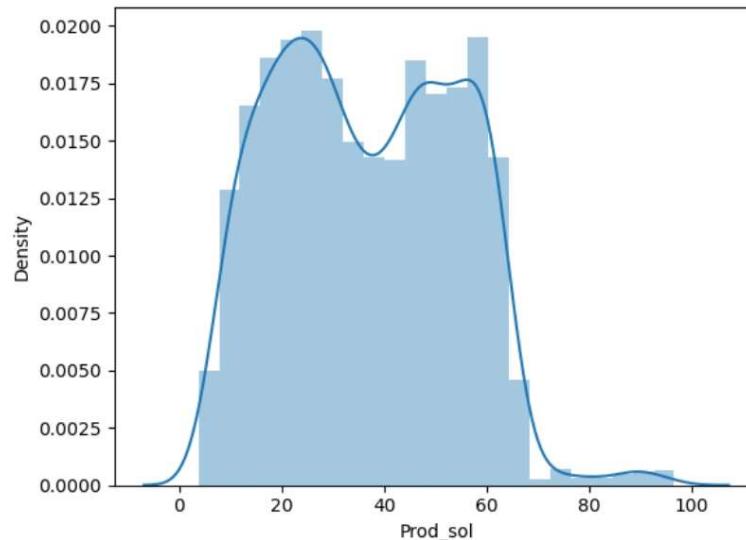
EN EL MERCADO ESPAÑOL

```
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (3039, 28)

In [328]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_sol'])

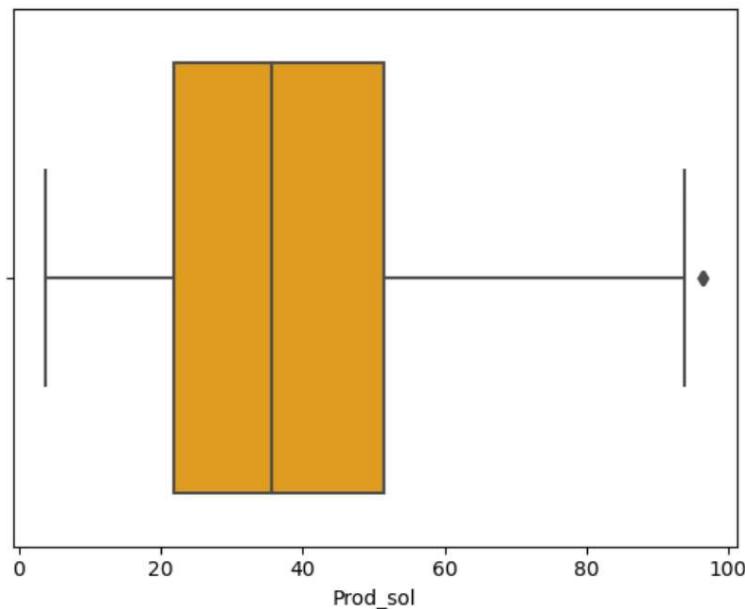
Out[328]: <AxesSubplot:xlabel='Prod_sol', ylabel='Density'>
```



```
In [329]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_sol'], color = "orange")

Out[329]: <AxesSubplot:xlabel='Prod_sol'>
```

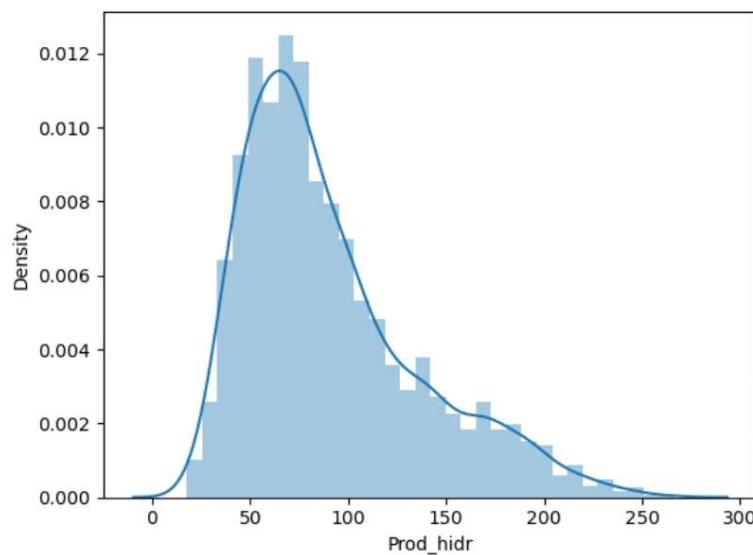
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



Outliers en variable "Prod_hidr"

```
In [330... # Obtenemos la distribución de la variable "Prod_hidr" en "train" y en "test"
sns.distplot(df_def_out['Prod_hidr'])

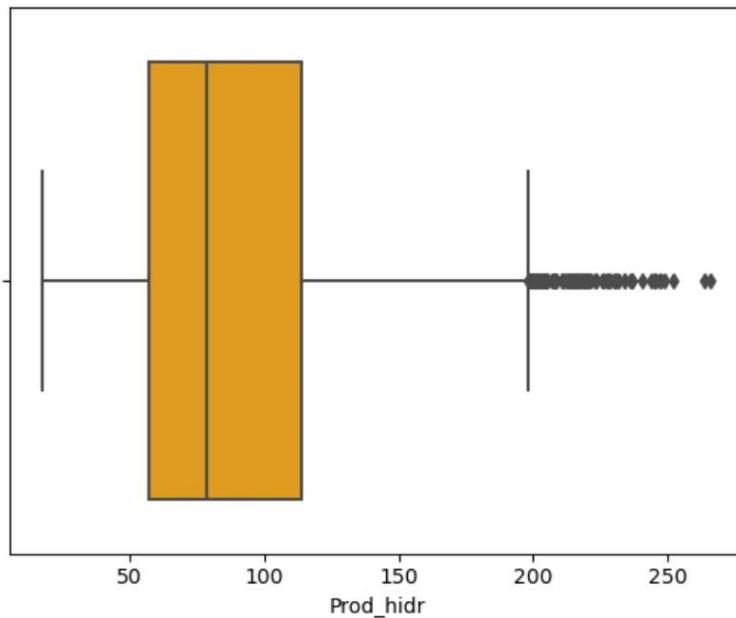
Out[330]: <AxesSubplot: xlabel='Prod_hidr', ylabel='Density'>
```



```
In [331... # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prod_hidr'], color = "orange")
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

```
Out[331]: <AxesSubplot:xlabel='Prod_hidr'>
```



```
In [332... # Obtenemos las estadísticas de los datos  
print(df_def_out.Prod_hidr.describe())
```

```
count    3039.000000  
mean     90.889001  
std      45.317254  
min      17.719687  
25%     57.308599  
50%     78.604563  
75%     113.717420  
max     266.074453  
Name: Prod_hidr, dtype: float64
```

```
In [333... # Realizamos el filtrado intercuartílico en los datos "train"  
df_def_out_Q1 = df_def_out['Prod_hidr'].quantile(0.25)  
df_def_out_Q3 = df_def_out['Prod_hidr'].quantile(0.75)  
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1  
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)  
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)  
print ('El valor Minimo para "Prod_hidr" en "train" es '+ str(df_def_out_LR))  
print ('El valor Maximo para "Prod_hidr" en "train" es '+ str(df_def_out_UR))
```

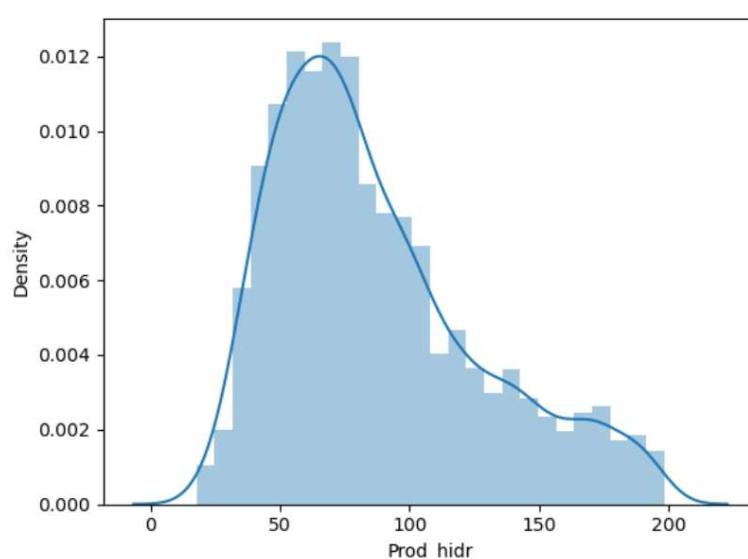
El valor Minimo para "Prod_hidr" en "train" es -27.3046
El valor Maximo para "Prod_hidr" en "train" es 198.3306

```
In [334... # Eliminamos los Outliers en los datos "train"  
df_def_out.drop(df_def_out[(df_def_out.Prod_hidr > df_def_out_UR) | (df_def_out.Prod_hidr < df_def_out_LR)], axis=0, inplace=True)  
  
# Obtenemos las nuevas dimensiones  
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))
```

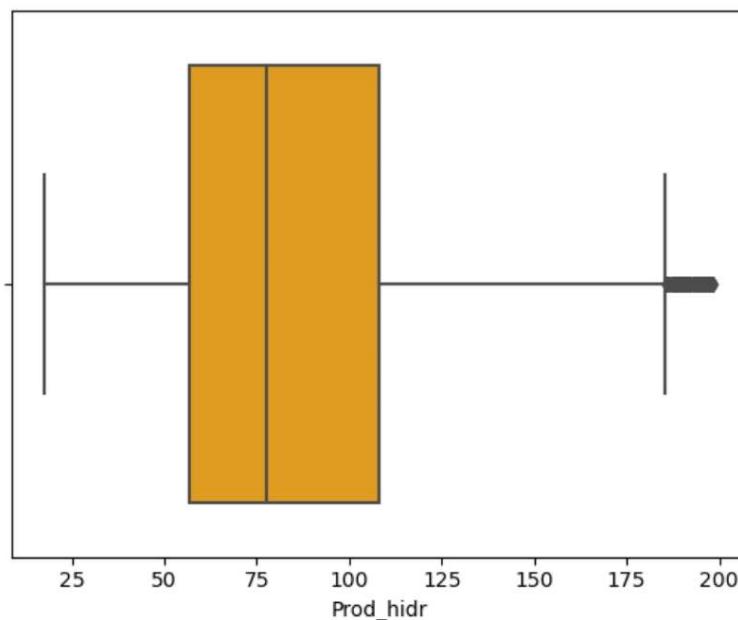
Las nuevas dimensiones de los datos "train" son (2951, 28)

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [335]: # Volvemos a graficar la variable para comprobar el cambio  
sns.distplot(df_def_out['Prod_hidr'])  
Out[335]: <AxesSubplot:xlabel='Prod_hidr', ylabel='Density'>
```



```
In [336]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Prod_hidr'], color = "orange")  
Out[336]: <AxesSubplot:xlabel='Prod_hidr'>
```

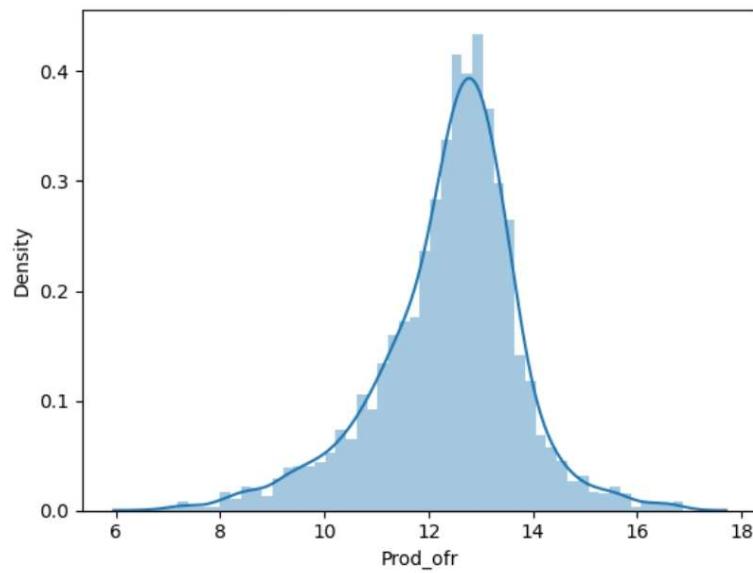


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

Outliers en variable "Prod_ofr"

```
In [337]: # Obtenemos la distribución de la variable "Prod_ofr" en "train" y en "t
sns.distplot(df_def_out['Prod_ofr'])

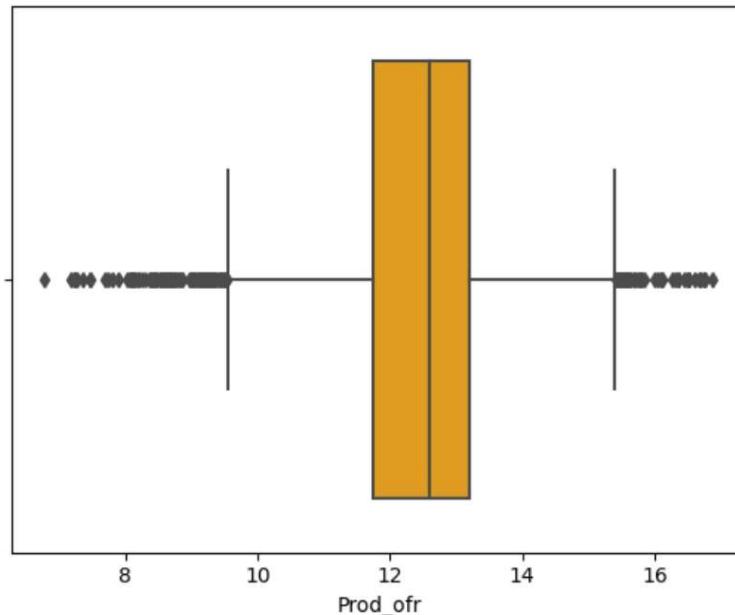
Out[337]: <AxesSubplot:xlabel='Prod_ofr', ylabel='Density'>
```



```
In [338]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prod_ofr'], color = "orange")

Out[338]: <AxesSubplot:xlabel='Prod_ofr'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [339... # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_ofr.describe())
count    2951.000000
mean      12.394471
std       1.354041
min       6.780660
25%      11.737096
50%      12.586330
75%      13.207234
max      16.882393
Name: Prod_ofr, dtype: float64

In [340... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_ofr'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_ofr'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_ofr" en "train" es '+ str(df_def_out_
print ('El valor Maximo para "Prod_ofr" en "train" es '+ str(df_def_out_
El valor Minimo para "Prod_ofr" en "train" es 9.5319
El valor Maximo para "Prod_ofr" en "train" es 15.4124

In [341... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_ofr > df_def_out_UR) | (df_def_out_Prod_ofr < df_def_out_LR)], axis=0, inplace=True)

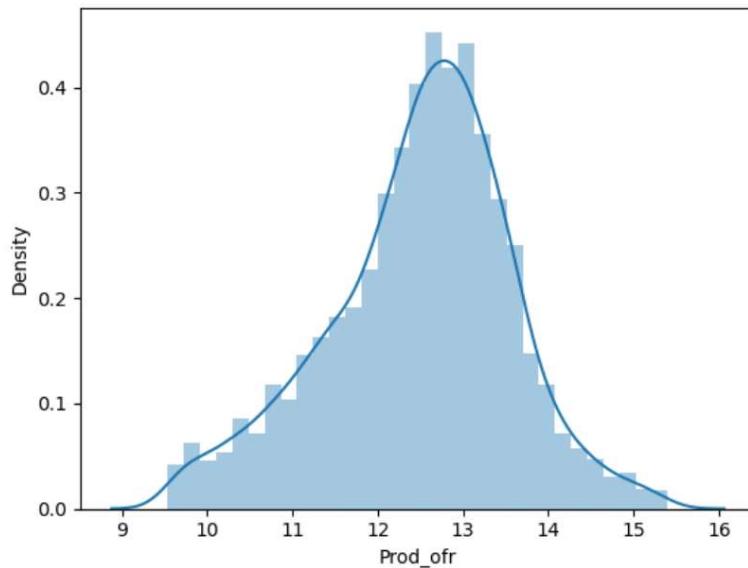
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2793, 28)

In [342... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_ofr'])
```

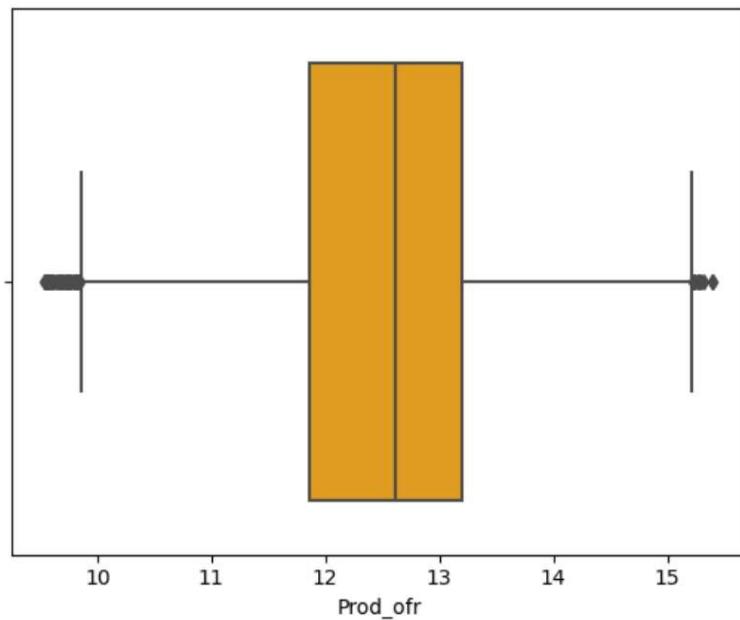
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[342]: <AxesSubplot:xlabel='Prod_ofr', ylabel='Density'>
```



```
In [343... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers
sns.boxplot(x = df_def_out['Prod_ofr'], color = "orange")
```

```
Out[343]: <AxesSubplot:xlabel='Prod_ofr'>
```

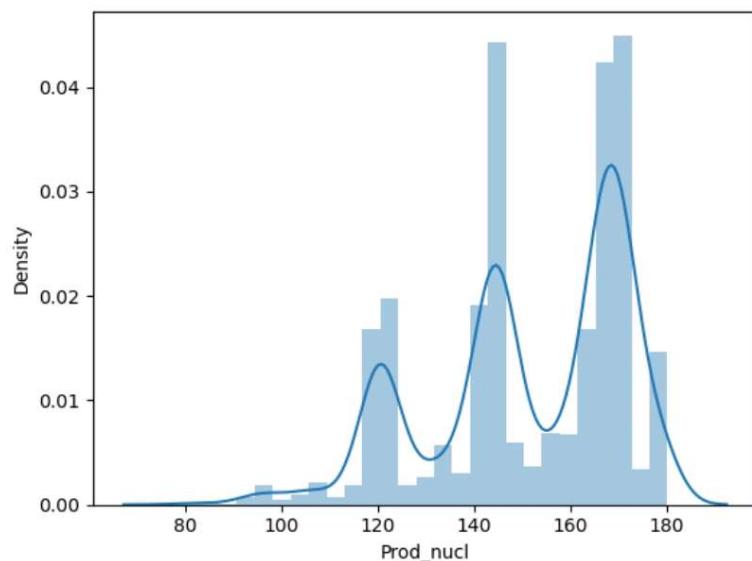


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

Outliers en variable "Prod_nucl"

```
In [344]: # Obtenemos la distribución de la variable "Prod_nucl" en "train" y en "test"
sns.distplot(df_def_out['Prod_nucl'])

Out[344]: <AxesSubplot:xlabel='Prod_nucl', ylabel='Density'>
```

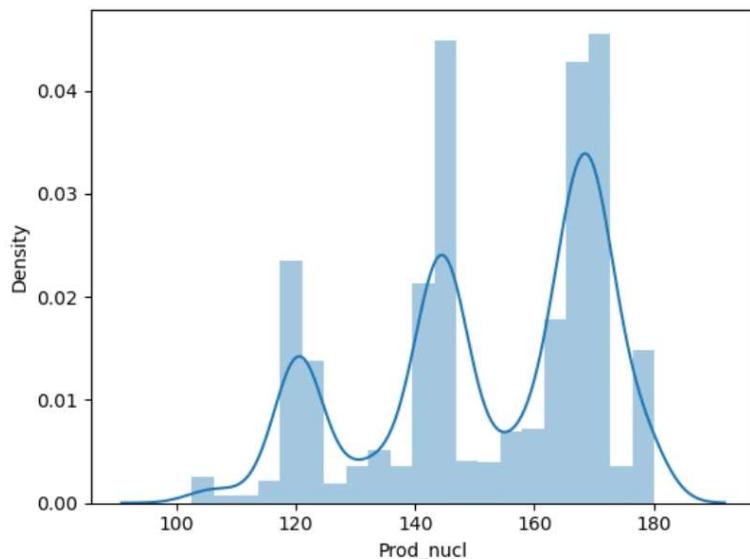


```
In [345]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prod_nucl'], color = "orange")

Out[345]: <AxesSubplot:xlabel='Prod_nucl'>
```

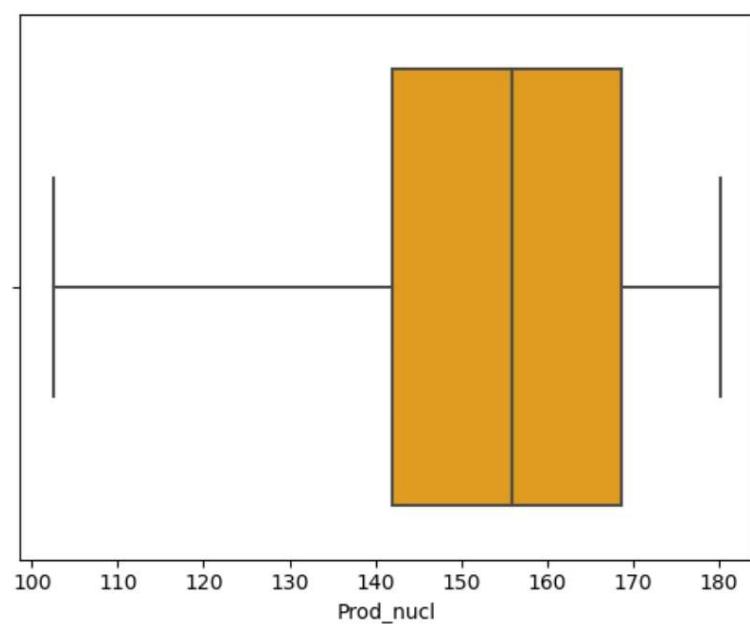
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[349]: <AxesSubplot:xlabel='Prod_nucl', ylabel='Density'>
```



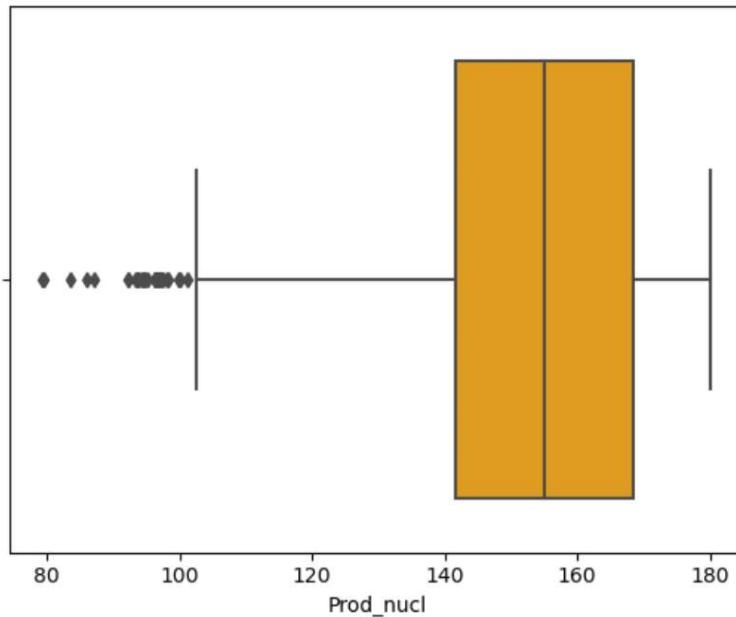
```
In [350... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_nucl'], color = "orange")
```

```
Out[350]: <AxesSubplot:xlabel='Prod_nucl'>
```



Outliers en variable "Prod_pet"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [346... # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_nucl.describe())
count    2793.000000
mean     151.392249
std      20.124441
min      79.415936
25%     141.653469
50%     155.127380
75%     168.490951
max     180.165816
Name: Prod_nucl, dtype: float64

In [347... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_nucl'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_nucl'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_nucl" en "train" es '+ str(df_def_out
print ('El valor Maximo para "Prod_nucl" en "train" es '+ str(df_def_out
El valor Minimo para "Prod_nucl" en "train" es 101.3972
El valor Maximo para "Prod_nucl" en "train" es 208.7472

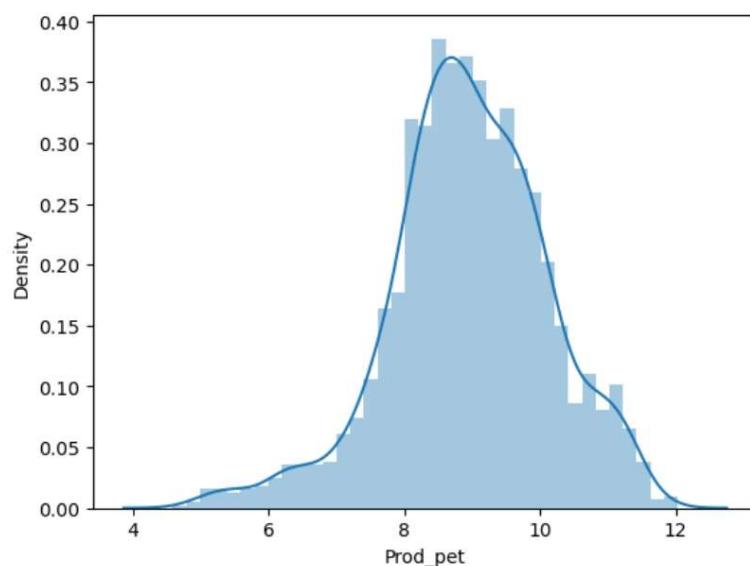
In [348... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_nucl > df_def_out_UR) | (df_
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out
Las nuevas dimensiones de los datos "train" son (2755, 28)

In [349... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_nucl'])
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [351... # Obtenemos la distribución de la variable "Prod_pet" en "train" y en "t
sns.distplot(df_def_out['Prod_pet'])
```

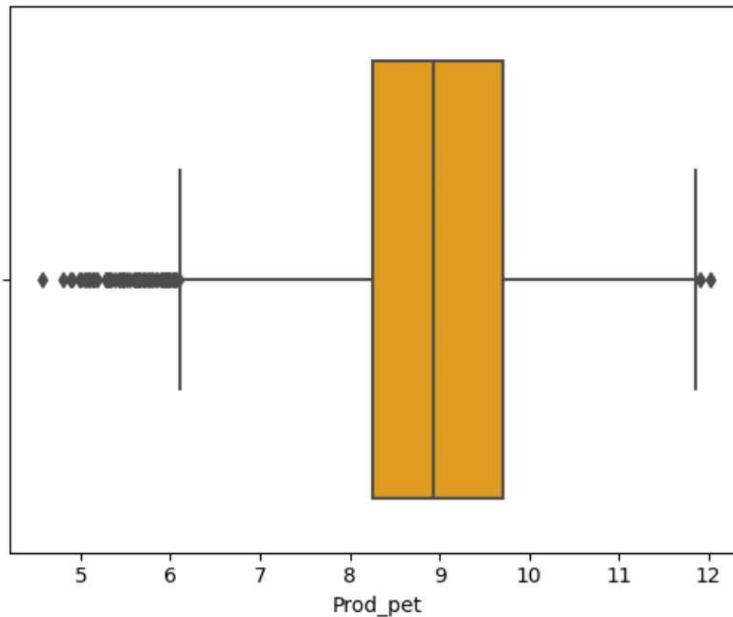
```
Out[351]: <AxesSubplot:xlabel='Prod_pet', ylabel='Density'>
```



```
In [352... # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prod_pet'], color = "orange")
```

```
Out[352]: <AxesSubplot:xlabel='Prod_pet'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [353... # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_pet.describe())
count    2755.000000
mean      8.949927
std       1.173196
min       4.579061
25%      8.263336
50%      8.936354
75%      9.710949
max      12.026109
Name: Prod_pet, dtype: float64

In [354... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_pet'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_pet'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_pet" en "train" es '+ str(df_def_out_
print ('El valor Maximo para "Prod_pet" en "train" es '+ str(df_def_out_
El valor Minimo para "Prod_pet" en "train" es 6.0919
El valor Maximo para "Prod_pet" en "train" es 11.8824

In [355... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_pet > df_def_out_UR) | (df_def_out_Prod_pet < df_def_out_LR)], axis=0, inplace=True)

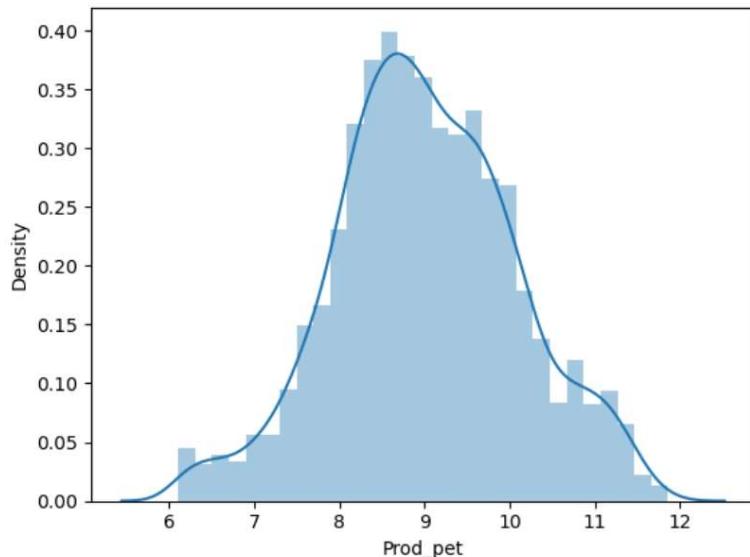
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2700, 28)

In [356... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_pet'])
```

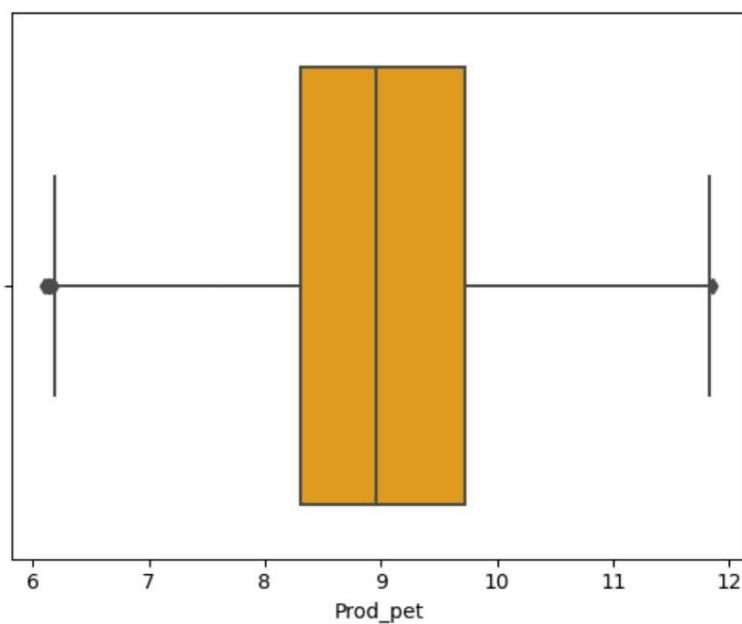
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[356]: <AxesSubplot:xlabel='Prod_pet', ylabel='Density'>
```



```
In [357... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Prod_pet'], color = "orange")
```

```
Out[357]: <AxesSubplot:xlabel='Prod_pet'>
```



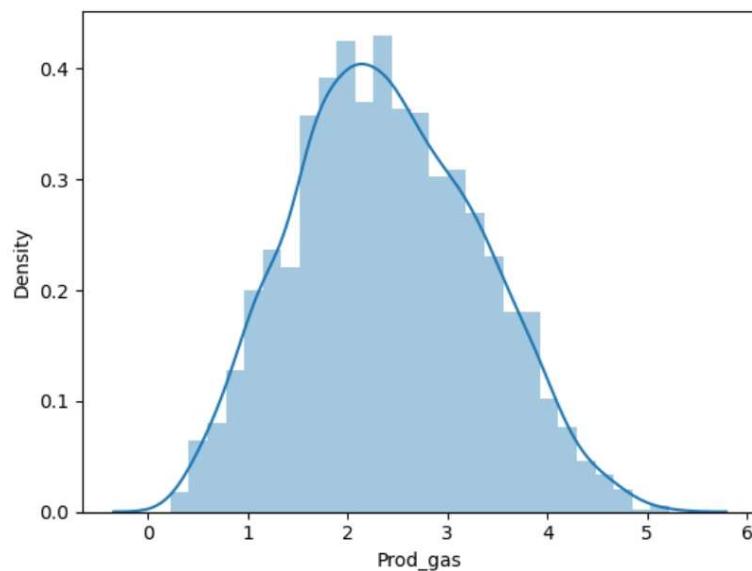
Outliers en variable "Prod_gas"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

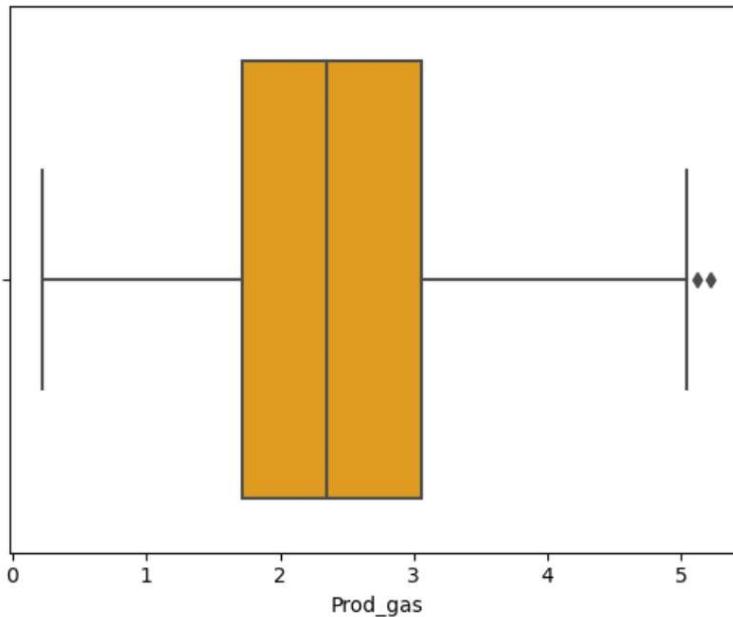
```
In [358]: # Obtenemos la distribución de la variable "Prod_gas" en "train" y en "t  
sns.distplot(df_def_out['Prod_gas'])
```

```
Out[358]: <AxesSubplot:xlabel='Prod_gas', ylabel='Density'>
```



```
In [359]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_gas'], color = "orange")
```

```
Out[359]: <AxesSubplot:xlabel='Prod_gas'>
```



```
In [360... # Obtenemos las estadísticas de los datos
print(df_def_out['Prod_gas'].describe())
count    2700.000000
mean      2.395668
std       0.922399
min       0.225596
25%      1.716987
50%      2.342792
75%      3.051452
max      5.220280
Name: Prod_gas, dtype: float64

In [361... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_gas'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_gas'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 - (1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 + (1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_gas" en "train" es ' + str(df_def_out_
print ('El valor Maximo para "Prod_gas" en "train" es ' + str(df_def_out_
El valor Minimo para "Prod_gas" en "train" es -0.2847
El valor Maximo para "Prod_gas" en "train" es 5.0532

In [362... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out['Prod_gas'] > df_def_out_UR) | (df_def_out['Prod_gas'] < df_def_out_LR)], inplace=True)

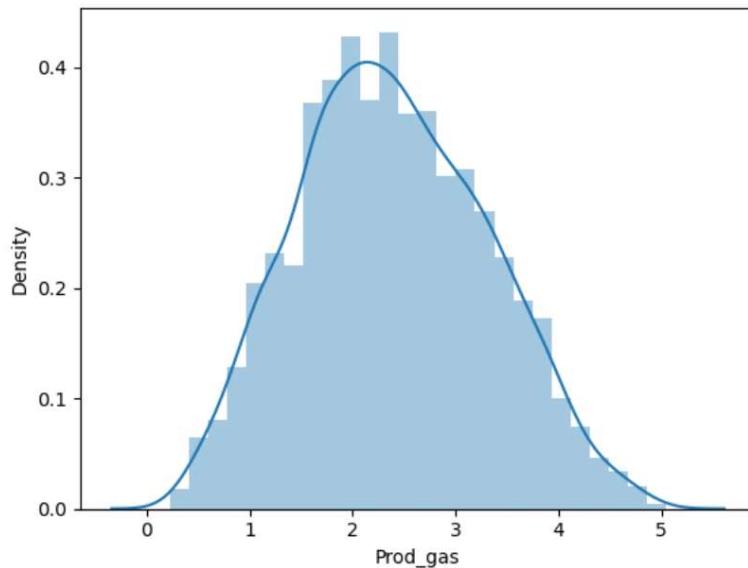
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2698, 28)

In [363... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_gas'])
```

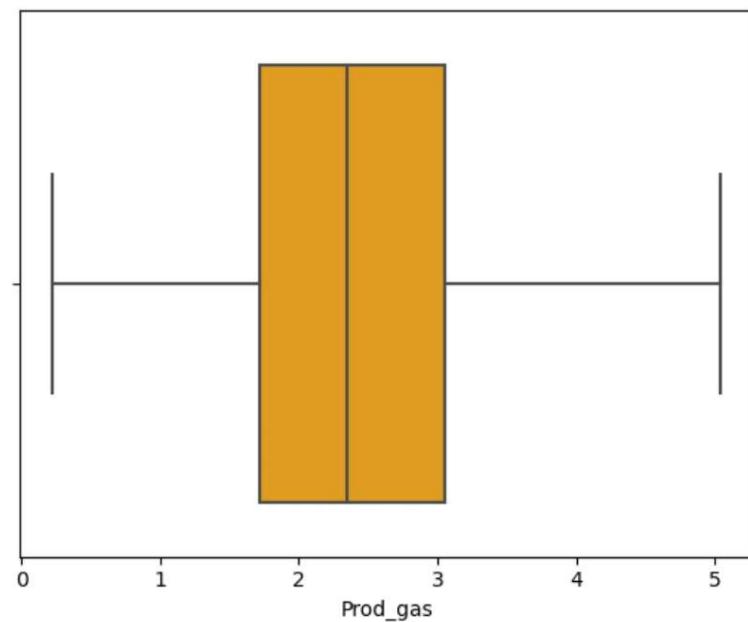
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[363]: <AxesSubplot:xlabel='Prod_gas', ylabel='Density'>
```



```
In [364... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers  
sns.boxplot(x = df_def_out['Prod_gas'], color = "orange")
```

```
Out[364]: <AxesSubplot:xlabel='Prod_gas'>
```

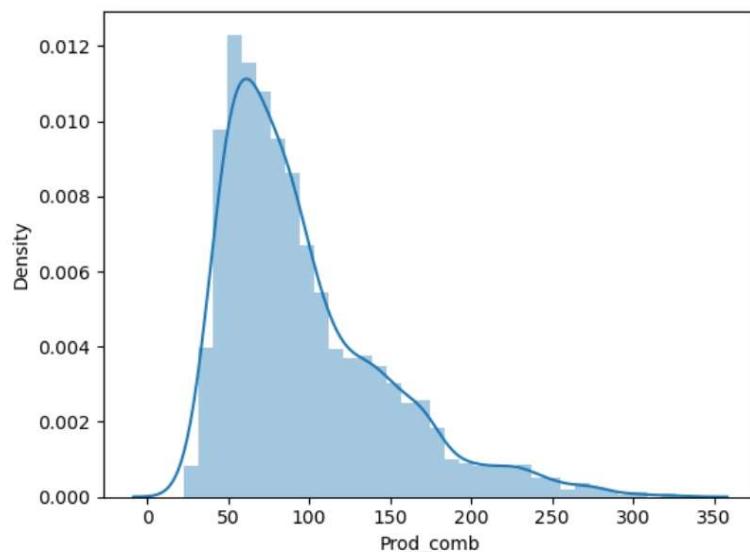


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

Outliers en variable "Prod_comb"

```
In [365]: # Obtenemos la distribución de la variable "Prod_comb" en "train" y en "test"
sns.distplot(df_def_out['Prod_comb'])

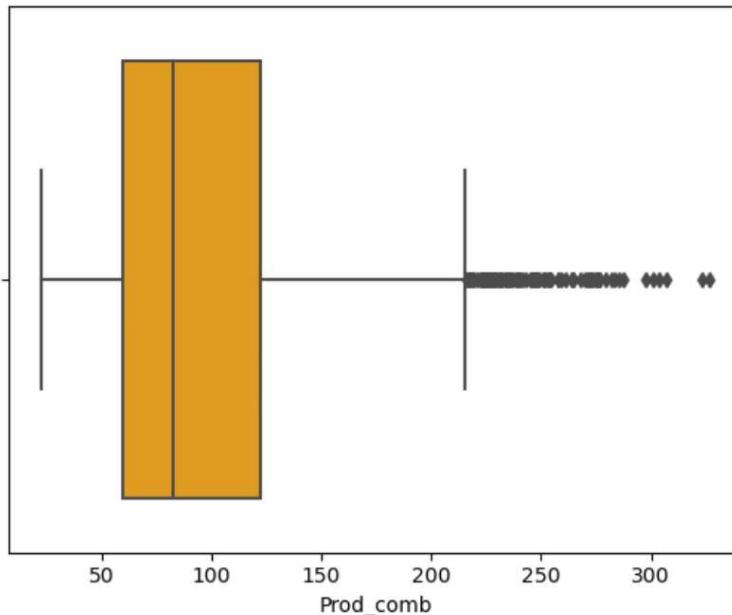
Out[365]: <AxesSubplot:xlabel='Prod_comb', ylabel='Density'>
```



```
In [366]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Prod_comb'], color = "orange")

Out[366]: <AxesSubplot:xlabel='Prod_comb'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [367...]: # Obtenemos las estadísticas de los datos
print(df_def_out['Prod_comb'].describe())
count    2698.000000
mean     96.738558
std      50.600248
min      22.726758
25%     59.487460
50%     82.624685
75%    121.954575
max     326.813456
Name: Prod_comb, dtype: float64

In [368...]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_comb'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_comb'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 - (1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 + (1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_comb" en "train" es ' + str(df_def_out_LR))
print ('El valor Maximo para "Prod_comb" en "train" es ' + str(df_def_out_UR))

El valor Minimo para "Prod_comb" en "train" es -34.2132
El valor Maximo para "Prod_comb" en "train" es 215.6552

In [369...]: # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out['Prod_comb'] > df_def_out_UR) | (df_def_out['Prod_comb'] < df_def_out_LR)], inplace=True)

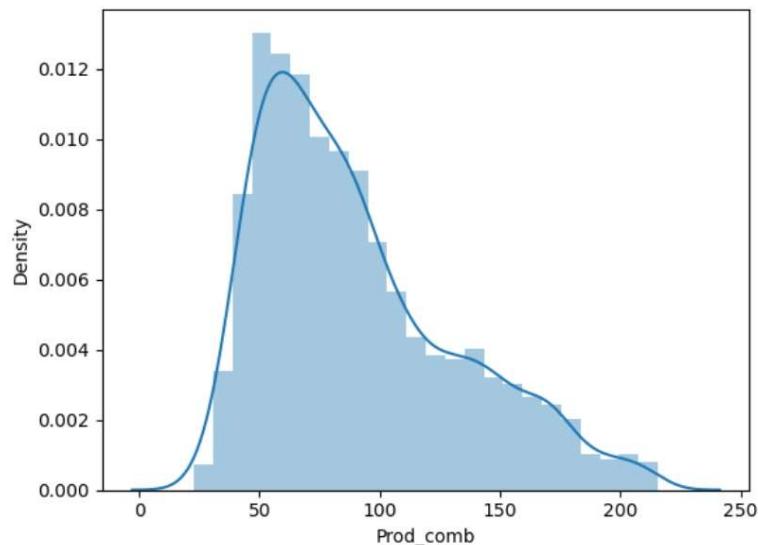
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2594, 28)

In [370...]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_comb'])
```

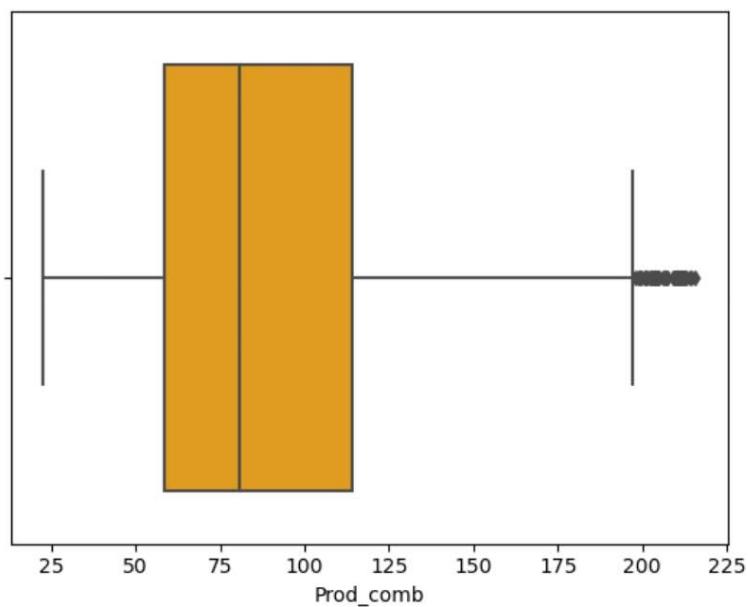
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[370]: <AxesSubplot:xlabel='Prod_comb', ylabel='Density'>
```



```
In [371]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Prod_comb'], color = "orange")
```

```
Out[371]: <AxesSubplot:xlabel='Prod_comb'>
```



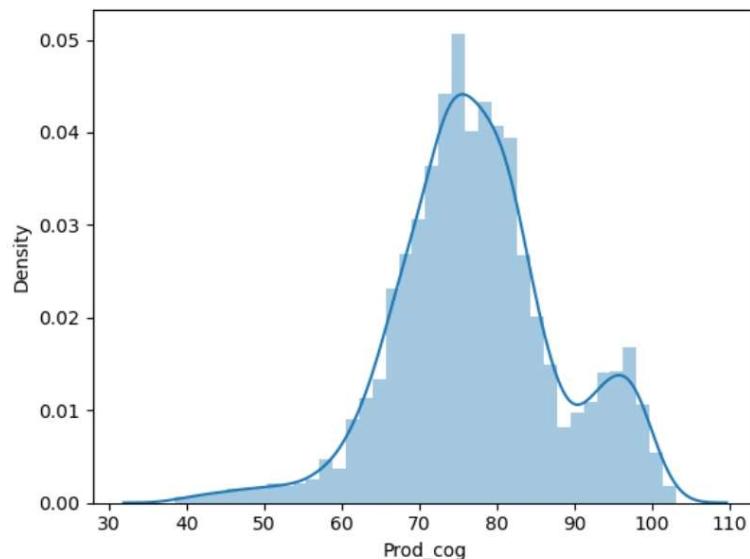
Outliers en variable "Prod_cog"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [372... # Obtenemos la distribución de la variable "Prod_cog" en "train" y en "t  
sns.distplot(df_def_out['Prod_cog'])
```

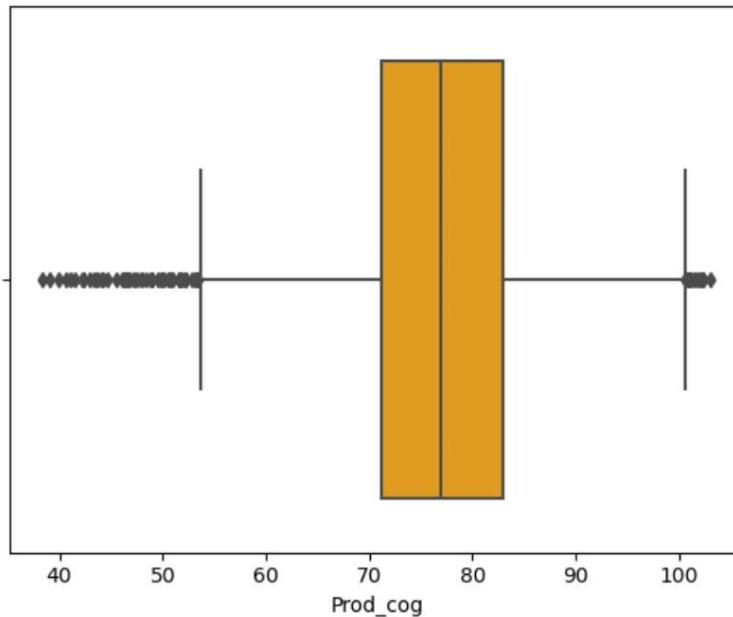
```
Out[372]: <AxesSubplot:xlabel='Prod_cog', ylabel='Density'>
```



```
In [373... # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_cog'], color = "orange")
```

```
Out[373]: <AxesSubplot:xlabel='Prod_cog'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [374... # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_cog.describe())
count    2594.000000
mean     77.447759
std      10.519073
min      38.390754
25%     71.238710
50%     76.908775
75%     83.002659
max     103.089470
Name: Prod_cog, dtype: float64

In [375... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_cog'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_cog'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_cog" en "train" es '+ str(df_def_out_
print ('El valor Maximo para "Prod_cog" en "train" es '+ str(df_def_out_
El valor Minimo para "Prod_cog" en "train" es 53.5928
El valor Maximo para "Prod_cog" en "train" es 100.6486

In [376... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Prod_cog > df_def_out_UR) | (df_def_out_Prod_cog < df_def_out_LR)], inplace=True)

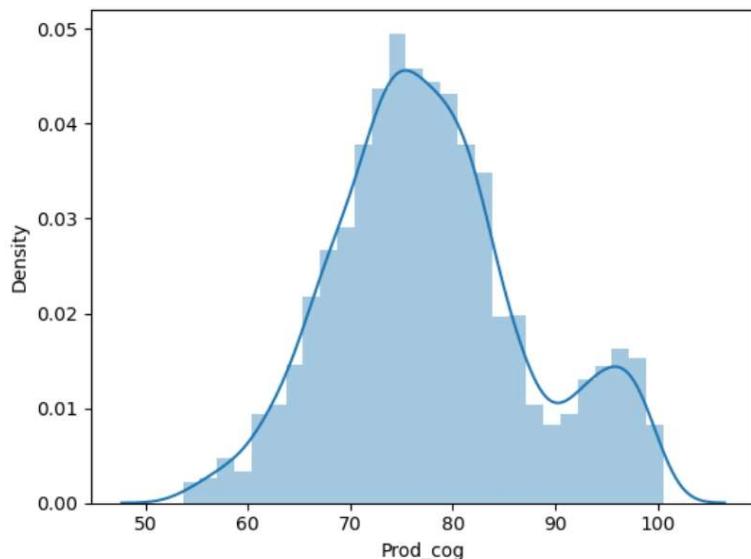
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2527, 28)

In [377... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_cog'])
```

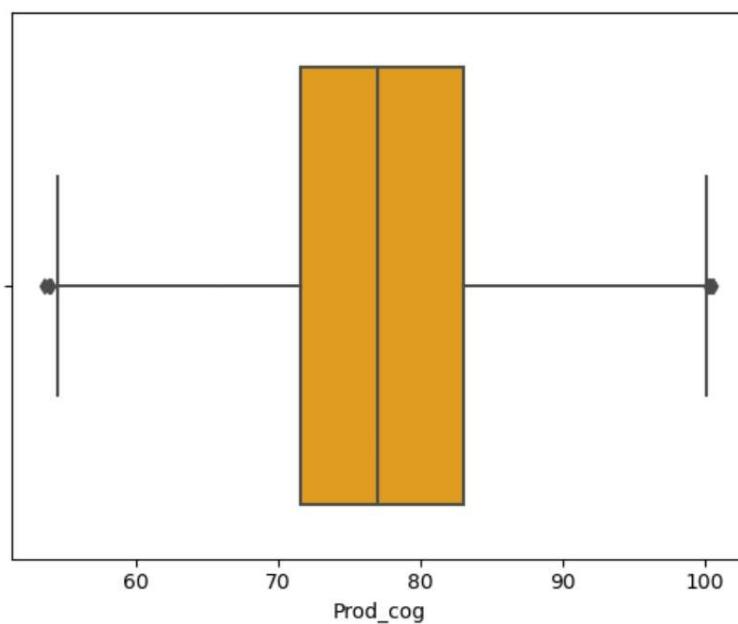
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[377]: <AxesSubplot:xlabel='Prod_cog', ylabel='Density'>
```



```
In [378... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Prod_cog'], color = "orange")
```

```
Out[378]: <AxesSubplot:xlabel='Prod_cog'>
```

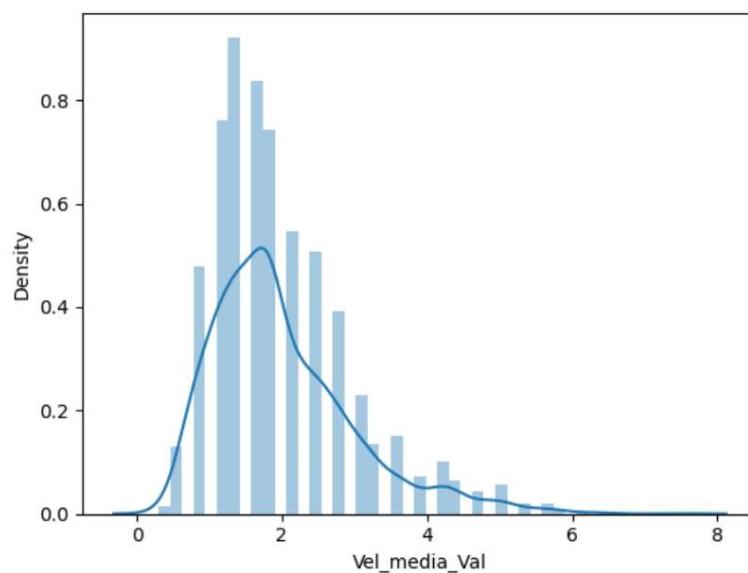


Outliers en variable "Vel_media_Val"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [379]: # Obtenemos la distribución de la variable "Vel_media_Val" en "train" y
sns.distplot(df_def_out['Vel_media_Val'])
```

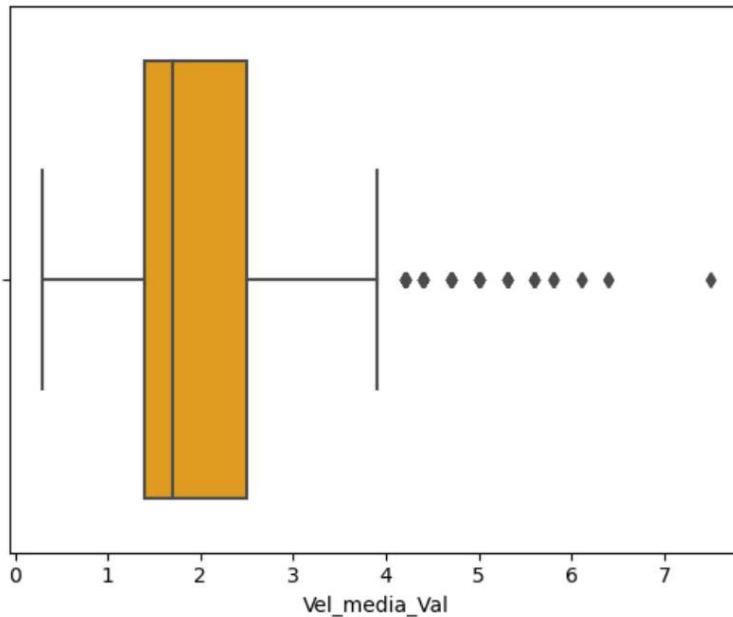
```
Out[379]: <AxesSubplot:xlabel='Vel_media_Val', ylabel='Density'>
```



```
In [380]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Vel_media_Val'], color = "orange")
```

```
Out[380]: <AxesSubplot:xlabel='Vel_media_Val'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [381... # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Val.describe())
count    2527.000000
mean      2.005540
std       0.991141
min       0.300000
25%      1.400000
50%      1.700000
75%      2.500000
max      7.500000
Name: Vel_media_Val, dtype: float64

In [382... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Val'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Val'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Val" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Vel_media_Val" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Vel_media_Val" en "train" es -0.25
El valor Maximo para "Vel_media_Val" en "train" es 4.15

In [383... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Vel_media_Val > df_def_out_UR) | (df_def_out.Vel_media_Val < df_def_out_LR)], axis=0, inplace=True)

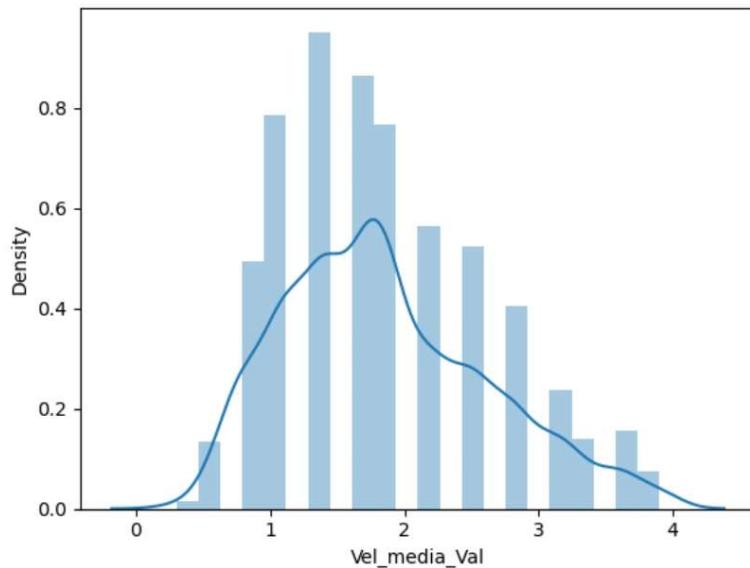
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2394, 28)

In [384... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Val'])
```

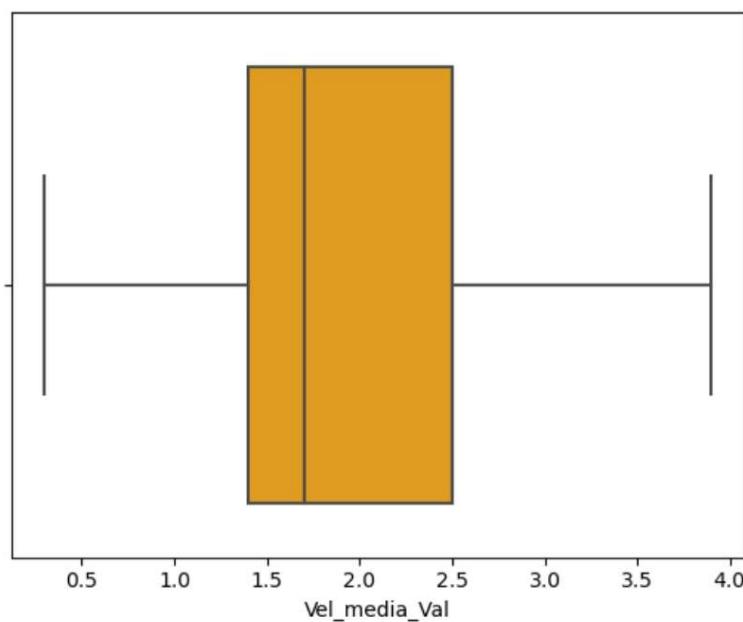
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[384]: <AxesSubplot:xlabel='Vel_media_Val', ylabel='Density'>
```



```
In [385... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers  
sns.boxplot(x = df_def_out['Vel_media_Val'], color = "orange")
```

```
Out[385]: <AxesSubplot:xlabel='Vel_media_Val'>
```



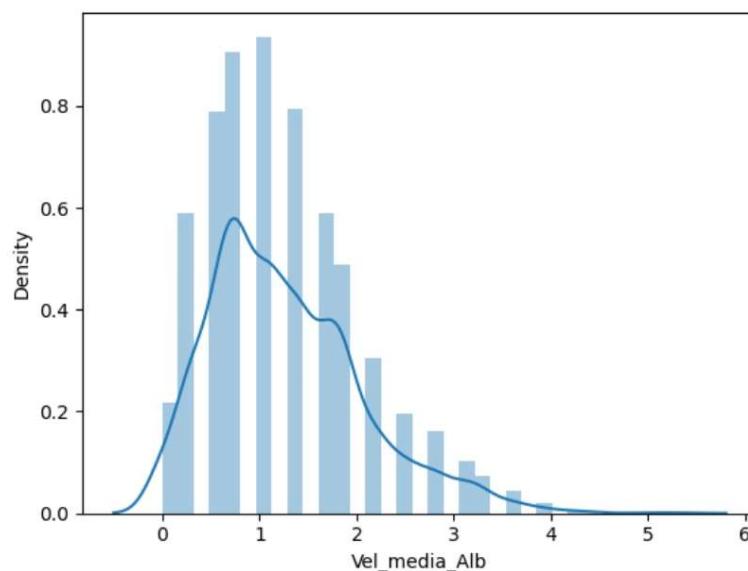
Outliers en variable "Vel_media_Alb"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [386]: # Obtenemos la distribución de la variable "Vel_media_Alb" en "train" y  
sns.distplot(df_def_out['Vel_media_Alb'])
```

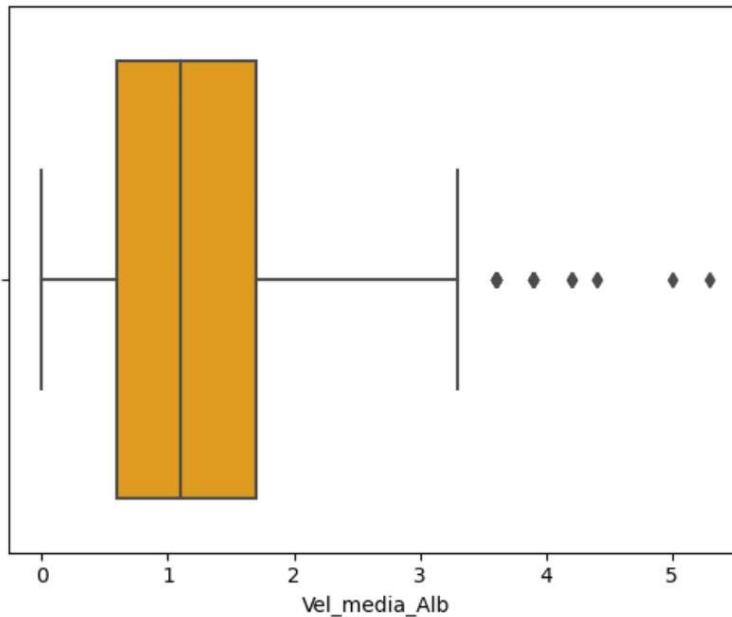
```
Out[386]: <AxesSubplot:xlabel='Vel_media_Alb', ylabel='Density'>
```



```
In [387]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Vel_media_Alb'], color = "orange")
```

```
Out[387]: <AxesSubplot:xlabel='Vel_media_Alb'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [388... # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Alb.describe())
count    2394.000000
mean     1.274144
std      0.797344
min      0.000000
25%     0.600000
50%     1.100000
75%     1.700000
max      5.300000
Name: Vel_media_Alb, dtype: float64

In [389... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Alb'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Alb'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Alb" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Vel_media_Alb" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Vel_media_Alb" en "train" es -1.05
El valor Maximo para "Vel_media_Alb" en "train" es 3.35

In [390... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Vel_media_Alb > df_def_out_UR) | (df_def_out.Vel_media_Alb < df_def_out_LR)], axis=0, inplace=True)

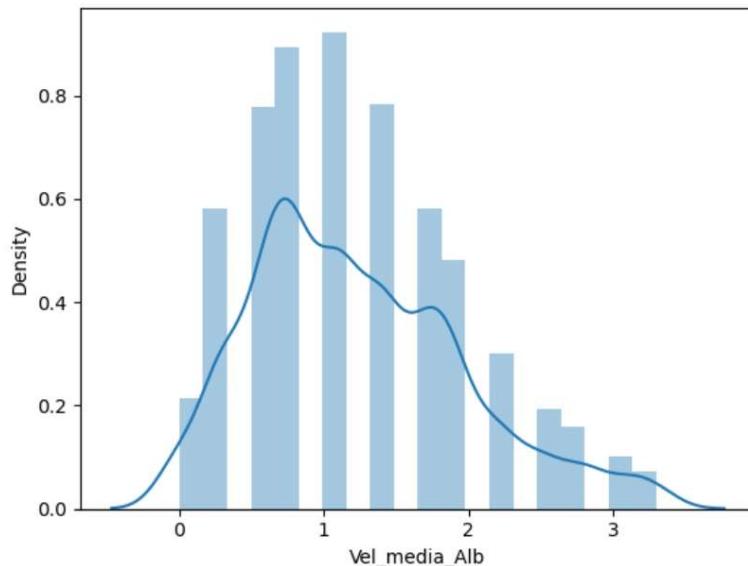
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2362, 28)

In [391... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Alb'])
```

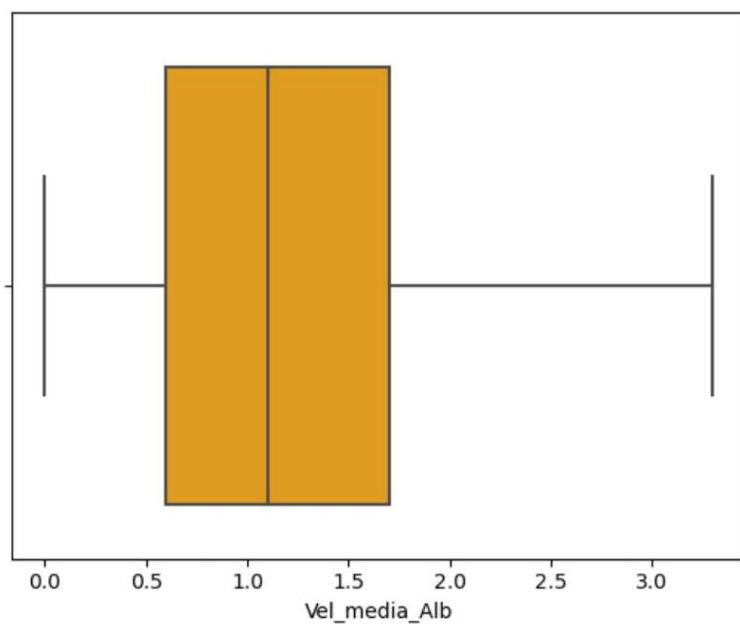
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[391]: <AxesSubplot:xlabel='Vel_media_Alb', ylabel='Density'>
```



```
In [392... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers
sns.boxplot(x = df_def_out['Vel_media_Alb'], color = "orange")
```

```
Out[392]: <AxesSubplot:xlabel='Vel_media_Alb'>
```

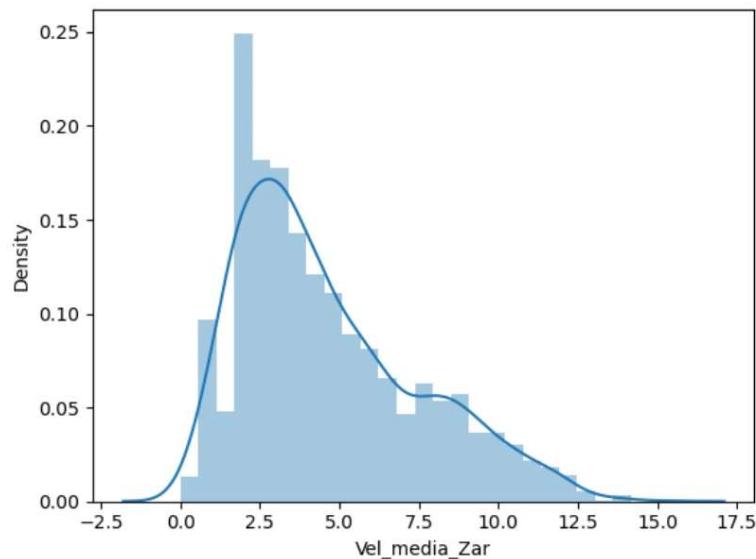


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

Outliers en variable "Vel_media_Zar"

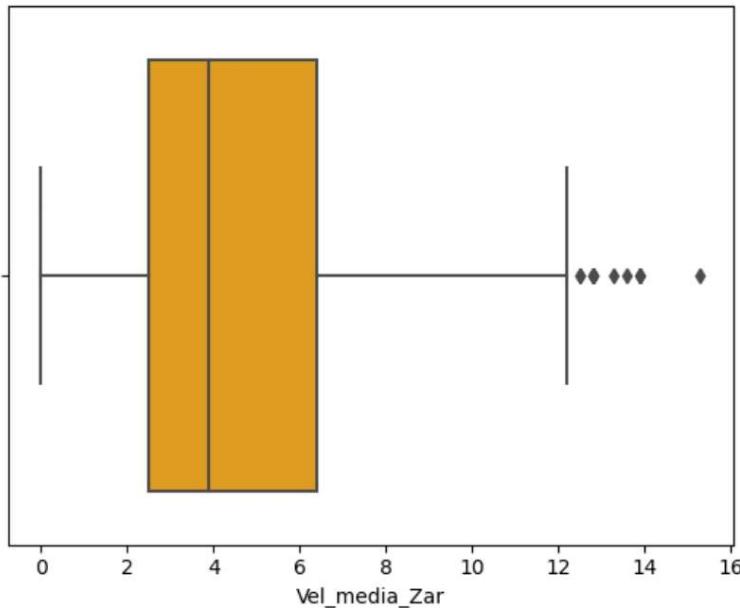
```
In [393]: # Obtenemos la distribución de la variable "Vel_media_Zar" en "train" y
sns.distplot(df_def_out['Vel_media_Zar'])

Out[393]: <AxesSubplot:xlabel='Vel_media_Zar', ylabel='Density'>
```



```
In [394]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Vel_media_Zar'], color = "orange")

Out[394]: <AxesSubplot:xlabel='Vel_media_Zar'>
```



```
In [395...]: # Obtenemos Las estadísticas de Los datos
print(df_def_out.Vel_media_Zar.describe())
count      2362.000000
mean       4.640432
std        2.831273
min        0.000000
25%       2.500000
50%       3.900000
75%       6.400000
max      15.300000
Name: Vel_media_Zar, dtype: float64

In [396...]: # Realizamos el filtrado intercuartílico en Los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Zar'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Zar'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Zar" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Vel_media_Zar" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Vel_media_Zar" en "train" es -3.35
El valor Maximo para "Vel_media_Zar" en "train" es 12.25

In [397...]: # Eliminamos Los Outliers en Los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Vel_media_Zar > df_def_out_UR) | (df_def_out.Vel_media_Zar < df_def_out_LR)], axis=0, inplace=True)

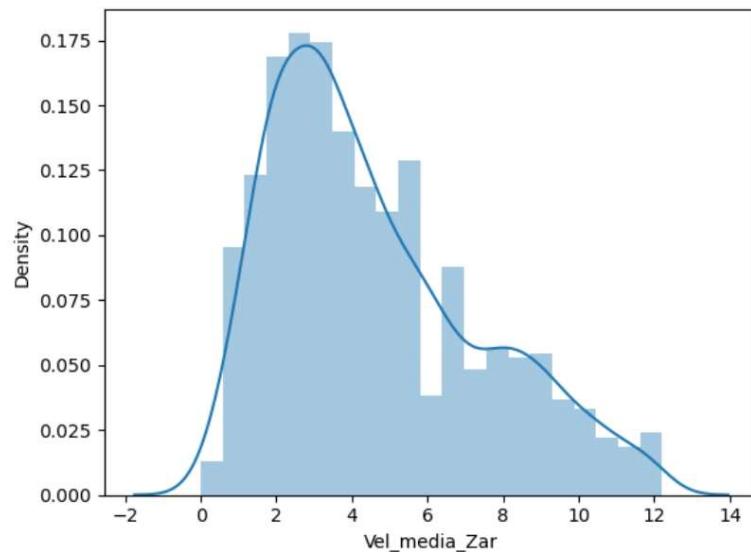
# Obtenemos Las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2349, 28)

In [398...]: # Volvemos a graficar La variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Zar'])
```

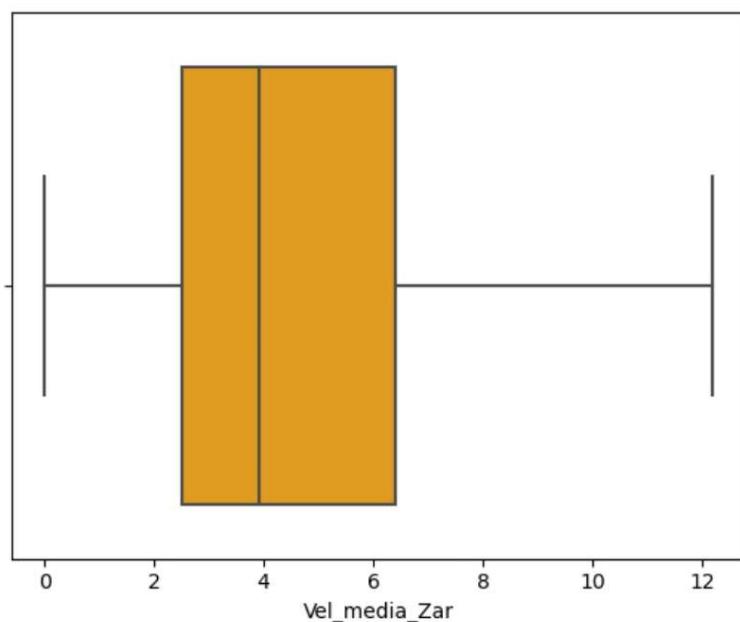
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[398]: <AxesSubplot:xlabel='Vel_media_Zar', ylabel='Density'>
```



```
In [399... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Zar'], color = "orange")
```

```
Out[399]: <AxesSubplot:xlabel='Vel_media_Zar'>
```



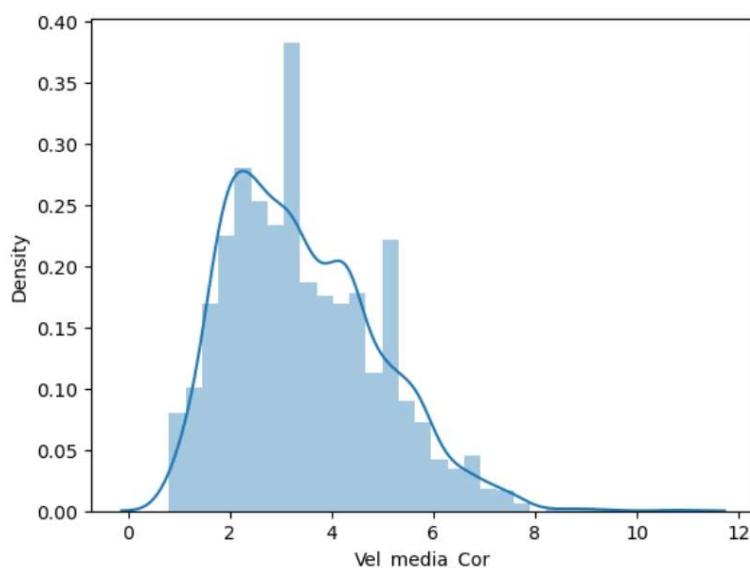
Outliers en variable "Vel_media_Cor"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [400]: # Obtenemos la distribución de la variable "Vel_media_Cor" en "train" y  
sns.distplot(df_def_out['Vel_media_Cor'])
```

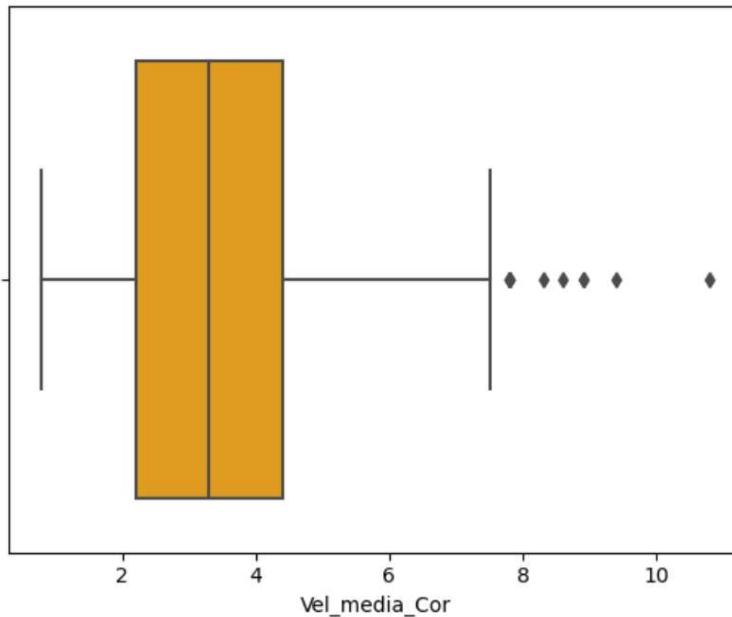
```
Out[400]: <AxesSubplot:xlabel='Vel_media_Cor', ylabel='Density'>
```



```
In [401]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Vel_media_Cor'], color = "orange")
```

```
Out[401]: <AxesSubplot:xlabel='Vel_media_Cor'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [402... # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Cor.describe())
count    2349.000000
mean      3.440996
std       1.469222
min       0.800000
25%      2.200000
50%      3.300000
75%      4.400000
max      10.800000
Name: Vel_media_Cor, dtype: float64

In [403... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Cor'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Cor'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Cor" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Vel_media_Cor" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Vel_media_Cor" en "train" es -1.1
El valor Maximo para "Vel_media_Cor" en "train" es 7.7

In [404... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Vel_media_Cor > df_def_out_UR) | (df_def_out.Vel_media_Cor < df_def_out_LR)], axis=0, inplace=True)

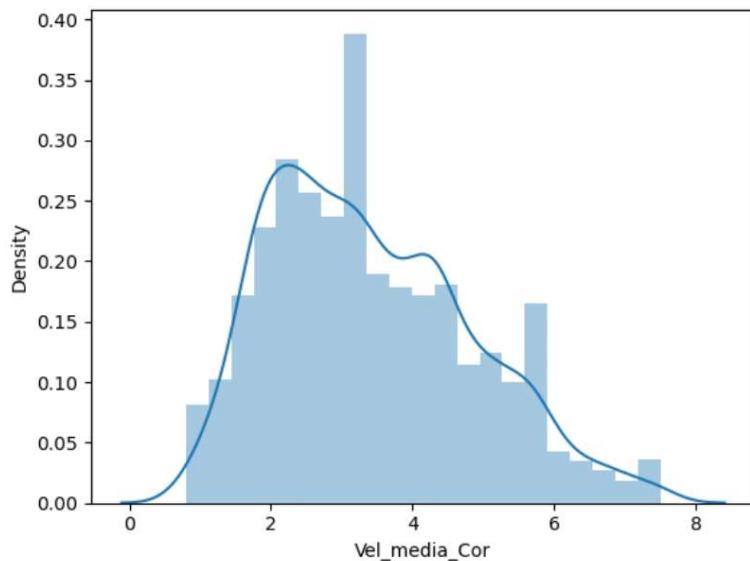
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2338, 28)

In [405... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Cor'])
```

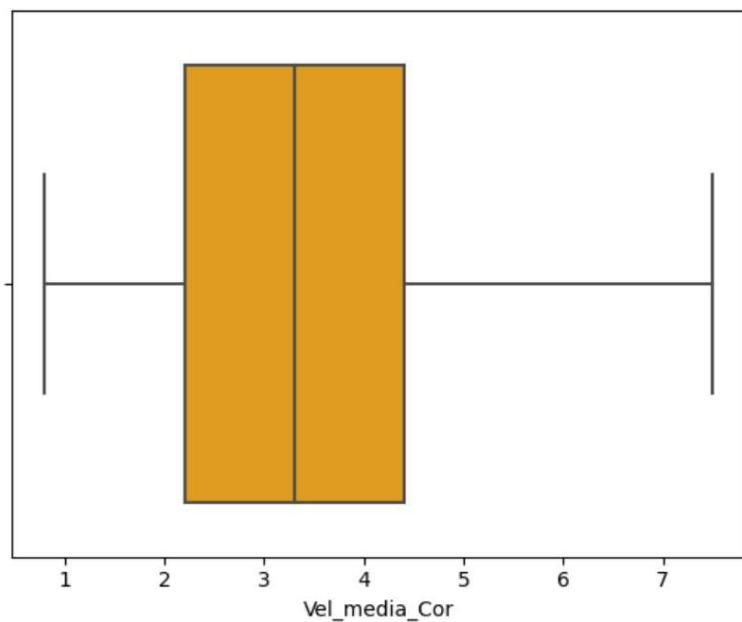
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[405]: <AxesSubplot:xlabel='Vel_media_Cor', ylabel='Density'>
```



```
In [406... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Vel_media_Cor'], color = "orange")
```

```
Out[406]: <AxesSubplot:xlabel='Vel_media_Cor'>
```

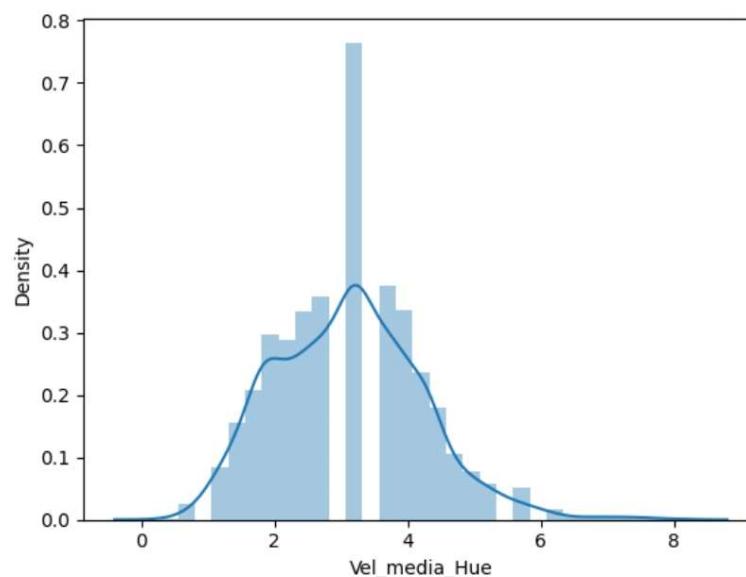


Outliers en variable "Vel_media_Hue"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [407... # Obtenemos la distribución de la variable "XXXXXX" en "train" y en "tes
sns.distplot(df_def_out['Vel_media_Hue'])
```

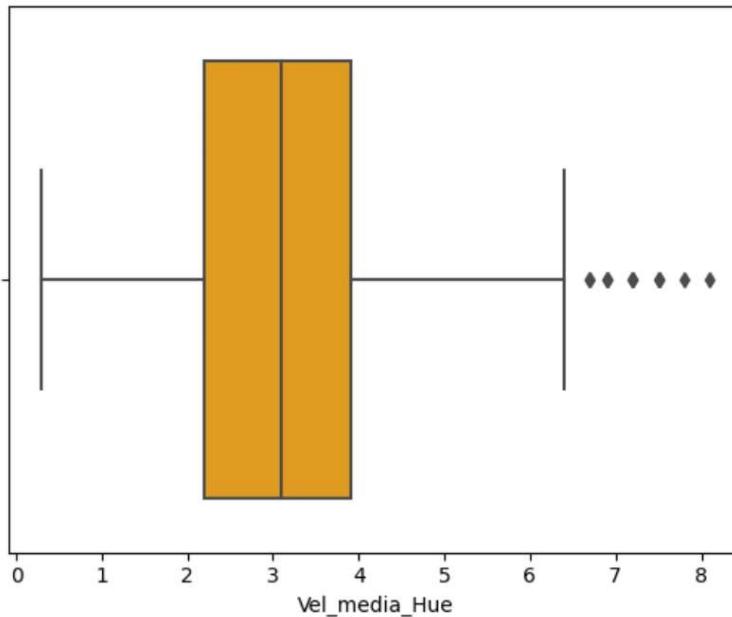
```
Out[407]: <AxesSubplot:xlabel='Vel_media_Hue', ylabel='Density'>
```



```
In [408... # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Vel_media_Hue'], color = "orange")
```

```
Out[408]: <AxesSubplot:xlabel='Vel_media_Hue'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [409... # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Hue.describe())
count    2338.000000
mean      3.113944
std       1.102478
min       0.300000
25%      2.200000
50%      3.100000
75%      3.900000
max      8.100000
Name: Vel_media_Hue, dtype: float64

In [410... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Hue'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Hue'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Hue" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Vel_media_Hue" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Vel_media_Hue" en "train" es -0.35
El valor Maximo para "Vel_media_Hue" en "train" es 6.45

In [411... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Vel_media_Hue > df_def_out_UR) | 
                           (df_def_out.Vel_media_Hue < df_def_out_LR)], axis=0, inplace=True)

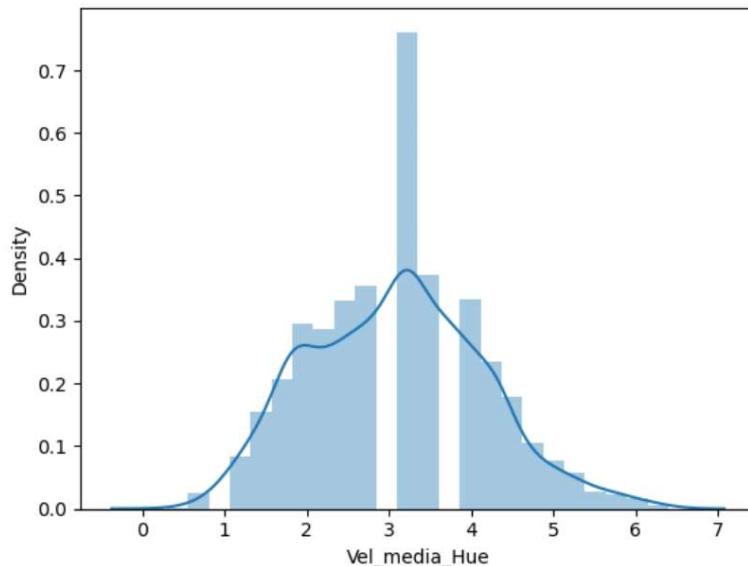
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (2325, 28)

In [412... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Hue'])
```

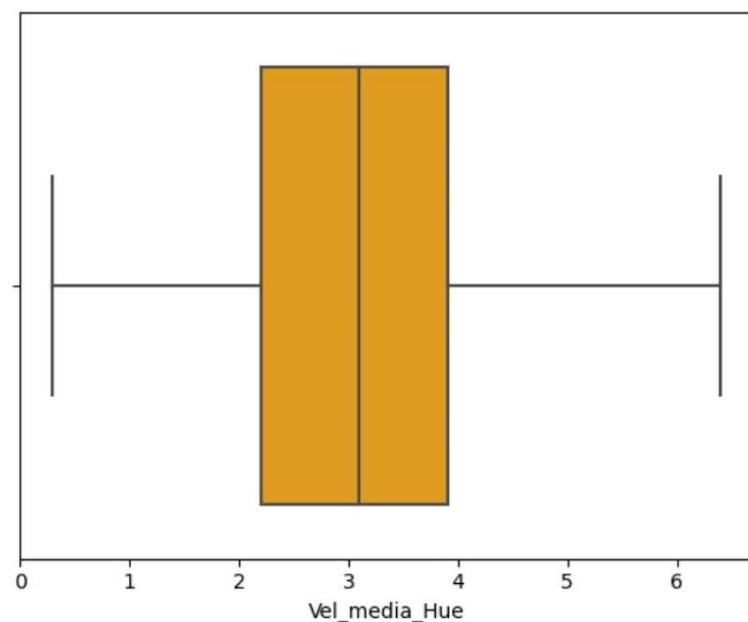
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[412]: <AxesSubplot:xlabel='Vel_media_Hue', ylabel='Density'>
```



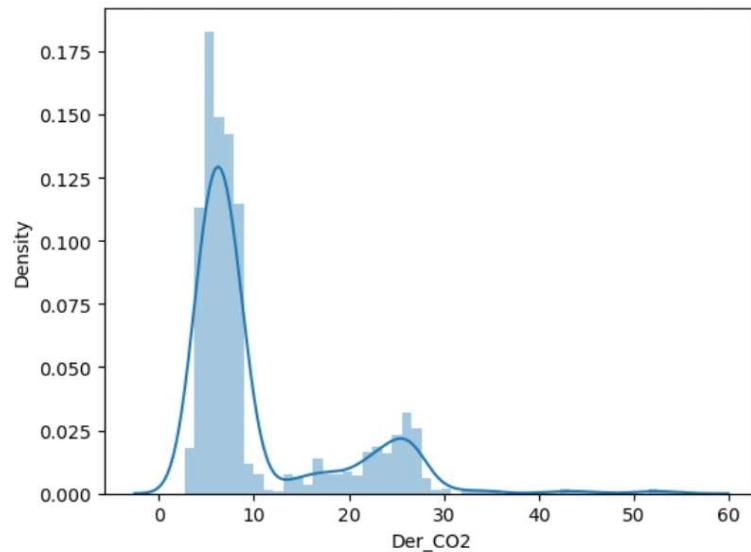
```
In [413... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers  
sns.boxplot(x = df_def_out['Vel_media_Hue'], color = "orange")
```

```
Out[413]: <AxesSubplot:xlabel='Vel_media_Hue'>
```



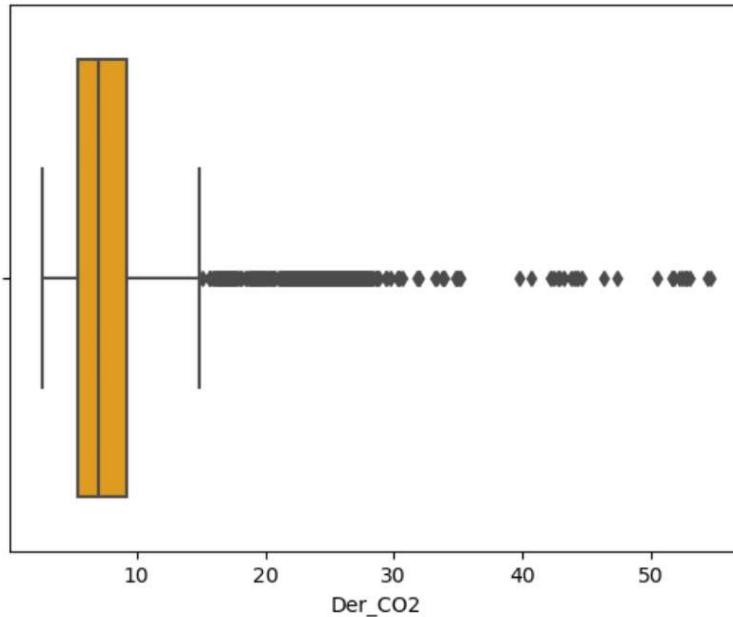
Outliers en variable "Der_CO2"

```
In [414]: # Obtenemos la distribución de la variable "Der_CO2" en "train" y en "te  
sns.distplot(df_def_out['Der_CO2'])  
  
Out[414]: <AxesSubplot:xlabel='Der_CO2', ylabel='Density'>
```



```
In [415]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Der_CO2'], color = "orange")  
  
Out[415]: <AxesSubplot:xlabel='Der_CO2'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
In [416... # Obtenemos las estadísticas de los datos
print(df_def_out.Der_CO2.describe())
count    2325.000000
mean      10.428069
std       8.320866
min       2.700000
25%      5.390000
50%      7.010000
75%      9.190000
max      54.670000
Name: Der_CO2, dtype: float64

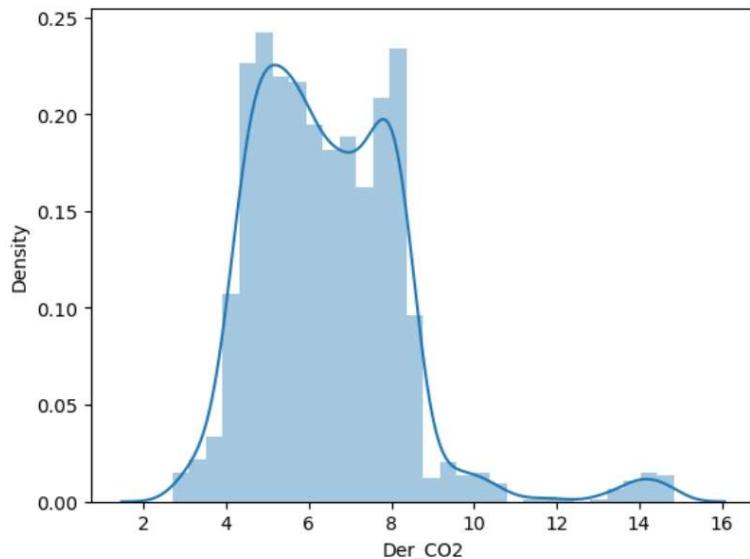
In [417... # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Der_CO2'].quantile(0.25)
df_def_out_Q3 = df_def_out['Der_CO2'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Der_CO2" en "train" es '+ str(df_def_out_L
print ('El valor Maximo para "Der_CO2" en "train" es '+ str(df_def_out_U
El valor Minimo para "Der_CO2" en "train" es -0.31
El valor Maximo para "Der_CO2" en "train" es 14.89

In [418... # Eliminamos los Outliers en los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Der_CO2 > df_def_out_UR) | (df_de
# Obtenemos las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out
Las nuevas dimensiones de los datos "train" son (1825, 28)

In [419... # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Der_CO2'])
```

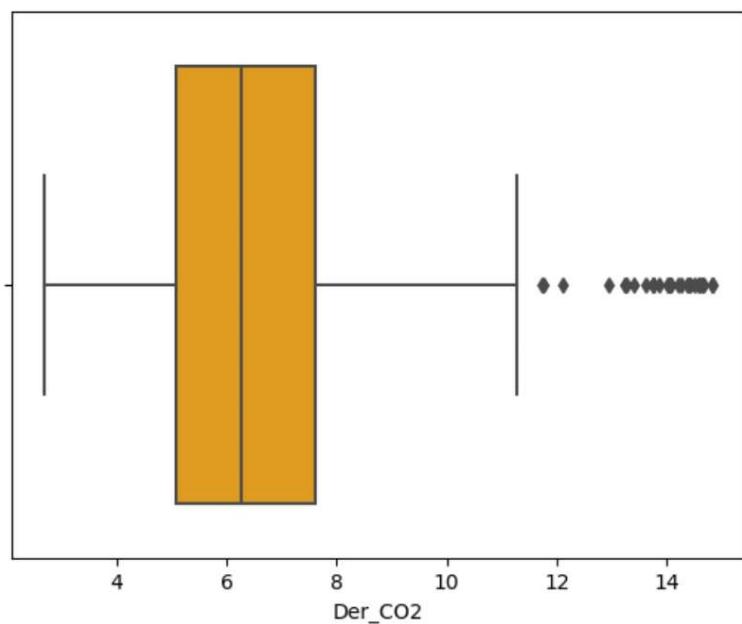
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
Out[419]: <AxesSubplot:xlabel='Der_CO2', ylabel='Density'>
```



```
In [420... # Volvemos a obtener el boxplot de "train" una vez tratados los outliers  
sns.boxplot(x = df_def_out['Der_CO2'], color = "orange")
```

```
Out[420]: <AxesSubplot:xlabel='Der_CO2'>
```

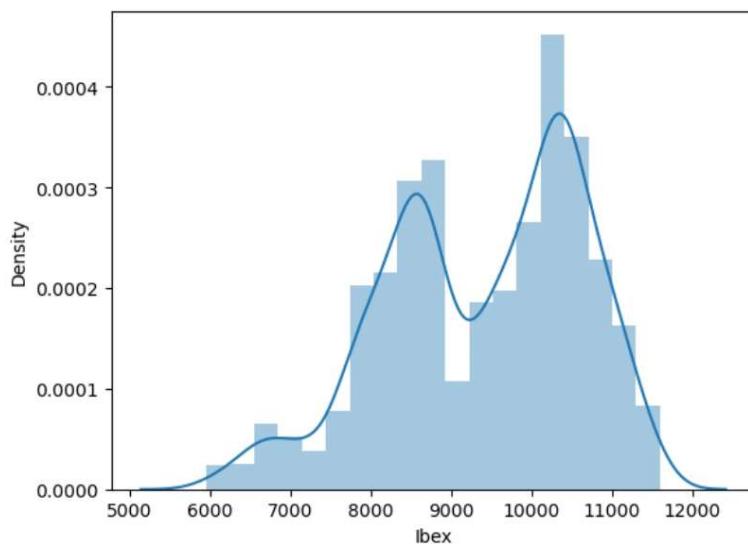


Outliers en variable "Ibex"

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

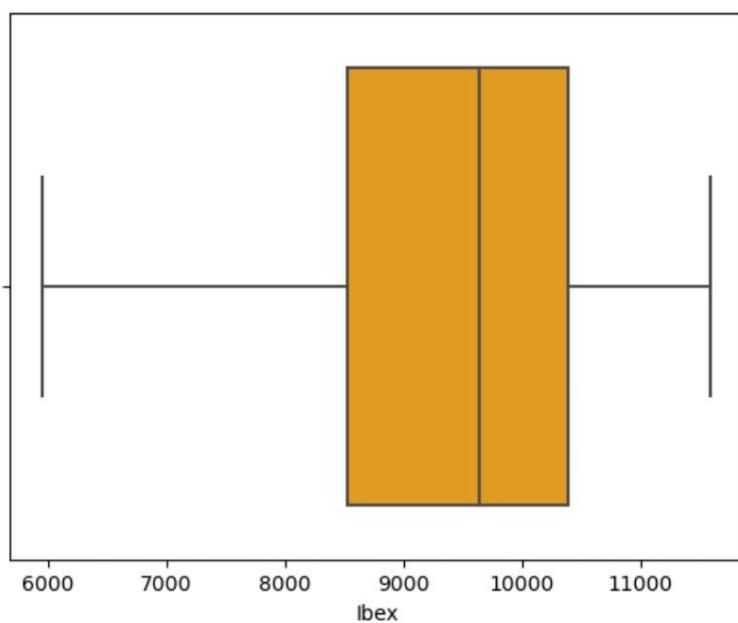
```
In [421]: # Obtenemos la distribución de la variable "Ibex" en "train" y en "test"  
sns.distplot(df_def_out['Ibex'])
```

```
Out[421]: <AxesSubplot: xlabel='Ibex', ylabel='Density'>
```



```
In [422]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Ibex'], color = "orange")
```

```
Out[422]: <AxesSubplot: xlabel='Ibex'>
```



MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [423...]: # Obtenemos Las estadísticas de Los datos
print(df_def_out.Ibex.describe())

count      1825.0000
mean       9405.31463
std        1223.89943
min        5956.30000
25%        8526.70000
50%        9639.60000
75%        10386.90000
max        11595.40000
Name: Ibex, dtype: float64

In [424...]: # Realizamos el filtrado intercuartílico en Los datos "train"
df_def_out_Q1 = df_def_out['Ibex'].quantile(0.25)
df_def_out_Q3 = df_def_out['Ibex'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Ibex" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Ibex" en "train" es '+ str(df_def_out_UR))

El valor Minimo para "Ibex" en "train" es 5736.4
El valor Maximo para "Ibex" en "train" es 13177.2

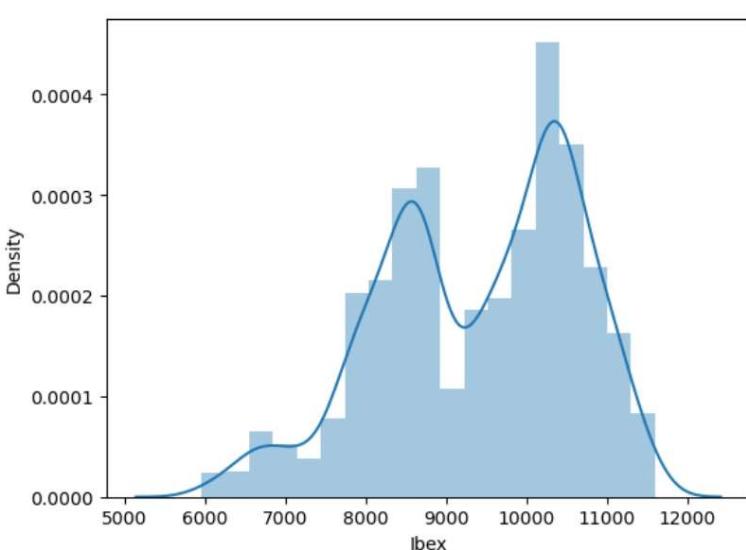
In [425...]: # Eliminamos Los Outliers en Los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Ibex > df_def_out_UR) | (df_def_out.Ibex < df_def_out_LR)], axis=0, inplace=True)

# Obtenemos Las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son '+ str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (1825, 28)

In [426...]: # Volvemos a graficar La variable para comprobar el cambio
sns.distplot(df_def_out['Ibex'])

Out[426]: <AxesSubplot:xlabel='Ibex', ylabel='Density'>
```

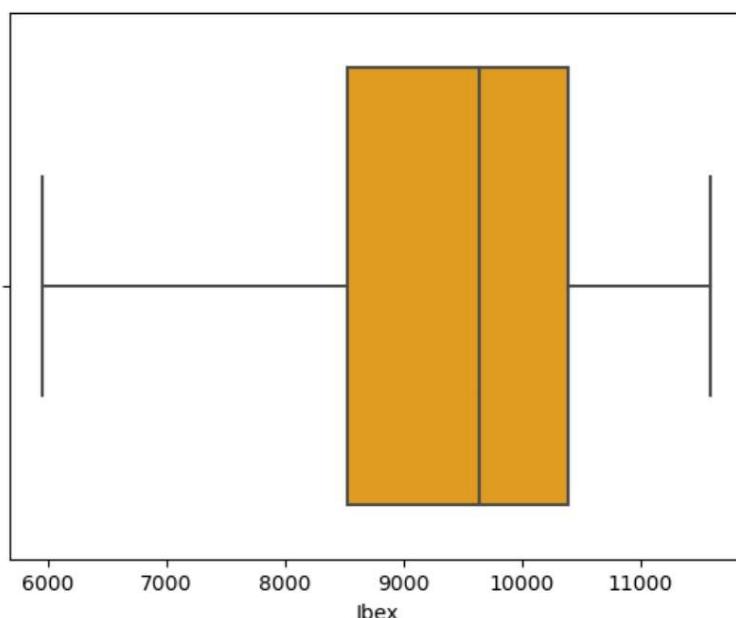


MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
In [427]: # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers  
sns.boxplot(x = df_def_out['Ibex'], color = "orange")
```

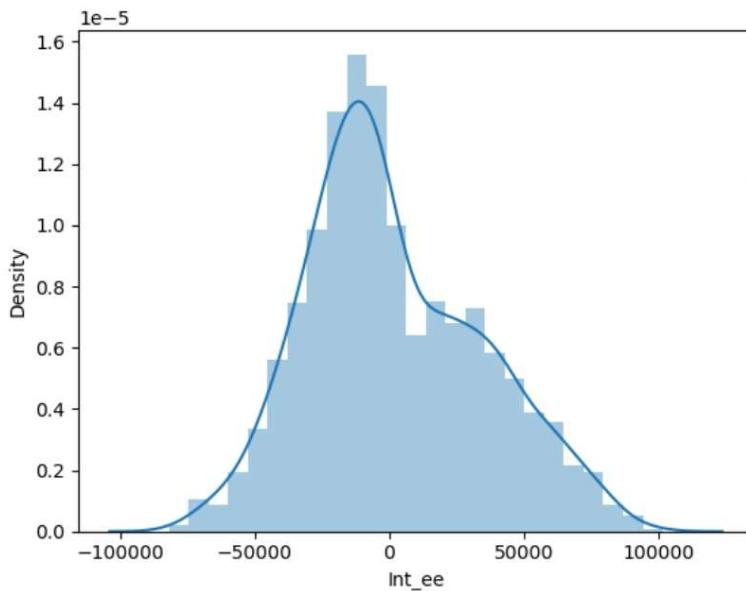
```
Out[427]: <AxesSubplot:xlabel='Ibex'>
```



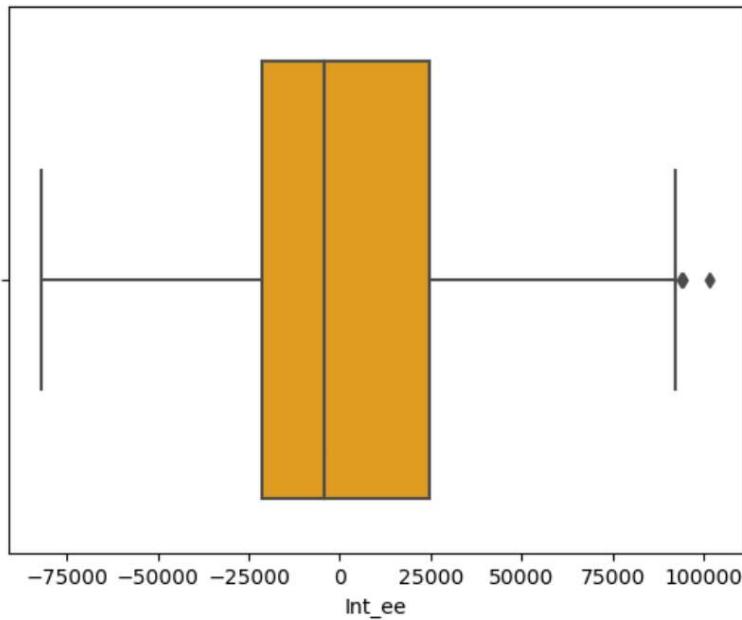
Outliers en variable "Int_ee"

```
In [428]: # Obtenemos La distribucion de La variable "Int_ee" en "train" y en "tes  
sns.distplot(df_def_out['Int_ee'])
```

```
Out[428]: <AxesSubplot:xlabel='Int_ee', ylabel='Density'>
```



```
In [429]: # Dibujamos el boxplot para Los datos "train"  
sns.boxplot(x = df_def_out['Int_ee'], color = "orange")  
Out[429]: <AxesSubplot:xlabel='Int_ee'>
```



```
In [430]: # Obtenemos Las estadísticas de Los datos  
print(df_def_out.Int_ee.describe())
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

```
count      1825.000000
mean       1847.630412
std        33108.499511
min       -81874.320000
25%      -21192.721000
50%       -4049.878000
75%      24832.163000
max      101779.882000
Name: Int_ee, dtype: float64
```

```
In [431...]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Int_ee'].quantile(0.25)
df_def_out_Q3 = df_def_out['Int_ee'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Int_ee" en "train" es ' + str(df_def_out_LR))
print ('El valor Maximo para "Int_ee" en "train" es ' + str(df_def_out_UR))

El valor Minimo para "Int_ee" en "train" es -90230.047
El valor Maximo para "Int_ee" en "train" es 93869.489

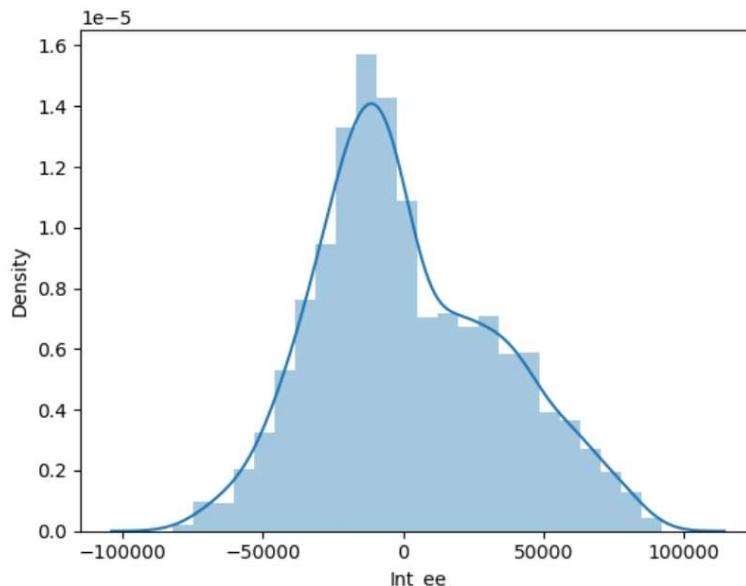
In [432...]: # Eliminamos Los Outliers en Los datos "train"
df_def_out.drop(df_def_out[(df_def_out.Int_ee > df_def_out.UR) | (df_def_out.Int_ee < df_def_out.LR)], inplace=True)

# Obtenemos Las nuevas dimensiones
print('Las nuevas dimensiones de los datos "train" son ' + str(df_def_out.shape))

Las nuevas dimensiones de los datos "train" son (1822, 28)

In [433...]: # Volvemos a graficar La variable para comprobar el cambio
sns.distplot(df_def_out['Int_ee'])

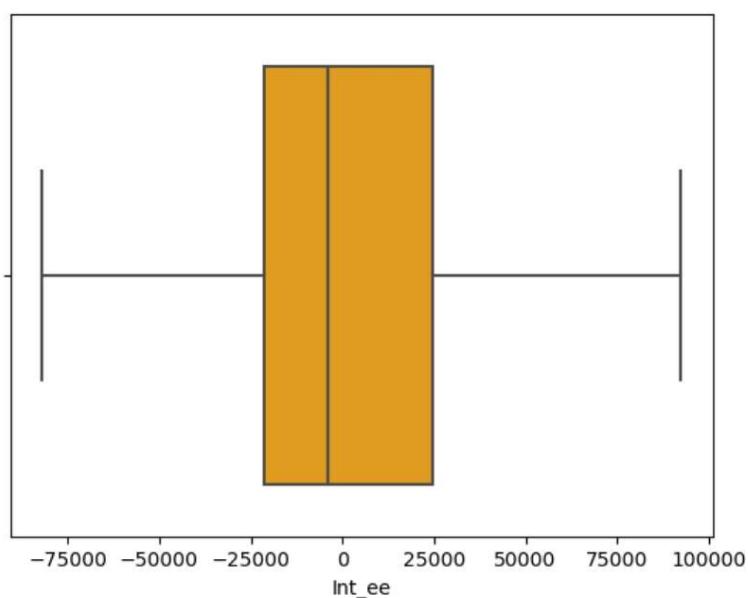
Out[433]: <AxesSubplot:xlabel='Int_ee', ylabel='Density'>
```



MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
In [434... # Volvemos a obtener el boxplot de "train" una vez tratados Los outliers
sns.boxplot(x = df_def_out['Int_ee'], color = "orange")
```

```
Out[434]: <AxesSubplot:xlabel='Int_ee'>
```



```
In [ ]:
```

Creamos un csv con los datos del dataframe obtenidos "df_def_out" para utilizarlos en los modelos

```
In [435... # Para descargarse los csv descomentar el codido
#df_def_out.to_csv('df_def_outliers.csv', header=True, index=False)
```

```
In [ ]:
```

```
In [ ]:
```

ANEXO VI: PROCESADO OUTLIERS (LIMT)

6.6.5 Procesado Outliers v07 (lim)

July 29, 2023

0.1 PROCESADO DE LOS OUTLIERS

Como el numero de variables que presentan valores atípicos es bastante elevado se ha creado este apartado para tratar este problema.

Las variables que presentan este tipo de valores son: - Dem: 5 valores (0.137%) - Prec_gas: 516 valores (14.129%) - Prec_car: 468 valores (12.815%) - Prod_eol: 33 valores (0.904%) - Prod_sol: 207 valores (5.668%) - Prod_hidr: 172 valores (4.710%) - Prod_ofr: 323 valores (8.844%) - Prod_nucl: 67 valores (1.835%) - Prod_pet: 20 valores (0.548%) - Prod_gas: 4 valores (0.110%) - Prod_comb: 160 valores (4.381%) - Prod_cog: 248 valores (6.791%) - Vel_media_Val: 197 valores (5.394%) - Vel_media_Alb: 101 valores (2.766%) - Vel_media_Zar: 22 valores (0.602%) - Vel_media_Cor: 22 valores (0.602%) - Vel_media_Hue: 76 valores (2.081%) - Der_CO2: 505 valores (13.828%) - Ibex: 5 valores (0.137%) - Int_ee: 24 valores (0.657%)

0.1.1 Cargamos las librerías necesarias

```
[1]: import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

0.1.2 Cargamos los datos

```
[2]: df_def = pd.read_csv(r'C:\Users\User\1.PYTHON\00.TFM\00.Archivos_utilizados\14.
Dataset final de datos.csv', sep=';')

[3]: print('Las dimensiones de los datos "df_def" son '+ str(df_def.shape))
```

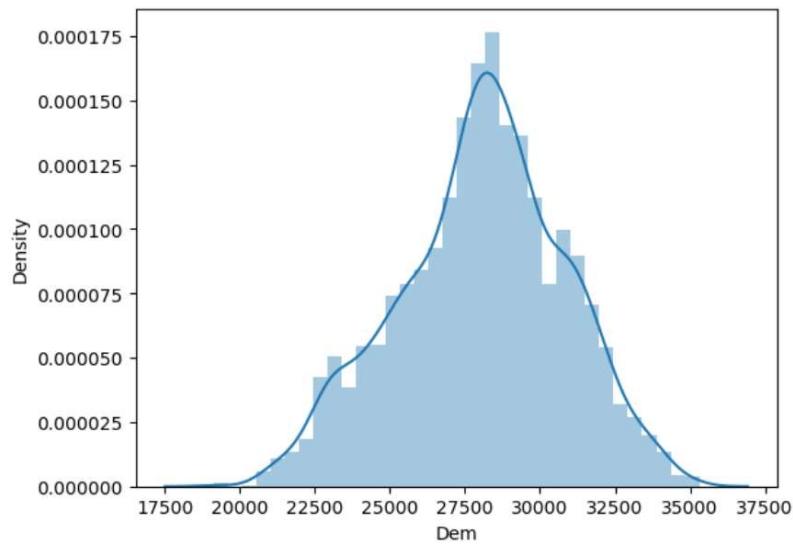
Las dimensiones de los datos "df_def" son (3652, 28)

0.1.3 Outliers en variable “Dem”

```
[4]: # Obtenemos la distribución de la variable "Dem" en "train" y en "test"
sns.distplot(df_def['Dem'])
```

```
[4]: <AxesSubplot: xlabel='Dem', ylabel='Density'>
```

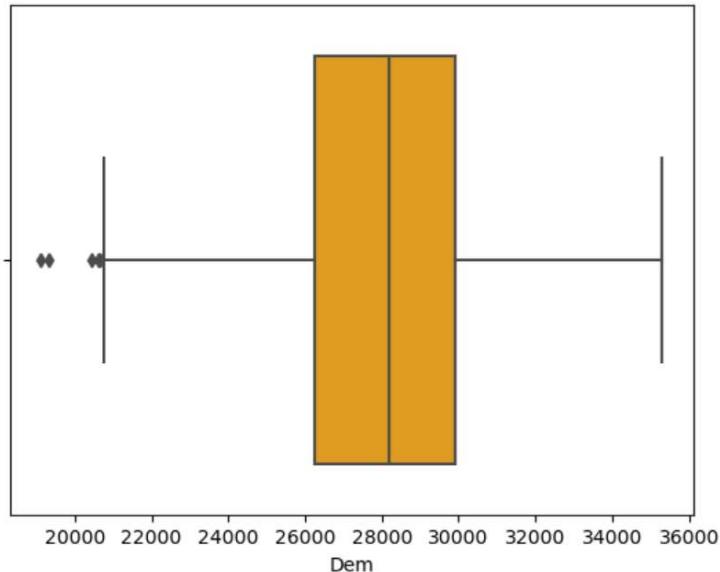
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[5]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def['Dem'], color = "orange")
```

```
[5]: <AxesSubplot:xlabel='Dem'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[6]: # Obtenemos las estadísticas de los datos
print(df_def.Dem.describe())
```

```
count    3652.000000
mean    28045.150005
std     2781.353082
min    19122.490000
25%    26232.872500
50%    28194.870000
75%    29903.192500
max    35306.410000
Name: Dem, dtype: float64
```

Para eliminar los valores extremos vamos a emplear el método de los percentiles

```
[7]: # Para obtener el Rango Intercuartilico tenemos que calcular la diferencia entre el tercer y el primer percentil.
# Luego en base a esto calcularemos los valores mínimos y máximos para definir qué observaciones
# serán descartadas.
```

```
[8]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out = df_def.copy()
df_def_out_Q1 = df_def_out['Dem'].quantile(0.25)
df_def_out_Q3 = df_def_out['Dem'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Dem" en "train" es ' + str(df_def_out_LR))
print ('El valor Maximo para "Dem" en "train" es ' + str(df_def_out_UR))
```

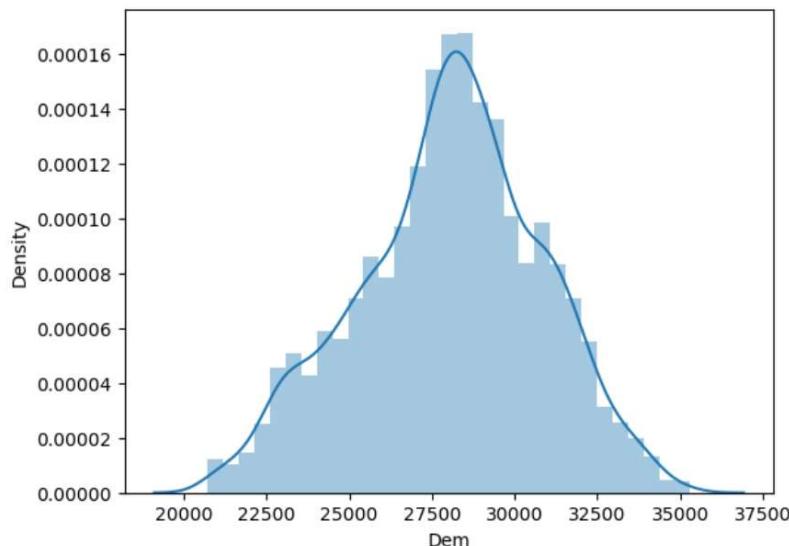
El valor Minimo para "Dem" en "train" es 20727.3925

El valor Maximo para "Dem" en "train" es 35408.6725

```
[9]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Dem'] < df_def_out_LR, 'Dem'] = df_def_out_LR
df_def_out.loc[df_def_out['Dem'] > df_def_out_UR, 'Dem'] = df_def_out_UR
```

```
[10]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Dem'])
```

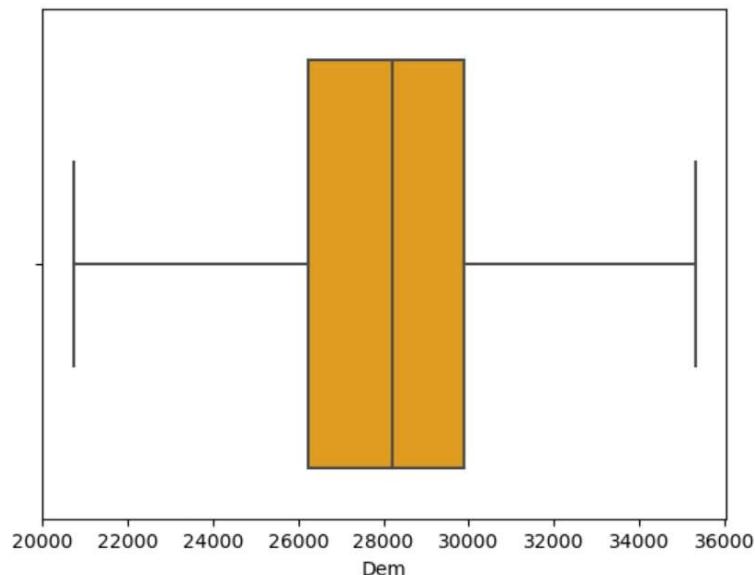
```
[10]: <AxesSubplot:xlabel='Dem', ylabel='Density'>
```



MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
[11]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Dem'], color = "orange")
```

```
[11]: <AxesSubplot:xlabel='Dem'>
```

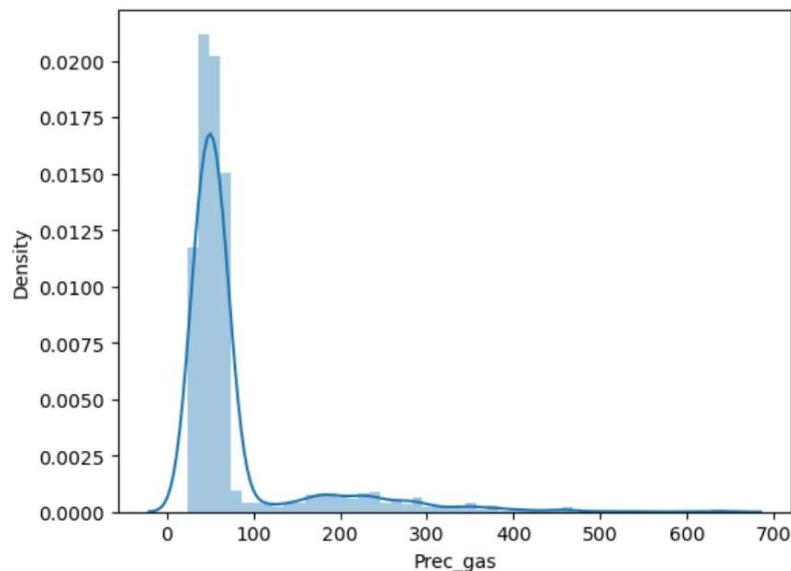


0.1.4 Outliers en variable “Prec_gas”

```
[12]: # Obtenemos la distribucion de la variable "Prec_gas" en "train" y en "test"
sns.distplot(df_def_out['Prec_gas'])
```

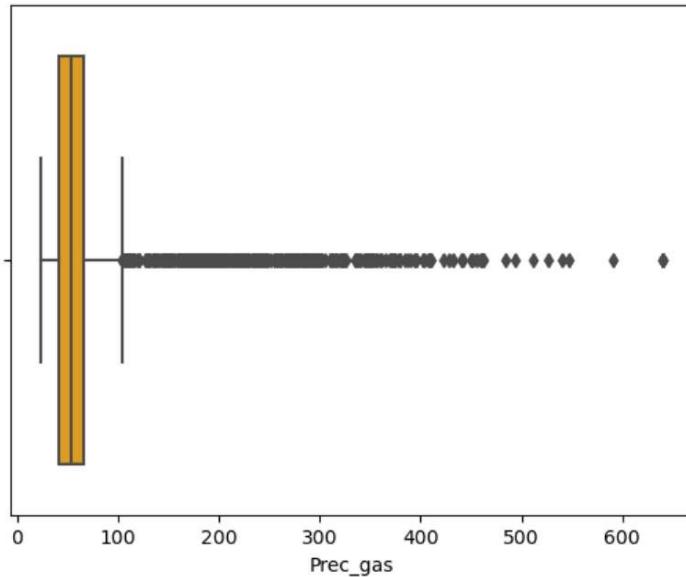
```
[12]: <AxesSubplot:xlabel='Prec_gas', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[13]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prec_gas'], color = "orange")  
  
[13]: <AxesSubplot:xlabel='Prec_gas'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[14]: # Podríamos llegar a una rápida conclusión, de que los valores de "Prec_gas" tienen demasiados valores fuera de rango, esto puede generar "ruido" en nuestro análisis, por lo que deberíamos deshacernos de esos valores y volver a graficar.  
# Para eliminar estos valores empleamos el metodo de los percentiles
```

```
[15]: # Obtenemos las estadísticas de los datos  
print(df_def_out.Prec_gas.describe())
```

```
count    3652.000000  
mean     77.796952  
std      77.067507  
min     24.180000  
25%    41.625000  
50%    53.890000  
75%    66.655000  
max    640.360000  
Name: Prec_gas, dtype: float64
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

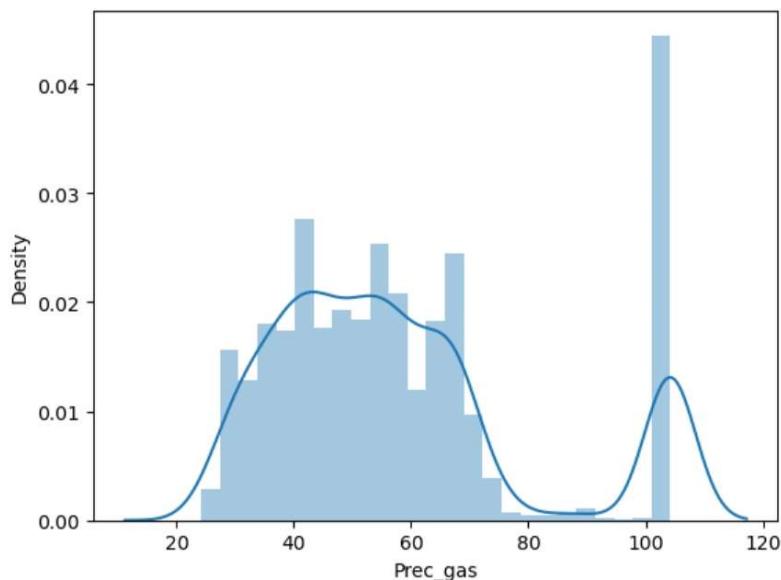
```
[16]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prec_gas'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prec_gas'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prec_gas" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prec_gas" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Prec_gas" en "train" es 4.08
El valor Maximo para "Prec_gas" en "train" es 104.2

```
[17]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prec_gas'] < df_def_out_LR, 'Prec_gas'] = df_def_out_LR
df_def_out.loc[df_def_out['Prec_gas'] > df_def_out_UR, 'Prec_gas'] = df_def_out_UR
```

```
[18]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prec_gas'])
```

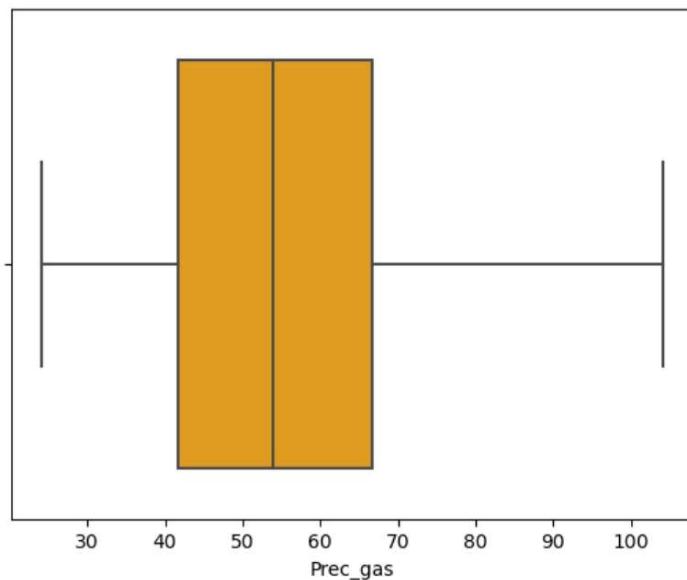
```
[18]: <AxesSubplot:xlabel='Prec_gas', ylabel='Density'>
```



MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

```
[19]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prec_gas'], color = "orange")
```

```
[19]: <AxesSubplot:xlabel='Prec_gas'>
```

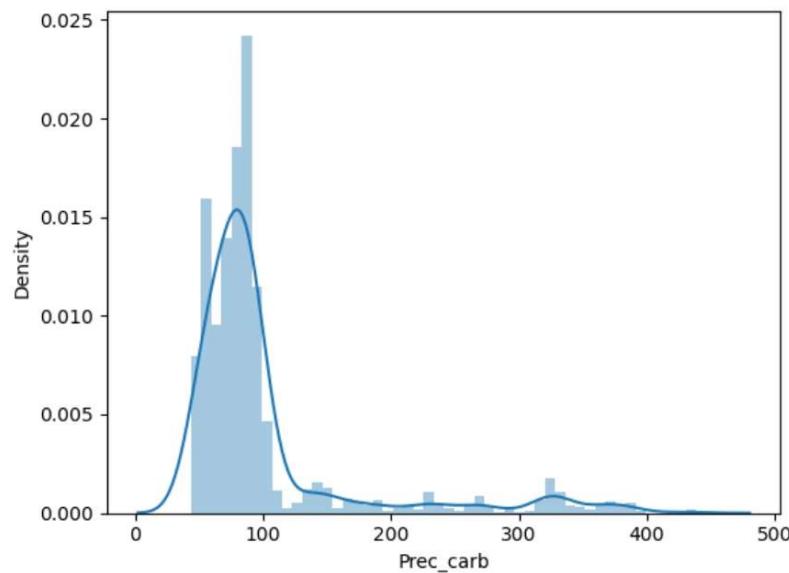


0.1.5 Outliers en variable “Prec_carb”

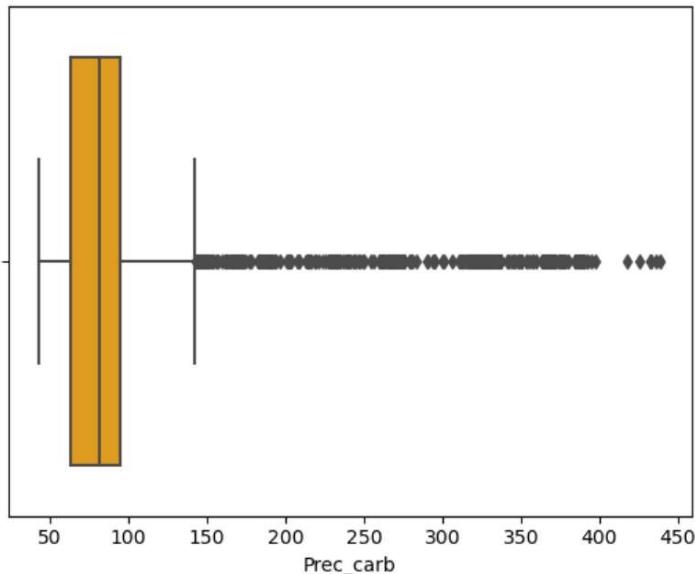
```
[20]: # Obtenemos la distribucion de la variable "Prec_carb" en "train" y en "test"
sns.distplot(df_def_out['Prec_carb'])
```

```
[20]: <AxesSubplot:xlabel='Prec_carb', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[21]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prec_carb'], color = "orange")  
  
[21]: <AxesSubplot:xlabel='Prec_carb'>
```



```
[22]: # Obtenemos las estadísticas de los datos  
print(df_def_out.Prec_carb.describe())
```

```
count    3652.000000  
mean     101.295485  
std      71.594866  
min      43.400000  
25%     63.037500  
50%     82.050000  
75%     94.800000  
max     439.000000  
Name: Prec_carb, dtype: float64
```

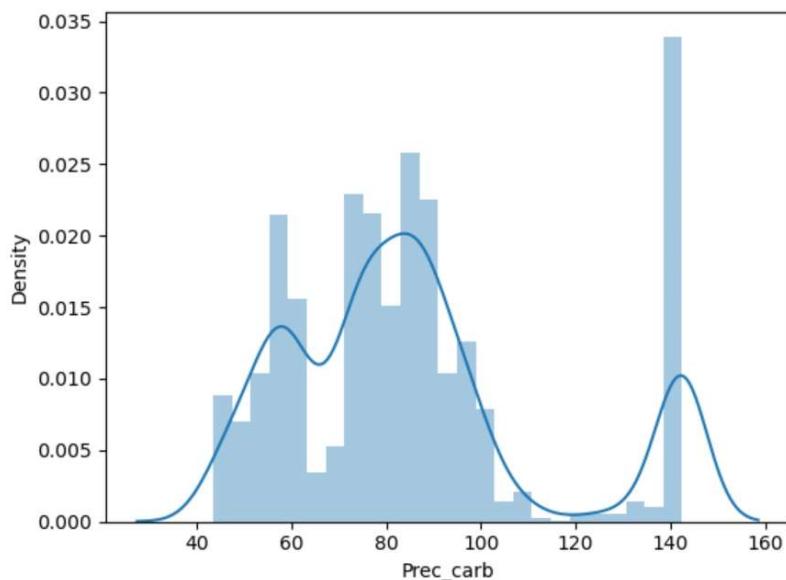
```
[23]: # Realizamos el filtrado intercuartílico en los datos "train"  
df_def_out_Q1 = df_def_out['Prec_carb'].quantile(0.25)  
df_def_out_Q3 = df_def_out['Prec_carb'].quantile(0.75)  
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1  
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)  
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)  
print ('El valor Minimo para "Prec_carb" en "train" es '+ str(df_def_out_LR))  
print ('El valor Maximo para "Prec_carb" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Prec_carb" en "train" es 15.3938
El valor Maximo para "Prec_carb" en "train" es 142.4438

```
[24]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prec_carb'] < df_def_out_LR, 'Prec_carb'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prec_carb'] > df_def_out_UR, 'Prec_carb'] =_
    df_def_out_UR
```

```
[25]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prec_carb'])
```

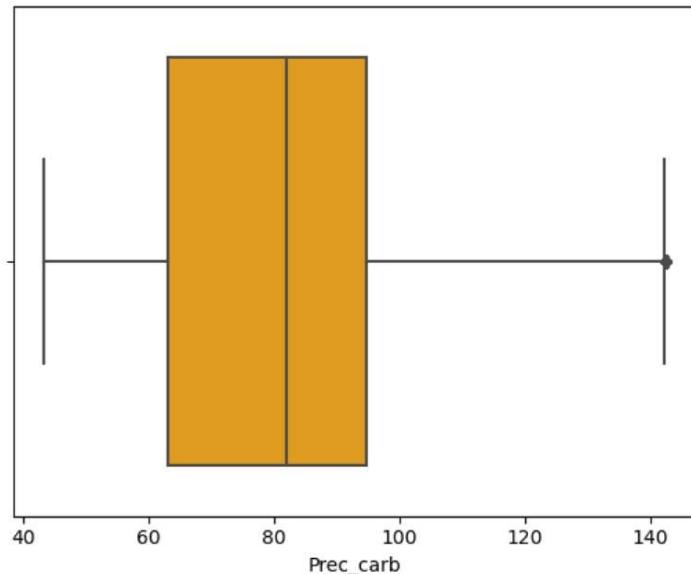
```
[25]: <AxesSubplot:xlabel='Prec_carb', ylabel='Density'>
```



```
[26]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prec_carb'], color = "orange")
```

```
[26]: <AxesSubplot:xlabel='Prec_carb'>
```

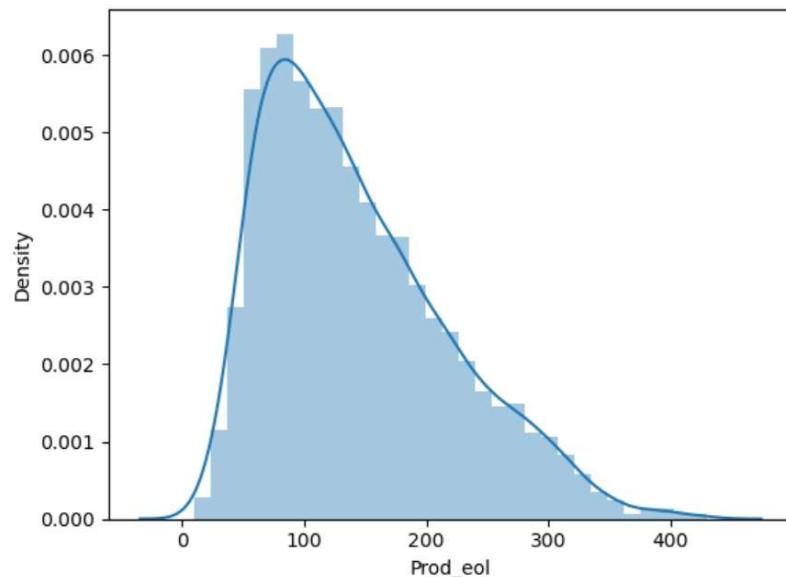
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.6 Outliers en variable “Prod_eol”

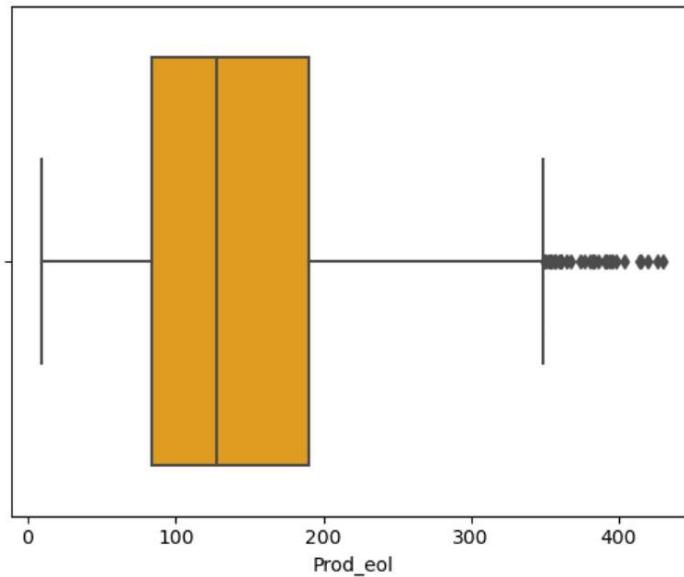
```
[27]: # Obtenemos la distribucion de la variable "Prod_eol" en "train" y en "test"  
sns.distplot(df_def_out['Prod_eol'])  
  
[27]: <AxesSubplot:xlabel='Prod_eol', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[28]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_eol'], color = "orange")
```

```
[28]: <AxesSubplot:xlabel='Prod_eol'>
```



```
[29]: # Obtenemos las estadísticas de los datos  
print(df_def_out.Prod_eol.describe())
```

```
count    3652.000000  
mean     143.394758  
std      75.930455  
min      9.687570  
25%     84.027658  
50%     127.642698  
75%     190.209523  
max     430.147947  
Name: Prod_eol, dtype: float64
```

```
[30]: # Realizamos el filtrado intercuartílico en los datos "train"  
df_def_out_Q1 = df_def_out['Prod_eol'].quantile(0.25)  
df_def_out_Q3 = df_def_out['Prod_eol'].quantile(0.75)  
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1  
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)  
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)  
print ('El valor Minimo para "Prod_eol" en "train" es '+ str(df_def_out_LR))  
print ('El valor Maximo para "Prod_eol" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_eol" en "train" es -75.2451
El valor Maximo para "Prod_eol" en "train" es 349.4823

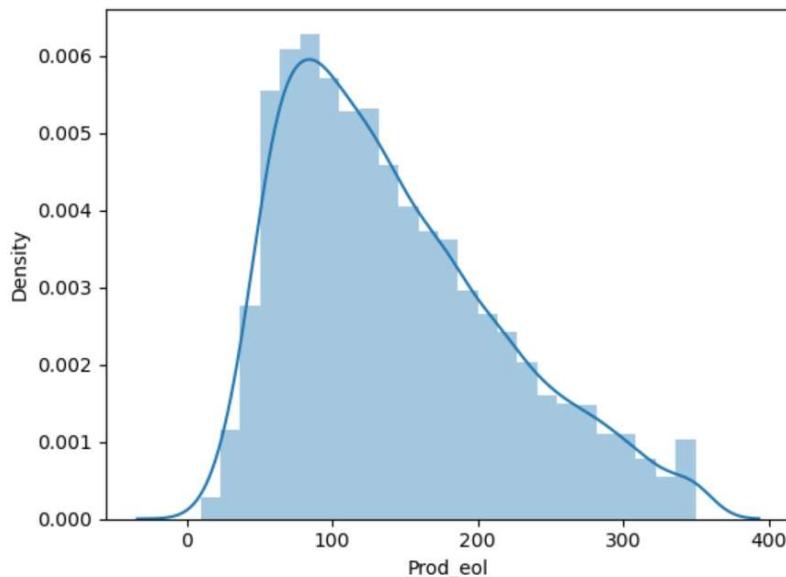
```
[31]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_eol'] < df_def_out_LR, 'Prod_eol'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prod_eol'] > df_def_out_UR, 'Prod_eol'] =_
    df_def_out_UR
```



```
[32]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_eol'])
```



```
[32]: <AxesSubplot:xlabel='Prod_eol', ylabel='Density'>
```

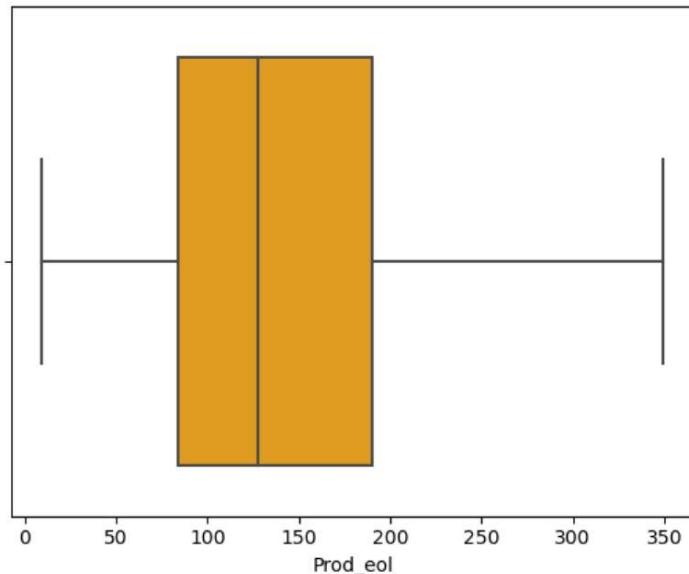


```
[33]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_eol'], color = "orange")
```



```
[33]: <AxesSubplot:xlabel='Prod_eol'>
```

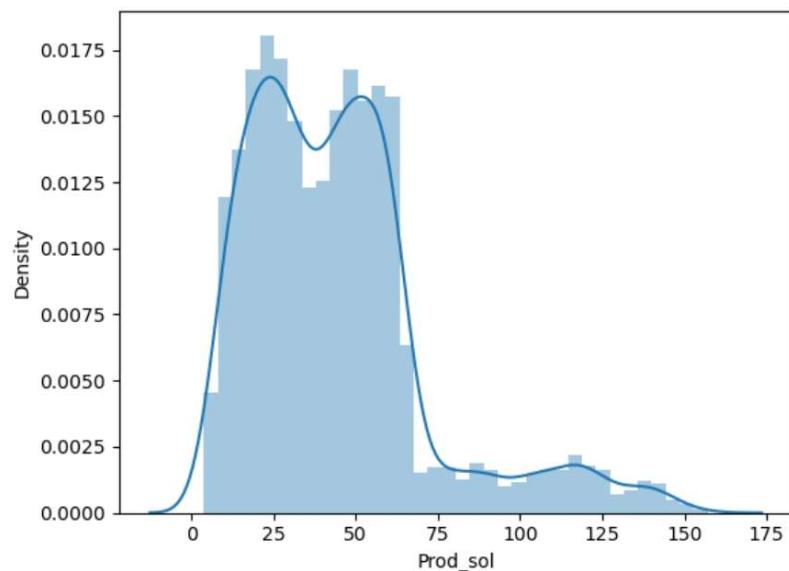
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



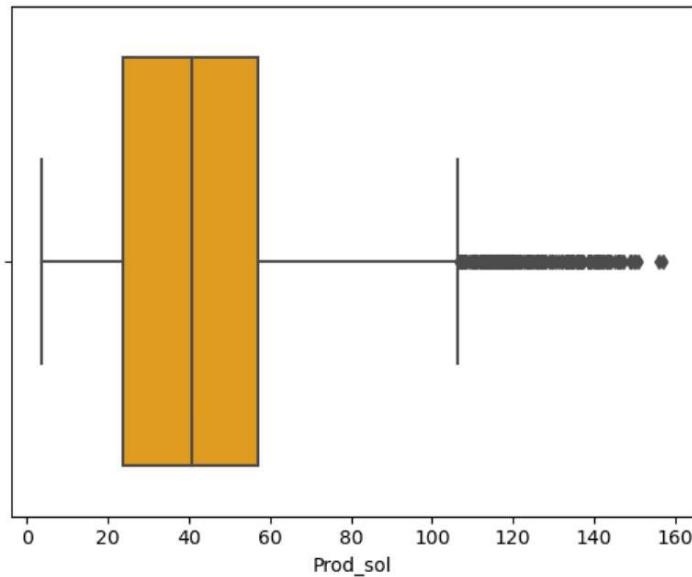
0.1.7 Outliers en variable “Prod_sol”

```
[34]: # Obtenemos la distribucion de la variable "Prod_sol" en "train" y en "test"  
sns.distplot(df_def_out['Prod_sol'])  
  
[34]: <AxesSubplot:xlabel='Prod_sol', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[35]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_sol'], color = "orange")  
  
[35]: <AxesSubplot:xlabel='Prod_sol'>
```



```
[36]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_sol.describe())
```

```
count    3652.000000
mean     44.367265
std      28.189011
min      3.794730
25%     23.711895
50%     40.570913
75%     56.877325
max     157.109191
Name: Prod_sol, dtype: float64
```

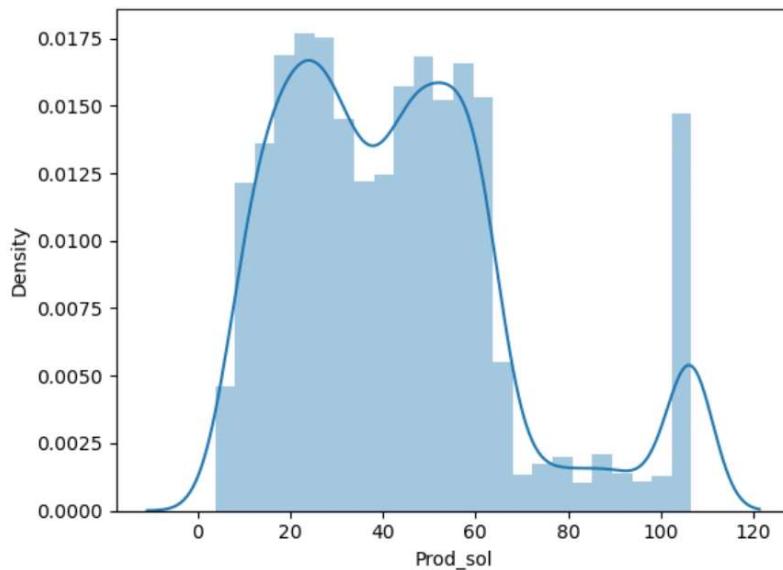
```
[37]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_sol'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_sol'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_sol" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_sol" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Prod_sol" en "train" es -26.0362
El valor Maximo para "Prod_sol" en "train" es 106.6255

```
[38]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_sol'] < df_def_out_LR, 'Prod_sol'] = df_def_out_LR
df_def_out.loc[df_def_out['Prod_sol'] > df_def_out_UR, 'Prod_sol'] = df_def_out_UR
```

```
[39]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_sol'])
```

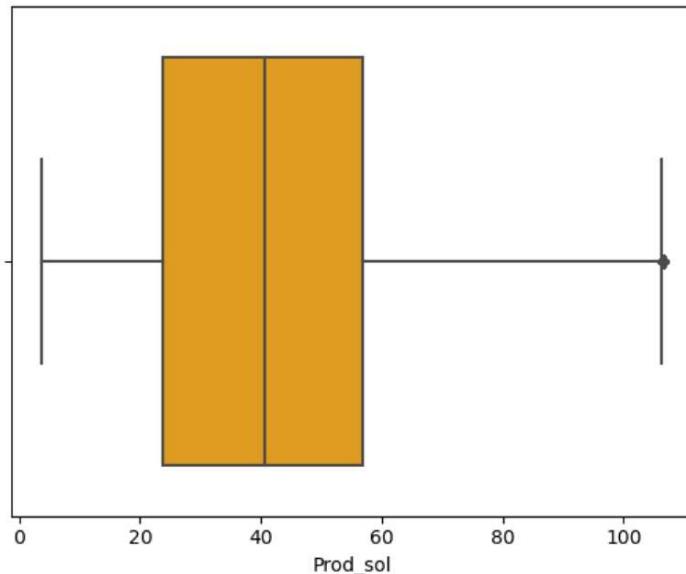
```
[39]: <AxesSubplot:xlabel='Prod_sol', ylabel='Density'>
```



```
[40]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_sol'], color = "orange")
```

```
[40]: <AxesSubplot:xlabel='Prod_sol'>
```

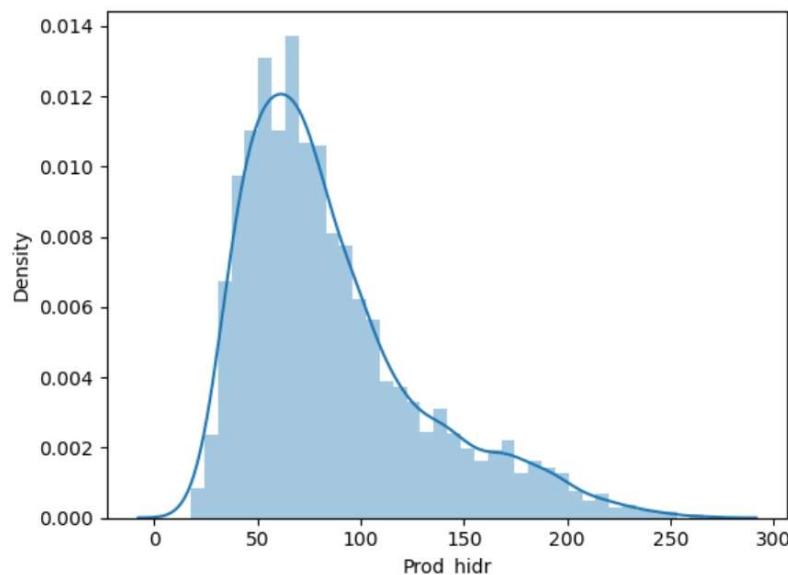
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.8 Outliers en variable “Prod_hidr”

```
[41]: # Obtenemos la distribucion de la variable "Prod_hidr" en "train" y en "test"  
sns.distplot(df_def_out['Prod_hidr'])  
  
[41]: <AxesSubplot:xlabel='Prod_hidr', ylabel='Density'>
```

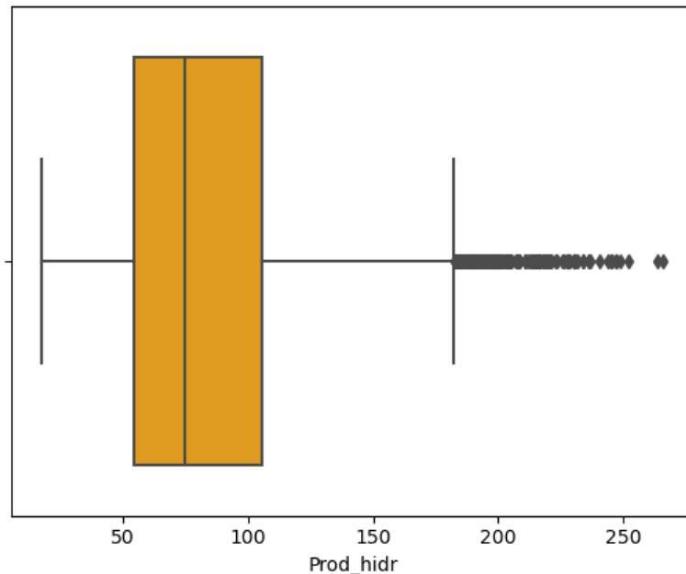
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[42]: # Dibujamos el boxplot para los datos "train"
      sns.boxplot(x = df_def_out['Prod_hidr'], color = "orange")
```



```
[42]: <AxesSubplot:xlabel='Prod_hidr'>
```



```
[43]: # Obtenemos las estadísticas de los datos  
print(df_def_out.Prod_hidr.describe())
```

```
count    3652.000000  
mean     86.196582  
std      43.802991  
min      17.719687  
25%      54.432246  
50%      74.952434  
75%      105.637833  
max      266.074453  
Name: Prod_hidr, dtype: float64
```

```
[44]: # Realizamos el filtrado intercuartílico en los datos "train"  
df_def_out_Q1 = df_def_out['Prod_hidr'].quantile(0.25)  
df_def_out_Q3 = df_def_out['Prod_hidr'].quantile(0.75)  
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1  
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)  
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)  
print ('El valor Minimo para "Prod_hidr" en "train" es '+ str(df_def_out_LR))  
print ('El valor Maximo para "Prod_hidr" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_hidr" en "train" es -22.3761
El valor Maximo para "Prod_hidr" en "train" es 182.4462

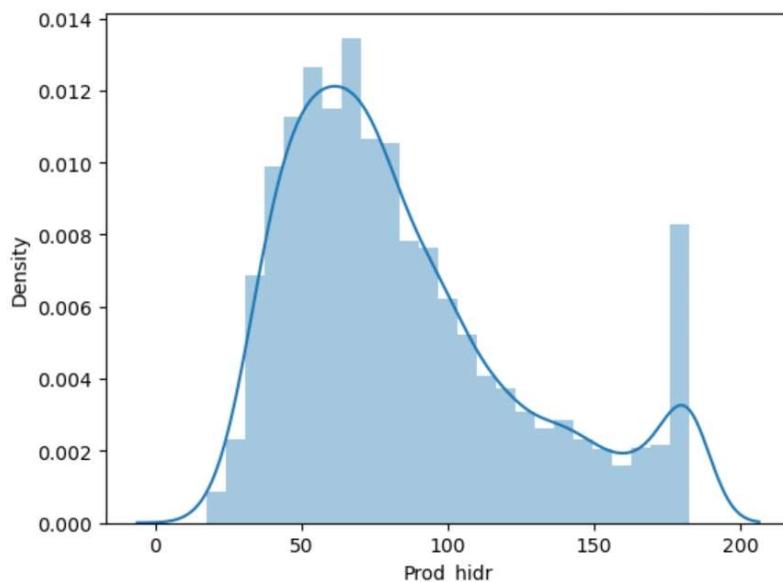
```
[45]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_hidr'] < df_def_out_LR, 'Prod_hidr'] =_
df_def_out_LR
df_def_out.loc[df_def_out['Prod_hidr'] > df_def_out_UR, 'Prod_hidr'] =_
df_def_out_UR
```



```
[46]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_hidr'])
```



```
[46]: <AxesSubplot:xlabel='Prod_hidr', ylabel='Density'>
```

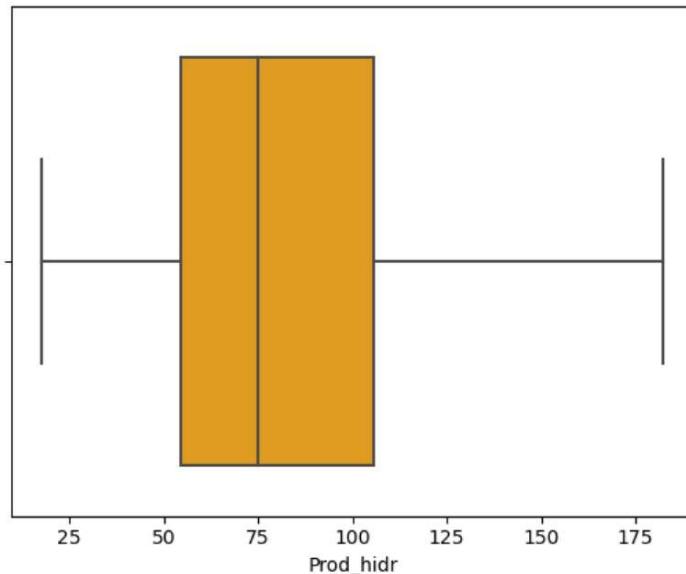


```
[47]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_hidr'], color = "orange")
```



```
[47]: <AxesSubplot:xlabel='Prod_hidr'>
```

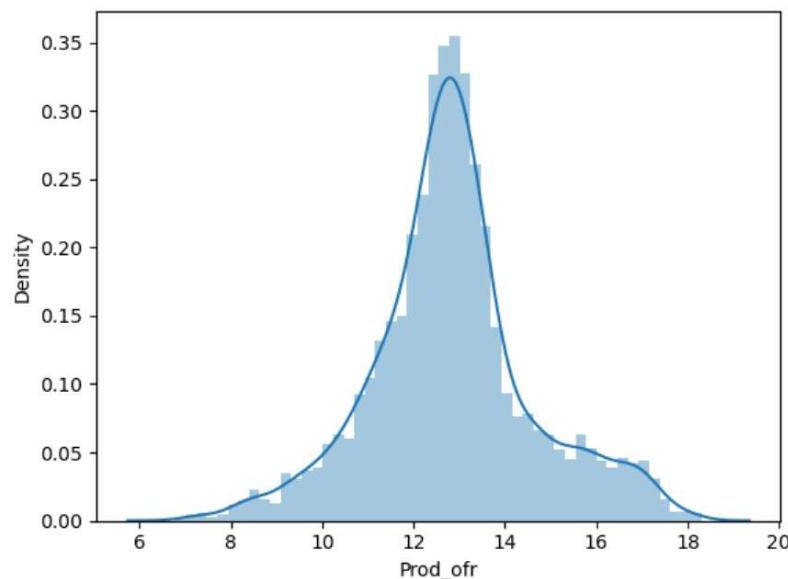
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.9 Outliers en variable “Prod_ofr”

```
[48]: # Obtenemos la distribucion de la variable "Prod_ofr" en "train" y en "test"  
sns.distplot(df_def_out['Prod_ofr'])  
  
[48]: <AxesSubplot:xlabel='Prod_ofr', ylabel='Density'>
```

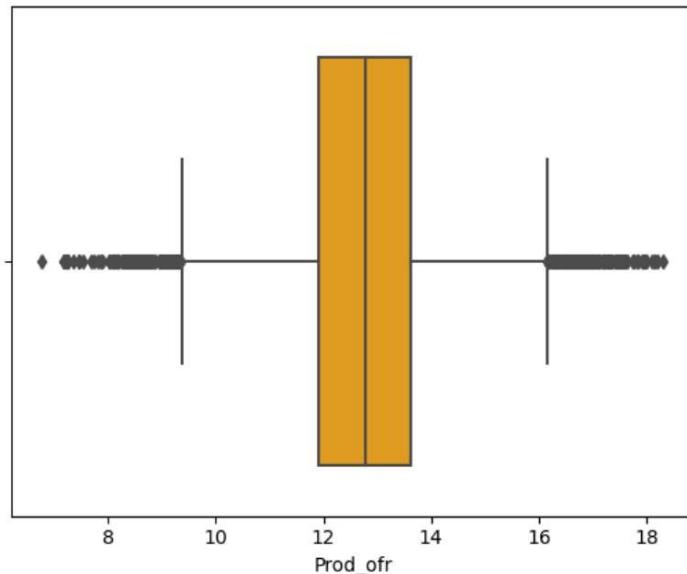
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[49]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_ofr'], color = "orange")
```

```
[49]: <AxesSubplot:xlabel='Prod_ofr'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[50]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_ofr.describe())

count    3652.000000
mean     12.846611
std      1.789711
min      6.780660
25%     11.914803
50%     12.779501
75%     13.609738
max     18.308854
Name: Prod_ofr, dtype: float64

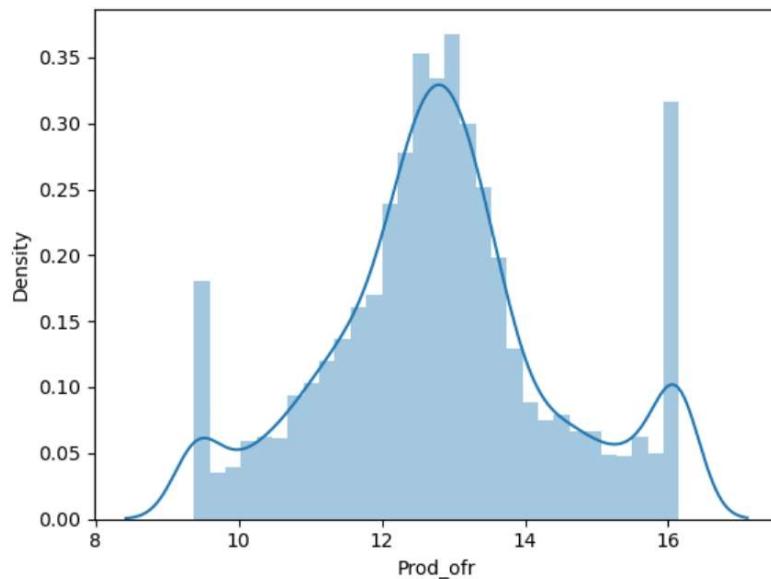
[51]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_ofr'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_ofr'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_ofr" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_ofr" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Prod_ofr" en "train" es 9.3724
El valor Maximo para "Prod_ofr" en "train" es 16.1521

```
[52]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_ofr'] < df_def_out_LR, 'Prod_ofr'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prod_ofr'] > df_def_out_UR, 'Prod_ofr'] =_
    df_def_out_UR
```

```
[53]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_ofr'])
```

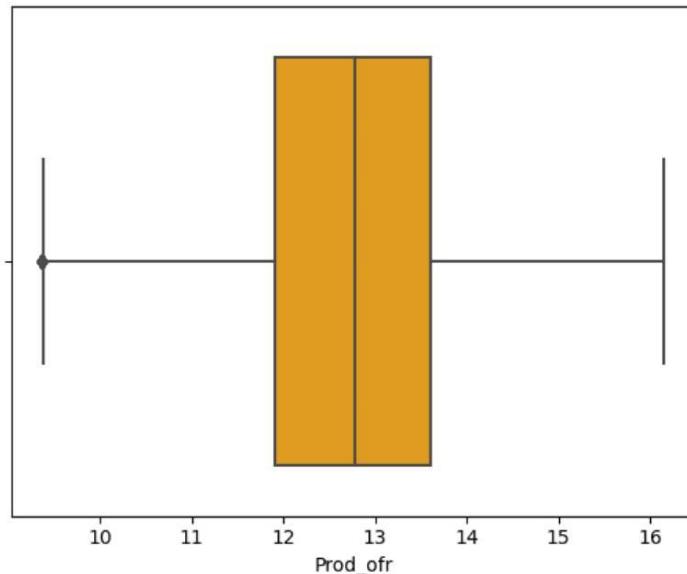
```
[53]: <AxesSubplot:xlabel='Prod_ofr', ylabel='Density'>
```



```
[54]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_ofr'], color = "orange")
```

```
[54]: <AxesSubplot:xlabel='Prod_ofr'>
```

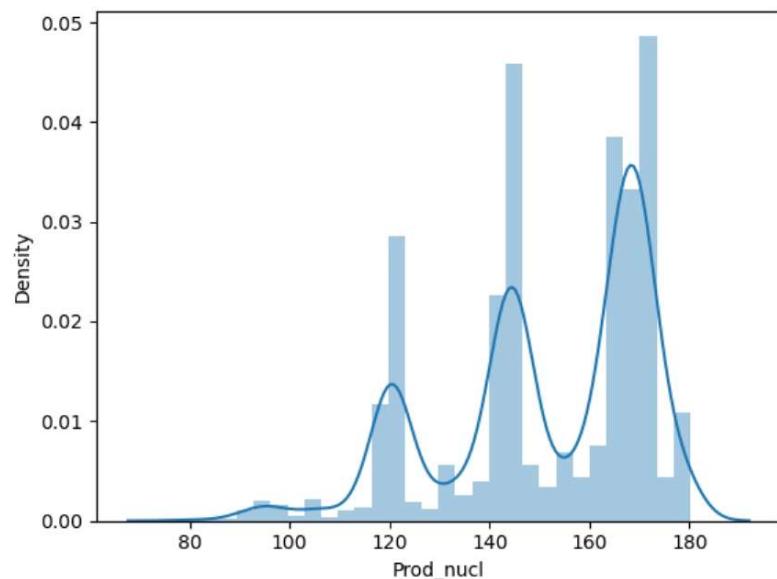
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.10 Outliers en variable “Prod_nucl”

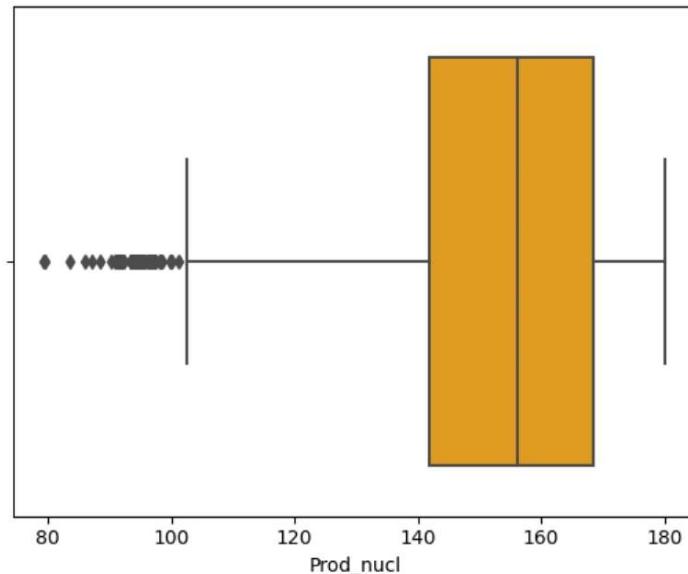
```
[55]: # Obtenemos la distribucion de la variable "Prod_nucl" en "train" y en "test"  
sns.distplot(df_def_out['Prod_nucl'])  
  
[55]: <AxesSubplot:xlabel='Prod_nucl', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[56]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_nucl'], color = "orange")
```

```
[56]: <AxesSubplot:xlabel='Prod_nucl'>
```



```
[57]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_nucl.describe())
```

```
count    3652.000000
mean     151.384671
std      20.277425
min      79.415936
25%     141.951267
50%     156.098112
75%     168.506047
max     180.165816
Name: Prod_nucl, dtype: float64
```

```
[58]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_nucl'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_nucl'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_nucl" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_nucl" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_nucl" en "train" es 102.1191
El valor Maximo para "Prod_nucl" en "train" es 208.3382

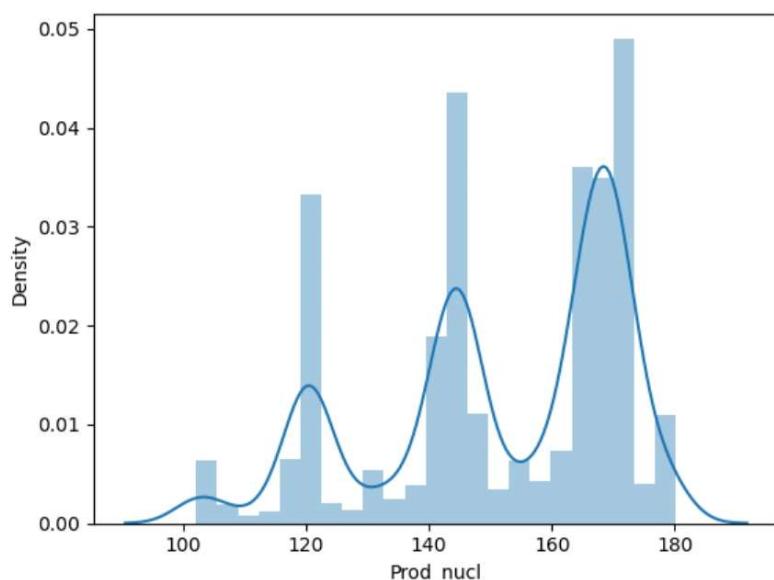
```
[59]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_nucl'] < df_def_out_LR, 'Prod_nucl'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prod_nucl'] > df_def_out_UR, 'Prod_nucl'] =_
    df_def_out_UR
```



```
[60]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_nucl'])
```



```
[60]: <AxesSubplot:xlabel='Prod_nucl', ylabel='Density'>
```

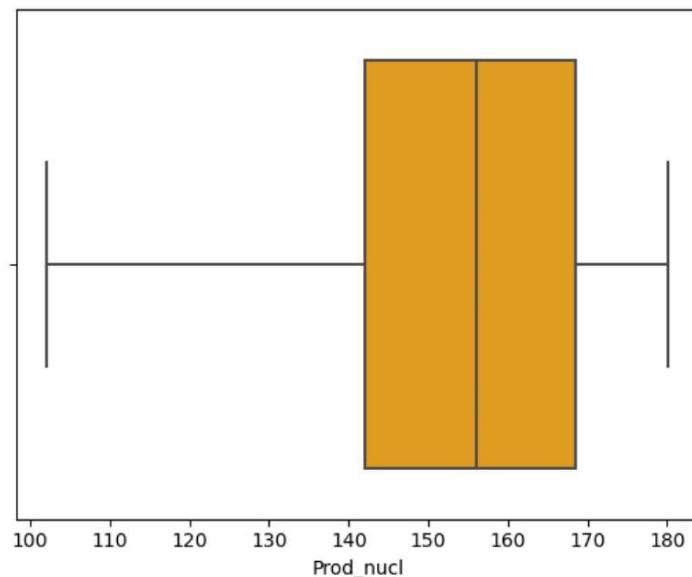


```
[61]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_nucl'], color = "orange")
```



```
[61]: <AxesSubplot:xlabel='Prod_nucl'>
```

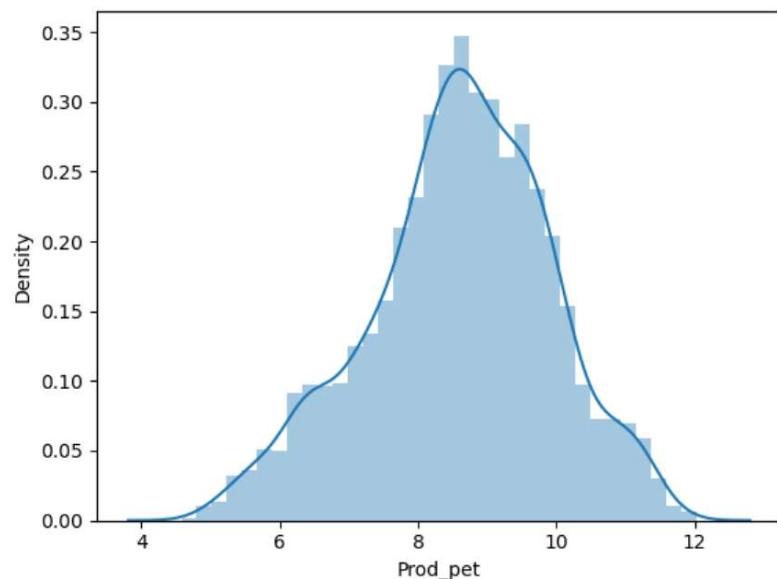
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



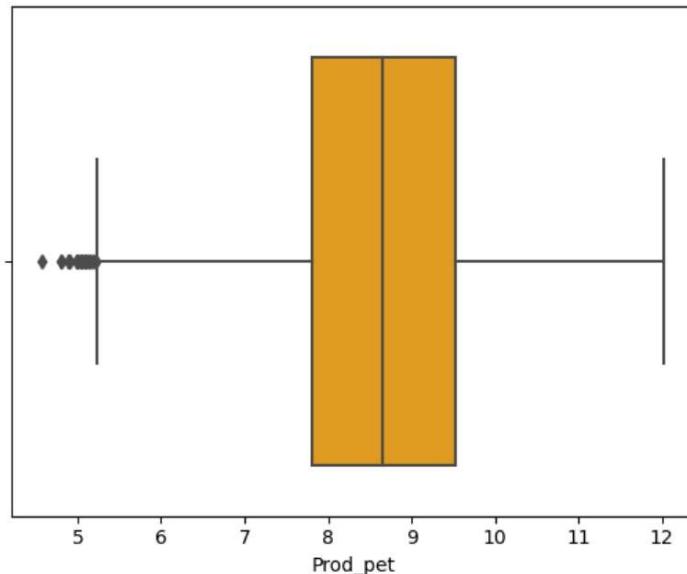
0.1.11 Outliers en variable “Prod_pet”

```
[62]: # Obtenemos la distribucion de la variable "Prod_pet" en "train" y en "test"  
sns.distplot(df_def_out['Prod_pet'])  
  
[62]: <AxesSubplot:xlabel='Prod_pet', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[63]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_pet'], color = "orange")  
  
[63]: <AxesSubplot:xlabel='Prod_pet'>
```



```
[64]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_pet.describe())
```

```
count    3652.000000
mean     8.599667
std      1.325793
min      4.579061
25%     7.804722
50%     8.655129
75%     9.527093
max     12.026109
Name: Prod_pet, dtype: float64
```

```
[65]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_pet'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_pet'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_pet" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_pet" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

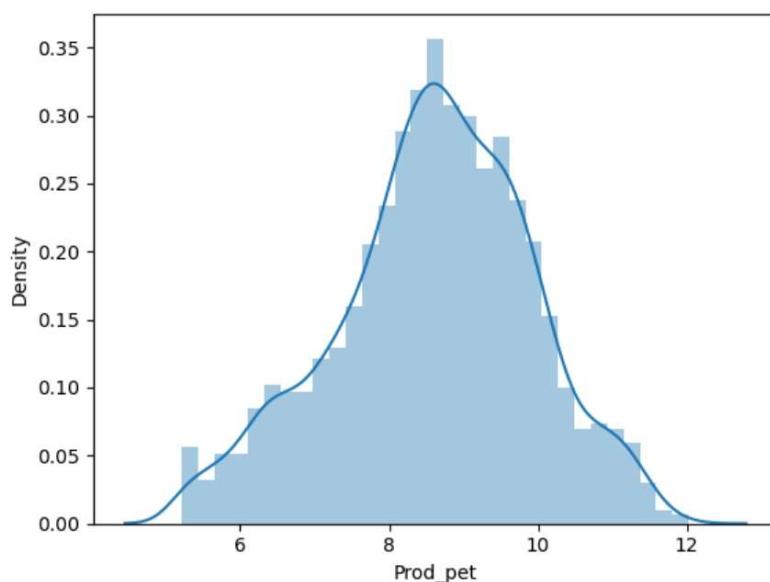
EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_pet" en "train" es 5.2212
El valor Maximo para "Prod_pet" en "train" es 12.1106

```
[66]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_pet'] < df_def_out_LR, 'Prod_pet'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prod_pet'] > df_def_out_UR, 'Prod_pet'] =_
    df_def_out_UR
```

```
[67]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_pet'])
```

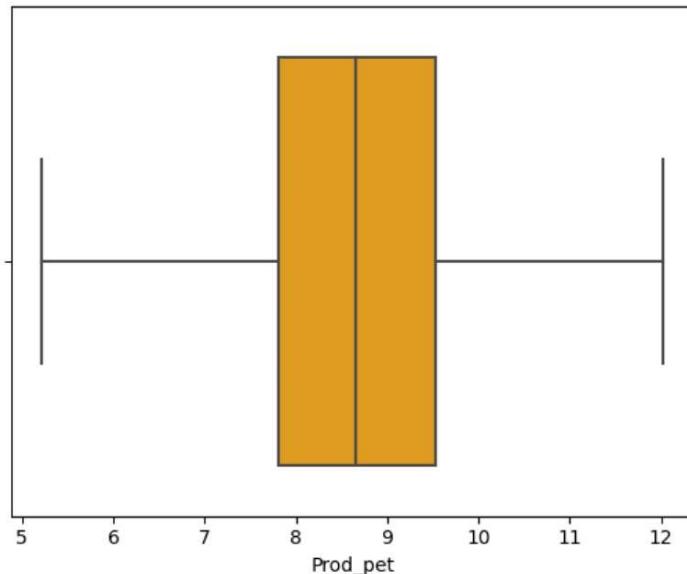
```
[67]: <AxesSubplot:xlabel='Prod_pet', ylabel='Density'>
```



```
[68]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_pet'], color = "orange")
```

```
[68]: <AxesSubplot:xlabel='Prod_pet'>
```

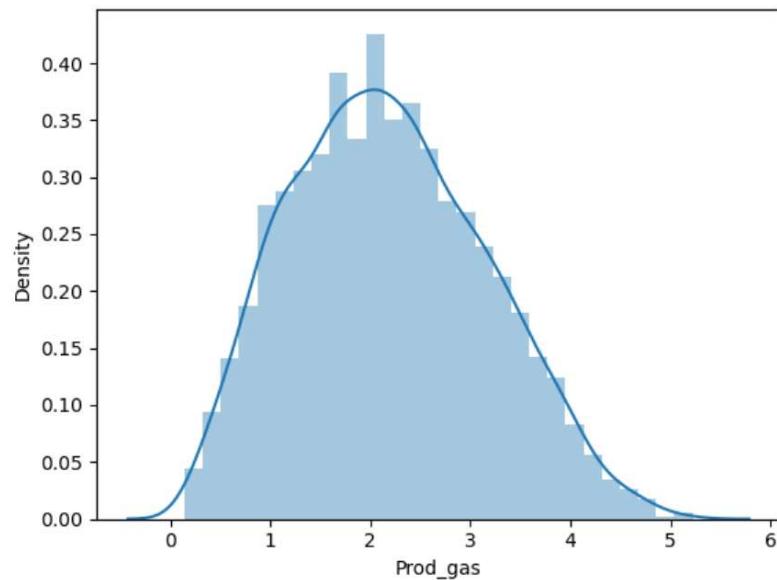
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.12 Outliers en variable “Prod_gas”

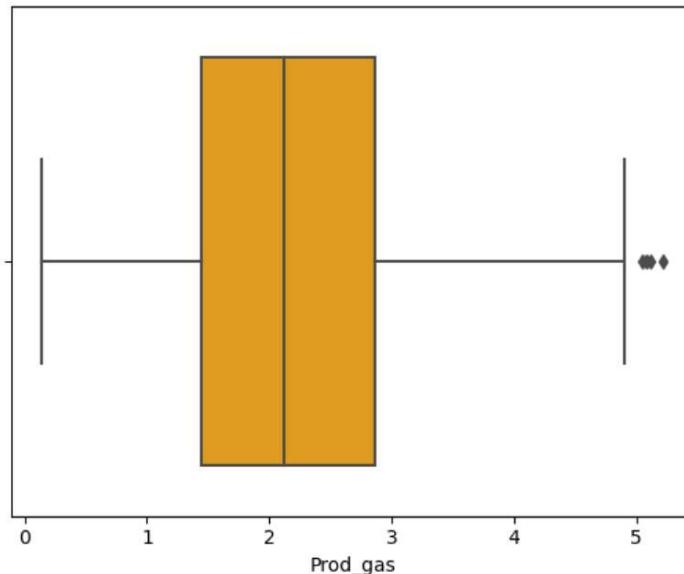
```
[69]: # Obtenemos la distribucion de la variable "Prod_gas" en "train" y en "test"  
sns.distplot(df_def_out['Prod_gas'])  
  
[69]: <AxesSubplot:xlabel='Prod_gas', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[70]: # Dibujamos el boxplot para los datos "train"  
      sns.boxplot(x = df_def_out['Prod_gas'], color = "orange")
```

```
[70]: <AxesSubplot:xlabel='Prod_gas'>
```



```
[71]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_gas.describe())
```

```
count    3652.000000
mean      2.183548
std       0.974275
min      0.143818
25%     1.443891
50%     2.119584
75%     2.868898
max      5.220280
Name: Prod_gas, dtype: float64
```

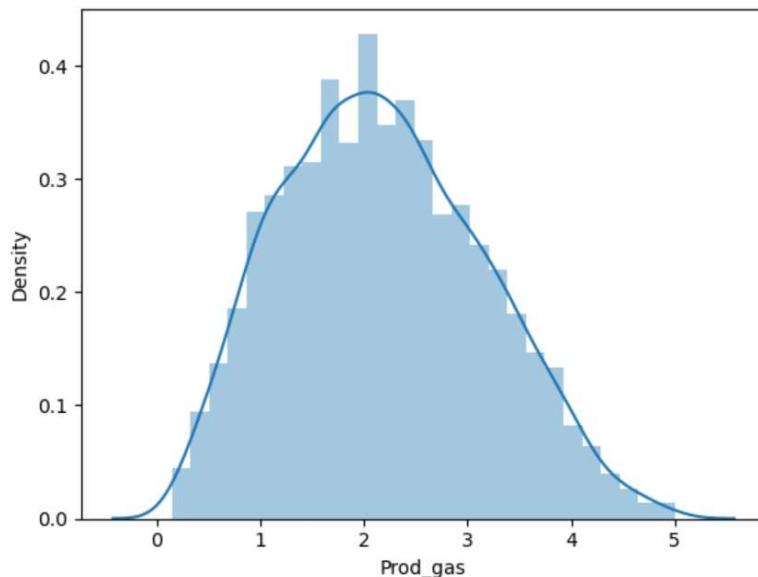
```
[72]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_gas'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_gas'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_gas" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_gas" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Prod_gas" en "train" es -0.6936
El valor Maximo para "Prod_gas" en "train" es 5.0064

```
[73]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_gas'] < df_def_out_LR, 'Prod_gas'] =_
    df_def_out_LR
df_def_out.loc[df_def_out['Prod_gas'] > df_def_out_UR, 'Prod_gas'] =_
    df_def_out_UR
```

```
[74]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_gas'])
```

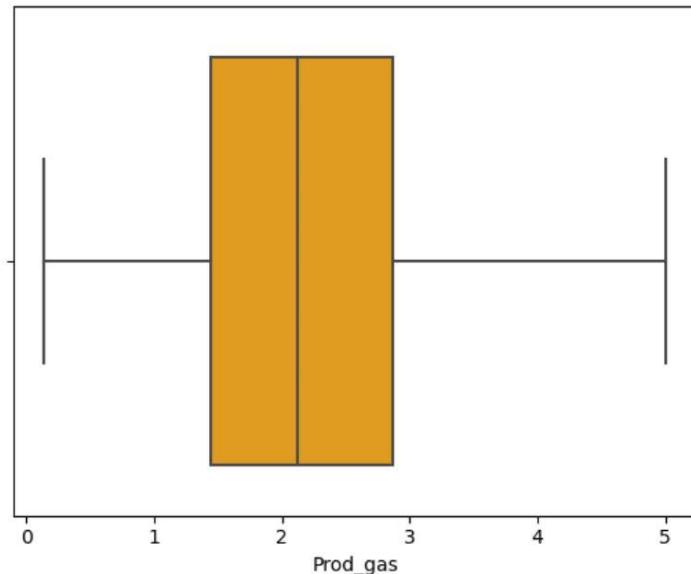
```
[74]: <AxesSubplot:xlabel='Prod_gas', ylabel='Density'>
```



```
[75]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_gas'], color = "orange")
```

```
[75]: <AxesSubplot:xlabel='Prod_gas'>
```

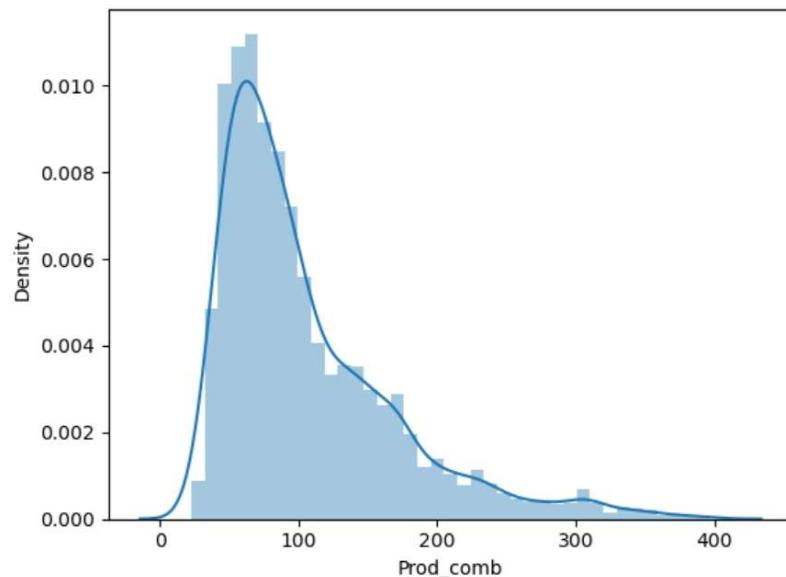
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.13 Outliers en variable “Prod_comb”

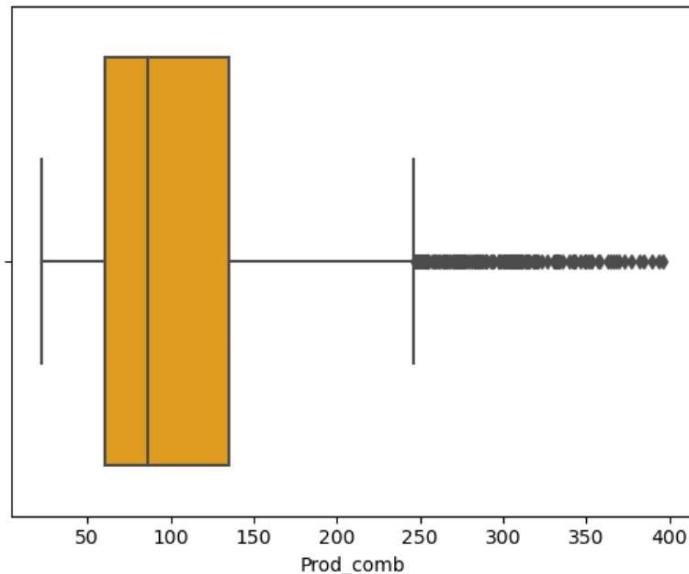
```
[76]: # Obtenemos la distribucion de la variable "Prod_comb" en "train" y en "test"  
sns.distplot(df_def_out['Prod_comb'])  
  
[76]: <AxesSubplot:xlabel='Prod_comb', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[77]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_comb'], color = "orange")
```

```
[77]: <AxesSubplot:xlabel='Prod_comb'>
```



```
[78]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_comb.describe())
```

```
count    3652.000000
mean     105.831537
std      63.749009
min      22.726758
25%     60.557129
50%     86.065697
75%    134.886550
max     396.453645
Name: Prod_comb, dtype: float64
```

```
[79]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_comb'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_comb'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_comb" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_comb" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_comb" en "train" es -50.937
El valor Maximo para "Prod_comb" en "train" es 246.3807

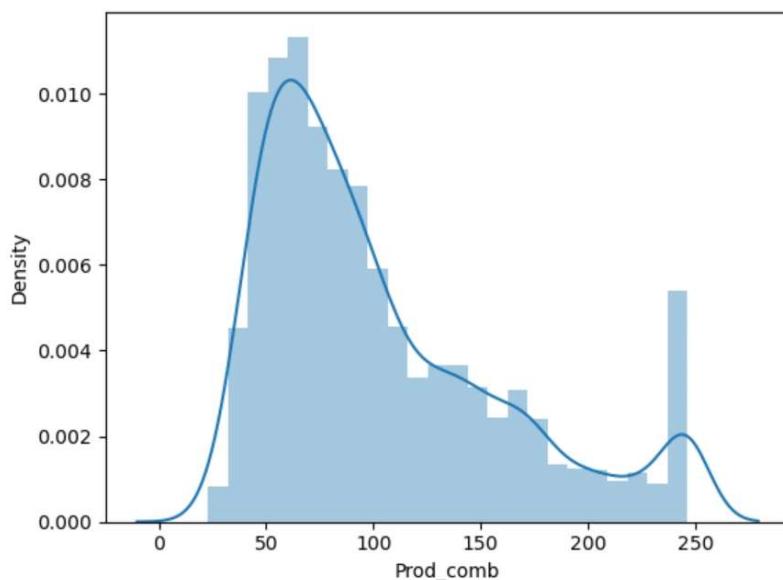
```
[80]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_comb'] < df_def_out_LR, 'Prod_comb'] =_
df_def_out_LR
df_def_out.loc[df_def_out['Prod_comb'] > df_def_out_UR, 'Prod_comb'] =_
df_def_out_UR
```



```
[81]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_comb'])
```



```
[81]: <AxesSubplot:xlabel='Prod_comb', ylabel='Density'>
```

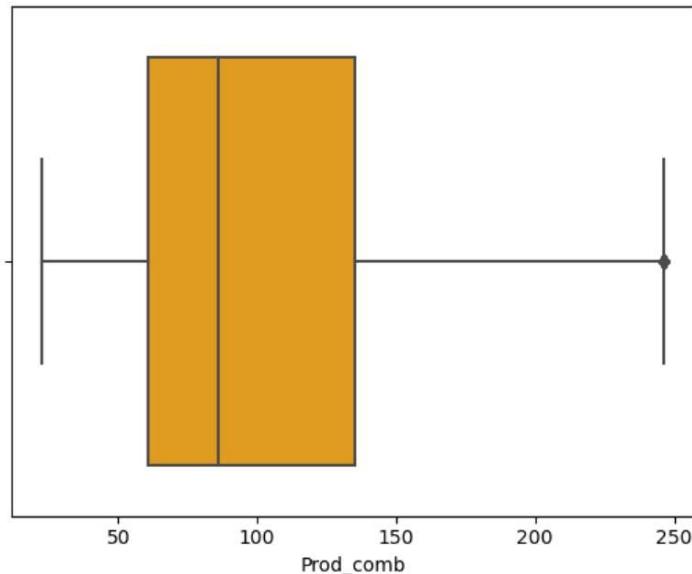


```
[82]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_comb'], color = "orange")
```



```
[82]: <AxesSubplot:xlabel='Prod_comb'>
```

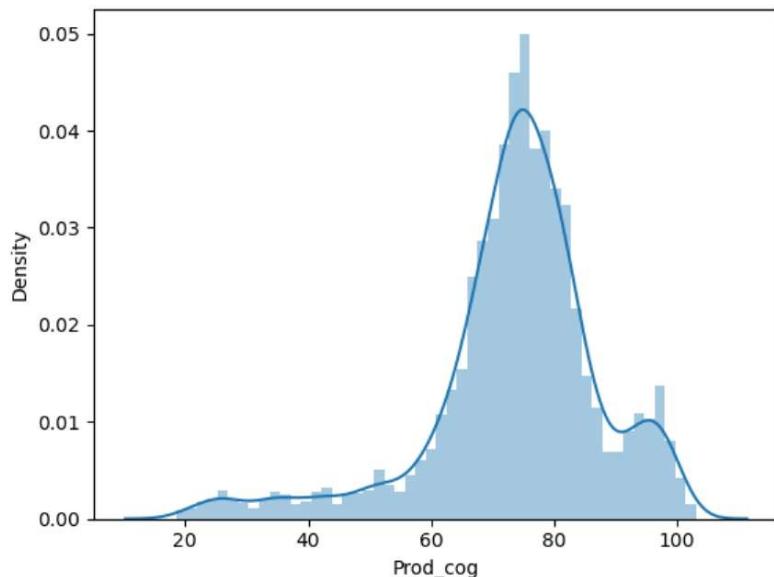
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.14 Outliers en variable “Prod_cog”

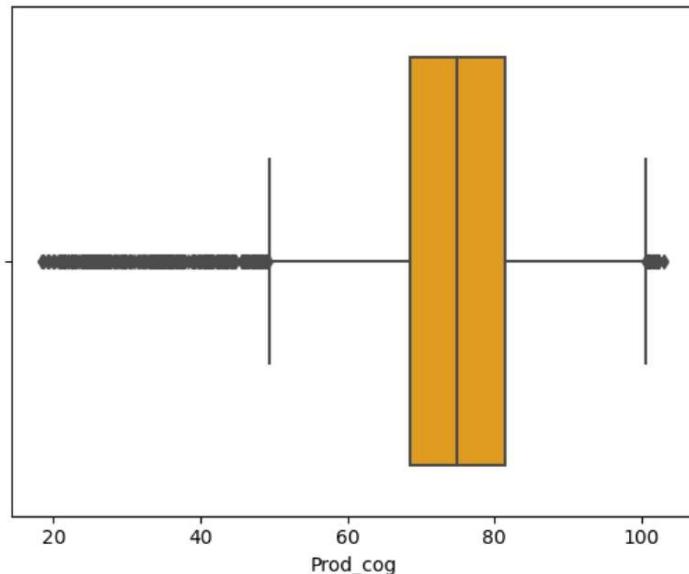
```
[83]: # Obtenemos la distribucion de la variable "Prod_cog" en "train" y en "test"  
sns.distplot(df_def_out['Prod_cog'])  
  
[83]: <AxesSubplot:xlabel='Prod_cog', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[84]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Prod_cog'], color = "orange")
```

```
[84]: <AxesSubplot:xlabel='Prod_cog'>
```



```
[85]: # Obtenemos las estadísticas de los datos
print(df_def_out.Prod_cog.describe())
```

```
count    3652.000000
mean     73.718862
std      14.195878
min      18.497770
25%     68.572433
50%     74.902375
75%     81.382255
max     103.089470
Name: Prod_cog, dtype: float64
```

```
[86]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Prod_cog'].quantile(0.25)
df_def_out_Q3 = df_def_out['Prod_cog'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Prod_cog" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Prod_cog" en "train" es '+ str(df_def_out_UR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

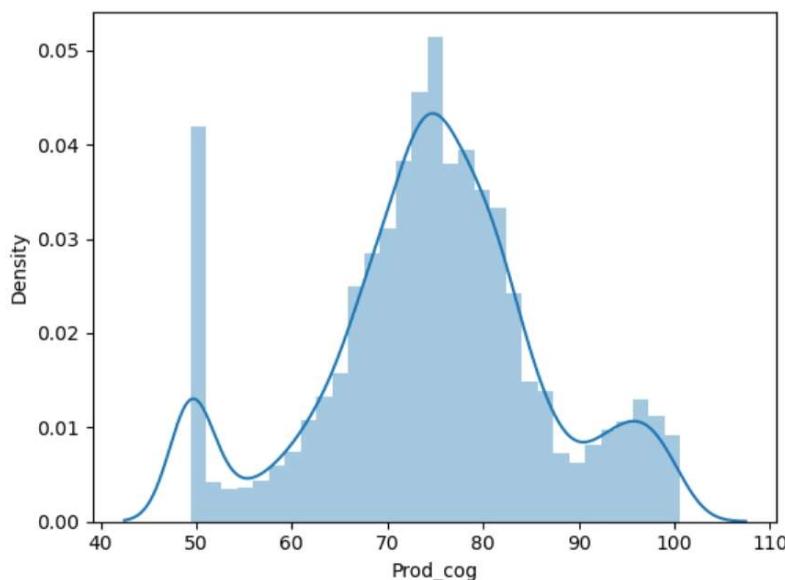
EN EL MERCADO ESPAÑOL

El valor Minimo para "Prod_cog" en "train" es 49.3577
El valor Maximo para "Prod_cog" en "train" es 100.597

```
[87]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Prod_cog'] < df_def_out_LR, 'Prod_cog'] =_
df_def_out_LR
df_def_out.loc[df_def_out['Prod_cog'] > df_def_out_UR, 'Prod_cog'] =_
df_def_out_UR
```

```
[88]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Prod_cog'])
```

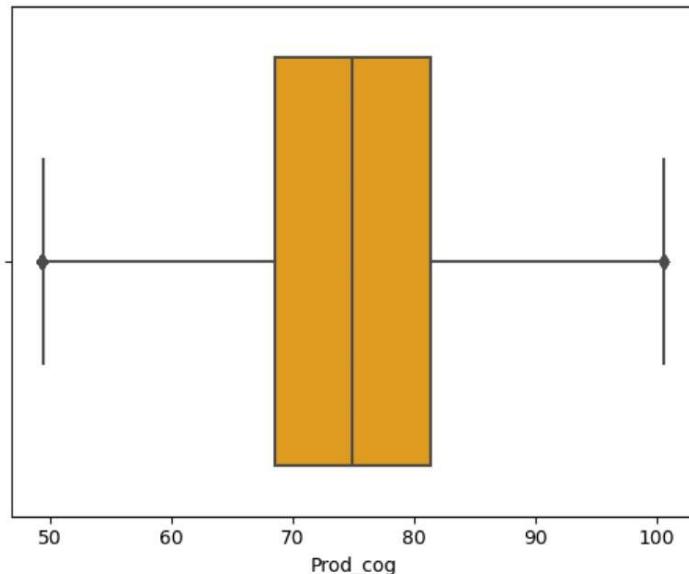
```
[88]: <AxesSubplot:xlabel='Prod_cog', ylabel='Density'>
```



```
[89]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Prod_cog'], color = "orange")
```

```
[89]: <AxesSubplot:xlabel='Prod_cog'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

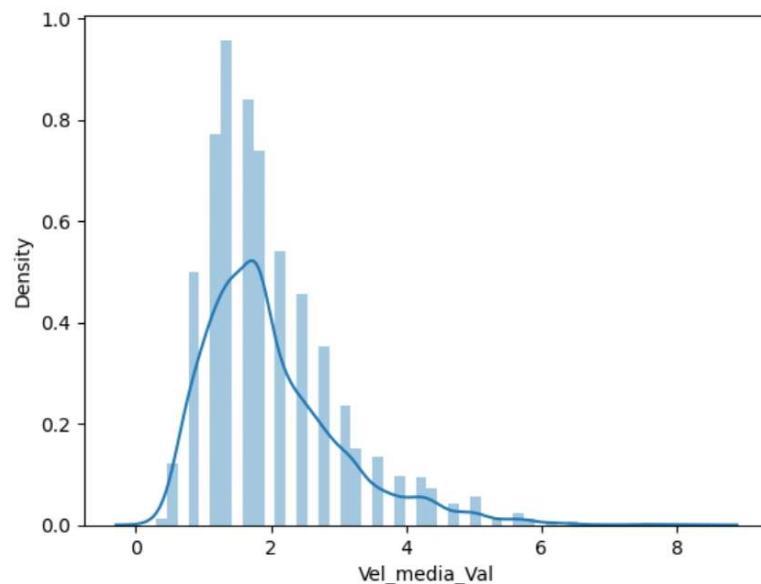


0.1.15 Outliers en variable “Vel_media_Val”

```
[90]: # Obtenemos la distribucion de la variable "Vel_media_Val" en "train" y en "test"
      sns.distplot(df_def_out['Vel_media_Val'])
```

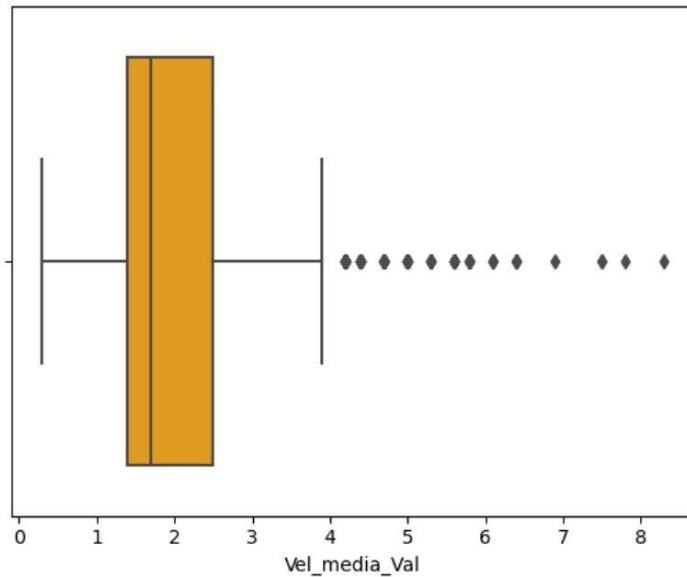
```
[90]: <AxesSubplot:xlabel='Vel_media_Val', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[91]: # Dibujamos el boxplot para los datos "train"
sns.boxplot(x = df_def_out['Vel_media_Val'], color = "orange")
```

[91]: <AxesSubplot:xlabel='Vel_media_Val'>



```
[92]: # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Val.describe())
```

```
count    3652.000000
mean     2.006298
std      1.019553
min      0.300000
25%     1.400000
50%     1.700000
75%     2.500000
max      8.300000
Name: Vel_media_Val, dtype: float64
```

```
[93]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Val'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Val'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Val" en "train" es '+_
      str(df_def_out_LR))
```

```

print ('El valor Maximo para "Vel_media_Val" en "train" es '+_
      str(df_def_out.UR))

```

El valor Minimo para "Vel_media_Val" en "train" es -0.25
 El valor Maximo para "Vel_media_Val" en "train" es 4.15

```

[94]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Vel_media_Val'] < df_def_out.LR, 'Vel_media_Val'] =_
      df_def_out.LR
df_def_out.loc[df_def_out['Vel_media_Val'] > df_def_out.UR, 'Vel_media_Val'] =_
      df_def_out.UR

```

```

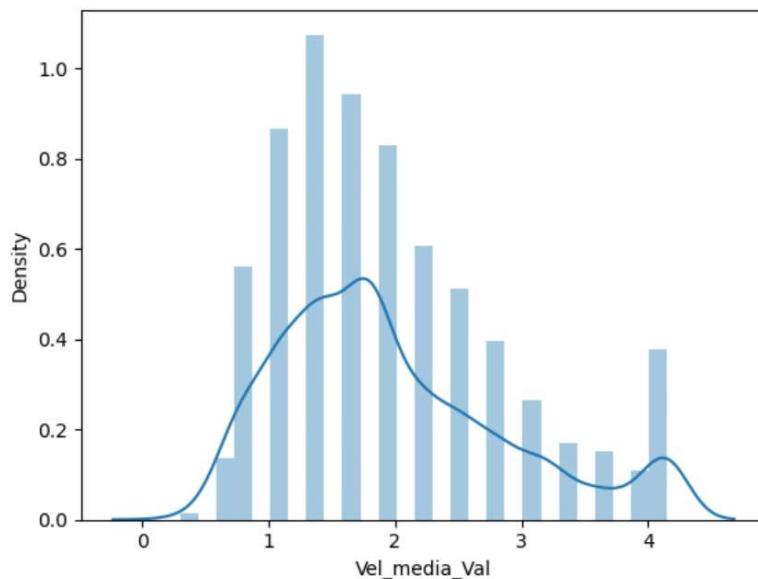
[95]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Val'])

```

```

[95]: <AxesSubplot:xlabel='Vel_media_Val', ylabel='Density'>

```



```

[96]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Val'], color = "orange")

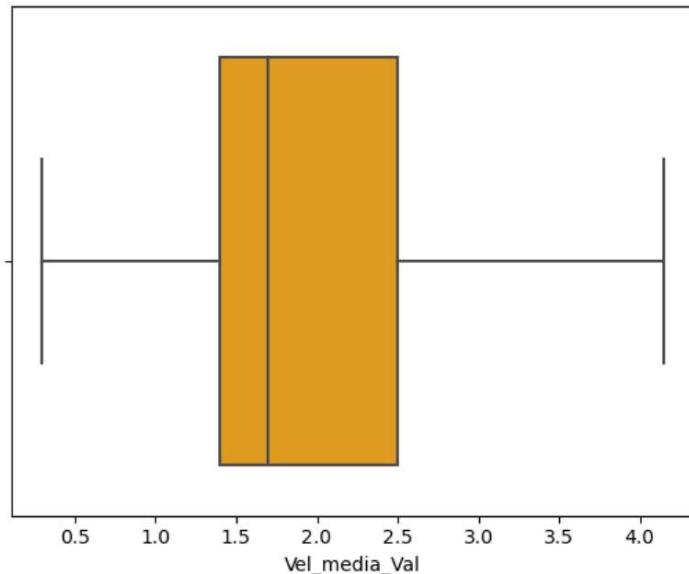
```

```

[96]: <AxesSubplot:xlabel='Vel_media_Val'>

```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



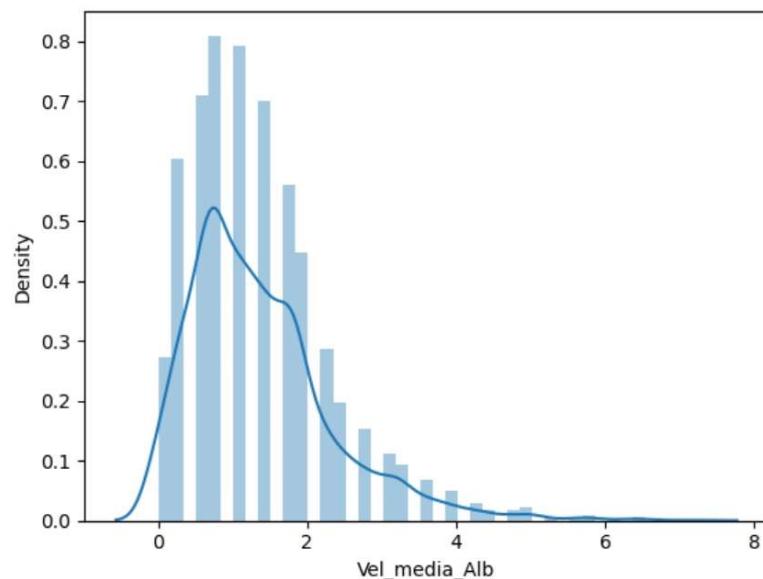
0.1.16 Outliers en variable “Vel_media_Alb”

```
[97]: # Obtenemos la distribucion de la variable "Vel_media_Alb" en "train" y en "test"
      sns.distplot(df_def_out['Vel_media_Alb'])
```



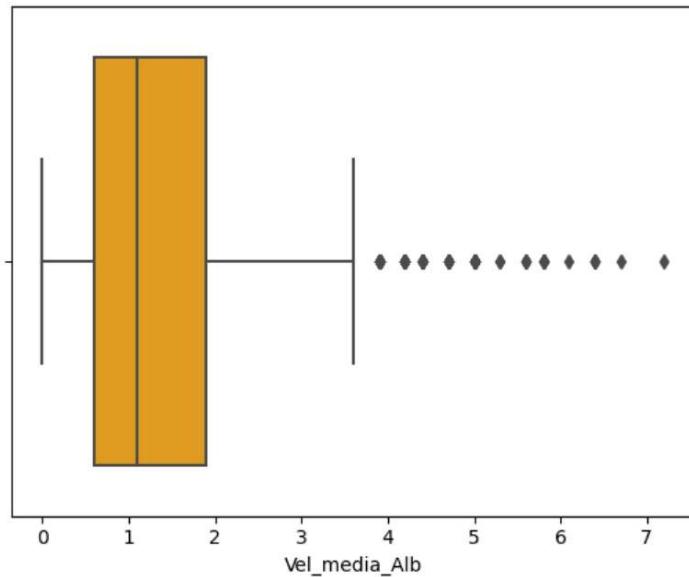
```
[97]: <AxesSubplot:xlabel='Vel_media_Alb', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[98]: # Dibujamos el boxplot para los datos "train"
      sns.boxplot(x = df_def_out['Vel_media_Alb'], color = "orange")
```

[98]: <AxesSubplot:xlabel='Vel_media_Alb'>



```
[99]: # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Alb.describe())
```

```
count    3652.000000
mean     1.360405
std      0.981072
min     0.000000
25%     0.600000
50%     1.100000
75%     1.900000
max     7.200000
Name: Vel_media_Alb, dtype: float64
```

```
[100]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Alb'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Alb'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Alb" en "train" es '+_
      str(df_def_out_LR))
```

```

print ('El valor Maximo para "Vel_media_Alb" en "train" es '+_
      str(df_def_out.UR))

```

El valor Minimo para "Vel_media_Alb" en "train" es -1.35
 El valor Maximo para "Vel_media_Alb" en "train" es 3.85

```

[101]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Vel_media_Alb'] < df_def_out.LR, 'Vel_media_Alb'] =_
      df_def_out.LR
df_def_out.loc[df_def_out['Vel_media_Alb'] > df_def_out.UR, 'Vel_media_Alb'] =_
      df_def_out.UR

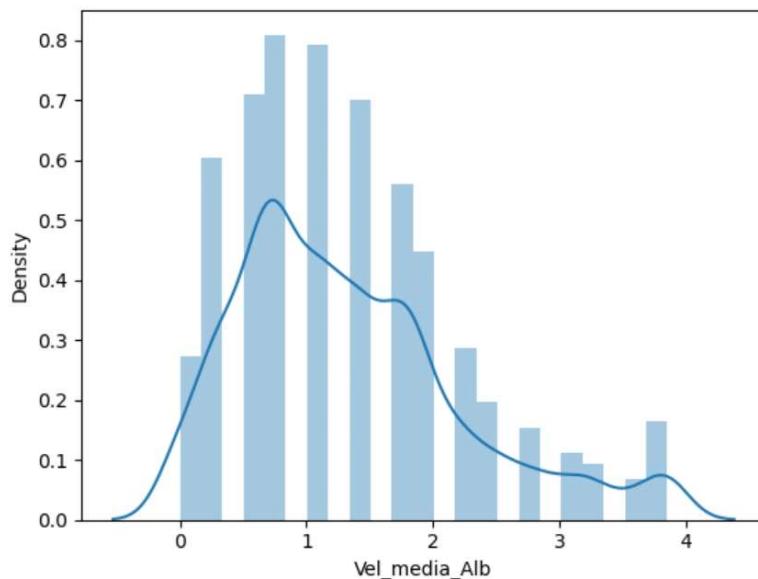
```

```

[102]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Alb'])

```

[102]: <AxesSubplot:xlabel='Vel_media_Alb', ylabel='Density'>



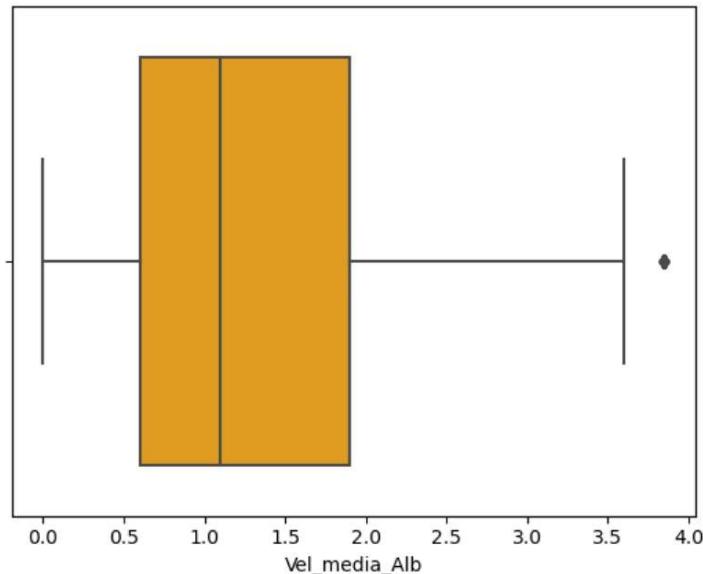
```

[103]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Alb'], color = "orange")

```

[103]: <AxesSubplot:xlabel='Vel_media_Alb'>

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

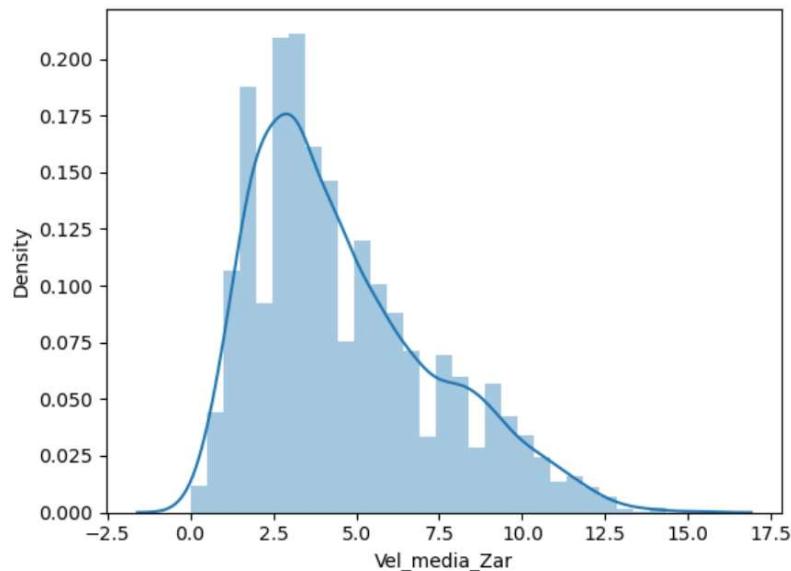


0.1.17 Outliers en variable “Vel_media_Zar”

```
[104]: # Obtenemos la distribucion de la variable "Vel_media_Zar" en "train" y en "test"
      sns.distplot(df_def_out['Vel_media_Zar'])
```

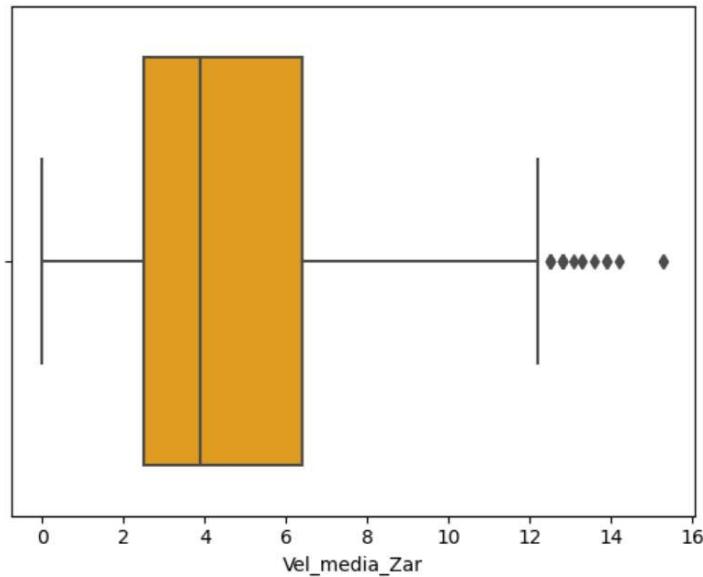
[104]: <AxesSubplot:xlabel='Vel_media_Zar', ylabel='Density'>

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[105]: # Dibujamos el boxplot para los datos "train"
      sns.boxplot(x = df_def_out['Vel_media_Zar'], color = "orange")
```

[105]: <AxesSubplot:xlabel='Vel_media_Zar'>



```
[106]: # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Zar.describe())
```

count	3652.000000
mean	4.644195
std	2.745516
min	0.000000
25%	2.500000
50%	3.900000
75%	6.400000
max	15.300000
Name:	Vel_media_Zar, dtype: float64

```
[107]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Zar'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Zar'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Zar" en "train" es '+_
      str(df_def_out_LR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

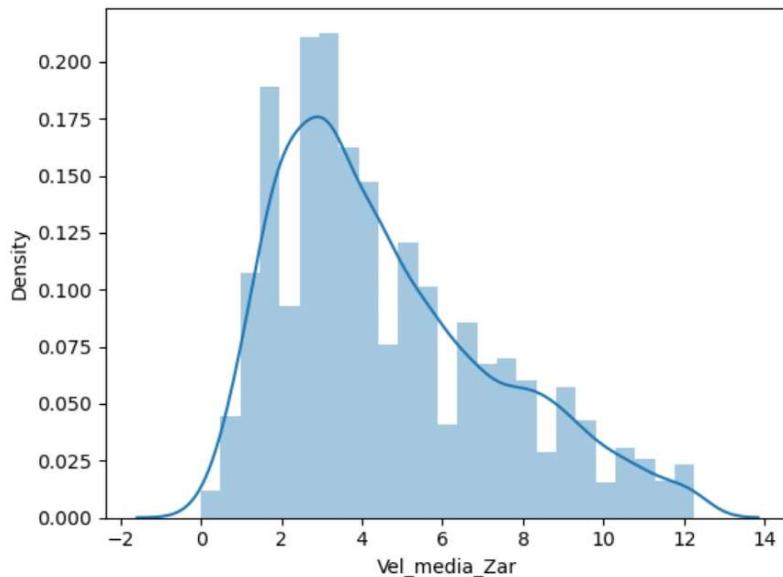
```
print ('El valor Maximo para "Vel_media_Zar" en "train" es '+_
+str(df_def_out.UR))
```

El valor Minimo para "Vel_media_Zar" en "train" es -3.35
El valor Maximo para "Vel_media_Zar" en "train" es 12.25

```
[108]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Vel_media_Zar'] < df_def_out.LR, 'Vel_media_Zar'] =_
+df_def_out.LR
df_def_out.loc[df_def_out['Vel_media_Zar'] > df_def_out.UR, 'Vel_media_Zar'] =_
+df_def_out.UR
```

```
[109]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Zar'])
```

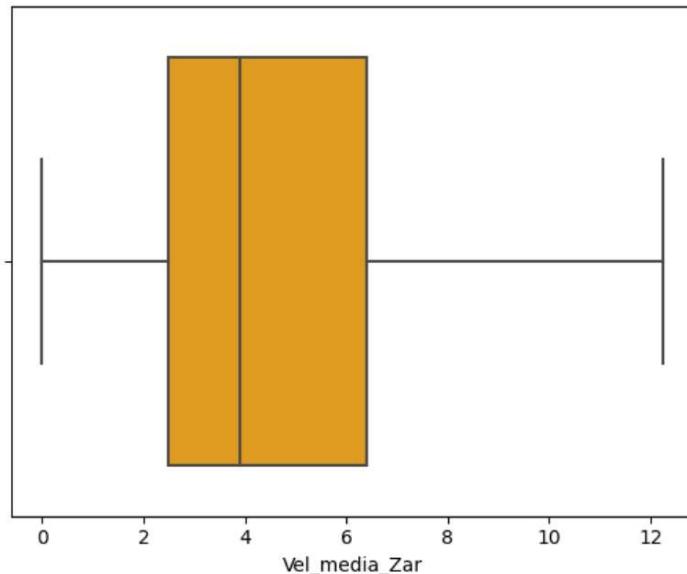
```
[109]: <AxesSubplot:xlabel='Vel_media_Zar', ylabel='Density'>
```



```
[110]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Zar'], color = "orange")
```

```
[110]: <AxesSubplot:xlabel='Vel_media_Zar'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



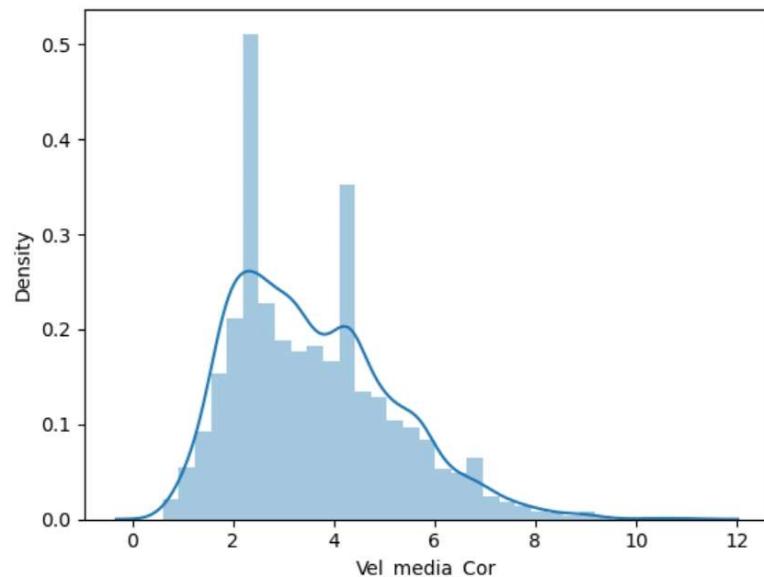
0.1.18 Outliers en variable “Vel_media_Cor”

```
[111]: # Obtenemos la distribucion de la variable "Vel_media_Cor" en "train" y en "test"
      sns.distplot(df_def_out['Vel_media_Cor'])
```



```
[111]: <AxesSubplot:xlabel='Vel_media_Cor', ylabel='Density'>
```

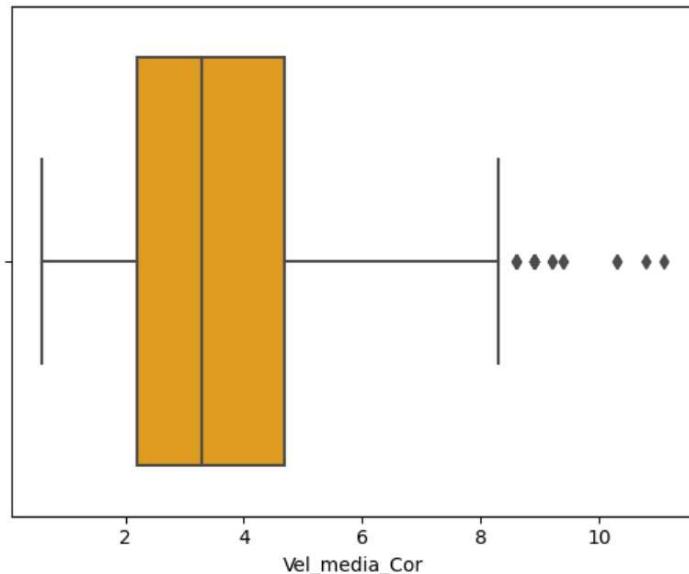
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[112]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Vel_media_Cor'], color = "orange")
```

```
[112]: <AxesSubplot:xlabel='Vel_media_Cor'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[113]: # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Cor.describe())
```

```
count    3652.000000
mean      3.610405
std       1.594199
min       0.600000
25%      2.200000
50%      3.300000
75%      4.700000
max      11.100000
Name: Vel_media_Cor, dtype: float64
```

```
[114]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Cor'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Cor'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Cor" en "train" es '+_
      str(df_def_out_LR))
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD

EN EL MERCADO ESPAÑOL

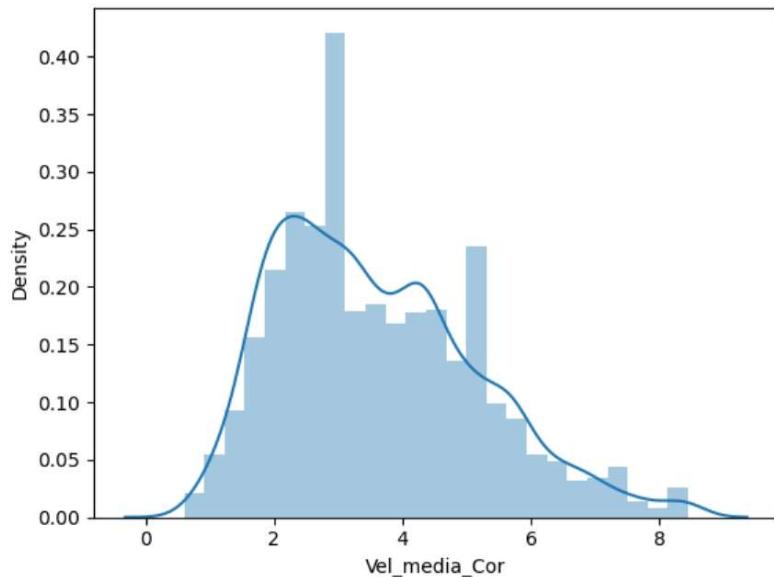
```
print ('El valor Maximo para "Vel_media_Cor" en "train" es '+_
+str(df_def_out.UR))
```

El valor Minimo para "Vel_media_Cor" en "train" es -1.55
El valor Maximo para "Vel_media_Cor" en "train" es 8.45

```
[115]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Vel_media_Cor'] < df_def_out.LR, 'Vel_media_Cor'] =_
+df_def_out.LR
df_def_out.loc[df_def_out['Vel_media_Cor'] > df_def_out.UR, 'Vel_media_Cor'] =_
+df_def_out.UR
```

```
[116]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Cor'])
```

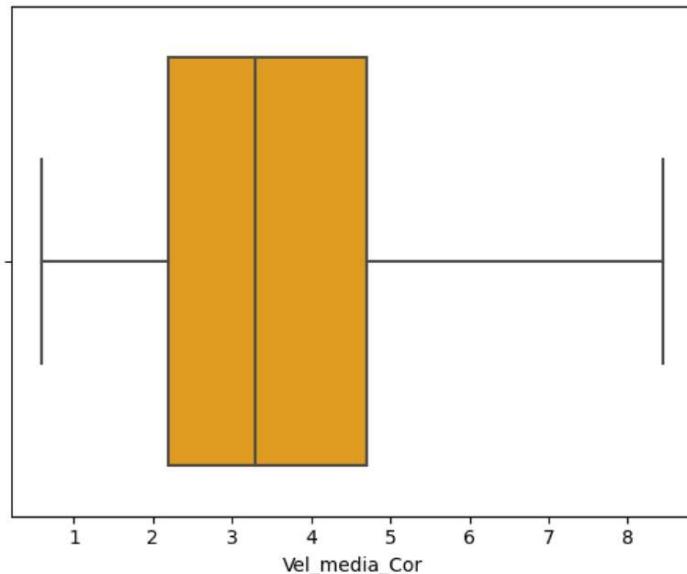
```
[116]: <AxesSubplot:xlabel='Vel_media_Cor', ylabel='Density'>
```



```
[117]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Cor'], color = "orange")
```

```
[117]: <AxesSubplot:xlabel='Vel_media_Cor'>
```

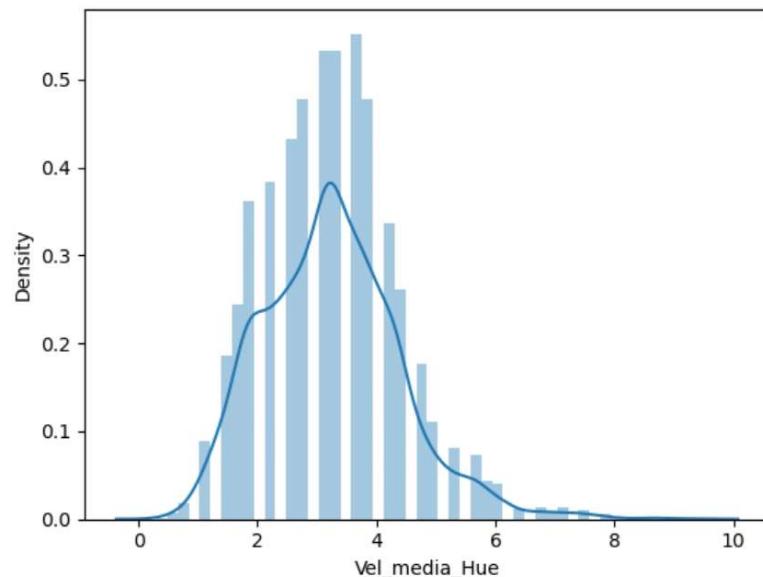
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.19 Outliers en variable “Vel_media_Hue”

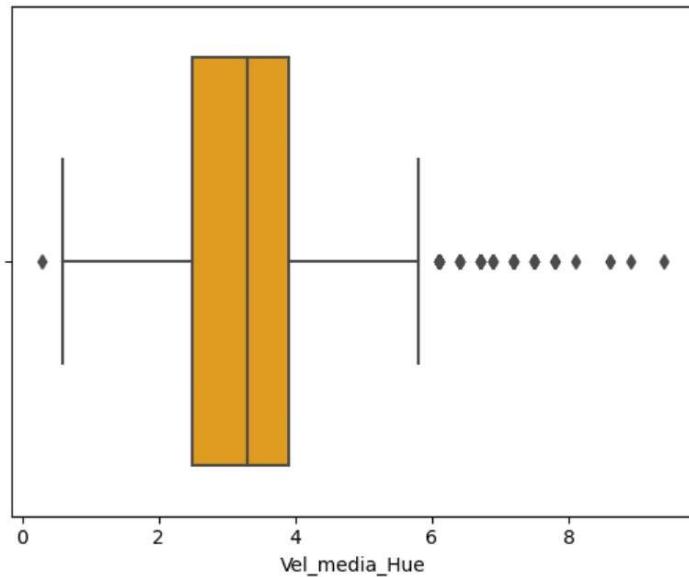
```
[118]: # Obtenemos la distribucion de la variable "XXXXXX" en "train" y en "test"  
sns.distplot(df_def_out['Vel_media_Hue'])  
  
[118]: <AxesSubplot:xlabel='Vel_media_Hue', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[119]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Vel_media_Hue'], color = "orange")
```

```
[119]: <AxesSubplot:xlabel='Vel_media_Hue'>
```



```
[120]: # Obtenemos las estadísticas de los datos
print(df_def_out.Vel_media_Hue.describe())
```

```
count    3652.000000
mean     3.243018
std      1.159843
min      0.300000
25%     2.500000
50%     3.300000
75%     3.900000
max     9.400000
Name: Vel_media_Hue, dtype: float64
```

```
[121]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Vel_media_Hue'].quantile(0.25)
df_def_out_Q3 = df_def_out['Vel_media_Hue'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Vel_media_Hue" en "train" es '+_
      str(df_def_out_LR))
```

```

print ('El valor Maximo para "Vel_media_Hue" en "train" es '+_
      str(df_def_out.UR))

```

El valor Minimo para "Vel_media_Hue" en "train" es 0.4
 El valor Maximo para "Vel_media_Hue" en "train" es 6.0

```

[122]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Vel_media_Hue'] < df_def_out.LR, 'Vel_media_Hue'] =_
      df_def_out.LR
df_def_out.loc[df_def_out['Vel_media_Hue'] > df_def_out.UR, 'Vel_media_Hue'] =_
      df_def_out.UR

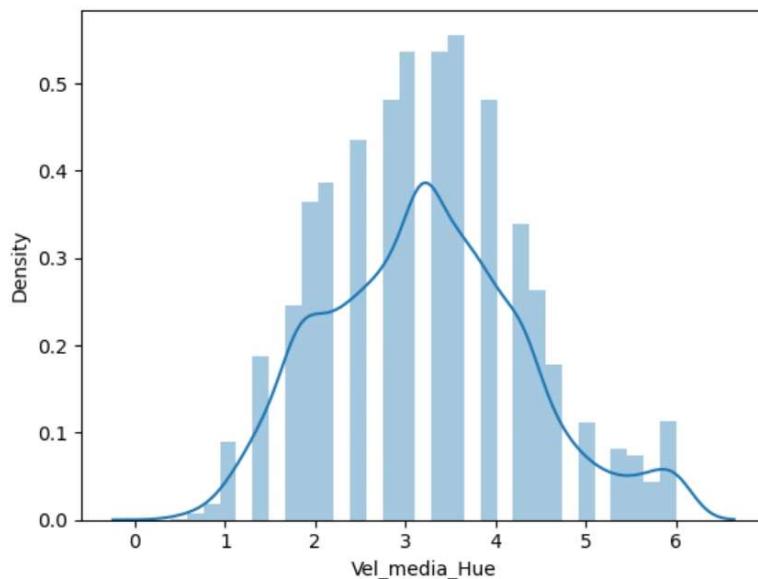
```

```

[123]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Vel_media_Hue'])

```

[123]: <AxesSubplot:xlabel='Vel_media_Hue', ylabel='Density'>



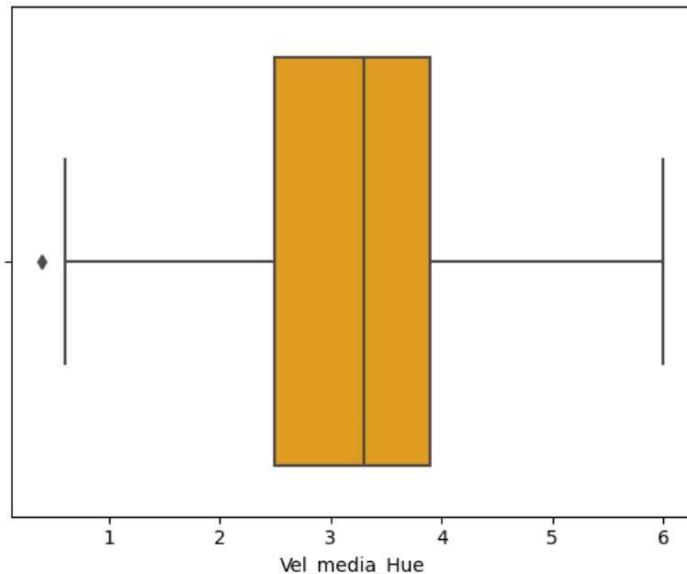
```

[124]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Vel_media_Hue'], color = "orange")

```

[124]: <AxesSubplot:xlabel='Vel_media_Hue'>

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

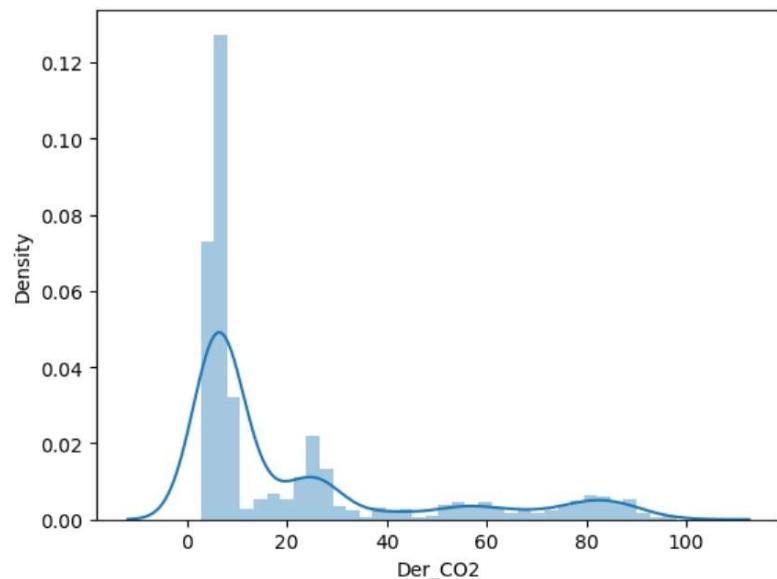


0.1.20 Outliers en variable “Der_CO2”

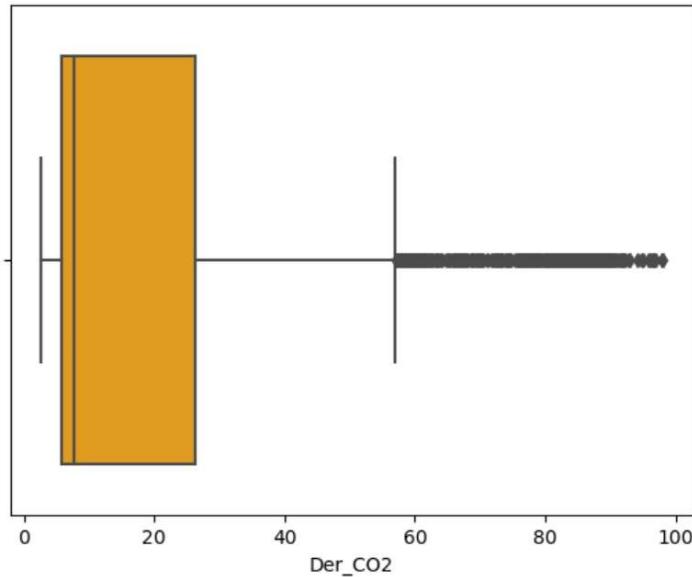
```
[125]: # Obtenemos la distribucion de la variable "Der_CO2" en "train" y en "test"
sns.distplot(df_def_out['Der_CO2'])

[125]: <AxesSubplot:xlabel='Der_CO2', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[126]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Der_CO2'], color = "orange")  
  
[126]: <AxesSubplot:xlabel='Der_CO2'>
```



```
[127]: # Obtenemos las estadísticas de los datos
print(df_def_out.Der_CO2.describe())
```

```
count    3652.000000
mean     21.554847
std      25.116090
min      2.700000
25%      5.720000
50%      7.745000
75%     26.282500
max     98.010000
Name: Der_CO2, dtype: float64
```

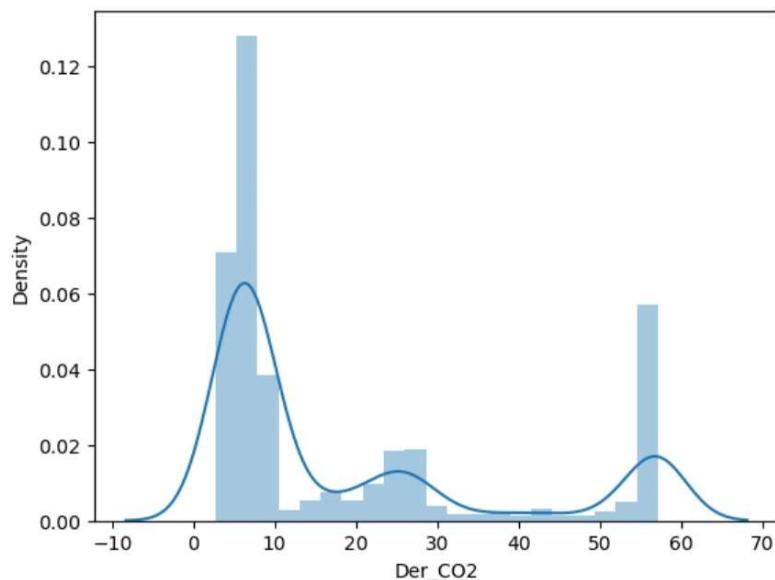
```
[128]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Der_CO2'].quantile(0.25)
df_def_out_Q3 = df_def_out['Der_CO2'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Der_CO2" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Der_CO2" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Der_CO2" en "train" es -25.1238
El valor Maximo para "Der_CO2" en "train" es 57.1262

```
[129]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Der_CO2'] < df_def_out_LR, 'Der_CO2'] = df_def_out_LR
df_def_out.loc[df_def_out['Der_CO2'] > df_def_out_UR, 'Der_CO2'] = df_def_out_UR
```

```
[130]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Der_CO2'])
```

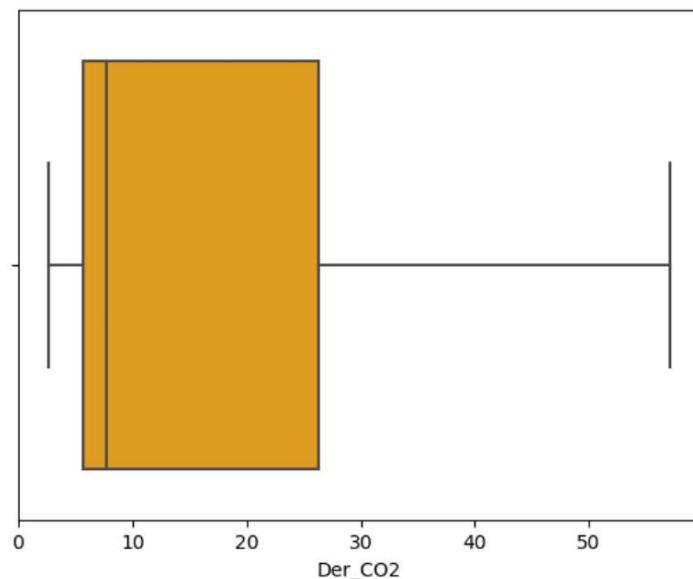
```
[130]: <AxesSubplot:xlabel='Der_CO2', ylabel='Density'>
```



```
[131]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Der_CO2'], color = "orange")
```

```
[131]: <AxesSubplot:xlabel='Der_CO2'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL

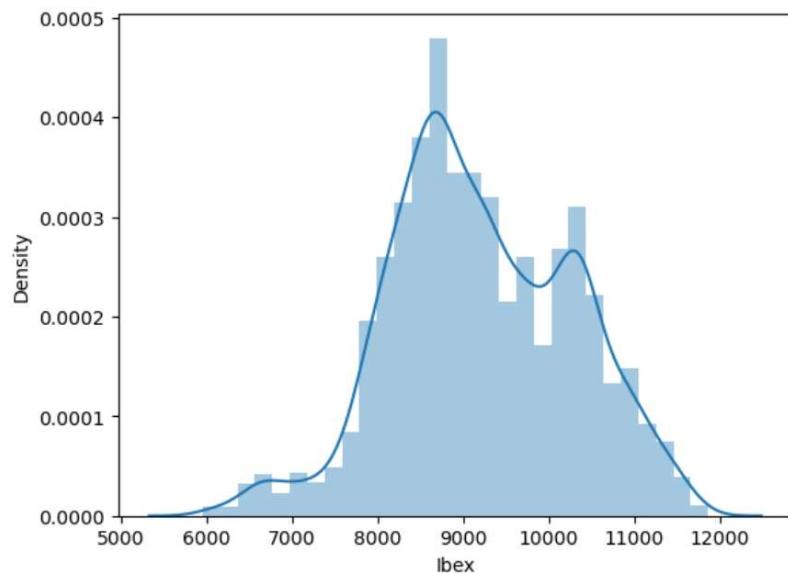


0.1.21 Outliers en variable “Ibex”

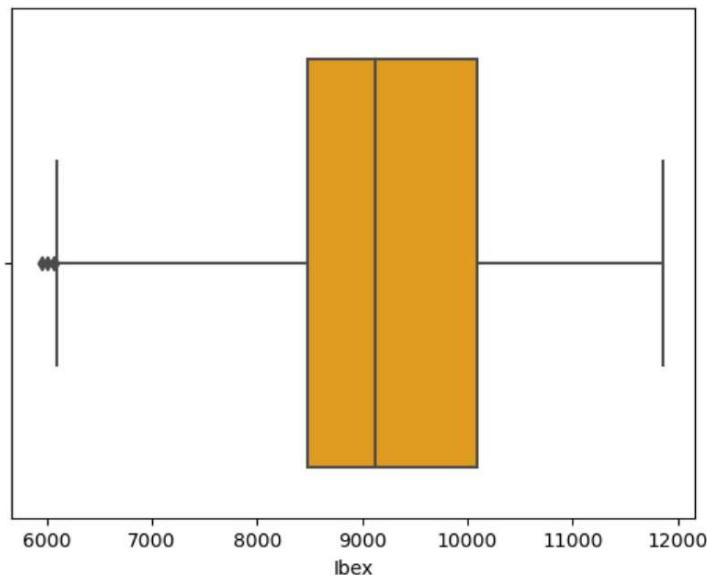
```
[132]: # Obtenemos la distribucion de la variable "Ibex" en "train" y en "test"  
sns.distplot(df_def_out['Ibex'])
```

```
[132]: <AxesSubplot:xlabel='Ibex', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[133]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Ibex'], color = "orange")  
  
[133]: <AxesSubplot:xlabel='Ibex'>
```



```
[134]: # Obtenemos las estadísticas de los datos
print(df_def_out.Ibex.describe())
```

```
count      3652.000000
mean      9215.483078
std       1071.750925
min      5956.300000
25%     8486.300000
50%     9124.650000
75%    10092.750000
max     11866.400000
Name: Ibex, dtype: float64
```

```
[135]: # Realizamos el filtrado intercuartílico en los datos "train"
df_def_out_Q1 = df_def_out['Ibex'].quantile(0.25)
df_def_out_Q3 = df_def_out['Ibex'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Ibex" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Ibex" en "train" es '+ str(df_def_out_UR))
```

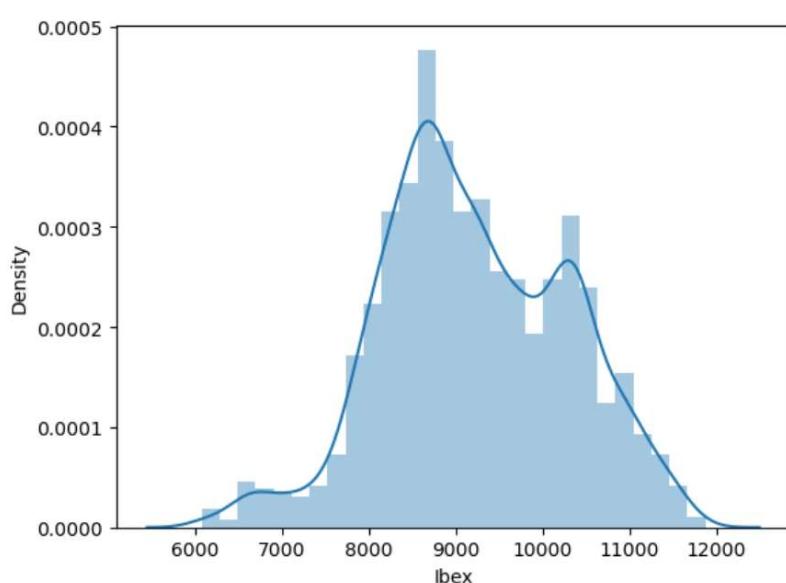
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD EN EL MERCADO ESPAÑOL

El valor Minimo para "Ibex" en "train" es 6076.625
El valor Maximo para "Ibex" en "train" es 12502.425

```
[136]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Ibex'] < df_def_out_LR, 'Ibex'] = df_def_out_LR
df_def_out.loc[df_def_out['Ibex'] > df_def_out.UR, 'Ibex'] = df_def_out.UR

[137]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Ibex'])

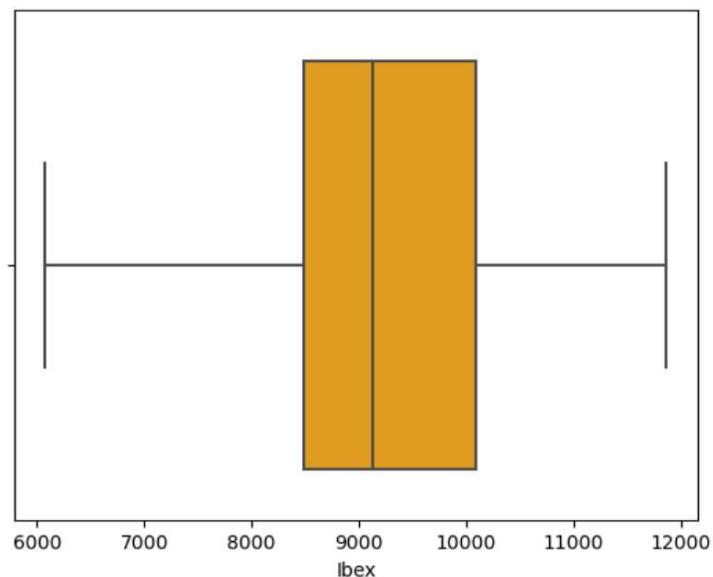
[137]: <AxesSubplot:xlabel='Ibex', ylabel='Density'>
```



```
[138]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Ibex'], color = "orange")
```

```
[138]: <AxesSubplot:xlabel='Ibex'>
```

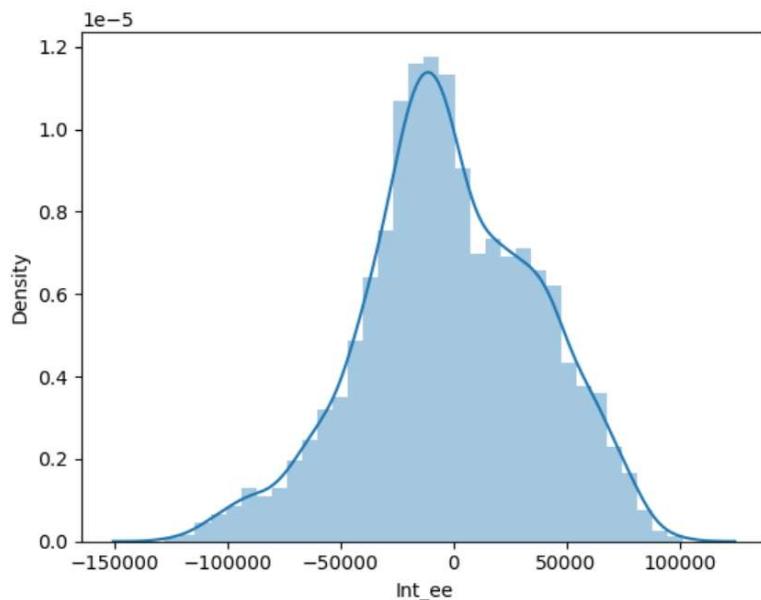
MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.22 Outliers en variable “Int_ee”

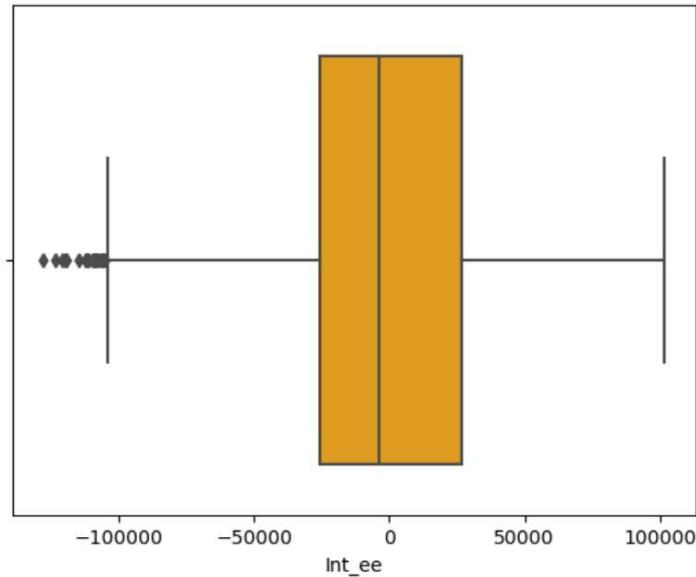
```
[139]: # Obtenemos la distribucion de la variable "Int_ee" en "train" y en "test"  
sns.distplot(df_def_out['Int_ee'])  
  
[139]: <AxesSubplot:xlabel='Int_ee', ylabel='Density'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



```
[140]: # Dibujamos el boxplot para los datos "train"  
sns.boxplot(x = df_def_out['Int_ee'], color = "orange")
```

```
[140]: <AxesSubplot:xlabel='Int_ee'>
```



```
[141]: # Obtenemos las estadisticas de los datos
print(df_def_out.Int_ee.describe())
```

```
count      3652.000000
mean     -1541.333354
std      39195.460710
min     -127714.695000
25%    -25758.695750
50%    -3786.574500
75%     26981.247000
max     101779.882000
Name: Int_ee, dtype: float64
```

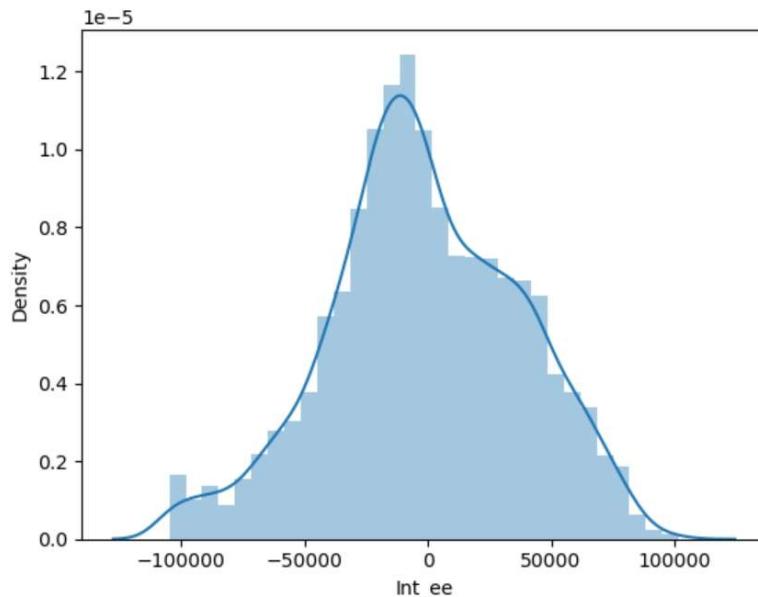
```
[142]: # Realizamos el filtrado intercuartilico en los datos "train"
df_def_out_Q1 = df_def_out['Int_ee'].quantile(0.25)
df_def_out_Q3 = df_def_out['Int_ee'].quantile(0.75)
df_def_out_IQR = df_def_out_Q3 - df_def_out_Q1
df_def_out_LR = round(df_def_out_Q1 -(1.5 * df_def_out_IQR), 4)
df_def_out_UR = round(df_def_out_Q3 +(1.5 * df_def_out_IQR), 4)
print ('El valor Minimo para "Int_ee" en "train" es '+ str(df_def_out_LR))
print ('El valor Maximo para "Int_ee" en "train" es '+ str(df_def_out_UR))
```

El valor Minimo para "Int_ee" en "train" es -104868.6099
El valor Maximo para "Int_ee" en "train" es 106091.1611

```
[143]: # Sustituimos los outliers por los valores maximos y minimos
df_def_out.loc[df_def_out['Int_ee'] < df_def_out_LR, 'Int_ee'] = df_def_out_LR
df_def_out.loc[df_def_out['Int_ee'] > df_def_out.UR, 'Int_ee'] = df_def_out.UR
```

```
[144]: # Volvemos a graficar la variable para comprobar el cambio
sns.distplot(df_def_out['Int_ee'])
```

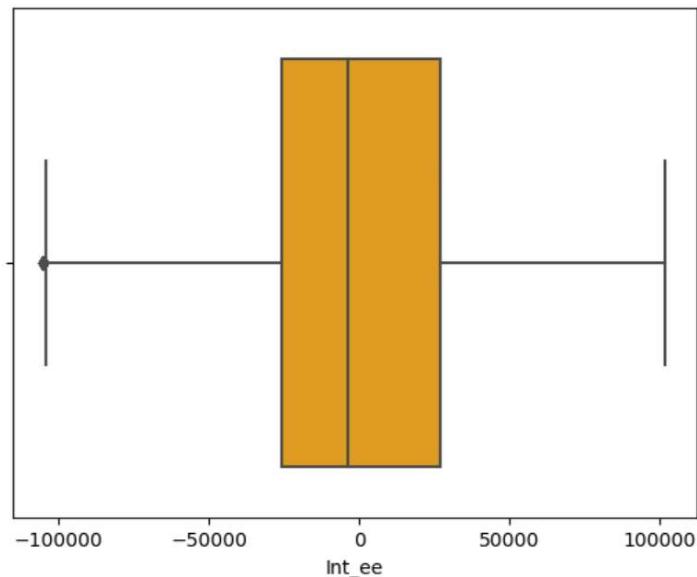
```
[144]: <AxesSubplot:xlabel='Int_ee', ylabel='Density'>
```



```
[145]: # Volvemos a obtener el boxplot de "train" una vez tratados los outliers
sns.boxplot(x = df_def_out['Int_ee'], color = "orange")
```

```
[145]: <AxesSubplot:xlabel='Int_ee'>
```

MODELOS PARA LA PREDICCIÓN DEL PRECIO DE LA ELECTRICIDAD
EN EL MERCADO ESPAÑOL



0.1.23 Creamos un csv con los datos del dataframe obtenidos “df_def_out” para utilizarlos en los modelos

```
[146]: # Para descargarse los csv descomentar el codido
df_def_out.to_csv('df_def_outliers_2.csv', header=True, index=False)
```

```
[ ]:
```

```
[ ]:
```