Pràctica 2: Com realitzar la neteja i l'anàlisi de dades?

Preprocessing document

Esther Manzano i Albert Queraltó

8 de enero 2023

LLibreries

Primer de tot, carregarem algunes llibreries que podríem necessitar per crear els diferents models, analitzar-los i generar gràfics.

1. Descripció del fitxer

Aquest document conté tota la part de preprocessat prèvia a la integració i selecció de les dades. En particular, s'han utilitzat les dades dels preus de l'electricitat i la producció d'energies renovables obtingudes en la primera pràctica mitjançant webscrapping de la web https://www.esios.ree.es/es/ (*dataset ESIOS*) i les dades meteorològiques de diferents estacions repartides per les províncies d'Espanya i extretes de la web https://opendata.aemet.es/ (*dataset AEMET*) utilitzant l'API proporcionada.

2. Preprocessat dels datasets

2.1. Càrrega, preprocessat i selecció del dataset ESIOS

2.1.1. Càrrega de les dades i anàlisi a alt nivell

Carreguem les dades del dataset ESIOS que hem creat a la pràctica 1 en un dataframe. Aquestes dades són les obtingudes a partir de la web https://www.esios.ree.es/es/ [6-7] i contenen la informació de l'evolució del preu de l'energia elèctrica a Espanya i la producció d'energies renovables en el període de temps comprès entre l'1 de Novembre del 2020 al 31 d'Octubre del 2022.

```
# Carreguem els datasets en un dataframe
esios_dataset <- read.csv('data/esios_dataset.csv', header = TRUE, sep=";")
```

Renombrem les columnes del dataset per a que siguin més fàcils de tractar.

```
# Renombrem el nom de les columnes
colnames(esios_dataset) <- c("date", "hour", "avg_total_price_eur_mwh",
                             "avg_price_free_market_eur_mwh",
                             "avg_price_ref_market_eur_mwh", "energy_total_mwh",
                             "energy_free_market_mwh", "energy_ref_market_mwh",
                             "free_market_share_perc", "ref_market_share_perc",
                             "renewable_generation_perc", "energy_source",
                             "renewable_generation_mw")

# Taula resum que representa les principals característiques de les diferents
# variables
str(esios_dataset)
```

```
## 'data.frame':    105120 obs. of  13 variables:
##  $ date                          : chr  "2022-01-12" "2022-01-12" "2022-01-12" "2022-01-12" ...
##  $ hour                          : chr  "00:00" "00:00" "00:00" "00:00" ...
##  $ avg_total_price_eur_mwh       : chr  "199,18" "199,18" "199,18" "199,18" ...
##  $ avg_price_free_market_eur_mwh: chr  "199,18" "199,18" "199,18" "199,18" ...
##  $ avg_price_ref_market_eur_mwh : chr  "199,19" "199,19" "199,19" "199,19" ...
##  $ energy_total_mwh              : chr  "26.714,2" "26.714,2" "26.714,2" "26.714,2" ...
```

```
##  $ energy_free_market_mwh       : chr  "23.764,1" "23.764,1" "23.764,1" "23.764,1" ...
##  $ energy_ref_market_mwh        : chr  "2.950,1" "2.950,1" "2.950,1" "2.950,1" ...
##  $ free_market_share_perc       : chr  "89,0" "89,0" "89,0" "89,0" ...
##  $ ref_market_share_perc        : chr  "11,0" "11,0" "11,0" "11,0" ...
##  $ renewable_generation_perc    : chr  "68,6\n%" "68,6\n%" "68,6\n%" "68,6\n%" ...
##  $ energy_source                : chr  "renewable generation (MW)" "wind generation (MW)" "water generation (MW)" "solar generation (MW)" ...
##  $ renewable_generation_mw      : chr  "20.548MW" "11.926MW" "998MW" "27MW" ...
```

El dataset conté dades estructurades en 13 columnes i 105120 files. També es pot veure com totes les columnes són de tipus *character*, malgrat que la majoria d'elles haurien de ser de tipus numèric per a portar a terme l'anàlisi pràctic correctament. Caldrà doncs fer un preprocessat previ per poder tractar-les com a numèriques. A més, també es preprocessaran les columnes *date* i *hour* per a poder tractar-les com a dates, així com també la columna *energy_source*.

2.1.2. Processat d'*energy_source*

Tractem els valors de la columna *energy_source* per a que tinguin un nom adequat. És a dir, els substituïm pels següents noms:

- "renewable generation (MW)" per "total".
- "wind generation (MW)" per "wind".
- "water generation (MW)" per "hydroelectric".
- "solar generation (MW)" per "solar".
- "nuclear generation (MW)" per "nuclear".
- "thermorenewable generation (MW)" per "thermorenewable".

```r
# Substituïm els valors de la columna "energy_source"
esios_dataset$energy_source <- gsub("thermorenewable generation (MW)",
                                    "thermorenewable",
                                    esios_dataset$energy_source, fixed = TRUE)
esios_dataset$energy_source <- gsub("renewable generation (MW)",
                                    "total",
                                    esios_dataset$energy_source, fixed = TRUE)
esios_dataset$energy_source <- gsub("wind generation (MW)",
                                    "wind",
                                    esios_dataset$energy_source, fixed = TRUE)
esios_dataset$energy_source <- gsub("water generation (MW)",
                                    "hydroelectric",
                                    esios_dataset$energy_source, fixed = TRUE)
esios_dataset$energy_source <- gsub("solar generation (MW)",
                                    "solar",
                                    esios_dataset$energy_source, fixed = TRUE)
esios_dataset$energy_source <- gsub("nuclear generation (MW)",
                                    "nuclear",
                                    esios_dataset$energy_source, fixed = TRUE)

# Mostrem els valors únics per "energy_source"
unique(esios_dataset$energy_source)
```

```
## [1] "total"           "wind"            "hydroelectric"   "solar"
## [5] "nuclear"         "thermorenewable"
```

2.1.3. Processat de *renewable_generation_perc* i *renewable_generation_mw*

Les columnes *renewable_generation_perc* i *renewable_generation_mw* contenen les unitats (% i MW) juntament amb el valor numèric i caldrà eliminar aquests strings per a poder tractar-les com a numèriques. També eliminarem el salt de línia de la columna *renewable_generation_perc*.

```r
# Eliminem les unitats de les columnes "renewable_generation_perc" i "renewable_generation_mw"
esios_dataset$renewable_generation_perc <- gsub("%", "",
                                    esios_dataset$renewable_generation_perc)
esios_dataset$renewable_generation_mw <- gsub("MW", "",
                                    esios_dataset$renewable_generation_mw)

# Eliminem els "\n" de la columna "renewable_generation_perc"
esios_dataset$renewable_generation_perc <- gsub("\n", "",
                                    esios_dataset$renewable_generation_perc)

# Eliminem els espais en blanc de totes les columnes
esios_dataset <- esios_dataset %>%
  mutate_at(vars(date, hour, energy_source), funs(gsub(" ", "", .)))
```

2.1.4 Processat dels separadors decimals i de milers, i transformació a columnes numèriques

Eliminem els separadors de milers de les columnes numèriques i substituïm els separadors decimals per un punt.

```
# Eliminem els separadors de milers de les columnes numèriques
esios_dataset <- esios_dataset %>%
  mutate_at(vars(energy_total_mwh, energy_ref_market_mwh,
  energy_free_market_mwh, renewable_generation_mw), funs(gsub(".", "", .,
                                                              fixed = TRUE)))

# Eliminem els separadors de milers de les columnes numèriques
esios_dataset <- esios_dataset %>%
  mutate_at(vars(avg_total_price_eur_mwh, avg_price_ref_market_eur_mwh,
  avg_price_free_market_eur_mwh, energy_total_mwh, energy_ref_market_mwh,
  energy_free_market_mwh, free_market_share_perc, ref_market_share_perc,
  renewable_generation_perc), funs(gsub(",", ".", ., fixed = TRUE)))
```

Finalment, transformem les columnes que haurien de ser numèriques a aquest tipus.

```
# Transformem les columnes numèriques a tipus numèric
esios_dataset <- esios_dataset %>%
  mutate_at(vars(avg_total_price_eur_mwh, avg_price_ref_market_eur_mwh,
  avg_price_free_market_eur_mwh, energy_total_mwh, energy_ref_market_mwh,
  energy_free_market_mwh, free_market_share_perc, ref_market_share_perc,
  renewable_generation_perc, renewable_generation_mw), funs(as.numeric(.)))
```

Es pot veure com durant la transformació hi ha alguns valors nuls que haurem de tractar posteriorment. Finalment, comprovem el tipus de dades de cada columna del dataset.

```
# Verifiquem el tipus de dades de cada columna del dataset
sapply(esios_dataset, class)
```

```
##                         date                         hour
##                  "character"                  "character"
##       avg_total_price_eur_mwh avg_price_free_market_eur_mwh
##                    "numeric"                    "numeric"
##  avg_price_ref_market_eur_mwh              energy_total_mwh
##                    "numeric"                    "numeric"
##        energy_free_market_mwh         energy_ref_market_mwh
##                    "numeric"                    "numeric"
##        free_market_share_perc         ref_market_share_perc
##                    "numeric"                    "numeric"
##     renewable_generation_perc                 energy_source
##                    "numeric"                  "character"
##       renewable_generation_mw
##                    "numeric"
```

```
# Mostrem les primeres files del dataset
head(esios_dataset)
```

```
##         date  hour avg_total_price_eur_mwh avg_price_free_market_eur_mwh
## 1 2022-01-12 00:00                 199.18                         199.18
## 2 2022-01-12 00:00                 199.18                         199.18
## 3 2022-01-12 00:00                 199.18                         199.18
## 4 2022-01-12 00:00                 199.18                         199.18
## 5 2022-01-12 00:00                 199.18                         199.18
## 6 2022-01-12 00:00                 199.18                         199.18
##   avg_price_ref_market_eur_mwh energy_total_mwh energy_free_market_mwh
## 1                       199.19          26714.2                23764.1
## 2                       199.19          26714.2                23764.1
## 3                       199.19          26714.2                23764.1
## 4                       199.19          26714.2                23764.1
## 5                       199.19          26714.2                23764.1
## 6                       199.19          26714.2                23764.1
##   energy_ref_market_mwh free_market_share_perc ref_market_share_perc
## 1                2950.1                     89                    11
## 2                2950.1                     89                    11
## 3                2950.1                     89                    11
## 4                2950.1                     89                    11
## 5                2950.1                     89                    11
## 6                2950.1                     89                    11
##   renewable_generation_perc   energy_source renewable_generation_mw
## 1                      68.6           total                   20548
## 2                      68.6            wind                   11926
## 3                      68.6   hydroelectric                     998
## 4                      68.6           solar                      27
## 5                      68.6         nuclear                    6994
## 6                      68.6 thermorenewable                     603
```

2.1.5. Transformar *date* i *energy_source* al format correcte

Transformem la columna *date* al tipus data i la columna *energy_source* al tipus factor.

```
# Transformem la columna date al format correcte
esios_dataset <- esios_dataset %>%
  mutate(date = as.Date(date, format = "%Y-%m-%d"))

# Transformem les columnes energy_source a factor
esios_dataset <- esios_dataset %>% mutate(energy_source = as.factor(energy_source))

# Verifiquem el tipus de dades de cada columna del dataset
sapply(esios_dataset, class)
```

```
##                         date                         hour
##                       "Date"                  "character"
##       avg_total_price_eur_mwh avg_price_free_market_eur_mwh
##                     "numeric"                    "numeric"
##  avg_price_ref_market_eur_mwh              energy_total_mwh
##                     "numeric"                    "numeric"
##        energy_free_market_mwh         energy_ref_market_mwh
##                     "numeric"                    "numeric"
##        free_market_share_perc         ref_market_share_perc
##                     "numeric"                    "numeric"
##     renewable_generation_perc                 energy_source
##                     "numeric"                     "factor"
##       renewable_generation_mw
##                     "numeric"
```

2.1.6. Tractament de valors nuls

A l'apartat 3.1, se'ns demana identificar i gestionar els valors nuls. Tanmateix, el preprocessat que realitzarem a l'apartat següent requereix haver tractat abans els valors nuls, per poder fer un càlcul correcte del valor promig. En primer lloc, comprovem si hi ha valors nuls:

```
# Comprovem si hi ha valors nuls a les dades per columnes
colSums(is.na(esios_dataset))
```

```
##                         date                         hour
##                            0                            0
##       avg_total_price_eur_mwh avg_price_free_market_eur_mwh
##                           18                           18
##  avg_price_ref_market_eur_mwh              energy_total_mwh
##                           18                           18
##        energy_free_market_mwh         energy_ref_market_mwh
##                           18                           18
##        free_market_share_perc         ref_market_share_perc
##                           18                          162
##     renewable_generation_perc                 energy_source
##                           18                            0
##       renewable_generation_mw
##                           18
```

Es pot veure que algunes columnes contenen valors nuls tot i que aquests representen un percentatge molt reduït si tenim en compte les dimensions del dataset (105120 files). Existeixen diversos mètodes per tractar-los, entre ells la imputació amb el valor promig. Tot i que aquest mètode podria ser vàlid en aquest cas, en treballar amb dades que presenten una forta dependència temporal es pot utilitzar la interpolació dels valors faltants:

```
# Interpolació dels valors nuls
esios_dataset <- esios_dataset %>%
  group_by(energy_source) %>%
  mutate_at(vars(avg_total_price_eur_mwh, energy_total_mwh,
  ref_market_share_perc, avg_price_ref_market_eur_mwh, avg_price_free_market_eur_mwh,
  energy_ref_market_mwh, energy_free_market_mwh, free_market_share_perc,
  renewable_generation_perc, renewable_generation_mw), funs(na.approx(.)))

# Comprovem si hi ha valors nuls a les dades per columnes
colSums(is.na(esios_dataset))
```

```
##                         date                         hour
##                            0                            0
##       avg_total_price_eur_mwh avg_price_free_market_eur_mwh
##                            0                            0
##  avg_price_ref_market_eur_mwh              energy_total_mwh
##                            0                            0
##        energy_free_market_mwh         energy_ref_market_mwh
```

```
##                            0                            0
##       free_market_share_perc        ref_market_share_perc
##                            0                            0
##    renewable_generation_perc                energy_source
##                            0                            0
##      renewable_generation_mw
##                            0
```

2.1.7. Calculem el promig de totes les columnes numèriques en funció de la

data i eliminem la columna *hora*

Calculem el promig de totes les columnes numèriques del dataset tenint en compte els valors de les columnes *date* i *energy_source*. El promig es calcula perquè el dataset AEMET que tractarem posteriorment té una granularitat menor, és a dir, les dades estan agrupades per dia i no per dia i hora.

```
# Calculem el promig de totes les columnes numèriques del dataset
esios_dataset <- esios_dataset %>%
  group_by(date, energy_source) %>%
  summarise_all(funs(mean(., na.rm = FALSE)))

# Eliminem la columna hora del dataset
esios_dataset <- esios_dataset %>% select(-hour)

# Mostrem les primeres files del dataset
head(esios_dataset)
```

```
## # A tibble: 6 x 12
## # Groups:   date [1]
##   date       energy_so~1 avg_t~2 avg_p~3 avg_p~4 energ~5 energ~6 energ~7 free_~8
##   <date>     <fct>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 2020-11-01 hydroelect~    31.6    31.3    34.1  22639.   19670   2969.    87.1
## 2 2020-11-01 nuclear        31.6    31.3    34.1  22639.   19670   2969.    87.1
## 3 2020-11-01 solar          31.6    31.3    34.1  22639.   19670   2969.    87.1
## 4 2020-11-01 thermorene~    31.6    31.3    34.1  22639.   19670   2969.    87.1
## 5 2020-11-01 total          31.6    31.3    34.1  22639.   19670   2969.    87.1
## 6 2020-11-01 wind           31.6    31.3    34.1  22639.   19670   2969.    87.1
## # ... with 3 more variables: ref_market_share_perc <dbl>,
## #   renewable_generation_perc <dbl>, renewable_generation_mw <dbl>, and
## #   abbreviated variable names 1: energy_source, 2: avg_total_price_eur_mwh,
## #   3: avg_price_free_market_eur_mwh, 4: avg_price_ref_market_eur_mwh,
## #   5: energy_total_mwh, 6: energy_free_market_mwh, 7: energy_ref_market_mwh,
## #   8: free_market_share_perc
```

2.1.8. Calcular percentatge d'energia renovable respecte el total

La columna *renewable_generation_perc* conté el percentatge d'energia renovable respecte el total. Per això, tornarem a recalcular-ne els valors tenint en compte les categories de la columna *energy_source* respecte les columnes *renewable_generation_mw* i *energy_total_mwh*.

```
# Calculem el percentatge d'energia renovable respecte el total
esios_dataset <- esios_dataset %>%
  mutate(renewable_generation_perc = renewable_generation_mw / energy_total_mwh * 100)

# Mostrem les primeres files del dataset
summary(esios_dataset$renewable_generation_perc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -4.023   5.652  16.628  23.585  26.898 104.715
```

2.1.9. Crear intervals per *renewable_generation_perc*

La generació d'energia renovable té valors en tot el rang de percentatges, per tant, és una variable que es pot discretitzar en intervals fent servir el *binning*. Això ens permetrà reduir la quantitat de dades i simplificarà algunes de les anàlisis posteriors.

```
# Mostrar resum de la variable renewable_generation_perc
summary(esios_dataset[, "renewable_generation_perc"])
```

```
##  renewable_generation_perc
##  Min.   : -4.023
##  1st Qu.:  5.652
##  Median : 16.628
##  Mean   : 23.585
```

```
##  3rd Qu.: 26.898
##  Max.   :104.715
```

Es consideraràn intervals amb percentatges negatius perquè hi ha valors de generació d'energia renovable negatius probablement deguts al consum energètic de la seva operació en dies en què no hi havia producció renovable. També es pot observar com hi ha algun valor amb percentatges de generació renovable superior. En concret és únicament un valor i, per tant, s'imputarà el valor 100 a la columna *renewable_generation_perc*.

```
# Imputem el valor 100 a la columna renewable_generation_perc
esios_dataset <- esios_dataset %>%
  mutate(renewable_generation_perc = ifelse(renewable_generation_perc > 100,
                                            100, renewable_generation_perc))
```

Tot seguit creem els intervals:

```
# Creem intervals per renewable_generation_perc
esios_dataset <- esios_dataset %>%
  mutate(renewable_generation_perc_bin = cut(renewable_generation_perc,
                                     breaks = c(-Inf, 0, 35, 75, 100),
                                     labels = c("Below 0", "0-35", "36-75", "76-100"),
                                     include.lowest = TRUE, right = TRUE))
```

Comprovem valors nuls de la variable *renewable_generation_perc*:

```
# Comprovem valors nuls per columnes
sapply(esios_dataset, function(x) sum(is.na(x)))
```

```
##                          date                 energy_source
##                             0                             0
##       avg_total_price_eur_mwh avg_price_free_market_eur_mwh
##                             0                             0
##  avg_price_ref_market_eur_mwh              energy_total_mwh
##                             0                             0
##        energy_free_market_mwh         energy_ref_market_mwh
##                             0                             0
##        free_market_share_perc         ref_market_share_perc
##                             0                             0
##     renewable_generation_perc       renewable_generation_mw
##                             0                             0
## renewable_generation_perc_bin
##                             0
```

2.1.10. Guardem el dataset ESIOS preprocessat

Guardem el dataset ESIOS preprocessat en un fitxer CSV:

```
# Guardem el dataset ESIOS preprocessat en un fitxer CSV
write.csv(esios_dataset, file = "data/esios_dataset_preprocessed.csv", row.names = FALSE)
```

2.2. Càrrega, preprocessat i selecció del dataset AEMET

2.2.1. Càrrega de les dades i anàlisi a alt nivell

Carreguem les dades del dataset AEMET en un dataframe. Aquestes dades s'han obtingut de la web https://opendata.aemet.es/ utilitzant l'API proporcionada. La informació continguda al dataset va des de l'1 d'Octubre del 2020 al 30 de Novembre del 2022. A més, el dataset està format per la data d'adquisició dels registres, el número identificador de l'estació meteorològica, el seu nom, la provincia, l'altitud sobre el nivell del mar (en metres), la temperatura mitjana diària (ºC), la precipitació diària (en mm), la temperatura mínima (ºC), l'hora i minuts d'adquisició de la temperatura mínima, la temperatura màxima (ºC), l'hora i minuts d'adquisició de la temperatura màxima, la direcció de la ratxa de vent màxima, la velocitat mitjana del vent (m/s), l'hora de la ratxa màxima, l'insolació (en hores), la pressió màxima (hPa), l'hora de la pressió màxima, la pressió mínima (hPa), l'hora de la pressió mínima.

```
# Carreguem les dades del dataset AEMET
aemet_dataset <- read.csv("data/aemet_dataset.csv", sep = ";", header = TRUE,
                          stringsAsFactors = FALSE)
```

Renombrem les columnes del dataset per fer-les més descriptives, facilitar-ne el tractament i la unió amb les dades del *dataset ESIOS*.

```
# Renombrem el nom de les columnes
colnames(aemet_dataset) <- c("date", "climate_indicator", "station_name",
```

```
                            "province", "height_above_sea", "avg_daily_temp",
                            "daily_precipitation", "min_daily_temp",
                            "hour_min_temp", "max_daily_temp",
                            "hour_max_temp", "max_wind_direction",
                            "avg_wind_speed", "max_wind_speed",
                            "hour_max_wind_speed", "insolation",
                            "max_pressure", "hour_max_pressure",
                            "min_pressure", "hour_min_pressure")

# Taula resum que representa les principals característiques de les diferents variables
str(aemet_dataset)
```

```
## 'data.frame':    175855 obs. of  20 variables:
##  $ date               : chr  "2020-11-01" "2020-11-01" "2020-11-01" "2020-11-01" ...
##  $ climate_indicator  : chr  "9091O" "9091R" "8178D" "8175" ...
##  $ station_name       : chr  "FORONDA-TXOKIZA" "VITORIAGASTEIZAEROPUERTO" "ALBACETE" "ALBACETEBASEAÉREA" ...
##  $ province           : chr  "ARABA/ALAVA" "ARABA/ALAVA" "ALBACETE" "ALBACETE" ...
##  $ height_above_sea   : int  513 513 674 702 880 605 81 43 15 575 ...
##  $ avg_daily_temp     : chr  "17,0" "17,7" "16,4" "14,6" ...
##  $ daily_precipitation: chr  "0,0" "0,0" "0,0" "0,0" ...
##  $ min_daily_temp     : chr  "13,0" "14,0" "7,9" "5,0" ...
##  $ hour_min_temp      : chr  "06:20" "06:26" "07:00" "05:30" ...
##  $ max_daily_temp     : chr  "21,1" "21,4" "24,9" "24,2" ...
##  $ hour_max_temp      : chr  "12:30" "12:34" "14:30" "14:40" ...
##  $ max_wind_direction : int  25 25 29 30 34 26 99 99 20 20 ...
##  $ avg_wind_speed     : chr  "5,8" "5,8" "0,3" "1,9" ...
##  $ max_wind_speed     : chr  "10,8" "10,8" "4,4" "6,1" ...
##  $ hour_max_wind_speed: chr  "17:53" "17:53" "13:40" "14:20" ...
##  $ insolation         : chr  "6,3" "7,7" "" "9,4" ...
##  $ max_pressure       : chr  "964,9" "964,9" "949,2" "945,4" ...
##  $ hour_max_pressure  : chr  "11" "Varias" "10" "9" ...
##  $ min_pressure       : chr  "962,4" "962,0" "946,6" "942,3" ...
##  $ hour_min_pressure  : chr  "24" "24" "17" "16" ...
```

El dataset conté 11 columnes i 190359 files. Totes les columnes són de tipus *character*, malgrat que la majoria d'elles haurien de ser de tipus numèric. Caldrà fer un preprocessat previ per poder tractar-les com a numèriques. A més, també es preprocessarà la columna *date* per a poder tractar-la com a data.

2.2.2. Processat dels separadors decimals i transformació a variables numèriques

Com que els separadors decimals són diferents als que utilitza R, hem de processar-los per tal de poder-los transformar a columnes numèriques.

```
# Processat dels separadors decimals
aemet_dataset <- aemet_dataset %>%
  mutate_at(vars(avg_daily_temp, daily_precipitation, min_daily_temp,
  max_daily_temp, avg_wind_speed, max_wind_speed, insolation, max_pressure,
  min_pressure), funs(gsub(",", ".", ., fixed = TRUE)))

# Transformem les columnes a numèriques
aemet_dataset <- aemet_dataset %>%
  mutate_at(vars(avg_daily_temp, daily_precipitation, min_daily_temp,
  max_daily_temp, avg_wind_speed, max_wind_speed, insolation, max_pressure,
  min_pressure), funs(as.numeric(.)))

# Verificació del tipus de dades
sapply(aemet_dataset, class)
```

```
##               date   climate_indicator        station_name            province
##        "character"         "character"         "character"         "character"
##    height_above_sea     avg_daily_temp daily_precipitation      min_daily_temp
##           "integer"          "numeric"           "numeric"           "numeric"
##       hour_min_temp     max_daily_temp       hour_max_temp  max_wind_direction
##         "character"          "numeric"         "character"           "integer"
##      avg_wind_speed     max_wind_speed hour_max_wind_speed          insolation
##           "numeric"          "numeric"         "character"           "numeric"
##        max_pressure   hour_max_pressure        min_pressure   hour_min_pressure
##           "numeric"         "character"           "numeric"         "character"
```

```
# Mostrem les primeres files del dataset
head(aemet_dataset)
```

```
##         date climate_indicator             station_name    province
## 1 2020-11-01             9091O          FORONDA-TXOKIZA ARABA/ALAVA
## 2 2020-11-01             9091R VITORIAGASTEIZAEROPUERTO ARABA/ALAVA
## 3 2020-11-01             8178D                 ALBACETE    ALBACETE
## 4 2020-11-01              8175        ALBACETEBASEAÉREA    ALBACETE
## 5 2020-11-01             8177A               CHINCHILLA    ALBACETE
```

```
## 6 2020-11-01            7096B                  HELLÍN    ALBACETE
##   height_above_sea avg_daily_temp daily_precipitation min_daily_temp
## 1              513           17.0                   0          13.0
## 2              513           17.7                   0          14.0
## 3              674           16.4                   0           7.9
## 4              702           14.6                   0           5.0
## 5              880           14.7                   0           8.9
## 6              605           19.2                   0          12.6
##   hour_min_temp max_daily_temp hour_max_temp max_wind_direction avg_wind_speed
## 1         06:20           21.1         12:30                25            5.8
## 2         06:26           21.4         12:34                25            5.8
## 3         07:00           24.9         14:30                29            0.3
## 4         05:30           24.2         14:40                30            1.9
## 5         06:40           20.5         14:30                34            1.7
## 6         05:40           25.7         15:00                26            1.7
##   max_wind_speed hour_max_wind_speed insolation max_pressure hour_max_pressure
## 1           10.8               17:53        6.3        964.9               11
## 2           10.8               17:53        7.7        964.9            Varias
## 3            4.4               13:40         NA        949.2               10
## 4            6.1               14:20        9.4        945.4                9
## 5            6.9               15:10         NA        925.7               10
## 6            5.8             Varias          NA        956.0               10
##   min_pressure hour_min_pressure
## 1        962.4               24
## 2        962.0               24
## 3        946.6               17
## 4        942.3               16
## 5        923.0               17
## 6        953.0               16
```

2.2.3. Transformar les columnes al format correcte

A continuació, transformarem les columnes *date* a tipus *Date*, les variables *climate_indicator*, *station_name*, *province*, *hour_min_temp*, *hour_max_temp*, *hour_max_wind_speed*, *hour_max_pressure* i *hour_min_pressure* a factor.

```
# Transformem la columna date a tipus Date
aemet_dataset$date <- as.Date(aemet_dataset$date, format = "%Y-%m-%d")

# Transformem les columnes a factor
aemet_dataset <- aemet_dataset %>%
  mutate_at(vars(climate_indicator, station_name, province, hour_min_temp,
  hour_max_temp, hour_max_wind_speed, hour_max_pressure, hour_min_pressure),
  funs(as.factor(.)))

# Verificació del tipus de dades
sapply(aemet_dataset, class)
```

```
##               date   climate_indicator        station_name            province
##             "Date"            "factor"            "factor"            "factor"
##    height_above_sea      avg_daily_temp daily_precipitation      min_daily_temp
##           "integer"           "numeric"           "numeric"           "numeric"
##       hour_min_temp      max_daily_temp       hour_max_temp  max_wind_direction
##            "factor"           "numeric"            "factor"           "integer"
##      avg_wind_speed      max_wind_speed hour_max_wind_speed          insolation
##           "numeric"           "numeric"            "factor"           "numeric"
##        max_pressure   hour_max_pressure        min_pressure   hour_min_pressure
##           "numeric"            "factor"           "numeric"            "factor"
```

2.2.4. Filtrem les columnes innecessàries

Filtrem les columnes *climate_indicator*, *station_name* i aqeulles que contenen les hores on s'han recollit els registres (*hour_min_temp*, *hour_max_temp*, *hour_max_wind_speed*, *hour_max_pressure* i *hour_min_pressure*)

```
# Filtrem les columnes climate_indicator i station_name using select
aemet_dataset <- aemet_dataset %>%
  select(-climate_indicator, -station_name, -hour_min_temp, -hour_max_temp,
          -hour_max_wind_speed, -hour_max_pressure, -hour_min_pressure)
```

2.2.5. Tractament de valors nuls

A l'apartat 3.1, se'ns demana identificar i gestionar els valors nuls. Tanmateix, el preprocessat que realitzarem a l'apartat següent requereix haver tractat abans els valors nuls, per poder fer un càlcul correcte del valor

promig. En primer lloc, comprovem si hi ha valors nuls:

```
# Comprovem si hi ha valors nuls a les dades per columnes
colSums(is.na(aemet_dataset))
```

```
##                date            province    height_above_sea      avg_daily_temp
##                   0                   0                   0                5761
## daily_precipitation      min_daily_temp      max_daily_temp  max_wind_direction
##                8576                5747                5697               12838
##      avg_wind_speed      max_wind_speed          insolation        max_pressure
##               12213               12838               74887               40567
##        min_pressure
##               40568
```

Es pot veure que algunes columnes contenen valors nuls. Existeixen diversos mètodes per tractar-los entre ells la imputació amb el valor promig. Tot i què aquest mètode podria ser vàlid en aquest cas, en treballar amb dades temporals, es pot utilitzar la interpolació dels valors faltants tal i com s'ha fet amb el dataset ESIOS:

```
# Interpolació dels valors nuls
aemet_dataset <- aemet_dataset %>% mutate_at(vars(avg_daily_temp, daily_precipitation, avg_wind_speed, max_wind_speed, max_wind_direction,
                                  min_daily_temp, max_daily_temp, insolation, max_pressure, min_pressure), funs(na.approx(.)))

# Comprovem si hi ha valors nuls a les dades per columnes
colSums(is.na(aemet_dataset))
```

```
##                date            province    height_above_sea      avg_daily_temp
##                   0                   0                   0                   0
## daily_precipitation      min_daily_temp      max_daily_temp  max_wind_direction
##                   0                   0                   0                   0
##      avg_wind_speed      max_wind_speed          insolation        max_pressure
##                   0                   0                   0                   0
##        min_pressure
##                   0
```

2.2.6. Calcular promitjos de les variables numèriques

Cada provincia pot tenir més d'una estació meteorològica. Així doncs, per tal de simplificar l'anàlisi de les dades, es procedirà a calcular el promig de les variables numèriques en base a les columnes *date* i *province*.

```
# Agrupem les dades per province i calculem els promitjos
aemet_dataset <- aemet_dataset %>%
  group_by(date, province) %>% summarise_all(funs(mean(., na.rm = FALSE)))

# Mostrem les primeres files del dataset
head(aemet_dataset)
```

```
## # A tibble: 6 x 13
## # Groups:   date [1]
##   date       province    heigh~1 avg_d~2 daily~3 min_d~4 max_d~5 max_w~6 avg_w~7
##   <date>     <fct>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 2020-11-01 ACORUÑA         154.    18.0 7.66       17.1    18.8    22.8    7.55
## 2 2020-11-01 ALBACETE        715.    16.2 0           8.6    23.8    29.8    1.4
## 3 2020-11-01 ALICANTE        178.    17.4 0.05       10.2    24.6    59.5    1.68
## 4 2020-11-01 ALMERIA         240.    17.2 0          11.9    22.5    37.2    1.68
## 5 2020-11-01 ARABA/ALAVA     513     17.4 0          13.5    21.2    25      5.8
## 6 2020-11-01 ASTURIAS        368.    17.8 0.00909    15.0    20.6    19.4    2.66
## # ... with 4 more variables: max_wind_speed <dbl>, insolation <dbl>,
## #   max_pressure <dbl>, min_pressure <dbl>, and abbreviated variable names
## #   1: height_above_sea, 2: avg_daily_temp, 3: daily_precipitation,
## #   4: min_daily_temp, 5: max_daily_temp, 6: max_wind_direction,
## #   7: avg_wind_speed
```

2.2.7. Creació de la variable *avg_pressure*

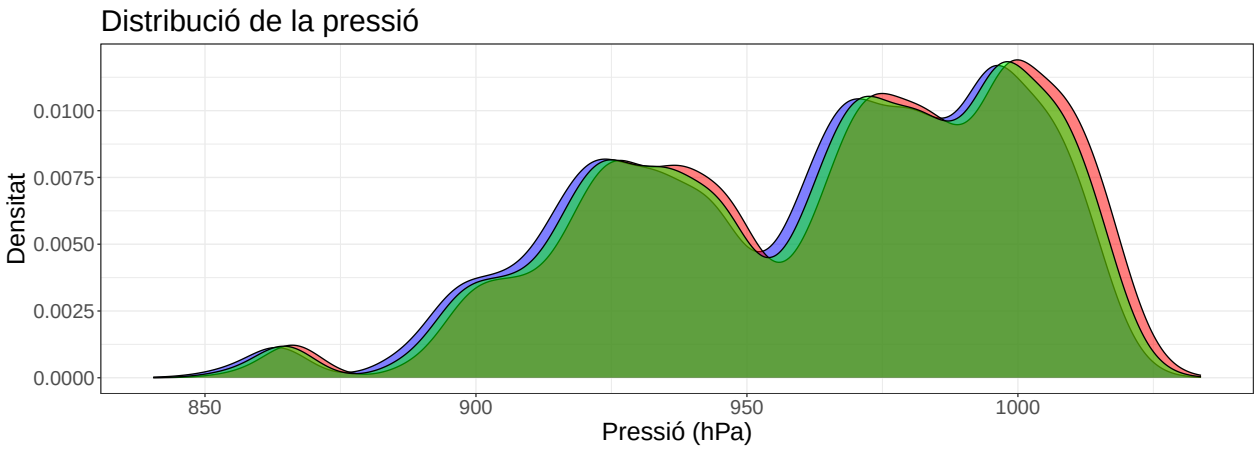
A continuació, creem la variable *avg_pressure* a partir de les variables *max_pressure* i *min_pressure*, això ens permetrà simplicar el dataset i eliminar dades innecessàries.

```
# Creació de la variable avg_pressure
aemet_dataset <- aemet_dataset %>% mutate(avg_pressure = (max_pressure + min_pressure) / 2)
```

Representem també la distribució de les dades per les variables *min_pressure*, *max_pressure* i *avg_pressure*:

```
# Histograma de la pressió
ggplot(aemet_dataset, aes(x = min_pressure)) +
  geom_density(fill = "blue", color = "black", alpha = 0.5) +
  geom_density(aes(x = max_pressure), fill = "red", color = "black", alpha = 0.5) +
  geom_density(aes(x = avg_pressure), fill = "green", color = "black", alpha = 0.5) +
  scale_x_continuous(name = "Pressió (hPa)") +
```

```
  scale_y_continuous(name = "Densitat") +
  scale_fill_manual(name = "Pressió",
                    values = c("blue", "red", "green"),
                    labels = c("Mínima", "Màxima", "Mitjana")) +
  labs(title = "Distribució de la pressió") +
  theme_bw() +
  theme(text = element_text(size = 20))
```



2.2.8. Discretització de la variable *avg_pressure*

A continuació, discretitzem la variable *avg_pressure*. La pressió atmosfèrica es classifica en funció de l'alçada respecte un nivell de referència, normalment el nivell del mar. Hi ha tres categories principals:

- Baixa pressió (*Depression*): els sistemes de baixes pressions estan associats amb núvols, precipitació i temps generalment inestable. Es caracteritza per pressions a nivell de superfície per sota de 1010 hPa.
- Pressió normal (*Normal*): aquesta es refereix a una superfície de pressió que està a la mateixa alçada que el nivell de referència. La pressió normal està associada amb temps "normal". Es caracteritza per pressions a nivell de superfície al voltant de 1013.25 hPa.
- Alta pressió (*Anticyclone*): els sistemes d'altes pressions estan associats amb temps estable i clar. Es caracteritza per pressions a nivell de superfície per sobre de 1015 hPa.

```
# Discretització de la pressió
aemet_dataset <- aemet_dataset %>% mutate(avg_pressure_bin = cut(avg_pressure,
                            breaks = c(-Inf, 1010, 1015, Inf),
                            labels = c("Depression", "Normal", "Anticyclone")))
```

2.2.9. Discretització de la insolació

La variable *insolation* es discretitzarà en 3 intervals de 5 hores cadascun:

```
# Discretització de la insolació
aemet_dataset <- aemet_dataset %>% mutate(insolation_bin = cut(insolation,
                                          breaks = c(-Inf, 0, 5, Inf),
                                          labels = c("0-5", "5-10", "10+")))
```

2.1.10. Guardem el dataset AEMET preprocessat

Guardem el dataset AEMET preprocessat en un fitxer CSV:

```
# Guardem el dataset AEMET preprocessat en un fitxer CSV
write.csv(aemet_dataset, file = "data/aemet_dataset_preprocessed.csv",
          row.names = FALSE)
```