Anàlisi de la qualitat de l'aire de Catalunya

Robert Garcia Ventura

04/09/2019

Contents

1) Descripció del problema

Actualment la contaminació de l'aire és un fet real el qual les administracions públiques estan estudiant per minimitzar el seu impacte a la població i a la terra.

Hi han molts estudis que demostren que la contaminació de l'aire prova milers de morts al any i efecte greument la salut de les persones.

2) Definició dels objectius

L'objectiu principal d'aquest anàlisis consistirà en crear un model per poder predir la contaminaicó atmostèrica en base a les dades climàtiques a tots els municipis de Catalunyai altres factors, com per exemple si un dia és laboral o no. La predicció de la contaminació atmosfèrica es farà a nivell de població, estació (estació de Gencat) i ZQA (Zona Qualitat Aire).

Per poder crear un model primer de tot s'obtindran dades principalment de dos fonts: Gencat i Meteocat.

Des de Gencat, obtindrem la qualitat de l'aire en temps real a nivell horari amb una frequencia d'actualització a cada hora. Les dades que obtindrem de la qualitat de l'aire provenen de les 76 estacions que la Generalitat de Catalunya té arreu del territori. Cal destacar que la majoria de les estacions no mesuren tots els components. Els components que es mesuren en total són:

- SO2 (g/m3)
- PM10 (g/m3)
- O3 (g/m3)
- Benze (g/m3)
- Cl2 (g/m3)
- HCl (g/m3)
- CO (mg/m3)
- H2S (g/m3)
- NO2 (g/m3)

Des del Meteocat, obtindrem la predicció del temps de totes les poblacions de Catalunya per al dia actual i per els pròxims dos dies. La frequencia d'actualització d'aquestes dades és diària, les dades que es proporcionen per a cada una de les mesures són a nivell horari. Les dades climàtiques que es mesuren són:

- EST_CEL (estat del cel)
- HUM_REL (humitat relativa)
- PREC_ACU (precipitació acumulada)
- TEMP (temperatura)
- TEMP_XAF (temperatura xafagor)
- VENT_DIR (direcció del vent)
- VENT_VELO (velocitat del vent)

Per realitzar l'anàlisi també disposem del llistat de municipis de tot Catalunya i les seves posicions GPS.

Els valos que s'intentaràn predir són el PM10 i O3. A part de ser uns dels components més importants de cares a determinar la qualitat d'aire, també són dos components que mesuren moltes estacions a tota Catalunya i per tant disposarem de moltes dades per a realitzar l'anàlisi.

3) Seleccionar/Identificar les dades

En aquesta secció es carregaran totes les dades guardades a la base de dades.

3.1) Comprobació dels paràmetres

Aquest informe està preperat perquè es puguin modificar els filtres de dates en els paràmetres del informe. Això permet executar l'informe amb diferents rangs de dates.

Per això, primer de tot és necessari comprobar que els filtres que l'usuari ha introduït són correctes.

```
#Check if the username is empty
stopifnot(nchar(params$mysql_username) > 0)

#Check if the password is empty
stopifnot(nchar(params$mysql_password) > 0)

#Check if start_date is lower or equal than end_date
stopifnot(params$start_date <= params$end_date)
```

3.2) Connexió a la base de dades MySQL i obtenir totes les dades

Un cop s'han comprobat que els parametres d'entrada són correctes ja podem obtenir les dades des de la base de dades i guardar els valors en dataframes.

```
#Create the MySQL connection
mydb = dbConnect(MySQL(), user=params$mysql_username, password=params$mysql_password, dbname=params$mys

#Set the encoding UTF-8
res <- dbSendQuery(mydb, "SET NAMES utf8;")

#Get all the stations in the database
res <- dbSendQuery(mydb, "SELECT * FROM stations WHERE station_user_id = 'Generalitat de Catalunya'")
stations <- dbFetch(res, n = -1)

#Get all the municipalities in the database
res <- dbSendQuery(mydb, "SELECT muni_id_meteocat, muni_id_gencat, muni_name_meteocat, muni_name_gencat
municipalities <- dbFetch(res, n = -1)

#Get all the regions in the database
res <- dbSendQuery(mydb, "SELECT * FROM regions;")
regions <- dbFetch(res, n = -1)

#Get all the measurments types in the database
res <- dbSendQuery(mydb, "SELECT * FROM measurtypes;")
measurtypes <- dbFetch(res, n = -1)

#Get all the weather data in the database
res <- dbSendQuery(mydb, paste0("SELECT * FROM wheathermeasures WHERE wmeasur_predict = '1' AND wmeasur
weather <- dbFetch(res, n = -1)

#Get all the air quality in the database
res <- dbSendQuery(mydb, paste0("SELECT * FROM airqualitymeasures_realtime WHERE airrmeasur_datetime >=
```

```r
airquality <- dbFetch(res, n = -1)

#Clear the result
result <- dbClearResult(res)

#Disconnect from the database
result <- dbDisconnect(mydb)

#Remove variables
remove(result)
remove(mydb)
remove(res)
```

4) Preperació de les dades

Per preperar el dataset final abans és necessari revisar les dades obtenides des de la base de dades. Aquestes s'han de formatar per així posterioment poder aplicar els models de machine learning que ens permetran fer la predicció de la qualitat de l'aire.

En aquesta secció netejarem, transformarem, imputarem, crearem noves variables i nous conjunts de dades.

4.1) Preperació de les dades d'estacions

El dataset 'stations' conté les dades de totes les estacions del nostre sistema. Això siginifica totes les estacions de qualitat de l'aire de la Generalitat de Catalunya i les pròpies de RoMi Box. Per a cada estació hi ha la seva posició GPS i en el cas de les de la Generalitat de Catalunya també tenim l'altura. També tenim l'ID del municipi en el qual es troba l'estació. Aquest camp serà especialment útil per posteriorment agrupar les dades per a municipi, comarca i ZQA (Zona de Qualitat de l'Aire).

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(stations)
```

```
##   station_id                          station_name
## 1    ES0584A                     Montcada i Reixac
## 2    ES0691A                   Barcelona (Poblenou)
## 3    ES0692A              L'Hospitalet de Llobregat
## 4    ES0694A Sant Vicen\303\247 dels Horts (Ribot)
## 5    ES0971A                Sant Andreu de la Barca
## 6    ES1018A                                Terrassa
##            station_user_id station_muni_id             station_muni_name
## 1 Generalitat de Catalunya               2             Montcada i Reixac
## 2 Generalitat de Catalunya               1                     Barcelona
## 3 Generalitat de Catalunya               1     l'Hospitalet de Llobregat
## 4 Generalitat de Catalunya               1 Sant Vicen\303\247 dels Horts
## 5 Generalitat de Catalunya               2       Sant Andreu de la Barca
## 6 Generalitat de Catalunya               2                      Terrassa
##   station_lat station_long station_alt station_height station_last_update
## 1    41.48197     2.188298          34           <NA> 2019-09-04 20:00:00
## 2    41.40388     2.204501           3           <NA> 2019-09-04 20:00:00
## 3    41.37048     2.114999          29           <NA> 2019-09-04 20:00:00
## 4    41.39219     2.009799          38           <NA> 2019-09-04 20:00:00
## 5    41.45080     1.974900          40           <NA> 2019-09-04 20:00:00
## 6    41.55611     2.007398         109           <NA> 2019-09-04 20:00:00
##   station_type_location station_type_source
## 1              suburban             traffic
## 2                 urban          background
## 3                 urban          background
## 4              suburban          background
## 5              suburban             traffic
## 6                 urban             traffic
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```
summary(stations)
```

```
##   station_id        station_name       station_user_id
##  Length:76          Length:76          Length:76
##  Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  station_muni_id    station_muni_name   station_lat     station_long
##  Length:76          Length:76          Min.   :40.55   Min.   :0.2884
##  Class :character   Class :character   1st Qu.:41.23   1st Qu.:1.1988
##  Mode  :character   Mode  :character   Median :41.42   Median :2.0036
##                                        Mean   :41.46   Mean   :1.7474
##                                        3rd Qu.:41.58   3rd Qu.:2.1876
##                                        Max.   :42.41   Max.   :3.2129
##   station_alt     station_height     station_last_update
##  Min.   :   3.0   Length:76          Length:76
##  1st Qu.:  34.0   Class :character   Class :character
##  Median :  93.0   Mode  :character   Mode  :character
##  Mean   : 207.9
##  3rd Qu.: 224.5
##  Max.   :1570.0
##  station_type_location station_type_source
##  Length:76             Length:76
##  Class :character      Class :character
##  Mode  :character      Mode  :character
##
##
##
```

Com podem veure la columna 'station_alt' té 18 valors NA.

Si agrupem per 'station_user_id' i es suma el total de NA per cada group podem veure que les estacions que no tenen el camp definit són les estacions de RoMi Box.

```
stations %>% select(station_user_id, station_alt) %>% group_by(station_user_id) %>% summarise(`Total NA
```

```
## # A tibble: 1 x 2
##   station_user_id          `Total NA`
##   <chr>                         <int>
## 1 Generalitat de Catalunya          0
```

Com que el model s'aplicarà només a les dades de les estacions de la Generalitat de Catalunya podem filtrar les estacions de RoMi Box.

```
stations <- stations %>% filter(station_user_id == "Generalitat de Catalunya")
```

Per veure el tipus de dades de cada columna podem utilitzar la funció 'str'.

```
str(stations)
```

```
## 'data.frame':    76 obs. of  12 variables:
##  $ station_id          : chr  "ES0584A" "ES0691A" "ES0692A" "ES0694A" ...
##  $ station_name        : chr  "Montcada i Reixac" "Barcelona (Poblenou)" "L'Hospitalet de Llobregat
##  $ station_user_id     : chr  "Generalitat de Catalunya" "Generalitat de Catalunya" "Generalitat de
##  $ station_muni_id     : chr  "2" "1" "1" "1" ...
##  $ station_muni_name   : chr  "Montcada i Reixac" "Barcelona" "l'Hospitalet de Llobregat" "Sant Vic
##  $ station_lat         : num  41.5 41.4 41.4 41.4 41.5 ...
##  $ station_long        : num  2.19 2.2 2.11 2.01 1.97 ...
##  $ station_alt         : num  34 3 29 38 40 109 41 39 97 56 ...
##  $ station_height      : chr  NA NA NA NA ...
##  $ station_last_update : chr  "2019-09-04 20:00:00" "2019-09-04 20:00:00" "2019-09-04 20:00:00" "20
```

```
##  $ station_type_location: chr  "suburban" "urban" "urban" "suburban" ...
##  $ station_type_source  : chr  "traffic" "background" "background" "background" ...
```

Un cop ja sabem el tipus i format de les dades ja podem començar amb la preperació del dataset.

Primer de tot cal veure que la variable 'station_height' és del tipus caràcter. Aquesta variable defineix l'altura en la qual esta l'estació respecta el terra. És diferent de 'station_alt' que diu l'altura respecte al nivell del mar.

```
#Convert 'station_height' to a numeric variable
stations$station_height <- as.numeric(stations$station_height)
```

Un cop convertida la variable a numéric podem veure si té valors o tots els valors són NA.

```
#Compare if the number of NA is the same of rows in the dataset stations
length(is.na(stations$station_height)) == nrow(stations)
```

```
## [1] TRUE
```

Com podem veure totes les files tenen aquesta columna com a NA per a tant no serà útil per a l'anàlisi i es pot eliminar.

```
#Remove column 'station_height' in the dataset stations
stations$station_height <- NULL
```

Tot seguit convertirem les variables categoriques a factors.

```
#Convert categorical columns to factors
stations$station_id <- as.factor(stations$station_id)
stations$station_name <- as.factor(stations$station_name)
stations$station_user_id <- as.factor(stations$station_user_id)
stations$station_muni_id <- as.factor(stations$station_muni_id)
stations$station_muni_name <- as.factor(stations$station_muni_name)
stations$station_last_update <- as.factor(stations$station_last_update)
```

4.2) Preperació de les dades dels municipis

El dataset 'municipalities' conté les dades de tots els municipis de Catalunya. Això inclou el codi de municipi, nom, codi de ZQA, codi de comarca i les coordenades GPS.

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(municipalities)
```

```
##   muni_id_meteocat muni_id_gencat          muni_name_meteocat
## 1           080018          08001                      Abrera
## 2           080023          08002           Aguilar de Segarra
## 3           080039          08003                      Alella
## 4           080044          08004                      Alpens
## 5           080057          08005 l'Ametlla del Vall\303\250s
## 6           080060          08006               Arenys de Mar
##              muni_name_gencat muni_region_id_gencat
## 1                      Abrera                    2
## 2           Aguilar de Segarra                    5
## 3                      Alella                    7
## 4                      Alpens                   10
## 5 l'Ametlla del Vall\303\250s                    2
## 6               Arenys de Mar                    7
```

```
##   muni_region_id_meteocat muni_lat_meteocat muni_long_meteocat
## 1                      11          41.51629           1.902257
## 2                       7          41.73895           1.630983
## 3                      21          41.49387           2.295160
## 4                      24          42.11928           2.101153
## 5                      41          41.66942           2.261534
## 6                      21          41.57957           2.551636
##   muni_lat_gencat muni_long_gencat
## 1        41.51752         1.901780
## 2        41.73921         1.627507
## 3        41.49197         2.294082
## 4        42.11895         2.101462
## 5        41.66832         2.260500
## 6        41.57971         2.550868
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```
summary(municipalities)
```

```
##  muni_id_meteocat   muni_id_gencat     muni_name_meteocat
##  Length:947         Length:947         Length:947
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  muni_name_gencat   muni_region_id_gencat muni_region_id_meteocat
##  Length:947         Min.   : 1.000        Min.   : 1.00
##  Class :character   1st Qu.: 5.000        1st Qu.: 9.00
##  Mode  :character   Median : 8.000        Median :20.00
##                     Mean   : 8.657        Mean   :20.16
##                     3rd Qu.:13.000        3rd Qu.:31.00
##                     Max.   :15.000        Max.   :42.00
##  muni_lat_meteocat muni_long_meteocat muni_lat_gencat muni_long_gencat
##  Min.   :40.54     Min.   :0.2506     Min.   :40.54   Min.   :0.2511
##  1st Qu.:41.43     1st Qu.:1.0698     1st Qu.:41.43   1st Qu.:1.0721
##  Median :41.71     Median :1.7434     Median :41.71   Median :1.7490
##  Mean   :41.73     Mean   :1.7509     Mean   :41.73   Mean   :1.7496
##  3rd Qu.:42.06     3rd Qu.:2.3485     3rd Qu.:42.06   3rd Qu.:2.3380
##  Max.   :42.84     Max.   :3.2758     Max.   :42.84   Max.   :3.2780
```

Per veure el tipus de dades de cada columna podem utilitzar la funció 'str'.

```
str(municipalities)
```

```
## 'data.frame':    947 obs. of  10 variables:
##  $ muni_id_meteocat       : chr  "080018" "080023" "080039" "080044" ...
##  $ muni_id_gencat         : chr  "08001" "08002" "08003" "08004" ...
##  $ muni_name_meteocat     : chr  "Abrera" "Aguilar de Segarra" "Alella" "Alpens" ...
##  $ muni_name_gencat       : chr  "Abrera" "Aguilar de Segarra" "Alella" "Alpens" ...
##  $ muni_region_id_gencat  : int  2 5 7 10 2 7 7 5 7 5 ...
##  $ muni_region_id_meteocat: int  11 7 21 24 41 21 21 6 21 7 ...
##  $ muni_lat_meteocat      : num  41.5 41.7 41.5 42.1 41.7 ...
##  $ muni_long_meteocat     : num  1.9 1.63 2.3 2.1 2.26 ...
##  $ muni_lat_gencat        : num  41.5 41.7 41.5 42.1 41.7 ...
##  $ muni_long_gencat       : num  1.9 1.63 2.29 2.1 2.26 ...
```

Els codis de municipi han de tenir una longitud de 6 caràcters, en aquest cas tot i ser del tipus char aquestes variables no tenen la mateixa longitud. Per fer tots els codis de la mateixa longitud ho farem de la següent forma:

```
#Create the code as a char with 0 at the beginning and always the lenght of 6
municipalities$muni_id_meteocat <- String_Right(paste0("000000",municipalities$muni_id_meteocat), 6)
municipalities$muni_id_gencat <- String_Right(paste0("000000",municipalities$muni_id_gencat), 6)
```

```
#Convert the IDs to a categorical variable
municipalities$muni_id_meteocat <- as.factor(municipalities$muni_id_meteocat)
municipalities$muni_id_gencat <- as.factor(municipalities$muni_id_gencat)
```

El que ens falta calcular en aquest dataset és l'estació de qualitat d'aire més pròxima per a cada municipi. Això ens serà útil posteriorment en el cas de voler fer la predicció de la qualitat de l'aire a nivell de municipi.

Per calcular l'estació de cada municipi farem una matriu de distàncies utilitzant les posicions GPS de cada estació i municipi. És molt important tenir en compte que s'assignarà a cada municipi l'estació més pròxima dins la ZQA en la qual es troba el municipi.

```
#Select the ID of the station and the region with the GPS position of each municipality
municipi <- municipalities %>% select(muni_id_meteocat, muni_region_id_gencat, muni_lat_gencat, muni_lo

#Create new matrix with 1 column
muni_airquality <- data.frame(matrix(ncol = 2, nrow = 0), stringsAsFactors = FALSE)

for(zqa in unique(municipi$muni_region_id_gencat)){

  #Subselect the stations of each zone
  sub_stations <- stations %>% filter(station_id != "ES1899A") %>% filter(station_muni_id == zqa)

  #Subselect the municipalities of each zone
  sub_municipi <- municipi[(municipi$muni_region_id_gencat == zqa),]

  #Calculate the distance for each farmer to all the weather stations
  muni_stations_matrix <- as.data.frame(geosphere::distm(sub_municipi[,c("muni_lat_gencat","muni_long_g

  #Set row names (ID air quality stations)
  row.names(muni_stations_matrix) <- sub_municipi$muni_id_meteocat

  #Set column names (ID municipalities)
  colnames(muni_stations_matrix) <- sub_stations$station_id

  #Calculate for each air quality station the most closer municipality
  for(i in 1:nrow(muni_stations_matrix)){
    row_muni <- as.data.frame(muni_stations_matrix[i,])
    colnames(row_muni) <- colnames(muni_stations_matrix)
    muni_airquality <- rbind(muni_airquality, c(sub_municipi[i,"muni_id_meteocat"], names(sort(row_muni
  }

}

#Rename column names
colnames(muni_airquality) <- c("muni_id","station_id")

#Rename row names
row.names(muni_airquality) <- muni_airquality$muni_id
```

```
#Remove stations matrix dataframe
remove(muni_stations_matrix)
```

En aquest punt tenim una taula anomenada 'muni_airquality' amb tantes files com a municipis i per a cada municipi el codi de l'estació de qualitat d'aire més pròxima al centre del municipi dins la seva ZQA.

Finalment només queda afegir la nova columna en al dataset 'municipalities'.

```
#Set values of much closer station to each municipality
municipalities$muni_station_id <- muni_airquality[municipalities$muni_id_meteocat,"station_id"]

#Set column 'muni_station_id' as factor
municipalities$muni_station_id <- as.factor(municipalities$muni_station_id)
```

4.3) Preperació de les dades de les regions

El dataset 'regions' conté els noms de totes les comàrques (regions) de Catalunya. Això inclou el codi de regió i el nom de la regió.

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(regions)
```

```
##   region_id           region_name
## 1         1              Alt Camp
## 2         2    Alt Empord\303\240
## 3         3    Alt Pened\303\250s
## 4         4            Alt Urgell
## 5         5 Alta Ribagor\303\247a
## 6         6                 Anoia
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```
summary(regions)
```

```
##    region_id     region_name
##  Min.   : 1.00   Length:42
##  1st Qu.:11.25   Class :character
##  Median :21.50   Mode  :character
##  Mean   :21.50
##  3rd Qu.:31.75
##  Max.   :42.00
```

Per veure el tipus de dades de cada columna podem utilitzar la funció 'str'.

```
str(regions)
```

```
## 'data.frame':    42 obs. of  2 variables:
##  $ region_id  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region_name: chr  "Alt Camp" "Alt Empord\303\240" "Alt Pened\303\250s" "Alt Urgell" ...
```

En aquest cas únicament es passarà la variable 'region_name' a categòrica.

```
#Convert 'region_name' to a categorical variable
regions$region_name <- as.factor(regions$region_name)
```

4.4) Preperació de les dades dels tipus de mesures

El dataset 'measurtypes' conté la llista dels diferents tipus de mesures. Per a cada mesure hi ha el seu id, nom, descripció i unitat de mesura.

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(measurtypes)
```

```
##   mtype_id mtype_name                    mtype_desc   mtype_unit
## 1        1        SO2                           SO2 \302\265g/m3
## 2       10       PM10        Part\303\255cules PM10 \302\265g/m3
## 3       14         O3 Oz\303\263 troposf\303\250ric \302\265g/m3
## 4       30       C6H6                         C6H6 \302\265g/m3
## 5       53        Cl2                         Clor \302\265g/m3
## 6       58        HCl           Clorur d'hidrogen \302\265g/m3
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```
summary(measurtypes)
```

```
##    mtype_id          mtype_name         mtype_desc
##  Length:9           Length:9           Length:9
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##   mtype_unit
##  Length:9
##  Class :character
##  Mode  :character
```

Com podem veure les 4 columnes són del tipus caràcter.

El que farem en aquest dataset serà convertir les 4 columnes de tipus caràcter a factors per així poder treballar millor les dades posteriorment.

```
#Convert categorical columns to factors
measurtypes$mtype_id <- as.factor(measurtypes$mtype_id)
measurtypes$mtype_name <- as.factor(measurtypes$mtype_name)
measurtypes$mtype_desc <- as.factor(measurtypes$mtype_desc)
measurtypes$mtype_unit <- as.factor(measurtypes$mtype_unit)
```

4.5) Preperació de les dades meteorològiques

El dataset 'weather' conté totes les dades meteorològiques per a totes els municipis de Catalunya a nivell horari. Les dades provenen de Meteocat.

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(weather)
```

```
##   wmeasur_wstation_id wmeasur_mtype_id    wmeasur_datetime wmeasur_predict
## 1              080155          EST_CEL 2019-07-01 00:00:00               1
## 2              080155          EST_CEL 2019-07-01 01:00:00               1
## 3              080155          EST_CEL 2019-07-01 02:00:00               1
## 4              080155          EST_CEL 2019-07-01 03:00:00               1
## 5              080155          EST_CEL 2019-07-01 04:00:00               1
## 6              080155          EST_CEL 2019-07-01 05:00:00               1
##   wmeasur_value
```

```
## 1             1
## 2             1
## 3             1
## 4             1
## 5             1
## 6             1
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```
summary(weather)
```

```
##  wmeasur_wstation_id wmeasur_mtype_id   wmeasur_datetime
##  Length:548352       Length:548352      Length:548352
##  Class :character    Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character
##
##
##
##  wmeasur_predict    wmeasur_value
##  Length:548352      Min.   : -0.231
##  Class :character   1st Qu.:  1.000
##  Mode  :character   Median : 21.000
##                     Mean   : 43.473
##                     3rd Qu.: 43.631
##                     Max.   :359.999
```

Per veure el tipus de dades de cada columna podem utilitzar la funció 'str'.

```
str(weather)
```

```
## 'data.frame':    548352 obs. of  5 variables:
##  $ wmeasur_wstation_id: chr  "080155" "080155" "080155" "080155" ...
##  $ wmeasur_mtype_id   : chr  "EST_CEL" "EST_CEL" "EST_CEL" "EST_CEL" ...
##  $ wmeasur_datetime   : chr  "2019-07-01 00:00:00" "2019-07-01 01:00:00" "2019-07-01 02:00:00" "2019
##  $ wmeasur_predict    : chr  "1" "1" "1" "1" ...
##  $ wmeasur_value      : num  1 1 1 1 1 1 1 1 1 1 ...
```

Un cop ja sabem el tipus i format de les dades ja podem començar amb la preperació del dataset.

En aquest cas convertirem la columna 'wmeasur_datetime' de tipus caràcter a POSIXct (format per treballar amb dates a R). També convertirem la resta de columnes de tipus caràcter a factors per així poder-les manipular molt millor posteriorment.

```
#Make weather as data type "dataframe"
weather <- as.data.frame(weather)

#Transform the column date to a date type
weather$wmeasur_datetime <- as.POSIXct(weather$wmeasur_datetime, format = "%Y-%m-%d %H:%M:%S")

#Convert categorical columns to factors
weather$wmeasur_wstation_id <- as.factor(weather$wmeasur_wstation_id)
weather$wmeasur_mtype_id <- as.factor(weather$wmeasur_mtype_id)
weather$wmeasur_predict <- as.factor(weather$wmeasur_predict)

#Reshape the weather data and create a new dataframe
weather_data <- reshape2::dcast(weather, wmeasur_wstation_id + wmeasur_datetime + wmeasur_predict ~ wmea
```

Utilitzant la funció 'reshape2::dcast' s'ha aconseguit passar tots els valors de files a columns agrupats per

municipi, data i preddició.

Si observem com ha quedat el dataset 'weather_data' després de la transformació tenim el següent.

```
summary(weather_data)
```

```
##  wmeasur_wstation_id wmeasur_datetime              wmeasur_predict
##  080155 : 1224       Min.   :2019-07-01 00:00:00   1:78336
##  080193 : 1224       1st Qu.:2019-07-13 17:45:00
##  080229 : 1224       Median :2019-07-26 11:30:00
##  080749 : 1224       Mean   :2019-07-26 11:30:00
##  080771 : 1224       3rd Qu.:2019-08-08 05:15:00
##  080898 : 1224       Max.   :2019-08-20 23:00:00
##  (Other):70992
##     EST_CEL         HUM_REL          PREC_ACU             TEMP
##  Min.   : 1.00   Min.   : 9.715   Min.   :-0.23100   Min.   : 7.112
##  1st Qu.: 1.00   1st Qu.:50.702   1st Qu.: 0.00000   1st Qu.:21.227
##  Median : 1.00   Median :67.357   Median : 0.00000   Median :24.132
##  Mean   : 5.37   Mean   :65.550   Mean   : 0.04504   Mean   :24.197
##  3rd Qu.: 4.00   3rd Qu.:82.137   3rd Qu.: 0.00000   3rd Qu.:27.075
##  Max.   :24.00   Max.   :99.796   Max.   :71.57300   Max.   :41.237
##
##     TEMP_XAF         VENT_DIR         VENT_VELO
##  Min.   : 6.167   Min.   :  0.013   Min.   : 0.002
##  1st Qu.:21.422   1st Qu.:119.448   1st Qu.: 1.092
##  Median :24.471   Median :176.853   Median : 2.237
##  Mean   :24.446   Mean   :182.351   Mean   : 2.353
##  3rd Qu.:27.692   3rd Qu.:251.023   3rd Qu.: 3.305
##  Max.   :38.788   Max.   :359.999   Max.   :14.487
##
```

4.6) Preperació de les dades de qualitat de l'aire

El dataset 'airquality' conté les dades de qualitat de l'aire a tot Catalunya a nivell horari. Les dades provenen de les 76 estacions de qualitat de l'aire de la Generalitat de Catalunya.

Les dades que s'han descarregat de la base de dades venen en el següent format:

```
head(airquality)
```

```
##   airrmeasur_airstation_id airrmeasur_mtype_id airrmeasur_datetime
## 1                  ES0584A                   1 2019-07-01 00:00:00
## 2                  ES0584A                   1 2019-07-01 01:00:00
## 3                  ES0584A                   1 2019-07-01 02:00:00
## 4                  ES0584A                   1 2019-07-01 03:00:00
## 5                  ES0584A                   1 2019-07-01 04:00:00
## 6                  ES0584A                   1 2019-07-01 05:00:00
##   airrmeasur_value
## 1                1
## 2                1
## 3                1
## 4                1
## 5                1
## 6                1
```

Per veure un petit resum de les dades ho podem fer utilitzant la funció 'summary'.

```r
summary(airquality)
```

```
##  airrmeasur_airstation_id airrmeasur_mtype_id airrmeasur_datetime
##  Length:833085            Length:833085       Length:833085
##  Class :character         Class :character    Class :character
##  Mode  :character         Mode  :character    Mode  :character
##
##
##
##
##  airrmeasur_value
##  Min.   :  0.1
##  1st Qu.:  2.0
##  Median : 12.0
##  Mean   : 24.3
##  3rd Qu.: 33.0
##  Max.   :243.0
##  NA's   :562367
```

Per veure el tipus de dades de cada columna podem utilitzar la funció 'str'.

```r
str(airquality)
```

```
## 'data.frame':    833085 obs. of  4 variables:
##  $ airrmeasur_airstation_id: chr  "ES0584A" "ES0584A" "ES0584A" "ES0584A" ...
##  $ airrmeasur_mtype_id     : chr  "1" "1" "1" "1" ...
##  $ airrmeasur_datetime     : chr  "2019-07-01 00:00:00" "2019-07-01 01:00:00" "2019-07-01 02:00:00" "
##  $ airrmeasur_value        : num  1 1 1 1 1 1 1 1 1 1 ...
```

Un cop ja sabem el tipus i format de les dades ja podem començar amb la preperació del dataset.

```r
#Convert categorical columns to factors
airquality$airrmeasur_airstation_id <- as.factor(airquality$airrmeasur_airstation_id)
airquality$airrmeasur_mtype_id <- as.factor(airquality$airrmeasur_mtype_id)

#Transform the column date to a date type
airquality$airrmeasur_datetime <- as.POSIXct(airquality$airrmeasur_datetime, format = "%Y-%m-%d %H:%M:%S

#Merge the dataset 'airquality' and 'measurtypes' by the measur type ID
airquality_data <- merge(airquality, measurtypes %>% select(mtype_id,mtype_name,mtype_unit), by.x = "air

#Convert categorical columns to factors
airquality_data$mtype_unit <- as.factor(airquality_data$mtype_unit)

#Change the levels of the column 'mtype_unit'
levels(airquality_data$mtype_unit) <- c("mg_m3","ug_m3")

#Reshape the airquality data to put all the parameters in on row by station and time
airquality_data <- reshape2::dcast(airquality_data, airrmeasur_airstation_id + airrmeasur_datetime ~ mty
```

Un cop hem creat el nou dataset amb tots els paràmetres per columnes i agrupat per estació, dia i hora ja podem revisar les dades si hi han NA. En cas que tinguem NA a les dades les imputarem amb els valors de les estacions més pròximes que tinguin els valors.

Primer de tot comprobarem la quantitat de NA per a cada columna, això es pot fer fàcilment utilitzant la següent funció.

```r
#Show the summary of the dataset 'airquality_data'
summary(airquality_data)
```

```
##  airrmeasur_airstation_id airrmeasur_datetime             C6H6_ug_m3
##  ES0584A: 1224            Min.   :2019-07-01 00:00:00   Min.   : 0.60
##  ES0691A: 1224            1st Qu.:2019-07-13 16:00:00   1st Qu.: 0.60
##  ES0692A: 1224            Median :2019-07-26 09:00:00   Median : 0.60
##  ES0694A: 1224            Mean   :2019-07-26 09:42:41   Mean   : 0.77
##  ES0971A: 1224            3rd Qu.:2019-08-08 03:00:00   3rd Qu.: 0.60
##  ES1018A: 1224            Max.   :2019-08-20 23:00:00   Max.   :24.20
##  (Other):85221                                          NA's   :86586
##     CO_mg_m3       Cl2_ug_m3       H2S_ug_m3       HCl_ug_m3
##  Min.   :0.10    Min.   :11.80   Min.   : 0.80   Min.   : NA
##  1st Qu.:0.20    1st Qu.:11.80   1st Qu.: 1.10   1st Qu.: NA
##  Median :0.20    Median :11.80   Median : 1.60   Median : NA
##  Mean   :0.26    Mean   :11.82   Mean   : 2.41   Mean   :NaN
##  3rd Qu.:0.30    3rd Qu.:11.80   3rd Qu.: 3.00   3rd Qu.: NA
##  Max.   :5.40    Max.   :12.30   Max.   :32.40   Max.   : NA
##  NA's   :68604   NA's   :91416   NA's   :77987   NA's   :92565
##    NO2_ug_m3        O3_ug_m3        PM10_ug_m3      SO2_ug_m3
##  Min.   :  1.0   Min.   :  1.00   Min.   : 4.00   Min.   :  1.00
##  1st Qu.:  6.0   1st Qu.: 46.00   1st Qu.:18.00   1st Qu.:  1.00
##  Median : 12.0   Median : 70.00   Median :22.00   Median :  2.00
##  Mean   : 17.3   Mean   : 68.98   Mean   :24.19   Mean   :  2.31
##  3rd Qu.: 24.0   3rd Qu.: 92.00   3rd Qu.:29.00   3rd Qu.:  3.00
##  Max.   :143.0   Max.   :243.00   Max.   :79.00   Max.   :175.00
##  NA's   :15238   NA's   :33573    NA's   :50548   NA's   :45850
```

Com podem veure totes les columnes amb valors de la qualitat de l'aire tenen valors NA, i en especial podem veure que la columna anomenada 'HCl_ug_m3' no té cap valor. En aquest cas no podem imputar res i per tant eliminarem la columna.

```r
#Check if the number of rows in the dataset is the same as the number of nulls in the column 'HCl_ug_m3
if(nrow(airquality_data) == length(airquality_data$HCl_ug_m3)){
  airquality_data$HCl_ug_m3 <- NULL
}
```

També podem calcular el nombre total de NA del dataframe.

```r
#Calculate total number of NA in the dataframe 'airquality_data'
sum(is.na(airquality_data))
```

```
## [1] 469802
```

Si analitzem el nombre de NA per cada variable tenim el següent:

En el cas del benze tenim 0 dades amb valors NA, això representa el 0% de les observacions són NA.

Com que tenim un percentatge de valors NA molt elevat i tampoc és una variable objectiu s'eliminarà ja que la mostra no és representativa a nivell de tot Catalunya, ja que només hi han 5 estacions a tot Catalunya que mesurin aquest contaminant.

```r
#Remove column 'benze_ug_m3'
airquality_data$benze_ug_m3 <- NULL
```

En el cas del Cl2 tenim 91416 dades amb valors NA, això representa el 98.7587101% de les observacions són NA.

Amb la variable Cl2 també tenim un percentatge molt elevat de valors NA. Això és perquè a tot Catalunya

només hi ha una única estació que mesuri aquest contaminant. Com que no és representativa a tot Catalunya s'eliminarà.

```
#Remove column 'Cl2_ug_m3'
airquality_data$Cl2_ug_m3 <- NULL
```

En el cas del CO tenim 68604 dades amb valors NA, això representa el 74.1144061% de les observacions són NA.

En aquest cas tot i que el percentatge és força elevat, les estacions que mesuren aquest contaminant estan ben repartides arreu del territori de Catalunya. Per tant, al imputar la mostra serà força bona ja que s'imputaran els valors amb el de les estacions més pròximes, i en aquest cas al estar ben repartides podrem tenir una bona aproximació.

En el cas del H2S tenim 77987 dades amb valors NA, això representa el 84.2510668% de les observacions són NA.

En aquest cas tenim un percentatge força elevat de valors NA. El problema d'aquest contaminant, a diferència de CO, és que la gran majoria de les estacions que mesuren aquest contaminant estan a l'àrea de Tarragona. Això és un problema perquè al moment de imputar no sería una bona aproximació. Per tant, com que no hi han suficients dades ni tampoc estan ben representades també s'eliminarà.

```
#Remove column 'H2S_ug_m3'
airquality_data$H2S_ug_m3 <- NULL
```

En el cas del NO2 tenim 15238 dades amb valors NA, això representa el 16.4619457% de les observacions són NA.

El contaminant NO2 té un percentatge de valors NA molt petit per tant aquest el mantidrem.

En el cas del O3 tenim 33573 dades amb valors NA, això representa el 36.2696484% de les observacions són NA.

El contaminant O3 també té un percentatge de valors NA molt petit per tant també el mantidrem.

En el cas del PM10 tenim 50548 dades amb valors NA, això representa el 54.6081132% de les observacions són NA.

En aquest cas, tot i tenir un percentatge elevat, a part de que és una de les variables objectius, també cal dir que les estacions que mesuren aquest contaminant també estan ben repartides arreu del territori català per tant no s'eliminarà.

En el cas del SO2 tenim 45850 dades amb valors NA, això representa el 49.5327608% de les observacions són NA.

Igual que amb PM10, en el cas del contaminant SO2 tot i tenir un percentatge de valors NA elevat, la mostra d'estacions que mesurent el contaminant està ben repartida arreu del territori català. Per tant, tampoc s'eliminarà.

Un cop s'han eliminat les columnes poc representatives i amb molts valors NA, ja podem imputar la resta de valors. Per fer-ho, primer de tot tenim que calcular per a cada estació, quines són les seves estacions veïnes. Per tant, calcularem una matriu de distàncies de totes les estacions entre elles utilitzant la posició GPS de cada una d'elles.

```
#Filter only the stations from the "Generalitat de Catalunya" and save it in a new dataframe
stations_gencat <- stations %>% filter(station_user_id == "Generalitat de Catalunya")

#Calculate the distance for each farmer to all the weather stations
stations_gencat_matrix <- geosphere::distm(stations_gencat[,c("station_lat","station_long")],
                                           stations_gencat[,c("station_lat","station_long")])
```

```r
#Set column names
colnames(stations_gencat_matrix) <- stations_gencat$station_id

#Set row names
row.names(stations_gencat_matrix) <- stations_gencat$station_id

#Create new matrix to store the stations in order
order_matrix <- data.frame(matrix(ncol = ncol(stations_gencat_matrix), nrow = 0), stringsAsFactors = FA

#Calculate for each weather station the rest of the weather stations ordered from the most close to the
for(i in 1:nrow(stations_gencat_matrix)){
  order_matrix <- rbind(order_matrix, names(sort(stations_gencat_matrix[i,])), stringsAsFactors = FALSE)
}

#Remove first column (always distance is 0)
order_matrix[,colnames(order_matrix)[1]] <- NULL

#Rename column names
colnames(order_matrix) <- 1:(ncol(stations_gencat_matrix) - 1)

#Rename row names
row.names(order_matrix) <- stations_gencat$station_id

#Remove stations matrix dataframe
remove(stations_gencat_matrix)
```

Amb la matriu de distàncies ja coneixem per a cada estació quines són les seves estacions més pròximes. Per tant, ja podem imputar per a cada estació i data els valors NA amb els valors de les estacions més pròximes.

Per fer-ho, primer de tot farem una còpia de les dades i a partir d'aqui per a cada fila es consultarà si té NA. En cas que tingui valors NA llavors s'imputaran amb valors de la matriu de distàncies original (backup).

```r
#Create a bakcup with the original data
airquality_data_backup <- airquality_data

#Imputation of weather data
for(n in 1:nrow(airquality_data)){

  #Sum number of NA in the row
  nulls <- sum(is.na(airquality_data[n,colnames(airquality_data)[3:7]]))

  if(nulls > 0){

    id <- airquality_data[n,]$airrmeasur_airstation_id
    date <- airquality_data[n,]$airrmeasur_datetime
    station_closer <- 1

    while((nulls > 0) & (station_closer < length(unique(stations_gencat$station_id)))){

      station_closer_id <- order_matrix[rownames(order_matrix) == id,station_closer]
      select_row_station_closer <- (airquality_data_backup$airrmeasur_airstation_id == station_closer_id

      #Check if exist data with the station and the date
      if(sum(select_row_station_closer) > 0){
```

```
        row_station_closer <- airquality_data_backup[select_row_station_closer,]

        colnames <- colnames(airquality_data)[is.na(airquality_data[n,]) & !is.na(row_station_closer)]

        data <- as.data.frame(row_station_closer[,colnames])

        colnames(data) <- colnames

        #Check if there are data that can be imuted
        if(length(data) > 0){
          #Impute the missing data with the neighbour stations
          airquality_data[n,colnames] <- data
        }
      }

      #Sum number of NA in the row
      nulls <- sum(is.na(airquality_data[n,colnames(airquality_data)[3:7]]))
      station_closer <- station_closer + 1
    }
  }
}
```

Després d'imputar els valors podem comprobar utilitzant la funció 'summary' si encara hi ha NA en el dataset.

```
summary(airquality_data)
```

```
##  airrmeasur_airstation_id airrmeasur_datetime            C6H6_ug_m3
##  ES0584A: 1224            Min.   :2019-07-01 00:00:00   Min.   : 0.6000
##  ES0691A: 1224            1st Qu.:2019-07-13 16:00:00   1st Qu.: 0.6000
##  ES0692A: 1224            Median :2019-07-26 09:00:00   Median : 0.6000
##  ES0694A: 1224            Mean   :2019-07-26 09:42:41   Mean   : 0.7681
##  ES0971A: 1224            3rd Qu.:2019-08-08 03:00:00   3rd Qu.: 0.6000
##  ES1018A: 1224            Max.   :2019-08-20 23:00:00   Max.   :24.2000
##  (Other):85221
##     CO_mg_m3        NO2_ug_m3         O3_ug_m3        PM10_ug_m3
##  Min.   :0.1000   Min.   :  1.00   Min.   :  1.00   Min.   : 4.00
##  1st Qu.:0.2000   1st Qu.:  5.00   1st Qu.: 44.00   1st Qu.:18.00
##  Median :0.2000   Median : 11.00   Median : 69.00   Median :22.00
##  Mean   :0.2467   Mean   : 16.14   Mean   : 67.24   Mean   :23.88
##  3rd Qu.:0.3000   3rd Qu.: 22.00   3rd Qu.: 90.00   3rd Qu.:29.00
##  Max.   :5.4000   Max.   :143.00   Max.   :243.00   Max.   :79.00
##
##    SO2_ug_m3
##  Min.   :  1.000
##  1st Qu.:  1.000
##  Median :  1.000
##  Mean   :  2.107
##  3rd Qu.:  3.000
##  Max.   :175.000
##
```

Com podem veure, no hi ha cap columna amb valors NA i tots estan correctament imputats.

Per estar-ne segurs podem executar la següent comanda per sumar tots els valors NA.

```r
#Calculate total number of NA in the dataframe 'airquality_data'
sum(is.na(airquality_data))
```

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

En aquest punt ja tenim el dataset preparat per ajuntar-lo amb les dades atmostferiques i posteriorment aplicar els models.

4.7) Combinar dades

Un cop ja hem preperat tots els dataset amb el format de dades correcte ja podem començar a crear nous dataset a partir de la unió de dos o més datasets.

4.7.1) Combinar municipis amb comarques

Primer de tot afegirem el nom de la regió en el dataset 'municipalities' a través del ID de regió.

```r
#Merge 'municipalities' and 'regions'
municipalities <- merge(municipalities, regions, by.x = 'muni_region_id_meteocat', by.y = 'region_id')
```

4.7.2) Combina dades meteorològiques amb els municipis

Si recordem, les dades del dataset 'weather_data' provenen de Meteocat i són prediccions de les condicions meteorològiques per a tots els municipis de catalunya. Això significa que les dades no provenen d'estacions climatològiques, per tant l'identificador d'estació és l'identificador del municipi.

En aquest cas unirem les dades dels datasets 'weather_data' i 'municipalities' a través del ID del municipi.

```r
#Merge 'weather_data' and 'municipalities'
weather_data <- merge(weather_data, municipalities, by.x = 'wmeasur_wstation_id', by.y = 'muni_id_meteo
```

Ara per a cada municipi, data i hora tenim totes les dades meteorològiques i a més tota la informació respecta a cada municipi. Això inclou el nom, codi de comarca, codi de ZQA (Zona Qualitat Aire) i les posicions GPS.

4.7.3) Combina dades de qualitat de l'aire amb les estacions

En el dataset 'airquality_data' tenim per a cada estació, data i hora les dades de contaminació del aire. El que farem en aquest cas serà afegir totes les metadades de cada estació amb l'identificador d'estació, d'aquesta forma aconseguim tenir el nom de l'estació, nom i codi del municipi, codi de la ZQA, posicions GPS i l'altura de l'estació respecta al nivell del mar.

```r
#Merge 'airquality_data' and 'stations'
airquality_data <- merge(airquality_data, stations, by.x = 'airrmeasur_airstation_id', by.y = 'station_
```

4.7.4) Combina dades de qualitat de l'aire amb dades meteorològiques

En aquest punt només falta ajuntar les dades de la qualitat de l'aire amb les dades meteorològiques. El problema és que per a cada dia i hora tenim 76 dades de la qualitat de l'aire i 64 dades climatològiques.

Per poder ajuntar els dos datasets s'ha d'assignar a cada estació de qualitat de l'aire un municipi. Així es podrà assignar a cada estació les dades climatològiques del municipi en el qual està l'estació.

Per calcular el municipi de cada estació farem una matriu de distàncies utilitzant les posicions GPS de cada municipi i de cada estació.

```
#Calculate the distance for each farmer to all the weather stations
stations_muni_matrix <- as.data.frame(geosphere::distm(stations_gencat[,c("station_lat","station_long")]
                                          municipalities[,c("muni_lat_meteocat","muni_long_meteocat")]))

#Set column names (ID municipalities)
colnames(stations_muni_matrix) <- municipalities$muni_id_meteocat

#Set row names (ID air quality stations)
row.names(stations_muni_matrix) <- stations_gencat$station_id
```

En aquest punt tenim una matriu anomenada 'stations_muni_matrix' amb tantes files com estacions de qualitat de l'aire hi han i tantes columnes com municipis hi ha a Catalunya.

El pròxim pas és calcular per a cada fila (estació) el municipi més pròxim.

```
#Create new matrix with 1 column
airquality_muni <- data.frame(matrix(ncol = 1, nrow = 0), stringsAsFactors = FALSE)

#Calculate for each air quality station the most closer municipality
for(i in 1:nrow(stations_muni_matrix)){
  airquality_muni <- rbind(airquality_muni, names(sort(stations_muni_matrix[i,]))[1], stringsAsFactors =
}

#Rename column names
colnames(airquality_muni) <- "muni_id"

#Rename row names
row.names(airquality_muni) <- stations_gencat$station_id

#Remove stations matrix dataframe
remove(stations_muni_matrix)
```

En aquest punt ja tenim per a cada estació de qualitat de l'aire el codi del municipi més pròxim, per tant ja podrem associar a cada estació les seves dades meteorològiques.

Un exemple de l'assignació entre les estacions de qualitat de l'aire i els municipis:

```
head(airquality_muni)
```

```
##         muni_id
## ES0584A  081252
## ES0691A  081944
## ES0692A  081017
## ES0694A  082634
## ES0971A  081960
## ES1018A  082798
```

Però algunes estacions estan molt pròximes entre sí i pot ser que més de una estació tingui assignat el mateix municipi. Per comprobar-ho ho podem fer amb la comanda 'table'.

```
table(airquality_muni$muni_id)
```

```
##
## 080155 080193 080229 080749 080771 080898 080961 081000 081017 081022
##      1      4      1      1      2      1      1      1      2      1
## 081120 081136 081141 081213 081249 081252 081379 081574 081691 081846
##      1      1      1      1      1      2      1      1      2      1
## 081878 081944 081960 082021 082055 082457 082515 082520 082592 082606
```

```
##      1      2      1      1      1      1      1      1      1      1
## 082634 082798 082830 083015 083054 083073 170010 170139 170792 171143
##      2      1      1      1      1      1      1      1      1      1
## 171254 250518 251193 251207 251728 251961 252094 430043 430056 430136
##      1      1      1      1      1      1      1      1      1      2
## 430141 430167 430445 430477 430607 430640 430666 430705 431188 431233
##      1      1      1      2      1      1      1      1      1      1
## 431482 431628 431711 439076
##      1      1      1      2
```

Com podem veure, efectivament hi ha municipis que tenen més de una estació assignada. Això significarà que pels diferents valors que donin les estacions del mateix municipi aquestes tindràn les mateixes dades meteorològiques. Aquest problema ja es solucionarà posteriorment on s'agragaran les dades a nivell de municipi.

Utilitzant la taula creada previament podem afegir una nova columna en el dataset 'airquality_data' amb el codi del municipi més pròxim. En aquest cas ja existeix una columna anomenada 'station_muni_id' però en realitat aquesta columna fa referència a la ZQA per tant es canviarà el nom d'aquesta columna.

```
#Rename column 'station_muni_id' and 'station_muni_name'
airquality_data$station_zqa_id <- airquality_data$station_muni_id
airquality_data$station_zqa_name <- airquality_data$station_muni_name

#Assign for each air quality station the most closer municipality
airquality_data$station_muni_id <- airquality_muni[airquality_data$airrmeasur_airstation_id,]

#Remove variable 'station_muni_name'
airquality_data$station_muni_name <- NULL

#Remove 'airquality_muni' because it's not needed anymore
remove(airquality_muni)
```

En aquest punt ja podem unir els dos grans datasets en un únic a on tindrem les dades de qualitat de l'aire i les dades meteorològiques.

```
#Merge 'airquality_data' and 'weather_data'
full_data <- merge(airquality_data, weather_data,
                   by.x = c("station_muni_id","airrmeasur_datetime"),
                   by.y = c("wmeasur_wstation_id","wmeasur_datetime"))
```

5) Analitzar i transformar les dades

Amb el dataset 'full_data' ja tenim totes les dades que necessitem per començar a realitzar anàlisis, transformar i crear noves variables a partir de les que ja tenim.

Primer de tot observarem les variables que tenim i si hi han NA.

```
summary(full_data)
```

```
##  station_muni_id    airrmeasur_datetime           airrmeasur_airstation_id
##  Length:92565       Min.   :2019-07-01 00:00:00   ES0584A: 1224
##  Class :character   1st Qu.:2019-07-13 16:00:00   ES0691A: 1224
##  Mode  :character   Median :2019-07-26 09:00:00   ES0692A: 1224
##                     Mean   :2019-07-26 09:42:41   ES0694A: 1224
##                     3rd Qu.:2019-08-08 03:00:00   ES0971A: 1224
##                     Max.   :2019-08-20 23:00:00   ES1018A: 1224
##                                                   (Other):85221
##    C6H6_ug_m3         CO_mg_m3        NO2_ug_m3         O3_ug_m3
##  Min.   : 0.6000   Min.   :0.1000   Min.   :  1.00   Min.   :  1.00
##  1st Qu.: 0.6000   1st Qu.:0.2000   1st Qu.:  5.00   1st Qu.: 44.00
##  Median : 0.6000   Median :0.2000   Median : 11.00   Median : 69.00
##  Mean   : 0.7681   Mean   :0.2467   Mean   : 16.14   Mean   : 67.24
##  3rd Qu.: 0.6000   3rd Qu.:0.3000   3rd Qu.: 22.00   3rd Qu.: 90.00
##  Max.   :24.2000   Max.   :5.4000   Max.   :143.00   Max.   :243.00
##
##    PM10_ug_m3      SO2_ug_m3
##  Min.   : 4.00   Min.   :  1.000
##  1st Qu.:18.00   1st Qu.:  1.000
##  Median :22.00   Median :  1.000
##  Mean   :23.88   Mean   :  2.107
##  3rd Qu.:29.00   3rd Qu.:  3.000
##  Max.   :79.00   Max.   :175.000
##
##                             station_name                   station_user_id
##  Agullana                         : 1224   Generalitat de Catalunya:92565
##  Alcanar                          : 1224
##  Alcover                          : 1224
##  Amposta                          : 1224
##  Badalona                         : 1224
##  Barber\303\240 del Vall\303\250s: 1224
##  (Other)                          :85221
##   station_lat     station_long     station_alt
##  Min.   :40.55   Min.   :0.2884   Min.   :   3.0
##  1st Qu.:41.22   1st Qu.:1.1930   1st Qu.:  34.0
##  Median :41.42   Median :2.0074   Median :  90.0
##  Mean   :41.46   Mean   :1.7478   Mean   : 208.2
##  3rd Qu.:41.60   3rd Qu.:2.1883   3rd Qu.: 238.0
##  Max.   :42.41   Max.   :3.2129   Max.   :1570.0
##
##          station_last_update station_type_location station_type_source
##  2019-09-04 18:00:00: 2007    Length:92565          Length:92565
##  2019-09-04 19:00:00:13464    Class :character      Class :character
##  2019-09-04 20:00:00:77094    Mode  :character      Mode  :character
##
##
```

```
##
##
##  station_zqa_id                                    station_zqa_name
##  1      :22032   Barcelona                                 : 9792
##  2      :15894   Tarragona                                 : 4896
##  15     :12240   Vandell\303\262s i l'Hospitalet de l'Infant: 3672
##  4      :11016   Montcada i Reixac                         : 2448
##  8      : 7344   Sant Vicen\303\247 dels Horts             : 2448
##  3      : 5679   el Prat de Llobregat                      : 2448
##  (Other):18360   (Other)                                   :66861
##  wmeasur_predict    EST_CEL          HUM_REL          PREC_ACU
##  1:92565         Min.   : 1.000   Min.   : 9.715   Min.   :-0.23100
##                  1st Qu.: 1.000   1st Qu.:52.519   1st Qu.: 0.00000
##                  Median : 1.000   Median :68.628   Median : 0.00000
##                  Mean   : 5.386   Mean   :66.602   Mean   : 0.04409
##                  3rd Qu.: 4.000   3rd Qu.:82.665   3rd Qu.: 0.00000
##                  Max.   :24.000   Max.   :99.796   Max.   :71.57300
##
##       TEMP           TEMP_XAF         VENT_DIR         VENT_VELO
##  Min.   : 7.112   Min.   : 6.167   Min.   :  0.013   Min.   : 0.002
##  1st Qu.:21.543   1st Qu.:21.815   1st Qu.:116.787   1st Qu.: 1.115
##  Median :24.344   Median :24.685   Median :175.997   Median : 2.252
##  Mean   :24.315   Mean   :24.598   Mean   :181.444   Mean   : 2.375
##  3rd Qu.:27.000   3rd Qu.:27.615   3rd Qu.:248.808   3rd Qu.: 3.305
##  Max.   :41.237   Max.   :38.788   Max.   :359.999   Max.   :14.487
##
##  muni_region_id_meteocat muni_id_gencat  muni_name_meteocat
##  Min.   : 1.00           008019 : 4896   Length:92565
##  1st Qu.:11.00           008077 : 2448   Class :character
##  Median :17.00           008101 : 2448   Mode  :character
##  Mean   :21.03           008125 : 2448
##  3rd Qu.:36.00           008169 : 2448
##  Max.   :41.00           008194 : 2448
##                          (Other):75429
##  muni_name_gencat   muni_region_id_gencat muni_lat_meteocat
##  Length:92565       Min.   : 1.000        Min.   :40.54
##  Class :character   1st Qu.: 2.000        1st Qu.:41.22
##  Mode  :character   Median : 4.000        Median :41.42
##                     Mean   : 5.592        Mean   :41.46
##                     3rd Qu.: 8.000        3rd Qu.:41.61
##                     Max.   :15.000        Max.   :42.41
##
##  muni_long_meteocat muni_lat_gencat muni_long_gencat muni_station_id
##  Min.   :0.2853     Min.   :40.54   Min.   :0.2843   ES1135A:13464
##  1st Qu.:1.1790     1st Qu.:41.22   1st Qu.:1.1881   ES2034A: 9792
##  Median :2.0097     Median :41.42   Median :2.0028   ES1851A: 8568
##  Mean   :1.7464     Mean   :41.46   Mean   :1.7477   ES1125A: 7326
##  3rd Qu.:2.1878     3rd Qu.:41.61   3rd Qu.:2.1821   ES1815A: 6903
##  Max.   :3.2073     Max.   :42.41   Max.   :3.2026   ES1225A: 4896
##                                                      (Other):41616
##                   region_name
##  Baix Llobregat          :13464
##  Barcelon\303\250s       :12240
##  Vall\303\250s Occidental: 9792
```

```
##  Tarragon\303\250s       : 7344
##  Vall\303\250s Oriental  : 6102
##  Baix Camp               : 3672
##  (Other)                 :39951
```

Si volem comprobar si hi ha valors NA ho podem fer de la següent manera.

```
sum(is.na(full_data))
```

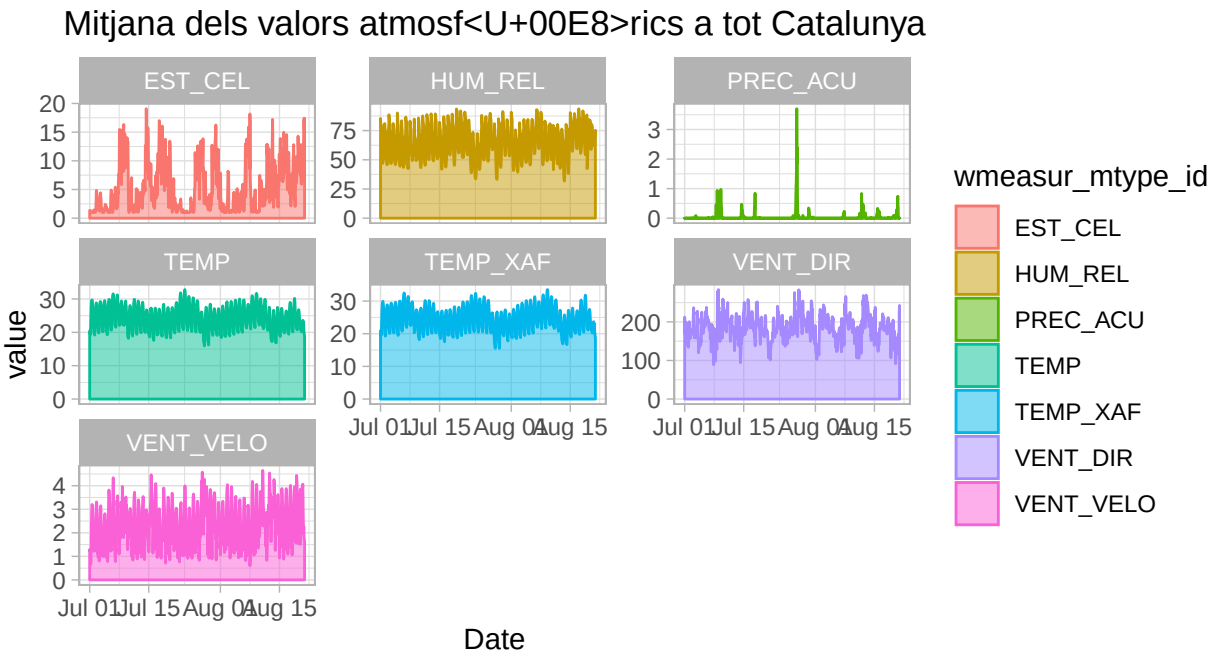
```
## [1] 0
```

Com podem veure no hi ha cap valor NA, per tant tenim les 3517470 dades correctament imputades sense cap NA.

5.1) Analitzar les dades meteorològiques

Primer de tot començarem amb l'anàlisi de les dades meteorològiques i climatològiques. En el següent gràfic es mostra la mitjana de totes les dades meteorològiques agragades per tipus de mesura, dia i hora durant tot el període de les dades descarregades.

```
ggplot(weather %>% group_by(wmeasur_datetime,wmeasur_mtype_id) %>% summarise(value = mean(wmeasur_value)
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_area(aes(color = wmeasur_mtype_id, fill = wmeasur_mtype_id), alpha = 0.5, position = position_do
  facet_wrap(~wmeasur_mtype_id, scales = "free_y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  Plot_AddFooter() +
  Plot_SetTheme("light") +
  Plot_SetPosFotter("center") +
  Plot_AddTitle("Mitjana dels valors atmosfèrics a tot Catalunya") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

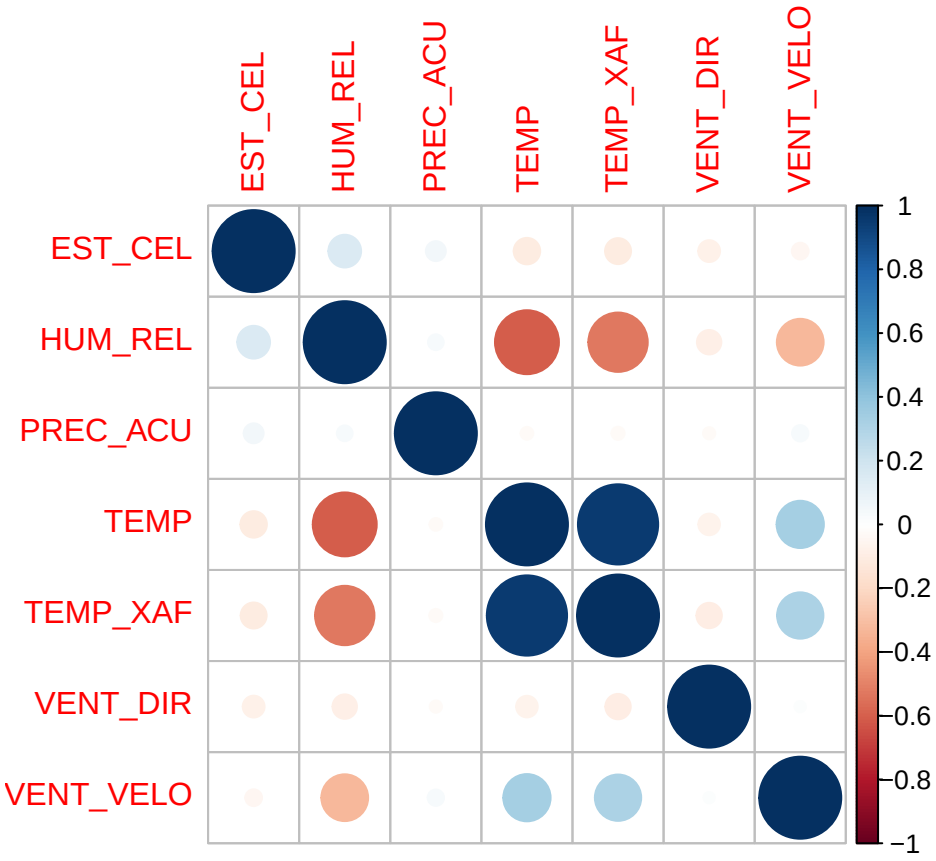
Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_mitjana_valors_atmosfèrics.png", width = 14, height = 8, dpi = 150, units = "in", devi
```

Si volem veure la correlació entre les dades climatològiques ho podem fer amb una gràfica de correlació.

```
data_cor <- cor(full_data %>% select("EST_CEL","HUM_REL","PREC_ACU","TEMP","TEMP_XAF","VENT_DIR","VENT_
corrplot::corrplot(data_cor)
```



Amb la matriu de correlació es pot veure en funció del color si la correlació és -1 (inversament proporcional), 0 (no tenen relació) i 1 (proporcionals), i també el grau de correlació va en funció de la mida del cercla, per tant com més gran és el cercla significarà que hi ha molta correlació entre les dos variables.

Com podem veure hi han alguns components que estan correlacionats entre les dades meteorològiques. Les dades més correlacionades són TEMP i TEMP_XAF que estan fortament correlacionades positivament, però no té molt de secret ja que les dos són pràcticament el mateix.

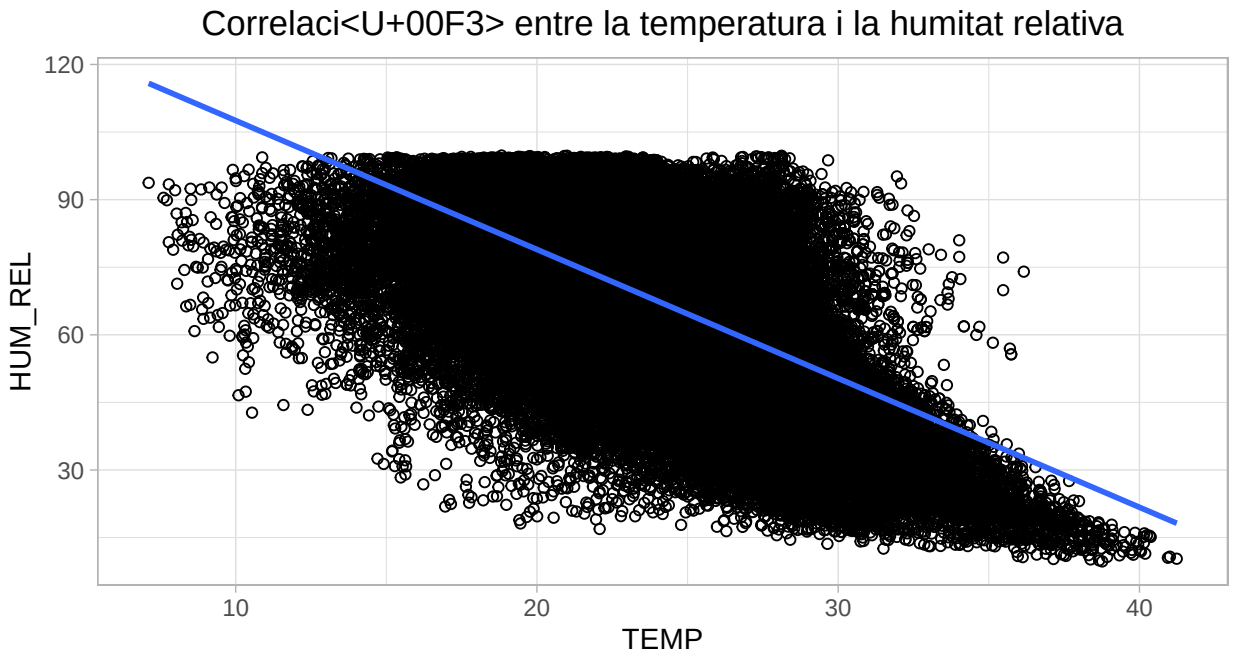
La segona correlació més gran és entre la temperatura i la humitat relativa que estat fortament correlacionades negativament.

```
cor(full_data$TEMP,full_data$HUM_REL)
```

```
## [1] -0.6078011
```

Si mostrem una gràfica d'aquestes dos variables podrem veure la coorelació entre elles.

```
ggplot(full_data, aes(x = TEMP, y = HUM_REL)) +
  geom_point(shape=1) +
  geom_smooth(method = "lm") +
  Plot_SetTheme("light") +
  Plot_AddTitle("Correlació entre la temperatura i la humitat relativa") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_cor_humitat_temp.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Com podem veure es veu clarament que quan la temparatura s'incrementa, sobretot per sobre dels 25 graus és quan la humitat decreix més. A la gràfica també s'ha afegit una línia blava que representa la regressió linial entre aquestes dos variables.

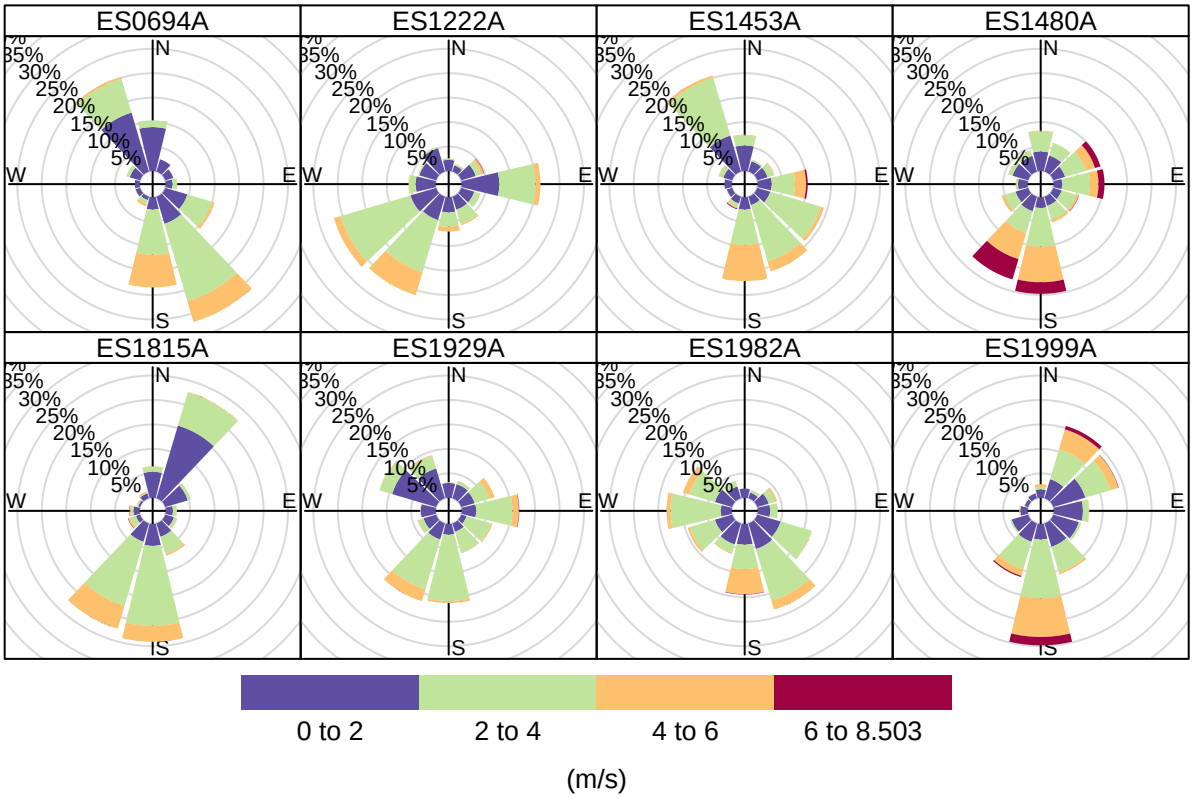
També estàn correlacionades la temparatura i la velocitat del vent. A la realitat també ho està la temparatura i la direcció del vent, ja que el vent del sud és més calid normalment. El problema és que la variable VENT_DIR tot i ser numèrica no és una variable continua sinò que és una variable circular, ja que 360 i 0 són casi el mateix. El fet de que sigui una variable circular fa que no es pugui veure la correlació utilitzant els mètodes tradicionals com el mètode de Pearson. En aquests casos es poden utilitzar altres mètodes com ara Spearman's rank correlation coefficient o Kendall rank correlation coefficient.

A les següents gràfiques podrem veure de forma visual la correlació entre el vent i la temperatura. Per fer-ho primer de tot escollirem de forma aleatoria 8 estacions de les 57 estacions de qualitat d'aire que hi ha.

```
#Get a sample of 6 stations random
sample_stations <- unique(full_data$airrmeasur_airstation_id)[sample(1:length(unique(full_data$airrmeasu
```

A la següent gràfica es podrà veure una rosa dels vents amb la direcció i velocitat del vent de les 8 estacions que s'han escollit.

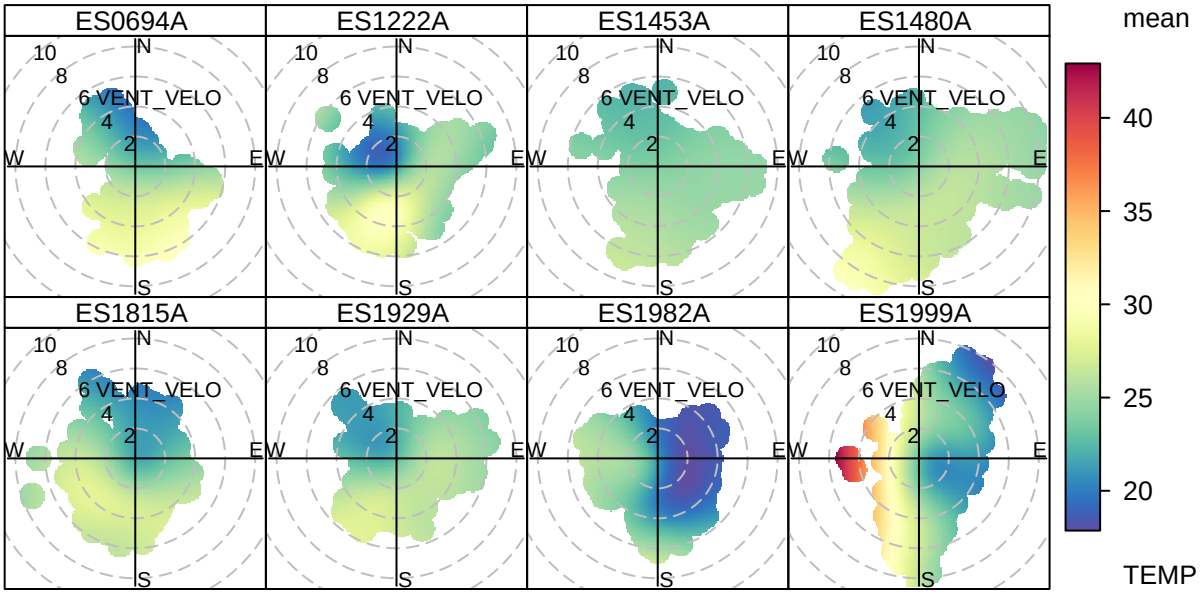
```
#Plot a wind rose with the wind speed and wind direction
openair::windRose(full_data %>% filter(airrmeasur_airstation_id %in% sample_stations),
                  ws = "VENT_VELO", wd = "VENT_DIR", type = "airrmeasur_airstation_id", auto.text= FALS
```



Si analitzem les roses de vents, podem veure que en general la tendència és que el vent prové més del sud i també té més velocitat el que ve del sud.

A la següent gràfica representarem la direcció i velocitat del vent i també amb la temperatura.

```
#Plot a polar plot with the wind speed, wind direcction and the temperature
openair::polarPlot(full_data %>% filter(airrmeasur_airstation_id %in% sample_stations), x = "VENT_VELO"
```



Com es pot veure d'aquesta mostra de 8 estacions, en general el vent amb molta velocitat i que ve del sud i també una mica del oest és molt més càlid que el vent del nort. Això ja és completament lògic ja que el vent del sud prové d'Àfrica i és molt més càlid.

5.1.1) Transformar la direcció del vent en una variable categòrica

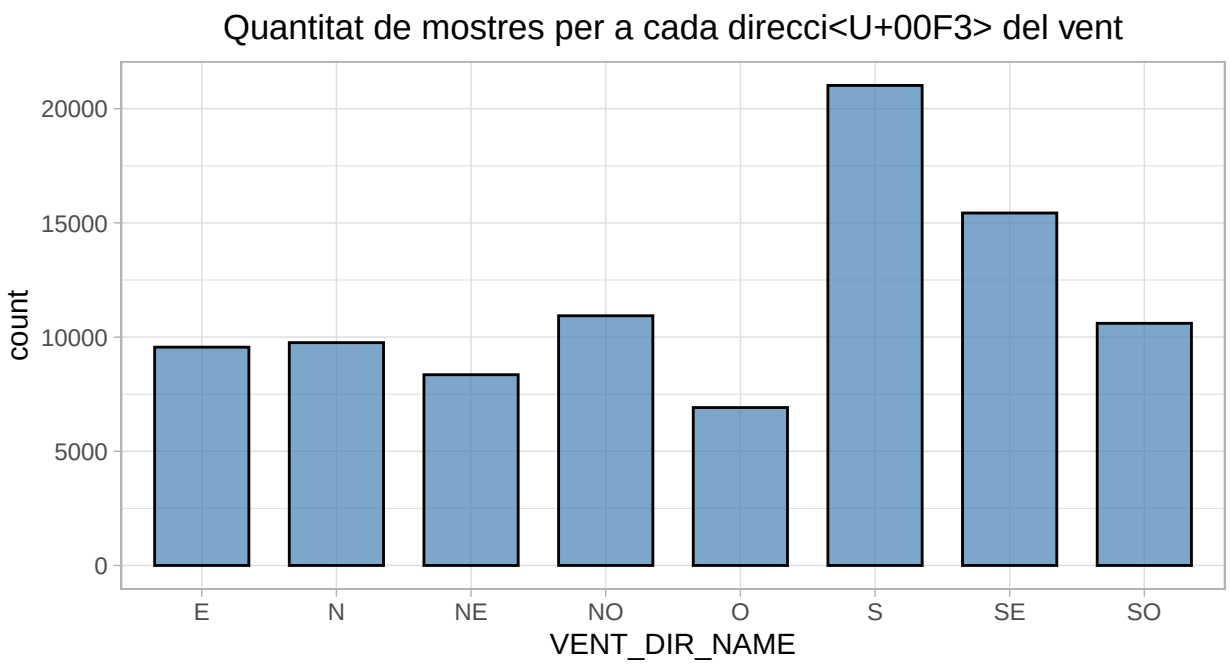
Com ja s'ha comentat, la variable VENT_DIR no és una variable contínua, sinò que és una variable circular o també podríem dir no lineal i ordinal. Per poder-la utilitzar en els models el que es farà serà convertir-la a una variable categòrica.

Per a convertir la variable VENT_DIR a una variable categòrica el que es farà serà dividir els 360 graus en 8 parts. Per tant, cada part serà un angle de 45 graus on el nord serà l'angle 0 i el 360.

```
#Convert 'VENT_DIR' to a categorical variable
full_data[(full_data$VENT_DIR <= 22.5) | (full_data$VENT_DIR >= 337.5),"VENT_DIR_NAME"] <- "N"   #Nord
full_data[(full_data$VENT_DIR > 22.5) & (full_data$VENT_DIR < 67.5),"VENT_DIR_NAME"] <- "NE"     #Nord-
full_data[(full_data$VENT_DIR >= 67.5 ) & (full_data$VENT_DIR <= 112.5),"VENT_DIR_NAME"] <- "E"  #Est
full_data[(full_data$VENT_DIR > 112.5) & (full_data$VENT_DIR < 157.5),"VENT_DIR_NAME"] <- "SE"   #Sud-E
full_data[(full_data$VENT_DIR >= 157.5) & (full_data$VENT_DIR <= 202.5),"VENT_DIR_NAME"] <- "S"  #Sud
full_data[(full_data$VENT_DIR > 202.5) & (full_data$VENT_DIR < 247.5),"VENT_DIR_NAME"] <- "SO"   #Sud-O
full_data[(full_data$VENT_DIR >= 247.5) & (full_data$VENT_DIR <= 292.5),"VENT_DIR_NAME"] <- "O"  #Oest
full_data[(full_data$VENT_DIR > 292.5) & (full_data$VENT_DIR < 337.5),"VENT_DIR_NAME"] <- "NO"   #Nord-
```

Si analitzem la freqüència de la nova variable creada 'VENT_DIR_NAME' tenim el següent gràfic de barres.

```
ggplot(full_data,aes(x = VENT_DIR_NAME)) +
  geom_bar(width=0.7, color="black", fill="steelblue", alpha=0.7) +
  Plot_SetTheme("light") +
  Plot_AddTitle("Quantitat de mostres per a cada direcció del vent") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```

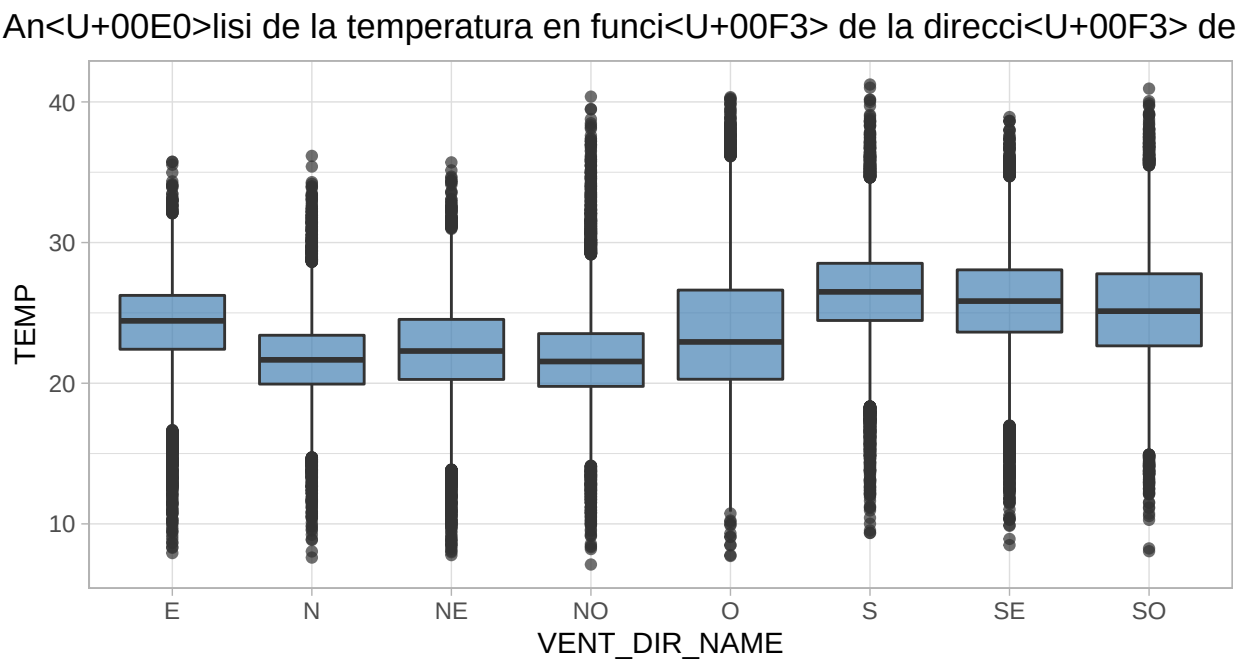


```
#Save the plot as a PNG image
ggsave("img/barplot_vent_quantitat.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Com podem veure, les freqüències més elevades són els vents del sud i en especial el vent del sud en concret que és més del doble de la resta de vents.

Per veure la distribució de freqüències de cada tipus de vent i la temperatura ho podem fer amb un diagrama de caixa.

```
ggplot(full_data,aes(x = VENT_DIR_NAME, y = TEMP)) +
  geom_boxplot(fill="steelblue", alpha=0.7) +
  Plot_SetTheme("light") +
  Plot_AddTitle("Anàlisi de la temperatura en funció de la direcció del vent") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```




```
#Save the plot as a PNG image
ggsave("img/boxplot_temp_vent.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

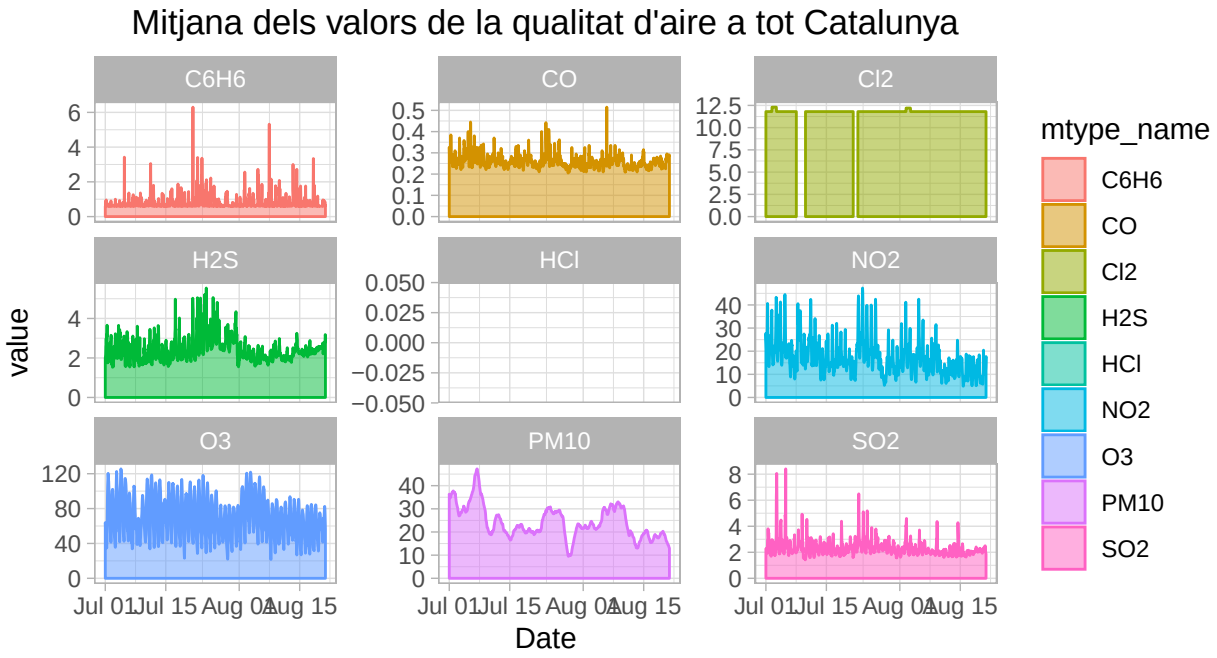
En aquest cas, podem veure que les medianes dels vents del sud són més elevades que la resta. I també podem veure que les medianes del nord són les més baixes de totes també. Amb això, també podem corrobora el fet de que els vents del sud són més càlids i els del nord més freds.

5.2) Analitzar les dades de qualitat de l'aire

Per altra banda també tenim les dades de qualitat d'aire.

En el següent gràfic es mostren la mitjana de totes les dades de qualitat d'aire agragades per tipus de mesura, dia i hora.

```
ggplot(merge(airquality %>% group_by(airrmeasur_datetime,airrmeasur_mtype_id) %>% summarise(value = mea
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_area(aes(color = mtype_name, fill = mtype_name), alpha = 0.5, position = position_dodge(0.8)) +
  facet_wrap(~mtype_name, scales = "free_y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  Plot_AddFooter() +
  Plot_SetTheme("light") +
  Plot_SetPosFotter("center") +
  Plot_AddTitle("Mitjana dels valors de la qualitat d'aire a tot Catalunya") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

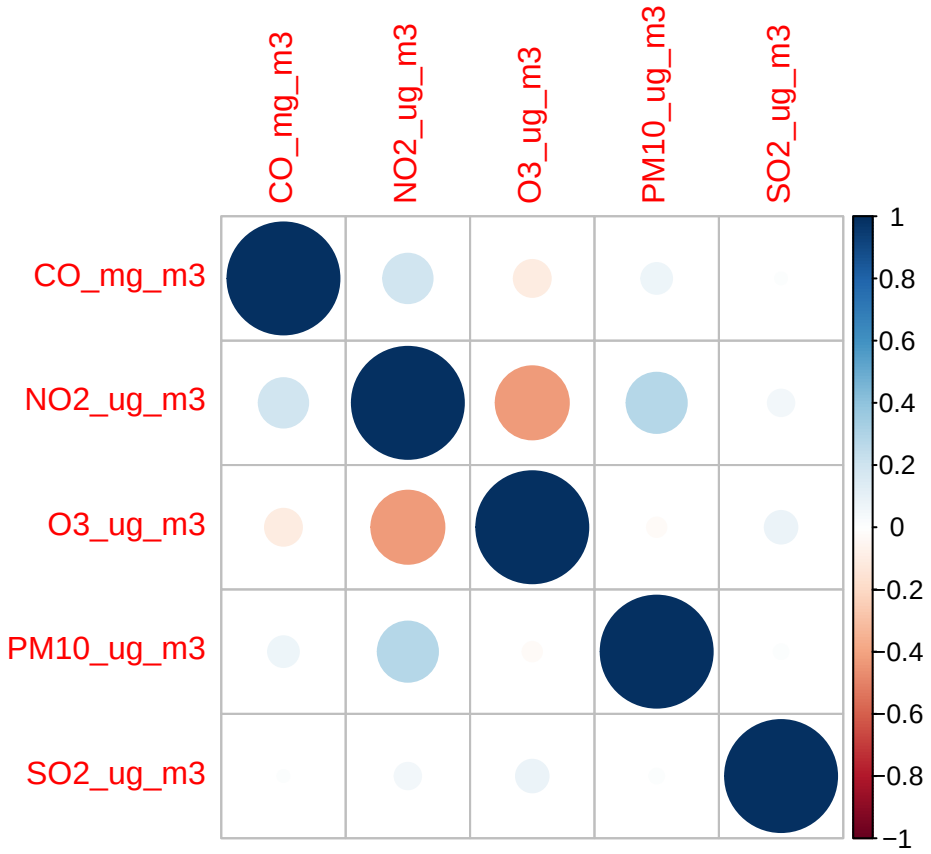
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_mitjana_valors_qualitat_aire.png", width = 14, height = 8, dpi = 150, units = "in", de
```

Com podem veure a la gràfica la variable HCl no té dades. Aquesta però durant la prepareció de les dades ja s'ha eliminat.

En aquest cas, si volem veure la correlació entre les dades de qualitat d'aire també ho podem fer amb una gràfica de correlació.

```
data_cor <- cor(full_data %>% select("CO_mg_m3","NO2_ug_m3","O3_ug_m3","PM10_ug_m3","SO2_ug_m3"), use="
corrplot::corrplot(data_cor)
```



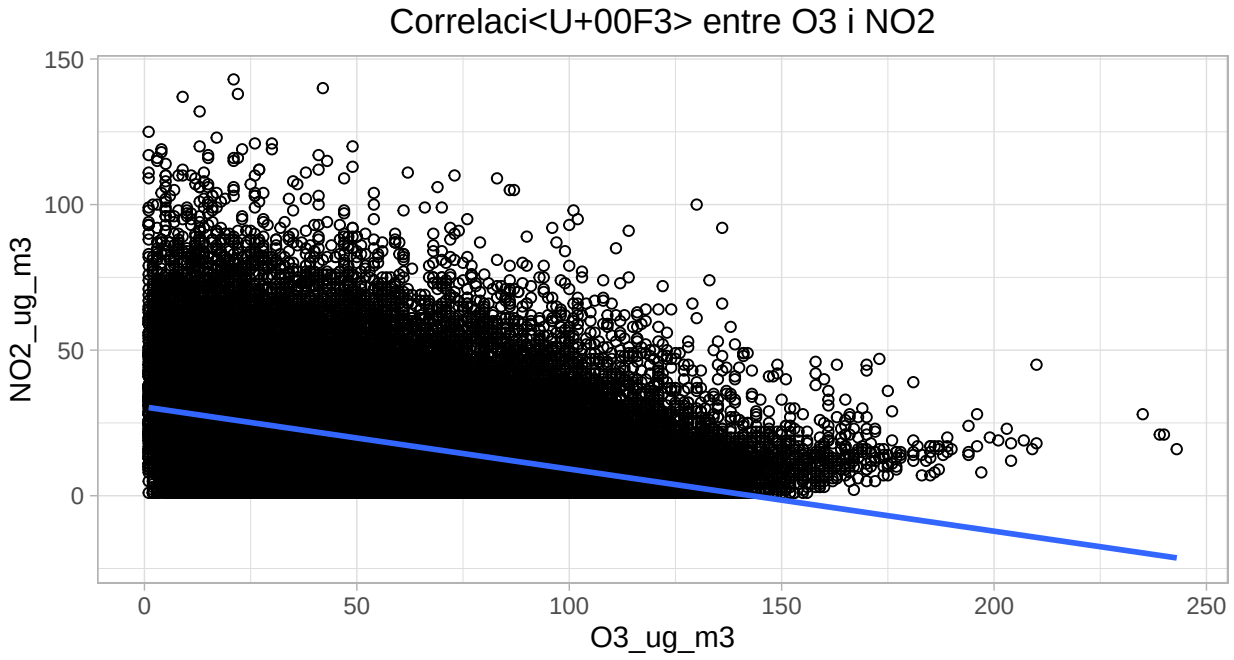
Com podem veure no hi ha tantes correlacions com en el cas de les dades meteorològiques. Tot i així, podem veure que hi ha una correlació negativa considerable entre les variables O3_ug_m3 i NO2_ug_m3.

```
cor(full_data$O3_ug_m3,full_data$NO2_ug_m3)
```

```
## [1] -0.4259517
```

Si mostrem una gràfica d'aquestes dos variables podrem veure la coorelació entre elles.

```
ggplot(full_data, aes(x = O3_ug_m3, y = NO2_ug_m3)) +
  geom_point(shape=1) +
  geom_smooth(method = "lm") +
  Plot_SetTheme("light") +
  Plot_AddTitle("Correlació entre O3 i NO2") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
ggsave("img/plot_cor_O3_NO2.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Efectivament, podem veure en el gràfic que quan O3 incrementa llavors NO2 disminueix. Igual que a l'apartat anterior, en aquest plot també s'ha afegit amb una línia blava la regressió lineal entre les dos variables.

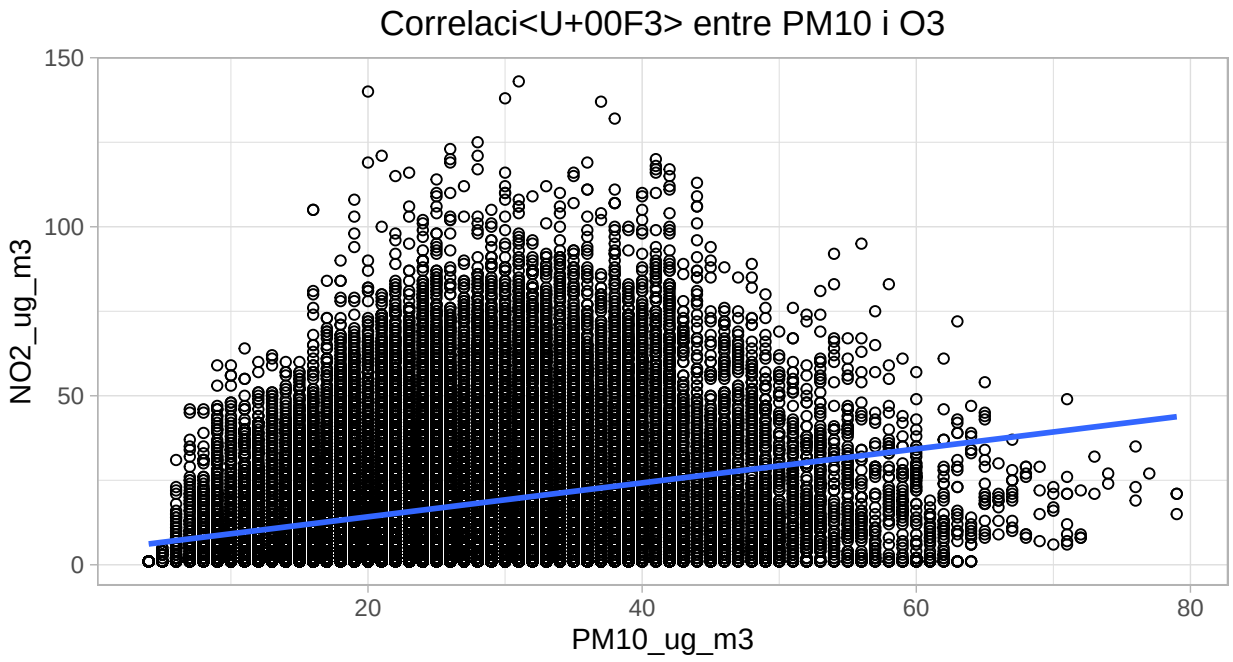
En el cas del PM10, es pot observar que només hi ha una petita correlació amb NO2 positiva.

```
cor(full_data$PM10_ug_m3,full_data$NO2_ug_m3)
```

```
## [1] 0.2886415
```

Si mostrem una gràfica d'aquestes dos variables podrem veure la coorelació entre elles.

```
ggplot(full_data, aes(x = PM10_ug_m3, y = NO2_ug_m3)) +
  geom_point(shape=1) +
  geom_smooth(method = "lm") +
  Plot_SetTheme("light") +
  Plot_AddTitle("Correlació entre PM10 i O3") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



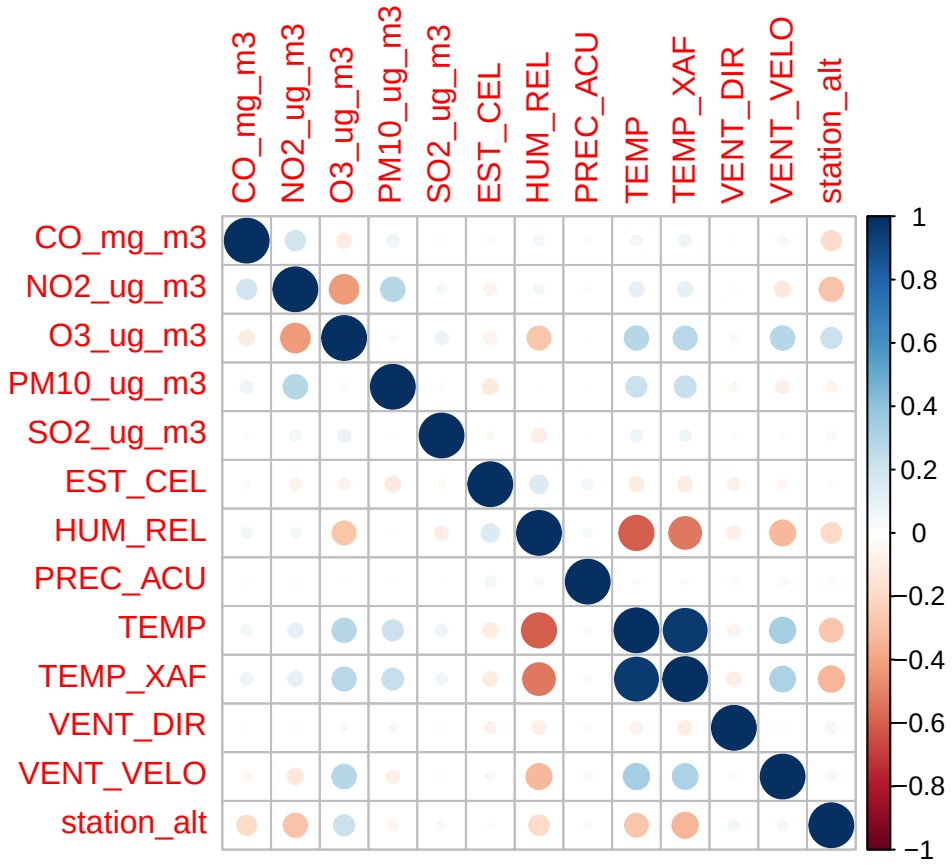

```
ggsave("img/plot_cor_PM10_O3.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Amb la gràfica es pot apreciar que realment hi ha una mica de correlació i que la tendència és positiva. Gràcies a la regressió lineal també es pot veure que el pendent de la regressió és positiu.

5.3) Analitzar la correlació entre les dades meteorològiques i la qualitat de l'aire

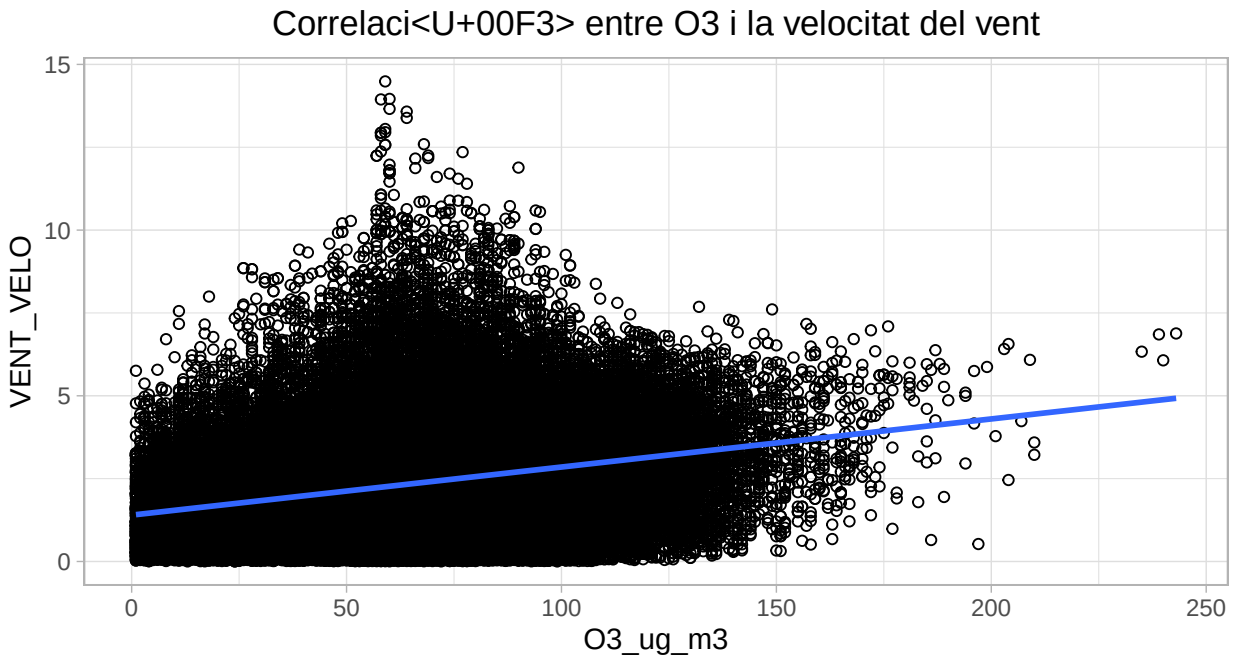
Si el que volem és veure si existeix alguna correlació entre les dades meteorològiques i les dades de qualitat d'aire ho podem fer també amb una matriu de correlació. En cas que hi hagin correlacions llavors aquestes seràn útils de cares a la creació dels models. També s'ha afegit l'altitud en la qual es troba l'estació de qualitat d'aire.

```
data_cor <- cor(full_data %>% select("CO_mg_m3","NO2_ug_m3","O3_ug_m3","PM10_ug_m3","SO2_ug_m3","EST_CE
corrplot::corrplot(data_cor)
```



Com podem veure a l'anterior taula de correlacions no hi han moltes correlacions entre les dades meteorològiques i les dades de qualitat d'aire. Per a la variable objectiu O3 la correlació més gran és amb la velocitat del vent que estan positivament correlacionades i també amb la temperatura. Si mostrem la correlació amb la velocitat del vent correlació visualment tenim el següent gràfic.

```
ggplot(full_data, aes(x = O3_ug_m3, y = VENT_VELO)) +
  geom_point(shape=1) +
  geom_smooth(method = "lm") +
  Plot_SetTheme("light") +
  Plot_AddTitle("Correlació entre O3 i la velocitat del vent") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

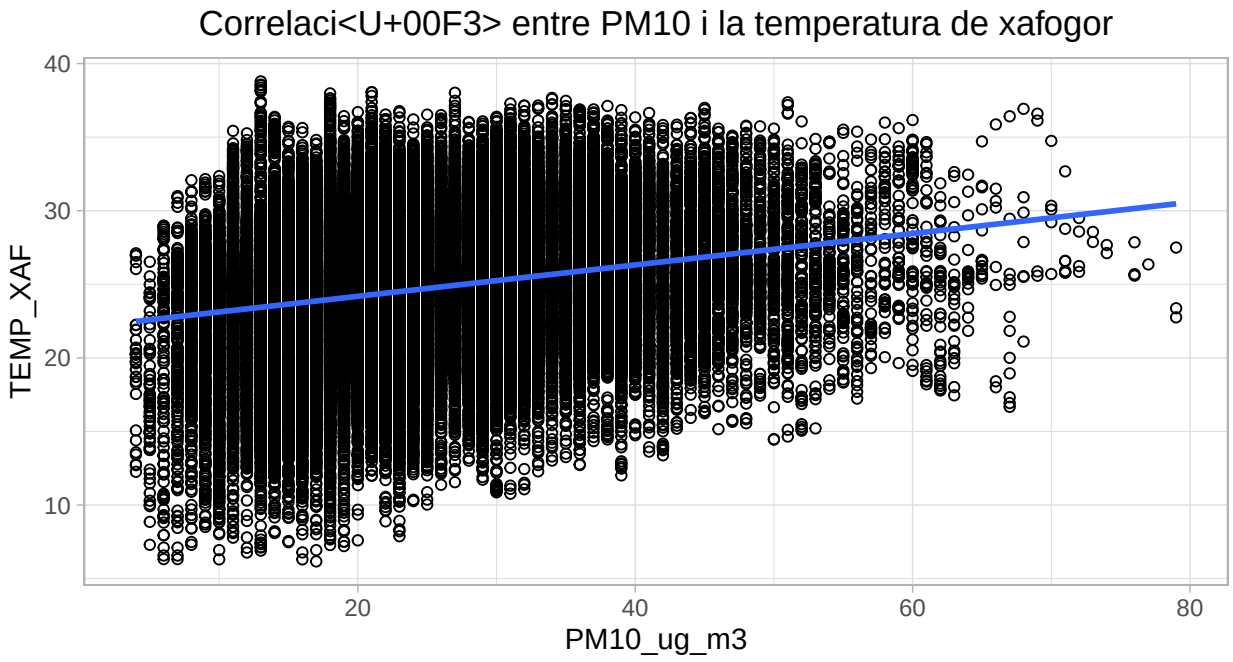
Per m..s info: https://www.respira.cat

```
ggsave("img/plot_cor_O3_veloc_vent.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Efectivament la correlació és positiva i es pot veure que a mesura que la velocitat del vent creix també ho fa O3.

L'altre variable objectiu que és PM10 no té pràcticament cap correlació amb les dades meteorològiques a excepció de la temperatura de xafogor i la temperatura, que estan una mica positivament correlacionades. Si ho mostrem visualment tenim la següent gràfica.

```
ggplot(full_data, aes(x = PM10_ug_m3, y = TEMP_XAF)) +
  geom_point(shape=1) +
  geom_smooth(method = "lm") +
  Plot_SetTheme("light") +
  Plot_AddTitle("Correlació entre PM10 i la temperatura de xafogor") +
  Plot_SetPosTitle("center") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

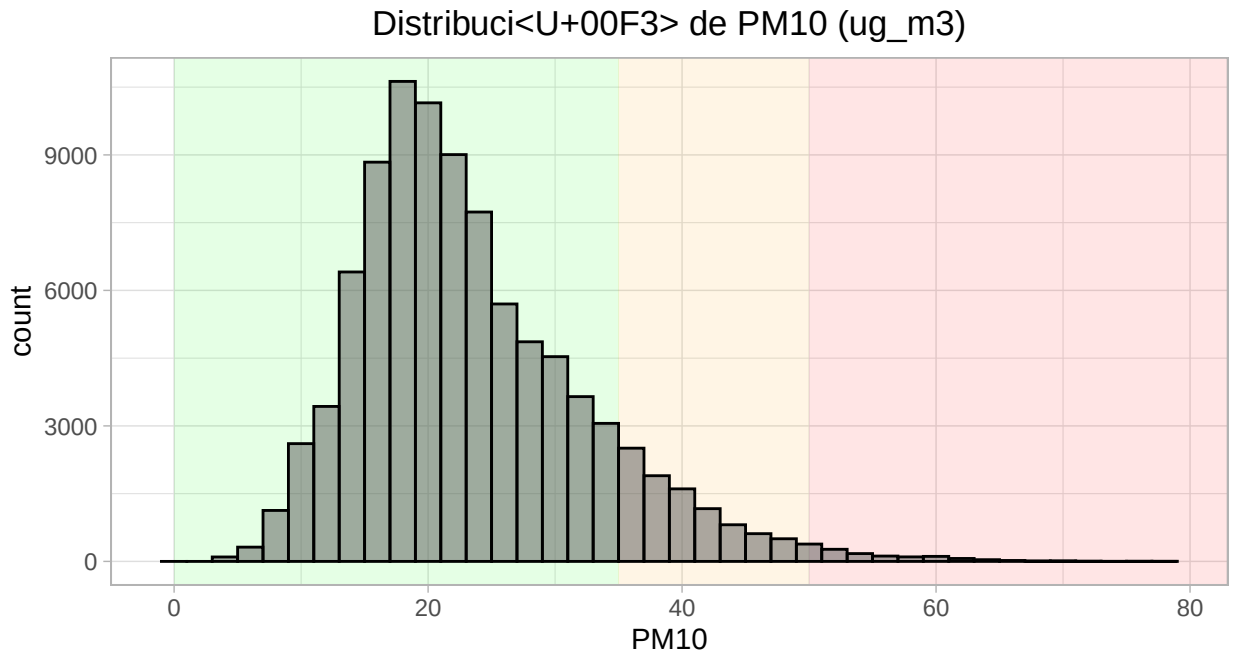
Per m..s info: https://www.respira.cat

```
ggsave("img/plot_cor_PM10_temp_xaf.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

5.4) Analitzar la variable objectiu PM10 (ug_m3)

Si volem analitzar la distribució dels diferents valors que té la variable objectiu PM10_ug_m3 ho podem fer amb un histograma:

```
ggplot(full_data, aes(x=PM10_ug_m3)) +
  Plot_SetTheme() +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=0,xmax=35, fill = "green", alpha = 0.1) +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=35,xmax=50, fill = "orange", alpha = 0.1) +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=50,xmax=Inf, fill = "red", alpha = 0.1) +
  geom_histogram(binwidth=2, alpha=0.5, color="black") +
  Plot_AddTitle("Distribució de PM10 (ug_m3)") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("PM10") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

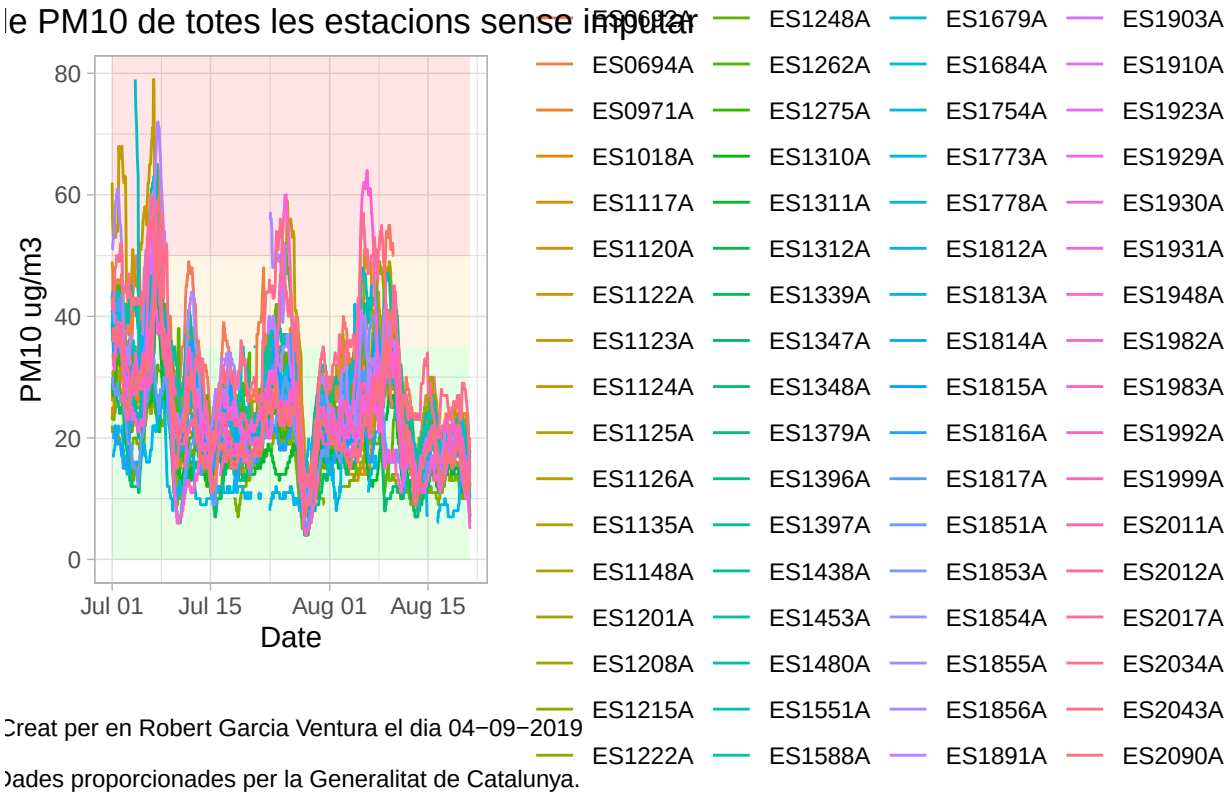
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/hist_PM10.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Com podem veure, la majoria dels valors estan dins la franja verda (considerada de bona qualitat). Posteriorment tenim la franja taronja i vermella a on la qualitat és regular i dolenta respectivament.

En el següent gràfic es representen els valors del nivell de PM10 ug/m3 de les estacions sense imputar cap valor.

```
ggplot(airquality_data_backup, aes(x = airrmeasur_datetime, y = PM10_ug_m3, color = airrmeasur_airstati
  annotate("rect",ymin=0, ymax=35, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=35, ymax=50, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_d
  annotate("rect",ymin=50, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de PM10 de totes les estacions sense imputar") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

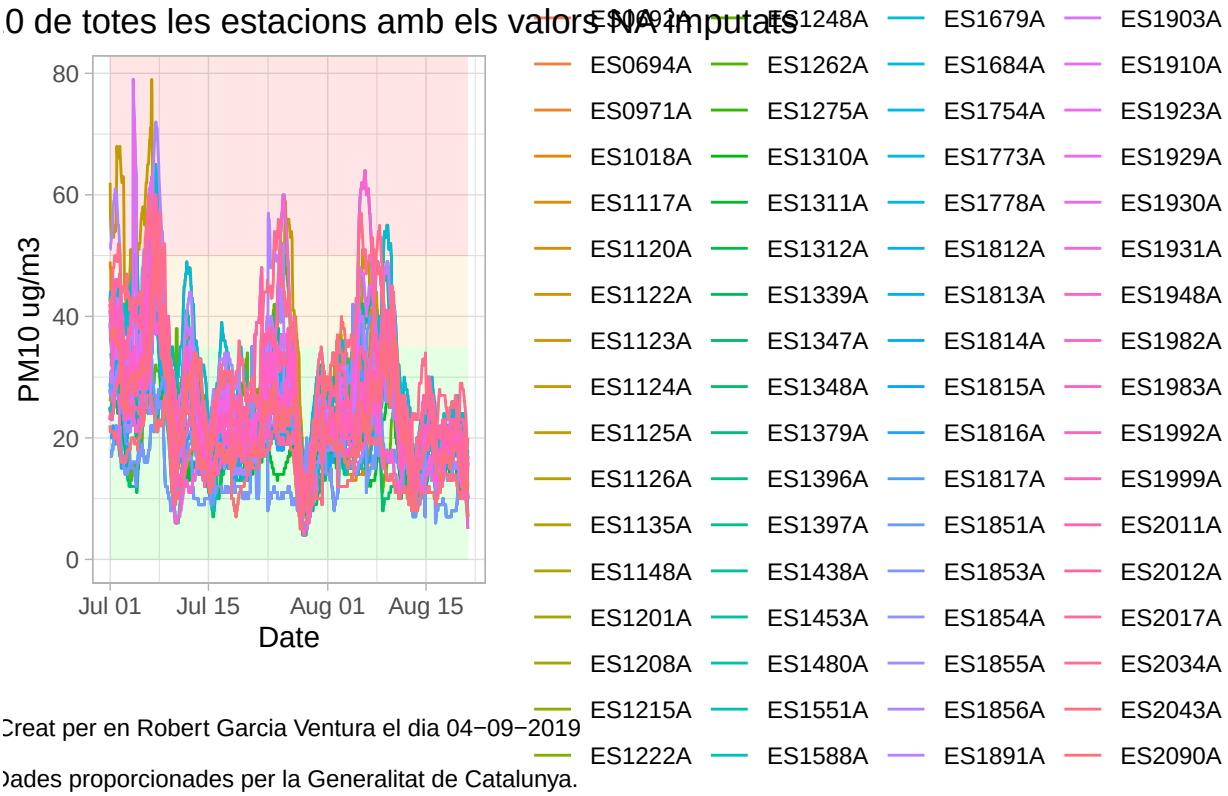
Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_PM10_sense_imputar.png", width = 14, height = 8, dpi = 150, units = "in", device='png'
```

En el següent gràfic es representen els valors del nivell de PM10 ug/m3 de les estacions amb el valors NA imputats amb els valors de les estacions més pròximes a cada una d'elles respectivament.

```
ggplot(airquality_data, aes(x = airrmeasur_datetime, y = PM10_ug_m3, color = airrmeasur_airstation_id),
  annotate("rect",ymin=0, ymax=35, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=35, ymax=50, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=50, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de PM10 de totes les estacions amb els valors NA imputats") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

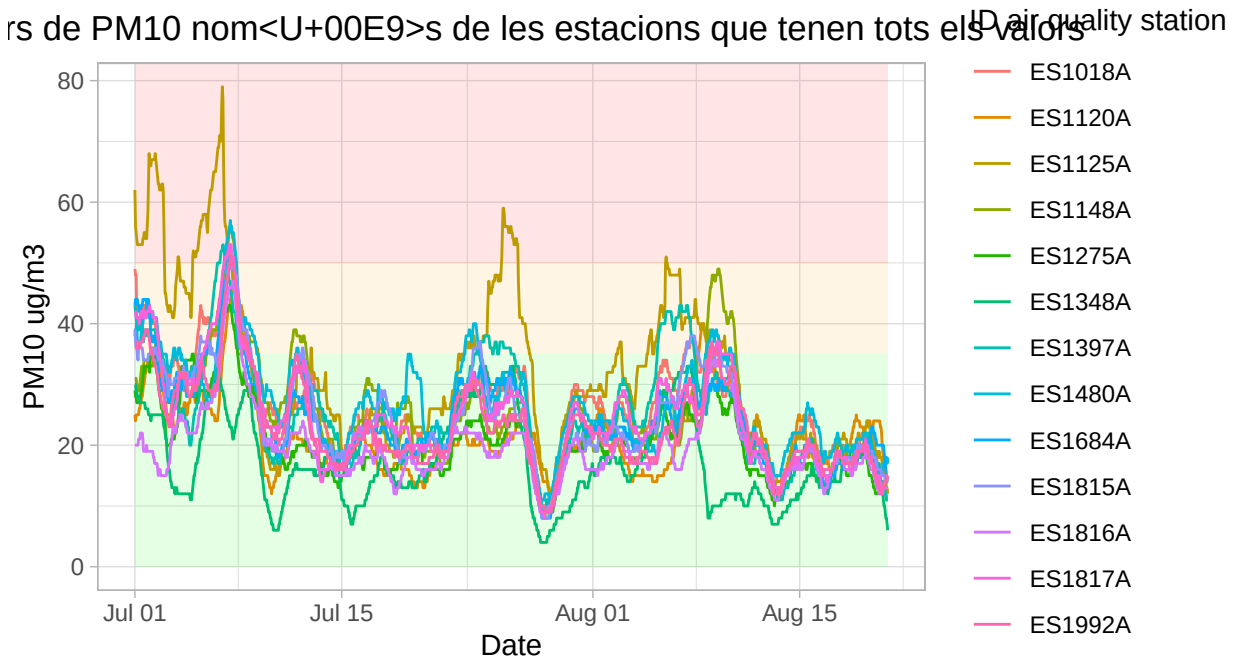
Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_PM10_imputats.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

En el següent gràfic es representen els valors del nivell de PM10 ug/m3 de les estacions que tenen totes les dades i no s'ha tingut que imputar cap valor.

```
ggplot(airquality_data %>% filter(!(airrmeasur_airstation_id %in% unique((airquality_data_backup %>% fil
  annotate("rect",ymin=0, ymax=35, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=35, ymax=50, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=50, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de PM10 només de les estacions que tenen tots els valors") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

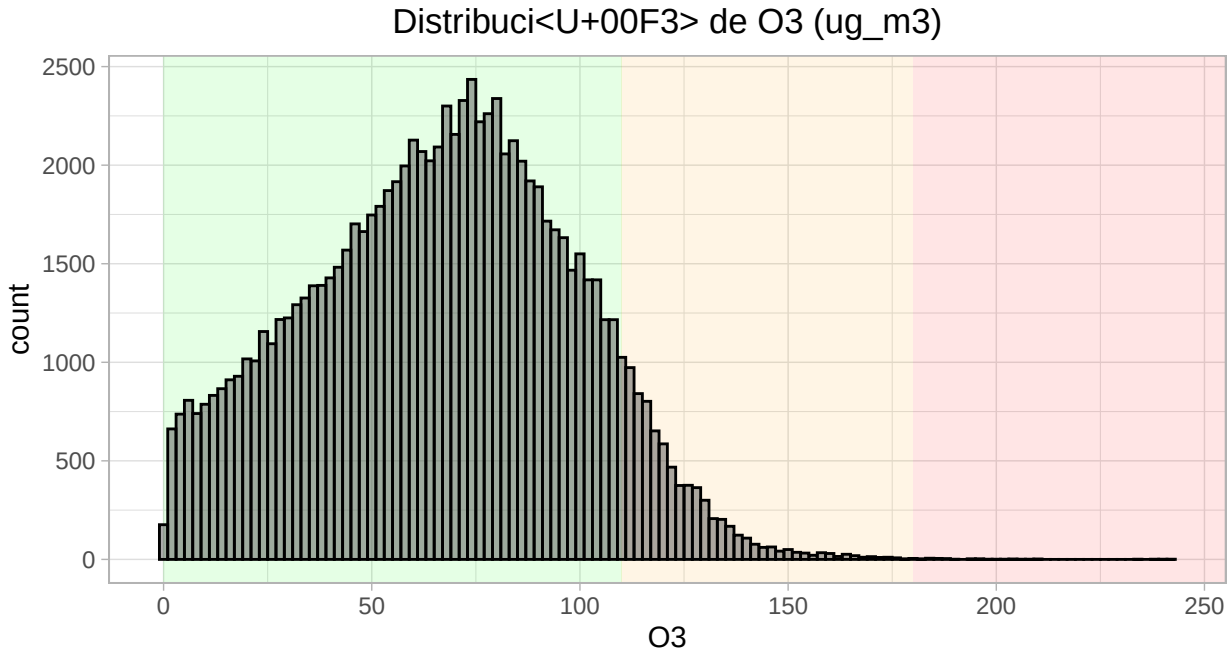
```
#Save the plot as a PNG image
ggsave("img/plot_PM10_tots_valors_sense_imputar.png", width = 14, height = 8, dpi = 150, units = "in", c
```

Si analizem les gràfiques podem veure que hi ha un patrò que segueixen la majoria de les estacions i que té unes fluctuacions al llarg dels dies que varia. En cap cas però es pot veure fàcilment un petrò de freqüència diaria o de varis dies.

5.5) Analitzar la variable objectiu O3 (ug_m3)

Si volem analitzar la distribució dels diferents valors que té la variable objectiu O3_ug_m3 ho podem fer amb un histograma:

```
ggplot(full_data, aes(x=O3_ug_m3)) +
  Plot_SetTheme() +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=0,xmax=110, fill = "green", alpha = 0.1) +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=110,xmax=180, fill = "orange", alpha = 0.1) +
  annotate("rect",ymin=-Inf, ymax=+Inf, xmin=180,xmax=Inf, fill = "red", alpha = 0.1) +
  geom_histogram(binwidth=2, alpha=0.5, color="black") +
  Plot_AddTitle("Distribució de O3 (ug_m3)") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("O3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

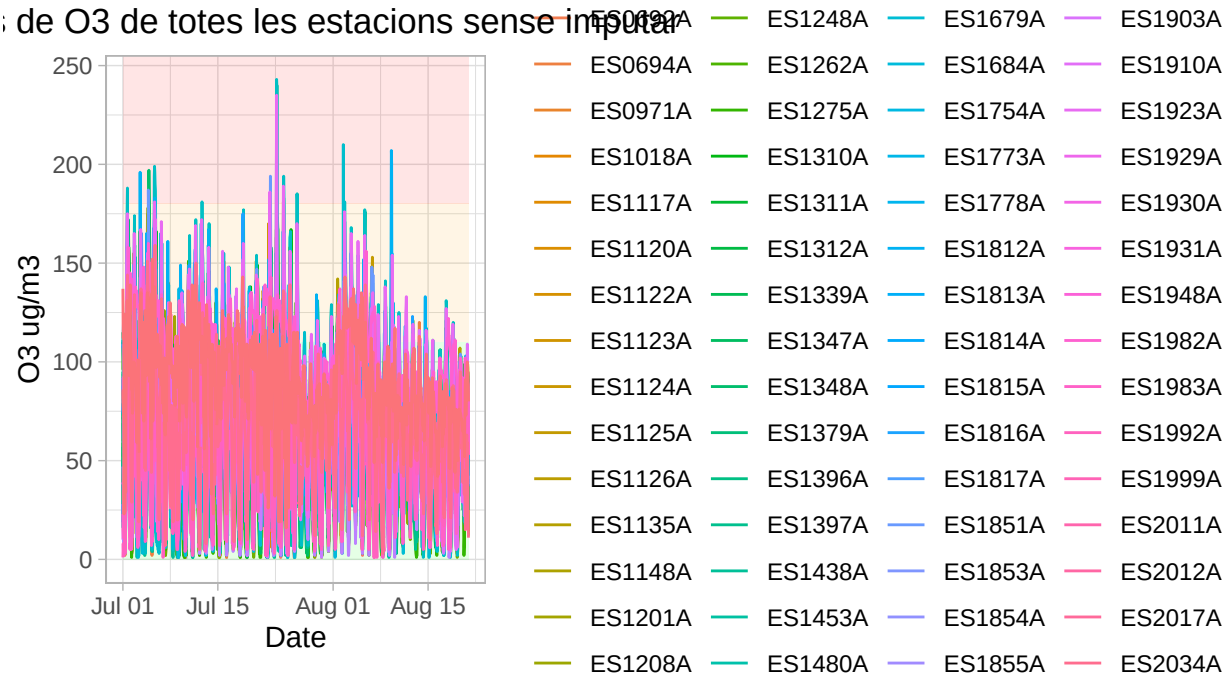
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/hist_O3.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Igual que en el cas del PM10, amb O3 també tenim la majoria dels valors a la franja verda. Posteriorment tenim la franja taronja i vermella a on la qualitat és regular i dolenta respectivament.

En el següent gràfic es representen els valors del nivell de O3 ug/m3 de les estacions sense imputar cap valor.

```
ggplot(airquality_data_backup, aes(x = airrmeasur_datetime, y = O3_ug_m3, color = airrmeasur_airstation_
  annotate("rect",ymin=0, ymax=110, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=110, ymax=180, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  annotate("rect",ymin=180, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 de totes les estacions sense imputar") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

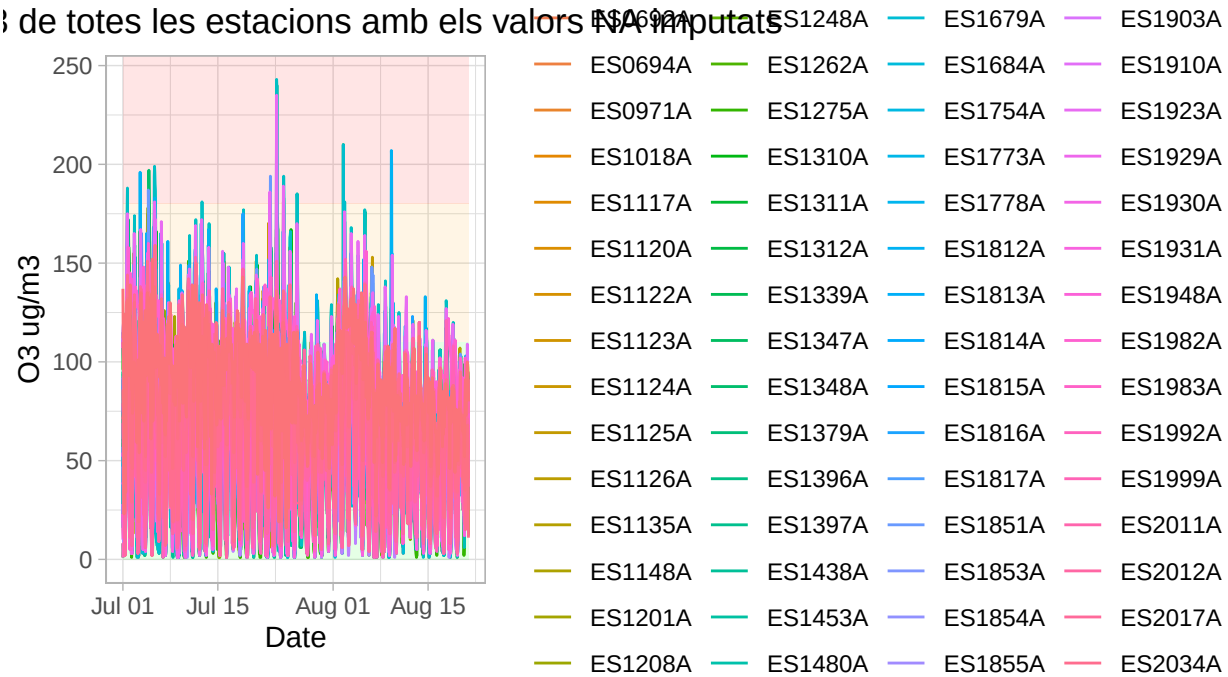
Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_O3_sense_imputar.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

En el següent gràfic es representen els valors del nivell de O3 ug/m3 de les estacions amb el valors NA imputats amb els valors de les estacions més pròximes a cada una d'elles respectivament.

```
ggplot(airquality_data, aes(x = airrmeasur_datetime, y = O3_ug_m3, color = airrmeasur_airstation_id), rm
  annotate("rect",ymin=0, ymax=110, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=110, ymax=180, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality
  annotate("rect",ymin=180, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 de totes les estacions amb els valors NA imputats") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

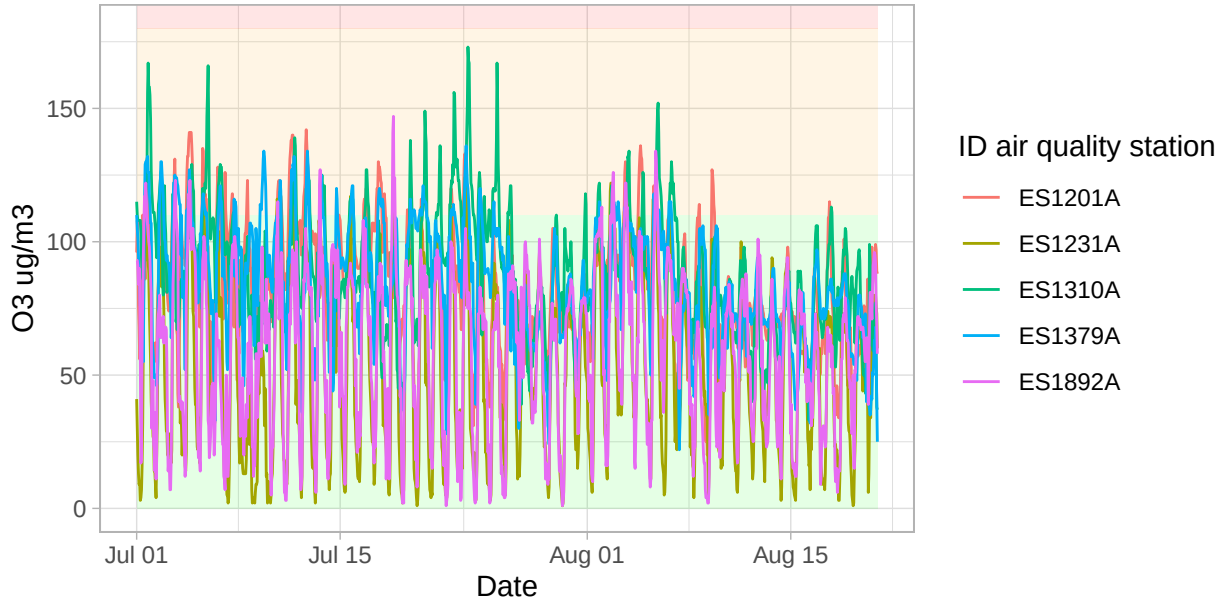
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_O3_imputats.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

En el següent gràfic es representen els valors del nivell de O3 ug/m3 de les estacions que tenen totes les dades i no s'ha tingut que imputar cap valor.

```
ggplot(airquality_data %>% filter(!(airrmeasur_airstation_id %in% unique((airquality_data_backup %>% fil
  annotate("rect",ymin=0, ymax=110, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_da
  annotate("rect",ymin=110, ymax=180, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  annotate("rect",ymin=180, ymax=Inf, xmin=min(airquality_data$airrmeasur_datetime),xmax=max(airquality_
  geom_line(na.rm=TRUE) +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 només de les estacions que tenen tots els valors") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Date") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "ID air quality station")
```

lors de O3 nom<U+00E9>s de les estacions que tenen tots els valors



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_O3_tots_valors_sense_imputar.png", width = 14, height = 8, dpi = 150, units = "in", de
```

Com podem veure, les gràfiques segueixen un patró força regular que té una freqüència diària. Durant la nit baixen els nivells de O3, llavors al matí comencen a pujar fins a finals de la tarda que tornen a baixar fins al matí següent.

6) Models lineals

En aquesta secció es crearàn dos subdatasets per poder entrenar i posteriorment testejar els dos models que es faran.

Al tractar-se de dades observades al llarg de diferents dies amb una freqüència horària es podría dir que són sèries temporals.

6.1) Dividir les dades en test i train

Primer de tot es seperarà les dades entre test i train. Al tractarse de dades temporals amb dia i hora i amb la mateixa freqüencia el que es farà serà dividir el conjunt de dades principal en dos parts. El 80% de dades inicials per al train i el 20% de dades restants pel test.

També s'ha de comentar que el dataset conté per a cada dia i hora tantes observacions com estacions de qualitat d'aire hi ha. Per tant a cada dia i hora tenim 57 observacions.

En aquest cas per aplicar els models únicament s'utilitzarà les observacions de una única estació.

```
#Good example of air quality station: ES1117A

#Create subset with data from a random one station
#full_data_station <- full_data[(full_data$airrmeasur_airstation_id == sample(unique(full_data$airrmeas

#Create subset with data from one station
full_data_station <- full_data[(full_data$airrmeasur_airstation_id == 'ES1117A'),]

#Create the train dataset (80%)
full_data_train <- full_data_station[(1:(nrow(full_data_station)*0.8)),]

#Create the test dataset (20%)
full_data_test <- full_data_station[(((nrow(full_data_station)*0.8)+1):nrow(full_data_station)),]
```

6.2) Model ANCOVA per a PM10

6.2.1) Entrenar el model ANCOVA per a PM10

El model ANCOVA és un model d'anàlisi de la covariància amb dades explicatives contínues i categòriques.

Quan estem treballant amb un model ANCOVA estem fent uns supòsits que tenen tots els models lineals. Aquests són:

- 1) Independència: Els elements de la mostra i, per tant, els residus del model, són independents entre sí.
- 2) Linealitat: La relació entre Y i X és lineal.
- 3) Normalitat: Els residus del model segueixen una distribució normal (si n $<$ 200).
- 4) Homoscedasticitat: La variància residual ha de ser constant.

Això a la pràctica no és cert, ja que les condicions climatològiques i la qualitat d'aire dels dies anteriors afectarà la qualitat d'aire del dia següent.

En aquest cas tenim com a variable independent PM10_ug_m3 i com a variables dependents totes aquelles variables que tenen un efecte a la variable independent. Com s'ha pogut veure a la gràfica de correlació els components que estan més correlacionats amb PM10 són NO2_ug_m3, O3_ug_m3, TEMP, TEMP_XAF,

VENT_VELO i VENT_DIR_NAME. Per tant, aquest components més correlacionats seran les variables dependents del model ANCOVA.

Per fer un model ANCOVA amb R s'utilitza la funció 'lm()' a on s'ha d'especificar la variable independent i les dependents.

```
ancova_PM10 <- lm(PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF + VENT_VELO + VENT_DIR_NAME, data
```

Un cop ja s'ha entrenat el model ja podem veure el resum amb la funció 'summary'.

```
summary(ancova_PM10)
```

```
##
## Call:
## lm(formula = PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF +
##     VENT_VELO + VENT_DIR_NAME, data = full_data_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6838  -4.6545  -0.9694   3.0033  22.5712
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     17.13677    3.77446   4.540 6.33e-06 ***
## NO2_ug_m3       -0.05301    0.02560  -2.070  0.03867 *
## O3_ug_m3        -0.08702    0.01163  -7.480 1.67e-13 ***
## TEMP            -0.94735    0.28850  -3.284  0.00106 **
## TEMP_XAF         1.52573    0.19954   7.646 4.99e-14 ***
## VENT_VELO       -1.54334    0.18961  -8.139 1.22e-15 ***
## VENT_DIR_NAMEN  -0.80861    1.05976  -0.763  0.44564
## VENT_DIR_NAMENE  0.12414    1.12893   0.110  0.91246
## VENT_DIR_NAMENO -0.46755    1.14147  -0.410  0.68219
## VENT_DIR_NAMEO  -2.55561    1.24575  -2.051  0.04049 *
## VENT_DIR_NAMES  -4.35677    0.99808  -4.365 1.41e-05 ***
## VENT_DIR_NAMESE -2.30295    1.02640  -2.244  0.02508 *
## VENT_DIR_NAMESO -1.66440    1.13291  -1.469  0.14212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.464 on 966 degrees of freedom
## Multiple R-squared:  0.2157, Adjusted R-squared:  0.2059
## F-statistic: 22.14 on 12 and 966 DF,  p-value: < 2.2e-16
```

Per a cada variable tenim l'estimació o també anomenat valor ajustat. També hi ha l'error o residu que és la diferència entre el valor ajustat i el valor observat. I també molt important, és la significació de cada variable. En altres paraules, com de important és la variable per predir la variable objectiu. Per poder-ho veure visualment fàcilment al costat de cada variable al final hi han unes estrelletes. Com més estrelletes significa que la variable és més important.

En general, podem veure que totes les variables contínues a excepció de la TEMP són molt importants. La variable categòrica 'VENT_DIR_NAME' la categoría més important és 'S', ve a ser el vent del sud. Això és lògic ja que el vent del sud prové d'Àfrica i normalment porta pols del desert del Sàhara.

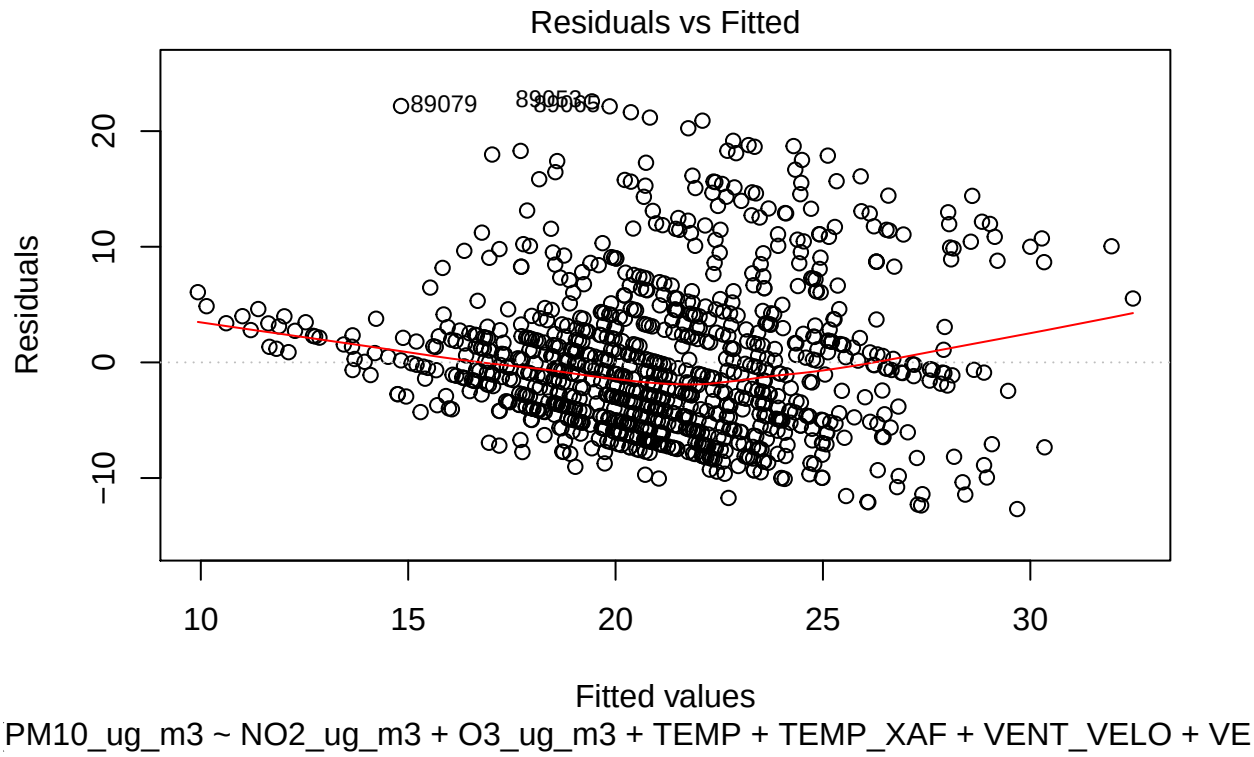
També si ens fixem amb els valors residuals en la mediana podem veure que els resultats estan una mica esbiaixats negativament. El valor de la mediana és -0.7716. Això significa que el nostre model dona resultats en la majoria dels casos amb una diferència de -0.7716. Per tant, són valors més baixos de lo real, però en general podem dir que esta força bé ja que els valors que dona són força pròxims al 0 (on 0 significa que el

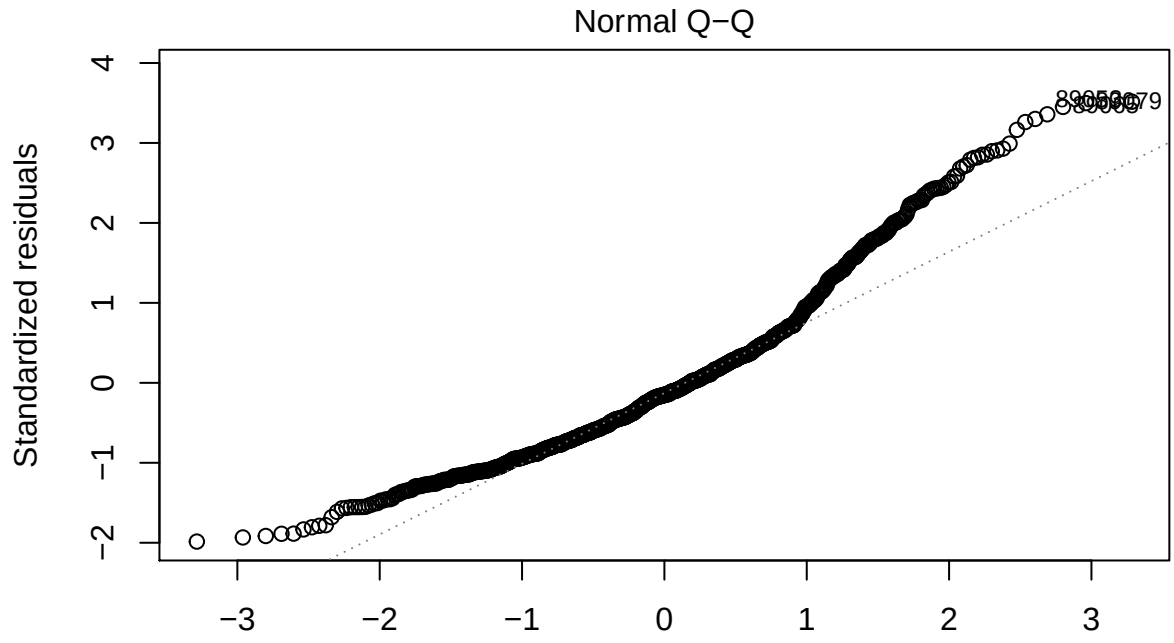
valor obtingut pel model i el valor real són iguals).

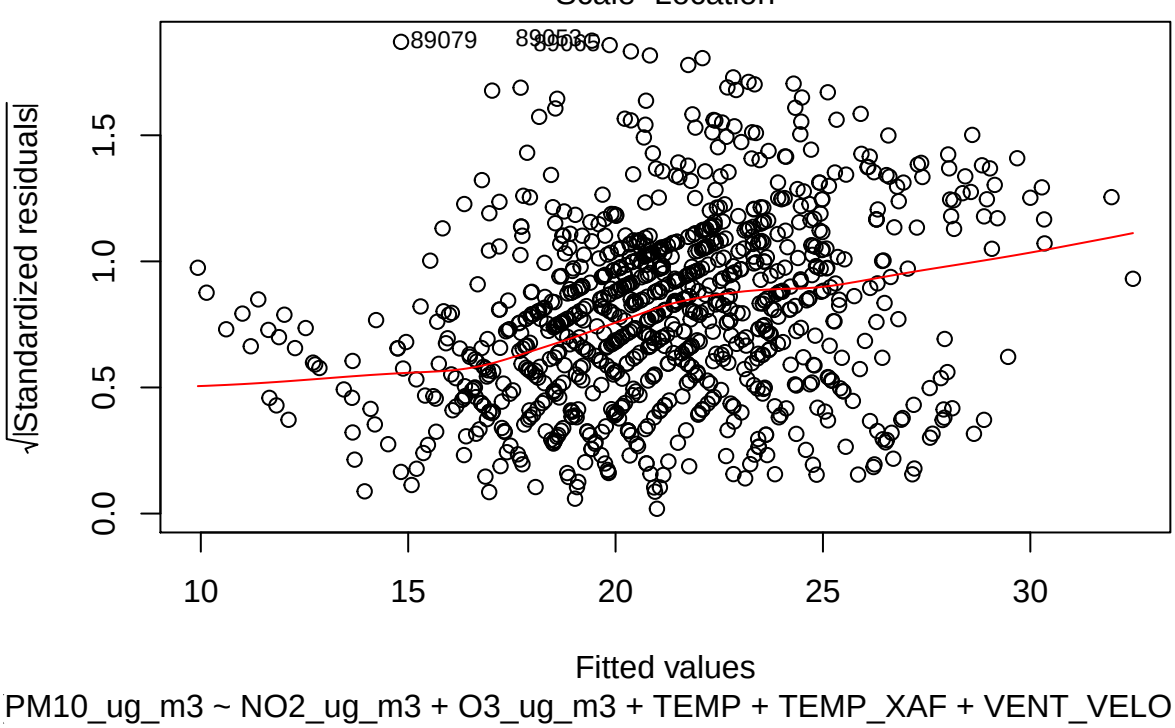
6.2.2) Provar i validar el model ANCOVA per a PM10

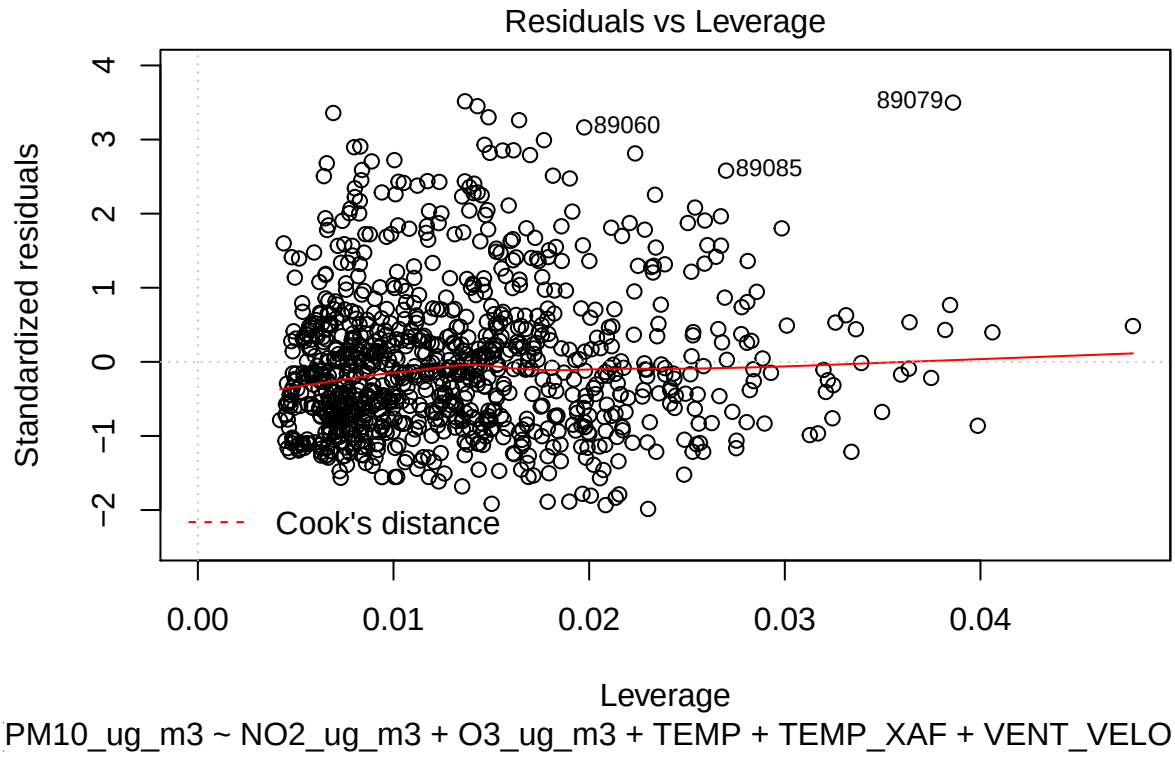
R ens permet generar diferents gràfiques del model que hem creat utilitzant la funció plot, en el nostre cas obtenim els següents gràfics.

```
plot(ancova_PM10)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF + VENT_VELO + VE


Scale−Location
89079
√|Standardized residuals|
Fitted values
PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF + VENT_VELO + VE



A la gràfica "Residuals vs Fitted" podem veure la relació dels valors residuals amb els valors fitted del model que hem creat en funció de les variables dependents.

A la gràfica "Normal Q-Q" podem veure la relació entre els valors residuals estandarditzats a l'eix Y i els quantils teòrics de les variables dependents.

A la gràfica "Scale-Location" podem veure la relació de l'arrel quadrada de l'estandarització dels valors residuals i els valors fitted del model.

Finalment, a la gràfica "Residuals vs Leverage" podem veure la relació dels valors estandarditzats residuals i l'apalancament (leverage).

Com es pot veure a la gràfica "Normal Q-Q" els residus no són del tot normals ja que aquests es desvien de la diagonal al final. Per comprovar estadísticament si els residus són normals podem utilitzar el test de Shapiro-Wilk. Aquest test comprova la hipòtesis nul · la que les dades són normals. Si rebutgem la hipòtesi nul · la (p-valor < 0.05) podem assumir que el nostre model NO és normal.

```
shapiro.test(residuals(ancova_PM10))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ancova_PM10)
## W = 0.94288, p-value < 2.2e-16
```

Com que el p-value és més petit de 0.05, s'ha de rebutgar la hipòtesis nul · la i per tant podem assumir que el nostre model no és normal.

Per comprovar estadísticament si els residus són homoscedàstics podem utilitzar el test de Breusch-Pagan. Aquest test comprova la hipòtesis nul · la que els residus són homoscedàstics. Si rebutgem la hipòtesi nul · la (p-valor < 0.05) podem assumir que els residus del model NO són homoscedàstics (són heteroscedàstics).

```
bptest(PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF + VENT_VELO + VENT_DIR_NAME, data = full_data
```

```
##
##  Breusch-Pagan test
##
## data:  PM10_ug_m3 ~ NO2_ug_m3 + O3_ug_m3 + TEMP + TEMP_XAF + VENT_VELO +     VENT_DIR_NAME
## BP = 117.58, df = 12, p-value < 2.2e-16
```

També hem obtingut un p-value inferior a 0.05 i per tant podem rebutjar la hipòsesi nul · la. Així doncs podem assumir que els residus del nostre model NO són homoscedàstics.

Un altre test per comprovar la hipòtesis de linealitat és amb el test "RESET". Aquest test comprova si X i Y es relacionen de forma lineal o, si al contrari, existeix una relació no lineal entre elles definida per potències de la variable resposta, la variable explicativa o el primer component principal de X. La hipòtesis nul · la és que es relacionen de forma lineal. Si el p-valor és molt petit (< 0.05) es rebutja la hipòtesi nul · la, el que indicarà algun tipus de relació no lineal.

```
resettest(ancova_PM10)
```

```
##
##  RESET test
##
## data:  ancova_PM10
## RESET = 7.8618, df1 = 2, df2 = 964, p-value = 0.0004104
```

Com podem veure, el p-value és més petit que 0.05 i per tant significarà que no existeix una relació lineal.

Un cop s'ha comprobat la independència, linealitat, normalitat i homoscedasticitat podem veure clarament que amb les dades que tenim no es pot aplicar un model lineal.

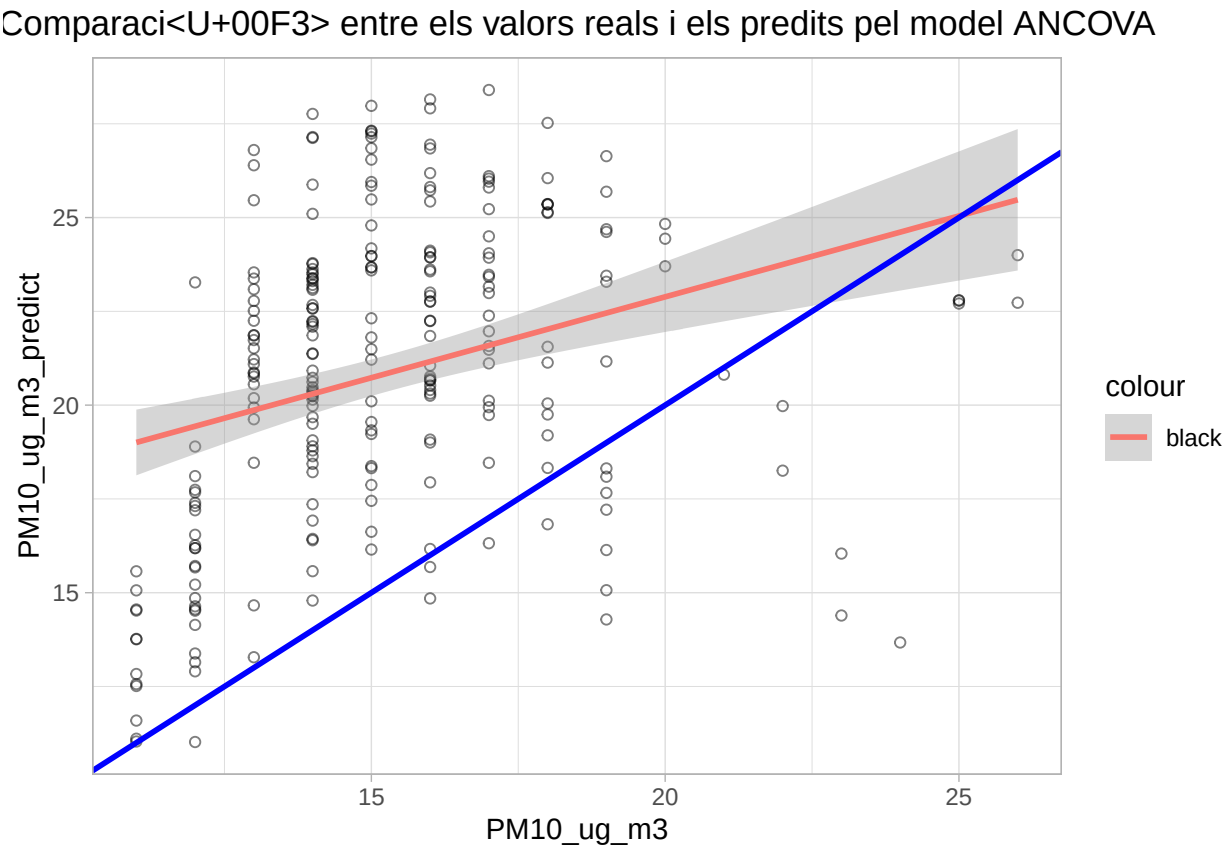
Tot i així, com que es realitza aquest model per tenir una primera aproximació, es farà el model i es consideren les variables independents.

Per fer la predicció s'utilitzarà la funció 'predict' i passant com a paràmetres el model i el dataset 'train'.

```
#Predict the data with the model
full_data_test$PM10_ug_m3_predict <- predict(ancova_PM10, full_data_test)
```

Per veure visualment com de bona és la predicció dels valors en comparació als valors reals ho podem fer amb la següent gràfica a on posem els valors reals al eix X i els valors predits al eix Y. Com més precisa sigui la predicció, els valors estaràn més pròxims a la línia blava.

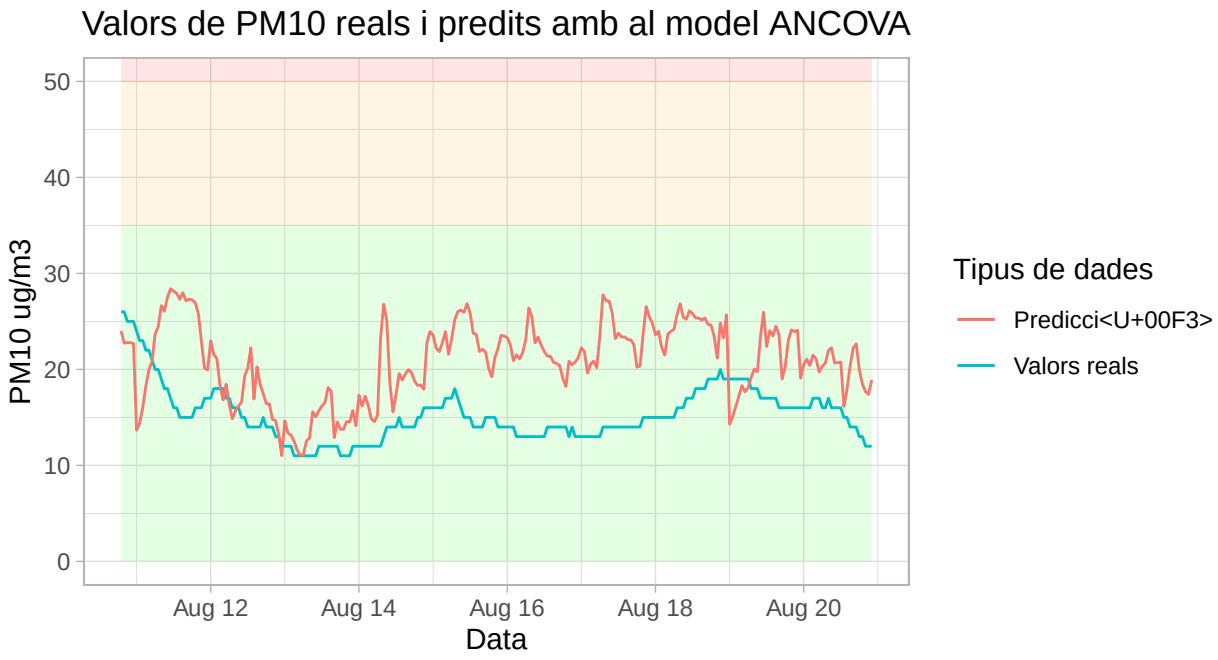
```
#Plot with the real values and the prediction
ggplot(full_data_test, aes(x = PM10_ug_m3, y = PM10_ug_m3_predict)) +
  geom_point(alpha = 0.5, shape=1) +
  stat_smooth(aes(color = "black"), method = "lm") +
  Plot_SetTheme() +
  Plot_AddTitle("Comparació entre els valors reals i els predits pel model ANCOVA") +
  Plot_SetPosTitle("center") +
  geom_abline(intercept = 0, slope = 1, size = 1, color = "blue")
```




```
#Save the plot as a PNG image
ggsave("img/plot_model_ancova_PM10_1.png", width = 14, height = 8, dpi = 150, units = "in", device='png
```

Una altre forma de veure visualment els resultats és comparant els valors en una gràfica amb dos línies de diferents colors.

```
#Plot with the real values and the prediction
ggplot() +
  annotate("rect",ymin=0, ymax=35, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_test$
  annotate("rect",ymin=35, ymax=50, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_tes
  annotate("rect",ymin=50, ymax=Inf, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_te
  geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = PM10_ug_m3, color = "Valors reals")
  geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = PM10_ug_m3_predict, color = "Predic
  Plot_SetTheme() +
  Plot_AddTitle("Valors de PM10 reals i predits amb al model ANCOVA") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ancova_PM10_2.png", width = 14, height = 8, dpi = 150, units = "in", device='png
```

6.3) Model ANCOVA per a O3

6.3.1) Entrenar el model ANCOVA per a O3

Per altra banda, amb la variable independent O3 s'han escollit les variables PM10_ug_m3, NO2_ug_m3, HUM_REL, TEMP, TEMP_XAF, VENT_VELO i VENT_DIR_NAME com a variables dependents.

Igual que en la secció anterior, per fer el model ANCOVA s'utilitzarà la funció 'lm()'.

```
ancova_O3 <- lm(O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP + TEMP_XAF + VENT_VELO + VENT_DIR_NAI
```

Un cop ja s'ha entrenat el model ja podem veure el resum amb la funció 'summary'.

```
summary(ancova_O3)
```

```
##
## Call:
## lm(formula = O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP +
##     TEMP_XAF + VENT_VELO + VENT_DIR_NAME, data = full_data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.282 -11.804  -0.498   9.557  79.933
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     97.90603   14.02163   6.982 5.39e-12 ***
## PM10_ug_m3      -0.63070    0.08447  -7.467 1.83e-13 ***
## NO2_ug_m3       -1.39840    0.05482 -25.507  < 2e-16 ***
## HUM_REL          0.01137    0.05355   0.212 0.831888
## TEMP            -2.93595    0.84207  -3.487 0.000511 ***
## TEMP_XAF         2.82522    0.55577   5.083 4.45e-07 ***
## VENT_VELO       -2.12951    0.60987  -3.492 0.000502 ***
## VENT_DIR_NAMEN  -1.89003    2.85707  -0.662 0.508433
## VENT_DIR_NAMENE -1.28217    3.03720  -0.422 0.673005
## VENT_DIR_NAMENO -0.59153    3.12023  -0.190 0.849678
## VENT_DIR_NAMEO   4.39907    3.40242   1.293 0.196346
## VENT_DIR_NAMES  10.68958    2.72396   3.924 9.32e-05 ***
## VENT_DIR_NAMESE 10.84889    2.75590   3.937 8.86e-05 ***
## VENT_DIR_NAMESO  6.00506    3.05969   1.963 0.049976 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.39 on 965 degrees of freedom
## Multiple R-squared:  0.5628, Adjusted R-squared:  0.5569
## F-statistic: 95.55 on 13 and 965 DF,  p-value: < 2.2e-16
```

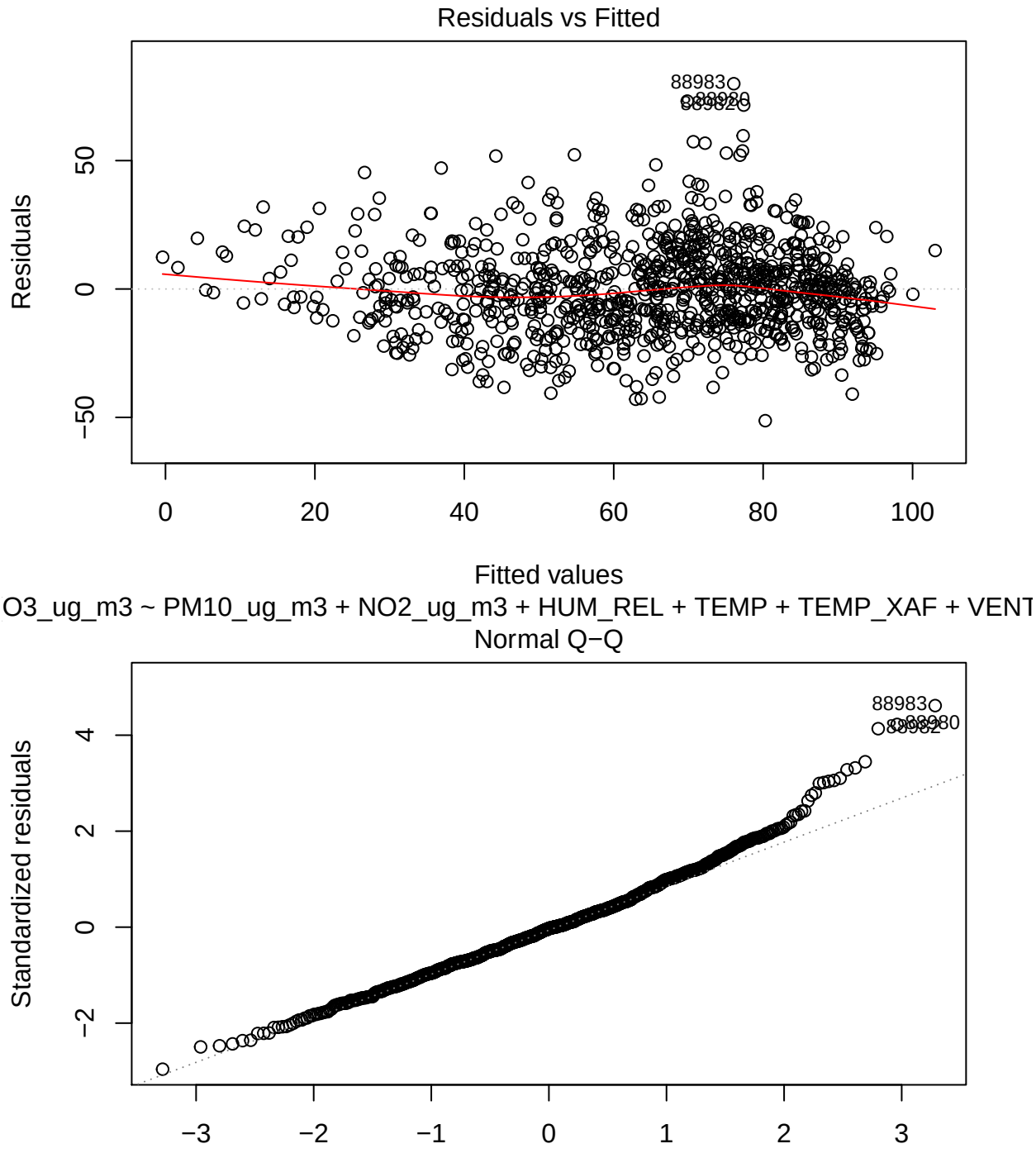
En general, podem veure que totes les variables contínues a excepció de la HUM_REL són molt importants. La variable categòrica 'VENT_DIR_NAME' les categoríes més importants són 'S' i 'SE', ve a ser el vent del sud i sud-est.

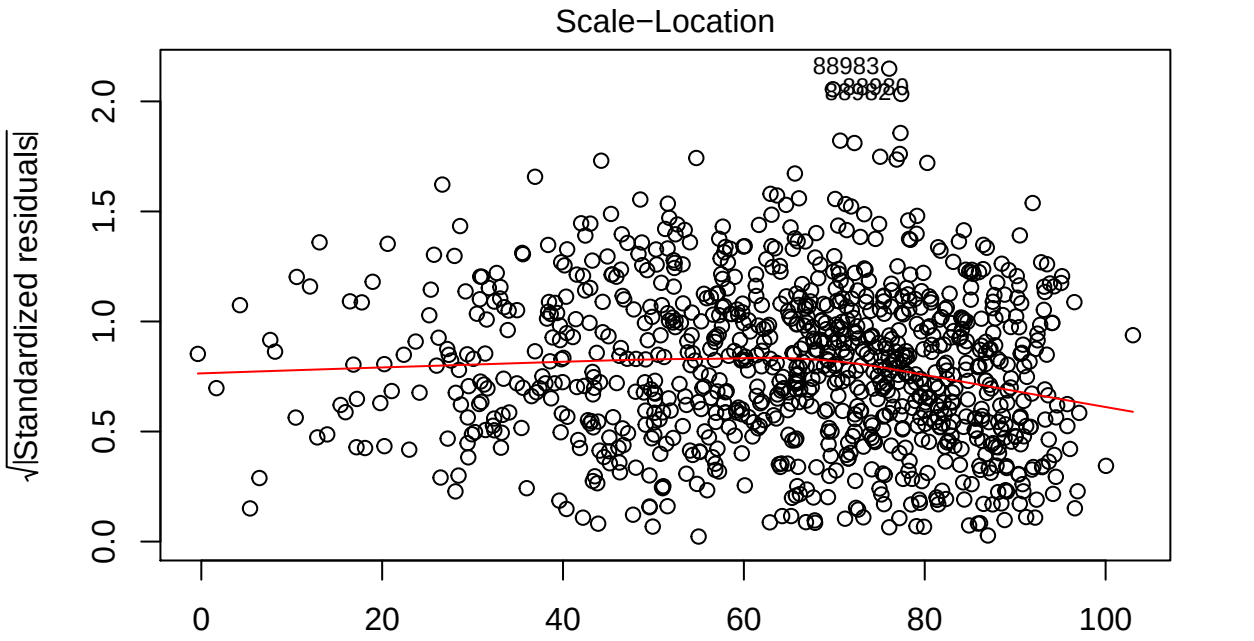
Respecta als valors residuals, podem veure que la mediana també és molt pròxima a 0 amb valor de -0.857.

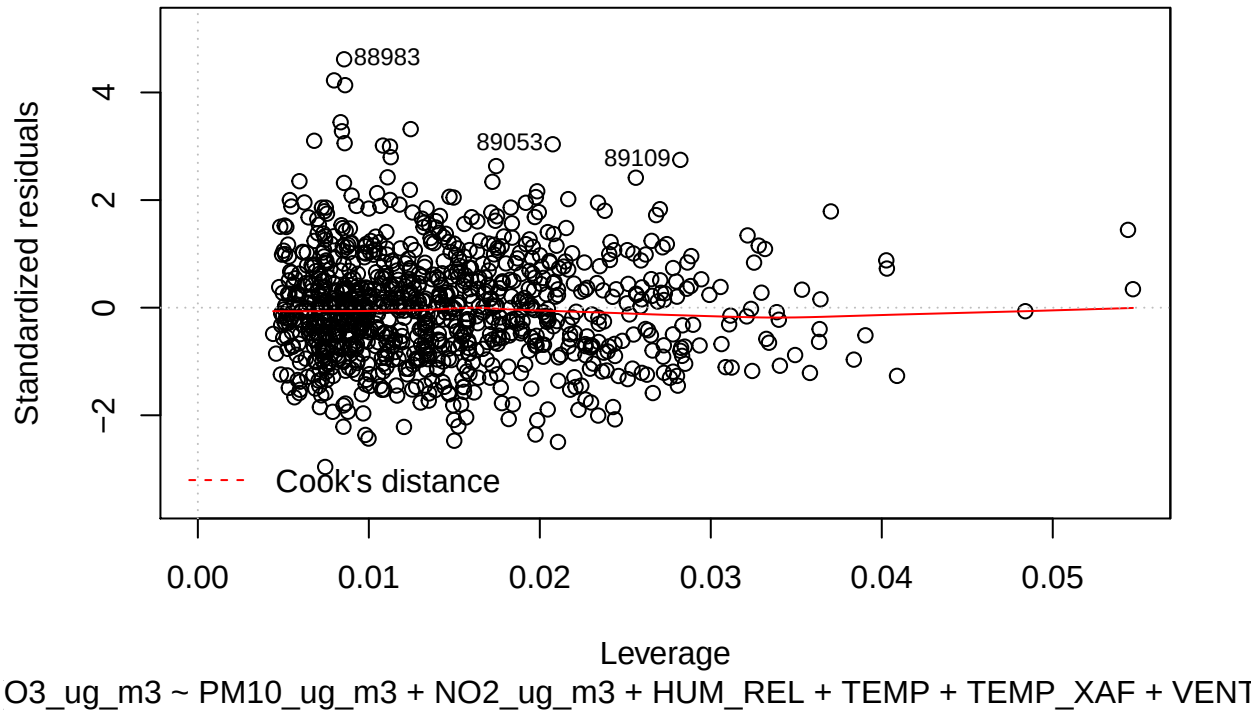
6.3.2) Provar i validar el model ANCOVA for O3

De la mateixa forma que hem fet amb PM10, també mostrarem les diferents gràfiques del model amb la funció 'plot'.

```
plot(ancova_O3)
```


Residuals vs Fitted
88983
88980
88982
Residuals
50
0
−50
0
20
40
60
80
100
Fitted values
O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP + TEMP_XAF + VENT
Normal Q−Q
88983
88980
88982
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP + TEMP_XAF + VENT





A la gràfica "Residuals vs Fitted" podem veure la relació dels valors residuals amb els valors fitted del model que hem creat en funció de les variables dependents.

A la gràfica "Normal Q-Q" podem veure la relació entre els valors residuals estandarditzats a l'eix Y i els quantils teòrics de les variables dependents.

A la gràfica "Scale-Location" podem veure la relació de l'arrel quadrada de l'estandardització dels valors

residuals i els valors fitted del model.

Finalment, a la gràfica "Residuals vs Leverage" podem veure la relació dels valors estandarditzats residuals i l'apalancament (leverage).

Com es pot veure a la gràfica "Normal Q-Q" els residus no són del tot normals ja que aquests es desvien de la diagonal al final. Per comprovar estadísticament si els residus són normals utilitzarem el test de Shapiro-Wilk explicat anteriroment.

```
shapiro.test(residuals(ancova_O3))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ancova_O3)
## W = 0.98423, p-value = 8.676e-09
```

Com que el p-value és més petit de 0.05, s'ha de rebutgar la hipòtesis nul · la i per tant podem assumir que el nostre model no és normal.

Per comprovar estadísticament si els residus són homoscedàstics ho farem amb el test de Breusch-Pagan explicat anteriroment.

```
bptest(O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP + TEMP_XAF + VENT_VELO + VENT_DIR_NAME, data =
```

```
##
##  Breusch-Pagan test
##
## data:  O3_ug_m3 ~ PM10_ug_m3 + NO2_ug_m3 + HUM_REL + TEMP + TEMP_XAF +     VENT_VELO + VENT_DIR_NAME
## BP = 80.382, df = 13, p-value = 9.349e-12
```

També hem obtingut un p-value inferior a 0.05 i per tant podem rebutjar la hipòsesi nul · la. Així doncs podem assumir que els residus del nostre model NO són homoscedàstics.

Per comprovar la hipòtesis de linealitat ho farem amb el test "RESET" explicat anteriroment.

```
resettest(ancova_O3)
```

```
##
##  RESET test
##
## data:  ancova_O3
## RESET = 16.043, df1 = 2, df2 = 963, p-value = 1.4e-07
```

Com podem veure, el p-value és més petit que 0.05 i per tant significarà que no existeix una relació lineal.

Un cop s'ha comprobat la independència, linealitat, normalitat i homoscedasticitat del model ja podem utilitzar el model amb el dataset 'test' per comprobar la predicció que fa el model.

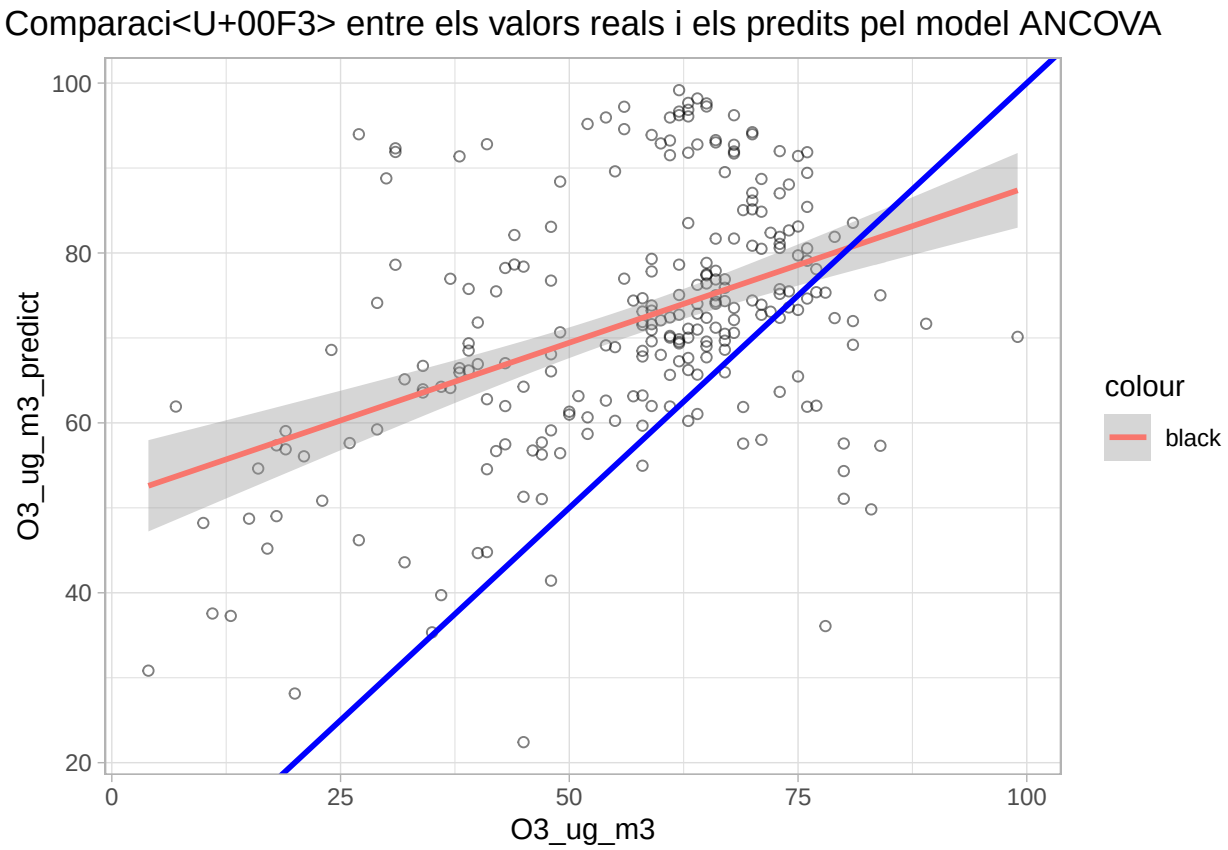
Per fer la predicció s'utilitzarà la funció 'predict' i passant com a paràmetres el model i el dataset 'train'.

```
#Predict the data with the model
full_data_test$O3_ug_m3_predict <- predict(ancova_O3, full_data_test)
```

Per veure visualment com de bona és la predicció dels valors en comparació als valors reals ho podem fer amb la següent gràfica a on posem els valors reals al eix X i els valors predits al eix Y. Com més precisa sigui la predicció, els valors estaràn més pròxims a la línia blava.

```
#Plot with the real values and the prediction
ggplot(full_data_test, aes(x = O3_ug_m3, y = O3_ug_m3_predict)) +
  geom_point(alpha = 0.5, shape=1) +
  stat_smooth(aes(color = "black"), method = "lm") +
```

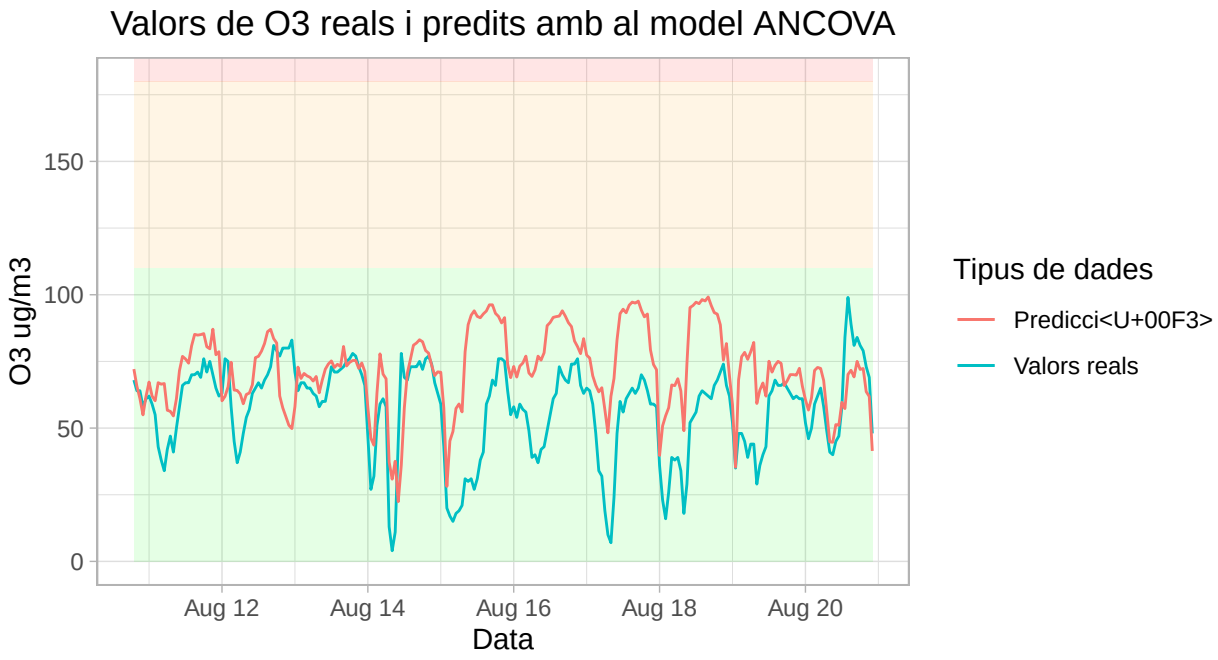
```
  Plot_SetTheme() +
  Plot_AddTitle("Comparació entre els valors reals i els predits pel model ANCOVA") +
  Plot_SetPosTitle("center") +
  geom_abline(intercept = 0, slope = 1, size = 1, color = "blue")
```




```
#Save the plot as a PNG image
ggsave("img/plot_model_ANCOVA_O3_1.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

Una altre forma de veure visualment els resultats és comparant els valors en una gràfica amb dos línies de diferents colors.

```
#Plot with the real values and the prediction
ggplot() +
  annotate("rect",ymin=0, ymax=110, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_tes
  annotate("rect",ymin=110, ymax=180, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_t
  annotate("rect",ymin=180, ymax=Inf, xmin=min(full_data_test$airrmeasur_datetime),xmax=max(full_data_t
  geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = O3_ug_m3, color = "Valors reals"),na
  geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = O3_ug_m3_predict, color = "Predicci
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 reals i predits amb al model ANCOVA") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ANCOVA_O3_2.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

7) Sèries temporals

7.1) Model ARIMA per a PM10

El model ARIMA és un model paramètric que permet modelar sèries temporals estacionàries i no estacionàries.

Els passos que s'han de seguir per a ajustar un model ARIMA es coneix com a metodologia Box-Jenkins, aquests són:

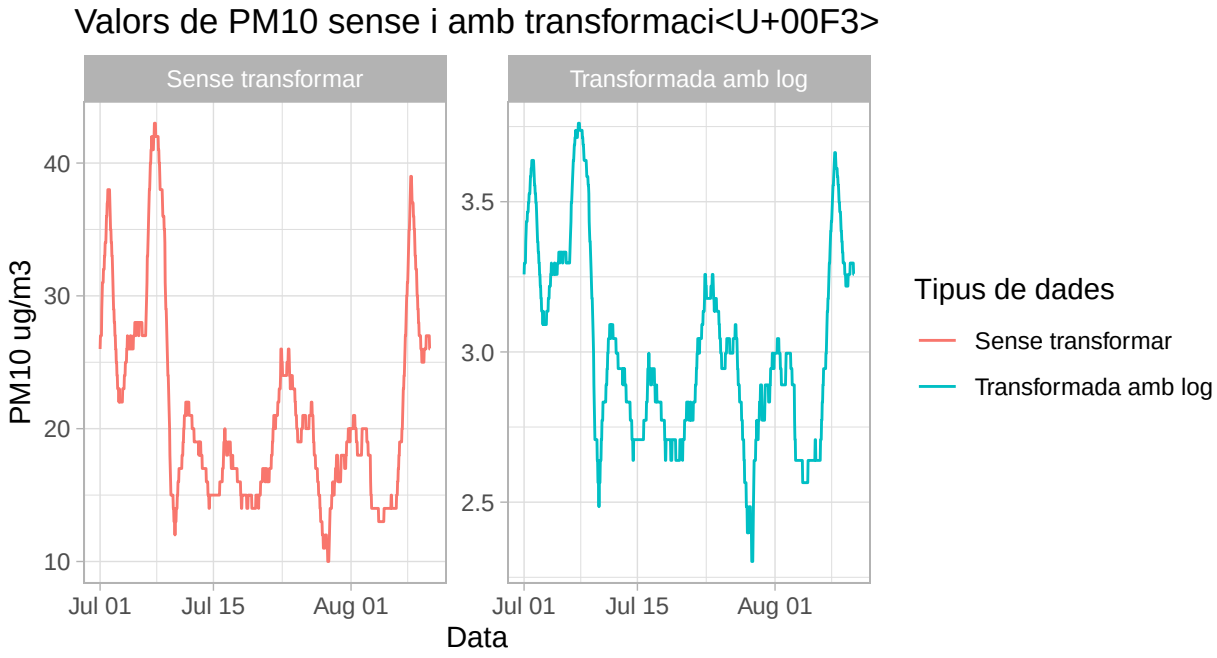
- 1) Assegurar/convertir la sèrie amb variància constant
- 2) Assegurar/convertir la sèrie estacionària
- 3) Identificació del model ARIMA.
- 4) Estimació dels paràmetres.
- 5) Validació del model. Si el model no és vàlid cal tornar al punt 1. Si el model és vàlid, es passa al següent punt.
- 6) Predicció de nous valors.

7.1.1) Transformació de PM10 per homogeneïtzar la variància

La condició que hi hagi homoscedasticitat és un requeriment per a les sèries estacionàries. Si la sèrie augmenta de variabilitat, cal transformar-la per a homogeneïtzar la variància. Una de les transformacions habituals és el log. Si tenim una sèrie que la variància augmenta a mesura que transcorre el temps, la transformació amb el logaritme pot ajudar a fer-la homoscedàstica.

Per veure la diferència ho podem veure visualment amb la següent gràfica.

```
#Plot with the real values and the prediction
ggplot(data = data.frame(rbind(cbind.data.frame(PM10_ug_m3 = full_data_train$PM10_ug_m3,airrmeasur_date
            aes(x = airrmeasur_datetime, y = PM10_ug_m3, color = type),na.rm=TRUE) +
  geom_line() +
  facet_wrap(~ type, scales = "free_y") +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de PM10 sense i amb transformació") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_PM10_arima_amb_i_sense_transforamcio.png", width = 14, height = 8, dpi = 150, units =
```

Sembla que les dades logtransformades tenen un perfil més homoscedàstic.

7.1.2) Convertir la sèrie de PM10 en estacionària

Si una sèrie no és estacionària, es pot transformar prenent diferències fins que la sèrie diferenciada sigui estacionària. Amb aquesta operació es diu que la sèrie original és un procés integrat i parlem de models ARIMA.

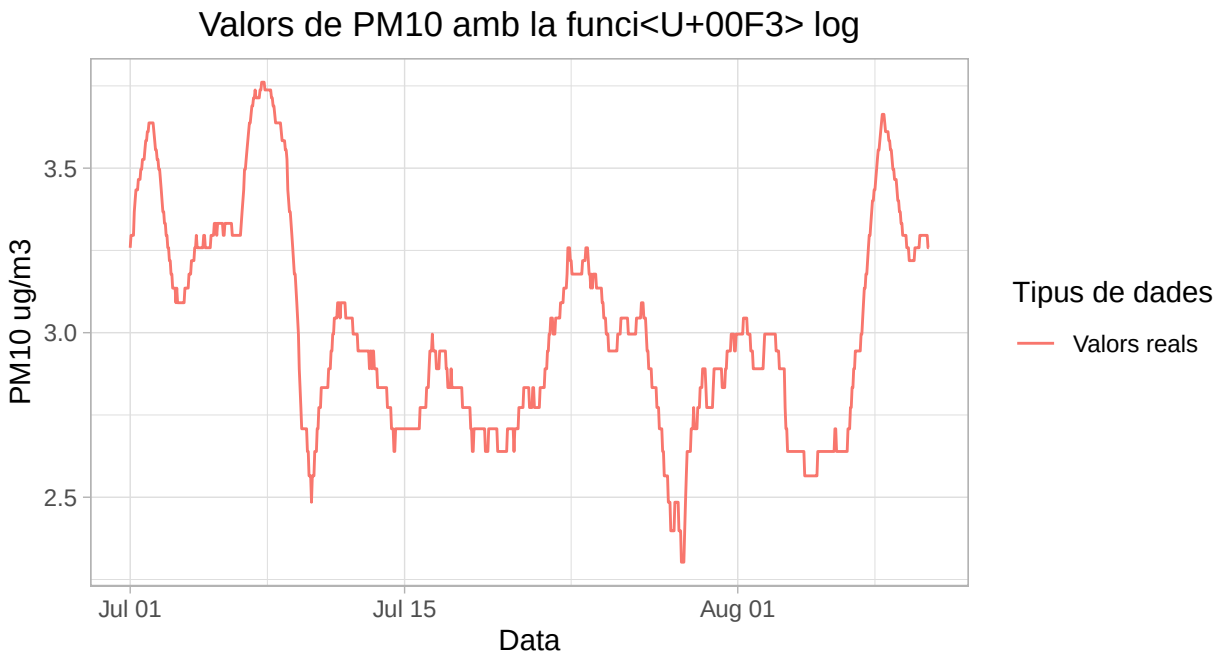
Les diferències regulars i estacionals permeten convertir una sèrie en estacionària. La decisió sobre quina transformació fer es basa fonamentalment en la combinació de quatre criteris:

- 1) Gràfics de la sèrie: Sense tendències ni estacionalitat
- 2) Desviacions típiques: Les transformacions han de proporcionar les sèries amb variàncies més petites. Si es sobrediferencia una sèrie sol augmentar la variància.
- 3) Contrast d'arrels unitàries: Contrastos per a veure si una sèrie és estacionària o si cal diferenciar-la. Un contrast és el de Dikey-Fuller augmentat (ADF). La hipòtesi nul · la és que cal diferenciar-la.
- 4) Diagrama d'autocorrelació (correlogram)

Si mostrem gràficament les dades de la sèrie PM10 logtrasformada podrem veure la següent gràfica.

```
#Plot with the real values and the prediction
ggplot() +
  geom_line(data = full_data_train, aes(x = airrmeasur_datetime, y = log(PM10_ug_m3), color = "Valors r
  Plot_SetTheme() +
```

```
  Plot_AddTitle("Valors de PM10 amb la funció log") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ancova_PM10_train_1.png", width = 14, height = 8, dpi = 150, units = "in", devic
```

Com es pot veure, no hi ha una clara tendència ni tampoc un component estacional. Per tant, podríem probar d'aplicar una diferència d'ordre 1 o una diferència d'ordre 24, ja que la sèrie té una freqüència horaria.

La variació de la variable sense aplicar la funció de log és:

```
var(full_data_train$PM10_ug_m3)
```

```
## [1] 52.61284
```

Amb la variable logtrasformada tenim una variància de:

```
var(log(full_data_train$PM10_ug_m3))
```

```
## [1] 0.09903976
```

La variància de la diferència regular és:

```
var(diff(log(full_data_train$PM10_ug_m3),1))
```

```
## [1] 0.0007061853
```

La variància de la diferència estacional és:

```
var(diff(log(full_data_train$PM10_ug_m3),24))
```

```
## [1] 0.06146018
```

La variància de la diferència regular i estacional és:

```
var(diff(diff(log(full_data_train$PM10_ug_m3),1),24))
```

```
## [1] 0.001462898
```

Podem veure que amb la diferenciació regular d'ordre 1 obtenim la menor variància.

El test d'arrels unitàries per a comprovar l'estacionarietat d'una sèrie temporal desestacionalizadas de Dikey-Fuller augmentat (ADF), ens pot ajudar en la nostra decisió de diferenciar o no la sèrie. Si el p-valor és molt petit (< 0.05) es rebutja la hipòtesi nul · la, per tant ens indicarà que no cal diferenciar-la.

```
adf.test(diff(log(full_data_train$PM10_ug_m3),1))
```

```
## Warning in adf.test(diff(log(full_data_train$PM10_ug_m3), 1)): p-value
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(full_data_train$PM10_ug_m3), 1)
## Dickey-Fuller = -7.044, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

Veiem que hem de rebutjar la H0 (cal diferenciar la sèrie) ja que tot i que ens dóna un p-valor de 0.01, el missatge ens avisa que el p-valor és inferior a aquest valor imprès. Per tant, sembla que caldrà només fer una diferència regular tal i com s'ha fet.

Anomenem a la nostre sèrie de treball, sèrie logtransformada i diferenciada regularment i estacional 'serie_arima_PM10'.

```
serie_arima_PM10 <- diff(log(full_data_train$PM10_ug_m3),1)
```

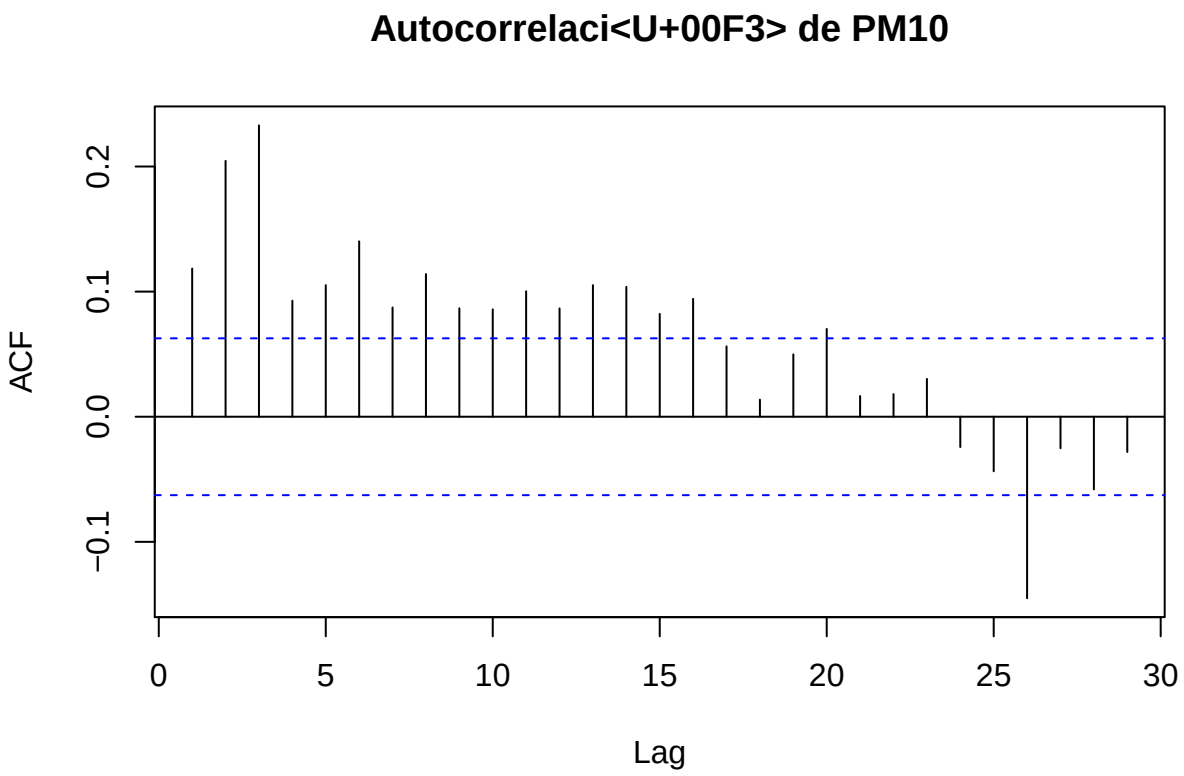
7.1.3) Identificació del tipus de model ARIMA per a PM10

Si una sèrie està ben identificada, quan s'ajusta el model dels residus no han de tenir estructura, és a dir, han de semblar-se a un soroll blanc. Un soroll blanc és una sèrie estacionària en la qual cap observació depèn de les altres i, per tant, tots els valors de l'ACF i la PACF són nuls. El correlograma i el correlograma parcial han de ser molt similars i els valors no són significativament diferents de zero.

Per tal de modelar la sèrie, anem a representar els diagrames de les funcions d'autocorrelació (ACF) i d'autocorrelació parcial (PACF).

El diagrama de funcions d'autocorrelació (ACF) el podem representar amb la següent funció.

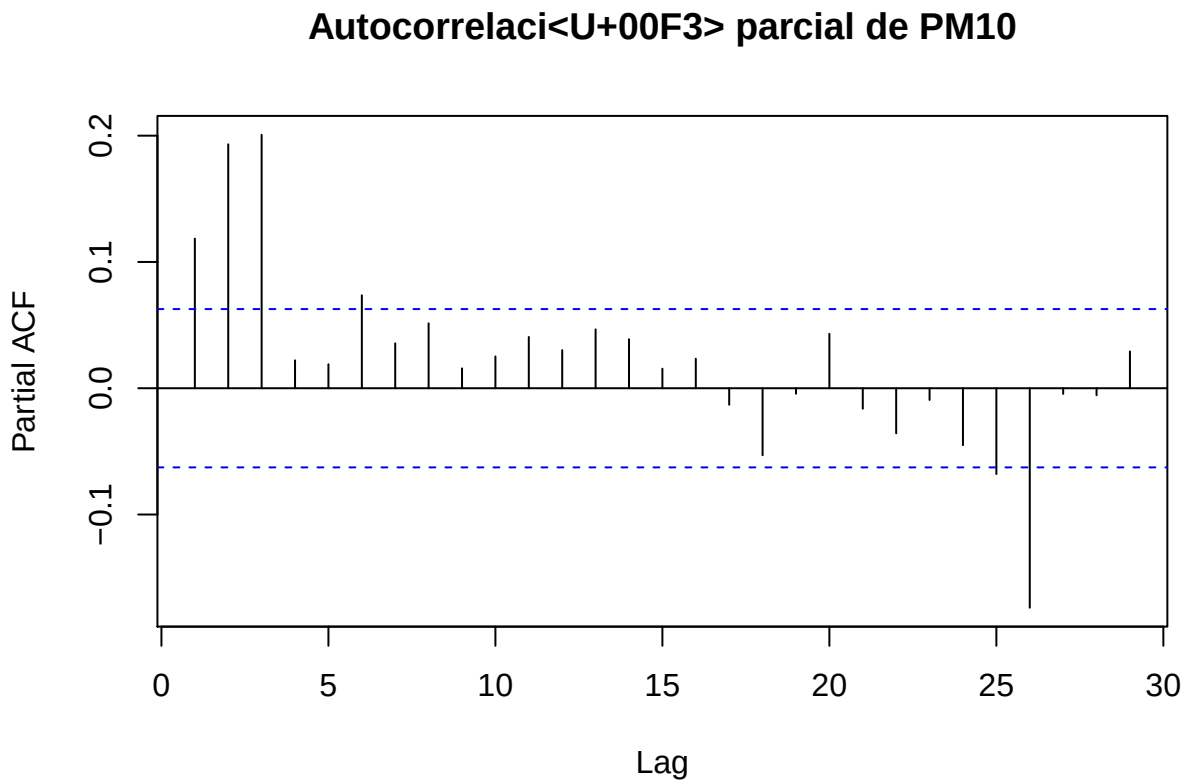
```
acf(serie_arima_PM10, main="Autocorrelació de PM10")
```



Cada barra representa un retard. Veiem que hi ha correlacions negatives significatives al final de la gràfica.

El diagrama d'autocorrelació parcial (PACF) el podem representar amb la següent funció.

```
pacf(serie_arima_PM10, main="Autocorrelació parcial de PM10")
```



Per identificar el tipus de ARIMA s'utilitzarà la funció 'auto.arima' que ens detectarà quin és el millor model a aplicar amb les dades de la nostra serie. Aquesta funció compara amb criteris d'informació (per defecte el Criteri d'Informació d'Akaike corregit, AICc) tots els possibles models amb un màxim d'ordre p i q per l'estructura regular i P i Q per a l'estructura estacionària definits per l'usuari.

```
arima_PM10 <-auto.arima(serie_arima_PM10, d=0, D=0, max.p=5, max.q=5, max.P=2, max.Q=2)
arima_PM10
```

```
## Series: serie_arima_PM10
## ARIMA(3,0,3) with zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2      ma3
##       0.4350  -0.1857  0.5561  -0.4055  0.3269  -0.4498
## s.e.  0.2501   0.2236  0.1401   0.2562  0.2233   0.1394
##
## sigma^2 estimated as 0.0006401:  log likelihood=2211.18
## AIC=-4408.37   AICc=-4408.25   BIC=-4374.17
```

Podem veure que el model que ens ajusta per a la nostra sèrie temporal 'serie_arima_PM10' és un ARMA(3,3) i amb un AIC de -4408.37.

7.1.4) Validació del model ARIMA per a PM10

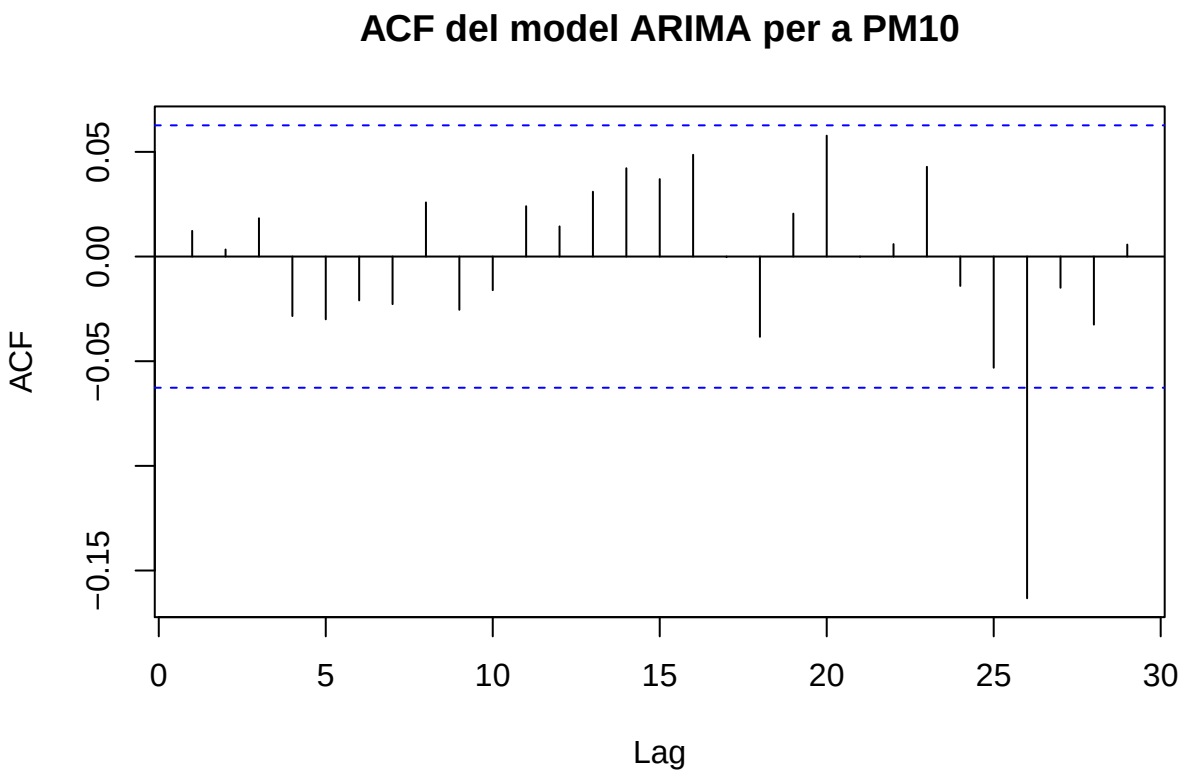
Si una sèrie està ben identificada, quan s'ajusta el model dels residus no han de tenir estructura, és a dir, han de semblar-se a un soroll blanc. Un soroll blanc és una sèrie estacionària en la qual cap observació depèn de les altres i, per tant, tots els valors de l'ACF i la PACF són nuls. El correlograma i el correlograma parcial han de ser molt similars i els valors no són significativament diferents de zero.

Primer de tot es realitzarà la validació del model. La ACF i la PACF dels residus han de ser molt semblants, no mostrar estructura i tenir gairebé tots els valors dins de les bandes de confiança.

Per calcular els residus estandarditzats ho farem amb la funció 'rstandard'.

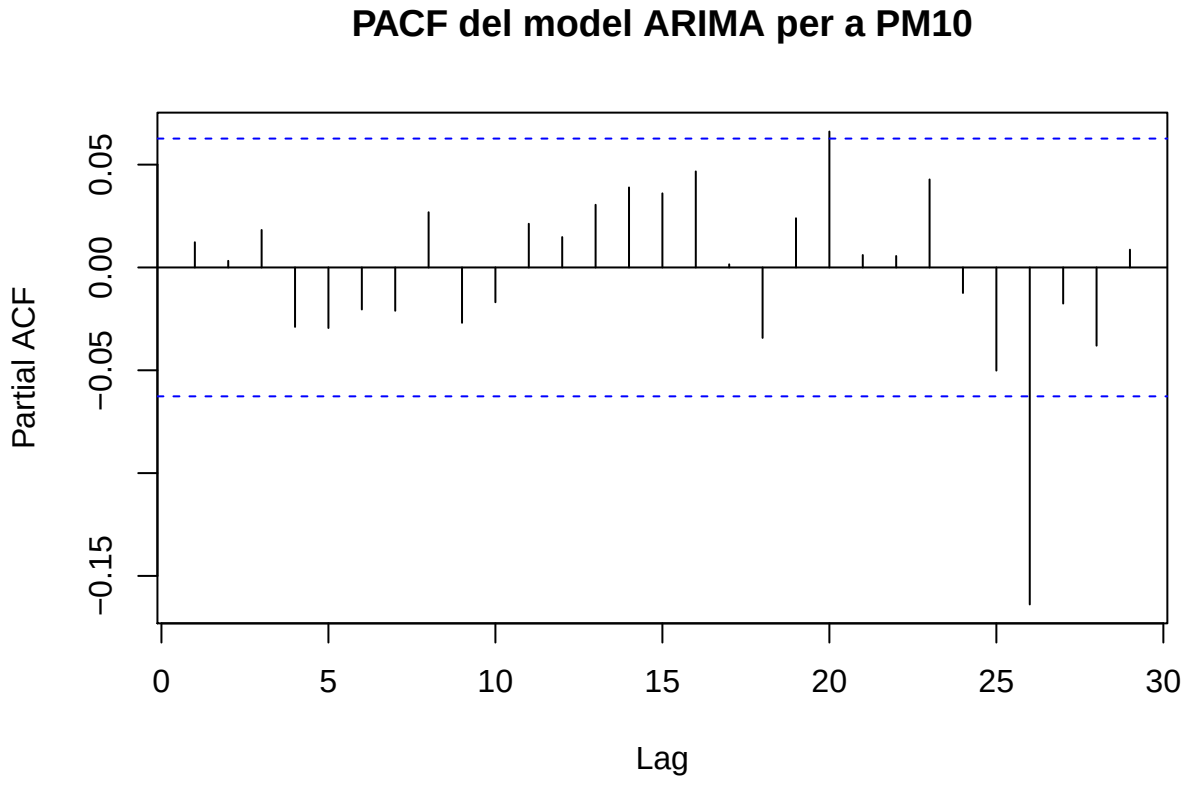
Si mostrem el ACF dels residus estandarditzats obtindrem la següent gràfica.

```
acf(rstandard(arima_PM10), main = "ACF del model ARIMA per a PM10")
```



Com podem veure la majoria dels valors estan a dins les bandes de confiança a excepció de una única banda que surt molt.

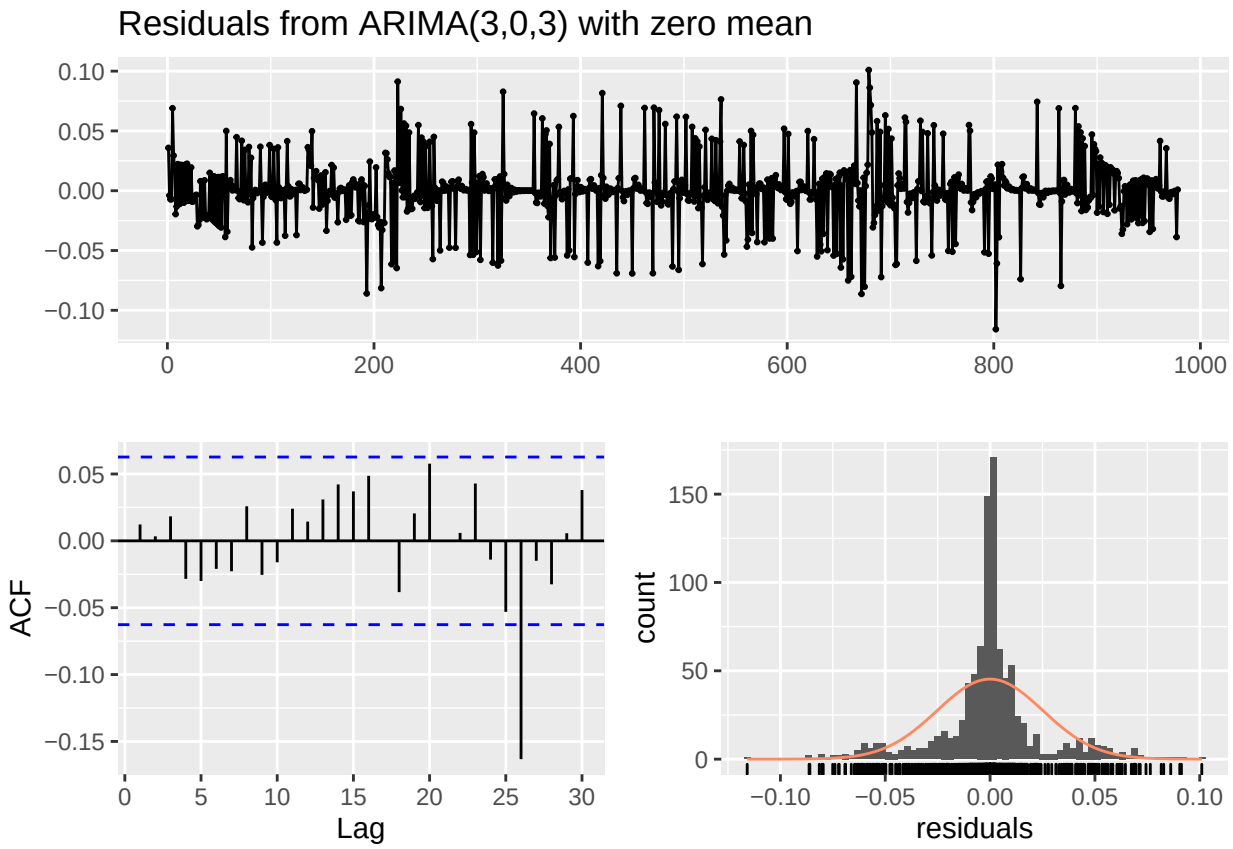
```
pacf(rstandard(arima_PM10), main = "PACF del model ARIMA per a PM10")
```



Amb la PACF també podem veure que la majoria dels valors estan a dins les bandes de confiança a excepció de una única banda que surt molt.

Una altre forma per veure els residus és utlitzant la funció 'checkresiduals'.

```
checkresiduals(arima_PM10)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,3) with zero mean
## Q* = 4.6674, df = 4, p-value = 0.3232
##
## Model df: 6.   Total lags used: 10
```

El contrast de Ljung-Box-Pierce, també conegut com a contrast de portmanteau. La hipòtesi nul · la és que les primeres autocorrelacions són nul · les. La hipòtesi alternativa d'aquest contrast implica que alguna de les correlacions és diferent de zero i, per tant, no es pot assumir que els residus siguin soroll blanc.

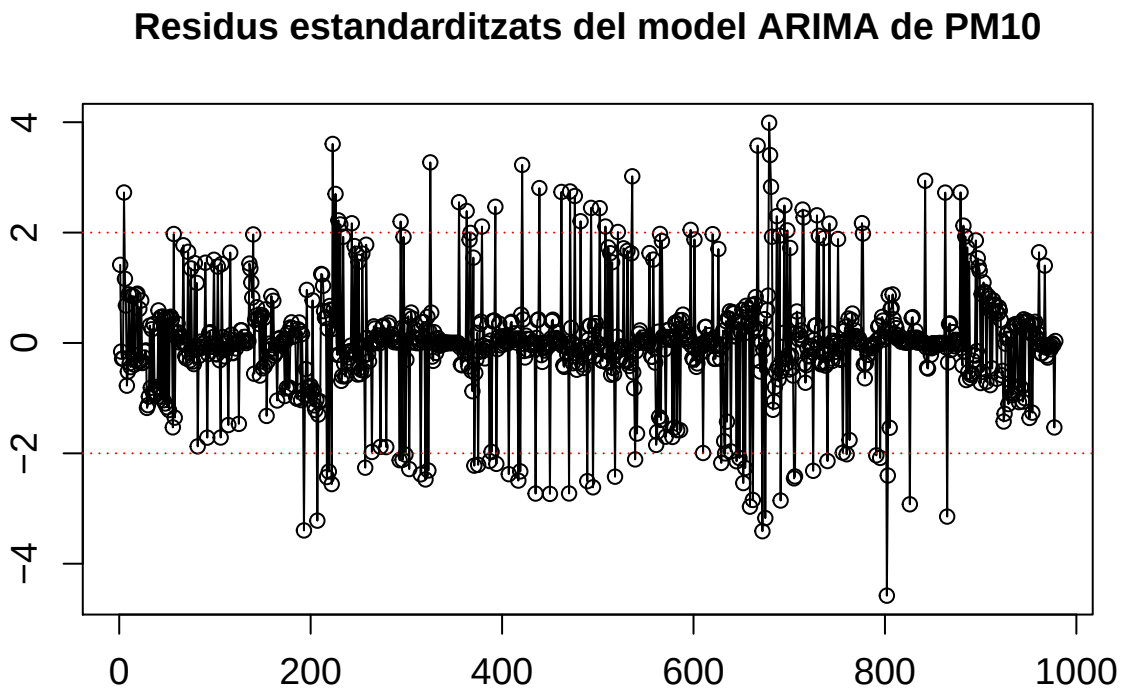
A la funcó 'checkresidus' també s'inclou el contrast de Ljung-Box-Pierce.

Com podem veure, hem obtingut un p-value superior a 0.05 i per tant no es pot rebutjar la hipòtesi nul · la, per la qual cosa assumim que les primeres autocorrelacions són nul · les.

El gràfic dels residus ha de mostrar que els residus varien al voltant del zero, sense tendències, la variància és constant i no hi ha valors atípics. Aproximadament el 95% dels residus estandarditzats han d'estar entre -2 i 2 desviacions típiques.

```
par(mfcol=c(1,1), cex.axis=1.2, cex.main=1.2, cex.lab=1.2)
plot(rstandard(arima_PM10), xlab="", ylab="", main="", type="o")
title("Residus estandarditzats del model ARIMA de PM10")
```

```
abline(h=2, lty=3, col="red")
abline(h=-2, lty=3, col="red")
```



Com podem veure hi ha una gran part dels residus que esta fora els marges de 2 i -2. Per calcular-los ho farem amb la següent comanda:

```
sum(abs(rstandard(arima_PM10))<=2)
```

```
## [1] 889
```

Tenim 889 dels 978 valors que estan a dins de les 2 desviacions típiques. Aixo representa un 90.8997955% del total de residus que està a dins de (-2,2).

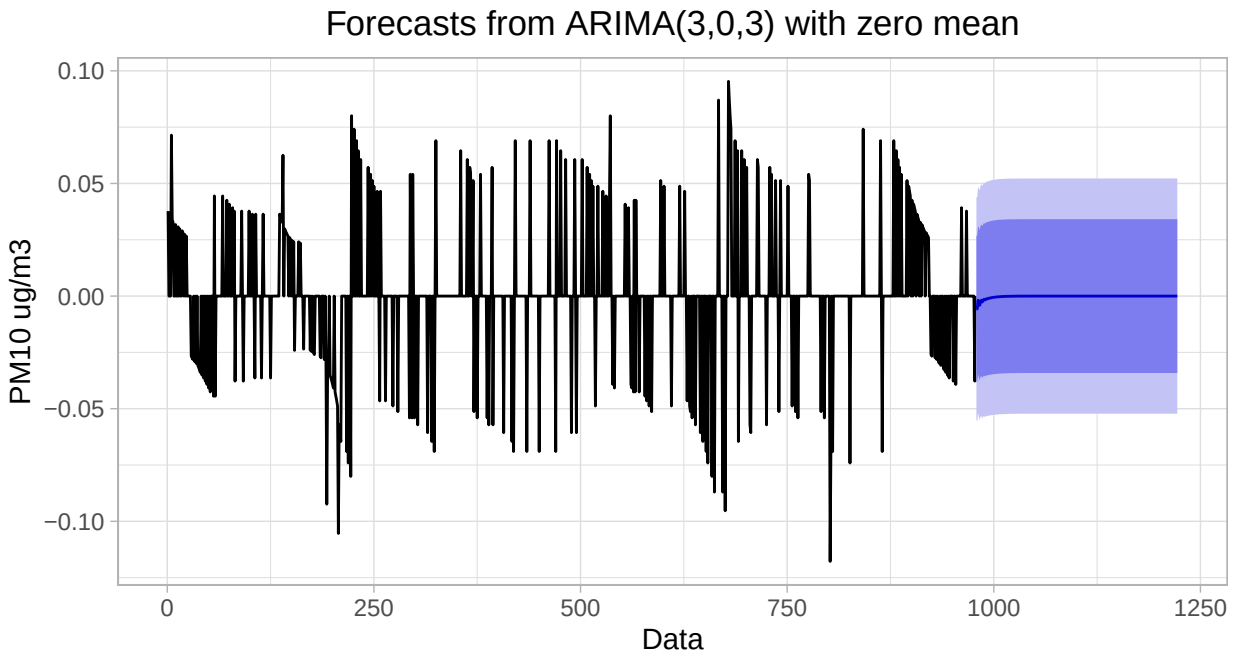
7.1.5) Predicció de nous valors amb el model ARIMA per a PM10

Per realitzar les prediccions s'utilitzarà la funció 'forecast' i es passarà com a paràmetre el model ARIMA i també el nombre de dades que es vol predir, en aquest cas serà el nombre files del dataset test.

```
#Create the forecast
result_arima_PM10 <- forecast(arima_PM10, nrow(full_data_test))
```

Un cop ja hem creat el forecast mostrarem les dades que s'han predit amb la funció 'autoplot'.

```
#Plot the forecast of the data
autoplot(result_arima_PM10) +
  Plot_SetTheme() +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("PM10 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

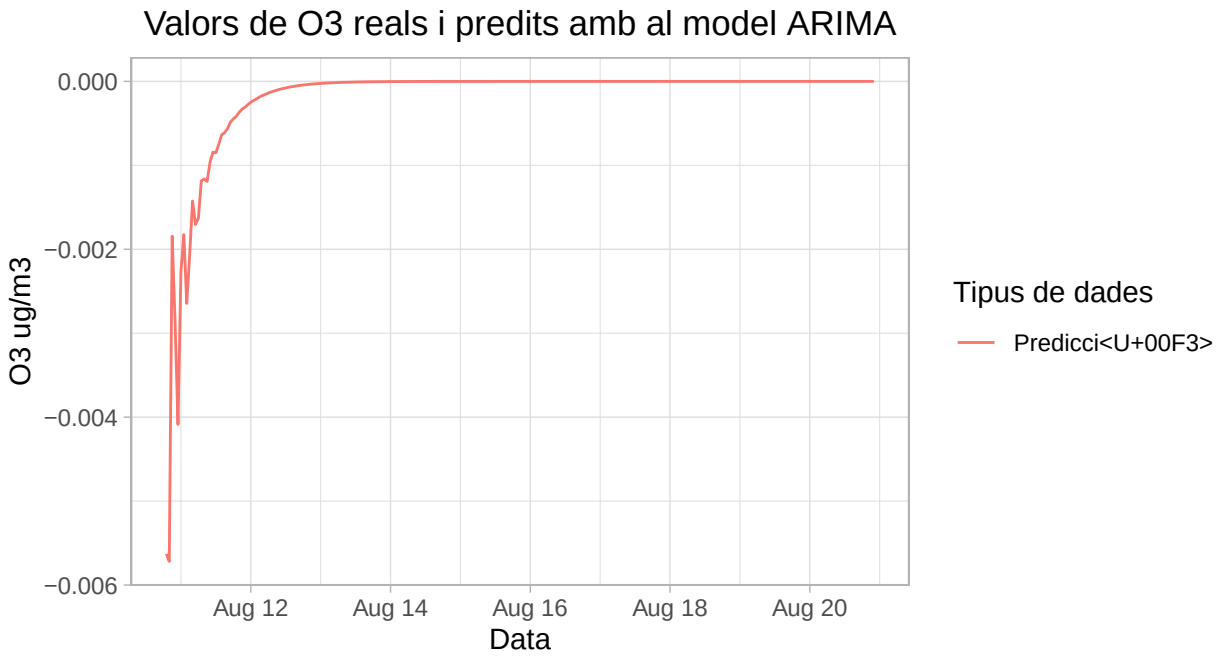
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ARIMA_PM10_1.png", width = 14, height = 8, dpi = 150, units = "in", device='png'
```

També podem mostrar únicament els valors predits amb la llibreria 'ggplot'.

```
#Plot with the real values and the prediction
ggplot() +
  #geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = log(PM10_ug_m3), color = "Valors r
  geom_line(data = as.data.frame(result_arima_PM10), aes(x = full_data_test$airrmeasur_datetime, y = res
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 reals i predits amb al model ARIMA") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ARIMA_PM10_2.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

7.2) Model ARIMA per a O3

Igual que s'ha fet a l'apartat anterior amb la variable PM10, en aquest apartat es seguirant els mateixos passos però utilitzant la variable O3.

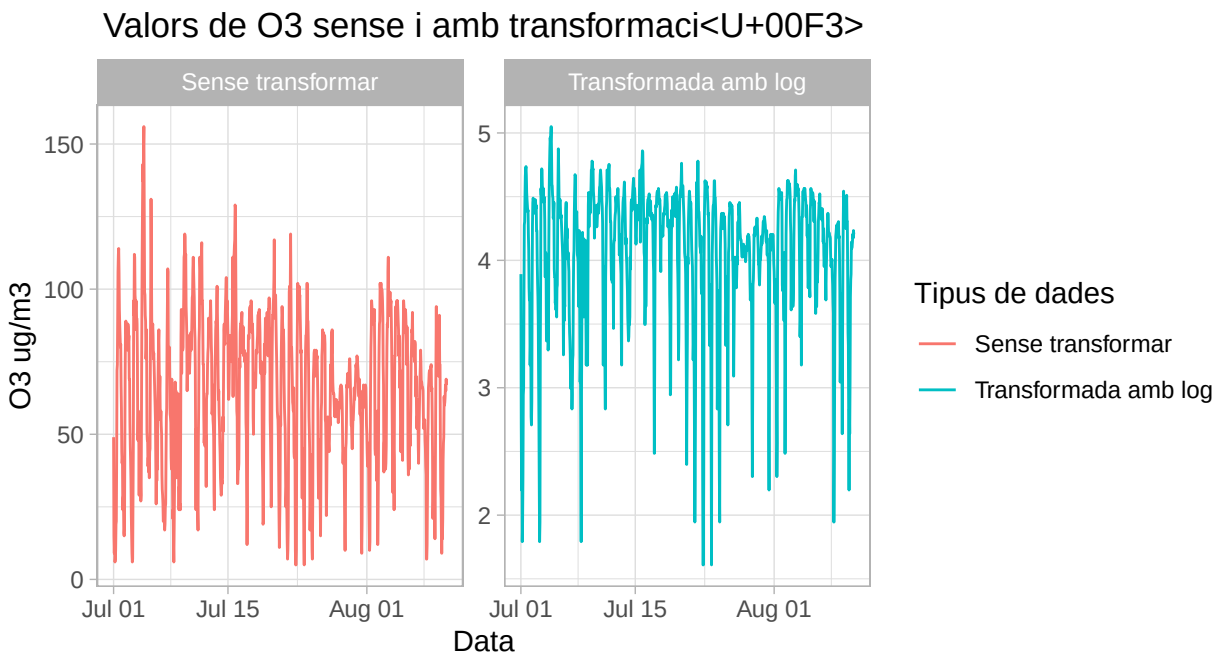
7.2.1) Transformació de O3 per homogeneïtzar la variància

Amb O3 també aplicarem una transformació utilitzant el log per així homogeneïtzar la variància de la variable.

Per veure la diferència ho podem veure visualment amb la següent gràfica.

```
#Plot with the real values and the prediction
ggplot(data = data.frame(rbind(cbind.data.frame(O3_ug_m3 = full_data_train$O3_ug_m3,airrmeasur_datetime
            aes(x = airrmeasur_datetime, y = O3_ug_m3, color = type),na.rm=TRUE) +
  geom_line() +
  facet_wrap(~ type, scales = "free_y") +
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 sense i amb transformació") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
```

```
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

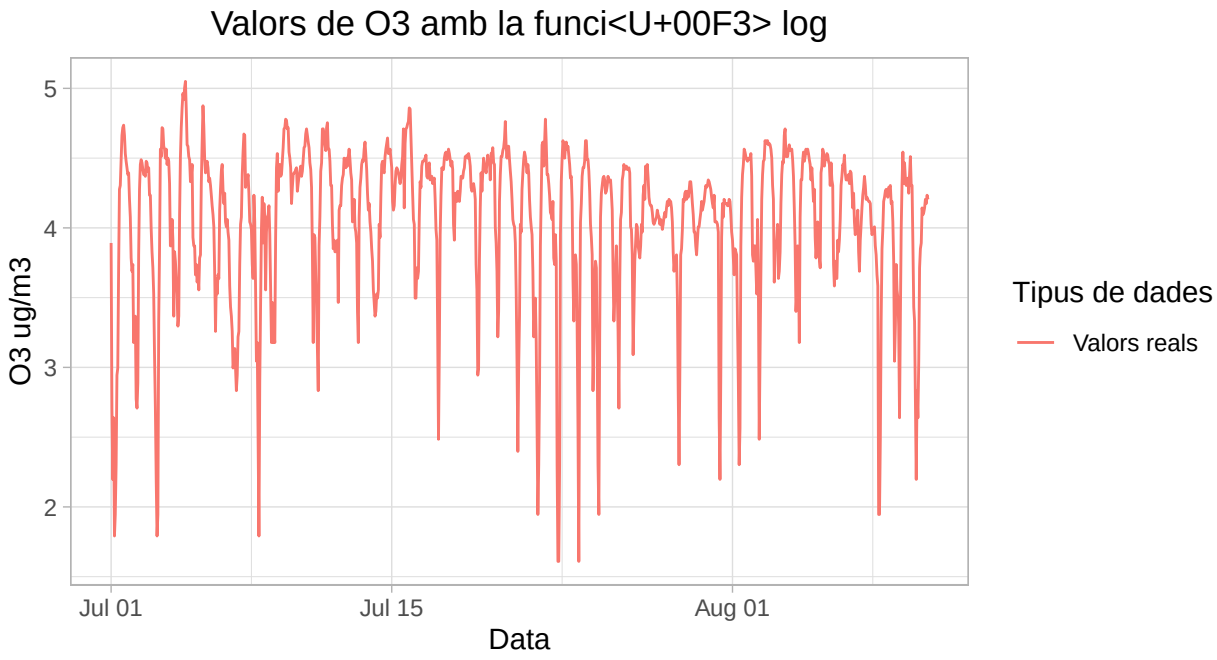
```
#Save the plot as a PNG image
ggsave("img/plot_O3_arima_amb_i_sense_transforamcio.png", width = 14, height = 8, dpi = 150, units = "in
```

7.2.2) Convertir la sèrie de O3 en estacionària

En aquest apartat es seguirant els mateixos passos per convertir la sèrie en estacionària.

Primer de tot, mostrem gràficament les dades de la sèrie O3 logtrasformada a la següent gràfica.

```
#Plot with the real values and the prediction
ggplot() +
  geom_line(data = full_data_train, aes(x = airrmeasur_datetime, y = log(O3_ug_m3), color = "Valors real
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 amb la funció log") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ANCOVA_O3_train_1.png", width = 14, height = 8, dpi = 150, units = "in", device=
```

Com es pot veure, no hi ha una clara tendència però sí un component estacional. Per tant, podríem probar d'aplicar una diferència d'ordre 1 o una diferència d'ordre 24, ja que la sèrie té una freqüència horaria.

La variació de la variable sense aplicar la funció de log és:

```
var(full_data_train$O3_ug_m3)
```

```
## [1] 682.414
```

Amb la variable logtrasformada tenim una variància de:

```
var(log(full_data_train$O3_ug_m3))
```

```
## [1] 0.3109559
```

La variància de la diferència regular és:

```
var(diff(log(full_data_train$O3_ug_m3),1))
```

```
## [1] 0.08586057
```

La variància de la diferència estacional és:

```
var(diff(log(full_data_train$O3_ug_m3),24))
```

```
## [1] 0.2067875
```

La variància de la diferència regular i estacional és:

```r
var(diff(diff(log(full_data_train$O3_ug_m3),1),24))
```

```
## [1] 0.09503709
```

Podem veure que amb la diferenciació regular d'ordre 1 obtenim la menor variància.

Igual que s'ha fet amb PM10, amb O3 també aplicarem el test d'arrels unitàries per a comprovar l'estacionarietat d'una sèrie temporal desestacionalizadas de Dikey-Fuller augmentat (ADF). Si el p-valor és molt petit (< 0.05) es rebutja la hipòtesi nul · la, per tant ens indicarà que no cal diferenciar-la.

```r
adf.test(diff(log(full_data_train$O3_ug_m3),1))
```

```
## Warning in adf.test(diff(log(full_data_train$O3_ug_m3), 1)): p-value
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(full_data_train$O3_ug_m3), 1)
## Dickey-Fuller = -12.823, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

Veiem que hem de rebutjar la H0 (cal diferenciar la sèrie) ja que tot i que ens dóna un p-valor de 0.01, el missatge ens avisa que el p-valor és inferior a aquest valor imprès. Per tant, sembla que caldrà només fer una diferència regular tal i com s'ha fet.

Anomenem a la nostre sèrie de treball, sèrie logtransformada i diferenciada regularment i estacional 'serie_arima_O3'.

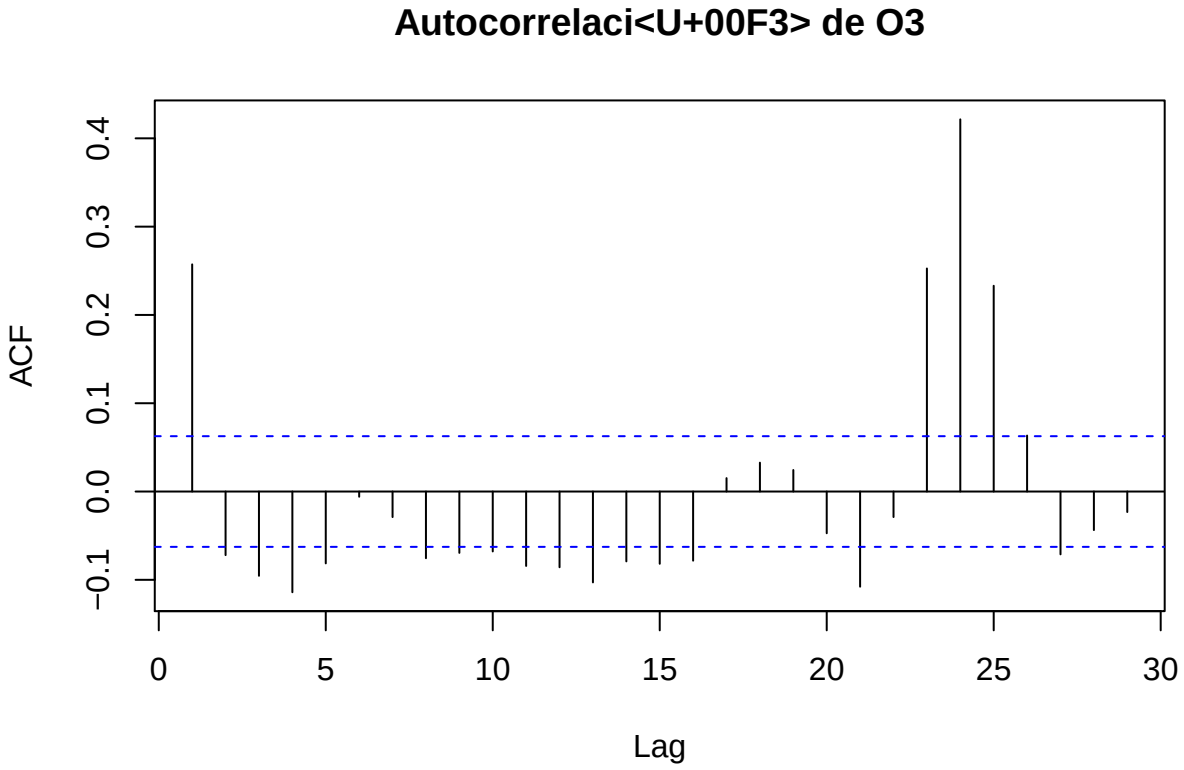
```r
serie_arima_O3 <- diff(log(full_data_train$O3_ug_m3),1)
```

7.2.3) Identificació del tipus de model ARIMA per a O3

Per tal de modelar la sèrie, anem a representar els diagrames de les funcions d'autocorrelació (ACF) i d'autocorrelació parcial (PACF).

El diagrama de funcions d'autocorrelació (ACF) el podem representar amb la següent funció.

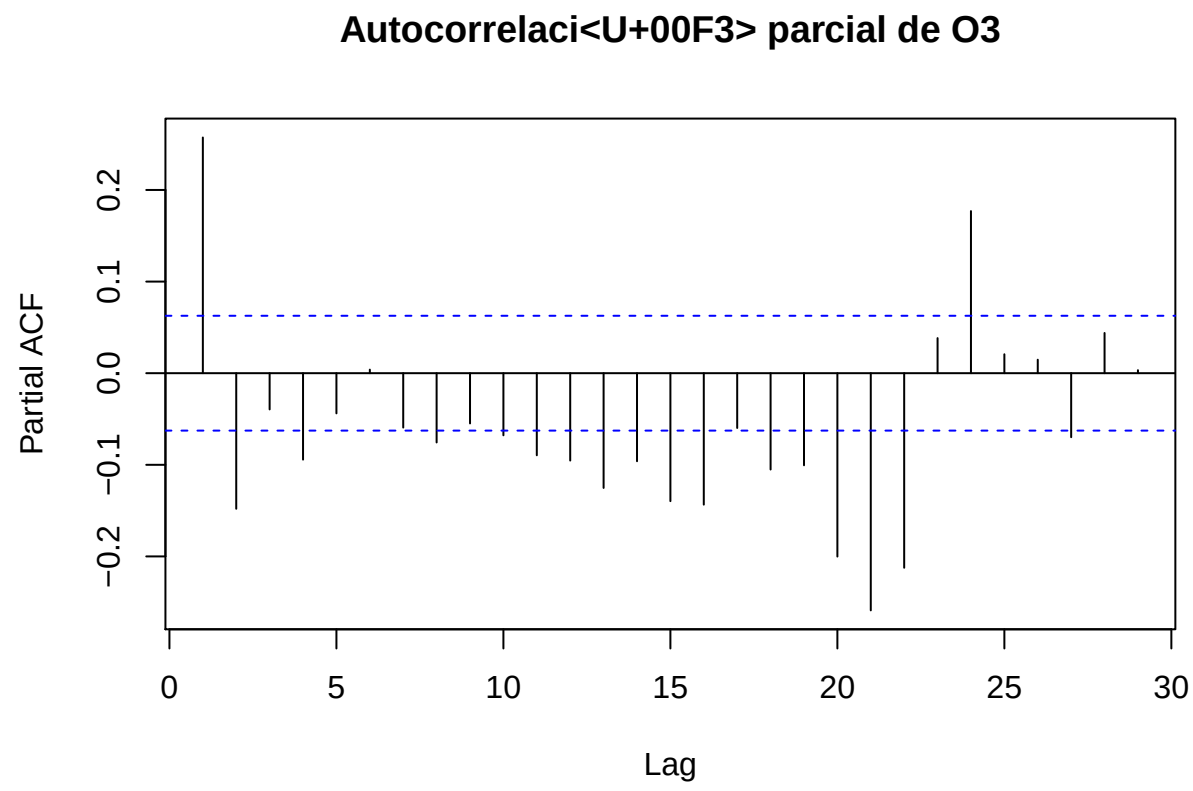
```r
acf(serie_arima_O3, main="Autocorrelació de O3")
```



Cada barra representa un retard. Veiem que hi ha correlacions negatives significatives al final de la gràfica.

El diagrama d'autocorrelació parcial (PACF) el podem representar amb la següent funció.

```
pacf(serie_arima_O3, main="Autocorrelació parcial de O3")
```



Per identificar el tipus de ARIMA s'utilitzarà la funció 'auto.arima', igual que s'ha fet en l'apartat anterior amb la variable PM10.

```
arima_O3 <-auto.arima(serie_arima_O3, d=0, D=0, max.p=5, max.q=5, max.P=2, max.Q=2)
arima_O3
```

```
## Series: serie_arima_O3
## ARIMA(5,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5
##       0.2843  -0.1509  -0.0223  -0.0780  -0.0482
## s.e.  0.0321   0.0333   0.0338   0.0337   0.0325
##
## sigma^2 estimated as 0.07767:  log likelihood=-135.75
## AIC=283.5   AICc=283.59   BIC=312.82
```

Podem veure que el model que ens ajusta per a la nostra sèrie temporal 'serie_arima_O3' és un AR(5) i amb un AIC de 283.5.

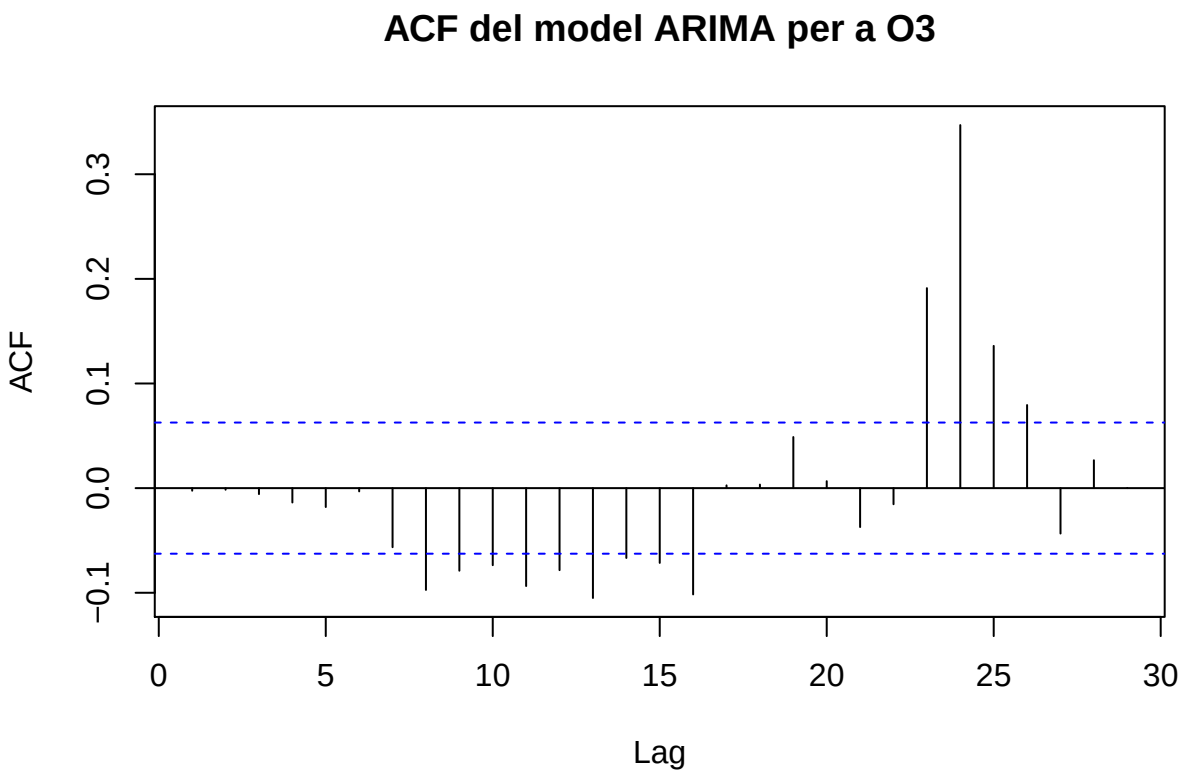
7.2.4) Validació del model ARIMA per a O3

Igual que en l'apartat anterior, primer es farà la validació del model ARIMA per O3 i posteriorment la predicció de nous valors.

Primer de tot es realitzarà la validació del model. La ACF i la PACF dels residus han de ser molt semblants, no mostrar estructura i tenir gairebé tots els valors dins de les bandes de confiança.

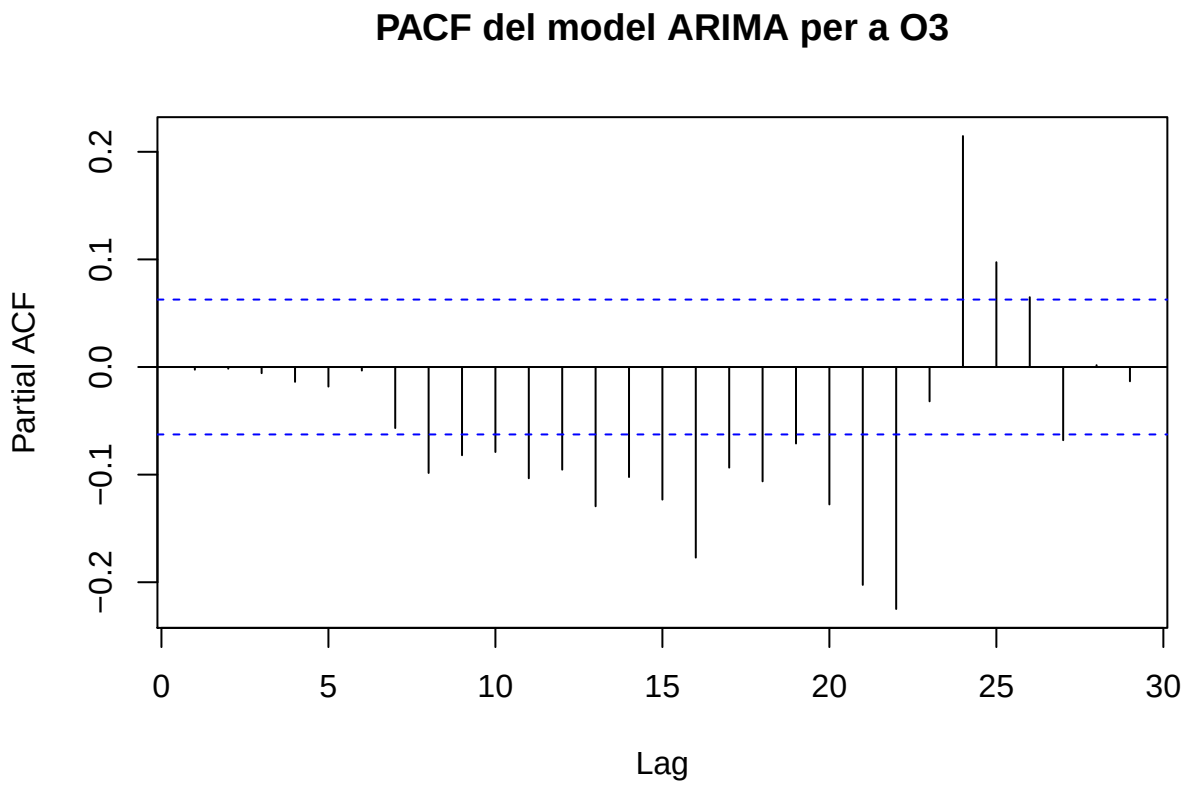
Per calcular els residus estandarditzats ho farem amb la funció 'rstandard' igual que s'ha fet anteriorment.

```
acf(rstandard(arima_O3), main = "ACF del model ARIMA per a O3")
```



A diferència de PM10, en aquest cas tenim molts més valors que estan a fora les bandes de confiança.

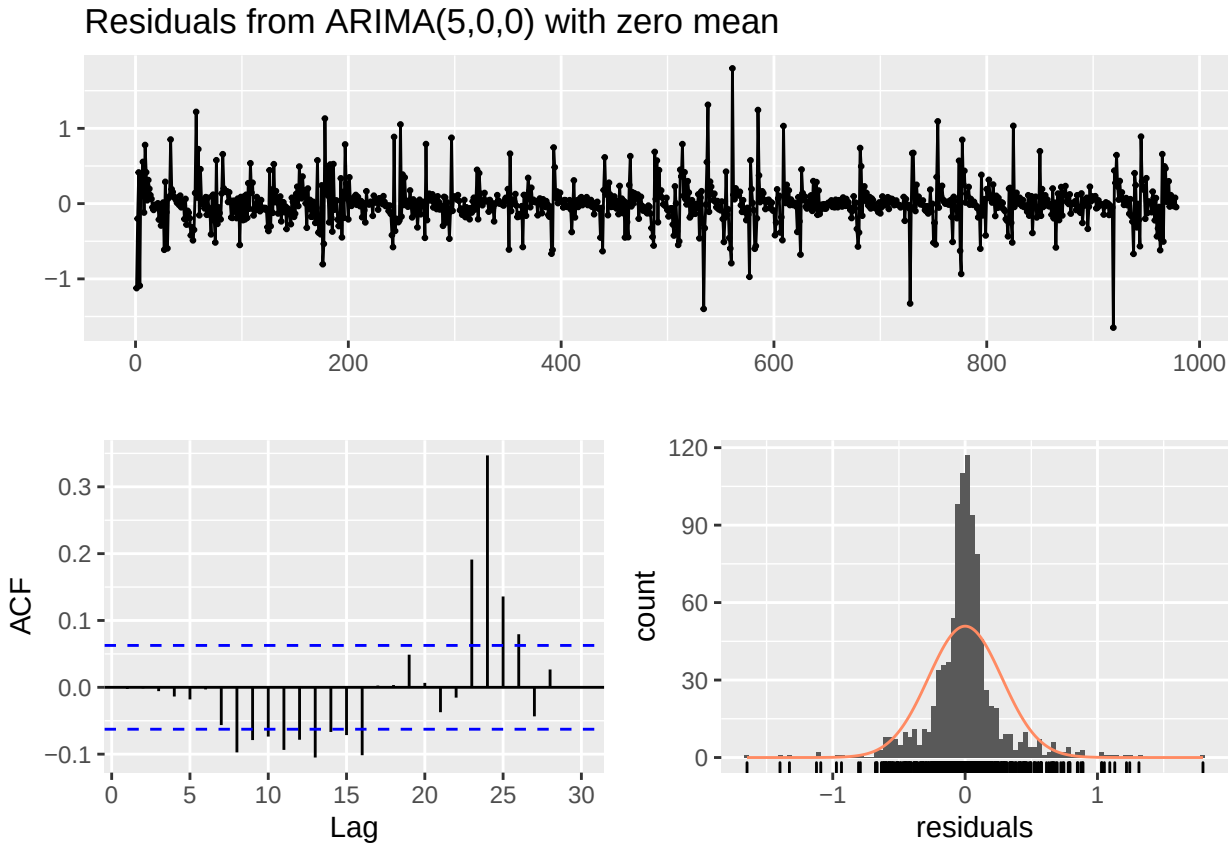
```
pacf(rstandard(arima_O3), main = "PACF del model ARIMA per a O3")
```



Amb la PACF també podem veure que la majoria dels valors estan a fora de les bandes de confiança.

Una altre forma per veure els residus és utlitzant la funció 'checkresiduals'.

```
checkresiduals(arima_O3)
```

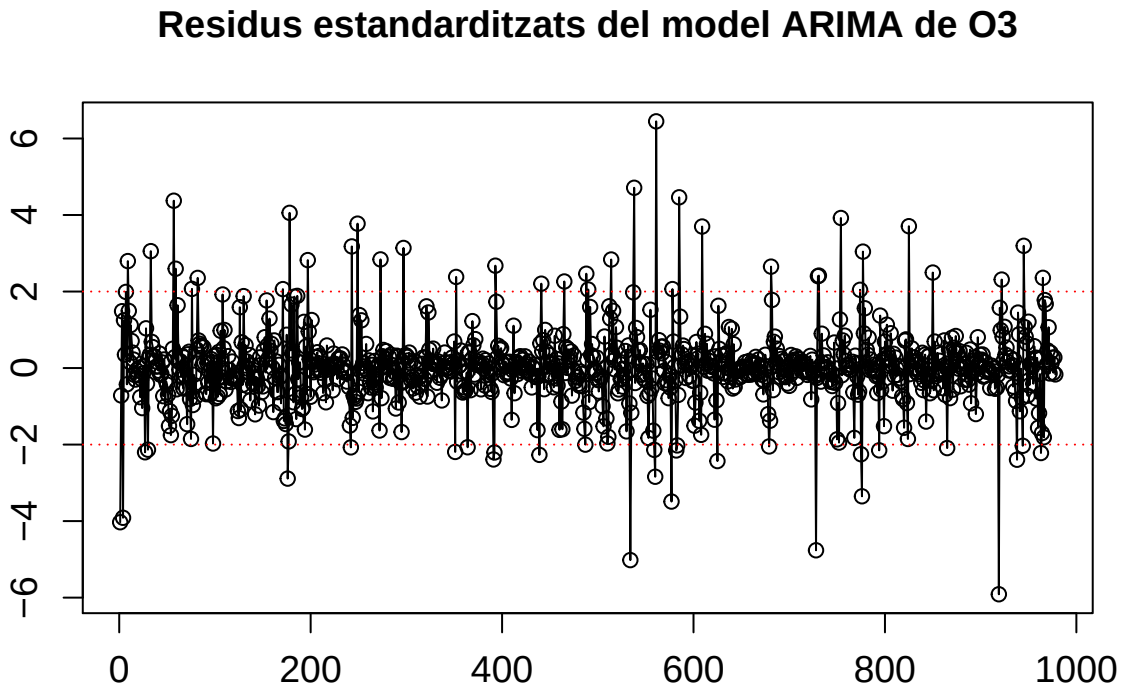



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,0,0) with zero mean
## Q* = 24.594, df = 5, p-value = 0.0001669
##
## Model df: 5.   Total lags used: 10
```

Com podem veure en el contrast de Ljung-Box-Pierce, hem obtingut un p-value inferior a 0.05 i això significarà que alguna de les correlacions és diferent de zero i, per tant, no es pot assumir que els residus siguin soroll blanc.

El gràfic dels residus ha de mostrar que els residus varien al voltant del zero, sense tendències, la variància és constant i no hi ha valors atípics. Aproximadament el 95% dels residus estandarditzats han d'estar entre -2 i 2 desviacions típiques.

```
par(mfcol=c(1,1), cex.axis=1.2, cex.main=1.2, cex.lab=1.2)
plot(rstandard(arima_O3), xlab="", ylab="", main="", type="o")
title("Residus estandarditzats del model ARIMA de O3")
abline(h=2, lty=3, col="red")
abline(h=-2, lty=3, col="red")
```



Com podem veure hi ha una gran part dels residus que esta fora els marges de 2 i -2. Per calcular-los ho farem amb la següent comanda:

```
sum(abs(rstandard(arima_O3))<=2)
```

```
## [1] 913
```

Tenim 913 dels 978 valors que estan a dins de les 2 desviacions típiques. Aixo representa un 93.3537832% del total de residus que està a dins de (-2,2).

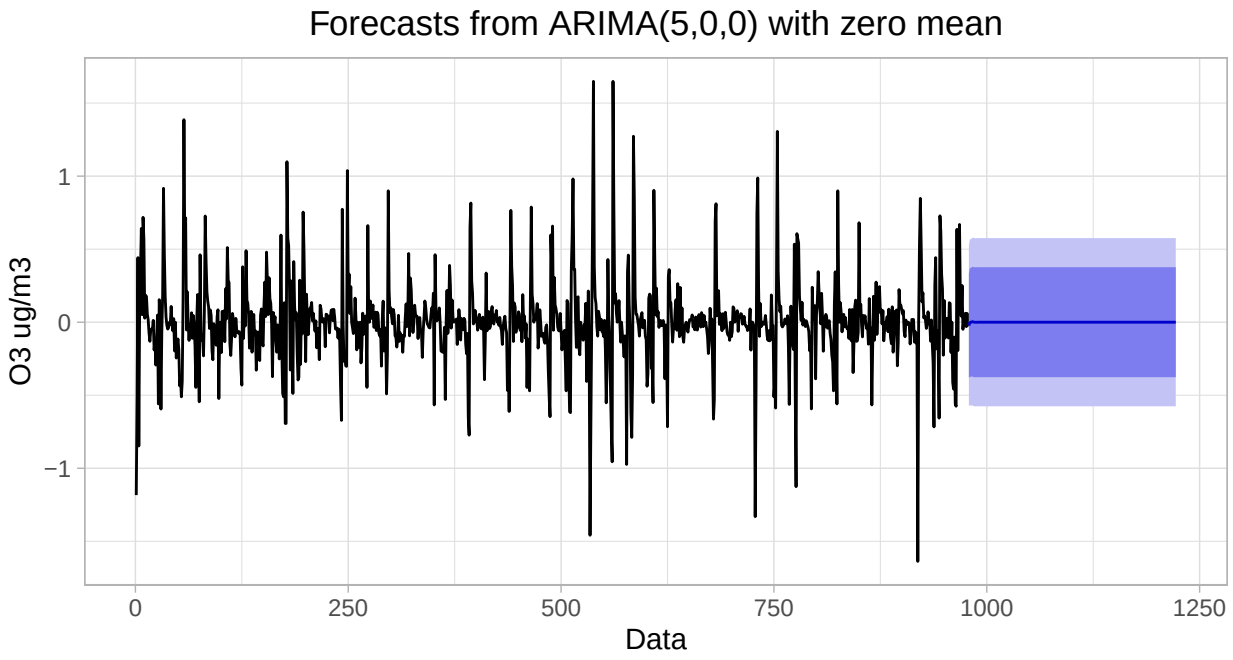
7.2.5) Predicció de nous valors amb el model ARIMA per a O3

Per realitzar les prediccions s'utilitzarà la funció 'forecast' i es passarà com a paràmetre el model ARIMA i també el nombre de dades que es vol predir, en aquest cas serà el nombre files del dataset test.

```
#Create the forecast
result_arima_O3 <- forecast(arima_O3, nrow(full_data_test))
```

Un cop ja hem creat el forecast mostrarem les dades que s'han predit amb la funció 'autoplot'.

```
#Plot the forecast of the data
autoplot(result_arima_O3) +
  Plot_SetTheme() +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center")
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

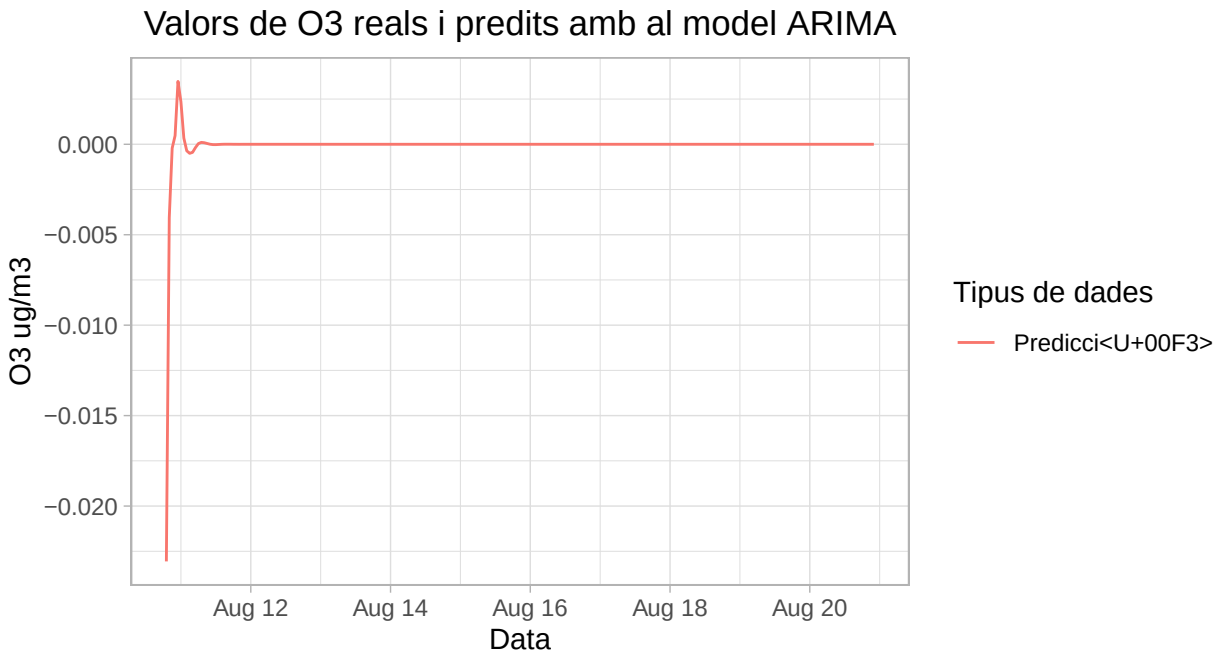
Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ARIMA_O3_1.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

També podem mostrar únicament els valors predits amb la llibreria 'ggplot'.

```
#Plot with the real values and the prediction
ggplot() +
  #geom_line(data = full_data_test, aes(x = airrmeasur_datetime, y = log(O3_ug_m3), color = "Valors rea
  geom_line(data = as.data.frame(result_arima_O3), aes(x = full_data_test$airrmeasur_datetime, y = resul
  Plot_SetTheme() +
  Plot_AddTitle("Valors de O3 reals i predits amb al model ARIMA") +
  Plot_SetPosTitle("center") +
  Plot_SetTextX("Data") +
  Plot_SetTextY("O3 ug/m3") +
  Plot_AddFooter() +
  Plot_SetPosFotter("center") +
  labs(color = "Tipus de dades")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Creat per en Robert Garcia Ventura el dia 04−09−2019

Dades proporcionades per la Generalitat de Catalunya.

Per m..s info: https://www.respira.cat

```
#Save the plot as a PNG image
ggsave("img/plot_model_ARIMA_O3_2.png", width = 14, height = 8, dpi = 150, units = "in", device='png')
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```