Air quality in Madrid (Spain) - From 2001 to 2018

Wagner Rosa

Introduction

Air pollution in the big metropoles around the world is one of the big concerns nowadays, since we can observe the negative effects of the pollutants in health and climate. Therefore, a control of the particles and gases with high potential to be harmful to our environment is needed. For that matter, Madrid is one of Europe's big capitals and has a series of air quality control stations distributed around the city, which can provide us a nice piece of information (and lots of data) about the pollutants presented at the Spanish capital. As a spanish myself and have lived in Madrid for many years, this topic is of very importance personally.

Recently, Madrid has launch the campaing "Madrid Central" in 2018, Wikipedia (https://en.wikipedia.org/wiki/Madrid_Central), which determine what type of vehicles are permitted to traffic in the capital's main area, depending on the year of the vehicle, motorization and so on. For instance, Madrid expect to reduce the pollution on Central area significantly. Hopefully, in this Capstone project, we will be able to predict if the ongoing trend is already a decline on pollutants or, worst case scenario, an increasing of pollutant in the spanish capital.

All the data sets were collected from Kaggle (https://www.kaggle.com/decide-soluciones/air-quality-madrid)

Loading the necessary libraries

```
#Loading the necessary libraries
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------------------------- tidyver
se 1.3.0 --
```

```
## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.3
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------ tidyverse_con
flicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: lubridate
```

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
if(!require(forecast)) install.packages("forecast", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

Now let's load the CSV files

```
#Setting the right directory path and importing csv files
setwd("C:/Users/DeOliW19830/Documents/RProjects/air-quality-madrid/air-quality-madrid")
stations.code <- read.csv("stations.csv", sep = ",")
```

Exploratory data analysis (EDA)

```
glimpse(stations.code)
```

```
## Observations: 24
## Variables: 6
## $ id        <int> 28079004, 28079008, 28079011, 28079016, 28079017, 28079018,…
## $ name      <fct> Pza. de EspaÃ±a, Escuelas Aguirre, Avda. RamÃ³n y Cajal, Ar…
## $ address   <fct> "Plaza de EspaÃ±a", "Entre C/ AlcalÃ¡ y C/ Oâ\200\231 Donel…
## $ lon       <dbl> -3.712247, -3.682319, -3.677356, -3.639233, -3.713322, -3.7…
## $ lat       <dbl> 40.42385, 40.42156, 40.45148, 40.44005, 40.34714, 40.39478,…
## $ elevation <int> 635, 670, 708, 693, 604, 630, 642, 621, 659, 685, 698, 674,…
```

```
#Assigning the list of csv files to a variable
files <- list.files(
  path = "C:/Users/DeOliW19830/Documents/RProjects/air-quality-madrid/air-quality-madrid/csvs
_per_year",
  pattern = "*.csv",
  full.names = T)

#Looping through the files and joining all the data sets to a single data frame
madrid.airquality <- sapply(files, read_csv, simplify=FALSE) %>% bind_rows
```

```
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
```

```
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
```

```
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
```

```
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   MXY = col_double(),
##   NMHC = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   OXY = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   PXY = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
```

```
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
```

```
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
```

```
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CH4 = col_logical(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   NOx = col_logical(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
```

```
## Warning: 59228 parsing failures.
##   row col           expected             actual
file
## 87457 NOx 1/0/T/F/TRUE/FALSE 37.0               'C:/Users/DeOliW19830/Documents/RProjects/
air-quality-madrid/air-quality-madrid/csvs_per_year/madrid_2017.csv'
## 87458 CH4 1/0/T/F/TRUE/FALSE 1.2200000286102295 'C:/Users/DeOliW19830/Documents/RProjects/
air-quality-madrid/air-quality-madrid/csvs_per_year/madrid_2017.csv'
## 87458 NOx 1/0/T/F/TRUE/FALSE 60.0               'C:/Users/DeOliW19830/Documents/RProjects/
air-quality-madrid/air-quality-madrid/csvs_per_year/madrid_2017.csv'
## 87459 NOx 1/0/T/F/TRUE/FALSE 19.0               'C:/Users/DeOliW19830/Documents/RProjects/
air-quality-madrid/air-quality-madrid/csvs_per_year/madrid_2017.csv'
## 87460 NOx 1/0/T/F/TRUE/FALSE 43.0               'C:/Users/DeOliW19830/Documents/RProjects/
air-quality-madrid/air-quality-madrid/csvs_per_year/madrid_2017.csv'
## ..... ... .................. ..................
.....................................................................................

## See problems(...) for more details.
```

```
## Parsed with column specification:
## cols(
##   date = col_datetime(format = ""),
##   BEN = col_double(),
##   CH4 = col_double(),
##   CO = col_double(),
##   EBE = col_double(),
##   NMHC = col_double(),
##   NO = col_double(),
##   NO_2 = col_double(),
##   NOx = col_double(),
##   O_3 = col_double(),
##   PM10 = col_double(),
##   PM25 = col_double(),
##   SO_2 = col_double(),
##   TCH = col_double(),
##   TOL = col_double(),
##   station = col_double()
## )
```

Looking into the file data frame format.

```
str(madrid.airquality, give.attr=F)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 3808224 obs. of  19 variables:
##  $ date   : POSIXct, format: "2001-08-01 01:00:00" "2001-08-01 01:00:00" ...
##  $ BEN    : num  NA 1.5 NA NA NA ...
##  $ CO     : num  0.37 0.34 0.28 0.47 0.39 ...
##  $ EBE    : num  NA 1.49 NA NA NA ...
##  $ MXY    : num  NA 4.1 NA NA NA ...
##  $ NMHC   : num  NA 0.07 NA NA NA ...
##  $ NO_2   : num  58.4 56.2 50.7 69.8 22.8 ...
##  $ NOx    : num  87.2 75.2 61.4 73.4 24.8 ...
##  $ OXY    : num  NA 2.11 NA NA NA ...
##  $ O_3    : num  34.5 42.2 46.3 40.7 66.3 ...
##  $ PM10   : num  105 100.6 100.1 69.8 75.2 ...
##  $ PXY    : num  NA 1.73 NA NA NA ...
##  $ SO_2   : num  6.34 8.11 7.85 6.46 8.8 ...
##  $ TCH    : num  NA 1.24 NA NA NA ...
##  $ TOL    : num  NA 10.8 NA NA NA ...
##  $ station: num  28079001 28079035 28079003 28079004 28079039 ...
##  $ PM25   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NO     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ CH4    : num  NA NA NA NA NA NA NA NA NA NA ...
```

Getting the names from the columns for each data frame.

```
colnames(madrid.airquality)
```

```
##  [1] "date"    "BEN"     "CO"      "EBE"     "MXY"     "NMHC"    "NO_2"
##  [8] "NOx"     "OXY"     "O_3"     "PM10"    "PXY"     "SO_2"    "TCH"
## [15] "TOL"     "station" "PM25"    "NO"      "CH4"
```

The names are related to the following measurements: - SO_2: sulphur dioxide level measured in μg/m³. High levels of sulphur dioxide can produce irritation in the skin and membranes, and worsen asthma or heart diseases in sensitive groups.

- CO: carbon monoxide level measured in mg/m³. Carbon monoxide poisoning involves headaches, dizziness and confusion in short exposures and can result in loss of consciousness, arrhythmias, seizures or even death in the long term.
- NO: nitric oxide level measured in μg/m³. This is a highly corrosive gas generated among others by motor vehicles and fuel burning processes.
- NO_2: nitrogen dioxide level measured in μg/m³. Long-term exposure is a cause of chronic lung diseases, and are harmful for the vegetation.

PM25: particles smaller than 2.5 μm level measured in μg/m³. The size of these particles allow them to penetrate into the gas exchange regions of the lungs (alveolus) and even enter the arteries. Long-term exposure is proven to be related to low birth weight and high blood pressure in newborn babies.

- PM10: particles smaller than 10 μm. Even though the cannot penetrate the alveolus, they can still penetrate through the lungs and affect other organs. Long term exposure can result in lung cancer and cardiovascular complications.
- NOx: nitrous oxides level measured in μg/m³. Affect the human respiratory system worsening asthma or other diseases, and are responsible of the yellowish-brown color of photochemical smog.
- O_3: ozone level measured in μg/m³. High levels can produce asthma, bronchytis or other chronic pulmonary diseases in sensitive groups or outdoor workers.
- TOL: toluene (methylbenzene) level measured in μg/m³. Long-term exposure to this substance (present in tobacco smkoke as well) can result in kidney complications or permanent brain damage.
- BEN: benzene level measured in μg/m³. Benzene is a eye and skin irritant, and long exposures may result in several types of cancer, leukaemia and anaemias. Benzene is considered a group 1 carcinogenic to humans by the IARC.
- EBE: ethylbenzene level measured in μg/m³. Long term exposure can cause hearing or kidney problems and the IARC has concluded that long-term exposure can produce cancer.
- MXY: m-xylene level measured in μg/m³. Xylenes can affect not only air but also water and soil, and a long exposure to high levels of xylenes can result in diseases affecting the liver, kidney and nervous system (especially memory and affected stimulus reaction).
- PXY: p-xylene level measured in μg/m³. See MXY for xylene exposure effects on health.
- OXY: o-xylene level measured in μg/m³. See MXY for xylene exposure effects on health.
- TCH: total hydrocarbons level measured in mg/m³. This group of substances can be responsible of different blood, immune system, liver, spleen, kidneys or lung diseases.
- CH4: methane level measured in mg/m³. This gas is an asphyxiant, which displaces the oxygen animals need to breath. Displaced oxygen can result in dizzinnes, weakness, nausea and loss of coordination.
- NMHC: non-methane hydrocarbons (volatile organic compounds) level measured in mg/m³. Long exposure to some of these substances can result in damage to the liver, kidney, and central nervous system. Some of them are suspected to cause cancer in humans.

As pointed in an article from Iberdrola (https://www.iberdrola.com/medio-ambiente/contaminacion-calidad-aire) - in spanish - the control standard is usally done measuring the levels of NO2, O3, CO, SO2. We are going to also look into PM10 and PM25, which are related to particles smaller than 10 adn 2.5 μm, respectively. This is

because the long exposure to these particles can can result in lung cancer and cardiovascular complications (PM10); low birth weight and high blood pressure in newborn babies (PM25).

Other reason to choose these level is because they are presented in all the data frames.

Summary of the pollutant measurments through the years

```
summary(madrid.airquality)
```

```
##       date                          BEN                CO
##  Min.   :2001-01-01 01:00:00   Min.   : 0.0      Min.   : 0.0
##  1st Qu.:2005-02-10 19:00:00   1st Qu.: 0.2      1st Qu.: 0.3
##  Median :2009-04-11 17:00:00   Median : 0.6      Median : 0.4
##  Mean   :2009-06-21 04:25:17   Mean   : 1.3      Mean   : 0.6
##  3rd Qu.:2013-10-17 23:15:00   3rd Qu.: 1.5      3rd Qu.: 0.6
##  Max.   :2018-05-01 00:00:00   Max.   :66.4      Max.   :18.0
##                                NA's   :2766540   NA's   :1157212
##       EBE               MXY               NMHC              NO_2
##  Min.   :  0.0     Min.   :  0       Min.   :0.0       Min.   :  0.00
##  1st Qu.:  0.3     1st Qu.:  1       1st Qu.:0.1       1st Qu.: 24.00
##  Median :  0.9     Median :  3       Median :0.2       Median : 44.00
##  Mean   :  1.4     Mean   :  5       Mean   :0.2       Mean   : 50.47
##  3rd Qu.:  1.6     3rd Qu.:  6       3rd Qu.:0.2       3rd Qu.: 69.58
##  Max.   :162.2     Max.   :178       Max.   :9.1       Max.   :628.60
##  NA's   :2806500   NA's   :3492809   NA's   :2722912   NA's   :21174
##       NOx               OXY               O_3               PM10
##  Min.   :   0.0    Min.   :  0       Min.   :  0.0    Min.   :  0.0
##  1st Qu.:  40.1    1st Qu.:  1       1st Qu.: 12.7    1st Qu.: 11.5
##  Median :  76.2    Median :  1       Median : 34.9    Median : 21.5
##  Mean   : 109.0    Mean   :  2       Mean   : 39.8    Mean   : 28.9
##  3rd Qu.: 139.4    3rd Qu.:  3       3rd Qu.: 60.0    3rd Qu.: 37.8
##  Max.   :2537.0    Max.   :103       Max.   :236.0    Max.   :695.0
##  NA's   :1484767   NA's   :3492529   NA's   :816492   NA's   :946969
##       PXY               SO_2              TCH               TOL
##  Min.   :  0       Min.   :  0.0     Min.   : 0.0      Min.   :  0.0
##  1st Qu.:  1       1st Qu.:  5.8     1st Qu.: 1.3      1st Qu.:  1.1
##  Median :  1       Median :  8.1     Median : 1.4      Median :  3.2
##  Mean   :  2       Mean   : 10.7     Mean   : 1.4      Mean   :  5.9
##  3rd Qu.:  3       3rd Qu.: 12.3     3rd Qu.: 1.5      3rd Qu.:  7.0
##  Max.   :106       Max.   :199.1     Max.   :10.5      Max.   :242.9
##  NA's   :3492640   NA's   :1032264   NA's   :2721783   NA's   :2769295
##     station              PM25               NO               CH4
##  Min.   :28079001   Min.   :-31.0     Min.   :   0.0    Min.   :0
##  1st Qu.:28079014   1st Qu.:  6.4     1st Qu.:   2.0    1st Qu.:1
##  Median :28079024   Median : 11.0     Median :   6.0    Median :1
##  Mean   :28079029   Mean   : 13.7     Mean   :  23.4    Mean   :1
##  3rd Qu.:28079040   3rd Qu.: 17.7     3rd Qu.:  20.0    3rd Qu.:1
##  Max.   :28079099   Max.   :506.9     Max.   :1146.0    Max.   :4
##                     NA's   :2991800   NA's   :2275827   NA's   :3799784
```

Getting the date right

```
#Ordering by date
madrid.airquality <- madrid.airquality[with(madrid.airquality, order(date)),]
madrid.airquality$full.date.time <- as.POSIXct(madrid.airquality$date,format = "%Y-%m-%d %H:%
M:%S", tz='CET')
madrid.airquality$date <- format(madrid.airquality$full.date.time, "%Y-%m-%d")
madrid.airquality$time <- format(madrid.airquality$full.date.time, "%T")
madrid.airquality$day <- format(madrid.airquality$full.date.time, "%d")
madrid.airquality$month <- format(madrid.airquality$full.date.time, "%m")
madrid.airquality$year <- format(madrid.airquality$full.date.time, "%Y")
```

Getting the total number of NA's in the data frame

```
#Total number of NA's
sum(is.na(madrid.airquality))
```

```
## [1] 38791297
```

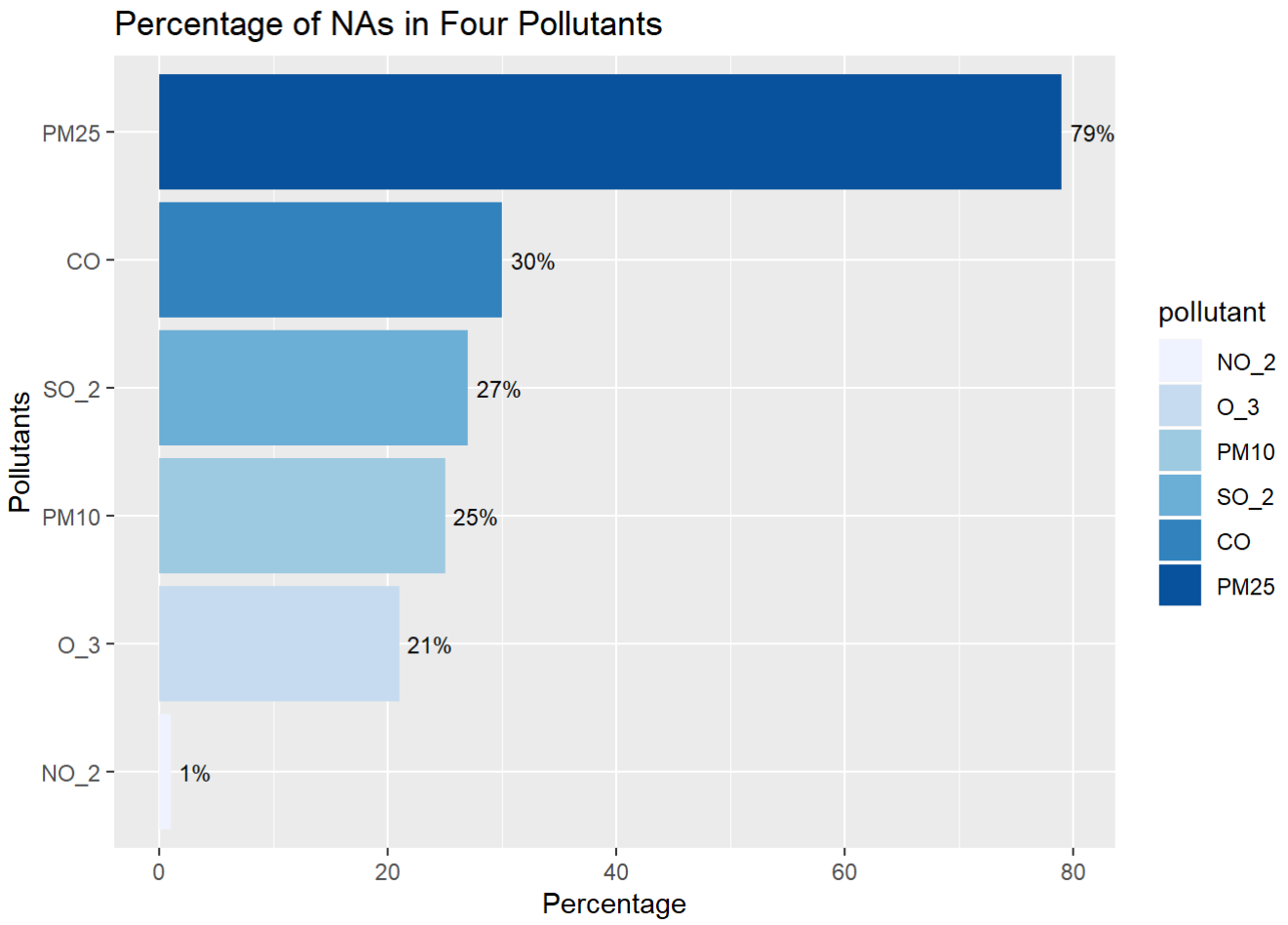
Now, let's filter the data to plot the amount of NA's for our selected indicators. For this, we will plot a graph to see the percentage of NA's in the data frame subset

```
indicators <- c("NO_2", "O_3", "SO_2", "CO", "PM10", "PM25")

#filtering only the subset of pollutants we want to analyze
madrid.airquality.indicators <- madrid.airquality[, indicators]

na.indicators <- data.frame(percent=round(
  colSums(
    is.na(madrid.airquality.indicators))/nrow(madrid.airquality.indicators)*100)
  )

na.indicators
```

```
##      percent
## NO_2       1
## O_3       21
## SO_2      27
## CO        30
## PM10      25
## PM25      79
```

Let's plot it into a bar graph to visualize the variation in NA's percentage for each pollutant measured in this historical series

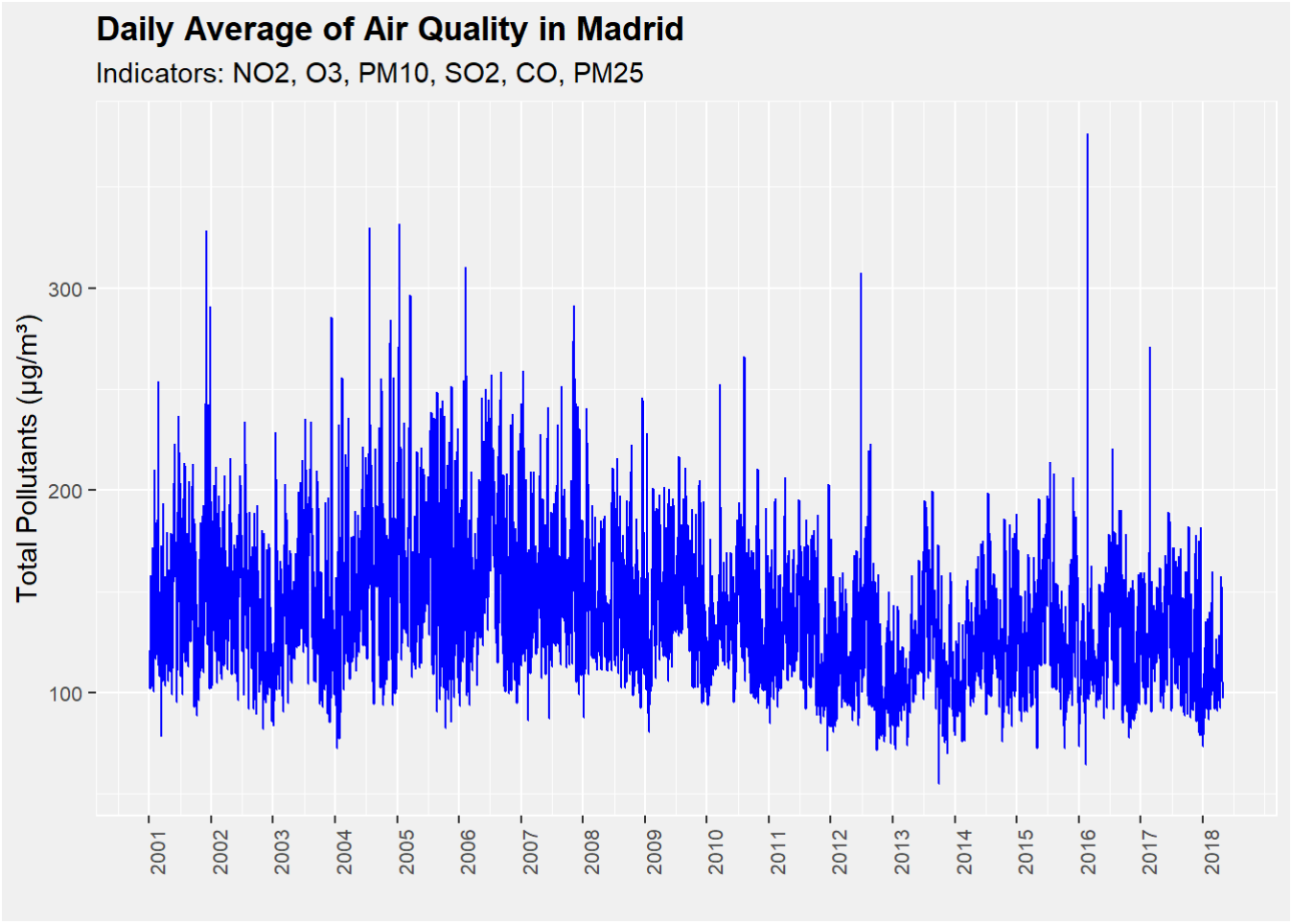
```
na.indicators$pollutant <- rownames(na.indicators)
na.indicators$pollutant<-factor(na.indicators$pollutant,
                                as.character(na.indicators$pollutant))

na.indicators %>%
  mutate(pollutant = fct_reorder(pollutant, percent)) %>%
  ggplot(aes(pollutant, percent,fill = pollutant)) +
  geom_bar(stat = "identity") +
  scale_fill_brewer() +
  geom_text(aes(label=paste0(percent,"%"), y=percent+0.7),
                size = 3,
                hjust = -0.01
                ) +
  labs(x = "Pollutants",
       y = "Percentage",
       title = "Percentage of NAs in Four Pollutants") +
  coord_flip()
```



PM25 has a considerable large number of NA's with 79% rate. The other pollutants have values between 20 and 30 percent, with only exeception of NO2 (1%)

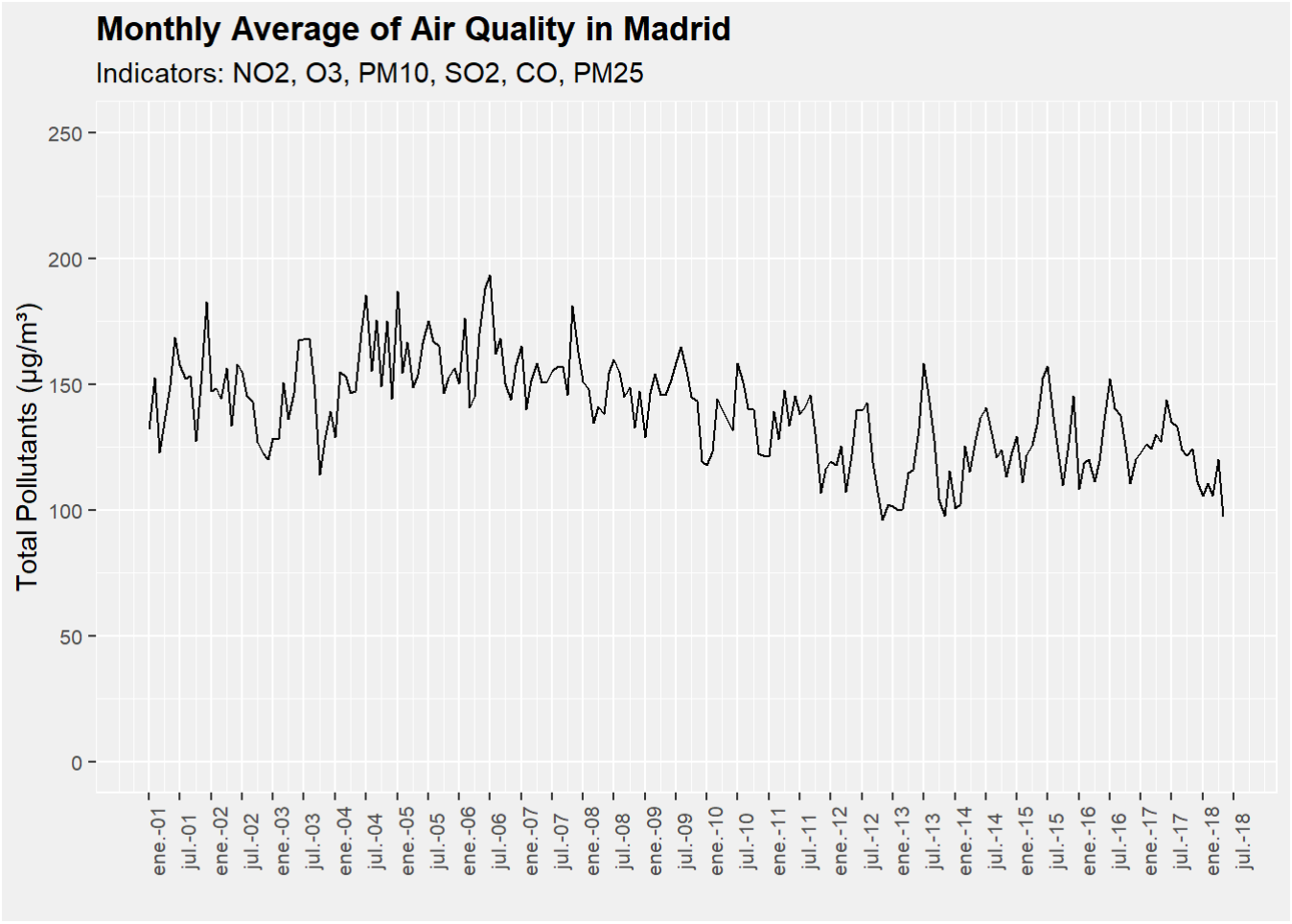
Now, let's take a look on how these pollutants/indicators vary over time.

```
#Daily variation
madrid.daily <- madrid.airquality %>%
  group_by(date) %>%
  summarise(NO_2.mean = mean(NO_2, na.rm = T),
            O_3.mean = mean(O_3, na.rm = T),
            PM10.mean = mean(PM10, na.rm = T),
            SO_2.mean = mean(SO_2, na.rm = T),
            CO.mean = mean(CO, na.rm = T),
            PM25.mean = mean(PM25, na.rm = T))

#Getting the total pollutant
madrid.daily <- mutate(madrid.daily,
                       tot_poll = rowSums(madrid.daily[,-1],
                       na.rm = T))

#Plottind the data frame
madrid.daily %>%
  ggplot(aes(x = as.Date(date),
             y = tot_poll)) +
    geom_line(color = 'blue') +
    theme(legend.position = "none",
          panel.background = element_rect(fill = "gray94", colour = "white",
                                          size = 0.5, linetype = "solid"),
          legend.background = element_rect(fill = "gray94"),
          plot.background = element_rect(fill = "gray94"),
          panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour =
"white"),
          panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour =
"white"),
          plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
    labs(x = '',
         y = 'Total Pollutants (µg/m³)',
         title='Daily Average of Air Quality in Madrid',
         subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
    theme(axis.text.y =element_text(angle = 0,
                                    size = 8),
          axis.text.x=element_text(angle = 90,
                                   size = 8)) +
    scale_x_date(breaks = seq(as.Date("2001-01-01"),
                              as.Date("2018-07-01"),
                              by="12 months"),
                 date_labels = "%Y")
```



To look how the average total pollution varies through the months, we're going to use the same approach as the one applied for the daily variation:

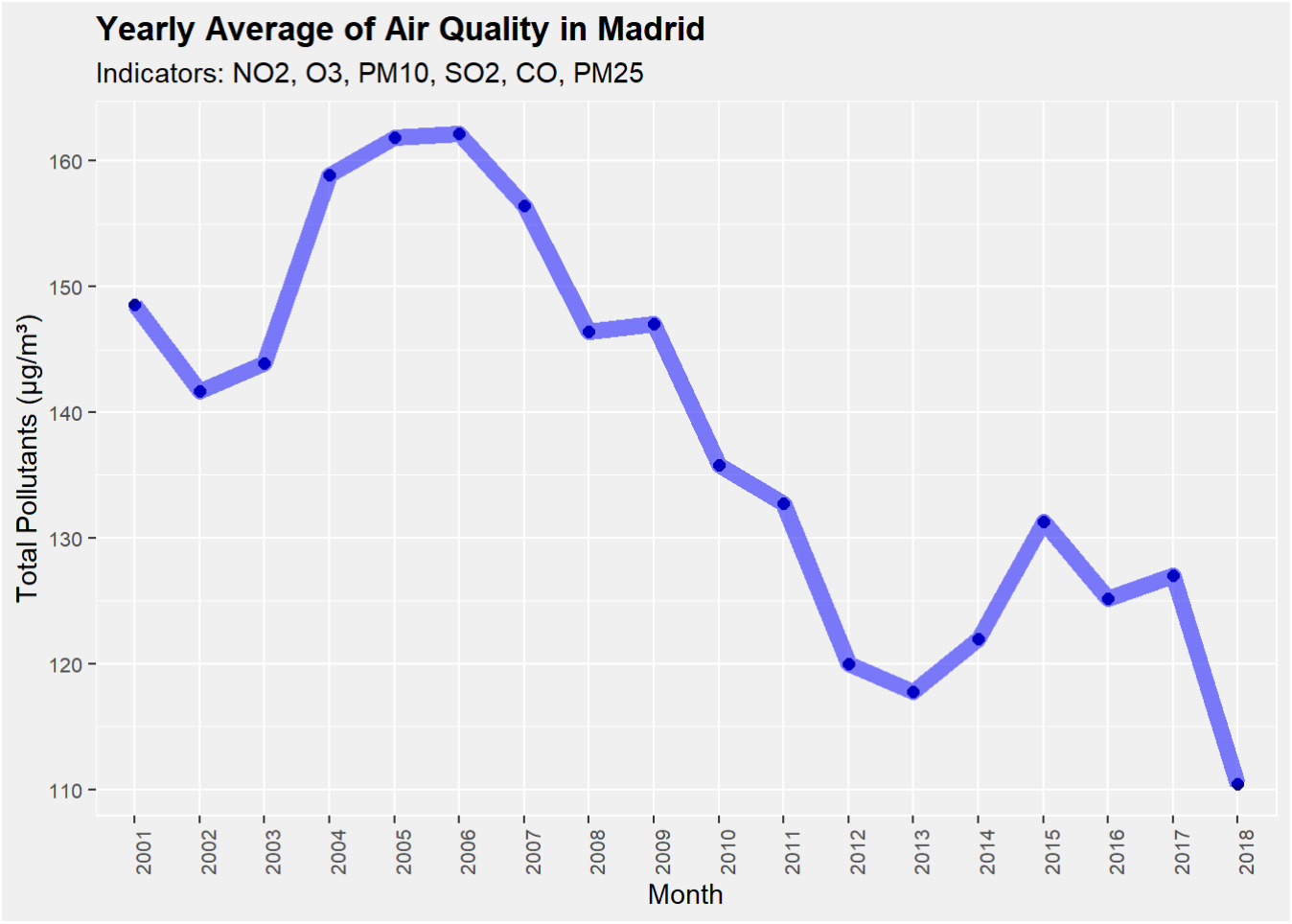
```
#Monthly variation
madrid.monthly <- madrid.airquality %>%
  group_by(year, month) %>%
  summarise(NO_2.mean = mean(NO_2, na.rm = T),
            O_3.mean = mean(O_3, na.rm = T),
            PM10.mean = mean(PM10, na.rm = T),
            SO_2.mean = mean(SO_2, na.rm = T),
            CO.mean = mean(CO, na.rm = T),
            PM25.mean = mean(PM25, na.rm = T)) %>%
  mutate(time = paste(year, "-", month, "- 01", sep = ""))

#Getting the total pollutant
madrid.monthly$tot_poll_mon <- rowSums(madrid.monthly[,3:8], na.rm = T) # this selects only t
he columns with the averages. In order to check it out, uncomment the next code line:
#str(madrid.monthly)
madrid.monthly$time_mon = as.Date(madrid.monthly$time, format = "%Y-%m-%d")

#Plottind the data frame
madrid.monthly %>%
  ggplot(aes(x = time_mon,
             y = tot_poll_mon)) +
    geom_line(color = 'black') +
    theme(legend.position = "none",
          panel.background = element_rect(fill = "gray94", colour = "white",
                                          size = 0.5, linetype = "solid"),
          legend.background = element_rect(fill = "gray94"),
          plot.background = element_rect(fill = "gray94"),
          panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour =
"white"),
          panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour =
"white"),
          plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
    labs(x = '',
         y = 'Total Pollutants (µg/m³)',
         title='Monthly Average of Air Quality in Madrid',
         subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
    theme(axis.text.y =element_text(angle = 0,
                                    size = 8),
          axis.text.x=element_text(angle = 90,
                                   size = 8)) +
    scale_x_date(breaks = seq(as.Date("2001-01-01"),
                              as.Date("2018-07-01"),
                              by="6 months"),
                 date_labels = "%b-%y") +
    scale_y_continuous(breaks = seq(0,250,50), lim = c(0,250))
```



This time series looks like a smooth out curve compared with the daily averages, as expected. In this plot, we can observe the presence of some peaks, which appear with certain frequency. This can be an indication of seasonal effects in the total concentration of pollutants.

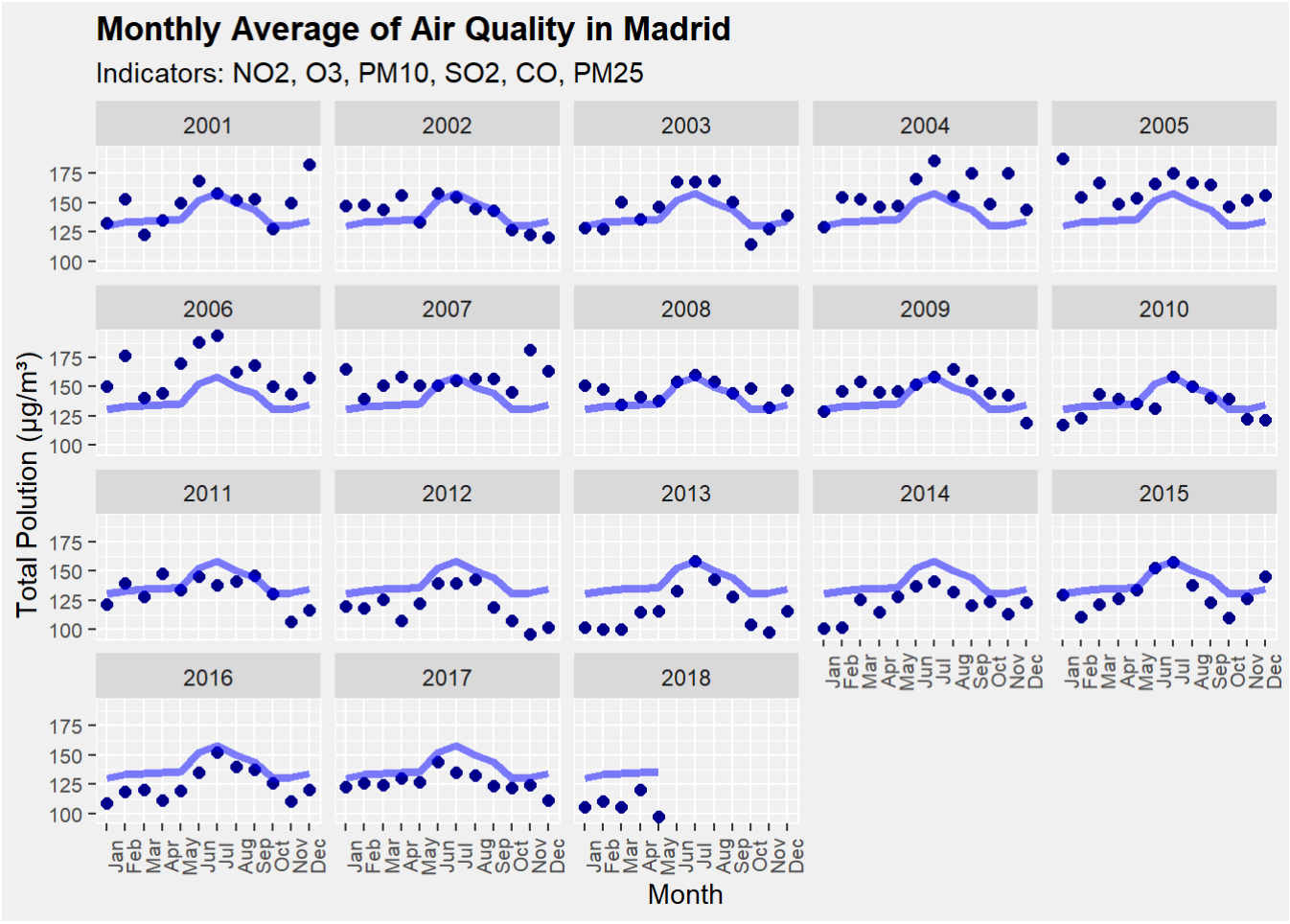
Let’s look at the yearly moving average

```
#Yearly variation
madrid.yearly <- madrid.airquality %>%
  select(year, station, NO_2, O_3, PM10, SO_2, CO, PM25) %>%
  group_by(year) %>%
  summarise(NO_2.mean = mean(NO_2, na.rm = T),
            O_3.mean = mean(O_3, na.rm = T),
            PM10.mean = mean(PM10, na.rm = T),
            SO_2.mean = mean(SO_2, na.rm = T),
            CO.mean = mean(CO, na.rm = T),
            PM25.mean = mean(PM25, na.rm = T))

#Getting the total pollutant
madrid.yearly$tot_poll_year <- rowSums(madrid.yearly[,2:7], na.rm = T)# this selects only the
columns with the averages. In order to check it out, uncomment the next code line:
#str(madrid.yearly)

#Plottind the data frame
madrid.yearly %>%
  ggplot(aes(year, tot_poll_year, group = 1)) +
    geom_point(aes(year, tot_poll_year), size = 2, color = 'darkblue') +
    geom_line(size = 3, alpha = 0.5, color = 'blue') +
    theme(legend.position = "none",
        panel.background = element_rect(fill = "gray94",
                                        colour = "white",
                                        size = 0.5,
                                        linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5,
                                        linetype = 'solid',
                                        colour = "white"),
        panel.grid.minor = element_line(size = 0.25,
                                        linetype = 'solid',
                                        colour = "white"),
        plot.title = element_text(hjust = 0,
                                  face = 'bold',
                                  color = 'black')) +
  labs(x = 'Month',
       y = 'Total Pollutants (µg/m³)',
       title='Yearly Average of Air Quality in Madrid',
       subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  theme(axis.text.y =element_text(angle = 0,
                                  size = 8),
          axis.text.x=element_text(angle = 90,
                                   size = 8))
```



After looking into the yearly moving average, we can notice a reduction tendency on the total number of pollutants. To look for seasonal effects, we have to evaluate how the pollutants vary during the year.

```
monthly <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "De
c")

madrid.monthly %>%
  group_by(month) %>%
  mutate(poll_mean = mean(tot_poll_mon)) %>%
  ggplot(aes(month, poll_mean, group = 1)) +
    geom_point(aes(month, tot_poll_mon), size = 2, color = 'darkblue') +
    geom_line(size = 1.5, alpha = 0.5, color = 'blue') +
    scale_x_discrete(labels = monthly) +
    theme(legend.position = "none",
        panel.background = element_rect(fill = "gray94",
                                        colour = "white",
                                        size = 0.5,
                                        linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5,
                                        linetype = 'solid',
                                        colour = "white"),
        panel.grid.minor = element_line(size = 0.25,
                                        linetype = 'solid',
                                        colour = "white"),
        plot.title = element_text(hjust = 0,
                                  face = 'bold',
                                  color = 'black')) +
  labs(x = 'Month',
       y = 'Total Polution (µg/m³)',
       title='Monthly Average of Air Quality in Madrid',
       subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  theme(axis.text.y = element_text(size = 8),
        axis.text.x=element_text(
          angle = 90,
          size = 8)) +
   facet_wrap(~year)
```



As one can observe, through all the years it seems that during the months of summer, approximately from May to September, the levels of pollutants are higher than at the rest of the year, with a peak around July. These are the months where the rain ocurrance is poor compared to the other periods of the year, therefore the dry weather can contribute to this behavior.

Making predictions - modelling algorithms

Simple exponential smooth (SES)

For our first model, we are going to use Exponential Smoothing method, which is an extension of the naive method, wherein the forecasts are produced using weighted averages of past observations, with the weights decaying exponentially as the observations get older. In simple words, higher weights are given to the more recent observations and vice versa. The value of the smoothing parameter for the level is decided by the parameter ‘alpha’.

In order to make predictions, we are goind to use the library forecast, which has all the functions needed.

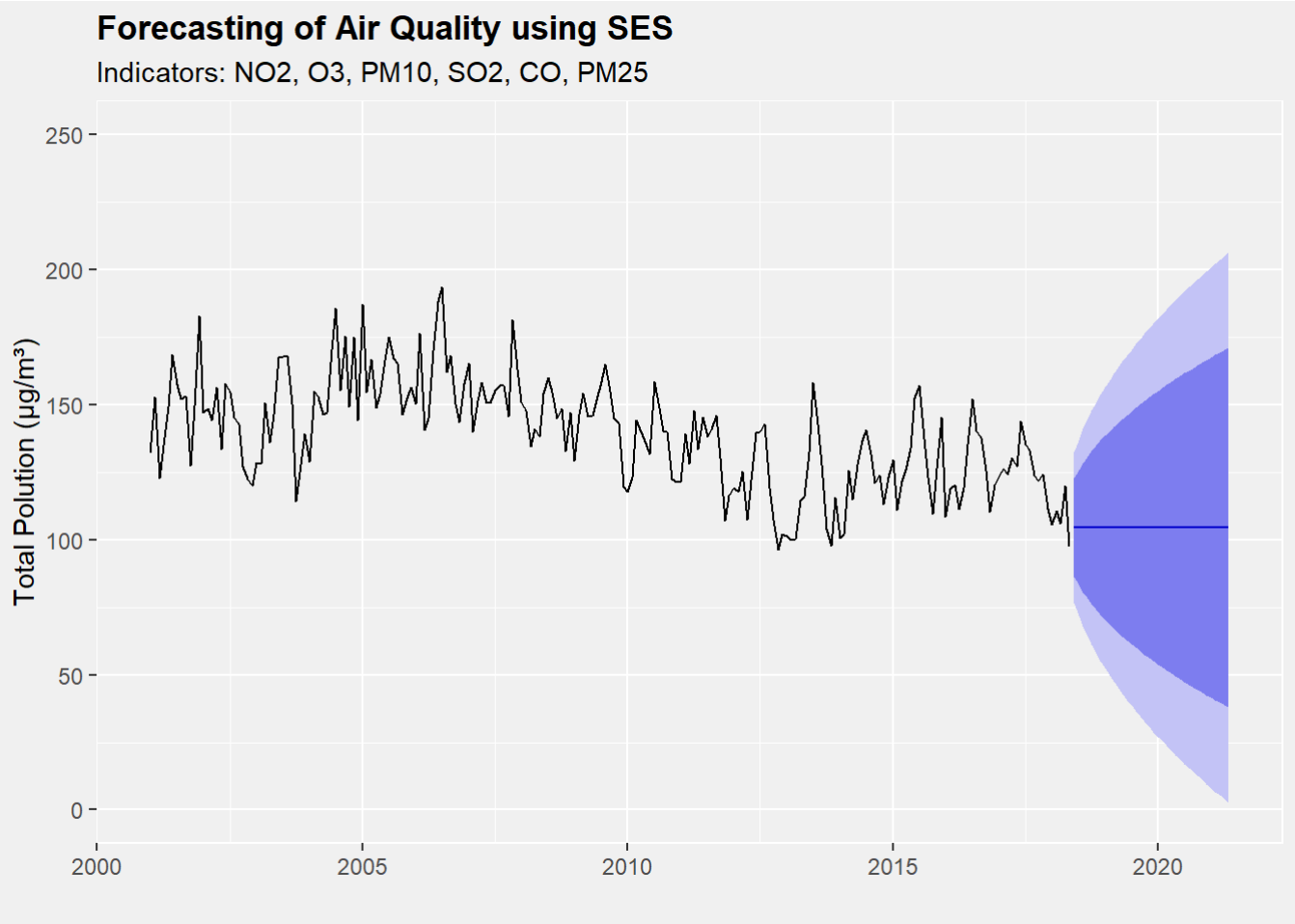
The “autoplot” function generates a ggplot object, therefore we can style it in the same way we have done so far.

```
#Predicting the total pollutant by month
library(forecast)
#The time series to be take into account. We are going to predict the total pollutants by mon
th
monthly.ts <- ts(madrid.monthly[,10],start=c(2001,1), end=c(2018,5), frequency = 12)

#Making predictions for the next 36 months
model.ses.ts <- ses(monthly.ts, h=36)
summary(model.ses.ts)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = monthly.ts, h = 36)
##
##   Smoothing parameters:
##     alpha = 0.5954
##
##   Initial states:
##     l = 136.9716
##
##   sigma:  14.1935
##
##      AIC     AICc      BIC
## 2229.403 2229.520 2239.430
##
## Error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.2608008 14.12546 11.17419 -0.9447616 8.110637 0.8498069
##                    ACF1
## Training set 0.05503343
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jun 2018       104.5165 86.32676 122.7063 76.697684 132.3353
## Jul 2018       104.5165 83.34649 125.6865 72.139755 136.8933
## Aug 2018       104.5165 80.73684 128.2962 68.148647 140.8844
## Sep 2018       104.5165 78.38654 130.6465 64.554170 144.4788
## Oct 2018       104.5165 76.23086 132.8022 61.257340 147.7757
## Nov 2018       104.5165 74.22821 134.8048 58.194561 150.8385
## Dec 2018       104.5165 72.35001 136.6830 55.322098 153.7109
## Jan 2019       104.5165 70.57559 138.4574 52.608347 156.4247
## Feb 2019       104.5165 68.88943 140.1436 50.029589 159.0034
## Mar 2019       104.5165 67.27954 141.7535 47.567483 161.4655
## Apr 2019       104.5165 65.73643 143.2966 45.207498 163.8255
## May 2019       104.5165 64.25242 144.7806 42.937893 166.0951
## Jun 2019       104.5165 62.82119 146.2118 40.749016 168.2840
## Jul 2019       104.5165 61.43748 147.5955 38.632821 170.4002
## Aug 2019       104.5165 60.09686 148.9362 36.582516 172.4505
## Sep 2019       104.5165 58.79553 150.2375 34.592303 174.4407
## Oct 2019       104.5165 57.53023 151.5028 32.657189 176.3758
## Nov 2019       104.5165 56.29811 152.7349 30.772838 178.2602
## Dec 2019       104.5165 55.09671 153.9363 28.935452 180.0976
## Jan 2020       104.5165 53.92383 155.1092 27.141685 181.8913
## Feb 2020       104.5165 52.77753 156.2555 25.388572 183.6444
## Mar 2020       104.5165 51.65608 157.3769 23.673466 185.3596
## Apr 2020       104.5165 50.55794 158.4751 21.993998 187.0390
## May 2020       104.5165 49.48170 159.5513 20.348035 188.6850
## Jun 2020       104.5165 48.42611 160.6069 18.733648 190.2994
## Jul 2020       104.5165 47.39002 161.6430 17.149088 191.8839
## Aug 2020       104.5165 46.37239 162.6606 15.592758 193.4403
## Sep 2020       104.5165 45.37227 163.6607 14.063202 194.9698
## Oct 2020       104.5165 44.38878 164.6442 12.559085 196.4739
```

```
## Nov 2020       104.5165 43.42112 165.6119 11.079177 197.9538
## Dec 2020       104.5165 42.46855 166.5645  9.622346 199.4107
## Jan 2021       104.5165 41.53038 167.5026  8.187545 200.8455
## Feb 2021       104.5165 40.60599 168.4270  6.773803 202.2592
## Mar 2021       104.5165 39.69477 169.3382  5.380220 203.6528
## Apr 2021       104.5165 38.79619 170.2368  4.005958 205.0271
## May 2021       104.5165 37.90973 171.1233  2.650234 206.3828
```

```
#Plotting the predictions
model.ses.ts %>%
  autoplot() +
  theme(axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray94", colour = "white",
                                        size = 0.5, linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"),
        panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"),
        plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
  labs(x = '', y = 'Total Polution (µg/m³)', title='Forecasting of Air Quality using SES', su
btitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  scale_y_continuous(breaks = seq(0,250,50), lim = c(0,250))
```



Holt's trend

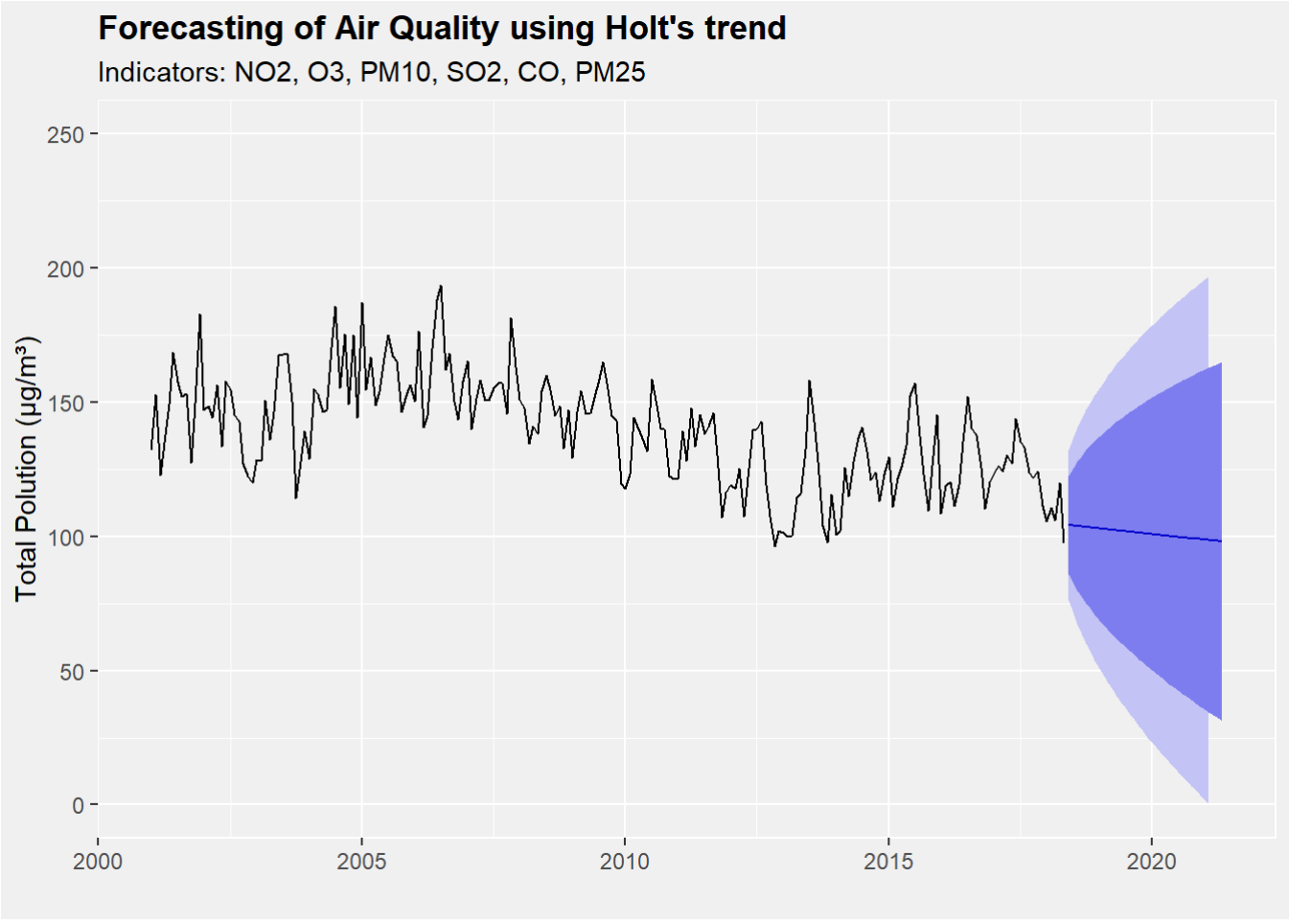
This is an extension of the simple exponential smoothing method which considers the trend component while generating forecasts. This method involves two smoothing equations, one for the level and one for the trend component.

```
#Making predictions for the next 36 months
model.holt.ts <- holt(monthly.ts, h=36)
summary(model.holt.ts)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = monthly.ts, h = 36)
##
##   Smoothing parameters:
##     alpha = 0.5931
##     beta  = 1e-04
##
##   Initial states:
##     l = 144.7659
##     b = -0.173
##
##   sigma:  14.2728
##
##      AIC     AICc      BIC
## 2233.703 2233.998 2250.414
##
## Error measures:
##                       ME    RMSE      MAE        MPE     MAPE      MASE
## Training set -0.03716885 14.1356 11.17017 -0.7814792 8.099194 0.8495011
##                    ACF1
## Training set 0.05541874
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80      Lo 95    Hi 95
## Jun 2018      104.25691 85.96553 122.5483 76.2826646 132.2312
## Jul 2018      104.08313 82.81585 125.3504 71.5576239 136.6086
## Aug 2018      103.90935 80.03342 127.7853 67.3942637 140.4244
## Sep 2018      103.73557 77.50844 129.9627 63.6246317 143.8465
## Oct 2018      103.56179 75.17685 131.9467 60.1507765 146.9728
## Nov 2018      103.38801 72.99745 133.7786 56.9096526 149.8664
## Dec 2018      103.21423 70.94183 135.4866 53.8578501 152.5706
## Jan 2019      103.04045 68.98947 137.0914 50.9639708 155.1169
## Feb 2019      102.86667 67.12496 138.6084 48.2044337 157.5289
## Mar 2019      102.69289 65.33635 140.0494 45.5609950 159.8248
## Apr 2019      102.51911 63.61421 141.4240 43.0191962 162.0190
## May 2019      102.34533 61.95088 142.7398 40.5673480 164.1233
## Jun 2019      102.17155 60.34008 144.0030 38.1958408 166.1473
## Jul 2019      101.99777 58.77658 145.2190 35.8966604 168.0989
## Aug 2019      101.82399 57.25594 146.3920 33.6630403 169.9849
## Sep 2019      101.65021 55.77440 147.5260 31.4892060 171.8112
## Oct 2019      101.47643 54.32869 148.6242 29.3701823 173.5827
## Nov 2019      101.30265 52.91600 149.6893 27.3016465 175.3037
## Dec 2019      101.12887 51.53384 150.7239 25.2798148 176.9779
## Jan 2020      100.95509 50.18004 151.7301 23.3013525 178.6088
## Feb 2020      100.78131 48.85267 152.7100 21.3633029 180.1993
## Mar 2020      100.60753 47.54999 153.6651 19.4630299 181.7520
## Apr 2020      100.43375 46.27048 154.5970 17.5981712 183.2693
## May 2020      100.25997 45.01273 155.5072 15.7666005 184.7533
## Jun 2020      100.08619 43.77548 156.3969 13.9663951 186.2060
## Jul 2020       99.91241 42.55761 157.2672 12.1958104 187.6290
## Aug 2020       99.73863 41.35806 158.1192 10.4532569 189.0240
```

```
## Sep 2020       99.56486 40.17590 158.9538  8.7372821 190.3924
## Oct 2020       99.39108 39.01024 159.7719  7.0465543 191.7356
## Nov 2020       99.21730 37.86029 160.5743  5.3798489 193.0547
## Dec 2020       99.04352 36.72530 161.3617  3.7360370 194.3510
## Jan 2021       98.86974 35.60461 162.1349  2.1140751 195.6254
## Feb 2021       98.69596 34.49757 162.8943  0.5129962 196.8789
## Mar 2021       98.52218 33.40360 163.6408 -1.0680977 198.1124
## Apr 2021       98.34840 32.32214 164.3746 -2.6300422 199.3268
## May 2021       98.17462 31.25270 165.0965 -4.1736166 200.5228
```

```
#Plotting the predictions
model.holt.ts %>%
  autoplot() +
  theme(axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray94", colour = "white",
                                        size = 0.5, linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"),
        panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"),
        plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
  labs(x = '', y = 'Total Polution (µg/m³)', title='Forecasting of Air Quality using Holt\'s
 trend', subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  scale_y_continuous(breaks = seq(0,250,50), lim = c(0,250))
```



ARIMA

Autoregressive Integrated Moving Average, aka ARIMA, as the data set seems to have a periodic behavior. In the package forecast we have the function “auto.arima”, which fits the best ARIMA model to an univariate time series. In order to make predictions for the next 3 years, we need to use the “forecast” funtion passing the number of months to predict.

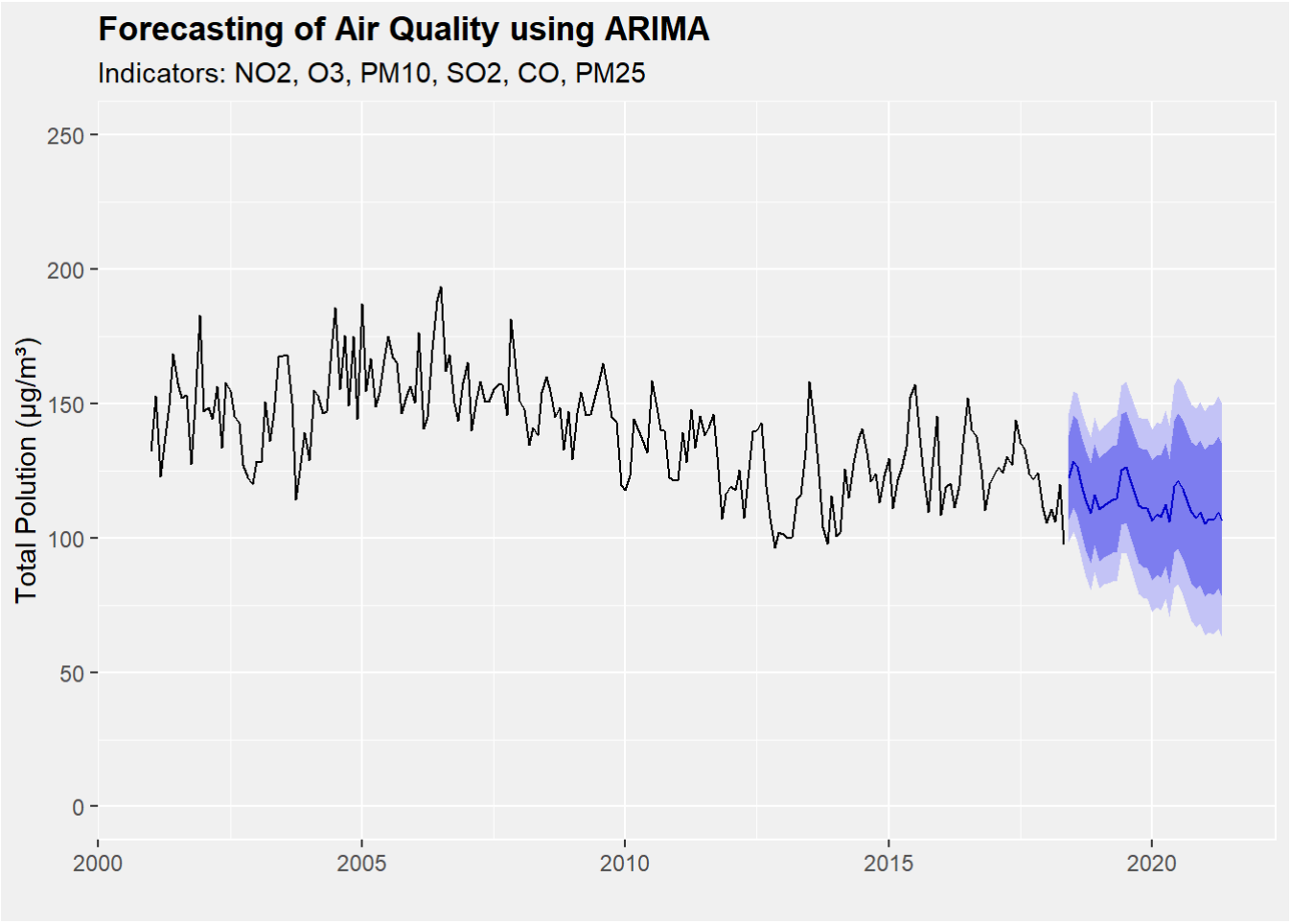
```
#Making predictions for the next 36 months
model.arima.ts <- auto.arima(monthly.ts)
summary(model.arima.ts)
```

```
## Series: monthly.ts
## ARIMA(5,1,1)(2,0,1)[12] with drift
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): Se han producido NaNs
```

```
##          ar1     ar2      ar3     ar4      ar5      ma1    sar1    sar2    sma1
##       0.2756  0.0796  -0.1368  0.0280  -0.0513  -0.8377  0.5289  0.2872  -0.488
## s.e.  0.0947  0.0854   0.0796  0.0828   0.0798   0.0759     NaN     NaN     NaN
##         drift
##       -0.1126
## s.e.   0.3717
##
## sigma^2 estimated as 149.3:  log likelihood=-813.8
## AIC=1649.6   AICc=1650.95   BIC=1686.31
##
## Training set error measures:
##                       ME     RMSE     MAE       MPE     MAPE      MASE
## Training set -0.02918269 11.89251 9.21039 -0.670073 6.724383 0.7004582
##                      ACF1
## Training set 0.001566434
```

```
#Plotting the predictions
model.arima.ts %>%
  forecast(h=36) %>%
  autoplot() +
  theme(axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray94", colour = "white",
                                        size = 0.5, linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"),
        panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"),
        plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
  labs(x = '', y = 'Total Polution (µg/m³)', title='Forecasting of Air Quality using ARIMA',
 subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  scale_y_continuous(breaks = seq(0,250,50), lim = c(0,250))
```



TBATS

The TBATS model combines several components of the already discussed techniques in this guide, making them a very good choice for forecasting. It constitutes the following elements:

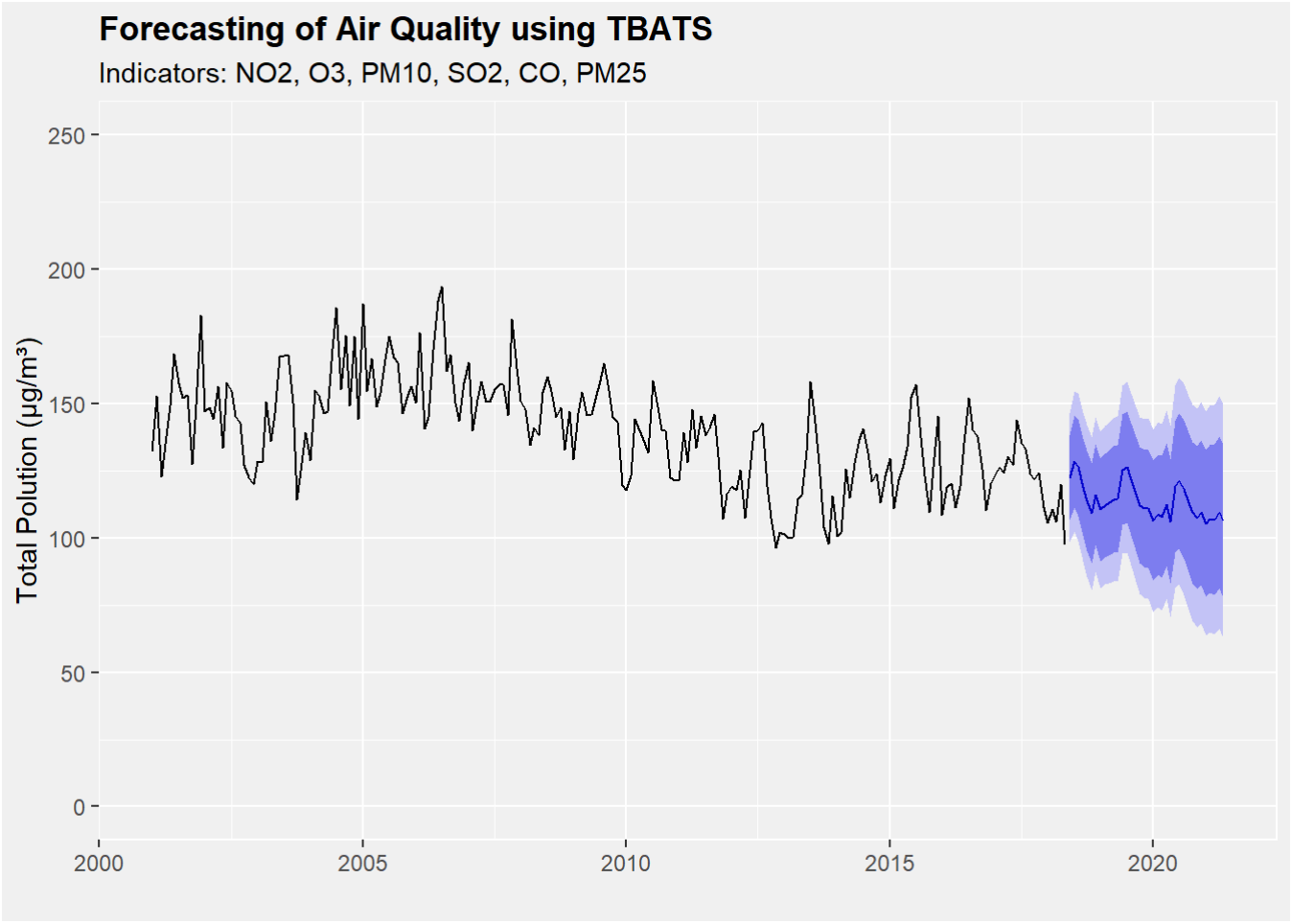
T: Trigonometric terms for seasonality
B: Box-Cox transformations for heterogeneity
A: Autoregressive Moving Average (ARMA) errors for short-term dynamics
T: Trend
S: Seasonal (including multiple and non-integer periods)

In this case, we need to use the “forecast” funtion passing the number of months to predict, same as we did for ARIMA.

```
#Making predictions for the next 36 months
model.tbats.ts <- tbats(monthly.ts)
summary(model.tbats.ts)
```

```
##                   Length Class  Mode
## lambda               0   -none- NULL
## alpha                1   -none- numeric
## beta                 0   -none- NULL
## damping.parameter    0   -none- NULL
## gamma.one.values     1   -none- numeric
## gamma.two.values     1   -none- numeric
## ar.coefficients      0   -none- NULL
## ma.coefficients      0   -none- NULL
## likelihood           1   -none- numeric
## optim.return.code    1   -none- numeric
## variance             1   -none- numeric
## AIC                  1   -none- numeric
## parameters           2   -none- list
## seed.states         11   -none- numeric
## fitted.values      209   ts     numeric
## errors             209   ts     numeric
## x                 2299   -none- numeric
## seasonal.periods     1   -none- numeric
## k.vector             1   -none- numeric
## y                  209   ts     numeric
## p                    1   -none- numeric
## q                    1   -none- numeric
## call                 2   -none- call
## series               1   -none- character
## method               1   -none- character
```

```
#Plotting the predictions
model.arima.ts %>%
  forecast(h=36) %>%
  autoplot() +
  theme(axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray94", colour = "white",
                                        size = 0.5, linetype = "solid"),
        legend.background = element_rect(fill = "gray94"),
        plot.background = element_rect(fill = "gray94"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"),
        panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"),
        plot.title = element_text(hjust = 0, face = 'bold',color = 'black')) +
  labs(x = '', y = 'Total Polution (µg/m³)', title='Forecasting of Air Quality using TBATS',
 subtitle='Indicators: NO2, O3, PM10, SO2, CO, PM25') +
  scale_y_continuous(breaks = seq(0,250,50), lim = c(0,250))
```



Conclusions

In summary, the analysis of the evolution of air quality in Madrid shows that the number of total pollutant are decreasing overall. We can also see in the time series, the seasonal effects that pumps the levels of pollutants up during the summer months, which are the months with lowest average rainfall rates (https://www.holiday-weather.com/madrid/averages/). In this brief analysis we only select 6 pollutants as being ones of the most important among them, nevertheless a broader comparison should be done in order to have an overall picture of the air quality in Madrid. From our models, we were able to note that the ARIMA and TBATS algorythms performs the best in the lowest RMSE's (around 11). Furthermore, as a future work, an analysis of the air quality as a function of the stations should be done. Such task can provide information of what part of Madrid is the less or more polluted, giving the total pollutant values.