

# Preprocesamiento AQI

(a) Limpiar los registros, guardar en un nuevo archivo .csv

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
data <- read.csv("C:/csv/PRSA_Data_20130301-20170228/PRSA_Data_Aotizhongxin_20130301-20170228.csv", head  
datos_limpios<- data[complete.cases(data),]  
head(datos_limpios,5)
```

```
##   No year month day hour PM2.5 PM10 SO2 NO2 CO O3 TEMP PRES DEWP RAIN wd  
## 1 1 2013      3  1    0      4   4   4  7 300 77 -0.7 1023.0 -18.8  0 NNW  
## 2 2 2013      3  1    1      8   8   4  7 300 77 -1.1 1023.2 -18.2  0  N  
## 3 3 2013      3  1    2      7   7   5 10 300 73 -1.1 1023.5 -18.2  0 NNW  
## 4 4 2013      3  1    3      6   6  11 11 300 72 -1.4 1024.5 -19.4  0  NW  
## 5 5 2013      3  1    4      3   3  12 12 300 72 -2.0 1025.2 -19.5  0  N  
##   WSPM      station  
## 1  4.4 Aotizhongxin  
## 2  4.7 Aotizhongxin  
## 3  5.6 Aotizhongxin  
## 4  3.1 Aotizhongxin  
## 5  2.0 Aotizhongxin
```

```
# Combinar las columnas "year", "month" y "day" en una columna de fecha  
datos_limpios <- datos_limpios %>%  
  unite(fecha, year, month, day, sep = "-")  
  
# Guardar los datos actualizados en otro archivo CSV  
write.csv(datos_limpios, file = "C:/csv/archivo_actualizado1.csv", row.names = FALSE)  
  
# Confirmar que se ha guardado correctamente  
cat("Los datos se han guardado en 'archivo_actualizado1.csv'")
```

```
## Los datos se han guardado en 'archivo_actualizado1.csv'
```

```
#datos_fecha <- read.csv("C:/csv/archivo_actualizado1.csv")
```

(b) Agrupar por fecha los gases.

```
library(dplyr)
```

```
datos_fecha <- read.csv("C:/csv/archivo_actualizado1.csv")
```

```
# Calcular la media de SO2, NO2 y PM10 agrupados por día
```

```
mean_so2 <- datos_fecha %>% group_by(station, fecha) %>% summarise(mean_so2 = mean(SO2), .groups= 'drop')
```

```
mean_no2 <- datos_fecha %>% group_by(station, fecha) %>% summarise(mean_no2 = mean(NO2), .groups= 'drop')
```

```
mean_pm10 <- datos_fecha %>% group_by(station, fecha) %>% summarise(mean_pm10 = mean(PM10), .groups= 'drop')
```

```
# Calcular el valor maximo de CO, o3
```

```
max_CO <- datos_fecha %>% group_by(station, fecha) %>% summarise(CO = max(CO), .groups= 'drop')
```

```
max_O3 <- datos_fecha %>% group_by(station, fecha) %>% summarise(O3 = max(O3), .groups= 'drop')
```

```
#.....
```

```
calculate_AQI_component <- function(concentration, pollutant) {
```

```
  if (pollutant == "PM10") {
```

```
    breakpoints <- c(0, 54, 154, 254, 354, 424, 504, 604)
```

```
  } else if (pollutant == "SO2") {
```

```
    breakpoints <- c(0, 91.7, 196.5, 484.7, 796.48, 1582.48, 2630.48)
```

```
  } else if (pollutant == "NO2") {
```

```
    breakpoints <- c(0, 99.64, 188, 676.8, 1220.12, 2348.12, 3852.12)
```

```
  } else if (pollutant == "CO") {
```

```
    breakpoints <- c(0, 5060, 10810, 14260, 17710, 34960, 57960)
```

```
  } else if (pollutant == "O3") {
```

```
    breakpoints <- c(0, 105.84, 137.2, 166.6, 205.8, 392, 604)
```

```
  } else {
```

```
    return(rep("Invalid pollutant", length(concentration)))
```

```
  }
```

```
index <- c(0, 50, 100, 150, 200, 300, 400, 500)
```

```
aqi <- rep(NA, length(concentration))
```

```
for (i in 1:(length(breakpoints) - 1)) {
```

```
  mask <- concentration >= breakpoints[i] & concentration <= breakpoints[i + 1]
```

```
  aqi[mask] <- ((index[i + 1] - index[i]) / (breakpoints[i + 1] - breakpoints[i])) * (concentration[mask] - breakpoints[i])
```

```
}
```

```
aqi[is.na(aqi)] <- "Out of range"
```

```
return(round(as.numeric(aqi)))
```

```
}
```

```
#.....Calculo el valor del AQI.....
```

```
aqi_so2 <- calculate_AQI_component(mean_so2$mean_so2, "SO2")
```

```

aqi_no2 <- calculate_AQI_component(mean_no2$mean_no2, "NO2")
aqi_pm10 <- calculate_AQI_component(mean_pm10$mean_pm10, "PM10")
aqi_CO <- calculate_AQI_component(max_CO$CO, "CO")
aqi_O3 <- calculate_AQI_component(max_O3$O3, "O3")

# Crear un data frame para almacenar los resultados del AQI
aqi_data <- data.frame( fecha = mean_so2$fecha, AQI_SO2 = aqi_so2, AQI_NO2 = aqi_no2, AQI_PM10 = aqi_pm10, AQI_CO = aqi_CO, AQI_O3 = aqi_O3)

# Imprimir el data frame con los resultados del AQI
#print(aqi_data)

df_datos <- aqi_data %>%
  rowwise() %>%
  mutate(AQI = max(c( AQI_PM10,AQI_SO2, AQI_NO2,AQI_CO,AQI_O3)))
  # ungroup()

write.csv(df_datos, file = "C:/Proyecto AQI/datos/calidad_aire3.csv", row.names = FALSE)

niveles <- c(0, 50, 100, 150, 200, 250, 300, 600)
labels <- c('Excelent', 'Good', 'Slightly Polluted', 'Lightly Polluted', 'Moderately Polluted', 'Heavily Polluted')

aqi_data$Nivel_Aire <- cut(df_datos$AQI,breaks = niveles, labels = labels)

nuevo_df <- df_datos %>%
  mutate(Nivel_Aire = cut(AQI, breaks = niveles, labels = labels))
write.csv(nuevo_df, file = "C:/Proyecto AQI/datos/calidad_aire4.csv", row.names = FALSE)
head(nuevo_df,4)

```

```

## # A tibble: 4 x 9
## # Rowwise:
##   fecha      AQI_SO2 AQI_NO2 AQI_PM10 AQI_CO AQI_O3 ciudad      AQI Nivel_Aire
##   <chr>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <chr>      <dbl> <fct>
## 1 2013-10-1 7      36     85     22     21 Aotizhongx~ 85 Good
## 2 2013-10-10 26     27     92     17     21 Aotizhongx~ 92 Good
## 3 2013-10-11 18     37     72     27     43 Aotizhongx~ 72 Good
## 4 2013-10-12 3      41     66     13     37 Aotizhongx~ 66 Good

```