Calidad y Análisis Exploratorio de los Datos

Índice:

1. **Cargar los datos**: Importación de archivos CSV y Excel.
2. **Descripciones necesarias**: Descripción de variables, web scraping y origen de los datos.
3. **Exploración de datos**: Distribución de contaminantes $PM_{2.5}$ y comparación entre ciudades y fuentes de emisión.
4. **Conclusiones sobre la calidad de los datos**: Problemas detectados y decisiones de limpieza.

1. Cargar los datos

Bibliotecas necesarias (actualizar una vez terminado el trabajo) y carga del primer excel:

```
library(readxl)
library(dplyr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(writexl)
library(knitr)
library(stringr)
library(rvest)
library(xml2)
library(tidyverse)
library(tidygeocoder)


df <- read_excel("Ciudades_origen_gases.xlsx")
summary(df)
```

```
##     CIUDAD           EXTRACCION            SECTOR          GAS
##  Length:174900      Length:174900      Min.   : 1.00   Length:174900
##  Class :character   Class :character   1st Qu.: 3.75   Class
:character
##  Mode  :character   Mode  :character   Median : 6.50   Mode
:character
##                                        Mean   : 6.50
##                                        3rd Qu.: 9.25
##                                        Max.   :12.00
##    POTENCIAL
##  Min.   :-2.3035
```

```
##  1st Qu.: 0.0000
##  Median : 0.0066
##  Mean   : 0.3636
##  3rd Qu.: 0.1213
##  Max.   :41.4720
```

Como partimos de una base de datos medianamente tratada no tenemos datos nulos, sin embargo tenemos un exceso de datos (174900). Vamos a realizar en **preparación de 1ª base de datos**.

```
head(df, 5)              # Primeras 5 filas
```

```
## # A tibble: 5 × 5
##   CIUDAD   EXTRACCION    SECTOR GAS   POTENCIAL
##   <chr>    <chr>          <dbl> <chr>     <dbl>
## 1 A Coruña A Coruña_City      1 NH3      0.0252
## 2 A Coruña A Coruña_City      1 NMVOC    0.0093
## 3 A Coruña A Coruña_City      1 NOx      0.123
## 4 A Coruña A Coruña_City      1 PPM25    0.163
## 5 A Coruña A Coruña_City      1 SOx      0.0143
```

```
df[60:65, ]              # Filas 60 a 65
```

```
## # A tibble: 6 × 5
##   CIUDAD   EXTRACCION    SECTOR GAS   POTENCIAL
##   <chr>    <chr>          <dbl> <chr>     <dbl>
## 1 A Coruña A Coruña_City     12 SOx      0
## 2 A Coruña A Coruña_Comm      1 NH3      0.0437
## 3 A Coruña A Coruña_Comm      1 NMVOC    0.0159
## 4 A Coruña A Coruña_Comm      1 NOx      0.202
## 5 A Coruña A Coruña_Comm      1 PPM25    0.291
## 6 A Coruña A Coruña_Comm      1 SOx      0.0559
```

```
df[177:182, ]            # Filas 177 a 182
```

```
## # A tibble: 6 × 5
##   CIUDAD   EXTRACCION        SECTOR GAS   POTENCIAL
##   <chr>    <chr>              <dbl> <chr>     <dbl>
## 1 A Coruña A Coruña_National     12 NMVOC   -0.006
## 2 A Coruña A Coruña_National     12 NOx      0.0006
## 3 A Coruña A Coruña_National     12 PPM25    0.138
## 4 A Coruña A Coruña_National     12 SOx      0.0005
## 5 Aachen   Aachen_City            1 NH3      0.0019
## 6 Aachen   Aachen_City            1 NMVOC    0.0029
```

Ahora cargamos la base de datos análoga en la que podremos encontrar información referente a las ciudades. El origen de esta base de datos es una combinación de otras, con las que hemos hecho un merge para obtener variables como `POBLACION CLASIFICACION PARTICULAS SUSPENSION`. A diferencia de la base de datos principal esta si que presenta datos faltantes por ello requerirá de un trabajo previo que describiremos a continuación, con el nombre de **preapración de 2ª base de datos**.

```r
RC <- read_excel("Ranking ciudades.xlsx")
head(RC)
```

```
## # A tibble: 6 × 7
##   PAIS     CLASIFICACION CIUDAD              `PARTICULAS µg/m3`
POBLACION RANGO
##   <chr>    <chr>         <chr>                            <dbl>
<dbl> <dbl>
## 1 Sweden   good          Uppsala                            3.5
219914     1
## 2 Sweden   good          Umeå                               3.6
125080     2
## 3 Portugal good          Faro                               3.6
61015      3
## 4 Iceland  good          Reykjavik                          3.9
132252     4
## 5 Finland  good          Oulu                               4
205489     5
## 6 Finland  good          Tampere / Tammerfors               4
238140     6
## # ℹ 1 more variable: `Coches/1000 habitantes` <dbl>
```

```r
RC[160:165, ]
```

```
## # A tibble: 6 × 7
##   PAIS        CLASIFICACION CIUDAD             `PARTICULAS µg/m3`
POBLACION RANGO
##   <chr>       <chr>         <chr>                           <dbl>
<dbl> <dbl>
## 1 France      fair          Nancy                             9.6
203610   164
## 2 Spain       fair          Telde                             9.7
102791   165
## 3 France      fair          Annemasse                         9.7
61518    166
## 4 Italy       fair          Siracusa                          9.7
119056   167
## 5 Netherlands fair          Greater Eindhoven                 9.7
276979   168
## 6 Austria     fair          Wien                              9.8
1766746   169
## # ℹ 1 more variable: `Coches/1000 habitantes` <dbl>
```

```r
RC[277:282, ]
```

```
## # A tibble: 6 × 7
##   PAIS    CLASIFICACION CIUDAD              `PARTICULAS µg/m3`
POBLACION RANGO
##   <chr>   <chr>         <chr>                           <dbl>
<dbl> <dbl>
## 1 Poland  moderate      Bydgoszcz                        13.7
```

```
350178   281
## 2 Italy   moderate      Forlì                              13.7
118292   282
## 3 Italy   moderate      Napoli (greater city)              13.7
2855958   283
## 4 Spain   moderate      Alicante/Alacant                   13.7
337482   284
## 5 Austria moderate      Graz                               13.8
269997   285
## 6 Cyprus  moderate      Lemesos                            14
189600   286
## # ℹ 1 more variable: `Coches/1000 habitantes` <dbl>

RC[377:382, ]

## # A tibble: 6 × 7
##   PAIS        CLASIFICACION CIUDAD            `PARTICULAS µg/m3`
POBLACION RANGO
##   <chr>       <chr>         <chr>                          <dbl>
<dbl> <dbl>
## 1 Italy       -             Bari                              NA
315284    NA
## 2 Switzerland -             Bern (greater ci…                 NA
227924    NA
## 3 France      -             Boulogne-sur-Mer                  NA
74740    NA
## 4 Germany     -             Braunschweig                      NA
249406    NA
## 5 Spain       -             Cádiz                             NA
115439    NA
## 6 Italy       -             Cagliari                          NA
151005    NA
## # ℹ 1 more variable: `Coches/1000 habitantes` <dbl>

summary(RC)

##      PAIS           CLASIFICACION         CIUDAD           PARTICULAS
µg/m3
##  Length:455         Length:455         Length:455         Min.   :
3.50
##  Class :character   Class :character   Class :character   1st Qu.:
8.50
##  Mode  :character   Mode  :character   Mode  :character   Median
:10.20
##                                                           Mean
:11.26
##                                                           3rd
Qu.:13.70
##                                                           Max.
:26.50
##                                                           NA's   :87
```

```
##    POBLACION           RANGO        Coches/1000 habitantes
##  Min.   :  40278   Min.   :  1.00   Min.   :418.0
##  1st Qu.:  89246   1st Qu.: 96.75   1st Qu.:474.0
##  Median : 146631   Median :188.50   Median :566.0
##  Mean   : 314994   Mean   :188.02   Mean   :573.3
##  3rd Qu.: 288664   3rd Qu.:280.25   3rd Qu.:664.0
##  Max.   :9845879   Max.   :372.00   Max.   :774.0
##                    NA's   :87
```

En la 4ª tabla se observa que tenemos datos fatantes en `CLASIFICACION`, `RANGO` y en `PARTICULAS µg/m3` dado que estas columnas son dependientes entre sí.

2. Descripciones necesarias

Las variables contenidas del primer archivo son:

- **CIUDAD**: Nombre de la ciudad donde se realiza la medición.
- **EXTRACCION**: Identificador de extracción de datos para la ciudad (Ciudad, Comunidad, Nacional e Internacional)
- **SECTOR**: Código numérico que indica el sector de origen de la emisión (cada numero corresponde con un sector "Agricultura, Shipping, Transporte, Industria...")
- **GAS**: Tipo de gas emitido (NH_3, NMVOC, NO_x, PM2.5, SO_x).
- **POTENCIAL**: porcentaje de la concentración media anual de PM2,5 puede atribuirse a un determinado precursor (como el NH_3 o NO_x) emitido por un sector específico.

La variable **EXTRACCION** tiene 4 identificadores distintos (`Ciudad, Comunidad, Nacional e Internacional`). El `Internacional` queda descartado porque como vemos en el primer muestreo, "A Coruña" no tiene ese tipo.

```r
df$EXTRACCION_TIPO <- sapply(str_split(df$EXTRACCION, "_"), function(x)
tail(x, 1))

df <- df[!grepl("International$", df$EXTRACCION_TIPO), ]
```

2.1 Preparación 1ª base de datos

Ahora vamos a determinar que categoría aporta más información diferenciada y cuáles dos categorías son menos relevantes para eliminarlas filas de manera más efectiva. Y realizar una selección de las categorías que más nos interesan.

```r
d <- list()

for (category in unique(df$EXTRACCION_TIPO)) {
  subset <- df$POTENCIAL[df$EXTRACCION_TIPO == category]
  subset <- subset[!is.na(subset)]  # Eliminar NAs si hay
```

```r
  media <- mean(subset)
  mediana <- median(subset)
  desv <- sd(subset)
  coef_var <- if (media != 0) desv / media else 0
  rango <- max(subset) - min(subset)

  d[[category]] <- data.frame(
    Media = media,
    Mediana = mediana,
    Desviacion_Estandar = desv,
    Coef_Variacion = coef_var,
    Rango = rango
  )
}

d_df <- do.call(rbind, d)

d_df
```

```
##              Media Mediana Desviacion_Estandar Coef_Variacion   Rango
## City     0.2154101  0.0020           1.0709457       4.971659 34.5929
## Comm     0.1544029  0.0013           0.7070538       4.579279 19.0268
## National 0.5913231  0.0161           1.9592203       3.313282 43.7755
```

Desviación estándar: Cuanto mayor sea, más dispersos están los datos.
Coeficiente de variación: Un valor alto indica mayor variabilidad relativa.
Rango: Un rango amplio sugiere más diversidad en los valores.

"City" tiene la mayor variabilidad, con la desviación estándar alta y el coeficiente de variación más alto.
"National" tiene la menor variabilidad, con valores casi constantes.
"Comm" tiene una variabilidad intermedia, pero sigue siendo baja comparada con City.

Como uno de los objetivos es el cruce de ciertas variables nos interesa elegir la categoría que tenga mayor variabilidad y permita distinguir diferencias significativas entre los datos.

```r
df <- df[!(df$EXTRACCION_TIPO %in% c("Comm", "National")), ]
df <- df %>% select(-EXTRACCION_TIPO)
df <- df %>% select(-EXTRACCION)
write_xlsx(df, "gases_city.xlsx")
```

2.2 Preparación 2ª base de datos

Dado que en la segunda base de datos tenemos una cantidad moderada de datos faltantes:

```r
sum(is.na(RC))
```

```
## [1] 174
```

Vamos a intentar completar los valores faltantes con el sumatorio del potencial de cada sector por ciudad de las partículas en suspensión (PPM25). Estos datos los añadiremos en una columna aparte para evitar que se solapen los datos que tiene nuestra base de datos original.

```
ranking <- read_excel("Ranking ciudades.xlsx")
gases <- read_excel("gases_city.xlsx")

gases_ppm25 <- gases %>%
  filter(GAS == "PPM25")

ppm25_sumado <- gases_ppm25 %>%
  group_by(CIUDAD) %>%
  summarise(PARTICULAS_ug_m3_calculadas = sum(POTENCIAL, na.rm = TRUE))

ranking_actualizado <- ranking %>%
  left_join(ppm25_sumado, by = c("CIUDAD" = "CIUDAD"))

write_xlsx(ranking_actualizado, "Ranking_actualizado.xlsx")
```

La nueva columna tiene valores distintos en comparación con la columna de datos original PARTICULAS µg/m3 para los valores no faltantes, por ello se descarta la idea de tratar de aunar ambas bases de datos a través de este metodo.

2.3 Web Scraping

Para la obtención del tráfico nos vamos a basar en una recogida de datos realizada por la WikiPedia (https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_Europe) en la cual se recogen los 100 aeropuertos con más pasajeros durante 2023 y 2024. Emplearemos los datos de 2023 para que cuandren con nuestra base de datos original referente a los gases.

```
url <-
"https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_Europe"
pagina <- read_html(url)
tablas <- pagina %>% html_nodes("table") %>% html_table(fill = TRUE)
tabla_aeropuertos <- tablas[[1]]
colnames(tabla_aeropuertos) <- make.names(colnames(tabla_aeropuertos),
unique = TRUE)
tabla_aeropuertos <- tabla_aeropuertos %>%
  mutate(across(.cols = everything(), ~ str_replace_all(., "\\[.*?\\]",
"")))
colnames(tabla_aeropuertos) <- c(
  "Rank_2024", "Rank_Change_2023_24", "Country", "Airport", "Ciudad",
  "Passengers_2024", "Passengers_2023", "Percent_Change_2023_24",
"Number_Change_2023_24"
)
```

```r
tabla_aeropuertos$Passengers_2024 <- as.numeric(gsub("[^0-9]", "",
tabla_aeropuertos$Passengers_2024))
tabla_aeropuertos$Passengers_2023 <- as.numeric(gsub("[^0-9]", "",
tabla_aeropuertos$Passengers_2023))
tabla_aeropuertos$Percent_Change_2023_24 <- as.numeric(gsub("[^0-9.]",
"", tabla_aeropuertos$Percent_Change_2023_24))
tabla_aeropuertos$Number_Change_2023_24 <- as.numeric(gsub("[^0-9-]", "",
tabla_aeropuertos$Number_Change_2023_24))

# Guardar la tabla limpia como xlsx
write_xlsx(tabla_aeropuertos, "Scrape aeropuertos.xlsx")
```

Una vez obtenida la tabla con la información podemos procesarla y emplearla para el estudio.

```r
aero <- read_excel("Scrape_aeropuertos.xlsx")
summary(aero)
```

```
##       Rank            PAIS              CIUDAD          PASAGEROS 2023
##  Min.   :  1.00   Length:100         Length:100         Min.   :
3891553
##  1st Qu.: 25.75   Class :character   Class :character   1st Qu.:
6619217
##  Median : 50.50   Mode  :character   Mode  :character   Median
:10364740
##  Mean   : 50.50                                         Mean
:17358982
##  3rd Qu.: 75.25                                         3rd
Qu.:22236660
##  Max.   :100.00                                         Max.
:79151723
```

Esta nueva base de datos pese a tener solo 100 aeropuertos nos sive para hacer la comparación con 100 otras ciudades que no tengar aeropuerto o cuyo aeropuerto no tenga tanto tráfico aéreo. Para ello vamos a implementar la distancia del aeropuerto al nucleo urbano de la ciudad, con el objetivo de emplear esto como posible factor de contaminación del aire.

```r
aeropuertos <- readxl::read_excel("Scrape aeropuertos.xlsx")

aeropuertos <- aeropuertos %>%
  mutate(full_name = paste(Airport, Ciudad, Country, sep = ", "))

aeropuertos_geo <- aeropuertos %>%
  geocode(address = full_name, method = "osm", lat = latitude, long =
longitude)

## Passing 101 addresses to the Nominatim single address geocoder

## Query completed in: 103.6 seconds
```

```r
writexl::write_xlsx(aeropuertos_geo, "Aeropuertos_con_coordenadas.xlsx")

ciudades <- read_excel("Ranking original.xlsx")

ciudades <- ciudades %>%
  mutate(full_city = paste(City, Country, sep = ", "))

ciudades_geo <- ciudades %>%
  geocode(address = full_city, method = "osm", lat = latitude, long =
longitude)

## Passing 455 addresses to the Nominatim single address geocoder

## Query completed in: 464.4 seconds

print(ciudades_geo)

## # A tibble: 455 × 14
##    Country `Classification Pm25 Conc Txt` City       `City Centroids
Latitude`
##    <chr>   <chr>                          <chr>
<dbl>
##  1 Austria fair                           Salzburg
48
##  2 Austria fair                           Innsbruck
47
##  3 Austria fair                           Wien
48
##  4 Austria moderate                       Linz
48
##  5 Austria moderate                       Klagenfurt
47
##  6 Austria moderate                       Graz
47
##  7 Belgium fair                           Liège
51
##  8 Belgium fair                           Charleroi
50
##  9 Belgium fair                           Mons
50
## 10 Belgium fair                           Namur
50
## # ℹ 445 more rows
## # ℹ 10 more variables: `City Centroids Longitude` <dbl>,
## #   `Fine particulate matter in µg/m3` <dbl>, `Population in the city`
<dbl>,
## #   Rank <dbl>, RankRange <dbl>, `Station Count` <dbl>,
## #   `Coches/1000 habitantes` <dbl>, full_city <chr>, latitude <dbl>,
## #   longitude <dbl>

writexl::write_xlsx(ciudades_geo, "Ciudades_con_coordenadas.xlsx")
```

Ahora procedemos con el segundo web scrape para obtener el tamaño de las ciudades y asi poder cruzarlo con la poblacion con el objetivo de realizar analisis entre ciudades demográficamente similares.

```
# Lista de países en el formato de la URL
countries <- c(
  "albania", "andorra", "austria", "belarus", "belgium", "bosnia",
"bulgaria",
  "croatia", "cyprus", "czechrep", "denmark", "estonia", "finland",
  "france", "germany", "greece", "hungary", "iceland", "ireland",
"italy",
  "latvia", "lithuania", "luxembourg", "malta", "moldova", "monaco",
"montenegro",
  "netherlands", "northmacedonia", "norway", "poland", "portugal",
"romania",
  "russia", "sanmarino", "serbia", "slovakia", "slovenia", "spain",
"sweden",
  "switzerland", "turkey", "ukraine", "vaticancity"
)

resultados <- list()

for (pais in countries) {
  url <- paste0("https://www.citypopulation.de/en/", pais, "/cities/")

  tryCatch({
    pagina <- read_html(url)

    tabla <- pagina %>% html_element("table.data") %>% html_table(fill =
TRUE)

    if (!is.null(tabla)) {
      colnames(tabla) <- str_trim(colnames(tabla))

      name_col <- grep("^(Name|City|Town)", colnames(tabla), value =
TRUE)[1]
      area_col <- grep("Area", colnames(tabla), value = TRUE)[1]

      if (!is.na(name_col) && !is.na(area_col)) {
        df1 <- tabla[, c(name_col, area_col)]
        colnames(df1) <- c("Name", "Area")
        df1$Area <- as.character(df1$Area)  # <- Forzar tipo común
        df1$Country <- pais
        resultados[[pais]] <- df1
      }
    }
  }, error = function(e) {
    message(paste("Error en", pais, ":", e$message))
  })
```

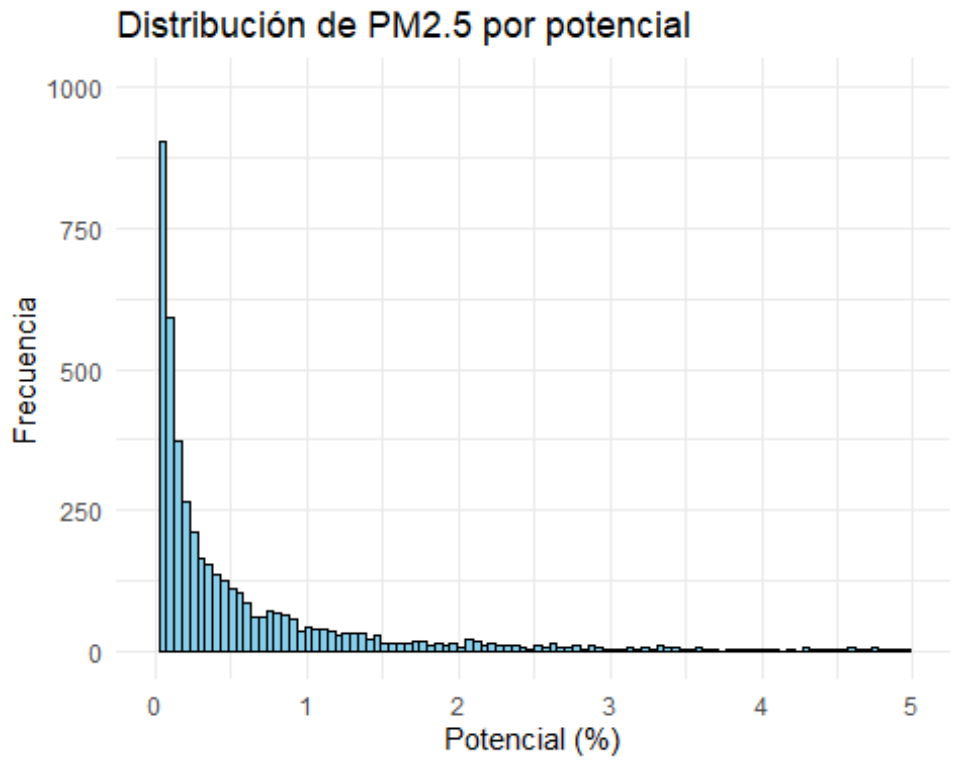
```
}

df_final <- bind_rows(resultados)
write_xlsx(df_final, "ciudades_europa.xlsx")
```

3. Exploración de datos

3.1 Distribución general de PM25:

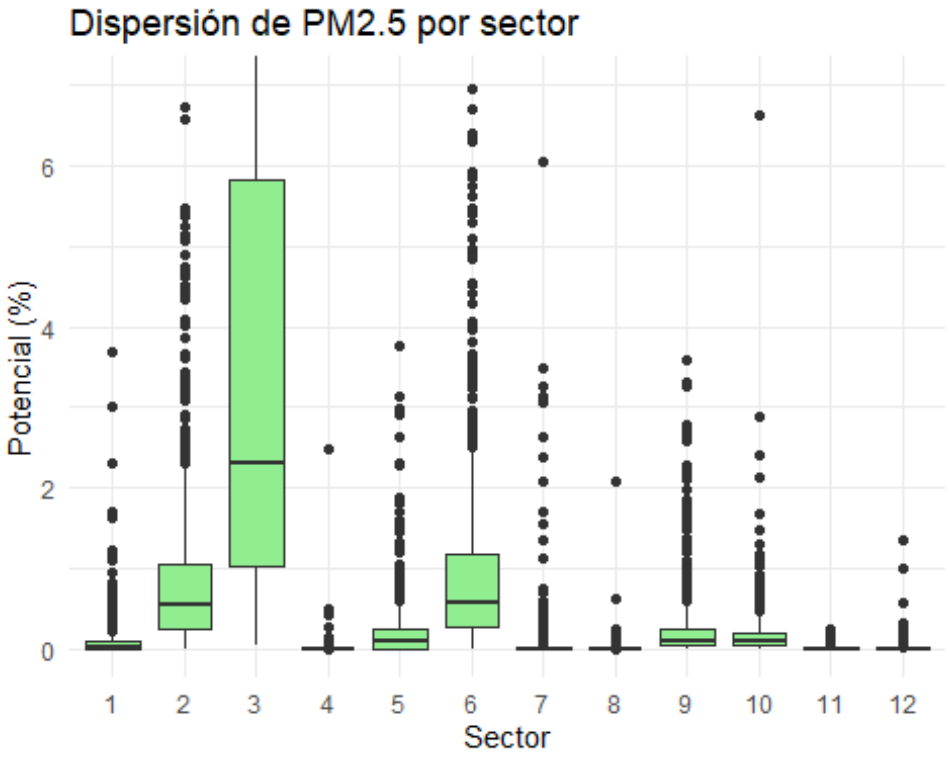
Para entender cómo se distribuyen los niveles de $PM_{2.5}$ entre las ciudades según el sector de emisión, filtramos primero los datos y generamos algunas estadísticas descriptivas.

```
gases_ppm25 <- df %>% filter(GAS == "PPM25")

ggplot(gases_ppm25, aes(x = POTENCIAL)) +
  geom_histogram(bins = 100, fill = "skyblue", color = "black") +
  labs(title = "Distribución de PM2.5 por potencial", x = "Potencial
(%)", y = "Frecuencia") +
  theme_minimal() +
  xlim(0, 5) +  # Recorte para ver el rango más relevante
  coord_cartesian(ylim = c(0, 1000))  # Límite superior en frecuencia
```




```
ggplot(gases_ppm25, aes(x = as.factor(SECTOR), y = POTENCIAL)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Dispersión de PM2.5 por sector", x = "Sector", y =
"Potencial (%)") +
```

```
  theme_minimal() +
  coord_cartesian(ylim = c(0, 7))
```



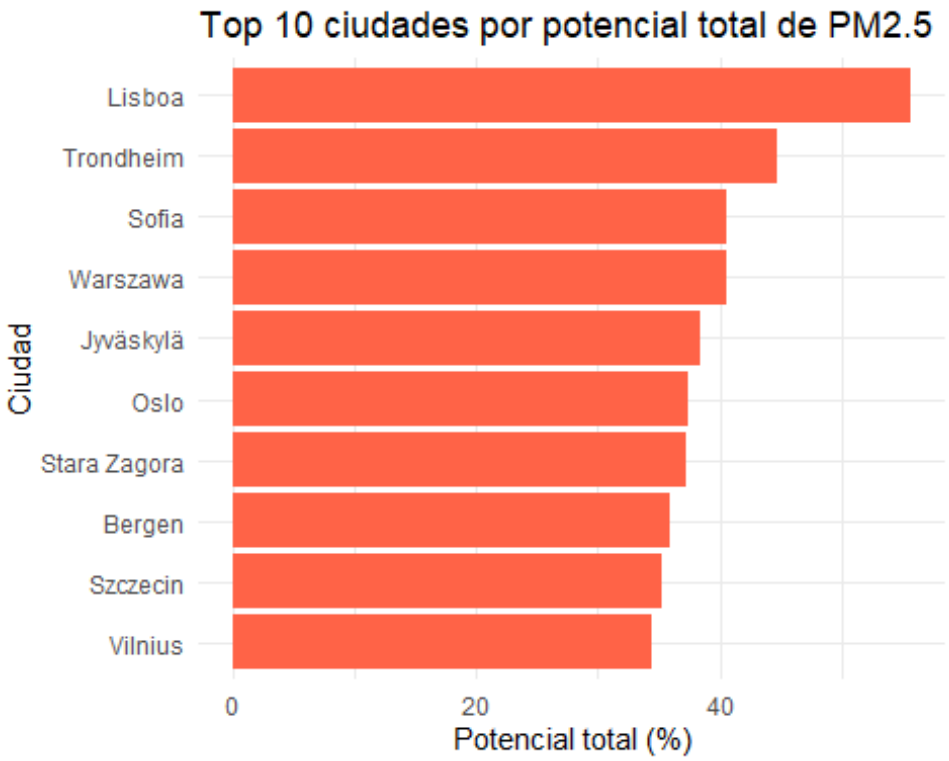
El análisis de boxplots muestra que el sector 3 relacionado con la **combustión estacionaria**, según la base de datos original, es con diferencia el más influyente en términos de potencial de PM2.5.

Este sector no solo presenta una mediana elevada, sino también una gran cantidad de outliers, lo que indica una alta heterogeneidad entre ciudades, cosa que se esperaba debido a la elección de la categoría `CITY` que era la que mayor variabilidad demostró en el análisis.

Sectores como el 2 y el 6 (industria y red de transporte) también tienen cierta relevancia, mientras que otros sectores (4, 5, 7–12) muestran un impacto generalmente bajo o puntual.

```
top_ciudades <- gases_ppm25 %>%
  group_by(CIUDAD) %>%
  summarise(POTENCIAL_TOTAL = sum(POTENCIAL, na.rm = TRUE)) %>%
  arrange(desc(POTENCIAL_TOTAL)) %>%
  slice(1:10)

ggplot(top_ciudades, aes(x = reorder(CIUDAD, POTENCIAL_TOTAL), y =
POTENCIAL_TOTAL)) +
  geom_bar(stat = "identity", fill = "tomato") +
  coord_flip() +
```

```
  labs(title = "Top 10 ciudades por potencial total de PM2.5", x =
"Ciudad", y = "Potencial total (%)") +
  theme_minimal()
```



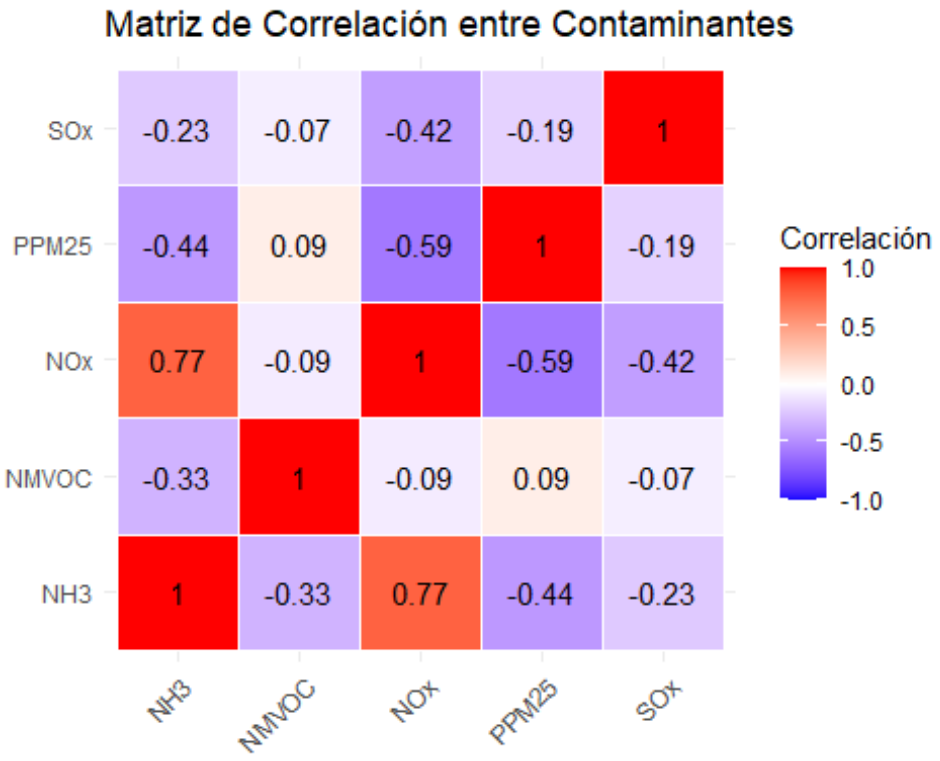
El gráfico de barras revela que ciudades como Lisboa, Trondheim y Sofia concentran los mayores valores acumulados de potencial de PM2.5.

Estas ciudades destacan muy por encima del resto, lo que sugiere una combinación de factores locales como tamaño poblacional, actividad industrial o trafico aereo que agravan la presencia de contaminantes.

3.2 Matriz de Correlación entre Contaminantes:

```
hmap <- read_excel("Ciudades_origen_gases.xlsx")

df_mapa <- hmap %>%
  group_by(CIUDAD, GAS) %>%
  summarise(POTENCIAL = sum(POTENCIAL, na.rm = TRUE), .groups = "drop")
%>%
  pivot_wider(names_from = GAS, values_from = POTENCIAL, values_fill = 0)

matriz_cor <- cor(df_mapa %>% select(-CIUDAD))

matriz_melt <- melt(matriz_cor)

ggplot(matriz_melt, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
```

```r
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name = "Correlación") +
  geom_text(aes(label = round(value, 2)), size = 4) +
  theme_minimal() +
  labs(title = "Matriz de Correlación entre Contaminantes",
       x = NULL, y = NULL) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



- **NH_3 y NO_x** presentan una **correlación positiva alta** (+0.77), lo que indica que comparten fuentes comunes, como **tráfico vehicular y actividades agrícolas**.
- **NO_x y PM_2.5** muestran una **correlación negativa significativa** (-0.59). Ciudades con altas emisiones de NO_x suelen tener menores niveles de PM_2.5 primario, reflejando una compensación entre fuentes de contaminación.
- **NH_3 y PM_2.5** también tienen una correlación negativa moderada (-0.44), sugiriendo que las emisiones de amoniaco en ciudades agrícolas no están relacionadas con altos niveles de material particulado urbano.
- **NO_x y SO_x** presentan una correlación negativa moderada (-0.42), indicando que ciertas ciudades son dominadas por fuentes de NO_x (tráfico) mientras que otras lo son por SO_x (industria).
- **NMVOC** no muestra correlaciones fuertes con otros contaminantes, lo que sugiere que sus fuentes (solventes industriales, tráfico) son más diversas e independientes de las emisiones de NO_x y PM_2.5.

4. Conclusiones sobre la calidad de los datos

4.1. Problemas detectados en la 1ª base de datos (`Ciudades_origen_gases.xlsx`)

- Aunque no hay valores nulos explícitos (`NA`), se detecta un **exceso de filas** (más de 174) que provienen del cruce entre ciudades, gases y sectores. Esto hace que sea necesario **agrupar los datos** por ciudad o por gas para facilitar su interpretación.
- Se observa también que el nombre de los gases no es homogéneo (`PPM25` en lugar de `PM2.5`), lo que requiere una normalización.

4.2. Problemas detectados en la 2ª base de datos (`Ranking ciudades.xlsx`)

- Las columnas `CLASIFICACION`, `RANGO` y `PARTICULAS µg/m3` presentan **valores faltantes**.
- En particular, se ha detectado que cuando falta `CLASIFICACION`, también suele faltar `RANGO` y `PARTICULAS µg/m3`, lo cual indica una **relación de dependencia** entre estas variables.
- Algunas ciudades aparecen duplicadas con nombres levemente distintos, lo que puede generar problemas a la hora de hacer un `merge` con otras fuentes. Será necesario **homogeneizar los nombres de ciudades** en caso de uniones posteriores.

4.3. Decisiones de limpieza adoptadas

1. **Eliminar registros con información totalmente ausente en columnas clave** (como `CLASIFICACION`, `RANGO`, `PARTICULAS µg/m3`).
2. **Estandarizar los nombres de gases** en la primera base de datos.
3. **Agrupar los datos por ciudad y gas** para crear una estructura más resumida y visualizable.
4. Se recomienda incorporar una **validación cruzada** de nombres de ciudad usando codificación NUTS o geolocalización para evitar ambigüedades futuras.