

Documento de cambios

Ángela Alarcón Ballester, Lucía Ponce Salmerón y Carlos Sánchez Polo

2023-11-14

Introducción

En este proyecto analizaremos y trataremos los datos por horas de calidad del aire de las estaciones de la red de vigilancia de la ciudad de Valencia entre los años 2016 y 2020. El conjunto de datos que se utiliza a lo largo de este proyecto se ha obtenido a través del portal de datos abiertos del Ayuntamiento de Valencia (https://valencia.opendatasoft.com/explore/embed/dataset/rvvcca_d_horarios_2016-2020/table/ link).

Comenzaremos realizando una exploración inicial de los datos y responderemos las preguntas que se deriven de ellos.

Importación y acondicionamiento de los datos

Cargamos la librerías que necesitaremos a lo largo del proyecto:

```
library(readr) # Entrada de datos
library(dplyr) # Manipulación de datos
library(tidyr) # Manipulación de datos
library(ggplot2) # Gráficas
library(gridExtra) # Gráficas
library(pheatmap) # Para realizar operaciones
library(purrr) # Mapas de calor
library(knitr) # Tablas
library(GGally)
```

Cargamos y visualizamos los datos:

```
rvvcca_d_horarios_2016_2020 <-
  read_delim("ProyectoAED2023_files/data/rvvcca_d_horarios_2016-2020.csv",
    delim = ";", escape_double = FALSE,
    col_types = cols(Fecha = col_date(format = "%Y-%m-%d"),
      `Fecha creacion` = col_date(format = "%Y-%m-%d"), ), trim_ws = TRUE)
```

Como se observa, las columnas `Fecha` y `Fecha creación` las definimos como fechas en formato “YYYY-MM-DD”.

```
head(rvvcca_d_horarios_2016_2020)
```

Convertimos la columna `Estacion` y `Dia de la semana` en factor:

```
rvvcca_d_horarios_2016_2020$Estacion <- as.factor(rvvcca_d_horarios_2016_2020$Estacion)
rvvcca_d_horarios_2016_2020$`Dia de la semana` <- factor(rvvcca_d_horarios_2016_2020$
`Dia de la semana`, levels = c("Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado",
"Domingo"))
```

Nos damos cuenta de que la columna Hora está en un formato complejo de manejar:

```
unique(rvvcca_d_horarios_2016_2020$Hora)
```

Cambiamos el formato de la columna Hora a uno más manejable:

```
rvvcca_d_horarios_2016_2020$Hora <- format(rvvcca_d_horarios_2016_2020$Hora,
format="%H:%M:%S")
unique(rvvcca_d_horarios_2016_2020$Hora)
```

Ordenamos por fecha:

```
indices_orden <- order(rvvcca_d_horarios_2016_2020$Fecha)
rvvcca_d_horarios_2016_2020 <- rvvcca_d_horarios_2016_2020[indices_orden, ]
```

Eliminamos todas las columnas vacías ya que no nos aportan ninguna información. En este caso, solo se eliminará la columna Fecha baja.

```
rvvcca_d_horarios_2016_2020 <- rvvcca_d_horarios_2016_2020[,
colSums(is.na(rvvcca_d_horarios_2016_2020)) < nrow(rvvcca_d_horarios_2016_2020)]
```

Utilizamos la función `glimpse` para obtener una visión general de los datos:

```
glimpse(rvvcca_d_horarios_2016_2020)
```

Observamos que contamos con las siguientes variables:

- Información general (Id, Fecha, Día de la semana, Día del mes, Hora, Fecha de creación). Son de tipo double, factor, fecha y caracter.
- Contaminantes atmosféricos (PM1, PM2.5, PM10, NO, NO2, NOx, O3, SO2, CO, NH3, C7H8, C6H6, C8H10) y Ruido. Estas variables miden la concentración de partículas en suspensión de diferentes tamaños, la concentración de óxidos de nitrógeno, ozono y azufre, entre otros. Todos son de tipo double.
- Información meteorológica (Velocidad del viento, Dirección del viento, Temperatura, Humedad relativa, Presión, Radiación, Precipitación, Velocidad máxima del viento). Todos son de tipo double.
- Estacion (Avda. Francia, Boulevard Sur, Molino del Sol, Pista Silla, Politécnico, Viveros, Centro, Consellería Meteo, Nazaret Meteo, Puerto València), la cual indica la ubicación de la estación de monitoreo. Es de tipo factor.

Valores faltantes

El estudio de los valores faltantes es esencial para garantizar que los análisis de datos sean sólidos y confiables. Al visualizar los datos inicialmente se han identificado valores faltantes, por lo que en este

Id	0	Velocidad del viento	7874
Fecha	0	Direccion del viento	6020
Dia de la semana	0	NH3	313088
Dia del mes	0	C7H8	313156
Hora	0	C6H6	313635
Estacion	0	Ruido	309508
PM1	286441	C8H10	313053
PM2.5	183359	Temperatura	9033
PM10	183354	Humedad relativa	16357
NO	88740	Presion	6276
NO2	71383	Radiacion	6814
NOx	88738	Precipitacion	7414
O3	87741	Velocidad maxima del viento	8045
SO2	91245	Fecha creacion	0
CO	236277		

Table 1: Número de valores faltantes de cada variable.

apartado procederemos a analizarlos. Primero, realizamos un resumen de la cantidad de valores faltantes en cada columna:

Podemos ver en la Tabla ?? como hay variables con un gran número de valores faltantes, en especial, en los contaminantes atmosféricos y la información meteorológica. Esto es debido a que cada estación dispone de unos sensores distintos de medición de estos fenómenos. Además, el ruido también dispone de muchos valores faltantes, ya que depende de lo tranquila que sea cada estación. Por ello, consideramos oportuno dividir nuestros datos en función de la estación meteorológica asociada, para poder hacer una mejor comparativa, ya que en cada una de ellas se han medido unos valores concretos.

Por tanto, para poder realizar un mejor análisis vamos a agrupar nuestro dataframe por **Estacion**:

```
grupos <- rrvcca_d_horarios_2016_2020 %>% group_by(Estacion)

lista_dataframes <- split(grupos, f = grupos$Estacion)

nombres_dataframes <- unique(rrvcca_d_horarios_2016_2020$Estacion)
```

Limpiamos las columnas vacías de cada dataframe:

```
for (estacion in nombres_dataframes) {
  lista_dataframes[[estacion]] <- lista_dataframes[[estacion]] %>%
    select_if(~!all(is.na(.)))
}
```

Convertimos la lista de dataframes en dataframes independientes:

```
list2env(lista_dataframes, envir = .GlobalEnv)
```

Ahora, veamos que parámetros se han medido en cada estación:

```
parametros <- names(rrvcca_d_horarios_2016_2020)

#Creamos dataframe vacío
tabla_zonas <- data.frame(Zona = character(0), stringsAsFactors = FALSE)
```

```

for (zona in 1:length(lista_dataframes)) {
  fila_zona <- data.frame(Zona = names(lista_dataframes)[zona], stringsAsFactors = FALSE)

  for (parametro in parametros) {
    fila_zona[[parametro]] <- parametro %in% colnames(lista_dataframes[[zona]])
  }

  tabla_zonas <- rbind(tabla_zonas, fila_zona)
}

rownames(tabla_zonas) <- tabla_zonas$Zona
colnames(tabla_zonas)[17] <- "V viento"
colnames(tabla_zonas)[29] <- "V_máx viento"
tabla_zonas <- subset(tabla_zonas, select = -c(Zona,Id,Fecha,`Dia de la semana`,
`Dia del mes`,Hora,Estacion,
`Fecha creacion`))
tabla_numeric <- as.data.frame(sapply(tabla_zonas, as.numeric))

# Crea el heatmap con colores azul y amarillo
pheatmap(tabla_numeric, color=hcl.colors(2,palette = "BluYl"),
  labels_row = rownames(tabla_zonas), cluster_rows = F,
  cluster_cols = F, legend_breaks = c(0,0.25,0.75, 1),
  legend_labels = c("", "No se mide", "Sí se mide", ""),
  border_color = "black")

```

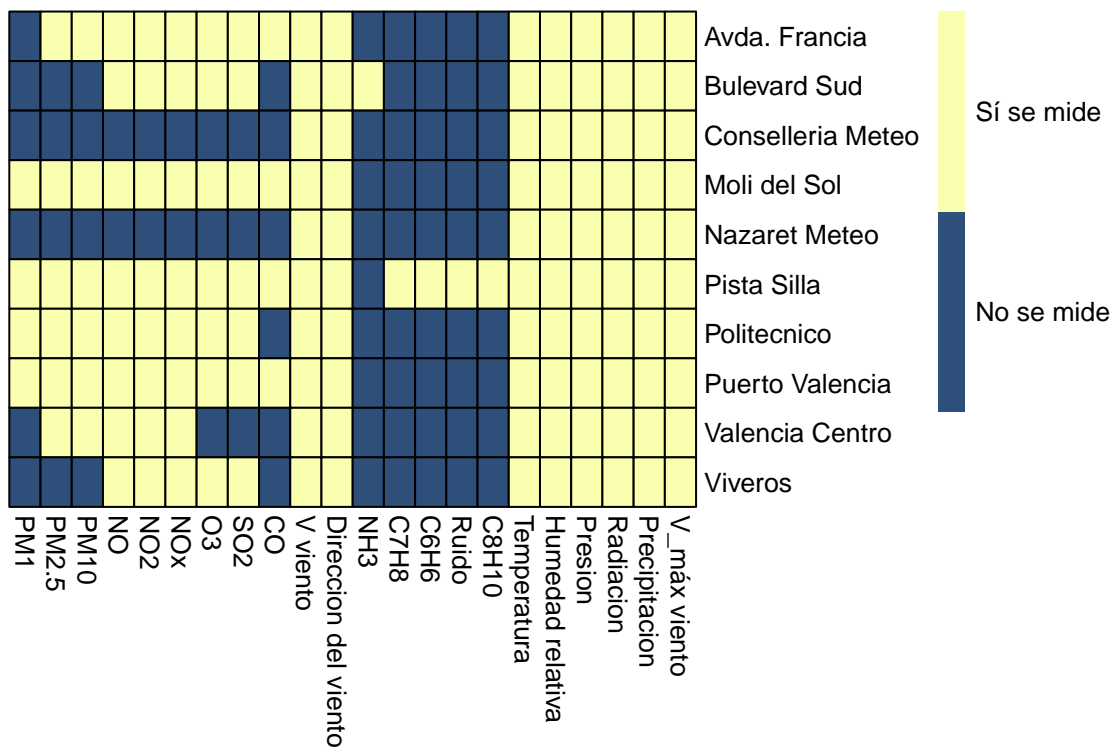


Figure 1: Variables medidas en cada estación.

Como la estación “Pista Silla” es la que menos columnas vacías tiene, es la que vamos a considerar para

nuestro estudio. Veamos cuál es el porcentaje de valores faltantes en cada una de sus columnas, para ver qué variables pueden sernos de utilidad:

PM1	0.636	C8H10	0.097
PM2.5	0.026	Temperatura	0.023
PM10	0.026	Humedad relativa	0.045
NO	0.026	Presion	0.016
NO2	0.026	Radiacion	0.015
NOx	0.026	Precipitacion	0.019
O3	0.065	Velocidad maxima del viento	0.021
SO2	0.070	CO	0.092

Table 2: Porcentaje de valores faltantes de cada variable para la estación "Pista Silla".

En la Tabla ?? observamos que el porcentaje de datos faltantes no es significativo en ninguna de las variables, a excepción de PM1. Por ello, podemos considerar que esta variable no nos aportará información suficiente y podemos eliminarla de nuestro análisis.

Una vez que hemos visualizado y analizado los datos, podemos plantearnos ciertas preguntas. En nuestro caso, hemos considerado que, teniendo en cuenta las variables de las que disponemos, sería interesante realizar un estudio que analice la relación entre la calidad del aire y las horas de mayor cantidad de desplazamientos en automóvil, es decir, las horas de más tráfico en carreteras. ¿Se observará un aumento en la concentración de contaminantes atmosféricos en las horas puntas? ¿Dependerá esto de la estación del año en la que nos encontremos? ¿Y los fines de semana nos encontraremos ante un aumento o una disminución?

Análisis de las variables

Análisis univariante

Como hemos mencionado, vamos a hacer un estudio de los contaminantes atmosféricos, por lo que únicamente consideraremos estos valores de ahora en adelante. Hemos visto anteriormente que todos los contaminantes atmosféricos sí se miden en la estación Pista Silla. Por tanto, vamos a calcular los estadísticos básicos de estos:

```
resultados_totales <- `Pista Silla` %>%
  summarise(across(c("NO", "NO2", "NOx", "O3", "SO2", "CO", "C6H6", "C7H8", "C8H10"),
    list(mean = mean, sd = sd,
    median = median, IQR = IQR), na.rm = TRUE)) %>%
  pivot_longer(cols = -Estacion, names_to = "variable", values_to = "valor") %>%
  separate(variable, into = c("Contaminante", "stat"), sep = "_") %>%
  pivot_wider(names_from = "stat", values_from = "valor") %>%
  mutate(Estacion = "Pista Silla") %>%
  select(Contaminante, mean, sd, median, IQR)

kable(resultados_totales, caption="Estadísticos básicos de la estación Pista Silla.")
```

Table 3: Estadísticos básicos de la estación Pista Silla.

	Contaminante	mean	sd	median	IQR
NO		19.7068219	28.4383802	10.0	20.0
NO2		33.0132081	23.6815240	28.0	33.0

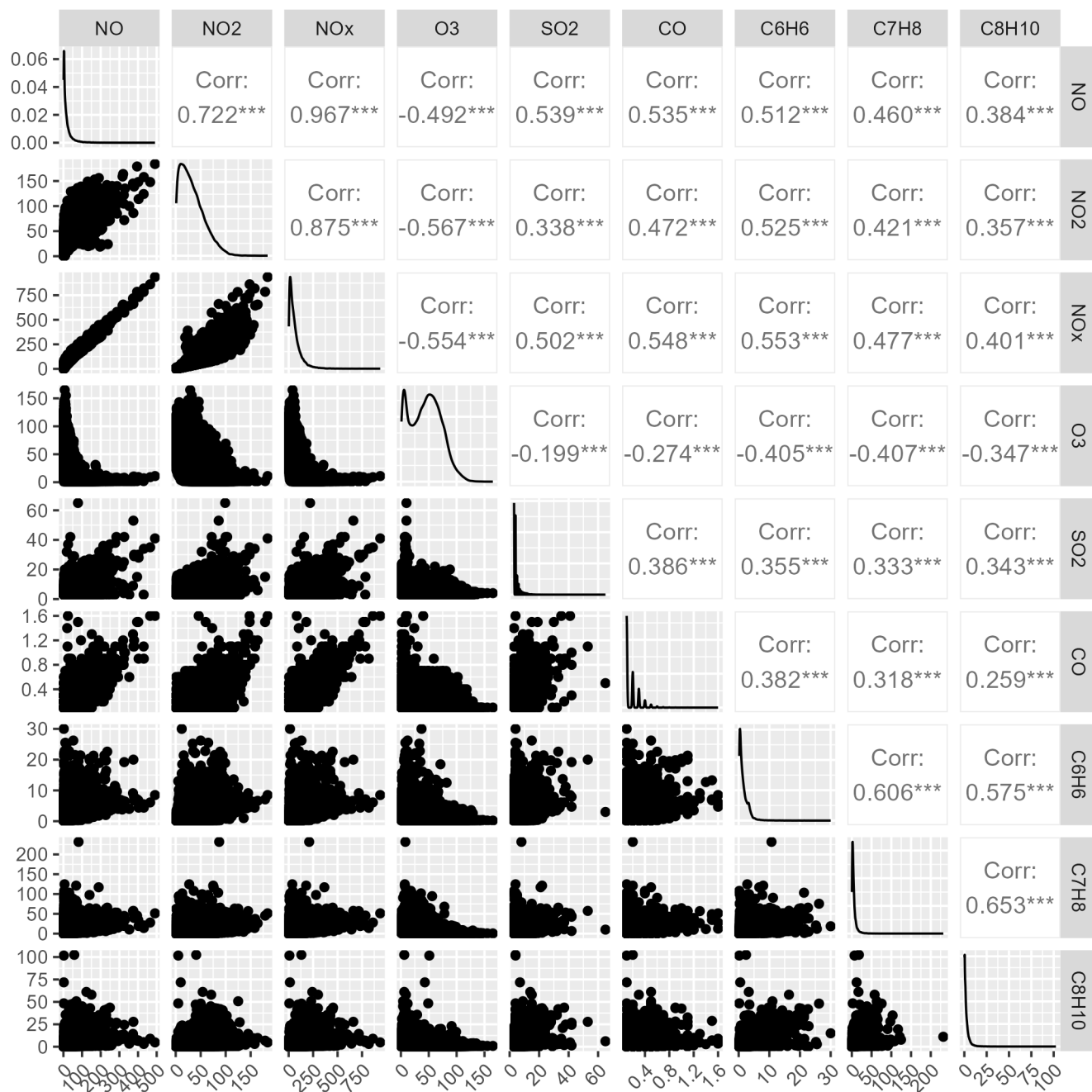
	Contaminante	mean	sd	median	IQR
	NOx	63.0690382	62.9793882	44.0	63.0
	O3	45.0142906	26.7762457	46.0	42.0
	SO2	4.3111563	2.2229034	4.0	1.0
	CO	0.1735941	0.1138526	0.1	0.1
	C6H6	1.5749097	1.6196948	1.1	1.6
	C7H8	5.1178385	6.1959639	3.4	4.4
	C8H10	2.3844622	3.3573891	1.3	2.5

Observamos que el dióxido de nitrógeno (NO₂) tiene la media más alta con 33.01, seguido por el dióxido de azufre (SO₂) con una media de 4.31. El monóxido de carbono (CO) tiene la concentración más baja con una media de 0.17. Además, se destaca que el rango intercuartílico (IQR) de los óxidos de nitrógeno (NO_x) es amplio, indicando una variabilidad significativa en sus niveles.

La desviación estándar (sd) indica la dispersión de los datos con respecto a la media. Por ejemplo, el dióxido de nitrógeno (NO₂) tiene una sd de 23.68, lo que sugiere una variabilidad considerable en las concentraciones de NO₂. En contraste, el monóxido de carbono (CO) tiene una sd mucho más baja de 0.11, indicando una menor variabilidad en sus niveles.

Análisis bivariante

```
knitr::include_graphics("grafs/correlation.png")
```



Se evidencia una relación lineal clara entre las variables NO₂, NO y NO_x, con una correlación superior al 0.7 entre ellas. Esta asociación tiene fundamentos sólidos, dado que los óxidos de nitrógeno (NO_x) son compuestos químicos formados por átomos de nitrógeno y oxígeno. Entre los óxidos de nitrógeno más prevalentes se encuentran el óxido nítrico (NO) y el dióxido de nitrógeno (NO₂), por lo que el término NO_x engloba la mezcla de estos óxidos, indicando que un incremento en NO y NO₂ conlleva un aumento en NO_x.

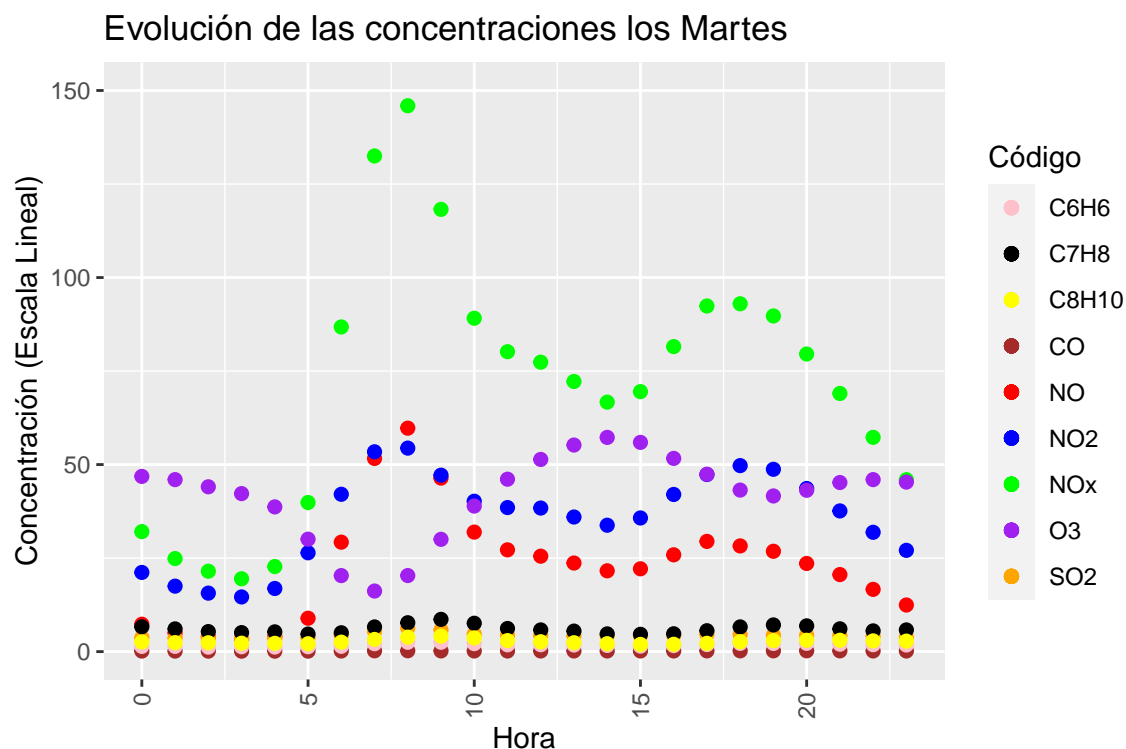
La justificación de esta relación se encuentra en el proceso de formación de estos compuestos, principalmente durante la combustión de combustibles a elevadas temperaturas, como la que ocurre en motores de vehículos, centrales eléctricas y otras instalaciones industriales. Es relevante destacar que el NO₂ es un contaminante atmosférico de considerable importancia.

Por consiguiente, a partir de este punto, se podría considerar enfocar el análisis exclusivamente en el NO_x

para simplificar la investigación, dada su representatividad y la relación intrínseca con los óxidos de nitrógeno clave en la atmósfera.

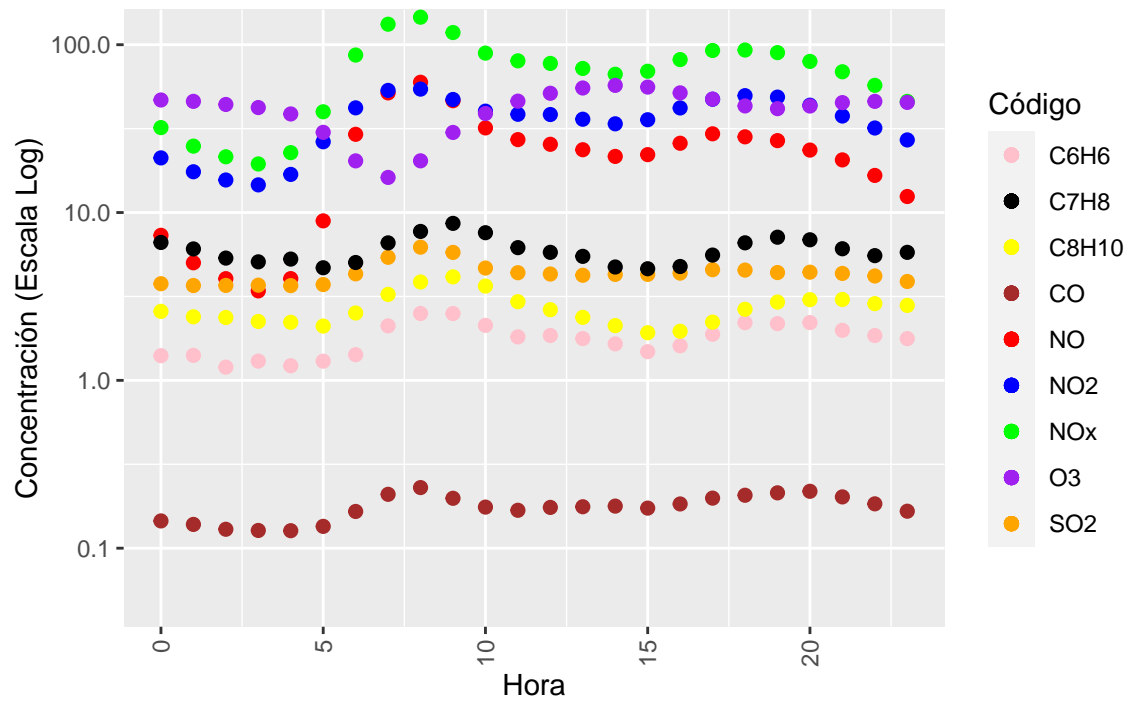
Impacto de las Horas de Desplazamiento Laboral en la Calidad del Aire

Una vez hecho el análisis podemos comenzar a dar respuesta a nuestras preguntas iniciales. Para ello, vamos a realizar la visualización de la concentración de contaminantes por horas del día, realizando a su vez, la comparación entre días de la semana.

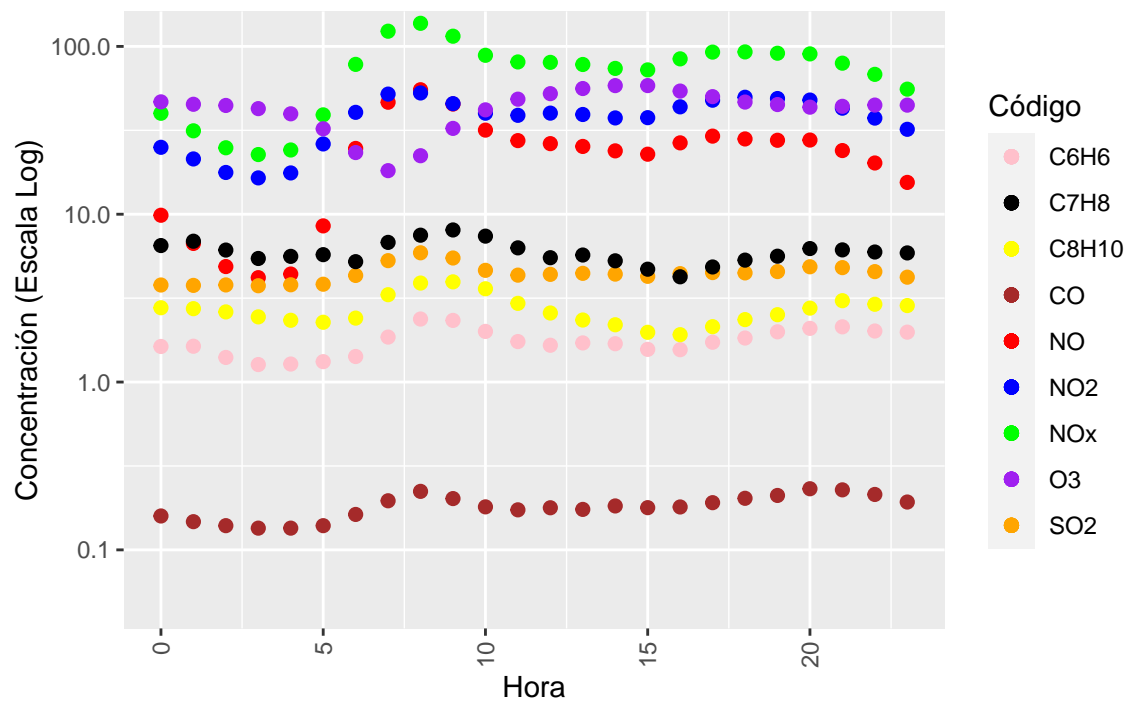


Somos capaces de apreciar los picos de contaminación en horas punta, sin embargo, los valores de las concentraciones de hidrocarburos son bajos, situándose en el extremo inferior de la imagen y solapándose entre ellos. Es por ello que hemos tomado la decisión de representar estos datos en escala logarítmica para varios días. Porque la forma de las gráficas son muy similares en los días laborables, solo mostraremos los martes, pero el resto de gráficas pueden consultarse en la documentación del trabajo.

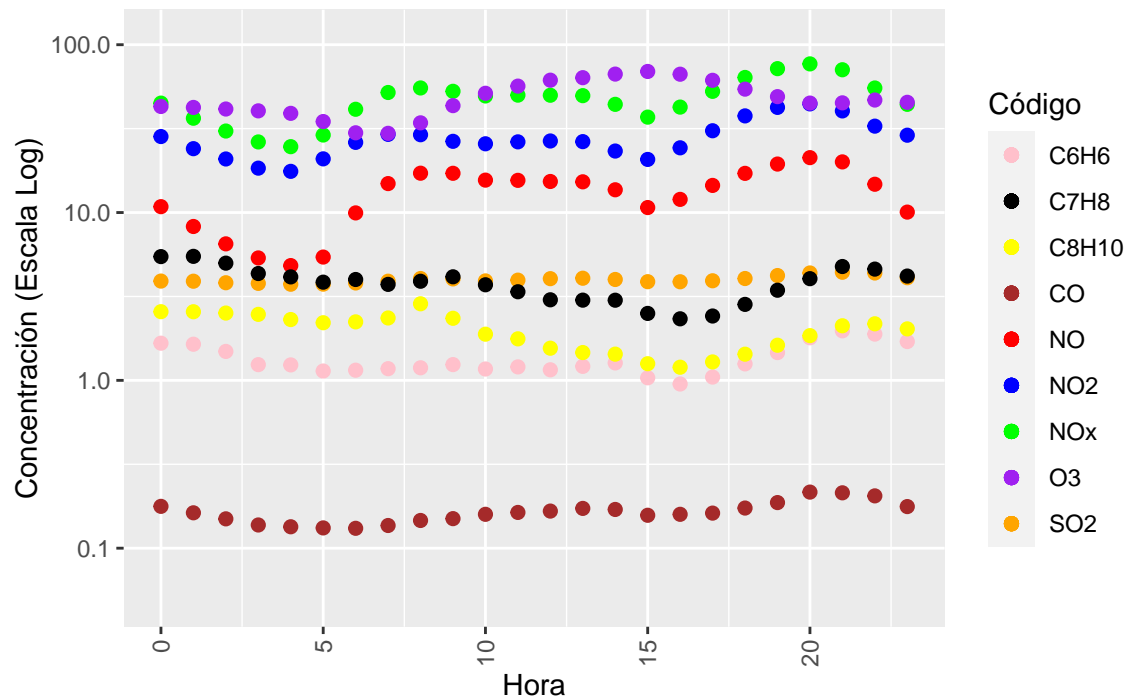
Evolución de las concentraciones los Martes



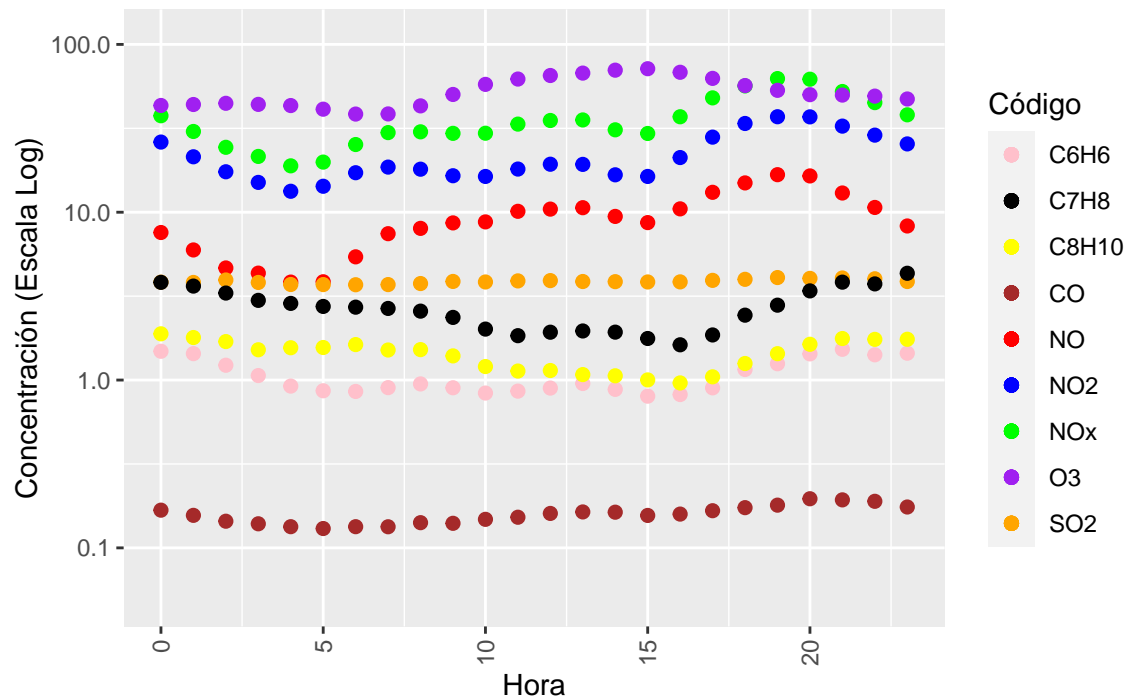
Evolución de las concentraciones los Viernes



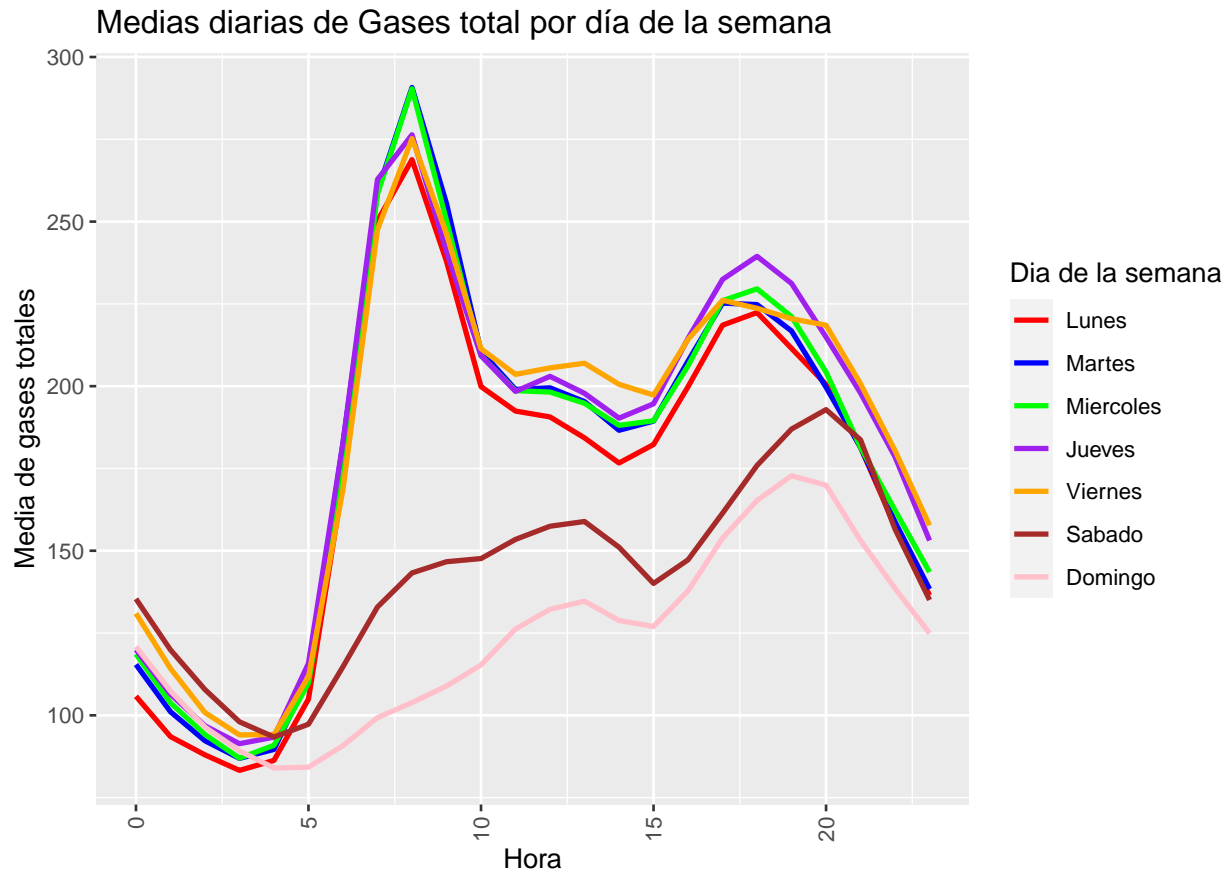
Evolución de las concentraciones los Sábados



Evolución de las concentraciones los Domingos



Ahora que hemos representado las concentraciones en escala logartítmica, somos capaces percibir los picos de concentración incluso en aquellas variables menos presentes. Es de interés estudiar la variabilidad de las concentraciones en función del día de la semana. Porque disponemos de 9 gases para cada día de la semana para cada una de las 24h, hemos decidido representar los valores absolutos de concentración. Véase la siguiente gráfica:



Como comentábamos con anterioridad, los días laborables y los que no son claramente distinguibles. Las concentraciones mantienen levemente la forma, a costa de reducir en gran medida sus valores absolutos. Esto es debido principalmente a la ausencia de necesidad de desplazarse al trabajo, al colegio o a otras zonas de interés.

Podemos comprobar como la conducción de vehículos emite una variedad de gases que tienen un impacto significativo en la calidad del aire y en la salud pública. Entre los principales contaminantes vehiculares nos encontramos los que se registran en nuestro **dataset**. Todos ellos son residuos típicos de los combustibles fósiles. Hemos recopilado información sobre ellos:

- Óxidos de Nitrógeno (NO , NO_2 y NO_x): Producidos durante la combustión a alta temperatura en motores de vehículos. Contribuyen a la formación de ozono troposférico y puede irritar las vías respiratorias.
- Monóxido de Carbono (CO): Resulta de la combustión incompleta de carbono en el motor. Es peligroso porque interfiere con la capacidad de la sangre para transportar oxígeno. Puede tener efectos tóxicos.
- Óxidos de Nitrógeno (NO_x): Contribuye a la formación de smog y afectan la calidad del aire
- Hidrocarburos (C_6H_6 , C_7H_8 y C_8H_{10}): Proviene de la evaporación de combustibles y emisiones de escape. Algunos hidrocarburos son cancerígenos, y otros pueden contribuir a la formación de ozono y smog.

El incremento sustancial de estos contaminantes durante las horas de máximo tráfico se debe a la mayor actividad vehicular y la consiguiente combustión de combustibles fósiles. Está muy claro que los picos de concentración de estos gases no son casuales, ocurren en aquellos momentos de tráfico elevado, las gráficas respaldan esta afirmación. Los picos ocurren:

- En días laborables
- A las horas de entrar a trabajar/estudiar.

Este fenómeno destaca la necesidad de estrategias para mitigar las emisiones vehiculares y mejorar la calidad del aire en entornos urbanos. Un estudio más exhaustivo podría encargarse a expertos químicos.

Análisis de outliers

A continuación, exploraremos la calidad del aire mediante boxplots que representan las concentraciones de dos sustancias fundamentales: dióxido de nitrógeno (NO₂) y ozono (O₃). Cada gráfico proporciona una visión detallada de la distribución de estas sustancias en las distintas estaciones de la red de vigilancia atmosférica de la ciudad de València.

```
generate_boxplot <- function(data, title) {
  boxplot(data, main = title,col = "lightpink", border = "darkblue")
}

par(mfrow = c(2, 4), mar = c(4, 4, 2, 1),cex.main = 0.8)

for (nombre_df in nombres_dataframes) {
  df <- get(nombre_df)

  if ("NO2" %in% names(df) && length(df$NO2) > 0) {
    generate_boxplot(df$NO2, paste("Outliers NO2", nombre_df, sep = " "))
  } else {
    cat("No se encontró la columna 'NO2' en", nombre_df, "\n")
  }
}
```

```
## No se encontró la columna 'NO2' en Conselleria Meteo
```

```
## No se encontró la columna 'NO2' en Nazaret Meteo
```

Observamos una notable presencia de outliers en la variable NO₂ en todas las estaciones, destacando en la parte superior de los boxplots. Estos boxplots son estrechos, lo que sugiere que la mayor parte de los datos se concentra en un rango reducido de concentraciones de dióxido de nitrógeno. Sin embargo, la presencia de numerosos outliers en la parte superior indica que a menudo se generan concentraciones inusualmente altas de esta sustancia en el aire.

```
par(mfrow = c(2, 4), mar = c(4, 4, 2, 1),cex.main = 0.8)

for (nombre_df in nombres_dataframes) {
  df <- get(nombre_df)

  if ("O3" %in% names(df) && length(df$O3) > 0) {
    generate_boxplot(df$O3, paste("Outliers O3", nombre_df, sep = " "))
  } else {
    cat("No se encontró la columna 'O3' en", nombre_df, "\n")
  }
}
```

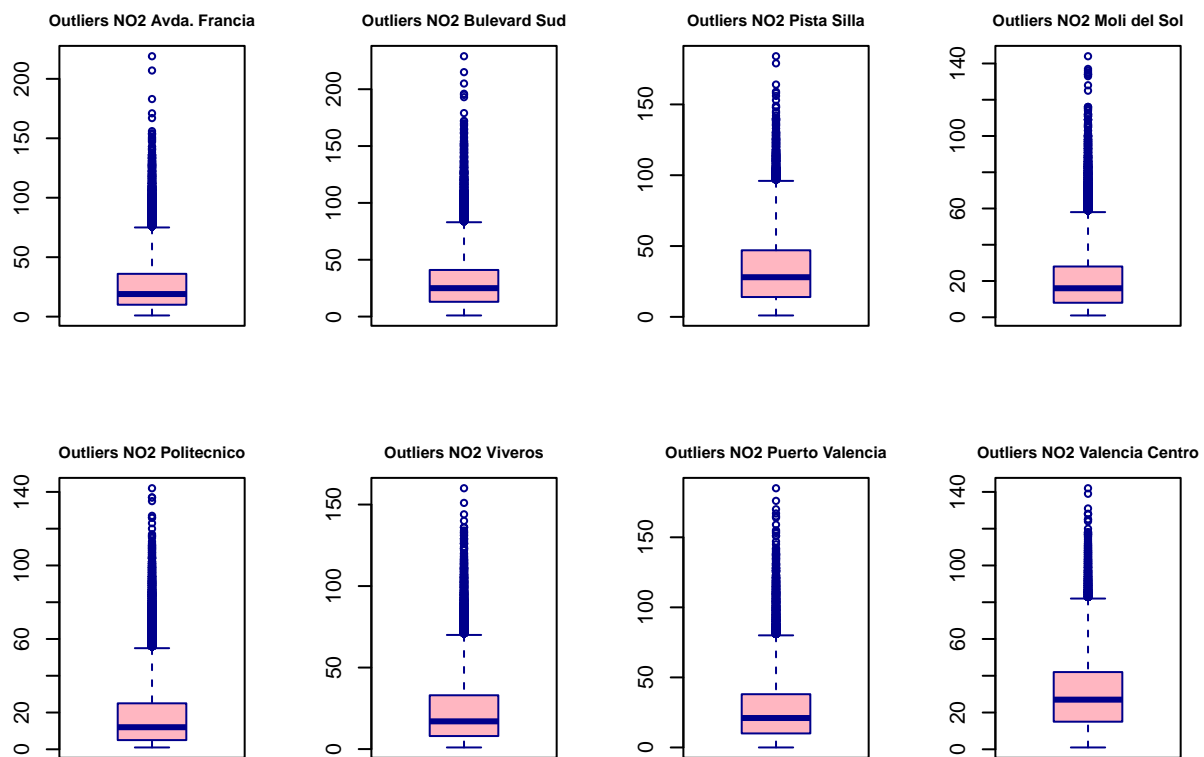
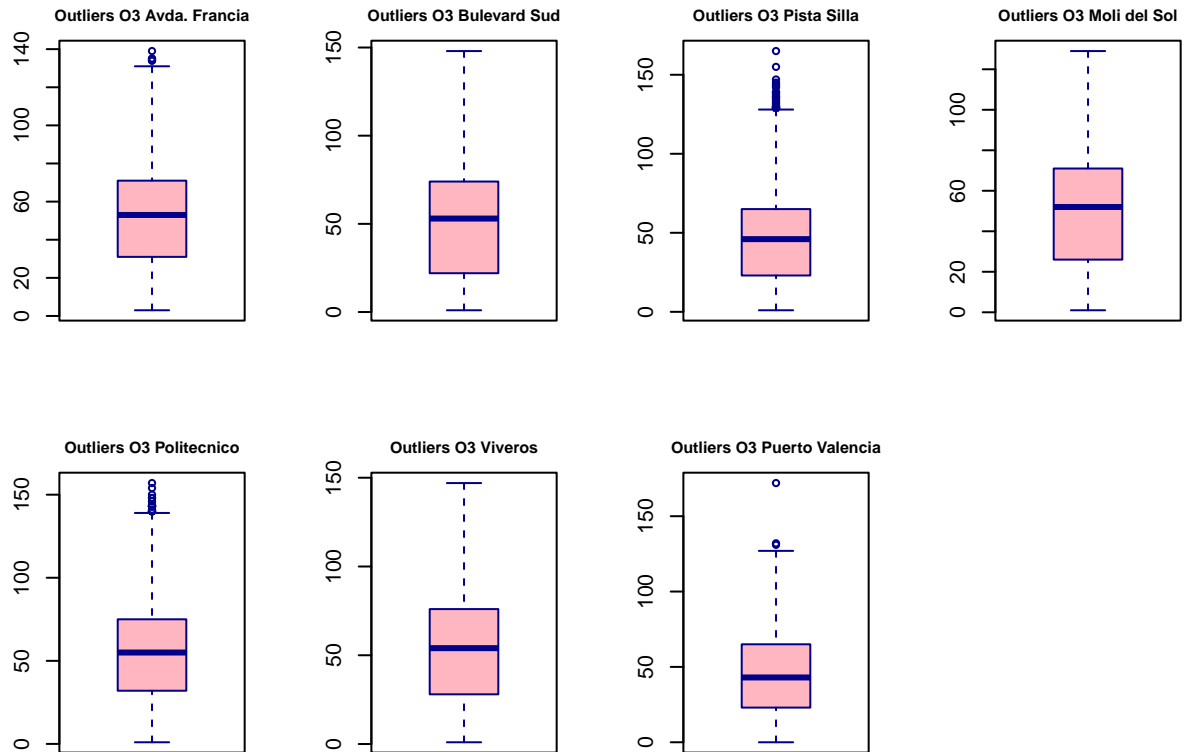


Figure 2: Diagramas de caja de NO2 en cada estación.

```
## No se encontró la columna 'O3' en Conselleria Meteo
```

```
## No se encontró la columna 'O3' en Valencia Centro
```

```
## No se encontró la columna 'O3' en Nazaret Meteo
```



En contraste, la presencia de outliers en la variable de ozono es menos pronunciada. Aunque en las estaciones de Viveros y Consellería Meteo se observa una mayor proporción de outliers, en general, los valores presentan una distribución más uniforme. Los boxplots son más anchos, indicando una dispersión más amplia de las concentraciones de ozono.

También hemos considerado oportuno implementar funciones específicas para la identificación de outliers utilizando distintos métodos estadísticos, como la regla de 3 sigma, el identificador Hampel y la regla de percentiles. Este es otro modo de resaltar valores que podrían considerarse atípicos y podemos comparar con los resultados obtenidos mediante el método boxplot.

Función para la regla de 3 sigma:

```
reglasigma <- function(x) {  
  media <- mean(x, na.rm = TRUE)  
  desviacion <- sd(x, na.rm = TRUE)  
  umbral_superior <- media + 3 * desviacion  
  umbral_inferior <- media - 3 * desviacion  
  outliers <- x[x > umbral_superior | x < umbral_inferior]  
  return(outliers)  
}
```

Función para el identificador Hampel:

```
reglahampel <- function(x) {
  mediana <- median(x, na.rm = TRUE)
  mad <- mad(x, na.rm = TRUE)
  umbral_superior <- mediana + 3 * mad
  umbral_inferior <- mediana - 3 * mad
  outliers <- x[x > umbral_superior | x < umbral_inferior]
  return(outliers)
}
```

Función para la regla del boxplot:

```
reglaboxplot <- function(x) {
  cuartil_75 <- quantile(x, 0.75, na.rm = TRUE)
  cuartil_25 <- quantile(x, 0.25, na.rm = TRUE)
  iqr <- cuartil_75 - cuartil_25
  umbral_superior <- cuartil_75 + 1.5 * iqr
  umbral_inferior <- cuartil_25 - 1.5 * iqr
  outliers <- x[x > umbral_superior | x < umbral_inferior]
  return(outliers)
}
```

Función para la regla de percentiles:

```
reglapercentil <- function(x) {
  percentil_5 <- quantile(x, 0.05, na.rm = TRUE)
  percentil_95 <- quantile(x, 0.95, na.rm = TRUE)
  outliers <- x[x < percentil_5 | x > percentil_95]
  return(outliers)
}
```

```
detectar_outliers <- function(x) {
  sol <- list(
    r_sigma = data.frame(Variable = colnames(x), Metodo = "Regla 3 sigma",
      Outliers = sapply(x, function(col) length(reglasigma(col)))),
    r_hampel = data.frame(Variable = colnames(x), Metodo = "Identificador Hampel",
      Outliers = sapply(x, function(col) length(reglahampel(col)))),
    r_boxplot = data.frame(Variable = colnames(x), Metodo = "Regla Boxplot",
      Outliers = sapply(x, function(col) length(reglaboxplot(col)))),
    r_percentiles = data.frame(Variable = colnames(x), Metodo = "Regla Percentiles",
      Outliers = sapply(x, function(col) length(reglapercentil(col))))
  )

  return(sol)
}
```

```
columnas_a_excluir <- c(1:6, ncol(df))
col <- setdiff(names(df), names(df)[columnas_a_excluir])
outliers_resultados <- detectar_outliers(Viveros[col])

plots <- lapply(outliers_resultados, function(result) {
  ggplot(result, aes(x = Variable, y = Outliers, fill = Variable)) +
    geom_bar(stat = "identity") +
```

```

labs(title = unique(result$Metodo),
     x = "Variable",
     y = "Cantidad de Outliers") +
theme_minimal() + theme(axis.text.x = element_blank(), legend.position = "none")
})

grid.arrange(grobs = plots, ncol = 2)

```

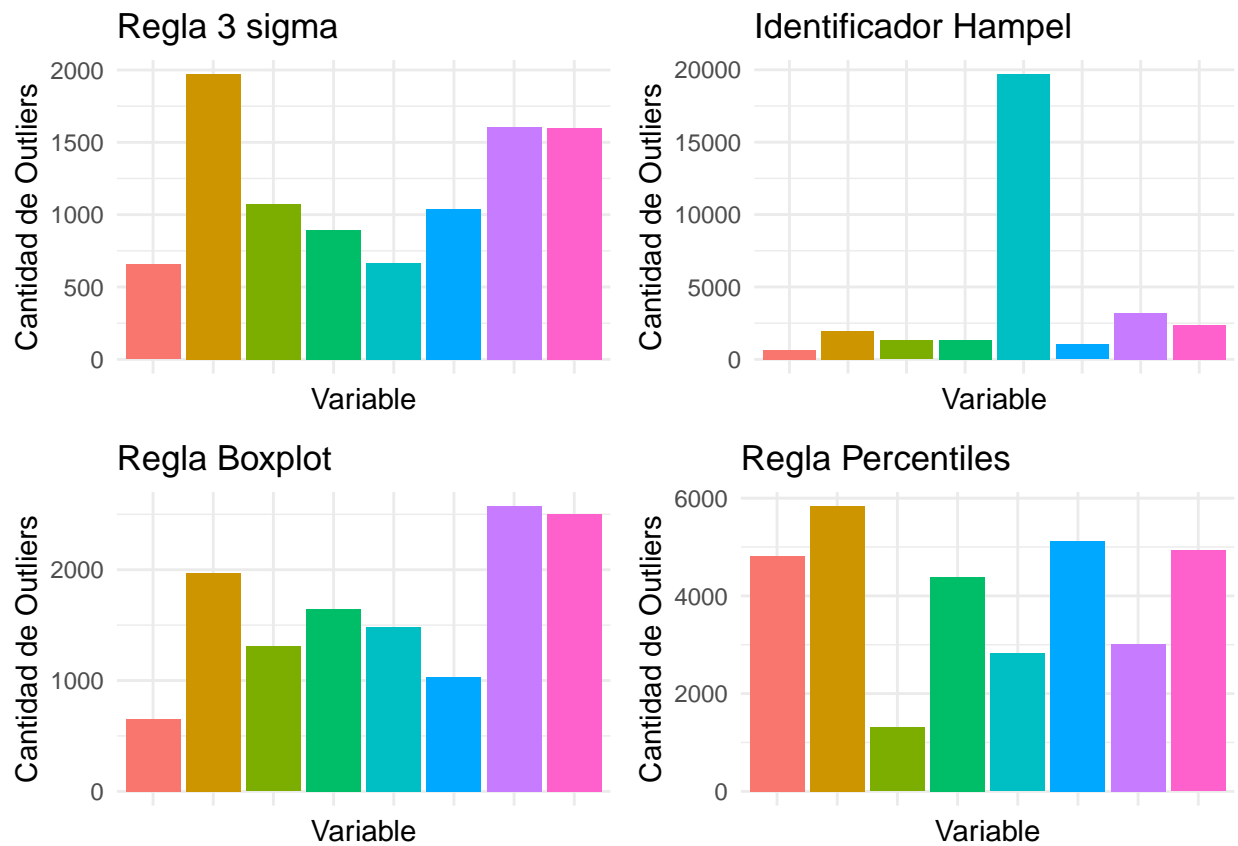


Figure 3: Gráficos de barras para detectar outliers.

La información resultante se presenta de manera clara y comparativa a través de gráficos de barras. Cada barra en estos gráficos representa la cantidad de outliers detectados para una variable específica, empleando diferentes métodos de detección. En este caso, lo hemos aplicado sobre los datos de la estación Viveros.

Conclusiones