MDPI

Article

ProyectoAED2023

Carlos, Diego, Miguel

[1] Universidad de Valencia - ETSE, UV Avinguda de l'Universitat, 46100 Burjassot, Valencia; leutnant@fh-muenster.de
[2] ;
* Correspondence:

Simple Summary: A Simple summary goes here.

Abstract: Análisis exploratorio de un conjunto de datos sobre la calidad del aire de la ciudad de Valencia entre 2004 y 2022

Keywords: Contaminación, Valencia, datos, análisis

1. Introducción

En el presente informe planteamos el análisis exploratorio de un conjunto de datos sobre la calidad del aire de la ciudad de Valencia entre 2004 y 2022. El dataset empleado continene observaciones obtenidas de distintas estaciones de la red de vigilancia de Valencia. Las observaciones están compuestas por variables respectivas a diversas moleculas y elementos presentes en el aire junto a otras de tipo meteorológico como la velocidad del viento, la temperatura, etc.

El procedimiento a seguir comenzará con la correcta importación del dataset, a lo que seguirá la preparación de los datos para resolver las preguntas planteadas. Se escogerán las variables de interés y los periodos temporales sobre los que se realizará el análisis, se aplicarán las transformaciones necesarias sobre las variables, se reestructurará el dataset acorde a nuestras necesidades y se gestionarán los outliers y datos faltantes.

Una vez preparado el dataset, procederemos a responder a las preguntas planteadas mediante diversas metodologías. Todo el proceso irá acompañado de las explicaciones pertinentes y finalizaremos con una conclusión del trabajo realizado.

1.1. Objetivos

El objetivo principal de este trabajo es familizarse con las herramientas y metodologías aprendidas para la carga, manipulación y análisis exploratorio de un conjunto de datos.

- analisis univariante, bivariante, NA, outliers

Para ello se van a plantear una serie de objetivos es forma de preguntas que requerirán la aplicación de las herramientas y metodologías aprendidas:

- ¿?Influencia del cambio en el tráfico hacia el centro de la ciudad en los últimos años. (carriles bici) : Comprobar posible reducción de gases emitidos por vehículos a motor en las estaciones más céntricas.
- ¿?Existe alguna correlación entre la calidad del aire y el día de la semana/año
- ¿?Existe cierta evolución de la contaminación sonora (años/zonas)
- ¿?Existe cierta evolución de la temperatura (A medias, se puede hacer algo más o incluso relacionarlo con las precipitaciones. No se me ocurre cómo pero se le podría dar una vuelta)

Citation: ProyectoAED2023. *Journal Not Specified* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

- ¿?Existe cierta evolución de la contaminación. Como medir la contaminación (Falta esta)

A lo largo del desarrollo del proyecto se irán planteando y resolviendo las preguntas anteriores junto a su razonamiento y conclusiones.

1.2. Carga de librerías y datos

```
pacman::p_load(readr, lubridate, tidyr, dplyr, ggplot2, plotly, stats)
```

El primer paso es importar nuestro dataset. Para ello, en primer lugar abrimos (por ejemplo en un bloc de notas) el archivo *rvvcca.csv* localizado en la carpeta data. De esta forma vemos el formato en el que están guardados los datos, lo cuál es importante para su correcta importación. Por ejemplo, vemos que el separador de las variables es ';', lo cuál debe ser especificado en la función *read_delim()* de readr que usamos para la importación. También definimos el formato con el que queremos importar ciertas columnas, como pueden ser las columnas de tipo fecha o algunos factores.

```
# Especificamos el tipo de columna para que los datos se importen directamente con
column_types <- cols(
  Fecha = col_date(format = "%Y-%m-%d"),
  `Dia de la semana` = col_factor(
    levels = c(
      "Lunes",
      "Martes",
      "Miercoles",
      "Jueves",
      "Viernes",
      "Sabado",
      "Domingo"
    )
  ),
  `Dia del mes` = col_factor(levels = as.character(1:31)),
  Estacion = col_factor(),
  `Fecha creacion` = col_date(format = "%Y-%m-%d"),
  `Fecha baja` = col_date(format = "%Y-%m-%d")
)

# importación del dataset
datos <-
  read_delim(
    "data/rvvcca.csv",
    delim = ";",
    escape_double = FALSE,
    trim_ws = TRUE,
    col_types = column_types
  )
head(datos)
```

```
## # A tibble: 6 x 34
##      Id Fecha      `Dia de la semana` `Dia del mes` Estacion     PM1 PM2.5 PM1
##   <dbl> <date>     <fct>              <fct>         <fct>      <dbl> <dbl> <dbl
## 1     5 2004-03-01 Sabado             3             Pista Sil~    NA    NA NA
## 2     7 2004-04-01 Domingo            4             Pista Sil~    NA    NA NA
## 3     8 2004-04-01 Domingo            4             Viveros       NA    NA 27.
```

```
## 4     9 2004-05-01 Lunes              5             Pista Sil~    NA    NA NA
## 5    14 2004-07-01 Miercoles          7             Viveros       NA    NA 43.
## 6    26 2004-01-13 Martes             13            Viveros       NA    NA 33.
## # i 26 more variables: NO <dbl>, NO2 <dbl>, NOx <dbl>, O3 <dbl>, SO2 <dbl>,
## #   CO <dbl>, NH3 <dbl>, C7H8 <dbl>, C6H6 <dbl>, Ruido <dbl>, C8H10 <dbl>,
## #   'Velocidad del viento' <dbl>, 'Direccion del viento' <dbl>,
## #   Temperatura <dbl>, 'Humidad relativa' <dbl>, Presion <dbl>,
## #   'Radiacion solar' <dbl>, Precipitacion <dbl>,
## #   'Velocidad maxima del viento' <dbl>, 'As (ng/m³)' <dbl>,
## #   'Ni (ng/m³)' <dbl>, 'Cd (ng/m³)' <dbl>, 'Pb (ng/m³)' <dbl>, ...
```

Para evitar posibles problemas, renombramos las varianles con espacios en su nombre para trabajar sin comillas ' ' y eliminamos el sufijo (ng/m³) que aparece en algunas variables

```
colnames(datos) <-
  gsub(x = colnames(datos),
       pattern = ' ',
       replacement = '_')
colnames(datos) <-
  gsub(x = colnames(datos),
       pattern = '_[(]ng[/]m³[)]$',
       replacement = '')
```

Para observar las características generales de nuestro dataset vamos a resumir en una tabla el tipo de cada variable *tyoe*, cuantos valores distintos tiene *levels*, su valor más frecuente *topLevel*, cuántas veces aparece *topCount*, y en qué proporción *topFreq* y por último la proporción de datos faltantes de cada variable *missFreq*:

```
levels <- function(x) {
  as.character(length(unique(x)))
}

topLevel <- function(x) {
  names(which.max(table(x)))
}

topCount <- function(x) {
  as.character(sum(as.character(x) == names(which.max(table(
    x
  ))), na.rm = T))
}

topFrec <- function(x) {
  as.character(round(sum(
    as.character(x) == names(which.max(table(x))), na.rm = T
  ) / length(x), digits = 3))
}

missFrec <- function(x) {
  as.character(round(sum(is.na(x)) / length(x), digits = 3))
}

sumario <-
  list(
    type = ~ class(.x),
```

```
    levels = ~ levels(.x),
    topLevel = ~ topLevel(.x),
    topCount = ~ topCount(.x),
    topFrec = ~ topFrec(.x),
    missFrec = ~ missFrec(.x)
  )

datos %>%
  summarise(across(
    .cols = all_of(colnames(datos)),
    .fns = sumario,
    .names = "{.col}-{.fn}"
  )) %>%
  pivot_longer(cols = 1:203,
               names_to = 'variable_estadistica',
               values_to = 'valor') %>%
  separate(variable_estadistica,
           into =  c('variable', 'estadistica'),
           sep = '-') %>%
  pivot_wider(names_from = 'estadistica', values_from = 'valor')
```

```
## # A tibble: 34 x 7
##    variable         type    levels topLevel    topCount topFrec missFrec
##    <chr>            <chr>   <chr>  <chr>       <chr>    <chr>   <chr>
##  1 Id               numeric 43388  1           1        0       0
##  2 Fecha            Date    6940   2022-01-01  12       0       0
##  3 Dia_de_la_semana factor  7      Sabado      6205     0.143   0
##  4 Dia_del_mes      factor  31     16          1426     0.033   0
##  5 Estacion         factor  13     Pista Silla 6940     0.16    0
##  6 PM1              numeric 179    4           978      0.023   0.748
##  7 PM2.5            numeric 221    9           1567     0.036   0.463
##  8 PM10             numeric 479    13          1104     0.025   0.336
##  9 NO               numeric 178    2           3890     0.09    0.23
## 10 NO2              numeric 120    16          961      0.022   0.23
## # i 24 more rows
```

- *Id* es un identificador para cada una de las filas y no aporta ninguna información concreta.

- *Fecha* es una variable tipo *Date* que nos proporciona la fecha en la que se tomaron los datos que componen la observación. Tomaremos esta variable para la ordenación ascendente de los datos. Vemos que hay datos de casi 7000 días distintos y que el día con más datos (esto se corresponde a más estaciones distintas midiendo 12/13) es el 01-01-2022.

```
datos <- arrange(datos, datos$Fecha)
```

- *Dia_de_la_semana* y *Dia_del_mes* son variables de tipo factor que indican en que día de la semana y en qué día del mes se realiza la medida. Puede ser extraído a partir de *Fecha* usando las funciones *wday()* y *day()*, de la librería *lubridate*. Vemos que el día de la semana más repetido es el Sábado y el del mes es el 31, sin embargo sus frecuecnias son prácticamente 1/7 y 1/30 por lo que parece indicar que todos los días de la semana y del mes están en la misma proporción. Esto estaría enconcordancia con el hecho de que tenemos datos realmente diarios.

- *Estacion* es una variable de tipo factor cuyos niveles son las distintas estaciones meteorológicas donde se tomaron las mediciones de las variables numéricas, haciendo un total de 13 estaciones. La más repetida es la estación de Pista de Silla con una frecuencia de 0.16 > 1/13 = 0.077. Esto significa que es la estación que aparece en más días distintos, y es indicativo de que no todas las estaciones funcionan durante el mismo tiempo.

```
unique(datos$Estacion)
```

```
##  [1] Pista Silla               Viveros                   
##  [3] Politecnico               Avda. Francia             
##  [5] Moli del Sol              Bulevard Sud              
##  [7] Conselleria Meteo         Puerto Valencia           
##  [9] Valencia Centro           Nazaret Meteo             
## [11] Puerto llit antic Turia   Puerto Moll Trans. Ponent 
## [13] Valencia Olivereta                                  
## 13 Levels: Pista Silla Viveros Politecnico Avda. Francia ... Puerto Valencia
```

- *PM1, PM2.5, PM10* son variables de tipo numérico con datos sobre la concentración de materiales particuldos (PM) de menos de 1, 2.5 y 10 micrómetros de diámetro respectivamente. Pese a ser variables numéricas supuestamente continuas llama a la atención que no hay un gran número de niveles distintos comparado con el número de filas del del dataset (alrededor de 10 veces menos). También vemos que hay un gran número de datos faltantes, sobre todo para PM1 (más de un 70%) y esto puede explicar en parte que no haya tantos valores distintos. Sin embargo, también podría deberse a que quizá durante muchos días el nivel de concentración de esas partículas se mantiene constante (o al menos la sensibilidad del sensor que las mide no es capaz de medir tal diferencia). La gran cantidad de datos faltantes puede deberse a que las varaibles no se miden durante ciertos períodos de tiempo o en todas las estaciones.

- *NO, NO2, NOx, O2, SO2, CO, NH3* son variables con datos sobre la concentración de estas moléculas inorgánicas consideradas contaminantes en el aire. Respecto a los óxidos de nitrógeno (NOx), estos son un grupo de gases compuestos por oxígeno y nitrógeno, es decir, NO y NO2 forman parte de este grupo y por lo tanto los valores de las variables estarán altamente correlacionados. La proporción de NA's en estas variables menor en este caso, alrededor del 20% con excepción del NH3 que presenta más de un 90% de datos faltantes. De nuevo el número de NA's puede deberse a los mismos motivos argumentados anteriormente y que comprobaremos más adelante en el análisis. Por otro lado, también tenemos unos pocos cientos de valores distintos para ser también variables numéricas continuas y esto puede deberse a las mismas causas argumentadas para los materiales particulados.

- *C7H8, C6H6, C8H10* son variables con datos sobre la concentración de estas moléculas orgánicas también consideradas contaminantes en el aire. De nuevo estas variables numéricas y continuas presentan datos faltantes por encima del 90% y muchos menos niveles distintos que observacioness de las variables. Podrían ofrecerse los mismos argumentos para explicar estos hechos. El porcentaje similar de datos faltantes podría se debido a que estas variables se miden al vez (en las mismas estaciones y períodos de tiempo).

- *Velocidad_del_viento, Direccion_del_viento, Temperatura, Humidad_relativa, Presion, Radiacion_solar, Precipitacion, Velocidad_maxima_del_viento* son variables numéricas con mediciones de estas distintas condiciones ambientales al momento de medir las concentraciones de moléculas contaminantes en el aire. Es interesante contar con estos datos para ver si influyen en los valores de contaminación. Estas variables presentan entre un 50-60% de datos faltantes que puede deberse a que no todas las estacione miden las condiciones ambientales.

- *As, Ni, Cd, Pb* son variables numéricas con datos de otras concentraciones de gases y metales contaminantes en el aire. Todos estas variables presentaban el prefijo (ng/m3) indicando las unidades en que se mide sus concentraciones. El resto de datos de concentraciones no presentan las unidades lo cuál nos hace sospechar que auiás los datos de estas variables fueron medidas en conjunto. Esto también explicaría que todos tengan el mismo porcentaje de datos faltantes (entorno al 98%). De nuevo, en el análisis comprobaremos a qué puede deberse esta cantidad de datos faltantes.
- *B(a)p* es una variable booleana que sólo presenta una observación. Elresto de datos son faltants y no podemos saber lo que representa esta variable.
- *Fecha_creación* y *Fecha_baja* son variables de tipo *Date* que parecen estar relacionadas con la creacion del dataset y que no parecen aportar información sobre nuestros datos. *Fecha_creación* solo presenta dos entradas distintas y *Fecha_baja* sólo presenta NA's por lo que estas variables parecen prescindibles.

Vamos a mantener sólo las variables que aporten alguna información relevante:

```
datos <-
  datos %>% select(
    -c(
      'Id',
      'Dia_de_la_semana',
      'Dia_del_mes',
      'Fecha_creacion',
      'Fecha_baja',
      'B(a)p'
    )
  )
```

2. Análisis de datos faltantes y especiales

En esta sección, previamente al análisis univariante y bivariante de las variables, vamos a realizar un análisis de la estructura del dataset. El dataset parece estar compuesto por la unión de un conjunto datasets, posiblemente por las distintas estaciones de las que provienen los datos. Con el objetivo de obtener un conjunto de datos consistente, analizaremos el porcentaje de NA's que hay por cada variable, luego veremos que datos proporcionan las estaciones y con que frecuencia, también comprobaremos cuando comenzaron a generar datos las estaciones y durante que periodos de tiempo se han obtenido las variables.

En primer lugar vamos reemplazar todos los posibles valores especiales (NULL, NaN, Inf...) que pudieran haber en las varianles numéricas de nuestro dataset por NA's. De esta forma podemos tratar todos estos datos que no tienen sentido a la vez.

```
variables_numericas <- colnames(datos[sapply(datos, is.numeric)])
reemplazo_especiales <- function(x) {
  return(ifelse(is.finite(x), x, NA))
}
datos <-
  mutate(datos, across(all_of(variables_numericas), reemplazo_especiales))
```

Ya hemos visto que nuestras variables numéricas tienen un alto porcentaje de NA's:

```
porc_na <- function(x) {
  100 * sum(is.na(x)) / length(x)
}
```

```
datos %>%
  summarise(across(.cols = all_of(colnames(datos)), .fns = porc_na)) %>%
  pivot_longer(cols = all_of(colnames(datos)),
               names_to = 'Variable',
               values_to = "Porcentaje NA's") %>%
  arrange(desc(`Porcentaje NA's`)) %>%
  mutate(`Porcentaje NA's` = round(`Porcentaje NA's`, digits = 2))
```

```
## # A tibble: 28 x 2
##    Variable `Porcentaje NA's`
##    <chr>                <dbl>
##  1 Ni                    97.8
##  2 Pb                    97.8
##  3 As                    97.8
##  4 Cd                    97.8
##  5 NH3                   94.0
##  6 C7H8                  90.8
##  7 C6H6                  90.2
##  8 C8H10                 90.1
##  9 Ruido                 75.1
## 10 PM1                   74.8
## # i 18 more rows
```

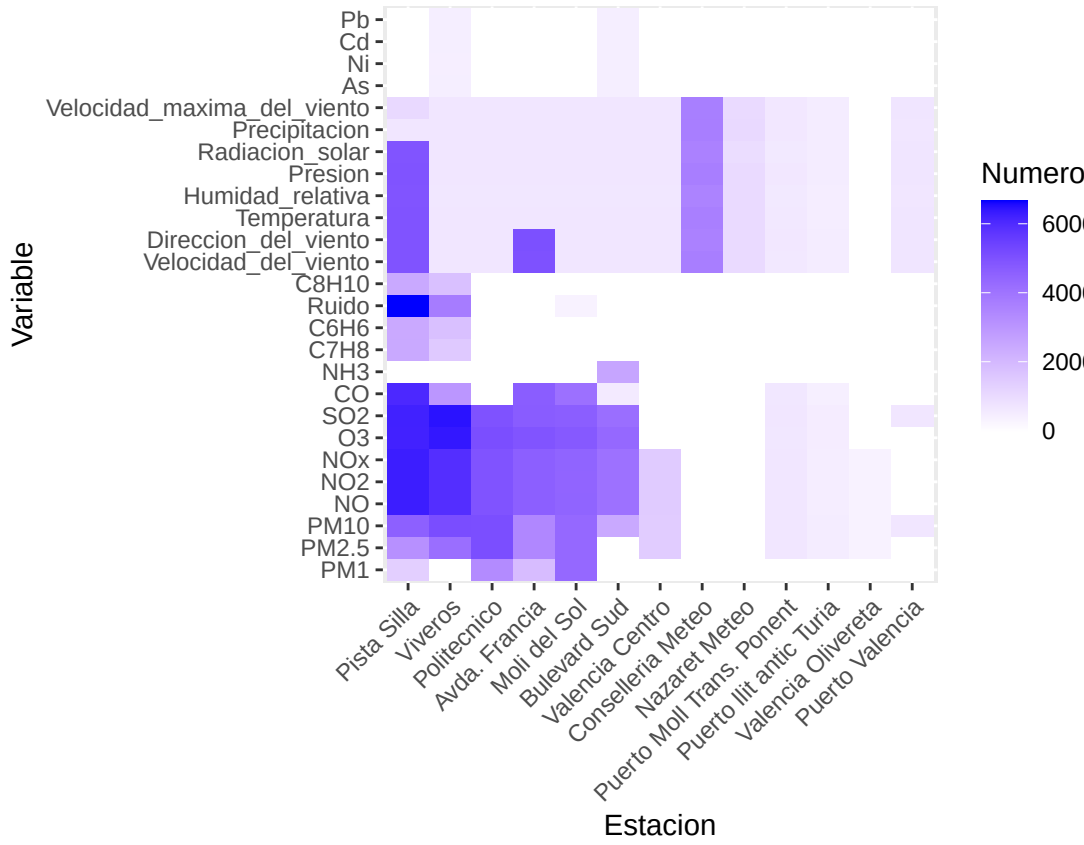
Como ya hemos comentado, pensamos que el elevado número de datos faltantes puede deberse a que algunas variables se miden en peródos de tiempo diferentes a las demás o a que no en todas las estaciones se miden todas las variables. Al unir los datos de las distintas estaciones por la fecha, se habrían dado lugar los NA's. Para comprobar esta teoría vamos a realizar una serie de representacions gráficas exploratorias.

En primer lugar, creamos un mapa de calor para comprobar cuáles son las variables que mide cada estación y cuántos datos de dicha variable aporta cada estación.

```
num_entradas <- function(x) {
  sum(!is.na(x))
}
est_var <- datos %>%
  group_by(Estacion) %>%
  summarise(across(.cols = all_of(variables_numericas), .fns = num_entradas)) %>%
  pivot_longer(
    cols = all_of(variables_numericas),
    names_to = 'Variable',
    values_to = "Numero_entradas"
  ) %>%
  mutate(Variable = factor(Variable, levels = variables_numericas, ordered = T)) %
  arrange(match(Variable, variables_numericas))

p <-
  ggplot(data = est_var, aes(x = Estacion, y = Variable, fill = Numero_entradas))
  geom_tile() + scale_fill_gradient(low = "white", high = "blue") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

show(p)
```

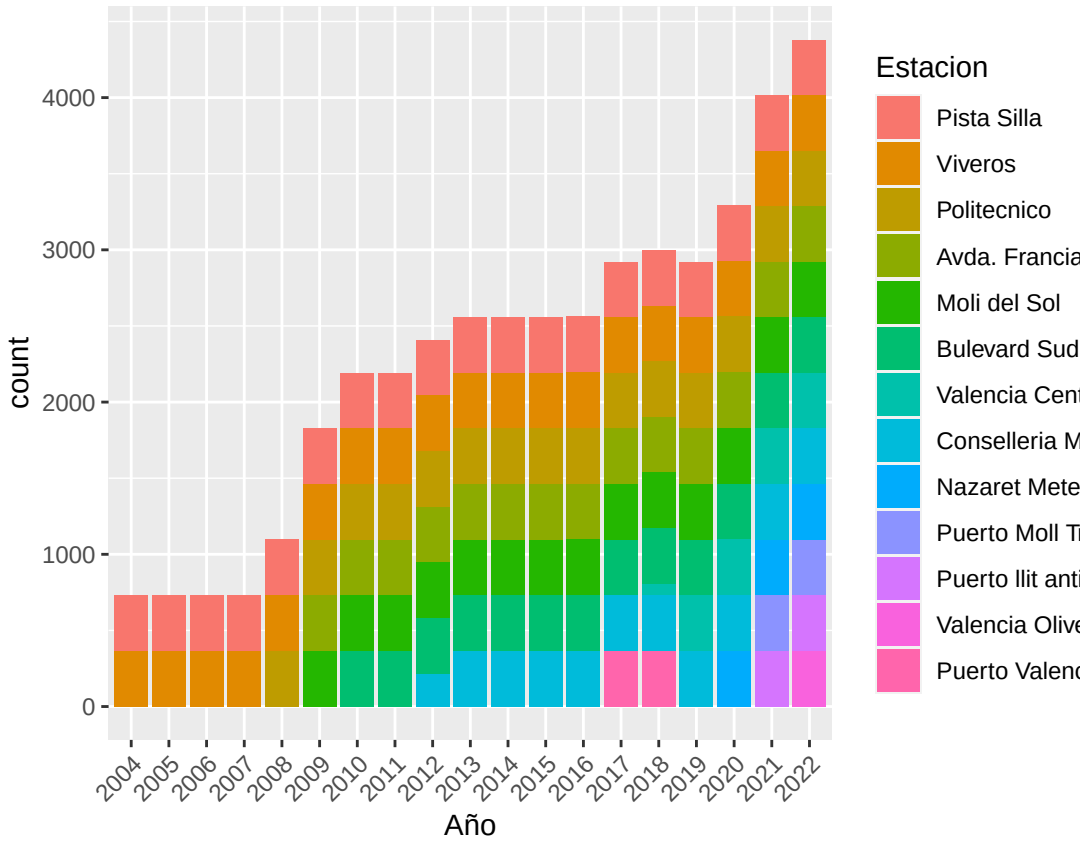



```
#plotly::ggplotly(p)
```

En este mapa de calor podemos ver qué variables mide cada estación meteorológica, además de la cantidad de datos que aporta cada una de ellas sobre cada varaible numérica. Vemos como sólo hay dos estaciones (Viveros y Boulevard Sud) que miden las variables *As*, *Pb*, *Ni* y *Cd* y que su número de entradas es bajo en comparación con otras variable, quizá debido a que se empezaron a tomar datos de estas variables más tarde que el resto. En el caso del *NH3* vemos que solo se ha medido en la estación Boulevard Sud. Las moléculas orgánicas *C8H10*, *C6H6*, *C7H8* sólo se miden en las estaciones de Pista de Silla y Viveros. Cabe destacar también que las estaciones que miden mayor cantidad de variables son Pista de Silla, Viveros y Boulevard Sud. Por otro lado, estaciones como Conselleria Meteo y Nazaret Meteo sólo miden las condiciones ambientales y no las concentraciones de contaminantes, al igual que la estación del Puerto de Valencia con la diferencia que en esta además se miden los niveles de partículas *PM10* y de *SO2*. Además, vemos como en la estación de Valencia Olivereta sólo se miden niveles de partículas *PM2.5*, *PM10* y óxidos de nitrógeno.

Ahora queremos ver qué estaciones han estado activas a lo largo de los años:

```
datos %>%
  mutate(ano = factor(year(Fecha))) %>%
  ggplot(aes(x = ano, fill = Estacion)) +
  geom_bar() +
  labs(x = 'Año') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

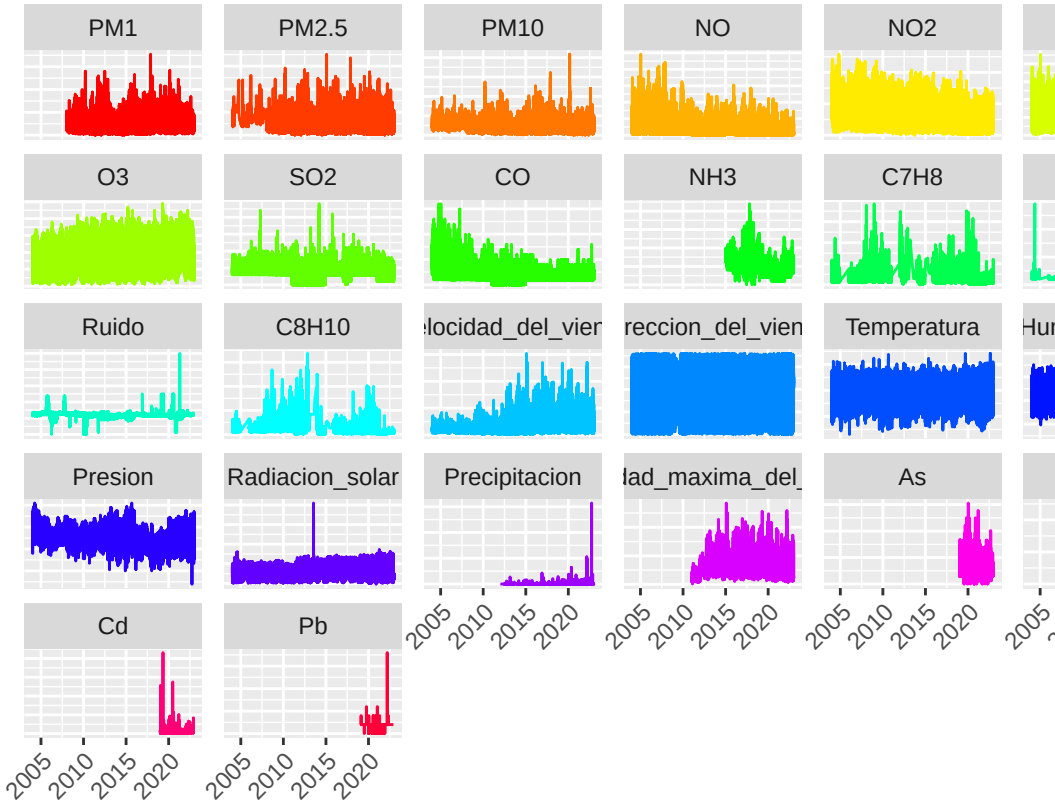


Como podemos ver, desde 2004 hasta 2007 sólo tenemos datos de las estaciones de Pista de Silla y Viveros, que se mantienen actvas hasta 2022. Posteriormente, en los siguientes 5 años (2008-2012) se van añadiendo datos de otras estaciones como la del Politécnico, Av. de Francia, Molí del Sol, Boulevard Sur, Valencia Centro y Conselleria a razón casi de una nueva estación al año y que se mantienen hasta el final (2022). Desde 2013 hasta 2016 se siguen viendo las mismas estaciones y en los siguientes dos años (2017-2018) se añaden datos de la estación del Puerto de Valencia, de la que posteriormente dejamos de obtener datos. Finalmente, desde 2019 hasta 2022 se van añadiendo progresivamente cada año las estaciones de Nazaret, Puerto Molí Trans. Ponent, Puerto antic Turia y Valencia Olivereta. 2022 es el año donde tenemos datos de más estaciones distintas.

A continuación queremos visualizar el período de tiempo entre la primera y la última medida de cada variable

```
datos %>%
  pivot_longer(
    cols = all_of(variables_numericas),
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  mutate(Variables = factor(Variables, levels = variables_numericas, ordered = T)) %>%
  ggplot(aes(x = Fecha, y = valor, color = Variables)) +
  geom_line() +
  facet_wrap(Variables ~ ., scales = "free_y") +
  theme(legend.position = "none") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank(),
```

```
    axis.ticks.y = element_blank()
  ) +
  labs(y = '') +
  scale_colour_manual(values = rainbow(26))
```



A partir de este gráfico podemos ver como algunas variables empiezan a ser medidas a partir de cierta fecha. En concreto, los datos de *As*, *Cd*, *Ni*, *Pb* empiezan a tomarse sólo a partir de 2019. Además, terminan antes que el resto de variables, a principios del 2022.. Otras variables como *Precipitacion*, *Velocidad_maxima_del_viento*,*PM1* o *NH3* empiezan a tener datos un poco más tarde que el resto pero antes del 2019. Todas las demás variables presentan almenos algún dato en 2004.

Tras esta primera visualización de nuestros datos, podemos concluir que la gran cantidad de datos faltantes se debe a una combinación del hecho de que no todas las estaciones miden todas las variables, sumado a que los datos de ciertas variables se empiezan a medir posteriormente que el resto. Esto puede deberse a que la(s) estación(es) que recoge(n) datos de esta variable comienza(n) a funcionar en una fecha posterior al resto de estaciones o simplemente debido a que hasta cierto año no se instalan sensores en las estaciones para medir esas variables. En cualquier caso, dependiendo del análisis que se quiera realizar se habrán de escoger unas variables, y por ende un determinado número de estaciones que midan estas variables y un período de tiempo en que se estén recogiendo datos de estas variables.

En nuestro caso vamos a escoger un período de 10 años, desde 2012 a 2022, ya que en este período de tiempo hay un número significativo de estaciones recogiendo datos y la mayoría de las variables son medidas en este intervalo de tiempo de forma consistente. No obstante, decidimos descartar las variables *Pb*, *Cd*, *Ni*, *As*, *B(a)p* y *NH3* debido a que se comienzan a medir después del 2012 y resultaría poco razonable imputar los datos de estas variables en esos años. De la misma forma, también descartamos las variables *C7H8*, *C6H6*, y *C8H10* debido a que presentan grandes períodos de tiempo con ausencia de datos intercalados entre 2012 y 2022, por lo que también sería poco razonable imputar estos datos.

3. Análisis univariante

En esta sección analizaremos las variables de forma individual con el objetivo de conocer sus magnitudes, es decir, las unidades de medida que representan, y como se distribuyen. Como se ha mencionado previamente, escogemos los datos que comienzan en 2012, concretamente cunado se empiezan a tomar datos de la precipitación, y los datos previos a 2023, concretamente cuando se dejan de tomar los oxidos de nitrogeno.

Variables de interés - Fecha - Estación - Gases: NO, NO2, NOx, SO2, CO, O3 - Meteorológicas: Temperatura, Humedad, Presion, Velocidad viento, Direccion viento, Radiación solar, Precipitación - Calidad de vida: Ruido

Comenzamos con un summary de las variables numéricas para analizar las magnitudes de las variables y como se distribuyen los datos.

```r
variables <-
  c(
    "Fecha",
    "Estacion",
    "NO",
    "NO2",
    "NOx",
    "SO2",
    "CO",
    "O3",
    "Temperatura",
    "Velocidad_del_viento",
    "Direccion_del_viento",
    "Humidad_relativa",
    "Presion",
    "Radiacion_solar",
    "Precipitacion",
    "Ruido"
  )

mystats <- function(x, na.omit = TRUE) {
  if (na.omit)
    x <- x[!is.na(x)]
  min <- min(x)
  Q1 <- quantile(x, 0.25)
  median <- median(x)
  mean <- mean(x)
  dt <- sd(x)
  Q3 <- quantile(x, 0.75)
  max <- max(x)

  n <- length(x)
  IQR <- IQR(x)

  return(round(
    c(
      min = min,
      Q1 = Q1,
      median = median,
      mean = mean,
      dt = dt,
      Q3 = Q3,
```

```
      max = max,
      n = n,
      IQR = IQR
    ),
    2
  ))
}

precipitacion <- datos[, c('Fecha', 'Precipitacion')] %>% na.omit
fecha_inicio <- precipitacion$Fecha[1]
oxidos_n <- datos[, c('Fecha', 'NOx')] %>% na.omit
fecha_fin <- oxidos_n$Fecha[nrow(oxidos_n)]

datos <-
  datos[datos$Fecha > fecha_inicio &
          datos$Fecha < fecha_fin, variables]

sapply(datos[-c(1, 2)], mystats)
```

```
##              NO      NO2      NOx      SO2       CO       O3 Temperatura
## min        0.00     0.00     0.00     0.00     0.00     3.00        4.40
## Q1.25%     3.00    14.00    18.00     3.00     0.10    38.00       13.90
## median     6.00    22.00    32.00     3.00     0.10    53.00       18.50
## mean      10.83    25.58    41.93     3.40     0.16    50.83       18.86
## dt        13.36    15.42    33.85     1.43     0.09    19.15        5.75
## Q3.75%    13.00    35.00    55.00     4.00     0.20    64.00       24.10
## max      152.00   104.00   331.00    24.00     0.80   122.00       34.20
## n      25107.00 25105.00 25107.00 24064.00 12261.00 23958.00    12950.00
## IQR       10.00    21.00    37.00     1.00     0.10    26.00       10.20
##        Velocidad_del_viento Direccion_del_viento Humidad_relativa  Presion
## min                    0.10                 0.00              14.00   968.00
## Q1.25%                 1.00                67.00              55.00  1002.00
## median                 1.60               220.00              67.00  1007.00
## mean                   2.01               179.31              65.81  1007.73
## dt                     1.56               110.28              14.51     8.56
## Q3.75%                 2.40               274.00              76.00  1014.00
## max                   15.20               360.00             100.00  1042.00
## n                  16261.00             16226.00           12652.00 13037.00
## IQR                    1.40               207.00              21.00    12.00
##        Radiacion_solar Precipitacion   Ruido
## min              -7.00          0.00    6.00
## Q1.25%          126.00          0.00   55.00
## median          200.00          0.00   59.00
## mean            200.75          1.26   58.08
## dt               94.03          8.93    6.77
## Q3.75%          281.00          0.00   61.00
## max            1025.00        629.00  250.00
## n             12841.00      11725.00 5157.00
## IQR             155.00          0.00    6.00
```

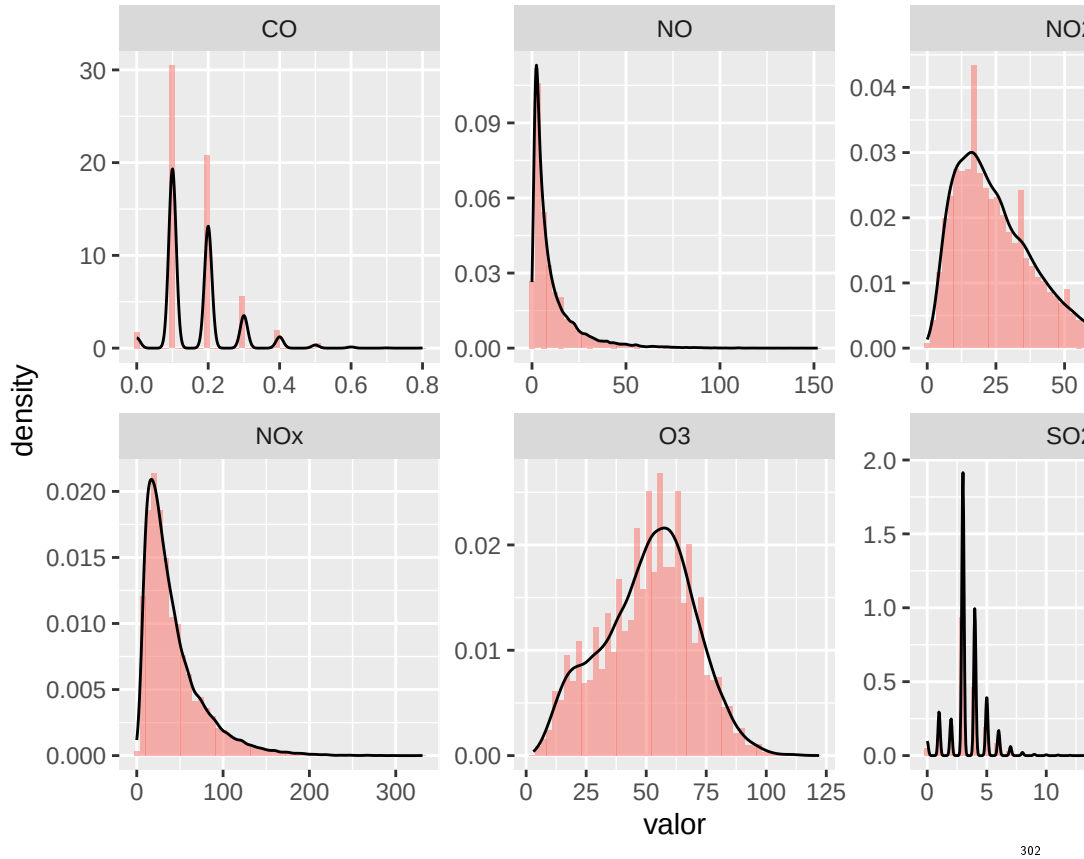
Mediante el summary anterior podemos averiguar cuales son las unidades de medida empleadas contrastando con información externa, ya que en la fuente de los datos no se proporcionan.

- Gases (NO, NO2, NOx, SO2, CO, O3): microgramos por metro cubico (μg/m3)

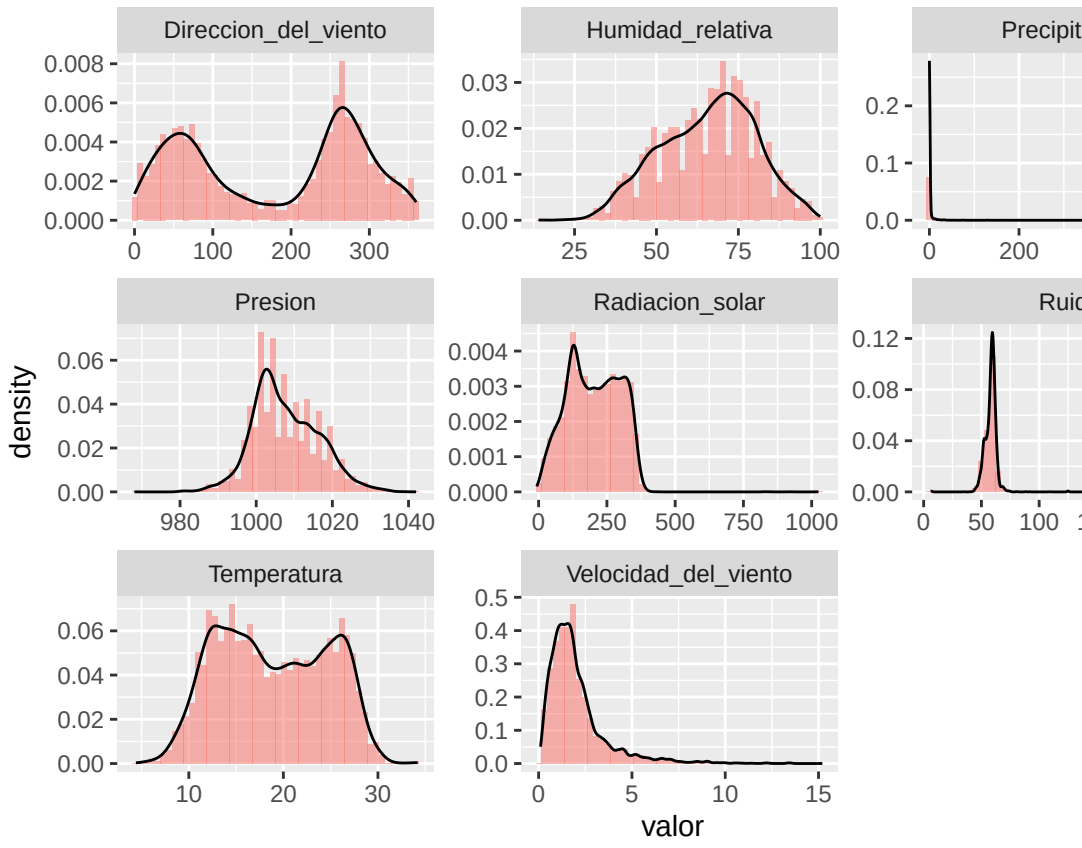
- Temperatura: grados centígrados (C^o)
- Humedad: porcentaje (%)
- Presión: hectopascales (hPa)
- Velocidad del viento: kilometros/hora (m/s)
- Dirección del viento: ángulo de 0 a 360 (^{o})
- Precipitación: litros de agua por metro cuadrado (mm)
- Radiación solar: Vatios por metro cuadrado (W/m2)
- Ruido: decibelios (dB)

Densidades:

```
# Gases
datos[2:8] %>%
  pivot_longer(
    cols = 2:7,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(x = valor)) +
  geom_histogram(aes(
    y = stat(density),
    fill = 'blue',
    alpha = 0.6
  ), bins = 50) +
  geom_density() +
  facet_wrap( ~ Variables, scales = "free") +
  theme(legend.position = "none")
```



```r
# Meteorológicas y Ruido
datos[c(2, c(9:16))] %>%
  pivot_longer(
    cols = 2:9,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(x = valor)) +
  geom_histogram(aes(
    y = stat(density),
    fill = 'blue',
    alpha = 0.6
  ), bins = 50) +
  geom_density() +
  facet_wrap( ~ Variables, scales = "free") +
  theme(legend.position = "none")
```



3.1. Detección y eliminación de outliers con métodos univariante

A continuación, procederemos a detectar y eliminar outliers. Primero, comenzaremos definiendo diferentes funciones posibles para aplicar la detección de estos. Todas estas funciones corresponden a métodos univariantes.

```r
tresSigma <- function(x) {
  media <- mean(x, na.rm = TRUE)
  desviacion <- sd(x, na.rm = TRUE)
  lim_low <- media - 3 * desviacion
```

```
  lim_upp <- media + 3 * desviacion

  outliers <- x[x < lim_low | x > lim_upp]

  return(outliers)
}

hampel <- function(x) {
  mediana <- median(x, na.rm = TRUE)
  madm <- mad(x, constant = 1.4826, na.rm = TRUE)
  umbral <- 3 * madm

  outliers <- x[abs(x - mediana) > umbral]

  return(outliers)
}

boxplot <- function(x) {
  q1 <- quantile(x, 0.25, na.rm = TRUE)
  q3 <- quantile(x, 0.75, na.rm = TRUE)

  iqr <- q3 - q1

  lower_limit <- q1 - 1.5 * iqr
  upper_limit <- q3 + 1.5 * iqr

  outliers <- x[x < lower_limit | x > upper_limit]

  return(outliers)
}

percentil <- function(x) {
  p5 <- quantile(x, 0.05, na.rm = TRUE)
  p95 <- quantile(x, 0.95, na.rm = TRUE)

  outliers <- x[x < p5 | x > p95]

  return(outliers)
}

remove_outliers <- function(x, func) {
  outliers <- func(x)
  return(x[!is.na(outliers)])
}

replace_outliers <- function(x, func) {
  outliers <- func(x)
  x[outliers] <- NA
  return(x)
}
```

Vamos a comprobar el funcionamiento de los métodos de detección de outliers. Lo aplicaremos sobre todas las variables numéricas y comprobaremos el número de outliers que detecta cada función para cada variable.

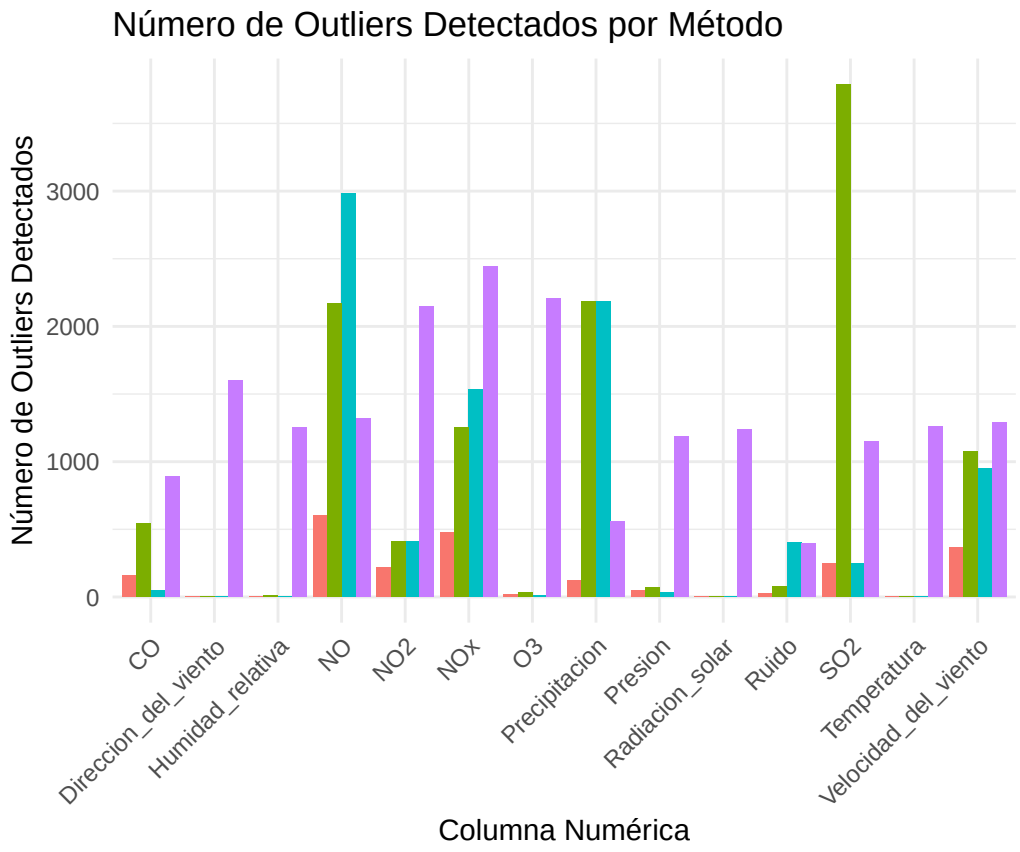
```r
columnas_numericas <- names(datos)[3:16]

m1 <- apply(datos[columnas_numericas], 2, tresSigma)
m2 <- apply(datos[columnas_numericas], 2, hampel)
m3 <- apply(datos[columnas_numericas], 2, boxplot)
m4 <- apply(datos[columnas_numericas], 2, percentil)

m1 <- sapply(sapply(m1, complete.cases), sum)
m2 <- sapply(sapply(m2, complete.cases), sum)
m3 <- sapply(sapply(m3, complete.cases), sum)
m4 <- sapply(sapply(m4, complete.cases), sum)

resultados <- data.frame(
  Metodo = rep(
    c("3-Sigma", "Hampel", "Boxplot", "Percentil"),
    each = length(columnas_numericas)
  ),
  Columna = rep(columnas_numericas, times = 4),
  Outliers_Detectados = c(m1, m2, m3, m4)
)

ggplot(resultados,
       aes(x = Columna, y = Outliers_Detectados, fill = Metodo)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Número de Outliers Detectados por Método",
       x = "Columna Numérica",
       y = "Número de Outliers Detectados") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Podemos observar cómo la regla del percentil siempre elimina datos, esto es debido a la naturaleza de la regla, la cual siempre eliminará el 2.5% que quede por arriba y el 2.5% por abajo. Podemos usar esto como comparador de el resto de métodos.

El resto de métodos, en la mayoría de ocasiones quedan por debajo del criterio del percentil. No obstante, es visible el agresivo comportamiento de la regla boxplot y el excesivamente poco agresivo método de la regla 3-sigma. En cuanto al comportamiento de la regla de Hampel, parece comportarse como una versión agresiva pero manteniendo la prudencia de la regla 3-sigma.

Pese a que, según las matemáticas, cuantos más datos tenemos, más se parece la distribución de los mismos a una Gaussiana, hemos visto anteriormente que no todos los datos siguen esta distribución.

Por este motivo, hemos decidido utilizar la regla de Hampel para la detección de los outliers, ya que esta es una regla robusta a distribuciones que no sean Gaussianas.

A continuación, reemplazaremos los outliers detectados por NA.

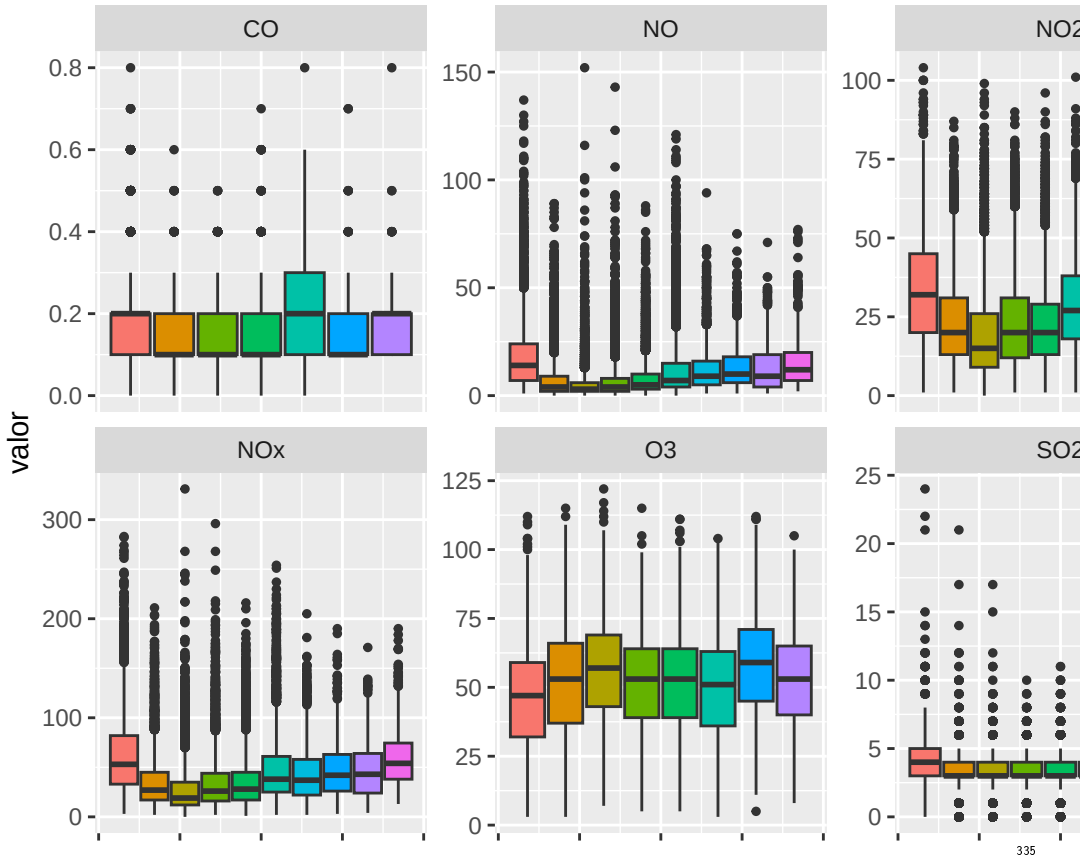
```
datos_no_outliers <- datos %>%
  mutate_at(vars(all_of(columnas_numericas)), ~ replace_outliers(., hampel))
```

4. Analisis bivariante

Partiendo de las conclusiones obtenidas en el apartado de análisis de datos faltantes, nos parece interesante ver si existe alguna influencia en el valor de la variable por parte de la estación de la que proviene. En caso de no existir una influencia considerable podemos utilizar los valores obtenidos en otras estaciones para imputar otros. Finalmente comprobaremos si año influye en el valor de la variable, para ver si a lo largo del tiempo ha ocurrido algún cambio considerable.

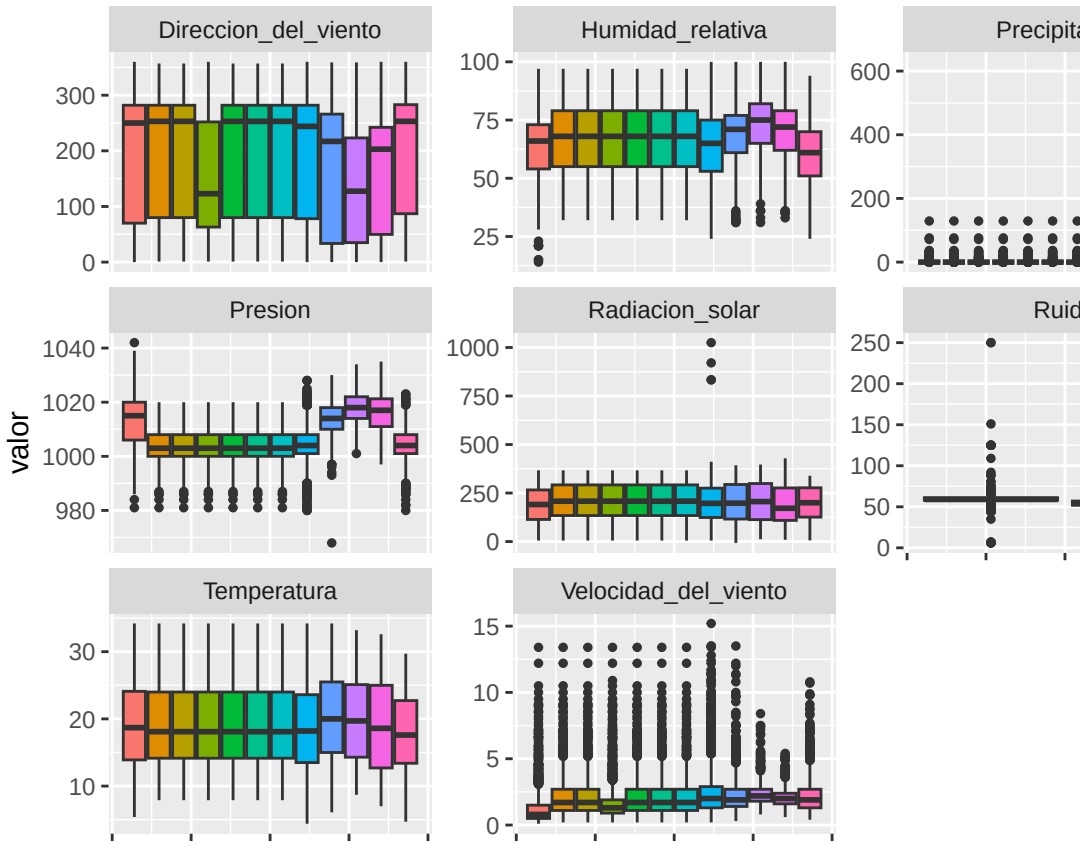
Mediante boxplots vamos a analizar como se distribuyen los datos respecto a las estaciones. En primer lugar los gases:

```r
# Gases
datos_no_outliers %>%
  select(all_of(variables[2:8])) %>%
  pivot_longer(
    cols = 2:7,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(y = valor, fill = Estacion)) +
  geom_boxplot(outlier.size = 1) +
  facet_wrap( ~ Variables, scales = "free_y") +
  theme(axis.text.x = element_blank(), legend.position = "none")
```



En segundo lugar las variables meteorológicas:

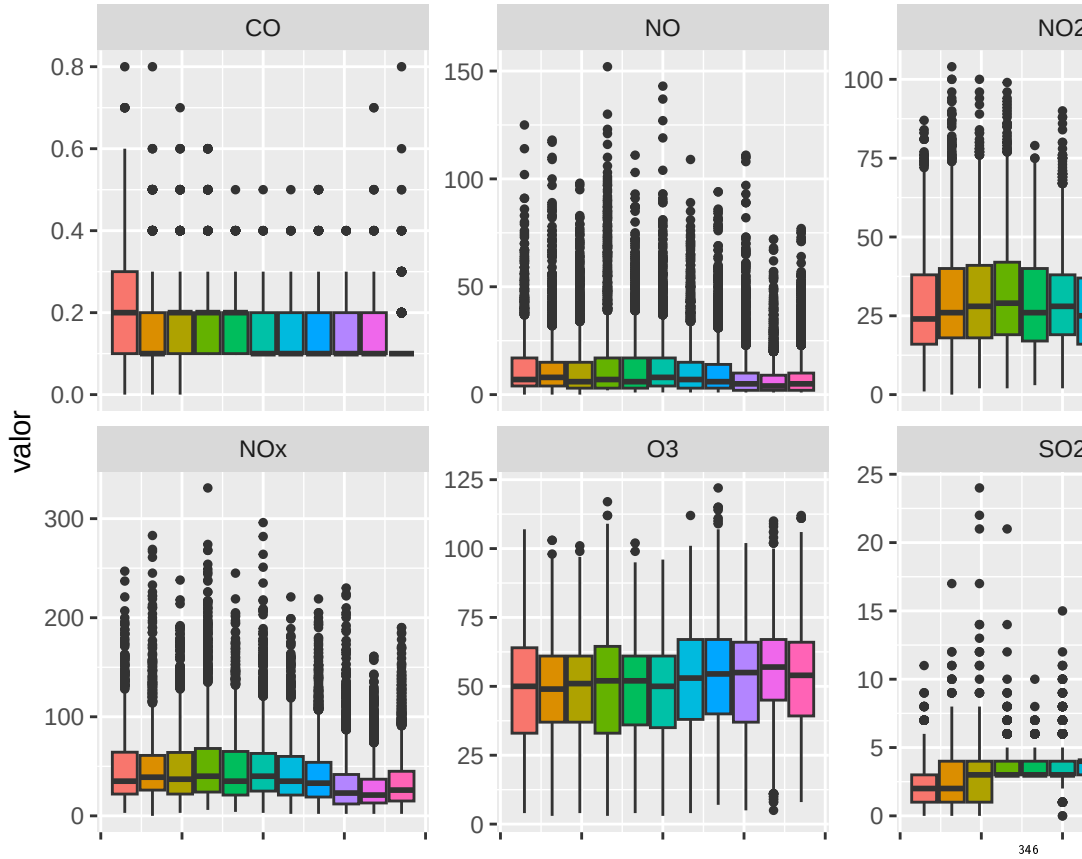
```r
# Variables meteorológicas y ruido
datos_no_outliers %>%
  select(all_of(variables[c(2, c(9:16))])) %>%
  pivot_longer(
    cols = 2:9,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(y = valor, fill = Estacion)) +
  geom_boxplot(outlier.size = 1) +
  facet_wrap( ~ Variables, scales = "free_y") +
  theme(axis.text.x = element_blank(), legend.position = "none")
```



Podemos concluir con que generalmente la estación no influye en los valores de las variables. Esta información nos es útil para la imputación de NA's, ya que podemos emplear la media de las medidas obtenidas por otras estaciones. Respecto al ruido, el cual únicamente se mide en las estaciones de "Pista Silla" y "Viveros" si parece tener un valor un poco mayor en "Pista Silla". Esto puede deberse a que esta estación se encuentra en un lugar con mayor tránsito de coches.

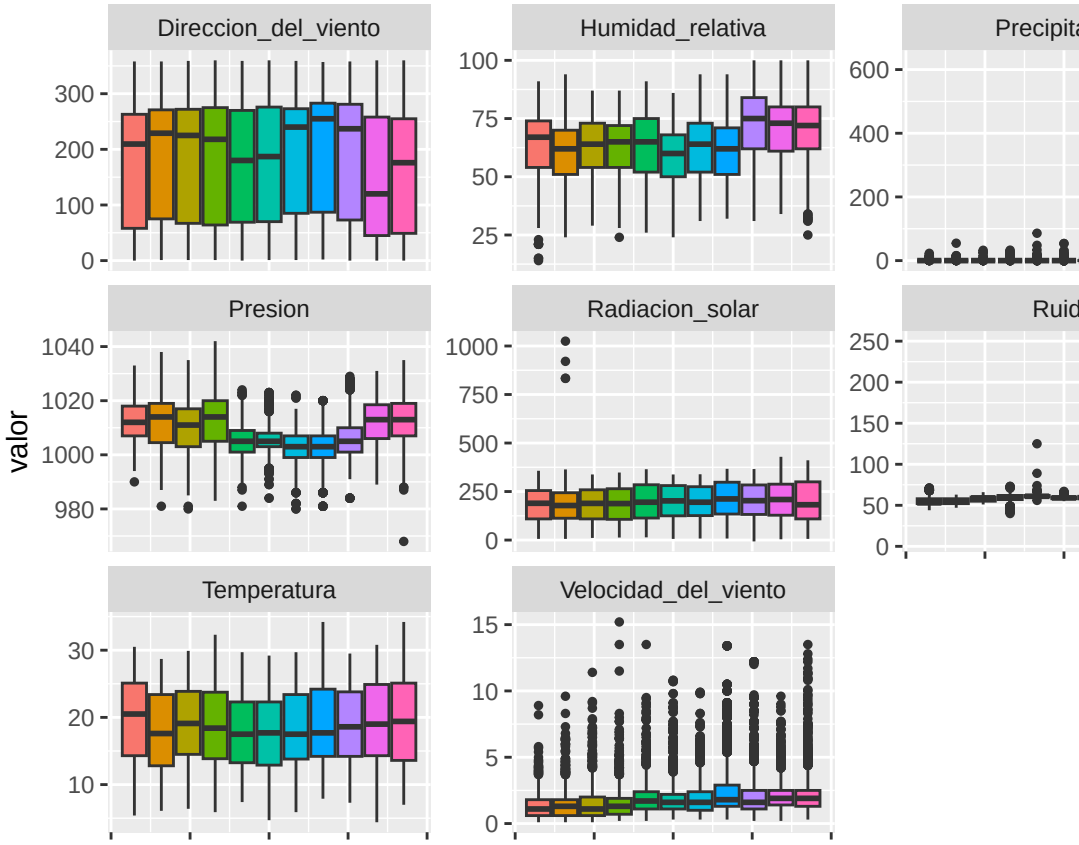
Ahora vamos a ver una sencilla evolución mediante boxplots de el valor de las variables a lo largo de los años. Comenzamos con los gases:

```
# Gases
datos_no_outliers %>%
  select(all_of(variables[c(1, 3:8)])) %>%
  mutate(ano = factor(year(Fecha))) %>%
  pivot_longer(
    cols = 2:7,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(y = valor, fill = ano)) +
  geom_boxplot(outlier.size = 1) +
  facet_wrap( ~ Variables, scales = "free_y") +
  theme(axis.text.x = element_blank(), legend.position = "none")
```



Se puede apreciar una leve reducción a lo largo de los años de los gases considerados como contaminantes a excepción del SO2 que parece mantenerse constante. Por otro lado vemos un leve aumento del ozono (O3), por lo que podría existir una relación con la disminución de gases contaminantes. Continuamos con las variables meteorológicas:

```
# Gases
datos_no_outliers %>%
  select(all_of(variables[c(1, 9:16)])) %>%
  mutate(ano = factor(year(Fecha))) %>%
  pivot_longer(
    cols = 2:9,
    names_to = 'Variables',
    values_to = 'valor',
    values_drop_na = T
  ) %>%
  ggplot(aes(y = valor, fill = ano)) +
  geom_boxplot(outlier.size = 1) +
  facet_wrap( ~ Variables, scales = "free_y") +
  theme(axis.text.x = element_blank(), legend.position = "none")
```



Como era de esperar, las variables meteorológicas no han sufrido variaciones a lo largo de los años. Por otro lado si que se aprecia un leve aumento en el ruido.

4.1. Imputación de NA's

Centrándonos en las variables de interés, si bien son más consistentes en lo que se refiere a las entradas diarias de cada variable, todavía hay muchos datos faltantes. No obstante, podemos aprovechar el hecho de que los datos diarios de las variables en todas las estaciones de Valencia suelen ser muy parecidos, por lo que decidimos imputar los datos faltantes en cada día con la media de esta variable sobre todas las estaciones. Tras realizar esta imputación, todavía hay algunas variables que presentan NA's, esto se debe a que hay algunos días donde donde esta variable no se ha medido en ninguna de las estaciones. Esto sólo ocurre en el 11% de las observaciones, por lo que decidimos desprendernos de estas observaciones, ya que el volúmen de datos es suficientemente grande y los datos todavía pueden considerarse diarios.

```
imputacion_NAs <- function(x) {
  ifelse(is.na(x), mean(x, na.rm = T), x)
}

datos_limpios <- datos_no_outliers %>%
  # Esta hecho arriba
  #filter(Fecha >= as.Date('2012-01-01', format = '%Y-%m-%d'), Fecha <= as.Date('2
  #select(-c('Pb_(ng/m3)','Cd_(ng/m3)','Ni_(ng/m3)','As_(ng/m3)','B(a)p_(ng/m3)','
  group_by(Fecha) %>%
  mutate(across(.cols = all_of(columnas_numericas), imputacion_NAs)) %>%
  mutate(across(all_of(columnas_numericas), reemplazo_especiales)) %>%
  ungroup
```

```r
100 * sum(complete.cases(datos_limpios)) / nrow(datos_limpios)
```

```
## [1] 87.88126
```

```r
datos_limpios <- datos_limpios[complete.cases(datos_limpios), ]
```

4.2. Correlaciones

La correlación puede tener valores en el rango [-1,1]. Siendo -1 el indicador de un comportamiento absolutamente contrario y siendo 1 la misma variable, pues tiene exactamente el mismo comportamiento.

Esto puede darnos información para reflexionar sobre la naturaleza y comportamiento de los datos.

```r
library(GGally)

# Correlacion Pearson
corr_P <- datos_limpios[3:15] %>%
  na.omit %>%
  cor
corr_P

# Correlacion Spearman
corr_S <- datos_limpios[3:15] %>%
  na.omit %>%
  cor(method = 'spearman')
corr_S

ggcorr(datos_limpios[3:15])
```

Basándonos en las observaciones, podemos extraer las siguientes conclusiones:

- Existe una correlación negativa importante entre NOs y Ozono. Esto podría deberse a que la presencia de óxidos de nitrógeno puede contribuir a la degradación del ozono en la atmósfera, lo que tiene implicaciones para la calidad del aire, la contaminación y el efecto hivernadero.
- Correlación positiva de NOx con el resto de NO (NO, NO2...). Como era de esperar y como se ha mencionado en la definición de las variables, NOx representa el conjunto de oxidos de nitrogeno, entre ellos el NO y el NO2.
- Correlación positiva entre Temperatura y Radiación Solar. Esto es coherente con las estaciones más cálidas que a menudo experimentan más horas de sol y mayor radiación solar.
- Correlación entre Radiación Solar y Ozono. Esto tiene sentido ya que la capa de ozono filtra la mayor parte de la radiación ultravioleta proveniente del sol, por lo tanto están muy relacionados entre ellos.

También hemos filtrado las correlaciones para mostrar únicamente las mayores correlaciones. Hemos puesto un umbral para escoger solo las correlaciones que sean mayores a 0.7 para visualizar las variables muy correlacionadas (eliminando las correlaciones de una variable con ella misma). Hemos puesto el ejemplo en el que las correlaciones son directas para la siguiente visualización. Podría realizarse esto para las correlaciones inversas.

```r
df <- cor(na.omit(datos_limpios[3:15])) %>%
  as.data.frame(.)

df$Variable1 <- rownames(df)

cor <- df %>%
  pivot_longer(cols = -Variable1,
               names_to = "Variable2",
               values_to = "Correlation")

correlaciones <-
  cor[cor$Correlation > 0.7, ] %>% arrange(desc(Correlation)) %>% filter(Variable1
correlaciones_inversas <-
  cor[cor$Correlation < -0.7, ] %>% arrange(Correlation) %>% filter(Variable1 != V
paste0(
  "El numero total de correlaciones inversas es ",
  nrow(correlaciones_inversas),
  ". Por lo tanto, solo estudiaremos las correlaciones directas"
)

ggplot(correlaciones,
       aes(x = Variable1, y = Variable2, color = Correlation)) +
  geom_tile(aes(fill = Correlation)) +
  theme_minimal() +
  labs(title = "Correlaciones Positivas")
```

4.3. Detección y eliminación de outliers con métodos multivariable

En este apartado, trabajaremos con el dataframe de datos que contiene la imputación de NA.

```r
distancia_mahalanobis <- function(x) {
  media = colMeans(x)
  S = cov(x)

  dist = mahalanobis(x, media, S)
  umbral <- qchisq(0.95, df = ncol(x))

  outliers <- which(dist > umbral)

  return(outliers)
}

instancias_outliers <-
  distancia_mahalanobis(datos_limpios[columnas_numericas])
paste0(
  "La distancia de mahalanobis detecta como outliers ",
  length(instancias_outliers),
  " datos con una confianza del 95%. Con estos datos, lo que haremos no es cambiar
)
```

```
## [1] "La distancia de mahalanobis detecta como outliers 1937 datos con una confi
```

```
datos_limpios_sin_outliers <- datos_limpios[-instancias_outliers, ]
```

5. Resolución de preguntas planteadas sobre los datos

En esta sección se resolverán las diferentes preguntas planteadas sobre los datos. ## Influencia de carril bici

Para estudiar la influencia de un carril bici por el centro de Valencia, estudiaremos la evolución de gases contaminantes en diversas estaciones de la ciudad. Mediremos la evolución de los gases:

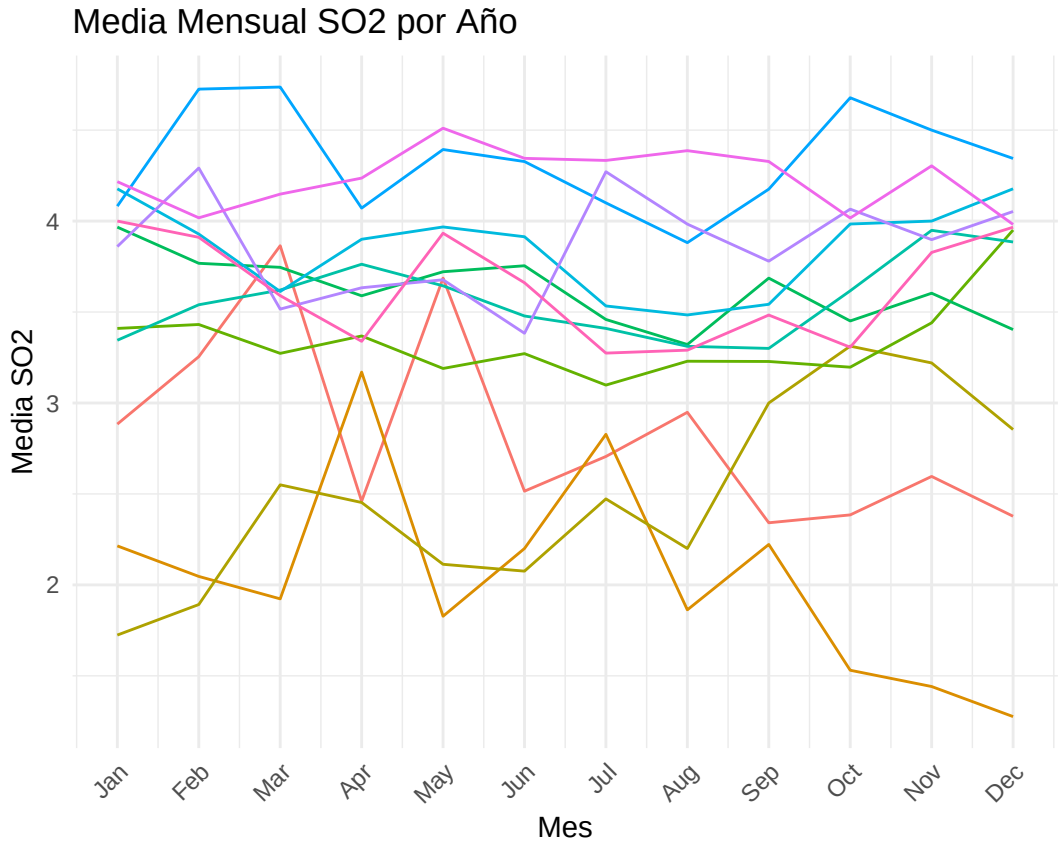
- SO2: Originado sobre todo durante la combustión de carburantes fósiles.
- NO2: Es un contaminante atmosférico cuyas fuentes fundamentales son el tráfico rodado, así como las emisiones de determinadas industrias y grandes instalaciones de combustión.

Además, las estaciones deberían ser las más céntricas, ya que ahí el efecto del carril bici y las restricciones de acceso deberían hacerse más notables. Observaremos las siguientes estaciones:

- Avda. Francia
- Bulevard Sud
- Valencia Centro
- Olivereta

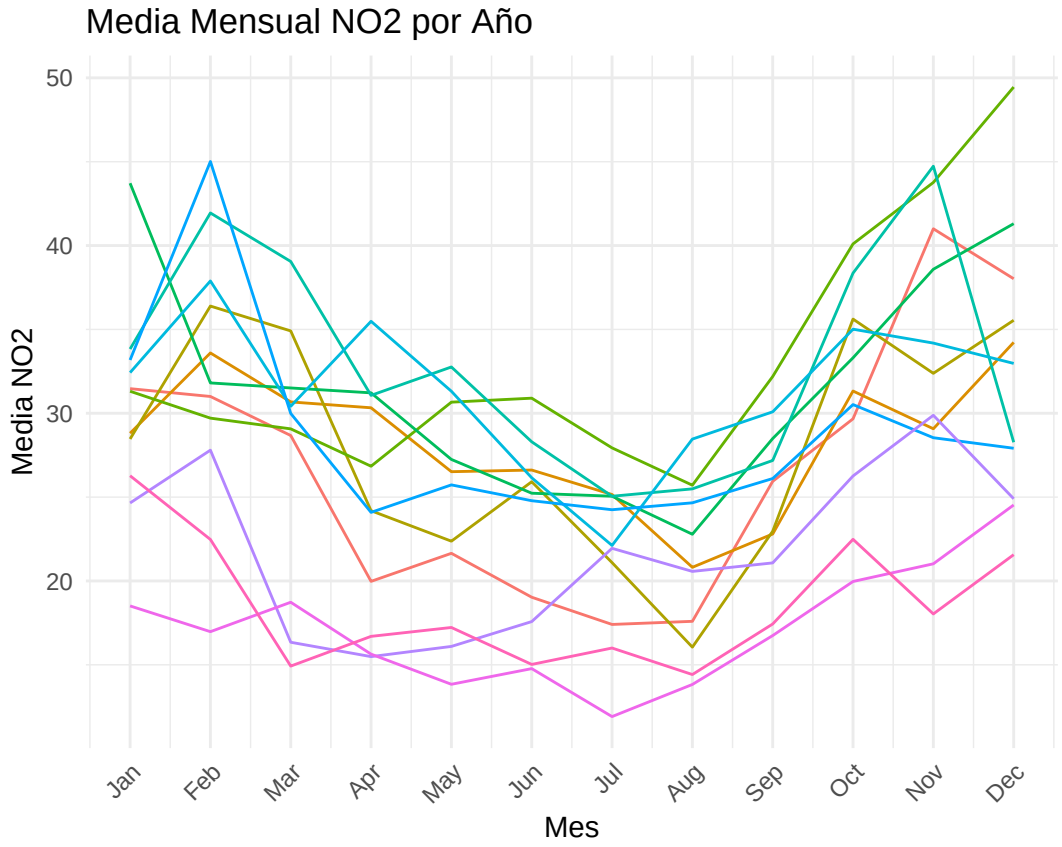
```
datos_mensuales <- datos[c("Fecha", "SO2", "NO2", "Estacion")] %>%
  filter(Estacion %in% c("Avda. Francia", "Bulevard Sud", "Valencia Centro", "Oliv
  mutate(ano = year(Fecha), mes = month(Fecha)) %>%
  arrange(Fecha) %>%
  group_by(ano, mes) %>%
  summarise(
    media_SO2 = mean(SO2, na.rm = TRUE),
    media_NO2 = mean(NO2, na.rm = TRUE),
    .groups = "drop"
  )

ggplot(data = datos_mensuales, aes(
  x = mes,
  y = media_SO2,
  group = ano,
  color = factor(ano)
)) +
  geom_line() +
  labs(x = "Mes", y = "Media SO2", title = "Media Mensual SO2 por Año") +
  scale_color_discrete(name = "Año") +
  scale_x_continuous(breaks = 1:12,
                     labels = month.abb,
                     limits = c(1, 12)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



412

```r
ggplot(data = datos_mensuales, aes(
  x = mes,
  y = media_NO2,
  group = ano,
  color = factor(ano)
)) +
  geom_line() +
  labs(x = "Mes", y = "Media NO2", title = "Media Mensual NO2 por Año") +
  scale_color_discrete(name = "Año") +
  scale_x_continuous(breaks = 1:12,
                     labels = month.abb,
                     limits = c(1, 12)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



NOTA: El gráfico de NO2 se ve mucho mejor la reducción cuando se observan todas las estaciones.

Si bien no se observan cambios significativos en los niveles de SO2, en los niveles de NO2 sí que podemos observar una evolución.

Respecto al NO2 si que se puede apreciar una reducción a lo largo de los años. Con unos niveles más bajos especialmente en los últimos 3 años. Además se puede apreciar una periodicidad, concretamente una reducción de los niveles de NO2 en los meses de verano. Esto coincide con el momento del año donde menos gente vive en la ciudad y por lo tanto menos tráfico hay. En conclusión parece que las medidas de transporte que se han tomado en la ciudad han influenciado en la reducción de gases contaminantes.

5.1. Relación entre la calidad del aire y el dia de la semana

Aqui observamos si, en general la calidad del aire se ve influenciada por el día de la semana. Queremos sobre todo fijarnos en la distinción entre días laborables y de descanso. Podemos comprobar cómo, efectivamente, existe una disminución en los gases contaminantes los días festivos con respecto a los días laborables.

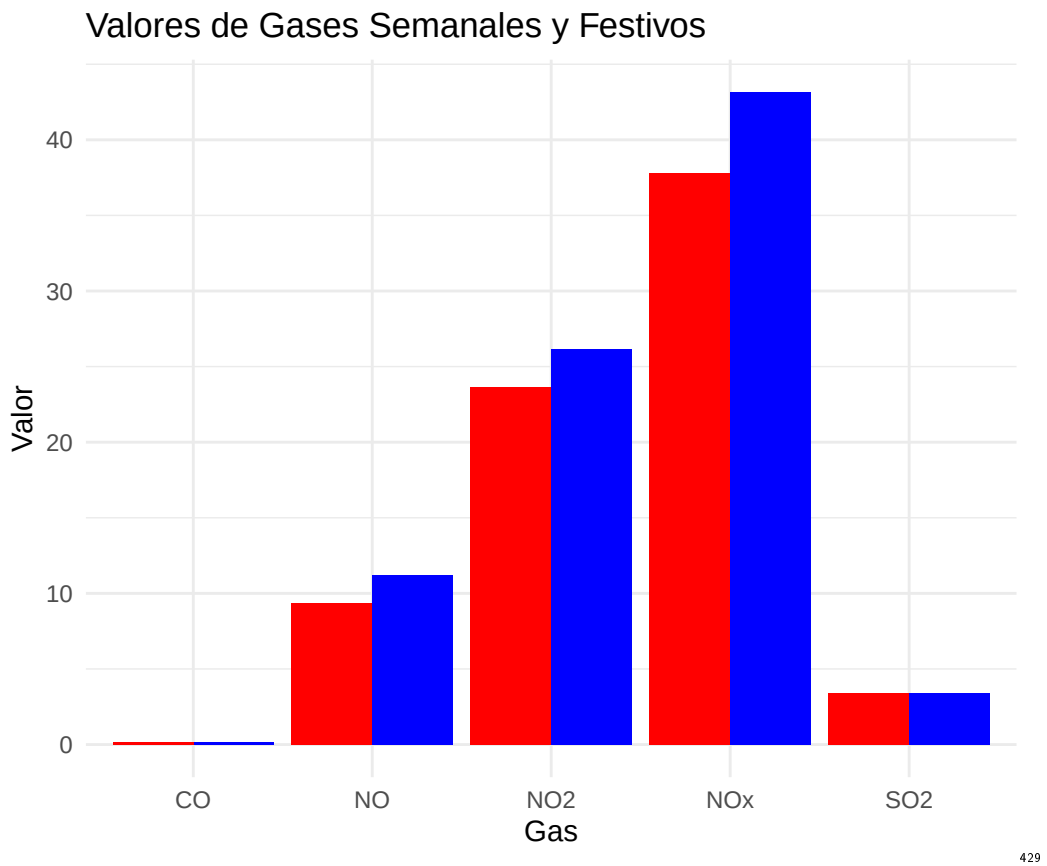
```
gases_interes <- c("NO", "NO2", "NOx", "SO2", "CO")

datos_semanales <- datos_limpios %>%
  mutate(Dia_de_la_semana = factor(wday(Fecha))) %>%
  filter(!(Dia_de_la_semana %in% c(1, 6))) %>%
  select(gases_interes) %>%
  summarise_all(.funs = list(mean)) %>%
  pivot_longer(cols = everything(),
               names_to = "Gas",
               values_to = "Valor")
```

```r
datos_festivos <- datos_limpios %>%
  mutate(Dia_de_la_semana = factor(wday(Fecha))) %>%
  filter(Dia_de_la_semana %in% c(1, 6)) %>%
  select(gases_interes) %>%
  summarise_all(.funs = list(mean)) %>%
  pivot_longer(cols = everything(),
               names_to = "Gas",
               values_to = "Valor")

datos_combinados <- rbind(
  transform(datos_semanales, Tipo = "Semanal"),
  transform(datos_festivos, Tipo = "Festivo")
)

ggplot(datos_combinados, aes(x = Gas, y = Valor, fill = Tipo)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Valores de Gases Semanales y Festivos",
       x = "Gas",
       y = "Valor") +
  scale_fill_manual(values = c("Semanal" = "blue", "Festivo" = "red")) +
  theme_minimal() +
  theme(legend.title = element_blank())
```

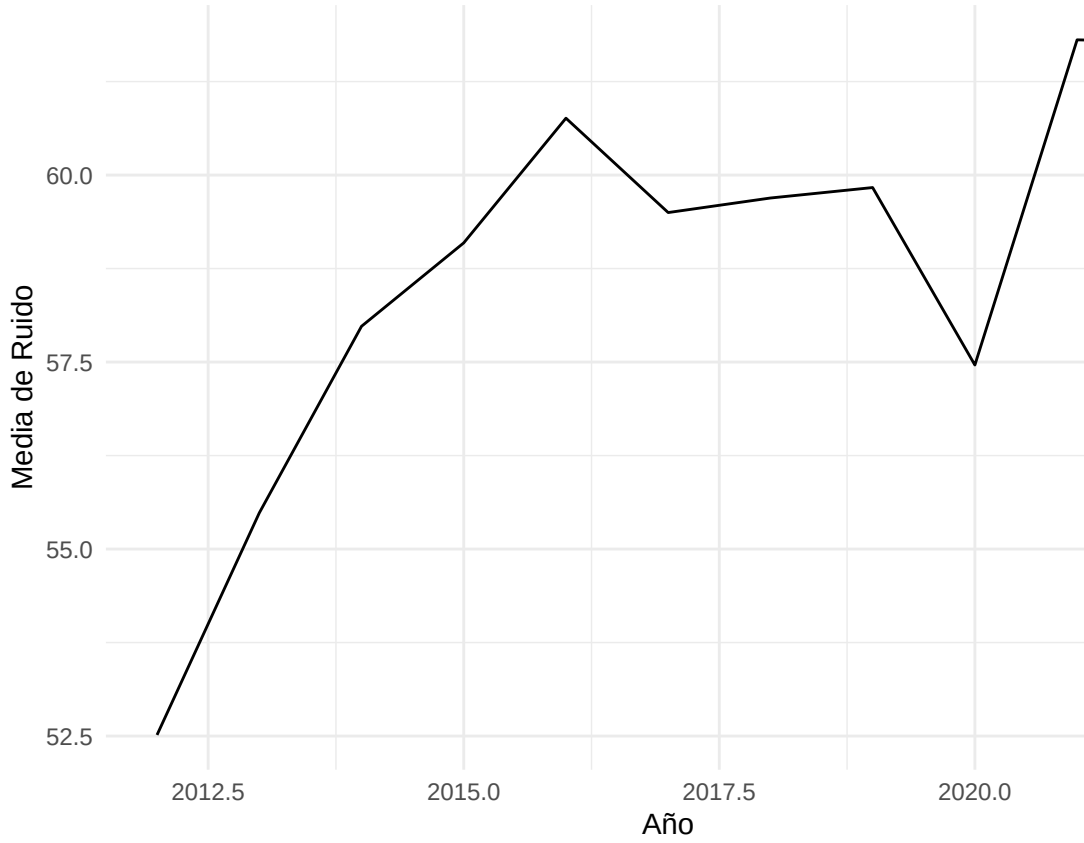


5.2. Evolución de la contaminación sonora a lo largo de los años

Podemos observar un incremento visibleen la medición del ruido medio a lo largo de los años. Obviamente, en los datos del 2020 podemos observar un decrecimiento de los mismos, producto del parón de la pandemia, pero vemos cómo continúa aun así subiendo.

```
datos_ruido <- datos_limpios[c("Fecha", "Ruido")] %>%
  mutate(Ano = year(Fecha)) %>%
  select(-Fecha) %>%
  group_by(Ano) %>%
  summarize(Media_Ruido = mean(Ruido, na.rm = TRUE, .groups = "drop"))

ggplot(datos_ruido, aes(x = Ano, y = Media_Ruido)) +
  geom_line() +
  labs(x = "Año", y = "Media de Ruido") +
  theme_minimal() +
  scale_y_continuous(limits = c(min(datos_ruido$Media_Ruido), max(datos_ruido$Medi
```

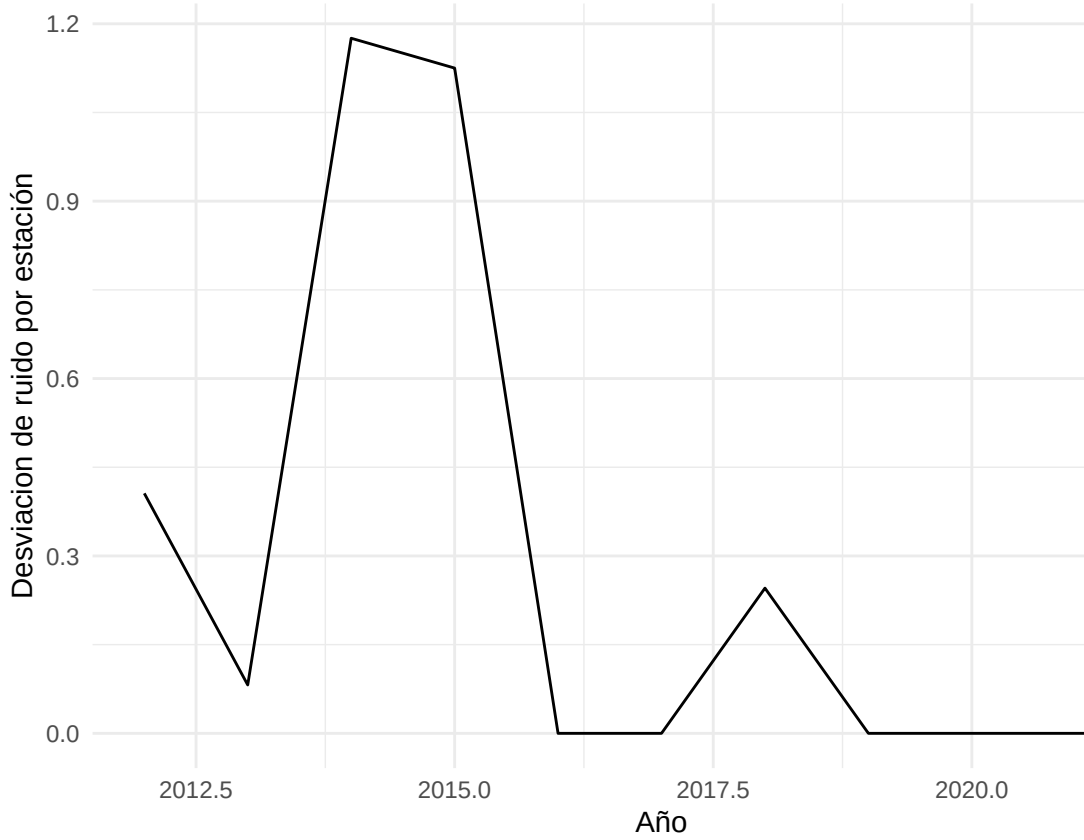


Ahora comprobaremos si este ruido medido depende de la zona dónde la midamos. Para ello, vamos a observar el comportamiento de la varianza de la media de los ruidos agrupados por estaciones. Calcularemos la desviación para cada año y, finalmente, la media de todas las desviaciones Si el valor es pequeño, entonces podemos afirmar que el incremento en el ruido no es de ciertas zonas, sino generalizado.

```
datos_ruido_estaciones <-
  datos_limpios[c("Fecha", "Estacion", "Ruido")] %>%
  mutate(Ano = year(Fecha)) %>%
  select(-Fecha) %>%
  group_by(Ano, Estacion) %>%
  summarize(Media_Ruido = mean(Ruido, na.rm = TRUE, .groups = "drop")) %>%
  group_by(Ano) %>%
  summarize(Desviacion_por_ano = sd(Media_Ruido, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Ano'. You can override using the `.groups`
## argument.
```

```
ggplot(datos_ruido_estaciones, aes(x = Ano, y = Desviacion_por_ano)) +
  geom_line() +
  labs(x = "Año", y = "Desviacion de ruido por estación") +
  theme_minimal()
```




```
mean(datos_ruido_estaciones$Desviacion_por_ano)
```

```
## [1] 0.2758076
```

Podemos observar que, para los valores que tratamos (sobre 50 y 60), una desviación del 0.3 es despreciable.

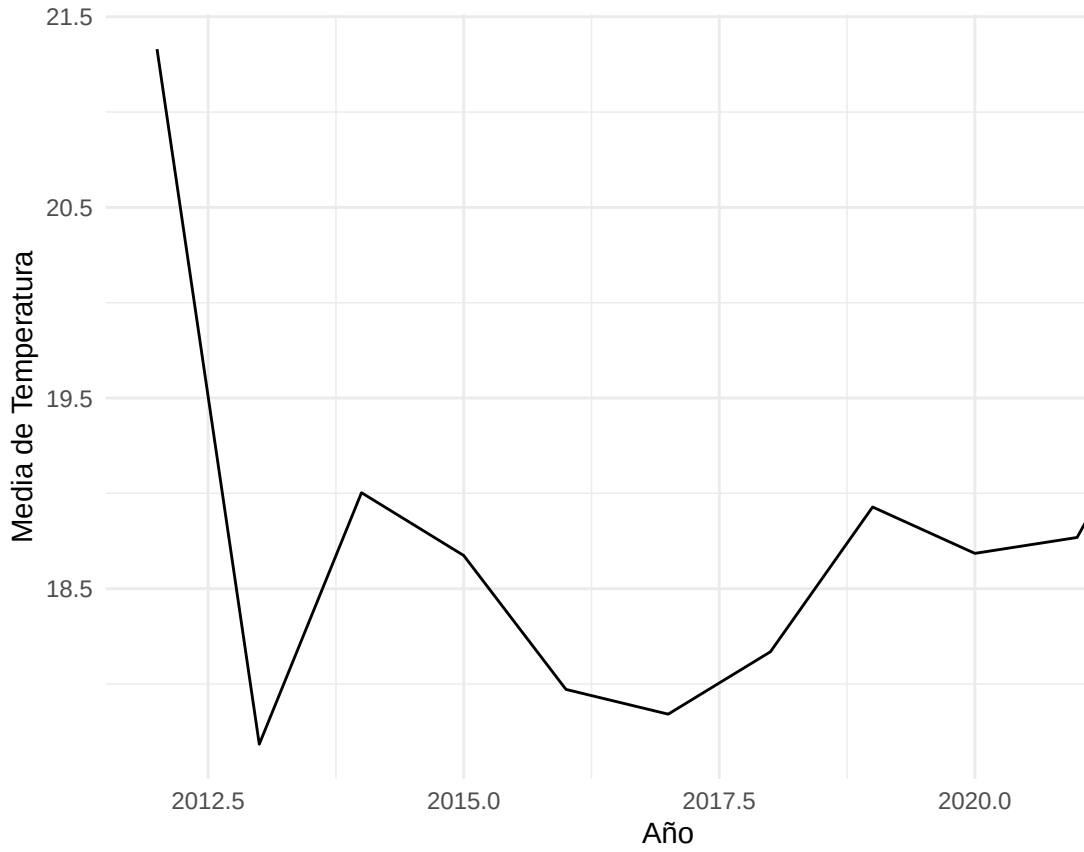
5.3. Evolución de la temperatura a lo largo de los años

También se puede observar el aumento generalizado de la temperatura a lo largo de los años. Es curioso que, pese a que en el mundo existe un aumento generalizado de la temperatura, en Valencia estuvo bajando esta tendencia y no es hasta el año 2017 que comenzaron a subir de nuevo. Pese a lo mencionado, bastante visible el hecho de que nos encontramos en una tendencia creciente.

```
datos_temperatura <- datos_limpios[c("Fecha", "Temperatura")] %>%
  mutate(Ano = year(Fecha)) %>%
  select(-Fecha) %>%
  group_by(Ano) %>%
  summarize(Media_Temperatura = mean(Temperatura, na.rm = TRUE, .groups = "drop"))

ggplot(datos_temperatura, aes(x = Ano, y = Media_Temperatura)) +
  geom_line() +
  labs(x = "Año", y = "Media de Temperatura") +
```

```
  theme_minimal() +
  scale_y_continuous(limits = c(
    min(datos_temperatura$Media_Temperatura),
    max(datos_temperatura$Media_Temperatura)
  ))
```



5.4. Cómo se ha comportado la contaminación a lo largo de los años

Cuando contestamos a la primera pregunta, nos centramos en gases que soltaban los carburantes y solo observamos las estaciones más cercanas al centro de Valencia. En esta ocasión, estudiaremos el comportamiento de todos los gases contaminantes en todas las estaciones.

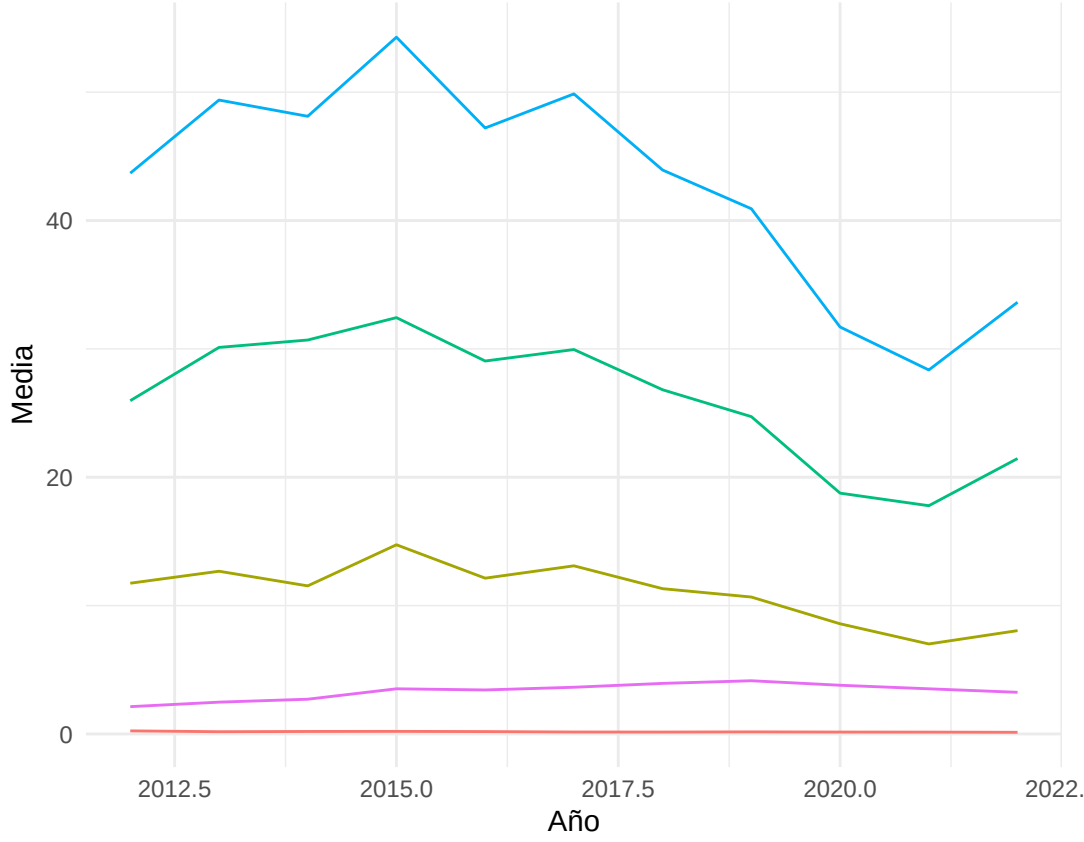
Esta vez podemos observar que se produjo un pico de todos los gases en el año 2016, pero que por lo general las medidas de estos se han mantenido bastante estables o incluso decrecientes en el tiempo.

```
datos_gases <-
  datos_limpios[c("Fecha", "NO", "NO2", "NOx", "SO2", "CO")] %>%
  mutate(Ano = year(Fecha)) %>%
  select(-Fecha) %>%
  pivot_longer(
    cols = c("NO", "NO2", "NOx", "SO2", "CO"),
    names_to = "Gas",
    values_to = "Valor"
  ) %>%
  group_by(Ano, Gas) %>%
  summarize(Media = mean(Valor, na.rm = TRUE, .groups = "drop"))
```

```
## `summarise()` has grouped output by 'Ano'. You can override using the `.groups`
```

```
## argument.
```

```
ggplot(datos_gases, aes(x = Ano, y = Media, color = Gas)) +
  geom_line() +
  labs(x = "Año") +
  theme_minimal()
```



Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	linear dichroism