Guía práctica de introducción al Análisis Exploratorio de Datos

GOBIERNO DE ESPAÑA
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
red.es
Iniciativa aporta

Índice de contenidos

Introducción

Una de las tareas habituales a las que se enfrenta todo analista o científico de datos es la de entender las características de las variables con las que se está trabajando. **Explorar, entender y evaluar la calidad de los datos es una condición previa al procesamiento de los mismos.** Estas acciones son necesarias para tener una aproximación a los datos antes de realizar cualquier análisis y, además, porque muchas de las técnicas estadísticas de análisis de datos presuponen el cumplimiento de unas condiciones previas para poder garantizar la objetividad e interoperabilidad de los datos. Por ejemplo, es necesario detectar y tratar los datos atípicos dado su impacto sobre algunos estadísticos, como el cálculo de la media. Una forma de llevar a cabo este pre-procesamiento es mediante **análisis exploratorios de datos (AED) o *Exploratory data analysis* (EDA)**.

El análisis exploratorio de los datos se refiere al **conjunto de técnicas estadísticas cuyo objetivo es explorar, describir y resumir la naturaleza de los datos y comprender las relaciones existentes entre las variables de interés, maximizando la comprensión del conjunto de datos**. Independientemente de la composición de los datos y de los análisis estadísticos que se realicen posteriormente, un análisis exploratorio de datos posee importantes ventajas: una exploración minuciosa de los datos permite identificar posibles errores (datos incorrectamente introducidos, detectar la ausencia de valores o una mala codificación de las variables), revelar la presencia de valores atípicos (*outliers*), comprobar la relación entre variables (correlaciones) y su posible redundancia o realizar un análisis descriptivo de los datos mediante representaciones gráficas y resúmenes de los aspectos más significativos. Desafortunadamente, esta exploración de los datos se descuida con demasiada frecuencia por parte de los reutilizadores de datos y es una parte esencial de cualquier análisis estadístico para que los resultados sean consistentes y veraces. Por esta razón, planteamos esta guía introductoria que te mostrará de forma sencilla una serie de tareas que siempre debes incorporar en tus EDA.

Es necesario indicar que existe cierta controversia respecto a qué tareas deben considerarse parte del análisis exploratorio de datos. Algunos autores consideran que la limpieza de datos (re-ajuste de las variables, detección y tratamiento de datos ausentes y valores atípicos, entre otras tareas) es una función previa al análisis exploratorio de datos, a la cual se refieren como *data cleaning*. En nuestra opinión, delimitar las acciones referentes al análisis exploratorio de datos es complejo, ya que la mayoría de los procesos están íntimamente ligados y el orden de los procesos puede verse alterado por la naturaleza de los datos. En esta guía introductoria se detalla una serie de tareas que, desde nuestro punto de vista, constituyen el conjunto mínimo a abordar para garantizar un punto de partida aceptable para una reutilización de datos eficaz.

Por último, hay que señalar que **el público objetivo de esta guía es principalmente el usuario de datos abiertos cuya meta es realizar algún tipo de tratamiento orientado a extraer la máxima información de los datos** a través, por ejemplo, del desarrollo de visualizaciones o la realización de estudios de investigación.

Para lograr la máxima comprensión del alcance de esta guía es recomendable que el lector posea competencias básicas en el lenguaje R que es el elegido para ilustrar

mediante ejemplos, las diferentes etapas involucradas en un EDA. Si no es así, te animamos igualmente a continuar la lectura de esta guía dado que, como verás a continuación, dispones de una interesante bibliografía que además de ayudarte a entender EDA, te permitirá conocer y obtener el máximo partido de este potente lenguaje de programación.

1. Metodología

Con esta guía se pretende facilitar al lector el aprendizaje de las técnicas propuestas mediante el **desarrollo de un caso práctico**, pudiendo experimentar de forma autodidacta con datos públicos y herramientas tecnológicas *Open Source* y gratuitas. El ejemplo que se detalla en esta guía **utiliza datos abiertos y se pone a disposición del lector** para que el código se pueda replicar o servir de base para otros análisis de datos.

Como herramienta para el caso práctico hemos utilizado el lenguaje de programación R y el entorno de desarrollo RStudio. **R es un lenguaje de uso frecuente por los profesionales en tratamiento de datos, ampliamente utilizado por su potencia y relativa sencillez para abordar las tareas que a continuación se describen.** El objetivo de esta guía no es explicar el código en detalle, sino describir los principales bloques de código, su implementación y el uso de determinadas funciones útiles disponibles en R para resolver cada tarea, por tanto, por sencillez para el lector no especialista, **el código aquí mostrado no está diseñado para maximizar su eficiencia sino para su fácil comprensión**. La forma de suministrar el código fuente completo es a través de un documento RMarkdown, que una vez cargado en el entorno de programación disponible en nuestra página de GitHub, el análisis puede reproducirse automáticamente ejecutando Knit en R o modificarse previamente si así se desea.

Para la realización del caso práctico hemos escogido un conjunto de datos relacionado con el **registro de la calidad del aire en la Comunidad Autónoma de Castilla y León** que se encuentra en el **portal de datos abiertos datos.gob.es**. Una vez cargadas las librerías y el conjunto de datos en el entorno de desarrollo procederemos a realizar el análisis exploratorio de datos siguiendo los pasos que a continuación se proponen.

2. Análisis exploratorio de datos

Para realizar esta guía hemos tomado como **referencia el análisis exploratorio de datos descrito en el libro** [R for Data Science](#) **de Wickman y Grolemund (2017)** disponible de forma gratuita y que además incluye una gran cantidad de ejemplos prácticos. El EDA que te proponemos seguirá los siguientes pasos:

1. **Realizar un análisis descriptivo de las variables**, para obtener una idea representativa del conjunto de datos.
2. **Re-ajustar los tipos de las variables** para que sean consistentes en el momento de realizar posteriores operaciones.
3. **Detección y tratamiento de datos ausentes**. El tratamiento o la eliminación de datos ausentes es esencial, ya que de otra manera no será posible procesar adecuadamente las variables numéricas.
4. **Identificación de datos atípicos y su tratamiento,** dado que pueden distorsionar futuros análisis estadísticos.
5. **Realizar un examen numérico y gráfico de las relaciones entre las variables analizadas para determinar el grado de correlación entre ellas**, pudiendo predecir el comportamiento de una variable en función de las otras.

El gráfico (Figura 1) siguiente representa de forma esquemática el conjunto de etapas del análisis exploratorio de datos que se relata en los contenidos de esta guía.

Fig1. Representación del conjunto de etapas del análisis exploratorio de datos.

Veamos a continuación de forma detallada cada una de las etapas propuestas para llevar a cabo un análisis exploratorio de datos. Cada capítulo incluye la sección **"Experimenta"** que, por medio de la aplicación práctica de diversas funciones en R, te ayudará a comprender los conceptos que se explican.

2.1. Análisis descriptivo

Una vez que se ha obtenido el dataset sobre el registro de la calidad del aire en la Comunidad Autónoma de Castilla y León del catálogo de datos abiertos y tenemos cargados los datos en nuestro entorno de desarrollo para posteriormente llevar a cabo alguna tarea de reutilización de los mismos como, por ejemplo, una visualización interactiva o el desarrollo de una aplicación, **es recomendable obtener una vista descriptiva sobre el contenido de las tablas de datos con las que vas a trabajar**. Con este fin **aplicaremos funciones de estadística descriptiva para explorar la estructura del conjunto de datos y examinar los datos y variables que presenta**. Asimismo, será muy útil **el uso de determinadas representaciones graficas que te ayudarán a intuir la forma que poseen las distribuciones de los datos**.

Experimenta

La tabla de datos cargada en nuestro entorno de desarrollo, con la que trabajaremos desde este momento y a la cual aplicaremos todos los procesos asociados al análisis exploratorio de datos propuestos en esta guía, se denomina "**calidad_aire**". Esta tabla contiene los datos de la calidad del aire de la Comunidad Autónoma de Castilla y León.

Para esta tarea usaremos funciones de R que nos mostrarán una visión general de la tabla de datos. Además de en esta etapa inicial, se utilizarán a lo largo del EDA con el fin de observar los cambios que progresivamente se irán efectuando sobre los datos a medida que vamos realizando las diferentes tareas de análisis exploratorio.

La función **view()** muestra el contenido de la tabla de datos que acabamos que cargar; la función **str()** permite conocer de forma compacta la estructura interna de la tabla de datos indicando el tipo de variables, los rangos de valores y una muestra de dichos valores donde de un vistazo se puede apreciar la presencia de valores ausentes identificados mediante las siglas NA (del inglés, *not allowed* o no disponible); **summary()** muestra un resumen general de las variables de la tabla, mostrando los valores: mínimo, máximo, media, mediana, primer y tercer cuartil para las variables numéricas, indicando además el número específico de valores NA presentes en cada una.

Veamos una muestra de la información descriptiva de la tabla que devuelve cada una de las funciones mencionadas:

```
view(calidad_aire)
```

	Fecha	CO (mg/m3)	NO (ug/m3)	NO2 (ug/m3)	O3 (ug/m3)	PM10 (ug/m3)	PM25 (ug/m3)	SO2 (ug/m3)	Provincia	Estación	Latitud	Longitud
1	01/01/1997	1.1	23	43	291	52	NA	6	Burgos	Burgos1	42.35083	-3.675556
2	01/01/1997	NA	NA	NA	58	NA	NA	14	Burgos	Burgos4	42.33611	-3.636111
3	01/01/1997	1.2	23	39	30	41	NA	16	Zamora	Zamora	41.50722	-5.738611
4	01/01/1997	3.2	180	79	7	NA	45	20	Salamanca	Salamanca3	40.96750	-5.668333
5	01/01/1997	1.2	43	54	29	124	NA	32	León	Leon1	42.60389	-5.587222
6	01/01/1997	0.9	14	26	38	37	NA	10	Valladolid	Medina del Campo	41.31639	-4.909167
7	01/01/1997	NA	18	29	20	NA	NA	13	Burgos	Miranda de Ebro3	42.68750	-2.954444
8	01/01/1997	NA	19	29	29	NA	NA	14	Burgos	Miranda de Ebro2	42.68806	-2.940556
9	01/01/1997	NA	31	44	9	36	NA	37	León	Leon2	42.58861	-5.571389
10	01/01/1997	NA	9	18	32	35	NA	40	Palencia	Guardo	42.79528	-4.840833
11	01/01/1997	2.6	68	70	NA	90	NA	14	Palencia	Palencia1	42.01000	-4.526944
12	01/01/1997	2.0	20	50	40	30	NA	27	Burgos	Burgos2	42.35111	-3.674444
13	01/01/1997	1.2	12	33	63	56	NA	19	Avila	Avila	40.65861	-4.688056

```
str(calidad_aire)
```

```
$ Fecha        : chr [1:446014] "01/01/1997" "01/01/1997" "01/01/1997" "01/01/1997" ...
$ CO (mg/m3)   : num [1:446014] 1.1 NA 1.2 3.2 1.2 0.9 NA NA NA NA ...
$ NO (ug/m3)   : num [1:446014] 23 NA 23 180 43 14 18 19 31 9 ...
$ NO2 (ug/m3)  : num [1:446014] 43 NA 39 79 54 26 29 29 44 18 ...
$ O3 (ug/m3)   : num [1:446014] 291 58 30 7 29 38 20 29 9 32 ...
$ PM10 (ug/m3): num [1:446014] 52 NA 41 NA 124 37 NA NA 36 35 ...
$ PM25 (ug/m3): num [1:446014] NA NA NA 45 NA NA NA NA NA NA ...
$ SO2 (ug/m3)  : num [1:446014] 6 14 16 20 32 10 13 14 37 40 ...
$ Provincia    : chr [1:446014] "Burgos" "Burgos" "Zamora" "Salamanca" ...
$ Estación     : chr [1:446014] "Burgos1" "Burgos4" "Zamora" "Salamanca3" ...
$ Latitud      : num [1:446014] 42.4 42.3 41.5 41 42.6 ...
$ Longitud     : num [1:446014] -3.68 -3.64 -5.74 -5.67 -5.59 ...
```

```
summary(calidad_aire)
```

```
    Fecha           CO (mg/m3)        NO (ug/m3)        NO2 (ug/m3)       O3 (ug/m3)        PM10 (ug/m3)
 Length:446014      Min.   : 0.0     Min.   :-441.00   Min.   :  0.00   Min.   :   0.00   Min.   :  0.00
 Class :character   1st Qu.: 0.3     1st Qu.:   2.00   1st Qu.:  8.00   1st Qu.:  37.00   1st Qu.: 11.00
 Mode  :character   Median : 0.7     Median :   5.00   Median : 16.00   Median :  54.00   Median : 18.00
                    Mean   : 0.9     Mean   :  13.23   Mean   : 21.41   Mean   :  53.05   Mean   : 22.69
                    3rd Qu.: 1.1     3rd Qu.:  15.00   3rd Qu.: 29.00   3rd Qu.:  68.00   3rd Qu.: 29.00
                    Max.   :25.1     Max.   : 634.00   Max.   :249.00   Max.   :4364.00   Max.   :557.00
                    NA's   :344856   NA's   :30984     NA's   :32517    NA's   :170600    NA's   :101435
  PM25 (ug/m3)      SO2 (ug/m3)       Provincia          Estación            Latitud        Longitud
 Min.   :  0.0     Min.   :-791.00   Length:446014      Length:446014      Min.   :38.94   Min.   :-6.782
 1st Qu.:  5.0     1st Qu.:   2.00   Class :character   Class :character   1st Qu.:41.65   1st Qu.:-6.484
 Median :  9.0     Median :   5.00   Mode  :character   Mode  :character   Median :42.54   Median :-4.909
 Mean   : 13.7     Mean   :   9.09                                         Mean   :42.15   Mean   :-5.179
 3rd Qu.: 15.0     3rd Qu.:  11.00                                         3rd Qu.:42.69   3rd Qu.:-4.538
 Max.   :223.0     Max.   : 364.00                                         Max.   :43.60   Max.   :-2.467
 NA's   :392230    NA's   :89739                                           NA's   :226     NA's   :226
```

Según los resultados obtenidos, las **principales características** que presenta la tabla de datos son:

- **El número total de observaciones del conjunto de datos es de 446.014**.
- **El número total de variables es 12**, tres de tipo cadena de caracteres y 9 numéricas.
- **El rango temporal abarca desde el 01/01/1997 hasta el 31/12/2020**.
- **Presenta dos variables del Sistema de Coordenadas Geográficas**: Longitud y Latitud, para georreferenciar cada una de las estaciones de calidad de aire de Castilla y León.

Es importante acompañar esta tarea con la representación de gráficos: histogramas, gráficos de líneas, barras o sectores, entre otros, para observar el comportamiento de la distribución de los datos. Una de las representaciones más útiles es la que se obtiene aplicando la función **hist()** que permite generar un histograma para observar la distribución de cualquier variable numérica presente en el conjunto de datos.

```
# Generamos los histogramas de dos variables numéricas que presenta la tabla de datos: 0³ (µg/m³) y PM10 ((µg/m³)

hist_O3 <- hist(calidad_aire$03, main ="",
                xlab = "O3 (ug/m3)",
                ylab = "Frecuencia",
                xlim = c(0, 150),
                breaks = 1000)

hist_PM10 <- hist(calidad_aire$PM10, main ="",
                  xlab = "PM10 (ug/m3)",
                  ylab = "Frecuencia",
                  xlim = c(0, 150),
                  breaks = 1000)
```

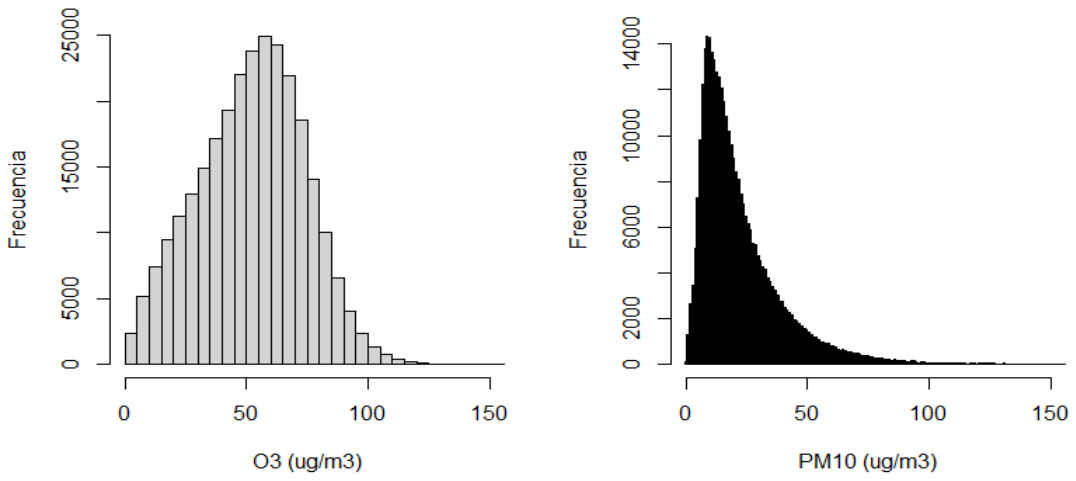

Fig1. Representación de los histogramas de dos variables donde se ha ajustado las etiquetas de cada eje, los límites inferior y superior del eje x y el número de barras (*breaks*) que se utilizan para representar la distribución.

Observando los histogramas de ambas variables, podemos concluir que presentan una distribución sesgada hacia la izquierda, con valores más cercanos al 0, aunque este sesgo es mucho más pronunciado en la variable **PM10.** Este resultado debe tenerse en cuenta en futuros análisis estadísticos, ya que las variables deben tratarse de manera distinta en función de la distribución que presenten sus datos.

Como hemos indicado anteriormente, **es aconsejable realizar un análisis descriptivo cada vez que se modifique la tabla de datos para comprobar de esta manera el efecto que producen sobre los datos los cambios aplicados en cada nueva etapa del EDA**.

2.2. Ajuste de los tipos de variables

Una de las primeras comprobaciones que hay que hacer tras cargar los datos en el entorno de trabajo, es **verificar que cada variable se ha almacenado con el tipo de valor que corresponde**. Por ejemplo, que las variables que contienen valores numéricos representan números y las cualitativas o categóricas están tipificadas como cadenas de caracteres y contienen una cantidad finita de elementos.

Los tipos de variables habituales, que puede albergar nuestra tabla de datos pueden ser:

- **numérico**, almacena números que pueden ser decimales o enteros.
- **carácter**: alberga cadenas de texto.
- **categórico**: contiene un número limitado de valores o categorías de información.
- **lógico o booleano**: variables binarias que solo pueden tomar dos valores: TRUE y FALSE ó 0 y 1; pueden ser resultado de una comparación o condición de otras variables presentes en el conjunto de datos.
- **fecha**: almacena intervalos específicos de tiempo.

Experimenta

Con la función **str()**, vista en el bloque anterior, podemos descubrir el tipo de cada una de las variables presentes en nuestro conjunto de datos, así como un ejemplo de los valores que toma.

En nuestro conjunto de datos, a priori, encontramos varias variables donde el tipo de dato no se corresponde con la naturaleza del valor que contiene, como es el caso de la variable **Fecha**, **Provincia** y **Estación**. Estas tres variables han sido codificadas como tipo carácter, sin embargo:

- la variable **Fecha**, debería ser de tipo fecha ya que almacena la fecha en la que se recogieron los datos para el resto de variables y de esta forma será posible aplicar determinadas funciones predefinidas en R para trabajar con datos de este tipo, por ejemplo, obtener la diferencia en días entre dos periodos de tiempo. Para transformar una cadena de caracteres a una variable fecha usaremos la función **as.Date()**, asegurando previamente si la secuencia correcta de día (d), mes (m) y año (y) o se utiliza otro formato internacional de fecha.
- las variables **Provincia** y **Estación**, son variables de tipo categórico. Para transformar una cadena de caracteres a una variable tipo categórico, usaremos la función **as.factor()**, conociendo previamente la lista de valores que puede tomar la variable. Para conocer la lista de valores que toma la variable y comprobar si finalmente esa variable toma un número finito de valores usaremos la función **unique()**, que devuelve un vector con los valores que presenta la variable sin duplicados.

El procedimiento a seguir es re-ajustar los tipos de estas variables para poder realizar posteriormente las operaciones, análisis y representaciones gráficas que sean necesarias.

```
# Ajustamos el tipo de la variable Fecha

calidad_aire$Fecha <- as.Date(calidad_aire$Fecha, format("%d/%m/%Y"))

# Ajustamos el tipo de la variable Provincia y Estación

unique(calidad_aire$Provincia)
calidad_aire$Provincia <- as.factor(calidad_aire$Provincia)

unique(calidad_aire$Estación)
calidad_aire$Estación <- as.factor(calidad_aire$Estación)
```

2.3. Detección y tratamiento de datos ausentes

a. Detección de datos ausentes

La presencia de datos ausentes, perdidos, *missing values*, o celdas vacías, representados habitualmente en R como NA, **es una problemática habitual en muchos conjuntos datos**. La mayoría de las veces se debe a fallos en la transcripción de los datos o problemas durante la recogida de datos, por ejemplo, debido a la imposibilidad para obtener cierta medida u observación.

Tratar con conjuntos de datos en los que existen datos ausentes puede generar problemas a la hora de aplicar diferentes análisis estadísticos o en la generación de representaciones gráficas. A fin de evitar problemas futuros, es necesario aprender a detectar y aplicar algún tipo de tratamiento.

Experimenta

Con la función **is.na()** comprobamos la existencia de valores ausentes. El resultado de esta función es un vector lógico (TRUE o FALSE), de tal forma que cuando el objeto (por ejemplo, un valor, una lista o la columna de una tabla) que se evalúa presenta un valor ausente devuelve TRUE, en caso contrario devuelve FALSE. Existen otras funciones muy útiles, que también se pueden usar para evaluar la existencia de NAs: la función **any(is.na())** devuelve TRUE si la tabla presenta al menos un valor ausente, sin indicar el número de valores perdidos que presenta la tabla, ni la posición; la función **sum(is.na())**, permite determinar el número de valores ausentes; **mean(is.na())**, muestra el porcentaje de valores perdidos que presenta la tabla con la cual estamos trabajando.

```
# Devuelve un vector lógico
is.na(calidad_aire)

# Devuelve un único valor lógico, cierto o falso, si existe algún valor ausente

any(is.na(calidad_aire))

# Devuelve el número de NAs que presenta la tabla

sum(is.na(calidad_aire))

# Devuelve el % de valores perdidos
mean(is.na(calidad_aire))
```

En ocasiones conviene realizar una detección de valores ausentes por columnas, en lugar de por filas, para identificar si alguna de las variables del dataset presenta un determinado nivel de datos perdidos. Para ello podemos utilizar la función **colMeans(is.na())** y la función **colSums(is.na())**.

```
# Detección del número de valores perdidos en cada una de las columnas que presenta la tabla

colSums(is.na(calidad_aire))

# Detección del % de valores perdidos en cada una de las columnas que presenta la tabla

colMeans(is.na(calidad_aire), round(2))
```

Analizando el conjunto de datos con el cual estamos trabajando en esta guía, podemos observar que la tabla "**calidad del aire**" presenta un total de 116.281 valores perdidos, el 21% del total. Si analizamos los NAs, en cada una de las variables, dos de ellas: **CO (mg/m^3)** y **PM25 (µg/m^3)**, presenta un porcentaje superior al 50%, del 77% y 88% respectivamente, lo que conlleva una ausencia significativa de información en el dataset de trabajo. **Esta anomalía debe ser tratada de alguna forma para disminuir su impacto en el objetivo de reutilización de los datos**.

b. Tratamiento de datos ausentes

Existen varias **maneras de tratar con valores ausentes**:

- Rellenar los valores con la media, mediana o el valor más frecuente de la variable.
- Completar los valores que faltan por el valor que esté directamente antes o después en la fila o columna.
- Completar todos los datos faltantes con 0, si se trata de valores numéricos. Esta opción es poco aconsejable ya que puedes modificar de manera significativa los resultados.
- Eliminar las filas que presenten valores ausentes, siempre y cuando el conjunto de datos sea lo suficientemente grande y no se pierde información relevante al eliminar esas filas.
- Y una forma abrupta de tratamiento que depende del contexto de análisis, es eliminar las variables que presentan un porcentaje mayor del 50% de datos ausentes.

Seleccionar la mejor manera de abordar el tratamiento de valores ausentes, depende del tipo de dato, del tratamiento posterior de los mismos o de la causa de la falta de esos valores (si se conoce). La estrategia más común es utilizar el valor medio, pero, que sea la opción más popular, no significa necesariamente que sea la elección correcta para el conjunto de datos con el cual estemos tratando en ese momento.

Los tratamientos de datos ausentes mencionados pueden modificar los resultados obtenidos en futuros análisis, disminuir el tamaño muestral o introducir un sesgo, por lo que, un diseño riguroso del EDA implica documentar esta decisión con el objetivo de mantener trazabilidad de los procesos llevados a cabo y poder en todo momento, retornar a uno de estos puntos de decisión ante una determinada inconsistencia o debilidad en etapas posteriores del análisis de datos.

Experimenta

Como ejemplo de aplicación de las opciones enumeradas, el primer tratamiento que vamos a realizar sobre los datos perdidos, es la eliminación de las dos variables que presentan un porcentaje superior al 50 %, ya que un número de NAs tan alto puede producir errores o distorsionar los análisis posteriores al no ser usadas las filas que presentan NAs (en este caso, no se usaría más del 50% de las observaciones).

```
# Eliminación de las variables que presentan un % de NAs superior al
50%, para ello se utiliza la función which() que permite realizar
selecciones de datos bajo alguna premisa.

calidad_aire <- calidad_aire
                    [,-which(colMeans(is.na(calidad_aire)) >= 0.50)]
```

Continuando con el ejemplo, los valores perdidos que presenta la tabla en el resto de variables, los sustituiremos por la media de cada una de las columnas, para no perder información significativa y los análisis posteriores no se vean alterados.

```
# Seleccionamos las variables numéricas que presenta la tabla iterando
sobre todas las columnas de la tabla mediante la función sapply()

columnas_numericas <- which(sapply(calidad_aire, is.numeric))

# Calculamos la media para cada una de las variables numéricas sin tener
en cuenta los NAs
cols_mean[columnas_numericas] <-
  colMeans(calidad_aire[, columnas_numericas], na.rm = TRUE)

# Sustituimos los valores NA por la media correspondiente a cada
variable

for (x in columnas_numericas) {
  calidad_aire[is.na(calidad_aire[,x]), x] <- round(cols_mean[x],2)
                                }
```

Las operaciones realizadas ilustran las posibilidades para el tratamiento de datos ausentes sin entrar a valorar la oportunidad de su aplicación respecto a un análisis riguroso de la calidad del aire dado que en ese caso influyen otros factores meteorológicos que en este ejemplo no se están teniendo en cuenta.

2.4. Detección y tratamiento de valores atípicos (outliers)

Un valor atípico u *outlier*, es una observación significativamente distinta del resto de datos que presenta una variable, de tal magnitud que se puede considerar un valor anómalo. **Estos valores pueden afectar a tareas siguientes pudiendo llegar a modificar los resultados**. Es necesario detectarlos y tratarlos para poder disminuir su

influencia en los análisis posteriores o, en casos muy extremos, eliminarlos del conjunto de datos.

Lo más recomendable para el tratamiento de los datos atípicos es reducir su posible influencia en los análisis. Aunque no es objeto de esta guía, hay que mencionar que existen métodos estadísticos robustos aplicables en los análisis, que permiten disminuir el impacto de los outliers. Estos métodos logran que los resultados se vean menos afectados por la presencia de valores atípicos.

Descartar los datos atípicos del conjunto de datos sin verificar que no se deben a un error de medición o de construcción del dataset, no es la solución. Por otro lado, **sustituir estos datos por la media o la mediana, tampoco es recomendable**. Estos dos tratamientos, al igual que los tratamientos aplicados a los datos ausentes, pueden modificar los resultados obtenidos en futuros análisis, disminuir el tamaño muestral, introducir un sesgo o puede afectar tanto a la distribución como a las varianzas de la variable de interés. Si finalmente se decide eliminar o sustituir los valores atípicos, **es muy recomendable repetir los análisis con y sin valores inusuales**, para observar el efecto que ocasionan. Si el efecto es mínimo, es razonable eliminarlos o sustituirlos. Si el efecto es sustancial, no deberían ser ignorados sin justificación.

Como se ha indicado anteriormente, independientemente de la opción considerada, es importante documentar cada decisión adoptada con el objetivo de que otros analistas de datos comprendan las posibles transformaciones efectuadas sobre el conjunto de datos en cada etapa del EDA llevada a cabo.

Manteniendo el objetivo didáctico de esta guía, mostramos a continuación como se eliminan los datos atípicos por si os encontráis en la tesitura de que podéis afirmar que los valores son errores de medición o derivados de la ingesta de datos y, por tanto, susceptibles de ser eliminarlos del conjunto de datos para que no produzcan distorsiones en futuros análisis estadísticos.

Experimenta

Para mostrar el proceso, debemos distinguir dos tipos de tratamiento en función del tipo de variables, continuas o discretas y categóricas.

a- Variables continuas

Detección de valores atípicos

Para mostrar el proceso de detección de valores atípicos en una variable continua, utilizaremos como ejemplo la variable numérica **O3.** El proceso es exactamente igual para el resto de variables numéricas que presente la tabla.

En primer lugar, **generamos un histograma para conocer la distribución de frecuencias que presenta la variable de estudio**:

```
histograma_O3 <- hist(calidad_aire$03,
                      main =””,
                      xlab = “O3 (ug/m3)”,
                      ylab = “Frecuencia”,
                      xlim = c(0, 150),
                      breaks = 1000)
```

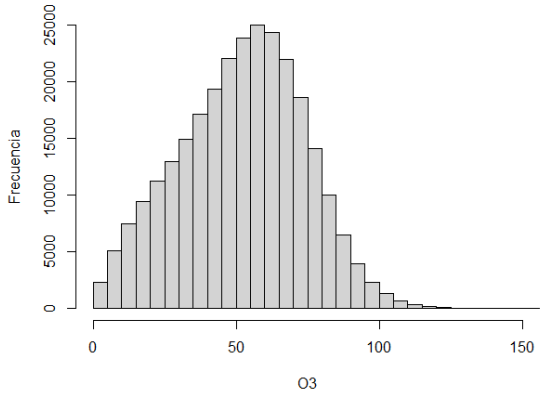

Fig2. Representación del histograma de la variable “O3 (µg/m3)”.

Como podemos observar en el histograma, los niveles de **O_3** en el aire están mayoritariamente en un rango aproximado de, entre 0 y 100 µg/m^3. Por encima de este valor la frecuencia es mínima, y podrían considerarse que es el rango de concentración de los valores atípicos. Para detectar esos valores atípicos, utilizaremos la representación más adecuada para esta tarea: un gráfico de cajas y bigotes.

Los gráficos de cajas y bigotes o *boxplots* (Fig. 2), **aportan una representación visual que describe la dispersión y simetría** que presentan los datos observando los cuartiles (división de la distribución en cuatro partes delimitadas por los valores 0,25; 0,50 y 0,75). Estos gráficos están compuestos por tres componentes:

- **Caja de rango intercuartílico** (*interquartile range* o IQR): Representa el 50% de los datos, comprende desde el percentil 25 de la distribución (Q1), hasta el percentil 75 (Q3). Dentro de la caja encontramos una línea que señala el percentil 50 de la distribución (Q2), la mediana. La caja aporta una idea sobre la dispersión de la distribución en función de la separación existente entre Q1 y Q3, así como también si la distribución es simétrica en torno a la mediana o si esta sesgada hacia alguno de los lados.
- **Bigotes**: Se extienden desde ambos lados de los extremos de la caja y representan los rangos del 25% de valores de la parte inferior (Q1 – 1,5 IQR) y el 25% de valores de la parte superior (Q3 + 1,5 IQR), excluyendo los valores atípicos.

- **Valores atípicos**: esta representación identifica como valores atípicos aquellas observaciones que presentan valores inferiores o superiores a los límites del gráfico (límite inferior: Q1 – 1,5 IQR y límite superior: Q3 + 1,5 IQR).

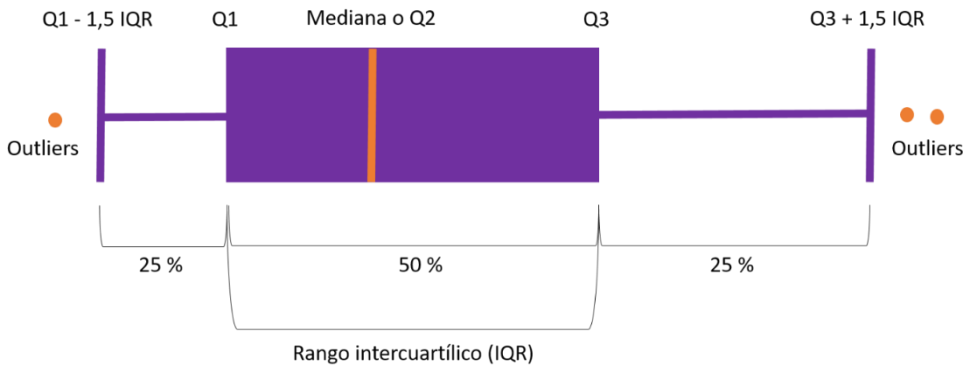

Fig3. Representación de un gráfico boxplot para identificar la presencia de valores atípicos.

Para la obtención de los estadísticos necesarios para la representación del gráfico, recurriremos a la función **boxplots.stats()** y para la representación del gráfico utilizaremos la función **boxplot()**.

```
# Estadísticas necesarias para reproducir el gráfico de cajas y bigotes
boxplot.stats(calidad_aire$ NO2)
# Construcción del gráfico de cajas y bigotes
boxplot(calidad_aire$O3, horizontal = TRUE, xlab = "O3 (ug/m3)")
```

La función boxplots.stats(), nos muestra los estadísticos necesarios para la representación del gráfico. A continuación, mostraremos el detalle de los 4 resultados que devuelve esta función:

- **$stats**: valores estadísticos que definen el gráfico:
  - Q1 – 1,5IQR = 0
  - Q1 = 37
  - Mediana o Q2 = 54
  - Q3 = 68
  - Q3 + 1,5IQR = 114
- **$n**: número de observaciones que presenta la variable, en este caso 275414.
- **$out**: lista de outliers que presenta el dataset para esa variable, en este caso 468.

Usando la función boxplot() hemos construido el gráfico de cajas y bigotes. En este gráfico, podemos observar que la variable representada presenta una alta cantidad de datos atípicos por encima del límite superior del gráfico (Q3 + 1,5 IQR).

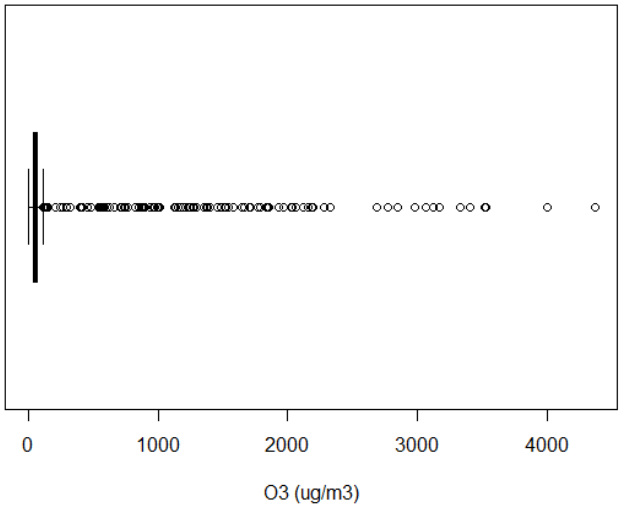

Fig4. Representación de un gráfico boxplot en el que se observa la presencia de valores atípicos y su distribución.

Los outliers detectados en esta variable coinciden con los valores más elevados de esta partícula en el aire, lo cual indica que, o bien son datos anómalos o realmente son valores extremadamente elevados de esta partícula detectados en días concretos. Un análisis exhaustivo de la calidad del aire debe contribuir a discernir esta duda, comparando también cuál ha sido el comportamiento de otras partículas en dichos días.

Eliminación de valores atípicos

Una forma de eliminar los valores atípicos de una variable numérica es generar una nueva tabla, en la cual eliminemos los valores identificados como atípicos.

```
# Se genera una nueva tabla que no contiene los valores almacenados en
el vector outliers$out, antes obtenido con la función boxplot.stats().

calidad_aire_NoOut <- calidad_aire[!(calidad_aire$O3 %in%
                                     outliers$out),]
```

Una vez que hemos eliminado los *outliers*, volvemos a realizar el gráfico de cajas y bigotes con esta nueva tabla para comparar la nueva distribución de valores con la anterior. Podemos observar que tanto la mediana como los cuartiles han cambiado, así como la ausencia de valores anómalos.

```
# Construcción de los gráficos de cajas y bigotes

boxplot(calidad_aire$`NO2 (ug/m3)`, xlab = "NO2 (ug/m3)")
boxplot(calidad_aire_NoOut$`NO2 (ug/m3)`, xlab = "NO2 (ug/m3)")
```

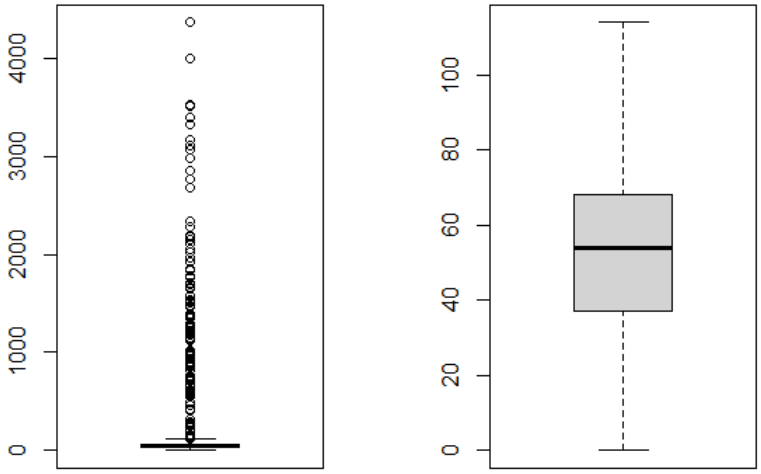

Fig5. Representación de dos gráficos boxplot en los que se observa la presencia y ausencia de valores atípicos antes y después de su eliminación.

b. Variables categóricas

Detección de valores atípicos

Al igual que en el caso anterior, **para detectar si existen valores atípicos en una variable categórica, debemos conocer su distribución**. Para ello lo más adecuado es observar su histograma representado mediante un gráfico de barras. Para ilustrar el caso, utilizaremos la variable **Provincia**, aunque el proceso es el mismo para cualquier variable categórica.

Para la realización de este gráfico R base dispone de funciones básicas como la función **hist()**, pero a su vez R dispone de múltiples librerías para realizar sofisticadas representaciones gráficas. Entre ellas, destaca el paquete **ggplot2**, usado para realizar la visualización de los datos que se muestra a continuación, que es uno de los más potentes en R y que incluimos en esta guía para ilustrar su uso en este ejemplo.

```
# Número de categorías que presenta la variable Provincia

count(calidad_aire, "Provincia")

# Construcción del gráfico de barras para la variable Provincia

ggplot_provincias <- ggplot(calidad_aire)+

                     geom_bar(aes(x = Provincia, fill = Provincia)) +
                     xlab("Provincias") + ylab("Nº observaciones") +
                     theme(axis.text.x = element_text(angle = 30))
ggplot_provincias
```

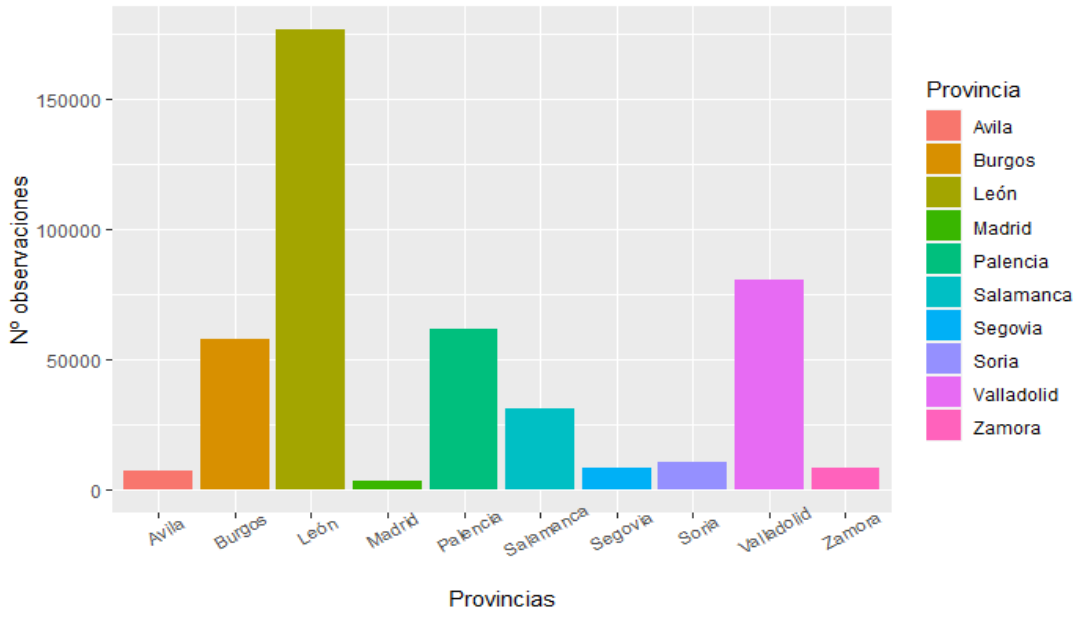

Fig6. Representación del histograma de la variable provincia utilizando la librería ggplot que ofrece una considerable riqueza gráfica.

Como se puede observar en la imagen, la variable Provincia presenta 10 factores o categorías, uno de los factores puede generar dudas, es la presencia del factor llamado "**Madrid**" con muy pocas observaciones, y que evidentemente no pertenece a la Comunidad Autónoma de Castilla y León. Esta categoría podría tratarse de un error o nos podría estar indicando la presencia de una estación de calidad del aire fuera de la Comunidad Autónoma de Castilla y León, algo que se podría comprobar al disponer de las coordenadas geográficas de las estaciones en el conjunto de datos. A priori se desconoce, ya que en la descripción del conjunto de datos no se especifica y en consecuencia procede realizar esta verificación para evitar un error a la hora de realizar nuestros análisis.

Eliminación de valores atípicos

Por la información de contexto derivada del análisis del dataset se puede deducir que la categoría de información detectada para esta variable se trata de un outlier y por tanto se procede a su eliminación.

Para eliminar los valores atípicos de una variable categórica, debemos eliminar la categoría que consideramos que no se ajusta a nuestros datos. En este caso, eliminaremos la categoría Madrid, de la variable categórica Provincia utilizando la función **droplevels()**.

```
# Eliminamos las filas que pertenecen al factor "Madrid"
eliminar_Madrid <- calidad_aire$Provincia %in% c("Madrid")
calidad_aire_SM <- calidad_aire[¡eliminar_Madrid,]

# Eliminamos el factor "Madrid"
calidad_aire_SM$Provincia <- droplevels(calidad_aire$Provincia)

# Con la función levels() verificamos la eliminación de la categoría
"Madrid" de la variable

levels(calidad_aire_SM$Provincia)
```

2.5. Análisis de correlación entre variables

La correlación (valor r, en el gráfico siguiente), **determina la relación lineal entre dos o más variables**, es decir, la fuerza y la dirección de una posible relación entre variables. Dicho de otra forma, si los valores de una variable tienden a subir, los de otra u otras variables, harán lo mismo si están correladas positivamente o a la inversa, si lo están negativamente. Esto no quiere decir, que una correlación entre variables indique una relación causa-efecto. De hecho, puedes encontrar cientos de ejemplos de correlaciones ficticias con las que puedes pasar un rato bien divertido. ¿En qué nos puede ayudar el análisis de correlación entre variables? La existencia de una relación fuerte en un determinado sentido entre dos variables podría inferir redundancia de información, pudiendo llegar a la eliminación de una de ellas con el fin de disminuir la complejidad en el procesamiento y análisis futuro de los datos. Esta práctica es habitual en EDA y está vinculada con la técnica de análisis de componentes principales (en muchos escritos lo verás cómo análisis PCA, por sus siglas en inglés). Sin entrar a definir esta técnica, la correlación se mide a través del **coeficiente de correlación "r" que oscila entre -1 y 1**. La correlación positiva perfecta se establece con el valor +1 e indica

que los valores de las variables varían de una forma similar y la correlación negativa perfecta se establece con el valor –1, indicando que varían de forma inversa. No existe relación entre las variables, es decir, son independientes, cuando el coeficiente es 0. A continuación, se muestra una imagen que representa diferentes niveles de correlación entre variables mediante un gráfico de dispersión.

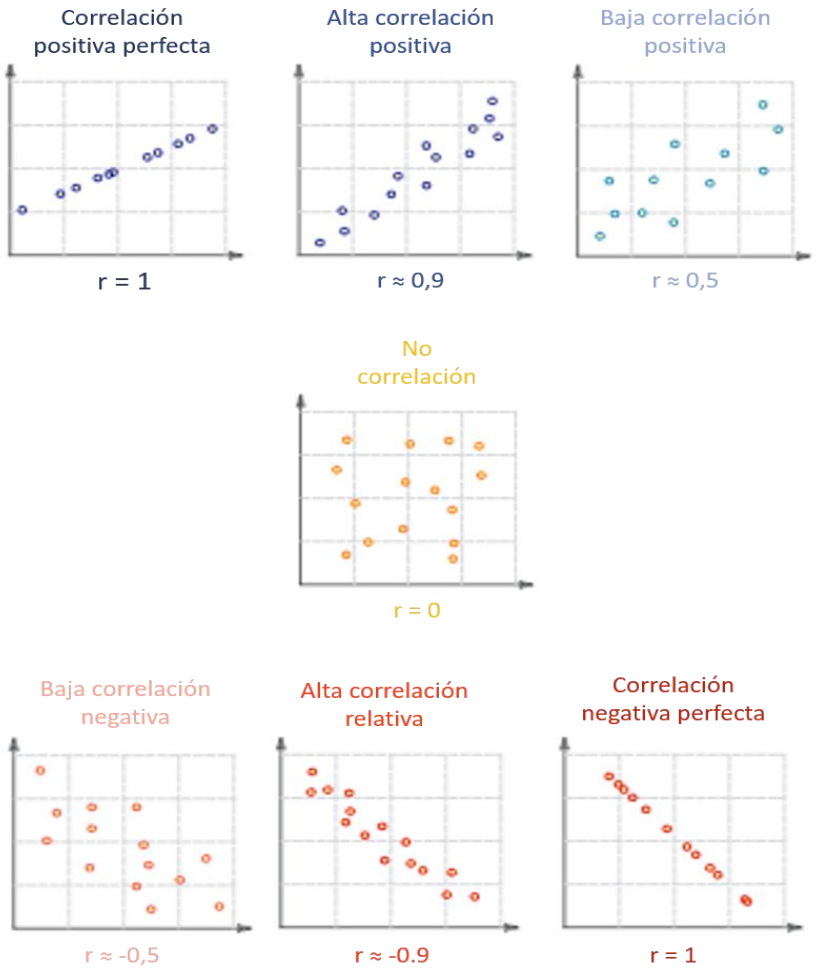

Fig3. Interpretación de la correlación lineal entre dos variables: si la correlación es positiva, cuando crece el valor de una de las variables también lo hace el valor de la variable relacionada y si es negativa, cuando crece el valor de una disminuye en la otra.

Experimenta

En primer lugar, debemos calcular la matriz de coeficientes de correlación (la fortaleza de la relación) para las variables numéricas, a partir de la cual consideramos si existe una relación entre ellas. Una vez que tengamos calculado la matriz de correlaciones, lo más habitual es mostrarlo gráficamente.

```
# Seleccionamos las variables numéricas situadas en las columnas 2 a 6
de la tabla de calidad del aire (NO, NO2, O3, PM10, PM25 y SO2)

num_variables <- calidad_aire[,c(2,3,4,5,6)]

#Calculamos la matriz de coeficientes de correlación entre las variables
numéricas

correlacion <- cor(num_variables)

#Gráfico de correlaciones indicando la forma en la que se representa la
correlación (un cuadrado que varía en tamaño según la fortaleza). Para
la generación de este gráfico es necesario instalar y cargar la librería
corrplot

corrplot(correlacion, method = "square")
```

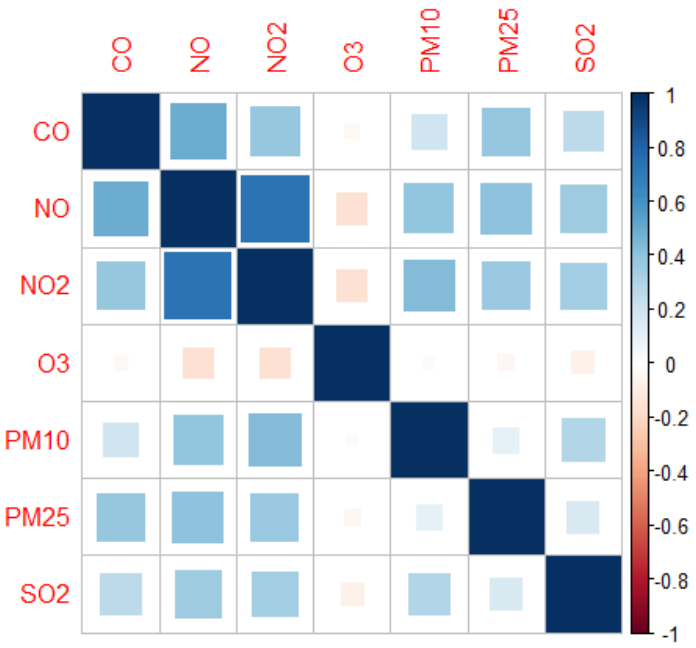

Fig3. Representación gráfica de correlaciones entre variables donde se observa la fortaleza de cada correlación bivariada mediante el tamaño de un cuadrado.

Existen múltiples formas de representar gráficamente la correlación entre variables en R. Hemos elegido una de las más sencillas que utiliza como recurso gráfico para representar la fortaleza y dirección de la correlación, formas y colores. En el gráfico

generado observamos la correlación entre las variables observadas. Cuando más cercano a 0 es el coeficiente de correlación r, los cuadrados son más pequeños y presentan un color más cercano al blanco. Observando el gráfico a simple vista, podemos hacer varias consideraciones al respecto:

- Las variables NO (ug/m3) y NO2 (ug/m3) presenta un coeficiente alto de **correlación positiva** (concretamente r = 0,73, según la matriz de correlación). Si fuese el caso, que no lo es porque en este ejemplo queremos analizar todas las partículas que conforman la calidad del aire, este resultado nos indica que podríamos prescindir de una de estas variables ante la necesidad de reducir el número de variables en un análisis posterior.
- En el resto de variables, el coeficiente de correlación es más bajo. Tal como se observa en el gráfico, los cuadros que representan la correlación son más pequeños y presenta un color más cercano al blanco. A simple vista, podríamos inferir que las variables O3 (ug/m3) y PM10 (ug/m3) se pueden considerar **variables independientes**.

Somos conscientes de que hemos hecho una introducción muy ligera del concepto de correlación, pero el objetivo de esta guía es explicarte algunas tareas relevantes que se deben llevar a cabo cuando realizas un EDA. En Internet podrás encontrar mucha información sobre el tratamiento de la correlación y la regresión lineal utilizando el lenguaje R.

3. Conclusiones

El Análisis Exploratorio de Datos o EDA es el conjunto de técnicas estadísticas cuyo fin es explorar los datos de forma preliminar a la aplicación de cualquier proceso posterior como una investigación científica o una visualización interactiva de datos. Se trata de un **proceso fundamental para el entendimiento básico de los datos y las relaciones que existen entre ellos**. Como hemos visto, a través de métodos sencillos, el EDA permite, entre otras acciones, organizar y preparar los datos, detectar fallos en el diseño y recogida de los mismos, el tratamiento y evaluación de los datos ausentes, la identificación de los casos atípicos y la posible relación que puedan existir entre las variables. **Es verdaderamente trascendental dedicar tiempo a aplicar estos métodos para que los resultados obtenidos a partir de los análisis estadísticos aplicados a esos datos sean altamente fiables y muestren la realidad de los mismos**.

En esta guía hemos realizado una introducción asequible, para todos los públicos, sobre los pasos más significativos a seguir para llevar a cabo este proceso, ilustrando su aplicación mediante un ejemplo con datos reales procedentes del catálogo de datos abiertos datos.gob.es. Los lectores podrán reproducir este caso práctico e incluso intentarlo con otros conjuntos de datos siguiendo los mismos pasos. Esperamos que os resulte útil esta nueva guía y seguiremos generando contenidos de interés relacionados con el mundo de los datos abiertos. ¡Hasta pronto!

4. Próxima parada

Si quieres seguir profundizando en el apasionante mundo del análisis exploratorio de los datos, te sugerimos los recursos que mencionamos a continuación:

- Algunos **libros disponibles gratuitamente**, que detallan el proceso del análisis exploratorio de los datos y suelen incluir conjuntos de datos de prueba y ejemplos con código (R o Python) para ilustrar el proceso:
    - Exploratory Data Analysis with R
    - R for Data Science
    - Exploratoty Data Analysis and Visualization
    - Exploratory Data Analysis with Python
    - Python for Data Analysis

- Además de libros, sin lugar a dudas, la mejor forma de aprender ciencia de datos es practicando. A continuación, os dejaremos unos enlaces a **tutoriales y cursos on-line** con una importante carga de programación práctica:
    - Python EDA: NLP process explanation
    - Comprehensive data exploration with Python
    - Visual data exploration
    - Exploratory Data Analysis (EDA) in Google Sheets
    - Tutorial: Análisis de datos de exploración con Python y Pandas
    - Exploratory Data Analysis in R
    - Exploratory Data Analysis with Seaborn

- Por último, os dejamos algunos **recursos adicionales** muy útiles que de una forma gráfica compilan la información más relevante:
    - Cheat sheet – 11 Steps for Data Exploration in R (with codes)
    - Cheat sheet - Estadística descriptiva
    - Cheat sheet – Dates and Time with Lubridate
    - Cheat sheet – Factors with forcats

¿QUIERES SABER MÁS SOBRE LA INICIATIVA APORTA?

Visita www.datos.gob.es
Twitter: @datosgob
Linkedin: datos.gob.es
Suscríbete a nuestro boletín
Escribe a contacto@datos.Gob.es

Puedes identificar los espacios de **datos abiertos** gracias a este logo

datos abiertos