Análisis exploratorio de datos (EDA)

Figure 1: Artwork por @allison_horst

Introducción

El análisis exploratorio de datos (conocido como **EDA**, su sigla en inglés) es un enfoque de análisis de datos para resumir y visualizar las características importantes de un conjunto de datos.

John Tukey, estadístico estadounidense, fue el principal propulsor contribuyendo de manera significativa al desarrollo del análisis exploratorio de datos al publicar, en 1977, su libro que lleva ese nombre donde entre otras cosas introdujo el *gráfico boxplot* (diagrama de caja y bigotes).

En términos simples, antes de avanzar con la etapa analítica y de construir modelos estadísticos, es relevante explorar, conocer y describir las variables de interés en nuestra tabla de datos.

Los principales objetivos perseguidos por EDA son:

- Conocer la estructura de la tabla de datos y sus tipos de variable
- Detectar observaciones incompletas (valores missing)
- Conocer la distribución de las variables de interés a partir de:
  - Resumir datos mediante estadísticos
  - Resumir datos mediante gráficos
- Detectar valores atípicos (outlier)

Aclaración: En este documento mostraremos funciones del lenguaje R que se pueden aplicar en este proceso basadas en la filosofía tidyverse. También aplicaremos otros paquetes diseñados para tareas específicas que le serán de mucha utilidad. Esto no quiere decir que no se pueda hacer la misma exploración con funciones del R `base` pero el ecosistema facilita el entendimiento de lo que estamos haciendo.

Presentaremos estas diferentes funciones de distintos paquetes que pueden servir en cada etapa de un EDA. Los paquetes con los que trabajaremos son:

- tidyverse
- skimr
- dlookr
- janitor

Para instalarlos puede copiar y ejecutar el siguiente código:

```
install.packages(c("tidyverse", "skimr", "janitor", "dlookr"))
```

Nota: Algunos paquetes, entre estos `dlookr`, pueden ocasionar un falso positivo en la detección del antivirus durante el proceso de instalación. Sugerimos que desactive momentáneamente su antivirus para instalarlo sin inconvenientes.

Una vez instalados los podemos activar:

```
library(skimr)
library(janitor)
library(dlookr)
library(tidyverse)
```

Cabe aclarar que no existe un solo camino y/o función del lenguaje para obtener la información requerida y que esta selección de paquetes puede cambiarse y ampliarse según la conveniencia del usuario. Es decir, aquellxs estudiantes que ya utilicen R y estén familiarizados con funciones y/o paquetes que realicen la misma tarea pueden seguir usándolos.

Con el fin de ejemplificar este análisis exploratorio vamos a utilizar un archivo con datos ficticios y variables de distinto tipo.

Conocer la estructura de la tabla de datos y sus tipos de variable

El primer paso en la exploración de un conjunto de datos es conocer su estructura y tamaño.

El tamaño está definido por la cantidad de observaciones (filas) y la cantidad de variables (columnas).

Llamamos estructura a la forma en se organizan sus variables, sus tipos de datos y sus categorías/valores.

```
datos <- read_csv2("datos/datos2.txt")
```

La función `glimpse()` del `tidyverse` le da un vistazo a los datos:

```
glimpse(datos)
```

```
Rows: 74
Columns: 7
$ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,…
$ sexo    <chr> "M", "M", "M", "M", "M", "M", "M", "M", NA, "F", "F", "M", "F"…
$ edad    <dbl> 76, 68, 50, 49, 51, 68, 70, 64, 60, 57, 83, 76, 27, 34, 17, 45…
$ peso    <dbl> 71, 71, 79, 71, 87, 75, 80, 83, 69, 73, 60, 70, 648, 718, 61, …
$ talla   <dbl> 167, 164, 164, 164, 1675, 170, 166, 160, 160, 155, 155, 167, 1…
$ trabaja <lgl> FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, NA, TRUE, TRUE, TRUE, …
$ fecha   <date> 2020-10-20, 2020-10-20, 2020-10-20, 2020-11-05, 2020-11-05, 2…
```

Nos informa que la tabla tiene 74 observaciones, 7 variables con su tipo de dato y los primeros valores de cada una al lado.

Los tipos de datos que nos podemos encontrar son:

- **int** (integer): números enteros
- **dbl** (double): números reales
- **lgl** (logical): valores lógicos
- **chr** (character): caracteres (texto)
- **Date**: fechas
- **fct** (factor): factores
- **dttm** (date-time): fechas y horas

Esta exploración inicial de la estructura generalmente viene acompañada por el "diccionario de datos" asociado a la tabla de datos, ya sea que esta tabla provenga de un proyecto de investigación propio (fuente primaria) o producto de una fuente secundaria.

En algunas situaciones el tipo de dato del dato coincidirá con la clasificación de la variable (por ejemplo, que sea numérica -dbl- para variables cuantitativas continuas) pero en otros casos podemos tener variables codificadas donde el dato es numérico pero representa una categoría de una variable cualitativa (por ejemplo, si a una variable de respuesta Si - No, la codificamos como 1 y 0).

Detectar observaciones incompletas (valores missing)

Sabemos que los valores perdidos o faltantes (conocidos en inglés como missing), que se gestionan en R mediante el valor especial reservado `NA`, constituyen un serio problema en nuestras variables de análisis.

Existen numerosos libros sobre como tratarlos y sobre diversos algoritmos de imputación que no vamos a incluir en este curso.

Sólo vamos a enfocarnos en como podemos utilizar algunas funciones de R para detectarlos, contabilizarlos y en algunas situaciones excluirlos.

Cada vez que ejecutemos un `count()` a una variable nos informará, al final de la tabla de salida, la cantidad de valores `NA`.

```
datos |>
  count(trabaja)
```

```
# A tibble: 3 × 2
  trabaja     n
  <lgl>   <int>
1 FALSE      26
2 TRUE       39
3 NA          9
```

Mucho mejor es la función `find_na()` que proviene del paquete `dlookr`:

```
find_na(datos, rate = T)
```

```
     id    sexo    edad    peso   talla trabaja   fecha
  0.000   4.054   0.000   0.000   0.000  12.162   0.000
```