Análisis Exploratorio de Datos 1

Rodrigo Zepeda (rodrigo.zepeda@itam.mx)

16 de junio 2020

Librerías

Para este análisis vamos a tener que llamar a las siguientes librerías previamente instaladas (por única vez) con `install.packages`:

```
library(tidyverse)
library(dplyr)
library(moments)
library(lubridate)
```

Si no tienes una librería puedes instalarla escribiendo en la consola el `install` junto con su nombre:

```
install.packages("lubridate")
```

Base a analizar

Como ejemplo analizaremos la base de *Carpetas de Investigación de la Fiscalía General de Justicia* de la CDMX para el año 2018 y mes de Diciembre misma que se encuentra en este link

> Si el link anterior no abre ve al sitio https://datos.cdmx.gob.mx/explore/dataset/carpetas-de-investigacion-pgj-cdmx/table/?refine.ao_hechos=2018 y elige la opción de año 2018, mes diciembre y descargar como csv.

La forma más fácil en RStudio es yéndonos a `Import Dataset` en el panel derecho seguido de `From Text` y seleccionamos el archivo. En este caso hay dos opciones cualquiera de las dos opciones funciona: si en tu ordenador no sirve una, ¡prueba la otra!

En mi caso el archivo está en el escritorio y se lee de la siguiente manera:

```
datos <- read.csv("~/Desktop/carpetas-de-investigacion-pgj-cdmx.csv")
```

Definiciones y notación

Denotamos el conjunto de datos observados como la matriz (*base de datos*) de $\ell \times n$

$$Z = \left(z_1 \middle| z_2 \middle| \ldots \middle| z_\ell\right)$$

donde $\ell \in \mathbb{N}$ con $\ell > 0$ y las z_i sin pérdida de generalidad, son vectores columna ($z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,n})^T$) de longitud n. Una columna z_k con $0 \leq k \leq \ell$ se le conoce como:

- *Numérica* si $z_k \in \mathbb{R}^n$. En particular es entera si $z_j \in \mathbb{Z}^n$.
- *Categórica* si cada entrada de z_k es una indicadora de pertenencia a algún conjunto (por ejemplo `Hombre / Mujer` ó `Ingresos Altos / Ingresos Medios / Ingresos Bajos`). Usualmente z_k se representa con un caracter o con un entero.
- *Lógica* si z_k es un indicador que toma alguno de los dos valores: TRUE ó FALSE.

- *Caracter* si z_k es un caracter o una cadena de caracteres donde los caracteres son el objeto de análisis en sí (no como pertenencia). Por ejemplo si cada entrada $z_{k,m}$ representa un Tweet.

OJO Los datos $z_{k,m}$ son variables fijas ya dadas y **NO SON ALEATORIAS**.

En el caso de nuestra base de datos podemos resumir la información contenida en la misma mediante `glimpse`:

```
datos %>% glimpse()
```

```
## Rows: 19,861
## Columns: 18
## $ año_hechos           <int> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018...
## $ mes_hechos           <chr> "Diciembre", "Diciembre", "Diciembre", "Diciem...
## $ fecha_hechos         <chr> "2018-12-13 12:00:00", "2018-12-22 19:00:00", ...
## $ delito               <chr> "USURPACIÓN DE IDENTIDAD", "SUSTRACCION DE MEN...
## $ categoria_delito     <chr> "DELITO DE BAJO IMPACTO", "DELITO DE BAJO IMPA...
## $ fiscalía             <chr> "INVESTIGACIÓN EN MIGUEL HIDALGO", "INVESTIGAC...
## $ agencia              <chr> "MH-2", "59", "BJ-1", "IZP-9", "75TER", "FDS-5...
## $ unidad_investigacion <chr> "UI-1SD", "UI-1CD", "UI-1SD", "UI-2SD", "3 S/D...
## $ colonia_hechos       <chr> "LOMAS DE SOTELO", NA, "DEL VALLE CENTRO", "AM...
## $ alcaldia_hechos      <chr> "MIGUEL HIDALGO", "CUAUTLA", "BENITO JUAREZ", ...
## $ fecha_inicio         <chr> "2019-06-16 12:14:09", "2019-06-06 16:26:15", ...
## $ mes_inicio           <chr> "Junio", "Junio", "Febrero", "Febrero", "Abril...
## $ ao_inicio            <int> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019...
## $ calle_hechos         <chr> "AV. CONSCRIPTO", "AVENIDFA DIEZ DE MARZO", "F...
## $ calle_hechos2        <chr> ".", "HECHOS EN CUAUTLA MORELOS", "ESQUINA COY...
## $ longitud             <dbl> -99.22535, NA, -99.17088, -99.03016, -99.13423...
## $ latitud              <dbl> 19.44028, NA, 19.37207, 19.34797, 19.54788, 19...
## $ Geopoint             <chr> "19.4402832543,-99.2253527208", "", "19.372068...
```

Notamos que el vector columna `año_hechos` es una variable numérica mientras que `mes_hechos` es categórica. No hay variables lógicas en esta base. Una variable caracter es el vector columna `calle_hechos` que no denota un conjunto sino una cadena de caracteres (véanse las faltas de ortografía, por ejemplo).

Al ser la tabla de datos una matriz podemos acceder a la entrada en la fila j y columna k haciendo:

$$\text{base}[j,k]$$

por ejemplo:

```
datos[4,6]
```

```
## [1] "INVESTIGACIÓN EN IZTAPALAPA"
```

> **NOTACIÓN** Para facilitar la notación en lo que sigue de estas notas y hasta nuevo aviso, si z_k es una columna categórica de Z denotaremos a los elementos de dicha columna como $C = (c_1, c_2, \ldots, c_n) = z_k^T$. Si z_k es numérica denotamos a los elementos de dicha columna como $\vec{x} = (x_1, x_2, \ldots, x_n) = z_k^T$.

Estadísticos univariados

Definición [Estadístico]

Un estadístico es una función cuyo dominio es la matriz de datos observados Z o una columna de la misma. Es decir, un estadístico es cualquier función de los datos. A continuación veremos algunos ejemplos de estadísticos así como su interpretación.

1. **Media poblacional** Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos la media

poblacional como:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad x_i \in \mathbb{R}$$

En el caso de nuestros datos podemos **calcular el promedio de delitos por día** como sigue. Primero necesitamos especificar a `R` que la `fecha_hechos` es una fecha. Esto lo hacemos mediante la función `ymd_hms` (`year-month-day_hour-minute-second`) del paquete de `lubridate` y la función `mutate` (que cambia una columna de la base de datos). El siguiente código le indica a `R` que cambie la columna `fecha_hechos` volviéndola a leer como fecha:

```
datos <- datos %>% mutate(fecha_hechos = ymd_hms(fecha_hechos))
```

Para mantener sólo la fecha y eliminar la hora de `fecha_hechos` podemos generar una nueva columna como sigue:

```
datos <- datos %>% mutate(fecha = date(fecha_hechos))
```

Finalmente podemos contar (`tally`) observaciones agrupadas (`group_by`) por día mediante la combinación de ambas funciones:

```
conteo_delitos <- datos %>% group_by(fecha) %>% tally()
```

```
## # A tibble: 6 x 2
##   fecha          n
##   <date>     <int>
## 1 2018-12-01   674
## 2 2018-12-02   584
## 3 2018-12-03   790
## 4 2018-12-04   640
## 5 2018-12-05   724
## 6 2018-12-06   718
```

Hay distintas formas de calcular la media. La primera es tomando la columna directo, para acceder a una columna utilizamos el signo de pesos $ como sigue:

```
base$columna
```

En nuestro caso:

```
mean(conteo_delitos$n)
```

```
## [1] 640.6774
```

O bien podemos usar la función `summarise` integrada en `dplyr`:

```
conteo_delitos %>% summarise(mean(n))
```

```
## # A tibble: 1 x 1
##   `mean(n)`
##       <dbl>
## 1      641.
```

2. Total poblacional Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos el total poblacional como:

$$t_{\vec{x}} = \sum_{i=1}^{N} x_i, \qquad x_i \in \mathbb{R}$$

En este caso de las carpetas de investigación el total nos daría todas las carpetas abiertas durante diciembre. Para ello calculamos el total sumando todos los elementos:

```r
sum(conteo_delitos$n)
```

```
## [1] 19861
```

O bien (y esto es una de las cosas interesantes de `tidyverse`) agregándolo a los cálculos previos:

```r
conteo_delitos %>% summarise(mean(n), sum(n))
```

```
## # A tibble: 1 x 2
##   `mean(n)` `sum(n)`
##       <dbl>    <int>
## 1      641.    19861
```

3. Varianza poblacional (no ajustada) Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos la varianza poblacional como:

$$\sigma^2_{\vec{x}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2, \qquad x_i \in \mathbb{R}$$

Misma que podemos calcular con el comando `var` ya sea directamente en la columna:

```r
var(conteo_delitos$n)
```

```
## [1] 10046.23
```

O bien a través del `summarise` integrando con el anterior:

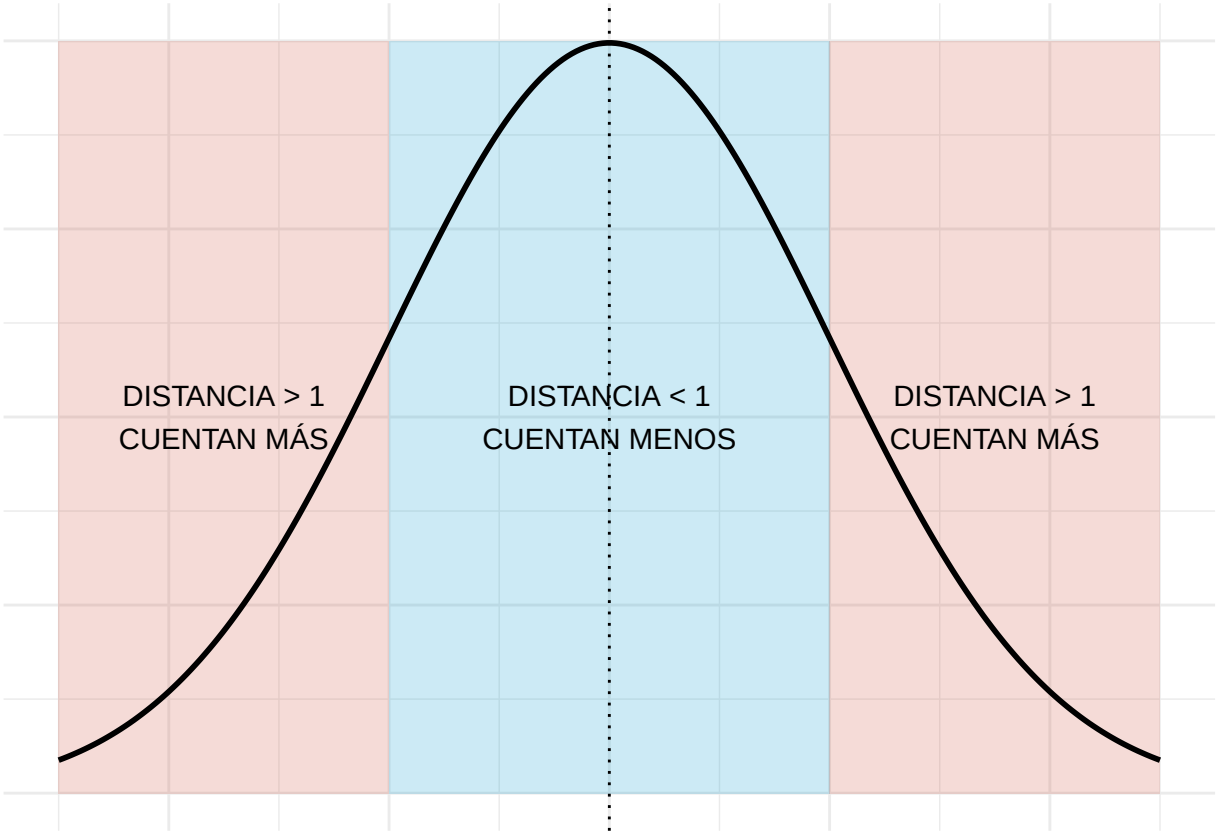
```r
conteo_delitos %>% summarise(mean(n), sum(n), var(n))
```

```
## # A tibble: 1 x 3
##   `mean(n)` `sum(n)` `var(n)`
##       <dbl>    <int>    <dbl>
## 1      641.    19861   10046.
```

La raíz cuadrada de la varianza se conoce como **desviación estándar** y se denota como sigue:

$$\sigma_{\vec{x}} = \sqrt{\sigma^2_{\vec{x}}}$$

Recuerda que la varianza se interpreta como la distancia cuadrática promedio a la que están los datos. En particular la varianza casi no considera valores que están a menos de 1 de distancia de $\bar{x}$ (pues $(x_i - \bar{x})^2 < 1$ en ese caso) pero le da mayor peso a valores que están muy lejanos (donde $(x_i - \bar{x})^2 \gg 1$ si x_i está muy lejos de $\bar{x}$). Gráficamente:



Si nos interesara que todos los valores (tanto los cercanos a $\bar{x}$ como los lejanos) pesaran de manera idéntica entonces usaríamos el MAD:

4. Desviación Media Absoluta (MAD) Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos la desviación media absoluta, MAD, como:

$$\text{MAD}_{\vec{x}} = \frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}|$$

Misma que se puede calcular en R como:

```
mad(conteo_delitos$n)
```

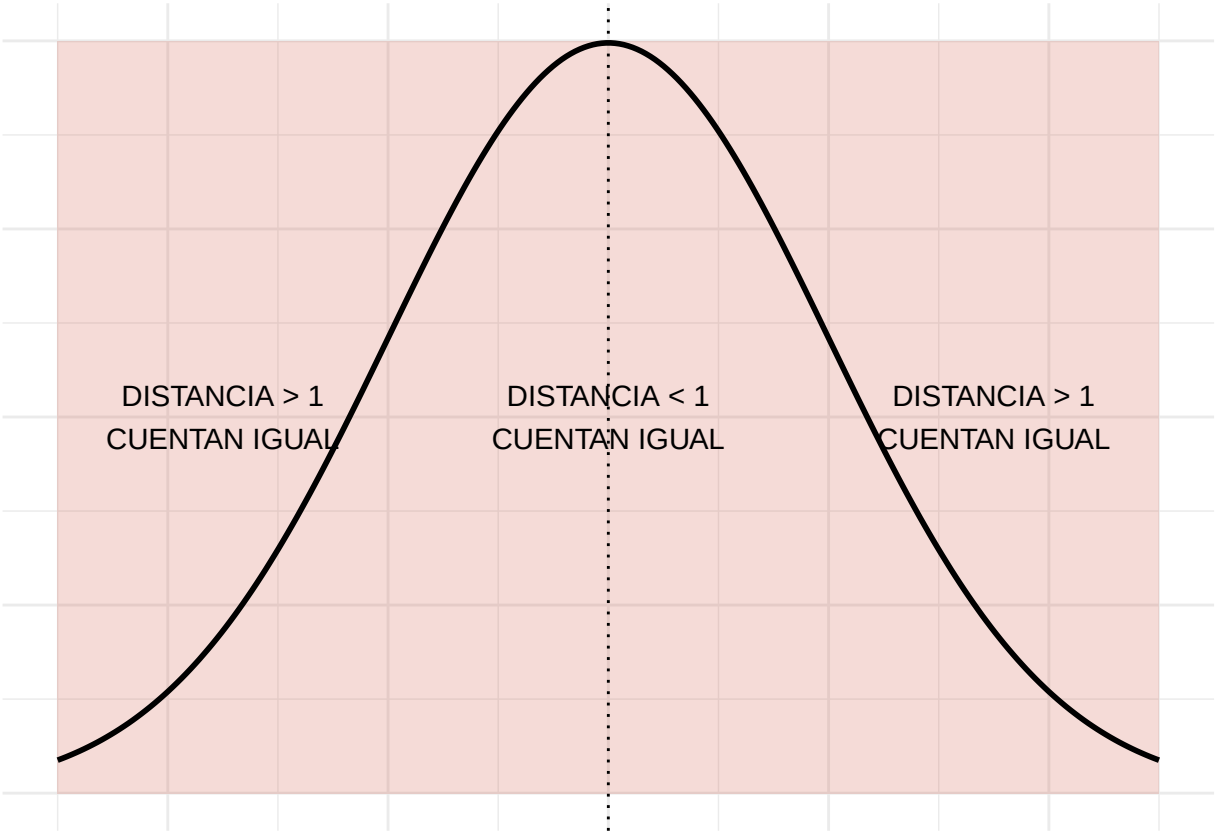
```
## [1] 115.6428
```

o bien dentro del `summarise`:

```
conteo_delitos %>% summarise(mean(n), sum(n), var(n), mad(n))
```

```
## # A tibble: 1 x 4
##   `mean(n)` `sum(n)` `var(n)` `mad(n)`
##       <dbl>    <int>    <dbl>    <dbl>
## 1      641.    19861   10046.     116.
```

La MAD también es una forma de medir distancia pero en este caso se tiene que todos aportan por igual los muy alejados y los que no:



Para pensarle: En el caso de una variable que se supone que es uniforme y no interesa penalizar valores lejanos de la media ¿cuál sería una mejor manera de cuantificar la dispersión MAD ó varianza? ¿en qué casos importaría la otra?

Las siguientes dos definiciones son con base en conceptos de proba. ¿Los recuerdas?

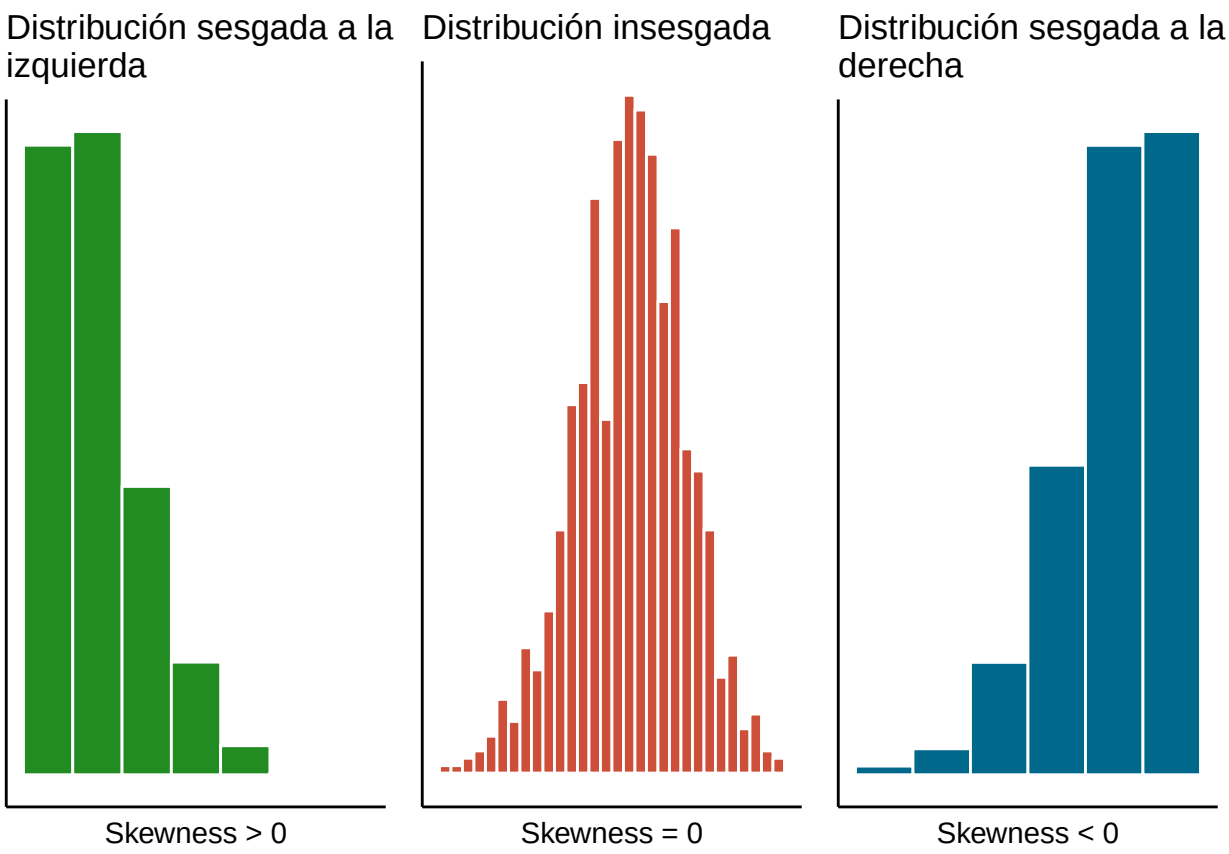
5. Coeficiente de asimetría Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos el coeficiente de asimetría de Pearson (*skewness*) como:

$$\text{Skewness}_{\vec{x}} = \frac{1}{n\sigma_X^3} \sum_{i=1}^{n} (x_i - \bar{x})^3$$

Para interpretar el coeficiente de asimetría podemos dividir esa suma en dos pedazos (olvidándonos de la constante):

$$\sum_{i=1}^{n} (x_i - \bar{x})^3 = \underbrace{\sum_{i=1}^{n} (x_i - \bar{x})^3}_{x_i > \bar{x}} + \underbrace{\sum_{i=1}^{n} (x_i - \bar{x})^3}_{x_i < \bar{x}}$$
$$\text{A} \qquad\qquad \text{B}$$

Notamos que si $|A| > |B|$ la mayor parte de las x_i (o las que se alejan más de la media) son mayores a $\bar{x}$ y por tanto los datos van a estar *sesgados a la derecha*:. Por otro lado si $|B| > |A|$ significa que hay más x_i (o con mayor peso) del lado izquierdo de la media que del lado derecho de la misma y por tanto los datos están *sesgados a la izquierda*. Datos *insesgados* son aquellos donde $\text{Skewness}_{\vec{x}} = 0$.



En el caso de las carpetas podemos calcular la asimetría que no se encuentra preprogramada en R como sigue:

```
#Estimación de la desviación estándar
desv.est <- sd(conteo_delitos$n)

#Estimación del x barra
x.barra <- mean(conteo_delitos$n)

#Obtención de la n (longitud del vector)
n.longitud <- length(conteo_delitos$n)

#Cálculo de la asimetría
(1/desv.est^3)*mean((conteo_delitos$n - x.barra)^3)
```

```
## [1] -0.4528209
```

¿Qué implica el resultado anterior?

6. Curtosis Dado el mismo vector X que en el enunciado anterior el coeficiente de curtosis se define como

$$\text{Curtosis}_{\vec{x}} = \frac{1}{n\sigma_X^4}\sum_{i=1}^{n}(x_i - \bar{x})^4$$

La interpretación de la curtosis es similar a la que hizimos de la varianza en el sentido que el elevar a la cuarta va a magnificar los efectos de aquellos valores que estén a más de σ de distancia de la media pues podemos reescribir la suma como:

$$\frac{1}{n\sigma_X^4}\sum_{i=1}^{n}(x_i-\bar{x})^4 = \frac{1}{n\sigma_X^4}\underbrace{\sum_{i=1}^{n}(x_i-\bar{x})^4}_{|x_i-\bar{x}|<\sigma \text{ A}} + \frac{1}{n\sigma_X^4}\underbrace{\sum_{i=1}^{n}(x_i-\bar{x})^3}_{|x_i-\bar{x}|>\sigma \text{ B}}$$

Notamos que la única parte importante que apota a la curtosis es la dada por **B** que es la que capta las *colas* de la distribución. De ahí que podamos decir que, entre dos vectores de datos, uno tiene colas más pesadas que el otro si su curtosis es mayor. En este caso podemos analizar la `latitud` y `longitud` de los datos a través de la curtosis:

```
datos %>% summarise(kurtosis(latitud, na.rm = T), kurtosis(longitud, na.rm = T))
```

```
##   kurtosis(latitud, na.rm = T) kurtosis(longitud, na.rm = T)
## 1                     2.857934                      3.045037
```

donde se agregó el comando `na.rm = T` para eliminar los valores de no respuesta (missing) marcados como `NA`. Del análisis notamos que la longitud tiene colas más pesadas que la latitud.

> **NOTACIÓN** Dado un vector $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ de valores numéricos denotamos el j-ésimo valor muestral $(1 \leq j \leq n)$ como $x_{(j)}$ tal que $x_{(1)} = \min\{x_1, x_2, \ldots, x_n\}$ y
>
> $$x_{(j)} = \min\{x_1, x_2, \ldots, x_n\} \setminus \{x_{(1)}, x_{(2)}, \ldots, x_{(j-1)}\}$$
>
> Es decir $x_{(j)}$ es el valor en orden j al momento de ordenar la muestra. Como nota adicional se define $x_{(0)} = 0$ y $x_{(n+1)} = 0$.

7. Mediana Dado un vector de valores numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ denotamos la mediana como:

$$\text{Mediana}_{\vec{x}} = \frac{x_{(\lfloor \frac{n}{2} \rfloor)} + x_{(\lfloor \frac{n}{2}+1 \rfloor)}}{2}$$

La mediana puede calcularse fácilmente haciendo:

```
median(conteo_delitos$n)
```

```
## [1] 646
```

8. Cuantil Dado un vector de valores numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)$ el α-ésimo cuantil está dado por:

$$\text{Cuantil}_{\vec{x}}(\alpha) = \frac{x_{(\lfloor \alpha n \rfloor)} + x_{(\lfloor \alpha n+1 \rfloor)}}{2}$$

donde $x_{(0)} = x_{(n+1)} = 0$. `R` no calcula los cuantiles de manera exacta sino que por velocidad los aproxima mediante la función `quantile`. Por ejemplo en el cálculo de los cuantiles $\alpha = 0.1$ y $\alpha = 0.66$:

```
conteo_delitos %>% summarise(quantile(n, c(0.1, 0.66)))
```

```
## # A tibble: 2 x 1
##   `quantile(n, c(0.1, 0.66))`
##                         <dbl>
## 1                         501
## 2                         707
```

La función `summary` también es bastante útil resumiendo múltiples observaciones de la base:

```
summary(conteo_delitos$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   397.0   568.0   646.0   640.7   721.0   790.0
```

Ésta incluye los **cuartiles** los cuales corresponden a los cuantiles asociados a $\alpha = 0.25, 0.5, 0.75$ y 1.

9. Rango intercuartílico Definimos el rango intercuartílico para valores numèricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ como la distancia entre el cuantil 0.75 y el 0.25 (primer y tercer cuartil):

$$\text{IQR}_{\vec{x}} = \text{Cuantil}_{\vec{x}}(0.75) - \text{Cuantil}_{\vec{x}}(0.25)$$

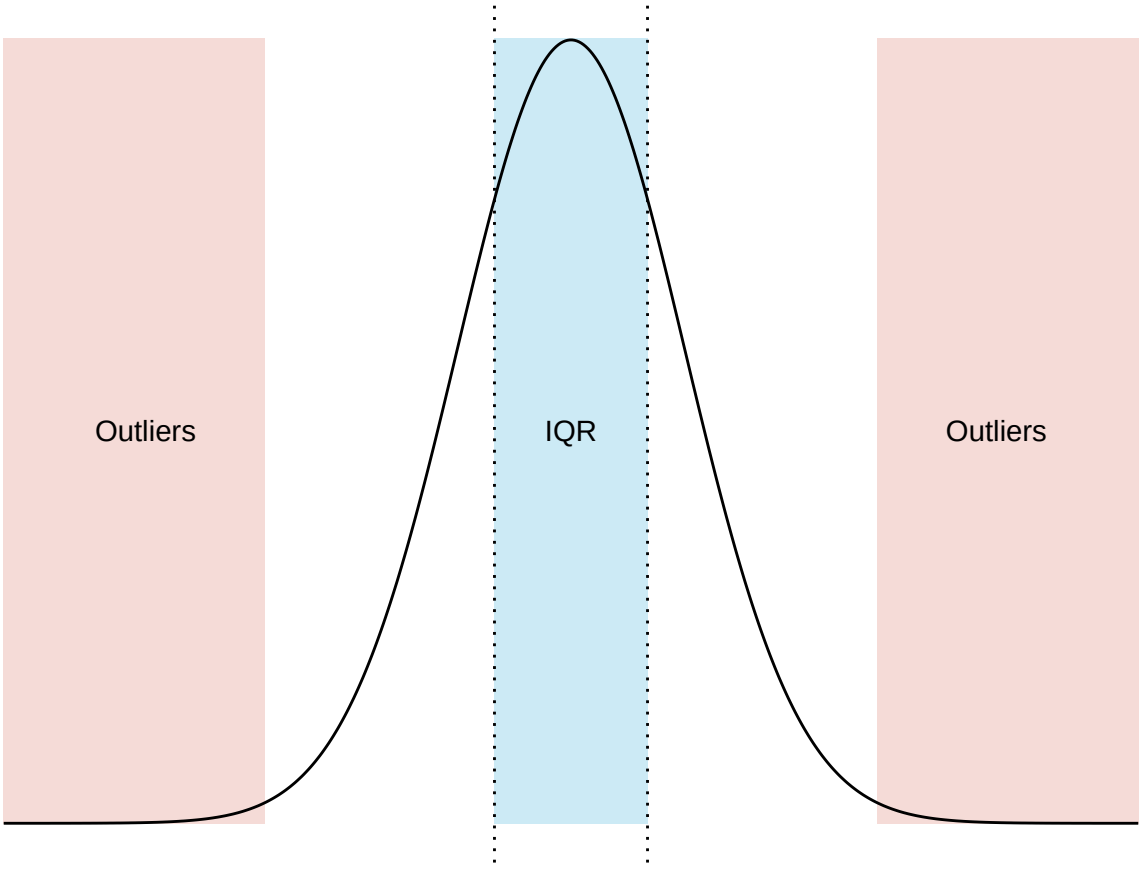
```
IQR(conteo_delitos$n)
```

```
## [1] 153
```

10. Valores atípicos (outliers) Dado un vector de datos numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ definimos los valores atípicos *outliers* como aquellas observaciones:

$$\text{Outliers}_{\vec{x}} = \left\{ x_i \in \vec{x} \middle| x_i \notin \left[\text{Cuantil}_{\vec{x}}(0.25) - \frac{3}{2}\text{IQR}_{\vec{x}}, \text{Cuantil}_{\vec{x}}(0.75) + \frac{3}{2}\text{IQR}_{\vec{x}} \right] \right\}$$

Los *outliers* en esta defínción son valores que serían verdaderamente improbables bajo una distribución normal.



Particularmente en el caso de la normal los *outliers* son valores que tienen una probabilidad de salir aproximadamente de 0.0069766 (por eso son atípicos porque no se esperaría que aparecieran nunca).

Para identificar los *outliers* calculamos el IQR primero y los cuartiles:

```
iqr      <- IQR(conteo_delitos$n)
cuartil1 <- quantile(conteo_delitos$n, 0.25)
cuartil3 <- quantile(conteo_delitos$n, 0.75)
```

después identificamos el límite inferior y superior del conjunto

```
lim.inf  <- cuartil1 - 3/2*iqr
lim.sup  <- cuartil3 + 3/2*iqr
```

finalmente preguntamos por cuáles están antes o después:

```
outliers <- conteo_delitos %>% filter(n < lim.inf | n > lim.sup)
```

En este caso no tenemos *outliers*.

11. Rango El rango se define como la diferencia entre el mínimo y el máximo de los valores de un vector numérico $\vec{x} = (x_1, x_2, \ldots, x_n)^T$:

$$\text{Rango}_{\vec{x}} = \max\{x_1, x_2, \ldots, x_n\} - \min\{x_1, x_2, \ldots, x_n\}$$

En R puede calcularse con la resta:

```
#Obtenemos máximo y mínimo
maximo <- max(conteo_delitos$n)
minimo <- min(conteo_delitos$n)

#Rango
maximo - minimo
```

```
## [1] 393
```

> **Nota** En algunos casos el rango se refiere al intervalo $[a, b]$ de valores donde $a = \min\{x_1, x_2, \ldots, x_n\}$ y $b = \max\{x_1, x_2, \ldots, x_n\}$. Éste es el caso de la función `range` en R:

```
range(conteo_delitos$n)
```

```
## [1] 397 790
```

12. Conteo asociado a un conjunto Sea $\vec{y} = (y_1, y_2, \ldots, y_n)^T$ un vector de datos de cualquier tipo (numéricos, categóricos, lógicos, caracteres, etc). Para un conjunto A definimos el conteo asociado al conjunto A como:

$$\text{Conteo}_{\vec{y}}(A) = \sum_{i=1}^{n} \mathbb{I}_A(y_i)$$

donde

$$\mathbb{I}_A(y) = \begin{cases} 1 & \text{si } y \in A, \\ 0 & \text{en otro caso }, \end{cases}$$

es una variable indicadora.

Una forma rápida de obtener dicho conteo en R es mediante `table`:

```
table(datos$delito)
```

```
##
## ABANDONO DE PERSONA              ABORTO  ABUSO DE AUTORIDAD  ABUSO DE CONFIANZA
##                  53                  15                 102                 276
##        ABUSO SEXUAL        ACOSO SEXUAL
##                 252                  30
```

O bien si se desean contar en la base de datos por ejemplo los delitos de `ABANDONO DE PERSONA` pueden hacerse mediante un filtro.

```
datos %>% filter(delito == "ABANDONO DE PERSONA") %>% tally()
```

```
##    n
## 1 53
```

Al filtro pueden agregársel grupos por si se desea obtener por fecha:

```
datos %>% filter(delito == "ABANDONO DE PERSONA") %>%
  group_by(fecha) %>%  tally()
```

```
## # A tibble: 21 x 2
##    fecha          n
##    <date>     <int>
##  1 2018-12-01     3
##  2 2018-12-02     3
##  3 2018-12-04     2
##  4 2018-12-05     8
##  5 2018-12-06     1
##  6 2018-12-07     1
##  7 2018-12-10     1
##  8 2018-12-12     2
##  9 2018-12-13     3
## 10 2018-12-14     2
## # ... with 11 more rows
```

El filtro funciona igual que un `if` pudiéndose usar `and` (&) u `or` (|):

```
datos %>%
  filter(delito == "ABANDONO DE PERSONA" |  delito == "ABORTO") %>%
  group_by(fecha) %>%  tally()
```

```
## # A tibble: 25 x 2
##    fecha          n
##    <date>     <int>
##  1 2018-12-01     3
##  2 2018-12-02     3
##  3 2018-12-04     5
##  4 2018-12-05     8
##  5 2018-12-06     1
##  6 2018-12-07     2
##  7 2018-12-10     1
##  8 2018-12-12     2
##  9 2018-12-13     4
## 10 2018-12-14     2
## # ... with 15 more rows
```

```
datos %>%
  filter(delito == "ABANDONO DE PERSONA" &
           fiscalía == "INVESTIGACIÓN EN IZTAPALAPA") %>%
  group_by(fecha) %>%  tally()
```

```
## # A tibble: 3 x 2
##   fecha          n
##   <date>     <int>
## 1 2018-12-02     1
## 2 2018-12-13     1
## 3 2018-12-20     1
```

13. Moda En términos simples, la moda es el conjunto de los valores que más se repiten. Matemáticamente

la moda es el conjunto $\text{Moda}_{\vec{y}} = \{m_1, m_2, \ldots, m_k\}$ tal que $m \in$ Moda sí y sólo si

$$\sum_{i=1}^{n} \mathbb{I}_{\{m\}}(y_i) \geq \sum_{i=1}^{n} \mathbb{I}_{\{\ell\}}(y_i) \qquad \forall \ell \neq m \text{ donde } y_i \in \vec{y}.$$

Para calcularla en `R` no existe una función predefinida para calcular la mode. Nosotros podemos crearla con el comando `function`. El término `function` nos sirve para construir funciones; por ejemplo, una función que eleva al cuadrado:

```
elevar.cuadrado <- function(x){
  return(x^2)
}
```

Observa la estructura que siempre será de esta forma:

$$\begin{aligned} &\text{Nombre de la función} \leftarrow \texttt{function}(\text{argumento 1}, \text{argumento 2}, \ldots, \text{argumento n})\{ \\ &\quad \#\text{Hacer algunos cálculos usando los argumentos} \\ &\quad \texttt{return}(\text{output}) \\ &\} \end{aligned} \tag{1}$$

Podemos llamar a la función con un número:

```
elevar.cuadrado(8)
```

```
## [1] 64
```

o bien con un vector:

```
elevar.cuadrado(12)
```

```
## [1] 144
```

En nuestro caso vamos a crear una función que se llame `moda` para estimar la moda:

```
#Función para estimar la moda de un vector x
moda <- function(x){

  #Contar cuántas veces aparecen las observaciones
  conteo <- table(x)

  #Obtengo el máximo que aparece
  max_aparece <- max(conteo)

  #Busco cuáles aparecen más y obtengo los nombres
  moda <- names(conteo)[which(conteo == max_aparece)]

  #Finalmente checo que si los datos eran numéricos moda debe
  #ser numérico
  if (is.numeric(x)){
    moda <- as.numeric(moda)
  }

  return(moda)
}
```

Podemos probar nuestra función con datos que ya sepamos su resultado nada más para asegurarnos que funciona:

```r
#Creamos un vector numérico con dos modas
vector.ejemplo.1 <- c(1,6,6,1,2,7,8,10)
moda(vector.ejemplo.1)
```

```
## [1] 1 6
```

Podemos probarlo también con caracteres:

```r
#Creamos un vector numérico con dos modas
vector.ejemplo.2 <- c("manzana","pera","guayaba","perejil","manzana")
moda(vector.ejemplo.2)
```

```
## [1] "manzana"
```

Una vez sabemos funciona podemos buscar el delito que ocurrió más:

```r
moda(datos$delito)
```

```
## [1] "VIOLENCIA FAMILIAR"
```

Ejercicios

1. Construye una función que tome de input dos variables: x un vector y k un entero de tal manera que calcule el **k-ésimo momento cenrtal** de los datos:

$$\text{Momento}_{\vec{x}}(k) = \frac{1}{n\sigma_{\vec{x}}^k}\sum_{i=1}^{n} x_i^k$$

   La función debe tener la siguiente estructura:

```r
kesimo.momento <- function(x, k){
  #Rellena aquí
}
```

2. Sin usar la opción de `trim` ni `trimmed.mean` crea una función que calcule la media de los datos que están entre el cuantil $\alpha/2$ y el cuantil $1-\alpha/2$ ($0 \leq \alpha \leq 1$. A esta media se le conoce como **media truncada al nivel** $\alpha \times 100\%$. Matemáticamente se define como:

$$\text{Media Truncada}_{\vec{x}}(\alpha) = \frac{1}{n_\alpha}\sum_{i=1}^{n} x_i \cdot \mathbb{I}_{[q_{\alpha/2}, q_{1-\alpha/2}]}(x_i)$$

   donde $n_\alpha = \sum_{i=1}^{n} \mathbb{I}_{[q_{\alpha/2}, q_{1-\alpha/2}]}(x_i)$ es la cantidad de x_i que están en el intervalo $[q_{\alpha/2}, q_{1-\alpha/2}]$ donde $q_{\alpha/2} = \text{Cuantil}_{\vec{x}}(\alpha/2)$ y $q_{1-\alpha/2} = \text{Cuantil}_{\vec{x}}(1-\alpha/2)$.

3. Una función llamada `jesimo.dato` de dos argumentos que dado un vector de datos $\vec{x}$ me devuelva el j-ésimo dato ordenado (es decir el $x_{(j)}$). **NOTA** No confundir con devolver el x_j que es la j-ésima entrada. Como sugerencia usar `arrange`, `order` ó `sort`. Un ejemplo de lo que debe hacer la función es:

```r
x <- c(12,8,9,7,14, 21)
jesimo.dato(x, 4)
```

```
## [1] 12
```

Gráficas univariadas

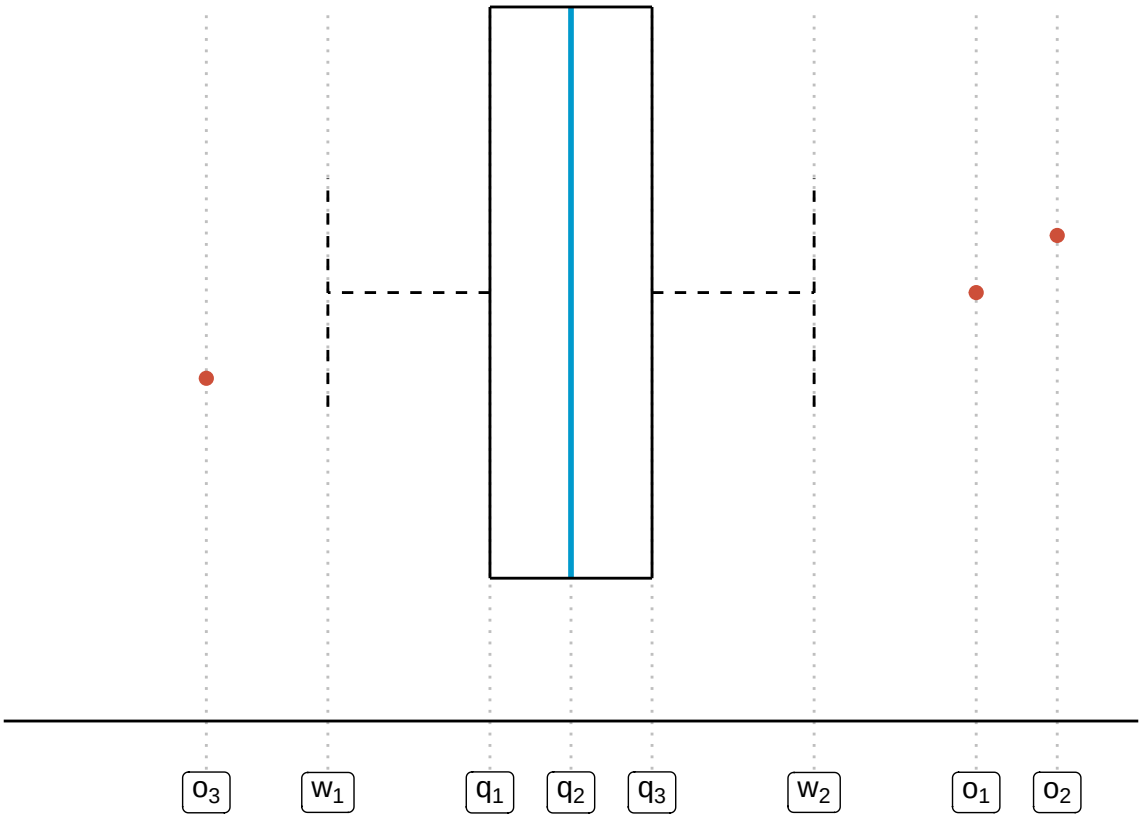
1. Gráfica de caja (boxplot) Una gráfica de caja pretende resumir los cuartiles, la mediana e identificar los *outliers* todo en una sola imagen. Para ello considera un vector numérico $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ tal que:

1. q_1 sea el primer cuartil ($\text{Cuantil}_{\vec{x}}(0.25)$), q_2 sea la mediana (que es lo mismo que el segundo cuartil o bien $\text{Cuantil}_{\vec{x}}(0.5)$) y q_3 corresponda al tercer cuartil ($\text{Cuantil}_{\vec{x}}(0.75)$).

2. $w_1 = \min\{x_j \in \vec{x} | x_j \geq q_1 - \frac{3}{2}IQR\}$ es el valor más pequeño de $\vec{x}$ que *no es outlier* y $w_2 = \max\{x_j \in \vec{x} | x_j \leq q_3 + \frac{3}{2}IQR\}$ es el valor más grande de $\vec{x}$ que *no es outlier*.
3. Sea $\text{Outliers}_{\vec{x}}$ el conjunto de outliers como lo definimos anteriormente:

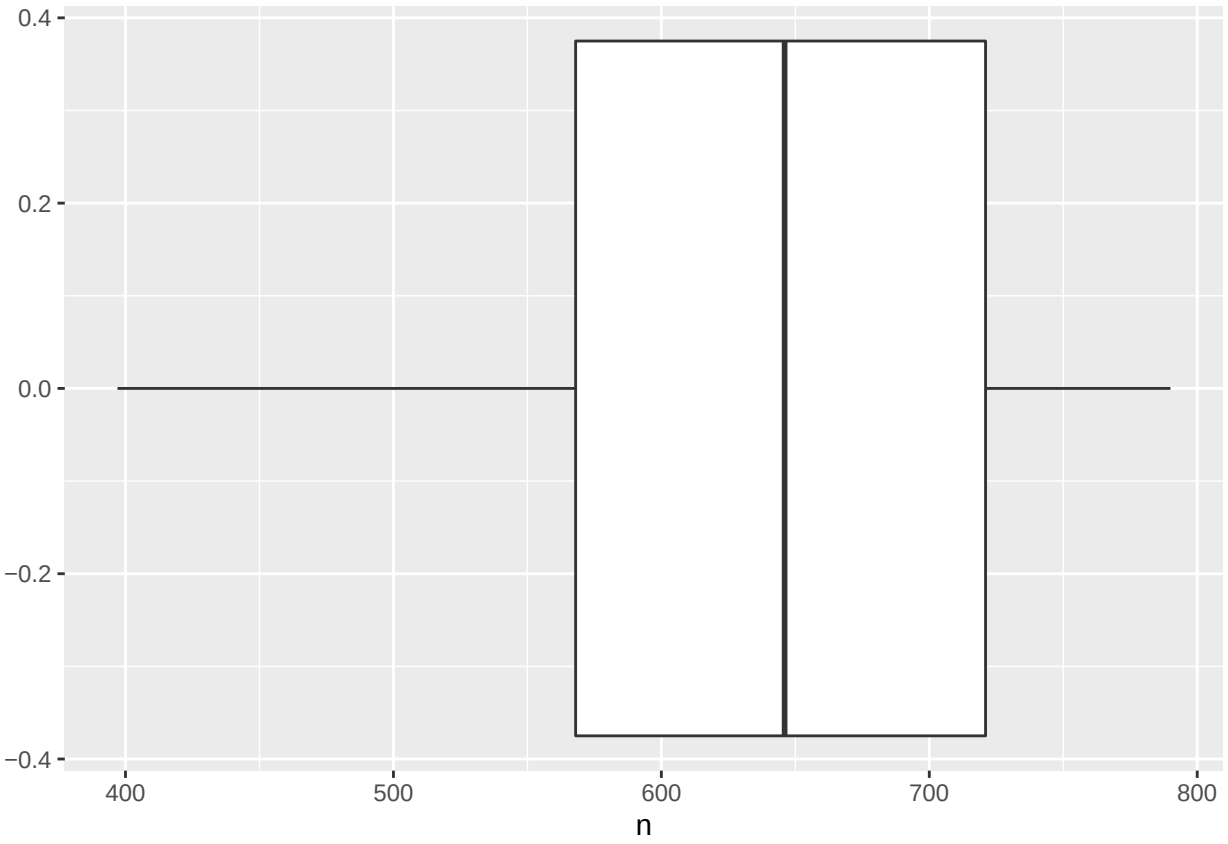
$$\text{Outliers}_{\vec{x}} = \left\{ x_i \in \vec{x} \middle| x_i \notin \left[q_1 - \frac{3}{2}\text{IQR}_{\vec{x}}, q_3 + \frac{3}{2}\text{IQR}_{\vec{x}} \right] \right\}$$

donde $\text{Outliers}_{\vec{x}} = \{o_1, o_2, \ldots, o_d\}$. Una gráfica de caja corresponde al siguiente diagrama:



La imagen anota la mediana, los cuartiles así como el rango de valores donde se sabe que no hay outliers. Finalmente la gráfica identifica los *outliers* si es que hay. Para armar una gráfica de boxplot usamos la librería de *ggplot2* especificando dentro de la función `ggplot` la base de datos de donde sale nuestra información:

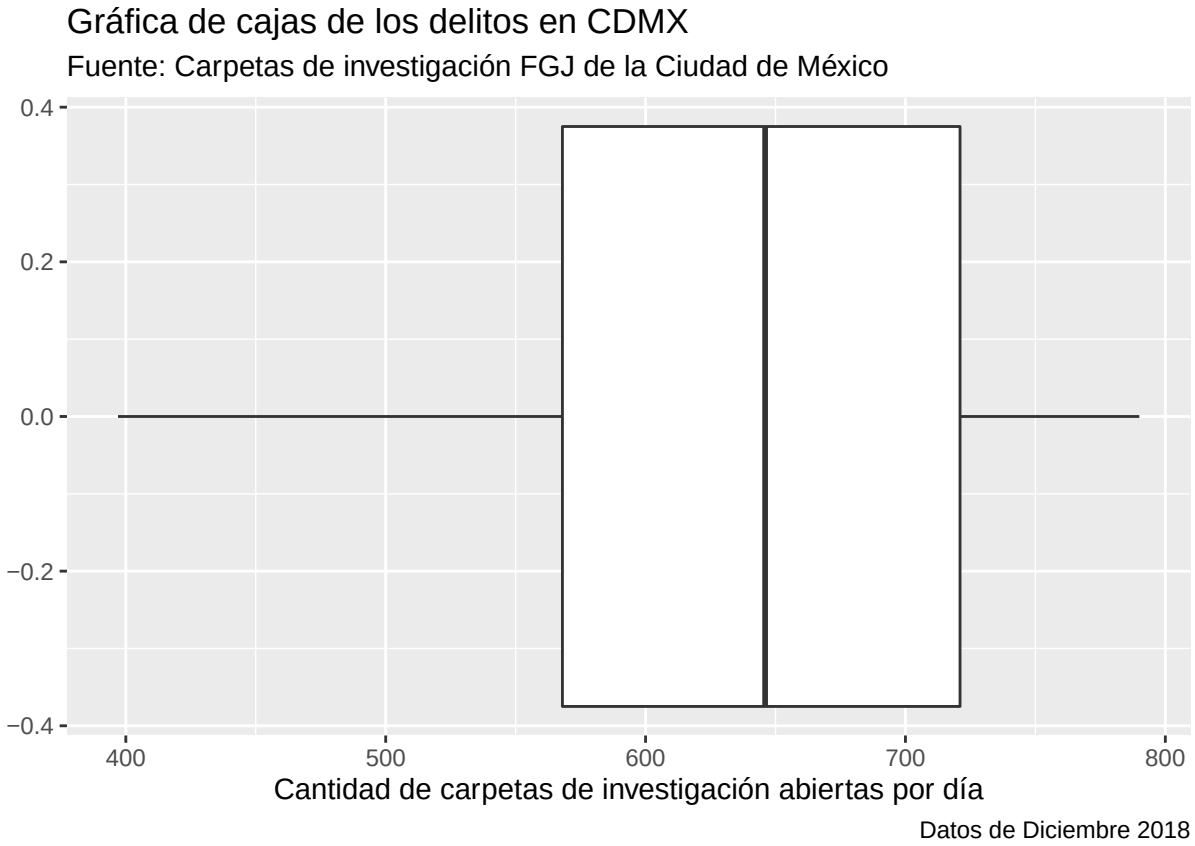
```
ggplot(conteo_delitos) +
  geom_boxplot(aes(x = n))
```



la cual pone la mediana en 646 como habíamos calculado, los cuartiles en 568 y 721 respectivamente. Finalmente no presenta *outliers* pues nuestro análisis previo nos mostraba que no había *outliers*.

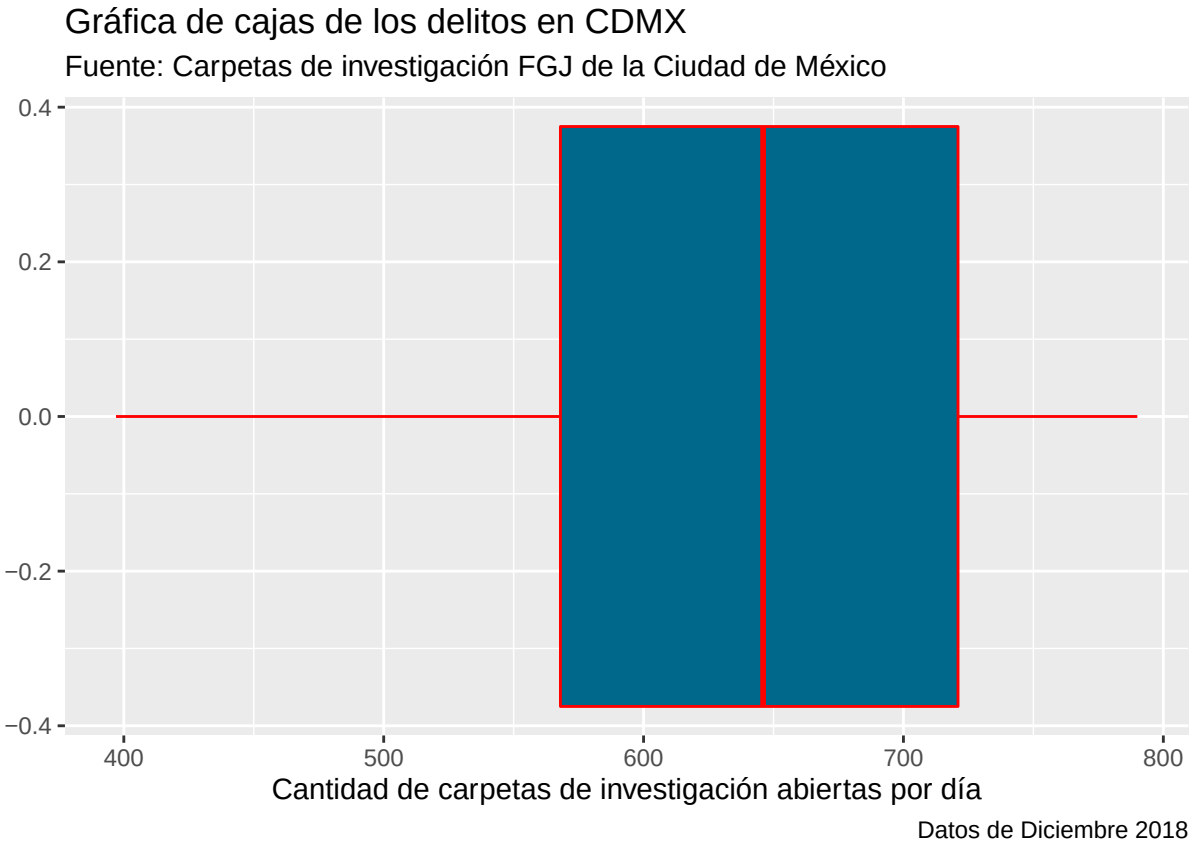
Podemos personalizar nuestra gráfica agregando títulos con la función lab:

```
ggplot(conteo_delitos) +
  geom_boxplot(aes(x = n)) +
  labs(
    x = "Cantidad de carpetas de investigación abiertas por día",
    y = "",
    title = "Gráfica de cajas de los delitos en CDMX",
    subtitle = "Fuente: Carpetas de investigación FGJ de la Ciudad de México",
    caption = "Datos de Diciembre 2018"
  )
```



Finalmente, podemos personalizar los colores de la gráfica editando directamente en el `geom_boxplot`:

```
ggplot(conteo_delitos) +
  geom_boxplot(aes(x = n), color = "red", fill = "deepskyblue4") +
  labs(
    x = "Cantidad de carpetas de investigación abiertas por día",
    y = "",
    title = "Gráfica de cajas de los delitos en CDMX",
    subtitle = "Fuente: Carpetas de investigación FGJ de la Ciudad de México",
    caption = "Datos de Diciembre 2018"
  )
```

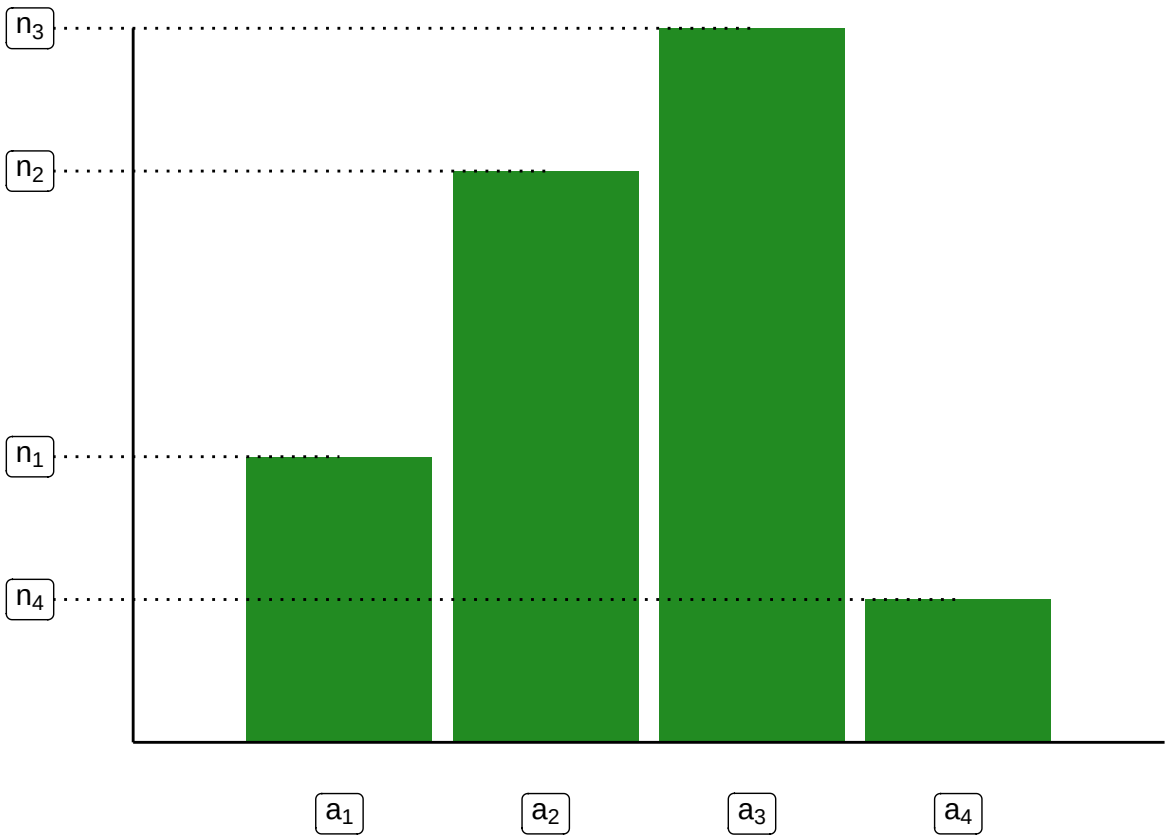


2. Gráfica de barras Sea $\vec{c} = (c_1, c_2, \ldots, c_n)^T$ un vector de datos categóricos. Sea $C = \{a_i | a_i \in \vec{c}\}$ el conjunto de ℓ valores únicos que se tienen registrados en el vector $\vec{c}$. Denotamos la cantidad de veces que aparece a_i en $\vec{c}$ como n_i; es decir:

$$n_i = \sum_{i=1}^{n} \mathbb{I}_{\{a_i\}}(c_i)$$

Una gráfica de barras consiste en una representación gráfica del conjunto:

$$\text{Barras} = \{(a_i, n_i) | a_i \in C\}$$



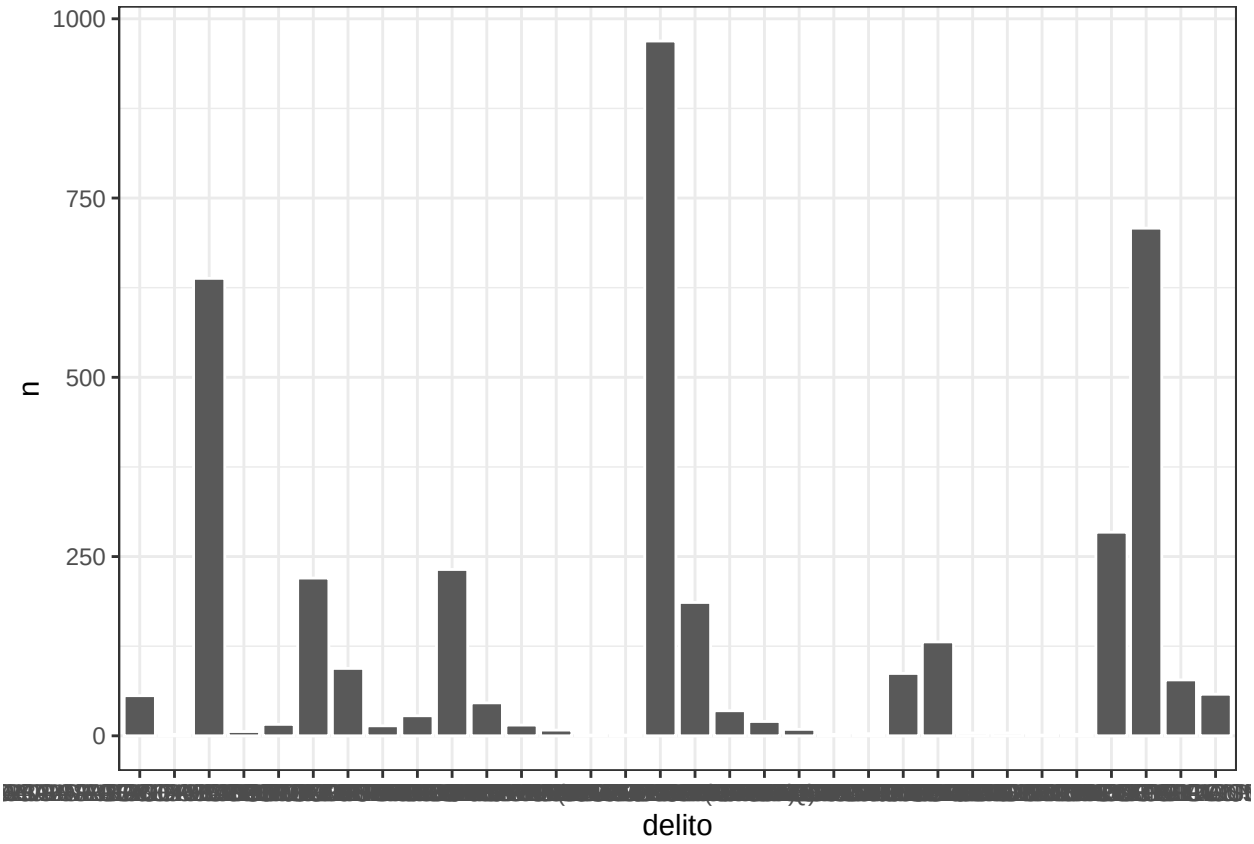
Gráficamente

Podemos crear una gráfica de barras con el comando `geom_col` para ello creemos unas barras correspondientes al tipo de delito (sólo en delitos que `categoria_delito` dice `ROBO`) haciendo una nueva base que cuente por delito:

```
conteo_tipo <- datos %>% filter(str_detect(categoria_delito,"ROBO")) %>%
  group_by(delito) %>% tally()
```

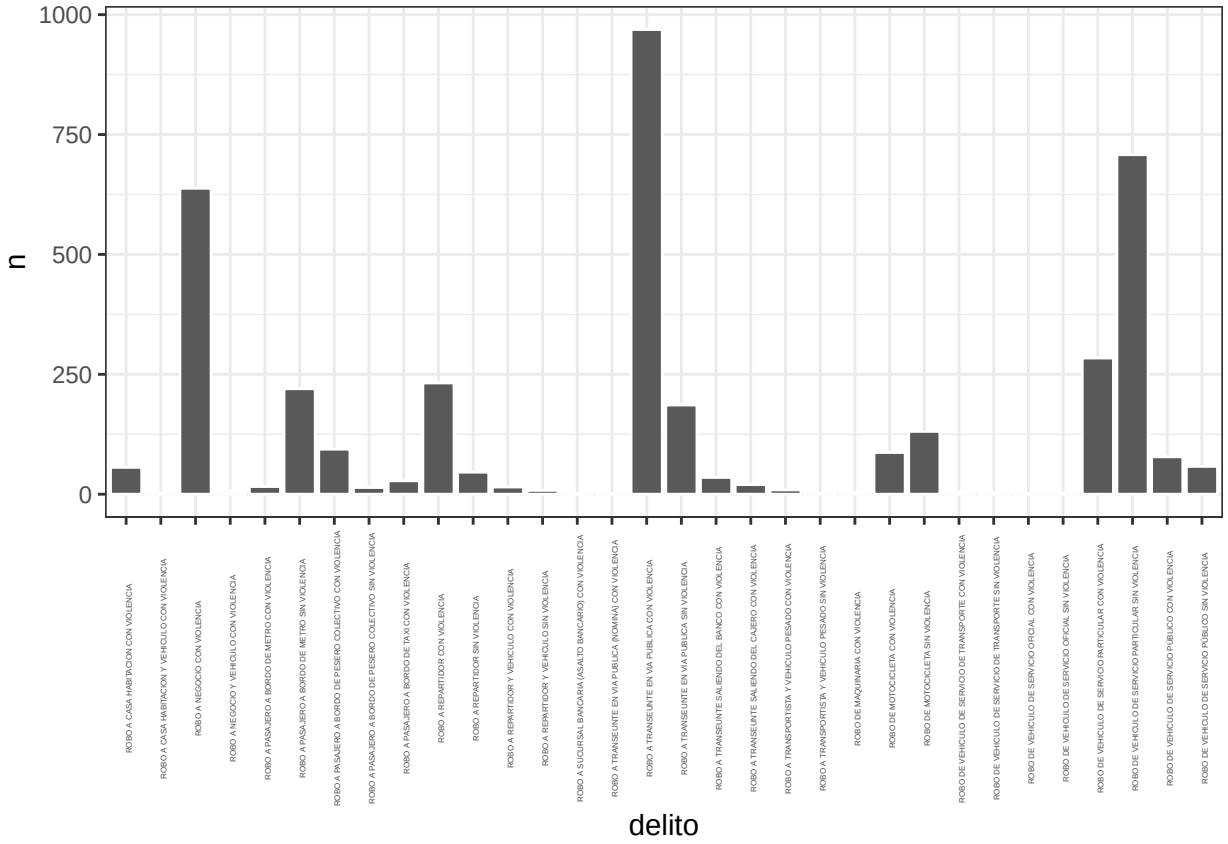
Y hagamos la gráfica:

```
ggplot(conteo_tipo) +
  geom_col(aes(x = delito, y = n), color = "white") +
  theme_bw()
```



Para evitar que se encime todo el texto podemos establecer un ángulo del mismo al usar `theme`:

```
ggplot(conteo_tipo) +
  geom_col(aes(x = delito, y = n), color = "white") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 3))
```

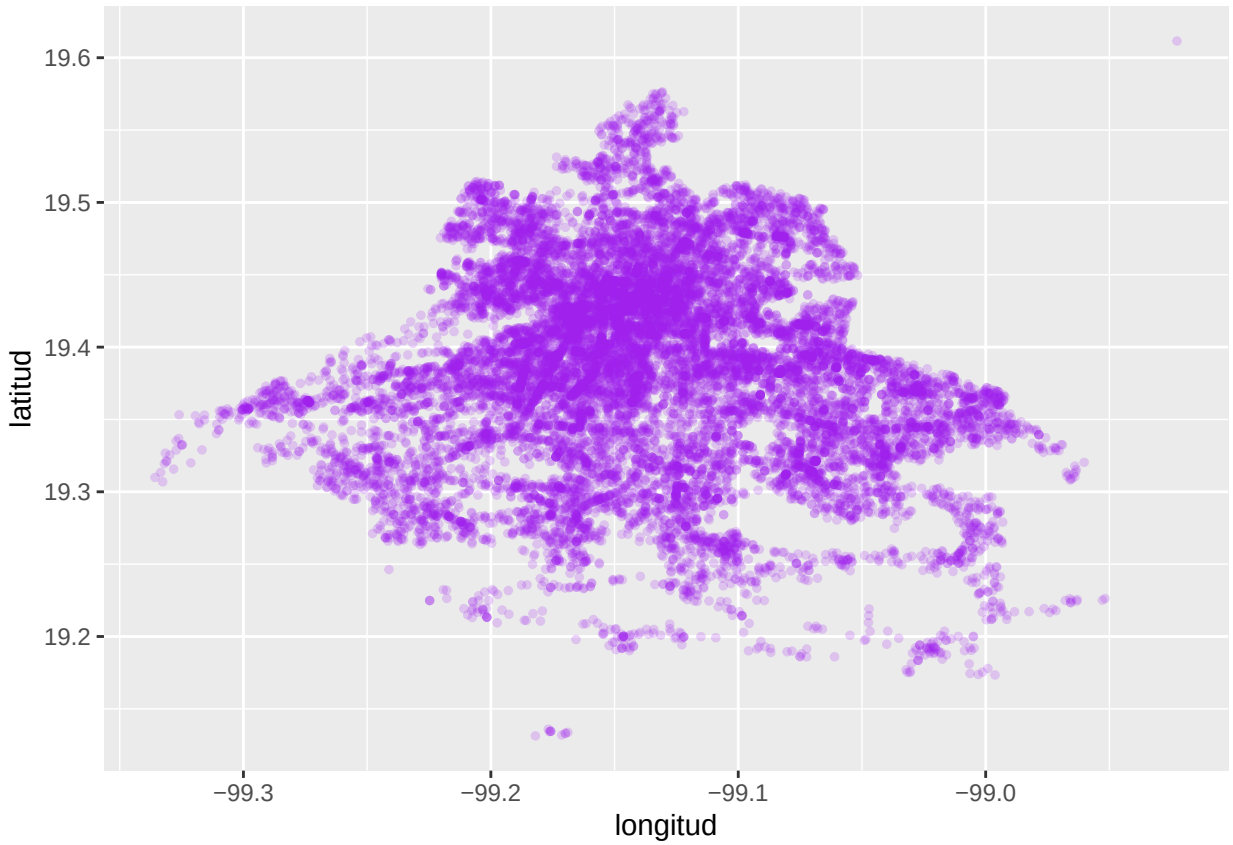


NOTA Una mala praxis es usar gráficas de pay pues es muy complicado contar una historia a partir de ellas. ¡No lo hagas!

Gráficas bivariadas

1. Gráfica de puntos (scatterplot) Dada una matriz de datos Z consideramos dos columnas numéricas z_i y z_j $(i \neq j)$ de dicha matriz. Sea $\mathbb{X} = \{(z_{i,1}, z_{j,1}), (z_{i,2}, z_{j,2}), \ldots, (z_{i,n}, z_{j,n})\}$ el conjunto de parejas ordenadas correspondientes a dichas columnas. Una gráfica de puntos consiste en la proyección de dichos puntos sobre $\mathbb{R}^2$. Para generarla en `R` podemos usar `ggplot`:

```
ggplot(datos) +
  geom_point(aes(x = longitud, y = latitud), size = 1, color = "purple",
             alpha = 0.2)
```



donde los parámetros `size` establecen el tamaño del punto, `color` su color y `alpha` su nivel de transparencia ($0 \leq \alpha \leq 1$).

2. Gráfica de líneas (lineplot) Dada una matriz de datos Z consideramos dos columnas numéricas z_i y z_j $(i \neq j)$ de dicha matriz. Sea $\mathbb{X} = \{(z_{i,1}, z_{j,1}), (z_{i,2}, z_{j,2}), \ldots, (z_{i,n}, z_{j,n})\}$ el conjunto de parejas ordenadas correspondientes a dichas columnas. Para evitar confusión de subíndices escribiré a las z_i como x y a las z_j como y de tal forma que $\mathbb{X} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ Supongamos, sin pérdida de generalidad que los datos están ordenados según las x: $x_1 \leq x_2 \leq \cdots \leq x_n$. Sea f la función de interpolación lineal dada por:

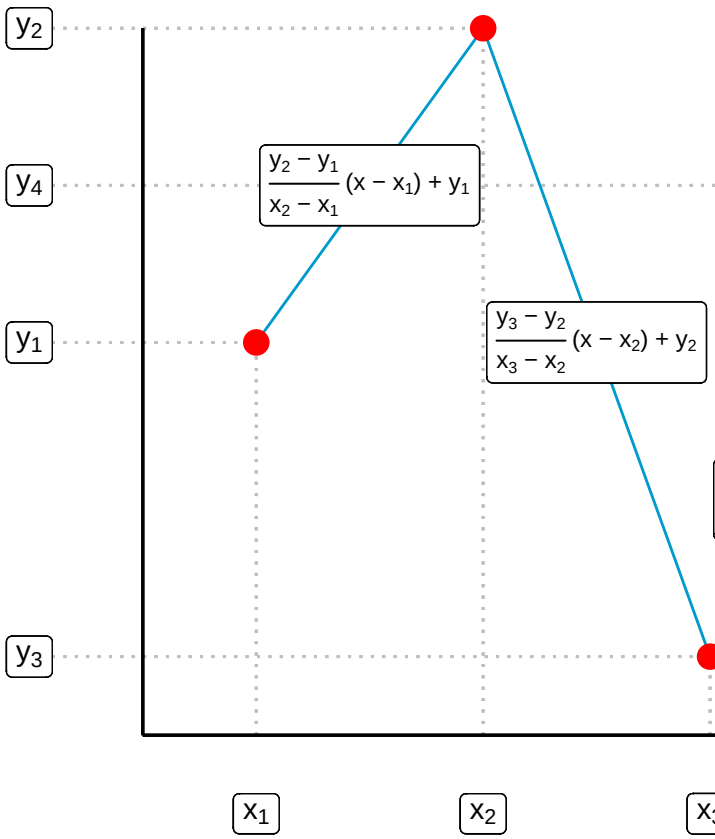
$$f(x) = \begin{cases} y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) & \text{si } x_1 \leq x \leq x_2 \\ \vdots & \\ y_{k-1} + \frac{y_k - y_{k-1}}{x_k - x_{k-1}}(x - x_{k-1}) & \text{si } x_{k-1} \leq x \leq x_k \\ \vdots & \\ y_{n-1} + \frac{y_n - y_{n-1}}{x_n - x_{n-1}}(x - x_{n-1}) & \text{si } x_{n-1} \leq x \leq x_n \end{cases}$$

Una gráfica de líneas corresponde a la representación gráfica del conjunto

$$\mathrm{Gr}_f = \Big\{ \big(x, f(x)\big) | z_{i,1} \leq x \leq z_{i,n} \Big\}$$

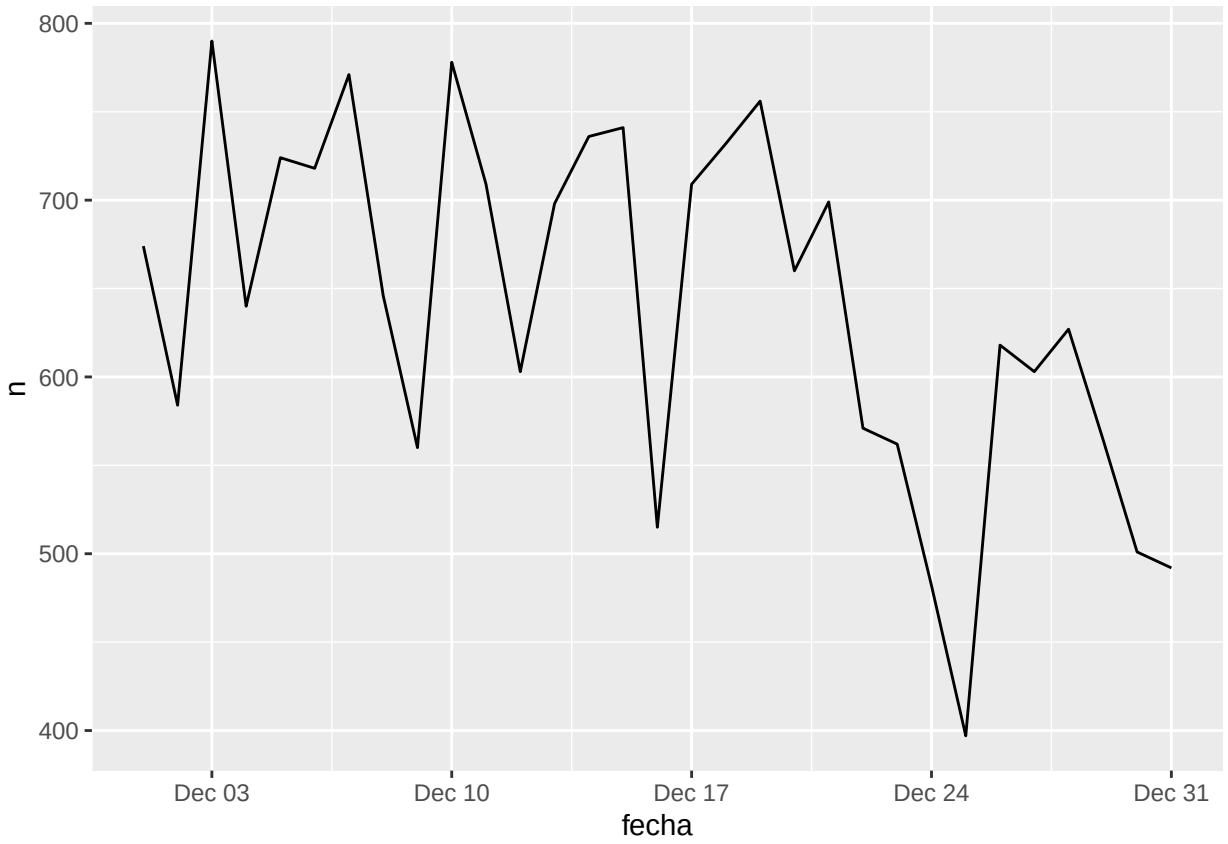
De manera un poco más intuitiva notamos que si tenemos, por ejemplo, $\mathbb{X} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$ una gráfica de líneas se construye interpolando una línea entre (x_1, y_1) y (x_2, y_2), otra línea entre (x_2, y_2) y (x_3, y_3) y, finalmente, otra recta entre (x_3, y_3) y (x_4, y_4). Usando la ecuación de la línea

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$



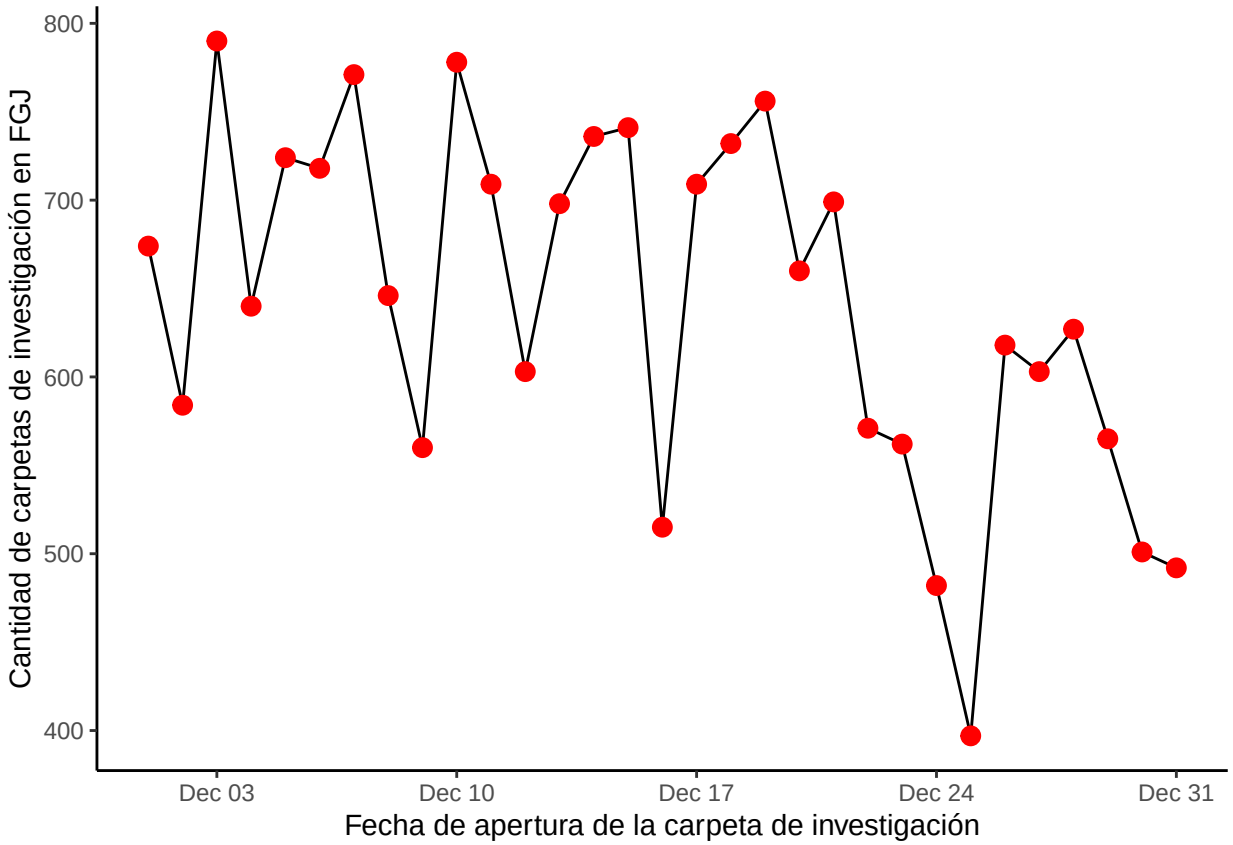
interpolamos cada uno de los puntos como en la gráfica siguiente:
Para realizar una gráfica de líneas podemos usar de nuevo ggplot2 con la opción de `geom_line`:

```
ggplot(conteo_delitos) +
  geom_line(aes(x = fecha, y = n))
```



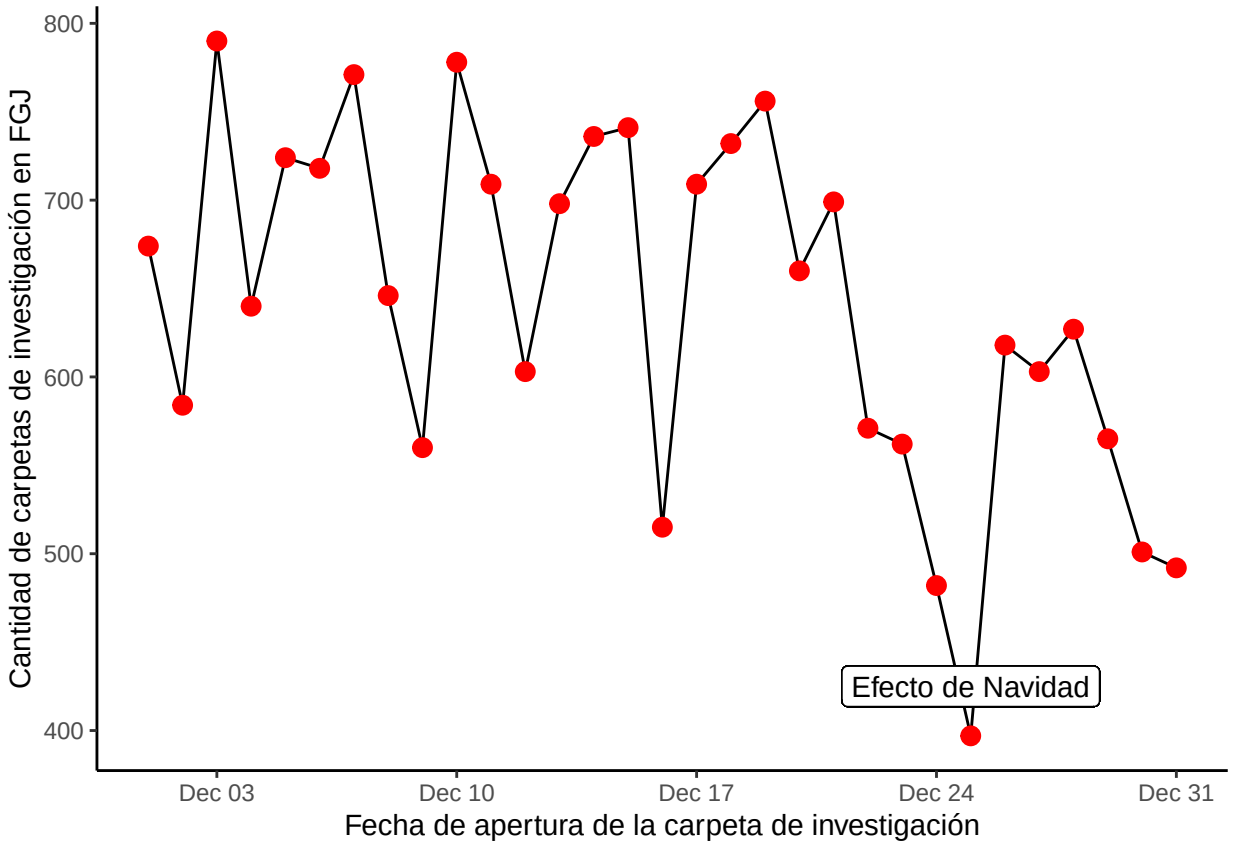
Podemos cambiar el tema y agregar puntos de otro color para que nuestra gráfica se vea más bonita:

```
ggplot(conteo_delitos) +
  geom_line(aes(x = fecha, y = n)) +
  geom_point(aes(x = fecha, y = n), color = "red", size = 3) +
  theme_classic() +
  labs(
    x = "Fecha de apertura de la carpeta de investigación",
    y = "Cantidad de carpetas de investigación en FGJ"
  )
```



Finalmente con `geom_label` podemos agregar anotaciones a nuestra gráfica:

```
ggplot(conteo_delitos) +
  geom_line(aes(x = fecha, y = n)) +
  geom_point(aes(x = fecha, y = n), color = "red", size = 3) +
  theme_classic() +
  labs(
    x = "Fecha de apertura de la carpeta de investigación",
    y = "Cantidad de carpetas de investigación en FGJ"
  ) +
  geom_label(aes(x = dmy("25/12/2018"), y = 425), label = "Efecto de Navidad")
```

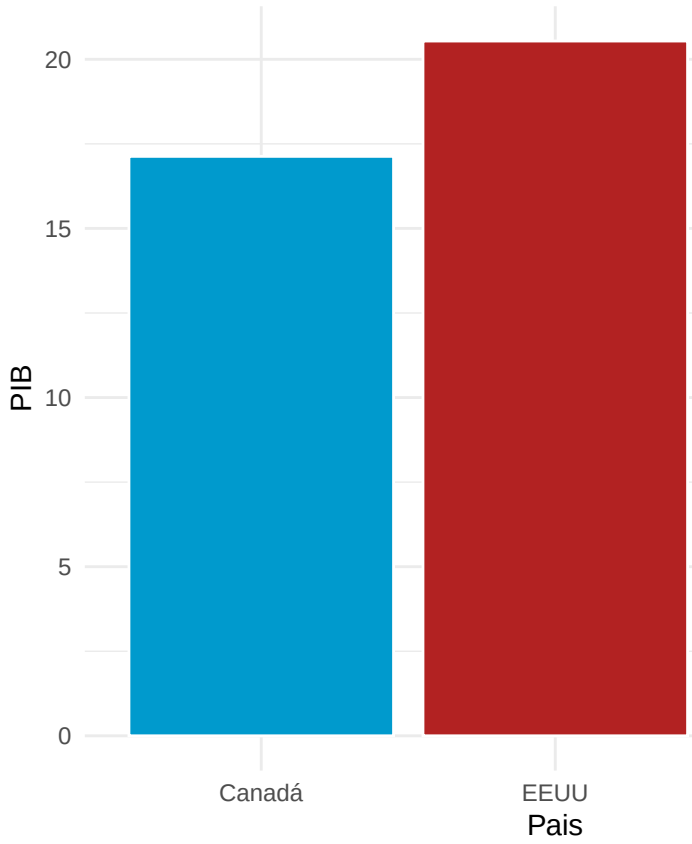


Ejercicio

Utiliza las siguiente bases de datos para replicar exactamente el formato de las gráficas que se muestran abajo de las bases. No todo viene en estas notas, la idea es que investigues y para ello te sugiero consultar este libro

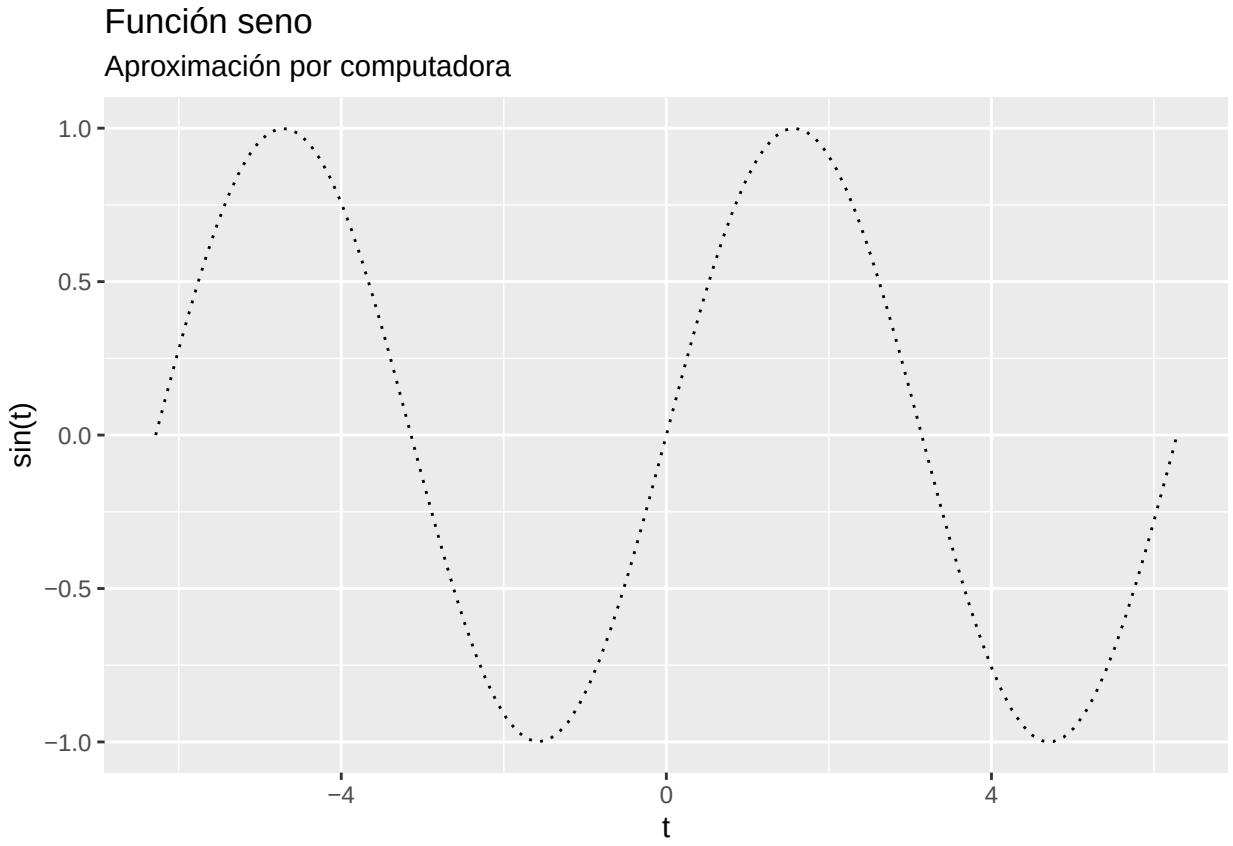
Gráfica de barras

```
datos.barras <- data.frame(Pais = c("EEUU","Canadá","México"),
                           PIB = c(20.54, 17.13, 1.21))
```



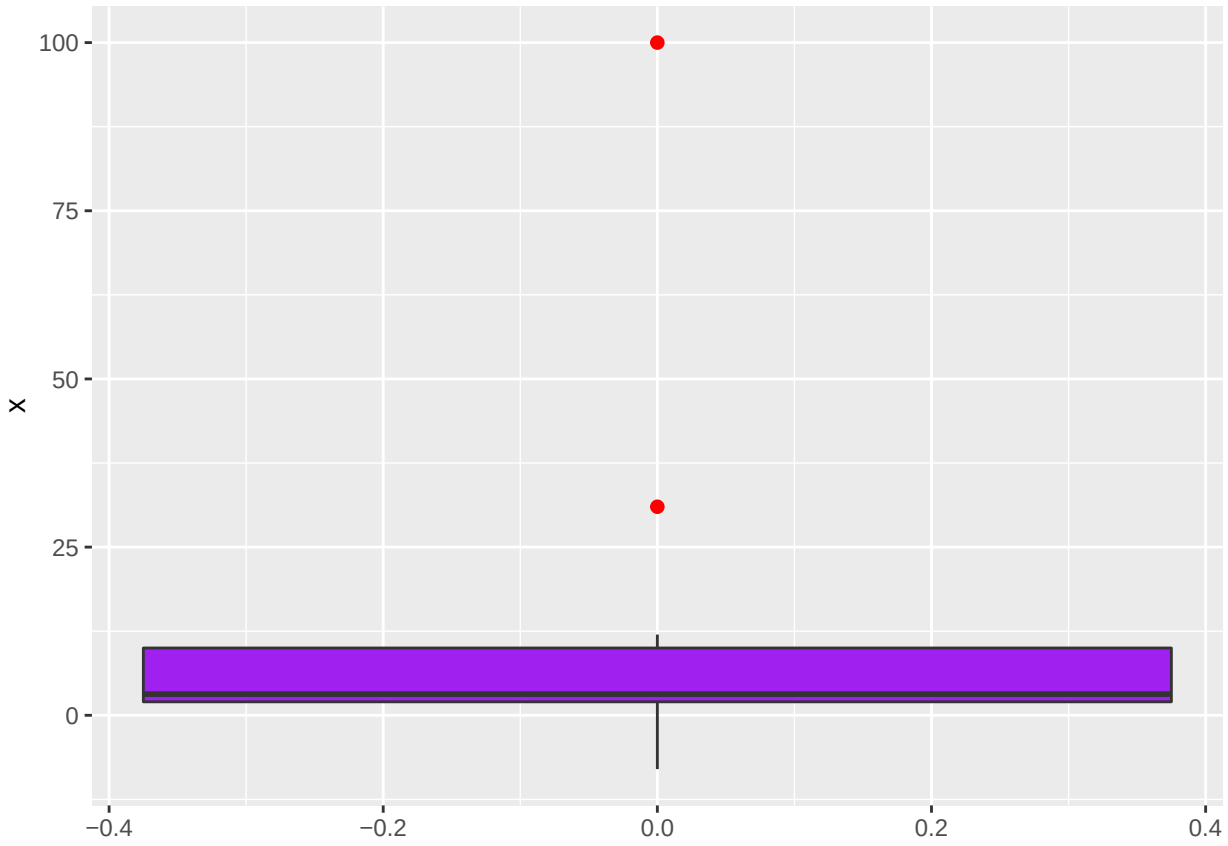
Los colores usados son `firebrick`, `deepskyblue3` y `forestgreen`:
Línea

```
x <- seq(-2*pi, 2*pi, length.out = 100)
datos.linea <- data.frame(x = x, y = sin(x))
```



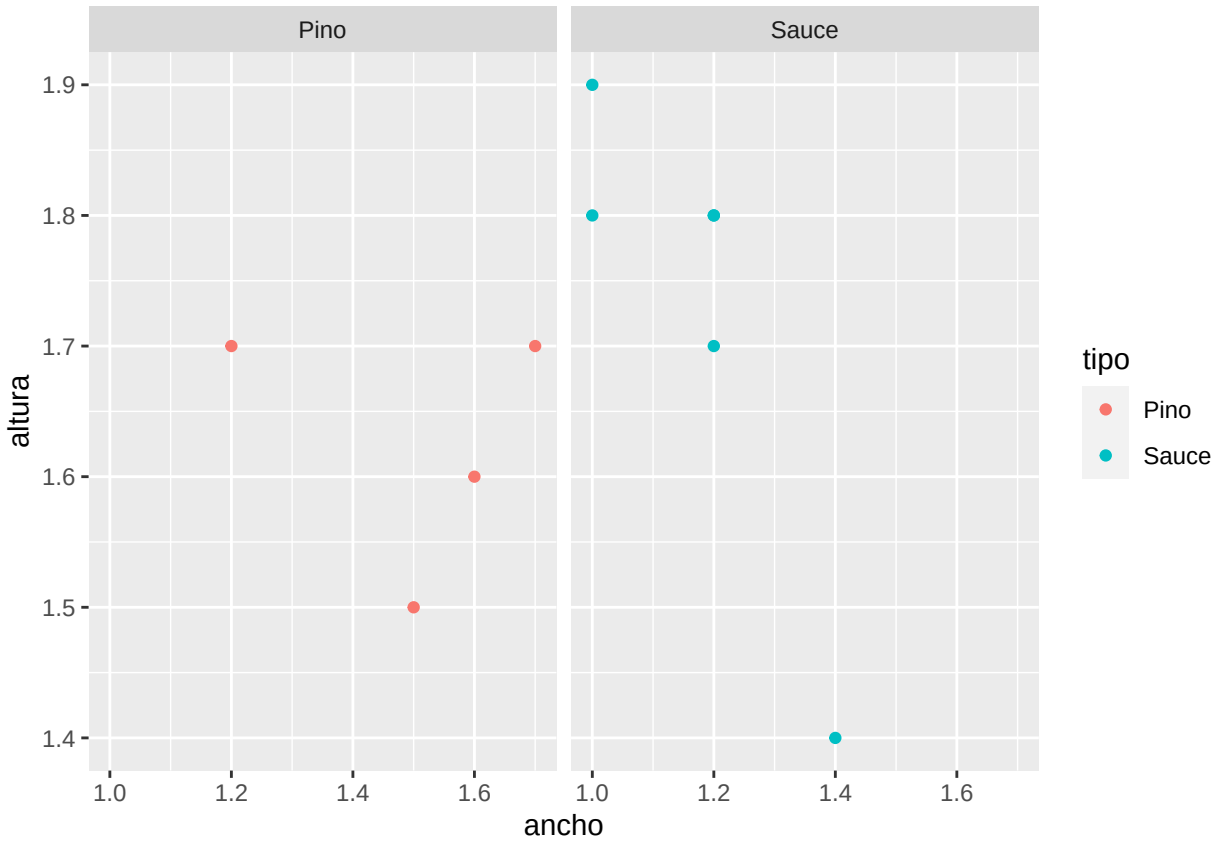
Boxplot

```
x <- c(1,10, 100, -2, 3, 5, 6, 12, -8, 31, 2, pi, 3)
datos.linea <- data.frame(Dientes = x)
```



Puntos

```
datos.arbol <- data.frame(altura = c(1.7, 1.4, 1.8, 1.9, 1.5, 1.7,
                                     1.6, 1.8, 1.7, 1.8),
                          ancho = c(1.2, 1.4, 1.2, 1, 1.5, 1.7, 1.6,
                                    1.2, 1.2, 1),
                          tipo = c("Pino","Sauce","Sauce","Sauce","Pino",
                                   "Pino","Pino","Sauce","Sauce","Sauce"))
```



Aproximaciones funcionales univariadas

Hasta ahora no hemos utilizado nada de probabilidad. Sin embargo, las siguientes aproximaciones suponen que para un vector observado de valores numéricos $\vec{x} = (x_1, x_2, \ldots, x_n)$ existe una variable aleatoria X_n que lo generó.

Por ejemplo, si $\vec{x}$ consiste en los registros ordenados de n tiros de una moneda (con $x_i \in \{0, 1\}$ donde se marcó 0 si era Águila y 1 en caso de Sol), podemos pensar que el vector $\vec{x}$ contiene distintas realizaciones de una variable aleatoria `Bernoulli`$(1/2)$. Asignar una variable aleatoria nos permite hacer ciertas inferencias sobre el objeto; por ejemplo si en un registro de 100 tiros de una moneda tuviéramos $\vec{x} = (1, 1, \ldots, 1)^T$ como un vector de sólo 1s entonces nuestro conocimiento de la probabilidad nos permitiría inferir que hay algo *extraño* pues el evento de observar sólo soles en el tiro de una moneda debería ser (bajo nuestra hipótesis de ser `Bernoulli`$(1/2)$) aproximadamente de $7.8886091 \times 10^{-31}$ (o sea, rarísimo).

Las siguientes son funciones que se pueden construir a partir de la muestra y donde es posible asignar una variable aleatoria.

1. Función de distribución empírica La función de distribución empírica está definida por:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{[x_i, \infty)}(x)$$

La cual es una aproximación a la probabilidad $\mathbb{P}(X \leq x)$ a partir de los datos.

La función de distribución empírica cumple las siguientes propiedades:

1. $\lim_{x \to -\infty} F_n(x) = 0$
2. $\lim_{x \to -infty} F_n(x) = 1$
3. Si $x < y$ entonces $F(x) \leq F(y)$ (no decreciente)

4. F es continua por la derecha con límites por la izquierda (cadlag).

Y por tanto es realmente una función de distribución.

2. Histograma

3. Densidad kernel

Ejercicio sugerido

Este ejercicio es para que tengas la seguridad de que comprendiste los conceptos previos y sabes calcularlos.
Es tedioso pero bueno para aclarar dudas.

Considera la siguiente base de datos:

```
dats <- data.frame(x = c(1,2,2,2,1,3), y = c(-100, -2, 1, 3, 1, 4), z = c("Rojo","Azul","Azul","Rojo","
kable(dats) %>% kable_styling(latex_options = "striped")
```

x	y	z
1	-100	Rojo
2	-2	Azul
2	1	Azul
2	3	Rojo
1	1	Verde
3	4	Amarillo

Calcula a mano:

1. La media y varianza de y
2. La curtosis y la asimetría de x
3. El cuantil 0.25 y el 0.75 de y así como su rango intercuartílico (IQR)
4. La moda de z
5. La mediana de y.
6. La MAD de x
7. Los outliers de y
8. El rango de y
9. Realiza el conteo de cuáles z pertenecen al conjunto $A = \{\text{Rojo}, \text{Amarillo}\}$

Grafica a mano:

1. Realiza una gráfica de caja (boxplot) para y
2. Realiza un scatterplot para (x, y)
3. Realiza una gráfica de líneas para (x, y) identificando la función de interpolación lineal $f(x)$ asociada.
4. Realiza una gráfica de barras de z
5. Identifica la función de distribución empírica para x, $F_n(x)$ y grafícala.
6. Realiza un histograma con $h = 2$ para x. Toma $I_2 = [2, 4)$.