Programación científica en R

Introducción a R

Marcos Ehekatzin García Guzmán

Agosto de 2024

1. Lectura de datos de un archivo

2. Análisis básico de una base de datos

Lectura de datos de un archivo

Lectura de datos de un archivo

- Los datos suelen leerse desde archivos externos, en lugar de crearlos manualmente.
- La clase anterior vimos que para importar un archivo en formato .rds utilizamos la función `readRDS()`
- Sin embargo, la mayor parte de archivos que se usan cotidianamente tienen formatos distintos (.txt, .csv, .xlsx ,.dta, .sav, etc)
- Para importar archivos de forma sencilla podemos utilizar distintas paqueterías.

Lectura de datos de un archivo

- Una paquetería es una colección de funciones y conjuntos de datos creados por la comunidad de 'R'.
- Para instalar una nueva paquetería utilizaemos la función `install.packages("paquetería")`.[1]
- Podemos revisar qué paqueterías tenemos instaladas utilizando la función `installed.packages()` y removerlas con las funciones `remove.packages()`.
- Podemos tener infinidad de paqueterías instaladas, por lo que debemos especificar cuales de esas paqueterías utilizaremos.
  - Esto lo hacemos con la función `library("paquetería")`.

[1] Dependiendo del repositorio esto puede cambiar.

La paquetería `haven`

- Para la importación de datos utilizaremos principalmente paquetería `haven`.
- Nos ayudará a importar datos de SAS (`read_sas()`), SPSS (`read_sav()`) y stata (`read_dta()`)

1. El primer paso es instalarla:

```
> install.packages('haven')
```

2. El segundo paso es cargar la paquetería:

```
> library('haven')
```

- Nota: Las paqueterías solamente las tenemos que instalar una vez, mientras que siempre debemos indicar qué paqueterías utilizaremos.

La paquetería `readxl`

- La paquetería `readxl` nos ayudará a importar archivos desde excel, ya sea .xls o .xlsx, con la función `read_excel()`.
- `readxl` contiene ejemplos de archivo, utilicemos uno de estos para ver más de sus funcionalidades.

```
readxl_example()
```

```
 [1] "clippy.xls"    "clippy.xlsx"   "datasets.xls"  "datas
 [5] "deaths.xls"    "deaths.xlsx"   "geometry.xls"  "geome
 [9] "type-me.xls"   "type-me.xlsx"
```

Ejemplos:

```
xlsx_example <- readxl_example("datasets.xlsx")
# Podemos especificar en qué hoja del archivo trabajaremos
excel_sheets(xlsx_example)
```

```
## [1] "iris"     "mtcars"   "chickwts" "quakes"
```

```
example1 <- read_excel(xlsx_example, sheet = "mtcars")
# Podemos especificar qué celdas vamos a importar
example2 <- read_excel(xlsx_example, n_max = 3)
example3 <- read_excel(xlsx_example, range = "C1:E4")
example4 <-read_excel(xlsx_example, range = "mtcars!B1:D5")
```

Análisis básico de una base de datos

Análisis básico de una base de datos

- La función más simple para describir una variable es la función summary.
  - Nos provee de los siguientes datos: Media, Mediana, Mínimo y Máximo y el primer y tercer cuartil.

```
df <- data.frame(unclass(summary(example1$mpg)))
df
```

```
##         unclass.summary.example1.mpg..
## Min.                           10.40000
## 1st Qu.                        15.42500
## Median                         19.20000
## Mean                           20.09062
## 3rd Qu.                        22.80000
## Max.                           33.90000
```

- En la función `summary()` también podemos incluir vectores lógicos.

- Para ver la relación entre dos variables podemos ver el coeficiente de correlación: $r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \times \sum(y_i - \bar{y})^2}}$
  - Básicamente, el coeficiente de correlación nos dice qué tan relacionadas están dos variables y en qué dirección.
  - Si el coeficiente es negativo, entonces la relación de las variables es inversa (si una sube, otra baja)
  - Si el coeficiente es positivo, el coeficiente dice que si una variable aumenta, la otra también.

```
# Ejemplo
cor(example1$mpg,example1$hp)
```

```
## [1] -0.7761684
```

- Para visualizar la relación entre dos variables podemos utilizar un gráfico de distpersión:

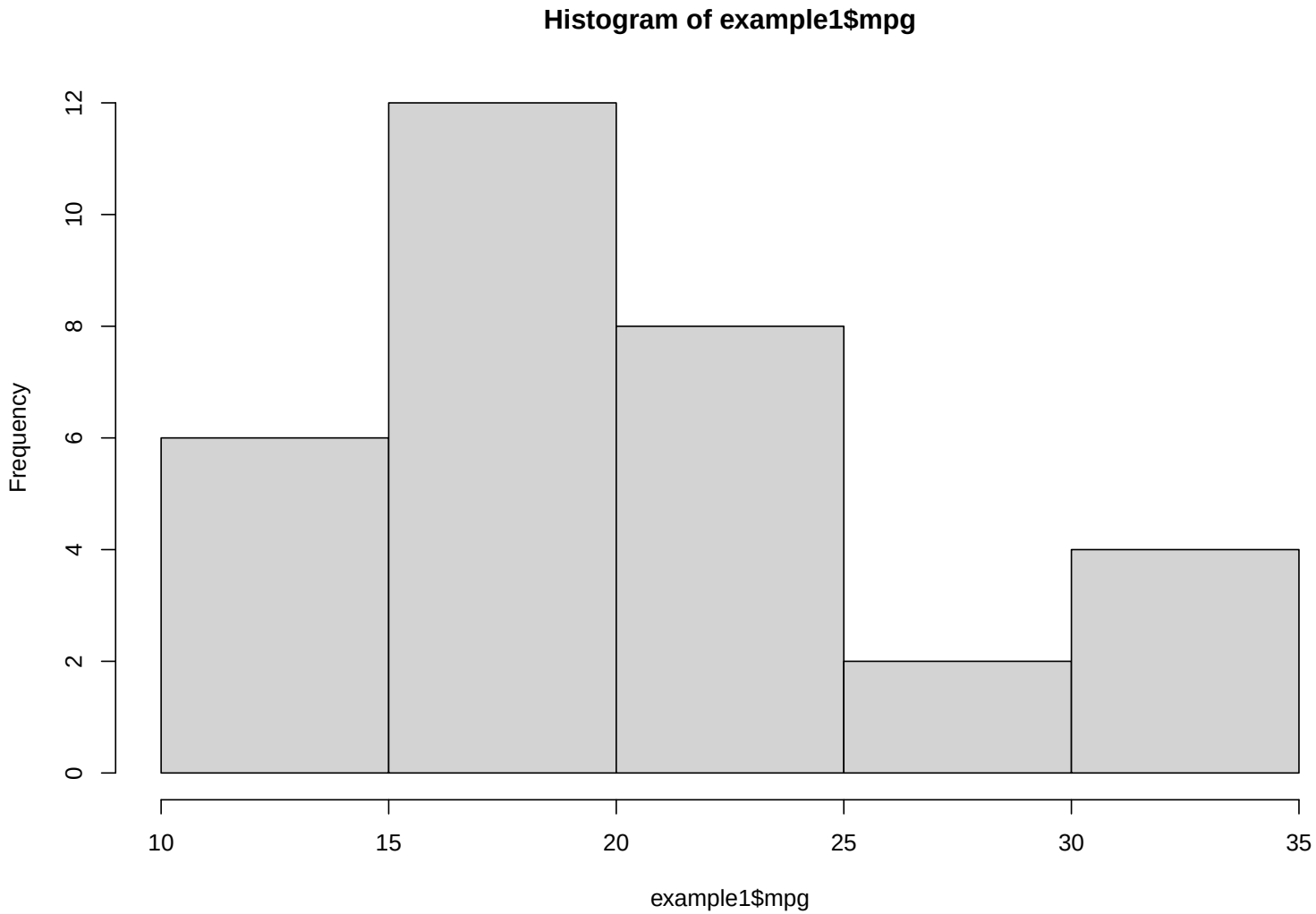
```
# Ejemplo
> disp <- plot(example1$mpg, example1$hp)
```

- Mientras que para visualizar como evoluciona una variable a través del tiempo podemos utilizar una gráfica de línea:

```
> lines <- plot(homicidios$Year[homicidios$Cve=="09"],
                homicidios$Total[homicidios$Cve=="09"],
                type = "l")
```

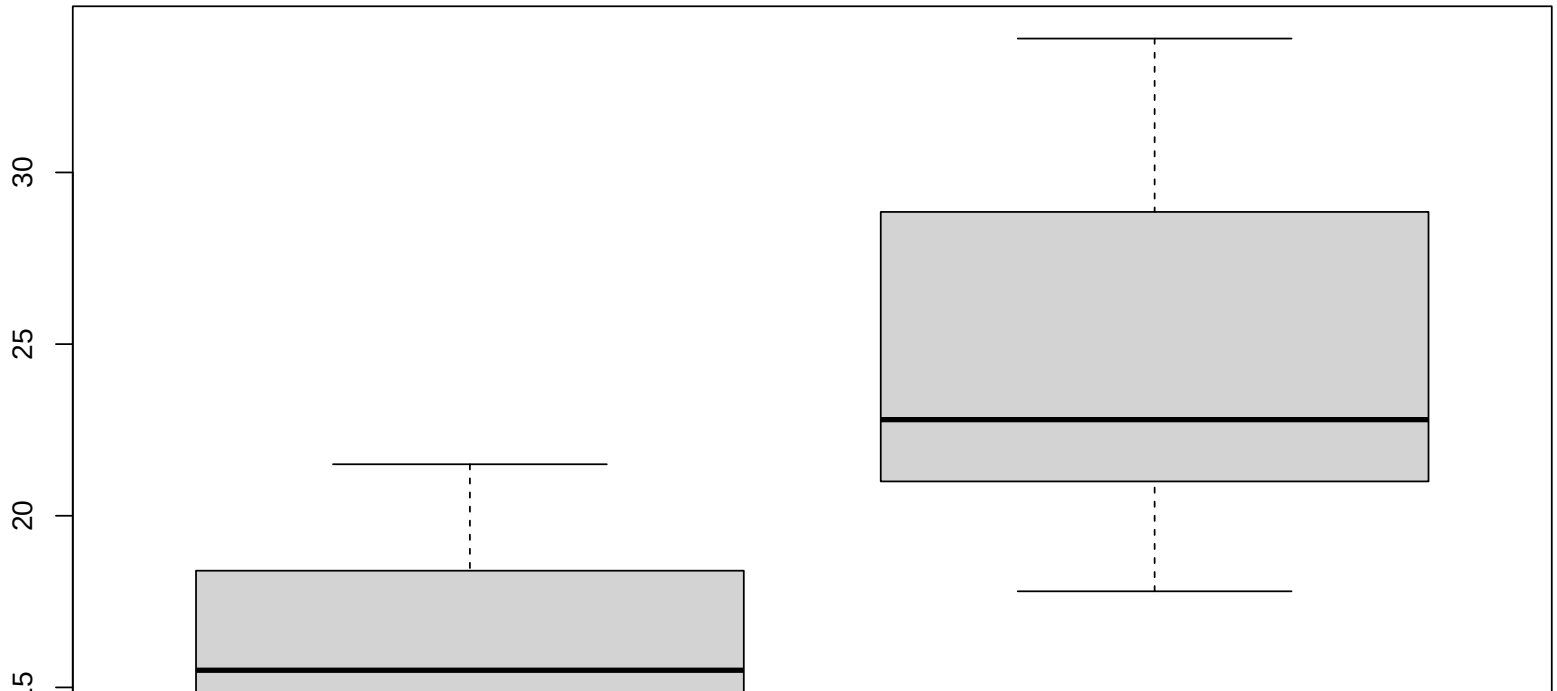
Análisis básico de una base de datos

- Además de los descriptivos básicos, muchas veces nos puede interesar cómo se ve la distribución de las variables.
- Para ello podemos hace histogramas, con la función `hist()`



- Para visualizar los descriptivos básicos podemos utilizar una gráfica de caja o *boxplot*, que a demás es útil para comparar dos muestras.

```
boxplot(example1$mpg[example1$gear==3],
        example1$mpg[example1$gear==4])
```



Ejercicios

- Ejercicio 1: Utilizando la base de homicidios

1. Genere un data.frame con el total de homicidios en el país por año.

- Hint: Utilice la función `rownames(df)` para crear una nueva columna en el df.
- Hint: Puede cambiar el nombre de las variables con: `names(df)<-c("name_1", ..., "name_n")`

2. Haga una gráfica que muestre la evolución de los homicidios en México.
3. Repita lo anterior pero solamente para los homicidios de hombres.

- Ejercicio 2: Utilizando la base de la ENOE

1. Obtenga las estadísticas descriptivas del ingreso (ingocup) para:

a. El total de la muestra
b. Mujeres y hombres por separado (Hombres: sex == 1, Mujeres: sex == 2)

2. Haga una gráfica de caja sobre el ingreso (log) separando a hombres y mujeres.
3. Haga un histograma sobre el ingreso separando a hombres y mujeres.
4. Haga un vector con el promedio de el ingreso (log) y la edad por estado (ent) y otro con el promedio de la edad.
5. Haga una gráfica que muestre la relación entre el ingreso (log) y la edad (eda)
6. Haga un data frame que contenga los dos vectores y haga una gráfica que muestre la relación entre el ingreso