

# Lab2\_Daniela\_Canabal.R

Perfil 1

2023-02-24

```
# Daniela Alexandra Canabal Valdes
# 22/02/2023
# Maestria Ciencias Forestales primer semestre

# HW_02
# Laboratorio 3: Importar datos a R

setwd("C:/Users/Perfil 1/Documents/Maestria/Estadistica/Estadistica/Scrips")

# Parte 1 -----

# Importar archivos .csv -----

# Después de tener la base de datos guardados en el formato .csv y nombre deseado, se procede a
# La importación usando la función read.csv. En el caso de que los datos se encuentren ubicados en
# otra carpeta diferente, se informa a R el directorio correcto añadiendo: "direccion/nombre_archi
# vo.csv".

trees <- read.csv("DBH_1.csv", header = TRUE)
head(trees)
```

```
##   Tree  dbh parcela
## 1    1  16.5      1
## 2    2  25.3      1
## 3    3  22.1      1
## 4    4  17.2      1
## 5    5  16.1      1
## 6    6   8.1      1
```

```
# Ingresar datos directo en la consola -----

# Una manera alternativa de capturar los datos de forma manual en el programa RStudio es creando
objetos y asignando datos:
# Supongamos que tenemos 30 mediciones de árboles de la especie Pinus pseudostrobus realizado me
diante un inventario en campo, los datos son los siguientes:

dbh <- c(16.5, 25.3, 22.1, 17.2, 16.1, 8.1, 34.3, 5.4, 5.7, 11.2, 24.1, 14.5, 7.7, 15.6, 15.9, 1
0, 17.5, 20.5, 7.8, 27.3, 9.7, 6.5, 23.4, 8.2, 28.5, 10.4, 11.5, 14.3, 17.2, 16.8)

# Almacenar y Accesar datos en la Nube -----

# Importar datos a R que se encuentran en una URL no segura: Las que comienzan con http: es senci
llo siempre que:
# * Los datos se almacenan en un formato simple, p. ej. Texto sin formato,
# * El archivo no está incrustado en un sitio web HTML más grande,

# Los datos los vamos a descargar en un objeto llamado Profepa. Los datos corresponden a las Acc
iones de Inspección y Vigilancia a los Recursos Forestales en ANPs.

prof_url <- "http://www.profepa.gob.mx/innovaportal/file/7635/1/accionesInspeccionfoanp.csv"
profepa <- read.csv(prof_url, encoding = "latin1" )
head(profepa)
```

##	Entidad	Inspección	Recorrido	Operativo
## 1	Aguascalientes	7	5	1
## 2	Baja California	0	12	3
## 3	Baja California Sur	5	9	3
## 4	Campeche	1	4	3
## 5	Chiapas	3	11	0
## 6	Chihuahua	48	25	2

```
# Servicios como Dropbox y GitHub ahora almacenan sus datos en URL seguras. Puedes saber si los
datos están almacenados en una dirección web segura si comienzan con https en lugar de http. Ten
emos que usar diferentes comandos para descargar datos de URL seguras.
# (Por default Dropbox añade un cero al final del link y siempre tendremos que sustituirlo por e
l número 1)
```

```
library(repmis)
conjunto <- source_data("https://www.dropbox.com/s/hmsf07bbayxv6m3/cuadro1.csv?dl=1")
```

```
## Downloading data from: https://www.dropbox.com/s/hmsf07bbayxv6m3/cuadro1.csv?dl=1
```

```
## SHA-1 hash of the downloaded data file is:
## 2bdde4663f51aa4198b04a248715d0d93498e7ba
```

```
head(conjunto)
```

```
##   Arbol Fecha Especie Clase Vecinos Diametro Altura
## 1     1    12      F     C        4     15.3  14.78
## 2     2    12      F     D        3     17.8  17.07
## 3     3     9      C     D        5     18.2  18.28
## 4     4     9      H     S        4      9.7   8.79
## 5     5     7      H     I        6     10.8  10.18
## 6     6    10      C     I        3     14.1  14.90
```

*# Para descargar los datos de Github podemos utilizar la paquetería readr y su función read\_csv:*

```
library(readr)
file <- paste0("https://raw.githubusercontent.com/mgtagle/",
               "202_Analisis_Estadistico_2020/master/cuadro1.csv")
inventario <- read_csv(file)
```

```
## Rows: 50 Columns: 7
```

```
## — Column specification —————
## Delimiter: ","
## chr (2): Especie, Clase
## dbl (5): Arbol, Fecha, Vecinos, Diametro, Altura
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(inventario)
```

```
## # A tibble: 6 × 7
##   Arbol Fecha Especie Clase Vecinos Diametro Altura
##   <dbl> <dbl> <chr>   <chr>   <dbl>   <dbl>   <dbl>
## 1     1     12 F       C        4     15.3   14.8
## 2     2     12 F       D        3     17.8   17.1
## 3     3      9 C       D        5     18.2   18.3
## 4     4      9 H       S        4      9.7    8.79
## 5     5      7 H       I        6     10.8   10.2
## 6     6     10 C       I        3     14.1   14.9
```

```
# Parte 2 -----
```

```
# Operaciones con La base de datos -----
```

*# Siempre al realizar una operación con las variables que contiene la base de datos debemos utilizar el nombre del objeto en este caso trees. Como primer ejemplo determinaremos la media y la desviación estándar de la variable dbh de la base de datos trees*

```
mean(trees$dbh)
```

```
## [1] 15.64333
```

```
# Desviación estándar  
sd(trees$dbh)
```

```
## [1] 7.448892
```

```
# Selección mediante restricciones -----
```

*# El uso de restricciones sirve para obtener una muestra bajo ciertas condiciones restrictivas de la base de datos trees y es por ejemplo, una operación útil en el manejo de información de los inventarios forestales.*

*# Los condicionantes restrictivos más empleadas son:*

*# igual o mayor (>=), mayor que (>), igual que (==)*

*# igual o menor (<=), menor que (<), no igual (!=)*

*# Para mostrar como funcionan las restricciones podemos realizar las siguientes preguntas: ¿Cuántos individuos tienen un diámetro menor (<) a 10 cm?*

```
sum(trees$dbh < 10)
```

```
## [1] 8
```

*# También es interesante saber cuáles son los individuos que son inferiores al diámetro (dbh < 10 cm). Para esto, hacemos uso de la función which que nos regresará cuáles individuos son los que poseen tal restricción.*

```
which(trees$dbh < 10)
```

```
## [1] 6 8 9 13 19 21 22 24
```

*# Excluir los diámetros que se encuentran en la parcela 2. El objeto resultante se puede grabar como trees.13. El símbolo ! indica NO.*

```
trees.13 <- trees[!(trees$parcela=="2"),]
trees.13
```

```
##      Tree  dbh parcela
## 1      1 16.5      1
## 2      2 25.3      1
## 3      3 22.1      1
## 4      4 17.2      1
## 5      5 16.1      1
## 6      6  8.1      1
## 7      7 34.3      1
## 8      8  5.4      1
## 9      9  5.7      1
## 10     10 11.2      1
## 21     21  9.7      3
## 22     22  6.5      3
## 23     23 23.4      3
## 24     24  8.2      3
## 25     25 28.5      3
## 26     26 10.4      3
## 27     27 11.5      3
## 28     28 14.3      3
## 29     29 17.2      3
## 30     30 16.8      3
```

*# Selección de submuestras -----*

*#Una submuestra se puede obtener de cualquier base de datos que este disponible en R mediante la función "subset"*

```
trees.1 <- subset(trees, dbh <= 10)
head(trees.1)
```

```
##      Tree  dbh parcela
## 6      6  8.1      1
## 8      8  5.4      1
## 9      9  5.7      1
## 13     13  7.7      2
## 16     16 10.0      2
## 19     19  7.8      2
```

```
# Representacion grafica -----
```

```
# Ahora tenemos dos base de datos: trees y trees.1 disponibles en R y con ambas podemos realizar operaciones y gráficas
```

```
mean(trees$dbh)
```

```
## [1] 15.64333
```

```
mean(trees.1$dbh)
```

```
## [1] 7.677778
```

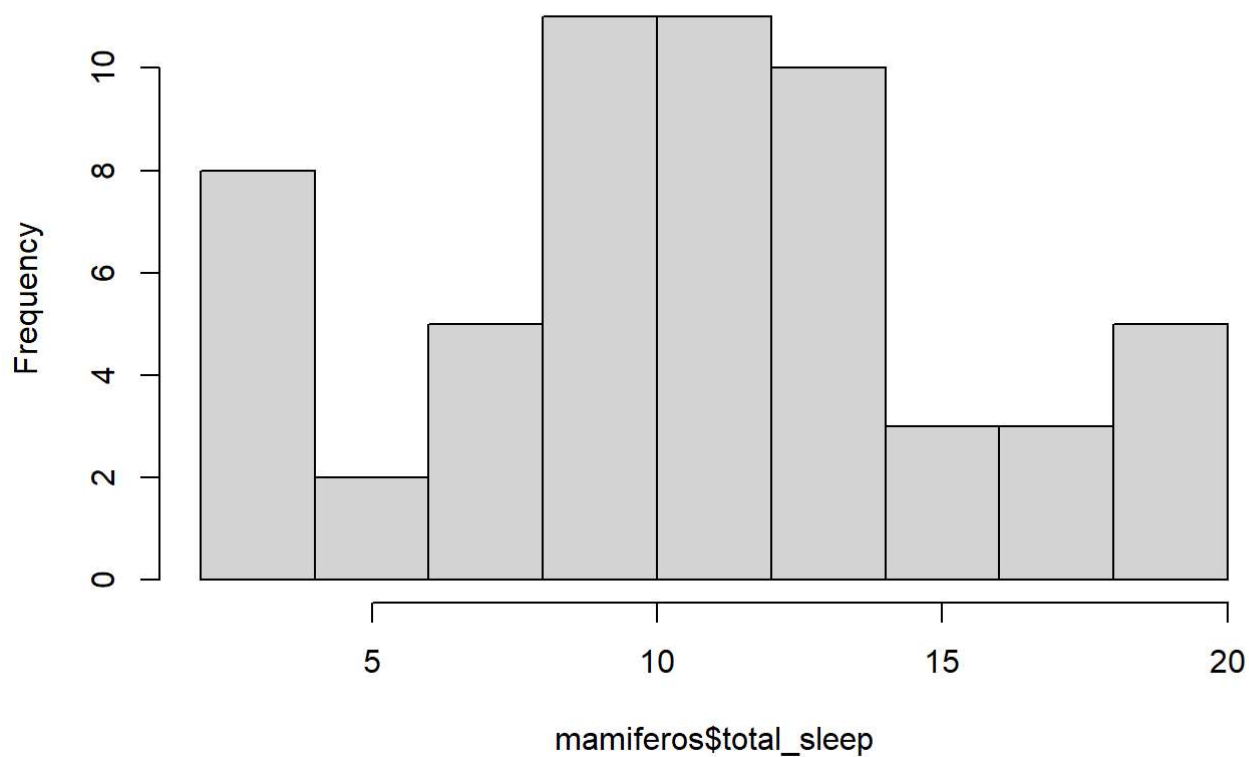
```
# Histograma -----
```

```
# Un histograma es una representación visual de la distribución de un conjunto de datos. Como tal, la forma de un histograma es su característica más evidente e informativa: Le permite ver fácilmente dónde se encuentra una cantidad relativamente grande de datos y dónde hay muy pocos datos.
```

```
mamiferos <- read.csv("https://www.openintro.org/data/csv/mammals.csv")
```

```
hist(mamiferos$total_sleep)
```

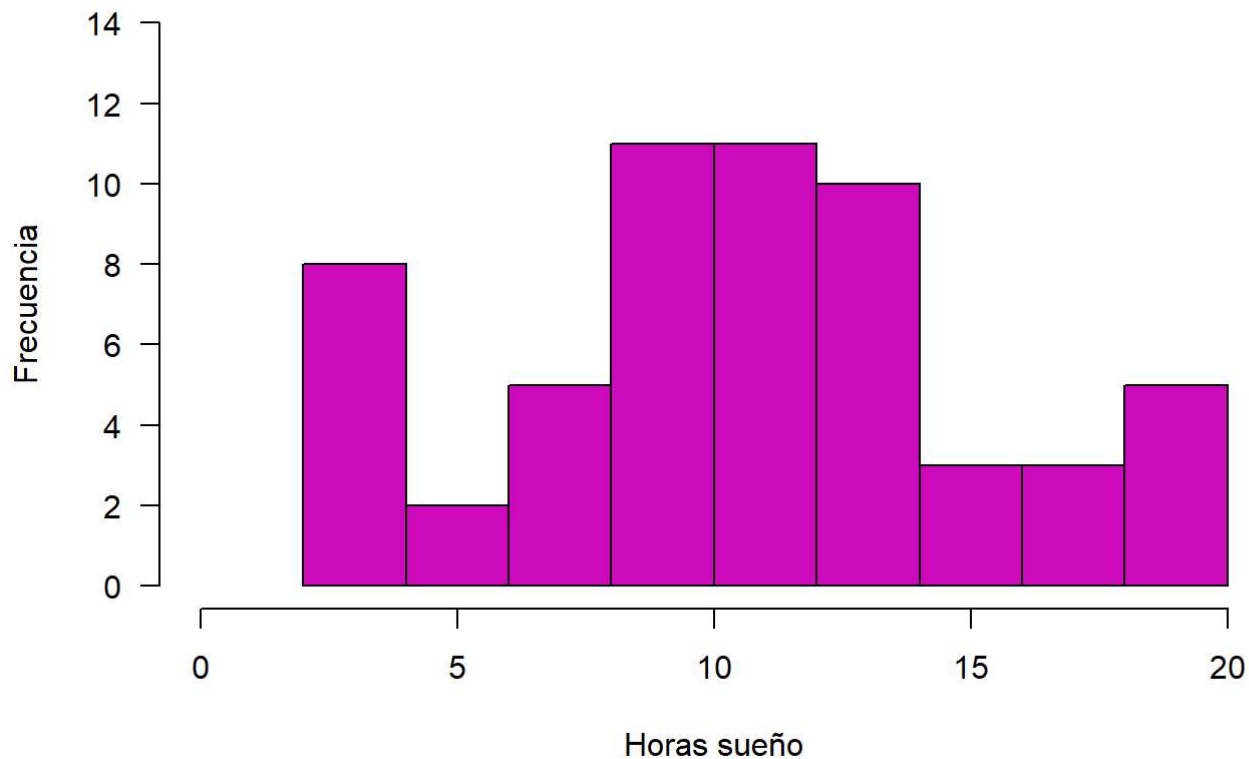
## Histogram of mamiferos\$total\_sleep



*# Como puedes observar en la figura 6, el título de la figura, así como el nombre del eje de x & y se dan por default, por lo tanto tendremos que trabajar el código para obtener un histograma más presentable*

```
hist(mamiferos$total_sleep,  
     xlim = c(0,20), ylim = c(0,14),  
     main = "Total de horas sueño de las 39 especies",  
     xlab = "Horas sueño",  
     ylab = "Frecuencia",  
     las = 1,  
     col = cm(9))
```

## Total de horas sueño de las 39 especies



*# Barplot o grafico de barras -----*

*# Un diagrama de barras (o gráfico de barras) es uno de los tipos de gráficos más comunes. Muestra la relación entre una variable numérica y una categórica. Cada entidad de la variable categórica se representa como una barra. El tamaño de la barra representa su valor numérico.*

```
data("chickwts")
head(chickwts[c(1:2,42:43, 62:64), ])
```

```
##      weight      feed
## 1      179 horsebean
## 2      160 horsebean
## 42     226 sunflower
## 43     320 sunflower
## 62     379   casein
## 63     260   casein
```

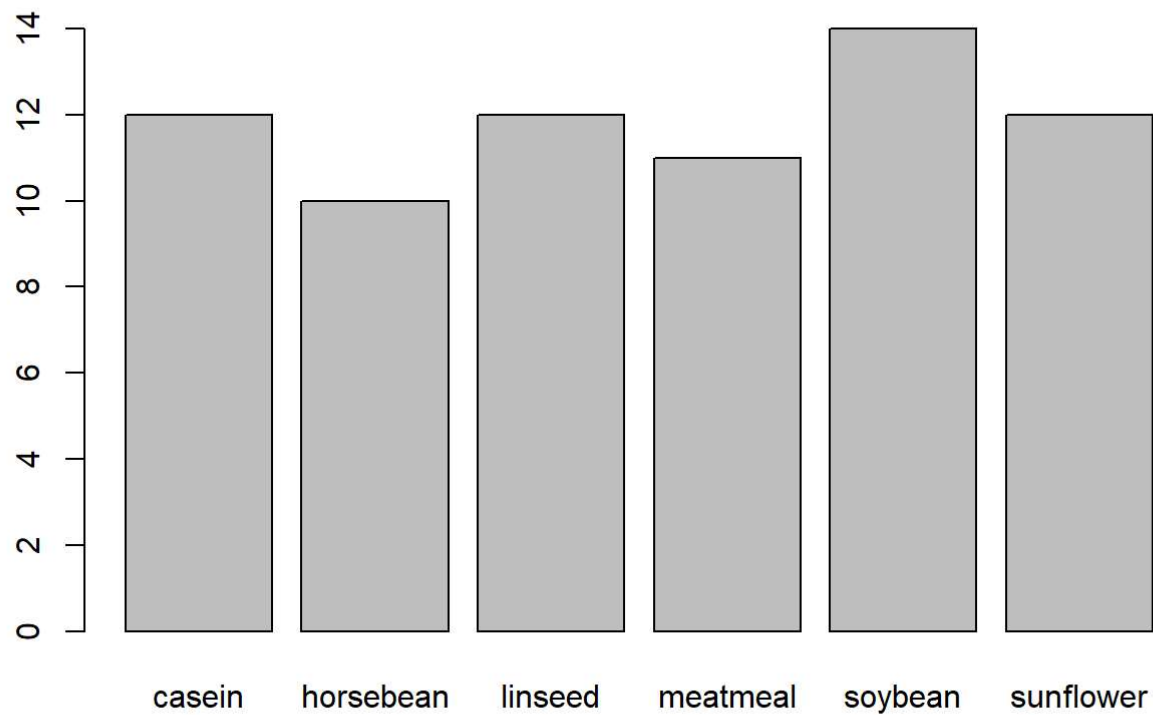
*# Primeramente tendremos que acomodar los datos en columnas (los datos originales están acomodados en dos columnas (weight, feed, Peso y tipo de alimentación de los pollos.)*

```
feeds <- table(chickwts$feed)
feeds
```



```
##  
##   casein horsebean  linseed  meatmeal  soybean  sunflower  
##      12       10       12       11       14       12
```

```
barplot(feeds)
```



```
# Ordenado  
barplot(feeds[order(feeds, decreasing = TRUE)],  
ylim = c(0,14),  
main = "Alimentacion de Aves",  
xlab = "Alimento",  
ylab = "peso (gr)",  
las = 1,  
col = rainbow(6))
```

## Alimentacion de Aves

