Bet Bardají Bofill
SP lab session

Introduction:

The main goal of this practical is to get familiarized with the use of R. In onder to achieve it, I'm including the non-guided exercises to show the capacities gained through the guide.

1. **Can you interpret the output of which(big) ?.**

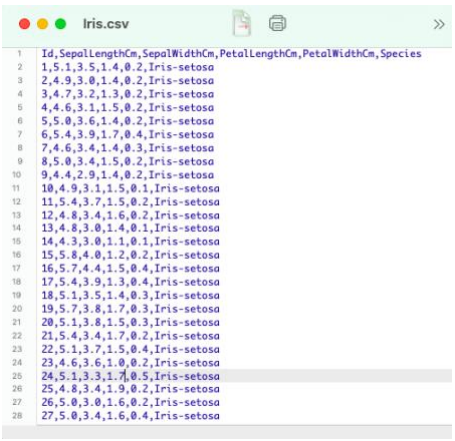
- Here the main goal is to understand how the function which works.

The which function as we can see, is returning the position of the values in the logical vector. In this case it's giving the positions of the vector where you find values greater than five.

```
> x <- c(2.3,3,5,7,7.5,7.5,4)
Warning message:
R graphics engine version 15 is not supported by this version of RStudio. The Plots
 tab will be disabled until a newer version of RStudio is installed.
> big <- x > 5
> which(big)
[1] 4 5 6
>
```

2. **Explain the differences between myDF1, myDF2 and my DF3. Which of them is correct?**

- Here the goal is to analyze the arguments included in the read function.
  - myDF1 <- read.table('iris.csv') #read file 'iris.csv' on the current workspace
    str(myDF) # describe the structure of myDF



  - myDF2 <- read.table('iris.csv', sep = ',') #read file by setting the field separator character as ','
    str(myDF)
  - myDF3 <- read.table('iris.csv', sep = ',', head = TRUE) #read file by setting the field separator character
    str(myDF)
    - sep argument is used to specify that the data should be separated with a blank space
    - header = TRUE is used when the first row in the file contains column names.

Since the third option includes sep with comas and header= TRUE , it is the correct option.

Bet Bardají Bofill
SP lab session

1. **Create a function that given a vector x = x1 · · · xN , computes the following expression.**

$$y = \frac{\sum_k k * x_k^2}{N}$$

Code:

```
-Creating a sum function
x <- c(x1,...xn)
ksum=function(k){
  out <- 0
  while (k){
   out <- out + (k<-k*x^2)
  return(out)
}
- Using the sum function
xk <- c(x1,...xn)
y <- readline(prompt = "Enter y: ")
xk <- readline(prompt = "Enter xk: ")  # is a value from the vector
N <- readline(prompt = "Enter N: ")
y <- as.integer(y)
x <- as.integer(x)
N<- as.integer(N)
y <- (sum((k) * (xk^2)))/N
```

2. Please try to explain the function calls and the output generated.

```
> data("iris")
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
  Species
1  setosa
2  setosa
3  setosa
4  setosa
5  setosa
6  setosa
> apply(iris[,-5],2,sd)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.8280661    0.4358663    1.7652982    0.7622377
```

Output:

You get a table with the 6 first values of each column
With the function apply in this case you get the standard deviation of al the variables (columns) of a single row.

Functions:

- data():This function returns a character vector of a specified dataset.
- head():The head() function in R is **used to display the first 6 rows present in the input data frame**.
- apply(): **apply()** takes Data frame or matrix as an input and gives output in vector, list or array
- split():takes a vector or data frame as an argument and divides the information into groups

Bet Bardají Bofill
SP lab session

```
> apply(iris[1:10,-5],1,summary)
            1     2     3     4    5     6     7     8
Min.    0.20 0.200 0.200 0.200 0.20 0.400 0.300 0.200
1st Qu. 1.10 1.100 1.025 1.175 1.10 1.375 1.125 1.175
Median  2.45 2.200 2.250 2.300 2.50 2.800 2.400 2.450
Mean    2.55 2.375 2.350 2.350 2.55 2.850 2.425 2.525
3rd Qu. 3.90 3.475 3.575 3.475 3.95 4.275 3.700 3.800
Max.    5.10 4.900 4.700 4.600 5.00 5.400 4.600 5.000
            9   10
Min.    0.200 0.10
1st Qu. 1.100 1.15
Median  2.150 2.30
Mean    2.225 2.40
3rd Qu. 3.275 3.55
Max.    4.400 4.90
> |
```

Here the output is the summary of 10 rows, including their median, mean, 1st and 3rd quarter and max and min values

3. Please try to explain the function calls and the output generated.

A very usefull function is the `split()` function, where we can retrieve the strata of a dataframe given an input factor. The output is a list with each strata in each element, named as the levels of the factor. An example with the *iris* dataset is:

```
> iris.strata <- split(iris,iris$Species)
> length(iris.strata)

[1] 3

> names(iris.strata)

[1] "setosa"     "versicolor" "virginica"

> summary(iris.strata$versicolor)
```

lenght():counts the number of characters in string, including any spaces, and returns the number.
In this case, the output is 3 indicating the are three categories of species

names():get or set the name of an Object. In this case, it gives us the name of the 3 categories of Species

summary(): produce result summaries of the results. In this case we are obtaining the stadistic information of the Species versicolor

4.Compute the same values through the tapply() function.

The tapply() helps to compute statistical measures (mean, median, min, max, etc..) or a self-written function operation for each factor variable in a vector. In this example, we are obtaining the mean of each factor variable in fvii combined and diseasef

Bet Bardají Bofill
SP lab session

```
> tapply(fvii,diseasef,mean)
Ehlers-Danlos syndrome            Hemophilia
              54.00000              61.00000
          Hemophilia A          Hemophilia B
              54.50000              46.00000
         Hhemophilia A     Myeloproliferative
              59.00000              58.00000
      Thrombocytopenia             Vasculitis
              53.22222              54.83333
```

3. Can you explain the function of tapply()?

- Basically, it helps us to create a subset of a vector and then apply some functions to each of the subsets.

A powerful function is the aggregate() function, which accepts the R formula interface for easy computations: > aggregate(. ~ Species, iris, mean)

```
> agg_mean = aggregate(iris[,1:4],by=list(iris$Species),FUN=mean, na.rm=TRUE)
> agg_mean
     Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
```

```
> sapply(iris,class)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   "numeric"    "numeric"    "numeric"    "numeric"
     Species
    "factor"
```

Sapply() helps in applying functions on a list, vector, or data frame and returns an array or matrix object of the same length.

4. Can you compute the maximum and minimum value of each variable and each Specie in the iris dataset?

```
> max(iris$Sepal.Length)
[1] 7.9
> max(iris$Sepal.Width)
[1] 4.4
> max(iris$Petal.Length)
[1] 6.9
> max(iris$Petal.Width)
[1] 2.5
```

```
> min(iris$Sepal.Length)
[1] 4.3
>
> min(iris$Sepal.Width)
[1] 2
> min(iris$Petal.Length)
[1] 1
> min(iris$Petal.Width)
[1] 0.1
```

Bet Bardají Bofill
SP lab session

Maximum and minum namber in variable Species:

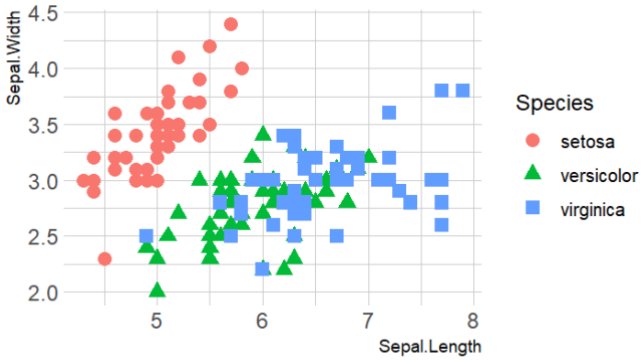
```
> sum(factor_vector == "setosa")
[1] 50
> sum(factor_vector == "versicolor")
[1] 50
> sum(factor_vector == "virginica")
[1] 50
```

As we can see, the number of species is proportional, equivalent(50 each).

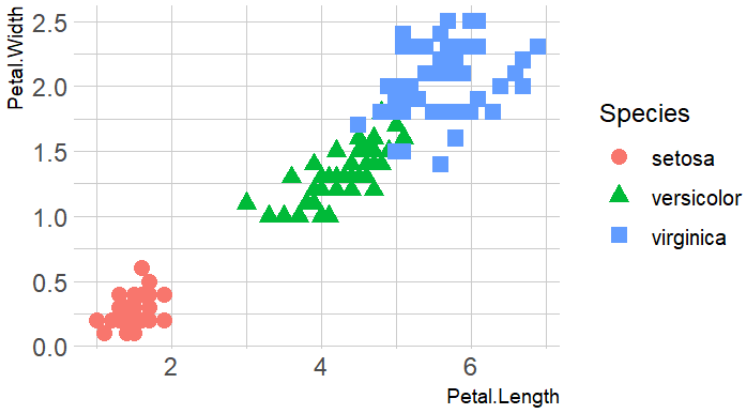
Summary(). Using this function, we obtain the max and minimum of all the variables. It is an fastest way compared with the ones used above.

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length
 Min.   :4.300   Min.   :2.000   Min.   :1.000
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
 Median :5.800   Median :3.000   Median :4.350
 Mean   :5.843   Mean   :3.057   Mean   :3.758
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
 Max.   :7.900   Max.   :4.400   Max.   :6.900
  Petal.Width          Species
 Min.   :0.100   setosa    :50
 1st Qu.:0.300   versicolor:50
 Median :1.300   virginica :50
 Mean   :1.199
 3rd Qu.:1.800
 Max.   :2.500
```

Graphical representation of the iris dataset



CODE:
> ggplot(iris,aes(x=Sepal.Length, y=Sepal.Width, shape=Species, color = Species))+ geom_point(size=3)+theme_ipsum ()



CODE

```
>ggplot(iris,aes(x=Petal.Length, y=Petal.Width, shape=Species, color = Species))+
geom_point(size=3)+theme_ipsum()
```

Please check the output of the data() function. Search for its help and explain its functionality. Search in the datasets package. Choose a dataset.

Here the goal is to understand how the data function works.
The data set function either loads specified data sets or returns a list of accessible data sets.

Input:

```
> help("data")
```

Output:

data {utils} R Documentation

Data Sets

Description

Loads specified data sets, or list the available data sets.

Usage

```
data(..., list = character(), package = NULL, lib.loc = NULL,
     verbose = getOption("verbose"), envir = .GlobalEnv,
     overwrite = TRUE)
```

Arguments

...	literal character strings or names.
list	a character vector.

Arguments

...	literal character strings or names.
list	a character vector.
package	a character vector giving the package(s) to look in for data sets, or NULL. By default, all packages in the search path are used, then the 'data' subdirectory (if present) of the current working directory.
lib.loc	a character vector of directory names of R libraries, or NULL. The default value of NULL corresponds to all libraries currently known.
verbose	a logical. If TRUE, additional diagnostics are printed.
envir	the environment where the data should be loaded.
overwrite	logical: should existing objects of the same name in envir be replaced?

Bet Bardají Bofill
SP lab session

Run examples

```
require(utils)
data()                          # list all available data sets
try(data(package = "rpart"), silent = TRUE) # list the data sets in the rpart package
data(USArrests, "VADeaths")     # load the data sets 'USArrests' and 'VADeaths'
## Not run: ## Alternatively
ds <- c("USArrests", "VADeaths"); data(list = ds)
## End(Not run)
help(USArrests)                 # give information on data set 'USArrests'
```

SOME PACKAGE DATASETS. I'M CHOOSING CHICKWEIGHT

Code:

```
> search()
 [1] ".GlobalEnv"        "tools:rstudio"     "package:stats"
 "package:graphics"  "package:grDevices"
 [6] "package:utils"     "package:datasets"  "package:methods"
 "Autoloads"         "package:base"
> help("data")
> library(help = "datasets")
```

```
                Information on package 'datasets'

Description:

Package:       datasets
Version:       4.2.1
Priority:      base
Title:         The R Datasets
               Package
Author:        R Core Team and
               contributors
               worldwide
Maintainer:    R Core Team
               <do-use-Contact-address@r-project.org>
Contact:       R-help mailing list
               <r-help@r-project.org>
Description:   Base R datasets.
License:       Part of R 4.2.1
Built:         R 4.2.1; ;
               2022-06-24 10:57:16
               UTC; unix

Index:

AirPassengers          Monthly Airline Passenger Numbers 1949-1960
BJsales                Sales Data with Leading Indicator
BOD                    Biochemical Oxygen Demand
CO2                    Carbon Dioxide Uptake in Grass Plants
ChickWeight            Weight versus age of chicks on different diets
DNase                  Elisa assay of DNase
EuStockMarkets         Daily Closing Prices of Major European Stock
                       Indices, 1991-1998
Formaldehyde           Determination of Formaldehyde
HairEyeColor           Hair and Eye Color of Statistics Students
Harman23.cor           Harman Example 2.3
Harman74.cor           Harman Example 7.4
Indometh               Pharmacokinetics of Indomethacin
InsectSprays           Effectiveness of Insect Sprays
JohnsonJohnson         Quarterly Earnings per Johnson & Johnson Share
LakeHuron              Level of Lake Huron 1875-1972
```

```
> search()
 [1] ".GlobalEnv"        "tools:rstudio"     "package:stats"
 "package:graphics"  "package:grDevices"
 [6] "package:utils"     "package:datasets"  "package:methods"
 "Autoloads"         "package:base"
> help("data")
> library(help = "datasets")
> library(help = "datasets")
> data("ChickWeight")
> head(ChickWeight)
  weight Time Chick Diet
1     42    0     1    1
2     51    2     1    1
3     59    4     1    1
4     64    6     1    1
5     76    8     1    1
6     93   10     1    1
```

I chose the chickweight dataset

5. Describe dataset size and type, including number of variables and observations.

ChickWeight it is a dataframe that compares the weight versus the age of chicks in order to study the effect of different diets. It presents 578 observations (rows) and 4 variables(columns) and a memory object of 22696 bytes. It is an object of class **c("nfnGroupedData", "nfGroupedData", "groupedData", "data.frame")** that contains 4 columns

Bet Bardají Bofill
SP lab session

```
> dim(ChickWeight)
[1] 578   4
```
→ with dim () function we find the dimensions of our dataset

```
> class(ChickWeight)
[1] "nfnGroupedData" "nfGroupedData"  "groupedData"
[4] "data.frame"
> object.size(ChickWeight)
22696 bytes

> length(as.matrix(ChickWeight))
[1] 2312

> data("ChickWeight")
> head(ChickWeight)
  weight Time Chick Diet
1     42    0     1    1
2     51    2     1    1
3     59    4     1    1
4     64    6     1    1
5     76    8     1    1
6     93   10     1    1
```

class () indicates the type of data is our dataset
object.size () indicates the memory bytes is using
length(as.matrix()) indicates the number of elements the data has, seen as a matrix

data and head() to input a dataset and head to show it.

Notice that the function head() shows us the first 6 rows of the dataset.

8. Try to describe the meaning of each variable and its type.

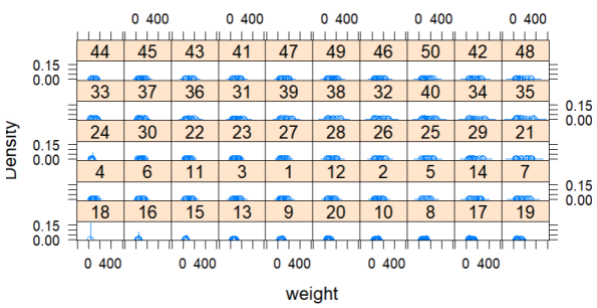
- **Weight:** the weight of every chick in gm. It is a numeric vector
- **Time**: it is a numeric vector, describes the age in days since birth when the measure was made
- **Chick:** it is an ordered factor that gives an unique identifier of chick studied. Presents different levels.
- **Diet:** It is a factor related to the type of diet the chick is eating. It has 4 levels

```
> class(ChickWeight$weight)
[1] "numeric"
> class(ChickWeight$Time)
[1] "numeric"
> class(ChickWeight$Chick)
[1] "ordered" "factor"
> class(ChickWeight$Diet)
[1] "factor"
```

The chicks' body weights were measured at birth and every other day until day 20. On day 21, they were also measured. Four groups of chicks were fed varied protein diets.

9. Provide with some statistical description of the dataset.

```
> dim(ChickWeight)
[1] 578   4
> summary(ChickWeight)
     weight           Time           Chick     Diet
 Min.   : 35.0   Min.   : 0.00   13     : 12   1:220
 1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120
 Median :103.0   Median :10.00   20     : 12   3:120
 Mean   :121.8   Mean   :10.72   10     : 12   4:118
 3rd Qu.:163.8   3rd Qu.:16.00   17     : 12
 Max.   :373.0   Max.   :21.00   19     : 12
                                 (Other):506
```



Bet Bardají Bofill
SP lab session

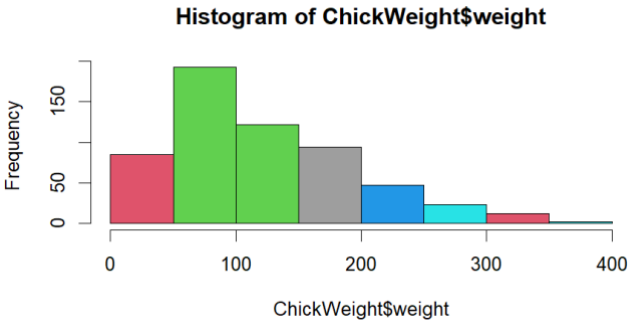
10. Agree with your teacher with three statistical questions to **answer with the dataset.**

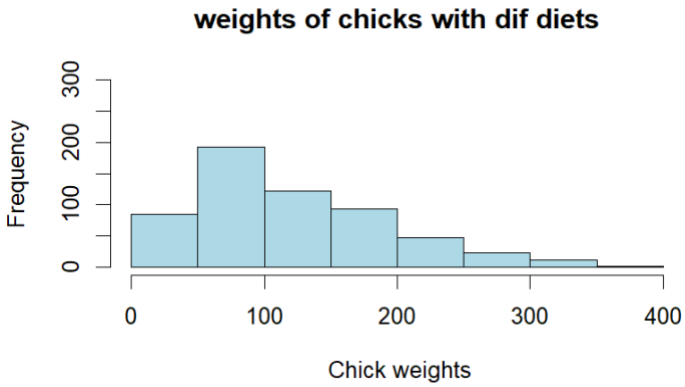
1.Which are the chickens that weight less.

```
> head(ChickWeight[order(ChickWeight$weight),])
    weight Time Chick Diet
196     35    2    18    1
26      39    2     3    1
195     39    0    18    1
293     39    0    27    2
305     39    0    28    2
317     39    0    29    2
```

2.Which are the mean of the columns weight and time. Does the weight follow a normal distribution?

```
> l <- list(ChickWeight$weight, ChickWeight$Time)
> l.mean <- sapply(l,mean)
> l.mean
[1] 121.81834  10.71799
```





CODE

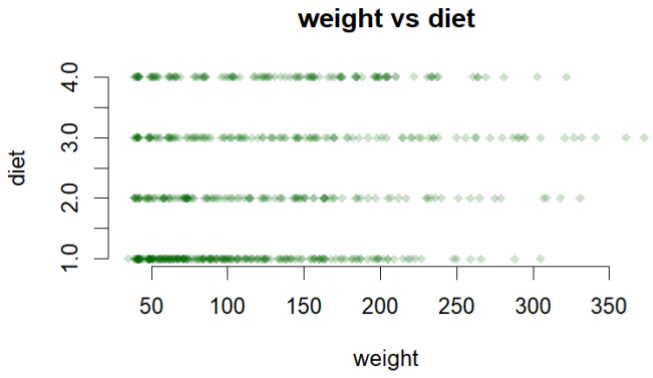
```
> hist(ChickWeight$weight, BREAKS = 8, main= "weights of chicks with dif diets", xlab = " Chick weights", col = ChcikWeight$Diets , prob = TRUE, ylim = c(0,300))

> hist(ChickWeight$weight, BREAKS = 8, main= "weights of chicks with dif diets", xlab = " Chick weights", col = "lightblue" , prob = TRUE, ylim = c(0,300))
```

The graph shows us that the distribution of the weight is not similar to a normal distribution. The mean and SD are most useful as summary statistics when the distribution is relatively symmetric. In this histogram the distribution is skewed left, so these 2 statistic parameters aren't the best in order to understand the behavior of the weight in this dataset

Bet Bardají Bofill
SP lab session

3.Compare the weight's mean and sd with the type of diet

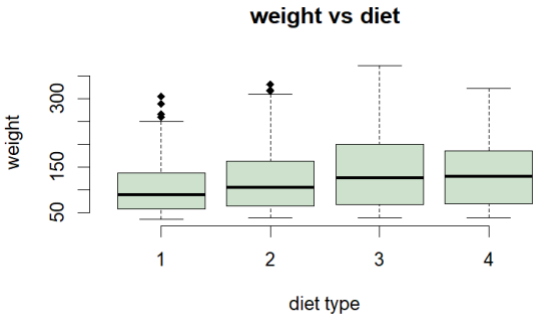
```
> by(ChickWeight$weight,ChickWeight$Diet,sd)
ChickWeight$Diet: 1
[1] 56.65655
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 2
[1] 71.60749
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 3
[1] 86.54176
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 4
[1] 68.82871
> by(ChickWeight$weight,ChickWeight$Diet,mean)
ChickWeight$Diet: 1
[1] 102.6455
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 2
[1] 122.6167
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 3
[1] 142.95
------------------------------------------------------------------------
------------------------------------
ChickWeight$Diet: 4
[1] 135.2627
```



X axis = diet
Y axis= weights

Code:

```
plot(ChickWeight$weight,Chic
kWeight$Diet, main = "weight
vs diet",ylab="diet",
xlab="weight",
col=rgb(0,100,0,50,maxColorV
alue = 255),pch=18,
frame=FALSE)
> library(ggplot2)
> library(hrbrthemes)
```



X axis = weight
Y axis= diets

Code:

```
plot(ChickWeight$Diet,ChickW
eight$weight, main = "weight
vs diet",ylab="weight",
xlab="diet",
col=rgb(0,100,0,50,maxColorV
alue = 255),pch=18,
frame=FALSE)
> library(ggplot2)
> library(hrbrthemes)
```

Bet Bardají Bofill
SP lab session