UNIVERSIDADE FEDERAL FLUMINENSE

Programa de Mestrado e Doutorado em Engenharia de Produção

Multivariate Data Analysis

Lesson: Descriptive Statistics with R

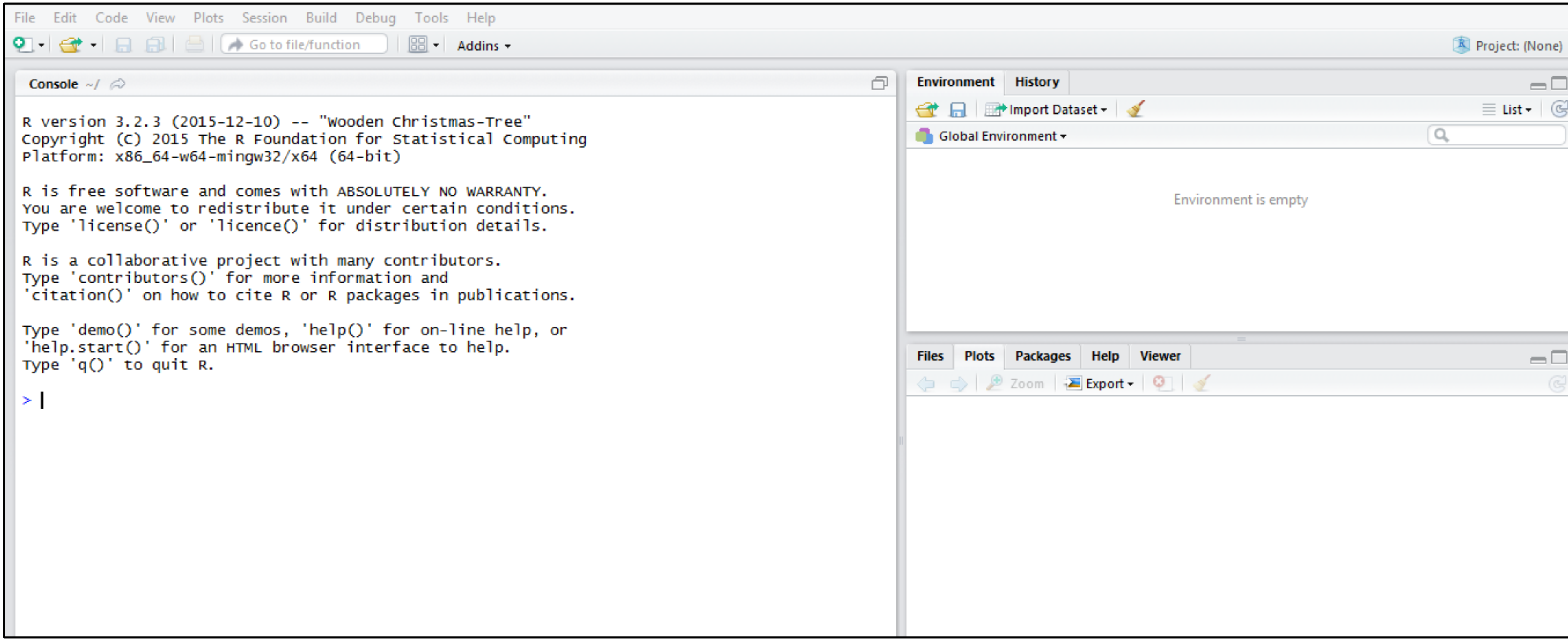
Professor: Valdecy Pereira, D. Sc.

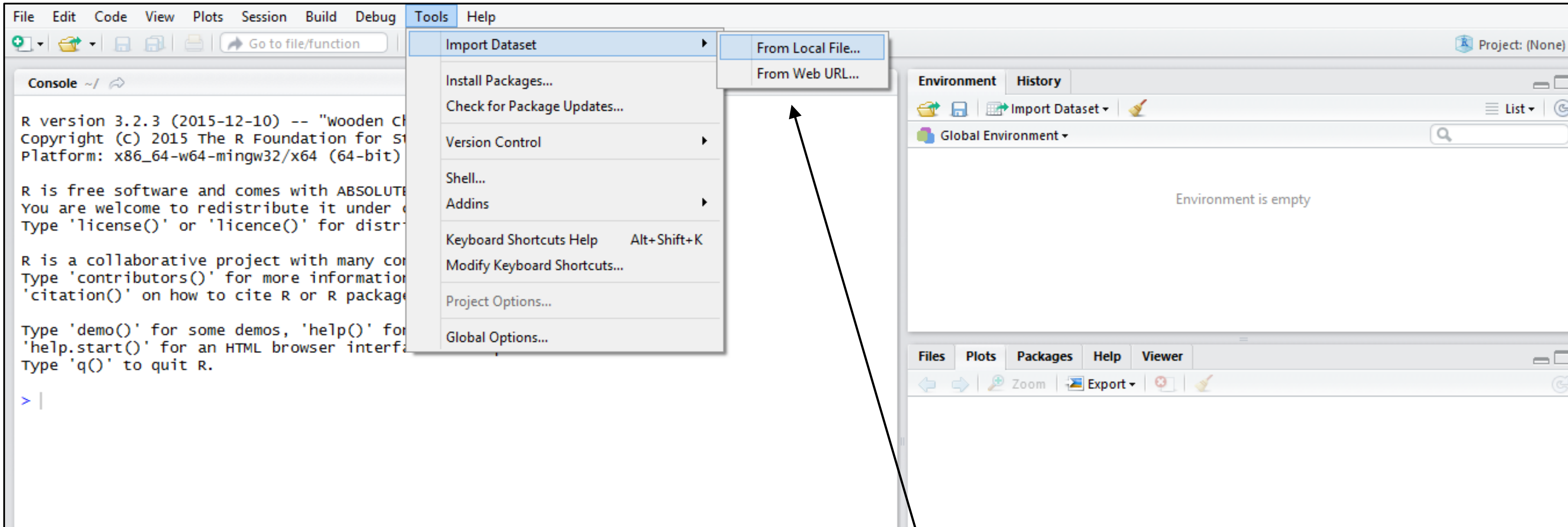
email: valdecy.pereira@gmail.com

Outline

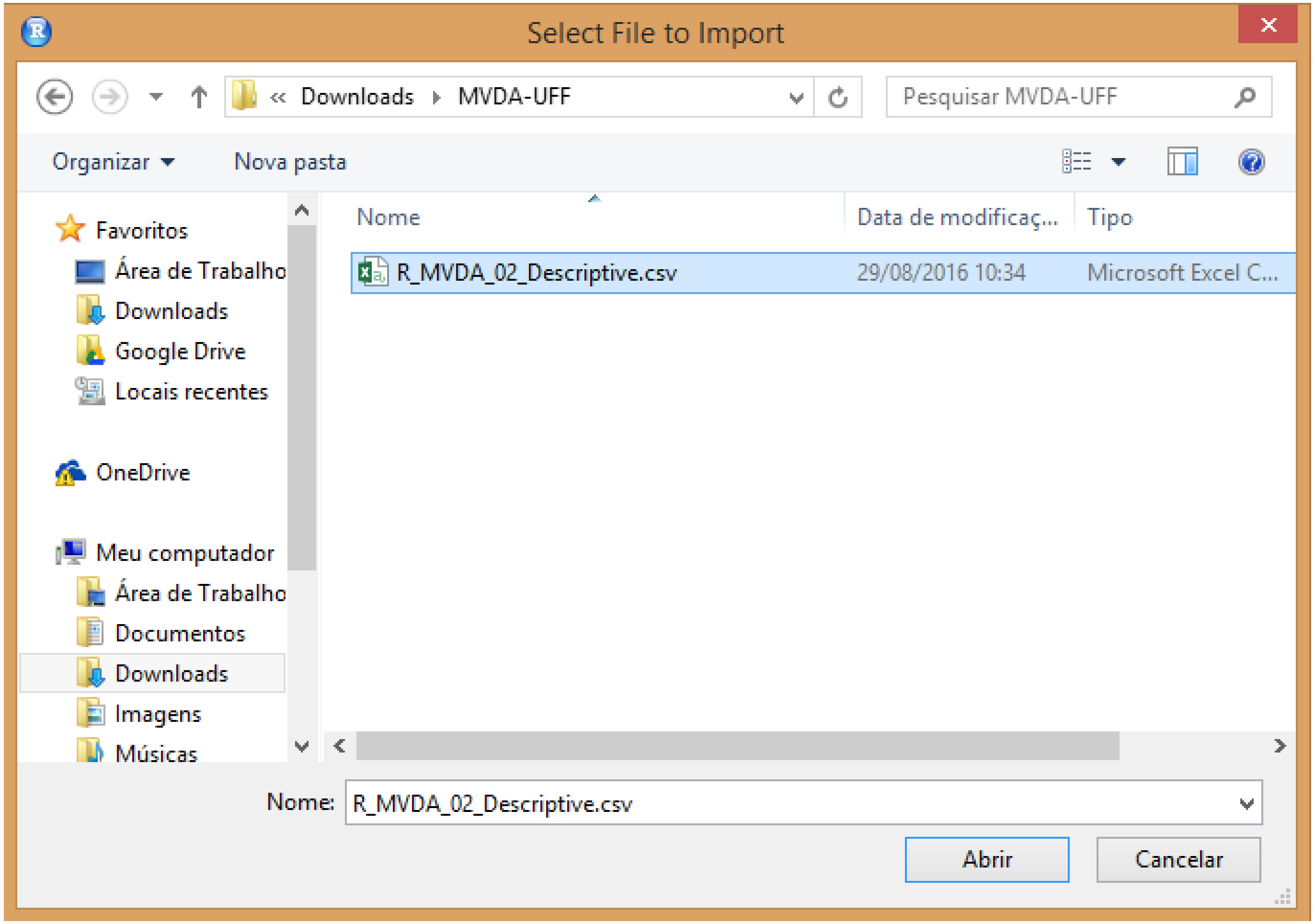
1. R-Studio
2. Class and Data
3. Packages
4. Descriptive Statistics
5. Normal Curve
6. Hypothesis Testing
7. Bibliography

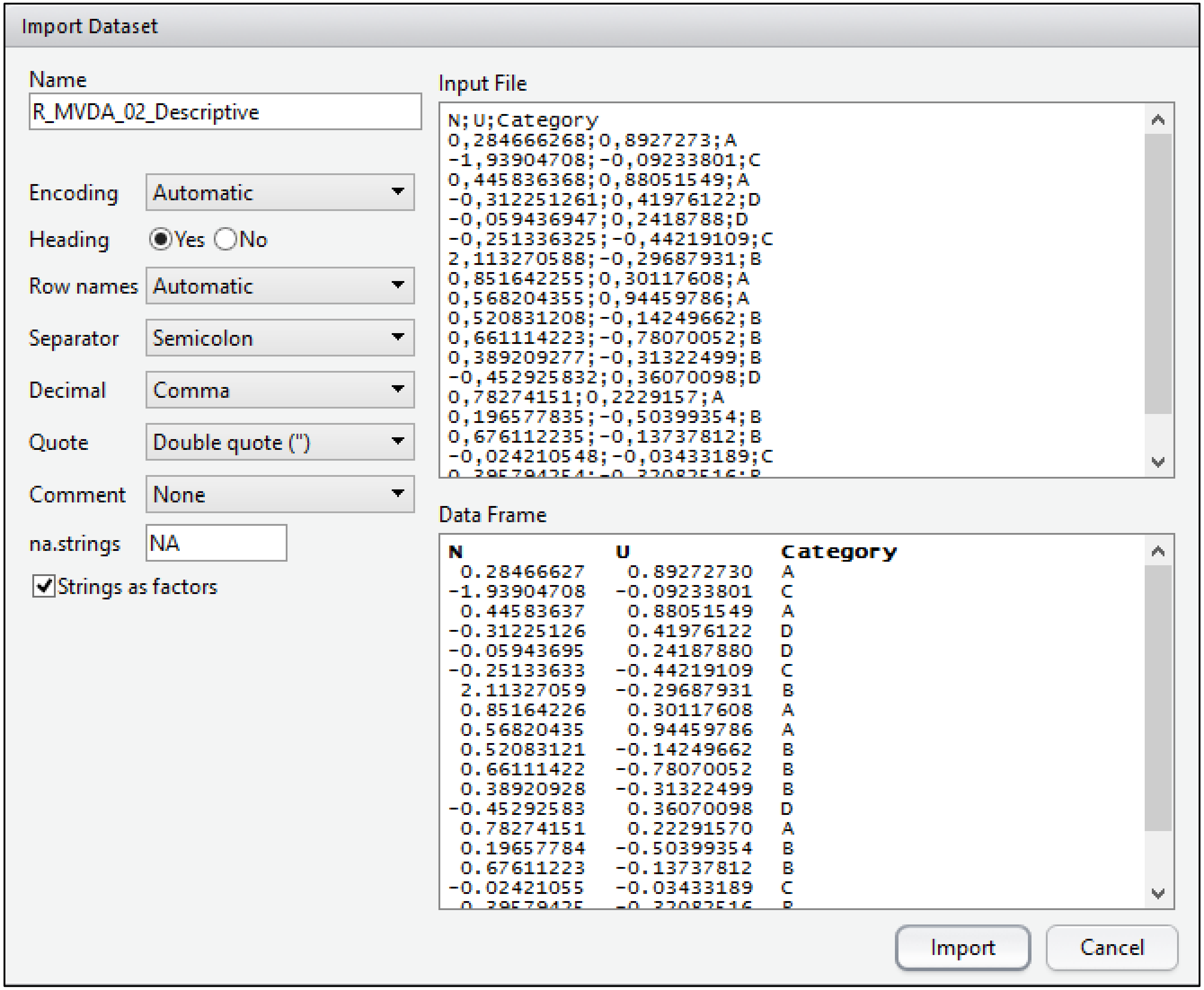
R-Studio

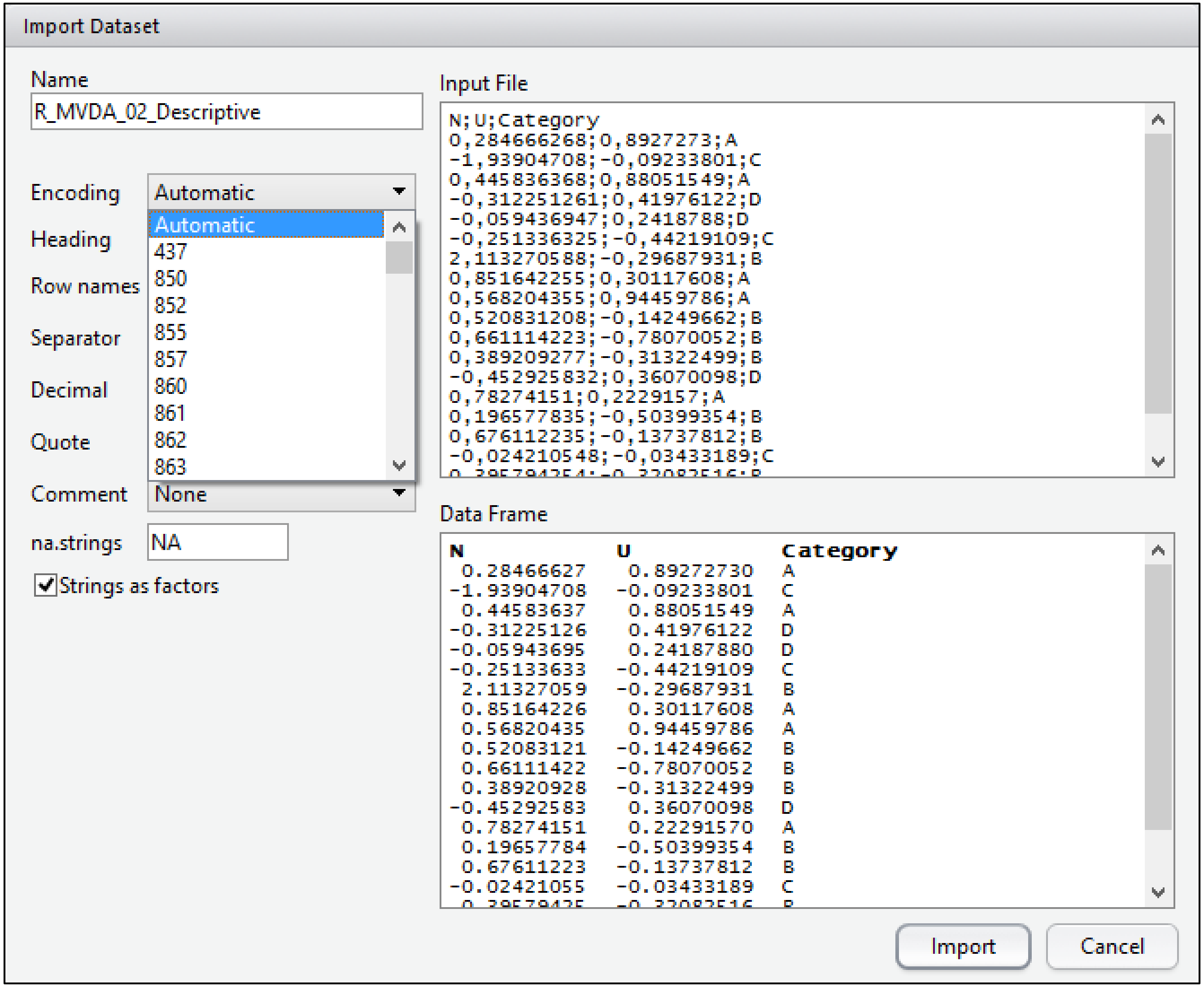

File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
Project: (None)
Console ~/
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment History
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
Zoom
Export

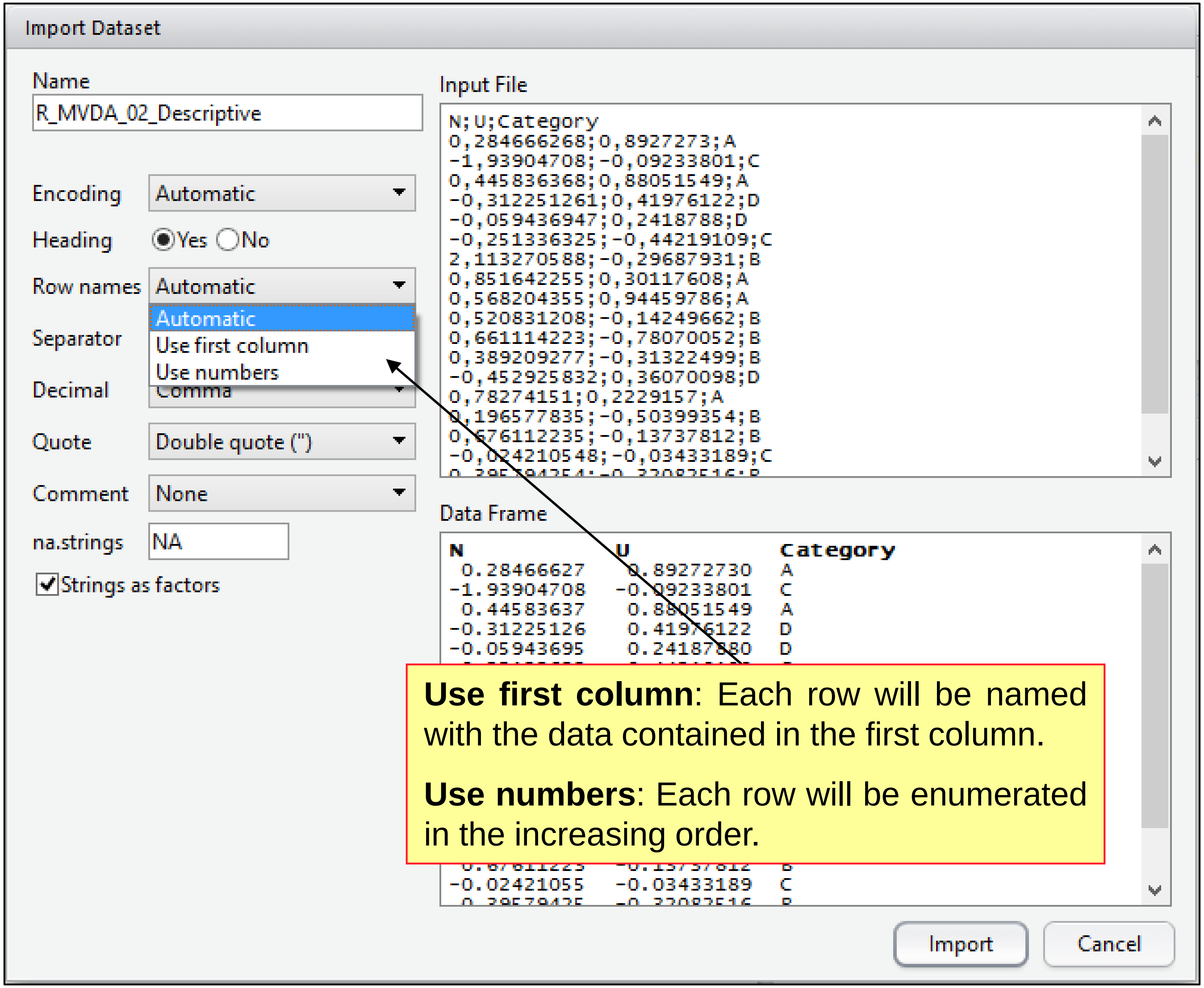


Importing data: Tools > Import Dataset


Select File to Import
Downloads
MVDA-UFF
Pesquisar MVDA-UFF
Organizar
Nova pasta
Favoritos
Área de Trabalho
Downloads
Google Drive
Locais recentes
OneDrive
Meu computador
Área de Trabalho
Documentos
Downloads
Imagens
Músicas
Nome
Data de modificaç...
Tipo
R_MVDA_02_Descriptive.csv
29/08/2016 10:34
Microsoft Excel C...
Nome:
R_MVDA_02_Descriptive.csv
Abrir
Cancelar

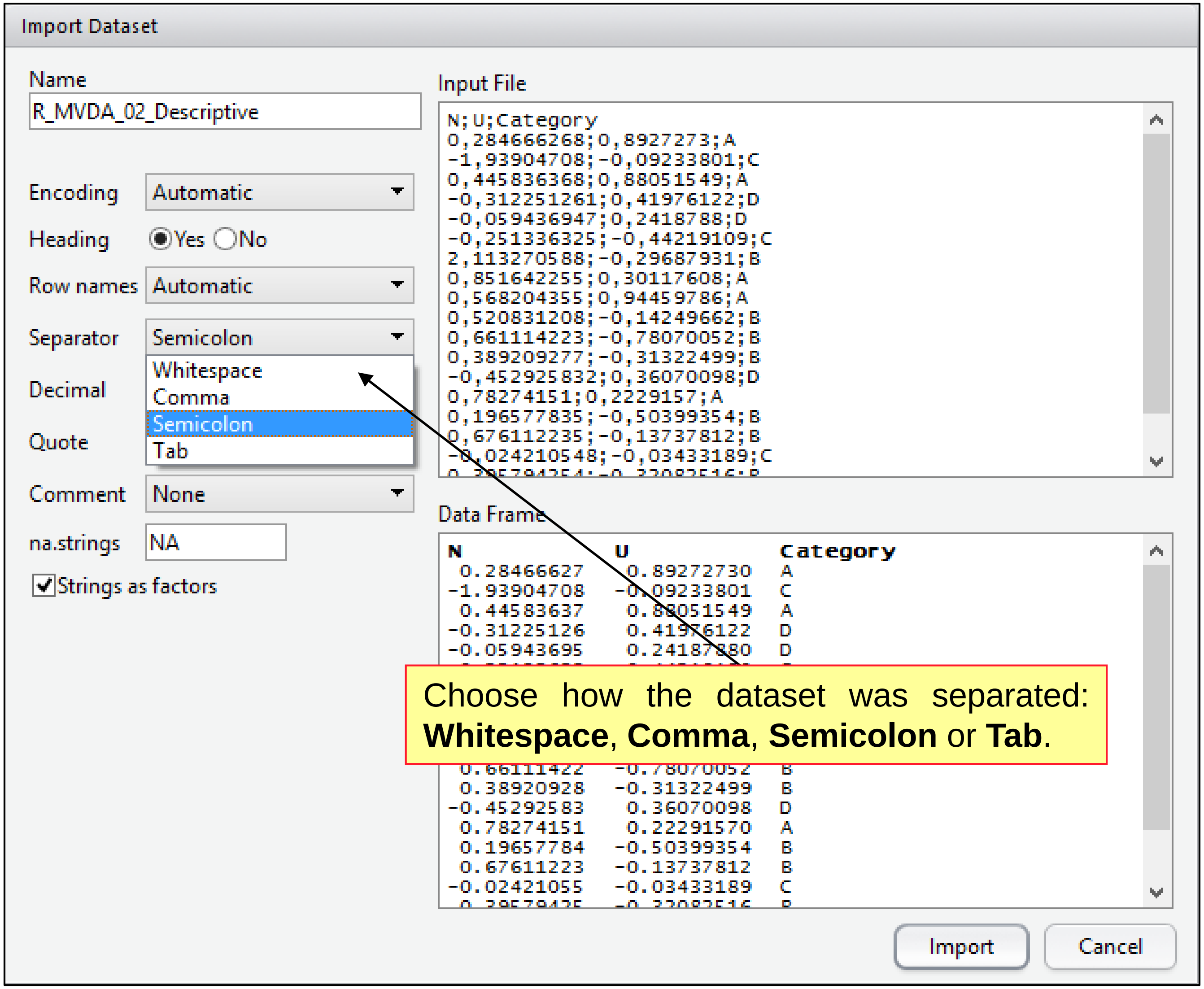

Import Dataset
Name
R_MVDA_02_Descriptive
Encoding
Automatic
Heading
Yes
No
Row names
Automatic
Separator
Semicolon
Decimal
Comma
Quote
Double quote (")
Comment
None
na.strings
NA
Strings as factors
Input File
N;U;Category
0,284666268;0,8927273;A
-1,93904708;-0,09233801;C
0,445836368;0,88051549;A
-0,312251261;0,41976122;D
-0,059436947;0,2418788;D
-0,251336325;-0,44219109;C
2,113270588;-0,29687931;B
0,851642255;0,30117608;A
0,568204355;0,94459786;A
0,520831208;-0,14249662;B
0,661114223;-0,78070052;B
0,389209277;-0,31322499;B
-0,452925832;0,36070098;D
0,78274151;0,2229157;A
0,196577835;-0,50399354;B
0,676112235;-0,13737812;B
-0,024210548;-0,03433189;C
Data Frame
N U Category
0.28466627 0.89272730 A
-1.93904708 -0.09233801 C
0.44583637 0.88051549 A
-0.31225126 0.41976122 D
-0.05943695 0.24187880 D
-0.25133633 -0.44219109 C
2.11327059 -0.29687931 B
0.85164226 0.30117608 A
0.56820435 0.94459786 A
0.52083121 -0.14249662 B
0.66111422 -0.78070052 B
0.38920928 -0.31322499 B
-0.45292583 0.36070098 D
0.78274151 0.22291570 A
0.19657784 -0.50399354 B
0.67611223 -0.13737812 B
-0.02421055 -0.03433189 C
Import
Cancel

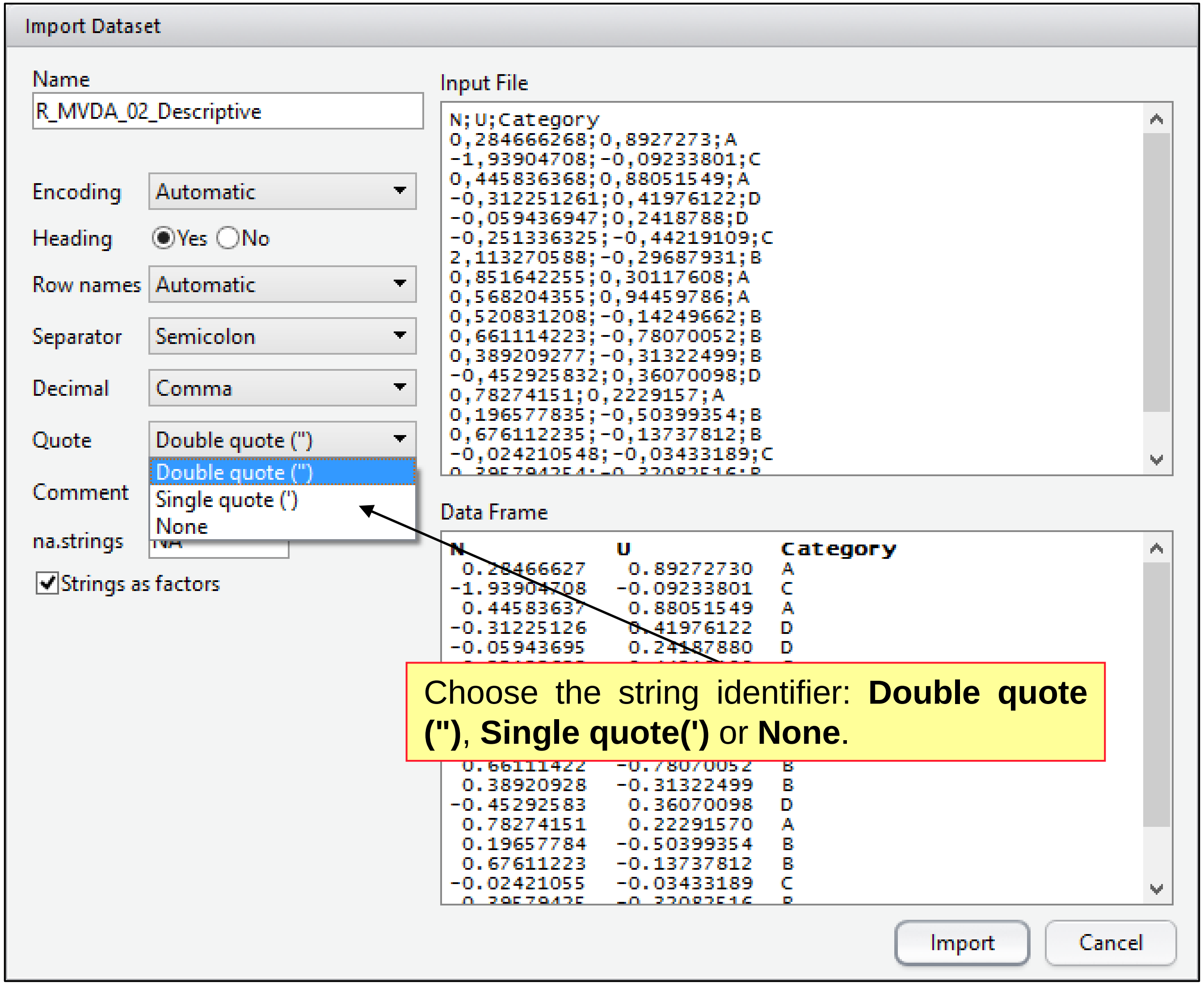

Import Dataset
Name
R_MVDA_02_Descriptive
Encoding
Automatic
Automatic
437
850
852
855
857
860
861
862
863
Heading
Row names
Separator
Decimal
Quote
Comment
None
na.strings
NA
Strings as factors
Input File
N;U;Category
0,284666268;0,8927273;A
-1,93904708;-0,09233801;C
0,445836368;0,88051549;A
-0,312251261;0,41976122;D
-0,059436947;0,2418788;D
-0,251336325;-0,44219109;C
2,113270588;-0,29687931;B
0,851642255;0,30117608;A
0,568204355;0,94459786;A
0,520831208;-0,14249662;B
0,661114223;-0,78070052;B
0,389209277;-0,31322499;B
-0,452925832;0,36070098;D
0,78274151;0,2229157;A
0,196577835;-0,50399354;B
0,676112235;-0,13737812;B
-0,024210548;-0,03433189;C
Data Frame
N U Category
0.28466627 0.89272730 A
-1.93904708 -0.09233801 C
0.44583637 0.88051549 A
-0.31225126 0.41976122 D
-0.05943695 0.24187880 D
-0.25133633 -0.44219109 C
2.11327059 -0.29687931 B
0.85164226 0.30117608 A
0.56820435 0.94459786 A
0.52083121 -0.14249662 B
0.66111422 -0.78070052 B
0.38920928 -0.31322499 B
-0.45292583 0.36070098 D
0.78274151 0.22291570 A
0.19657784 -0.50399354 B
0.67611223 -0.13737812 B
-0.02421055 -0.03433189 C
Import
Cancel

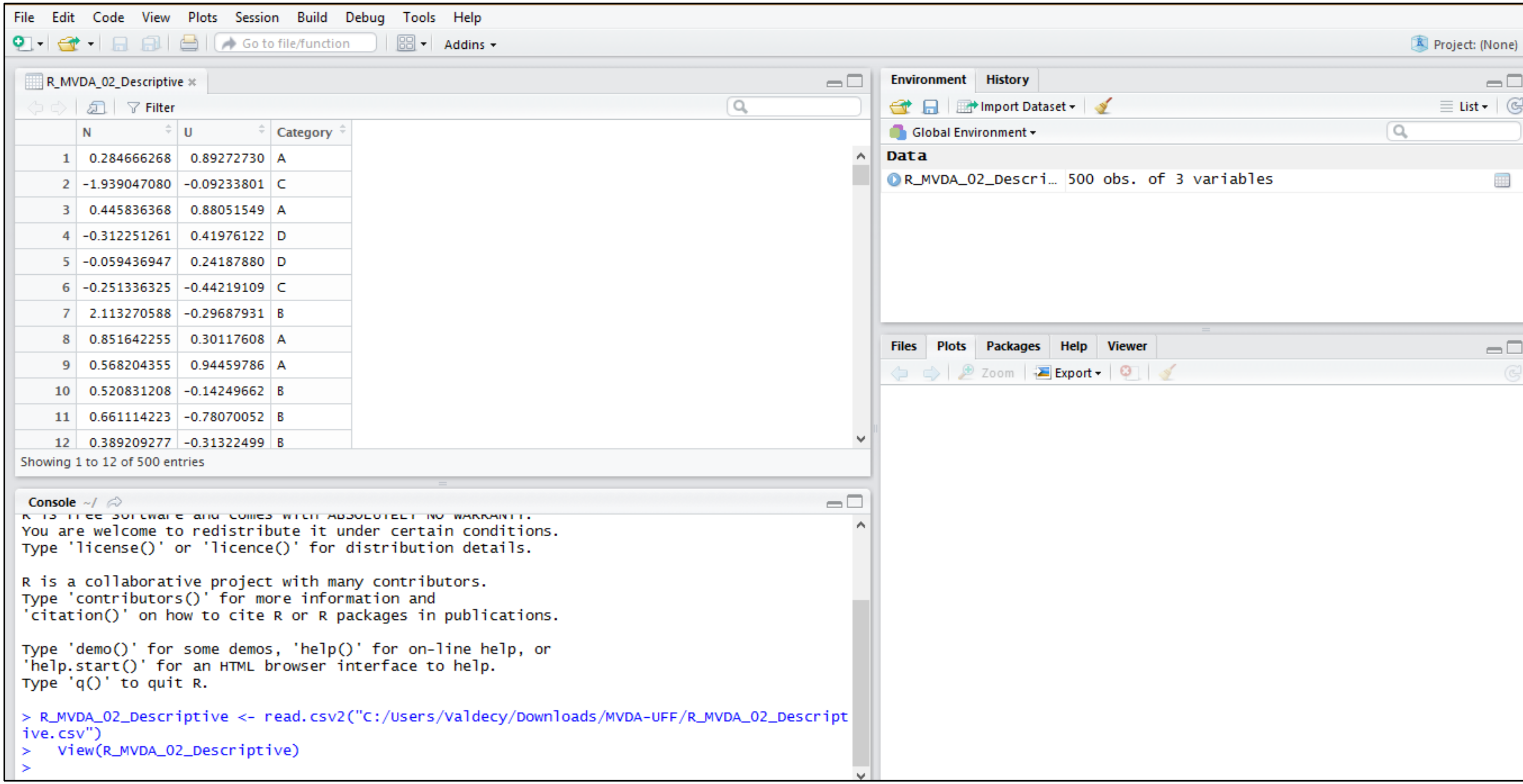


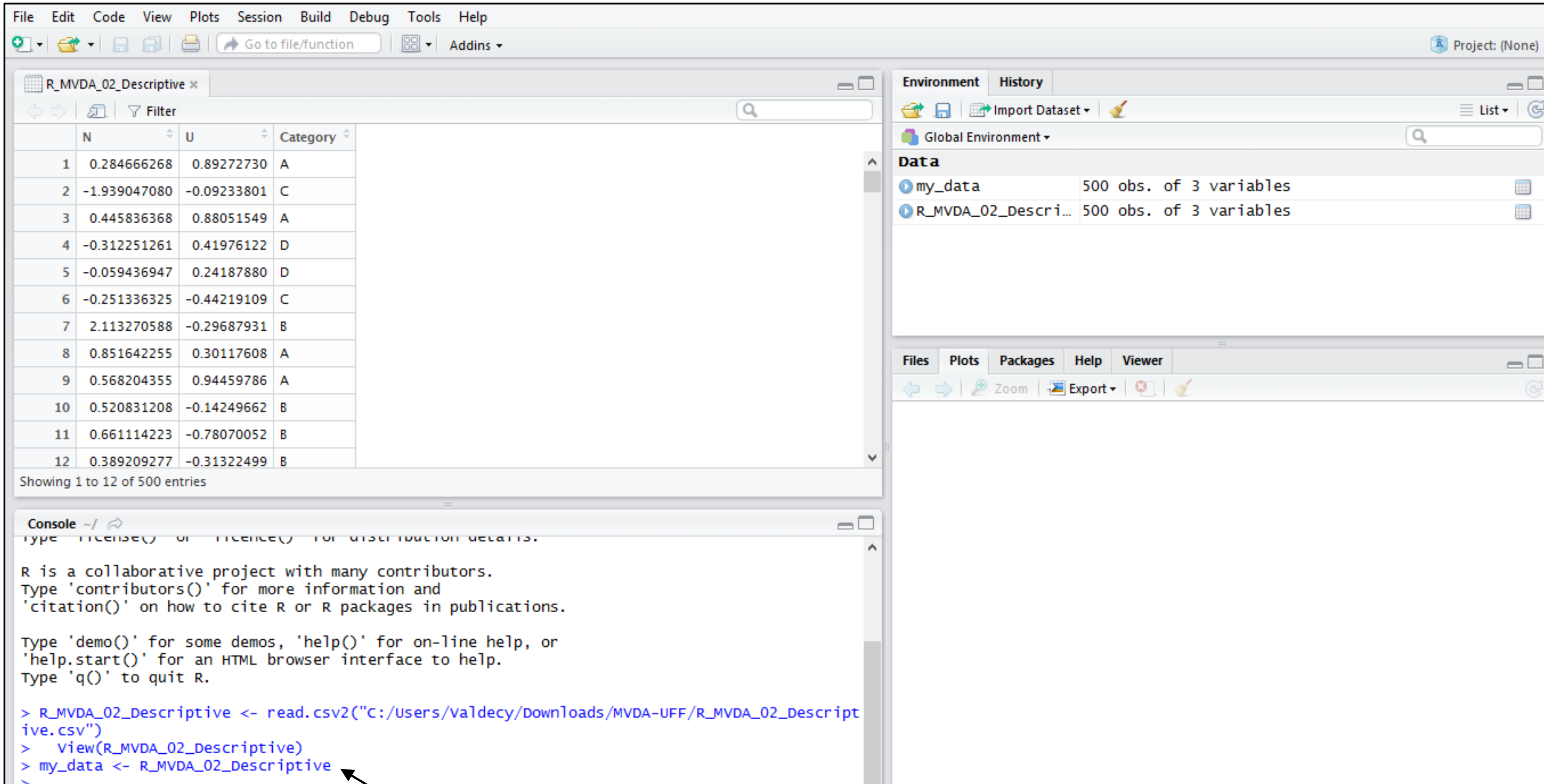
Use first column: Each row will be named with the data contained in the first column.

Use numbers: Each row will be enumerated in the increasing order.

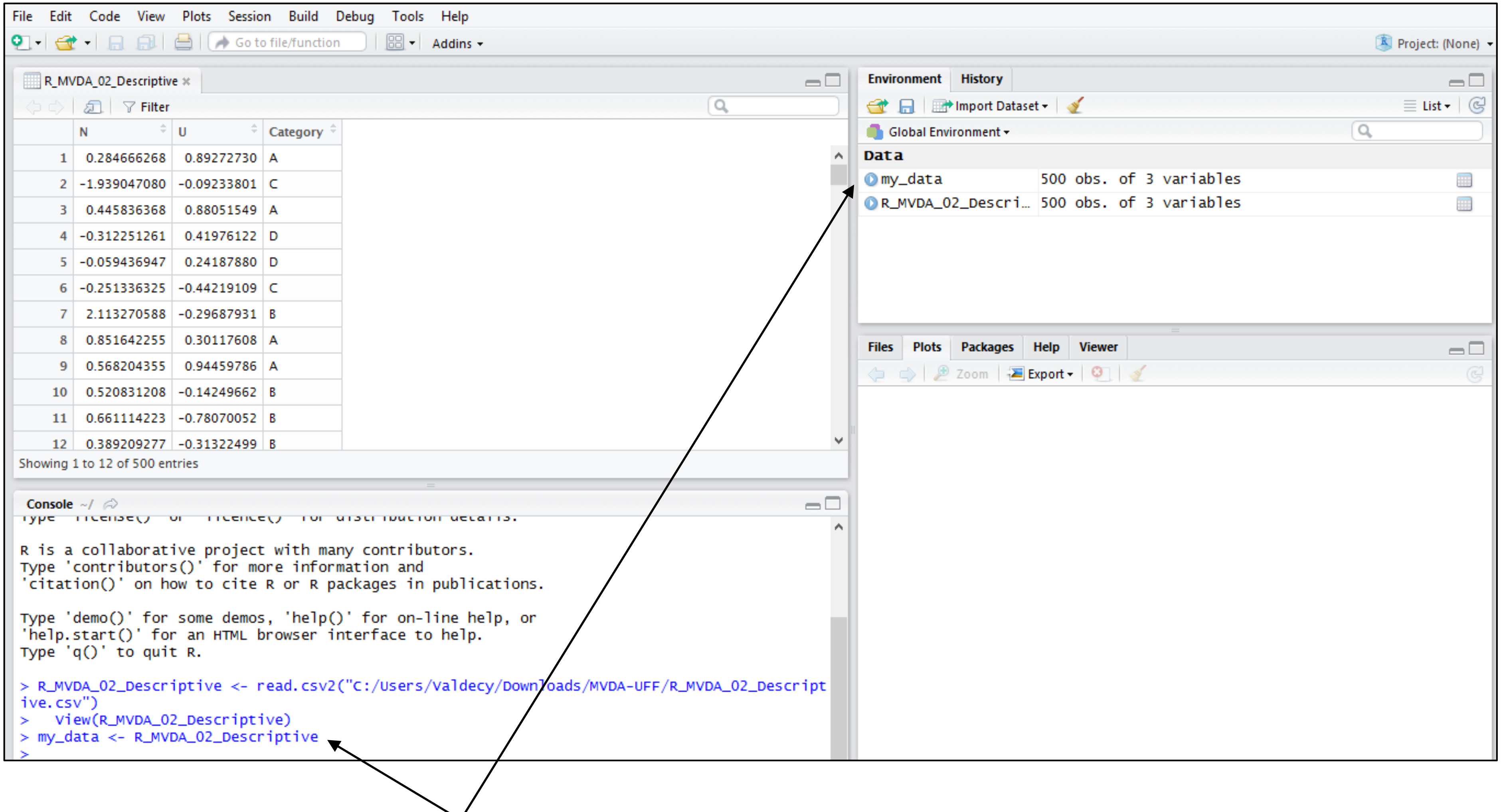

Import Dataset
Name
R_MVDA_02_Descriptive
Encoding
Automatic
Heading
Yes
No
Row names
Automatic
Separator
Semicolon
Whitespace
Comma
Semicolon
Tab
Decimal
Quote
Comment
None
na.strings
NA
Strings as factors
Input File
N;U;Category
0,284666268;0,8927273;A
-1,93904708;-0,09233801;C
0,445836368;0,88051549;A
-0,312251261;0,41976122;D
-0,059436947;0,2418788;D
-0,251336325;-0,44219109;C
2,113270588;-0,29687931;B
0,851642255;0,30117608;A
0,568204355;0,94459786;A
0,520831208;-0,14249662;B
0,661114223;-0,78070052;B
0,389209277;-0,31322499;B
-0,452925832;0,36070098;D
0,78274151;0,2229157;A
0,196577835;-0,50399354;B
0,676112235;-0,13737812;B
-0,024210548;-0,03433189;C
Data Frame
N U Category
Import
Cancel
Choose how the dataset was separated: **Whitespace**, **Comma**, **Semicolon** or **Tab**.


Import Dataset
Name
R_MVDA_02_Descriptive
Encoding
Automatic
Heading
Yes
No
Row names
Automatic
Separator
Semicolon
Decimal
Comma
Quote
Double quote (")
Double quote (")
Single quote (')
None
Comment
na.strings
Strings as factors
Input File
N;U;Category
0,284666268;0,8927273;A
-1,93904708;-0,09233801;C
0,445836368;0,88051549;A
-0,312251261;0,41976122;D
-0,059436947;0,2418788;D
-0,251336325;-0,44219109;C
2,113270588;-0,29687931;B
0,851642255;0,30117608;A
0,568204355;0,94459786;A
0,520831208;-0,14249662;B
0,661114223;-0,78070052;B
0,389209277;-0,31322499;B
-0,452925832;0,36070098;D
0,78274151;0,2229157;A
0,196577835;-0,50399354;B
0,676112235;-0,13737812;B
-0,024210548;-0,03433189;C
Data Frame
N U Category
0.28466627 0.89272730 A
-1.93904708 -0.09233801 C
0.44583637 0.88051549 A
-0.31225126 0.41976122 D
-0.05943695 0.24187880 D
0.38920928 -0.31322499 B
-0.45292583 0.36070098 D
0.78274151 0.22291570 A
0.19657784 -0.50399354 B
0.67611223 -0.13737812 B
-0.02421055 -0.03433189 C
Choose the string identifier: Double quote ("), Single quote(') or None.
Import
Cancel


File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
Project: (None)
R_MVDA_02_Descriptive
Filter
N U Category
1 0.284666268 0.89272730 A
2 -1.939047080 -0.09233801 C
3 0.445836368 0.88051549 A
4 -0.312251261 0.41976122 D
5 -0.059436947 0.24187880 D
6 -0.251336325 -0.44219109 C
7 2.113270588 -0.29687931 B
8 0.851642255 0.30117608 A
9 0.568204355 0.94459786 A
10 0.520831208 -0.14249662 B
11 0.661114223 -0.78070052 B
12 0.389209277 -0.31322499 B
Showing 1 to 12 of 500 entries
Environment History
Import Dataset
List
Global Environment
Data
R_MVDA_02_Descri… 500 obs. of 3 variables
Files Plots Packages Help Viewer
Zoom
Export
Console ~/
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> R_MVDA_02_Descriptive <- read.csv2("C:/Users/Valdecy/Downloads/MVDA-UFF/R_MVDA_02_Descriptive.csv")
> View(R_MVDA_02_Descriptive)
>



my_data <- R_MVDA_02_Descriptive



my_data <- R_MVDA_02_Descriptive

Class Type - Vector

MVDA – *Class Type*

A $VECTOR$ is a collection of ordered homogeneous elements.

```
# Numeric Vector
a <- c(1, 3.33, 8.87, 6, -2, 7)
# To retrieve the elements of vector use [ ]
a [ c (2,4) ]
[1] 3.33  6.00


a [ c (1:4) ]
[1] 1.00  3.33  8.87  6.00

# Character (String) Vector
b <- c("alpha", "bravo")

# Logical Vector
c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)  # or #  c <- c (T, T, T, F, T, F)
*****************************************************************************************************************
length(a)
[1] 6
names(a)  <- c("A", "B", "C", "D", "E", "F")
   A      B      C      D       E      F
1.00  3.33  8.87  6.00  -2.00  7.00
```

Class Type - Matrix

MVDA – *Class Type*

A $MATRIX$ is a $VECTOR$ with two-dimensional shape. The information contained must be of the same type (numeric, character or logic).

```
# Matrix 5 x 4 (5 rows and 4 columns) with a sequence of numbers from 1 to 20.
matrix(1:20, nrow = 5, ncol = 4, byrow = FALSE)

     [,1] [,2] [,3]  [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20

matrix(1:20, nrow = 5, ncol = 4, byrow = TRUE)

     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
```

MVDA – *Class Type*

```
# Another example
my_matrix <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2, byrow = TRUE)
rownames(my_matrix) <- c ("1st_row", "2nd_row")
colnames(my_matrix) <- c ("column_A", "column_B")

        column_A   column_B
1st_row        1          2
2nd_row        3          4

my_matrix[2,1]
[1] 3
*********************************************************************************************************
length(my_matrix)
[1] 4

dim(my_matrix)
[1] 2  2

my_matrix[ -1,  ]
column_A   column_B
       3          4
```

MVDA – *Class Type*

```
my_matrix <- cbind(my_matrix, c(5, 6))
colnames(my_matrix)[ 3 ] <- c("column_C")
```

	column_A	column_B	column_C
1st_row	1	2	5
2nd_row	3	4	6

```
my_matrix <- rbind(my_matrix, c(7, 8, 9))
rownames(my_matrix)[ 3 ] <- c("3rd_row")
```

	column_A	column_B	column_C
1st_row	1	2	5
2nd_row	3	4	6
3rd_row	7	8	9

Class Type - Data Frame

MVDA – *Class Type*

A *DATA-FRAME* is a general form of a *MATRIX*, and the information contained can be from different types (numeric, character or logic).

```
my_data_frame <- data.frame (c (1, 2, 3, 4), c ("one", "two", "five", "ten"), c (T, T, T, F))
colnames(my_data_frame) <- c("A", "B", "C")
rownames(my_data_frame) <- c("r1", "r2", "r3", "r4")
```

	A	B	C
r1	1	one	TRUE
r2	2	two	TRUE
r3	3	five	TRUE
r4	4	ten	FALSE

```
# my_data_frame [ ,1] ; my_data_frame$A  or my_data_frame["A"] returns the values from the first column
```

To add, remove or name rows and columns in a *DATA-FRAME*, consult *MATRIX* commands for the same purpose.

Class Type - List

MVDA – *Class Type*

A ***LIST*** is an ordered collection of objects.

```
my_list <- list(my_matrix, my_data_frame, b)
[ [1] ]
            column_A   column_B   column_C
  1st_row      1          2          5
  2nd_row      3          4          6
  3rd_row      7          8          9

[ [2] ]
[1] "alpha"  "bravo"

my_list[[2]]
[ [2] ]
[1] "alpha"  "bravo"

my_list[[2]][1]
[1] "alpha"
```

Scales

MVDA – *Scales*

Scale	Type	R
Nominal	Non-Metric	Factor
Ordinal	Non-Metric	Factor
Interval	Metric	Numeric
Ratio	Metric	Numeric

- **Non-metric** data are attributes, characteristics or categories that identify or describe an observation.

- **Metric** data are precision measures and the differences between scale points can be made.

MVDA – *Scales*

NOMINAL

- The numbers serve as labels to name, identify, classify and (or) categorize data on persons, objects, events or facts.

- Ex: Vehicle plate. The numbers have no meaning unless identify the person or number associated with the object.

- Descriptive Statistics: **Mode**.

```
# Strings are converted automatically in factors
factor(c("small", "medium", "large"), levels = c("small", "medium", "large"))
[1] small  medium large
Levels: small medium large
```

MVDA – *Scales*

ORDINAL

- It represents an order relation between objects. An ordinal scale is one in which the numbers are in addition to name, identify, classify, also to order, according to a comparison process, people, objects or events in a certain characteristic.

- Ex: Rank - 1st place, 2nd place, 3rd place, etc.

- Descriptive Statistics: **Mode** and **Median**.

```
# Orders are created from the lowest to the greater attribute.
ordered(c("small", "medium", "large"), levels = c("small", "medium", "large"))
[1] small  medium large
Levels: small < medium < large
```

MVDA – *Scales*

INTERVAL

- Numbers sort the objects such that the distance (range) between them correspond to the distances between objects, people or events in the characteristic being measured, although there is an arbitrary zero.

- Ex: Celsius, Fahrenheit, etc.

- Descriptive Statistics: **Mode**, **Median** and **Mean**.

MVDA – *Scales*

RATIO

- They have the same characteristics as the INTERVAL scale, with the advantage of having absolute zero.

- Ex: Age, income, sales of a product, market share, cost, number of consumers, weight, height, distance, etc.

- Descriptive Statistics: **Mode**, **Median** and **Mean**.

Packages

MVDA – *Packages*

Packages are a set of functions made by R users or companies, having as the main objective, extending the capabilities of R.

```
# Downloading and installing a Package
install.packages("package name")

# Accessing a Package
library("package name")
```

Useful Codes

MVDA – *Codes*

str(object)	# structure of an object
head(object)	# object's first 6 rows
tail(object)	# object's last 6 rows
summary(object)	# summarize an object
sort(object)	# sort an object
help(command)	# help
as.numeric(object)	# transform to numeric
as.factor(object)	# transform to factor
as.matrix(object)	# transform to matrix
as.vector(object)	# transform to vector
as.data.frame(object)	# transform to data frame
as.list(object)	# transform to list
ls()	# list objects
rm(object)	# delete an object
fix(object)	# edit object
View(object)	# View object in a window

MVDA – *Codes*

```r
# For
for (i in 1:3){
  print(i)
}
[1] 1
[1] 2
[1] 3
```

```r
# If-then-else
x <- 0
if (x < 0) {
   print("Negative number")
} else if (x > 0) {
   print("Positive number")
} else
   print("Zero")
[1] "Zero"
```

```r
# While
x <- 1
while(x < 5) {
   x <- x + 1
   print(x)
}
[1] 2
[1] 3
[1] 4
```

MVDA – *Codes*

Logic Operator	Description
`<`	# less than
`<=`	# less than or equal to
`>`	# greater than
`>=`	# greater than or equal to
`==`	# exactly equal to
`!=`	# not equal to
`!x`	# not x
`x | y`	# x OR y
`x & y`	# x AND y

Descriptive Statistics

MVDA - *Mean*

The **Arithmetic Mean**, **Mean** or **Average** is the sum of a collection of numbers divided by the number of members of the same collection. It is often used to report central tendencies, however it is not a robust statistic, meaning that it is greatly influenced by outliers.

$$Mean = \frac{\sum x}{n}$$

```
# Arithmetic mean for dataset that holds a sequence from 1 to 100
sum(1:100)/length(seq(1:100))
[1]  50.5

mean(x = 1:100) # mean(1:100)  also works!
[1]  50.5
```

MVDA - *Mean*

In the **Weighted Mean** each element of the set may have different importance (weight), and in this case the calculation should take into account the weights of each element.

Tests	Grade	Weigth
T1	80	0.30
T2	90	0.30
T3	96	0.40

$$\overline{x}_p = \frac{\sum_{i=1}^{n} x_i p_i}{\sum_{i=1}^{n} p_i}$$

```
# Weighted mean for dataset in the example
weighted.mean(x = c(80, 90, 96), w = c(0.3, 0.3, 0.4))
[1]  89.4
```

MVDA - *Mean*

The **Trimmed Mean** is obtained by eliminating the data set the "m" largest and "m" lower values. Usually "m" corresponds from 2.5% to 5% of the observed values that may be considered as outliers.

```
# Trimmed mean for data set that holds a sequence from 1 to 100 and the value 5897 as an outlier
mean(x = c(1:100, 5897))
[1]  108.39

mean(x = c(1:100, 5897), trim = 0.10) # Both sides are trimmed
[1]  51

mean(x = 2:100) # This is the same as reducing de data from 2 to 100
[1]  51
```

MVDA - *Median*

The **Median** of a set of values is the middle value of this set when they are in increasing order, dividing it in half.

```
# Median
median(x = c(3, 7, 5, 5, 1, 9, 15, 13, 17, 13, 11))
[1]  9
```

Median

01, 03, 05, 05, 07, **9** 11, 13, 13, 15, 17

MVDA - *Mode*

The **Mode** of a data set is the one which is has the higher frequency. It can not exist, or it may not be unique.

```
# Mode
mode_table <- table(c(3, 7, 5, 5, 1, 9, 15, 13, 17, 13, 17))

 1  3  5  7  9 13 15 17
 1  1  2  1  1  2  1  2

names(mode_table)[mode_table  == max(mode_table)]

[1] "5"  "13"  "17"
```

MVDA - *Dispersion*

The **Variance** and the **Standard Deviation** are the most widely used measures of dispersion. The **Standard Deviation** is most commonly used because its result is in the same unit of the studied variable, while the **Variance** results are squared.

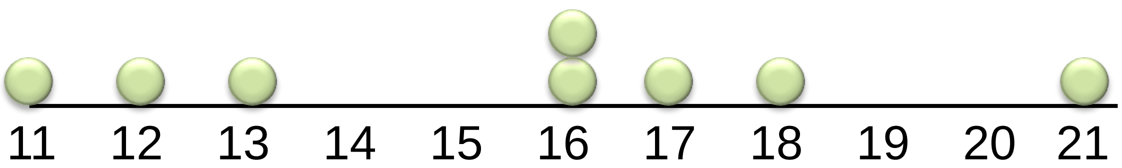
x_i	$\bar{x}$	$(x_i - \bar{x})^2$
2	6	16
4	6	4
6	6	0
8	6	4
10	6	16

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

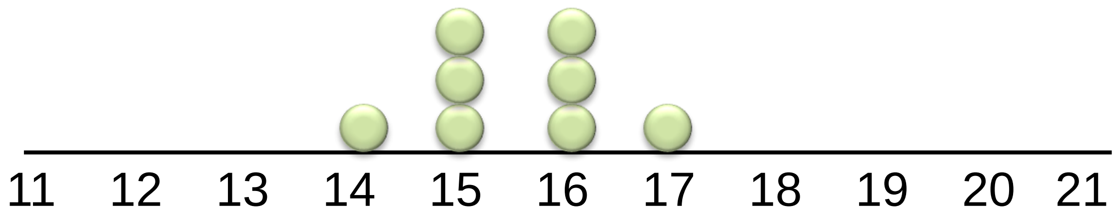
$$\hat{\sigma} = \sqrt{\hat{\sigma}^2}$$

```
# Variance
var(c(2, 4, 6, 8, 10))
[1] 10

# Standart Deviation
sd(c(2, 4, 6, 8, 10))
[1] 3.16
```

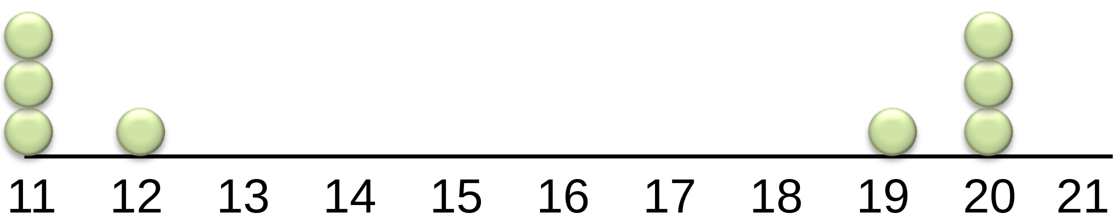
MVDA - *Dispersion*



$$\bar{x} = 15{,}5$$
$$\hat{\sigma} = 3{,}338$$



$$\bar{x} = 15{,}5$$
$$\hat{\sigma} = 0{,}926$$



$$\bar{x} = 15{,}5$$
$$\hat{\sigma} = 4{,}570$$

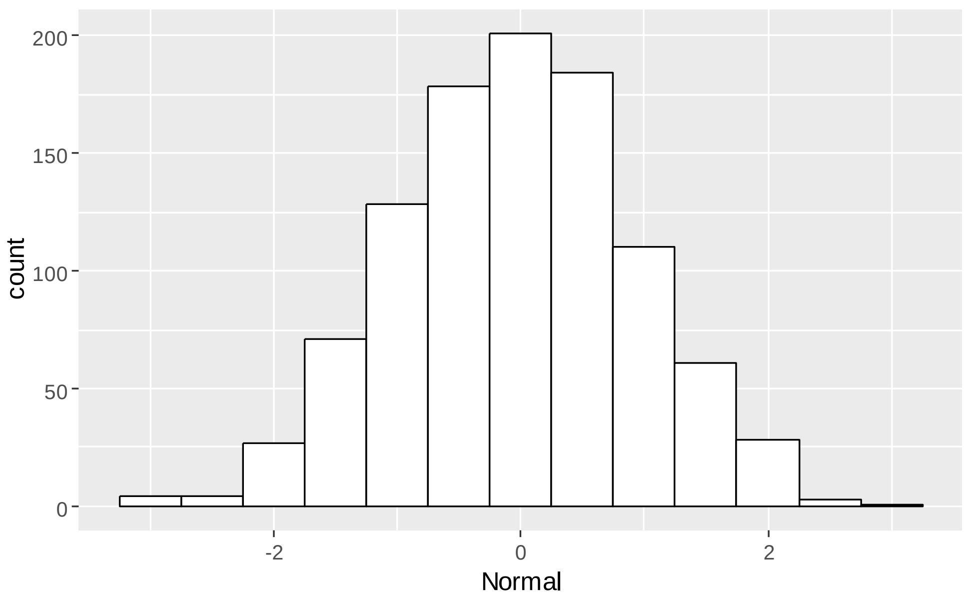
MVDA – *Normal Distribution*

A **Normal Distribution** is fully described by its mean and standard deviation parameters, that is, knowing these values any probability can be determined in a normal distribution. Also **Central Limit Theorem** states that the sum of independent random variables is approximately Normally limited, provided that the number of variables of the sum is large enough (≥ 50).

```
# Generate a data set with 1000 observations with mean 0 and standart deviation 1.
set.seed(101)
n_data <- as.data.frame(rnorm(1000, 0, 1))  # or n_data <- as.data.frame(rnorm(n = 1000, mean = 0, sd = 1))

# Central Limit Theorem
cl <- data.frame(runif(500))
for (i in 1:50) {
    cl[ , i] <- data.frame(runif(500))
}
cl$Sum <- rowSums(cl)

library("ggplot2")
ggplot(data = cl, aes( x = Sum)) + geom_density() + xlab("Normal")
```

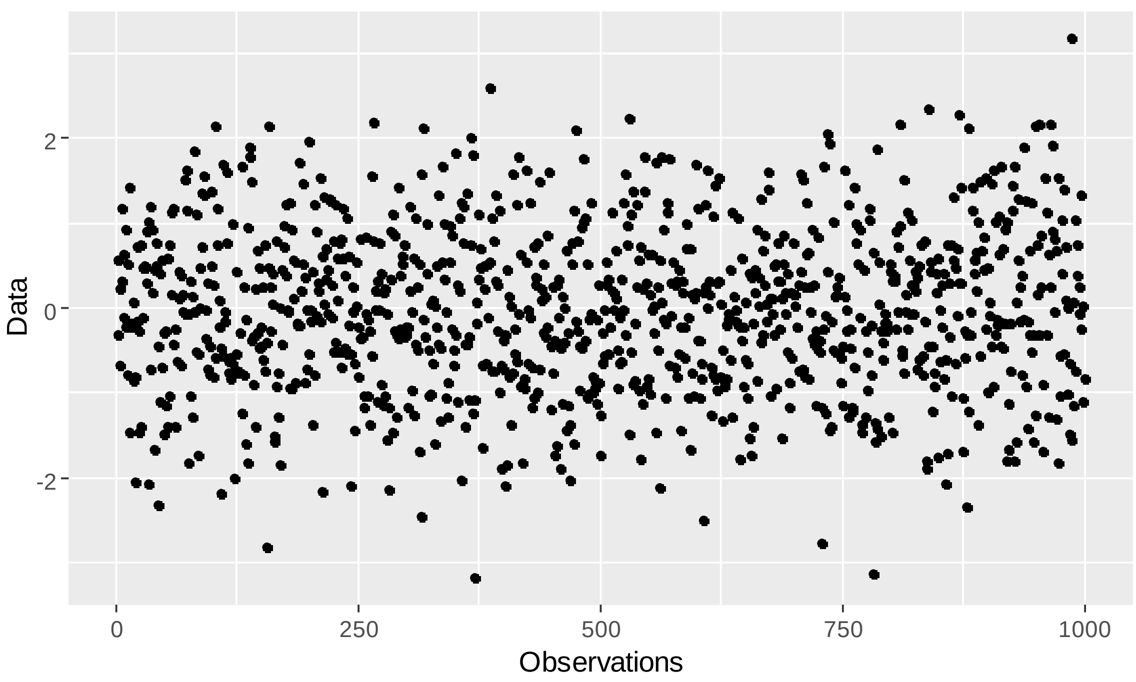
MVDA – *Normal Distribution*

```
# histogram
library("ggplot2")
ggplot(data = n_data, aes( x = n_data[,1])) + geom_histogram(binwidth = 0.5, colour = "black", fill = "white") + xlab("Normal")
```



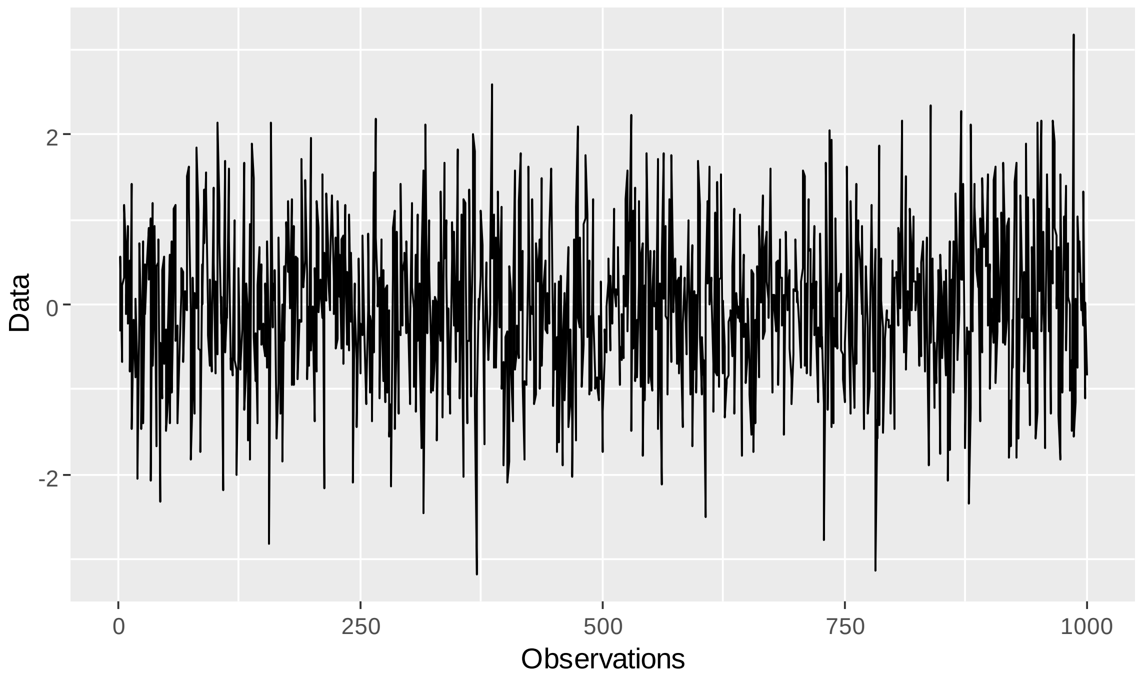
MVDA – *Normal Distribution*

```
# scatter plot
library("ggplot2")
ggplot(data = n_data, aes( x = 1:1000, y = n_data[,1])) + geom_point() + xlab("Observations") +
ylab("Data")
```



MVDA – *Normal Distribution*

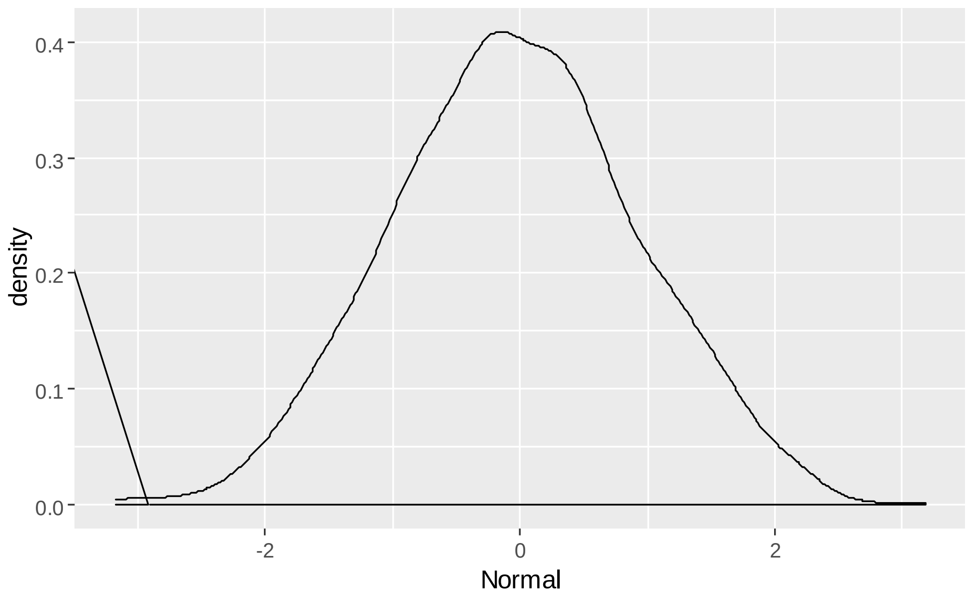
```
# line plot
library("ggplot2")
ggplot(data = n_data, aes( x = 1:1000, y = n_data[,1])) + geom_line() + xlab("Observations") + ylab("Data")
```



MVDA – *Normal Distribution*

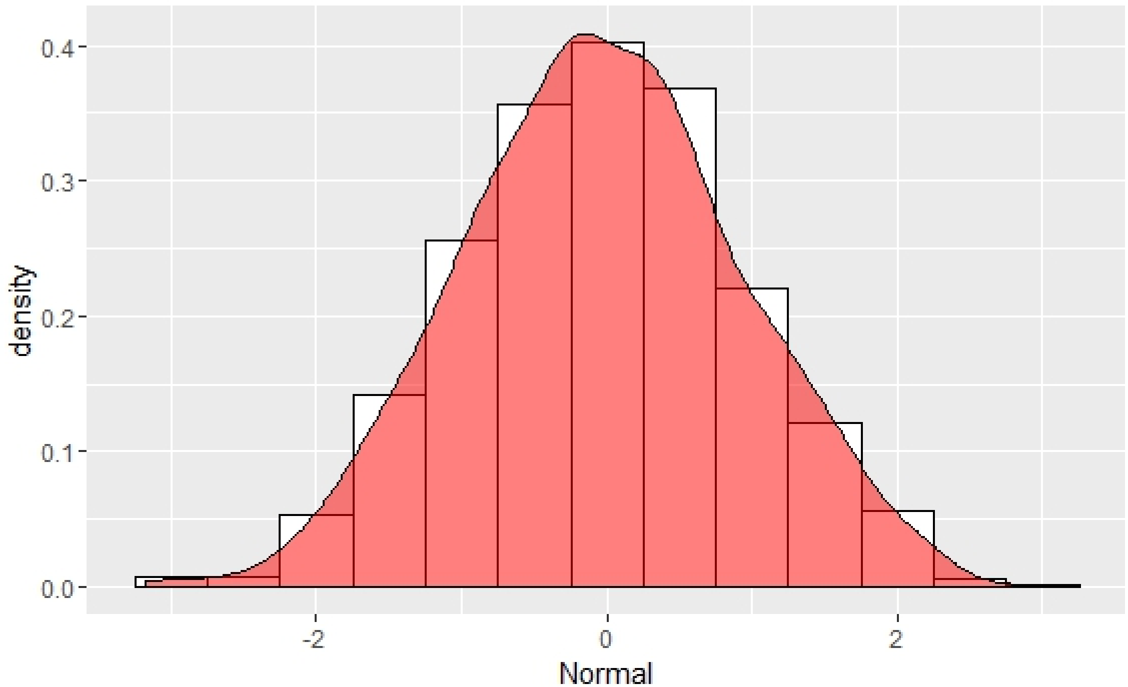
A **probability density function** (**PDF**) describes the relative likelihood of a random variable to assume a value (Integral of the function over a range). The integral over the entire space is always equal to one.

```
# density curve
ggplot(data = n_data, aes( x = n_data[,1])) + geom_density() + xlab("Normal")
```



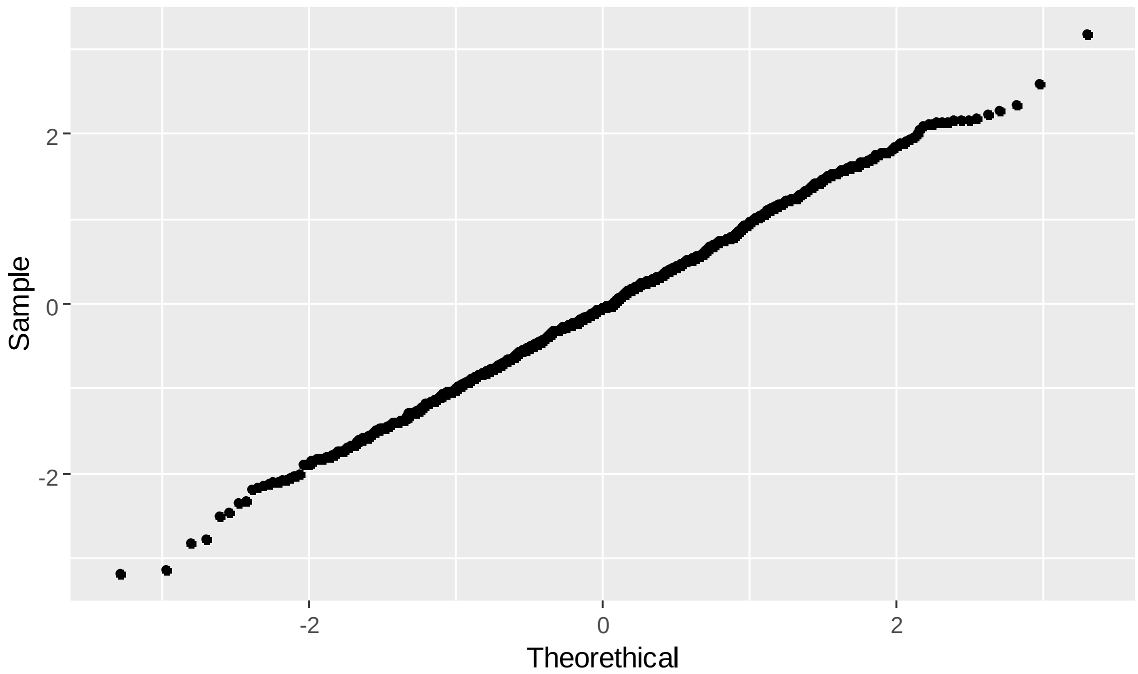
MVDA – *Normal Distribution*

```
# both curves
ggplot(data =  n_data, aes( x = n_data[,1])) + geom_histogram(aes(y = ..density.. ), binwidth= 0.5,
colour = "black", fill = "white") + geom_density(alpha = 0.2, fill = "red") + xlab("Normal")
```



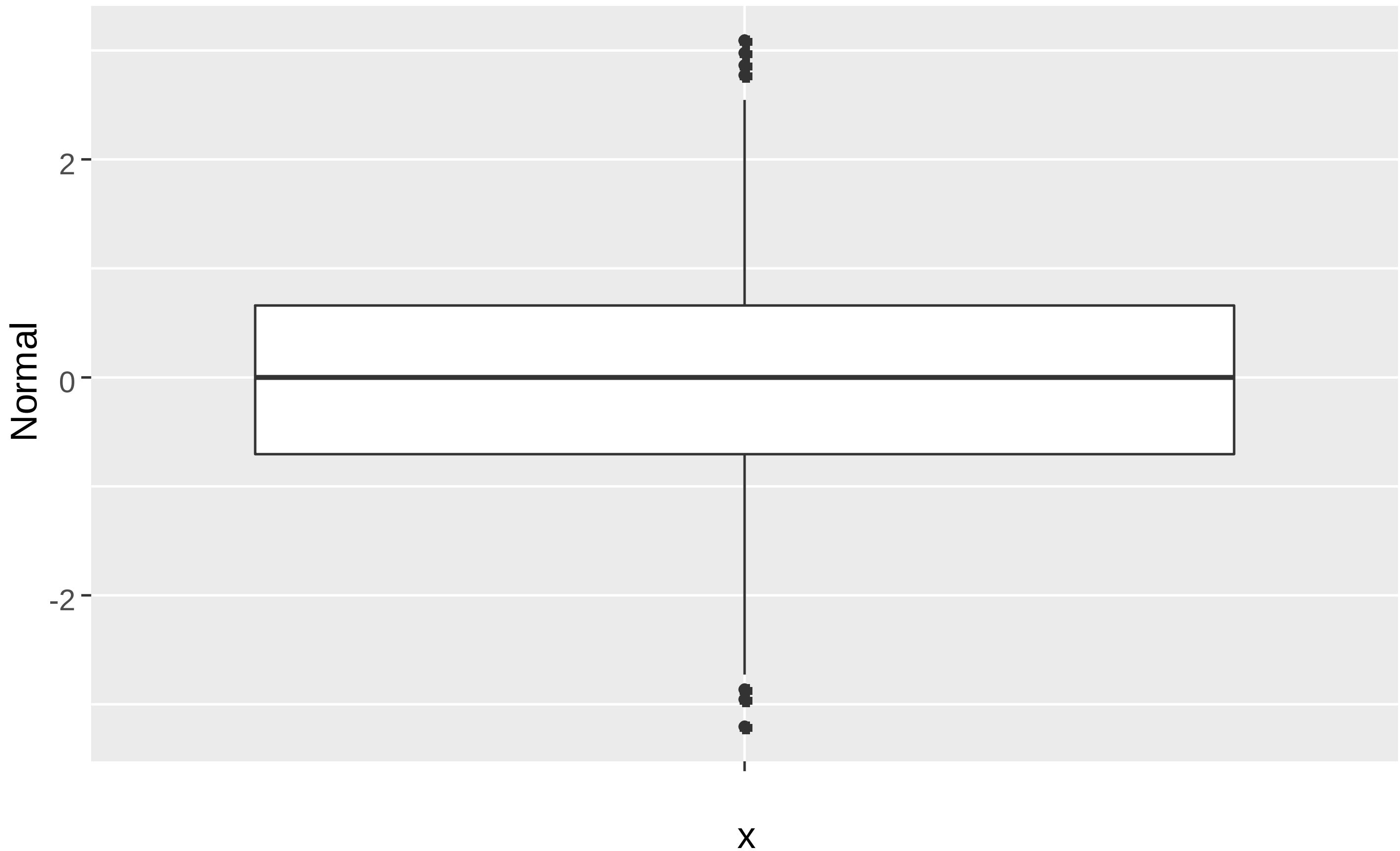
MVDA – *Normal Distribution*

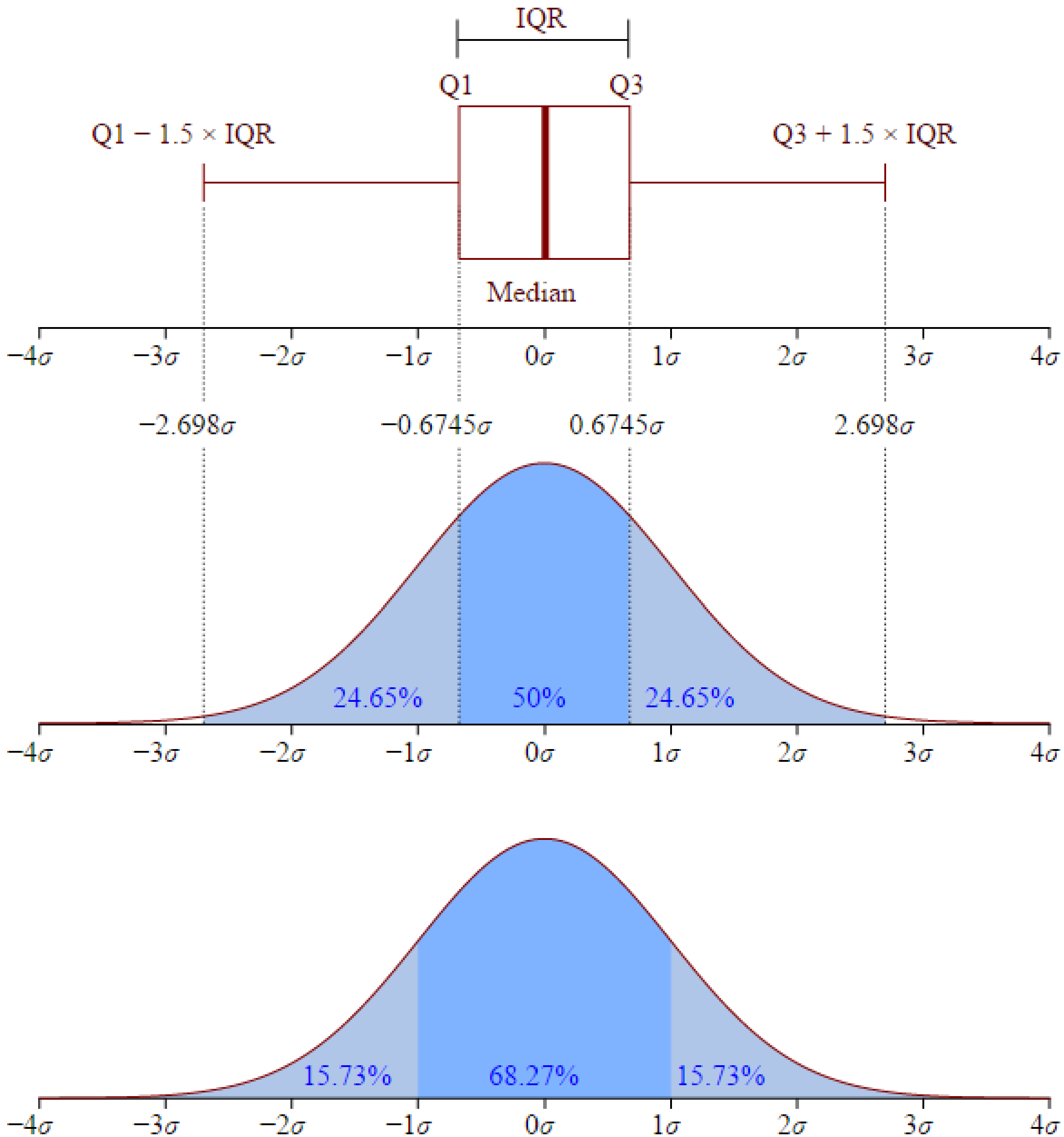
```
# qq plot
ggplot(data = n_data, aes( sample = n_data[ ,1])) + stat_qq()+ xlab("Theorethical") +
ylab("Sample")
```



MVDA – *Normal Distribution*

```
# box plot
ggplot(data = n_data, aes(x = " ", y = n_data[ ,1])) + geom_boxplot() + ylab("Normal")
# Q1 – 1.5IQR, Q1, Q2, Q3, Q3 + 1.5IQR
boxplot.stats(n_data[,1])$stats
[1] -2.5069914 -0.6919844 -0.0543911  0.5855897  2.3370023
```

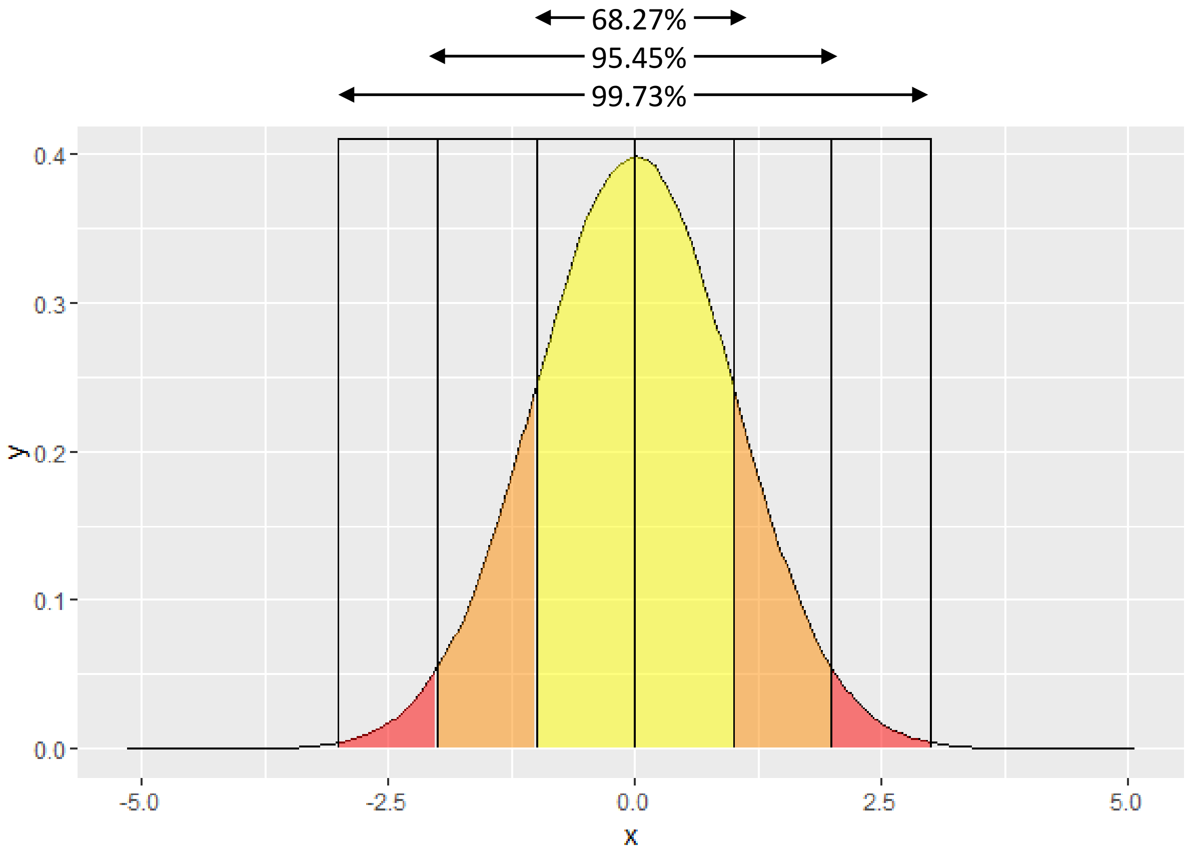




By Jhguch at en.wikipedia, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=14524285>

MVDA – *Normal Distribution*

The **yellow area** is within **one standard deviation** (σ) of the mean, representing 68.27% of the total number of observations. Within **two standard deviations** from the mean (**orange and yellow areas**) represents 95.45% of the total number of observations, and finally, within **three standard deviations** (**yellow, orange and red areas**) covers 99.73% of the total number of observations. This fact is known as empirical rule or the rule of the 3-$sigmas$.



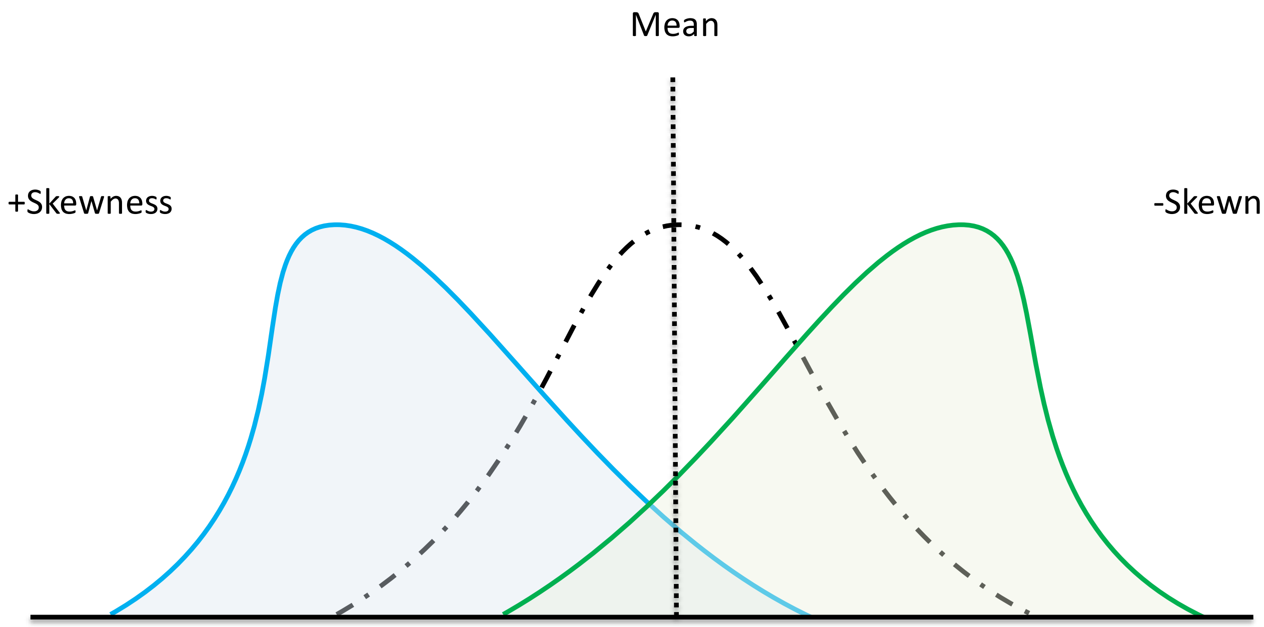
```
# Normal Plot
set.seed(101)
n_data <- as.data.frame(rnorm(900000, 0, 1))
density_n_data <- density(n_data[ , 1])
df_d <- data.frame(density_n_data$x, density_n_data$y)
colnames(df_d ) <- c ("x", "y")

ggplot(data =  df_d, aes( x = x , y = y)) + geom_line() + geom_ribbon(data = subset(df_d, x >= -1 &
x <= 1), aes( ymax = y), ymin = 0, fill= "yellow", colour = NA,  alpha=0.5) + geom_ribbon(data =
subset(df_d, x >= -2 & x < -1), aes( ymax = y), ymin = 0, fill = "darkorange", colour = NA,  alpha =
0.5) + geom_ribbon(data = subset(df_d, x >= 1 & x < 2), aes( ymax = y), ymin = 0, fill =
"darkorange", colour = NA,  alpha=0.5) + geom_ribbon(data = subset(df_d, x >= -3 & x < -2), aes(
ymax = y), ymin = 0, fill = "red", colour = NA,  alpha=0.5) + geom_ribbon(data = subset(df_d, x >= 2
& x < 3), aes( ymax = y), ymin = 0, fill = "red", colour = NA,  alpha=0.5) + geom_segment(data =
df_d , x = -3, y = 0.41, xend = 3, yend = 0.41) + geom_segment(data = df_d , x = -3, y = 0, xend = -
3, yend = 0.41) + geom_segment(data = df_d , x = -2, y = 0, xend = -2, yend = 0.41) +
geom_segment(data = df_d , x = -1, y = 0, xend = -1, yend = 0.41) + geom_segment(data = df_d , x
= 0, y = 0, xend = 0, yend = 0.41) + geom_segment(data = df_d , x = 1, y = 0, xend = 1, yend =
0.41) + geom_segment(data = df_d , x = 2, y = 0, xend = 2, yend = 0.41) + geom_segment(data =
df_d , x = 3, y = 0, xend = 3, yend = 0.41) + xlab("x") + ylab("y")
```

MVDA – *Normal Distribution*

The **Skewness** is a measure of the **Asymmetry** of a distribution.

```
library(e1071)
skewness(n_data[ ,1])
```

If $v < 0$, then the distribution has a left tail (values below average - positive skewness).
If $v = 0$, then the distribution is approximately symmetrical.
If $v > 0$, then the distribution has a right tail (above average values - negative skewness).



MVDA – *Normal Distribution*

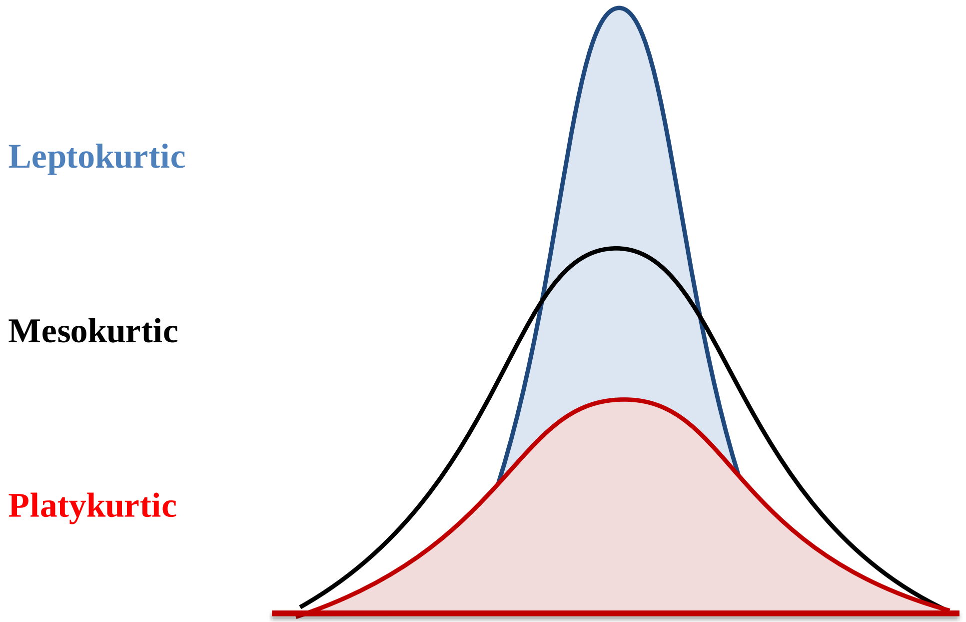
The **Kurtosis** is a measure of dispersion that characterizes the peak or flattening of the pdf curve.

```
library(e1071)
kurtosis(n_data[ ,1])
```

If $k > 0$, **Leptokurtic**: higher than the normal distribution.

If $k = 0$, **Mesokurtic**: same flattening the normal distribution.

If $k < 0$, **Platykurtic**: flatter than the normal distribution.

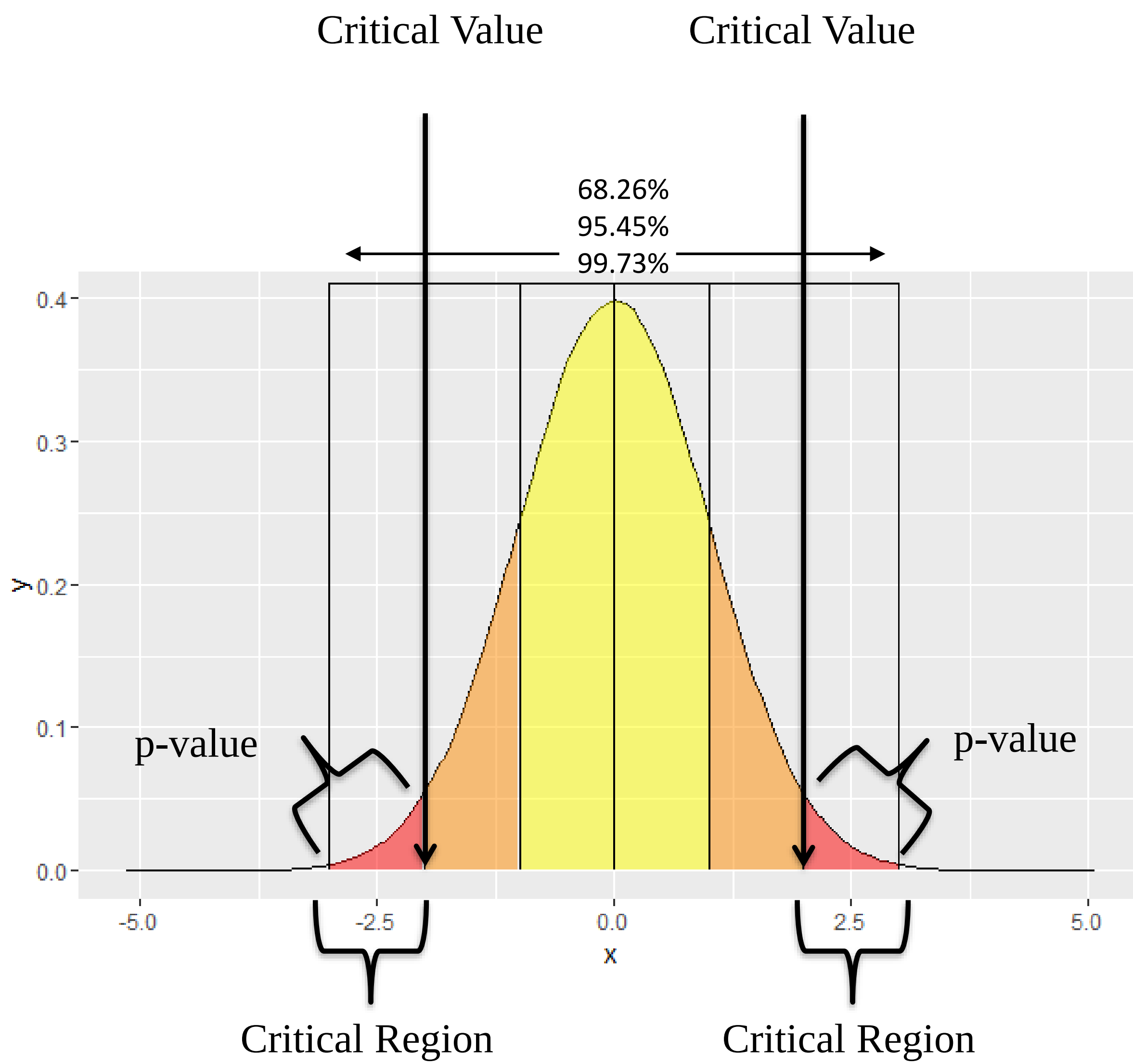


MVDA – ***Hypothesis Testing***

A **hypothesis** in statistics is a statement about some property of a population and **hypothesis testing** is a method that accepts or rejects this claim through a sample of that population.

- **Null hypothesis** (H_0): The statement that is being tested.
- **Alternative Hypothesis** (H_1): What is believed to be true, when the null hypothesis is considered false.
- **Critical Region**: The region which the values cause the rejection of the null hypothesis.
- **Critical Value**: It's value that starts the critical region.
- **Confidence Interval** ($1 - \alpha$; α = significance; $0 \leq \alpha \leq 1$): An estimated range of a parameter of interest of a population.
- **p – value**: The area under the pdf curve that rejects the null hypothesis if its value is less than the significance level α ($0 \leq \alpha \leq 1$).
- **Type I error** (False Positive): It is the incorrect rejection of a true null hypothesis.
- **Type II error** (False Negative): It is the incorrect acceptance of a false null hypothesis.

MVDA – *Hypothesis Testing*



MVDA – *Normality Test*

The **Shapiro-Wilk test** (for samples ≤ 5000 observations) and the **Kolmogorov-Smirnov test** (for samples > 5000 observations) checks whether a sample came from a normally distributed population. With the following hypothesis:

$$H_0: The\ sample\ is\ normally\ distributed$$
$$H_1: The\ sample\ is\ not\ normally\ distributed$$

```
# R Shapiro-Wilk test supports up to 5000 observations
shapiro.test()

# The KS test needs a reference distribution y.
set.seed(101)
n_data <- as.data.frame(rnorm(900000, 0, 1))
r_data <- as.data.frame(runif(900000))
ks.test(x = m_data[,1], y = n_data[,1])

p-value < 2.2e-16  # rejects  the null hypothesis, thus x is not normally distributed
```

MVDA

https://github.com/Valdecy/Multivariate_Data_Analysis

##
##

Created by: Prof. Valdecy Pereira, D.Sc.
UFF - Universidade Federal Fluminense (Brazil)
email: valdecypereira@yahoo.com.br
Course: Multivariate Data Analysis
Lesson: Descriptive Statistics with R

Citation:

PEREIRA, V. (2016). Project: Multivariate Data Analysis, File: R-MVDA-Descriptive.pdf, GitHub repository: <https://github.com/Valdecy/Multivariate_Data_Analysis>

##
##

Bibliography

CORRAR, L.J.; PAULO, E.; DIAS FILHO, J. M. **Análise Multivariada para Cursos de Administração, Ciências Contábeis e Economia**. ATLAS, 2009.

FÁVERO, L. P.; BELFIORE, P.; SILVA, F. L.; CHAN, B. **Análise de Dados: Modelagem Multivariada para Tomada de Decisões**. CAMPUS, 2009.

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E.; TATHAM, R. L. **Análise Multivariada de Dados**. BOOKMAN, 2009.

LATTIN, J.; CARROLL, J. D.; GREEN, P. E. **Análise de Dados Multivariados**. CENGAGE Learning, 2011.

LEVINE, D. M.; STEPHAN, D. F.; KREHBIEL, T. C.; BERENSON, M. L. **Estatística - Teoria e Aplicações - Usando Microsoft Excel**. LTC, 2012.