R

ROBERTO ALEJANDRO CÁRDENAS OVANDO

Outline

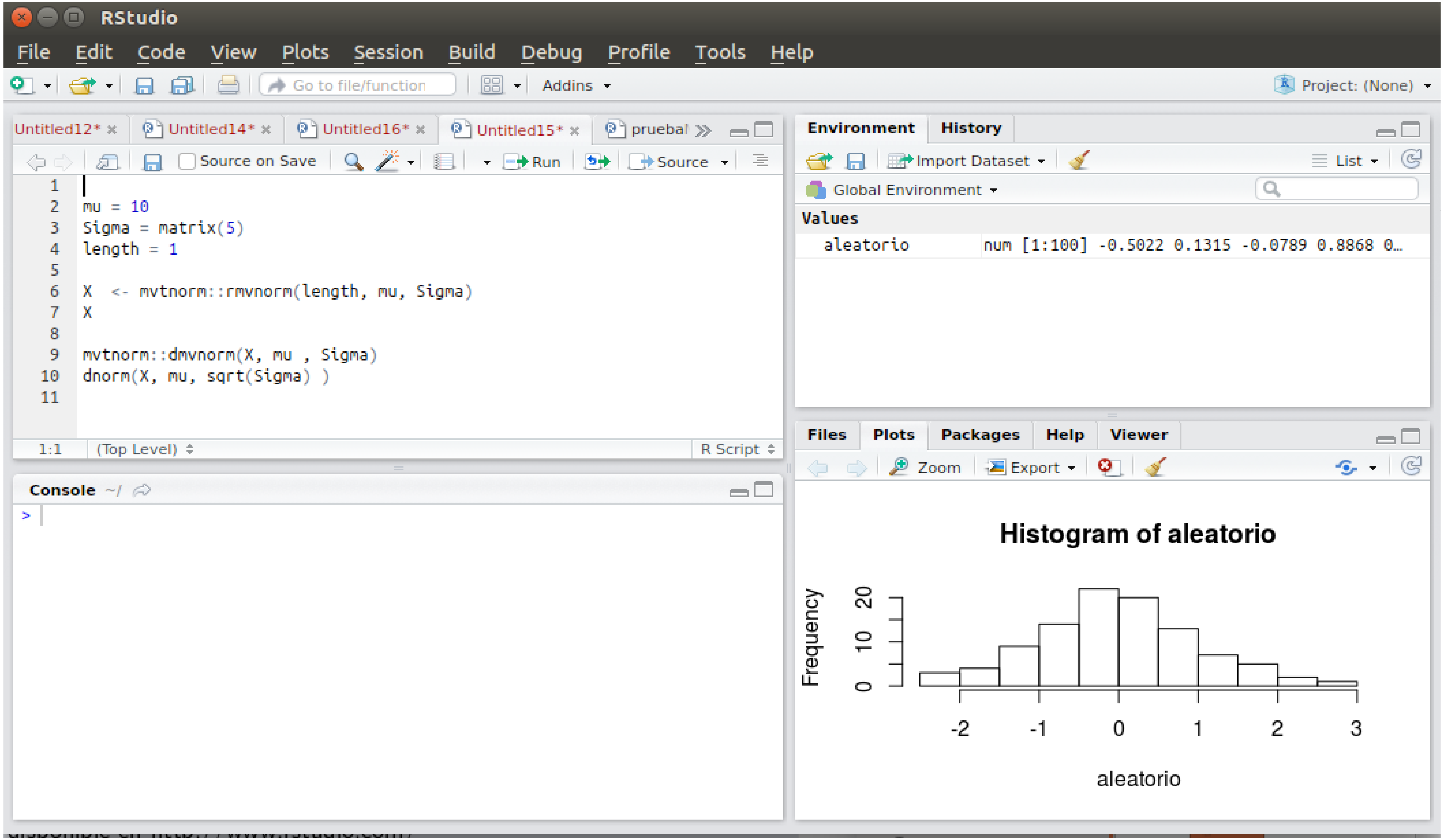
- R
- Using R – basics
- Control flow

What is R ?

- Programming language to do statistical analyses in a quick and easy way
- Good to visualize data
- Free and open source
- Multiplatform (MacOS, Linux and Windows)

What is R Studio ?

- RStudio is an Integrated Development Environment
- It is not necessary, but it is pretty useful
- It has 4 windows:
  - Script
  - Shell
  - Workspace
  - Help/Files/Plots
- Free for personal use



```r

mu = 10
Sigma = matrix(5)
length = 1

X  <- mvtnorm::rmvnorm(length, mu, Sigma)
X

mvtnorm::dmvnorm(X, mu , Sigma)
dnorm(X, mu, sqrt(Sigma) )

```

```
Values
aleatorio        num [1:100] -0.5022 0.1315 -0.0789 0.8868 0…
```

```
>
```

disponible en http://www.rstudio.com/

Install R and RStudio

- Download the installable R app at:

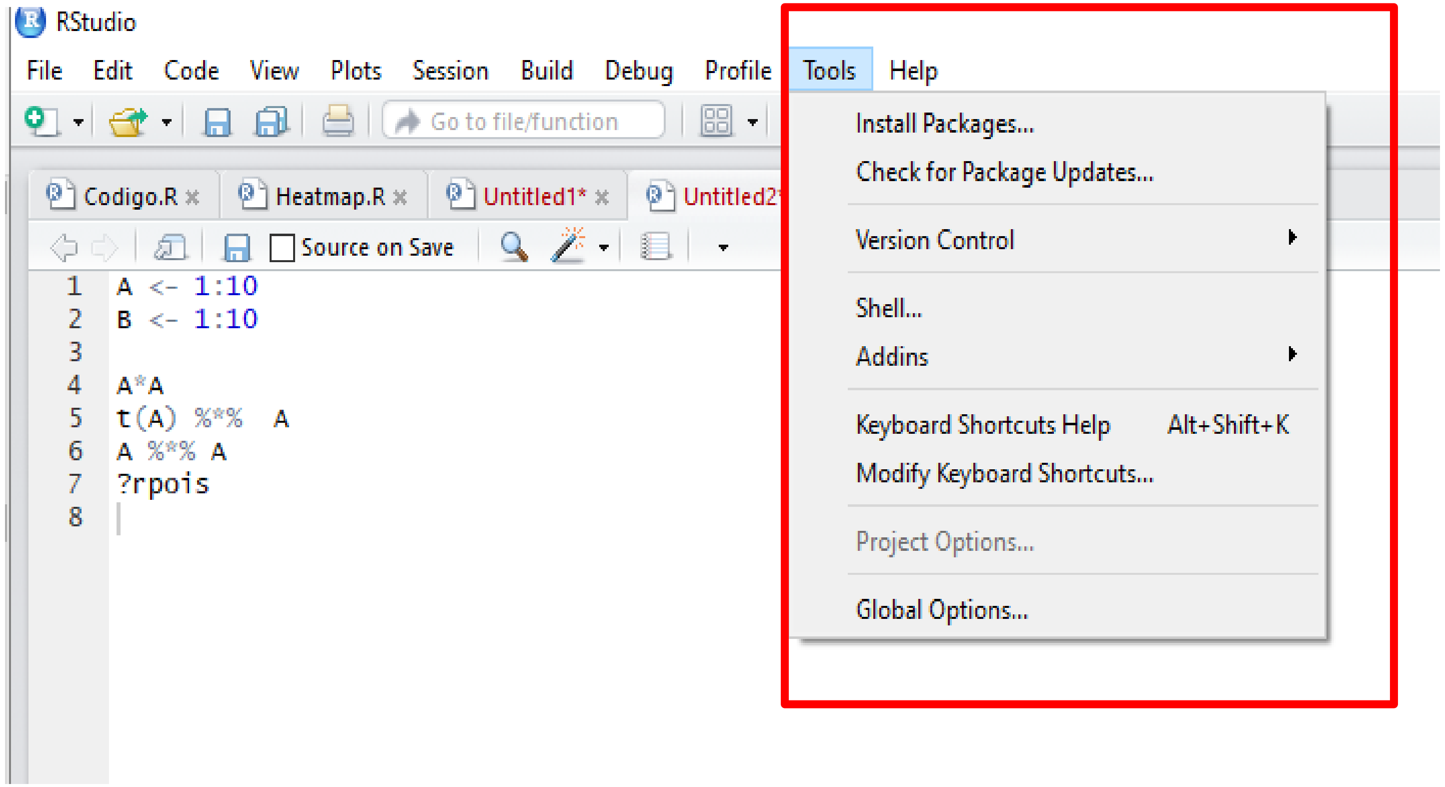
  https://cran.r-project.org/

- Download Rstudio from:

  https://www.rstudio.com/products/rstudio/download/

Install more libraries

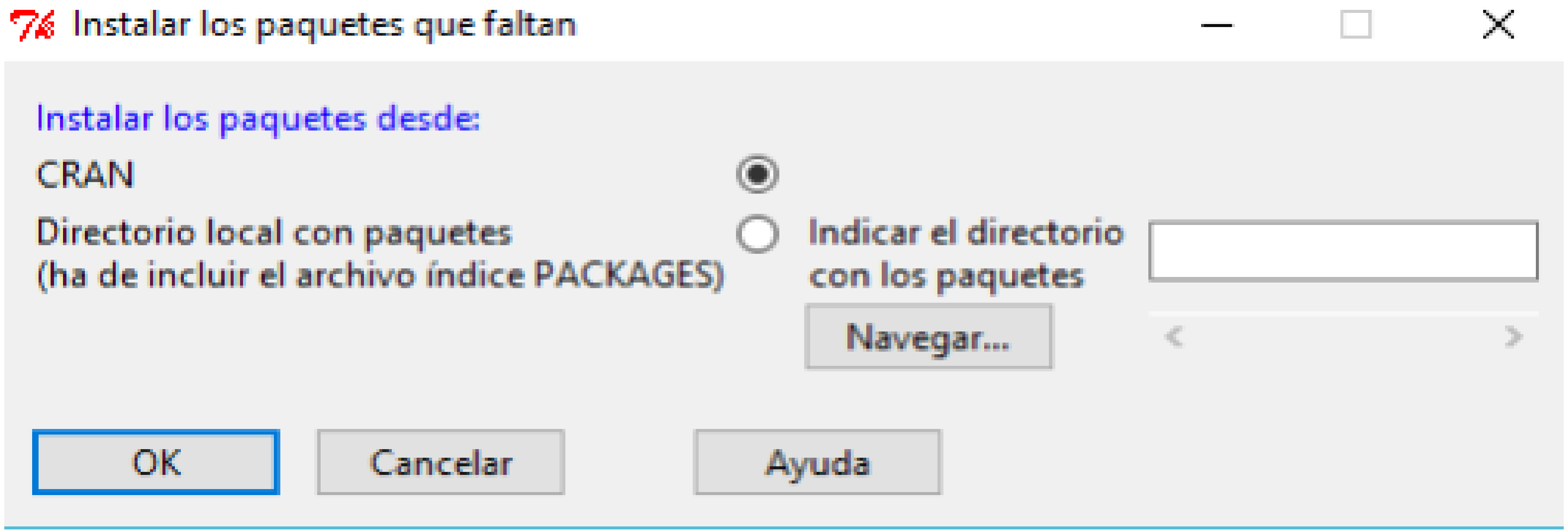
- From Rstudio interface:



```
A <- 1:10
B <- 1:10

A*A
t(A) %*%  A
A %*% A
?rpois

```

Install more libraries

- If it is the first time using Rstudio, it will ask for the necessary Repository. Select CRAN and Mexico Server



Instalar los paquetes que faltan

Instalar los paquetes desde:

CRAN ●

Directorio local con paquetes (ha de incluir el archivo índice PACKAGES) ○ Indicar el directorio con los paquetes

Coffee Break

Install the necessary Software

Using R

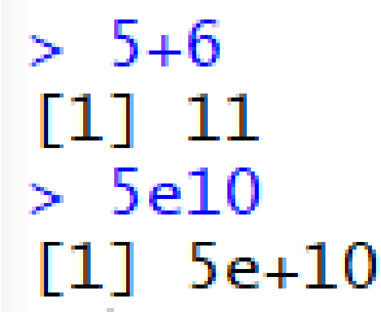
- R is usually used in the shell/command prompt

- The command prompt has a > symbol to show the place where the instructions, functions and variables must be entered

- If an instruction has a # symbol at the start of the lines, it will be recognized as a comment

Using R command prompt

- The command prompt can be used as a scientific calculator

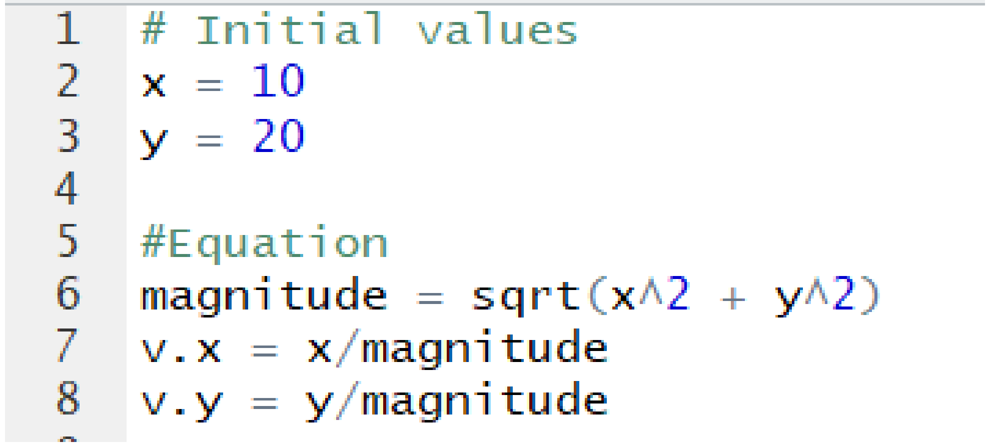


```
> 5+6
[1] 11
> 5e10
[1] 5e+10
```

- However, the most important feature to use in R is the vector and matrix operations

Using R to solve algebraic equations

- Variable: An element, feature, or factor that is liable to vary or change. Used to store data.
- Example normal vector equation:
  - Initial Values
    - x = 10
    - y = 20
  - Equation:
  - $\hat{v} = \left(\frac{x}{\sqrt{x^2+y^2}}, \frac{y}{\sqrt{x^2+y^2}}\right)$



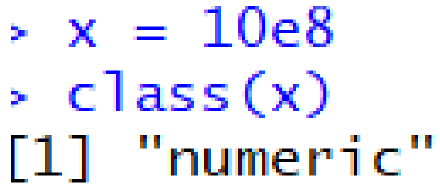
```
# Initial values
x = 10
y = 20

#Equation
magnitude = sqrt(x^2 + y^2)
v.x = x/magnitude
v.y = y/magnitude
```

NOTE: Use the Rstudio Script window, select the line to execute and use the button

Using R with different kind of data

- Most used data types in R:
  - Logical: TRUE, FALSE
  - Numeric: Natural or Real numbers. E.g. 10, 1.5, 10e8
  - Character: Text. E.g. “Mexican”, “French”
  - Data Frame: Multiple kind of data in the same variable (Matrix)

- To know the data type of a variable, we use the function “class”

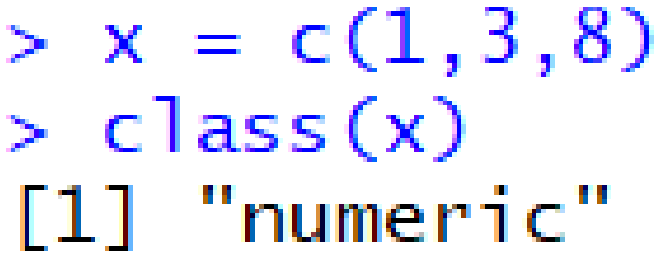


```
> x = 10e8
> class(x)
[1] "numeric"
```

Using R with vectors

- To use a vector it is necessary to link or concatenate elements of the same data type. The function to us is “c”

- Example: I want a vector with the values 1,3,8

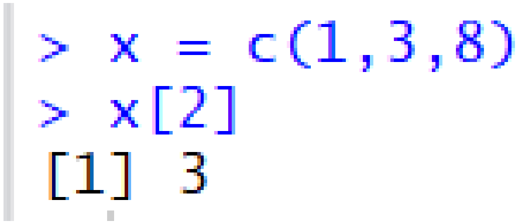


```
> x = c(1,3,8)
> class(x)
[1] "numeric"
```

Using R and vector functions

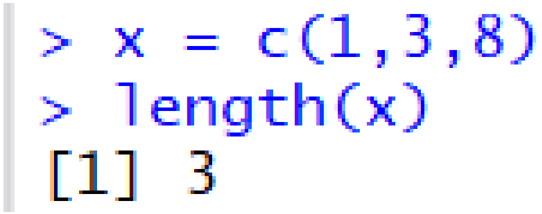
- To use a specific element from the vector, we use brackets after the vector variable name.

- Example: I want to use the second element of the vector (1,3,8)



```
> x = c(1,3,8)
> x[2]
[1] 3
```

- To know how many elements are in the vector, we use the function “length”



```
> x = c(1,3,8)
> length(x)
[1] 3
```

Exercises - Part 1

- Use R and Rstudio to do the following exercises:
  - Compute the operation: log(((3+2)*5)+6). Store each operation (the innermost parenthesis operation) in a different variable and use it to compute the next parenthesis.
  - E.g.
    - First compute x = 3+2.
    - Then compute y = x*5
    - And so on

Exercises - Part 1

- Create a character vector with 5 elements, then print the fourth element.
- Create a numeric vector with 4 elements, then print the sum of the first element and the third element

Matrices

☛ ❖Column vector

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad (n \ \times 1)$$

❖Row vector

$$[y_1 \quad \cdots \quad y_n] \qquad (1 \ \times n)$$

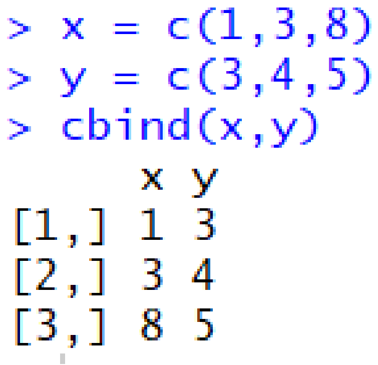
Matrices

- Matrix

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \qquad (n \ \times p)$$

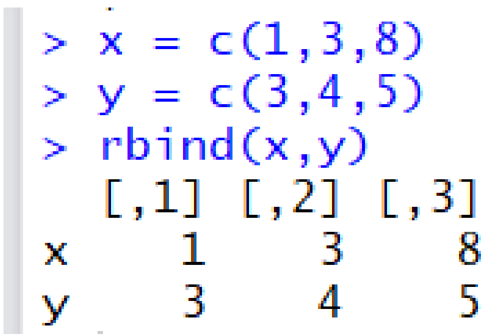
Using R and matrices

- We have 2 ways to create a matrix.
  - Concatenate vectors
    - Concatenate each vector as a column.
    - Function cbind



```
> x = c(1,3,8)
> y = c(3,4,5)
> cbind(x,y)
     x y
[1,] 1 3
[2,] 3 4
[3,] 8 5
```

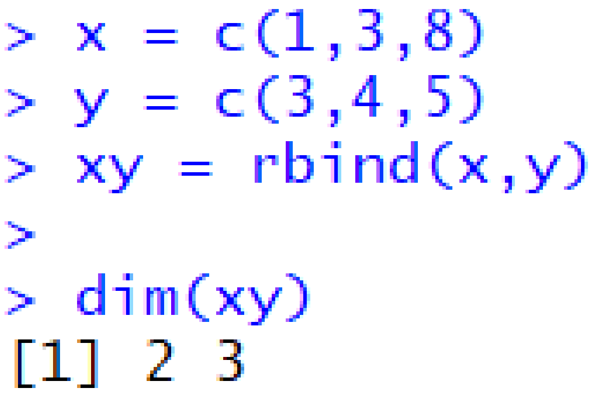
    - Concatenate each vector as a row.
    - Function rbind



```
> x = c(1,3,8)
> y = c(3,4,5)
> rbind(x,y)
  [,1] [,2] [,3]
x    1    3    8
y    3    4    5
```

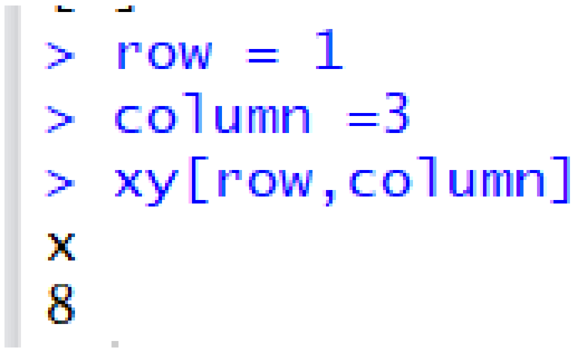
Using R and matrices

- To get the matrix size we use the dim function



```
> x = c(1,3,8)
> y = c(3,4,5)
> xy = rbind(x,y)
>
> dim(xy)
[1] 2 3
```

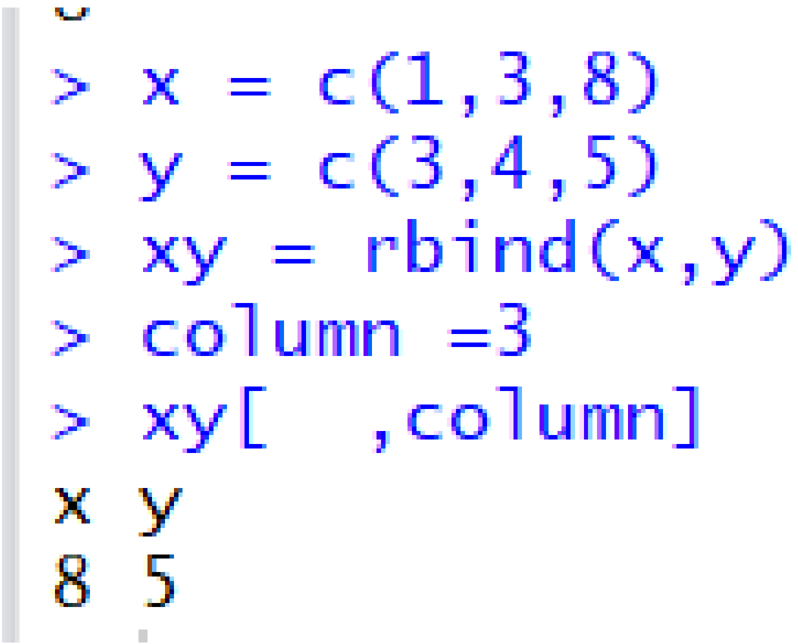
- To use a specific element from the matrix, we use brackets after the matrix variable name.



```
> row = 1
> column =3
> xy[row,column]
x
8
```

Using R and matrices

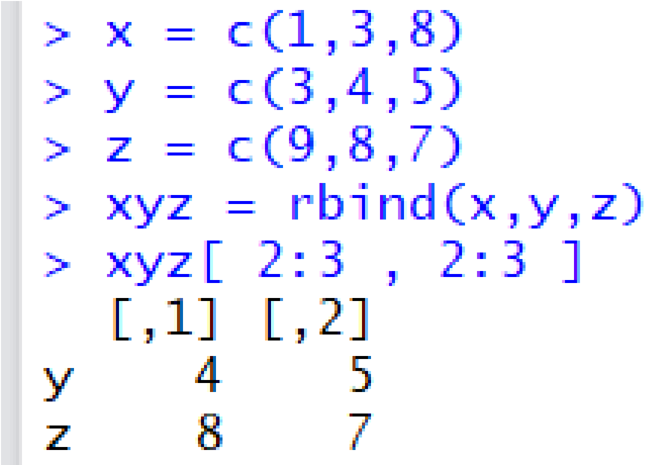
- If we want to retrieve all the values in a column the only value to be specified is the desired column, the same strategy is used if a row is desired.
- All values in a specified column



```
> x = c(1,3,8)
> y = c(3,4,5)
> xy = rbind(x,y)
> column =3
> xy[  ,column]
x y
8 5
```

Using R and matrices

- If we want a submatrix from the matrix, we can specified the start and end indices for each dimension.
- Example. From a 3x3 matrix get the inferior-right submatrix (2x2)



```
> x = c(1,3,8)
> y = c(3,4,5)
> z = c(9,8,7)
> xyz = rbind(x,y,z)
> xyz[ 2:3 , 2:3 ]
  [,1] [,2]
y    4    5
z    8    7
```

NOTE: We can create a vector of consecutive numbers with the format   initialValue : finalValue

Matrices

- Element-wise operations
  - Addition, subtraction and scalar product

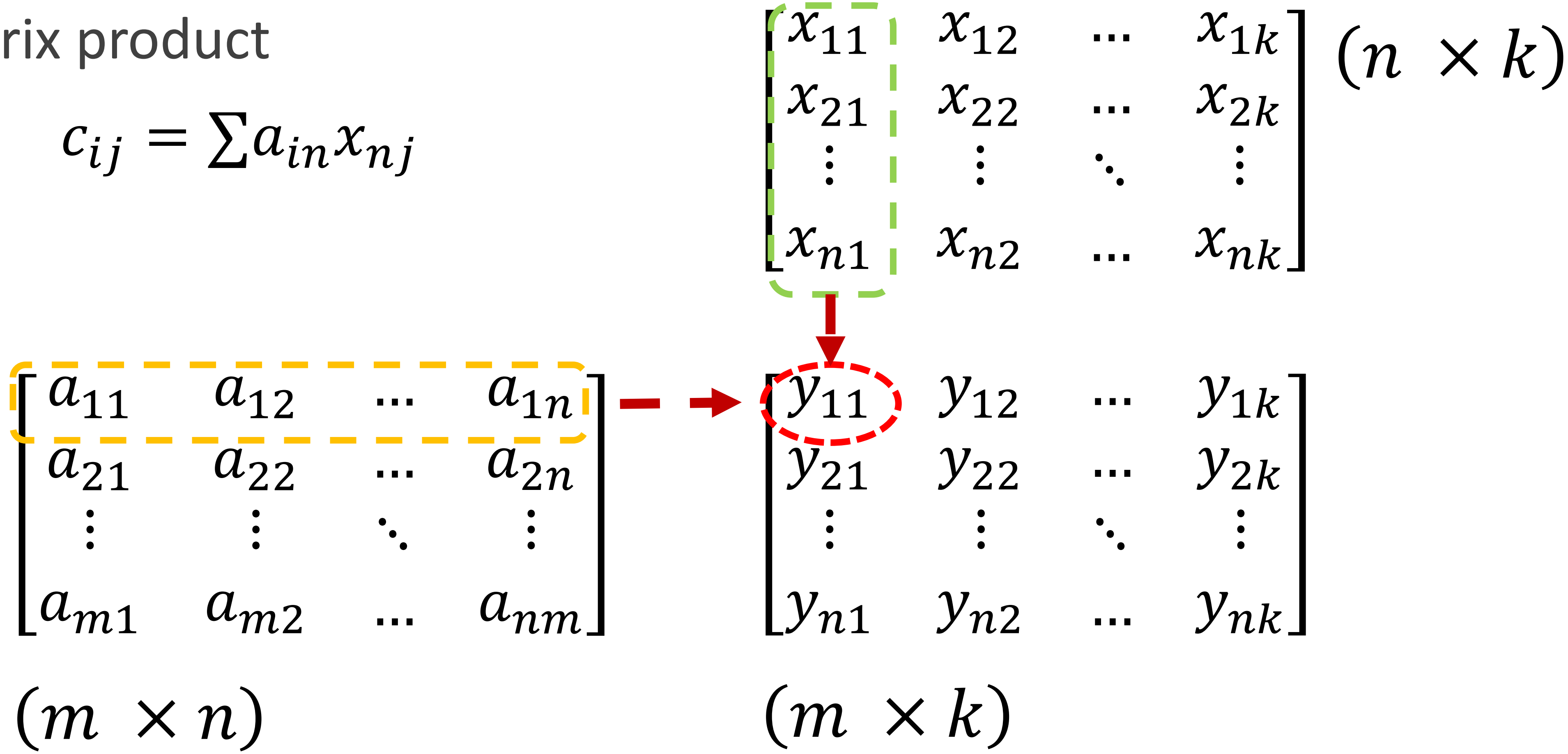
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \pm \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$$

$$\mathrm{k} * \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} k * x_1 \\ k * x_2 \\ \vdots \\ k * x_n \end{bmatrix}$$

Matrices

- Matrix product

$$c_{ij} = \sum a_{in} x_{nj}$$



$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad (n \ \times k)$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{nm} \end{bmatrix} \qquad \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1k} \\ y_{21} & y_{22} & \cdots & y_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nk} \end{bmatrix}$$

$$(m \ \times n) \qquad\qquad (m \ \times k)$$

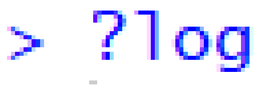
Using R - Products

- Element wise product: C = A * B
  - Only the * symbol between two vectors of the same size
  - Input: A (nx1) and B (nx1)
  - Output: C (nx1)

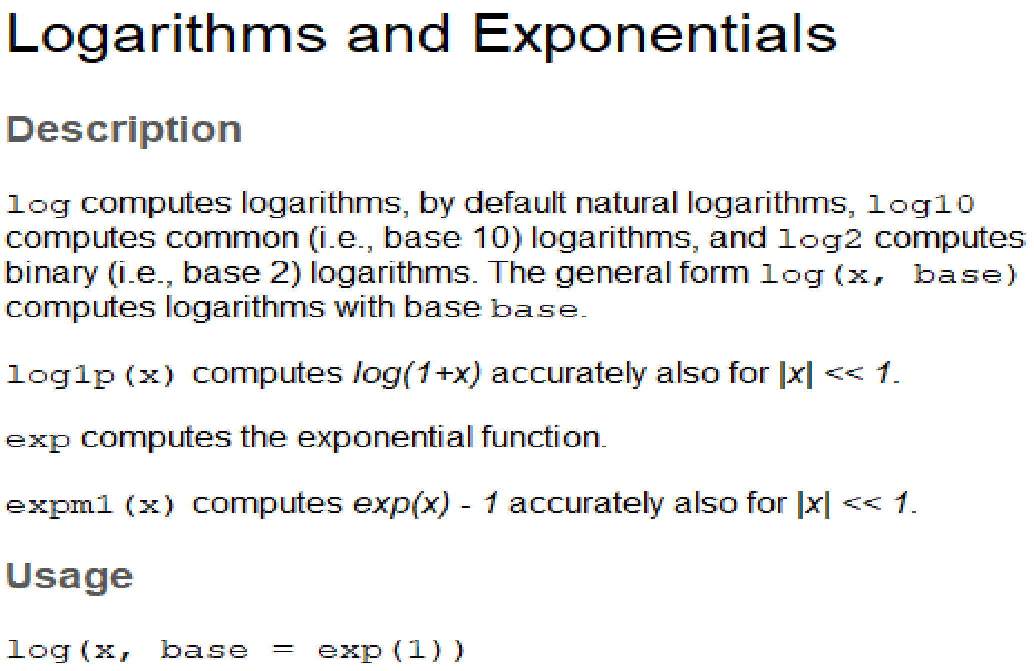
- Matrix product: C = A %*% B
  - The operation * must be surrounded by the % symbol
  - Input: A (nxm) and B (mxk)
  - Output: C (nxk)

Using R's help

- In R we can use the help function to know how a function is used.
- Example: I want to know to use the log function.



```
> ?log
```



Logarithms and Exponentials

Description

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

`log1p(x)` computes *log(1+x)* accurately also for |*x*| << *1*.

`exp` computes the exponential function.

`expm1(x)` computes *exp(x) - 1* accurately also for |*x*| << *1*.

Usage

```
log(x, base = exp(1))
```

Exercises – Part 2

- Use the help function to learn how to use the following functions:
  - hist
  - rnorm

- From the Usage section of the rnorm function, generate one thousand values with a mean of 10 and standard deviation of 3 and store them in a variable called: randomValues

Scripts

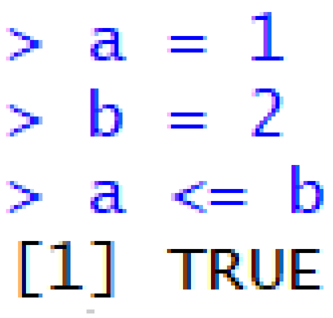
- A script is a file that has all the instructions necessary to fulfill a purpose.
- In R, the scripts have a “.R” extension
- The scripts are necessary to have reproducible experiments.
- i.e. If you run a script and I run a script, it must do the same for both of us.

Control flow

- All the instructions are done from top to bottom in a script, and we can manipulate them to do some parts of the script if a condition is fulfilled and even repeat the same subsection of the code depending in our needs.

Conditions

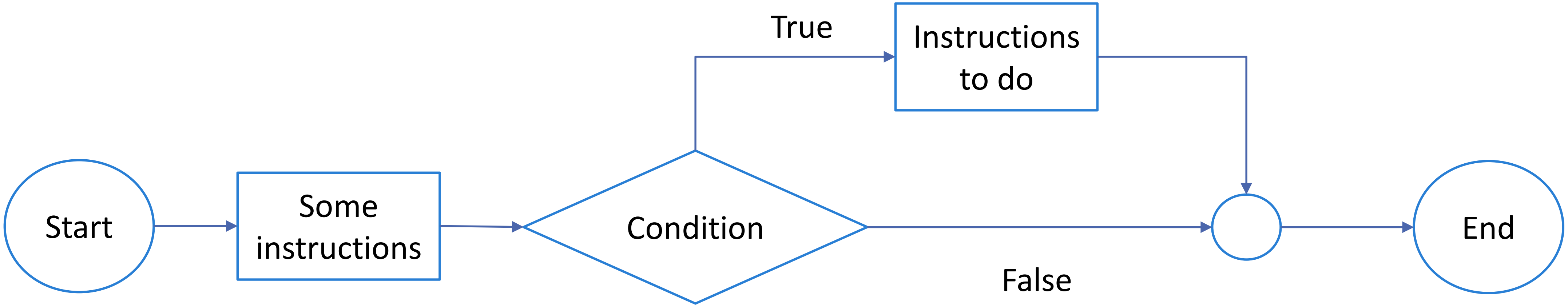
- To compare two values we can use relational operators, if the comparison is satisfied it returns TRUE; if not, it returns FALSE:
  - A is equal to B: A == B
  - A is different than B: A != B
  - A is greater than B: A > B
  - A is greater or equal to B: A >= B
  - A is less than B: A < B
  - A is less or equal to B: A <= B



```
> a = 1
> b = 2
> a <= b
[1] TRUE
```

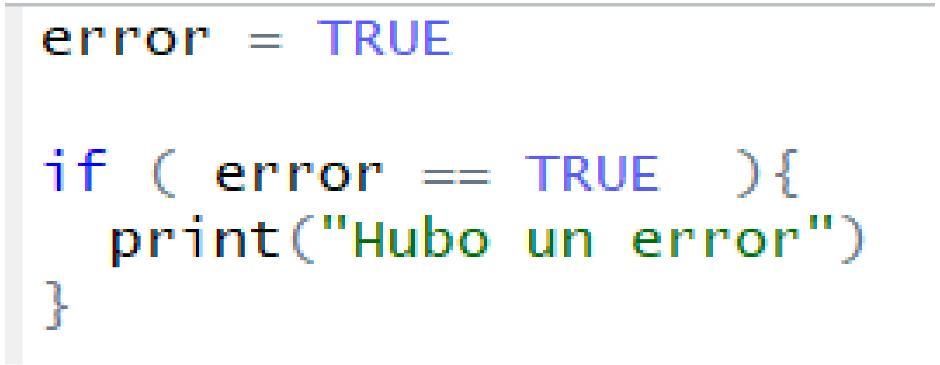
Control flow - Conditions

- Single condition



Control flow - Conditions

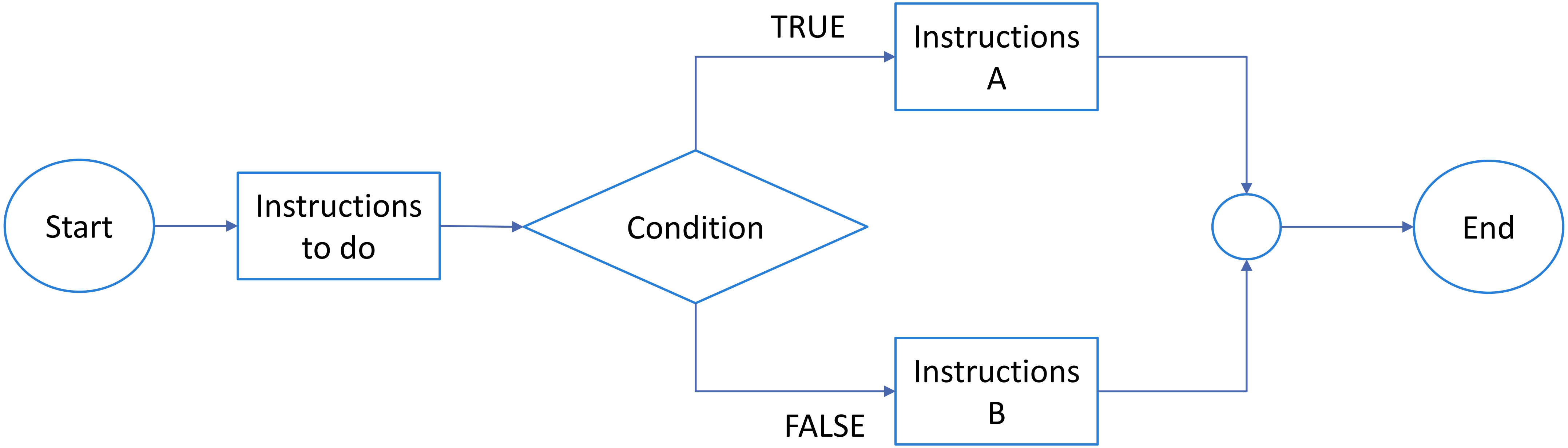
- Single condition



```
error = TRUE

if ( error == TRUE  ){
  print("Hubo un error")
}
```

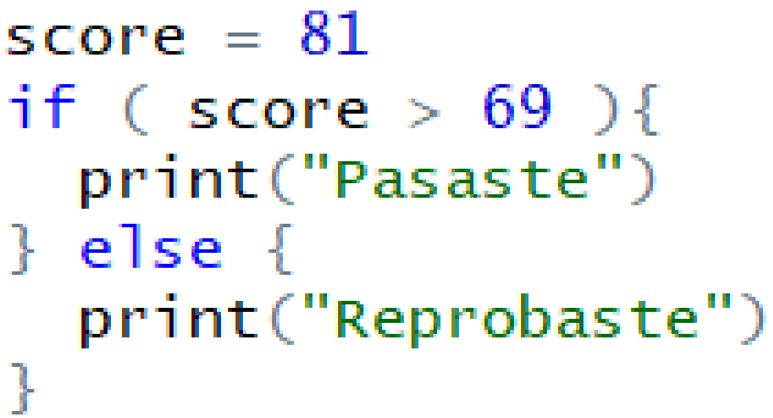
Control flow - Conditions

- Two possible results from 1 condition



Control flow - Conditions

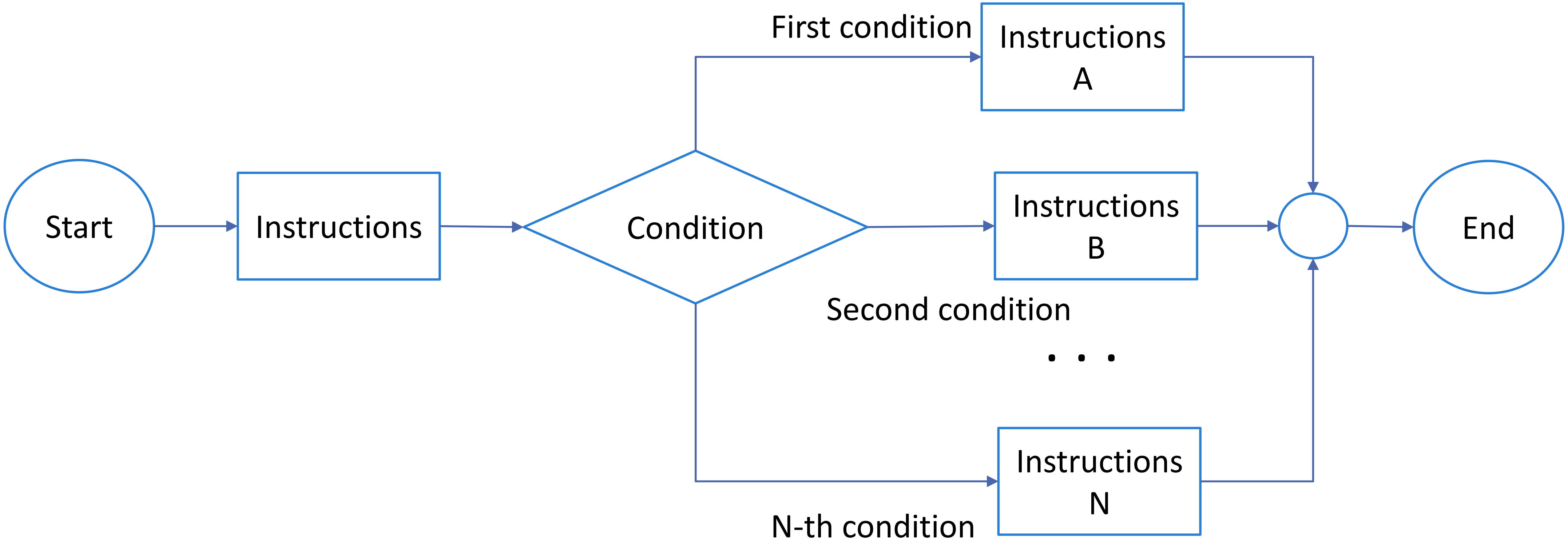
- Two possible results from 1 condition



```
score = 81
if ( score > 69 ){
  print("Pasaste")
} else {
  print("Reprobaste")
}
```

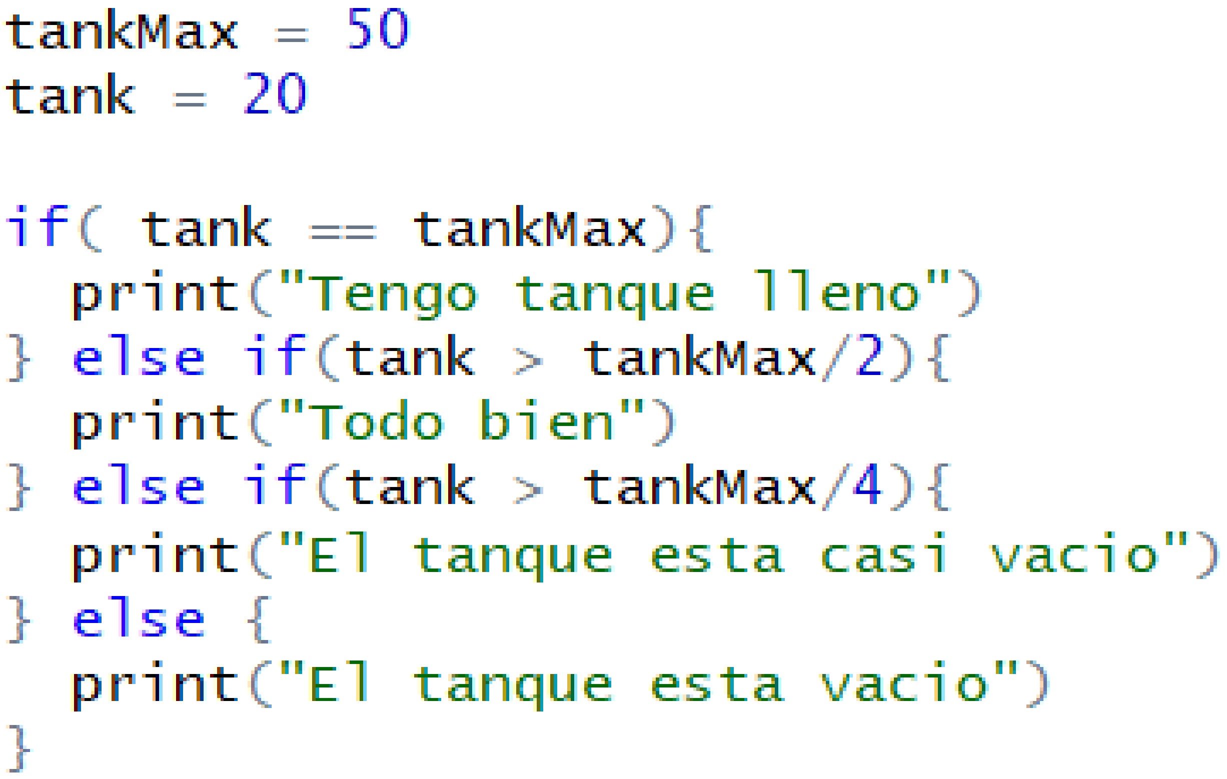
Control flow - Conditions

- Multiple results



Control flow - Conditions

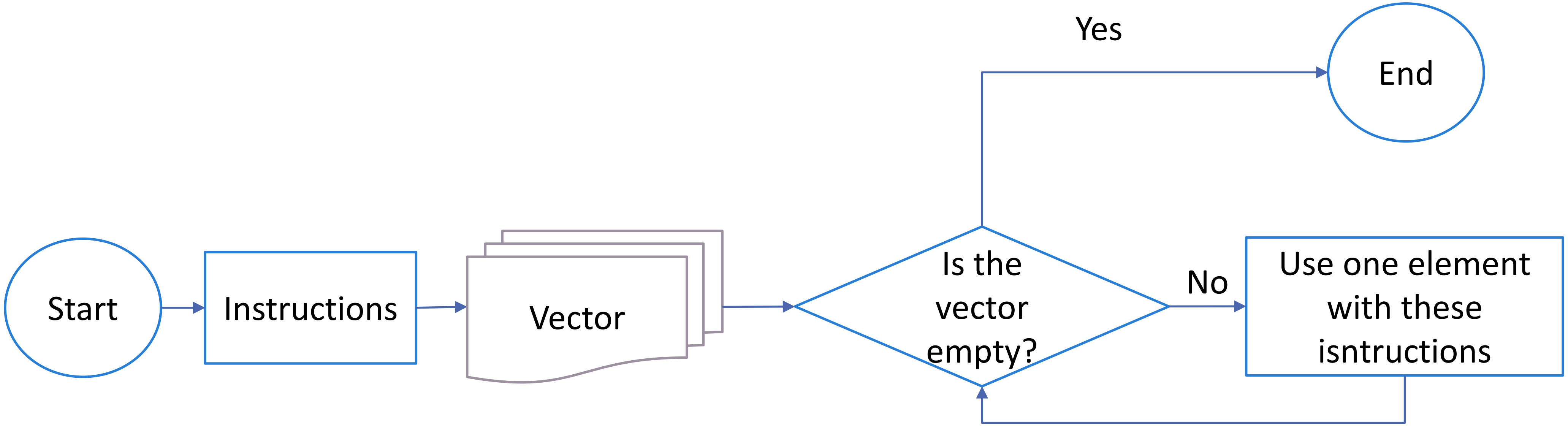
- Multiple results



```
tankMax = 50
tank = 20

if( tank == tankMax){
  print("Tengo tanque lleno")
} else if(tank > tankMax/2){
  print("Todo bien")
} else if(tank > tankMax/4){
  print("El tanque esta casi vacio")
} else {
  print("El tanque esta vacio")
}
```

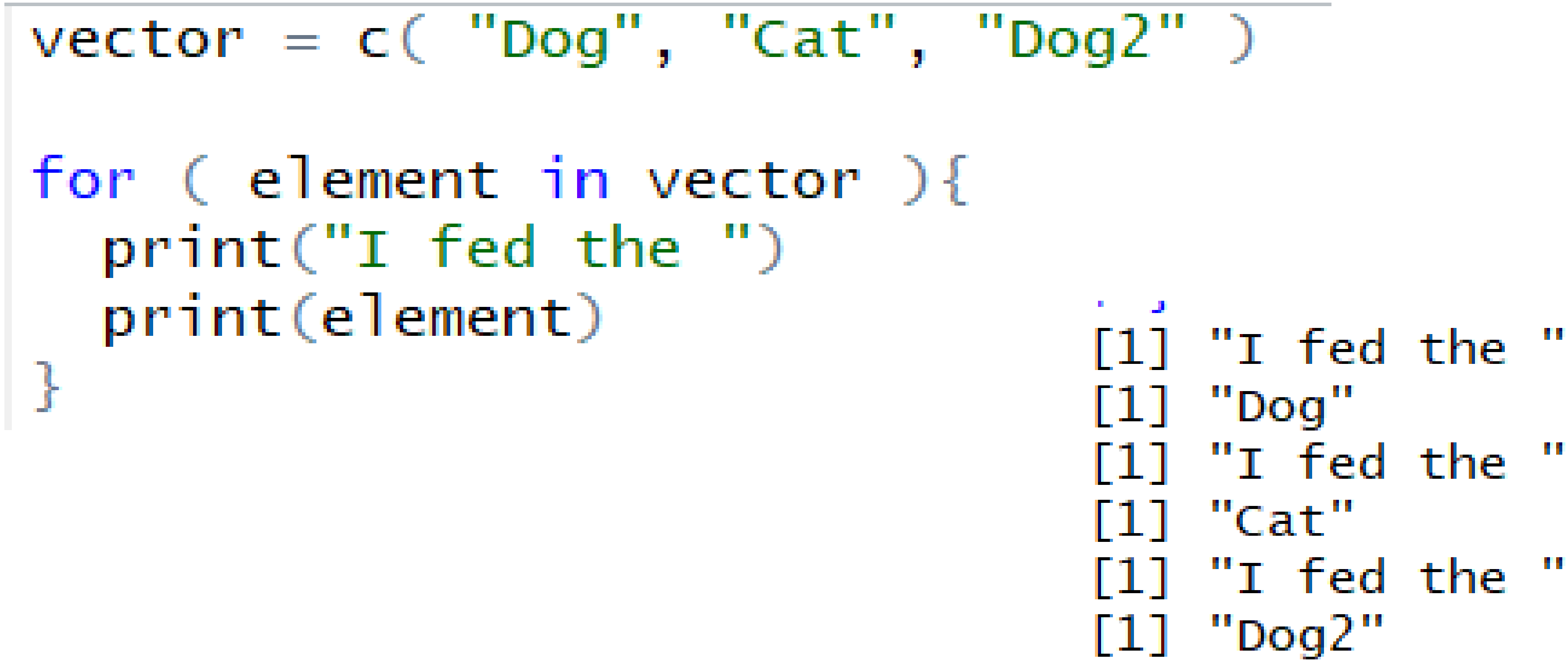
Control flow - Loops

- Repeat the same instructions for each element in a vector:



Control flow - Loops

- Repeat the same instructions for each element in a vector:

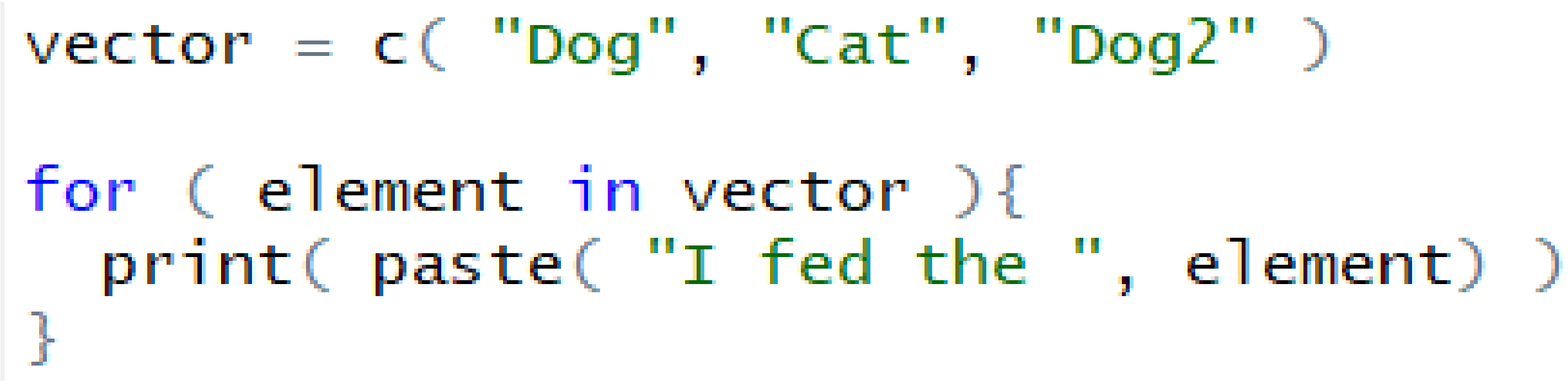


```r
vector = c( "Dog", "Cat", "Dog2" )

for ( element in vector ){
  print("I fed the ")
  print(element)
}
```

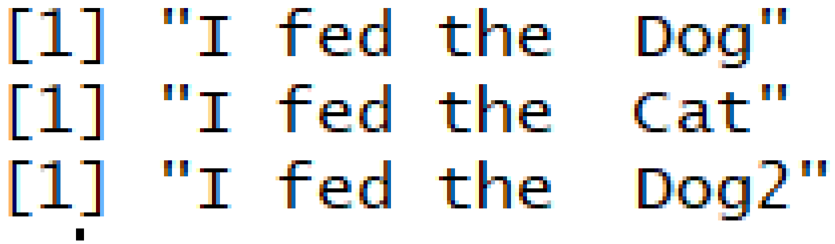
```
[1] "I fed the "
[1] "Dog"
[1] "I fed the "
[1] "Cat"
[1] "I fed the "
[1] "Dog2"
```

Control flow - Loops

- Repeat the same instructions for each element in a vector:



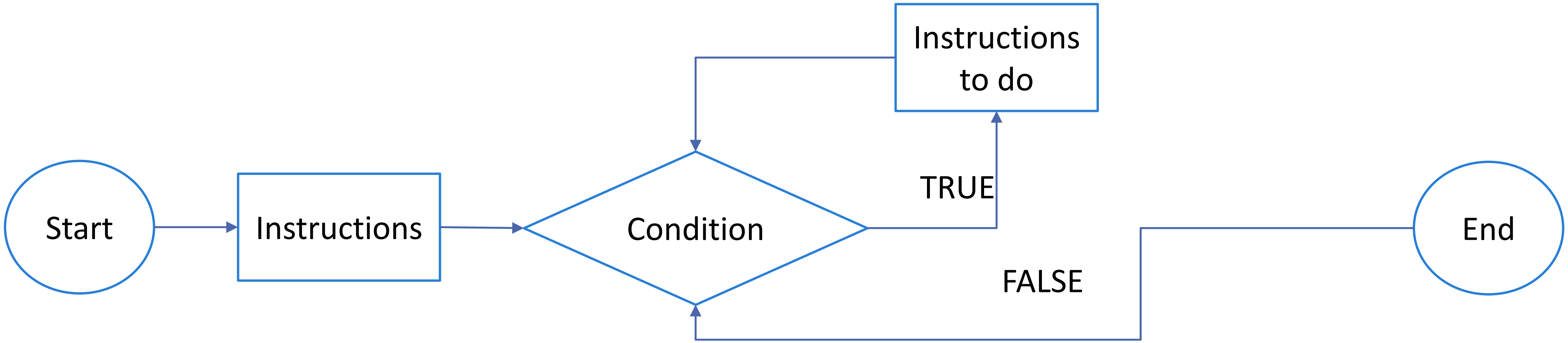
```r
vector = c( "Dog", "Cat", "Dog2" )

for ( element in vector ){
  print( paste( "I fed the ", element) )
}
```



```
[1] "I fed the  Dog"
[1] "I fed the  Cat"
[1] "I fed the  Dog2"
```

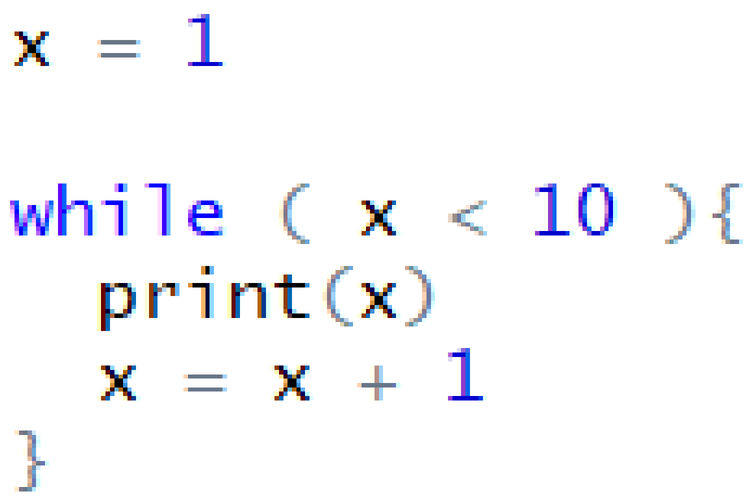
Control flow - Loops

- Repeat the same instructions until a condition is NOT satisfied

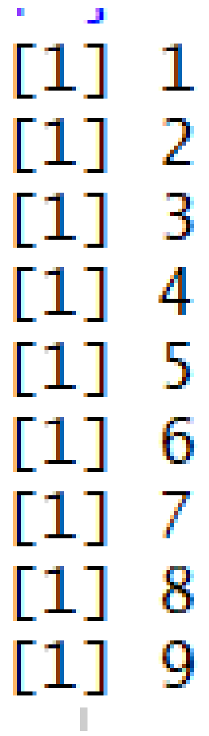


Control flow - Loops

- Repeat the same instructions until a condition is NOT satisfied



```
x = 1

while ( x < 10 ){
  print(x)
  x = x + 1
}
```



```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
```

Exercises – Part 3

- Use control flow to print all the values from 1 to 100 that are divisible by 3

- Use control flow to print all the numbers in the Fibonacci series till its 20th element

- Use control flow to print if a numeric variable is a prime number (Intermediate level)

Data analysis

- Most of the formats are accepted in R:
  - xlsx
  - csv
  - spss
  - sql
  - txt
- Each function is explicit and easy to use:
  - read.csv
  - read.spss
- The most general function is read.table and can be used to read most of the formats

Exercises – Part 4

- Use the help function and read the Dataset.csv file in Bb

- Use the following functions to skim our database:
  - How many columns? – ncol
  - How many rows? – nrow
  - How many elements? – length, dim
  - What data types are included? – class
  - Use the head function to see if we have repeated data

Advanced data types

- Matrix
  - They must be rectangular matrices
  - Dimensionality – 2D
  - Same data type values

Advanced data types

- data.frame
  - Different data types values per column
  - How to retrieve a column by name?
    - Variable$FirstColumnName

More functions

- names – Return the column names of a 2D variable
- colnames – Return the column names of a 2D variable
- rownames – Return the row names of a 2D variable
- summary – Returns a brief statistic result of the data contained in a variable

Exercises – Part 5

- Use the help function to understand how to use the following functions:
  - as.factor
  - levels

- Use those functions to know how many different countries were used in the database

Data analysis in R

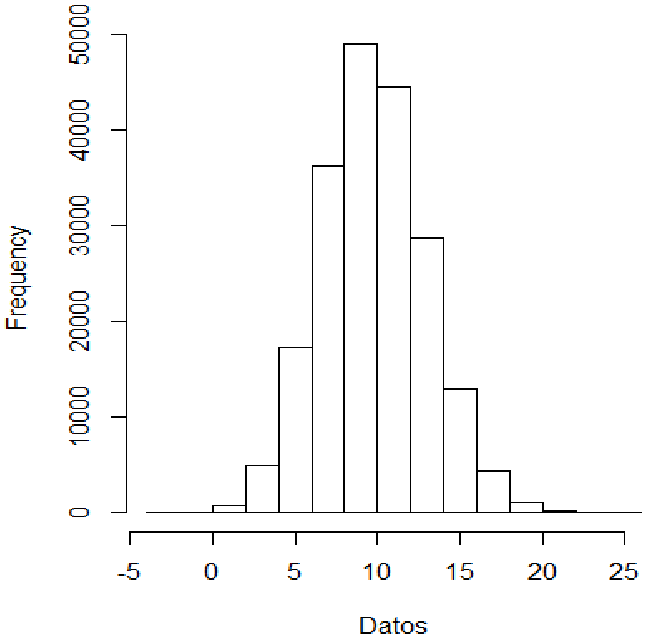
- Exploratory Data Analysis (EDA)
  - Are there anomalies?
  - Do we care about the outliers?
  - Do we have repeated data?
  - Do I need to clean the database?
  - Are all the variables in the database of interest?
  - Lets see if all the assumptions are satisfied!

Visualization

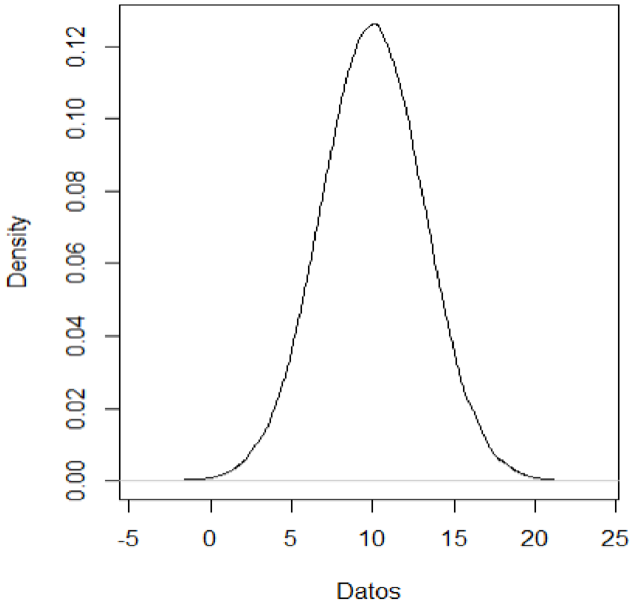
- It is necessary to find the best way to show the results
  - Lists/tables
  - Summaries
  - Plots

- What kind of plots?
  - Histograms – hist
  - Densities – density
  - Scatter plots – plot
  - Box plots – boxplot

Exploratory data analysis

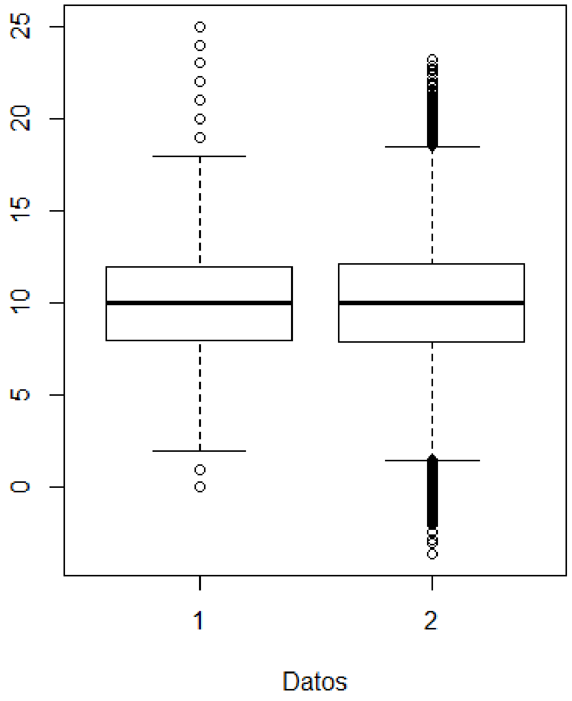


Histogram

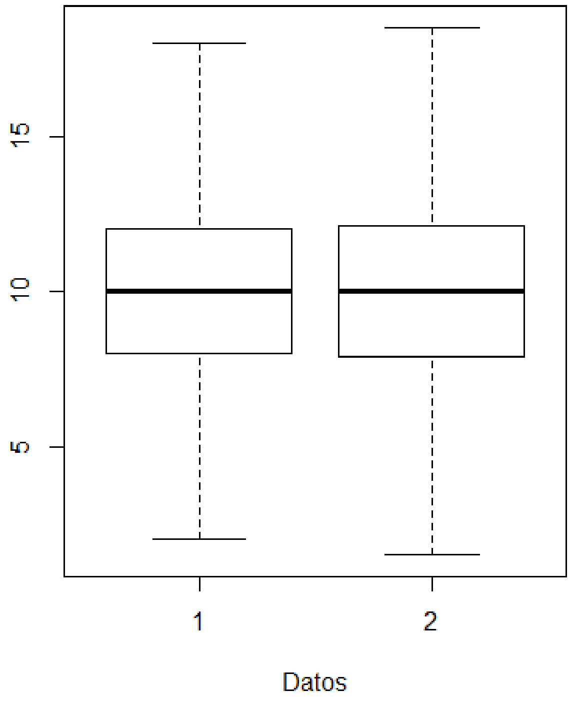


Density

Exploratory data analysis



Boxplot



Boxplot without outliers

Dataset.csv

Lets explore this database!

Dataset.csv

- Lets see all the dataypes included in the dataset

- Lets explore the numeric data with different plots
- Is there something wrong?
  - Do we need to filter our data?

- Research question:
  - Is there a difference between year 2010 total deaths compared to 2012?
  - EDA plots

Dataset.csv

- Plot all the death values per year in only one plot without outliers

- Plot parameters:
  - main – Plot name
  - xlab – x-label
  - ylab - y-label
  - col - Colors

Homework 1

- Load the dataset.csv
- The new dataset must have:
  - Only relevant variables (Not repeated data accross columns)
  - Only the counts related to “All causes of death”
  - And the measure must be “Number of total deaths”
  - All years and all countries must be included
  - Save the new database in another csv file called “filteredDatabase.csv”
- TIP: Use matrix and submatrix notation to get the desired values and the write.csv function

Homework 1

- Select another cause of death and measure to get a second dataset.
- With this new data set:
- Plot the boxplot of these deaths in the first year reported and compare it with the last year (both boxplots in the same plot)
- Upload the .R file with all your code in the Assignment tab in Bb
  - Assignment - R Homework
  - 1 per pair. Both names in a comment at the beginning of the .R file
  - DO NOT upload the csv file