R Language

What is R

R is popular programming language used for statistical computing and graphical presentation
Its most common use is to analyze and visualize data.

R Variables

```
name <- "John"
age <- 48
```

age: variables; 48: values

Print / Output Variables

To print a variable just type the name

```
age  # output: 48
```

There are times we must use the print() function to output code, for ex when working with for loops

Concatenate Elements

Join 2 or more element using the paste() function

```
text <- "awesome"
paste("R is", text)
```

R Comments

```
# This is a comment
```

Basic Data Types

- numeric - (10.5, 55, 787)
- integer - (1L, 55L, 100L)
- complex - (9 + 3i)
- character - ("K", "Hello", "11")
- logical - (TRUE or FALSE)

we can use the class() function to check the data type of a variable

Type Conversion

- as.numeric()
- as.integer()
- as.logical()
- as.character()
- as.complex

→ Arithmetic Operators

- a + b Sums 2 variables
- a - b Substracts 2 variables
- a * b multiply 2 variables
- a / b divide 2 variables
- a ^ b power (a^b)
- a %% b Remainder
- a %/% b Integer division

→ Relational Operators

- a == b equality
- a != b inequality
- a > b greater than
- a < b lower than
- a >= b greater or equal
- a <= b lower or equal

→ Logical Operators

- ! logical NOT
- & Element-wise logical AND
- && logical AND
- | Element-wise logical OR
- || logical OR

→ Other Operators

- : - creates series of nb - x <- 1:10
- %in% element belong to vector?
- %*% Matrix Multiplication
- $ Access objects stored within an object

→ Math Functions

- log(x) - Natural log
- exp(x) - Exponential
- max(x) - Largest element
- min(x) - Smallest element
- round(x, n) - Round to n decimal place
- signif(x, n) - Round to n significant figure
- corr(x, y) - Correlation between 2 vectors
- sum(x) - sum of a vector
- mean(x) - mean of a vector
- median(x) - median of a vector
- quantile(x) - percentage quantile
- rank(x) - rank of elements
- var(x) - variance of a vector
- sd(x) - Standard deviation of a vector
- sqrt() - Square root of a nb
- abs() - absolute value

→ R Strings

- cat(): concatenates multiple strings and outputs the result without any separation or space.
  ↳ cat("Hello", "World!")
  # Output: Hello World!

- paste(): concatenates multiple strings and allows to specify a separator between them using the 'sep' argument
  ↳ paste("Hello", "World!", sep=" ")
  # Output: "Hello World!"

- Additionally 'paste()' can handle vectors and other dsa by pasting corresponding elements together
  ↳ x<-c("apple", "banana", "cherry")
  y<-c("red", "yellow", "red")
  paste(x, y, sep = " - ")
  # Output: "apple - red" "banana - yellow" "cherry - red"
  cat(x, y)
  # Output: apple banana cherry red yellow red

- nchar(x): nb of characters in a string

- grep(patern, x): search for a specified pattern in a character vector or a string. It return a numeric vector containing the indices of the elements that match the patter. If we add the 'value = True' parameter, it returns a character vector containing the actual matching elements

- grepl(patern, x): test if a pattern is present in a string or in each element of a character vector. It returns a logical vector of the same length as the input vector where 'TRUE' => match 'FALSE' => no match
  ↳ x<-c("apple", "banana", "orange")
  grep("an", x) # Output: 2 3
  grep("an", x, value = TRUE)
  # Output: "banana" "orange"
  grepl("an", x)
  # Output: FALSE TRUE TRUE

If Statements

```
if (condition) {
    Do something
} else {
    Do something different
}
i <- 5
if (i > 3) {
    print ('Yes')
} else {
    print ('No')
} # output: "Yes"
```

While Loop

```
while (condition) {
    Do Something
}
i <- 0
while (i < 5) {
    print (i)
    i <- i + 1
} # output: 0
          1
          2
          3
          4
```

For Loop

```
for (variable in sequence) {
    Do Something
}
for (i in 1:4) {        # 1 2 3 4
    j <- i + 10
    print (j)
} # output: 11
           12
           13
           14
```

Functions

```
function_name <- function (var) {
    Do Something
    return (new_variable)
}
square <- function (x) {
    squared <- x * x
    return (squared)
}
# call the function
square (4) # output: 16
```

R Vectors

Creating vectors

Input	Output	Description
c(1,3,5)	1 3 5	Creates a vector using elements separated by commas
1:7	1 2 3 4 5 6 7	Creates a vector of integers between two numbers
seq(2,8,by=2)	2 4 6 8	Creates a vector between two numbers, with a specified interval between each element
rep(2,8,times=4)	2 8 2 8 2 8 2 8	Create a vector of given elements repeated a nb of times
rep(2,8,each=3)	2 2 2 8 8 8	Creates a vector of given elements repeating each element a nb of times

Vector Functions

- sort(my_vector) - sort
- rev(my_vector) - reverse
- table(my_vector) - count the values in a vector
- unique(my_vector) - remove duplicate values in a vector
- length(my_vector) - how many items a vector has

Selecting Vector Elements

By Position

x[4] - The fourth element

x[-4] - All except the fourth

x[2:4] - Elements two to four

x[-(2:4)] - All element except two to four

x[c(1,5)] - Element one and five

By Value

x[x == 10] - Elements wich are equal to

x[x < 0] - All elements less than 0

x[x %in% c(1,2,5)] - Elements in the set 1, 2, 5

→ R Lists

l <- list(x=1:5, y=c('a','b'))

A list is collection of elements wich can be of different types

↳ Access Lists

- l[[2]] - second element of l
- l[1] - New list with the 1st element
- l$x - Element named x
- l['y'] - New list with element named y

→ R Matrices

m <- matriv(x, nrow=3, ncol=


m[2,] select a row


m[,1] select a column


m[2,3] select an element

- t(m): transpose
- m %*% n: matrix multiplication

→ Data Frames

A Data Frame has the variable of a data set as columns and the observations as rows.

```
# This creates the data frame df
  seen on the right
df <- data.frame(x=1:3, y=c("h","i","j"),
  z=12:14)
```

x	y	z
1	h	12
2	i	13
3	j	14

```
# This select all columns of the third row
df[3, ]
```

x	y	z
1	h	12
2	i	13
3	j	14

```
# This select the column z
df$z
```

x	y	z
1	h	12
2	i	13
3	j	14

this selects all rows of the second column

```
df [ , 2]
```

x	y	z
1	h	12
2	i	13
3	j	14

This select the third element (column) of the second row

```
df [ 2 , 3 ]
```

x	y	z
1	h	12
2	i	13
3	j	14

Understanding a Dataframe

View (df) - See the full dataframe

head (df) - See the first 6 rows

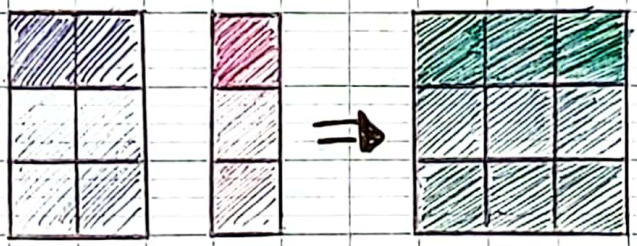
nrow (df) - Nb of rows

ncol (df) - Nb of columns

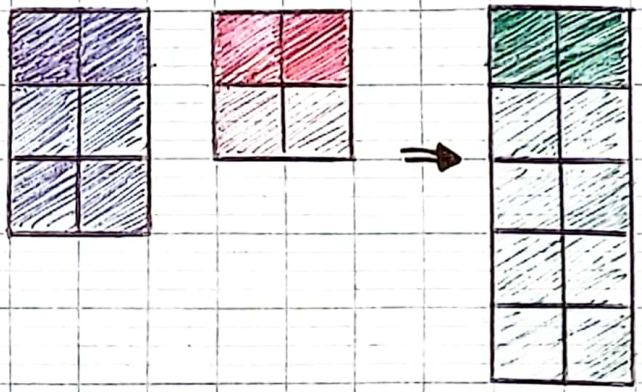
dim (df) - Nb of columns and rows

Manipulating Data Frames

cbind - Bind columns



rbind - Binds rows



Statistics

. lm (x ~ y, data = df) : linear model

. summary : get info about a model

. t.test (x, y) : t-test for diff between means

. aov : Analysis of Variance (ANOVA)

. pairwise.t.test : Perform a t-test for paired data

. prop.test : Test for diff between proportions

Read A File

```
df <- read.csv ('File.csv')
```