

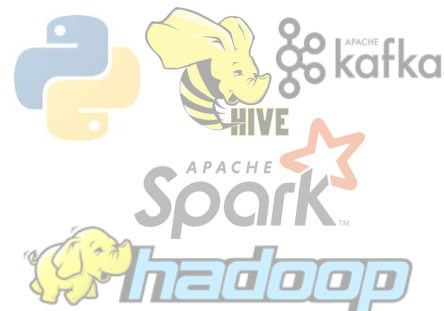


BIG DATA & ANALYTICS

Program Learning

Programa Integral en BigData & Analytics

Business Intelligence (30hr)	Advanced Analytics (40hr)	Bigdata Technologies (42hr)	Agile Data (12hr)	Workshops (12hr)
DataWareHouse	Business Analytics	Big Data Fundamentals	SCRUM Data	Analytics with Unstructure Data
SQL Essentials	R Essentials	Python Essentials	DataOps	
Data Modeling	Modeling Process	Tecnologías Cloud y OnP	Data Transformations	
Data Engineering	Deploy and Monitoring	Storage technologies		
Data Visualization	Machine Learning Intro	Batch/RealTime Process		





Essentials for Business Analytics

“ ... Generando valor a través de los Datos ”

CONTENIDO

ASPECTOS
GENERALES

ELEMENTOS
DEL LENGUAJE

MANIPULACION
DE DATOS

VISUALIZACIÓN
DE DATOS

CONTENIDO

ASPECTOS
GENERALES

ELEMENTOS
DEL LENGUAJE

MANIPULACION
DE DATOS

VISUALIZACIÓN
DE DATOS

Aspectos Generales

R es un entorno de software libre para computación estadística y gráficos. Compila y se ejecuta en una amplia variedad de plataformas UNIX, Windows y MacOS

- La versión de R 3.x en adelante soporta enteros de 64 bits
- R es case sensitive
- Los índices en R empiezan en 1

Soporte y consideraciones

Accessing the help files

?mean

Get help of a particular function.

help.search('weighted mean')

Search the help files for a word or phrase.

help(package = 'dplyr')

Find help for a package.

More about an object

str(iris)

Get a summary of an object's structure.

class(iris)

Find the class an object belongs to.

Uso de Paquetes

install.packages('dplyr')

Download and install a package from CRAN.

library(dplyr)

Load the package into the session, making all its functions available to use.

dplyr::select

Use a particular function from a package.

data(iris)

Load a built-in dataset into the environment.

Working Directory

getwd()

Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')

Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Tipos

La conversión entre tipos de datos comunes en R. Siempre puede ir desde un valor más alto en la tabla a un valor más bajo.

<code>as.logical</code>	<code>TRUE, FALSE, TRUE</code>	Boolean values (TRUE or FALSE).
<code>as.numeric</code>	<code>1, 0, 1</code>	Integers or floating point numbers.
<code>as.character</code>	<code>'1', '0', '1'</code>	Character strings. Generally preferred to factors.
<code>as.factor</code>	<code>'1', '0', '1', levels: '1', '0'</code>	Character strings with preset levels. Needed for some statistical models.

Funciones matemáticas

<code>log(x)</code>	Natural log.	<code>sum(x)</code>	Sum.
<code>exp(x)</code>	Exponential.	<code>mean(x)</code>	Mean.
<code>max(x)</code>	Largest element.	<code>median(x)</code>	Median.
<code>min(x)</code>	Smallest element.	<code>quantile(x)</code>	Percentage quantiles.
<code>round(x, n)</code>	Round to n decimal places.	<code>rank(x)</code>	Rank of elements.
<code>signif(x, n)</code>	Round to n significant figures.	<code>var(x)</code>	The variance.
<code>cor(x, y)</code>	Correlation.	<code>sd(x)</code>	The standard deviation.

CONTENIDO

ASPECTOS
GENERALES

ELEMENTOS
DEL
LENGUAJE

MANIPULACION
DE DATOS

VISUALIZACIÓN
DE DATOS

Variables y Entorno

Variable Assignment

```
> a <- 'apple'  
> a  
[1] 'apple'
```

The Environment

`ls()`

List all variables in the environment.

`rm(x)`

Remove x from the environment.

`rm(list = ls())`

Remove all variables from the environment.

You can use the environment panel in RStudio to browse variables in your environment.

Vectores

Creating Vectors

<code>c(2, 4, 6)</code>	2 4 6	Join elements into a vector
<code>2:6</code>	2 3 4 5 6	An integer sequence
<code>seq(2, 3, by=0.5)</code>	2.0 2.5 3.0	A complex sequence
<code>rep(1:2, times=3)</code>	1 2 1 2 1 2	Repeat a vector
<code>rep(1:2, each=3)</code>	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

sort(x)

Return x sorted.

table(x)

See counts of values.

rev(x)

Return x reversed.

unique(x)

See unique values.

Selección de elementos de un vector

By Position

`x[4]` The fourth element.

`x[-4]` All but the fourth.

`x[2:4]` Elements two to four.

`x[-(2:4)]` All elements except two to four.

`x[c(1, 5)]` Elements one and five.

By Value

`x[x == 10]` Elements which are equal to 10.

`x[x < 0]` All elements less than zero.

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

Named Vectors

`x['apple']` Element with name 'apple'.

Matrices

```
m <- matrix(x, nrow = 3, ncol = 3)
```

Create a matrix from x.



`m[2,]` - Select a row



`m[, 1]` - Select a column



`m[2, 3]` - Select an element

`t(m)`

Transpose

`m %*% n`

Matrix Multiplication

`solve(m, n)`

Find x in: $m * x = n$

Listas

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

A list is a collection of elements which can be of different types.

`l[[2]]`

Second element
of l.

`l[1]`

New list with
only the first
element.

`l$x`

Element named
x.

`l['y']`

New list with
only element
named y.

Strings y Factores

Strings

Also see the **stringr** package.

<code>paste(x, y, sep = ' ')</code>	Join multiple vectors together.
<code>paste(x, collapse = ' ')</code>	Join elements of a vector together.
<code>grep(pattern, x)</code>	Find regular expression matches in x.
<code>gsub(pattern, replace, x)</code>	Replace matches in x with a string.
<code>toupper(x)</code>	Convert to uppercase.
<code>tolower(x)</code>	Convert to lowercase.
<code>nchar(x)</code>	Number of characters in a string.

Factors

<code>factor(x)</code>	<code>cut(x, breaks = 4)</code>
Turn a vector into a factor. Can set the levels of the factor and the order.	Turn a numeric vector into a factor by 'cutting' into sections.

Data Frames

Also see the
dplyr package.

Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
```

A special case of a list where all elements are the same length.

x	y
1	a
2	b
3	c

List subsetting

df\$x

df[[2]]

Understanding a data frame

`View(df)`

See the full data frame.

`head(df)`

See the first 6 rows.

Data Frames Subsettings

Matrix subsetting

`df[, 2]`



`df[2,]`



`df[2, 2]`



`nrow(df)`
Number of rows.

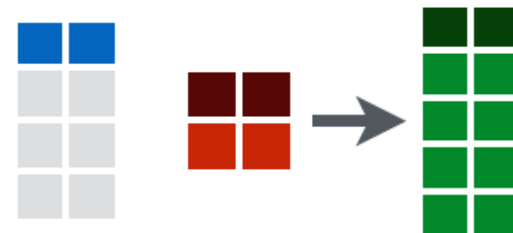
`ncol(df)`
Number of columns.

`dim(df)`
Number of columns and rows.

`cbind` - Bind columns.



`rbind` - Bind rows.



Statistics

lm(y ~ x, data=df)

Linear model.

glm(y ~ x, data=df)

Generalised linear model.

summary

Get more detailed information
out a model.

t.test(x, y)

Perform a t-test for
difference between
means.

pairwise.t.test

Perform a t-test for
paired data.

prop.test

Test for a
difference
between
proportions.

aov

Analysis of
variance.

Distribuciones

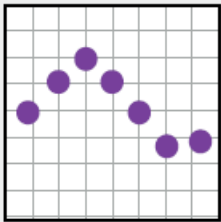
Distributions

	Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>

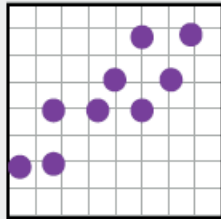
Gráficos

Plotting

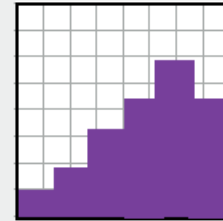
Also see the **ggplot2** package.



plot(x)
Values of x in
order.



plot(x, y)
Values of x
against y.



hist(x)
Histogram of
x.

Instrucciones de Control

For Loop

```
for (variable in sequence){  
  Do something  
}
```

Example

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

While Loop

```
while (condition){  
  Do something  
}
```

Example

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

If Statements

```
if (condition){  
  Do something  
} else {  
  Do something different  
}
```

Example

```
if (i > 3){  
  print('Yes')  
} else {  
  print('No')  
}
```

Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

Conditions

a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

CONTENIDO

ASPECTOS
GENERALES

ELEMENTOS
DEL LENGUAJE

MANIPULACION
DE DATOS

VISUALIZACIÓN
DE DATOS

Import Data

Reading and Writing Data

Also see the **readr** package.

Input	Ouput	Description
<code>df <- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>	Read and write a delimited text file.
<code>df <- read.csv('file.csv')</code>	<code>write.csv(df, 'file.csv')</code>	Read and write a comma separated value file. This is a special case of read.table/write.table.
<code>load('file.RData')</code>	<code>save(df, file = 'file.Rdata')</code>	Read and write an R data file, a file type special for R.

