CHEAT SHEET - INTRO TO R

Accessing help

Get help of a particular function:
`?round`
`help(round)`

Using packages

Download and install a package from CRAN:
`install.packages("Publish")`

Load the package into the session
making all its functions available to use:
`library(Publish)`

Working Directory

The Working Directory is the file path that `R` will use as a starting point for file paths.
Find the current working directory:
`getwd()`

Set the working directory:
`setwd(//file/path")`

Operators

Arithmetic Operators	
a+b	sum two variables
a-b	subtract two variables
a*b	multiply two variables
a/b	divide two variables
a ^ b	exponentiation of a variable
Comparison operators	
a==b	test for equality
a!=b	test for inequality
a>=b	test for disequality
%in %	Identifies whether an element belongs to a vector
Logical operators	
&	AND
|	OR
!	NOT

Math functions

`log(x)`	Natural log	`exp(x)`	Exponential
`sum(x)`	Sum	`prod(x)`	Product
`round(x,digits=n)`	Round to n decimal		

Types

Check type of an object:
`typeof()`

`as.numeric()`	1,2.5,3	integers or decimal numbers
`as.character()`	"b", "treated"	Character strings
`as.logical()`	TRUE,FALSE	Logical values
`as.factor()`	"B", "treated"	Character strings with preset levels

Vectors

Creating vectors:

`c(1,2,3)`	join elements into a vector
`a:b`	sequence of integers from a to b
`seq(a,b, by=c)`	sequence from a to b with step c
`rep(1:2, n)`	repeat a vector n times

Accessing a vector:

`v[i]`	select element i in vector v
`v[ - i ]`	all elements but the ith
`v[2:4]`	elements 2 to 4
`v[c(2,4)]`	elements 2 and 4
`v[v==2]`	elements that respects the condition

Functions for vectors:

`length(v)`	number of elements in vector v
`rbind(v1,v2)`	bind by row two or more vectors
`cbind(v1,v2)`	bind by column two or more vectors
`factor(v)`	transform v into a factor
`cut(v, breaks=c(0,2,5),`	create a categorical variable
`labels=c("low","high")`	by defining the cut-off (breaks) and the labels
`ifelse(`*condition*`,a, b)`	check the condition assigning a if TRUE and b if FALSE

Data.Frame

Creating a data.frame:
`db<-data.frame(ID=1:3, trt=c( .A ","B ",.A "), weight=c(62,73,58)`

Accessing a data.frame:

`db$ID`	Select the column ID
`db[, c]`	Select columns indexed by c
`db[r, ]`	Select row indexed by r
`db[r,c]`	Select a specific element
`db[r, c(ÏD")]`	select ID of row r
`db$ID[r]`	select ID of row r

Functions for data.frame:

`dim() nrow(), ncols()`	number of rows and/or columns
`head()`	print first 6 lines of the data.frame
`str()`	internal structure of the data.frame
`summary()`	summary of each variable of the data.frame
`subset(db, `*condition*`)`	subset by condition
`reshape()`	long to wide and vice-versa
`merge(db1,db2, by)`	join two data sets by one or more columns

Descriptive analysis

`mean(x)`	Mean	`sd(x)`	standrad deviation
`max(x)`	Largest element	`min(x)`	Smallest element
`median(x)`	Median	`quantile(x)`	Percentage quantiles
`table(x)`	count by category	`prop.table()`	proportions by category

Calculation by group:
`tapply()`
`aggregate()`

Graphics

`plot(x)`	plot valuse of x in order
`plot(x,y)`	scatterplot for x and y
`line()`	add a line to the plot
`points()`	add points to the plot
`abline()`	add vertical/horizontal line to the plot
`legend()`	add legend to the plot
`hist()`	plot histogram for a variable
`boxplot(`$x \sim y$ `)`	boxplot of a continuous variable x by levels of y

Specifications for all graphics:

`col`	color(s) of the plot (factor)
xlim,ylim	margins of the plot
xlab,ylab	labels of the axes
`main`	title of the plot
`lty,pch`	type of lines and points
`lwd`	line width

Combine together several plots:
`par(mfrow=c(nrow,ncol))` + `dev.off()`