

# R Reference Card

A. Shipunov, 20140301

## Miscellaneous

q(): quit  
; separate commands  
c(a, b): concatenate, vectorize  
savehistory(file1): save command history  
mat1[, 2]: second column of matrix or data frame mat1  
mat1[, 2:5] or mat1[, c(2, 3, 4, 5)]: columns 2–5  
list1[[-5]]: all elements of list except fifth list1  
df1\$a1: variable (column) a1 in data frame df1  
NA: missing value is.na(x1): true if x1 == NA  
install.packages("package1"): install package package1  
library(xlsx): load package xlsx

## Help

help(com1) or ?com1: help about command com1  
??"topic1": finds topic1 in all help files (slow!)  
args(com1): which arguments command com1 have  
example(com1): run examples for command com1  
help(package = rpart): help for the package, e.g. rpart

## Entering and saving data

obj2 <- edit(obj1): edit object obj1 in the external editor defined in options(), edited data will go to obj2  
fix(obj1): same, but obj1 will change  
fix(df1): edit data frame df1 in the embedded table editor  
read.table("file1", ...): read data from file  
save(file = "file1", list = ls() and load("file1")): save (in binary format) and load all objects  
dir(...) and setwd(): list files in directory, go to another  
sink("file1"): output to file1 until sink()  
source("file1.r"): run commands from file file1.r  
write.table(x1, "file1"): write object x1 to the file file1

## Manage variables and objects

1:3 or c(1, 2, 3): vector 1, 2, 3  
rep(x1, n1): repeat vector x1 n1 times  
seq(n1, n2, n3): sequence from n1 to n2 by n3 steps  
sample(x1, n1): sample n1 elements from x1  
attach(x1): place components of x1 in search path  
detach(x1): remove x1 from search path  
with(x1, ...): manage components of x1 within environment  
ls(): list all active objects  
rm(obj1): remove object obj1  
dim(mat1): number of rows and columns in matrix mat1  
dimnames(mat1), or names(df1) and row.names(df1): names of rows and columns of mat1 or data frame df1  
length(v1): length of vector v1  
nrow(mat1): how many rows?  
str(obj1): structure of object obj1  
as.data.frame(x1), as.matrix(x1): conversion  
cbind(a1, b1, c1), rbind(a1, b1, c1): join columns and rows into matrix  
cut(v1, c(0, 100, 200), labels = c("small", "big")): split vector v1 in two intervals  
data.frame(v1, v2): list from vectors v1 and v2  
matrix(vector1, r1, c1): vector1 to matrix with r1 rows and c1 columns  
merge(df1, df2): merge two data frames  
t(mat1): transpose (rotate 90°) matrix or data frame  
grep("topic1", x1): search topic1 in x1  
gsub("topic1", "topic2", x1): replace topic1 to the topic2 in x1  
paste("cow", "boy", sep = "") outputs "cowboy"  
unique(x1): list of unique elements of x1

## Cycles, conditions and functions

plot(..., pch = ...): 1 ○ 2 △ 3 † 4 × 5 ◇ 6 ▽ 7 ☒ 8 ✱ 9 ⬠ 10 ⊕ 11 ⊗ 12 ⊞ 13 ⊗ 14 ⊞ 15 ■  
16 ● 17 ▲ 18 ◆ 19 ● 20 ● 21 ○ 22 □ 23 ◇ 24 △ 25 ▽ \* \* . . a a ? ? 0 □

if (condition1) ...else ...: condition  
ifelse(condition1, yes, no): vectorized condition  
for (i1 in vector1) command1: cycle  
fun1 <- function(args1) something1: define function  
Recode <- function(var, from, to) {  
  x <- as.vector(var); x.tmp <- x  
  for (i in 1:length(from))  
  {x <- replace(x, x.tmp == from[i], to[i])}  
  if(is.factor(var)) factor(x) else x}: useful function for fast merge

## Logic and math

!<, &, |, ==: "not less", "and", "or", "equal"  
x1 %in% x2, match(x1, x2): if elements of x1 are in x2  
is.factor(obj1), is.matrix(obj1), is.vector(obj1): check the type of object obj1  
mat1[mat1 > 0]: elements of mat1 which are positive  
\*, ^, sqrt(pi), abs(-3), log(1): multiplication, degree, √π, 3, natural logarithm  
round(x1): round

## Descriptive statistics

summary(x1); IQR(x1); cumsum(x1); diff(x1);  
  fivenum(x1); mad(x1); max(x1); mean(x1); median(x1);  
  min(x1); prod(x1); sd(x1); sum(x1); var(x1): statistics  
colSums(mat1): calculate sum of every column  
aggregate(...): pivot table  
rev(x1), order(x1), scale(x1), sort(x1): reverse etc.  
table(x1, x2): cross-tabulation  
apply(x1, n1, f1): apply function f1 (e.g., mean) to all rows (n1 = 1) or columns (n2 = 2) of x  
tapply(x1, list1, f1): apply function f1 to x1 grouping by list1  
lapply(...); mapply(...): other useful vectorized functions

## Inferential statistics

ks.test(...); prop.test(...); t.test(...),  
  wilcox.test(...): tests  
chisq.test(tab1): χ<sup>2</sup>-test for table tab1  
cor(df1): correlations between all columns of the data frame  
cor.test(x1, x2): correlation test  
lm(...); glm(...); aov(...); anova(...): linear and non-linear models, analyses of variation (ANOVA)  
lm(y ~ x + z, data = ...): formula interface to the linear model, y responses on two variables, x and z  
predict(model1): predict from model

## Multivariate statistics

cmdscale(...): metric multidimensional scaling (MDA)  
hclust(...): hierarchical cluster analysis  
princomp(...); prcomp(...): principal component analyses (PCA)

## Plots

plot(...), barplot(...), boxplot(...), stem(...),  
  hist(...): main plots  
plot(..., type = "p|l|s|n"): points, lines, stairs and no plot  
plot(..., col = 0|1|2|3): white, black, red, green color  
plot(..., lty = 1|2): no lines, straight line, dashed line  
plot(..., cex = 1|2): normal dot size, double dot size  
png("file1.png|pdf"): write subsequent plot into file1.png|pdf until dev.off()  
identify(...): reveal information from points using mouse  
lines(...); points(...): add lines then points  
legend(..., legend = "..."): add legend  
qqnorm(vec1); qqline(vec1): check normality