**R reference card**, by Jonathan Baron

Parentheses are for functions, brackets are for indicating the position of items in a vector or matrix. (Here, items with numbers like `x1` are user-supplied variables.)

**Miscellaneous**

`q()`: quit `<-`: assign `INSTALL` package: install package `m1[,2]` column 2 of matrix `m1` `m1[,2:5]` or `m1[,c(2,3,4,5)]`: columns 2–5 `m1$a1`: variable `a1` in data frame `m1` `NA`: missing data `is.na`: true if data missing `library(mva)`: load (e.g.) the `mva` package

**Help**

`help(command1)`: get help with `command1` (NOTE: USE THIS FOR MORE DETAIL THAN THIS CARD CAN PROVIDE.)

`help.start()`: start browser help

`help(package=mva)`: help with (e.g.) package `mva`

`apropos("topic1")`: commands relevant to `topic1` `example(command1)`: examples of `command1`

**Input and output**

`source("file1")`: run the commands in `file1`. `read.table("file1")`: read in data from `file1` `data.entry()`: spreadsheet `scan(x1)`: read a vector `x1` `download.file(url1)`: from internet `url.show(url1)`, `read.table(url(url1))`: remote input `sink("file1")`: output to `file1`, until `sink()` `write(object, "file1")`: writes an object to `file1` `write.table(dataframe,"file1")`: writes a table

**Managing variables and objects**

`attach(x1)`: put variables in `x1` in search path `detach(x1)`: remove from search path `ls()`: lists all the active objects. `rm(object)`: removes object `dim(matrix1)`: dimensions of matrix `dimnames(matrix1)`: names of dimensions of `x1` `length(vector1)`: length of vector `1:3`: the vector `1,2,3` `c(1,2,3)`: creates the same vector `rep(x1,n1)`: repeats the vector `x1` `n1` times `cbind(a1,b1,c1)`, `rbind(a1,b1,c1)`: binds columns or rows into a matrix `merge(df1,df2)`: merge data frames `matrix(vector1,1,c1)`: make `vector1` into a matrix with `1` rows and `c1` columns `data.frame(y1,y2)`: make a data frame from vectors `y1` and `y2` `as.factor()`, `as.matrix()`, `as.vector()`: conversion `is.factor()`, `is.matrix()`, `is.vector()`: what it is `t()`: switch rows and columns `which(x1==a1)`: returns indices of `x1` where `x1==a1`

**Control flow**

`for (i1 in vector1)`: repeat what follows

`if (condition1) ... else ...`: conditional

**Arithmetic**

`%*%`: matrix multiplication `%^%`, `%^`, `sqr()`: integer division, power, modulus, square root

**Statistics**

`max()`, `min()`, `mean()`, `median()`, `sum()`, `var()`: as named `summary(data.frame)`: prints statistics `rank()`, `sort()`: rank and sort `ave(x1,y1)`: averages of `x1` grouped by factor `y1` `by()`: apply function to data frame by factor `apply(x1,n1,function1)`: apply function1 (e.g. `mean`) to `x1` by rows (n1=1) or columns (n2=2) `tapply(x1, list1, function1)`: apply function to `x1` by `list1` `table()`: make a table `tabulate()`: tabulate a vector

**Basic statistical analysis**

`aov()`, `anova()`, `lm()`, `glm()`: linear and nonlinear models, `anova` `t.test()`: `t` test `prop.test()`, `binom.test()`: sign test `chisq.test(x1)`: chi-square test on matrix `x1` `fisher.test()`: Fisher exact test `cor(a)`: show correlations `cor.test(a,b)`: test correlation `friedman.test()`: Friedman test

**Some statistics in mva package**

`prcomp()`: principal components `kmeans()`: kmeans cluster analysis `factanal()`: factor analysis `cancor()`: canonical correlation

**Graphics**

`plot()`, `barplot()`, `boxplot()`, `stem()`, `hist()`: basic plots `matplot()`: matrix plot `pairs(matrix)`: scatterplots `coplot()`: conditional plot `stripplot()`: strip plot `qqPlot()`: quantile-quantile plot `qqnorm()`, `qline()`: fit normal distribution