R Reference Card

This table provides a short overview of (most) R commands presented during the exercise sessions. More information on both the commands and their arguments can be retrieved from the R help pages. These are accessed via `?command` or `help(command)`. If you the corresponding package is not loaded, use instead `??command`. A more comprehensive overview is offered by the R reference card, found at
`http://cran.r-project.org/doc/contrib/Short-refcard.pdf`.

Command	Syntax/Example	Description
Basic Operations		
`library`	`library(x)`	Loads a package with additional functions
`setwd`	`setwd("path")`	Changes working directory to a path
`c`	`c(x, y)`	Concatenate objects to a vector
`seq`	`seq(from=start, to=end, by=steps)`	Generates a sequence with defined steps
`:`	`start:end`	Sequence from start to end, possibly descending
`length`	`length(x)`	Length of a vector
`sum`	`sum(x)`	Sum of all values in $\boldsymbol{x}$
`min, max`	`min(x), max(x)`	Minimum/maximum of vector $\boldsymbol{x}$
`mean`	`mean(x)`	Average value of vector $\boldsymbol{x}$
`sqrt`	`sqrt(x)`	Square root of x
`round`	`round(x, digits=0)`	Rounds x to the chosen number of significant digits
`log10`	`log10(x)`	Logarithm to base 10
`sin, cos, tan`	`sin(x), cos(x), tan(x)`	Sine/cosine/tangent of x
`abs`	`abs(x)`	Absolute value of x
`rep`	`rep(x, times=n)`	Replicates number/vector n times
`"`	`"text"`	Denote a string (i. e. a sequence of characters)
`TRUE, FALSE`	`TRUE, FALSE`	Literals denoting boolean values
Data Collection		
`names`	`names(DataFrame)`	Column names of a data frame
`colnames`	`colnames(DataFrame)`	Column names of a data frame
`head`	`head(DataFrame)`	Shows the first 6 rows of a data frame
`$`	`DataFrame$Columnname`	Accessing specific columns of a data frame
`cbind, rbind`	`cbind(x), rbind(x)`	Binding vectors column-wise or row-wise to form matrix
`as.data.frame`	`as.data.frame(x)`	Coerces an object to a data frame
`read.csv`	`read.csv("file.csv", sep=",", ↩ header=TRUE)`	Reading a csv file as specified
`read.csv`	`read.csv(file.choose(), sep=",", ↩ header=TRUE)`	Reading a csv file according to manual selection
`str`	`str(x)`	Gives the structure of an object
`dim`	`dim(x)`	Dimensions of a data frame x
`ncol, nrow`	`ncol(DataFrame), nrow(DataFrame)`	Number of columns/rows in data frame or matrix
`numeric`	`numeric(n)`	Create vector of length n filled with zeros
`attributes`	`attributes(x)`	Access an objects attributes
Indexing		
`[]`	`x[n]`	n-th element of vector $\boldsymbol{x}$
	`x[-n]`	Vector $\boldsymbol{x}$ without the n-th element

	`x[1:n]`	Elements 1 to n from vector x
	`x[c(1,3,4)]`	First, third and fourth element of vector x
	`x[x>3]`	Selects elements fulfilling condition, i. e. greater than 3
	`mat[i,j]`	Matrix element in i-th row and j-th column
	`mat[,j]`	j-th column of matrix
	`DataFrame[["Columnname"]]`	Accessing specific columns of a data frame
`which`	`which(x>3)`	Indices of element fulfilling condition
`which.min`	`which(d)`	Index of element with minimum value
	Visualization	
`plot`	`plot(x,y)`	Two-dimensional point plot
	`plot(x, y, type='l')`	Two-dimensional line
`points`	`points(x, y, shape, color)`	Adding points to existing plot
`text`	`text(x, y, data, position)`	Adding text to points
`outer`	`outer(x, y, f)`	Evaluating f for all combinations of x and y
`persp`	`persp(x, y, z)`	Surface in 3D
	`persp(x, y, z, theta, phi)`	3D plot with left/right (θ) and up/down (ϕ) viewing angle
`image`	`image(x, y, z)`	Image colored according to z value
`contour`	`contour(x, y, z, add=TRUE)`	Adding contour lines to existing plot
`trans3d`	`trans3d(x, y, z, pmat=res)`	Adding data points to an existing 3D plot
`ggplot`	`gplot(...)`	Fancy graphics with additional options
	Control Flow	
`function`	`function(){ body }`	Self-defined function
`return`	`return(value)`	Returning a value from within a function
`print`	`print(text)`	Printing text to the screen from within a function
`expression`	`expression(...)`	Creating a symbolic expression for later evaluation
`eval`	`eval(f)`	Evaluating an expression f with global varialbes
	`eval(f, list)`	Evaluating an expression f with a given list of variables
`if, else`	`if, else`	Conditional statements
`ifelse`	`ifelse(cond, expr1, expr2)`	Executing statements depending on a condition
`for`	`for (i in D){ body }`	Loop iterating over a fixed vector of values
`while`	`while (cond){ body }`	Executing a loop body while a condition is fulfilled
`proc.time`	`proc.time()`	Stopwatch for measuring duration
`system.time`	`system.time(expr)`	Measuring execution time of an expression
	Linear Algebra & Numerical Analysis	
`digitsBase`	`digitsBase(num, base=b)`	Converting number from base 10 to another base b
`strtoi`	`strtoi(num, base=b)`	Converting a number from any base b to base 10
`%*%`	`a %*% b`	Dot product, matrix multiplication
`drop`	`drop(A)`	Deleteing dimensions in A with only one value
`t`	`t(A)`	Transpose a matrix A
`diag`	`diag(n)`	Identity matrix of size $n \times n$
`solve`	`solve(A)`	Inverse of a matrix square A if existent
`ginv`	`ginv(A)`	Pseudoinverse of a non-square matrix A
`det`	`det(A)`	Determinant of A if existent
`eigen`	`eigen(A)`	Eigenvalues and eigenvectors of a matrix
`is.positive.definite`	`is.positive.definite(A)`	Tests if matrix A is positive definite
`D`	`D(f, x)`	Derivative of a function f regarding x
`optimHess`	`optimHess(x, f, control)`	Approximating Hessian matrix of f at x

`taylor`	`taylor(f, x0, n)`	Order n Taylor approximation of f at x_0
`polyval`	`polyval(p, x)`	Evaluating a Taylor polynomial at x
Optimization		
`lp`	`lp(direction="min", objective.in, const.mat, const.dir, const.rhs, int.vec)`	Optimize linear function
`solve.QP`	`solve.QP(Dmat, dvec, Amat, bvec, meq=0)`	Minimize quadratic function $-\boldsymbol{d}^T\boldsymbol{b}+\frac{1}{2}\boldsymbol{b}^T D\boldsymbol{b}$ with constraints $A^T\boldsymbol{b} >= \boldsymbol{b}_0$
`optimize`	`optimize(f=fn, interval=c(a,b), ..., maximum=FALSE, tol=0.0001)`	Minimize one-dimensional, non-linear function fn on interval $[a,b]$ with certain tolerance
`optimx`	`optimx(par, fn, ..., method=c("Nelder-Mead","BFGS"), ..., control=list(), ...)`	Optimize multi-dimensional, non-linear function fn
Necessary Packages		
car	`library(car)`	Datasets and companion to applied regression
ggplot2	`library(ggplot2)`	Fancy plots
sfsmisc	`library(sfsmisc)`	Base conversions
MASS	`library(MASS)`	Pseudoinverse and datasets
matrixcalc	`library(matrixcalc)`	Tests for definiteness of matrices
pracma	`library(pracma)`	Taylor approximations
lpSolve	`library(lpSolve)`	Linear programming
quadprog	`library(quadprog)`	Quadratic programming
optimx	`library(optimx)`	Various algorithms for multi-dimensional, non-linear optimization