C Compendium

Elementary

Commands

`ls()` or `objects()`	List objects in workspace
`rm(object)`	Delete `object`
`search()`	Search path

Variable names

Combinations of letters, digits and period. Must not start with a digit. Avoid starting with period.

Assignments

`<-`	Assign value to variable
`->`	Assignment "to the right"
`<<-`	Global assignment (in functions)

Operators

Arithmetic

`+`	Addition
`-`	Subtraction, sign
`*`	Multiplication
`/`	Division
`^`	Raise to power
`%/%`	Integer division
`%%`	Remainder from integer division

Logical and relational

`==`	Equal to
`!=`	Not equal to
`<`	Less than
`>`	Greater than
`<=`	Less than or equal to
`>=`	Greater than or equal to
`is.na(x)`	Missing?
`&`	Logical AND
`|`	Logical OR
`!`	Logical NOT

`&` and `|` are *elementwise*. See "Programming" (p. 258) for `&&` and `||`.

Vectors and data types

Generating

`numeric(25)`	25 zeros
`character(25)`	25 × `""`
`logical(25)`	25 × `FALSE`
`seq(-4,4,0.1)`	Sequence: −4.0 −3.9 3.8 ... 3.9 4.0
`1:10`	Same as `seq(1,10,1)`
`c(5,7,9,13,1:5)`	Concatenation: 5 7 9 13 1 2 3 4 5
`rep(1,10)`	1 1 1 1 1 1 1 1 1 1
`gl(3,2,12)`	Factor with 3 levels, repeat each level in blocks of 2, up to length 12 (i.e., 1 1 2 2 3 3 1 1 2 2 3 3)

Coercion

`as.numeric(x)`	Convert to numeric
`as.character(x)`	Convert to text string
`as.logical(x)`	Convert to logical
`factor(x)`	Create factor from vector `x`

Re. factors, see also "Tabulation, grouping, recoding" (p. 253).

Data frames

`data.frame(height = c(165,185), weight = c(90,65))`	Data frame with two named vectors
`data.frame(height, weight)`	Collect vectors into data frame
`dfr$var`	Select vector `var` in data frame `dfr`
`attach(dfr)`	Put data frame in search path
`detach()`	— and remove it from path

Attached data frames always come *after* `.GlobalEnv` in the search path.

Attached data frames are copies; subsequent changes to `dfr` have no effect.

Numerical functions

Mathematical

`log(x)`	Logarithm of x, natural (base-e) logarithm
`log10(x)`	Base-10 logarithm
`exp(x)`	Exponential function e^x
`sin(x)`	Sine
`cos(x)`	Cosine
`tan(x)`	Tangent
`asin(x)`	Arcsin (inverse sine)
`acos(x)`	
`atan(x)`	
`min(x)`	Smallest value in vector
`min(x1,x2,...)`	minimum over several vectors (one number)
`max(x)`	Largest value in vector
`range(x)`	Like `c(min(x),max(x))`
`pmin(x1,x2,...)`	*Parallel* (elementwise) minimum over multiple equally long vectors
`pmax(x1,x2,...)`	Parallel maximum
`length(x)`	Number of elements in vector
`sum(complete.cases(x))`	Number of non-missing elements in vector

Statistical

`mean(x)`	Average
`sd(x)`	Standard deviation
`var(x)`	Variance
`median(x)`	Median
`quantile(x,p)`	Quantiles
`cor(x,y)`	Correlation

Indexing/selection

`x[1]`	First element
`x[1:5]`	Subvector containing first five elements
`x[c(2,3,5,7,11)]`	Element nos. 2, 3, 5, 7, and 11
`x[y<=30]`	Selection by logical expression
`x[sex=="male"]`	Selection by factor variable
`i <- c(2,3,5,7,11); x[i]`	Selection by numerical variable
`l <- (y<=30); x[l]`	Selection by logical variable

Matrices and data frames

`m[4,]`	Fourth row
`m[,3]`	Third column
`dfr[dfr$var<=30,]`	Partial data frame
`subset(dfr,var<=30)`	Same, often simpler

Input of data

`data(name)`	Built-in data set
`read.table("filename")`	Read from external file

Common arguments to `read.table`

`header=TRUE`	First line has variable names
`sep=","`	Data are separated by commas
`dec=","`	Decimal point is comma
`na.strings="."`	Missing value is dot

Variants of `read.table`

`read.csv("filename")`	Comma separated
`read.delim("filename")`	Tab delimited
`read.csv2("filename")`	Semicolon separated, comma decimal point
`read.delim2("filename")`	Tab delimited, comma decimal point

These all set `header=TRUE`

Missing values

Functions

`is.na(x)`	Logical vector. `TRUE` where `x` has `NA`
`complete.cases(x1,x2,...)`	Neither missing in `x1`, nor `x2`, nor...

Arguments to other functions

`na.rm=`	In statistical functions. Remove missing if TRUE, return `NA` if FALSE.
`na.last=`	In `sort` `TRUE`, `FALSE` and `NA` means, respectively, "last", "first", and "throw away".
`na.action=`	In `lm`, etc., values `na.fail`, `na.omit`, `na.exclude`. Also in `options("na.action")`.
`na.print=`	In `summary` and `print.default`: How to represent `NA` in output.
`na.strings=`	In `read.table()`: Code(s) for `NA` in input.

Tabulation, grouping, recoding

`table(f1,...)`	(Cross)tabulation
`tapply(x,f,mean)`	Table of means
`factor(x)`	Convert vector to factor
`cut(x,breaks)`	Groups from cutpoints for continuous variable

Arguments to `factor`

`levels`	Values of x to code. Use if some values are not present in data, or if the order would be wrong.
`labels`	Values associated with factor levels.
`exclude`	Values to exclude. Default `NA`. Set to `NULL` to have missing values included as a level.

Arguments to `cut`

`breaks`	Cutpoints. Note values of `x` outside `breaks` gives `NA`. Can also be a single number, the number of cutpoints (not very useful).
`labels`	Names for groups. Default is `(0,30]`, etc.
`right`	Right endpoint included? (`FALSE`: left)

Recoding factors

`levels(f) <- names`	New level names
`factor(newcodes[f])`	Combining levels: `newcodes`, e.g., `c(1,1,1,2,3)` to amalgamate the first 3 of 5 groups.

Statistical distributions

Normal distribution

`dnorm(x)`	Density
`pnorm(x)`	Cumulative distribution function, $P(X \leq x)$
`qnorm(p)`	p-quantile, $x : P(X \leq x) = p$
`rnorm(n)`	n (pseudo-)random normally distributed numbers

Distributions

`pnorm(x,mean,sd)`	Normal
`plnorm(x,mean,sd)`	Lognormal
`pt(x,df)`	Student's t
`pf(x,n1,n2)`	F distribution
`pchisq(x,df)`	χ^2
`pbinom(x,n,p)`	Binomial
`ppois(x,lambda)`	Poisson
`punif(x,min,max)`	Uniform
`pexp(x,rate)`	Exponential
`pgamma(x,shape,scale)`	Gamma
`pbeta(x,a,b)`	Beta

Same convention (d-q-r) for density, quantiles, and random numbers as for normal distribution.

Statistical standard methods

Continuous response

`t.test`	One- and two-sample t test
`pairwise.t.test`	Pairwise comparisons
`cor.test`	Correlation
`var.test`	Comparison of two variances (F test)
`lm(y ~ x)`	Regression analysis
`lm(y ~ f)`	One-way analysis of variance
`lm(y ~ f1 + f2)`	Two-way analysis of variance
`lm(y ~ f + x)`	Analysis of covariance
`lm(y ~ x1 + x2 + x3)`	Multiple regression analysis
`bartlett.test`	Bartlett's test (k variances)
Nonparametric:	
`wilcox.test`	One- and two-sample Wilcoxon test
`kruskal.test`	Kruskal–Wallis test
`friedman.test`	Friedman's two-way analysis of variance
`cor.test` variants:	
`method="kendall"`	Kendall's τ
`method="spearman"`	Spearman's ρ

Discrete response

`binom.test`	Binomial test (incl. sign test)
`prop.test`	Comparison of proportions
`prop.trend.test`	Test for trend in relative proportions
`fisher.test`	Exact test in small tables
`chisq.test`	χ^2 test
`glm(y ~ x1+x2+x3, binomial)`	Logistic regression

Models

Model formulas

~	Described by
+	Additive effects
:	Interaction
*	Main effects + interaction (a*b = a + b + a:b)
-1	Remove intercept

Classifications are represented by descriptive variable being a *factor*.

Linear and generalized linear models

`lm.out <- lm(y ~ x)`	Fit model and save result
`summary(lm.out)`	Coefficients, etc.
`anova(lm.out)`	Analysis of variance table
`fitted(lm.out)`	Fitted values
`resid(lm.out)`	Residuals
`predict(lm.out, newdata)`	Predictions for new data frame
`glm(y ~ x, binomial)`	Logistic regression

Diagnostics

`rstudent(lm.out)`	Studentized residuals
`dfbetas(lm.out)`	Change in β if obs. removed
`dffits(lm.out)`	Change in fit if obs. removed

Survival analysis

`S <- Surv(time, ev)`	Create survival object
`survfit(S)`	Kaplan–Meier estimate
`plot(survfit(S))`	Survival curve
`survdiff(S ~ g)`	(Log-rank) test for equal survival curves
`coxph(S ~ x1 + x2)`	Cox's proportional hazards model

Graphics

Standard plots

`plot()`	Scatterplot (and more)
`hist()`	Histogram
`boxplot()`	Box-and-whiskers plot
`stripplot()`	Stripplot
`barplot()`	Bar diagram
`dotplot()`	Dot diagram
`piechart()`	Cakes...
`interaction.plot()`	Interaction plot

Plotting elements

`lines()`	Lines
`abline()`	Line given by intercept and slope (and more)
`points()`	Points
`segments()`	Line segments
`arrows()`	Arrows (NB: `angle=90` for error bars)
`axis()`	Axis
`box()`	Frame around plot
`title()`	Title (above plot)
`text()`	Text in plot
`mtext()`	Text in margin
`legend()`	List of symbols

These are all *added* to existing plots.

Graphical parameters

`pch`	Symbol (*p*lotting *ch*aracter)
`mfrow, mfcol`	Several plots on one (*m*ulti*f*rame)
`xlim, ylim`	Plot limits
`lty,lwd`	Line type/width
`col`	Colour
`cex, mex`	Character size and line spacing in margins

See the help page for `par` for more details.

Programming

Conditional execution	`if(p<0.05)` `  print("Hooray!")`

— with alternative	`if(p<0.05)`
	`  print("Hooray!")`
	`else`
	`  print("Bah.")`

Loop over list	`for(i in 1:10)` `  print(i)`

Loop	`i <- 1`
	`while(i<10) {`
	`  print(i)`
	`  i <- i + 1`
	`}`

User-defined function	`f <- function(a,b,doit=FALSE){`
	`  if (doit)`
	`  a + b`
	`  else`
	`  0`
	`}`

In flow control one uses `a && b` and `a || b` where `b` is only computed if necessary; that is, `if a then b else FALSE` and `if a then TRUE else b`	