Cheatography

R essentials Cheat Sheet

by xeonkai via cheatography.com/31513/cs/9576/

R help functions

help.start()	General help
help(plot) or ?plot	Help on function `plot`
help.search("geo")-	Search help system for instances of string *geo*
example(plot)	Examples of function `plot`
RSiteSearch("geo")	Search for string *geo* in online help manuals and archived mailing lists

R help functions (cont)

apropos("geo", mode="function")	List all available functions with *geo* in their name
data()	List all example datasets in loaded packages
vignette()	List all vignettes for installed packages
vignette("geo")	Display specific vignettes for topic *geo*

Managing R workspace

getwd()	Get working directory
setwd("mydirectory")	Set working directory to *mydirectory*
ls()	List the objects in workspace
rm(objectlist)	Remove one or more objects
help(options)	Learn about available options
options()	View or set current options
history(#)	Display the last # of commands (default=25)

Managing R workspace (cont)

savehistory("myfile")	Save the commands history to *myfile* (default = .Rhistory)
loadhistory("myfile")	Reload a commands history (default = .Rhistory)
save.image("myfile")-	Save the workspace to myfile (default = .RData)
save(objectlist, file = "myfile")	Save specific objects to a file

By **xeonkai**
cheatography.com/xeonkai/

Not published yet.
Last updated 5th November, 2016.

Cheatography

R essentials Cheat Sheet

by xeonkai via cheatography.com/31513/cs/9576/

Managing R workspace (cont)

load("myfile")	Load a workspace in the current session (default = .RData)
q()	Quit R. Will be prompted to save the workspace.

```
setwd("C:/myprojects/project1")
options()
options(digits=3)
x <- runif(20)
summary(x)
hist(x)
savehistory()
save.image()
q()
```

Saving graphic output

pdf("filename.pdf")	PDF file
win.metafile("filename.wmf")	Windows metafile
png("filename.png")	PBG file

Saving graphic output (cont)

jpeg("filename.jpg")	JPEG file
bmp("filename.bmp")	BMP file
postscript("filename.ps")	PostScript file

```
jpeg("rplot.jpeg")
plot(x,y)
dev.off()
```

Packages

.libPaths()	Shows where library is located
library()	Shows the packages saved in the library
search()	Shows which packages have been loaded and ready to use

Packages (cont)

install.packages("ggplot")	Install a package for the first time
update.packages("ggplot")	Update packages that have been installed
installed.packages()	Lists packages with version numbers, dependencies, and other info
library(ggplot)	Load package

Packages (cont)

help(package="ggplot")	Brief description of package and index of functions and datasets

```
help.start()
install.packages("vcd")
help(package="vcd")
library(vcd)
help(Arthritis)
Arthritis
example(Arthritis)
q()
```

Data input

Keyboard input	`mydata <- data.frame(age=numeric(0),gender=character(0),weight=numeric(0))`
	`mydata <- edit(mydata)` **or** `fix(mydata)`
Delimited text file	`mydataframe <- read.table("file", header=logical_value, sep="delimiter", row.names="name", stringsAsFactors=logical_value)`
Excel file	`library(RODBC)`
	`channel <- odbcConnectExcel("myfile.xls")`

Data input (cont)

```
mydataframe <- sqlFetch(channel, "mysheet")
```

```
odbcClose(channel)
```

or

```
library(xlsx)
```

```
workbook <- "c:/myworkbook.xlsx"
```

```
mydataframe <- read.xlsx(workbook, 1)
```

Others	XML, Webscrapping, SPSS, SAS, Stata, netCDF, HDF5, DBMSs

Accessing DBMSs - RODBC package

`odbcConnect(dsn, uid=" ", pwd=" ")`	Open a connection to an ODBC database
`sqlFetch(channel, sqltable)`	Read a table from an ODBC database into a dataframe

Accessing DBMSs - RODBC package (cont)

`sqlQuery(channel, query)`	Submit a query to an ODBC database and return the results
`sqlSave(channel, mydf, tablename=sqtable, append=FALSE)`	Write or update (append=TRUE) a data frame to a table in the ODBC database
`sqlDrop(channel, sqtable)`	Remove a table from the ODBC database
`close(channel)`	Close the connection

```
library(RODBC)
myconn <- odbcConnect("mydsn", uid="Rob", pwd="aardvark")
crimedat <- sqlFetch(myconn, Crime)
pundat <- sqlQuery(myconn, "select * from Punishment")
close(myconn)
```

Data structures

Vectors

Hold numeric, character, or logical data e.g.

```
c(1,2,3,4,5),
c("one","two","three"),
c(TRUE,TRUE,FALSE,FALSE)
```

Matrices

2D array where each element has the same mode

```
mymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns, byrow=logical_value, dimnames=list(char_vector_rownames, char_vector_colnames))
```

Arrays

Similar to matrices but can have more than 2 dimensions

```
myarray <- array(vector, dimensions, dimnames)
```

Data frames

Different columns can contain different modes of data

```
mydata <- data.frame(col1, col2, col3, ...)
```

Data structures (cont)

Factors

Nominal, ordinal or continuous

```
status <- c("Poor", "Improved", "Excellent", "Poor")
```

```
status <- factor(status, order=TRUE, levels=c("Poor","Improved","Excellent"))
```

Lists

Ordered collection of objects

```
mylist <- list(name1=object1, name2=object2, ...)
```

Example code

```
cells <- c(1,26,24,68)
rnames <- c("R1","R2")
cnames <- c("C1","C2")
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE, dimnames=list(rnames,cnames))
dim1 <- c("A1","A2")
dim2 <- c("B1","B2","B3")
```

Example code (cont)

```
dim3 <-
c("C1","C2","C3","C4")
z <- array(1:24, c(2,3,4),
dimnames=list(dim1, dim2,
dim3))
patientID <- c(1,2,3,4)
age <- c(25,34,28,52)
diabetes <-
c("Type1","Type2","Type1",
"Type1")
status <-
c("Poor","Improved","Exce
llent","Poor")
patientdata <-
data.frame(patientID, age,
diabetes, status)
patientdata[1:2]
patientdata[c("diabetes","
status")]
patientdata$age
```

Example code (cont)

```
# cross tabulate diabetes
by status
table(patientdata$diabetes
, patientdata$status)
g <- "My First List"
h <- c(25,26,18,39)
j <- matrix(1:10, nrow=5)
k <-
c("one","two","three")
mylist <- list(title=g,
ages=h, j, k)
mylist[[2]]
mylist[["ages"]]
```

Useful tips

Attach, Detach and With

```
attach (mtcars)
summary(mpg)
plot(mpg,wt)
detach(mtcars)
with(mtcars, {
summary(mpg,disp,wt)
plot(mpg,disp)
plot(mpg,wt)
})
with(mtcars, {
nokeepstats <-
summary(mpg)
keepstats <<-
summary(mpg)
})
nokeepstats # object not
found
```

Useful tips (cont)

```
keepstats # object exists
in global env with
superassignment '<<-'
```

Useful functions

`length(object)`	Number of elements/components
`dim(object)`	Dimensions of an object
`str(object)`	Structure of an object
`class(object)`	Class or type of an object
`mode(object)`	How an object is stored
`names(object)`	Names of components in an object

Useful functions (cont)

`cbind(object, object, ...)`	Combines objects as columns
`rbind(object, object, ...)`	Combines objects as rows
`head(object)`	Lists the first part of the object
`tail(object)`	Lists the last part of the object
`rm(object, object, ...)`	Deletes one or more objects. `rm(list=ls())` will remove most objects.

Useful functions (cont)

`newobject <- edit(object)`	Edits object and saves as newobject
`fix(object)`	Edits in place
`set.seed()`	To have reproducible results

Arithmetic operators

`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`^` or `**`	Exponentiation
`x%%y`	Modulus (x mod y)
`x%/%y`	Integer division

Logical operators

`<`	Less than
`<=`	Less than or equal to
`>`	Greater than
`>=`	Greater than or equal to
`==`	Exactly equal to
`!=`	Not equal to
`!x`	Not `x`
`x | y`	`x` or `y`
`x & y`	`x` and `y`
`isTRUE(x)`	Test if `x` is `TRUE`

By **xeonkai**
cheatography.com/xeonkai/

Not published yet.
Last updated 5th November, 2016.