

R help functions	R help functions (cont)	Managing R workspace	Managing R workspace (cont)
help.start ( )	apropos("geo", mode="function")	getwd()	savehistory("myfile")
help(plot) or ?plot	data()	setwd("mydirectory")	loadhistory("myfile")
help.search("geo")	vignette()	ls()	save.image("myfile")
example(plot)	vignette("geo")	rm(objectlist)	save(objectlist, file = "myfile")
RSiteSearch("geo")		help(options)	
		options()	
		history(#)	



By **xeonkai**  
[cheatography.com/xeonkai/](https://cheatography.com/xeonkai/)

Not published yet.  
 Last updated 5th November, 2016.  
 Page 1 of 5.

Sponsored by **ApolloPad.com**  
 Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>

### 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=sqltable,
append=FALSE)
```

Write or update (append=TRUE) a data frame to a table in the ODBC database

```
sqlDrop(channel,
sqltable)
```

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
or_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)
})
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(o bject)	Number of elements/components
dim(obje ct)	Dimensions of an object
str(obje ct)	Structure of an object
class(ob ject)	Class or type of an object
mode(obj ect)	How an object is stored
names(ob ject)	Names of components in an object



By [xeonkai](https://cheatography.com/xeonkai/)  
[cheatography.com/xeonkai/](https://cheatography.com/xeonkai/)

Not published yet.  
Last updated 5th November, 2016.  
Page 4 of 5.

Sponsored by [ApolloPad.com](https://apollopad.com)  
Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>

Useful functions (cont)		Useful functions (cont)		Logical operators	
<code>cbind(object, object, ...)</code>	Combines objects as columns	<code>newobject</code>	Edits object and saves as <code>newobject</code>	<code>&lt;</code>	Less than
<code>rbind(object, object, ...)</code>	Combines objects as rows	<code>edit(object)</code>	Edits in place	<code>&lt;=</code>	Less than or equal to
<code>head(object)</code>	Lists the first part of the object	<code>set.seed()</code>	To have reproducible results	<code>&gt;</code>	Greater than
<code>tail(object)</code>	Lists the last part of the object	<b>Arithmetic operators</b>		<code>&gt;=</code>	Greater than or equal to
<code>rm(object, object, ...)</code>	Deletes one or more objects. <code>rm(list=ls())</code> will remove most objects.	<code>+</code>	Addition	<code>==</code>	Exactly equal to
		<code>-</code>	Subtraction	<code>!=</code>	Not equal to
		<code>*</code>	Multiplication	<code>!x</code>	Not x
		<code>/</code>	Division	<code>x   y</code>	x or y
		<code>^</code> or <code>**</code>	Exponentiation	<code>x &amp; y</code>	x and y
		<code>x%%y</code>	Modulus (x mod y)	<code>isTRUE(x)</code>	Test if x is TRUE
		<code>x%/%y</code>	Integer division		



By [xeonkai](https://cheatography.com/xeonkai/)  
[cheatography.com/xeonkai/](https://cheatography.com/xeonkai/)

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

Sponsored by [ApolloPad.com](https://apollopad.com)  
 Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>