

Base R

Cheat Sheet

Getting Help

Accessing the help files

?mean
Get help of a particular function.
help.search('weighted mean')
Search the help files for a word or phrase.
help(package = 'dplyr')
Find help for a package.

More about an object

str(iris)
Get a summary of an object's structure.
class(iris)
Find the class an object belongs to.

Using Packages

install.packages('dplyr')
Download and install a package from CRAN.
library(dplyr)
Load the package into the session, making all its functions available to use.
dplyr::select
Use a particular function from a package.
data(iris)
Load a built-in dataset into the environment.

Working Directory

getwd()
Find the current working directory (where inputs are found and outputs are sent).
setwd('C://file/path')
Change the current working directory.
Use projects in RStudio to set the working directory to the folder you are working in.

Vectors

Creating Vectors

c(2, 4, 6)	2 4 6	Join elements into a vector
2:6	2 3 4 5 6	An integer sequence
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

sort(x) Return x sorted.	rev(x) Return x reversed.
table(x) See counts of values.	unique(x) See unique values.

Selecting Vector Elements

By Position

x[4]	The fourth element.
x[-4]	All but the fourth.
x[2:4]	Elements two to four.
x[-(2:4)]	All elements except two to four.
x[c(1, 5)]	Elements one and five.

By Value

x[x == 10]	Elements which are equal to 10.
x[x < 0]	All elements less than zero.
x[x %in% c(1, 2, 5)]	Elements in the set 1, 2, 5.

Named Vectors

x['apple']	Element with name 'apple'.
-------------------	----------------------------

Programming

For Loop

```
for (variable in sequence){  
  Do something  
}
```

Example

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

While Loop

```
while (condition){  
  Do something  
}
```

Example

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

If Statements

```
if (condition){  
  Do something  
} else {  
  Do something different  
}
```

Example

```
if (i > 3){  
  print('Yes')  
} else {  
  print('No')  
}
```

Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

Reading and Writing Data

Also see the **readr** package.

Input	Ouput	Description
df <- read.table('file.txt')	write.table(df, 'file.txt')	Read and write a delimited text file.
df <- read.csv('file.csv')	write.csv(df, 'file.csv')	Read and write a comma separated value file. This is a special case of read.table/write.table.
load('file.RData')	save(df, file = 'file.Rdata')	Read and write an R data file, a file type special for R.

Conditions	a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
	a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

Types

Converting between common data types in R. Can always go from a higher value in the table to a lower value.

<code>as.logical</code>	TRUE, FALSE, TRUE	Boolean values (TRUE or FALSE).
<code>as.numeric</code>	1, 0, 1	Integers or floating point numbers.
<code>as.character</code>	'1', '0', '1'	Character strings. Generally preferred to factors.
<code>as.factor</code>	'1', '0', '1', levels: '1', '0'	Character strings with preset levels. Needed for some statistical models.

Maths Functions

<code>log(x)</code>	Natural log.	<code>sum(x)</code>	Sum.
<code>exp(x)</code>	Exponential.	<code>mean(x)</code>	Mean.
<code>max(x)</code>	Largest element.	<code>median(x)</code>	Median.
<code>min(x)</code>	Smallest element.	<code>quantile(x)</code>	Percentage quantiles.
<code>round(x, n)</code>	Round to n decimal places.	<code>rank(x)</code>	Rank of elements.
<code>signif(x, n)</code>	Round to n significant figures.	<code>var(x)</code>	The variance.
<code>cor(x, y)</code>	Correlation.	<code>sd(x)</code>	The standard deviation.

Variable Assignment

```
> a <- 'apple'
> a
[1] 'apple'
```

The Environment

<code>ls()</code>	List all variables in the environment.
<code>rm(x)</code>	Remove x from the environment.
<code>rm(list = ls())</code>	Remove all variables from the environment.

You can use the environment panel in RStudio to browse variables in your environment.

Matrices

```
m <- matrix(x, nrow = 3, ncol = 3)
Create a matrix from x.
```



`m[2,]` - Select a row



`m[, 1]` - Select a column



`m[2, 3]` - Select an element

`t(m)`

Transpose

`m %*% n`

Matrix Multiplication

`solve(m, n)`

Find x in: $m \cdot x = n$

Lists

```
l <- list(x = 1:5, y = c('a', 'b'))
A list is a collection of elements which can be of different types.
```

`l[[2]]`

Second element of l.

`l[1]`

New list with only the first element.

`l$x`

Element named x.

`l['y']`

New list with only element named y.

Also see the **dplyr** package.

Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
A special case of a list where all elements are the same length.
```

x	y
1	a
2	b
3	c

List subsetting

`df$x`



`df[[2]]`



Understanding a data frame

`View(df)`

See the full data frame.

`head(df)`

See the first 6 rows.

Matrix subsetting

`df[, 2]`



`df[2,]`



`df[2, 2]`

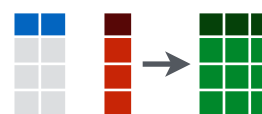


`nrow(df)`
Number of rows.

`ncol(df)`
Number of columns.

`dim(df)`
Number of columns and rows.

`cbind` - Bind columns.



`rbind` - Bind rows.



Strings

Also see the **stringr** package.

<code>paste(x, y, sep = ' ')</code>	Join multiple vectors together.
<code>paste(x, collapse = ' ')</code>	Join elements of a vector together.
<code>grep(pattern, x)</code>	Find regular expression matches in x.
<code>gsub(pattern, replace, x)</code>	Replace matches in x with a string.
<code>toupper(x)</code>	Convert to uppercase.
<code>tolower(x)</code>	Convert to lowercase.
<code>nchar(x)</code>	Number of characters in a string.

Factors

<code>factor(x)</code>	Turn a vector into a factor. Can set the levels of the factor and the order.
<code>cut(x, breaks = 4)</code>	Turn a numeric vector into a factor by 'cutting' into sections.

Statistics

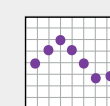
<code>lm(y ~ x, data=df)</code> Linear model.	<code>t.test(x, y)</code> Perform a t-test for difference between means.	<code>prop.test</code> Test for a difference between proportions.
<code>glm(y ~ x, data=df)</code> Generalised linear model.	<code>pairwise.t.test</code> Perform a t-test for paired data.	<code>aov</code> Analysis of variance.
<code>summary</code> Get more detailed information out a model.		

Distributions

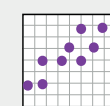
	Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>

Plotting

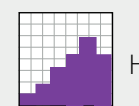
Also see the **ggplot2** package.



`plot(x)`
Values of x in order.



`plot(x, y)`
Values of x against y.



`hist(x)`
Histogram of x.

Dates

See the **lubridate** package.

RStudio IDE :: CHEAT SHEET



Documents and Apps

Open Shiny, R Markdown, knitr, Sweave, LaTeX, .Rd files and more in Source Pane

Check spelling, Render output, Choose output format, Configure render options, Insert code chunk, Publish to server

Jump to previous chunk, Jump to next chunk, Run code, Show file outline, Visual Editor (reverse side)

Jump to section or chunk, Run this and all previous code chunks, Run this code chunk, Set knitr chunk options

Access markdown guide at **Help > Markdown Quick Reference**
See reverse side for more on **Visual Editor**

RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app

Run app, Choose location to view app, Publish to shinyapps.io or server, Manage publish accounts

Source Editor

Navigate backwards/forwards, Open in new window, Save, Find and replace, Compile as notebook, Run selected code

Re-run previous code, Source with or w/out Echo or as a Local Job, Show file outline

Multiple cursors/column selection with **Alt + mouse drag**, Code diagnostics that appear in the margin. Hover over diagnostic symbols for details, Syntax highlighting based on your file's extension, Tab completion to finish function names, file paths, arguments, and more, Multi-language code snippets to quickly use common blocks of code, Jump to function in file, Change file type

Working Directory, Run scripts in separate sessions, Maximize, minimize panes, Ctrl/Cmd + ↑ to see history, R Markdown Build Log, Drag pane boundaries

Tab Panes

Import data with wizard, History of past commands to run/copy, Manage external databases, View memory usage, R tutorials

Load workspace, Save workspace, Clear R workspace, Search inside environment

Choose environment to display from list of parent environments, Display objects as list or grid

Displays saved objects by type with short description, View in data viewer, View function source code

Create folder, Delete file, Rename file, More file options, Change directory

Path to displayed directory, A File browser keyed to your working directory. Click on file or directory name to open.

Version Control

Turn on at **Tools > Project Options > Git/SVN**

Added, Deleted, Modified, Renamed, Untracked

Stage files, Commit staged files, Push/Pull to remote, View History, Current branch

Show file diff to view file differences

Debug Mode

Use **debug()**, **browser()**, or a breakpoint and execute your code to open the debugger mode.

Launch debugger mode from origin of error, Open traceback to examine the functions that R called before the error occurred

Console, Terminal, Jobs

Error, Show Traceback, Rerun with Debug

Package Development

Create a new package with **File > New Project > New Directory > R Package**

Enable roxygen documentation with **Tools > Project Options > Build Tools**

Roxygen guide at **Help > Roxygen Quick Reference**

See package information in the **Build Tab**

Install package and restart R, Run devtools::load_all() and reload changes

Run R CMD check, Customize package build options, Clear output and rebuild, Run package tests

RStudio opens plots in a dedicated **Plots** pane

Files, Plots, Packages, Help, Viewer

Navigate recent plots, Open in window, Export plot, Delete plot, Delete all plots

GUI **Package** manager lists every installed package

Files, Plots, Packages, Help, Viewer

Install Packages, Update Packages, Browse package site, Click to load package with **library()**. Unclick to detach package with **detach()**, Package version installed, Delete from library

RStudio opens documentation in a dedicated **Help** pane

Files, Plots, Packages, Help, Viewer

Home page of helpful links, Search within help file, Search for help file

Viewer pane displays HTML content, such as Shiny apps, RMarkdown reports, and interactive visualizations

Files, Plots, Packages, Help, Viewer

Stop Shiny app, Publish to shinyapps.io, rpubs, RSConnect, ..., Refresh

View(<data>) opens spreadsheet like view of data set

Filter rows by value or value range, Sort by values, Search for value

Click next to line number to add/remove a breakpoint, Highlighted line shows where execution has paused

Environment, History, Build, Git, Tutorial

Run commands in environment where execution has paused, Examine variables in executing environment, Select function in traceback to debug

Console, Terminal, Jobs

Next, Continue, Stop

Step through code one line at a time, Step into and out of functions to run, Resume execution, Quit debug mode



Keyboard Shortcuts

RUN CODE

	Windows/Linux	Mac
Search command history	Ctrl+↑	Cmd+↑
Interrupt current command	Esc	Esc
Clear console	Ctrl+L	Ctrl+L

NAVIGATE CODE

Go to File/Function	Ctrl+.	Ctrl+.
---------------------	--------	--------

WRITE CODE

Attempt completion	Tab or Ctrl+Space	Tab or Ctrl+Space
Insert <- (assignment operator)	Alt+-	Option+-
Insert %>% (pipe operator)	Ctrl+Shift+M	Cmd+Shift+M
(Un)Comment selection	Ctrl+Shift+C	Cmd+Shift+C

MAKE PACKAGES

	Windows/Linux	Mac
Load All (devtools)	Ctrl+Shift+L	Cmd+Shift+L
Test Package (Desktop)	Ctrl+Shift+T	Cmd+Shift+T
Document Package	Ctrl+Shift+D	Cmd+Shift+D

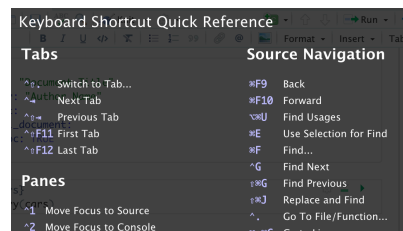
DOCUMENTS AND APPS

Knit Document (knitr)	Ctrl+Shift+K	Cmd+Shift+K
Insert chunk (Sweave & Knitr)	Ctrl+Alt+I	Cmd+Option+I
Run from start to current line	Ctrl+Alt+B	Cmd+Option+B

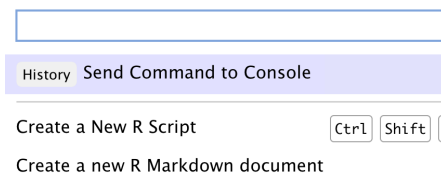
MORE KEYBOARD SHORTCUTS

Keyboard Shortcuts Help	Alt+Shift+K	Option+Shift+K
Show Command Palette	Ctrl+Shift+P	Cmd+Shift+P

View the Keyboard Shortcut Quick Reference with **Tools > Keyboard Shortcuts** or **Alt/Option + Shift + K**



Search for keyboard shortcuts with **Tools > Show Command Palette** or **Ctrl/Cmd + Shift + P**.



Visual Editor

Block format

Check spelling

Render output

Choose output format

Choose output location

Insert code chunk

Jump to previous chunk

Jump to next chunk

Run selected lines

Publish to server

Show file outline

Back to Source Editor (front page)

Heading 2

B

I

U

</>

Format

Insert

Table

File outline

R Markdown Including Plots

Insert and edit tables

Lists and block quotes

Links

Citations

Images

More formatting

Insert blocks, citations, equations, and special characters

Clear formatting

Insert verbatim code

Block format

title: "Document Title"

author: "Author Name"

output:

html_document:

toc: TRUE

Insert verbatim code

{r setup, include=FALSE}

knitr::opts_chunk\$set(echo = TRUE)

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

{r cars}

summary(cars)

Jump to chunk or header

Add/Edit attributes

Set knitr chunk options

Run this and all previous code chunks

Run this code chunk

RStudio Workbench



WHY RSTUDIO WORKBENCH?

Extend the open source server with a commercial license, support, and more:

- open and run multiple R sessions at once
- tune your resources to improve performance
- administrative tools for managing user sessions
- collaborate real-time with others in shared projects
- switch easily from one version of R to a different version
- integrate with your authentication, authorization, and audit practices
- work in the RStudio IDE, JupyterLab, Jupyter Notebooks, or VS Code

Download a free 45 day evaluation at

www.rstudio.com/products/workbench/evaluation/

Share Projects

File > New Project

RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

Start new R Session in current project

Close R Session in project

Active shared collaborators

Name of current project

Select R Version

Share Project with Collaborators

Run Remote Jobs

Run R on remote clusters (Kubernetes/Slurm) via the Job Launcher

Monitor launcher jobs

Launch a job

Run launcher jobs remotely

Job Name	Status	Location	Time	Actions
fast.R	Running	Local	0:09	Stop
sleepy.R	Succeeded 11:22 AM	Local	0:41	Refresh
sleepy.R	Idle	KubernetesX	Waiting	Stop

