

Wie bekommt man Hilfe?

Jan-Philipp Kolb

3 Mai 2017

Wie bekommt man Hilfe?

- Um generell Hilfe zu bekommen:

```
help.start()
```

- Online Dokumentation für die meisten Funktionen:

```
help(name)
```

- Nutze ? um Hilfe zu bekommen.

```
?mean
```

- `example(lm)` gibt ein Beispiel für die lineare Regression

```
example(lm)
```

Nutzung Suchmaschinen


- Ich nutze meistens google
- Tippe:

R-project + Was ich schon immer wissen wollte



- Das funktioniert natürlich mit jeder Suchmaschine!

Stackoverflow

- Für Fragen zum Programmieren
- Ist nicht auf R fokussiert
- Sehr detaillierte Diskussionen

 **stackoverflow**

QuestionsJobsDocumentationBETATagsUsers

 [Log In](#) [Sign Up](#)

Tagged Questions

info newest **8** featured frequent votes active unanswered

R is a free, open-source programming language and software environment for statistical computing, bioinformatics, and graphics. Please supplement your question with a minimal reproducible example. Use `dput()` for data and specify all non-base packages with library calls. For statistical questions ...

[learn more...](#) [top users](#) [synonyms \(2\)](#) [r jobs](#)

1776 votes

22 answers

147k views

[r](#) [r-faq](#)


How to make a great R reproducible example?

When discussing performance with colleagues, teaching, sending a bug report or searching for guidance on mailing lists and here on SO, a reproducible example is often asked and always helpful. What ...

[community wiki](#)
11 revs, 8 users 54%
[Hack-R](#)

22,187 frequent questions tagged

[r](#) [about »](#)

 **R Language**
DOCUMENTATION

[Find a request to handle](#) or [browse 121 topics](#).

Related Tags

[ggplot2](#) × 2875

[dataframe](#) × 1351

[plot](#) × 1105

Ein Schummelzettel - Cheatsheet

<https://www.rstudio.com/resources/cheatsheets/>

Base R Cheat Sheet

Getting Help

Accessing the help files

Focus
Get help of a particular function.
`help.search("weighted.mean")`
Search the help files for a word or phrase.
`help(package = "dplyr")`
Look help for a package.

View manual and code

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

Using Libraries

`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(list)`
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:/Users/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

<code>c(1, 4, 6)</code>	<code>1 4 6</code>	Arithmetic vector
<code>1:4</code>	<code>1 2 3 4</code>	Integer sequence
<code>seq(1, 4, length=3)</code>	<code>1.5 2.5 3.5</code>	A numeric sequence
<code>rep(1:3, times=2)</code>	<code>1 2 3 1 2 3</code>	Repeats each element twice
<code>rep(1:3, each=2)</code>	<code>1 1 2 2 3 3</code>	Repeats elements 2x each

Vector Functions

<code>sort(x)</code>	Return a sorted version of x
<code>unique(x)</code>	Return only unique values of x
<code>length(x)</code>	Return the number of elements in x

Selecting Vector Elements

By Position

<code>x[4]</code>	The fourth element
<code>x[-4]</code>	All but the fourth
<code>x[2:4]</code>	Elements two to four
<code>x[(1+4)]</code>	All elements except the 1st
<code>x[(1, 5)]</code>	Elements one and five

By Value

<code>x[x == 10]</code>	Elements which are equal to 10
<code>x[x < 0]</code>	All elements less than zero
<code>x[x %>% 2]</code>	Elements in the set 1, 2, 5

Named Vectors

<code>x["apple"]</code>	Element with name 'apple'
-------------------------	---------------------------

Programming

For Loop

For Variable in sequence()
Do something

Example

```
for (i in 1:10) {  
  i = i + 10  
  print(i)  
}
```

While Loop

while condition()
Do something

Example

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

IF Statements

if condition()
Do something
else
Do something different

Example

```
if (i > 10) {  
  print("Yes")  
} else {  
  print("No")  
}
```

Functions

function_name <- function(parameters)
Do something
`return(value)`

Example

```
squares <- function(x) {  
  squared <- x^2  
  return(squared)  
}
```

Reading and Writing Data

Input	Output	Description
<code>df <- read.table("file.csv")</code>	<code>write.table(df, "file.csv")</code>	Read and write a delimited text file.
<code>df <- read.csv("file.csv")</code>	<code>write.csv(df, "file.csv")</code>	Read and write a comma-separated value file. This is a special case of read.table() with default arguments.
<code>load("file.Rsave")</code>	<code>save(df, file = "file.Rsave")</code>	Read and write an R data file, a R script generated file.

Comparison

<code>0 < 10</code>	Not equal	<code>0 < 10</code>	Greater than	<code>0 < 10</code>	Less than	<code>0 < 10</code>	Is equal	<code>0 < 10</code>	Is not equal	<code>0 < 10</code>	Is not equal
------------------------	-----------	------------------------	--------------	------------------------	-----------	------------------------	----------	------------------------	--------------	------------------------	--------------

RStudio: an introduction to RStudio, 1.0.150 (2019-07-06) - RStudio: an introduction to RStudio, 1.0.150 (2019-07-06)

Read more or change the page layout - RStudio: an introduction to RStudio, 1.0.150 (2019-07-06)