

# 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

|  |  |
|--|--|
| <b>sort(x)</b><br>Return x sorted.       | <b>rev(x)</b><br>Return x reversed.    |
| <b>table(x)</b><br>See counts of values. | <b>unique(x)</b><br>See unique values. |

#### Selecting Vector Elements

##### By Position

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

##### By Value

|                             |                                 |
|-----------------------------|---------------------------------|
| <b>x[x == 10]</b>           | Elements which are equal to 10. |
| <b>x[x &lt; 0]</b>          | All elements less than zero.    |
| <b>x[x %in% c(1, 2, 5)]</b> | Elements in the set 1, 2, 5.    |

##### Named Vectors

|                   |                            |
|-------------------|----------------------------|
| <b>x['apple']</b> | 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',<br>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 |

### Matrix subsetting

`df[, 2]`



`df[2, ]`



`df[2, 2]`



### List subsetting

`df$x`



`df[[2]]`



Understanding a data frame

`View(df)`

See the full data frame.

`head(df)`

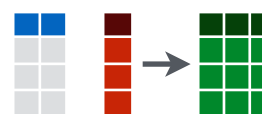
See the first 6 rows.

`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

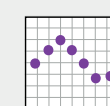
|  |   |  |
|--|---|--|
| <code>lm(y ~ x, data=df)</code><br>Linear model.                   | <code>t.test(x, y)</code><br>Perform a t-test for difference between means. | <code>prop.test</code><br>Test for a difference between proportions. |
| <code>glm(y ~ x, data=df)</code><br>Generalised linear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

## Distributions

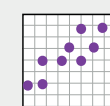
|          | Random Variates     | Density Function    | Cumulative Distribution | Quantile            |
|----------|---------------------|---------------------|-------------------------|---------------------|
| Normal   | <code>rnorm</code>  | <code>dnorm</code>  | <code>pnorm</code>      | <code>qnorm</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

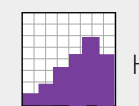
Also see the **ggplot2** package.



`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.