Basics of 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.

Variable Assignment

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

The Environment

ls()

List all variables in the environment.

rm(x)

Remove x from the environment.

rm(list = ls()) Remove all variables from the

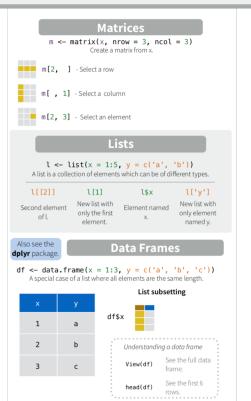
You can use the environment panel in RStudio to browse variables in your environment.

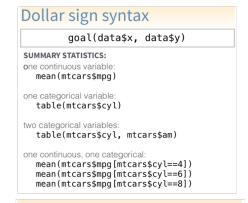
Types

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

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

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





Formula syntax

goal(y~x|z, data=data, group=w)

SUMMARY STATISTICS: one continuous variable:

mosaic::mean(~mpg, data=mtcars)

one categorical variable:

mosaic::tally(~cyl, data=mtcars)

two categorical variables:

mosaic::tally(cyl~am, data=mtcars)

one continuous, one categorical:

mosaic::mean(mpg~cyl, data=mtcars)

ilde

Tidyverse syntax



Syntax is the set of rules that govern what code works and doesn't work in a programming language. Most programming languages offer one standardized syntax, but R allows package developers to specify their own syntax. As a result, there is a large variety of (equally valid) R syntaxes.

The three most prevalent R syntaxes are:

- 1. The dollar sign syntax, sometimes called base R
 - **syntax**, expected by most base R functions. It is characterized by the use of dataset\$variablename, and is also associated with square bracket subsetting, as in dataset[1,2]. Almost all R functions will accept things passed to them in dollar sign syntax.
- The **formula syntax**, used by modeling functions like lm(), lattice graphics, and mosaic summary statistics. It uses the tilde (~) to connect a response variable and one (or many) predictors. Many base R functions will accept formula syntax.
- The **tidyverse syntax** used by dplyr, tidyr, and more. These functions expect data to be the first argument, which allows them to work with the "pipe" (%>%) from the magrittr package. Typically, ggplot2 is thought of as part of the tidyverse, although it has its own flavor of the syntax using plus signs (+) to string pieces together. ggplot2 author Hadley Wickham has said the package would have had different syntax if he had written it after learning about the pipe.