

R at a Glance:

Useful functions and syntax for Psych 252

Getting started

Install library: `install.packages('mylibrary')`

Load library: `library(mylibrary)`

Useful libraries to start with:

Load data	Clean up data	Statistics	Share results
xlsx	plyr	car	ggplot2
R.Matlab	dplyr	nlm	rmarkdown
jsonlite	tidyr	lme4	

Set working directory: `setwd('~/.path/to/my/data')`

Get current working directory: `getwd()`

Search for a function in the R documentation:

By its exact name: `?functionName`

By part of its name: `??fun`

Data frames

Reading and writing data

Combine elements into a vector	<code>c(1,2,3,4)</code>
Create number sequence	<code>seq(start, end)</code>
Combine vectors by rows	<code>rbind(vectr1, vectr2)</code>
Combine vectors by columns	<code>cbind(vectr1, vectr2)</code>
Create data frame from vectors	<code>data.frame(tag = value)</code>
Load an existing data set ^[1]	<code>read.csv('mydata.csv')</code> <code>read.xlsx('mydata.xls')</code>

Exploring datasets

View topmost rows	<code>head(data)</code>
View structure of data	<code>str(data)</code>
View summary of data ^[2]	<code>summary(data)</code>

Data types

(In the examples below, fill in datatype with the type of data you want, such as factor, character, numeric, or logical.)

Converting to other data types	<code>as.datatype(vectr)</code>
Check data type of a vector	<code>is.datatype(vectr)</code>
View/set levels of a factor	<code>levels(data\$columnName)</code>

Selecting and extracting data

View all column names	<code>names(data)</code>
Get column by name	<code>data\$columnName</code>
Get i-th row	<code>data[i,]</code>
Get j-th column	<code>data[, j]</code>
Get element at row i, column j	<code>data[i, j]</code>
Select rows using logical ^[3]	<code>data[data\$group == 'A',]</code>

Notes:

[1] Make sure you are in the correct working directory!

[2] summary returns different results depending on the object you want to summarize. Try it out with data frames, models, statistical tests, etc., to see what information it gives you.

[3] Don't forget the comma! This will return all rows that match the condition in the brackets. You can use this syntax to subset your data frame by any criterion, such as subjects that are above a certain age, all measurements taken in a drug trial before treatment, etc.

Basic math & statistics

	Name	Example
Arithmetic	Variable assignment	<code>x <- 10</code> <code>x = 10</code>
	Addition	<code>x + y</code>
	Subtraction	<code>x - y</code>
	Multiplication	<code>x * y</code>
	Exponent	<code>x ** y</code> <code>x^y</code>
Comparison	Modulus	<code>x %% y</code>
	Less than	<code>x < y</code>
	Less or equal to	<code>x <= y</code>
	Greater than	<code>x > y</code>
	Greater or equal to	<code>x >= y</code>
Logic	Equal to	<code>x == y</code>
	Not equal to	<code>x != y</code>
	NOT x	<code>!x</code>
	x OR y	<code>x y</code>
	X AND y	<code>x & y</code>
Other operations	Exponential	<code>exp(x)</code>
	Logarithm	<code>log(x)</code>
	Square root	<code>sqrt(x)</code>
	Round	<code>round(x)</code>
	Absolute value	<code>abs(x)</code>
Statistics	Sum	<code>sum(vectr)</code>
	Scale & center	
	Maximum	
	Minimum	
	Mean	<code>mean(vectr)</code>
	Median	<code>median(vectr)</code>
	Std. dev.	<code>sd(vectr)</code>
	Variance	<code>var(vectr)</code>
	Correlation	<code>cor(vectr1, vectr2)</code>
	Covariance	<code>cov(vectr1, vectr2)</code>
	T-Test	<code>t.test(y ~ x, data)</code> <code>t.test(vectr1, vectr2)</code>
	Chi-squared test	<code>chisq.test(table)</code>
	Linear model	<code>lm(y ~ x, data)</code>
	Logistic regression	<code>glm(y ~ x, family = "binomial")</code>
	Mixed-effects model	<code>lmer(y ~ x + (int slope), data)</code>

Missing ANOVA, length

Plotting with ggplot2

Making any plot with ggplot follows the same basic steps:

1. Choosing a **dataset** to plot
2. Using **geoms** to specify what kinds of marks (such as lines, dots, or bars) will appear on the plot
3. Using **aesthetic mappings** to specify how different properties of the dataset will appear on the plot. The most basic of these is choosing which variables will appear on the x and y axis.

Basic syntax

```
qplot(x = time, y = score, data = mydata,
      geom = 'point')
      Aesthetic mappings      Dataset
      Geoms
```

Making a plot with ggplot command provides more control. Use + to add elements, layers, and custom options.

```
myplot <- ggplot(mydata, aes(time, score)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  theme_bw() # custom options
```

Geoms

[Table with icon, command, available options, notes.
Include geom_density, geom_histogram, geom_bar, geom_jitter, geom_point, geom_line, geom_smooth, geom_errorbar]

Other options

Adding a title

```
myplot + ggtitle('My Plot')
```

Changing x- and y-axis labels

```
myplot + xlab('Time') + ylab('Score')
```

Faceting

Changing bounds of plot

Changing theme

Programming basics

Defining functions

if, for, while

Other useful

commands:

Paste, attach, detach,
with, by

Sampling from distributions

qnorm, rnorm, etc.

Debugging?

Useful resources