

Introduction to R

by Debby Kermer

Data Services Research Consultant
George Mason University

<http://dataservices.gmu.edu>
datahelp@gmu.edu

<http://dataservices.gmu.edu/software/r>

www.r-project.org

R \approx S \approx S-Plus

Free

Statistical
Programming
Environment

rseek.org



The R Project for Statistical Computing

[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Contributors](#)

[What's New?](#)

[Mailing Lists](#)

[Bug Tracking](#)

[Conferences](#)

[Search](#)

R Foundation

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

Documentation

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.2.2 (Fire Safety)** has been released on 2015-08-14.
- **The R Journal Volume 7/1** is available.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.
- **useR! 2015**, will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.

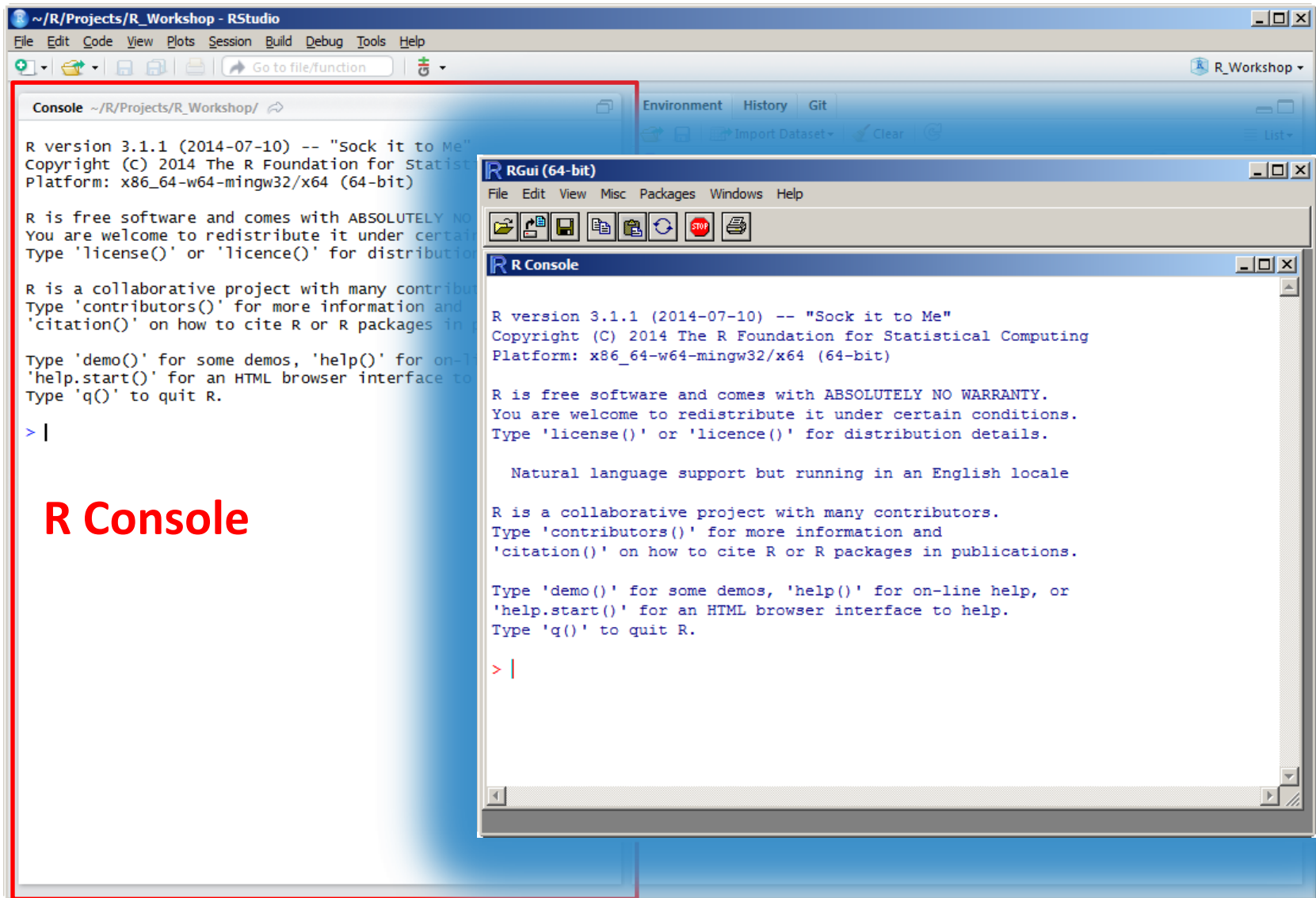
• **useR! 2014**, took place at the University of California, Los

Why R can be Hard

- R is for Statisticians / Programmers
 - Easy to create new analyses
 - Multiple ways to accomplish task
 - But, tutorials have complete examples!
- R is Open Source
 - Constantly changing
 - Varying quality of code and documentation
 - But, lots of both!

The Interface

RStudio: www.rstudio.com



Console

- > prompt
- + waiting for more
- [#] element number

3 + 2 {ENTER}

3 - {ENTER} 2

Logic Symbols

| or

& and

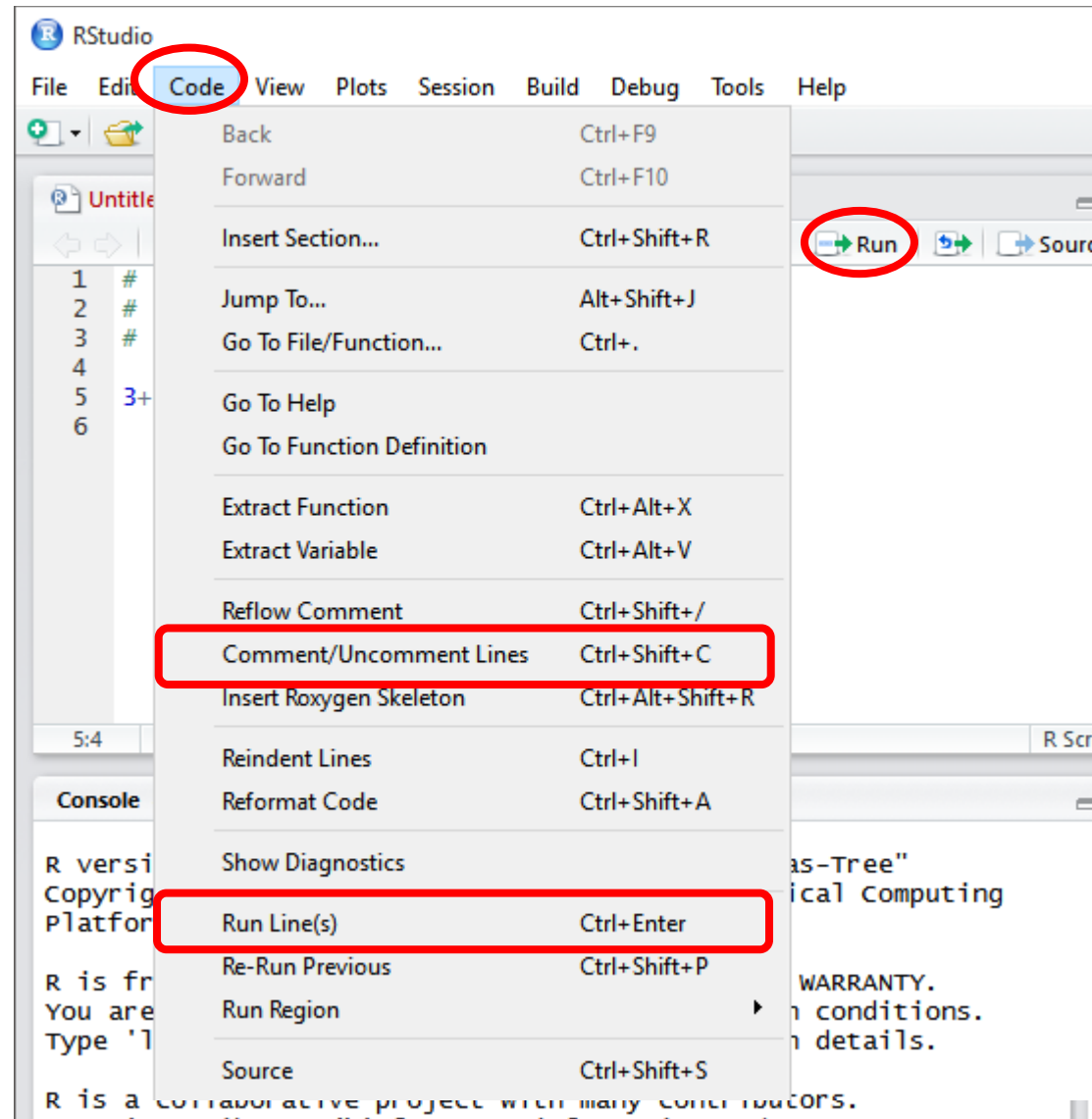
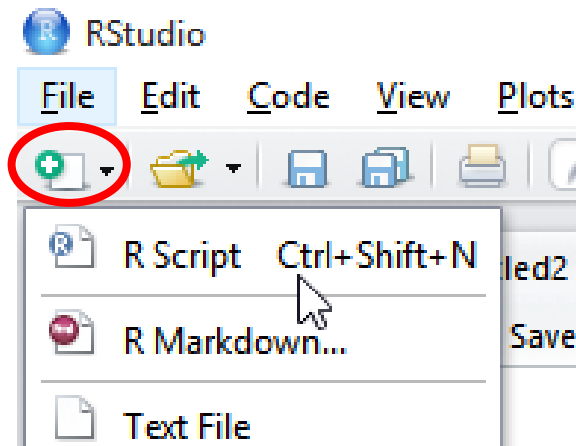
== is equal to

!= not equal to

Type 'de
'help.st
Type 'q

```
> 3+2
[1] 5
> 3-
+ 2
[1] 1
> |
```

Script Files



Objects

Assignment <-

RStudio: Type <- with Alt + - [dash]

```
my.school <- "Mason"
```

```
nine <- 9
```

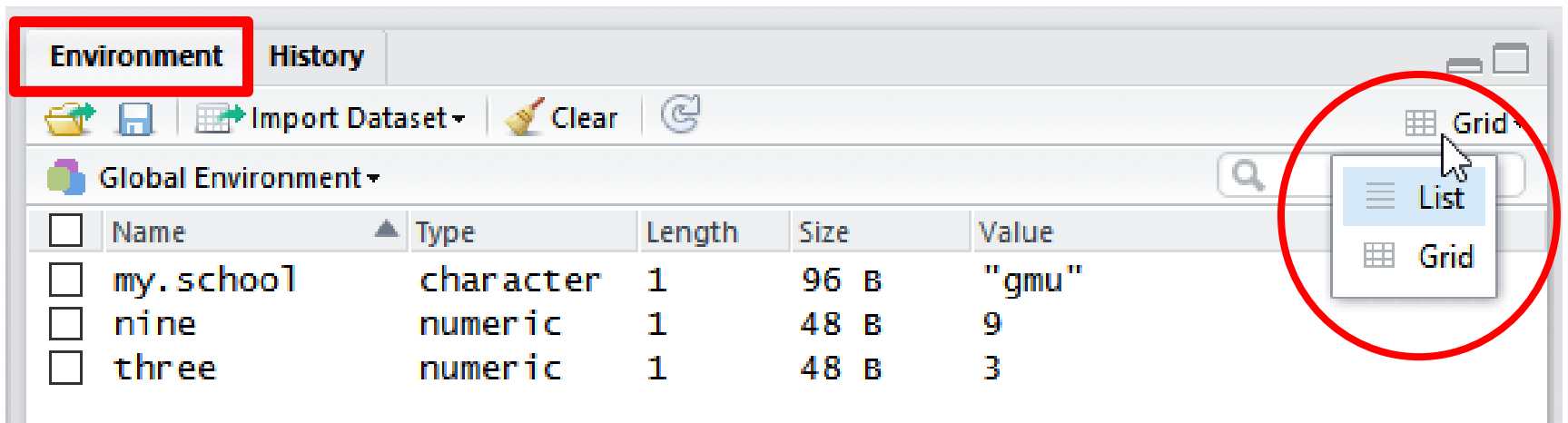
```
three <- nine / 3
```

```
> nine <- 9
```

```
> nine
```

```
[1] 9
```

It is best to use <-
but = also works



Vectors [Variables]

```
numbers <- c(101,102,103,104,105)
```

```
numbers <- 101:105
```

```
numbers <- c(101:104,105)
```

*think **c** combine*

the same

```
numbers[ 2 ]
```

```
numbers[ c(2,4) ]
```

```
numbers[ numbers > 102 ]
```

```
> numbers <- 101:105
```

```
> numbers
```

```
[1] 101 102 103 104 105
```

```
> numbers[ 2 ]
```

```
[1] 102
```

```
> numbers[ c(2,4) ]
```

```
[1] 102 104
```

```
> numbers[ numbers > 102 ]
```

```
[1] 103 104 105
```

```
>
```

Functions & Packages

Functions & Arguments

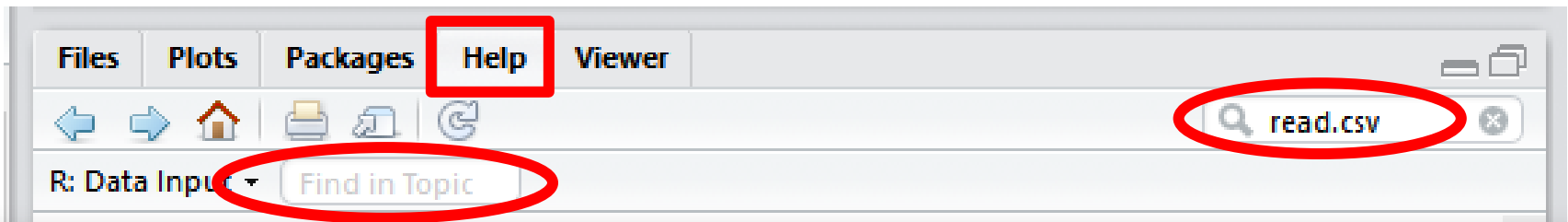
`read.table(datafile, header=TRUE, sep = ",",)`

Function

Positional
Argument

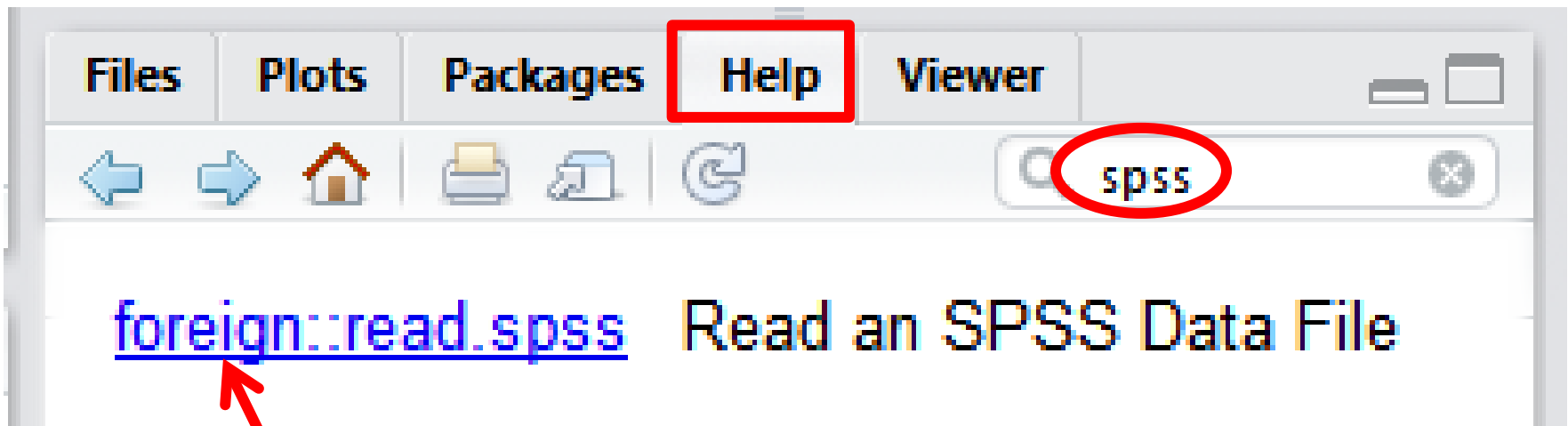
Named
Argument

Named
Argument

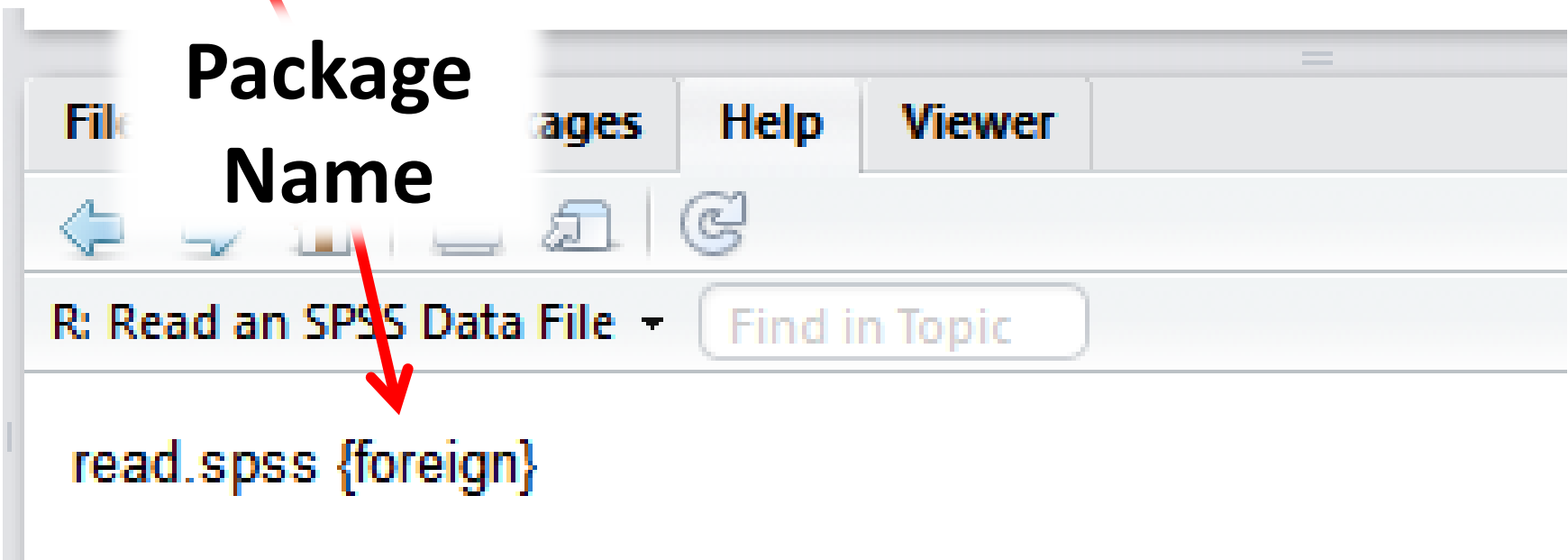


```
read.table(file, header = FALSE, sep = "", quote = "\"",  
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
           row.names, col.names, as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrows = -1,  
           skip = 0, check.names = TRUE, fill = "blank.lines.skip",  
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)  
  
read.csv(file, header = TRUE, sep = ",", quote = "\"",  
         dec = ".", fill = TRUE, comment.char = "", ...)  
  
fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

Packages



**Package
Name**



Using Packages

1) Install

Click "Install"

or

`install.packages("name")`

2) Load

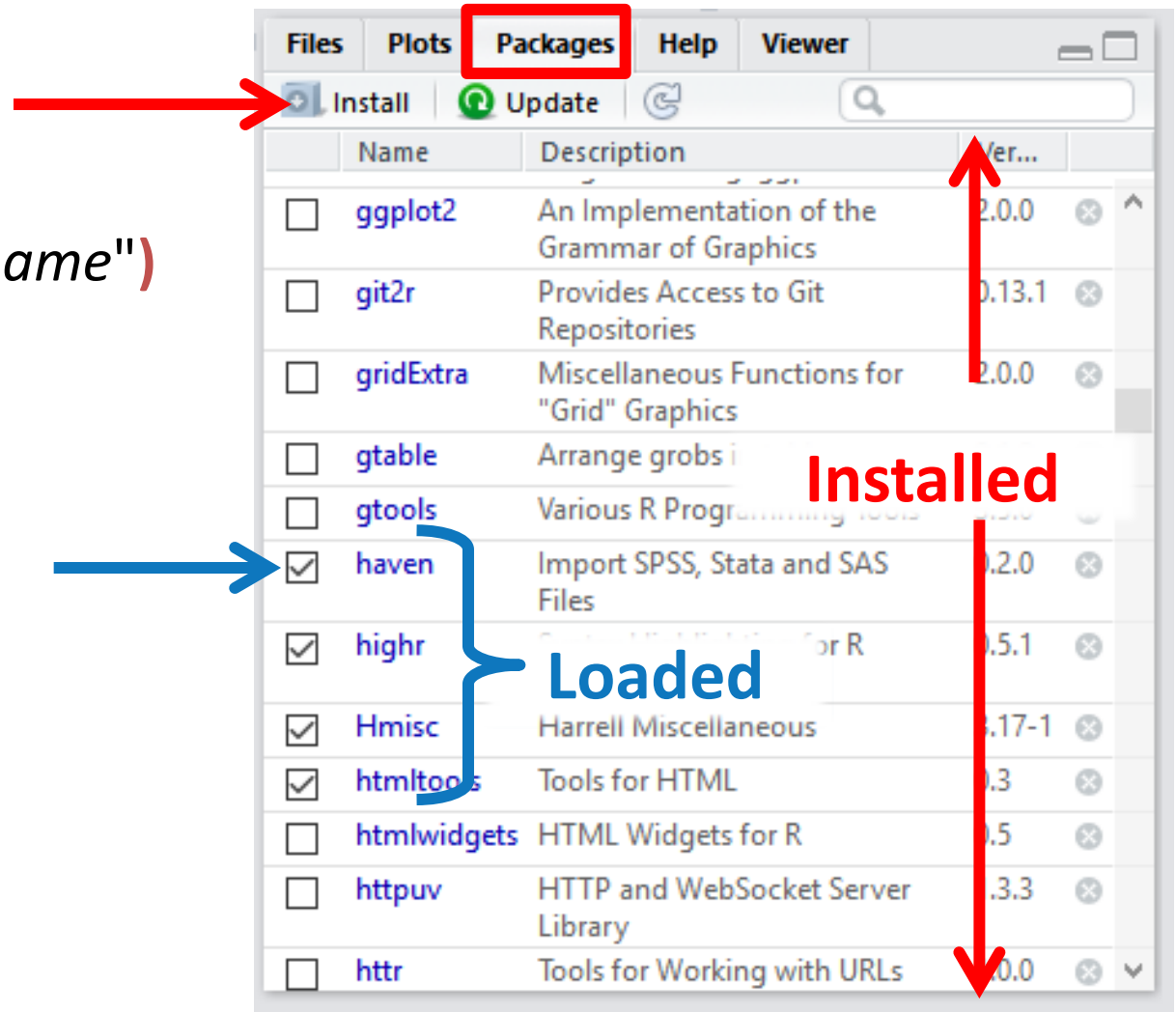
Click the checkbox

or

`library(name)`

alternatively

`require(name)`



Choosing Packages

Thousands of packages

Not all are good

Download from **C**omprehensive **R** **A**rchive **N**etwork (CRAN)

<http://dataservices.gmu.edu/software/r/packages>

Working with Datasets

```
mydata <- read.csv("titanic.csv")
```


Data Frame

`str(mydata)` think *structure*

```
> str(mydata)
```

```
'data.frame':    1309 obs. of  10 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ name    : Factor w/ 1307 levels "Abbing, Mr. Anthony",...: 1 2 3 4
 $ survived: Factor w/ 2 levels "Died","Survived": 1 1 1 2 2 1 2 1
 $ age     : num  42 13 16 35 16 25 30 28 20 18 ...
 $ gender  : Factor w/ 2 levels "female","male": 2 2 2 1 1 2 2 1 2 1
 $ sibsp   : int   0 0 1 1 0 0 1 1 0 0 ...
 $ parch   : int   0 2 1 1 0 0 0 0 0 0 ...
 $ pclass  : Factor w/ 3 levels "1st","2nd","3rd": 3 3 3 3 3 3 2 2 3
 $ fare    : num   7.55 20.25 20.25 20.25 7.65 ...
 $ embarked: Factor w/ 4 levels "", "C", "Q", "S": 4 4 4 4 4 4 2 2 4 2
```

int / num = Numeric
(Interval / Ratio)

Factor = Categorical
(Nominal / Ordinal)

`mydata$pclass`

`mean(mydata$fare)`

Variables in Data Frames

Object

`mydata$age`

`mydata[, 'age']`

`mydata[['age']]`

Named Argument

`qplot(age, data = mydata)`

`library(lessr)`

`Histogram(age, data = mydata)`

Attach

`attach(mydata)`

`age`

Positional Argument

`select(mydata, age)`

`mydata %>% select(age) %>% head`



Pipe - read "*then*"

Ctrl + Shift + M

Review: What to look for

<- =	Assignment	Create Objects	> list <- c("one","two") > list = c("one","two")
[] \$	Object Parts	Indexing Column Name	> dataset[row, column] > dataset\$columnname
() ,	Functions	Packages Arguments	> library(something) > with(data, column=two)
+ +	Continuation	Unfinished	> ggplot(one, data=data) + + geom_points()

Statistical Analysis

Model for Statistical Analysis

```
object <- goal( formula , data = data )
```

~ predicted from : interaction
+ include * factorial

$Y \sim X$

$Y \sim X + Z$

$Y \sim X * Z$

```
tt.ttest <- t.test( fare ~ gender, data = mydata )
```

```
tt.anova <- aov( fare ~ pclass + gender, data = mydata )
```

```
tt.logistic <- glm( survived ~ pclass * gender, data = mydata ,  
family = binomial )
```

Analysis Objects

```
> tt.ttest <- t.test(fare ~ gender, data = mydata )  
> tt.ttest
```

Welch Two Sample t-test

data: fare by gender

t = 6.116, df = 701.72, p-value = 0.000000001594

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

13.60912 26.47787

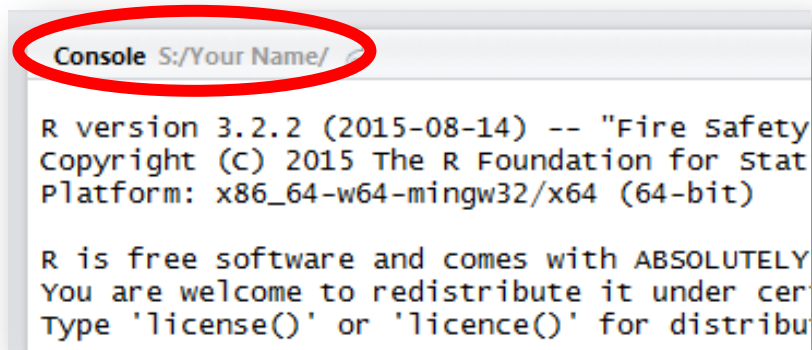
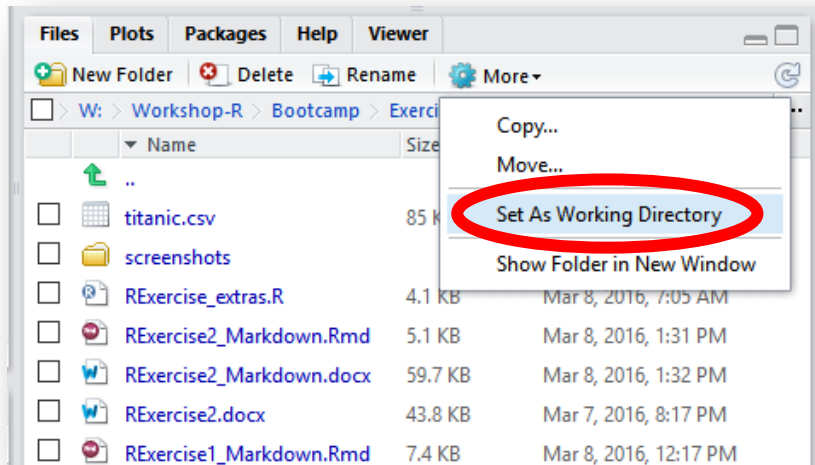
sample estimates:

mean in group female	mean in group male
46.1981	26.1546

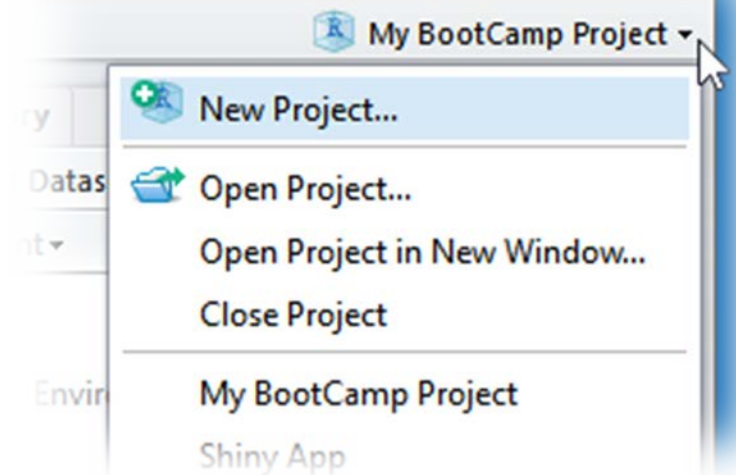
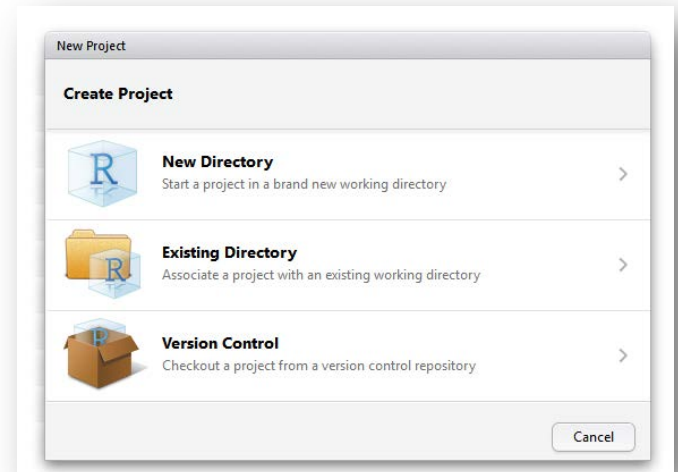
Project Management

Have a Project Directory

Working Directories

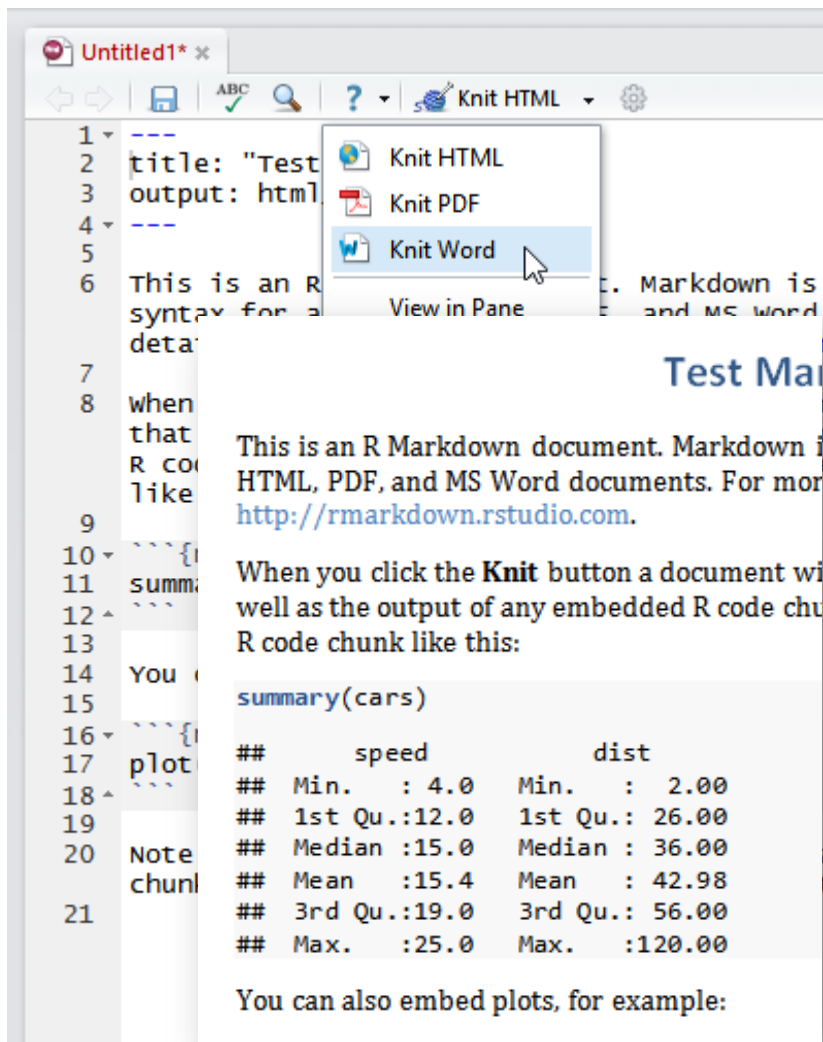


RStudio Projects



Combine Tools

RMarkdown (Script)



The screenshot shows the RStudio interface with an RMarkdown document titled 'Untitled1*.rmd'. The 'Knit' button in the toolbar is clicked, opening a menu with options: 'Knit HTML', 'Knit PDF', 'Knit Word', and 'View in Pane'. The 'Knit Word' option is selected. Below the menu, the document content is visible, including a title 'Test Markdown', a paragraph about R Markdown, and an R code chunk that has been executed, displaying the output of the `summary(cars)` function.

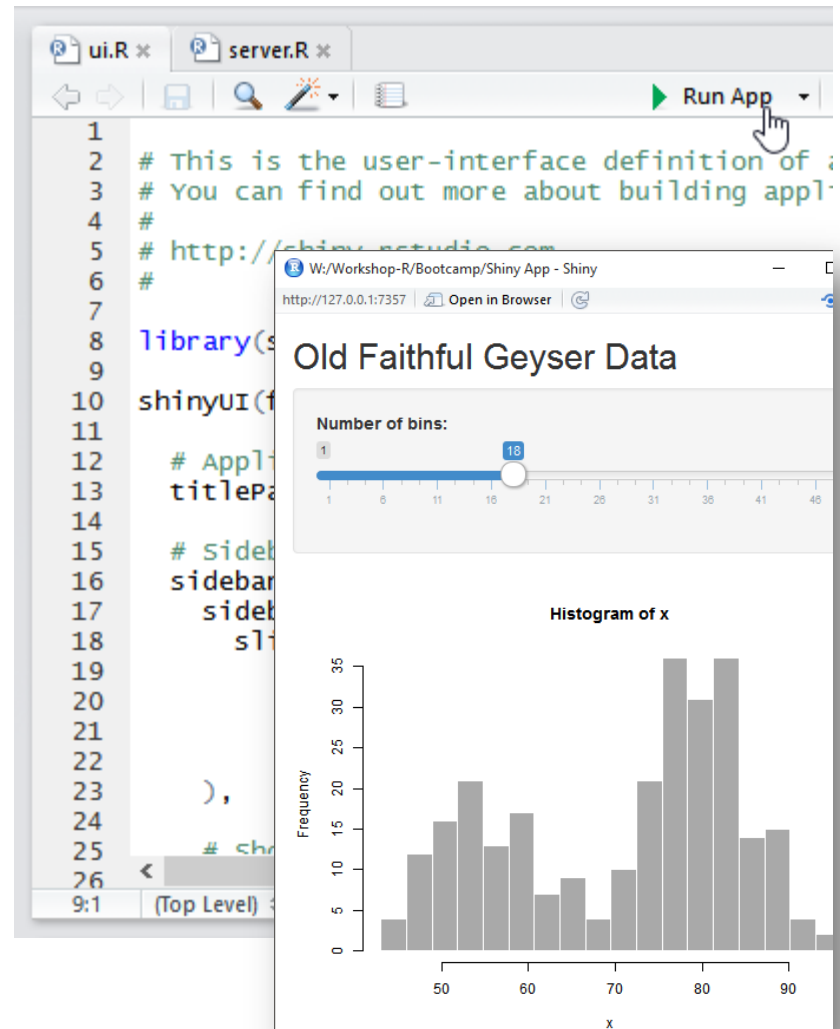
```
1 title: "Test Markdown"
2 output: html_document
3
4 ---
5
6 This is an R Markdown document. Markdown is a simple
7 syntax for creating formatted text documents. You can
8 learn more about it in http://rmarkdown.rstudio.com.
9
10 When you click the Knit button a document will be generated
11 containing the contents of the R code chunks along with the
12 output of any embedded R code chunks or R graphics.
13
14 You can embed R code in a document and have it executed
15 when you click the Knit button. Here is an example:
16
17 

```
summary(cars)
```


18
19 ##      speed      dist
20 ##  Min.   : 4.0    Min.   :  2.00
21 ## 1st Qu.:12.0    1st Qu.: 26.00
22 ## Median :15.0    Median : 36.00
23 ## Mean   :15.4    Mean   : 42.98
24 ## 3rd Qu.:19.0    3rd Qu.: 56.00
25 ## Max.   :25.0    Max.   :120.00
```

You can also embed plots, for example:

Shiny Interactive Graphics



The screenshot shows the RStudio interface with two files: 'ui.R' and 'server.R'. The 'Run App' button in the toolbar is clicked. The 'server.R' file contains the following code:

```
1 # This is the user-interface definition of a Shiny web application.
2 # You can find out more about building apps here:
3 # http://shiny.rstudio.com/articles/building.html
4
5 # Load the necessary libraries
6 library(shiny)
7
8 # Define the UI
9 shinyUI(fluidPage(
10   titlePanel("Old Faithful Geyser Data"),
11   sliderInput("bins", "Number of bins:",
12     min = 1, max = 48, value = 18),
13   plotOutput("histogram")
14 ))
15
16 # Define the server logic
17 server({
18   data <- read.csv("data/oldfaithful.csv")
19   x <- data[, "eruptions"]
20   hist(x, breaks = get("bins"), col = "gray", border = "black")
21 })
```

The 'Run App' button is clicked, and a web browser window opens showing the Shiny application. The application displays a title 'Old Faithful Geyser Data' and a histogram of the 'eruptions' data. A slider input allows the user to interactively change the number of bins, currently set to 18. The histogram shows the frequency distribution of eruption times.

Now What?

1. Download
2. Unzip
3. Open **R-Exercise** file