

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.

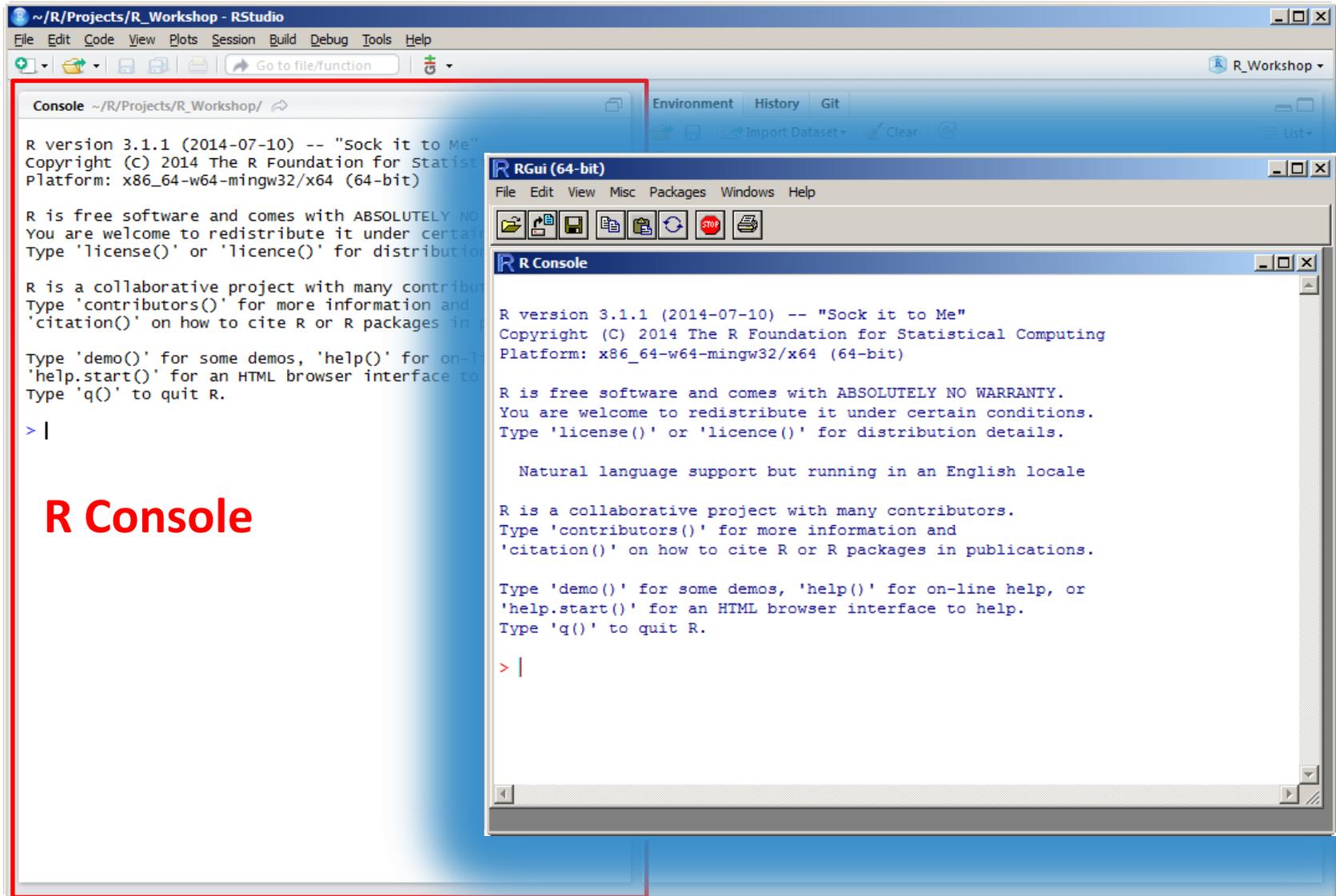
- **useR! 2014**, took place at the University of California, Los

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

3 > 2

3 >= 2

2 == 2

| or

& and

== is equal to

!= not equal to

```
Type 'de  
'help.st  
Type 'q
```

```
> 3+2
```

```
[1] 5
```

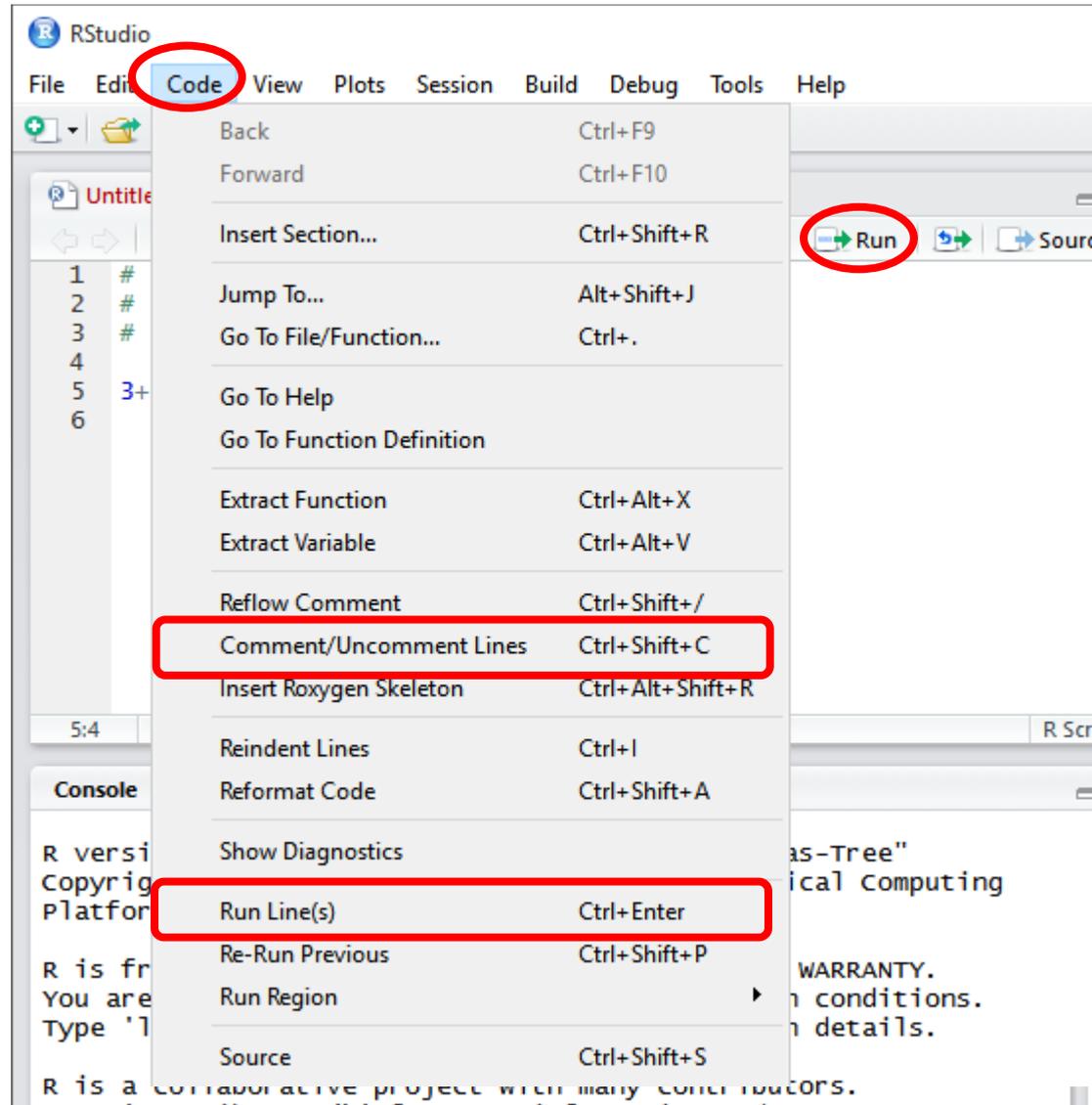
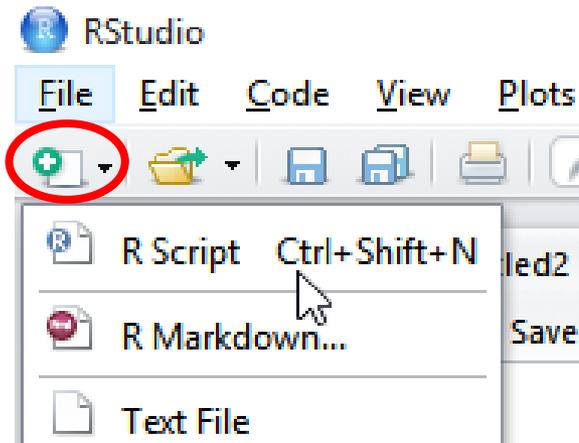
```
> 3-
```

```
+ 2
```

```
[1] 1
```

```
> |
```

Script Files



Objects

Assignment <-

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

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

```
nine <- 9
```

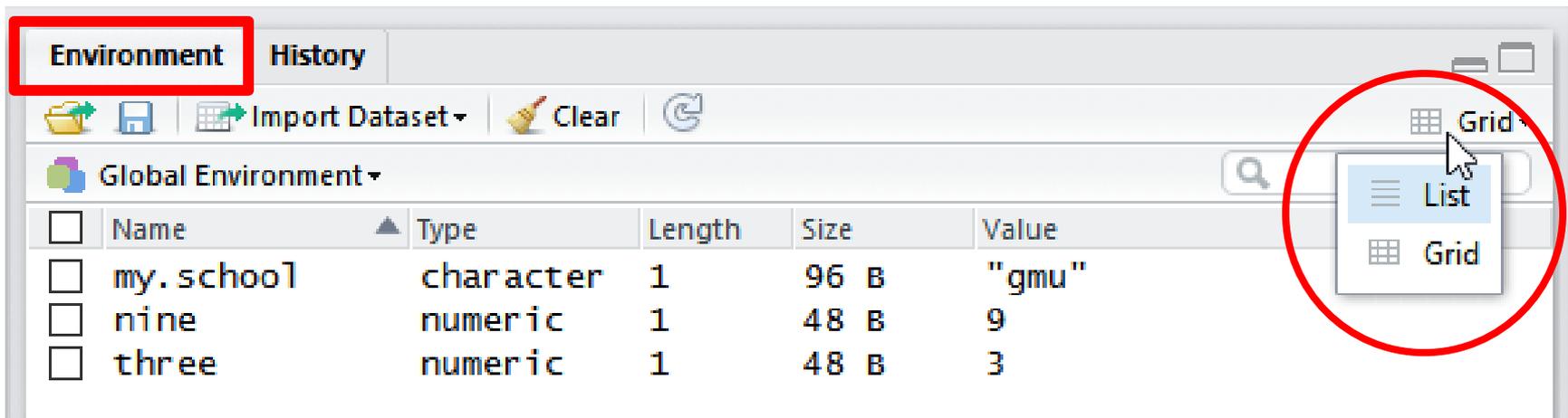
```
three <- nine / 3
```

```
> nine <- 9
```

```
> nine
```

```
[1] 9
```

It is best to use <-
but = also works



The screenshot shows the RStudio Environment pane. The 'Environment' tab is selected and highlighted with a red box. Below the toolbar, the 'Global Environment' is displayed with a search icon. A table lists the current objects in the environment:

<input type="checkbox"/>	Name	Type	Length	Size	Value
<input type="checkbox"/>	my.school	character	1	96 B	"gmU"
<input type="checkbox"/>	nine	numeric	1	48 B	9
<input type="checkbox"/>	three	numeric	1	48 B	3

On the right side of the Environment pane, a red circle highlights a context menu with 'List' and 'Grid' options. A mouse cursor is pointing at the 'List' option.

Vectors [Variables]

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

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

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

think 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

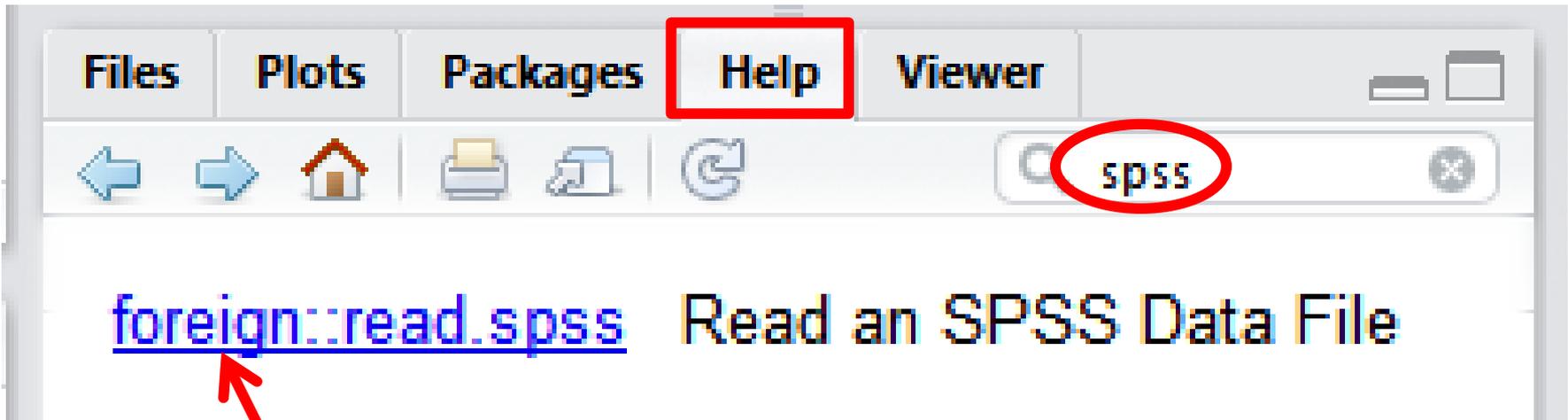
Named
Argument

Named
Argument

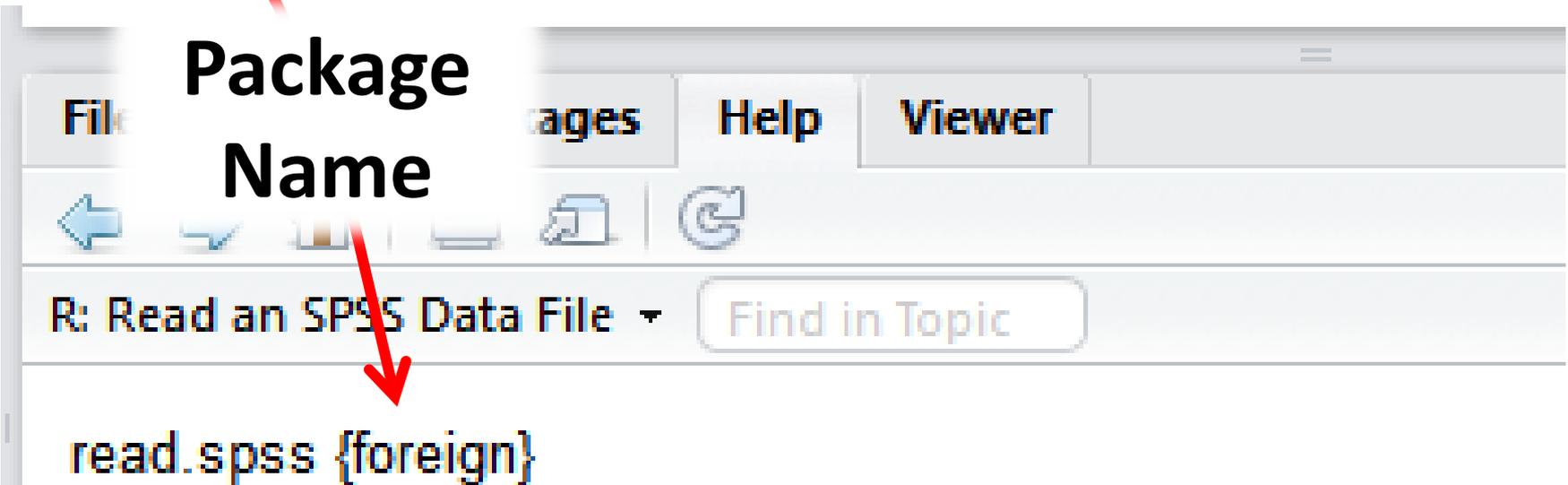


```
read.table(file, header = FALSE, sep = "", quote = "\"",  
          dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
          row.names, col.names, as.is = !stringsAsFactors,  
          na.strings = "NA", colClasses = NA, nrows = -1,  
          skip = 0, check.names = TRUE, fill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

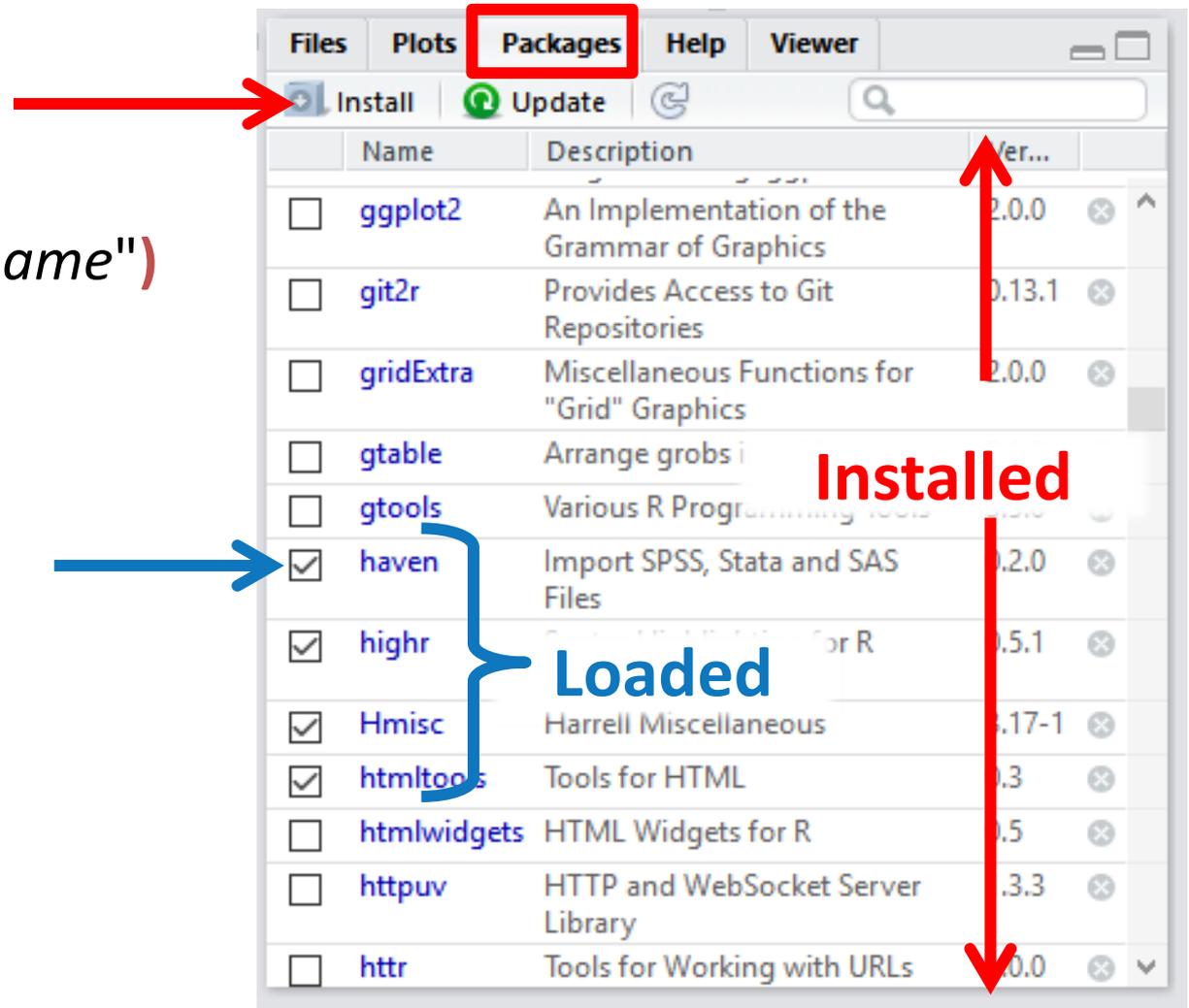
Click the checkbox

or

`library(name)`

alternatively

`require(name)`



Choosing Packages

Thousands of packages

Not all are good

Download from **Comprehensive R Archive Network (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 2 1 2 3
 $ 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

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

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
```

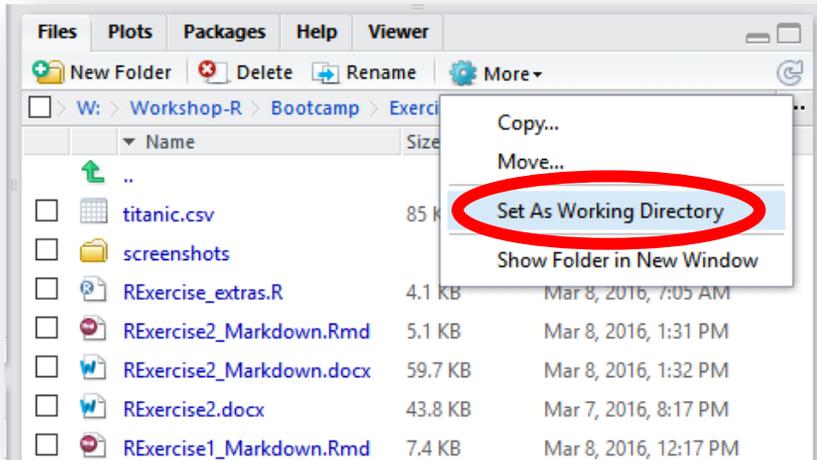
```
sample estimates:
```

```
mean in group female    mean in group male  
      46.1981             26.1546
```

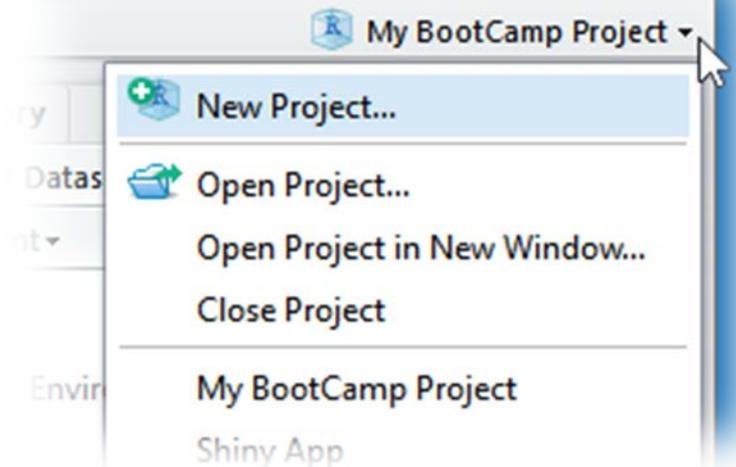
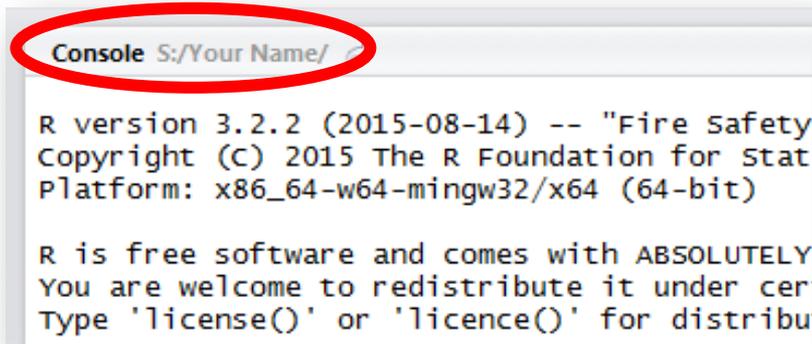
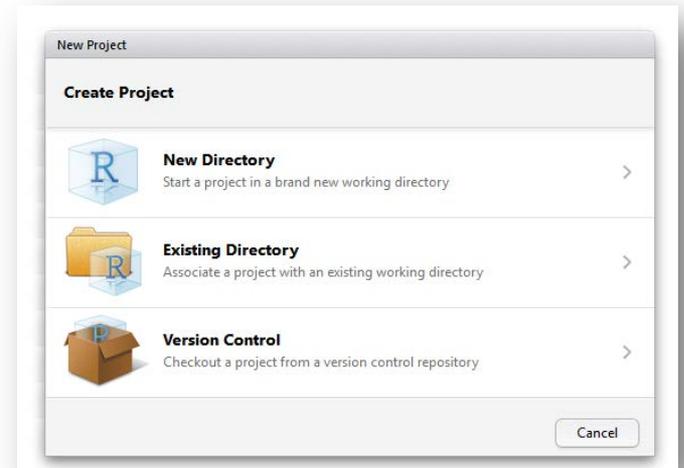
Project Management

Have a Project Directory

Working Directories

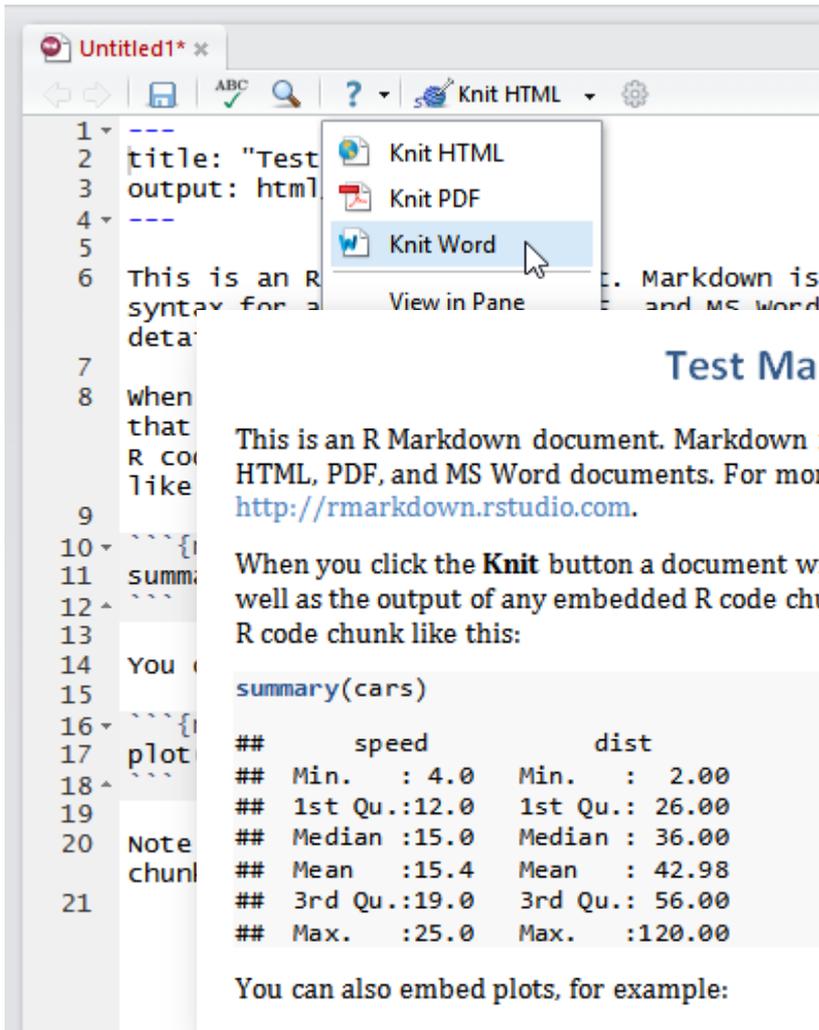


RStudio Projects



Combine Tools

RMarkdown (Script)



The screenshot shows the RStudio interface with an RMarkdown document. The top toolbar includes the 'Knit HTML' button. A dropdown menu is open, showing options to 'Knit HTML', 'Knit PDF', and 'Knit Word'. The main editor shows RMarkdown code with a code chunk for `summary(cars)`. The right-hand pane displays the rendered HTML output, which includes a title 'Test Markdown', introductory text about RMarkdown, and a table of summary statistics for the 'cars' dataset.

```
1 ---  
2 title: "Test  
3 output: html  
4 ---  
5  
6 This is an R Markdown document. Markdown is a simple  
7 syntax for writing structured text in a way that can be  
8 easily converted to HTML, PDF, and MS Word documents.  
9  
10 You can embed an R code chunk in any document.  
11  
12 When you click the Knit button a document will be  
13 rendered (HTML output by default) and the contents of  
14 any embedded R code chunks will be executed, the results  
15 will be stored in a file called outputs/ and will be  
16 automatically loaded into the document and placed below  
17 each code chunk like this:  
18  
19 

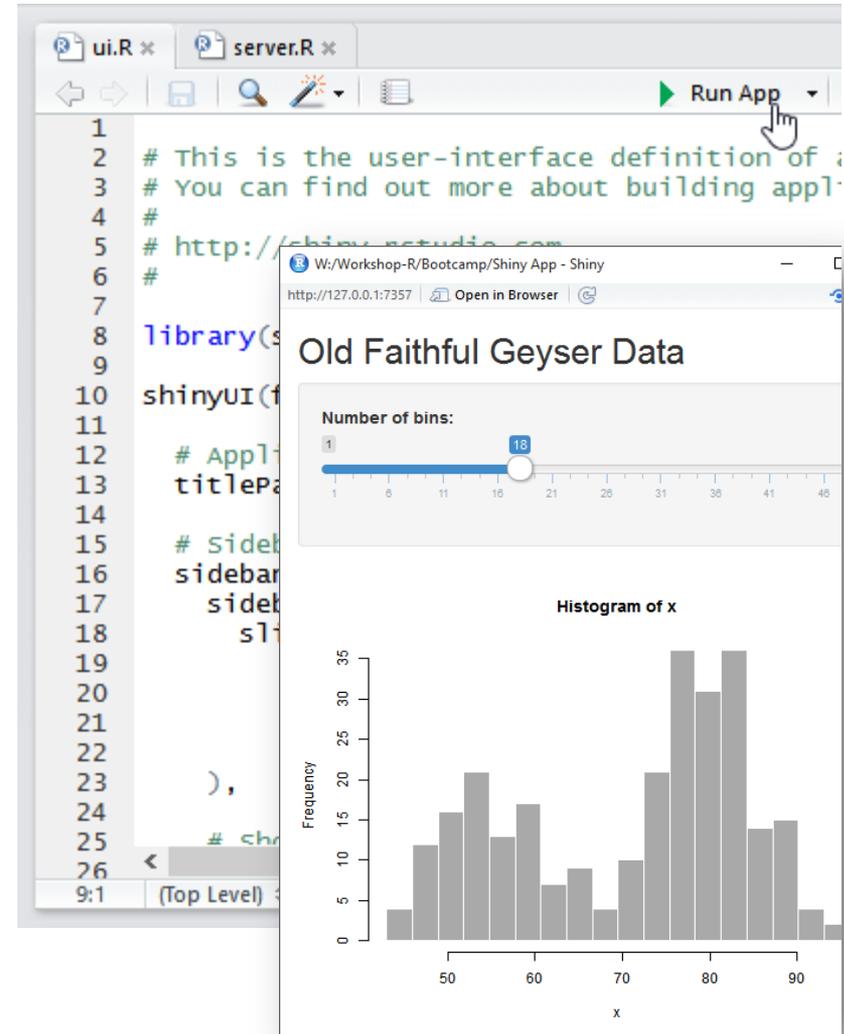
```
summary(cars)
```

  
20  
21
```

##	speed	dist
## Min.	: 4.0	Min. : 2.00
## 1st Qu.	:12.0	1st Qu.: 26.00
## Median	:15.0	Median : 36.00
## Mean	:15.4	Mean : 42.98
## 3rd Qu.	:19.0	3rd Qu.: 56.00
## Max.	:25.0	Max. :120.00

You can also embed plots, for example:

Shiny Interactive Graphics



The screenshot shows the RStudio interface with a Shiny application. The top toolbar includes the 'Run App' button. The main editor shows R code for a Shiny app using `library(shiny)` and `shinyUI(fluidPage(...))`. The right-hand pane displays the rendered Shiny application in a browser window. The application title is 'Old Faithful Geyser Data'. It features an interactive slider for 'Number of bins' with a value of 18. Below the slider is a histogram titled 'Histogram of x' showing the frequency distribution of the 'x' variable.

```
1  
2 # This is the user-interface definition of a Shiny app  
3 # You can find out more about building apps with  
4 # Shiny here: http://shiny.rstudio.com/  
5 # http://shiny.rstudio.com/  
6 #  
7  
8 library(shiny)  
9  
10 shinyUI(fluidPage(  
11   titlePanel("Old Faithful Geyser Data"),  
12   # Application title panel  
13   titlePanel("Old Faithful Geyser Data"),  
14   # Sidebars  
15   # sidebarLayout(  
16     sidebarMenu(  
17       sidebarLink("Home", href="#"),  
18       sidebarLink("About", href="#"),  
19       sidebarLink("Help", href="#"),  
20     ),  
21     # Main content area  
22     mainPanel(  
23       # Shiny app content  
24     ),  
25   ),  
26   # Shiny app content  
27 )
```

Number of bins: 18

Histogram of x

Frequency	x
4	45
12	50
16	55
21	60
13	65
17	70
7	75
9	80
21	85
35	90
35	95
14	100
15	105
4	110
3	115

Now What?

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