



Paul Intrevado

Exploratory Data Analysis with R

Summer 2018





Table of Contents

1 A Brief Introduction

2 Introduction to R

3 RMarkdown



Section 1

A Brief Introduction



The Warden





Who Am I?

- Ph.D. Operations Management (McGill University)
 - Research focused on service operations
 - Model and solve optimization problems (e.g., MIPs)
- B.Sc., M.Sc. Industrial Engineering (Purdue University)
- B. Commerce (McGill University)
- Assistant Prof @ USF as of August 2014
- I teach or have taught
 - MSDS 593 - Exploratory Data Analysis with R
 - MSDS 601 - Linear Regression Analysis
 - MSDS 605/625/627/632 - Practicum
 - CS 686 - Machine Learning
 - BSDS 100 - Introduction to Data Science with R
- MS Data Science Practicum Director [2014-2017]
- Founding Associate Director of Data Institute [2016-2017]



What I Do

$$\min_{y, z^+, z^-} \sum_{j \in \mathbb{J}} \sum_{\nu \in \mathbb{V}} \sum_{t \in \mathbb{T}} r^{(t-1)} \left[\sum_{w \in \Omega} \left[f_{j\nu w}^+ z_{j\nu wt}^+ - f_{j\nu w}^- z_{j\nu wt}^- \right] + \gamma_\nu (\kappa_{j\nu t} - \rho_{j\nu t} + \sum_{i \in \mathbb{I}} y_{ij\nu t}) \right] \quad (4.1)$$

subject to

$$\kappa_{j\nu t} = \kappa_{j\nu(t-1)} + \sum_{w \in \Omega} C_{\nu w} (z_{j\nu wt}^+ - z_{j\nu wt}^-) \quad \forall j \in \mathbb{J}, \nu \in \mathbb{V}, t \in \mathbb{T} \quad (4.2)$$

$$\begin{aligned} \rho_{j\nu t} = & \rho_{j\nu(t-1)} + \sum_{w \in \Omega} C_{\nu w} (z_{j\nu wt}^+ - z_{j\nu wt}^-) - \sum_{i \in \mathbb{I}} y_{ij\nu(t-1)} + \alpha_{j\nu(t-1)} - \psi_{j(t-1)}^{(\nu+1 \rightarrow \nu)} + \psi_{j(t-1)}^{(\nu \rightarrow \nu-1)} \\ & \forall j \in \mathbb{J}, \nu \in \mathbb{V}, t \in \mathbb{T} \end{aligned} \quad (4.3)$$

$$\alpha_{j\nu t} = \mu_{\nu t} (\kappa_{j\nu t} - \rho_{j\nu t} + \sum_{i \in \mathbb{I}} y_{ij\nu t}) \quad \forall j \in \mathbb{J}, \nu \in \mathbb{V}, t \in \mathbb{T} \setminus |\mathbb{T}| \quad (4.4)$$

$$\psi_{jt}^{(\nu \rightarrow \nu-1)} = \tau_t^{(\nu \rightarrow \nu-1)} (\kappa_{j\nu t} - \rho_{j\nu t} + \sum_{i \in \mathbb{I}} y_{ij\nu t} - \alpha_{j\nu t}) \quad \forall j \in \mathbb{J}, \nu \in \mathbb{V} \setminus \{1\}, t \in \mathbb{T} \setminus |\mathbb{T}| \quad (4.5)$$

$$\eta_{it} \lambda_{it} + \psi_{i(t-1)}^{(\nu+1 \rightarrow \nu)} = \sum_{j \in \mathbb{J}} y_{ij\nu t} \quad \forall i \in \mathbb{I}, \nu \in \mathbb{V}, t \in \mathbb{T} \quad (4.6)$$

$$\eta_{it} \lambda_{it} \pi_{it} + \psi_{i(t-1)}^{(\nu+1 \rightarrow \nu)} \leq y_{ij\nu t} \quad \forall i = j \in \mathbb{J}, \nu \in \mathbb{V}, t \in \mathbb{T} \quad (4.7)$$



Awards from Alumni

*Most Likely to have Been a
General in a Former Life*

Awarded to

Paul Intrevado



UNIVERSITY OF
SAN FRANCISCO



Awards from Alumni

*Most Likely to Call You an Idiot in the
Form of An Impeccably Worded Email*

Awarded to

Paul Intrevado



UNIVERSITY OF
SAN FRANCISCO



Awards from Alumni

MSAN Superlative Award 2016

Most likely to date his teacher

Paul Intrevado

Uni. Of San Francisco



MSAN Class of 2016



Software Programming Experience

- C / C++
- MATLAB
- CPLEX
- Gurobi
- R
- Python
- SQL
- CSS / HTML (web)
- MS Excel / MS Access / VBA



Subsection 2

About this Class



Course Material

- All digital homework submissions and grades will be submitted/posted through Canvas
- All other course material can be found at the following link

<https://goo.gl/xzbwzw>



Syllabus

MSDS 593 – Exploratory Data Analysis with R

Instructor: Paul Intrevado
Course Syllabus
Summer 2018

SUMMARY INFORMATION

Offices: Shared Office (Downtown) / McLaren Hall, Room 103 (Main Campus)

Office Hours: Wednesdays, 10.00 - 11.30 and 13.45 - 15.15 in Shared Office

Office Phone: 415/422.2527

Email: pintrevado@usfca.edu

Class Time: 10:00 - 12:00 / 13:00 - 15:00 Thursdays and Fridays

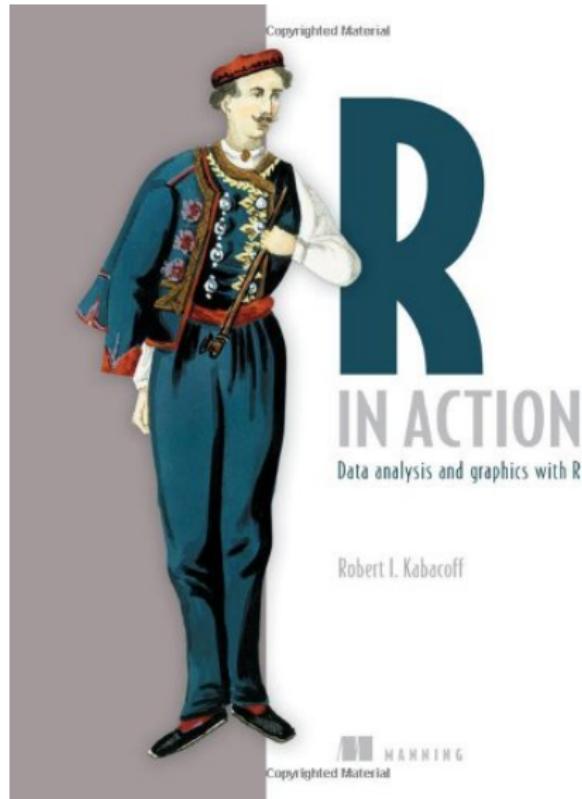
Final Exam: 13.00 - 15.00 Friday, August 10th, 2018

Quizzes: 09:15 - 10:00 Thursdays



Subsection 3

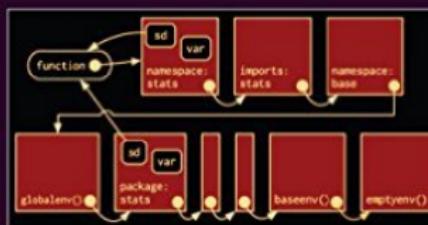
Reference Textbooks





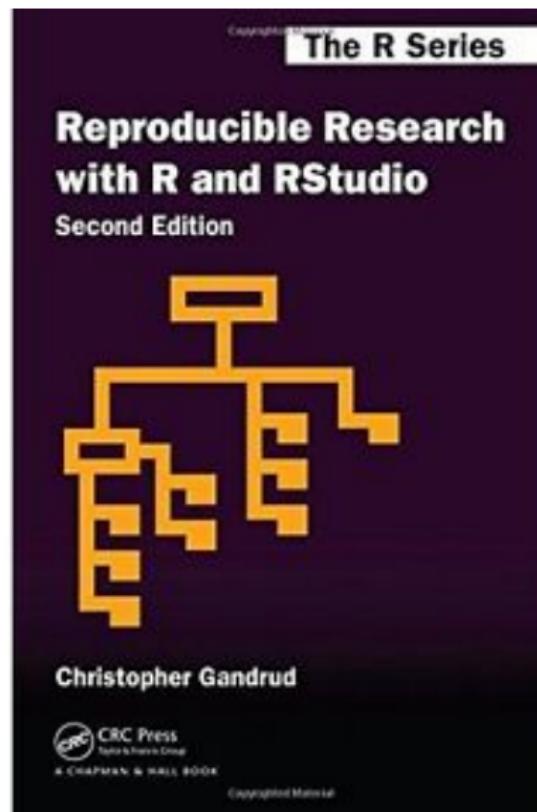
The R Series

Advanced R



Hadley Wickham

 CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK





The R Series

Dynamic Documents with R and knitr

Second Edition

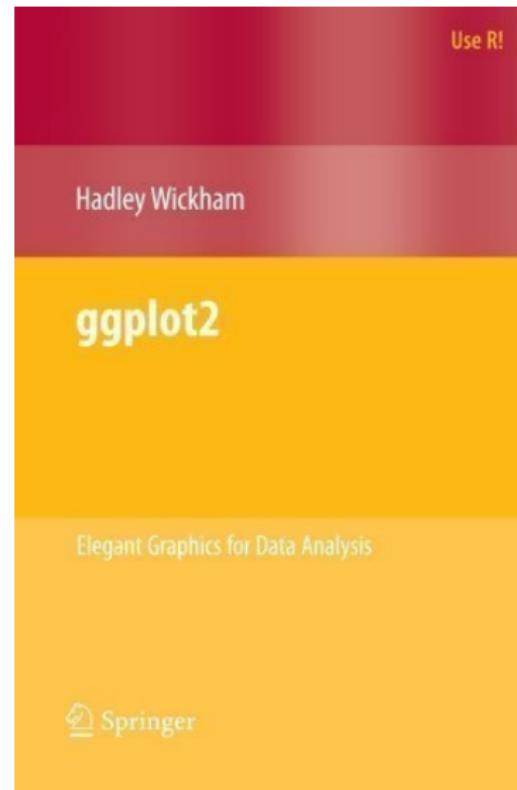


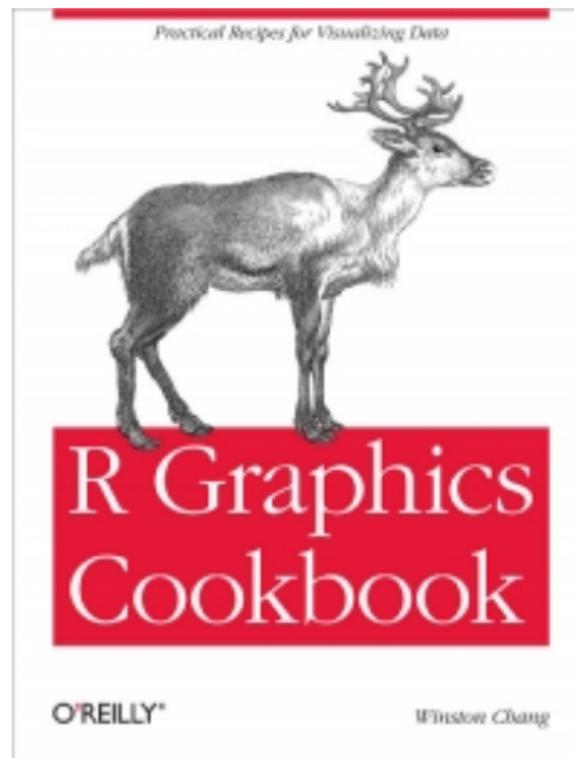
Yihui Xie

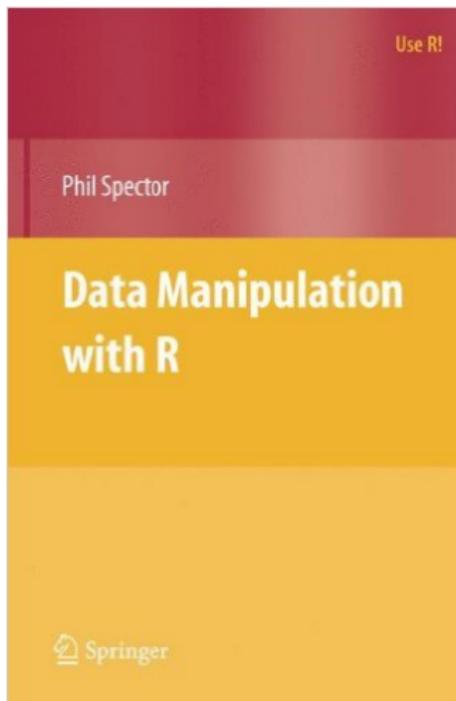


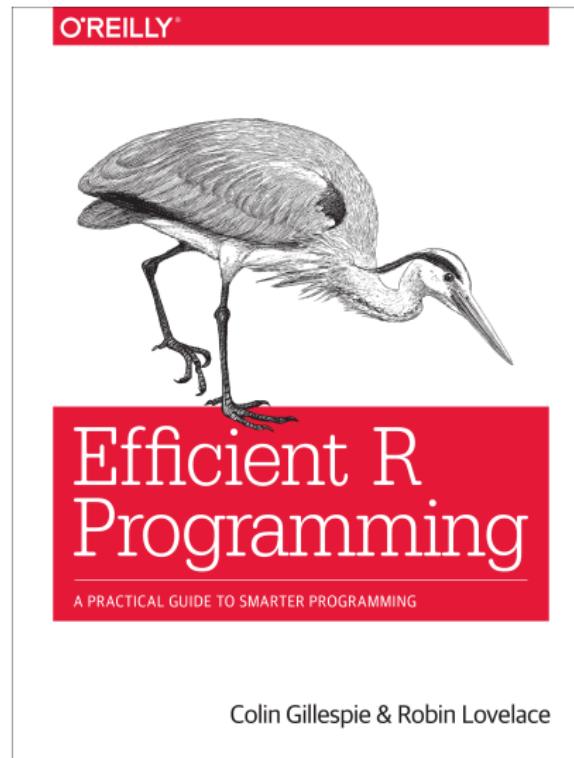
CRC Press
Taylor & Francis Group

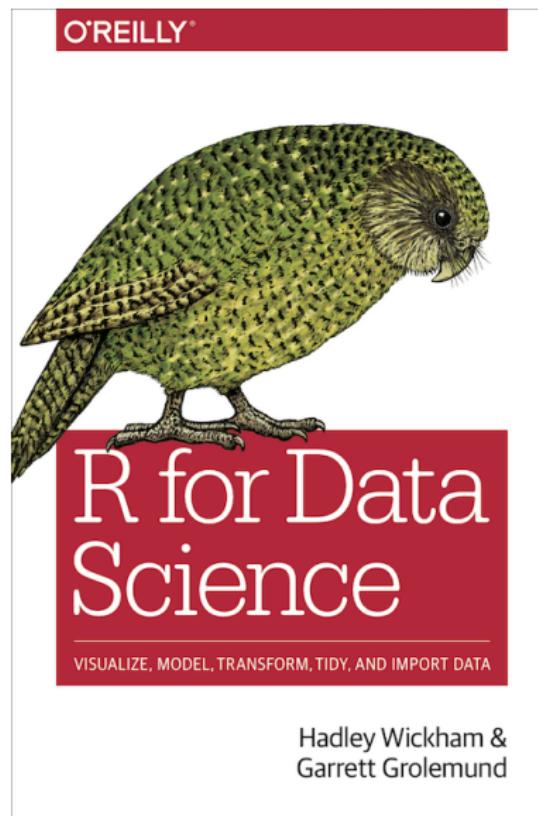
A CHAPMAN & HALL BOOK













Section 2

Introduction to R



What About Excel?





Excel is Great for Certain Things...

Screenshot of Microsoft Excel showing a student grades dataset.

The table has 28 rows and 14 columns. The columns are labeled as follows:

- A: Row
- B: Student
- C: Midterm
- D: Final
- E: Asn #1
- F: Asn #2
- G: Asn #3
- H: Asn #4
- I: Asn #5
- J: Asn #6
- K: V1
- L: V2
- M: Final Points
- N: Letter Grade

The data shows various student names and their scores across different assignments and final points. The letter grade column uses conditional formatting to color the text based on the value in column M. The cell containing "82.19" in row 11 is highlighted with a green border.

Row	Student	Midterm	Final	Asn #1	Asn #2	Asn #3	Asn #4	Asn #5	Asn #6	V1	V2	Final Points	Letter Grade
1	Student												
2	Student 1	95.5	91.78	100	100	100	100	100	100	94.54	93.42	94.54	A
3	Student 2	93.2	89.04	100	100	100	100	100	100	92.48	91.23	92.48	A
4	Student 3	95.5	86.3	100	100	100	100	100	100	91.80	89.04	91.80	A
5	Student 4	94.3	86.3	100	100	100	100	100	100	91.44	89.04	91.44	A
6	Student 5	95.5	82.88	100	100	75	100	100	100	89.26	85.47	89.26	A
7	Student 6	79.5	86.3	100	100	100	100	100	100	87.00	89.04	89.04	A
8	Student 7	84.1	85.6	100	100	100	100	100	100	88.03	88.48	88.48	A
9	Student 8	94.3	80.14	100	100	100	100	100	100	88.36	84.11	88.36	A
10	Student 9	94.3	80	100	100	100	100	100	100	88.29	84.00	88.29	A
11	Student 10	89.8	82.19	100	100	100	100	100	100	88.04	85.75	88.04	A
12	Student 11	90.9	81.51	100	100	100	100	100	100	88.03	85.21	88.03	A
13	Student 12	93.2	78.77	100	100	100	100	100	100	87.35	83.02	87.35	A
14	Student 13	89.8	81.5	100	75	100	100	100	100	86.86	84.37	86.86	A
15	Student 14	86.4	81.5	100	100	100	100	100	100	86.67	85.20	86.67	A
16	Student 15	93.2	76.71	100	100	100	100	100	100	86.32	81.37	86.32	A
17	Student 16	97.7	71.23	100	100	100	100	100	100	84.93	76.98	84.93	A-
18	Student 17	86.4	76.71	100	100	100	100	100	100	84.28	81.37	84.28	A-
19	Student 18	87.5	77.4	100	100	100	100	75	100	84.12	81.09	84.12	A-
20	Student 19	90.9	75.34	100	100	100	75	100	100	84.11	79.44	84.11	A-
21	Student 20	81.8	78.77	100	100	100	100	100	100	83.93	83.02	83.93	A-
22	Student 21	76.1	78.77	100	100	100	100	100	100	82.22	83.02	83.02	B+
23	Student 22	84.1	71.92	100	100	100	100	100	100	81.19	77.54	81.19	B+
24	Student 23	85.2	71.23	100	100	100	100	100	100	81.18	76.98	81.18	B+
25	Student 24	67	76.03	100	100	100	100	100	100	78.12	80.82	80.82	B+
26	Student 25	85.2	69.18	100	100	100	100	100	100	80.15	75.34	80.15	B+
27	Student 26	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+
28	Student 27	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+



...but Not Everything

Sample Data

- Six columns of data with ~ 1.05 million rows
- Column 5: `startDate`
- Column 6: `endDate`
- **Objective:** test to see if `endDate < startDate`



...but Not Everything

Sample Data

- Six columns of data with ~ 1.05 million rows
- Column 5: `startDate`
- Column 6: `endDate`
- **Objective:** test to see if `endDate < startDate`

RESULTS

- **Excel:** good luck...
- R: 33 min (poor coding technique)
- R: 58.5 sec (improved coding technique)



R or Python?





Vectorization in R

- Vectorized code saves time asking **type** questions
- There is an optimized engine—a basic linear algebra system (BLAS)—that is highly efficient at solving linear algebra problems
- A lot of R functions are written in C (or variants)
- MATLAB, Mathematica and the NumPy package for Python are also vectorized

<http://www.noamross.net/blog/2014/4/16/vectorization-in-r-why.html>



Why Use R?

- Open source (free)
- Runs on just about any platform
- Great visualization capabilities ([ggplot2](#))
- Read/write from/to various data sources
- Scripting language (interpreted)
- Deep library of advanced data manipulation and statistical packages



Installing R

- RStudio is a nice, user-friendly integrated development environment (IDE), but can be quirky at times — still highly recommended and what I will use in class
- Required to install R regardless
- You can even run R from a terminal window if you wish
- If you are curious to see some basic R demos, type `demo()` to see the list of demonstration code available and then run one if you like, e.g., `demo(persp)`



This is what R Looks Like

R version 3.2.4 (2016-05-10) -- very secure Vienna
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.67 (7152) x86_64-apple-darwin13.4.0]
[Workspace restored from /Users/Paul/.RData]
[History restored from /Users/Paul/.Rapp.history]

> |



This is what RStudio Looks Like

The screenshot displays the RStudio environment with the following components:

- Top Bar:** Shows the title bar with "userTrend.R" and "Q1Report.Rnw" tabs, and the status bar indicating the current working directory is "/workbench - RStudio".
- Left Panel:** Contains the R Script pane showing the following R code:

```
1 # User Trend Analysis
2 # Breakdown of active and non-active users
3
4 library(plyr)
5 library(ggplot2)
6
7(userData <- read.csv("userDataTrends.csv"))
8 userData <- subset(userData, select = -c(id, group))
9 userData$active <- as.factor(userData[,1])
10
11 states <- levels(userData$state)
12
13 names(userData)
14 count(userData, "active == 1")
15 View(userData)
16
17 summary(subset(userData, active == 1)$state)
18 summary(subset(userData, active == 0)$state)
19
20 ggplot(state, age, color = active, data = userData,
21   main = "Breakdown of Users by Age and State") +
22   opts(plot.title = theme_text(size = 19))
23 |
```

- Right Panel:** Shows the "Workspace" tab with the following information:
 - Data: userData (580 obs. of 5 variables)
 - Values: active (integer[270]), states (character[11])
 - Functions: split(group, location, ...)
- Bottom Panel:** Shows the "Plots" tab with a dot plot titled "Breakdown of Users by Age and State". The plot displays user age (y-axis, 0-70) against state (x-axis, IA, IL, IN, KS, MI, MN, MO, ND, NE, OH, SD). The legend indicates "active": 0 (red dots) and 1 (blue dots).



RStudio

RStudio has Four Panels

- Console
- Scripting/Viewing
- Files/Packages/Help/Viewer
- Environment/History/Plots

Open up RStudio for a guided tour `class01intro.R`



Notes on R

- R is case-sensitive
- It is best practice to use the assignment operator '`<-`' instead of the equality operator '`=`' for all code, even though both work, e.g.,

Syntax	Comments
<code>x <- 5</code>	standard syntax
<code>x = 5</code>	poor syntax, not permitted
<code>5 -> x</code>	awkward syntax, not permitted (but it works)

- Keyboard shortcut for assignment operator
 - Option (alt) + -



Basic R Functions

Function	Action
?foo	Help on the function <code>foo</code>
??foo	Search the help system for instances of the function <code>foo</code>
<code>RSiteSearch("foo")</code>	Search for the string <code>foo</code> in online help manuals and archived mailing lists
<code>data()</code>	List all available example datasets contained in currently loaded packages
<code>getwd()</code>	List the current working directory
<code>setwd("~/Desktop")</code>	Change working directory to <code>Desktop</code>
<code>rm()</code>	Remove (delete) one or more objects
<code>ls()</code>	List the objects in the current directory



The Best Place for Answers to R Questions?





Useful R Keyboard Shortcuts: Autocomplete

The screenshot shows the RStudio interface with the following details:

- Console:** Shows the command `> q`. A dropdown menu is open over the command, listing completions for the function `q` from the `base` package. The first item in the list is `q {base}`.
- Message:** Below the dropdown, a message states: "The function quit or its alias q terminate the current R session. Press F1 for additional help".
- Workspace:** Shows a list of files and folders in the current workspace:

Name	Size	Modified
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-data		
- File Explorer:** Shows a tree view of the project structure under the `04-large-data` folder.

A large, stylized `[Tab]` character is overlaid at the bottom left of the screenshot.



Useful R Keyboard Shortcuts: History

The screenshot shows the RStudio interface with the following details:

- Console (Left Panel):** Displays the history of R code entered. The last few lines shown are:

```
qplot(table ~ depth, data = diamonds,
qplot(day, data = email)
qplot(day, mails, data = daily, geom = "line", colo
qplot(day, mails, data = daily, geom = "smooth", co
qplot(day, variants, data = daily, geom = "line", c
qplot(wday, hour, data = wh, size = freq)
qplot(mpg, wt, data = mtcars)
qplot(mpg, wt, data = mtcars, colour = cyl)
```
- File Browser (Right Panel):** Shows the directory structure for the project "04-large-data". The contents of the "04-large-data" folder are listed in a table:

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
B		Feb 7, 2013, 9:37 AM
B		Feb 7, 2013, 9:37 AM
B		Feb 7, 2013, 9:37 AM
B		Feb 7, 2013, 9:37 AM

[Cmd/Ctrl + ↑]



Useful R Keyboard Shortcuts: Execute Subset of Code

The screenshot shows the RStudio interface. In the top-left pane, titled 'Untitled1*', there is a single line of R code: `library(ggplot2)`. In the bottom-left pane, titled 'Console', the same line is shown along with a prompt: `> library(ggplot2)`. To the right of the console is a large text overlay containing the keyboard shortcut: **[Cmd/ctrl + enter]**. The top-right pane is titled '04-large-data' and shows the file tree for the current project. The bottom-right pane is the 'Files' tab of the file browser, listing various files and their details.

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		



Useful R Keyboard Shortcuts: Execute All Code

The screenshot displays the RStudio interface. The top bar shows the path: ~/Documents/rstudio/training/Introduction to R/1-r-basics/1-basic-visualization/04-large-data - RStudio. The title bar says '04-large-data'. The left pane is the 'Console' with the command 'library(ggplot2)' entered. The right pane is the 'File Explorer' showing a file tree for '04-large-data' containing files like '04-large-data.html', 'overplot.png', 'transparent.png', etc. A large text overlay '[Cmd/ctrl + shift + enter]' is centered over the console area.

Name	Size	Modified
..		Feb 7, 2013, 9:37 AM
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		



Useful R Keyboard Shortcuts: Restarting an R Session

The screenshot shows the RStudio interface. In the top-left pane, the code `library(ggplot2)` is entered. In the bottom-left pane, the message `Restarting R session...` is displayed. On the right side, there is a file browser window showing various files in the directory `~/Documents/rstudio/training/Introduction to R/1-r-basics/1-basic-visualization/04-large-data`. Overlaid on the center of the screen is a large text box containing the keyboard shortcut **[Cmd/ctrl + shift + F10]**.

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		



IMPORTANT R Setting

Options

General

Code

Appearance

Pane Layout

Packages

Sweave

Spelling

Git/SVN

Publishing

Default working directory (when not in a project): ~

Restore most recently opened project at startup

Restore previously open source documents at startup

Restore .RData into workspace at startup

Save workspace to .RData on exit:

Always save history (even when not saving .RData)

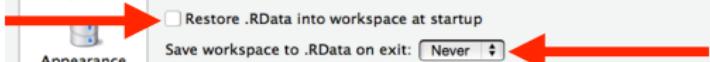
Remove duplicate entries in history

Use debug error handler only when my code contains errors

Automatically expand tracebacks in error inspector

Default text encoding: UTF-8

Automatically notify me of updates to RStudio





TRY THIS [IN A SCRIPTING WINDOW]

- ➊ $10,352 + 987,653$
- ➋ $10,352 / 987,653$
- ➌ $2^6 \times 99$
- ➍ $91,4545 \text{ modulus } 33$



TRY THIS [IN A SCRIPTING WINDOW]

- ① $10,352 + 987,653$
- ② $10,352 / 987,653$
- ③ $2^6 \times 99$
- ④ $91,4545 \text{ modulus } 33$

SOLUTIONS

① > 10352 + 987653

② > 10352 / 987653

③ > > 2**6 * 99 OR 2^6 * 99

④ > 914545 %% 33



A Brief Digression

- Whenever writing code, you want to be sure to clear your environment to ensure the fidelity of your results
- In each and every R script file I write, I always include the following two lines of code

```
rm(list=ls())
cat("\014")
```

- ① `rm(list=ls())` removes all objects in the current environment
- ② For Mac, `cat("\014")` clears the console window
 - same as keystroke `ctrl + l`



Packages in R

- Vanilla R comes with extensive capabilities
- ...BUT...
- some of the most exciting features in R are available as optional modules called Packages that you can download and install
- Packages are—like R—free, user-contributed modules that you can download and install
- There are ~ 10,000+ R packages [July 2018]



Packages in R

- Vanilla R comes with extensive capabilities
- ...BUT...
- some of the most exciting features in R are available as optional modules called Packages that you can download and install
- Packages are—like R—free, user-contributed modules that you can download and install
- There are ~ 10,000+ R packages [July 2018]
- Given R packages are user-contributed, some contain errors (some of which we have found right here in MSAN), so feel free to use R for analysis, but maybe double-check your output before you buy or sell billions of dollars of stock based on R package calculations



What are R Packages?

- Packages are collections of functions, data and compiled code in a well-defined format
 - The `library()` function shows you which packages are **saved** in your library (i.e., which packages have been downloaded)
- n.b.** `library()` **does not** tell you which packages are loaded, it only tells you which packages are downloaded
- `search()` tells you which packages are loaded and ready to use
 - `install.packages("myPackageName")` is used to install packages, and `library("myPackageName")` loads the package into your current working session
 - You only ever need to download a package once, but you always need to load packages when restarting an R session



Loading, Using & Maintaining R Packages [EXAMPLE]

```
> install.packages("dplyr")
    % Total      % Received % Xferd  Average Speed   Time     Time     Time   ...
      Dload  Upload Total    Spent   Left
0       0       0      0 --::-- --::-- --::-- ...
```

The downloaded binary packages are in
/var/folders/jm/3w7pqfms0nvg_ypvnvkkk83h0000gn/...

```
> summarise(iris, meanVal = mean(Sepal.Length))

Error: could not find function "summarise"

> dplyr::summarise(iris, meanVal = mean(Sepal.Length))
  meanVal
1 5.843333
```



Loading, Using & Maintaining R Packages [EXAMPLE] [CONT'D]

```
# this loads the package
> library(dplyr)

# it can now be called without the dplyr:: prefix
> summarise(iris, meanVal = mean(Sepal.Length))
  meanVal
1 5.843333
```

- Packages are often updated by their authors, so be sure to keep your packages up to date
 - `update.packages()` will update all packages tab
 - `installed.packages()` will list all packages you have downloaded, along with their version numbers, dependencies and other information



Loading, Using & Maintaining R Packages [CONT'D]

A far less cumbersome way to install, load and update packages in RStudio is using the appropriate icons in the “Packages” window

The screenshot shows the RStudio interface with the "Packages" tab selected in the top navigation bar. Below the navigation bar, there are two buttons: "Install" and "Update". A search bar and a refresh icon are also present. The main area displays a table of installed packages:

	Name	Description	Version	Actions
<input type="checkbox"/>	geepack	Generalized Estimating Equation Package	1.2-0.1	
<input checked="" type="checkbox"/>	ggplot2	An Implementation of the Grammar of Graphics	2.1.0	
<input type="checkbox"/>	ggvis	Interactive Grammar of Graphics	0.4.2	
<input type="checkbox"/>	git2r	Provides Access to Git Repositories	0.15.0	
<input type="checkbox"/>	googleVis	R Interface to Google Charts	0.5.10	
<input checked="" type="checkbox"/>	graphics	The R Graphics Package	3.2.4	
<input checked="" type="checkbox"/>	grDevices	The R Graphics Devices and Support for Colours and Fonts	3.2.4	
<input type="checkbox"/>	grid	The Grid Graphics Package	3.2.4	



Notes on R Packages

Packages sometimes (often?) have dependencies on other packages, e.g.,

```
#load the package MatchIt  
> library(MatchIt)  
Loading required package: MASS
```

- Without asking, when loading the `MatchIt` package, the `MASS` package is automatically loaded
 - This is helpful in one respect, since `MatchIt` leverages certain functionality from `MASS`; if `MASS` wasn't automatically loaded, then calling certain functions from `MatchIt` might throw an error
- n.b.** Be careful of function masking: because there are so many R packages available from so many different authors, it often happens that different packages have identically named functions



Notes on R Packages [CONT'D]

When calling a function, R searches the Global Environment first, then iterates through all packages for the function, beginning from the most recently added

```
> search()
[1] ".GlobalEnv"      "package:reshape2"   "package:plyr"
[4] "package:MatchIt"  "package:MASS"     "package:ggplot2"
[7] "tools:rstudio"    "package:stats"    "package:graphics"
[10] "package:grDevices" "package:utils"    "package:datasets"
[13] "package:methods"   "Autoloads"      "package:base"

> (.packages())
[1] "reshape2"    "plyr"        "MatchIt"     "MASS"       "ggplot2"    "stats"
[7] "graphics"    "grDevices"   "utils"      "datasets"   "methods"    "base"
```



Notes on R Packages [CONT'D]

When calling a function, R searches the Global Environment first, then iterates through all packages for the function, beginning from the most recently added

```
> search()
[1] ".GlobalEnv"      "package:reshape2"   "package:plyr"
[4] "package:MatchIt"  "package:MASS"       "package:ggplot2"
[7] "tools:rstudio"    "package:stats"     "package:graphics"
[10] "package:grDevices" "package:utils"     "package:datasets"
[13] "package:methods"   "Autoloads"        "package:base"

> (.packages())
[1] "reshape2"    "plyr"        "MatchIt"     "MASS"        "ggplot2"    "stats"
[7] "graphics"    "grDevices"   "utils"       "datasets"   "methods"    "base"
```

- If you are using a lot of packages and want to be certain you are calling a function from a specific package, use the double colon operator, e.g., `plyr::rename()`



Notes on R Packages [EXAMPLE]

TRY THIS

```
> library(magrittr)
> library(dplyr)

> iris %>% select(Petal.Length, Species) %>% group_by(Species) %>%
  summarize(meanVal = mean(Petal.Length))

> library(Hmisc)

> iris %>% select(Petal.Length, Species) %>% group_by(Species) %>%
  summarize(meanVal = mean(Petal.Length))

Error in summarize(., meanVal = mean(Petal.Length)) :
  argument "by" is missing, with no default
```



Notes on R Packages [EXAMPLE]

TRY THIS

```
> library(magrittr)
> library(dplyr)

> iris %>% select(Petal.Length, Species) %>% group_by(Species) %>%
  summarize(meanVal = mean(Petal.Length))

> library(Hmisc)

> iris %>% select(Petal.Length, Species) %>% group_by(Species) %>%
  summarize(meanVal = mean(Petal.Length))

Error in summarize(., meanVal = mean(Petal.Length)) :
  argument "by" is missing, with no default
```

SOLUTION

```
> iris %>% select(Petal.Length, Species) %>% group_by(Species) %>%
  dplyr::summarize(meanVal = mean(Petal.Length))
```



Notes on R Packages [CONT'D]

- If you feel you have too many packages loaded and want to unload (detach) one, you can use the following command:
`detach("package:MatchIt", unload = TRUE)`
- **n.b.** Be aware that when unloading MatchIt you do not automatically unload the dependency that was loaded when you loaded `MatchIt`, namely, `MASS`
- You may choose to use `require("myPackageName")` in lieu of `library("myPackageName")` to load a package, but be cautious when doing so
 - `require()` attempts to load a package and returns a logical to indicate whether or not the could not be loaded; if the package could not be loaded, a `warning` is thrown
 - `library()` attempts to load a package and throws an `error` if the package could not be loaded



How Big is Big Data in R?

- R holds data in memory, effectively limiting data to the amount of RAM a computer has access to
- It is not uncommon to work with a data set containing 100,000,000 elements (e.g., 100,000 observations of 1,000 variables or 1,000,000 observations of 100 variables) without difficulty
- Of course all of the above approximations depend on what type of data is contained in each variable, e.g., I am currently working on a data set with 2.2 million records and twenty variables, which takes approximately one minute to load into memory
- The other issue to consider is what techniques and/or functions will be applied to the data



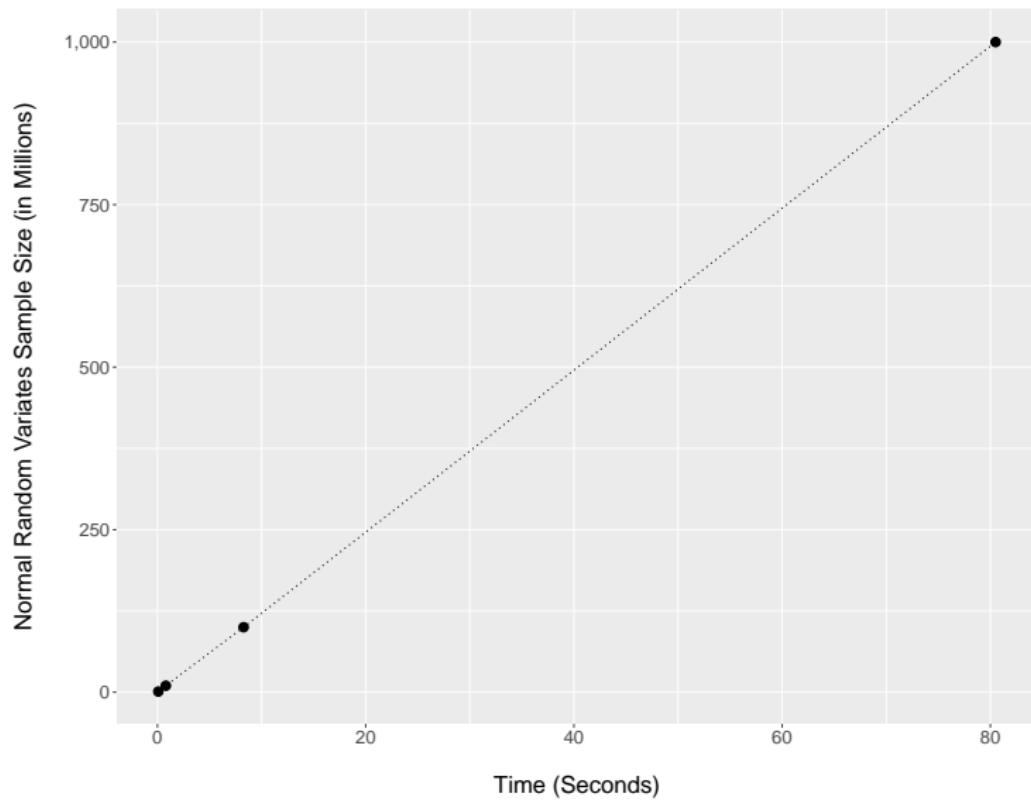
How Big is Big Data in R? [CONT'D]

- The more complex and memory intensive the task, the smaller the data will be required to be
- Basic plotting would likely require far less computational exertion than a complex statistical learning model



TRY THIS

- ① Create a vector named `myVec` with 1,000,000 random $\sim \mathcal{N}(10, 5)$ variates. How much time does this take?
- ② Repeat and time with 10,000,000 random variates
- ③ Repeat and time with 100,000,000 random variates
- ④ Repeat and time with 1,000,000,000 random variates
- ⑤ Generate a graph with *Sample Size* on the *y* axis and *Time* on the *x* axis





SOLUTION

```
# create vector and time execution
system.time(
  myVec <- rnorm(1000000, mean = 10, sd = 5)
)

# load tidyverse to access tibble and ggplot packages
library(tidyverse)

myDF <- tibble(
  sampleSize = c(1, 10, 100, 1000),
  time = c(0.079, 0.805, 8.274, 80.5)
)

library(magrittr) # for piping
library(scales)   # for commas in y axis

# create graph in ggplot
myDF %>%
  ggplot(aes(x = time, y = sampleSize)) +
  geom_line(linetype = "dotted") +
  geom_point(size = 3) +
  xlab("\nTime (Seconds)") +
  ylab("Normal Random Variates Sample Size (in Millions)\n") +
  scale_y_continuous(labels = comma)
```



Data Sets in R?

- R comes built in with multiple data sets you can play with
- Many (most?) packages also have data sets
- `data()` will bring up a list of all data sets available across all loaded packages
- `help(<nameOfDataSet>)` will provide you a detailed description of the data set in question