

Lecture 1: Introduction



UNIVERSITY OF
SAN FRANCISCO

James D. Wilson

BSDS 100 - Intro to Data Science with R



Outline

- Course Overview
- What is Data Science?
 - Where is Data Science?
 - A brief history
- The R Programming Language and RStudio
 - Comparison with other programming languages
 - Installation
 - Handy Shortcuts

Part I: Course Overview



A Little About Me

- Ph.D. Statistics and Operations Research (UNC Chapel Hill, '15)
 - Research focused on statistical analysis of networks
 - Explore, model, and analyze network data (e.g., social networks)
- M.S. Mathematical Sciences (Clemson University, '10)
- B.S. Mathematics and Chemistry (Campbell University '08)



A Little About Me

Classes I teach:

- BSDS 100 - Intro to Data Science with R
- MATH 106 - Business Statistics
- MATH 370 - Probability with Applications
- MATH 373 - Statistical Learning
- MSAN 601 - Linear Regression Analysis
- MSAN 630 - Advanced Computational Statistics
- MSAN 700 - Social Network Analysis



A Little About Me

- Born and raised in NC (near Raleigh)
- Live in Rockridge, Berkeley.
- A huge college basketball fan! (Go Heels!)
- Have loved college football since 2008 (Go Tigers!)
- Enjoy tasting beers (bourbon-barrel stouts are my favorite).

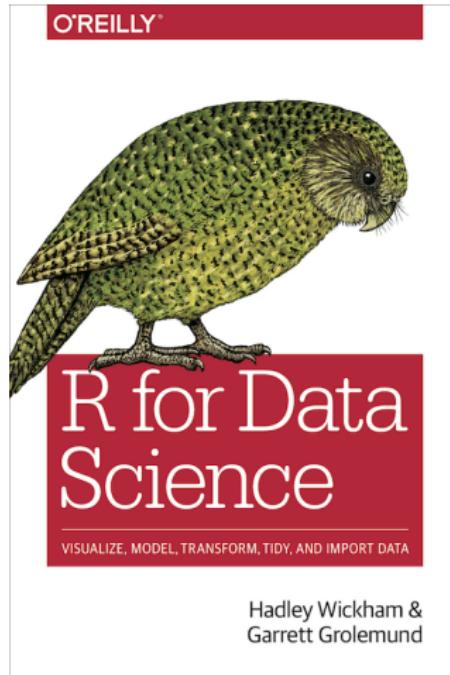


Course Description and Syllabus

All lecture notes, the syllabus, assignments, and course description are available at this site: <https://github.com/jdwilson4/Intro-Data-Science-2017>



Main Text



Available online here: <http://r4ds.had.co.nz/index.html>



Other Resources



The Best Place for Answers to R Questions?



Part II: What is Data Science?



What is Data Science?

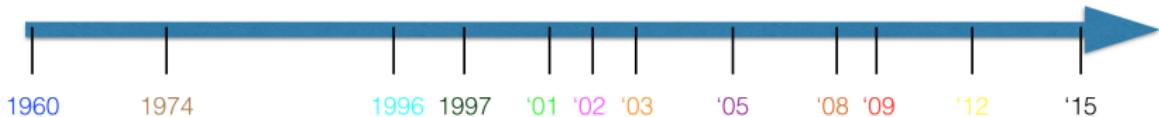
- **Wikipedia:** “the extraction of knowledge from data.”
- A precise definition is a bit unclear and has faced much controversy... (we'll see more on this in a moment)
- Practitioners tend to agree on the *components* of data science:
 - gathering and cleaning data
 - database management
 - exploratory analysis
 - predictive modeling
 - data summary and visualization

Where is Data Science?



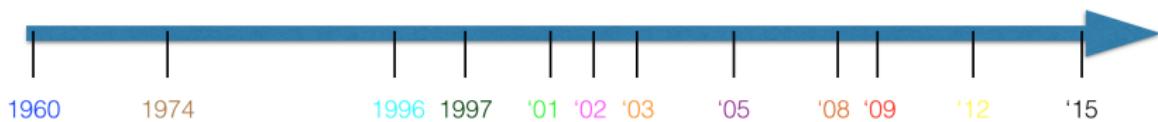
- Twitter feed, December 2014

The Evolution of Data Science



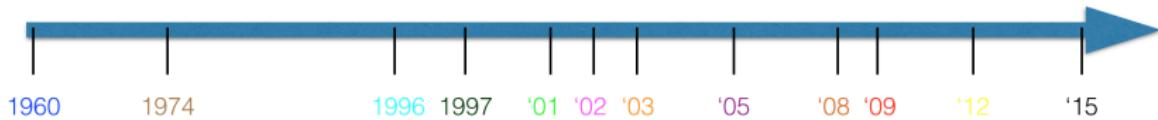
- 1960: Peter Naur (CS Ph.D.) published *Datalogy: the science of data and its place in education*.
- 1974: Peter Naur published *Concise Survey of Computer Methods*.
 - defines data science as “the science of dealing with data, once they have been established.”
 - continues to say that “... the relation of the data to what they represent is delegated to other fields and sciences.”

The Evolution of Data Science



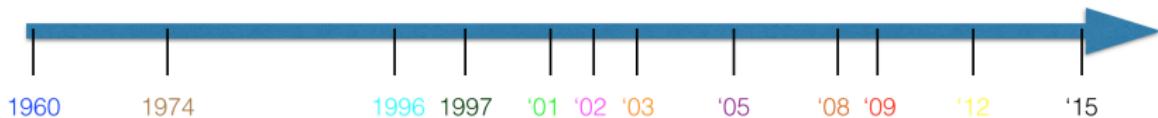
- 1996: International Federation of Classification Societies meet in Tokyo and for the first time include "data science" in the conference title: "Data science, classification, and related methods."
- 1997: C.F. Jeff Wu gave the inaugural lecture "Statistics = Data Science?" for appointment to the H. C. Carver Professorship at the University of Michigan.

The Evolution of Data Science



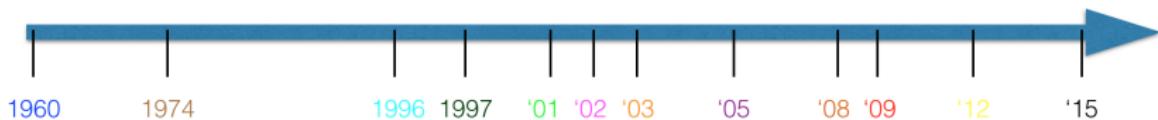
- **2001:** William Cleveland (Bell Labs) published *Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics*.
 - Sets forth 6 areas for a university department involving statistics.
- **2002:** *Data Science Journal* is launched
 - Focus on data systems, publications on internet, and applications
- **2003:** *Journal of Data Science* is launched
 - Focus on application of statistical and quantitative methods

The Evolution of Data Science



- **2005:** National Science board redefines data scientists:
 - "The information and computer scientists, data and software programmers, disciplinary experts, ... who are crucial to successful management of a digital data collection whose primary activity is to conduct creative inquiry and analysis"
- **2008:** DJ Patil (LinkedIn) and Jeff Hammerbacher (Facebook) coined the term "data scientist" to define their jobs

The Evolution of Data Science



- **January, 2009:** Hal Varian (chief economist at Google) writes that "... the sexy job in the next 10 years will be statisticians."
- **October, 2012:** Harvard Business Review publishes "Data Scientist: The Sexiest Job of the 21st Century."
- **February 5th, 2015:** DJ Patil appointed as the first Chief Data Scientist in the White House.



Applications



Marketing analytics, sports analytics, biotechnology, social experiments, e-commerce, government analysis, ...

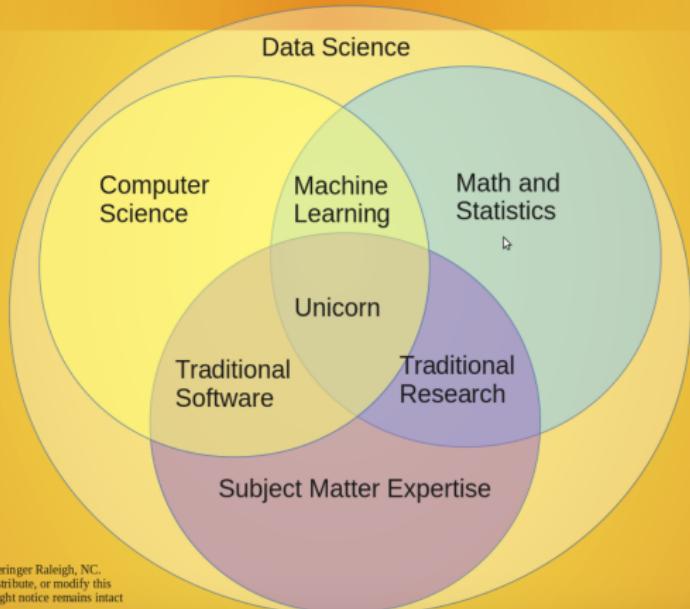


Why Data Science?

- Size, complexity, and amount of data
 - Predicted ≈ 40 trillion gigabytes of data in 2020; up from 130 billion in 2005!
 - **Big data** requires innovative techniques for analysis
- *McKinsey*: "The U.S. faces a shortage of 140K - 190K people with analytical expertise and 1.5 million managers and analysts with the skills to understand and make decisions based on the analysis of big data." (May, 2011)
- *Harvard Business Review*: "Data Scientist: The Sexiest Job of the 21st Century." (October, 2012)



Data Science Venn Diagram v2.0



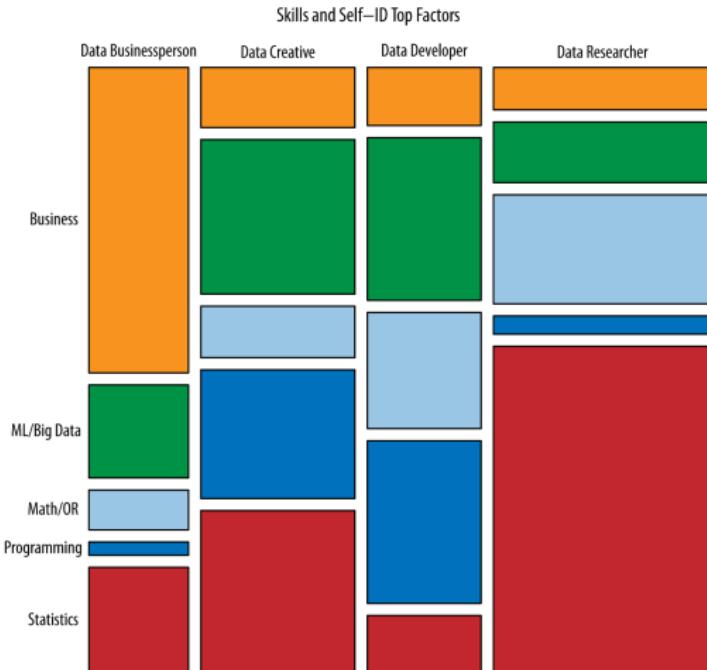
Copyright © 2014 by Steven Geringer Raleigh, NC.
Permission is granted to use, distribute, or modify this
image, provided that this copyright notice remains intact.



- The field is inherently interdisciplinary
 - mathematical statistics
 - computer science
 - domain expertise
- The magical **Unicorn**: having all three skills
 - In 2014, these jobs go unfilled for 6 months or longer on average
- Has lead to the development of data science *teams*
 - hope is to merge skills of analysts



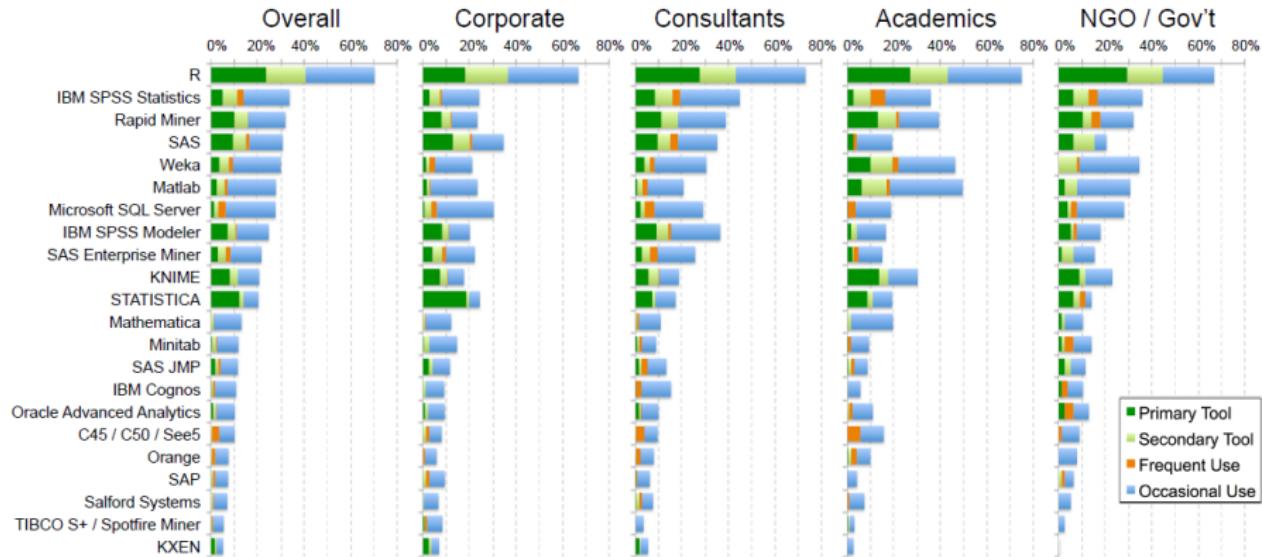
The Analysts of Data Science



"Analyzing the Analyzers (2013) by Harry, Murphy, and Vaisman."



Software: A Data Scientist's first weapon



-www.datasciencecentral.com



Data Science in Academia

ABOUT USF DESTINATIONS GATEWAYS SEARCH

USF College of Arts and Sciences

Berkeley School of Information

datascience@berkeley

GEORGETOWN UNIVERSITY

SCHOOL OF INFORMATION STUDIES SYRACUSE UNIVERSITY

CAS in Data Science

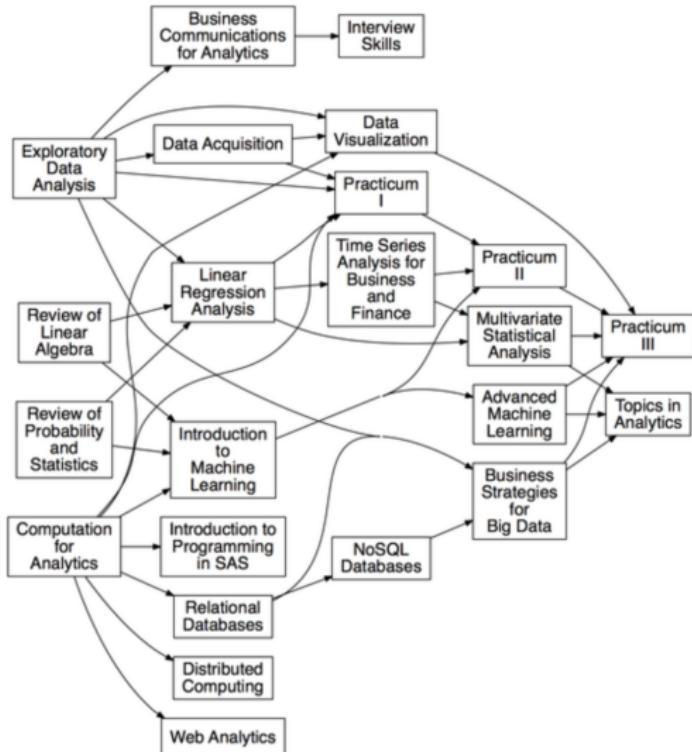
James D. Wilson (USF)

Lecture 1: Introduction

25 / 53



Academic Programs





A Data Scientist's Toolkit

Harvard's data science [toolkit](#):

- ① **Wrangle the data:** gather, clean, and sample data
- ② **Manage the data:** access big data quickly and reliably
- ③ **Explore the data:** to make a hypothesis
- ④ **Make predictions:** statistical methods
- ⑤ **Communicate the results:** visualization, presentations, summaries



Get Involved! Great Resources

- Flowingdata.com
 - Contemporary visualization and data manipulation techniques
- Kaggle.com
 - Kaggle competitions: win money for solving problems!
- Coursera.org
 - Free online courses in data science and machine learning
 - 972 courses. Great resource for coding, data analysis, etc.
 - Recent notable course: "The Data Scientist's Toolbox."

Part III: R and rStudio



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)
- Massive library of data manipulation and statistical packages



But... what about Excel?





Excel is Great for Certain Things...

Screenshot of Microsoft Excel showing a student grade sheet. The table has 28 rows and 14 columns. The columns are labeled A through M, and the rows are numbered 1 through 28. Row 1 contains the headers: Student, Midterm, Final, Asn #1, Asn #2, Asn #3, Asn #4, Asn #5, Asn #6, V1, V2, Final Points, and Letter Grade. Row 28 contains the last student's data. The 'Letter Grade' column uses a formula to determine the grade based on the 'Final Points'. The 'Final Points' column is calculated as the average of the Midterm, Final, and six assignments. The 'Letter Grade' column uses a color-coded formula to assign grades (A, A-, B+, etc.). The 'Final Points' column for Student 11 is currently being edited, showing '82.19'.

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

RESULTS

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



...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)

How about 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>



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
- You can even run R from a terminal window if you wish
- Download and install at this website:
<https://www.r-project.org>
- **Important:** You will have to re-install R from time-to-time to maintain the newest version so that code remains compatible!
New versions generally come out every 4 - 6 months.



Installing RStudio

- RStudio has a very nice graphical user interface (GUI) that is easier to use than base R
- We will be using this throughout the course
- Make sure that you have R installed first. Then, download and install at this website:

<https://www.rstudio.com/products/RStudio/>

The R Graphical User Interface (GUI)



R Console

R version 3.2.4 (2016-05-10) -- very secure DNSes
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]

> |





The RStudio GUI

~/workbench - RStudio

userTrend.R* | Q1Report.Rnw | userData*

Source on Save Run Source

```
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 aplot(state, age, color = active, data = userData,
21     main = "Breakdown of Users by Age and State") +
22     opts(plot.title = theme_text(size = 19))
23
```

(Top Level) R Script

Console ~ / ↵

```
active....1 freq
1 FALSE 310
2 TRUE 270
> View(userData)
> summary(subset(userData, active == 1)$state)
IA IL IN KS MI MN MO ND NE OH SD
19 26 21 21 27 49 22 26 19 16 24
> summary(subset(userData, active == 0)$state)
IA IL IN KS MI MN MO ND NE OH SD
26 27 18 31 27 49 23 32 19 33 26
> aplot(state, age, color = active, data = userData,
+     main = "Breakdown of Users by Age and State") +
+     opts(plot.title = theme_text(size = 19))
>
```

Workspace History

Data

userData 580 obs. of 5 variables

Values

active integer[270]

states character[11]

Functions

split(group, location, ...)

Files Plots Packages Help

Zoom Export Clear All

Breakdown of Users by Age and State

active
0
1



RStudio has Four Panels

- Console (bottom left)
- Scripting/Viewing (top left)
- Files/Packages/Help/Viewer (top right)
- Environment/History/Plots (bottom right)



Really Advanced Calculators

At their core, R and RStudio are just calculators! Try the following

$$3 + 2 = ? \quad \log(10) = ? \quad \sqrt{32} = ?$$

```
> 3 + 2
```

```
> log(10)
```

```
> sqrt(32)
```



Notes on R

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

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



Basic R Help 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>data()</code>	List all available example datasets contained in currently loaded packages
<code>getwd()</code>	List the current working directory
<code>ls()</code>	List the objects in the current directory



Basic R Workspace Functions

Function	Action
<code>getwd()</code>	List the current working directory.
<code>setwd("mydirectory")</code>	Change the current working directory to <i>mydirectory</i> .
<code>ls()</code>	List the objects in the current workspace.
<code>rm(objectlist)</code>	Remove (delete) one or more objects.
<code>help(options)</code>	Learn about available options.
<code>options()</code>	View or set current options.
<code>history(#)</code>	Display your last # commands (default = 25).
<code>savehistory("myfile")</code>	Save the commands history to <i>myfile</i> (default = .Rhistory).
<code>loadhistory("myfile")</code>	Reload a command's history (default = .Rhistory).
<code>save.image("myfile")</code>	Save the workspace to <i>myfile</i> (default = .RData).
<code>save(objectlist, file="myfile")</code>	Save specific objects to a file.
<code>load("myfile")</code>	Load a workspace into the current session (default = .RData).
<code>q()</code>	Quit R. You'll be prompted to save the workspace.



Useful R Keyboard Shortcuts: Autocomplete

The screenshot shows the RStudio interface. In the top-left pane, there's an empty script editor window titled "Untitled1". The main pane, labeled "Console", contains the command "q" followed by a list of suggestions: "q {base}", "qbeta {stats}", "qbinom {stats}", "qbirthday {stats}", "qcauchy {stats}", and "achieve_estratal". A tooltip explains that "The function quit or its alias q terminate the current R session." The bottom-left area features a large "[Tab]" character. The right side of the interface includes the "Workspace", "History", and "Git" tabs, along with a file browser showing files like "caratcut.png", "expensive.png", "blue.png", "caratbox.png", "blue2.png", "slides.md", and "04-large-data.RData".



Useful R Keyboard Shortcuts: History

The screenshot shows the RStudio interface with the following details:

- Code Editor:** An R script named "Untitled1" is open, showing code related to ggplot2 and mtcars datasets.
- History:** The history pane at the bottom displays a series of command-line entries, including "qplot(mpg, wt, data = mtcars, colour = cyl)" and "q".
- File Explorer:** The right-hand sidebar shows a file tree for a directory named "04-large-data".
- Keyboard Shortcut Overlay:** A large, semi-transparent overlay at the bottom features the text "[Cmd/Ctrl] + ↑" in a large, bold, sans-serif font, indicating the keyboard shortcut for navigating through the history.



Useful R Keyboard Shortcuts: Execute Code

The screenshot shows the RStudio interface. In the top-left code editor, the command `library(ggplot2)` is typed. A large bracketed text overlay "[Cmd/ctrl + enter]" is centered over the code editor area. In the bottom-left console, the command is run and its output is shown. The bottom-right file browser lists several files and folders related to the "04-large-data" project.

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: Restarting an R Session

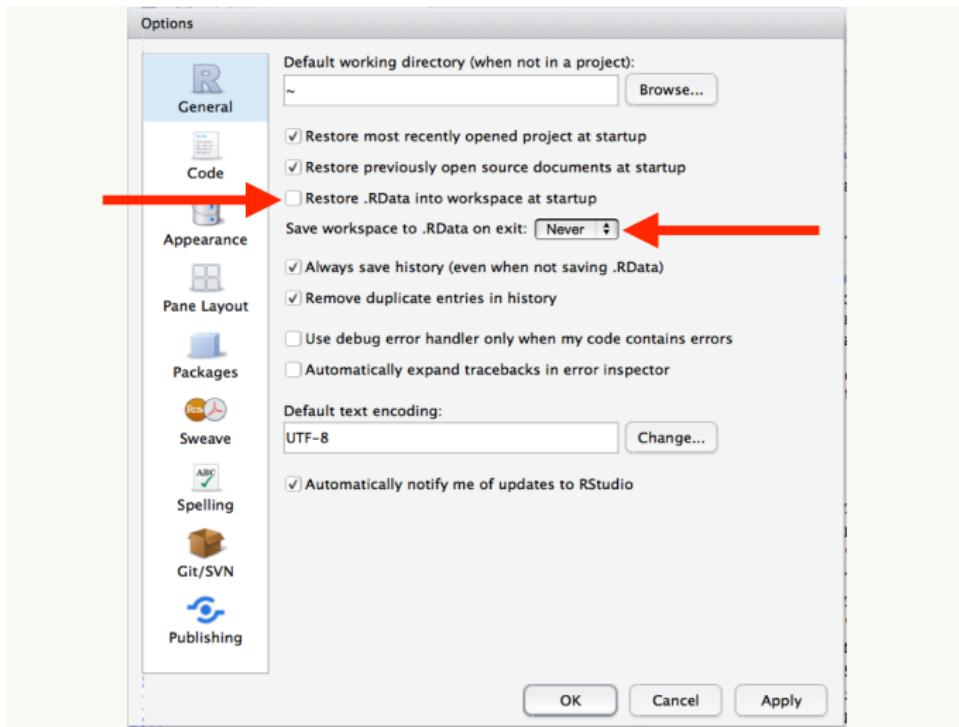


The screenshot shows the RStudio interface. In the top-left corner, there's a large text box containing the keyboard shortcut: **[Cmd/ctrl + shift + F10]**. Below this, the RStudio window is visible, divided into several panes:

- Console** pane: Shows the command `> library(ggplot2)` being entered.
- Output** pane: Shows the message "Restarting R session...".
- File Explorer** pane: Shows a file tree for a directory named "04-large-data". The tree includes files like "04-large-data.html", "overplot.png", "transparent.png", etc., all modified on Feb 7, 2013, at 9:37 AM.



IMPORTANT R Setting





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")
```

- ➊ `rm(list=ls())` removes all objects in the current environment
- ➋ On a Mac, `cat ("\014")` clear the console windows (same as `ctrl + l`)



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



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
- Approximations depend on what type of data is contained in each variable, e.g., a data set with 2.2 million records and twenty variables, which takes approximately one minute to load into memory



How Big is *Big Data* in R?

- Also depends on what techniques and/or functions will be applied to the data
- The more complex and memory intensive the task, the smaller the data will be required to be
- Basic plotting requires far less computational exertion than a complex statistical learning model
- **Common Definition:** *Big Data* refers to any data set that cannot be loaded into working memory on your personal computer



Assignment 1

- Read Chapter 1 of Doing Data Science by Cathy O'Neil and Rachel Schutt entitled "What is Data Science?" [here](#). Be ready to answer questions about this in the next class.