Analysis of EHS Data

Class 1:
Part A: Course overview/expectations
Part B: Introduction to R/Markdown/ggplot2

January 11, 2020

What we will cover today:

A. Course overview
B. Getting started
    - R
    - Markdown
    - ggplot

Introductions. . .

Analytic skills: the demand is high

https://www.pwc.com

Where do you want to go next?

Health Department?

Non-profit?

For profit?

PhD program?

Medical school?

Before you start, you will need to be comfortable with data!

- From the syllabus:

Data is the foundation of all research and becoming comfortable reading about, describing, analyzing, interpreting, summarizing and presenting EHS data is critical for the success of all environmental health scientists.

Therefore, all EHS MPH (and PhD, MS, etc) graduates should know how to work with the types of data they will typically encounter in EHS-related research settings.

Class Flow

By Thursday before class:

- Lecture posted (via courseworks)
- Learning Check posted (via courseworks)

On Mondays:

- **12 noon:** Homework due (via courseworks); Learning Check due (via courseworks)
- **1p:** Homework "answers" released (via courseworks: recording, RMD, and HTML files will be posted); you should review before 2p.
- **2p:** Live session begins (via zoom)
    - **2p-2:10p**: Housekeeping issues; answer any questions about homework that was due
    - **2:10p-2:30p**: Review weekly learning check and go over main points of new material (from recorded lecture)
    - **2:30p-3:30p**: Live demonstration of new material
- ~3:30p: Live session ends; new homework released

Weekly Learning Checks

- Each week after you listen to the lecture, you should complete the weekly “Learning Check”, which you can find in the Quizzes tab on courseworks.
- They will consist of 3-5 short, multiple choice questions that you should be able to answer pretty easily after listening to the lecture.

- Learning Checks will be worth 10% of your grade.

Homeworks

- Turn in: html file from .Rmd code (more later)
  - html file should include code, notes, and output
    - If you can generate an html file, your code "works"; if there are parts of your code that do not work, we will teach you how to block them out
    - If you can't, there's an error; you can submit your .rmd and a word doc instead but will not get full credit.
- Due MONDAY 12 noon the day class - late homework will generally not be accepted*
  - All assignments will be submitted through courseworks
  - Syntax inventory will be submitted through courseworks
- Put your name at the top of the html file (in YAML) and also include it in your file name!!

  `"Herbstman_HW1.html"`
- Can work together, but the work you submit must be your own

* We are in the middle of a global pandemic. If you are unable to turn your homework in on-time, please reach out to me (Julie) about an exceptions, which will be made on a case-by-case basis.

Additional points about HW

- Grading:
  - 1 point for code that runs with no errors and annotation of what you're doing
  - 5 points for correct answers (partial credit given)
  - Graded within ~1 week with comments on why points were deducted (via courseworks)
  - Contact TA/TFs if you have questions about grading (if you request regrading, we will regrade the whole assignment)
- Homework assignments will be due weekly
  - 13 total, highest 12 will be counted)
  - Will count for 65% of your final grade

Final Assignments and Project

- Final Assignment: 15%
  - 2 weeks to complete (will be due near the end of finals)

- Compiled syntax inventory: 10%
  - We will walk through an example today. . .

Office Hours

- We will set up 2 office hours/week, with the goal of having at least one session that that you can 'generally' make.
- A note about contacting us other times:
  - Email TA/TF's first (include both) - we will try to respond within 24 hours
  - If you have questions about assignments, please use the Discussion Board on Courseworks
    - If you email us with homework questions, likely we will tell you to check the Discussion Board or to post your question on the Discussion Board
      - We will monitor the Discussion Board but please feel free to respond to each other as well.

Registration

- Everyone needs to be registered for the class to have access to CANVAS/COURSEWORKS
- If you are not yet registered for this class and need approval, please contact Nina Kulacki (njk2128@cumc.columbia.edu)
    - She will be able to sign-off on your registration, if necessary

To conduct data analysis, you need software

- Quick comparison. . .

Features	Stata	SPSS	SAS	R
Learning curve	Steep/gradual	Gradual/flat	Pretty steep	Pretty steep
User interface	Programming/point-and-click	Mostly point-and-click	Programming	Programming
Data manipulation	Very strong	Moderate	Very strong	Very strong
Data analysis	Powerful	Powerful	Powerful/versatile	Powerful/versatile
Graphics	Very good	Very good	Good	Excellent
Cost	Affordable (perpetual licenses, renew only when upgrade)	Expensive (but not need to renew until upgrade, long term licenses)	Expensive (yearly renewal)	Open source

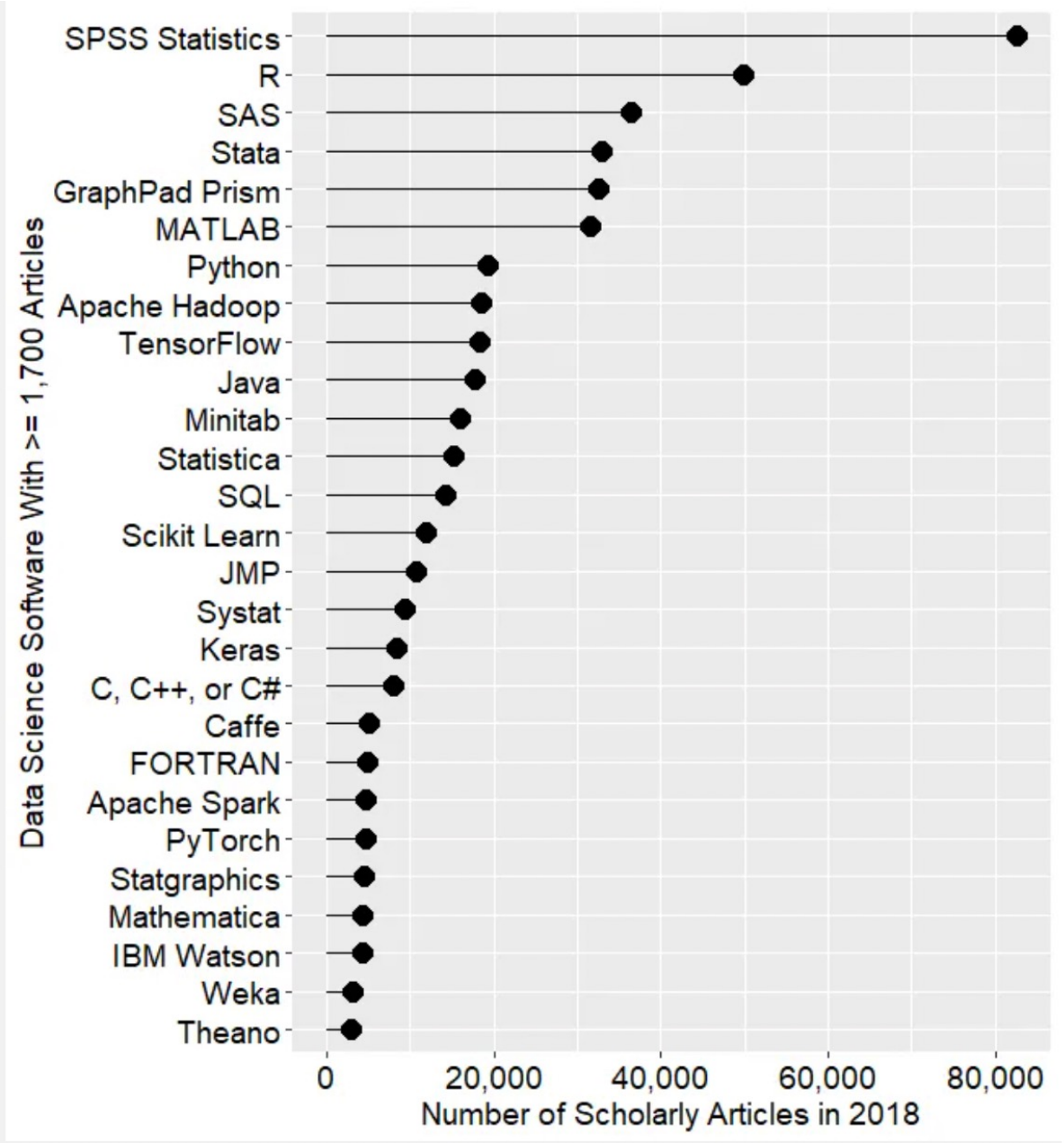
[https://dss.princeton.edu/training/RStata.pdf]

A major difference between R and others:
R is not a statistical package, it is a programming language that can be used for statistical analysis.

Why R?

- Very flexible (tons of packages)
  - Lots of applications
- It can hold more than one dataset in memory at the same time (this is a convenient feature!)
- Reproducibility (e.g., Excel vs. anything that uses a script/Markdown)
- Ease of workflow (e.g., copy and paste vs. Markdown)
- FREE

R vs Others



[http://r4stats.com/articles/popularity/]

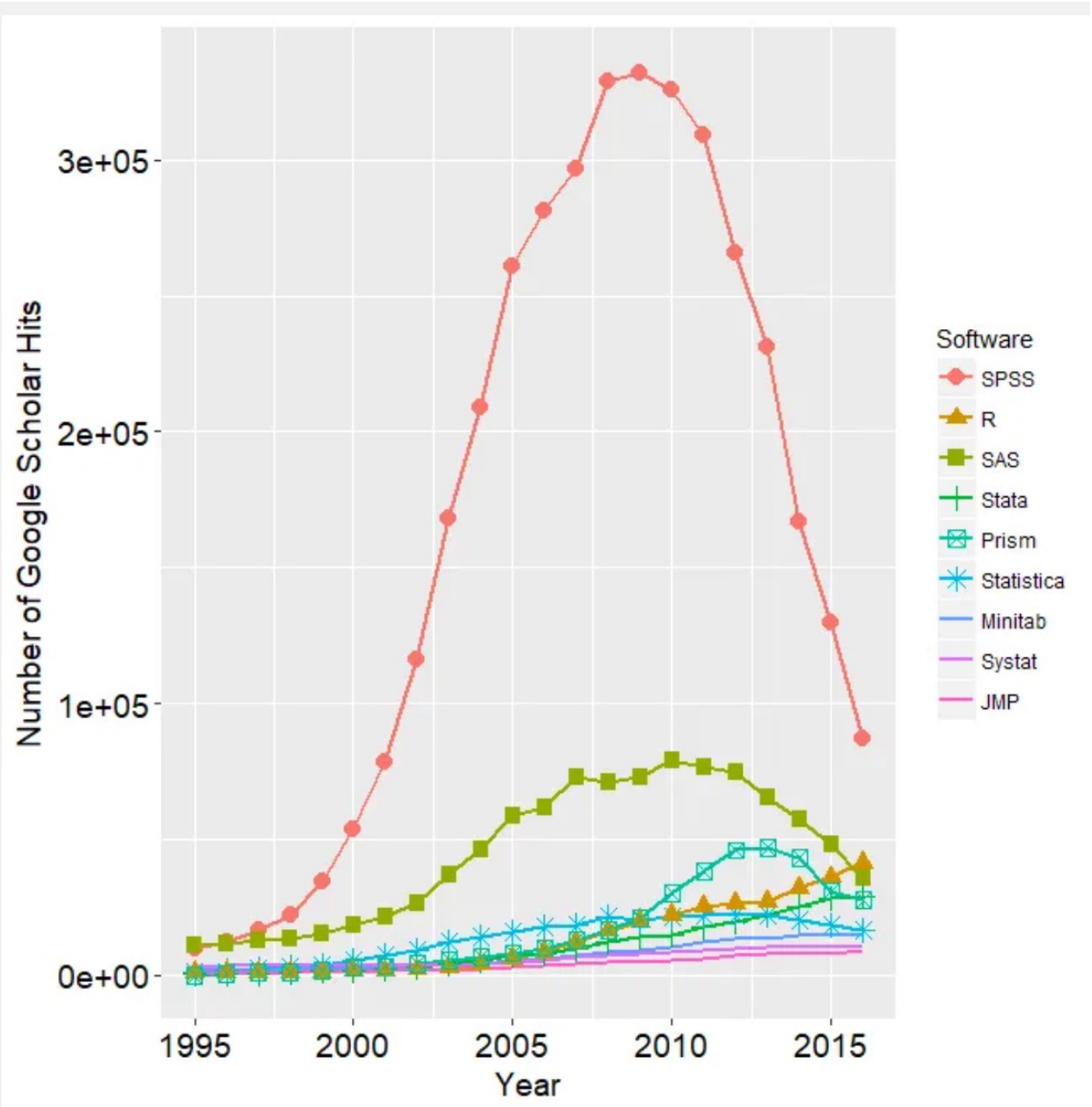


Figure 2d. The number of Google Scholar citations for each classic statistics package per year from 1995 through 2016.

[http://r4stats.com/articles/popularity/]

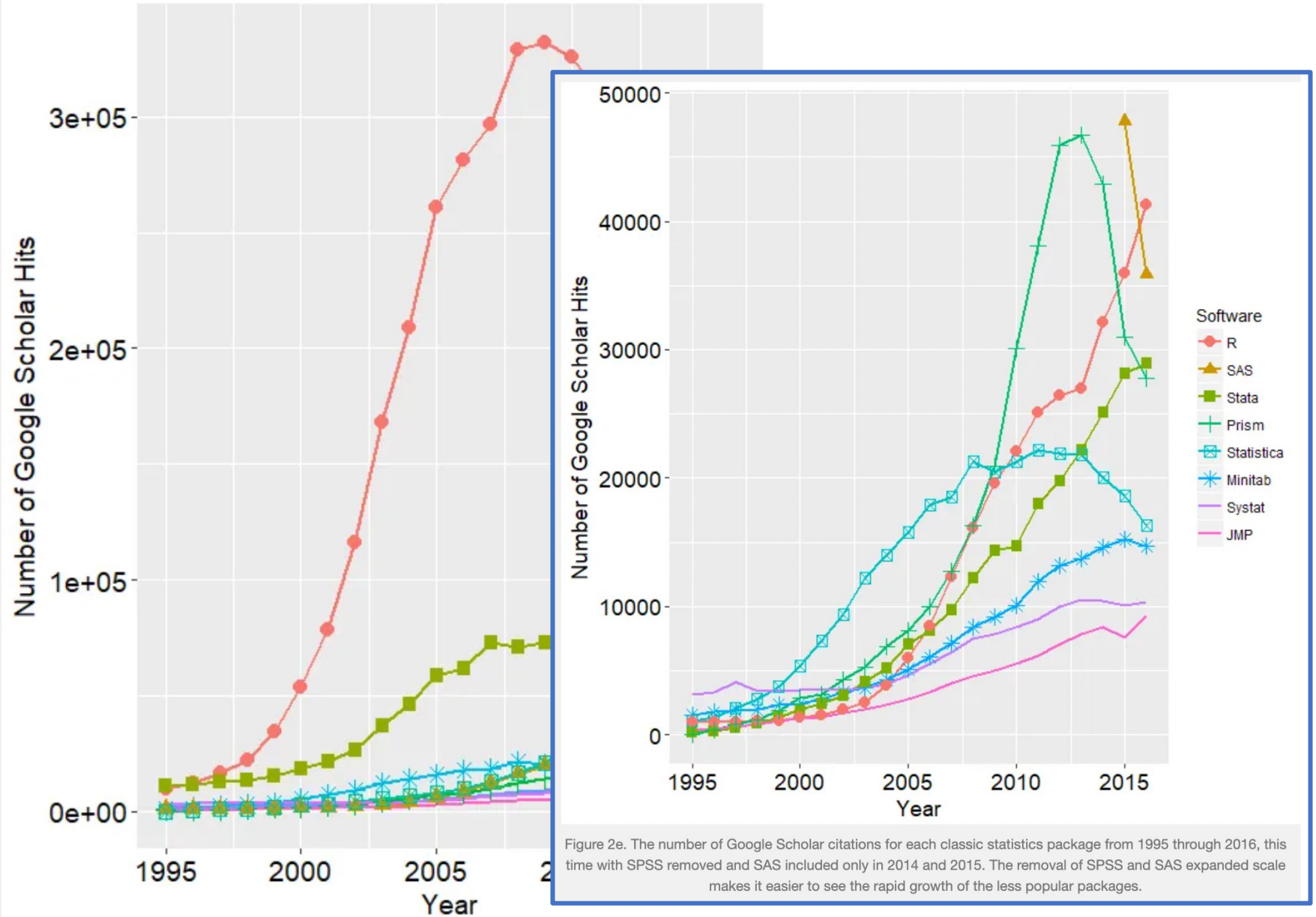


Figure 2e. The number of Google Scholar citations for each classic statistics package from 1995 through 2016, this time with SPSS removed and SAS included only in 2014 and 2015. The removal of SPSS and SAS expanded scale makes it easier to see the rapid growth of the less popular packages.

Figure 2d. The number of Google Scholar citations for each classic statistics package per year from 1995 through 2016.

[http://r4stats.com/articles/popularity/]

Best Practices: general concepts

- PLAN: in most cases, statistical approach can be selected prior to looking at the data
  - vs. Exploratory Analyses
- ORGANIZE
- EXECUTE: run code
- DOCUMENT: keeping track of what you did and what you thought (similar to a “lab notebook”)
  - There’s no such thing as too much documentation

Best Practices: general concepts

- Why document?

  So you (or someone else) can REPLICATE!

- How?
  - EXCEL vs. [STATA, R, SAS, SPSS, etc.]

Best Practices: general concepts

- Goal: if you start out with the same dataset, you can re-run your analyses (including all 'pre-processing') to arrive at the same results

- To do this, we will always use scripts (more later. . .)

Big Picture: steps for data analysis

- Cleaning data (tidying)
  - Importing data: did it import correctly?
  - Ranges: are the numbers what you'd expect?
  - N: is the sample size what you'd expect?
  - Variables and data labels
- Performing analysis
- Presenting findings
  - Picking out what pieces of your results to present
  - Deciding how best to present them
- Saving your work
  - Archiving (e.g., scripts, html logs)

Best Practices: Significant Figures

- Significant figures (digits)
  - Don't over-represent the amount of precision

- Which digits give you information?
  - 0.00700 (3 sig figs)
  - 0.052 (2 sig figs)
  - 370. (3 sig figs)
  - 10.0 (3 sig figs)
  - 705.001 (6 sig figs)
  - 37,000 (2 sig figs? 5 sig figs: 37,000.)

Best Practices: Significant Figures

[please review on your own]

- Rules:
  - All nonzero digits are significant
    - 1.234 (4 sf); 1.2 (2 sf)
  - Zeros between nonzero digits are significant
    - 1002 (4 sf); 3.07 (3 sf)
  - Leading zeros to the left of the first nonzero digits are not significant
    - 0.001 (1 sf); 0.012 (2 sf)
  - Trailing zeros that are to the right of a decimal point are significant
    - 0.0234 (3 sf); 0.20 (2 sf)

Best Practices: Significant Figures

[please review on your own]

- More Rules:
  - When a number ends in zeros that are not to the right of a decimal point, zeros are not necessarily significant
    - 190 (2 sf); 190. (3 sf)
  - Scientific notation:
    - 5.06×10^4 (3 sf)
    - 5.060×10^4 (4 sf)
    - 5.0600×10^4 (5 sf)

Best Practices: Rounding

[please review on your own]

- What about within calculations?
  - Accuracy of a calculated result is limited by the least accurate measurement
    - 100 (3 sf) + 23.643 (5 sf) = 123.643 → 124 (3 sf)
    - 3.0 (2 sf) x 12.60 (4 sf) = 37.8000 → 38 (2 sf)

- Within a long calculation?
  - Carry as many digits as possible through the entire set of calculations; then round the final result as appropriate
    - (5.00 / 1.235) + 3.000 + (6.35 / 4.0)=4.04858... + 3.000 + 1.5875=8.630829... → 8.3 (2 sf)

Best Practices: Significant Figures

[please review on your own]

- What about rounding at the end?
  - Round up if digit dropped >5
    - 12.6 → 13
  - Round down if digit dropped <5
    - 12.4 → 12
  - If digit dropped = 5, use any additional non-zero digits
    - 12.51 → 13
  - If digit dropped = 5, and no additional non-zero digits, round up if odd, down if even
    - 11.5 → 12
    - 12.5 → 12

Precision in reporting vs. analysis

- Over-reporting precision (AKA “false” or “spurious” precision) can be misleading
  - Indicates that you are able to measure something more precisely than is actually possible.
    - E.g., Height: 5.56 inches
- Over-reporting precision can hide the pattern or relationship you are trying to present

In which example is the relationship most apparent?

A. The number of women teachers in training rose from 29,942 to 94,347, and that of men from 13,410 to 36,051.

B. The number of women teachers in training rose from 29,900 to 94,300, and that of men from 13,400 to 36,100.

C. The number of women teachers in training rose from 30,000 to 94,000, and that of men from 13,000 to 39,000.

(Ehrenberg 1981)

Significant Figures: some concluding thoughts

- Just because software outputs lots of numbers, this does not mean that they are significant or important!

- During analysis, don't lose information; during presentation, consider how best you can convey your message and what is appropriate.
  - In some cases, graphical approaches may convey information and trends better than tables (more to come. . .)

Set Up

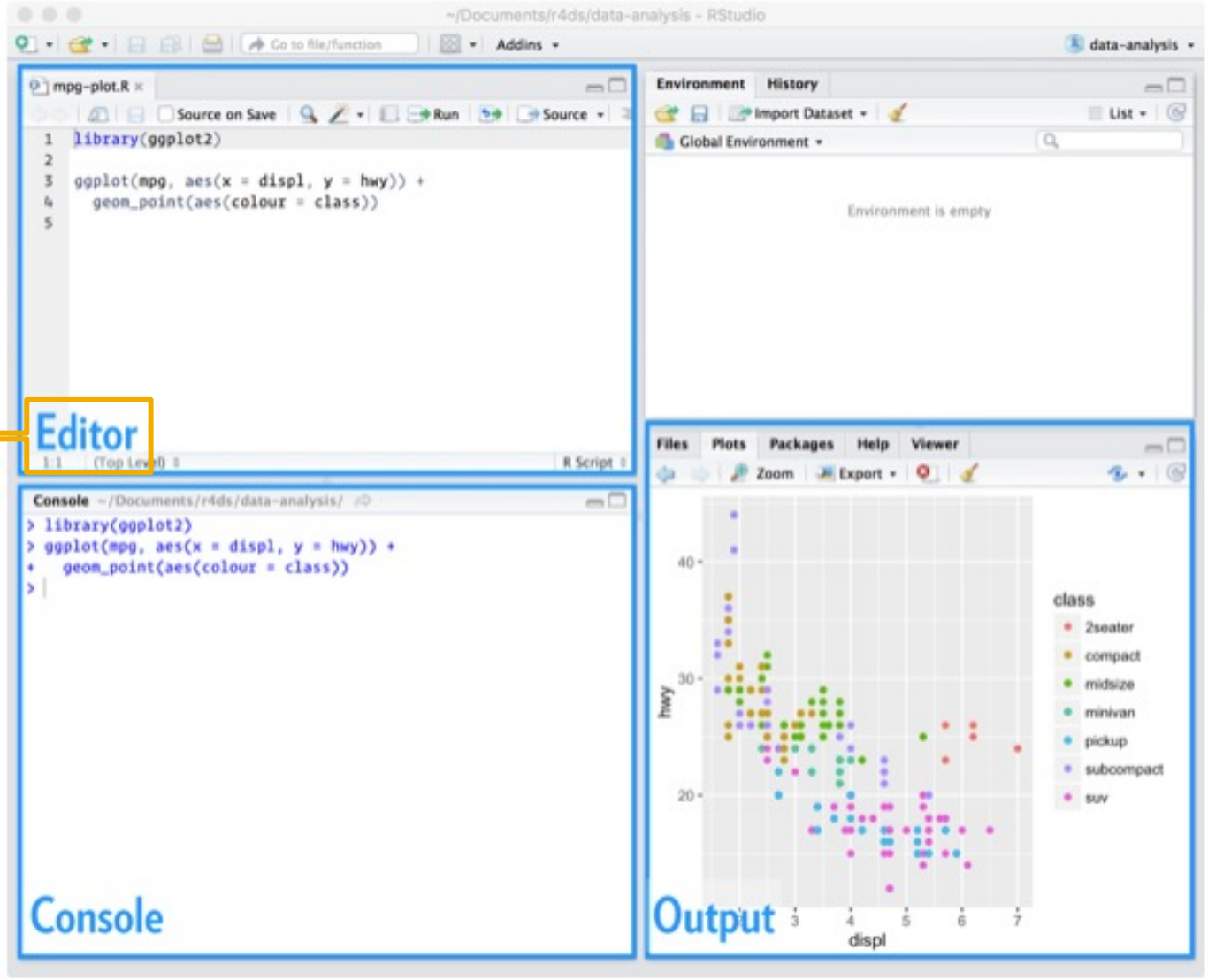
- Install R (cran.r-project.org)
- Install RStudio IDE (www.rstudio.com/download/)

Per my email: I am assuming that all of you have R and R studio installed.

- If not, we will help you at the end of class or will schedule a time to work with you 1:1 to get it running.

Orientation to R Studio

this is where your scripts go



https://r4ds.had.co.nz/workflow-scripts.html

A bit about scripts

- Using scripts (vs. console) keeps a record of everything you've done.
  - Why is this important?
    - If you make a mistake, you can easily correct it and re-run your code
    - If you don't finish a project in one sitting, you can pick up where you left off
    - Ensures reproducibility

Notes in Scripts

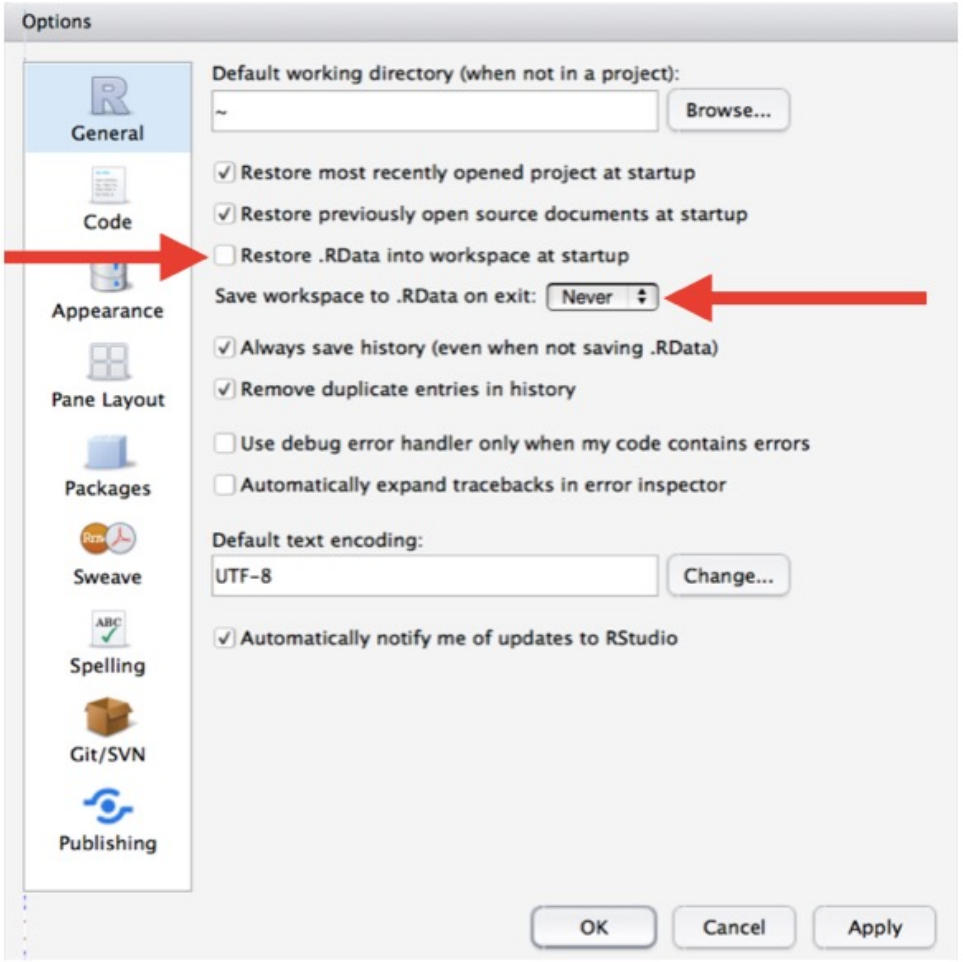
- In your script, you will want more than just commands.
  - You will make decisions along the way that you will want to remember
  - If someone else picks up your code, they will understand what you are doing
  - If you pick up your code later, you will understand what you did

Dear past-Hadley: PLEASE COMMENT YOUR CODE BETTER. Love present-Hadley
— @hadleywickham tweet

Your closest collaborator is you six months ago, but you don't reply to emails.
— @gonuke, after @kcranstn tweet quoting @mtholder

Good habits

- You want to get in the habit of relying on your script (not your environment); we will enforce this "good habit" in 2 ways:
  - To get full credit for your homework, you will create an HTML file that will only run if your script has no errors.
  - We suggest that you set your preferences in RStudio so that it does not preserve your workspace between sessions.



R Projects (1)

- Where do your R files live?
- R has a neat feature where you can set up Projects
  - A great alternative to setting working directory with facilitates collaboration and reproducibility
  - Each assignment can be considered its own Project so the files associated with this Project can be stored in one place
  - How?
    - File > New Project
    - New Directory > New Project > Empty Project [put it somewhere on your computer that makes sense]

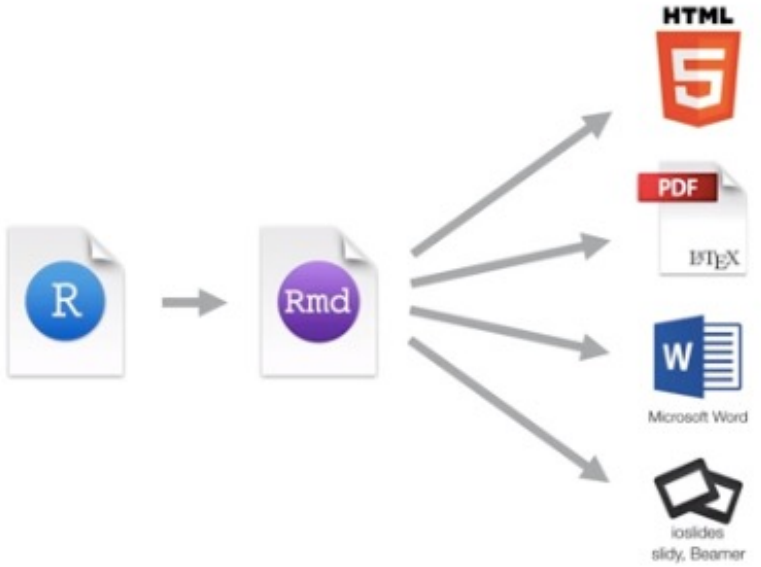
Markdown

- Markdown: language that uses plain text and formatting syntax that can be converted to a variety of formats (e.g., html, pdf, word)
- Advantages:
  - Allows for text/comments, code, and output (including plots) to be displayed in a single document
  - Presentable results can be easily updated if data changes

Encourages "*literate programming* and *reproducible research*"

markstat.bib

R Markdown



Required:

https://campus.datacamp.com/courses/reporting-with-r-markdown/

R Markdown

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>

<body>

  <h1>A header</h1>

  <p>A list with three items:</p>
  <ul style="list-style-type:circle">
    <li>Moe</li>
    <li>Larry</li>
    <li>Curly</li>
  </ul>

  <p>Some <strong>bold</strong> text.</p>

</body>
```

Code required in html language

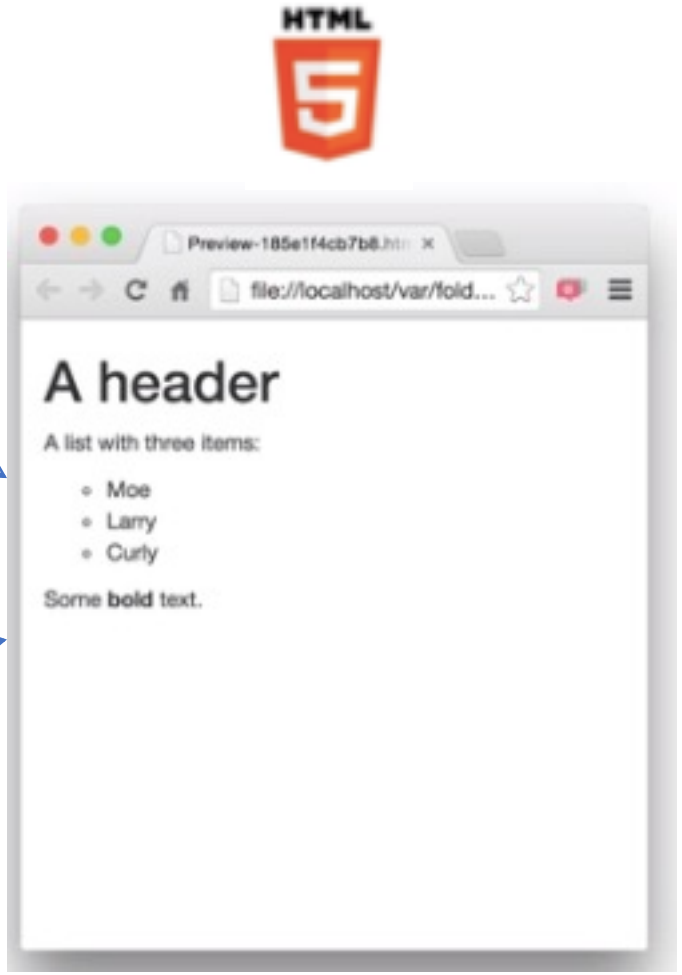
```
# A header

A list with three items:

* Moe
* Larry
* Curly

Some **bold** text.
```

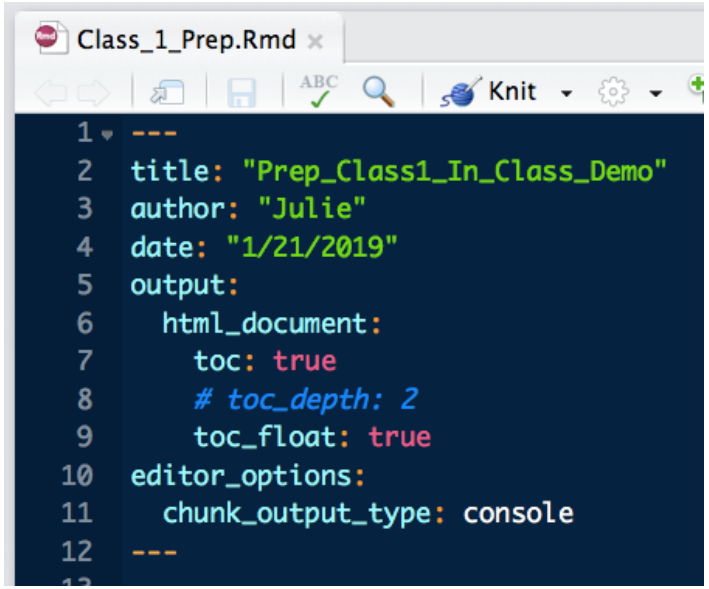
Code required using Markdown



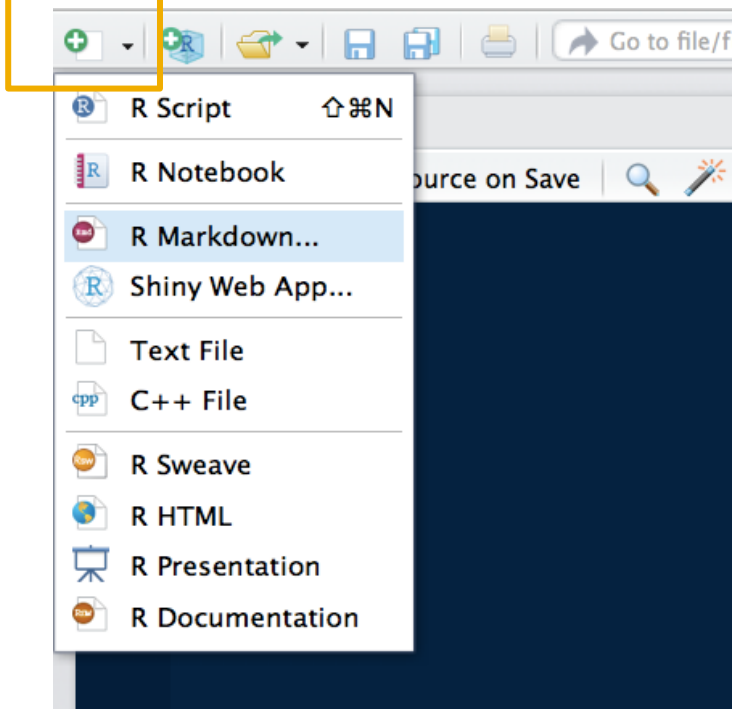
https://campus.datacamp.com/courses/reporting-with-r-markdown/

R Markdown Scripts

- If you open a new R Markdown tab, a template will pop up.
- In the YAML template:
  - Change title to whatever make sense
  - Change the name and date
  - Change the output:



this is where you can open a new script or markdown tab



R Markdown Scripts

- Your homework assignments will be written in an R Markdown file. This will give you a template to work from.
  - You will need to download this template from courseworks, save it in the R Project that you create for this session and open it.
  - You will need to edit the template document to answer the questions
- You can edit these templates as you like (make them more fancy/less fancy).

Other R Markdown Scripts you will get:

- Demo file:
  - Each week, I will demonstrate some new commands in an R Markdown script file.
- Syntax Inventory file:
  - I will provide you with a starting document to work from; ideally, you would edit/add to it each week, including the new commands introduced.
    - From now until Spring break, you can do this as you would any other file (on your hard drive, etc).
    - Beginning after Spring break, you can start doing this in GitHub (optional, not mandatory)
    - You will need to submit the final inventory in GitHub so it is good to get used to it.

R Packages

- Packages = collection of functions, data & documentation
- Installing packages and loading contents:
    - Note: you need to be connected to the internet for the package to download from CRAN and to install on your computer

```
install.packages("package_name")

library(package_name)
```

R Packages (a warning!)

- Why use packages?
  - There are (as of 1/2019) 13,664 packages on CRAN
  - They make it easier to work independently and collaborate with others
- However, CRAN guarantees nothing except that R will load the package on your machine
  - There are (as of 1/2019) 13,664 packages on CRAN
  - Does not guarantee correct results
  - Does not guarantee that it will be updated
  - Does not guarantee that it will work well with other packages

[www.linkedin.com/learning/learning-the-r-tidyverse]

@hadleywickham

@tidyverse

Why use tidyverse (vs. ‘Base R’)

- tidyverse is a set of packages (an “Ecosystem”)
  - Born 2014 and is constantly evolving. . .
    - You can contribute, if you have ideas! #Tidyverse
  - Not a replacement for ‘Base R’
- Why use it?
  - building everything yourself is time consuming
  - assumes some base knowledge.
- tidyverse was developed by RStudio
- reputable
  - RStudio is likely to have sustained support
  - tidyverse is ‘validated’
  - is more readable by a human
- Covers:
  - Data importation
  - Data wrangling
  - Data visualization

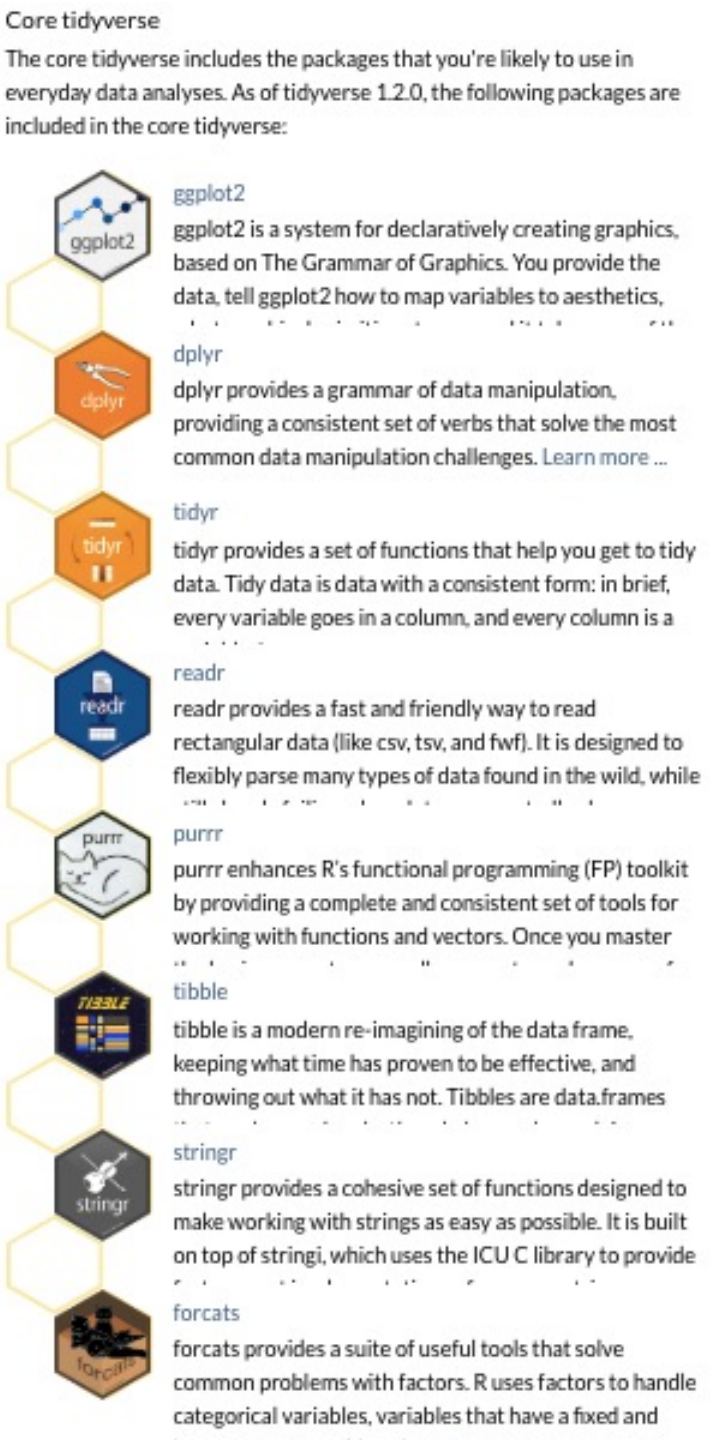
[www.linkedin.com/learning/learning-the-r-tidyverse]

Additional Packages

- There are many, many packages that can do the same thing. . .
  - There are many, many ways to do the same thing (this is especially true in R)
- Look for packages that work with tidyverse

Tidyverse

- There are 8 “core” packages in the tidyverse
- There are many (~70?) tidyverse-adjacent packages that are designed to work with the tidyverse but that you need to install separately
- To install:
  - `install.packages(“tidyverse”)`
    - Do this in your console (not script)
    - If it asks: “Do you want to install from sources the package which needs compilation?” ANSWER: y
  - `library(tidyverse)`
    - Do this in your script
    - It will load the “core” packages



Core tidyverse

The core tidyverse includes the packages that you're likely to use in everyday data analyses. As of tidyverse 1.2.0, the following packages are included in the core tidyverse:

ggplot2

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics,

dplyr

dplyr provides a grammar of data manipulation, providing a consistent set of verbs that solve the most common data manipulation challenges. Learn more ...

tidyr

tidyr provides a set of functions that help you get to tidy data. Tidy data is data with a consistent form: in brief, every variable goes in a column, and every column is a

readr

readr provides a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while

purrr

purrr enhances R's functional programming (FP) toolkit by providing a complete and consistent set of tools for working with functions and vectors. Once you master

tibble

tibble is a modern re-imagining of the data frame, keeping what time has proven to be effective, and throwing out what it has not. Tibbles are data.frames

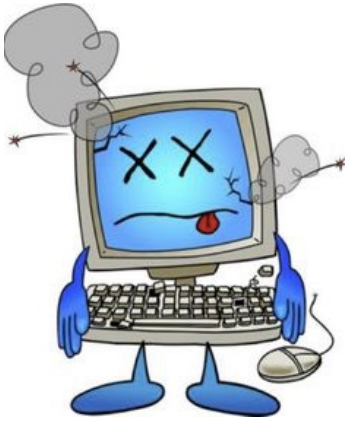
stringr

stringr provides a cohesive set of functions designed to make working with strings as easy as possible. It is built on top of stringi, which uses the ICU C library to provide

forcats

forcats provides a suite of useful tools that solve common problems with factors. R uses factors to handle categorical variables, variables that have a fixed and

Issue with Updates



- Right now (1/11/2021):
  - Base R
  - RStudio
  - Tidyverse
- All 3 will be updated pretty regularly; if you update, things may change.
  - We will plan NOT to update any of these 3 over the course of the class (unless we decide otherwise)
  - If your computer dies over the course of the semester (hopefully not), you will need to re-load (could be problematic but hopefully not)

Very basic coding concepts (1)

%>%

This is called a pipe; a pipe is used to improve coding “flow”

- Allows you to manipulate data
- Allows you to combine multiple operations in a single command (chaining commands together)

Getting help

- `?command`
- Google
  - Very helpful for error messages (not always intuitive)
- stackoverflow (http://stackoverflow.com)
- If you like reference books:
  - R for Data Science by Hadley Wickham and Garrett Grolemund
  - free download: http://r4ds.had.co.nz

Very basic coding concepts (2)

- object_name <- value

(read: “object name gets value”)

- Put spaces between everything

(otherwiseitishardtoread)

- Convention is good

(snake_case: object_names_separated_by_underscores)

- Order matters

(R will do first whatever you write first)

- Case matters

(EHS ≠ ehs)

Very basic coding concepts (3)

- Parentheses and quotations must come in pairs

(if not, R will not compute)

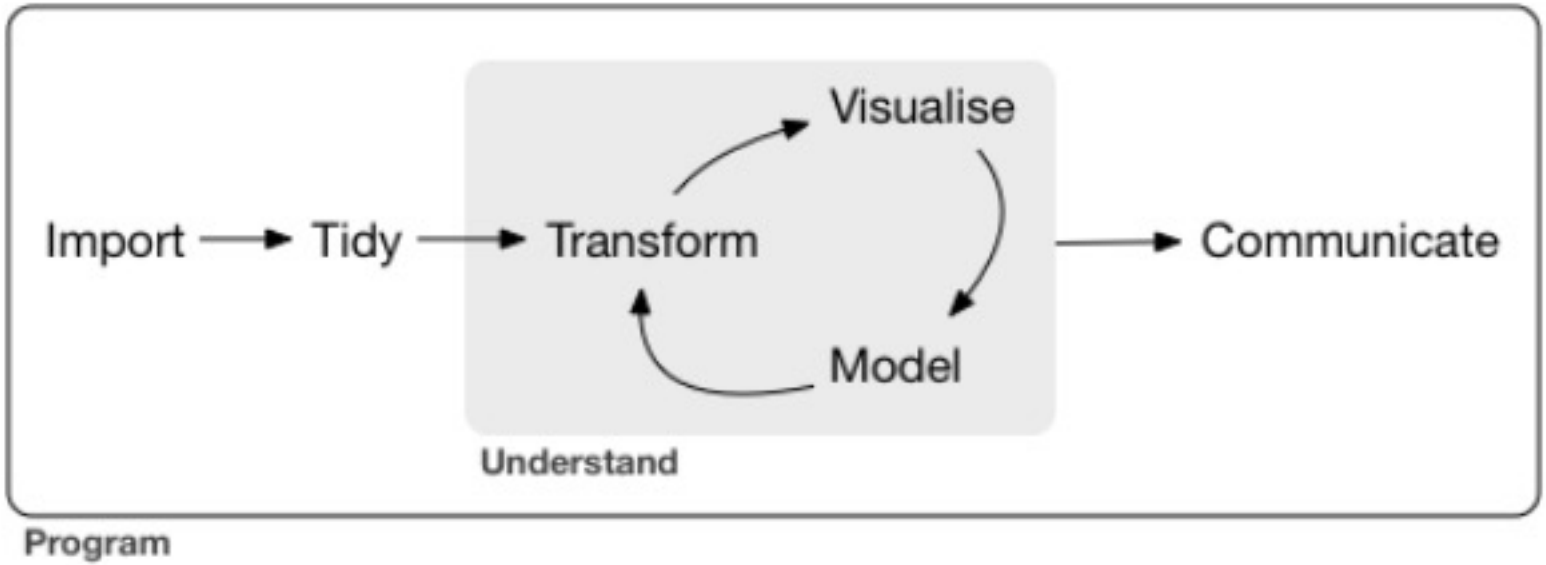
- For a list of keyboard shortcuts: Alt-Shift-K

- If you want to see your data in an excel-like format:

`View(<DATASET>)` will have it pop up in another tab next to your syntax

  - `View(NYCH14)`

How we'll spend our time



[Wickham and Grolemund 2018]

Looking at Data

- Before you can look at any data, you need to:
  - Import it
  - Install and load packages that give you tools to explore it
    - Or you can use default data that is loaded with packages (e.g., mtcars and diamonds)

Importing data (1)

- You need to tell your computer where to look for the data
  - Set up your Project
  - Or go to existing Project

- You need the data to be in a format that R can read it (more on this later)
  - CSV ("comma separated values") is a generic and easy format to work in (but `readr`, `readxl`, and `haven` will easily import all kinds of data formats)

```
dataname <- read.csv("data_name", header=TRUE)
```

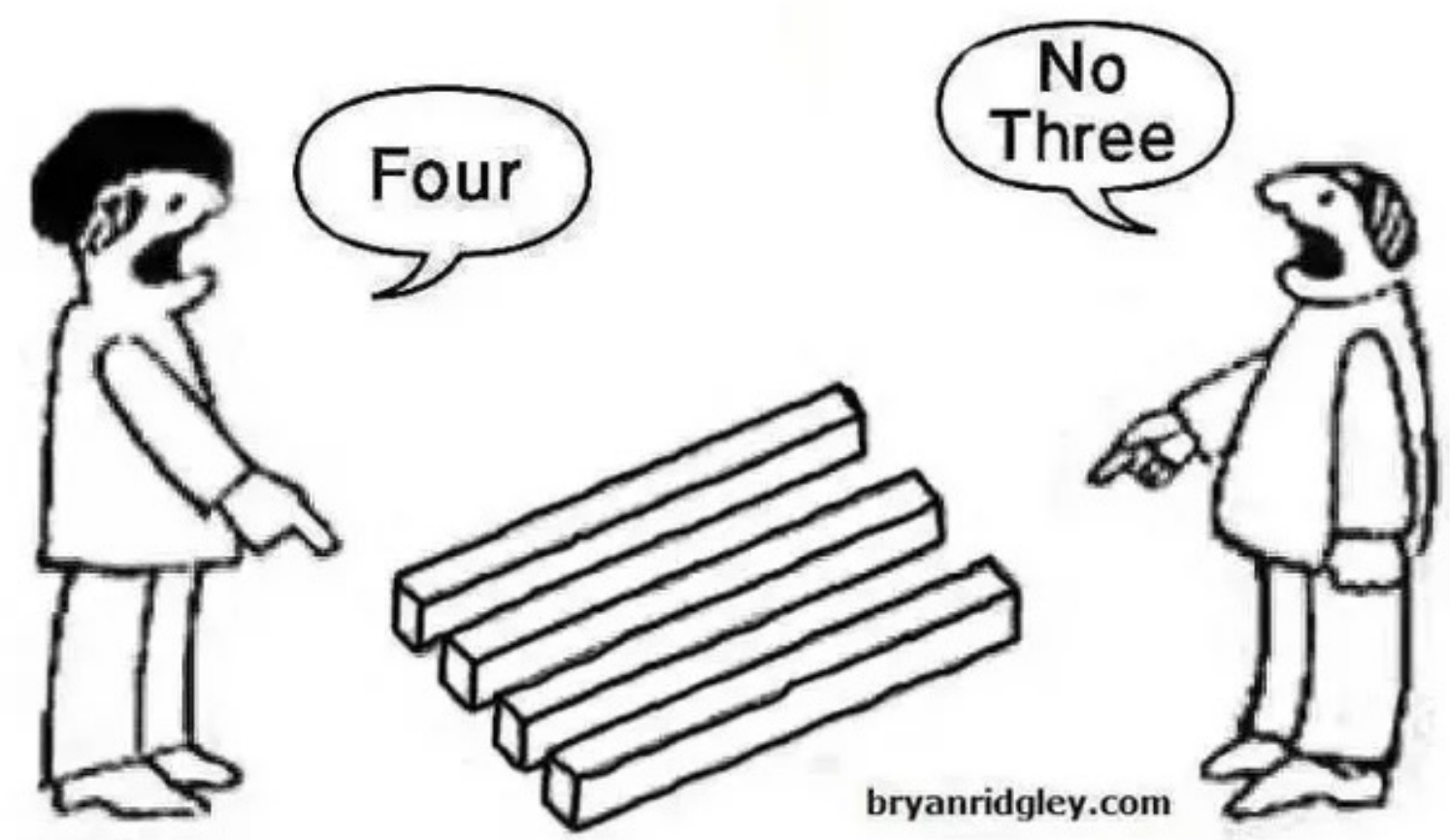
Importing data (2)

- Right now, let's focus on importing .csv (later, we will talk about ways to import data other than .csv)

```
NYCH14_ <- read.csv("NYCH14_sub.csv",
header=TRUE)
```

- You will typically download data from courseworks; you will want to save the data in the R Project that it is associated with. . .

Looking at data. . .

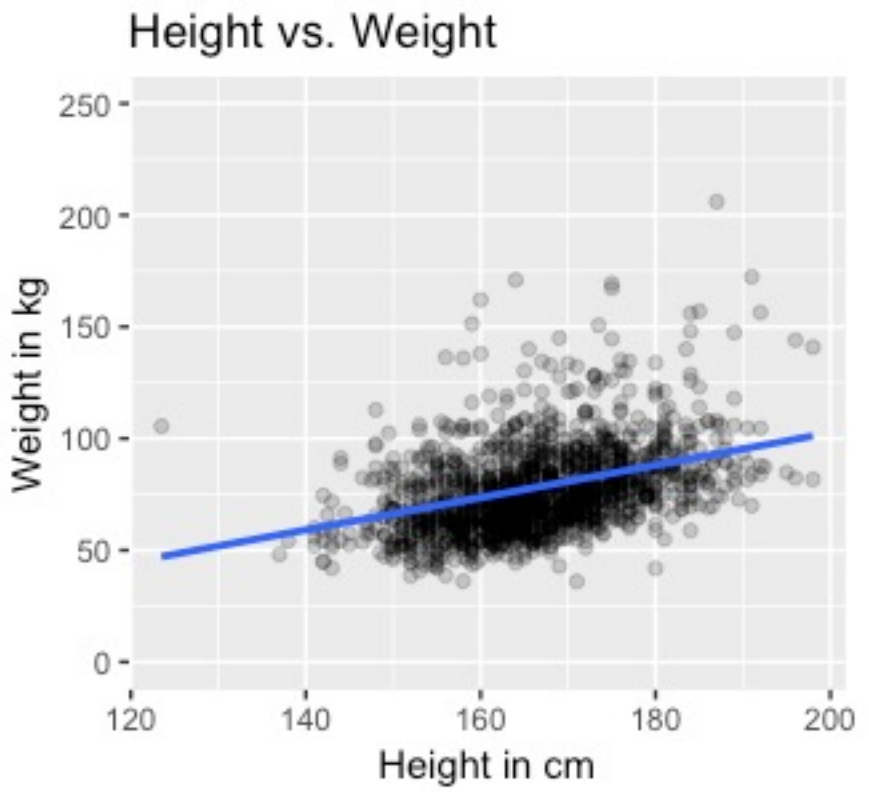


Using graphs to look at data

- It is often easier to see a pattern with a graph than with a table.

	A	B	J	K	L	M
1	KEY	GENDER	HEIGHTTRUE	ANTHRO2_W	BMI	BMI4CAT
2	133370A	Female	165	66	24.24	Normal
3	133370B	Male	167	69.3	24.85	Normal
4	133380A	Female	163	81	30.49	Obese
5	133380B	Male	166	77.9	28.27	Overwt
6	133390A	Female	159	71	28.08	Overwt
7	133410A	Male	173	121.3	40.53	Obese
8	133410B	Male	184	79	23.33	Normal
9	133430A	Female	159	69.4	27.45	Overwt
10	133450A	Female	163.5	77.7	29.07	Overwt
11	133460A	Female	153	56.9	24.31	Normal
12	133460B	Male	153.5	43.4	18.42	Underwt
13	133470A	Female	161	68	26.23	Overwt
14	133500B	Female	153	73.8	31.53	Obese
15	133570A	Female	159	66	26.11	Overwt
16	133570B	Male	175.5	90.2	29.29	Overwt
17	133580A	Male	164	58.4	21.71	Normal
18	133660A	Female	162.5	63.4	24.01	Normal
19	133660B	Female	177	59.9	19.12	Normal
20	133710A	Male	174	67.4	22.26	Normal
21	133720A	Male	166	65.2	23.66	Normal
22	133720B	Female	165	60.5	22.22	Normal
23	133750A	Female	172	61.2	20.69	Normal
24	133770A	Male	179.5	104.6	32.46	Obese
25	133830A	Female	158	61.8	24.76	Normal
26	133890A	Male	171	84.6	28.93	Overwt
27	134010A	Female	167	59	21.16	Normal
28	134010B	Female	164	62.3	23.16	Normal
29	134040A	Male	170	66	22.84	Normal
30	134040B	Female	164	57.9	21.53	Normal
31	134090A	Female	149	56.5	25.45	Overwt
32	134240A	Female	164	59.9	22.27	Normal
33	134240B	Male	178	73.4	23.17	Normal
34	134260A	Female	151	52.7	23.11	Normal
35	134260B	Female	152	44.8	19.39	Normal
36	134290A	Female	157.5	55.2	22.25	Normal

-vs-



Data Visualization (Data Viz)

"Data and information is now a tool for creating beautiful visuals. What used to be simple charts and scatter plots are now complex and creative pieces of data art—some beautiful enough to hang up on your wall."

Data Visualization = Communicating with Data

We will do this but initially, we just want to take a look. . .

https://visme.co/blog/best-data-visualizations/

We will use ggplot2 for graphing

- gg: grammar of graphics
  - build it up (vs. starting from the ‘result’)
- Built in layer(s):
  - data
  - aesthetic mapping (controls x, y, and scale)
  - geom: visual marks that represent datapoints
  - stat: describes relationship between data and geom (e.g., bar = identity vs. bin)
    - only x: geom_bar()
    - x and y: geom_bar(stat = “identity”)
  - position: describes relationship between data and mapping (e.g., put 30 at 30)
- You can stack layers

Graphing in ggplot

- Minimum components of plots

  You need to tell ggplot:

  - What data to use (data = `<DATA>`)
  - What to plot: `<GEOM_FUNCTION>( )`

    There are many geoms, which to use depends on what you are trying to visualize:

    - geom_point [this is a scatterplot]
    - geom_smooth [this is a smoothed line graph]
    - geom_bar [this is a bar graph]
    - geom_box [this is a box plot]
    - geom_histogram [this is a histogram]

ggplot template:

```
ggplot(data = <DATA>) +
     <GEOM_FUNCTION>(
          mapping = aes(<MAPPINGS>),
          stat = <STAT>,
          position = <POSITION>) +
     <COORDINATE_FUNCTION> +
     <FACET_FUNCTION>
```

Required inputs
Not required; ggplot has defaults

What kind of graph should I use?

- Depends on what kind of data you have:
  - Discrete/Categorical vs. Continuous
- Depends on how many factors you want to compare at once:
  - 1 factor: distribution of age
  - 2 factors: distribution of age by sex
  - 3 factors: distribution of age by sex by country
  - ETC

Let's take a break and then look at this using ggplot

Helpful resource: http://www.sthda.com/english/wiki/be-awesome-in-ggplot2-a-practical-guide-to-be-highly-effective-r-software-and-data-visualization#geom_point-scatter-plot

About the data. . .NYC HANES 2014

NYC HANES

New York City Health and Nutrition Examination Survey

ABOUT | DATA | RESEARCH OPPORTUNITIES | PUBLICATIONS

In 2004 and 2013, NYC HANES measured the health of adults living throughout New York City. With data and bio-specimens for more than 3,500 New Yorkers across the two time periods, we are now pleased to offer these data to researchers around the world.

Since the 2004 survey was completed, more than 20 academic articles have been published. Now we invite researchers in public health, medicine and other fields to download the most recent dataset or submit a proposal to receive specimens for your own research. Read More »

http://nychanes.org/