A-Lab Tutorial
Data Wrangling and Visualization in R

Sebastian Steffen

ssteffen@mit.edu

September 26, 2021

Overview

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Workshop Outline

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.
- Understand key data concepts.

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.
- Understand key data concepts.
- Know vocabulary and libraries for better search queries.

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.
- Understand key data concepts.
- Know vocabulary and libraries for better search queries.
- Overview of the basics.

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.
- Understand key data concepts.
- Know vocabulary and libraries for better search queries.
- Overview of the basics.

- Ultimate Goal is to help you help yourself.

Teaching Learning Objectives (TLOs)

- Install and Set Up R and tidyverse.
- Understand key data concepts.
- Know vocabulary and libraries for better search queries.
- Overview of the basics.

- Ultimate Goal is to help you help yourself.
- Q&A.

Before We Begin..

Please take a few minutes to think about the data for your own project (alternatively, do this once you have access to it):

- Which questions do you have to answer? Which can you answer? Which do you want to answer? Why are they important? List a few.

Before We Begin..

Please take a few minutes to think about the data for your own project (alternatively, do this once you have access to it):

- Which questions do you have to answer? Which can you answer? Which do you want to answer? Why are they important? List a few.
- What would the ideal data look like? What variables would you need? What format does the data need to be in (i.e. what should each row look like)?

Before We Begin..

Please take a few minutes to think about the data for your own project (alternatively, do this once you have access to it):

- Which questions do you have to answer? Which can you answer? Which do you want to answer? Why are they important? List a few.
- What would the ideal data look like? What variables would you need? What format does the data need to be in (i.e. what should each row look like)?
- What are your hypotheses?

Before We Begin..

Please take a few minutes to think about the data for your own project (alternatively, do this once you have access to it):

- Which questions do you have to answer? Which can you answer? Which do you want to answer? Why are they important? List a few.
- What would the ideal data look like? What variables would you need? What format does the data need to be in (i.e. what should each row look like)?
- What are your hypotheses?
- What method do you need to apply?

Before We Begin..

Please take a few minutes to think about the data for your own project (alternatively, do this once you have access to it):

- Which questions do you have to answer? Which can you answer? Which do you want to answer? Why are they important? List a few.
- What would the ideal data look like? What variables would you need? What format does the data need to be in (i.e. what should each row look like)?
- What are your hypotheses?
- What method do you need to apply?
- Is there prior work (i.e. literature, news, . . .) that attempts to answer it that you can leverage?

Example

Want to understand the value of different skills. Long history in Economics on estimating the 'Returns to "Skill"'.

- How valuable is learning to code? What's the $-value associated with R?
- My Research: What skills do employers demand and how much are they willing to pay.
- Ideal data: Panel data of job postings annotated with skills demanded and employer. Each row is a bundle of skills and a $-value (wage).
- Hypothesis: Value of coding is high and has increased over time.
- Method: OLS Regression (for interpretability) with employer fixed effects

- Conversely, if I wanted to predict the future value of R, I'd instead need a time series of $-values for R and might use ARIMA.

Data Wrangling

Data Wrangling

Methods to transform raw data into usable a format to answer a specific research question.

- Different research questions require different data formats.

Workshop Outline

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Installation of R

- R: https://cloud.r-project.org/
- RStudio Desktop IDE: https://www.rstudio.com/products/rstudio/download/
- Install packages via 'Tools' or via `install.packages(<package_name>)`:
  - tidyverse
    - ★ includes several extremely useful packages: tidyr, dplyr, purrr, stringr, forcats, lubridate, ggplot2
  - ...
  - mice, tidytext, philentropy, lfe, rpart, causaltree, glmnet, fuzzywuzzy, shinyr, nnet, neuralnet, tensorflow, sf, ...

Project Environment Tips

- Keep project folder organized (i.e. via Gentzkow, Shapiro (2014)).

Project Environment Tips

- Keep project folder organized (i.e. via Gentzkow, Shapiro (2014)).
- Never change the raw data - always work on a copy!

Project Environment Tips

- Keep project folder organized (i.e. via Gentzkow, Shapiro (2014)).
- Never change the raw data - always work on a copy!
- Try to use functional programming for any task you do more than twice. (a bit of a learning curve with dplyr
  →hrefhttps://dplyr.tidyverse.org/articles/programming.htmlquosures)

Project Environment Tips

- Keep project folder organized (i.e. via Gentzkow, Shapiro (2014)).
- Never change the raw data - always work on a copy!
- Try to use functional programming for any task you do more than twice. (a bit of a learning curve with dplyr
  →hrefhttps://dplyr.tidyverse.org/articles/programming.htmlquosures)
- Use (Jupyter) Notebooks to explore data. Later, move to automating stuff via command line and `screen` (or batch files, i.e. on Sloan's Engaging Server).

Workshop Outline

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Tidy Data

- **Every column is a variable.**

NOT

brand_model	year	mileage
Audi_A4	1999	18
Audi_A4	2008	20
VW_GTI	1999	19
VW_GTI	2008	22
...	...	...

- Can be fixed with `separate()`.

Tidy Data

- **Every column is a variable.**
- **Every row is an observation.**

NOT

brand	model	mileage_1999	mileage_2008
Audi	A4	18	20
VW	GTI	19	22
...	...	...	

- Can be fixed with `pivot_longer()`, `pivot_wider()` (formerly `gather()`, `spread()`).

Tidy Data

- **Every column is a variable.**
- **Every row is an observation.**
- **Every cell is a single value.**

brand	model	mileages_1999_2008
Audi	A4	(18, 20)
VW	GTI	(19 , 22)
...	...	...

- Can be fixed with `separate()` and `pivot_longer()`.

Tidy Data

- **Every column is a variable.**
- **Every row is an observation.**
- **Every cell is a single value.**

brand	**model**	**year**	mileage	...
Audi	A4	1999	18	...
Audi	A4	2008	20	...
VW	GTI	1999	19	...
VW	GTI	2008	22	...
VW	GTI	2012	24	...
Mazda	CX-5	2017	23	...
...	...	...	...	...

- Tidy data: Each row has a **unique key (or index)** and **values**.
- Here, the key/index is **brand x model x year**, values are mileage.

Tidy Data - Easier Said than Done

- Example I: Which car has the lowest fuel consumption?

Tidy Data - Easier Said than Done

- Example I: Which car has the lowest fuel consumption?
- Example II: Which car brand is the most innovative (in terms of updated models)?

Tidy Data - Easier Said than Done

- Example I: Which car has the lowest fuel consumption?
- Example II: Which car brand is the most innovative (in terms of updated models)?
- Example III: Predict average fuel consumption in 2016.

Tidy Data - Easier Said than Done

- Example I: Which car has the lowest fuel consumption?
- Example II: Which car brand is the most innovative (in terms of updated models)?
- Example III: Predict average fuel consumption in 2016.

- Definition of clean data depends on the question(s). $\rightarrow$ Need to know how to transform one data format into another.

Workshop Outline

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Chaining with Pipes

- Pipes let you read code from left to right by chaining commands together. $\rightarrow$ More legible code.

Without Chaining	With Chaining
f(x)	$x \rightarrow f$
f(x, y)	$x \rightarrow f(y)$
f(g(x))	$x \rightarrow f \rightarrow g$

Basic Operations

- `filter`: filter data based on logical condition.
- `mutate`: create new (and change old) variables.
- `group_by`, `ungroup()`: define (temporary) keys.
- `summarize`: Used with `group_by` to aggregate columns. `arrange`: reorder rows (i.e. sort). `select`: subset (and reorder) columns.

Quick Example

Common Operations

- Log: $log(x + 1)$
- %-Change over time: $group\_by(Group)\% > \%arrange(-T)\% > \%mutate(Change = last(X)/first(X) - 1$
-

More Advanced Words

- `across` (supersedes `mutate_at` and `mutate_if`).
- `pivot_longer`, `pivot_wider`: Make dataset longer or wider (i.e. from time series to panel or vice versa).

Types of Data

- Cross-Sectional Data: Many subjects, one point in time.
- Time Series: One subject, many points in time.
- Panel Data: Many subjects, many points in time. May be unbalanced, i.e. some (subject, time) combinations are missing.
- Multidimensional Panel Data: More than 2 dimensions, i.e. (subject, time, group)

Long versus Wide Data

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

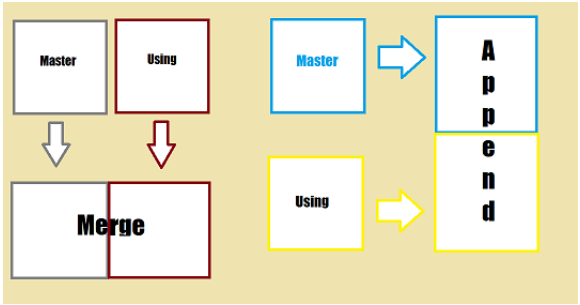
- The first allows analyses at the id (1, 2) level, the second at the id, key level.

Aggregation

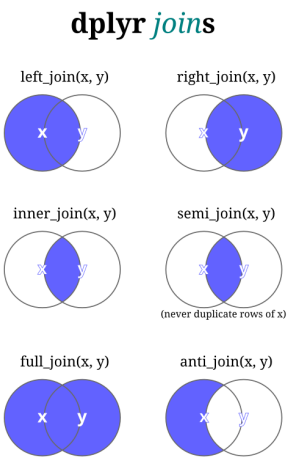
- Groups/Reduces multiple rows into one:
- `df <- cars %>% group_by(brand, year) %>% summarize(num_models = n(), mean_mpg = mean(mpg, na.rm = TRUE)`
- Common Aggregation functions: `mean()`, `sd()`, `first()`, `last()`, `sum()`, `min()`, `max()`, `n()`

Joins/Merges and Appends

- Append: Requires same column names (`bind_rows()`, `bind_cols()`
- Join/Merge: Requires key to identify matching rows.
  - Left Outer Join: Keep all (including unmatched) rows from left. (`left_join`).
  - Inner Join: Only keep matching rows (`inner_join()`).
  - Anti Join: Only keep unmatched rows from left (`anti_join()`).
  - Full Join: Keep all rows from both left and right (`full_join()`).



Join Types Visualized



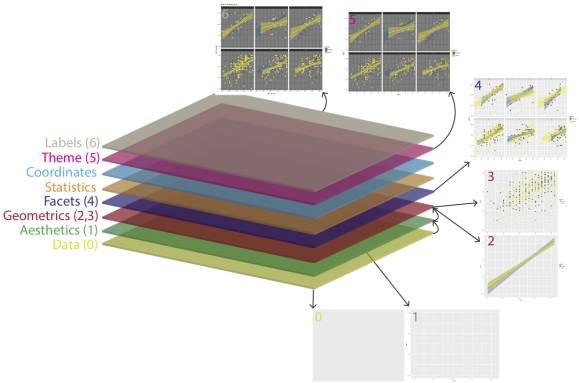
Package Overview

Type	R	Python
Missing Values	`mice`	`fancyimpute, statsmodels`
Data Wrangling	`dplyr`	`pandas`
Strings	`stringr`	built-in, `pandas`
(Fuzzy) String Matching	`fuzzywuzzyR`	`fuzzywuzzy`
Dates	`lubridate`	built-in, `pandas`
Geographical	`ggmap, maps, maptools`	`geopandas`
Visualization	`ggplot2, bokeh`	`plotnine, bokeh, matplotlib`
Regressions (with p-values)	`lm, lfe`	`statsmodels`
Machine Learning	`caret, glmnet, rpart`	`scikit-learn`
Deep Learning	`ttensorflow, keras`	`tensorflow, keras, pytorch`

- All Python packages can also be accessed in R with the R package `reticulate` (may be slower than built-ins though).

Visualizations with ggplot2

- Plots are built layer-by-layer.
- Most important layer: the `geom layer`



Visualization Advice

- No double Y-axes $\rightarrow$ Very misleading.
- Maximize the 'Data-Ink-Ratio' $\rightarrow$ Keep it simple.
- use themes like `theme_bw()` to make figures consistent.
- thick/horizontal lines to highlight, no or muted tick lines
- Think about aspect ratio of your presentation (on Zoom it's usually 16:9, while in-person it's often 4:3) versus write-up. Often need to make axes titles larger than defaults.

Tips & Tricks I

- Start early!!!

- Don't oversell your data.
- Communicate well with your team, your mentor, your company.
- Robustness checks, Validity tests, Simplicity (start with a baseline).
- Clean, consistent figures and formatting.
- Practice for your final presentation.

Tips & Tricks II

- Use flags in the header of your script to set constant variables, i.e. path/file names, etc.
- Keep a diary of what you've tried. Spending < 2 Minutes per day on this is already enough and really helps.
- Read the documentation.
- Use github for version control (especially good with team mates).
- Proper motivation and embedding in existing literature or context for final write-up.
- Cite references (use citation management like Zotero or Mendeley).

Workshop Outline

1. Teaching Learning Objectives (TLOs)
2. Boring Stuff
3. Concepts
4. R and Tidyverse Crashcourse
5. Q & A

Q & A

- These were just very general points, vocab, and code snippets to get you started.
- While there's a lot to learn and do, don't feel overwhelmed. A lot of the material here may not apply to your specific project.
- There are incredible online resources - use them!
- You can always talk to you mentors!

Thank You!

For feedback, questions, or comments,
please email me: ssteffen@mit.edu
sebastiansteffen.com