Thursday's class

Mick McQuaid

python vs r

plenty of discussion online

datacamp

ease of learning, python

Python is widely thought to be easier to learn than R because of its simple syntax: no braces, indentation enforced

ease of learning, r

Most people learn one of two flavors of R

- base R with `data.table` (more difficult, terse, powerful for data wrangling)
- tidyverse R (what I've been teaching you, better for visualization)

general data wrangling

Two kinds

- formal (corporate-enforced standards like Pentaho)
- informal (roll your own, individualistic)

Focus on informal

- you won't have a choice about formal anyway
- you need daily practice with informal, like doing pushups
- informal data wrangling is the most time-consuming part of data science, so important to tackle

Focus on regular expressions

- You need regular expressions for informal data wrangling
- Most regular expressions are PCRE (perl compatible)
- Bible remains Friedl, 1997, *Mastering Regular Expressions*, because regular expressions don't change much from year to year
- That durability makes them more valuable to you, like SQL

Focus on one of three data wrangling approaches

Get really good at one and maintain a passing familiarity with the other two

- Python (easiest)
- R, using tidyverse (middle)
- R, using data.table (hardest)

Data visualization

- Hadley Wickham is the most original developer
- Hence R gets the new ideas first
- Python gets them eventually
- I recommend you do most in ggplot with ggthemes
- Develop your identity as a visualizer

Machine learning

- For statistical machine learning, use R
- For deep learning, use Python
- But both work for both
- R gets statistical ideas first
- Python gets neural network ideas first

Production systems

- Neither one
- C, Rust, or C++ are all better
- But prefer Python over R for production

Notebooks

- available for both R and Python
- I prefer R Markdown over Jupyter Notebooks
- I find R Markdown more flexible than Jupyter Notebooks
- R Studio makes a big difference compared to Jupyter Notebooks

graphic usage

which graphic should you use?

- ggplot is silent on the issue
- tries to make up for it with the ggplot cheatsheet
- pause to look at the ggplot cheatsheet

Jamal's diagram

(pause for Jamal's diagram)

(take a screen shot if he doesn't mind)

useful functions

my favorite

```
library(tidyverse)
df %>%
    select(Borough)           %>%
    group_by(Borough)         %>%
    dplyr::mutate(count=n()) %>%
    unique()                  %>%
    arrange(desc(count))
```

preceding function

equivalent to the following `bash` pipeline

```
cut -f 3 -d , df | sort | uniq -c | sort -nr
```

but more verbose, easier to remember?

exercise

- Make a function out of preceding
- Put it in `.Rprofile`
- Don’t have to remember it

function

```
uniqdashc <- function(x,y) {
    x %>%
        select({{y}}) %>%
        group_by({{y}}) %>%
        dplyr::mutate(count=n()) %>%
        unique() %>%
        arrange(desc(count))
}
uniqdashc(df,Borough)
```

a more compact way

```
uniqdashc <- function(x,y) x %>% count({{y}})
```

after this course

survey

How was the statistics module? Tell me the top three good points and top three bad points about it.

general advice

- Don’t discount SQL (fastest way to massage production data)
- Use pgexercises to brush up on SQL
- Read publications like KDNuggets (but take them with a grain of salt)
- Follow Hadley Wickham
- Follow conferences like NeurIPS, KDD
- Work on storytelling (should have done more in this course)
- Never stop learning