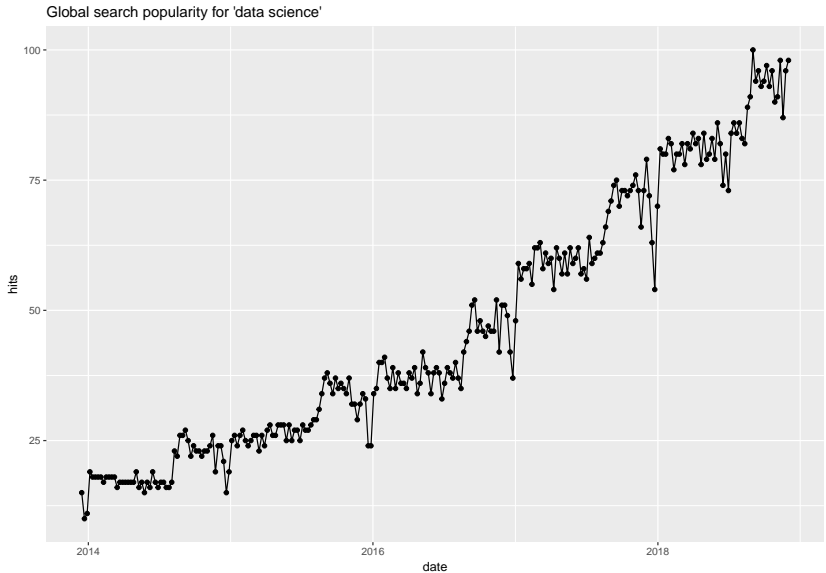


# Introduction to Computational Tools and Techniques in Social Science

Jae Yeon Kim

10 December, 2018

# Motivation



- ▶ Why should we care?
- ▶ Yes, big data (or machine learning) is a trend.
- ▶ But computational tools and techniques are much broader and fundamental to how we research:
  - ▶ Data collection (e.g., APIs, webscraping)
  - ▶ Analysis (e.g., text analysis, machine learning)
  - ▶ Visualization (e.g., maps, social networks)

- ▶ Using Excel:
  - ▶ 3 mins for copying, pasting, and reorganizing one article
  - ▶ 80,000 newspaper articles
  - ▶ Taking **4,000** hours or **166 days**

- ▶ Using python:
  - ▶ A few hours for coding
  - ▶ Less than 5 mins for creating the dataset
  - ▶ Also, the code is reusable.

```
In [1]: def parsing_proquest(x):  
        # load libs  
        from bs4 import BeautifulSoup  
        import re  
        # load file  
        soup = BeautifulSoup(open(x,"r"), 'html.parser')  
        # filter by strong tag  
        doc = soup('strong')  
        # save filtered results to new objects  
        doc.text = soup.findAll(text=re.compile('Full text:'))  
        doc.date = soup.findAll(text=re.compile('Publication year:'))  
        doc.source = soup.findAll(text=re.compile('Publication date:'))  
        doc.author = soup.findAll(text=re.compile('Publication info:'))
```

- ▶ Yet it takes some **efforts** to take advantages of these new tools.
  - ▶ You need to learn how to code **a little bit**.
  - ▶ However, learning on your own is inefficient.
  - ▶ More important, you can get **bad** habits.

- ▶ The following examples are adapted from <https://style.tidyverse.org>

*# Good*

```
fit_models.R
```

```
if (y < 0 && debug) {  
  message("y is negative")  
}
```

*# Bad*

```
fit models.R
```

```
if (y < 0 && debug)  
message("Y is negative")
```

*# Good*

```
do_something_very_complicated(  
    something = "that",  
    requires = many,  
    arguments = "some of which may be long"  
)
```

*# Bad*

```
do_something_very_complicated("that", requires, many, arguments,  
                                "some of which may be long"  
                                )
```



- ▶ Three commandments
  - ▶ Thou shall comment.
  - ▶ Thou shall reuse functions (no copy and paste).
  - ▶ Thou shall practice version control (no final\_final\_final.Rmd).

- ▶ Coding is similar to **cooking**.
  - ▶ So many different cuisines (programming languages).
  - ▶ But there are fundamentals.
    - ▶ Ingredients (data)
    - ▶ Techniques (logic)
    - ▶ Recipes (workflow)

# Objectives

- ▶ Tasting a wide range of computational tools
- ▶ Getting programming fundamentals right
  - ▶ Concepts
  - ▶ Techniques
- ▶ Learning by doing
  - ▶ Learning from your own MANY trials and errors
  - ▶ Learning from others (please, do Google search before asking me)

▶ **Don't expect**

- ▶ Becoming a data scientist within one semester
- ▶ I can answer all of your questions.

▶ We focus on learning **how to learn**.

- ▶ Programming is one endless Google Search (aka “Rochelle’s Law”)

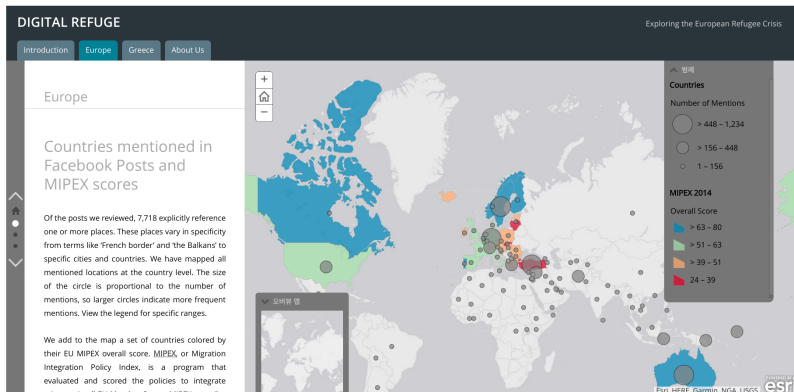
# Syllabus

- ▶ Introduction
  - ▶ Jan. 16: Introduction and Setup (“Installfest”)
- ▶ Version control
  - ▶ Jan. 21/23: Unix, Bash, and Git
- ▶ R and Python fundamentals
  - ▶ Jan. 28/30: Data Structure in R
  - ▶ Feb. 4/6: Data Analysis in R
  - ▶ Feb. 11/13: Data Visualization in R
  - ▶ Feb. 18/20: Intro to Python

- ▶ Online data collection
  - ▶ Feb. 25/27: Webscraping (project proposal draft due)
  - ▶ Mar. 4/6: APIs
  - ▶ Mar. 11/13: HTML/CSS/Javascript
  - ▶ Mar. 18/20: Online Sampling, Survey, and Field Experiments
- ▶ Mar. 25/27: SPRING BREAK (final project proposal due)

- ▶ Machine Learning and text analysis
  - ▶ Apr. 1/3: Text Analysis in R (guest lecture by Marla Stuart, Social Work & BIDS Data Science Fellow)
  - ▶ Apr. 8/10: Machine Learning and Text Analysis
  - ▶ Apr. 15/17: Supervised Machine Learning in R (guest lecture by Chris Kennedy, Biostats & BIDS Data Science Fellow)
- ▶ Review
  - ▶ Apr. 22/24: Wrap-up: Frontiers in Computational Methods (project demo)

# Previous final projects by students





# Class

- ▶ Participation (25%)
  - ▶ Be nice to each other. We're all learning (especially me).
- ▶ Homework (50%)
  - ▶ Every week.
  - ▶ Learning how to code is like learning how to drive.
- ▶ Final project (25%)
  - ▶ Feasibility is your friend. Late Feb proposal, April presentations.

# Logistics

- ▶ Learning by doing
  - ▶ Pair-programming on in-class challenges
- ▶ Section is required
- ▶ Julia Christensen is a technical assistant to the course.

## Special thanks

- ▶ Rochelle Terman (Chicago)
- ▶ Rachel Bernhard (Oxford, UC Davis)