Software Engineering for Data Scientists

Course Introduction

David Beck[1,2], Joseph Hellerstein[1,3], Jake VanderPlas[1,4]

[1]eScience Institute

[2]Chemical Engineering

[3]Computer Science Engineering

[4]Astronomy

University of Washington

March 13, 2017

UNIVERSITY of WASHINGTON
eScience Institute

What's the class about? Who are we?

- Objectives
  - Teach how to create and collaborate on data- and computation-intensive research projects
  - Provide practical software skills for data analysis in research & industry.
  - Elevate coding in science to the level of technical writing.
- Instructor introductions

Student Introductions

Prepare For Team Formation During Break

- What to say
  - What data you analyze (generically)
    - Analyze numbers/text/images?
  - Programming experience
    - if-statements? for-statements? functions? modules?
    - python? matplotlib? pandas?
    - Size of biggest project (lines of code or files)
  - What you want to learn from the class
    - Programming? Engineering (design, testing)? Software collaboration techniques?

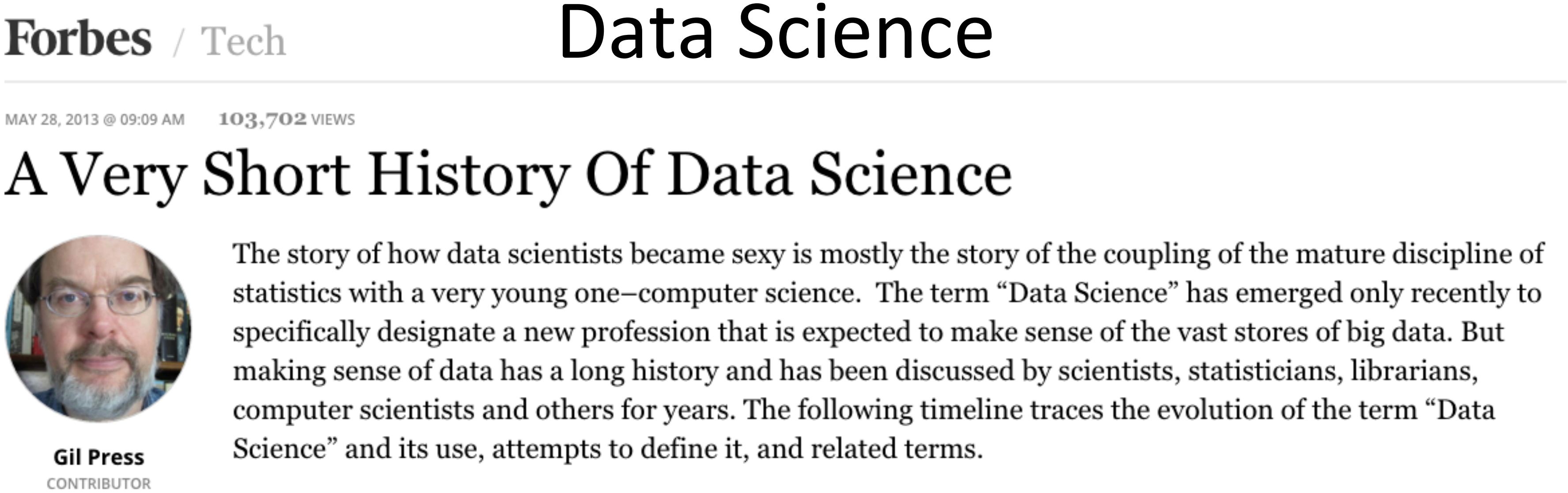
2 minutes max!

Agenda

- Why data science?
- Course overview
- Pronto data
- Getting data with shell scripts

Why Data Science?

Data Science



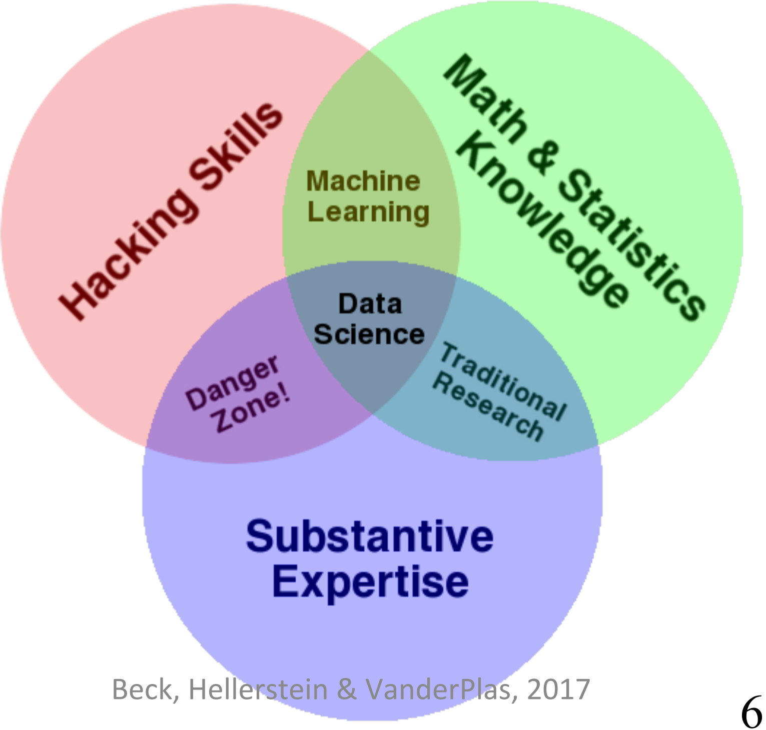
Forbes / Tech

MAY 28, 2013 @ 09:09 AM 103,702 VIEWS

A Very Short History Of Data Science

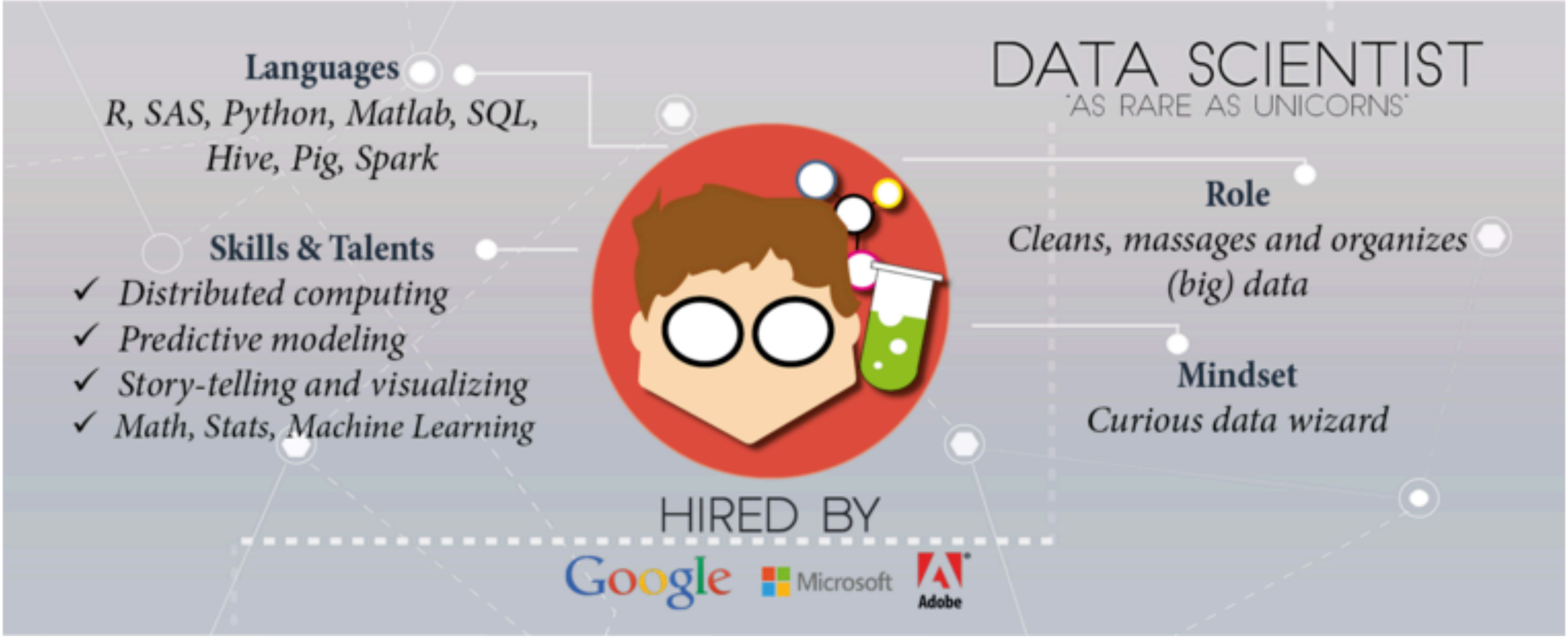
Gil Press
CONTRIBUTOR

The story of how data scientists became sexy is mostly the story of the coupling of the mature discipline of statistics with a very young one–computer science. The term "Data Science" has emerged only recently to specifically designate a new profession that is expected to make sense of the vast stores of big data. But making sense of data has a long history and has been discussed by scientists, statisticians, librarians, computer scientists and others for years. The following timeline traces the evolution of the term "Data Science" and its use, attempts to define it, and related terms.



Beck, Hellerstein & VanderPlas, 2017

Data Scientist – Industry Perspective



DATA SCIENTIST

"AS RARE AS UNICORNS"

Languages

R, SAS, Python, Matlab, SQL, Hive, Pig, Spark

Skills & Talents

- ✓ *Distributed computing*
- ✓ *Predictive modeling*
- ✓ *Story-telling and visualizing*
- ✓ *Math, Stats, Machine Learning*

Role

Cleans, massages and organizes (big) data

Mindset

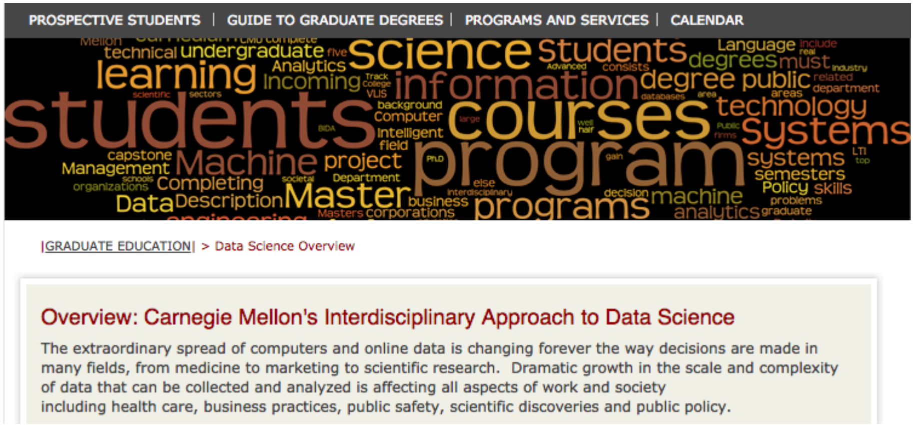
Curious data wizard

HIRED BY

Degrees in Data Science

Certificate in Data Science

Approved by the UW Department of Computer Science & Engineering and developed under the guidance of the eScience Institute

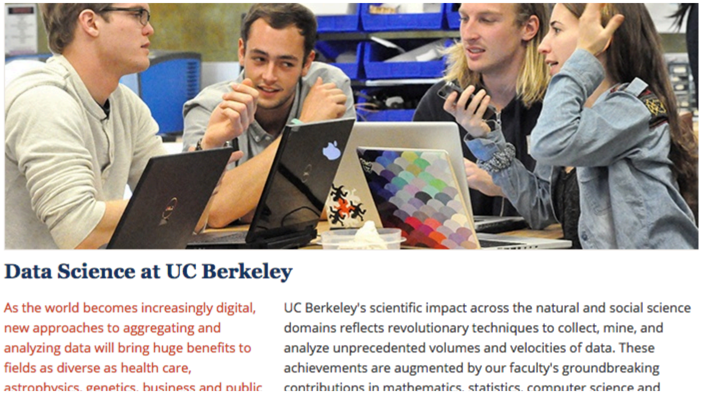


PROSPECTIVE STUDENTS | GUIDE TO GRADUATE DEGREES | PROGRAMS AND SERVICES | CALENDAR

|GRADUATE EDUCATION| > Data Science Overview

Overview: Carnegie Mellon's Interdisciplinary Approach to Data Science

The extraordinary spread of computers and online data is changing forever the way decisions are made in many fields, from medicine to marketing to scientific research. Dramatic growth in the scale and complexity of data that can be collected and analyzed is affecting all aspects of work and society including health care, business practices, public safety, scientific discoveries and public policy.



Data Science at UC Berkeley

As the world becomes increasingly digital, new approaches to aggregating and analyzing data will bring huge benefits to fields as diverse as health care, astrophysics, genetics, business and public

UC Berkeley's scientific impact across the natural and social science domains reflects revolutionary techniques to collect, mine, and analyze unprecedented volumes and velocities of data. These achievements are augmented by our faculty's groundbreaking contributions in mathematics, statistics, computer science and

Course Overview

Skills Taught

- Program in python using the Python scientific stack, including numpy, pandas, and matplotlib.
- Search, evaluate, and integrate into a project externally developed Python packages; create your own Python packages.
- Develop unit tests that validate important aspects of the project implementation.
- Develop software that it can be used by others including: shared code on github, documentation, installing packages, setup, and running computational studies.
- Create technical specifications for what a program should do and how this is accomplished.

Assumptions On Student Background

Question	Response
Years of programming?	Mostly > 1 yr
Years of python?	< 1 yr
Experience with a text editor	Mostly "Yes"
Comfort with if-statements?	Yes
Comfort with for-statements?	Yes
Comfort with functions?	Mostly "Yes"
Python packages (scipy, pandas, ...)	Mostly "No"
Experience with iPython?	Very little
Experience with github?	Very little

Course Structure

- Programming basics
- Version control, python, data manipulation
- Software development
  - Debugging, documentation, design, collaboration
- Software engineering practicum

Class syllabus

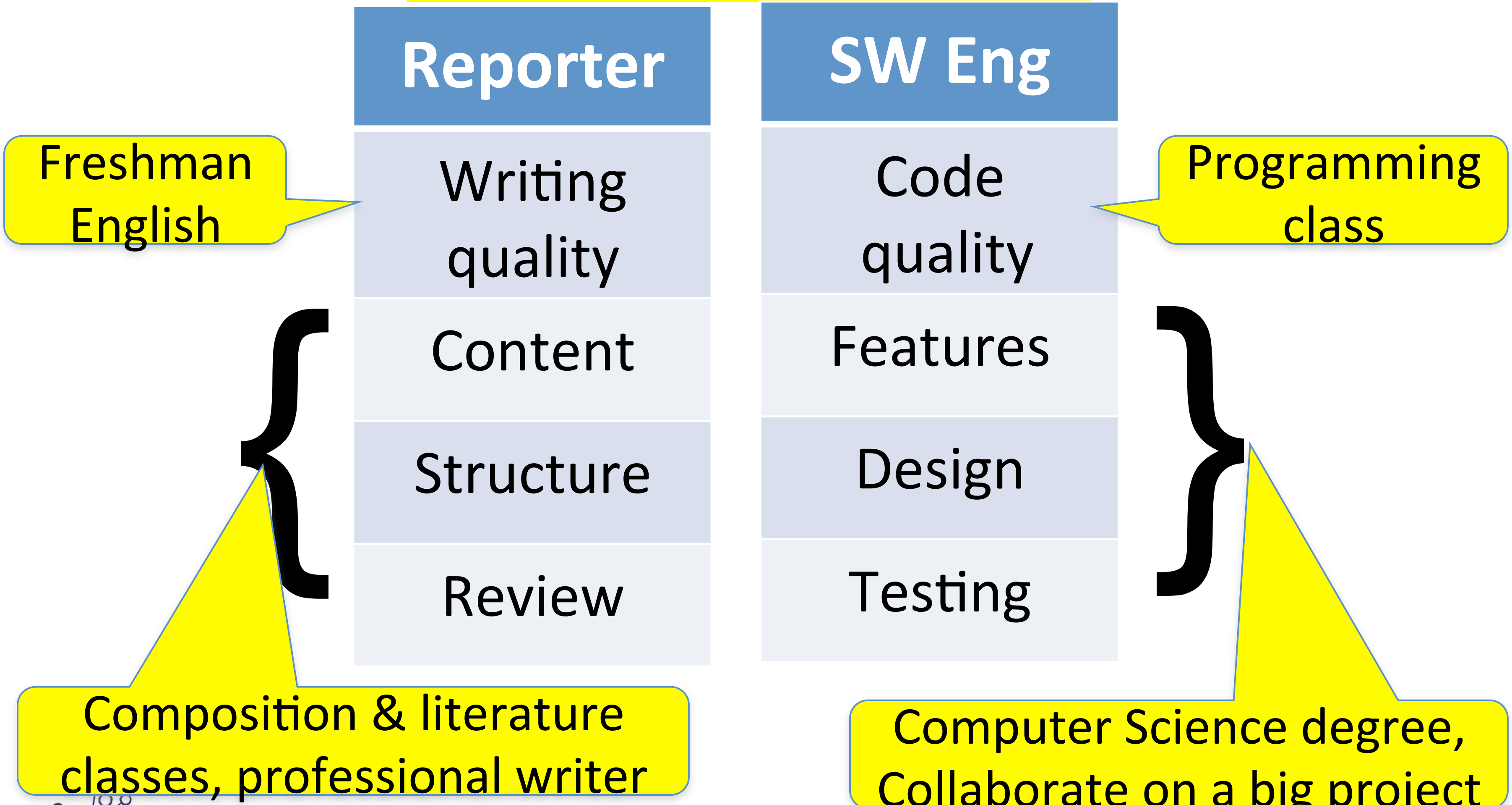
Programming vs. Software Engineering

Analogy: What is the difference between the following kinds of writing:

1. Note to yourself
2. An article in the NY Times

Relating Writing to Software

How Learn Skills



Reporter	SW Eng
Writing quality	Code quality
Content	Features
Structure	Design
Review	Testing

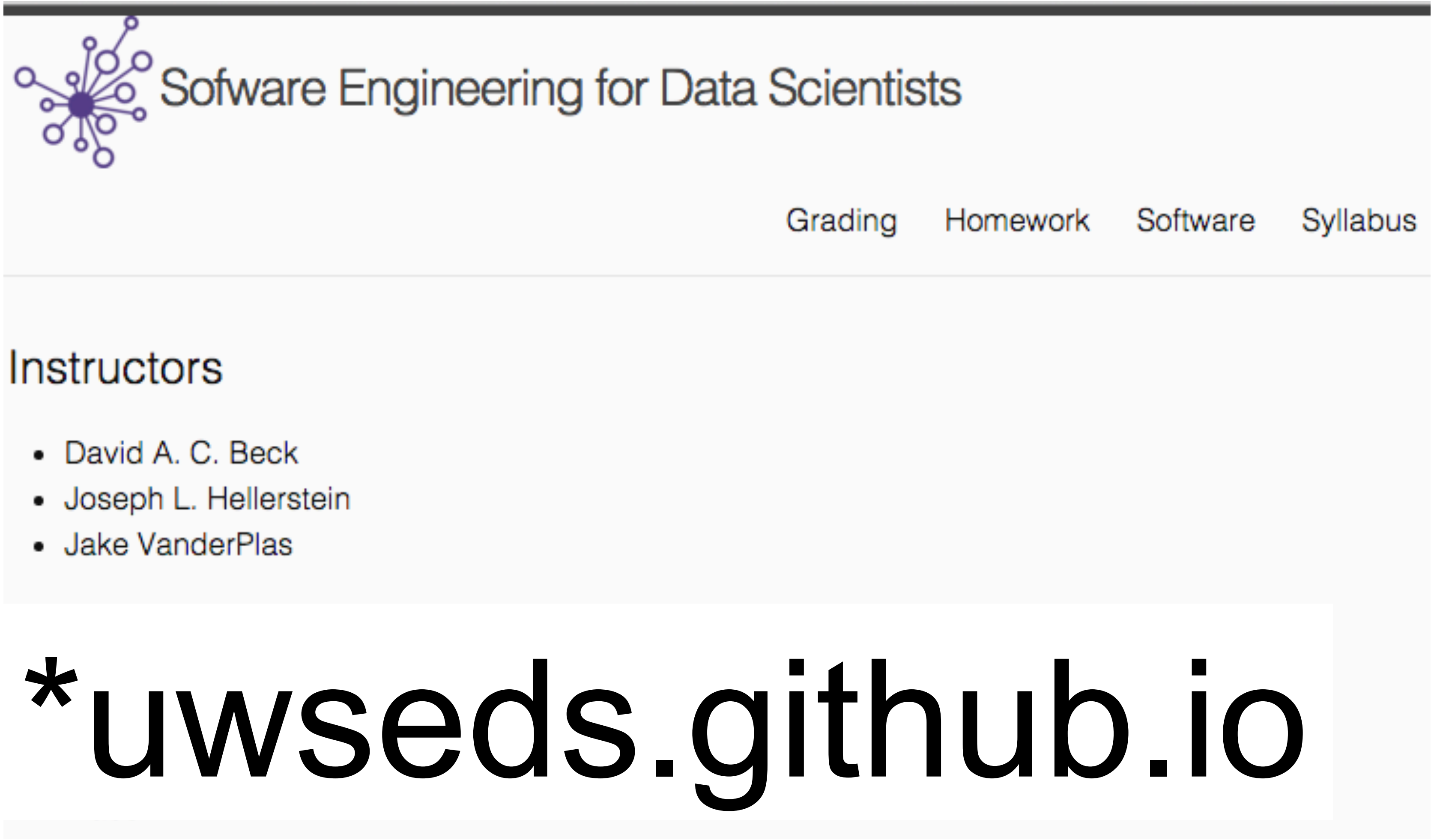
Freshman English

Programming class

Composition & literature classes, professional writer

Computer Science degree, Collaborate on a big project

Course Web Page*



Sofware Engineering for Data Scientists

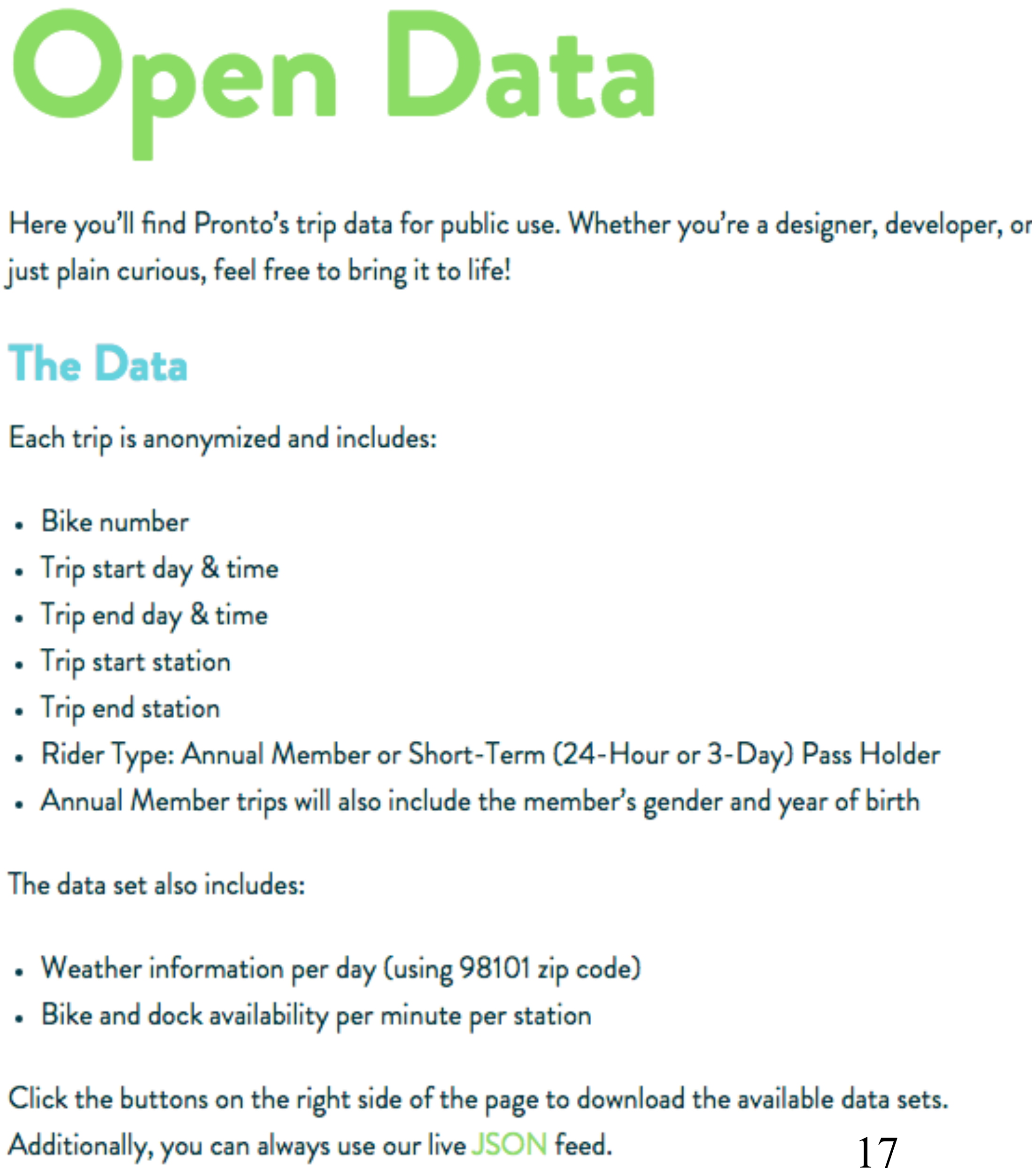
Instructors

- David A. C. Beck
- Joseph L. Hellerstein
- Jake VanderPlas

*uwseds.github.io

Pronto Data

https://www.prontocycleshare.com/datachallenge



Open Data

Here you'll find Pronto's trip data for public use. Whether you're a designer, developer, or just plain curious, feel free to bring it to life!

The Data

Each trip is anonymized and includes:

- Bike number
- Trip start day & time
- Trip end day & time
- Trip start station
- Trip end station
- Rider Type: Annual Member or Short-Term (24-Hour or 3-Day) Pass Holder
- Annual Member trips will also include the member's gender and year of birth

The data set also includes:

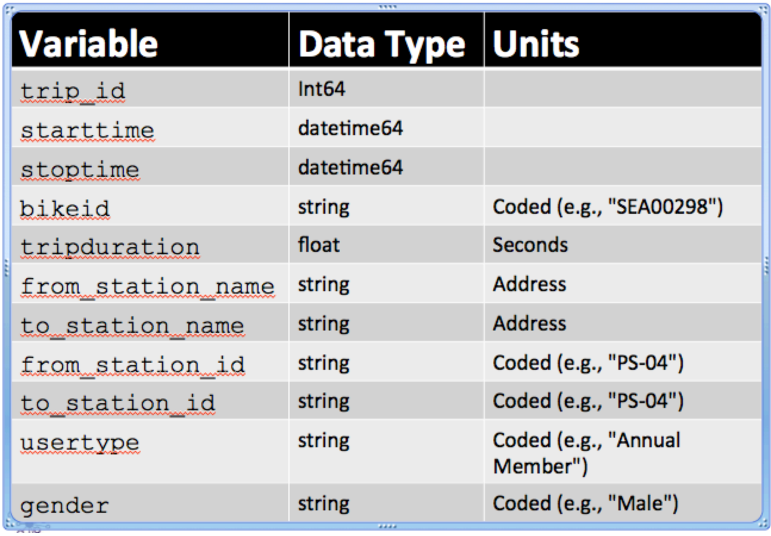
- Weather information per day (using 98101 zip code)
- Bike and dock availability per minute per station

Click the buttons on the right side of the page to download the available data sets. Additionally, you can always use our live JSON feed.

Fields in Pronto Data

Variable	Data Type	Units
`trip_id`	Int64	
`starttime`	datetime64	
`stoptime`	datetime64	
`bikeid`	string	Coded (e.g., "SEA00298")
`tripduration`	float	Seconds
`from_station_name`	string	Address
`to_station_name`	string	Address
`from_station_id`	string	Coded (e.g., "PS-04")
`to_station_id`	string	Coded (e.g., "PS-04")
`usertype`	string	Coded (e.g., "Annual Member")
`gender`	string	Coded (e.g., "Male")

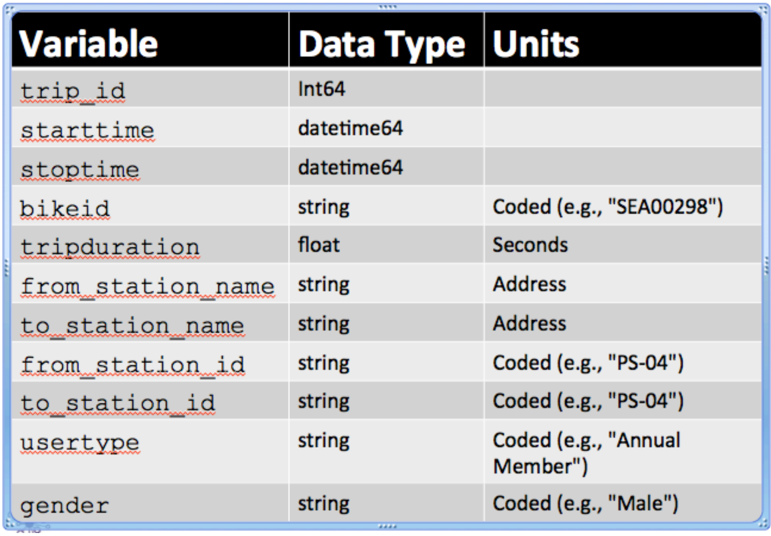
Data Considerations



Variable	Data Type	Units
trip_id	Int64	
starttime	datetime64	
stoptime	datetime64	
bikeid	string	Coded (e.g., "SEA00298")
tripduration	float	Seconds
from_station_name	string	Address
to_station_name	string	Address
from_station_id	string	Coded (e.g., "PS-04")
to_station_id	string	Coded (e.g., "PS-04")
usertype	string	Coded (e.g., "Annual Member")
gender	string	Coded (e.g., "Male")

- Do similar fields have the same data type and/or code (e.g., `from_station_id`, `to_station_id`)?
- Do coded data have useful information hidden in the codes (e.g., "PS-04")?
- How merge with other data (e.g., weather)?

Data Schema



Variable	Data Type	Units
trip_id	Int64	
starttime	datetime64	
stoptime	datetime64	
bikeid	string	Coded (e.g., "SEA00298")
tripduration	float	Seconds
from_station_name	string	Address
to_station_name	string	Address
from_station_id	string	Coded (e.g., "PS-04")
to_station_id	string	Coded (e.g., "PS-04")
usertype	string	Coded (e.g., "Annual Member")
gender	string	Coded (e.g., "Male")

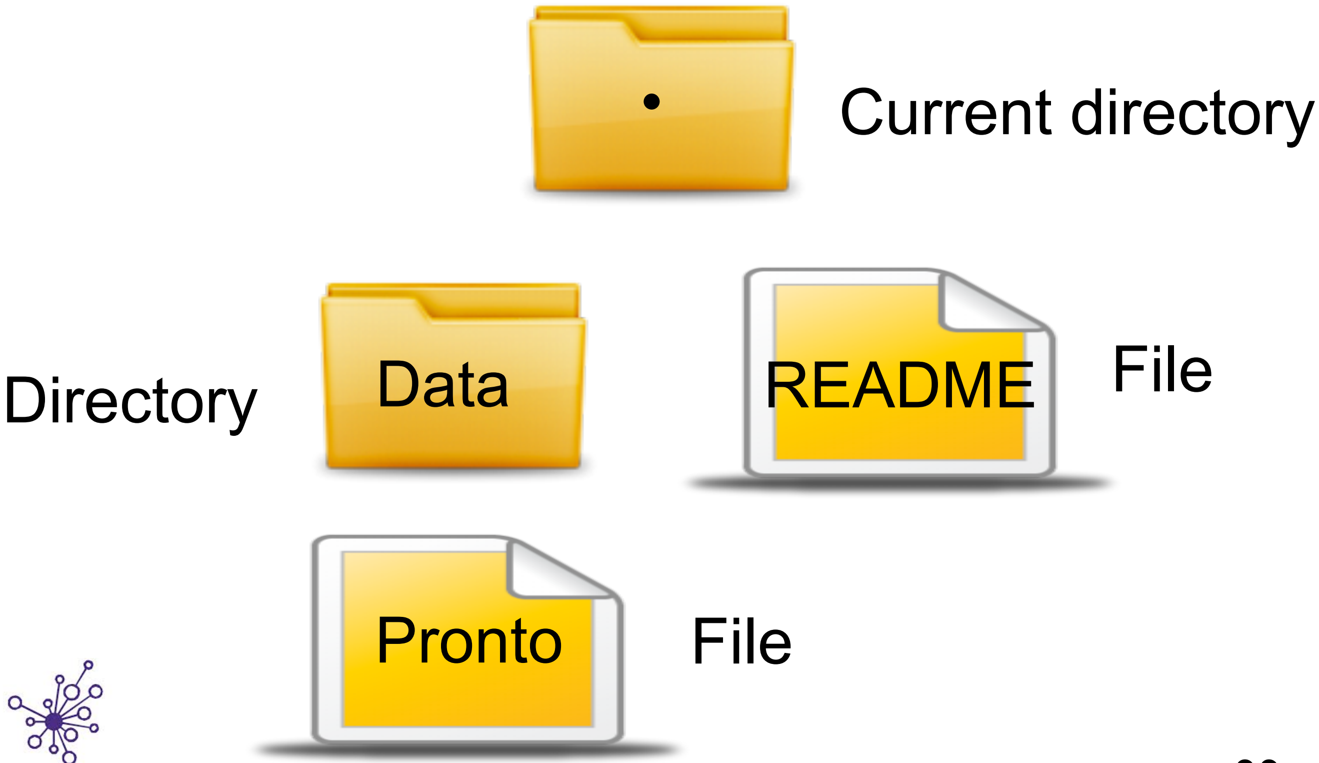
- "Meta data" – describes the data
  - data types
  - units
  - "keys" (how relate one data set to another)

Getting Data With Shell Scripts

File System Basics

- File – container of data
- Directory – container of files and directories

Directories are organized into a tree



.

Current directory

Directory

Data

README

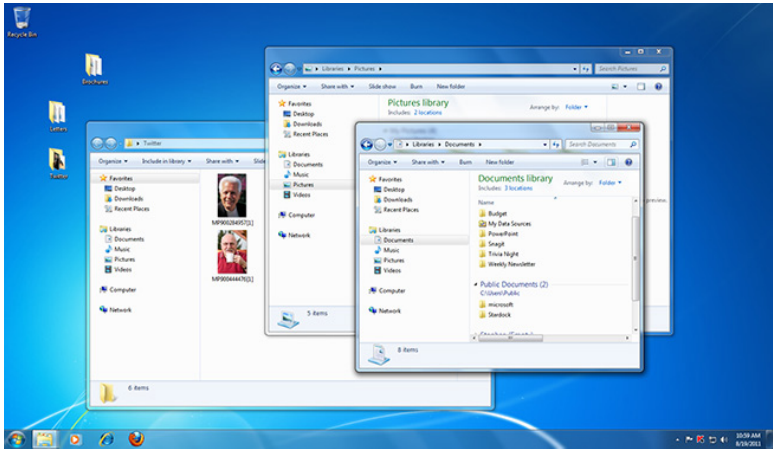
File

Pronto

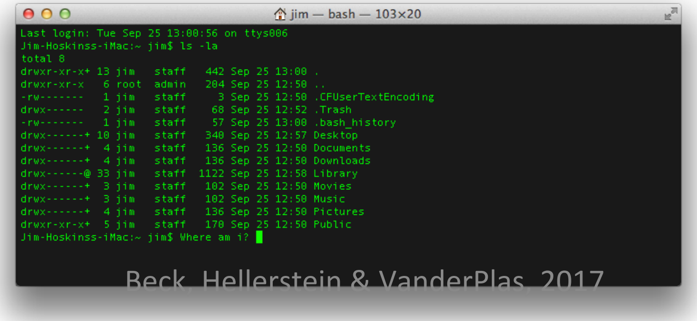
File

Command Line Tools

Graphical User Interface (GUI)



Command Line Interface (CLI)



```
Last login: Tue Sep 25 13:00:56 on ttys006
Jim-Hoskinss-iMac:~ jim$ ls -la
total 8
drwxr-xr-x+ 13 jim   staff   442 Sep 25 13:00 .
drwxr-xr-x   6 root  admin   204 Sep 25 12:50 ..
-rw-------   1 jim   staff     3 Sep 25 12:50 .CFUserTextEncoding
drwx------   2 jim   staff    68 Sep 25 12:52 .Trash
-rw-------   1 jim   staff    57 Sep 25 13:00 .bash_history
drwx------+ 10 jim   staff   340 Sep 25 12:57 Desktop
drwx------+  4 jim   staff   136 Sep 25 12:50 Documents
drwx------+  4 jim   staff   136 Sep 25 12:50 Downloads
drwx------@ 33 jim   staff  1122 Sep 25 12:58 Library
drwx------+  3 jim   staff   102 Sep 25 12:50 Movies
drwx------+  3 jim   staff   102 Sep 25 12:50 Music
drwx------+  4 jim   staff   136 Sep 25 12:50 Pictures
drwxr-xr-x+  5 jim   staff   170 Sep 25 12:50 Public
Jim-Hoskinss-iMac:~ jim$ Where am i?
```

Command Line Tools

What is the command line?

- Also known as a 'shell'
- Most common is `bash` (what we will use)
  - Bourne Again Shell
  - Reimplementation of a shell from 1977
  - Every OS/X Mac
  - Every Linux
  - Every supercomputer
- For later…
  - Programming language itself!

Command Line Tools

Where is the command line?

Mac (pre-installed)

Applications -> Utilities -> Terminal

Windows (after install Git / Bash)

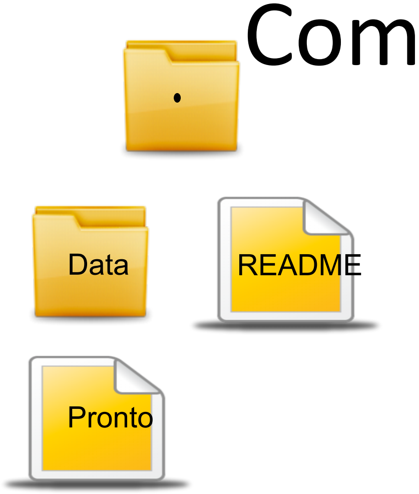
Start -> Git -> Git Bash

(See the "software" tab of the course web page to install Gitbash.)

Command Line Tools

- Commands take arguments (stuff after cmd.)
  - Arguments alter the function of commands, e.g.
    - Specify what file to use as input
  - Many commands accept the special argument to return help, usually one of
    - `--help`
    - `-help`
    - `-h`
- Tab completion is your friend!
  - When entering file arguments
  - Hitting tab key will autocomplete the filename

Commands for Files & Directories



- By category
  - Create
    - Directory: `mkdir`
    - File: various (e.g., `cp`)
  - View contents
    - Directory: `ls`
    - File: `cat`
  - Remove
    - Directory: `rmdir`
    - File: `rm`

Demo

1. Create the project directory structure and README
2. Get the pronto data from the Internet
3. Unpack the data

   Comma separated variable (CSV) files

4. Automate the workflow using a shell script

Useful Shell Commands

Command	Task	Example usage
ls	List files	ls
cp	Copy files	cp original_file new_file
mv	Move / rename files	mv original_file new_file
rm	Remove / delete files	rm original_file
cd	Change directory	cd some_directory
pwd	Print working / current directory	pwd
mkdir	Create directory	mkdir some_directory
rmdir	Remove / delete directory	rmdir some_directory
cat	View files	cat some_file
head	View begining of file	head some_file
tail	View end of file	tail some_file
grep	Search file for matching lines	grep search.text some_file
sort	Sort lines	sort some_file
uniq	Print unique lines	uniq some_file
diff	Compare to files	diff original_file new_file
unzip	Uncompress a file	unzip compressed_file.zip
curl	Download a file using its URL	curl some URL

Also see http://www.pixelbeat.org/cmdline.html

Also search shell + <cmd name>

Lecture Review: Data Essentials

- Structure of data
  - Schema
  - File format (CSV)
- File systems (directories, files)
- Terminal sessions
- Shell commands
  - File system operations (create, view, delete)
  - Data access (download URL, decompress)