

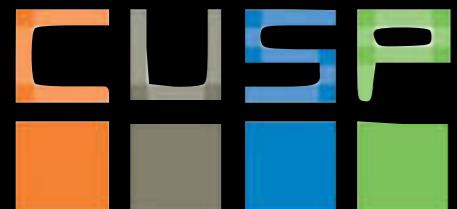
Urban Informatics

Fall 2015

dr. federica bianco fb55@nyu.edu

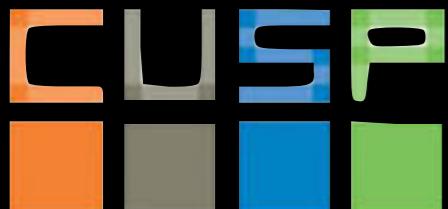


@fedhere



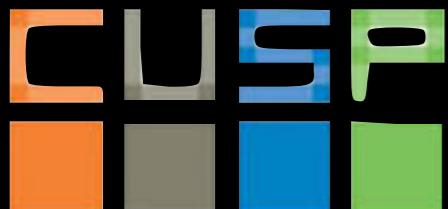
Summary:

- **Epistemological concepts:**
falsifiability, law of parsimony,
- **Good scientific practice:**
reproducibility of research



Summary:

- **Epistemological concepts:**
falsifiability, law of parsimony,
- **Good scientific practice:**
reproducibility of research,
- **Gathering and parsing data:**
data munging or wrangling, data jujitsu.
 - types of data
 - reporting data for reproducibility
 - cleaning and tabulating data
 - reading data from CSV files
 - reading data from JSON files
 - reading data from the CUSP dataHub
 - reading data from APIs

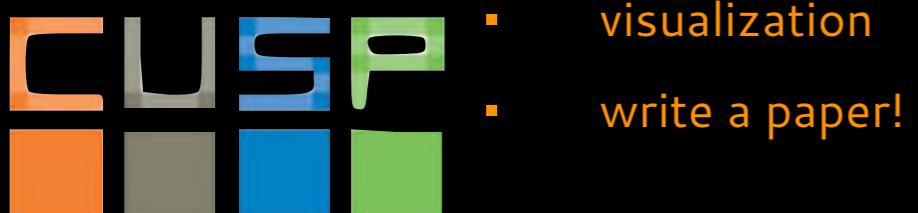


- IDEA
- dataset
 - define ideal data
 - figure out best data available
 - figure out if you can get new data
 - obtain data (including policy issues + technical issues)

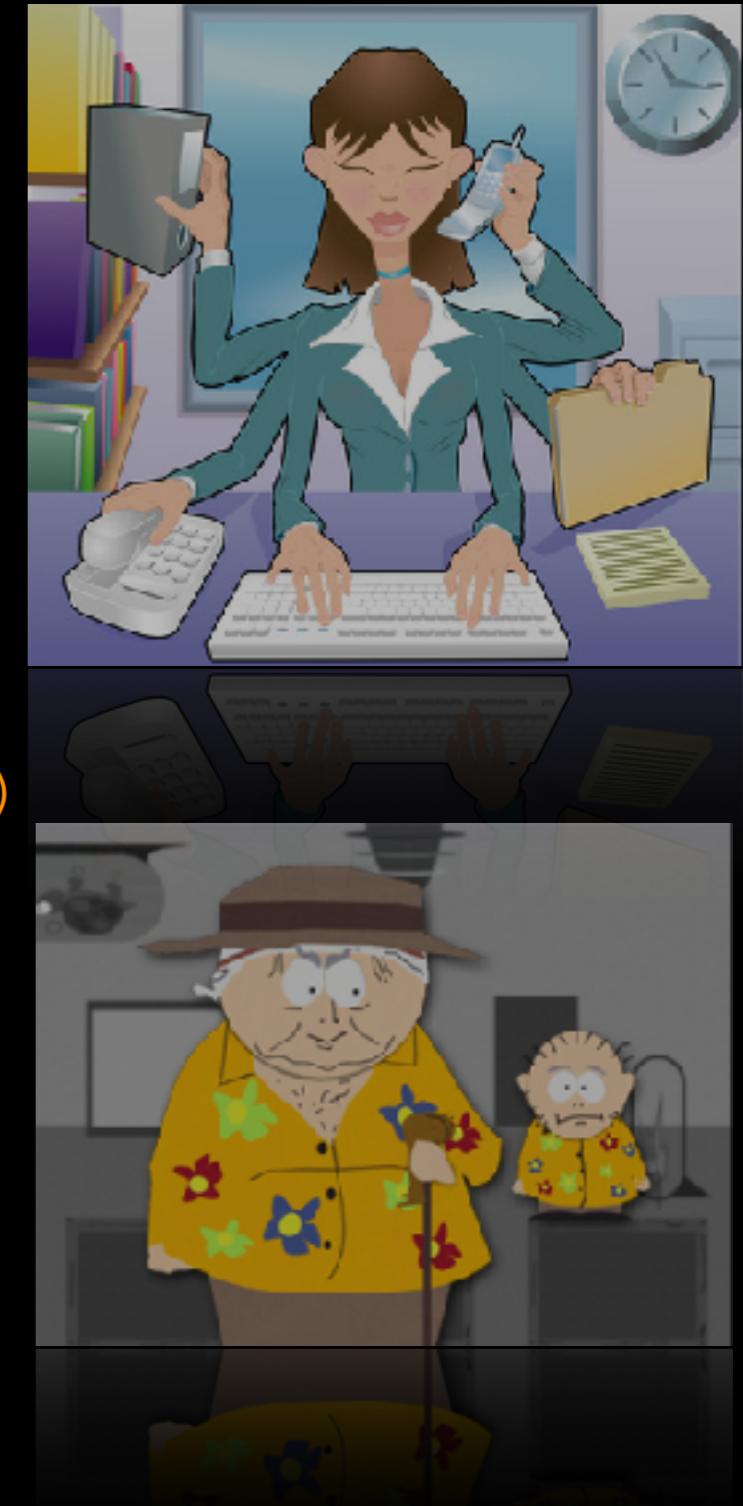
- data handling
 - joining databases
 - formatting data
- exploratory data analysis
 - machine learning (clustering? dimensionality reduction?)

- statistics
 - models (regression)
 - prediction
 - validation (simulations)

- interpretation
- presentation



- visualization
- write a paper!



- IDEA

- dataset

PROBLEM IDENTIFICATION

- figure out best data available

FORMULATING HYPOTHESIS

- obtain data (including policy issues + technical issues)

- data handling

- joining databases
- formatting data

- exploratory data analysis

- machine learning (clustering? dimensionality reduction?)

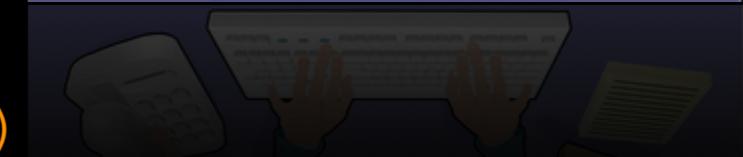
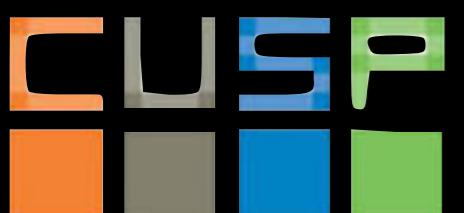
- statistics

- models (regression)
- prediction
- validation (simulations)

- interpretation

- presentation

- visualization
- write a paper!



- IDEA
- dataset

- define ideal data
- figure out best data available
- figure out if you can get new data
- obtain data (including policy issues + technical issues)

- data handling

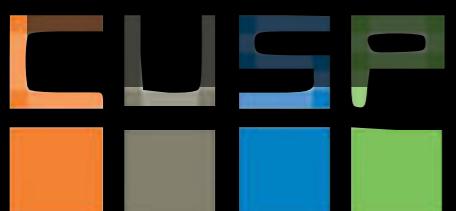
- joining databases
- formatting data

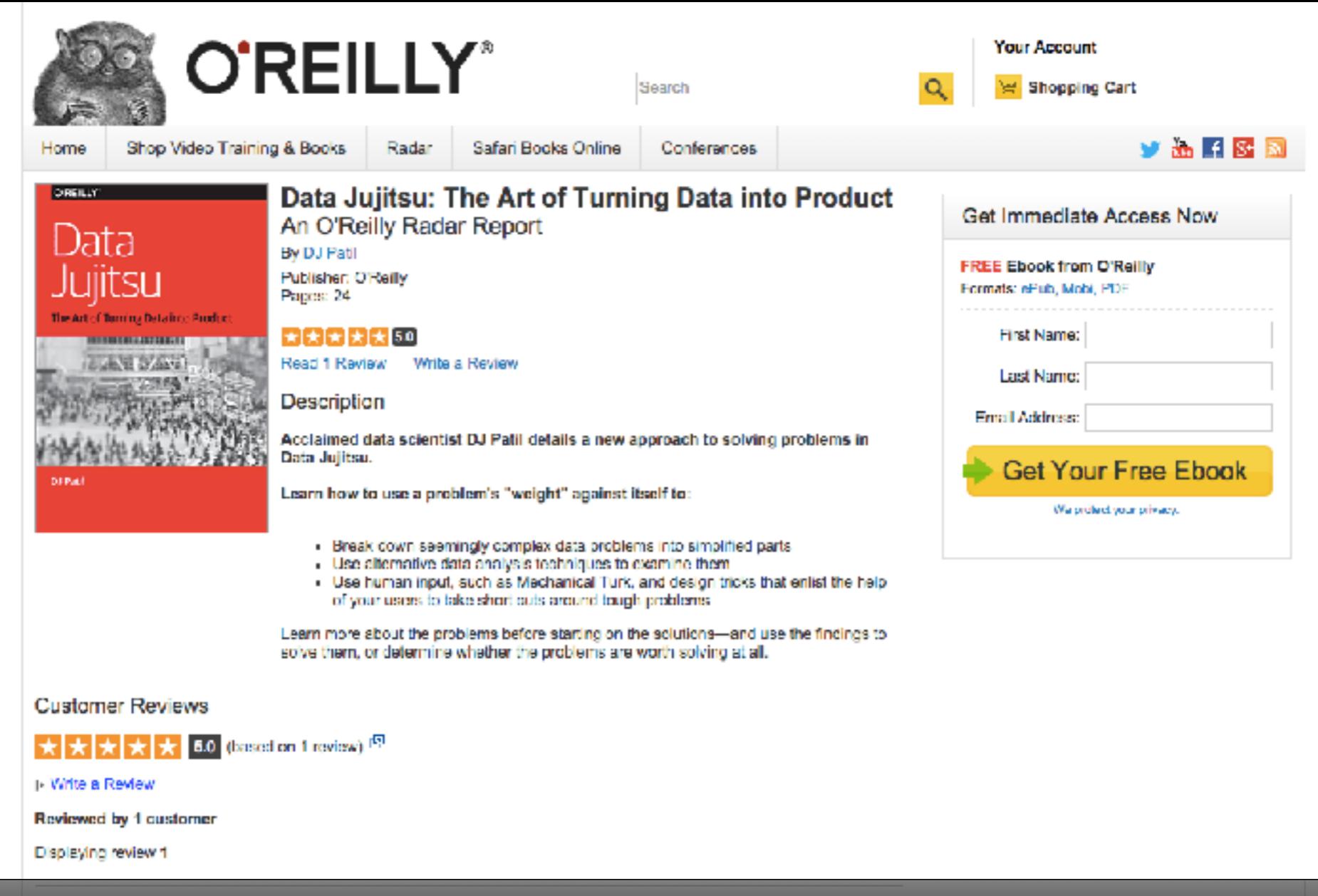
- exploratory data analysis
DATA PREPARATION
machine learning (clustering? dimensionality reduction?)

- statistics
generally takes 70-90%
of the life of a data driven
investigation
 - prediction
 - validation (simulations)

- interpretation
- presentation

- visualization
- write a paper!





The screenshot shows the O'Reilly website product page for "Data Jujitsu: The Art of Turning Data into Product".

Product Information:

- Title:** Data Jujitsu: The Art of Turning Data into Product
- Author:** An O'Reilly Radar Report
- By:** DJ Patil
- Publisher:** O'Reilly
- Pages:** 24

Reviews: ★★★★☆ 5.0 (based on 1 review) [Read 1 Review](#) [Write a Review](#)

Description:

Acclaimed data scientist DJ Patil details a new approach to solving problems in Data Jujitsu.

Learn how to use a problem's "weight" against itself to:

- Break down seemingly complex data problems into simplified parts
- Use alternative data analysis techniques to examine them
- Use human input, such as Mechanical Turk, and design tricks that enlist the help of your users to take short cuts around tough problems

Learn more about the problems before starting on the solutions—and use the findings to solve them, or determine whether the problems are worth solving at all.

Customer Reviews:

★★★★★ 5.0 (based on 1 review) [Read 1 Review](#) [Write a Review](#)

Reviewed by 1 customer

Displaying review 1

Get Immediate Access Now:

FREE Ebook from O'Reilly
Formats: ePub, Mobi, PDF

First Name: Last Name: Email Address:

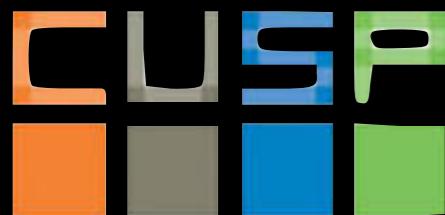
Get Your Free Ebook

We protect your privacy.

CUSP SWIGVIA SWIGVIA is a trademark of CUSP

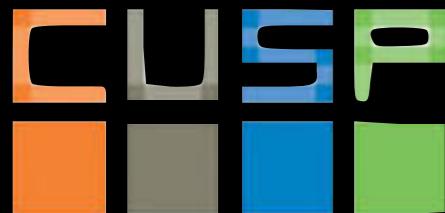
[...]data products are unique in that they are often extremely difficult, and seemingly intractable for small teams with limited funds. Yet, they get solved every day.

How? Are the people who solve them superhuman data scientists who can come up with better ideas in five minutes than most people can in a lifetime? Are they magicians of applied math who can cobble together millions of lines of code for high-performance machine learning in a few hours? No.
(continue in the next page...)



(...continued)

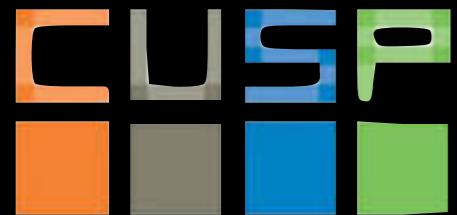
Many of them are incredibly smart, but meeting big problems head-on usually isn't the winning approach. There's a method to solving data problems that avoids the big, heavyweight solution, and instead, concentrates on building something quickly and iterating. Smart data scientists don't just solve big, hard problems; they also have an instinct for making big problems small.



Data are messy

...

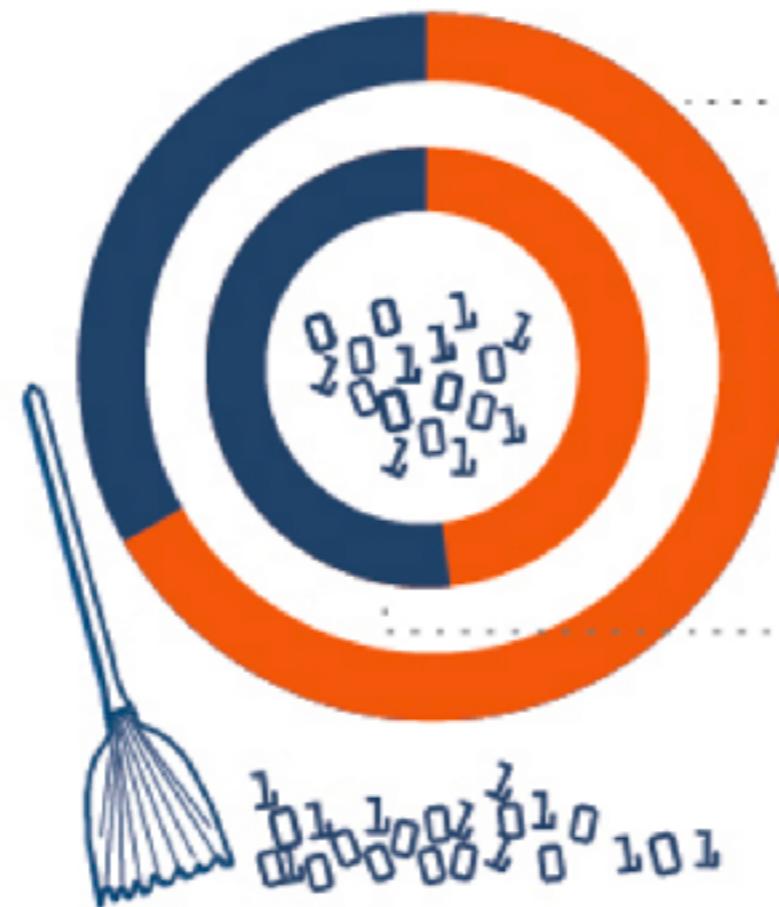
(and you have to clean the mess!)



II: Data Wrangling

WHAT ARE THEIR BIGGEST CHALLENGES?

Dirty data is the #1 hurdle for data scientists.



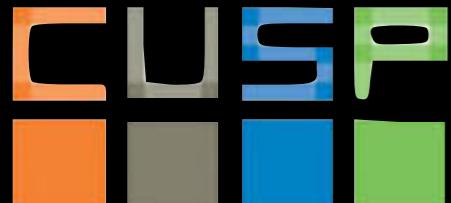
66.7%

of data scientists say cleaning and organizing data is their most time-consuming task.

52.3%

of data scientists cite poor quality data as their biggest daily obstacle.

2015 survey of data scientists



<https://www.crowdflower.com/the-data-behind-todays-data-scientists-an-infographic/>
II: Data Wrangling

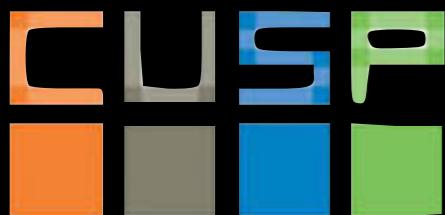
Reproducible research means:

code raw data

Raw data formats: API, Json, binary, paper...

Tidy data: tabulated data, binary, csv, tsv, ascii,
Excel (ugh!) or (object oriented) Json,...

	var 1	var 2
obs 1	7.4	9.2
obs 2	NaN	13.1



Reproducible research means:

code raw data

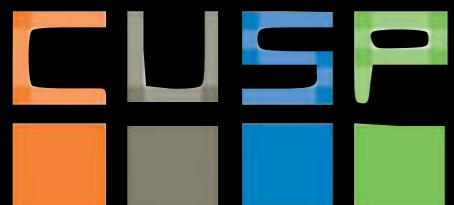
Remove sensitive data

Some day you or a collaborator may accidentally commit sensitive data, such as a password or SSH key, into a Git repository. Although you can remove the file from the latest commit with `git rm`, the file will still exist in the repository's history. Fortunately, there are other tools that can entirely remove unwanted files from a repository's history. This article will explain how to use two of them: `git filter-branch` and the [BFG Repo-Cleaner](#).

Danger: Once you have pushed a commit to GitHub, you should consider any data it contains to be compromised. If you committed a password, change it! If you committed a key, generate a new one.

This article tells you how to make commits with sensitive data unreachable from any branches or tags in your GitHub repository. However, it's important to note that those commits may still be accessible in any clones or forks of your repository, directly via their SHA-1 hashes in cached views on GitHub, and through any pull requests that reference them. You can't do anything about existing clones or forks of your repository, but you can permanently remove all of your repository's cached views and pull requests on GitHub by contacting [GitHub support](#).

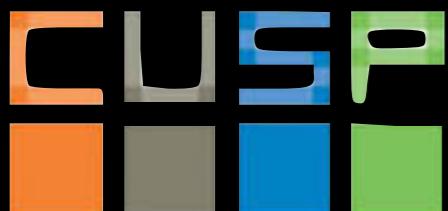
<https://help.github.com/articles/remove-sensitive-data/>



Reproducible research:

How to share your data

- Share/Reference the source of your raw data
- Share the “tidy” data
- Share the code used to process the data at each step

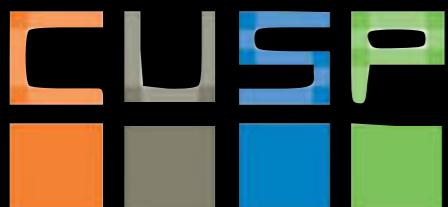


Reproducible research:

How to share your data

- Share/Reference the source of your raw data
- Share the “tidy” data
- Share the code used to process the data at each step

[https://github.com/jakevdp/PythonicPerambulations/blob/master/
content/downloads/notebooks/ProntoData.ipynb](https://github.com/jakevdp/PythonicPerambulations/blob/master/content/downloads/notebooks/ProntoData.ipynb)



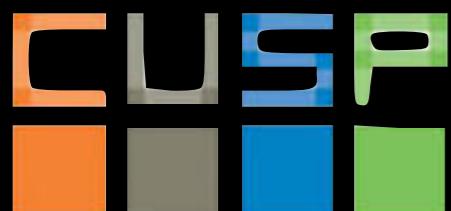
Reproducible research:

How to share your data

let's agree on a single location for PUI data. That way the graders home dirs do not fill up with copies of the same dataset.

PUI RULE: place
all your data in
PUIdata

where PUIdata is
pointed to by an
env variable
PUIDATA



Reproducible research:

How to share your data

let's agree on a single location for PUI data. That way the graders home dirs do not fill up with copies of the same dataset.

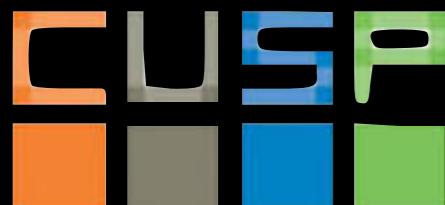
PUI RULE: place all your data in PUIdata

where PUIdata is pointed to by an env variable
PUIdata

```
1 #downloading data
2 !curl -O https://s3.amazonaws.com/pronto-data/open_data_year_one.zip
3 #unpacking into $PUIDATA
4 !unzip open_data_year_one.zip -d $PUIDATA

% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
               Dload  Upload   Total   Spent  Left  Speed
100 70.8M  100 70.8M    0      0  8587k      0  0:00:08  0:00:08  --:--:-- 9783k
Archive: open_data_year_one.zip
inflating: /Users/fbianco/science/Dropbox/UI/PUIdata/2015_station_data.csv
inflating: /Users/fbianco/science/Dropbox/UI/PUIdata/2015_status_data.csv
inflating: /Users/fbianco/science/Dropbox/UI/PUIdata/2015_trip_data.csv
inflating: /Users/fbianco/science/Dropbox/UI/PUIdata/2015_weather_data.csv
inflating: /Users/fbianco/science/Dropbox/UI/PUIdata/README.txt

1 !ls $PUIDATA
2015_station_data.csv  nybb_16d.zip          zbp06totals.zip
2015_status_data.csv   nycb2010_16d        zbp07totals.zip
2015_trip_data.csv    rows.csv            zbp08totals.zip
2015_weather_data.csv t_sf1_dp_cd.xlsx    zbp09totals.zip
ACS_10_SF4_DP03       t_sf1_dp_nyc.xlsx  zbp10totals.txt
```



export PUIdata="\$HOME/PUIdata" II: Data Wrangling

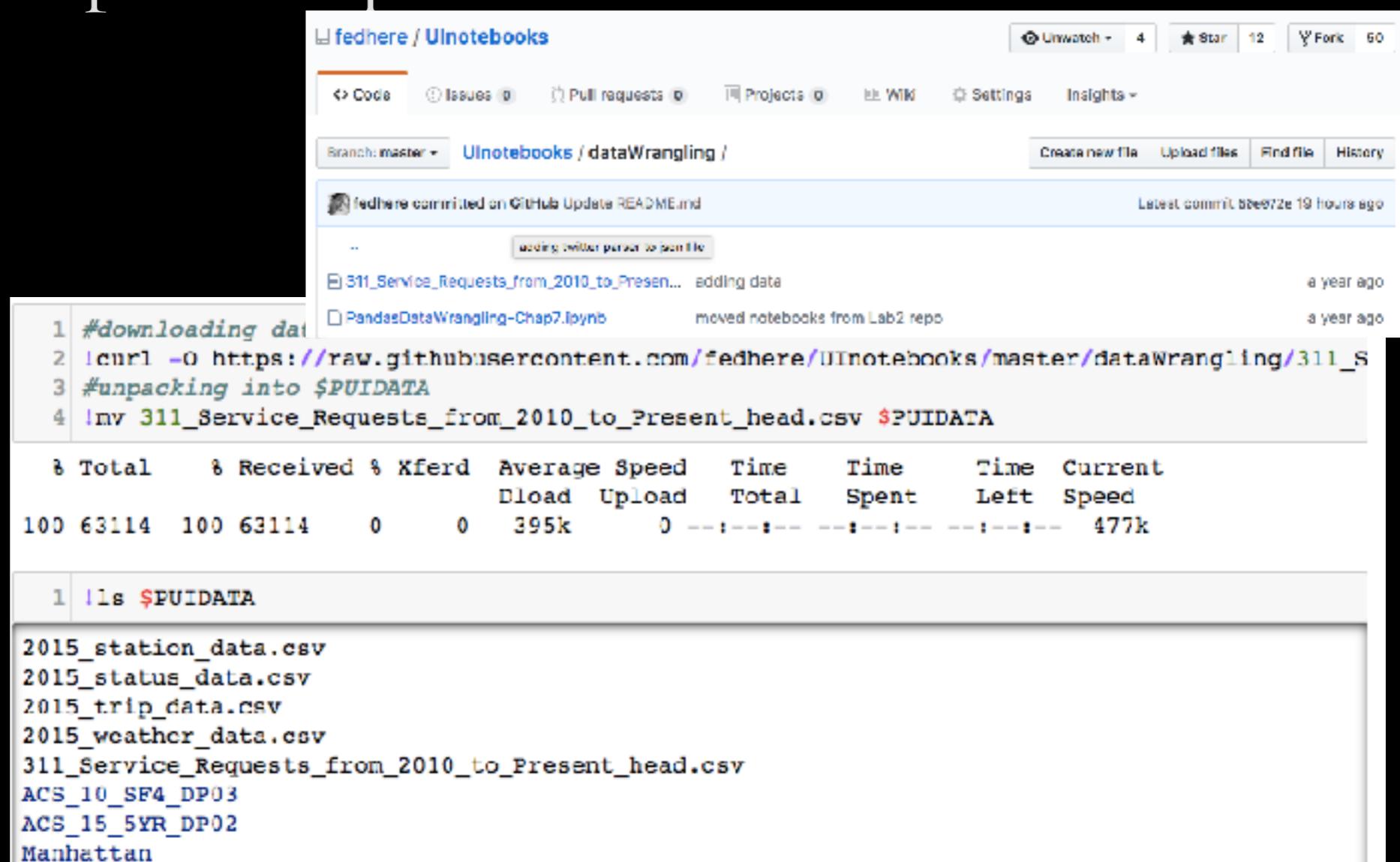
Reproducible research:

How to share your data

let's agree on a single location for PUI data. That way the graders home dirs do not fill up with copies of the same dataset.

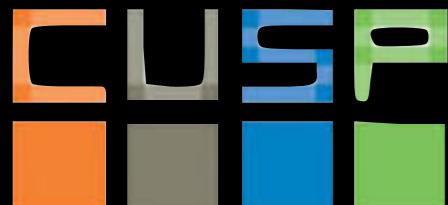
PUI RULE: place all your data in PUIdata

where PUIdata is pointed to by an env variable
PUIDATA



The screenshot shows a GitHub repository page for 'fedhere/UInotebooks'. The repository has 4 forks and 12 stars. The commit history shows several commits from 'fedhere' over the past year. The terminal output shows the process of downloading data from a GitHub raw URL, unpacking it into a directory, and then listing the files in that directory. The output includes a progress bar for the download and a list of CSV files.

```
#downloading data
curl -O https://raw.githubusercontent.com/fedhere/UInotebooks/master/dataWrangling/311_Service_Requests_from_2010_to_Present_head.csv
#unpacking into $PUIDATA
!ls $PUIDATA
2015_station_data.csv
2015_status_data.csv
2015_trip_data.csv
2015_weather_data.csv
311_Service_Requests_from_2010_to_Present_head.csv
ACS_10_SF4_DP03
ACS_15_5YR_DP02
Manhattan
```



[https://github.com/fedhere/UInotebooks/
tree/master/dataWrangling](https://github.com/fedhere/UInotebooks/tree/master/dataWrangling)

II: Data Wrangling

Reproducible research:

How to share your data

if you are working on compute or jupyterhub at CUPS and you are working with data available in the CUSP Data Facility (DF) you can directly read the data

```
from __future__ import print_function
__author__ = '__fb2__'
```

```
import os
import pandas as pd
%pylab inline
```

```
DFDATA = os.getenv("DFDATA")
DFDATA
```

```
Populating the interactive namespace from numpy and matplotlib
```

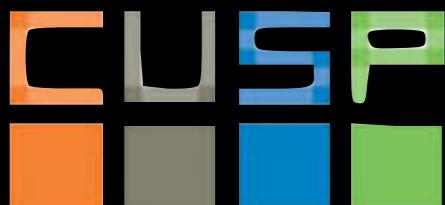
```
'/gws/open/NYCOpenData/nycopendata/data'
```

```
DFDATA = "/gws/open/NYCOpenData/nycopendata/data/"
```

Reads in CSV files from the CUSP Data Facility and plots numerical values and time series NOTE: must have the DFDATA environmental variable pointing to "/gws/open/NYCOpenData/nycopendata/data/" Part 1 uses the Natural Gas Consumption data <https://datahub.cusp.nyu.edu/dataset/uedp-fegm> Part 2 uses the Social Media metrics of NYC agencies data <https://datahub.cusp.nyu.edu/dataset/5t3a-rs48>

Part 1. Gas Data

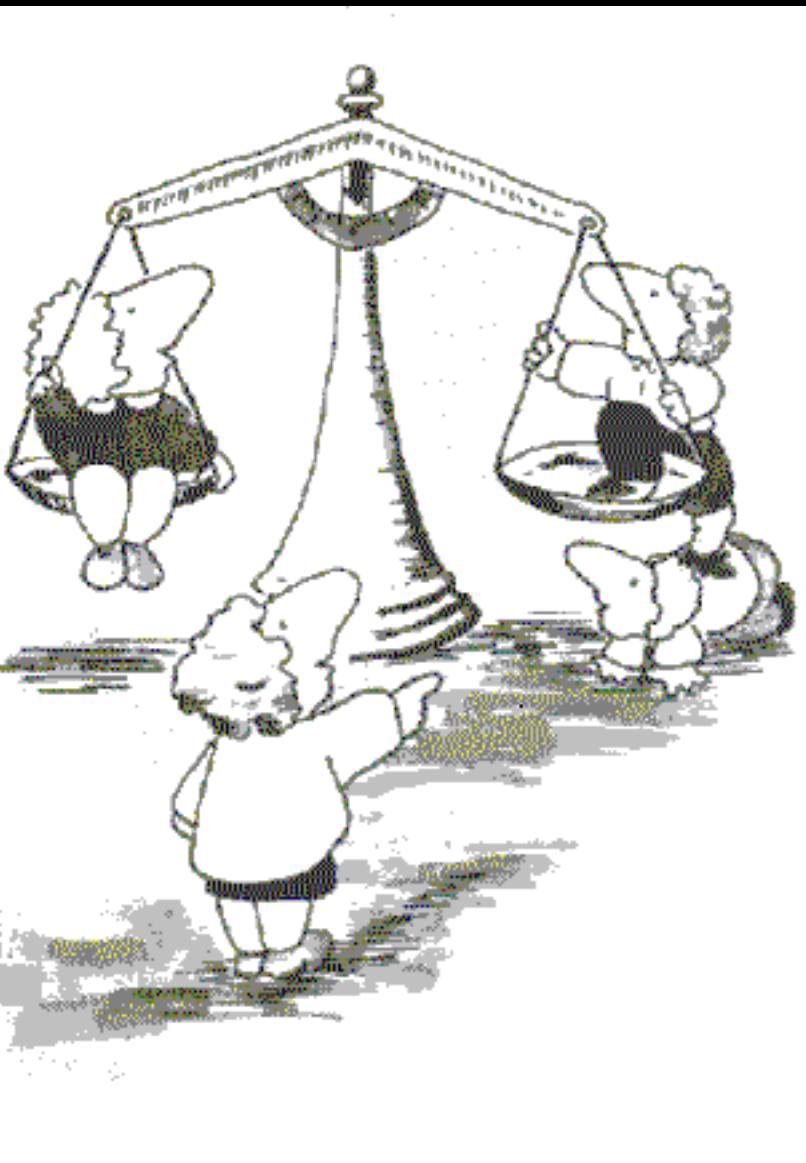
```
# read in gas data
df_gas = pd.read_csv(DFDATA + "/uedp-fegm/1414245967/uedp-fegm")
```



Reproducible research means:

Data are noisy

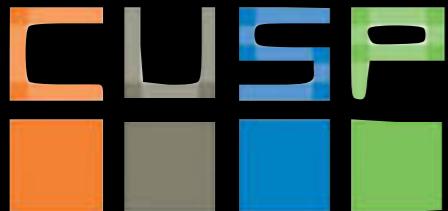
- Data entry errors
- Measurement errors
- Extraction errors



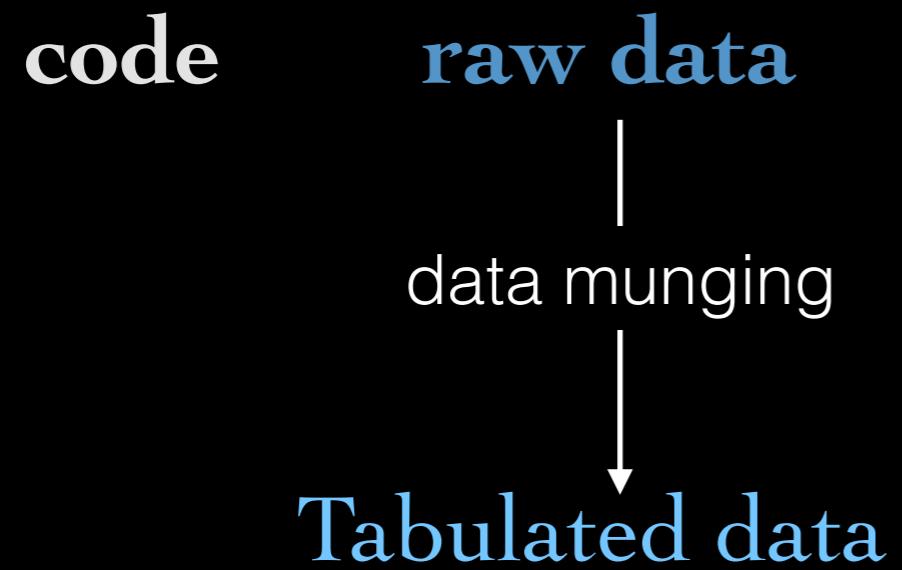
Make sure you do not make them noisier...

Typical data wrangling issues

- Licensing issues/Privacy (some data unavailable)
- Missing required field
- Primary key violation (two people with the same social security number)
- Parsing text into fields (separator issues)
- Naming conventions: NYC vs New York
- Different representations (2 vs Two)
- Fields corrupted (too long get truncated)
- Formatting issues – especially dates
- Redundant Records (exact match or other)



Reproducible research means:



Reproducible research means:

code raw data formats: API, Json, binary, paper...



data munging



Tabulated data

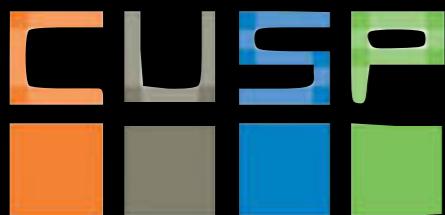
tabulated data: binary, csv, tsv, ascii, Excel (ugh!) or (object oriented) Json,...

Reproducible research means:

code raw data

Proper data table orientations:
columns -> variables
rows -> observations

	var 1	var 2
obs 1	7.4	9.2
obs 2	NaN	13.1

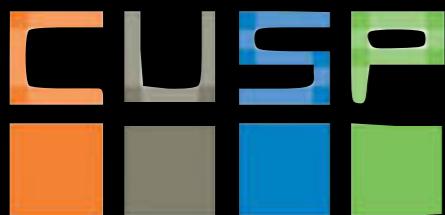


Reproducible research means:

code raw data

Proper data table orientations:
columns -> variables
rows -> observations

	var 1	var 2
obs 1	7.4	9.2
obs 2	NaN	13.1

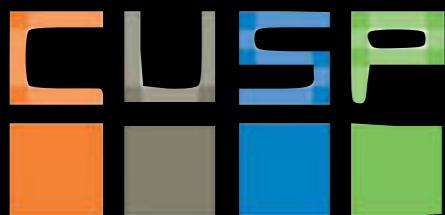


Reproducible research means:

code raw data

provide detailed observation info if possible

	var 1	var 2	link
obs 1	7.4	9.2	URL 1
obs 2	NaN	13.1	URL 1



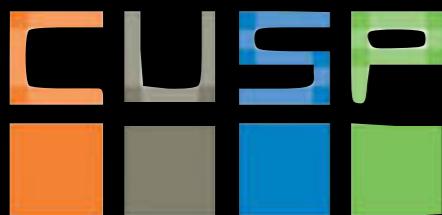
Reproducible research means:

code raw data

impute missing data correctly and unambiguously

	var 1	var 2	link
obs 1	7.4	9.2	URL 1
obs 2	Nan	13.1	URL 1

never use
fake values
for missing
data!



Reproducible research means:

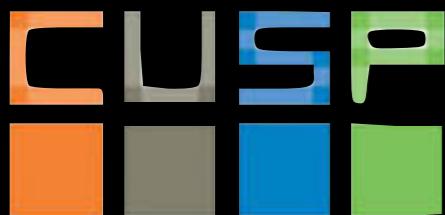
code raw data

Raw data formats: API, Json, binary, paper...

Tidy data: tabulated data, binary, csv, tsv, ascii, Excel (ugh!) or (object oriented) Json,...

Report separately :

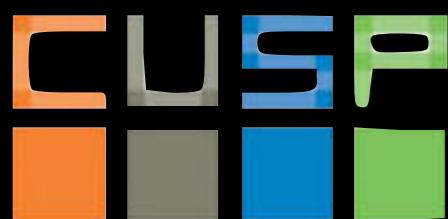
- **Details about variables:** e.g. date: “date when test was performed”
- **Units:** remember the Mars Rover that was lost at sea cause scientists got Metric and Imperial units mixed up?!?!



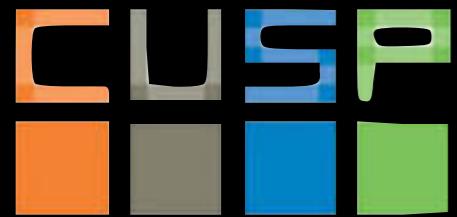
Reproducible research means:

code raw data

	speed (m/s)	distance (m)	link	accessed date
obs 1	7.4	9.2	URL 1	09/16/17
obs 2	NaN	13.1	URL 1	09/23/16



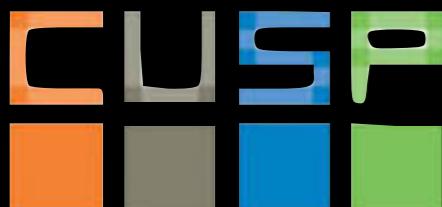
data ingestion



Types of Data Files:

- CSV comma separated values (also TSV - tab)
- JSON: corresponds to a Python data dictionary
- XML: similar to HTML

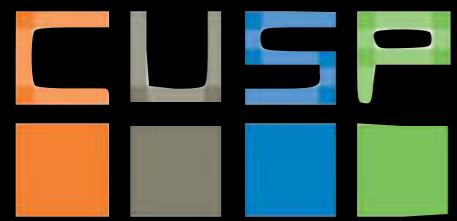
support nested structures





loading data on a computer

[https://github.com/fedhere/UInotebooks/blob/master/
dataWrangling/readingData.ipynb](https://github.com/fedhere/UInotebooks/blob/master/dataWrangling/readingData.ipynb)

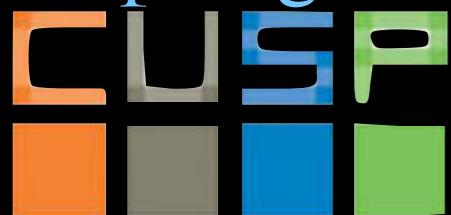


II: Data Wrangling



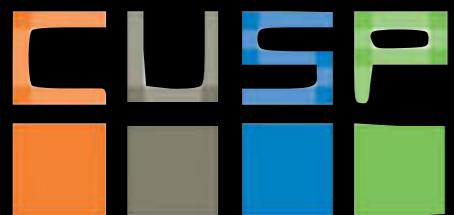
Python for Data Analysis (Pandas manual) Chapter 7

[https://github.com/fedhere/UInotebooks/blob/master/dataWrangling/
PandasDataWrangling-Chap7.ipynb](https://github.com/fedhere/UInotebooks/blob/master/dataWrangling/PandasDataWrangling-Chap7.ipynb)



II: Data Wrangling

data types



Introduction to Statistics

Online Edition

Primary author and editor: David M. Lanel

Introduction to Statistics

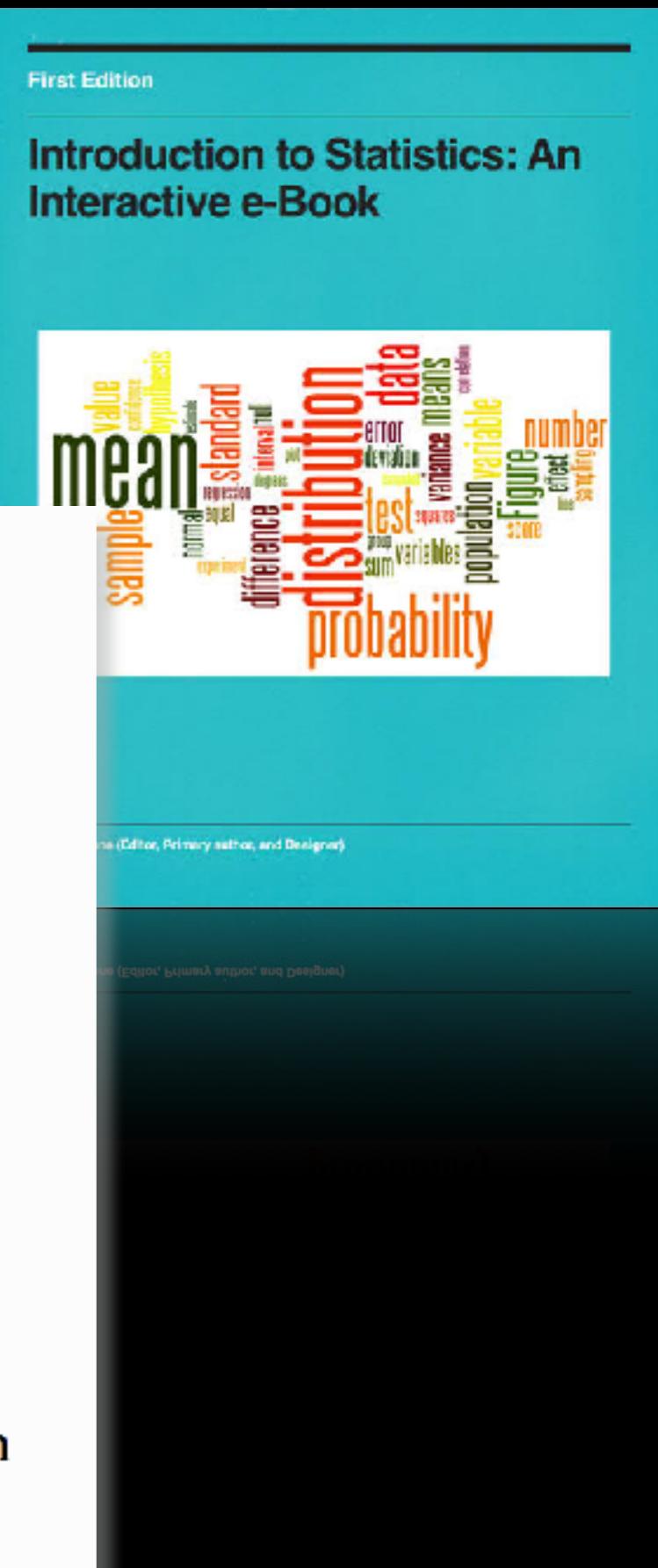
Online Edition

Primary author and editor:
David M. Lane¹

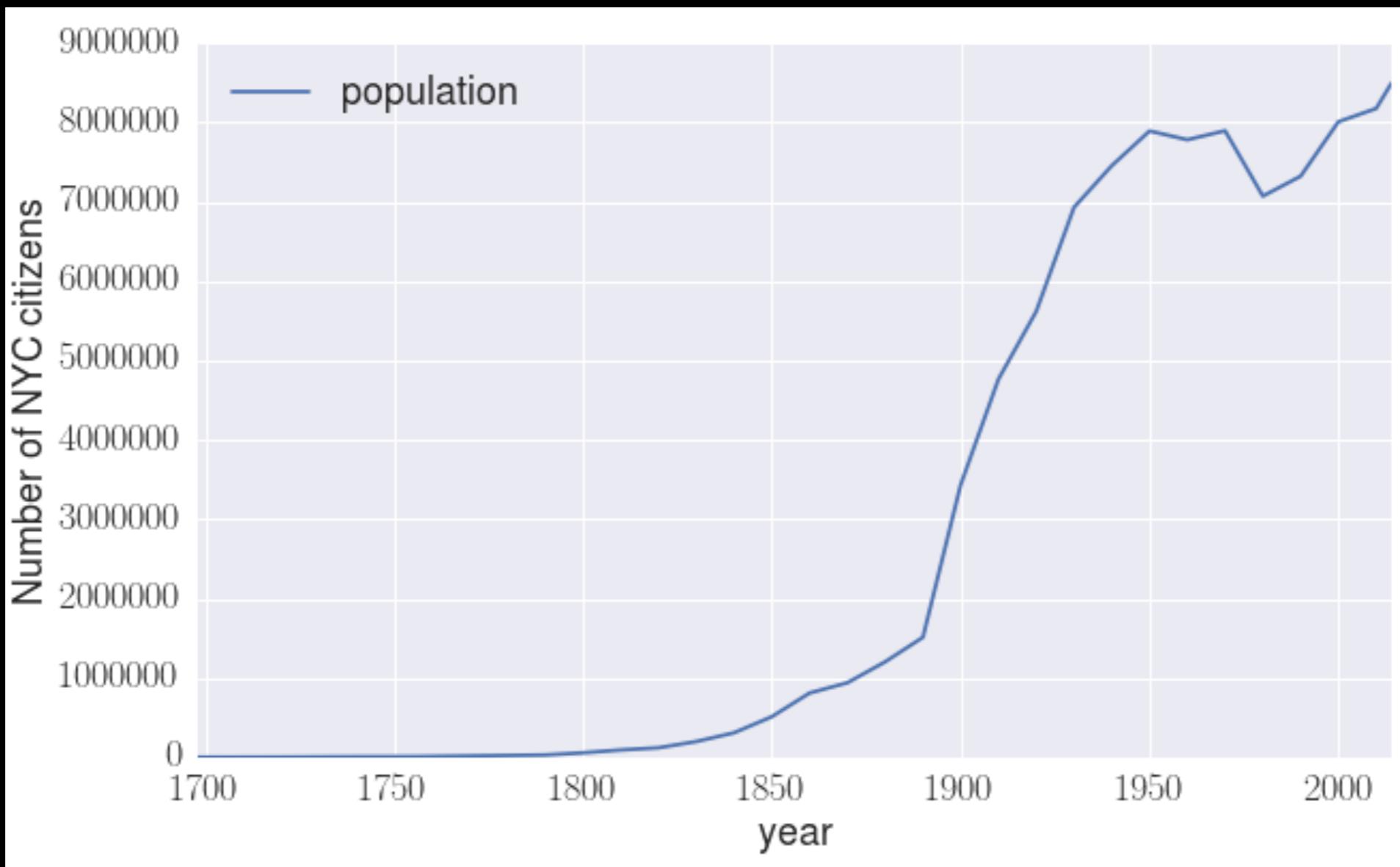
Other authors:

David Scott¹, Mikki Hebl¹, Rudy Guerra¹, Dan Osherson¹, and Heidi Zimmer²

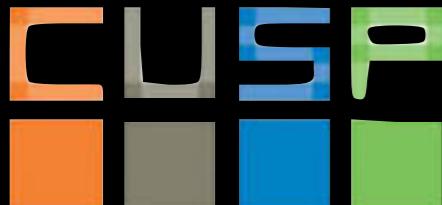
¹Rice University; ²University of Houston, Downtown Campus



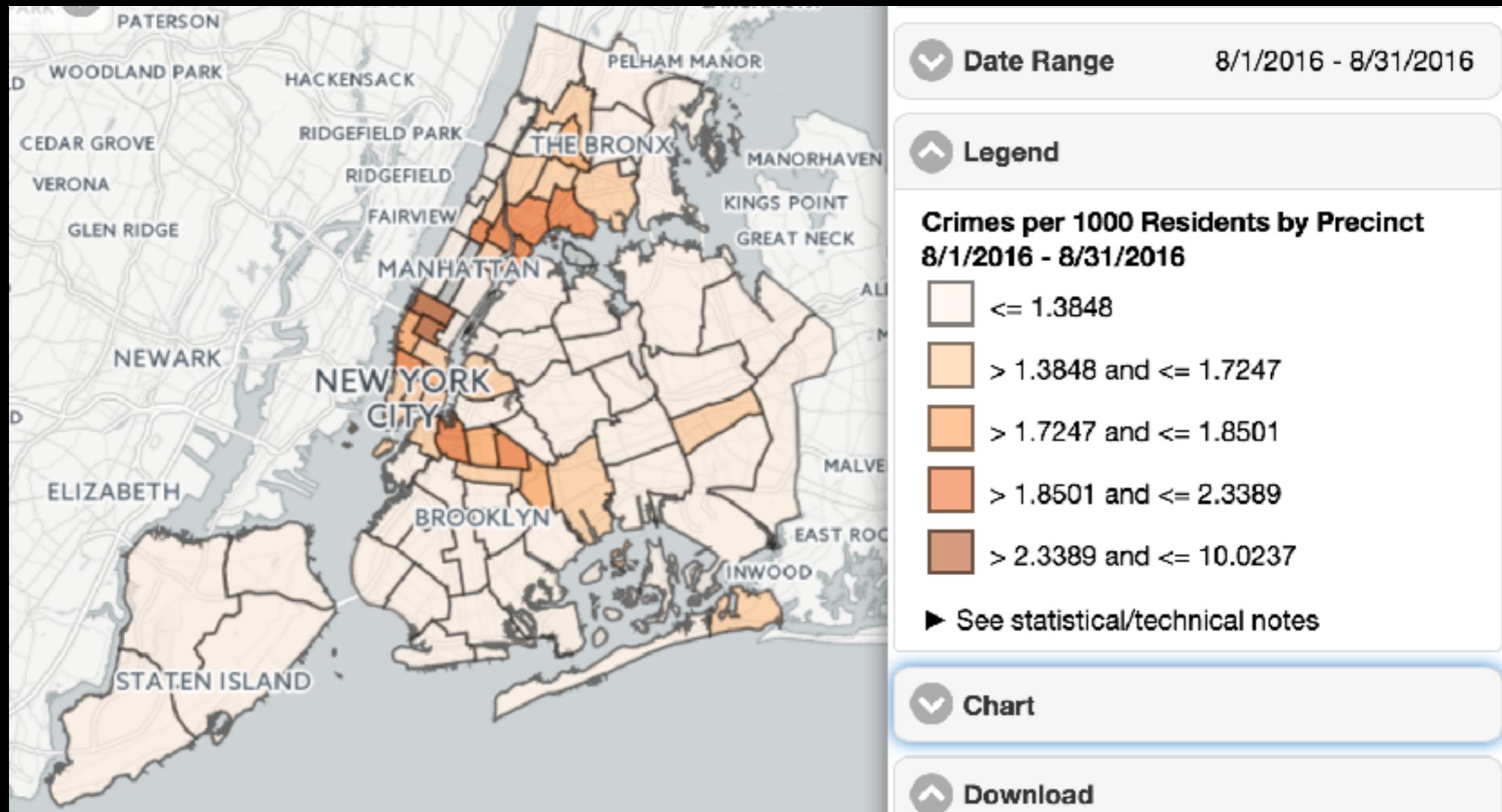
Dependent vs Independent Variable



<http://flowingdata.com/2016/09/12/watch-bacteria-evolve-resistance-to-antibiotic/>



Dependent vs Independent Variable

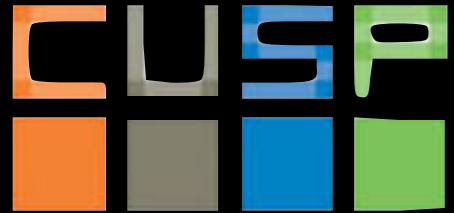


<https://maps.nyc.gov/crime/>

Dependent vs Independent Variable

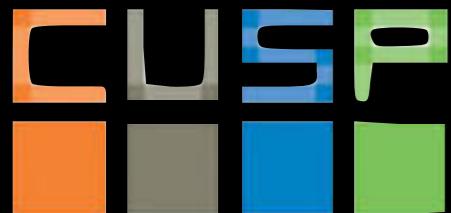


<http://flowingdata.com/2016/09/12/watch-bacteria-evolve-resistance-to-antibiotic/>



Dependent vs Independent Variable

Levels of an independent variable:
the number of experimental conditions.
e.g.: control test and experiment are 2 levels



Types of Data: *Qualitative* variables

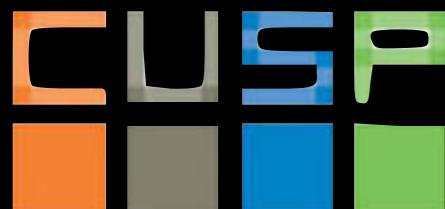
- No ordering

UrbanScience e.g. precinct, state, gender
Also called *Nominal, Categorical*

Types of Data: *Quantitative* variables

- Ordering is meaningful

Time, Distance, Age, Length, Intensity, Satisfaction



Types of Data: *Quantitative* variables

- Continuous: _____
- Discrete: • • • • • • • • • • • • • • •

Types of Data: *Quantitative* variables

- **Continuous:** distance to the closest park
- **Discrete:** *any* countable, e.g. number of crimes

Discrete data may be:

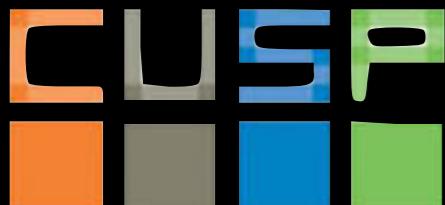
- **Counts:** number of bacteria at time t in section A
- **Ordinal:** survey response Good/Fair/Poor

Continuous data may be:

- **Continuous Ordinal:** Earthquakes (not linear scale)
- **Interval:** F temperature - interval size preserved
- **Ratio:** Car speed - 0 is naturally defined

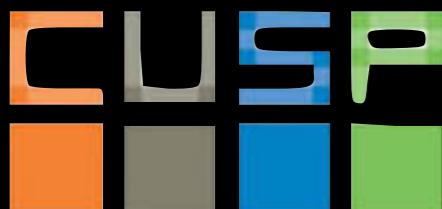
Data may also be:

- **Censored:** age > 90
- **Missing:** “Prefer not to answer” (NA / NaN)



Data Definitions

- **Data:** observations that have been collected
- **Population:** the complete body of subjects we want to infer about
- **Sample:** the subset of the population we actually studied
- **Census:** collection of data from the entire population
- **Parameter:** numerical value describing an attribute of the *population*
- **Statistics:** numerical value describing an attribute of the *sample*



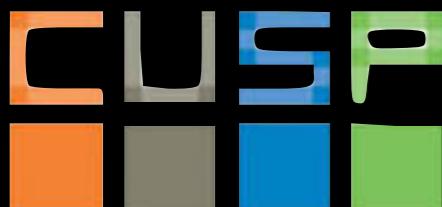
Data Definitions

The analysis of our _____
showed that for our 10 _____ the mean height is 6.4 ft.
The standard deviation of the _____ means is 0.5 ft.
From this _____ we infer for the _____ a mean
height _____ $6.4 +/ - 5$ ft

Data Definitions

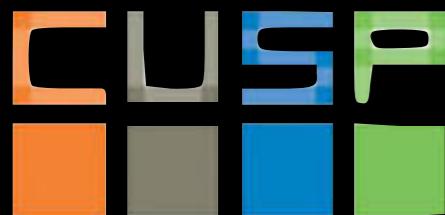
The analysis of our _____ showed that for our 10 _____ the mean height is 6.4 ft. The standard deviation of the _____ means is 0.5 ft. From this _____ we infer for the _____ a mean height _____ 6.4 ± 0.5 ft

sample parameter data population statistics



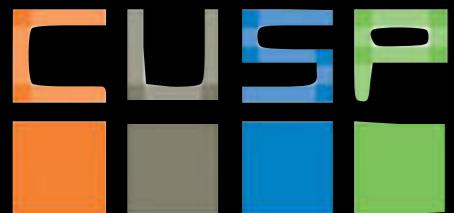
Data Definitions

The analysis of our **data** showed that for our 10 **samples** the mean height is 6.4 ft. The standard deviation of the **sample** means is 0.5 ft. From this **statistic** we infer for the **population** a mean height **parameter** 6.4 ± 0.5 ft





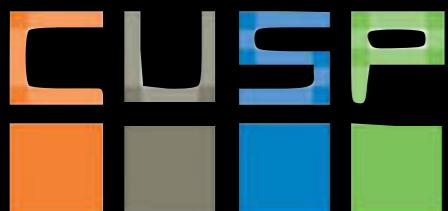
basic data inputation and
manipulation Lab:
FormattingTables.ipynb



II: Data Wrangling

How to acquire data

- Downloadable files
- API *Application Program Interface*
- Databases (MySQL, MongoDB...
Prof. Huy Vo guest lecture)



How to acquire data

- **Downloadable files**

PROS: easy to do
reproducibility: you can provide the data

CONS: store the data locally
(bad if large)

- **API Application Program Interface**

PROS: works directly with the source

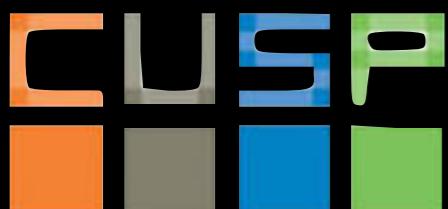
CONS: store the data locally
reproducibility: relies on data source to be permanent & API stable and accessible

- **Databases (MySQL, MongoDB...)**

Prof. Huy Vo guest lecture)

PROS: fast and flexible
can query only data of interest
reproducibility : can host remotely & provide access

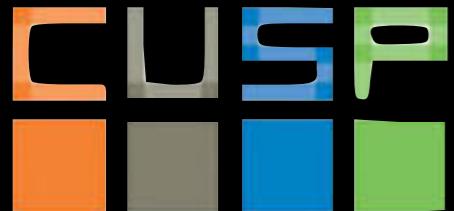
CONS: must build a database





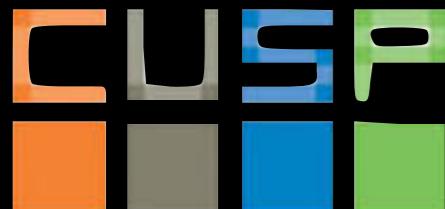
Downloading files directly
in your notebook

[https://github.com/fedhere/UInotebooks/blob/master/
dataWrangling/acquiringData.ipynb](https://github.com/fedhere/UInotebooks/blob/master/dataWrangling/acquiringData.ipynb)



Types of Data Files:

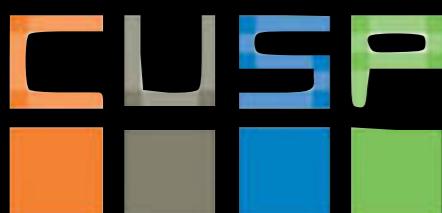
- CSV comma separated values (also TSV - tab)
- JSON: corresponds to a Python data dictionary
- XML: similar to HTML



Types of Data Files:

- CSV comma separated values (also TSV - tab)
- JSON: corresponds to a Python data dictionary
- XML: similar to HTML

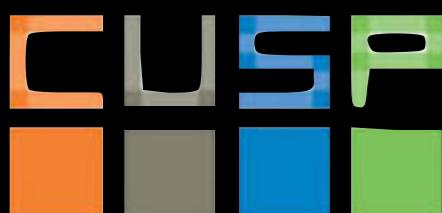
support nested structures



Types of Data Files:

- CSV comma separated values (also TSV - tab)
- JSON: corresponds to a Python data dictionary
- XML: similar to HTML

support nested structures

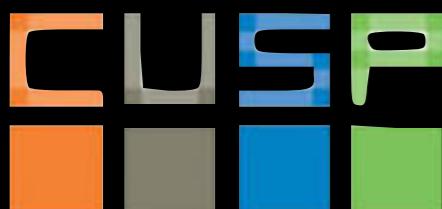


Types of Data Files:

- CSV comma separated values (also TSV - tab)
- JSON: corresponds to a Python data dictionary
- XML: similar to HTML

support nested structures

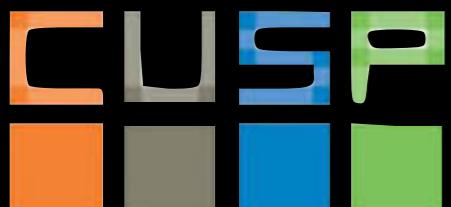
<https://dev.twitter.com/rest/tools/console>



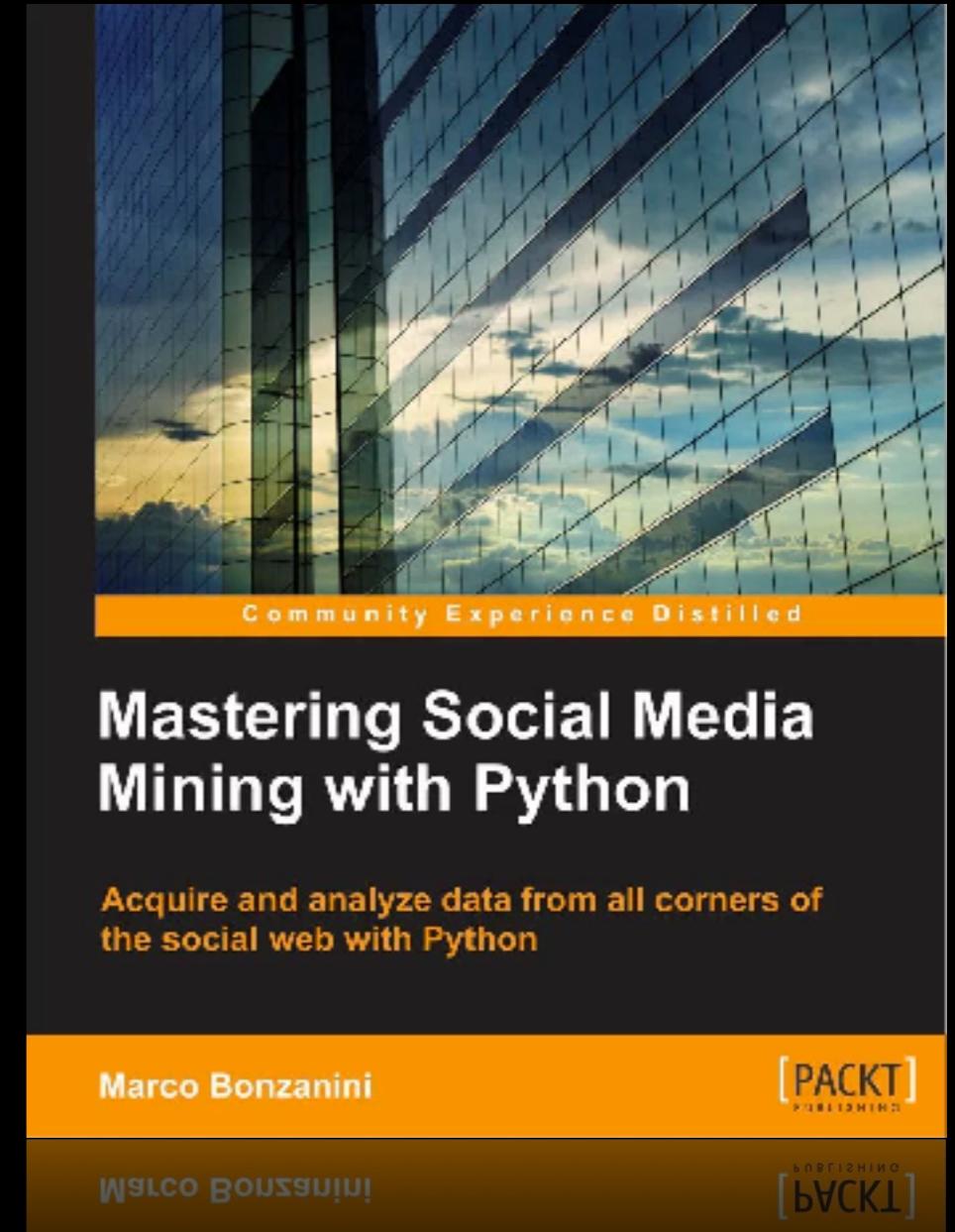
[https://github.com/fedhere/
PUI2016_fb55/blob/master/Lab2_fb55/
twitterJson.py](https://github.com/fedhere/PUI2016_fb55/blob/master/Lab2_fb55/twitterJson.py)



download an
online, nested
JSON file



<http://json.parser.online.fr/>

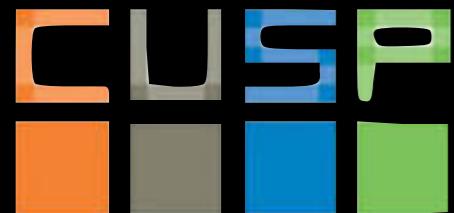


[https://marcobonzanini.com/
2015/03/02/mining-twitter-data-
with-python-part-1/](https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/)

II: Data Wrangling

Access data through API

<http://openweathermap.org/api>

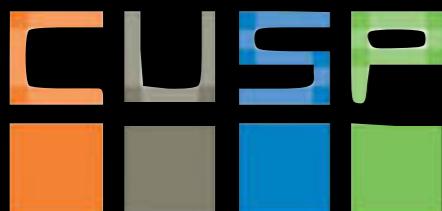




API scraping and JSON manipulation lab

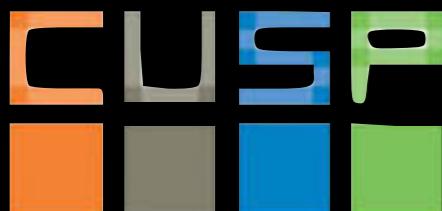
Key Concepts:

- importance of reproducibility
 - data ethics and data curation
 - types of variables
 - CSV, JSON file formats properties
-
- DATA Facility data access
 - API data access
 - JSON data manipulation
 - working with Pandas Dataframes



Key Concepts:

- importance of reproducibility
 - data ethics and data curation
 - types of variables
 - CSV, JSON file formats properties
-
- DATA Facility data access
 - API data access
 - JSON data manipulation
 - working with Pandas Dataframes



Resources:

D.Lane 2014

Introduction to Statistics Chapters 1.6-1.8 & 6.1-6.3

<http://onlinestatbook.com/>

Mohit Sharma 2016

CUSP UCSL tutorials Python Dictionaries

<https://sharmamohit.com/tutorials/ucsl/Python-Dictionaries/>

Allen B. Downey 2012

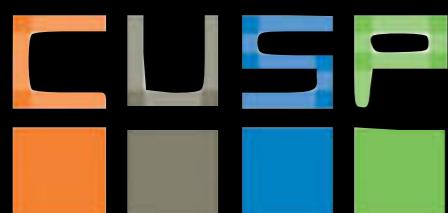
Thinking Python (free online) Appendix A ASSIGNED READING

<http://greenteapress.com/thinkpython/html/thinkpython021.html>

Wes McKinney 2013

Pandas for Data Analysis Chapter 7

<http://tinyurl.com/hydko7p>



[https://github.com/fedhere/UInotebooks/blob/master/
dataWrangling](https://github.com/fedhere/UInotebooks/blob/master/dataWrangling)

II: Data Wrangling