Research Design and *pandas*

Ivan Corneillet

Data Scientist

GA GENERAL ASSEMBLY

Learning Objectives

After this lesson, you should be able to:

- Setup and manage your personal GitHub repository for submitting assignments
- Define a problem and types of data
- Identify dataset types
- Apply the data science workflow in the *pandas* context
- Write an iPython notebook to import, format, and clean data using the *pandas* library

Announcements and Exit Tickets

Announcements and Exit Tickets

- 5/12 (session 3)
  - Guest speaker and recent alum will share his experience on his final project
- 5/24 (session 6)
  - Data Science Happy Hour with DS-SF-22 on Tuesday 5/24, 8:30PM to 9:30PM

Q & A

Python Onboarding

Wrapping Up

DS

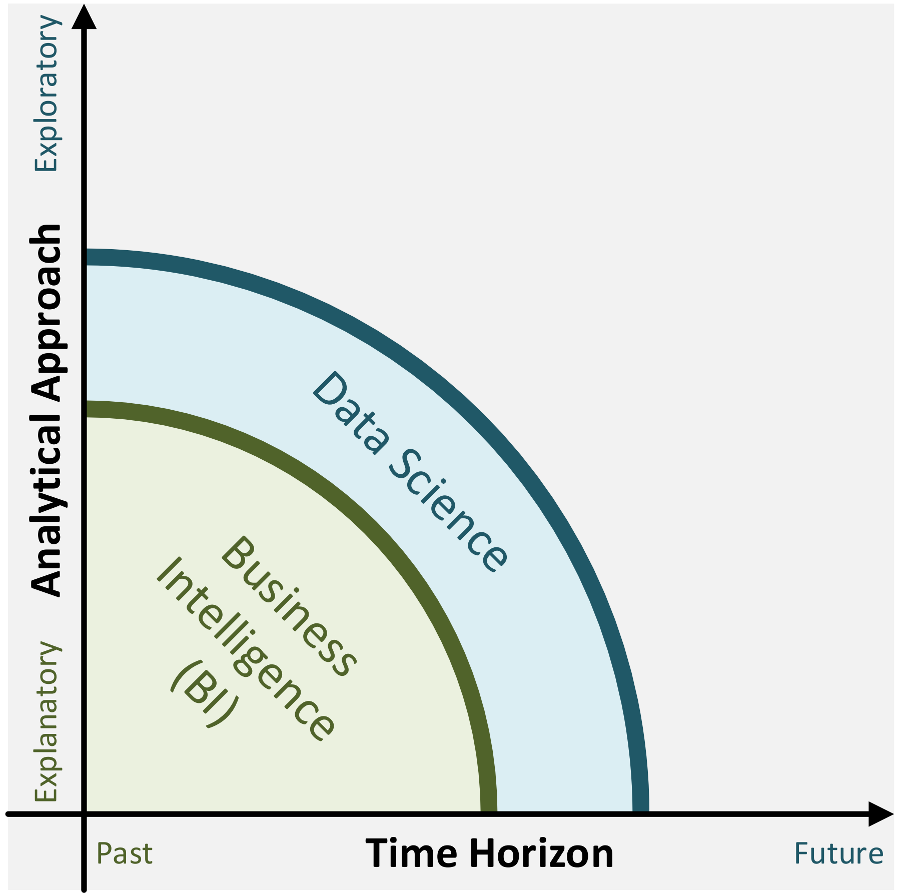
Review

DS

Review

What is Data Science?

Review | What is Data Science? (cont.)



Data Science (Data Mining and Predictive Analytics)	
Typical techniques and data types	• Statistical analysis, optimization, predictive modeling, forecasting • Structured/unstructured data, many types of sources, very large datasets
Common questions	• What if …? • What’s the optimal scenario for our business? • What will happen next? What if these trends continue? Why is it happening?

Business Intelligence (BI)	
Typical techniques and data types	• Standard and *ad-hoc* reporting, dashboards, alerts, queries, details on demand • Structured data, traditional sources, manageable datasets
Common questions	• What happened last quarter? • How many units were sold? • Where is the problem? In which situations?

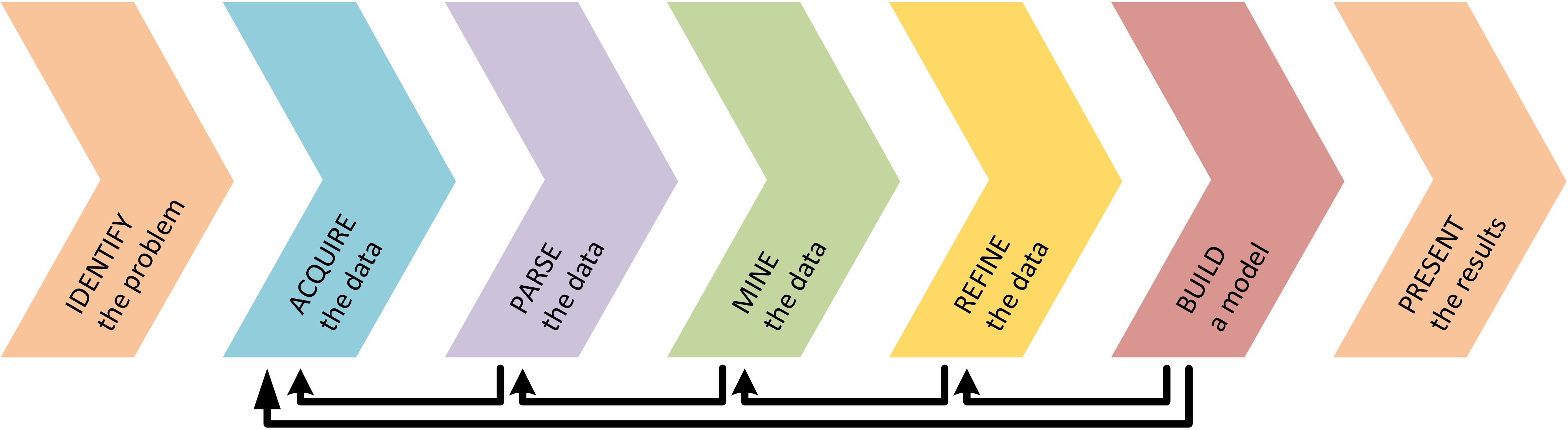
Source: Data Science and Big Data Analytics

DS

Review

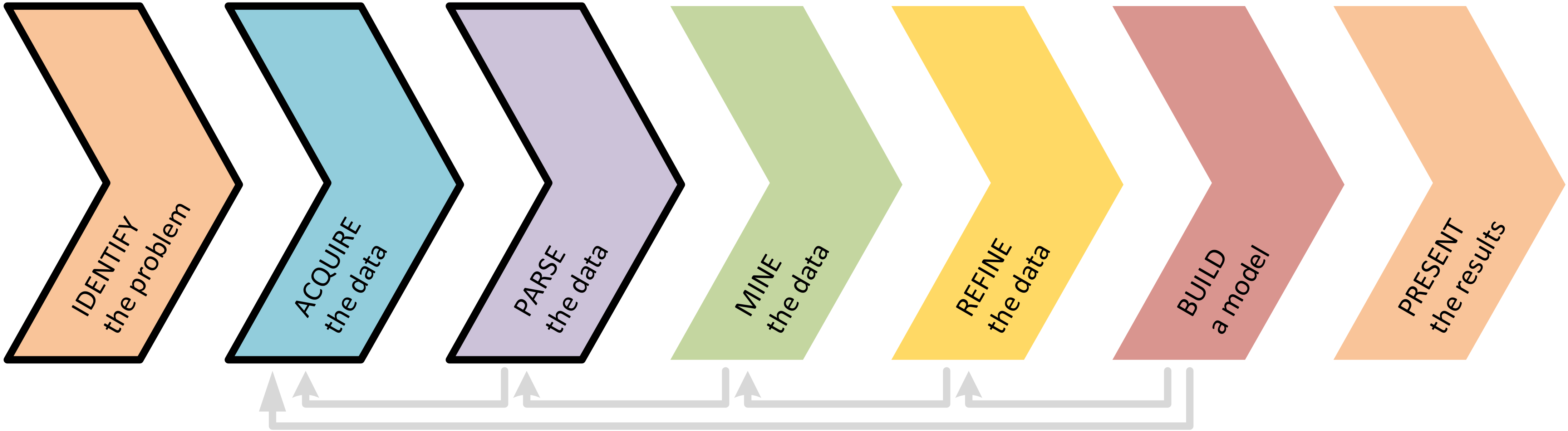
Data Science Workflow

The Data Science Workflow



Today

Today we'll focus on the first three (IDENTIFY the problem, ACQUIRE the data, and PARSE the data)



Today, we are covering Research Design and introducing the *pandas* library

Research Design and Data Analysis	Research Design	Data Visualization in *pandas*	Statistics	Exploratory Data Analysis in *pandas*
Foundations of Modeling	Linear Regression	Classification Models	Evaluating Model Fit	Presenting Insights from Data Models
Data Science in the Real World	Decision Trees and Random Forests	Time Series Data	Natural Language Processing	Databases

Here’s what’s happening today:

- Announcements and Exit Tickets
- Review
- Pre-Work
- Git and GitHub Primer
- ❶ Identify the problem
  - The Why’s and How’s of a Good Question
  - The SMART Goals Framework
- ❷ Acquire the Data
  - Data Types
  - Logistics of Acquiring Data
  - SF Housing Dataset from Zillow
  - Tidying Data
  - File Formats
- ❸ Parse the Data
  - Documentation and Data Dictionaries
  - Codealong – Introduction to *pandas*
  - Codealong – Tidying up the SF housing dataset
- Unit Project 1
- Lab – Introduction to *pandas*
- Review
- **Unit Project 1 (due next session on 5/12)**

Pre-Work

Pre-Work

- Have completed the onboarding pre-work
- Install either GitHub Desktop or SourceTree (really a matter of which one you prefer) on your laptop
  - These applications provide a GUI to interact with Git repositories; they also seamlessly setup SSH keys (to authenticate you for your own repositories) should you decide to use the `git` command line tool
- Start looking first unit project
  - Unit Project 1 (due next session on 5/12)

DS

Q & A

Git and GitHub Primer

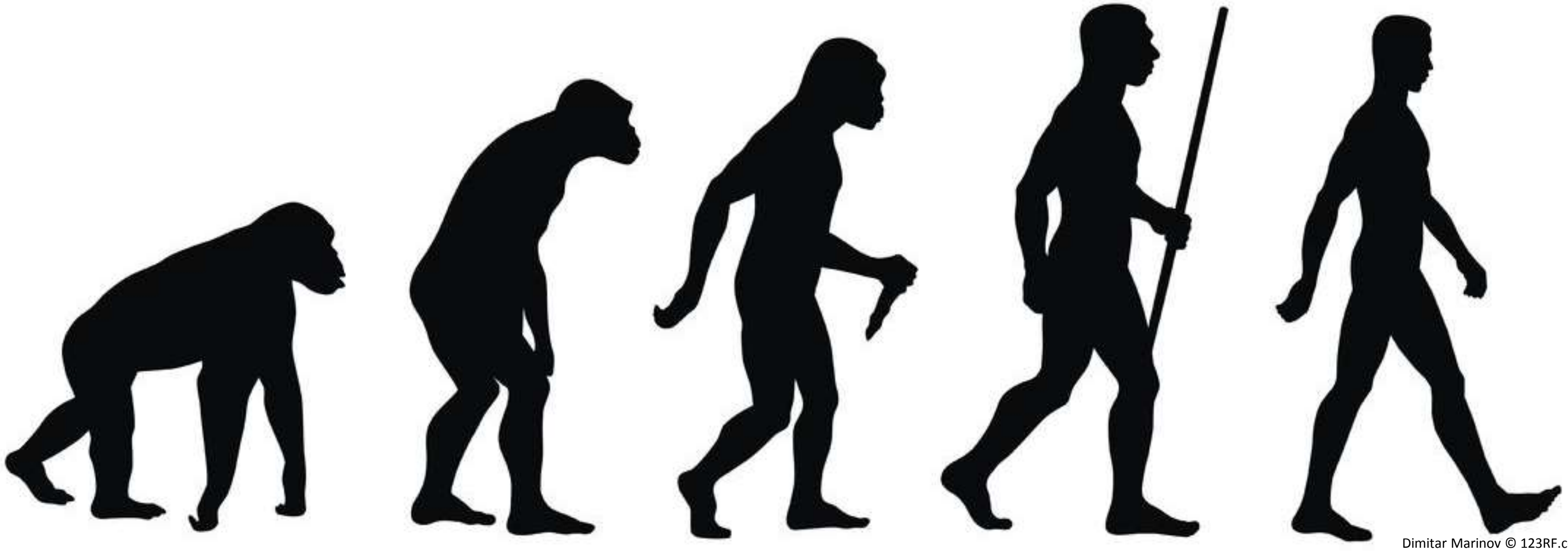
yupiramos © 123RF.com

MY COMPUTER HARD DRIVE CRASHED

THE NSA WON’T SEND ME THEIR BACKUP COPY

(http://techland.time.com/2013/08/30/no-the-nsa-wont-restore-your-crashed-hard-drive)

Version Control (cont.)



Dimitar Marinov © 123RF.com

Nothing

(incidentally this includes the NSA strategy...)

Copy and paste files and directories

(it quickly becomes a giant mess)

Time Machine

(great at home to track individual files but it isn't geared for tracking a collection of files that change together)

Old-School Version Control Systems (a.k.a., VCS)

`git`, a modern Version Control System

Why should you learn Git/GibHub?

- Git (or another modern VCS) is essential when you write code; as a data scientist you write code. It was created by programmers for programmers to enable them to collaborate on the same codebase

- GitHub is a popular social coding website and provides a Git repository web-based hosting service. GitHub provides easy-to-use web-based, desktop, and mobile applications as well as access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project

- GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub account in order to qualify for a job

Source: Wikipedia

GitHub Desktop/SourceTree GitHub/Bitbucket

GitHub Desktop/SourceTree and GitHub/Bitbucket

- We will use GitHub for the course repository but another popular option is BitBucket
  - One notable difference: Bitbucket offers unlimited free private repositories while GitHub charges for them
- Feel free to use either one for your projects

- GitHub Desktop and SourceTree are desktop applications providing a similar UX to interact with Git repositories
  - Even if you only use the command line tool they seamlessly setup your SSH keys (to authenticate you against your own repositories) so your don’t have to
- GitHub Desktop and SourceTree work on both GitHub and Bucket repositories

Step ⓿ | Open a terminal window to check that you are correctly authenticated

- To commit code into GitHub/BitBucket, you need to be authenticated (using SSH keys): Open a terminal and check the output of the following command:

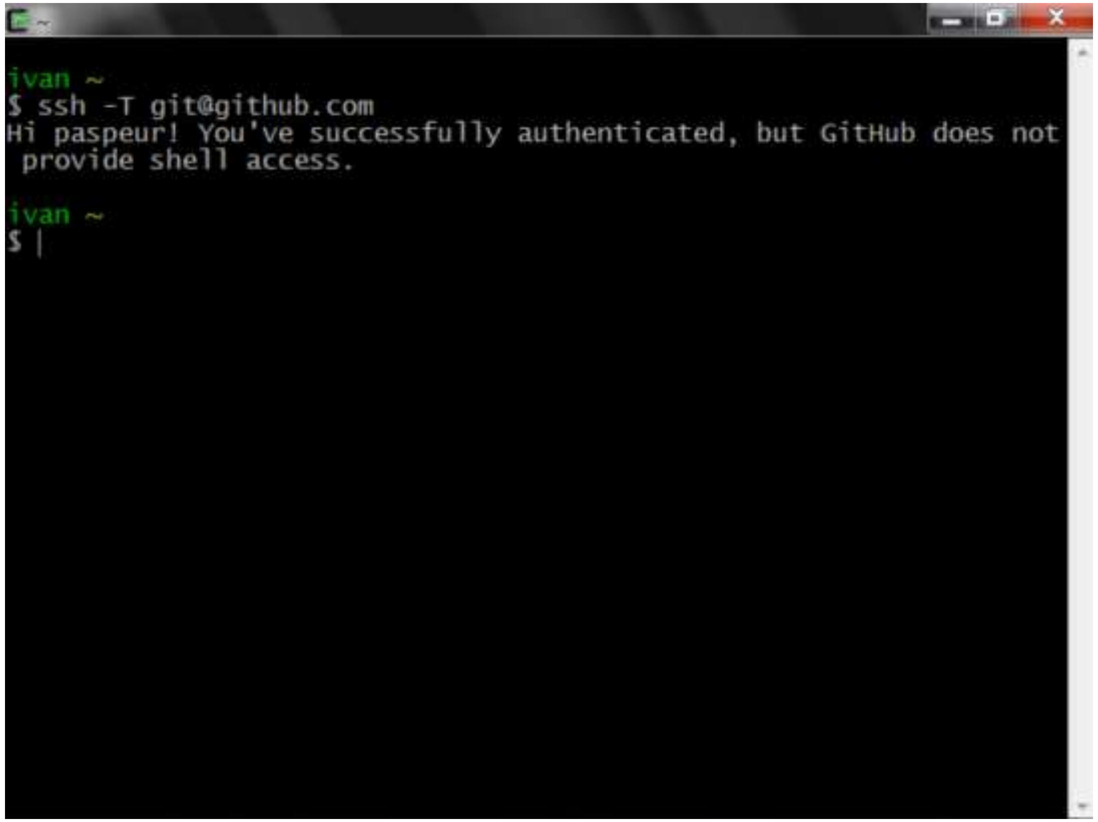
`ssh -T git@github.com` *(for GitHub)*

`ssh -T git@bitbucket.org` *(for BitBucket)*

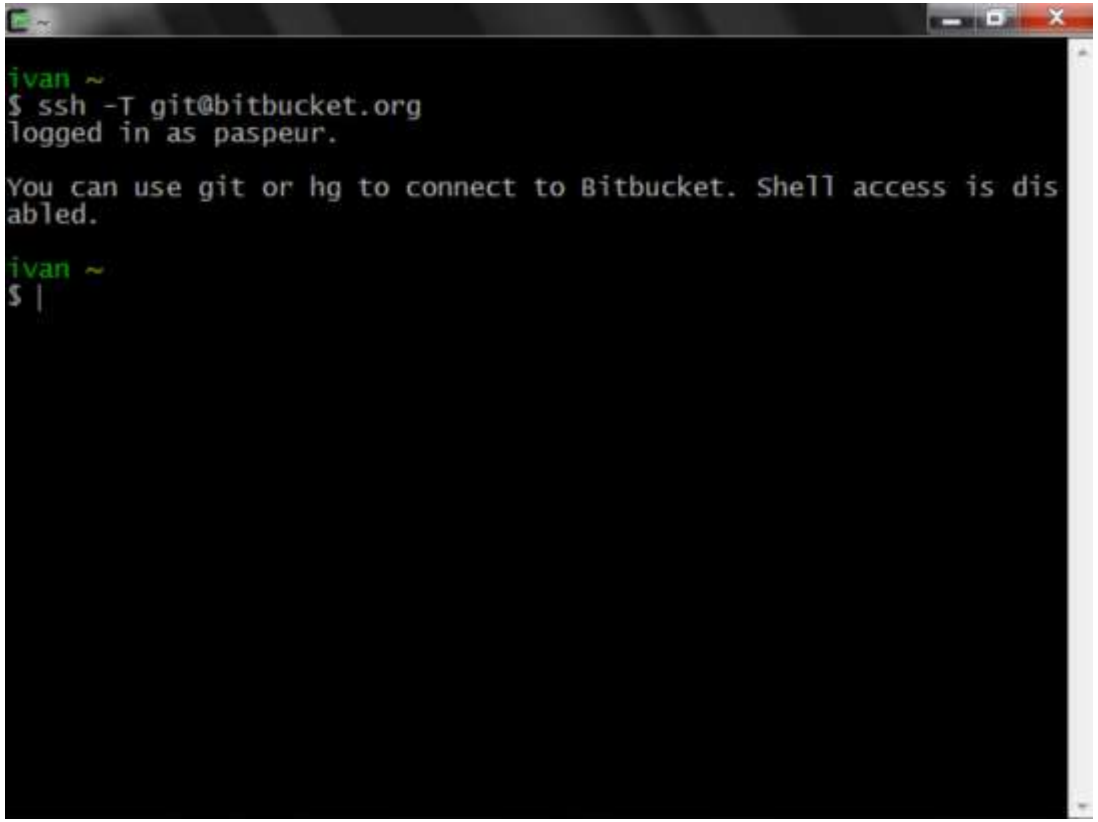
- If you get the following message along these lines, you are good to go:

```
Hi paspeur! You've successfully authenticated, but GitHub does not
provide shell access.
```

Step ⓿ | Open a terminal window to check that you are correctly authenticated (cont.)



```
ivan ~
$ ssh -T git@github.com
Hi paspeur! You've successfully authenticated, but GitHub does not
 provide shell access.

ivan ~
$ |
```



```
ivan ~
$ ssh -T git@bitbucket.org
logged in as paspeur.

You can use git or hg to connect to Bitbucket. Shell access is dis
abled.

ivan ~
$ |
```

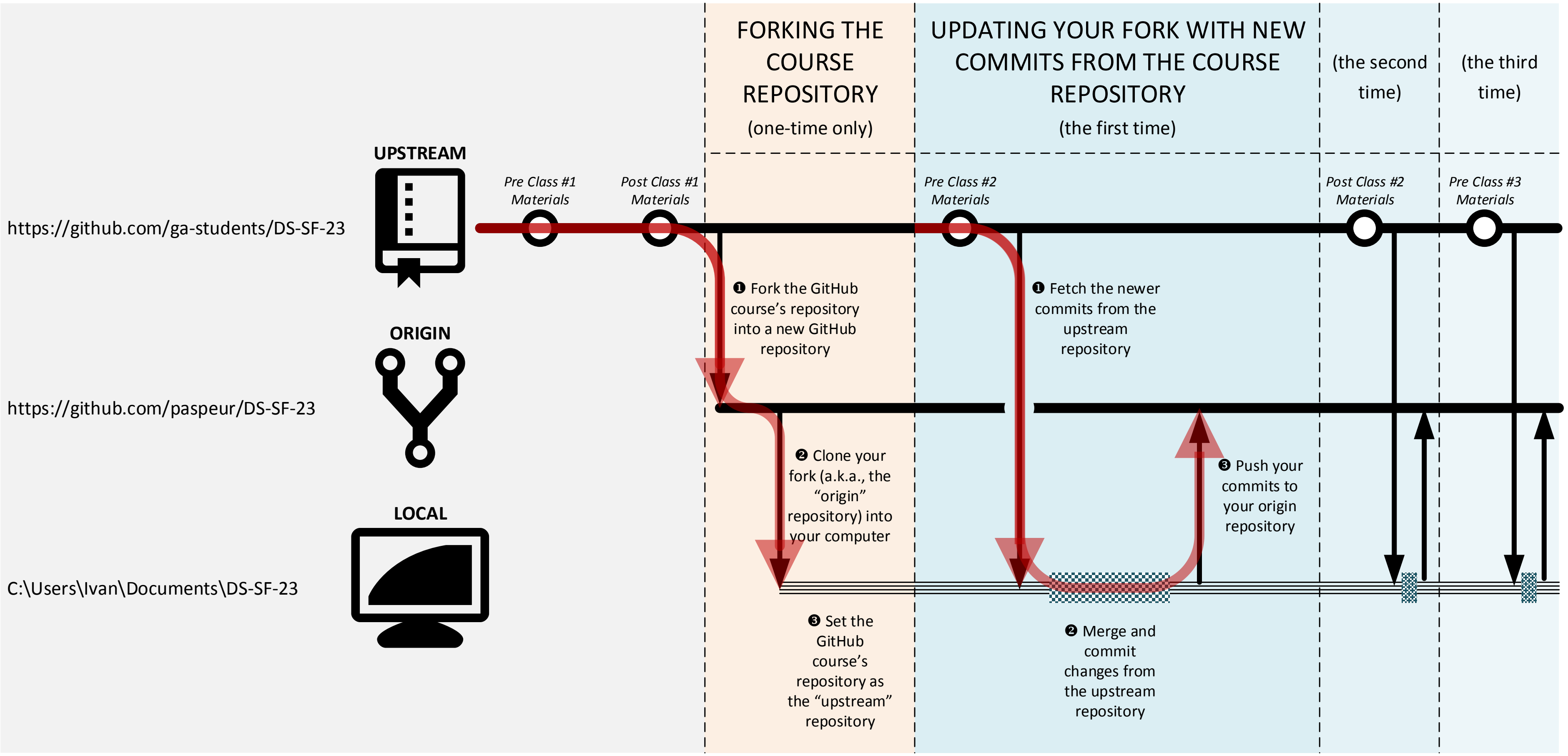
Step ➓’ | Set some Git configuration

- `git config --global push.default simple`

Git and GitHub Primer

Practice #1 | Fork the course repository; clone your fork; update your fork and clone as needed

Practice #1 | Fork the course repository; clone your fork; update your fork and clone as needed (cont.)


FORKING THE COURSE REPOSITORY
(one-time only)
UPDATING YOUR FORK WITH NEW COMMITS FROM THE COURSE REPOSITORY
(the first time)
(the second time)
(the third time)
UPSTREAM
https://github.com/ga-students/DS-SF-23
Pre Class #1 Materials
Post Class #1 Materials
Pre Class #2 Materials
Post Class #2 Materials
Pre Class #3 Materials
ORIGIN
https://github.com/paspeur/DS-SF-23
LOCAL
C:\Users\Ivan\Documents\DS-SF-23
❶ Fork the GitHub course's repository into a new GitHub repository
❷ Clone your fork (a.k.a., the "origin" repository) into your computer
❸ Set the GitHub course's repository as the "upstream" repository
❶ Fetch the newer commits from the upstream repository
❷ Merge and commit changes from the upstream repository
❸ Push your commits to your origin repository

Practice #1 | How to fork the course repository (one-time only)

- ❶ Fork the GitHub course's repository into a new GitHub repository
  - Login into GitHub; open https://github.com/ga-students/DS-SF-23; click on the Fork button on the top right; your fork is at https://github.com/paspeur/DS-SF-23 (replace *paspeur* with your username)
- ❷ Clone your fork (a.k.a., the "origin" repository) into your computer
  - Open a terminal window; type "`git clone https://github.com/paspeur/DS-SF-23`"; your clone is under the DS-SF-23 folder (type "`cd DS-SF-23`" to change the current directory to your clone's root directory)
- ❸ Set the GitHub course's repository as the "upstream" repository
  - With a terminal window, `cd` to your clone's root directory; type "`git remote add upstream https://github.com/ga-students/DS-SF-23`"

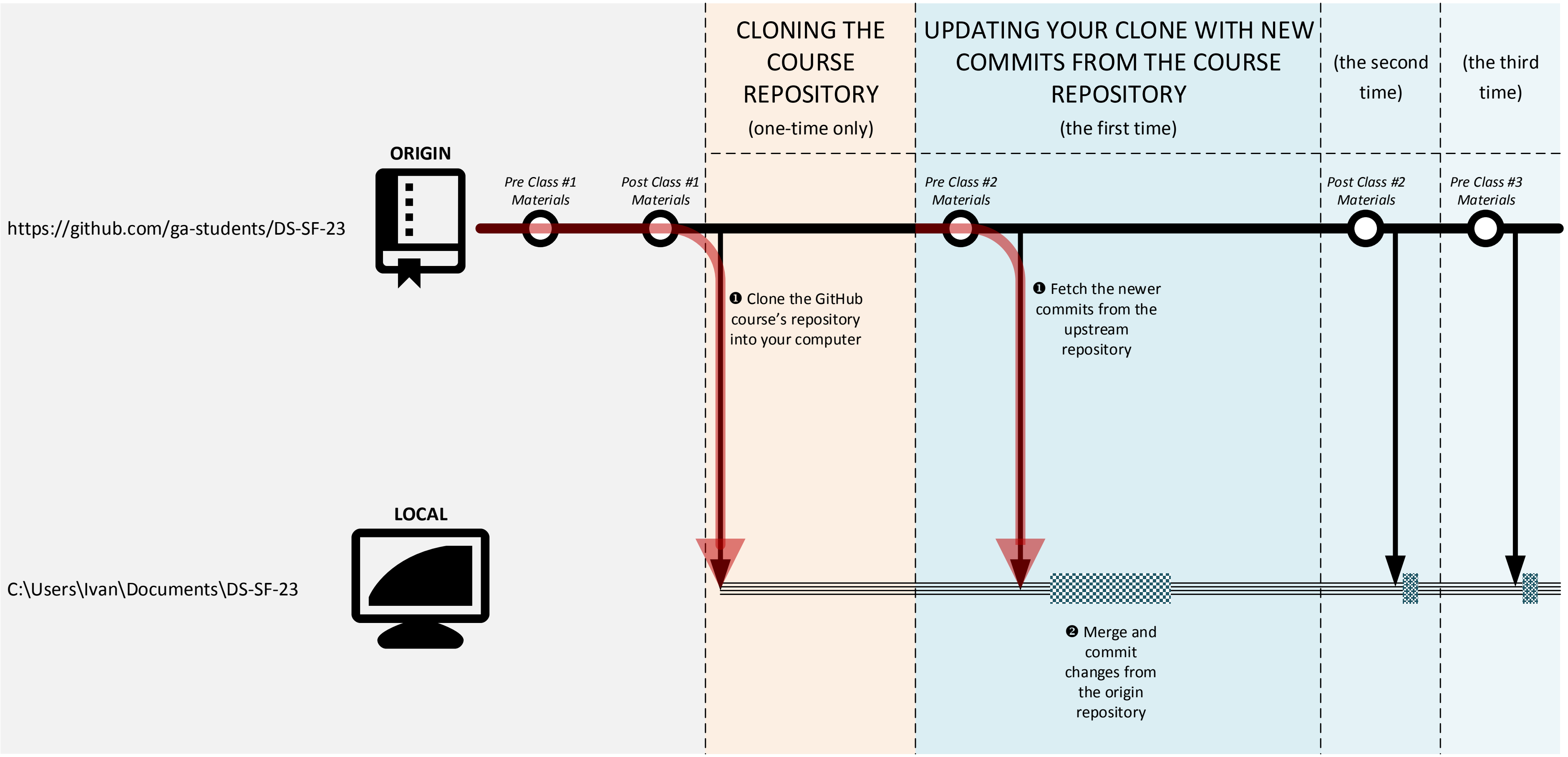
Practice #1 | How to update your fork and clone with new commits from the course repository

- ❶ Fetch the newer commits from the upstream repository
  - `git fetch upstream`
- ❷ Merge and commit changes from the upstream repository
  - `git merge upstream/master`
  - `git commit –m "Merged commits from ga-students/SF-DAT-23 up to 2016-05-10"` (if the merge was “Fast-forward”, i.e., trivial, there is no need to commit these changes)
- ❸ Set the GitHub course’s repository as the “upstream” repository
  - `git push` (Git might ask you your GitHub credentials the first time around)

Git and GitHub Primer

Practice #2 | Clone the course repository; update your clone as needed

Practice #2 | Clone the course repository; update your clone as needed


CLONING THE COURSE REPOSITORY
(one-time only)
UPDATING YOUR CLONE WITH NEW COMMITS FROM THE COURSE REPOSITORY
(the first time)
(the second time)
(the third time)
ORIGIN
https://github.com/ga-students/DS-SF-23
Pre Class #1 Materials
Post Class #1 Materials
Pre Class #2 Materials
Post Class #2 Materials
Pre Class #3 Materials
❶ Clone the GitHub course's repository into your computer
❶ Fetch the newer commits from the upstream repository
LOCAL
C:\Users\Ivan\Documents\DS-SF-23
❷ Merge and commit changes from the origin repository

Practice #2 | How to clone the course repository (one-time only)

- ❶ Clone the GitHub course's repository (a.k.a., the "origin" repository) into your computer
  - Open a terminal window; type "`git clone https://github.com/ga-students/DS-SF-23`"; your clone is under the DS-SF-23 folder (type "`cd DS-SF-23`" to change the current directory to your clone's root directory)

Practice #2 | How to update your clone with new commits from the course repository

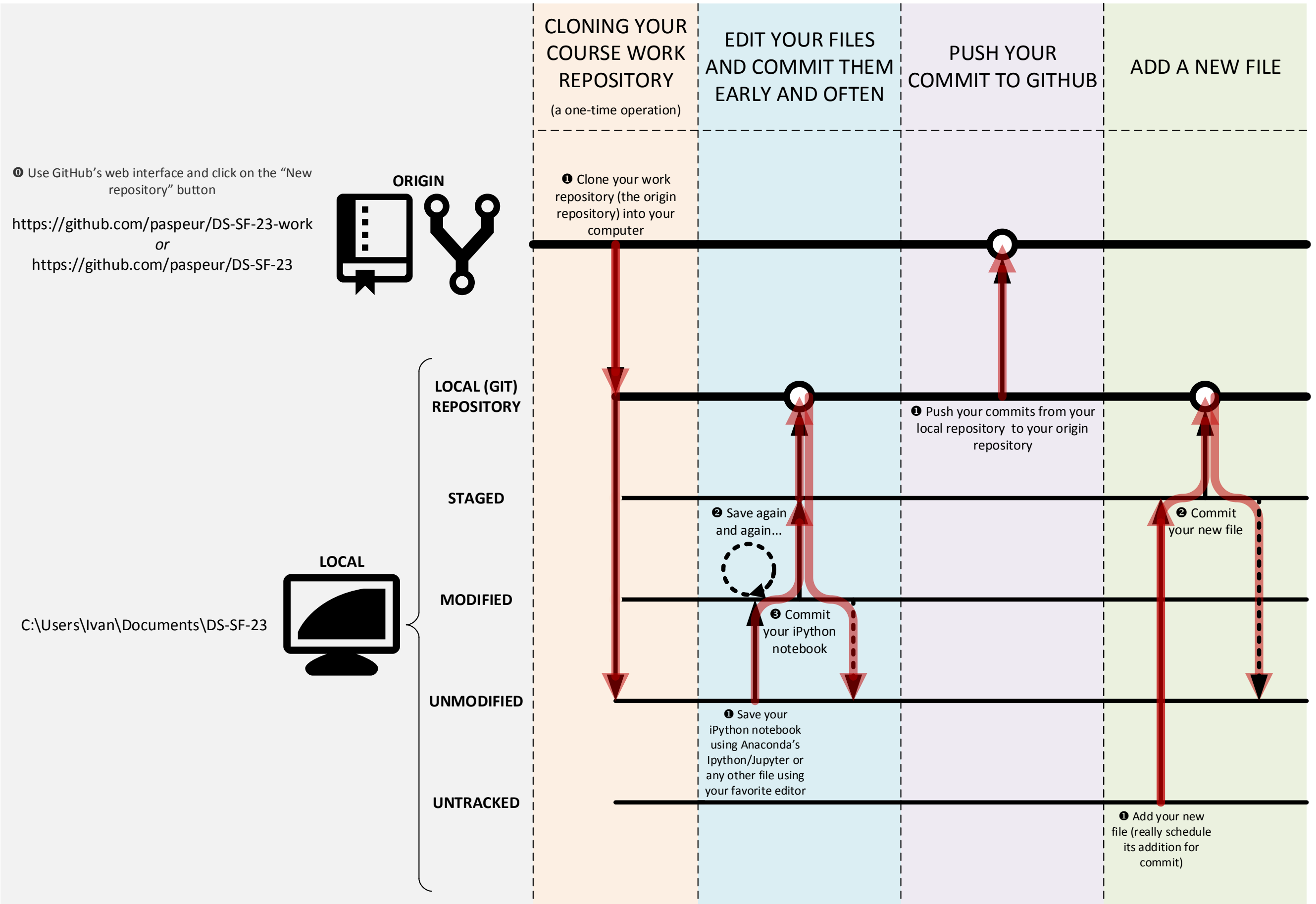
- ❶ Fetch the newer commits from the origin repository
  - `git fetch`
- ❷ Merge and commit changes from the upstream repository
  - `git merge`
  - `git commit –m "Merged commits from ga-students/SF-DAT-23 up to 2016-05-10"`
- Note: "`git pull`" combines "`git fetch`" and "`git merge`" as a single command

DS

Git and GitHub Primer

Practice #3 | Create your own work repository (to reuse your forked course repository) for working on your projects and for submitting them

Practice #3 | Create your own work repository (to reuse your forked course repository) for working on your projects and for submitting them



CLONING YOUR COURSE WORK REPOSITORY

(a one-time operation)

EDIT YOUR FILES AND COMMIT THEM EARLY AND OFTEN

PUSH YOUR COMMIT TO GITHUB

ADD A NEW FILE

⓿ Use GitHub's web interface and click on the "New repository" button

https://github.com/paspeur/DS-SF-23-work
or
https://github.com/paspeur/DS-SF-23

ORIGIN

❶ Clone your work repository (the origin repository) into your computer

LOCAL (GIT) REPOSITORY

STAGED

LOCAL

MODIFIED

C:\Users\Ivan\Documents\DS-SF-23

UNMODIFIED

UNTRACKED

❶ Save your iPython notebook using Anaconda's Ipython/Jupyter or any other file using your favorite editor

❷ Save again and again...

❸ Commit your iPython notebook

❶ Push your commits from your local repository to your origin repository

❶ Add your new file (really schedule its addition for commit)

❷ Commit your new file

Practice #3 | How to create your own course work repository *(optional; you can reuse your fork of the course repository as well)*

- ⓿ Use GitHub’s web interface and click on the “New repository” button (or use this link: https://github.com/new)
  - There is no hard rule but we suggest “DS-SF-23-work” for the name of your work repository
  - Choose “Python” as your “Add .ignore:” option
- ❶ Clone your work repository (the origin repository) into your computer
  - `git clone https://github.com/paspeur/DS-SF-23-work`

Practice #3 | How to edit your files and commit them

- ❶ Save your iPython notebook using Anaconda's IPython/Jupyter Notebook
  - (or any other file using your favorite editor)
- ❷ Save again and again…
- ❸ Commit your iPython notebook

  -
    ```
    git commit –m "a descriptive message so you know what this change is about in 3 months or next week..." classes/02/code/codealong-02-introduction-to-pandas.ipynb
    ```

- Note: Commit early and often (http://blog.codinghorror.com/check-in-early-check-in-often)

Practice #3 | Your commits are stored in your local repository (on your laptop). How to update your remote (GitHub) repository

- ❶ Push your commits from your local repository to your origin repository (your remote GitHub reposirory)
  - `git push`
  - (Git may complain that there are changes in your origin repository that you need to synchronize first on your local repository; check the `git fetch/merge/pull` slide from the course repository)

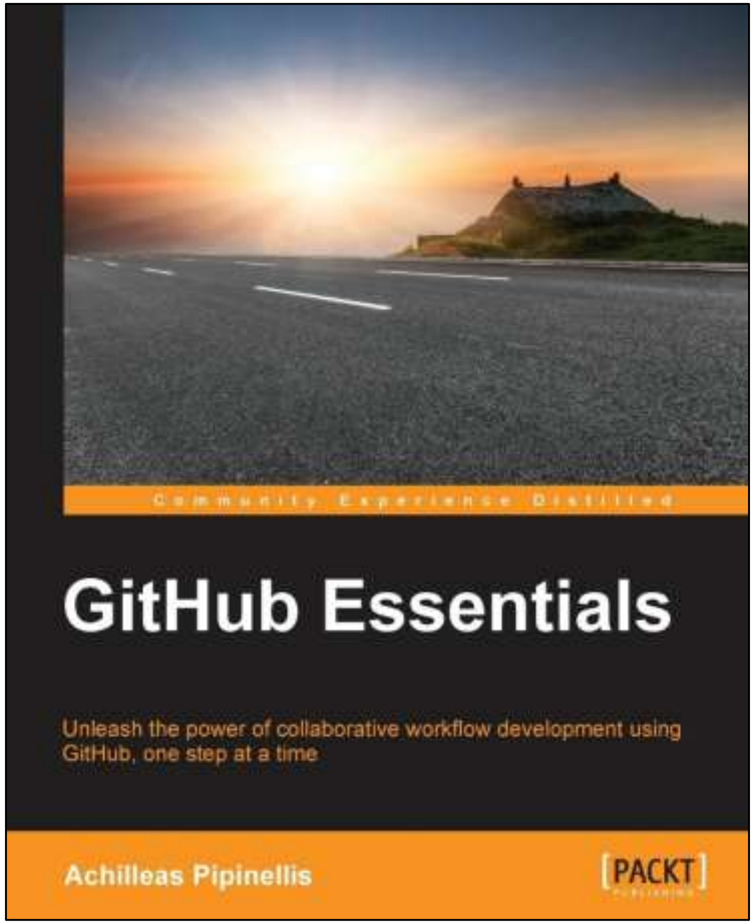
Practice #3 | How to add a new file

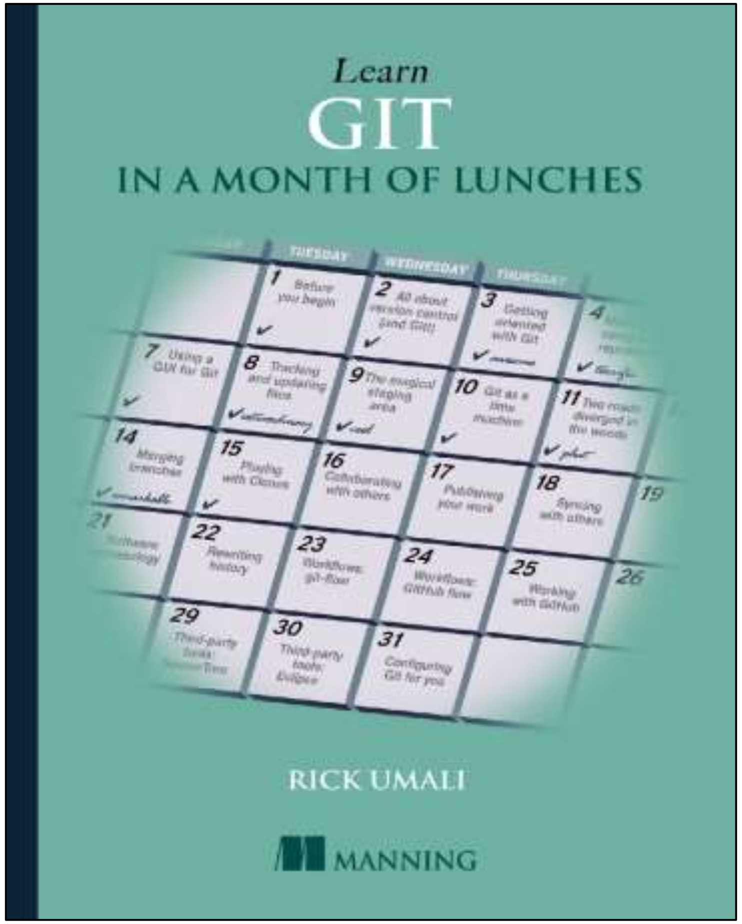
- ❶ Add your new file (really schedule its addition for commit)

  -
    ```
    git add classes/02/code/introduction-to-pandas-notes.ipynb
    ```
- ❷ And commit it

  -
    ```
    git commit –m "another descriptive message so you know what you did here in 3 months or next week..." classes/02/code/introduction-to-pandas-notes.ipynb
    ```

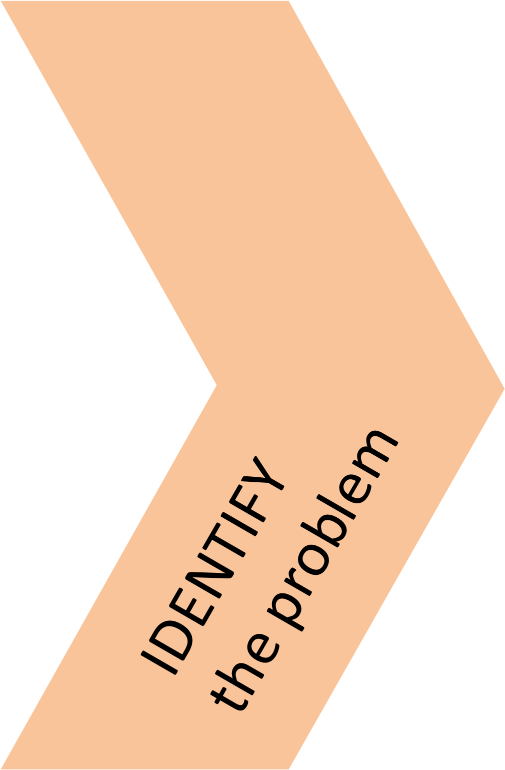
A couple of resources to get started with Git *(optional; not required for the course)* (the styles are different but the content overlaps so only pick one if any)


Community Experience Distilled
GitHub Essentials
Unleash the power of collaborative workflow development using GitHub, one step at a time
Achilleas Pipinellis
PACKT PUBLISHING


Learn
GIT
IN A MONTH OF LUNCHES
TUESDAY
WEDNESDAY
THURSDAY
1 Before you begin
2 All about version control (and Git)
3 Getting oriented with Git
4
7 Using a GUI for Git
8 Tracking and updating files
9 The magical staging area
10 Git as a time machine
11 Two roads diverged in the woods
14 Merging branches
15 Playing with Clones
16 Collaborating with others
17 Publishing your work
18 Syncing with others
19
21
22 Rewriting history
23 Workflows: git-flow
24 Workflows: GitHub flow
25 Working with GitHub
26
29 Third-party tools: SourceTree
30 Third-party tools: Eclipse
31 Configuring Git for you
RICK UMALI
MANNING

➊ IDENTIFY the Problem

❶ Identify the Problem



- Identify the Problem
  - Identify business/product objectives
  - Identify and hypothesize goals and criteria for success
  - Create a set of questions for identifying correct dataset

❶ Identify the Problem (cont.)

- Identify the Problem
  - Identify business/product objectives
  - Identify and hypothesize goals and criteria for success
  - Create a set of questions for identifying correct dataset

- The Why's and How's of a Good Question
- The SMART Goals Framework

❶ IDENTIFY the Problem

The Why's and How's of a Good Question

Why do we need a good question?

- “The scientist is not a person who gives the right answers, he's one who asks the right questions.” – Claude Lévi-Strauss

- “If they can get you asking the wrong questions, they don't have to worry about answers.” – Thomas Pynchon

- “Judge a man by his questions rather than by his answers.” – Voltaire

By asking a good question and setting a clear aim:

- You set yourself up for success
  - "A problem well stated is half solved" – Charles Kettering
- You help other data scientists learn from and reproduce your work
  - You establish the basis for making your analysis reproducible
- You also help them expand on your work in the future

➊ IDENTIFY the Problem

The SMART Goals Framework

The SMART Goals Framework provides a good foundation to set a clear aim

SPECIFIC	Who, What, Where, When, Why, Which	Define the goal as much as possible, with no ambiguous language. WHO is involved, WHAT do I want to accomplish, WHERE will it be done, WHY am I doing this — reasons, purpose, WHICH constraints and/or requirements do I have?
MEASURABLE (MEANINGFUL)	From and To	Can you track the progress and measure the outcome? How much, how many, how will I know when my goal is accomplished?
ATTAINABLE (ACTION ORIENTED)	How	Is the goal reasonable enough to be accomplished? How so? Make sure the goal is not out reach or below standard performance
RELEVANT (REALISTIC)	Worthwhile	Is the goal worthwhile and will it meet your needs? Is each goal consistent with other goals you have established and fits with your immediate and long term plans?
TIMELY (TIME-BOUND)	When	Your objective should include a time limit. "I will complete this step by month/day/year." It will establish a sense of urgency and prompt you to have better time management

The SMART Framework tuned up for Data Science:

SPECIFIC	The dataset and key variables are clearly defined
MEASURABLE	The type of analysis and major assumptions are articulated
ATTAINABLE	The question you are asking is feasible for your dataset and is not likely to be biased
REPRODUCIBLE	Another person (or you in 6 months!) can read your state and understand exactly how your analysis is performed
TIME-BOUND	You clearly state the time period and population for which this analysis will pertain

Trends often change over time and vary by the population of source of your data. It is important to clearly define who/what you included in your analysis as well as the time period for the analysis

❶ IDENTIFY the Problem

Activity | A SMART Goal for Your Final Project

Activity | A SMART Goal for Your Final Project

EXERCISE

DIRECTIONS (10 minutes)

1. After the first class, you probably started brainstorming on an idea for your final project. If not, here's an opportunity!
2. If your idea is cool and interesting, that's great. But it is a SMART idea?
3. Assess your idea using the Data Science-tuned SMART Framework
    a. If you have just a couple of gaps, how can you close them?
    b. On the other end, if you have too many and closing these gaps would be difficult, you might want to consider something else
4. After 5 minutes, share your idea and gaps in pairs and offer advise to each other, again using the SMART Framework (2.5 minutes each)

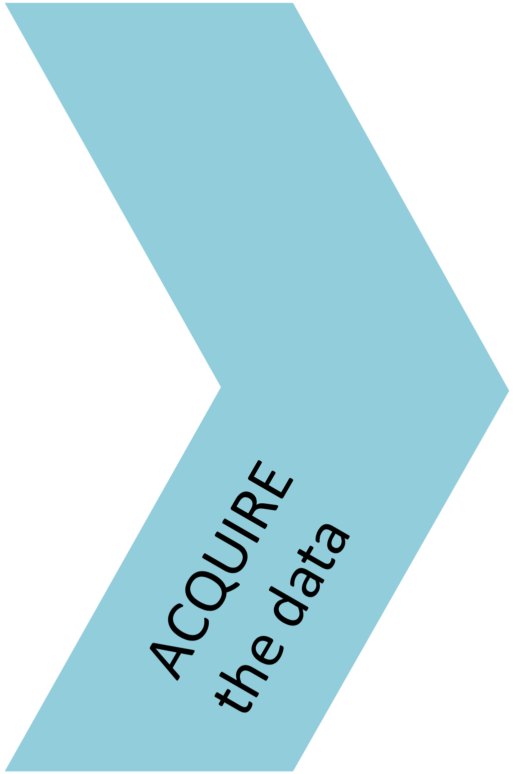
DELIVERABLE

Answers to the above questions

DS

❷ ACQUIRE the Data

❷ Acquire the Data



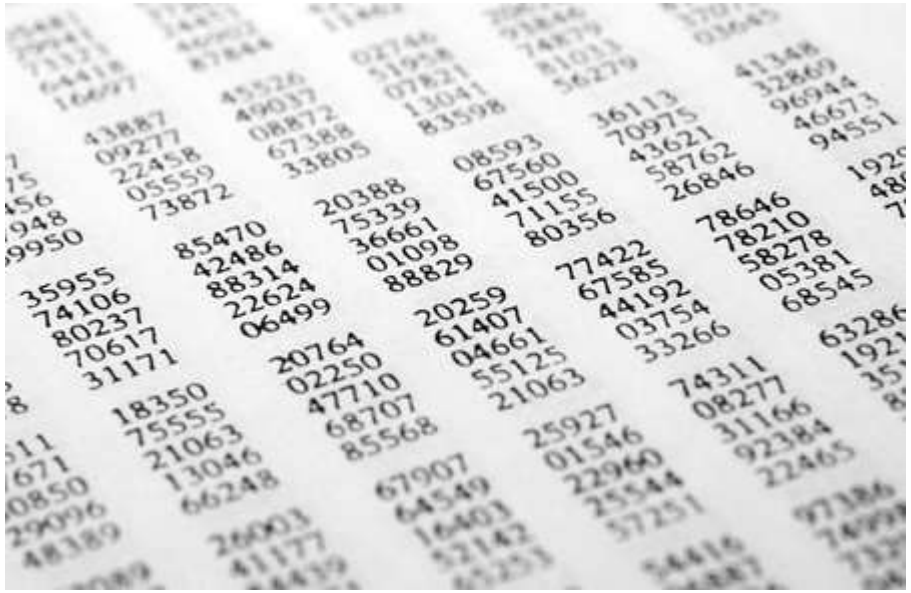
ACQUIRE
the data

- Acquire the Data
  - Identify the “right” dataset(s)
  - Import data and set up local or remote data structure
  - Determine most appropriate tools to work with data

The data can be either unstructured or structured data

Today, we will focus in structured data…

but classes 13 and 14 in unit 3 will focus on Natural Language Processing



milosb © 123RF.com

Bundit Chuangboonsri © 123RF.com

❷ Acquire the Data (cont.)

- Questions to ask:
  - What type of data is it, cross-sectional or longitudinal?
  - How well was the data collected?
  - Is there much missing data?
  - Was the data collection instrument calibrated?
  - Is the dataset aggregated?
  - Do we need pre-aggregated data?

- Data Types
- Logistics of Acquiring Data
  - The SF housing Dataset
- Tidying Up Data
- File Formats

❷ ACQUIRE the Data

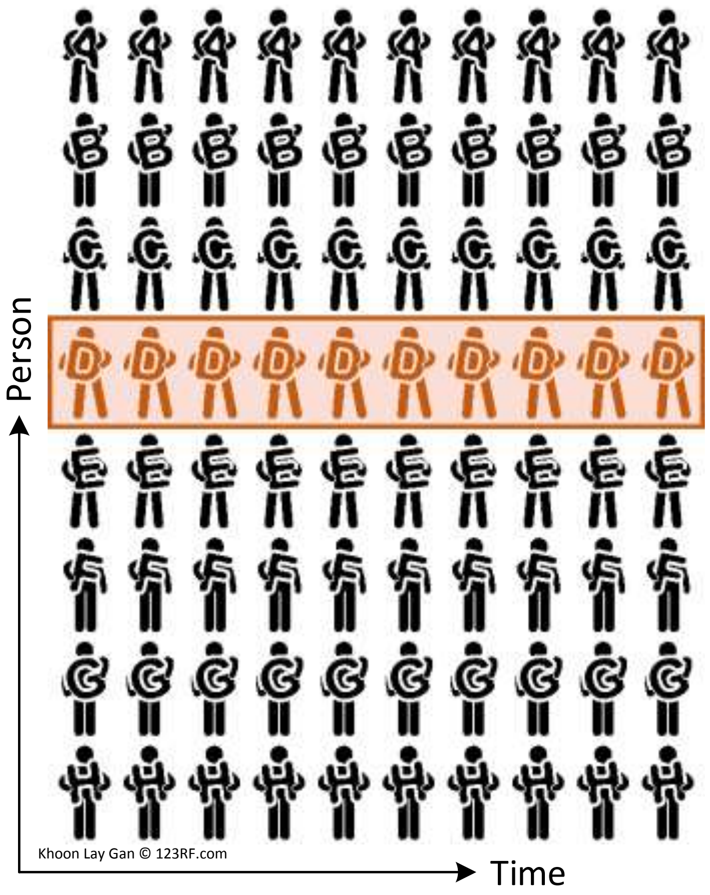
Data Types

Why Data Types Matter

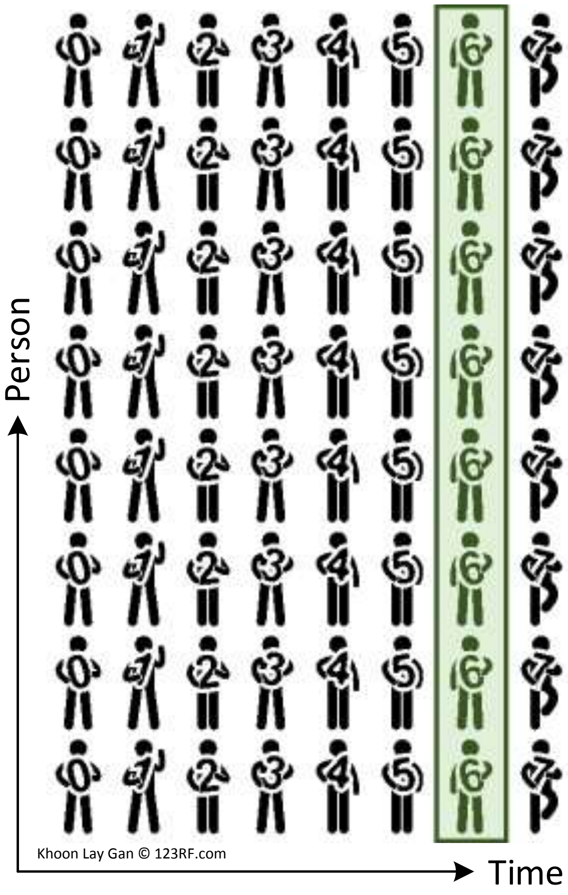
- Different data types have different limitations and strengths
- Certain types of analyses aren’t possible with certain data types
- There are 2 types of data which we may might use for analysis

Time Series/Longitudinal Data

- Information is collected over a period of time
- Sessions 15 and 16 in Unit 3
  - Time Series



Cross-Sectional Data



Khoon Lay Gan © 123RF.com

- All information is determined at the same time; all data comes from the same time period
  - Note: There is no distinction between exposure and outcome
- Most of the course will focus on this type of data

Data Types: Strengths and Weaknesses

	Strengths	Weaknesses
Cross-Sectional Data	❑ Often population-based ❑ Generalizable ❑ Less expensive compared to other types of data collection methods	❑ Separation of cause and effect may be difficult (or impossible) ❑ Variables/cases with long duration are over-represented
Time Series/Longitudinal Data	❑ Unambiguous temporal sequence; exposure precedes outcome ❑ Multiple outcomes can be measured	❑ Takes a long time to collect data ❑ Vulnerable to missing data ❑ More expense compared to other types of data collection methods

❷ ACQUIRE the Data

Activity | Knowledge Check

Activity | Knowledge Check

EXERCISE

DIRECTIONS (10 minutes)

1. What type of data is shown by Zillow? (http://www.zillow.com/san-francisco-ca/sold/)
2. Can you create a cross-sectional analysis from a longitudinal data collection? How? Is this applicable from the data above?
3. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

❷ ACQUIRE the Data

Logistics of Acquiring Data

Logistics of Acquiring Data

- Data can be acquired through a variety of sources
  - Web (e.g., Google Analytics, HTML)
  - Databases
    - SQL (Structured Query Language)
    - NoSQL (“Not only SQL”)

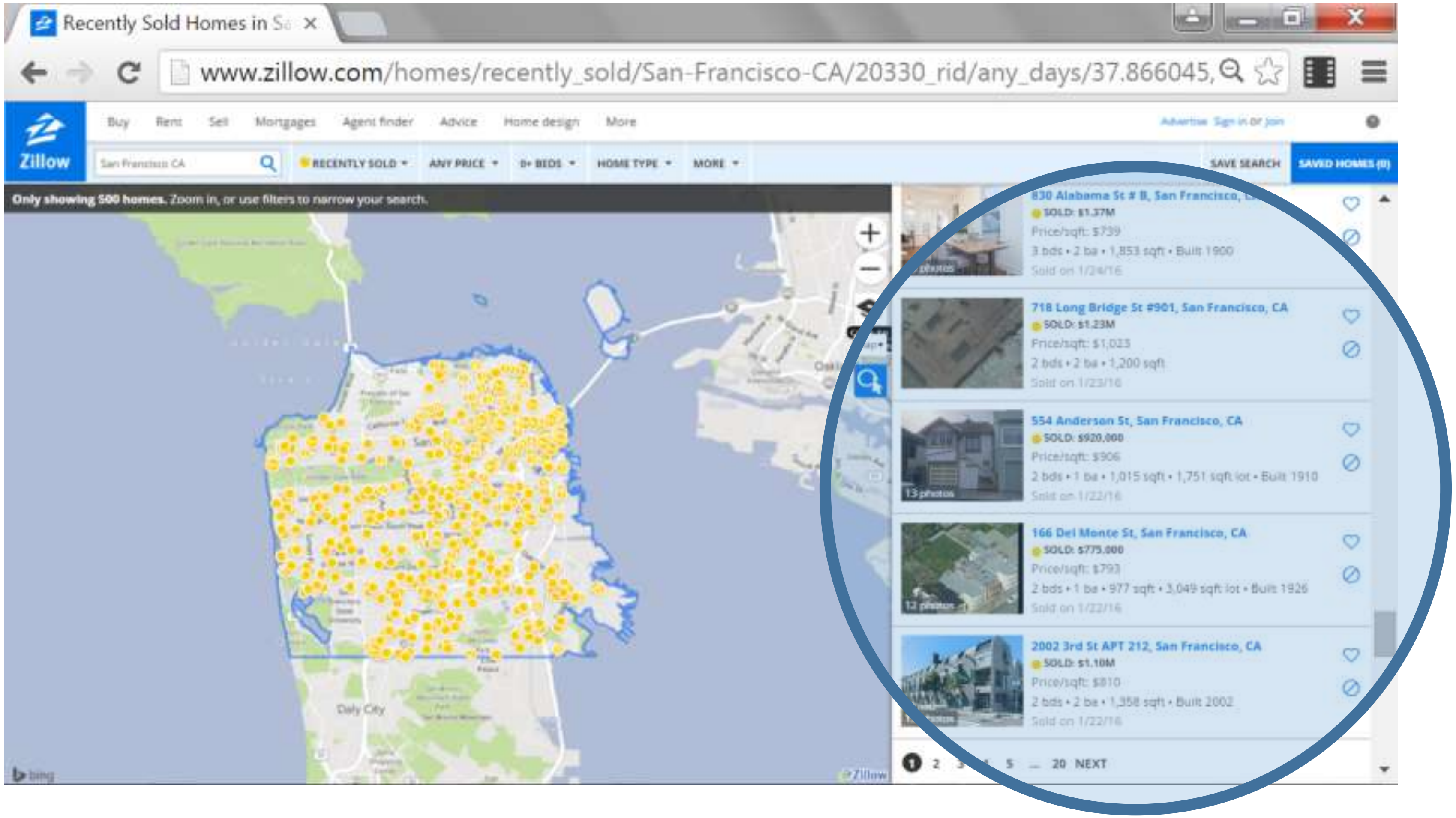
- Files
  - CSV (Comma-Separated Values)
  - TSV/TXT (Tab-Separated Values)
  - JSON (JavaScript Object Notation)
  - XML (eXtensible Markup Language)

SF Housing Dataset from Zillow: a dataset we will use throughout this course

CASE STUDY

- Recently Sold Homes (Source: Zillow)
  - 1,000 homes sold in San Francisco between 11/10/2015 and 2/12/2106

Raw data was scrapped from the Zillow website (20 pages, each listing 50 homes = 1,000 homes)



Only showing 500 homes. Zoom in, or use filters to narrow your search.

830 Alabama St # B, San Francisco, CA
SOLD: $1.37M
Price/sqft: $739
3 bds • 2 ba • 1,853 sqft • Built 1900
Sold on 1/24/16

718 Long Bridge St #901, San Francisco, CA
SOLD: $1.23M
Price/sqft: $1,023
2 bds • 2 ba • 1,200 sqft
Sold on 1/23/16

554 Anderson St, San Francisco, CA
SOLD: $920,000
Price/sqft: $906
2 bds • 1 ba • 1,015 sqft • 1,751 sqft lot • Built 1910
Sold on 1/22/16

166 Del Monte St, San Francisco, CA
SOLD: $775,000
Price/sqft: $793
2 bds • 1 ba • 977 sqft • 3,049 sqft lot • Built 1926
Sold on 1/22/16

2002 3rd St APT 212, San Francisco, CA
SOLD: $1.10M
Price/sqft: $810
2 bds • 2 ba • 1,358 sqft • Built 2002
Sold on 1/22/16

Raw data is Messy™…

```
<div class="property-info"
id="yui_3_18_1_1_1456167242885_71870"><strong
id="yui_3_18_1_1_1456167242885_71869"><dt class="property-address"
id="yui_3_18_1_1_1456167242885_71868"><a href="/homedetails/149-
Shipley-St-San-Francisco-CA-94107/15147894_zpid/" class="hdp-link
routable" title="149 Shipley St, San Francisco, CA Real Estate"
id="yui_3_18_1_1_1456167242885_71873">149 Shipley St, San
Francisco, CA</a></dt></strong><dt class="listing-type zsg-
content_collapsed" id="yui_3_18_1_1_1456167242885_71875"><span
class="zsg-icon-recently-sold type-icon"></span>Sold:
$1.18M</dt><dt class="zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71877">Price/sqft: $1,116</dt><dt
class="property-data" id="yui_3_18_1_1_1456167242885_71880"><span
class="beds-baths-sqft">3 bds • 2 ba • 1,057 sqft</span><span
class="built-year" id="yui_3_18_1_1_1456167242885_71879"> • Built
1992</span></dt><dt class="sold-date zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71975">Sold on 2/22/16</dt></div>
```

… and needs to be parsed and tidied up (a.k.a., organized)

```
<div class="property-info"
id="yui_3_18_1_1_1456167242885_71870"><strong
id="yui_3_18_1_1_1456167242885_71869"><dt class="property-address"
id="yui_3_18_1_1_1456167242885_71868"><a href="/homedetails/149-
Shipley-St-San-Francisco-CA-94107/15147894_zpid/" class="hdp-link
routable" title="149 Shipley St, San Francisco, CA Real Estate"
id="yui_3_18_1_1_1456167242885_71873">149 Shipley St, San
Francisco, CA</a></dt></strong><dt class="listing-type zsg-
content_collapsed" id="yui_3_18_1_1_1456167242885_71875"><span
class="zsg-icon-recently-sold type-icon"></span>Sold:
$1.18M</dt><dt class="zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71877">Price/sqft: $1,116</dt><dt
class="property-data" id="yui_3_18_1_1_1456167242885_71880"><span
class="beds-baths-sqft">3 bds • 2 ba • 1,057 sqft</span><span
class="built-year" id="yui_3_18_1_1_1456167242885_71879"> • Built
1992</span></dt><dt class="sold-date zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71975">Sold on 2/22/16</dt></div>
```

❷ ACQUIRE the Data

Tidying Up Data

Tidying Up Data

- Tidying up data is the most fruitful skill you can learn as a data scientist
  - It will save you hours of time and make your data much easier to visualize, manipulate, and model
- Many data science tools follow a set of conventions that makes one layout of tabular data much easier to work with than others. Your data will be easier to work with if you follow three rules:
  - Each observation is placed in its own row
  - Each variable in the dataset is placed in its own column
  - Each value is placed in its own cell

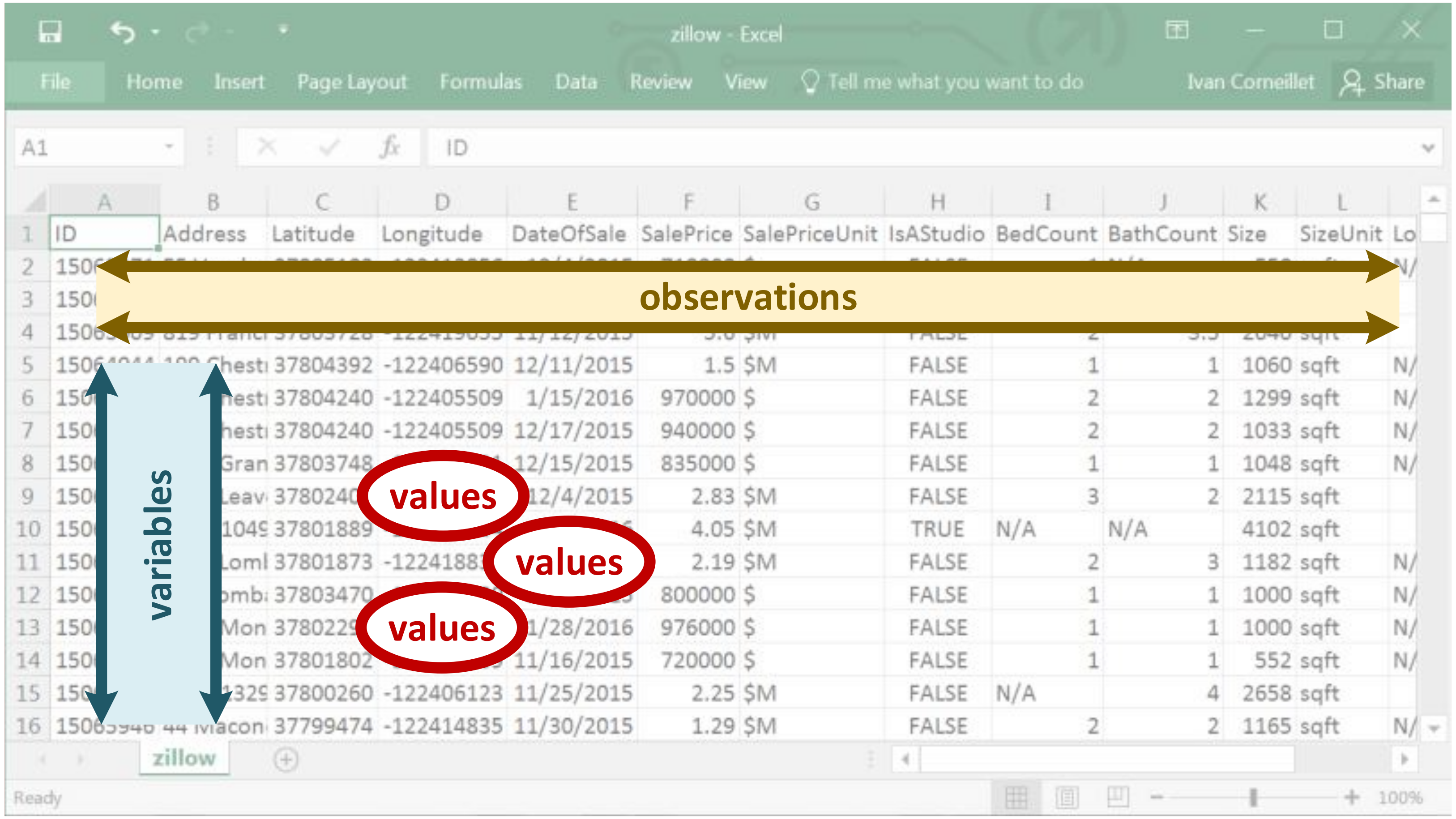
Really, data can be incredibly raw

EXAMPLE

- Trouble tickets inspect and maintain manholes in New Year City
- "Service box," a common piece of infrastructure, had at least 38 variants, including `SB`, `S`, `S/B`, `S.B`, `S?B`, `S.B.`, `SBX`, `S/BX`, `SB/X`, `S/XB`, `/SBX`, `S.BX`, `S  &BX`, `S?BX`, `S  BX`, `S/B/X`, `S  BOX`, `SVBX`, `SERV  BX`, `SERV-BOX`, `SERV/BOX`, and `SERVICE  BOX`

(Source: Big Data: A Revolution That Will Transform How We Live, Work, and Think)

The Tidy SF Housing Dataset



	A	B	C	D	E	F	G	H	I	J	K	L	
1	ID	Address	Latitude	Longitude	DateOfSale	SalePrice	SalePriceUnit	IsAStudio	BedCount	BathCount	Size	SizeUnit	Lo
2	150[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	N/
3	150	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
4	1506[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
5	150	Chesti	37804392	-122406590	12/11/2015	1.5	$M	FALSE	1	1	1060	sqft	N/
6	150	esti	37804240	-122405509	1/15/2016	970000	$	FALSE	2	2	1299	sqft	N/
7	150	hesti	37804240	-122405509	12/17/2015	940000	$	FALSE	2	2	1033	sqft	N/
8	150	Gran	37803748	[illegible]	12/15/2015	835000	$	FALSE	1	1	1048	sqft	N/
9	150	Leav	3780240	[illegible]	12/4/2015	2.83	$M	FALSE	3	2	2115	sqft	
10	150	1049	37801889	[illegible]	[illegible]	4.05	$M	TRUE	N/A	N/A	4102	sqft	
11	150	Loml	37801873	-1224188	[illegible]	2.19	$M	FALSE	2	3	1182	sqft	N/
12	150	ombi	37803470	[illegible]	[illegible]	800000	$	FALSE	1	1	1000	sqft	N/
13	150	Mon	3780229	[illegible]	1/28/2016	976000	$	FALSE	1	1	1000	sqft	N/
14	150	Mon	37801802	[illegible]	11/16/2015	720000	$	FALSE	1	1	552	sqft	N/
15	150	1329	37800260	-122406123	11/25/2015	2.25	$M	FALSE	N/A	4	2658	sqft	
16	15063946	44 Macon	37799474	-122414835	11/30/2015	1.29	$M	FALSE	2	2	1165	sqft	N/

observations

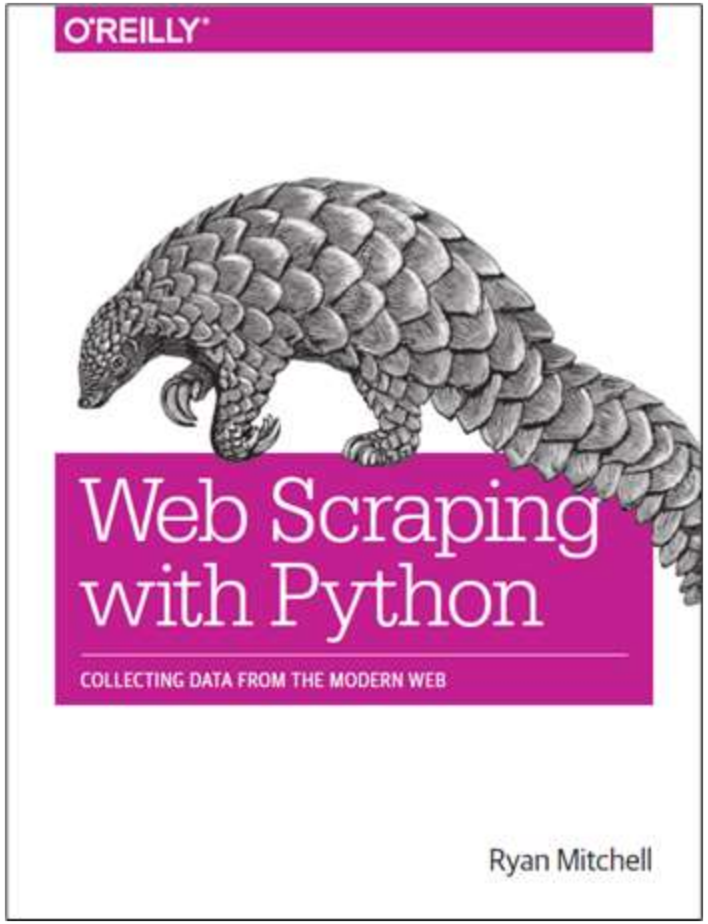
variables

values

values

values

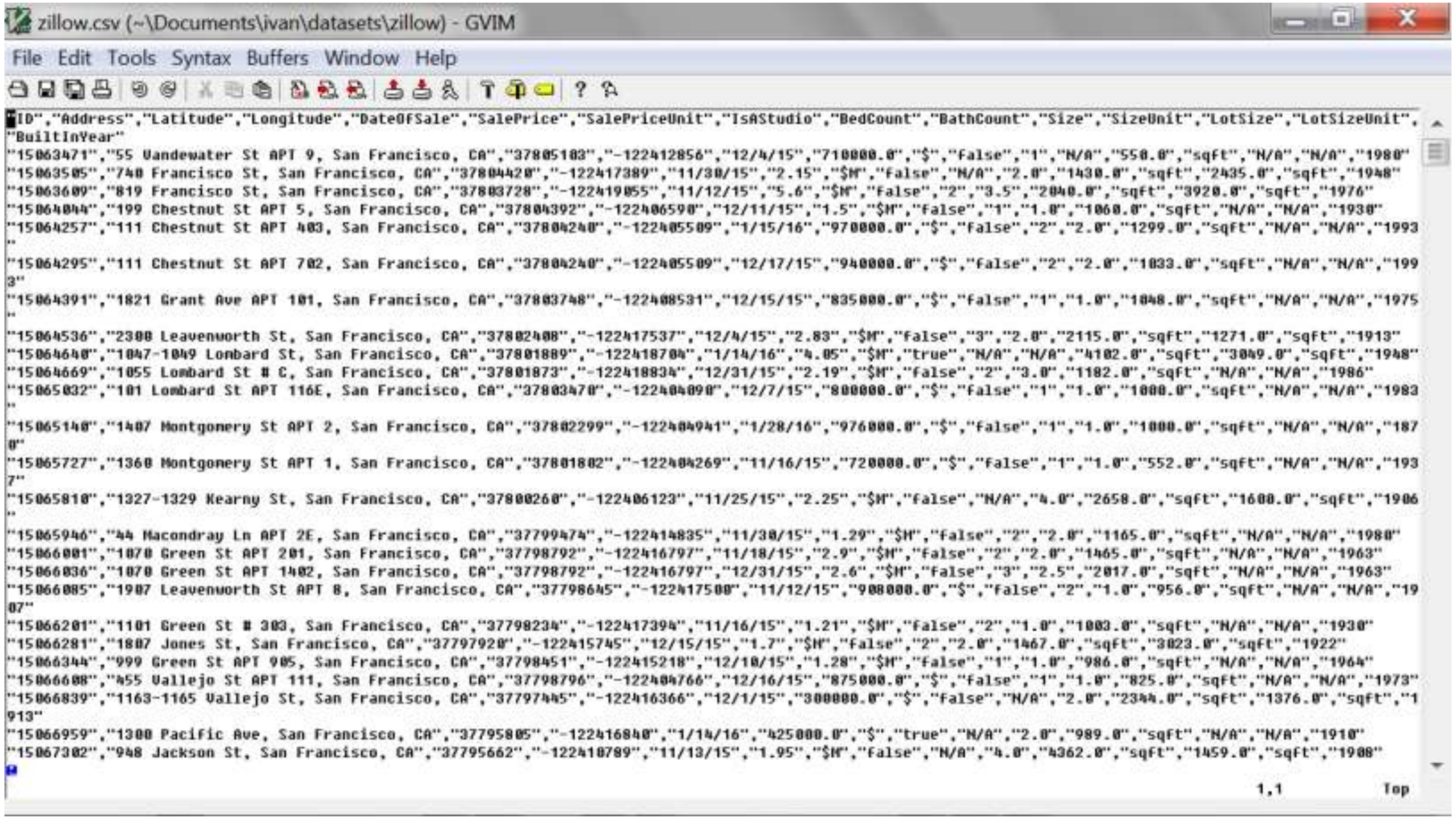
A good resource to get started with web scraping using Python *(optional; not required for the course)*



❷ ACQUIRE the Data

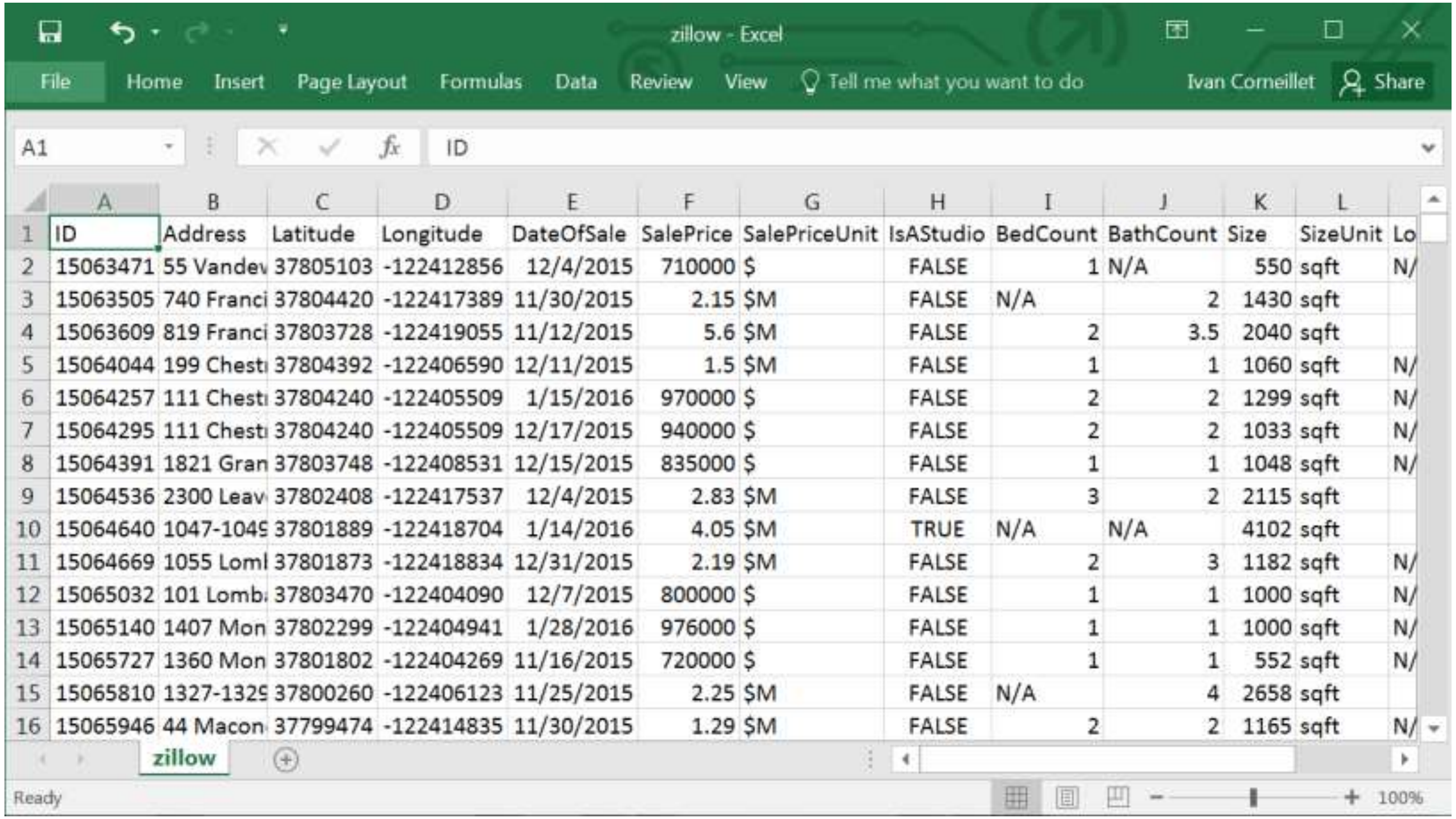
File Formats

Our tidy SF housing dataset in CSV format: each observation is in one line; within each line, variables are separated with commas (and here delimited with double-quotes)



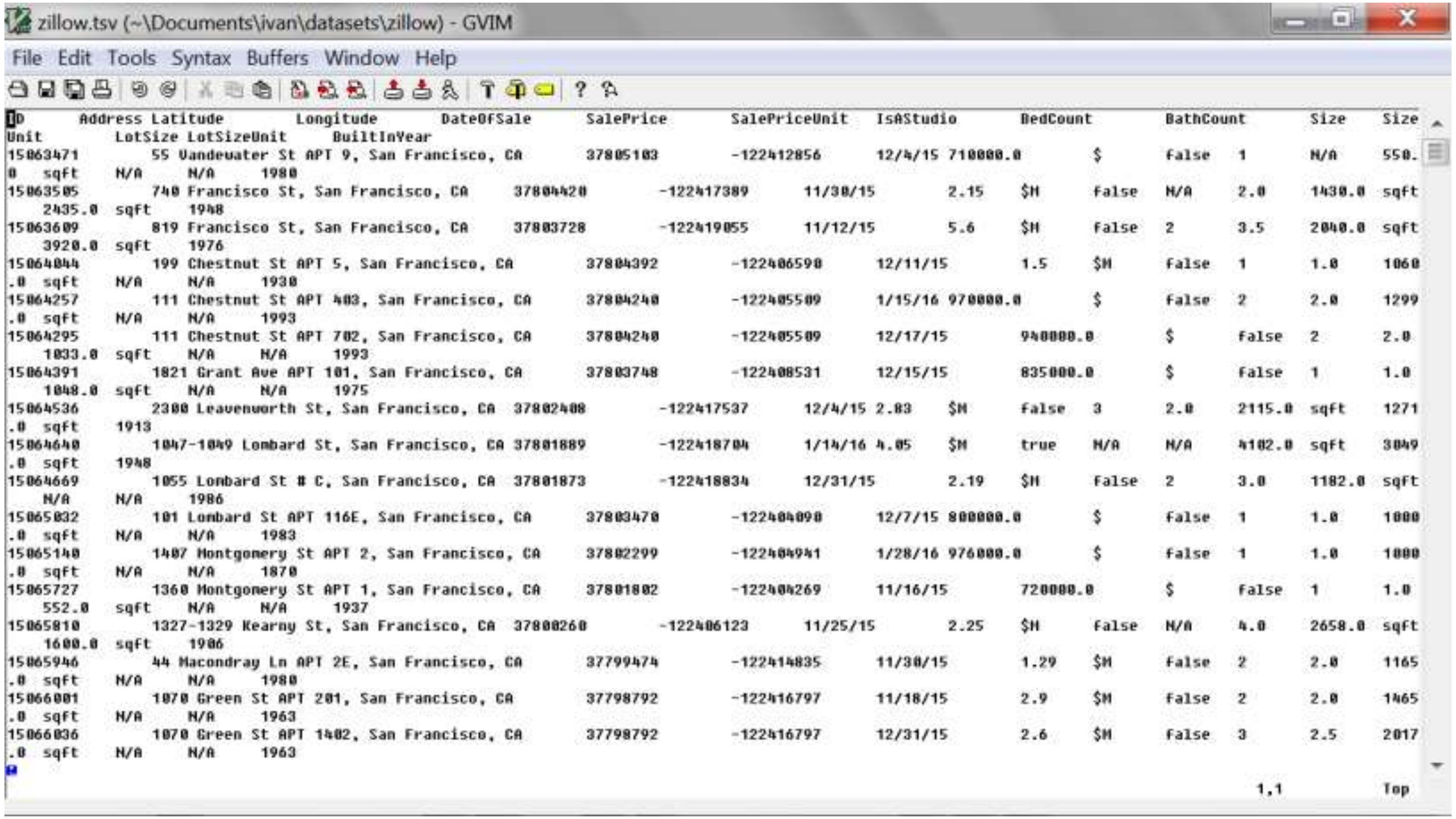
```
"ID","Address","Latitude","Longitude","DateOfSale","SalePrice","SalePriceUnit","IsAStudio","BedCount","BathCount","Size","SizeUnit","LotSize","LotSizeUnit",
"BuiltInYear"
"15063471","55 Vandewater St APT 9, San Francisco, CA","37805103","-122412856","12/4/15","710000.0","$","false","1","N/A","550.0","sqft","N/A","N/A","1980"
"15063505","740 Francisco St, San Francisco, CA","37804420","-122417389","11/30/15","2.15","$M","false","N/A","2.0","1430.0","sqft","2435.0","sqft","1948"
"15063609","819 Francisco St, San Francisco, CA","37803728","-122419055","11/12/15","5.6","$M","false","2","3.5","2040.0","sqft","3920.0","sqft","1976"
"15064044","199 Chestnut St APT 5, San Francisco, CA","37804392","-122406590","12/11/15","1.5","$M","false","1","1.0","1060.0","sqft","N/A","N/A","1930"
"15064257","111 Chestnut St APT 403, San Francisco, CA","37804240","-122405509","1/15/16","970000.0","$","false","2","2.0","1299.0","sqft","N/A","N/A","1993
"
"15064295","111 Chestnut St APT 702, San Francisco, CA","37804240","-122405509","12/17/15","940000.0","$","false","2","2.0","1033.0","sqft","N/A","N/A","199
3"
"15064391","1821 Grant Ave APT 101, San Francisco, CA","37803748","-122408531","12/15/15","835000.0","$","false","1","1.0","1048.0","sqft","N/A","N/A","1975
"
"15064536","2300 Leavenworth St, San Francisco, CA","37802408","-122417537","12/4/15","2.83","$M","false","3","2.0","2115.0","sqft","1271.0","sqft","1913"
"15064640","1047-1049 Lombard St, San Francisco, CA","37801889","-122418704","1/14/16","4.05","$M","true","N/A","N/A","4102.0","sqft","3049.0","sqft","1948"
"15064669","1055 Lombard St # C, San Francisco, CA","37801873","-122418834","12/31/15","2.19","$M","false","2","3.0","1182.0","sqft","N/A","N/A","1986"
"15065032","101 Lombard St APT 116E, San Francisco, CA","37803470","-122404090","12/7/15","800000.0","$","false","1","1.0","1000.0","sqft","N/A","N/A","1983
"
"15065140","1407 Montgomery St APT 2, San Francisco, CA","37802299","-122404941","1/28/16","976000.0","$","false","1","1.0","1000.0","sqft","N/A","N/A","187
0"
"15065727","1360 Montgomery St APT 1, San Francisco, CA","37801802","-122404269","11/16/15","720000.0","$","false","1","1.0","552.0","sqft","N/A","N/A","193
7"
"15065810","1327-1329 Kearny St, San Francisco, CA","37800260","-122406123","11/25/15","2.25","$M","false","N/A","4.0","2658.0","sqft","1600.0","sqft","1906
"
"15065946","44 Macondray Ln APT 2E, San Francisco, CA","37799474","-122414835","11/30/15","1.29","$M","false","2","2.0","1165.0","sqft","N/A","N/A","1980"
"15066001","1070 Green St APT 201, San Francisco, CA","37798792","-122416797","11/18/15","2.9","$M","false","2","2.0","1465.0","sqft","N/A","N/A","1963"
"15066036","1070 Green St APT 1402, San Francisco, CA","37798792","-122416797","12/31/15","2.6","$M","false","3","2.5","2017.0","sqft","N/A","N/A","1963"
"15066085","1907 Leavenworth St APT 8, San Francisco, CA","37798645","-122417500","11/12/15","908000.0","$","false","2","1.0","956.0","sqft","N/A","N/A","19
07"
"15066201","1101 Green St # 303, San Francisco, CA","37798234","-122417394","11/16/15","1.21","$M","false","2","1.0","1003.0","sqft","N/A","N/A","1930"
"15066281","1807 Jones St, San Francisco, CA","37797920","-122415745","12/15/15","1.7","$M","false","2","2.0","1467.0","sqft","3023.0","sqft","1922"
"15066344","999 Green St APT 905, San Francisco, CA","37798451","-122415218","12/10/15","1.28","$M","false","1","1.0","986.0","sqft","N/A","N/A","1964"
"15066608","455 Vallejo St APT 111, San Francisco, CA","37798796","-122404766","12/16/15","875000.0","$","false","1","1.0","825.0","sqft","N/A","N/A","1973"
"15066839","1163-1165 Vallejo St, San Francisco, CA","37797445","-122416366","12/1/15","300000.0","$","false","N/A","2.0","2344.0","sqft","1376.0","sqft","1
913"
"15066959","1300 Pacific Ave, San Francisco, CA","37795805","-122416840","1/14/16","425000.0","$","true","N/A","2.0","989.0","sqft","N/A","N/A","1910"
"15067302","948 Jackson St, San Francisco, CA","37795662","-122410789","11/13/15","1.95","$M","false","N/A","4.0","4362.0","sqft","1459.0","sqft","1908"

                                                                                                                        1,1           Top
```

Excel reads CSV files natively (and our Python code will too…)



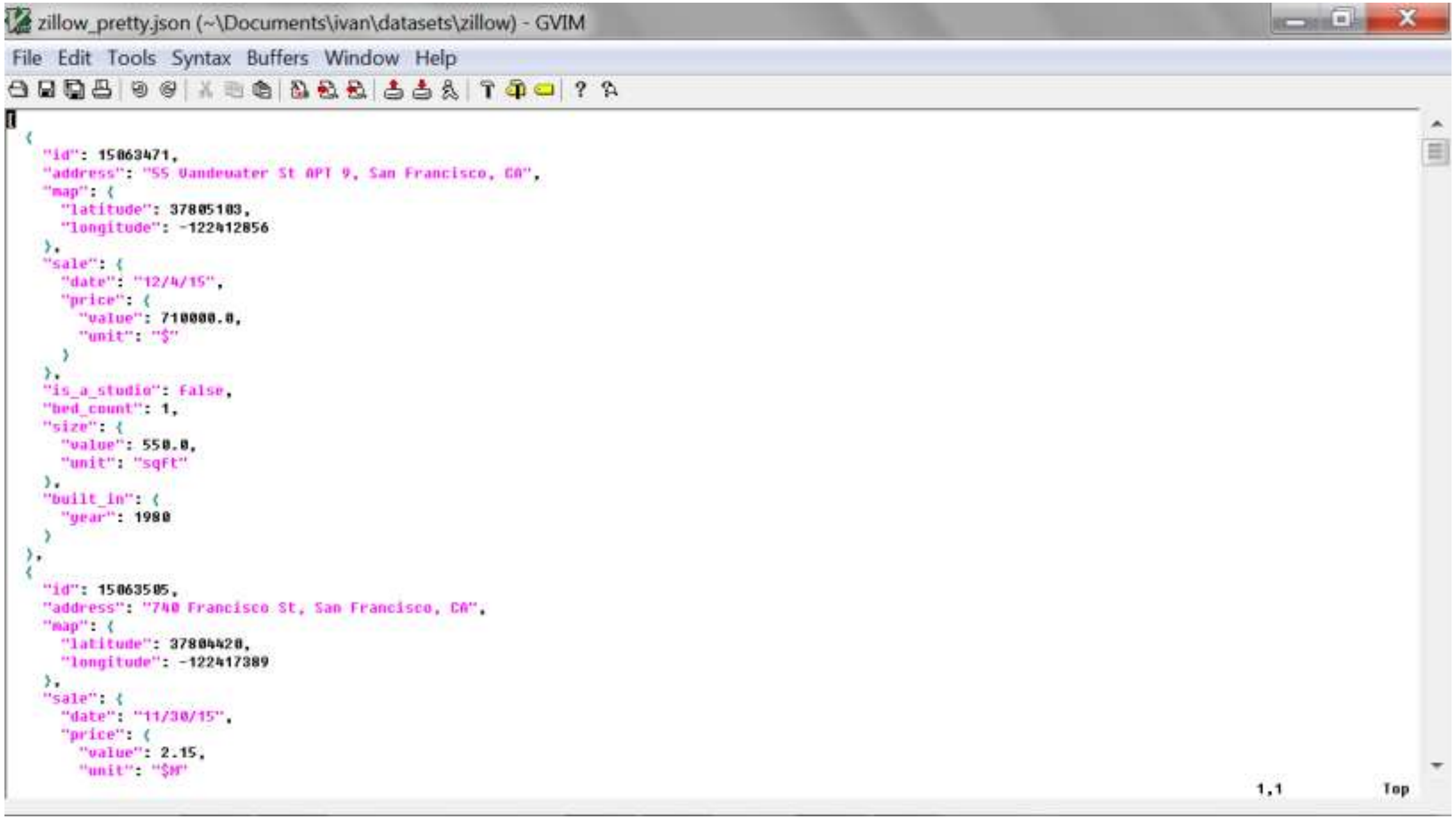
	A	B	C	D	E	F	G	H	I	J	K	L	
1	ID	Address	Latitude	Longitude	DateOfSale	SalePrice	SalePriceUnit	IsAStudio	BedCount	BathCount	Size	SizeUnit	Lo
2	15063471	55 Vandev	37805103	-122412856	12/4/2015	710000	$	FALSE	1	N/A	550	sqft	N/
3	15063505	740 Franci	37804420	-122417389	11/30/2015	2.15	$M	FALSE	N/A	2	1430	sqft	
4	15063609	819 Franci	37803728	-122419055	11/12/2015	5.6	$M	FALSE	2	3.5	2040	sqft	
5	15064044	199 Chest	37804392	-122406590	12/11/2015	1.5	$M	FALSE	1	1	1060	sqft	N/
6	15064257	111 Chest	37804240	-122405509	1/15/2016	970000	$	FALSE	2	2	1299	sqft	N/
7	15064295	111 Chest	37804240	-122405509	12/17/2015	940000	$	FALSE	2	2	1033	sqft	N/
8	15064391	1821 Gran	37803748	-122408531	12/15/2015	835000	$	FALSE	1	1	1048	sqft	N/
9	15064536	2300 Leav	37802408	-122417537	12/4/2015	2.83	$M	FALSE	3	2	2115	sqft	
10	15064640	1047-1049	37801889	-122418704	1/14/2016	4.05	$M	TRUE	N/A	N/A	4102	sqft	
11	15064669	1055 Lomb	37801873	-122418834	12/31/2015	2.19	$M	FALSE	2	3	1182	sqft	N/
12	15065032	101 Lomb	37803470	-122404090	12/7/2015	800000	$	FALSE	1	1	1000	sqft	N/
13	15065140	1407 Mon	37802299	-122404941	1/28/2016	976000	$	FALSE	1	1	1000	sqft	N/
14	15065727	1360 Mon	37801802	-122404269	11/16/2015	720000	$	FALSE	1	1	552	sqft	N/
15	15065810	1327-1329	37800260	-122406123	11/25/2015	2.25	$M	FALSE	N/A	4	2658	sqft	
16	15065946	44 Macon	37799474	-122414835	11/30/2015	1.29	$M	FALSE	2	2	1165	sqft	N/

TSV is another simple text format for storing data in a tabular structure: each observation in the table is one line of the text file and each variable is separated from the next by a tab character



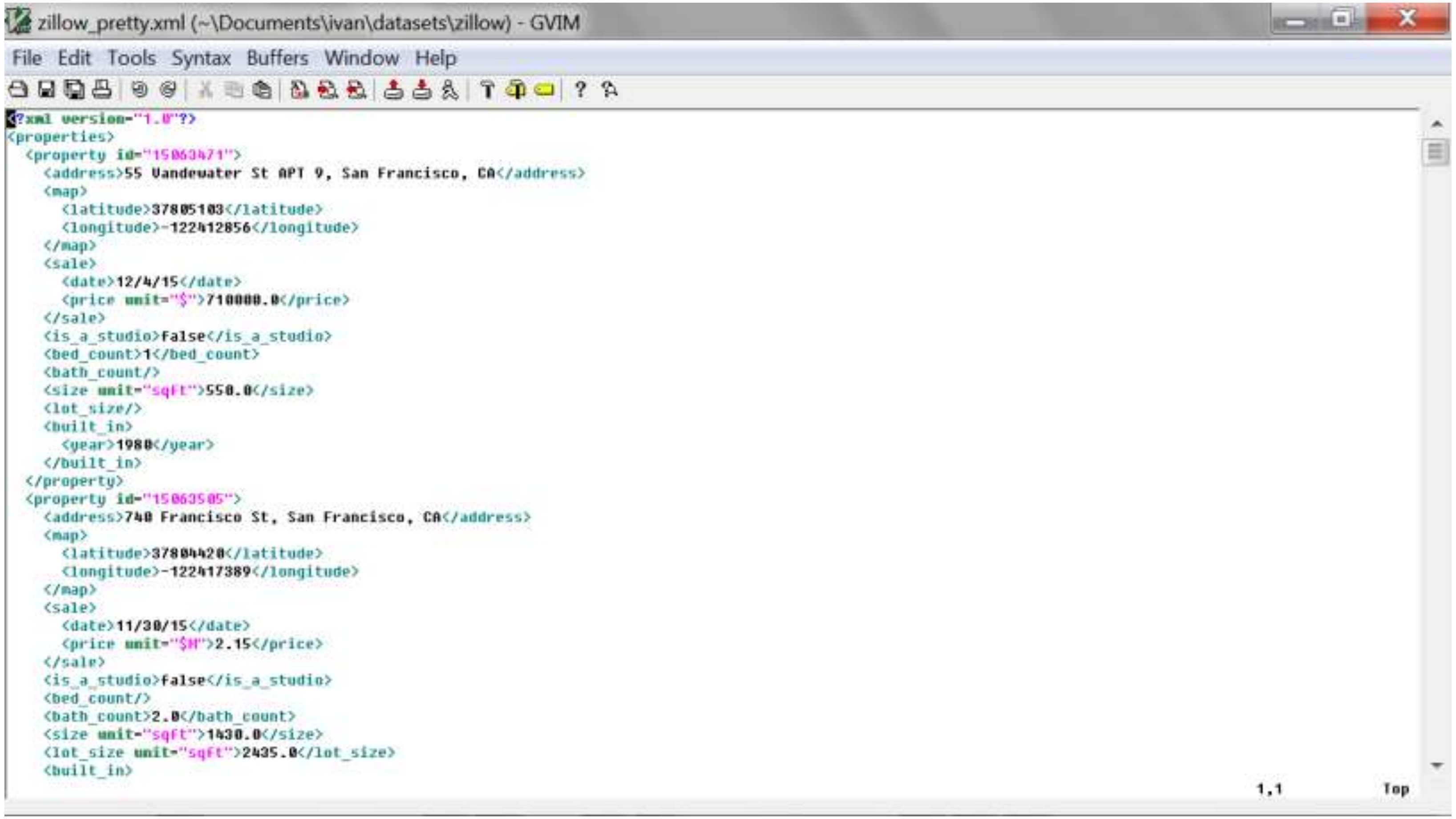
```
ID      Address Latitude        Longitude       DateOfSale      SalePrice       SalePriceUnit   IsAStudio       BedCount        BathCount       Size    Size
Unit        LotSize LotSizeUnit     BuiltInYear
15063471        55 Vandewater St APT 9, San Francisco, CA       37805103        -122412856      12/4/15 710000.0        $       False   1       N/A     550.
0   sqft    N/A     N/A     1980
15063505        740 Francisco St, San Francisco, CA     37804420        -122417389      11/30/15        2.15    $M      False   N/A     2.0     1430.0  sqft
    2435.0  sqft    1948
15063609        819 Francisco St, San Francisco, CA     37803728        -122419055      11/12/15        5.6     $M      False   2       3.5     2040.0  sqft
    3920.0  sqft    1976
15064044        199 Chestnut St APT 5, San Francisco, CA        37804392        -122406590      12/11/15        1.5     $M      False   1       1.0     1060
.0  sqft    N/A     N/A     1930
15064257        111 Chestnut St APT 403, San Francisco, CA      37804240        -122405509      1/15/16 970000.0        $       False   2       2.0     1299
.0  sqft    N/A     N/A     1993
15064295        111 Chestnut St APT 702, San Francisco, CA      37804240        -122405509      12/17/15        940000.0        $       false   2       2.0
    1033.0  sqft    N/A     N/A     1993
15064391        1821 Grant Ave APT 101, San Francisco, CA       37803748        -122408531      12/15/15        835000.0        $       false   1       1.0
    1048.0  sqft    N/A     N/A     1975
15064536        2300 Leavenworth St, San Francisco, CA  37802408        -122417537      12/4/15 2.03    $M      false   3       2.0     2115.0  sqft    1271
.0  sqft    1913
15064640        1047-1049 Lombard St, San Francisco, CA 37801889        -122418704      1/14/16 4.05    $M      true    N/A     N/A     4102.0  sqft    3049
.0  sqft    1948
15064669        1055 Lombard St # C, San Francisco, CA  37801873        -122418834      12/31/15        2.19    $M      False   2       3.0     1182.0  sqft
    N/A     N/A     1986
15065032        101 Lombard St APT 116E, San Francisco, CA      37803470        -122404090      12/7/15 800000.0        $       False   1       1.0     1000
.0  sqft    N/A     N/A     1983
15065140        1407 Montgomery St APT 2, San Francisco, CA     37802299        -122404941      1/28/16 976000.0        $       False   1       1.0     1000
.0  sqft    N/A     N/A     1870
15065727        1360 Montgomery St APT 1, San Francisco, CA     37801802        -122404269      11/16/15        720000.0        $       False   1       1.0
     552.0  sqft    N/A     N/A     1937
15065810        1327-1329 Kearny St, San Francisco, CA  37800260        -122406123      11/25/15        2.25    $M      False   N/A     4.0     2658.0  sqft
    1600.0  sqft    1906
15065946        44 Macondray Ln APT 2E, San Francisco, CA       37799474        -122414835      11/30/15        1.29    $M      False   2       2.0     1165
.0  sqft    N/A     N/A     1980
15066001        1070 Green St APT 201, San Francisco, CA        37798792        -122416797      11/18/15        2.9     $M      False   2       2.0     1465
.0  sqft    N/A     N/A     1963
15066036        1070 Green St APT 1402, San Francisco, CA       37798792        -122416797      12/31/15        2.6     $M      False   3       2.5     2017
.0  sqft    N/A     N/A     1963

                                                                                                                                            1,1         Top
```

JSON, the most common open standard data format used for asynchronous browser/server communication, uses human-readable text to store data using key–value pairs and lists. Unlike CVS/TSV format, data can be represented hierarchically


zillow_pretty.json (~\Documents\ivan\datasets\zillow) - GVIM
File Edit Tools Syntax Buffers Window Help
1,1
Top

```
[
  {
    "id": 15063471,
    "address": "55 Vandewater St APT 9, San Francisco, CA",
    "map": {
      "latitude": 37805103,
      "longitude": -122412856
    },
    "sale": {
      "date": "12/4/15",
      "price": {
        "value": 710000.0,
        "unit": "$"
      }
    },
    "is_a_studio": false,
    "bed_count": 1,
    "size": {
      "value": 550.0,
      "unit": "sqft"
    },
    "built_in": {
      "year": 1980
    }
  },
  {
    "id": 15063505,
    "address": "740 Francisco St, San Francisco, CA",
    "map": {
      "latitude": 37804420,
      "longitude": -122417389
    },
    "sale": {
      "date": "11/30/15",
      "price": {
        "value": 2.15,
        "unit": "$M"
```

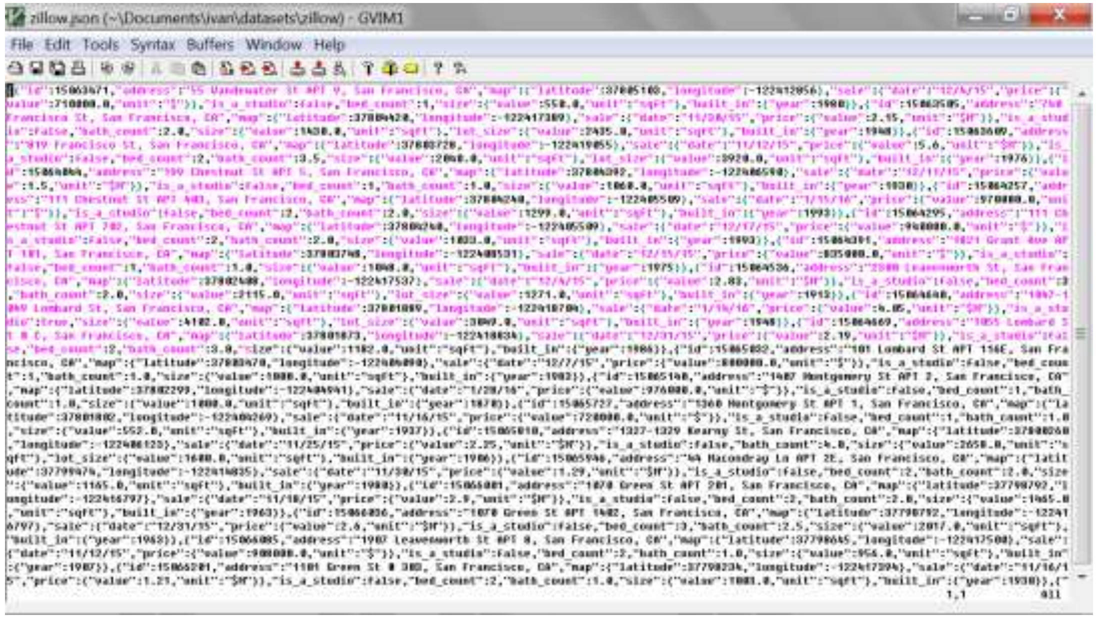
XML, another data format used for asynchronous browser/server communication, also uses human-readable text to store hierarchical data

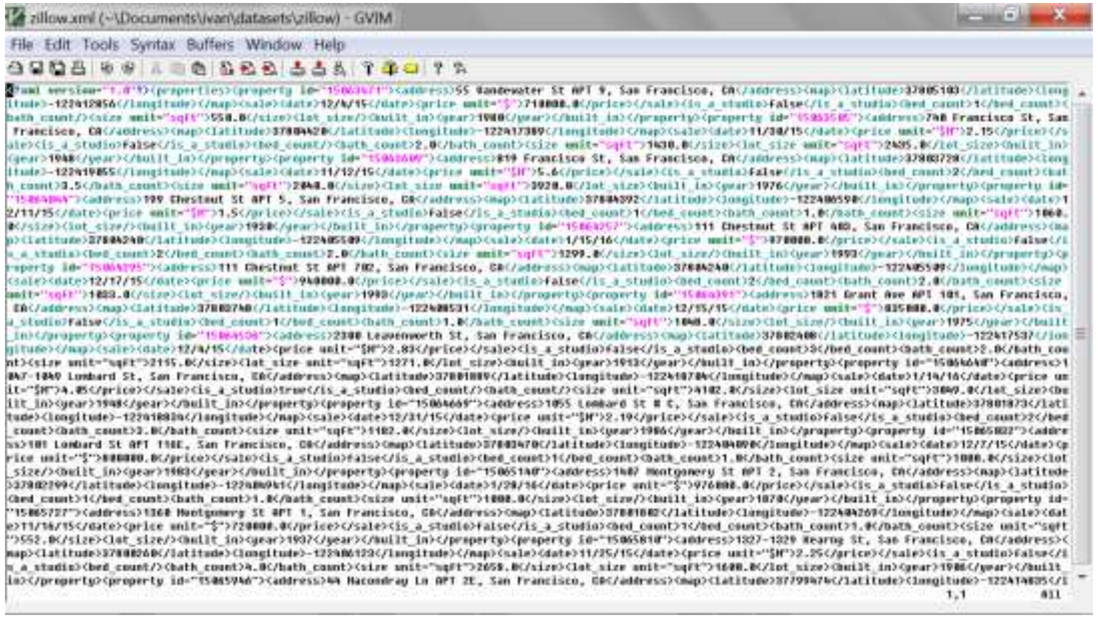


zillow_pretty.xml (~\Documents\ivan\datasets\zillow) - GVIM
File Edit Tools Syntax Buffers Window Help
1,1
Top

```xml
<?xml version="1.0"?>
<properties>
  <property id="15063471">
    <address>55 Vandewater St APT 9, San Francisco, CA</address>
    <map>
      <latitude>37805103</latitude>
      <longitude>-122412856</longitude>
    </map>
    <sale>
      <date>12/4/15</date>
      <price unit="$">710000.0</price>
    </sale>
    <is_a_studio>False</is_a_studio>
    <bed_count>1</bed_count>
    <bath_count/>
    <size unit="sqft">550.0</size>
    <lot_size/>
    <built_in>
      <year>1980</year>
    </built_in>
  </property>
  <property id="15063505">
    <address>740 Francisco St, San Francisco, CA</address>
    <map>
      <latitude>37804420</latitude>
      <longitude>-122417389</longitude>
    </map>
    <sale>
      <date>11/30/15</date>
      <price unit="$M">2.15</price>
    </sale>
    <is_a_studio>False</is_a_studio>
    <bed_count/>
    <bath_count>2.0</bath_count>
    <size unit="sqft">1430.0</size>
    <lot_size unit="sqft">2435.0</lot_size>
    <built_in>
```

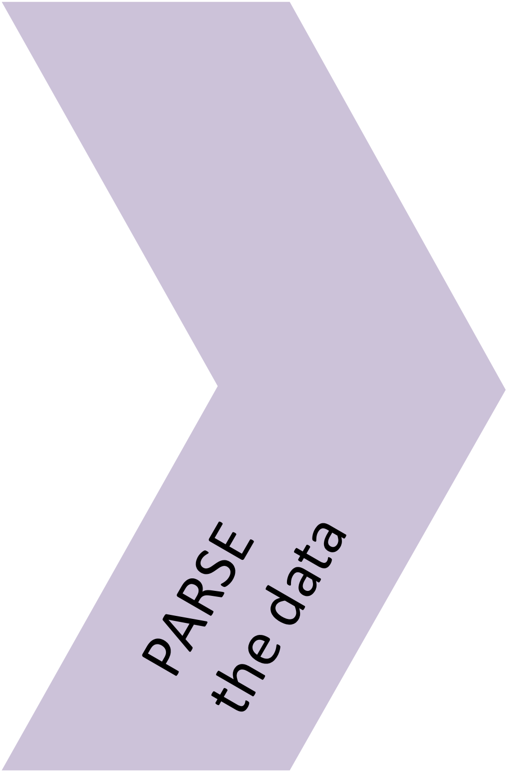
JSON and XML are harder to read by humans when indentation is removed (usually the default) although it is still straightforward for machines…


zillow.json (~\Documents\ivan\datasets\zillow) - GVIM1
File Edit Tools Syntax Buffers Window Help
1,1 All


zillow.xml (~\Documents\ivan\datasets\zillow) - GVIM
File Edit Tools Syntax Buffers Window Help
1,1 All

❸ PARSE the Data

❸ Parse the Data



PARSE
the data

- Parse the Data
  - Read any documentation provided with the data (session 2)
  - Perform exploratory data analysis (session 3)
  - Verify the quality of the data (sessions 2/3)

❷ Acquire the Data (cont.)

- You need to understand what you’re working with
- To better understand your data
  - Create or review the data dictionary
  - Perform exploratory surface analysis
  - Describe data structure and information being collected
  - Explore variables and data types

- Documentation and Data Dictionary
- Introduction to *pandas* + codealong
- Codealong: Tidying up (more) the SF housing dataset
- Lab

❸ PARSE the Data

Documentation and Data Dictionary

Documentation and Data Dictionary

- Data dictionaries
  - Help you judge the quality of the data
  - Also help understand how it’s coded
    - Does “`gender = 1`” mean female or male?
    - Is the currency dollars or euros?
  - Help identify any requirements, assumptions, and constraints of the data
  - Make it easier to share data

Kaggle’s Titanic Data Dictionary

EXAMPLE

```
VARIABLE DESCRIPTIONS:
survival        Survival
                (0 = No; 1 = Yes)
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown;
                 S = Southampton)

SPECIAL NOTES:
Pclass is a proxy for socio-economic status (SES)
 1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

Age is in Years; Fractional if Age less than One (1)
 If the Age is Estimated, it is in the form xx.5
```

```
With respect to the family relation variables (i.e.
sibsp and parch) some relations were ignored.  The
following are the definitions used for sibsp and
parch.

Sibling:  Brother, Sister, Stepbrother, or
          Stepsister of Passenger Aboard Titanic
Spouse:   Husband or Wife of Passenger Aboard
          Titanic (Mistresses and Fiancés Ignored)
Parent:   Mother or Father of Passenger Aboard
          Titanic
Child:    Son, Daughter, Stepson, or Stepdaughter of
          Passenger Aboard Titanic

Other family relatives excluded from this study
include cousins, nephews/nieces, aunts/uncles, and
in-laws.  Some children travelled only with a nanny,
therefore parch=0 for them.  As well, some travelled
with very close friends or neighbors in a village,
however, the definitions do not support such
relations.
```

❸ PARSE the Data

Introduction to pandas

pandas is a Python library to manipulate and perform statistical and mathematical analysis on tabular and multidimensional datasets

- *pandas* provides the ability to index, retrieve, tidy, reshape, combine, slice, and perform various analyses on both single and multidimensional data
- It also includes loading and saving data from local and Internet-based resources
- We will use *pandas* to explore and manipulate the SH housing dataset

❸ PARSE the Data

Codealong | Introduction to pandas

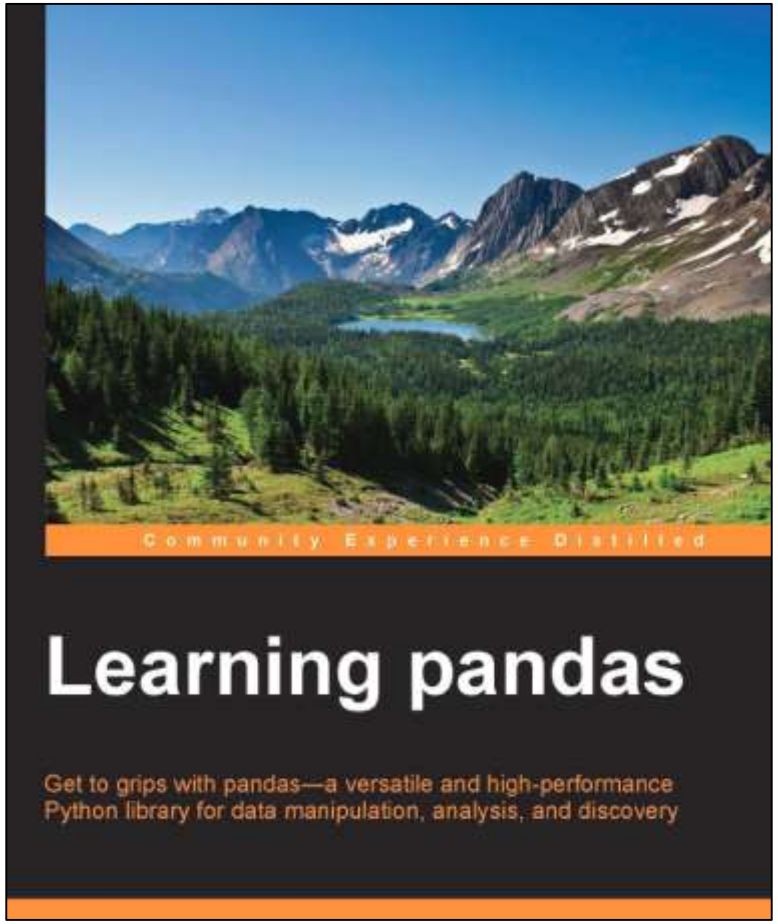
The codealong was just the tip of the iceberg. There is much more. Check out the following:

- *pandas* documentation (which is very well written…)
  - http://pandas.pydata.org/pandas-docs/stable/

Some other good online resources

http://pandas.pydata.org/pandas-docs/stable/tutorials.html	Guide to many pandas tutorials, geared mainly for new users
https://github.com/jvns/pandas-cookbook	Great resource with examples from weather, bikes, and 311 calls from Julia Evans
https://bitbucket.org/hrojas/learn-pandas	Great series of Pandas tutorials from Dave Rojas
https://github.com/ResearchComputing/Meetup-Fall-2013/tree/master/python	Awesome set of python notebooks from a meetup-based course exclusively devoted to pandas

As well as a good book *(again optional; not required for the course)*


Community Experience Distilled
Learning pandas
Get to grips with pandas—a versatile and high-performance
Python library for data manipulation, analysis, and discovery

❸ PARSE the Data

Codealong | Tidying up (more) the SF housing dataset

Unit Project 1

Lab

Introduction to pandas

Review

Review

You should now be able to:

- Setup and manage your personal GitHub repository for submitting assignments
- Define a problem and types of data
- Identify dataset types
- Apply the data science workflow in the *pandas* context
- Write an iPython notebook to import, format, and clean data using the *pandas* library

DS

Q & A

Before Next Class

Before Next Class

- Projects
  - **Unit Project 1 (due next time on 5/12)**
  - Final Project 1 (due 2 weeks from now on 5/24)

Next Class

Statistics Fundamentals

Learning Objectives

After the next lesson, you should be able to:

- ID variable types
- Use the *pandas* (and *NumPy*) libraries to analyze datasets using basic summary statistics: mean, median, mode, max, min, quartile, inter-quartile range, variance, standard deviation, and correlation
- Create data visualizations – including: boxplots, histograms, and scatter plots – to discern characteristics and trends in a dataset

Exit Ticket

Don’t forget to fill out your exit ticket here

Slides © 2016 Ivan Corneillet Where Applicable
Do Not Reproduce Without Permission