Reading Data Using Python

Zachary Hafen and Gion Matthias Schelbert

Purpose

Most modern data analysis is done using programming, and one of the simplest yet most powerful programming languages is Python. This lesson teaches how to download a data set of the students' choosing, and then load that data into Python, the first step in any data analysis that uses Python. Students will become aware of one of the many data sources available on the internet, as well as how feasible it is to access that data. In addition, students will become more comfortable using Python, and manipulating technical files on computers.

Overview

The lesson will be divided up into three in-class parts. During the first part the instructor will demonstrate downloading data and then loading it into Python. This involves going to an online data portal for data collected by the City of Chicago, choosing a data set, downloading it, opening up a Python notebook on the instructor's computer, and then loading the data into the notebook. During the second part the instructor will repeat what she/he did in the first part, but will be guided through each step by the class, instead of leading the process. During the third part the students will do the process on their own, and then turn in the end result. Before class it's recommended that the students attempt the activity individually, not under the expectation that they will succeed, but such that they become more familiar with the overall setup.

Student Outcomes

Students will be able to:

- Navigate to a data portal (in this case the City of Chicago data portal)
- Browse through data sets on a data portal
- Download a data set of interest in a useable format (.csv in this case)
- Locate a data set on the computer they are using and move the data set to an ideal location
- Start a Python notebook
- Execute cells in a Python notebook
- Load a data set into a Python notebook
- Display very basic features of a data set in a Python notebook

Standards Addressed

NGSS Practices: "Analyzing and Interpreting Data", "Using Mathematics and Computational Thinking", and "Obtaining, Evaluating, and Communicating Information".
NGSS Standards:
HS-PS2-1 Motion and Stability: Forces and Interactions, i.e. analyzing data to evaluate the claim of Newton's second law, which can be done well in Python using imported data.

Reach for the Stars is a GK-12 program supported by the National Science Foundation under grant DGE-0948017. However, any opinions, findings, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the Foundation.

HS-LS3-3 Heredity: Inheritance and Variation of Traits, i.e. applying statistics to explain the distribution of traits in a population, which requires a data set on which to apply statistics, and may be enhanced by the many statistical tools available in Python.
HS-LS4-3 Biological Evolution: Unity and Diversity, i.e. applying statistics to support explanations that organisms with an advantageous heritable trait tend to increase in proportion to organisms lacking this trait, which requires a data set on which to apply statistics, and may be enhanced by the many statistical tools available in Python.
HS-ESS1-4 Earth's Place in the Universe, i.e. using computational representations to predict the motion of orbiting objects, which may involve loading the data accurately from an external data set.
HS-ESS2-2 Earth's Systems, i.e. analyzing geoscience data to understand how one change on Earth's surface affects other Earth systems.
HS-ESS3-5 Earth and Human Activity, i.e. analyzing geoscience data to understand climate change.
HS-ESS3-5 Earth and Human Activity, i.e. using a computational representation to illustrate the behavior of Earth's systems, which may require loading and importing data.
HS-ETS1-4 Engineering Design, i.e. using a computer simulation to model real world systems, one part of which is checking the robustness of your model by comparing to data.

Time

One class period, with students encouraged to spend up to an hour of additional time outside of class, prior to class, and possibly an hour of additional time after class for students unable to finish in class.

Level

11th Grade Science, especially science classes involving some programming.

Materials and Tools

One computer per student or group of students.

- The computer must have internet access, especially to the City of Chicago Data Portal .
- The computer must be able to run Python notebooks (this can be done by installing Enthought Canopy).
- The computer is recommended to have Google Chrome installed (this is because Python notebooks run in browsers, and Google Chrome is one that is compatible with Python notebooks).

One computer connected to a projector, for the instructor (same requirements as the student computers).

The example Python notebook and data set, found in this folder:
https://northwestern.box.com/s/n2idz6f0ayg7jdm0g40wfuuyc7ezqkyk

Preparation

The teacher should make sure that each student (or each group of students) has access to a computer that can run Python notebooks. The recommended way to do this is to install Enthought Canopy (a free Python software package) on each computer.
Also, the teacher should fully download and load a Python data set themselves in advance (following the instructions), using one of the computers the students will use. This is for the same reason a science lab teacher would test the equipment before giving it to the students: even if the teacher fully understands each step in principle, each computer has its own quirks, and understanding how to address those is

important, so that the teacher can demonstrate that to the students and/or point them in the right direction.

Prerequisites

Students should have some basic familiarity with Python prior to this lesson, and be comfortable working with basic programming concepts, such as variables. Students should also be able to navigate around the file system for a given computer, and know how to move files around in that computer.

Background

Data analysis allows people to extract from a catalogue of information interesting trends, behaviors, and patterns. More importantly, those trends can then be used to inform real, every-day decisions, and provide an evidence-grounded view of the world. Modern data analysis is done using programming, and one of the simplest yet most powerful programming languages is Python. To perform data-analysis in Python, one must first load the data, which is the purpose of this lesson.

Teaching Notes

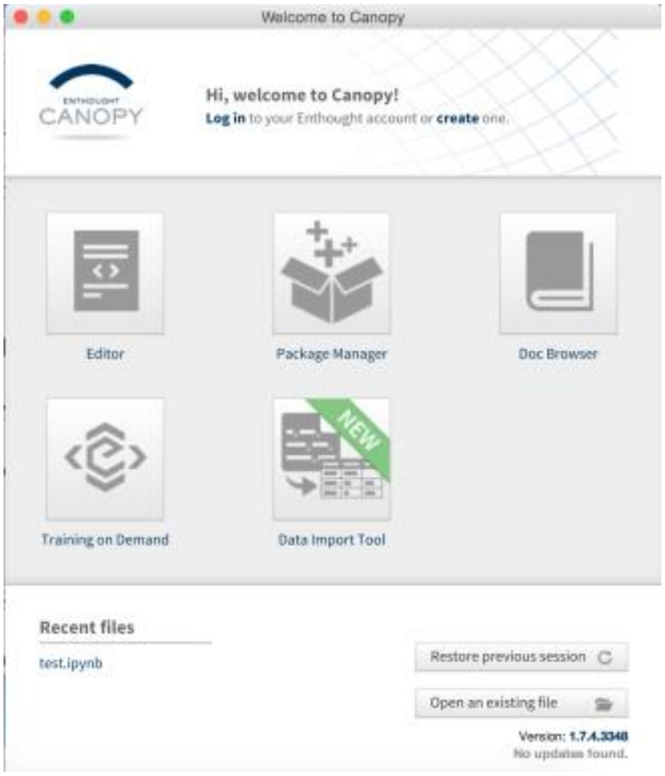
The lesson consists of going through steps I-III (listed below) three to four times, in different contexts. The reason the steps are repeated is because the process itself is somewhat messy, even if it's not particularly complicated, and it's easy to get lost.

- Before the first honest attempt to go through steps I-III, have the students try to do it themselves at home, just to become familiar with the challenges at hand. Successful students can help other students around them during the class period.
- The first time through, the instructor demonstrates the steps to the class by hooking his/her computer up to the projector and going through each step in turn, with explanation.
- During the second time through the instructor should still use his/her computer to show each step on the projector, but should solicit the students for directions. If the students are stuck on what to do next, then the instructor can step in and inform the students. It's recommended to use a different data set than was used for the first time through, just to show that it's not hard to adapt the process (one option is to display the data sets, and then have the students quickly vote for a favorite).
- The third time through the students will be in charge of going through the steps independently, on their own computers, using data sets chosen by each student/set of students. Students can either work individually, or in groups. If in groups, one recommendation is to have one person responsible for understanding and recording the steps, one to two people responsible for actually inputting commands into the computer, and one person responsible for getting the group back on track if they get lost (e.g. responsible for debugging).

<u>Pre-I: Running and Looking at the Example Notebook (Instructor Only):</u>

This step is just so the teacher can see what the end result looks like, for use as a guide.

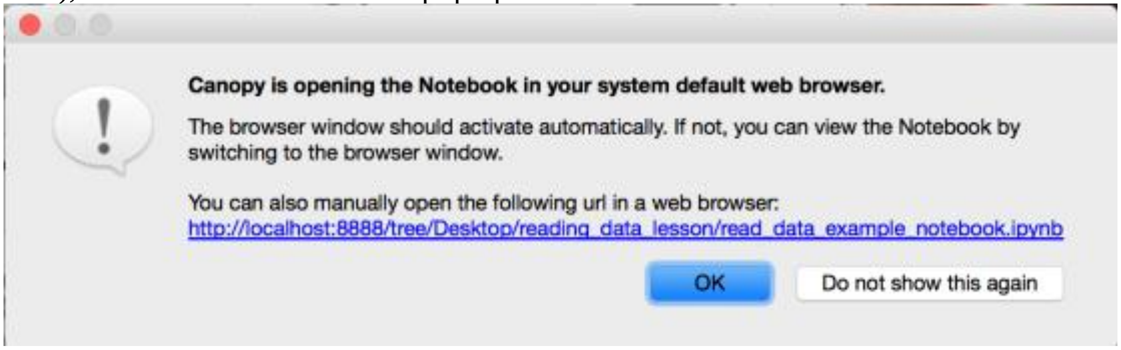
1. Download the [Reading Data with Python]() folder and move it to a convenient location (like the computer desktop).
2. Open up Canopy. You should see a window like this:



Hi, welcome to Canopy!
Log in to your Enthought account or create one.

Version: 1.7.4.3348
No updates found.

3. Click on the “Open an existing file” button in the bottom right corner, and navigate to unzipped example folder.
4. Click on the “read_data_example_notebook.ipynb” (the .ipynb is the extension for Python notebooks), and wait a minute for it to load. When successful (it can take a little bit of time to load), a window like this should pop up:



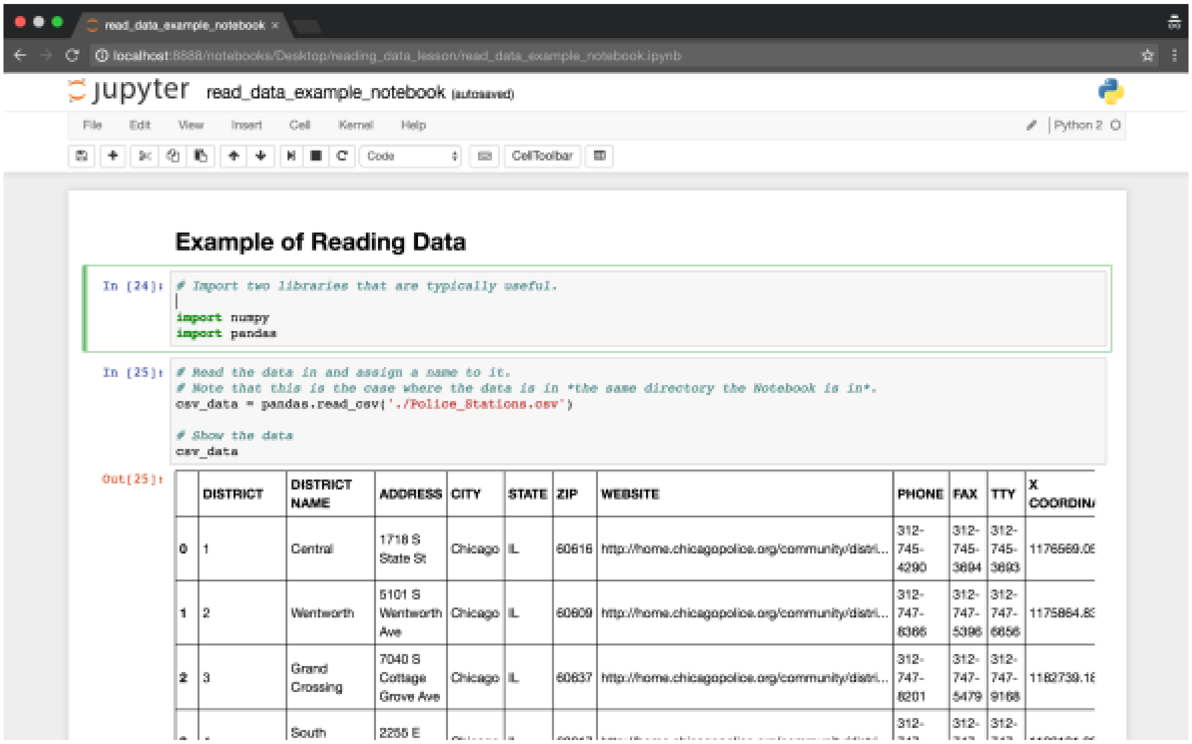
Canopy is opening the Notebook in your system default web browser.

The browser window should activate automatically. If not, you can view the Notebook by switching to the browser window.

You can also manually open the following url in a web browser:
http://localhost:8888/tree/Desktop/reading_data_lesson/read_data_example_notebook.ipynb

Simultaneously, a window should open in your default web browser, which is where you can access the Python notebook. *Some web browsers work better than others for Python Notebooks: Chrome works well, while Internet Explorer typically does not.*

The opened Python notebook should look like this:



read_data_example_notebook

localhost:8888/notebooks/Desktop/reading_data_lesson/read_data_example_notebook.ipynb

jupyter read_data_example_notebook (autosaved)

File Edit View Insert Cell Kernel Help | Python 2

Code | CellToolbar

Example of Reading Data

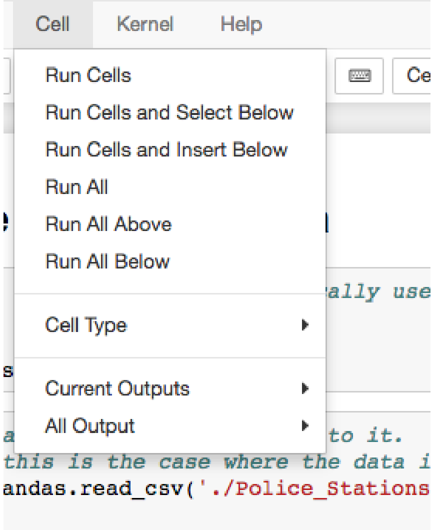
```
In [24]: # Import two libraries that are typically useful.

         import numpy
         import pandas
```

```
In [25]: # Read the data in and assign a name to it.
         # Note that this is the case where the data is in *the same directory the Notebook is in*.
         csv_data = pandas.read_csv('./Police_Stations.csv')

         # Show the data
         csv_data
```

Out[25]:

	DISTRICT	DISTRICT NAME	ADDRESS	CITY	STATE	ZIP	WEBSITE	PHONE	FAX	TTY	X COORDIN
0	1	Central	1718 S State St	Chicago	IL	60616	http://home.chicagopolice.org/community/distri...	312-745-4290	312-745-3694	312-745-3693	1176569.05
1	2	Wentworth	5101 S Wentworth Ave	Chicago	IL	60609	http://home.chicagopolice.org/community/distri...	312-747-8366	312-747-5396	312-747-6656	1175864.83
2	3	Grand Crossing	7040 S Cottage Grove Ave	Chicago	IL	60637	http://home.chicagopolice.org/community/distri...	312-747-8201	312-747-5479	312-747-9168	1182739.18
		South	2255 E					312-	312-	312-	

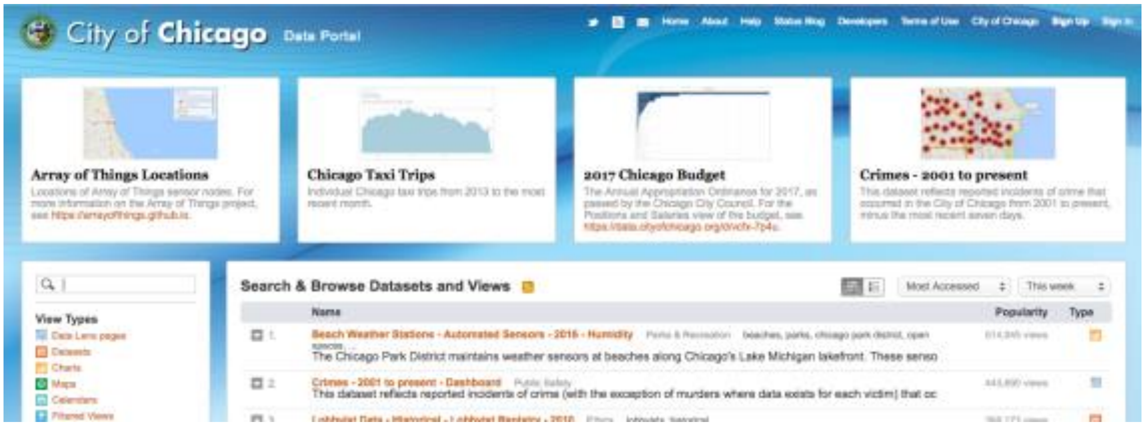
5. Mess around with the opened notebook for a bit, to get the feeling of things. The Python notebook consists of a number of different “cells”, each of which can be selected and edited, and run independently. When a cell is run, the results are then stored in memory and can be used by any of the other cells. To run a cell, select it and then press SHIFT+ENTER. Try running all cells, as well. To run all cells, select the “Cell” menu and then “Run All”, as seen here:


Cell
Kernel
Help
Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output
Ce
ally use
to it.
this is the case where the data i
andas.read_csv('./Police_Stations

I. Obtaining the Data:

This is the first step that both the students and the teacher will do.

1. Navigate on the web to the data portal, https://data.cityofchicago.org/
2. The list of available data sets should be immediately visible under the heading “Search & Browse Datasets and Views”, like as shown here:



City of Chicago Data Portal
Home About Help Status Blog Developers Terms of Use City of Chicago Sign Up Sign In
View Types
Data Lens pages
Datasets
Charts
Maps
Calendars
Filtered Views
Most Accessed
This week
Name
Popularity
Type

Array of Things Locations
Locations of Array of Things sensor nodes. For more information on the Array of Things project, see https://arrayofthings.github.io.

Chicago Taxi Trips
Individual Chicago taxi trips from 2013 to the most recent month.

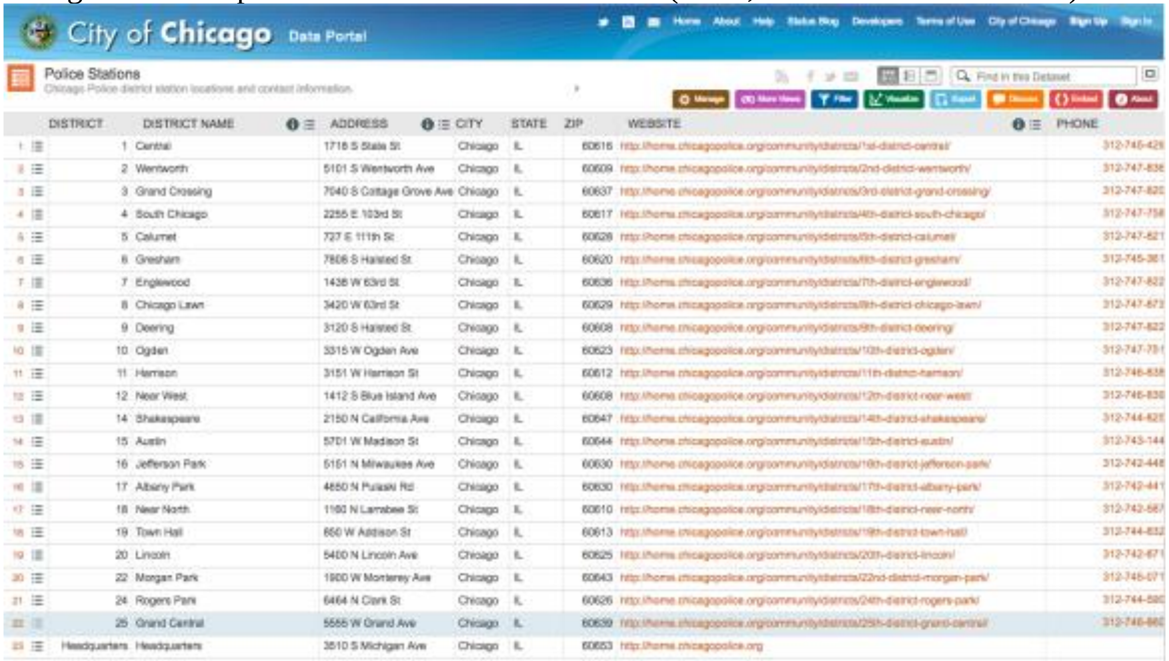
2017 Chicago Budget
The Annual Appropriation Ordinance for 2017, as passed by the Chicago City Council. For the Positions and Salaries view of the budget, see https://data.cityofchicago.org/d/vcfx-7p4u.

Crimes - 2001 to present
This dataset reflects reported incidents of crime that occurred in the City of Chicago from 2001 to present, minus the most recent seven days.

Search & Browse Datasets and Views

1. Beach Weather Stations - Automated Sensors - 2016 - Humidity — Parks & Recreation — beaches, parks, chicago park district, open spaces ... — The Chicago Park District maintains weather sensors at beaches along Chicago's Lake Michigan lakefront. These senso
2. Crimes - 2001 to present - Dashboard — Public Safety — This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that oc
3. Lobbyist Data - Historical - Lobbyist Registry - 2016 — Ethics — lobbyists, historical

3. Click on the link for a data set of interest (for the example notebook, I chose Police Stations). Doing so should open a window that looks like this (if not, then choose another file):

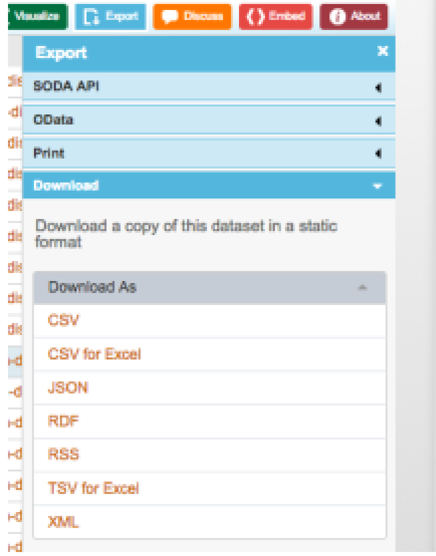


City of Chicago Data Portal
Home About Help Status Blog Developers Terms of Use City of Chicago Sign Up Sign In
Find in this Dataset
Manage
More Views
Filter
Visualize
Export
Discuss
Embed
About

Police Stations
Chicago Police district station locations and contact information.

	DISTRICT	DISTRICT NAME	ADDRESS	CITY	STATE	ZIP	WEBSITE	PHONE
1	1	Central	1718 S State St	Chicago	IL	60616	http://home.chicagopolice.org/community/districts/1st-district-central/	312-745-429
2	2	Wentworth	5101 S Wentworth Ave	Chicago	IL	60609	http://home.chicagopolice.org/community/districts/2nd-district-wentworth/	312-747-838
3	3	Grand Crossing	7040 S Cottage Grove Ave	Chicago	IL	60637	http://home.chicagopolice.org/community/districts/3rd-district-grand-crossing/	312-747-820
4	4	South Chicago	2255 E 103rd St	Chicago	IL	60617	http://home.chicagopolice.org/community/districts/4th-district-south-chicago/	312-747-758
5	5	Calumet	727 E 111th St	Chicago	IL	60628	http://home.chicagopolice.org/community/districts/5th-district-calumet/	312-747-821
6	6	Gresham	7808 S Halsted St	Chicago	IL	60620	http://home.chicagopolice.org/community/districts/6th-district-gresham/	312-745-381
7	7	Englewood	1438 W 63rd St	Chicago	IL	60636	http://home.chicagopolice.org/community/districts/7th-district-englewood/	312-747-822
8	8	Chicago Lawn	3420 W 63rd St	Chicago	IL	60629	http://home.chicagopolice.org/community/districts/8th-district-chicago-lawn/	312-747-873
9	9	Deering	3120 S Halsted St	Chicago	IL	60608	http://home.chicagopolice.org/community/districts/9th-district-deering/	312-747-822
10	10	Ogden	3315 W Ogden Ave	Chicago	IL	60623	http://home.chicagopolice.org/community/districts/10th-district-ogden/	312-747-731
11	11	Harrison	3151 W Harrison St	Chicago	IL	60612	http://home.chicagopolice.org/community/districts/11th-district-harrison/	312-746-838
12	12	Near West	1412 S Blue Island Ave	Chicago	IL	60608	http://home.chicagopolice.org/community/districts/12th-district-near-west/	312-746-830
13	14	Shakespeare	2150 N California Ave	Chicago	IL	60647	http://home.chicagopolice.org/community/districts/14th-district-shakespeare/	312-744-825
14	15	Austin	5701 W Madison St	Chicago	IL	60644	http://home.chicagopolice.org/community/districts/15th-district-austin/	312-743-144
15	16	Jefferson Park	5151 N Milwaukee Ave	Chicago	IL	60630	http://home.chicagopolice.org/community/districts/16th-district-jefferson-park/	312-742-448
16	17	Albany Park	4650 N Pulaski Rd	Chicago	IL	60630	http://home.chicagopolice.org/community/districts/17th-district-albany-park/	312-742-441
17	18	Near North	1160 N Larrabee St	Chicago	IL	60610	http://home.chicagopolice.org/community/districts/18th-district-near-north/	312-742-587
18	19	Town Hall	850 W Addison St	Chicago	IL	60613	http://home.chicagopolice.org/community/districts/19th-district-town-hall/	312-744-832
19	20	Lincoln	5400 N Lincoln Ave	Chicago	IL	60625	http://home.chicagopolice.org/community/districts/20th-district-lincoln/	312-742-871
20	22	Morgan Park	1900 W Monterey Ave	Chicago	IL	60643	http://home.chicagopolice.org/community/districts/22nd-district-morgan-park/	312-745-071
21	24	Rogers Park	6464 N Clark St	Chicago	IL	60626	http://home.chicagopolice.org/community/districts/24th-district-rogers-park/	312-744-580
22	25	Grand Central	5555 W Grand Ave	Chicago	IL	60639	http://home.chicagopolice.org/community/districts/25th-district-grand-central/	312-746-860
23	Headquarters	Headquarters	3510 S Michigan Ave	Chicago	IL	60653	http://home.chicagopolice.org	

4. Click the blue "Export" button, and choose the top option from the Download tab, "CSV", as shown here:



Download a copy of this dataset in a static format

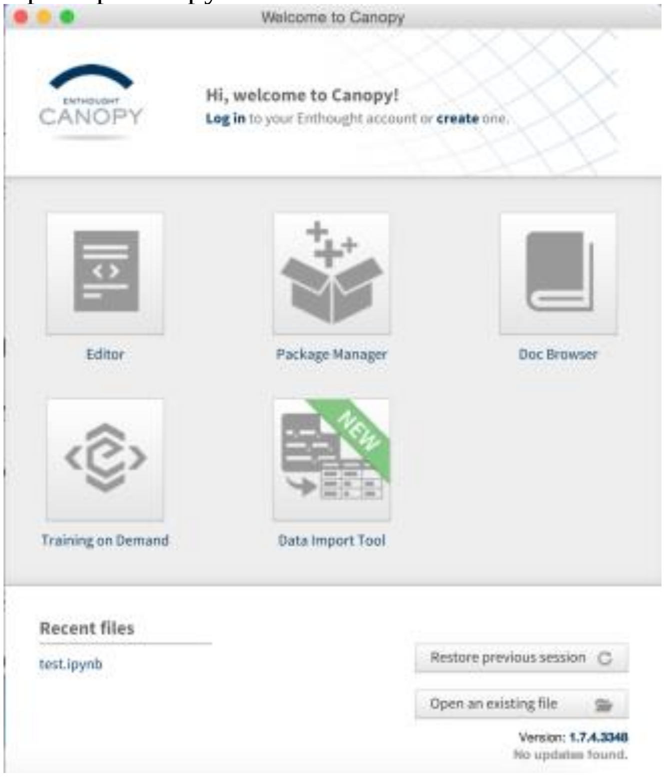
   Note that you can use other data types, but CSV (.csv; comma separated values) is one of the easiest to deal with, so I recommend using data in that format.

5. Locate where the file was downloaded onto the computer, e.g. the “Downloads” folder.

II. Creating a New Python Notebook:

Here I will describe how to open a Python notebook, assuming you are using Canopy. There are other ways to open Python Notebooks, for example in your computer terminal (the preferred method by most researchers), but I won’t discuss those here.

1. Open up Canopy. You should see a window like this:



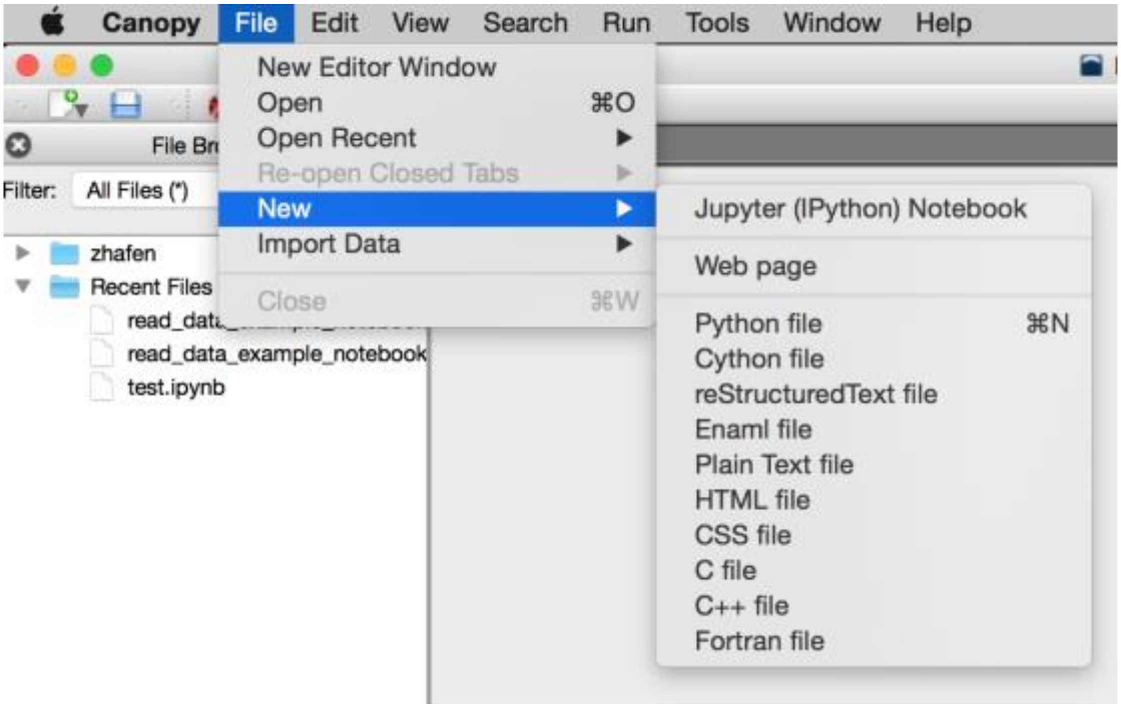
Welcome to Canopy

Hi, welcome to Canopy!
Log in to your Enthought account or create one.

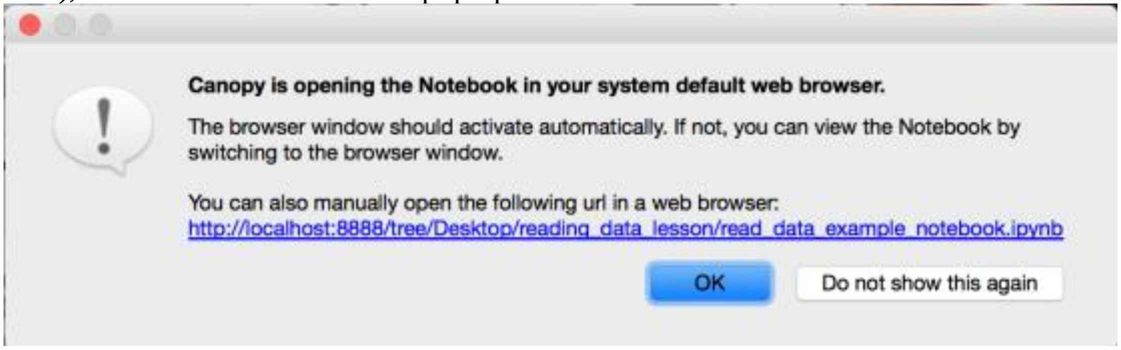
Recent files
test.ipynb

Version: 1.7.4.3348
No updates found.

2. Select the “Editor” button. A new window will open up. From the Editor window, go to File->New and select Jupyter (IPython) Notebook. It will be the top option, as shown below.



3. Choose a name (e.g. ‘read_data’) and save location that are convenient. The computer desktop is usually sufficient for these purposes.
4. Wait a minute for your new notebook to load. When successful (it can take a little bit of time to load), a window like this should pop up:



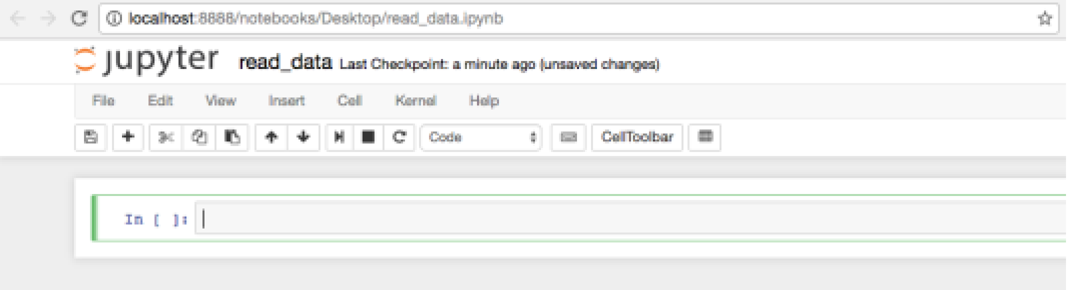
Canopy is opening the Notebook in your system default web browser.

The browser window should activate automatically. If not, you can view the Notebook by switching to the browser window.

You can also manually open the following url in a web browser:
http://localhost:8888/tree/Desktop/reading_data_lesson/read_data_example_notebook.ipynb

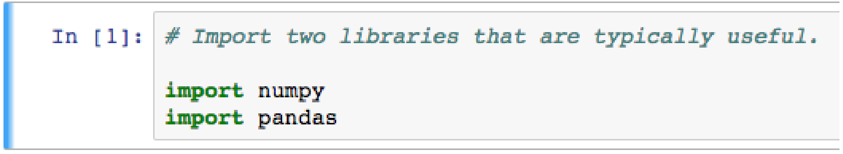
Simultaneously, a window should open in your default web browser, which is where you can access the Python notebook. *Some web browsers work better than others for Python Notebooks: Chrome works well, while Internet Explorer typically does not.* As the message states, you can also open the notebook manually in the web browser of your choosing by copying and pasting the link into that web browser.

5. You should now see something like the following in your browser:


localhost:8888/notebooks/Desktop/read_data.ipynb
jupyter
read_data Last Checkpoint: a minute ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Code
CellToolbar
In [ ]:

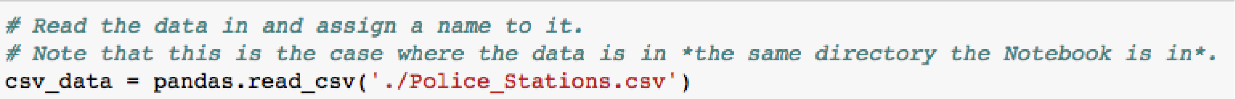
III. Loading the Data into the Python Notebook

1. We'll first import libraries that are useful for loading and analyzing data. These libraries are NumPy (numerical Python; numpy.org) and pandas (Python data analysis library; pandas.pydata.org). Click inside the first cell, and import them (the explicit commands are shown in the screenshot below).



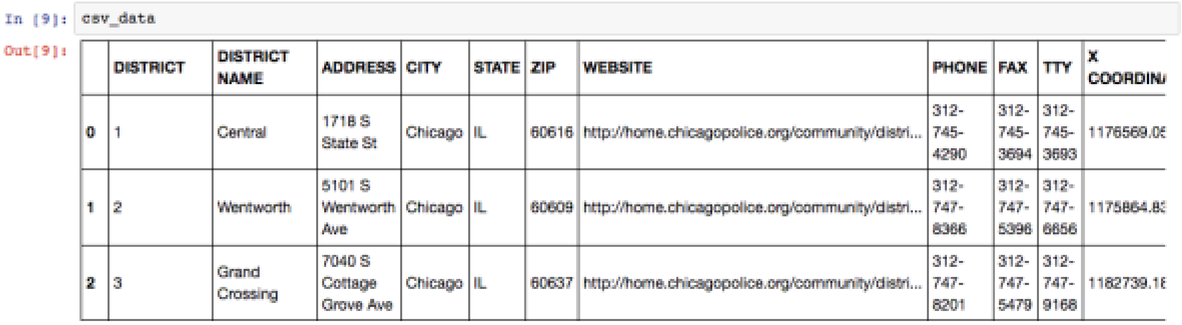
```
In [1]: # Import two libraries that are typically useful.

        import numpy
        import pandas
```

2. Cells are run individually by pressing SHIFT+ENTER. While the first cell is still selected, press SHIFT+ENTER to run it. Now both NumPy and Pandas are loaded and available for use in the Python notebook (you'll know it was successful if there is now a number next to the cell containing the code for importing).
3. Move the data set you downloaded earlier *to the same folder* as your Python notebook. This is crucial, and a common mistake. (Note that it is possible to have the data in a different location, however that requires more advanced knowledge which I will not address here.)
4. We'll now actually load the data in, using the read_csv() function from pandas. The argument you will pass to the function is a string consisting of './<name of your data set>'. The "./" means it will look for the file in the folder the Python notebook is stored in. Here's an example of the actual syntax:



```
# Read the data in and assign a name to it.
# Note that this is the case where the data is in *the same directory the Notebook is in*.
csv_data = pandas.read_csv('./Police_Stations.csv')
```

   In this case, the data is read in, and then stored as a variable for easy access, "csv_data." Don't forget to press SHIFT+ENTER (while the cell is selected) after putting things into a cell, in order to actually run that cell.
5. Finally, you can do a basic printout of what the data looks like by just entering the name of the stored variable by itself, as such:



```
In [9]: csv_data
Out[9]:
```

	DISTRICT	DISTRICT NAME	ADDRESS	CITY	STATE	ZIP	WEBSITE	PHONE	FAX	TTY	X COORDIN
0	1	Central	1718 S State St	Chicago	IL	60616	http://home.chicagopolice.org/community/distri...	312-745-4290	312-745-3694	312-745-3693	1176569.0
1	2	Wentworth	5101 S Wentworth Ave	Chicago	IL	60609	http://home.chicagopolice.org/community/distri...	312-747-8366	312-747-5396	312-747-6656	1175864.8
2	3	Grand Crossing	7040 S Cottage Grove Ave	Chicago	IL	60637	http://home.chicagopolice.org/community/distri...	312-747-8201	312-747-5479	312-747-9168	1182739.1

6. Save the notebook by pressing the save button in the upper left corner.
7. That's it! There are no further steps. The data is loaded in successfully if you're able to display the above output. However, if you would like to see an example of how to use the loaded data to pull out some interesting statistics, look at the example notebook.

Assessment

At the end of the class period the students will be responsible for turning in a working notebook that has successfully loaded a data set. It is up to the instructor's discretion on whether or not to grade it. It is recommended that students who do not complete the assignment during class do it as homework.

Additional Information

For those struggling with Python, a great resource is Codecademy.