

Computer Science CS134C (Fall 2018)

Duane A. Bailey

Laboratory 5

Presenting Data (due noon, Thursday/Friday)

Objective. To learn how to set up a personalized environment for data presentation.

This week we will experiment with using Python to present data. We'll use the module `matplotlib` to present our data as a graph. Before we get started, however, we'll have to install this tool in our own directory. This is a simple process that you must do once for each computer you use. Once the tools have been installed, you may choose to *activate* them, or not.

Virtual Environments & Installing Additional Packages.

Few Python environments have *all* the modules you might want to eventually have installed. This week we'll use the *virtual environment* system to allow us to experiment with adding new modules. First, we set up and *activate* the environment, then we install the new software. Here are the steps:

1. We assume that you're using Python 3.6 or greater. You can find python's version with

```
-> python3 --version
Python 3.7.0
```

2. Go to your home directory and create the environment in your `cs134` folder (the `->` is a prompt):

```
-> python3 -m venv --system-site-packages ~/cs134
```

The phrase `--system-site-packages` indicates that you want to *extend* the environment you already have set up on your machine. The specification of the directory simply indicates where the environment is to be installed. Installing it in your `cs134` folder makes it easy to access in all of your future labs.

3. We now go into our `cs134` folder and *activate* this new environment:

```
-> cd cs134
-> source bin/activate
(cs134) -> python --version
Python 3.7.0
(cs134) ->
```

If the activation worked, you'll see your usual prompt, prefixed with the name of the folder that contains your virtual environment. In addition, `python` now refers to `python3`. The `source` command is simply saying "execute the commands found in the `activate` script in the `bin` subdirectory." Those commands changed your prompt, and told `bash` where to install new modules.

Once an environment is activated, you can de-activate it (if you want to) by typing `deactivate`. Don't do that now, or everything will go back to `python`'s normal behavior, with `python` referring to version 2 of Python and not 3 and not knowing how to install new modules..

4. New modules are installed with pip (a silly recursive acronym for “Pip Installs Python”). We would like to install the non-standard package, matplotlib:

```
(cs134) -> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.0.0...
    100% |#####| 14.1MB 17.MB/s
... output omitted ...
Installing collected packages:...matplotlib
Successfully installed...matplotlib-3.0.0...
(cs134) ->
```

The first time you import matplotlib is does some one-time initialization, like building fonts that work specifically with your machine. We'll force matplotlib to build those fonts now:

```
(cs134) -> python
>>> import matplotlib.pyplot as plt
...
UserWarning: Matplotlib is building.... This may take a moment.
...
>>> quit()
```

The matplotlib environment is now installed and ready for use. You'll be able to use matplotlib for any new code you write, as long as you activate the environment at the beginning of each new session (*i.e.* whenever you log in, open a Terminal window, *etc.*).

The remainder of the lab will generate a plot or two. Let's grab some starter files by cloning the lab5 repository (use your CS id instead of 22xyz9):

```
(cs134) -> cd ~/cs134
(cs134) -> git clone ssh://22xyz9@davey.cs.williams.edu/~cs134/22xyz9/lab5.git lab5
```

Required Tasks. We encourage you to learn more by investigating the matplotlib tutorial at

http://matplotlib.org/users/pyplot_tutorial.html

1. First, we'll experiment together with `matplotlib.pyplot`. We'll show you how to draw a simple graph, to get you started. The general approach is similar to the following:

```
# at the top:
import matplotlib.pyplot as plt
... compute two lists of values whose relationship you wish to plot ...
n = length of the lists
plt.title('Put your title here')
plt.xlabel('Description of the independent variable')
plt.ylabel('Description of the dependent variable')
plt.plot(independent, dependent, 'k-') # black lines
plt.savefig('PlotFileName.pdf') # always use pdf files for cs134
```

2. Live Poll Data. Recently, the New York Times and Siena College have taken to producing *live polling data*. These polls, available at

<https://www.nytimes.com/interactive/2018/upshot/elections-polls.html>

provide polling results in *real time*, as the polling occurs. When the poll is complete, they publish the *microdata* that documents the responses of each caller, in the order the calls are made.

In the lab5 directory, you'll find `txsen-2.csv`. This file is associated with a live poll of 800 people in Texas from October 8 to October 11, responding to the race between Ted Cruz and Beto O'Rourke. You can use this csv file, or download another completed poll from the New York Times.

Your job is to plot the Democratic (blue) and Republican (red) party trends on a single graph. Each data point indicates the percentage of voters indicating they'll vote for a specific party. Be aware that some respondents will not indicate a preference for either party. Their preference will not be explicitly displayed, but will reduce the percentage of votes earned by the other two parties. You might think about why the curves are shaped the way they are. When you are finished, make sure the README file is updated so we can see which race you have decided to use.

Remember to add and commit your (1) specific CSV file, (2) your plotting code, (3) the PDF of your plot, and (3) your updated README.

3. Optional: other trends. If you'd like to get a bit more credit, investigate some other time-oriented data of your choice. You could, for example, use another column from your specific live polling microdata, or you could pick an entirely different domain. We also give credit for demonstrating mastery of some new skills in `matplotlib`. This visual should be submitted as `trend.pdf`. Please add some notes in README describing the origin of the dataset you used to generate `trend.pdf` and a sentence or two describing what's being displayed.

That's it! *Good luck!*

★