CME 252: Homework 0
Due: Not Graded

AJ Friend

We'll use this homework to get everyone set up with the Python environment you'll use for the whole class, and do a test run for homework submission.

Part 1: Python Environment

We'll want to get the following things working:

- Python interpreter
- "SciPy stack" python packages:
  - `numpy`
  - `scipy`
  - `matplotlib`
  - `ipython`
  - `jupyter`
- `cvxpy`

It doesn't matter **how** you get these working, as long as you do. I'll outline **one** possible path to get these working below.

As a quick test, you can skip this part if you can download, open on your own computer, and run all the cells in the example Time Series Smoothing notebook. If you have any trouble, this homework should help. We can also iron out installation issues in class, in office hours, or on Piazza.

Step 1. Install Anaconda Python distribution

Anaconda is a Python distribution that includes Python itself, and many useful packages (including the ones we need: `numpy`, `scipy`, `matplotlib`, `jupyter`, and `ipython`). Installing Anaconda is probably the easiest way (especially for beginners) to get a working setup.

- Go to https://www.continuum.io/downloads, download the installer for **Python 3** for your operating system, and run it. The code we use in this

class will probably work for Python 2.7 (and you can complain to me if it doesn't), but I'll be using 3, so that's your safest bet.
- Open a **new** terminal window and run `conda info` to ensure that everything is working.

Step 2. Install `cvxpy`

- Run `pip install cvxpy`.

Step 3. Check that everything is working

- Download the Time Series Smoothing notebook to some directory on your computer.
- In a terminal window, and in the same directory where you downloaded the notebook, type `jupyter notebook`. This should open up a new browser window.
- Locate `smooth.ipynb` in the list of files and open it.
- You should be looking at a Jupyter (previously named to "IPython") notebook. It's just an interactive Python interpreter that allows you to split code into cells and execute them independently. It also allows for inline plotting of images, and inclusion of descriptive text (and LaTex for math). I find it useful both for experimenting with code, and for explaining concepts. I'll give many examples using notebooks throughout the class.
- Execute the cells, from first to last, to make sure that `numpy`, `matplotlib`, and `cxvpy` are all working. On my computer, I execute a cell with `Shift-Enter`. A cell is busy computing an answer if there is an asterisk (*) next to it.
- If you got pretty plots and no error messages, then everything works! If you have any trouble, please ask on Piazza, in class, or in office hours.

Part 2: Homework Submission

I'll use this initial and **ungraded** homework as a trial run to see how well homework submission and autograding will work.

- Please download `hw0.py` and `grade_hw0.py` from the Homework page of the course website and place them in the same directory.
- From that directory run `python grade_hw0.py` and notice that you currently have 0 out of a possible 2 points.
- Modify `hw0.py` according to the instructions in the file so that you pass the tests to get the full 2 points.

- Once you're happy with your `hw0.py`, please submit it to this Dropbox File request link: https://www.dropbox.com/request/KoUUxj6Oy9j2gNh98hZ8.
- Please keep the filename the same, `hw0.py`, but do make sure your name and Stanford email are listed properly in `hw0.py`.
- I'm planning on providing a `grade_hwX.py` file for each homework as a **sanity check**. Getting full credit with the grading file on your local machine doesn't guarantee that I'll grant full credit when I grade it. I'll likely use a slightly different grading script, and may look at the source code individually to determine a grade.