Tutorial

May 26, 2019

1 Tutorial: Sage and Jupyter notebooks

When studying linear algebra, it's vitally important that you learn to use a computer to perform important algebraic operations. The reason linear algebra is an increasingly important mathematical tool is because computers faciliate computations that are not accessible without them. In particular, typical applications may work with $10,000 \times 10,000$ matrices, and there's no way to do that without a computer.

In our class, most of the computations will be performed in Sage, which is a software environment that is designed for learning mathematics. You don't need to know this, but Sage is built in Python so some of our Sage commands will look familiar if you know Python (but it's ok if you don't!).

There are several ways to access Sage. For one, I'll show you how you can do some simple computations within a web page. You can also create Sage notebooks within cocalc, and you can install Sage for free on your own laptop though you don't certainly need to do that. However, one way to access Sage is through a Jupyter notebook, like the one you're reading now. The advantage to this format is that I can share notebooks with you that contain explanatory text (like you're reading) and specialized commands to automate some of what we're learning and that allow you to import data sets to investigate. In addition, you can save your work and come back to it at a later time.

Within a Jupyter notebook, Sage computations are performed in "cells" like the one below. To evaluate this cell, click in it and press Shift-Enter. The result of evaluating this cell will appear below.

```
In [1]: 2+2
```

```
Out[1]: 4
```

Now let's review some linear algebra. You can define the matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 1 & 0 \end{bmatrix}$ by entering

A = matrix(2,3,[1,2,3,-1,1,0])

Notice that the first number in parentheses is the number of rows, followed by the number of columns, followed by a list of the entries. Go ahead and enter this in the cell below and evaluate the cell by pressing Shift-Enter.

```
In [3]: A = matrix(2,3,[1,2,3,-1,1,0])
        A.rref()
```

```
Out[3]: [1 0 1]
        [0 1 1]
```

You probably noticed that evaluating this cell did not produce a reply message. In general, evaluating a cell will only report the result of the last line in the cell. In this case, there is only one line, but all this does is define a matrix and give it the name A. You can find the reduced row echelon form of A by evaluating

A.rref()

We can enter more than one line in a cell. In the cell below, define the matrix $B = \left[\begin{array}{rr} -2 & 1 \\ 0 & 2 \\ 1 & -1 \end{array}\right]$ on one line and add a second line to evaluate AB using

A*B.

We can also define vectors, such as $\mathbf{v} = \left[\begin{array}{r} 1 \\ -2 \\ 3 \end{array}\right]$ using

v = vector([1, -2, 3])

In the cell below, define the vector **v** and evaluate $A\mathbf{v}$ using A*v.

If A is a square matrix, we can find its inverse using A.inverse(). Define the matrix $A = \left[\begin{array}{rrr} 3 & 0 & -2 \\ 1 & -2 & -1 \\ 2 & 3 & 1 \end{array}\right]$ and use its inverse to solve the equation $A\mathbf{x} = \left[\begin{array}{r} -5 \\ -8 \\ 8 \end{array}\right]$.

Now that we've learned how to performs some basic Sage operations within a Jupyter notebook, let me demonstrate how we can pull a data set into our notebook. A frequently-studied data set consists of the measurements of 150 irises and can be found here. For each of the 150 irises, four measurements are given in centimeters. The cell below will pull in the data set, create a vector for each of the 150 flowers using two of the measurements, and then plot them. You do not need to worry about how this code works. It's enough to know that you feel comfortable executing it.

```
In [0]: import csv
        irises = csv.reader(open("iris.data"))
        data = [vector([float(flower[0]), float(flower[1])]) for flower in irises]
        list_plot(data, size=20, aspect_ratio=1)
```

One Python feature that is nice to know about is lists. Basically, a list is just what it sounds like, a list of some kind of items. To create a list, we just put the things we want in the list between square brackets and separated by commas. For instance,

heights = [68.1, 72.2, 63.5, 65.2, 71.4, 66.0]

could be a list of the heights, in inches, of six people. We can find the mean of these heights using

mean(heights).

Use the cell below to define these heights and find their mean.

We can also create a list of vectors. Define vectors v1, v2, and v3 to be $\left[\begin{array}{r} -1 \\ 2 \end{array}\right]$ \$,\$ $\left[\begin{array}{r} 0 \\ 1 \end{array}\right]$, and $\left[\begin{array}{r} 3 \\ 4 \end{array}\right]$. Put these vectors into a list and then find the average of the vectors using the `mean` command.