Plots and Linear Regression

Please do a **git pull** to update your repository with today’s lecture materials. We will cover a little more about plots today and then I will introduce to you one of the most important models in data science. A model will tell you how your data features relate to the response variable. Recall that a feature is a column of data in your DataFrame. I will talk about response variables later.

HW1 and Quiz1 solutions are also uploaded already.

[Plotting Stock Prices]

--- Here I demonstrate how to plot a time series. A time series is simply data plotted over time. Stock prices are a really good example of a time series. I also demonstrate how to set titles, axis labels, etc. ---

Below is a brief summary of the functions used:

**pd.read\_csv(“spx.csv”, index\_col=0)**

**dataframe.plot(style= “<color><linestyle>”)**

**plt.title()**

**plt.xlabel()**

**plt.ylabel()**

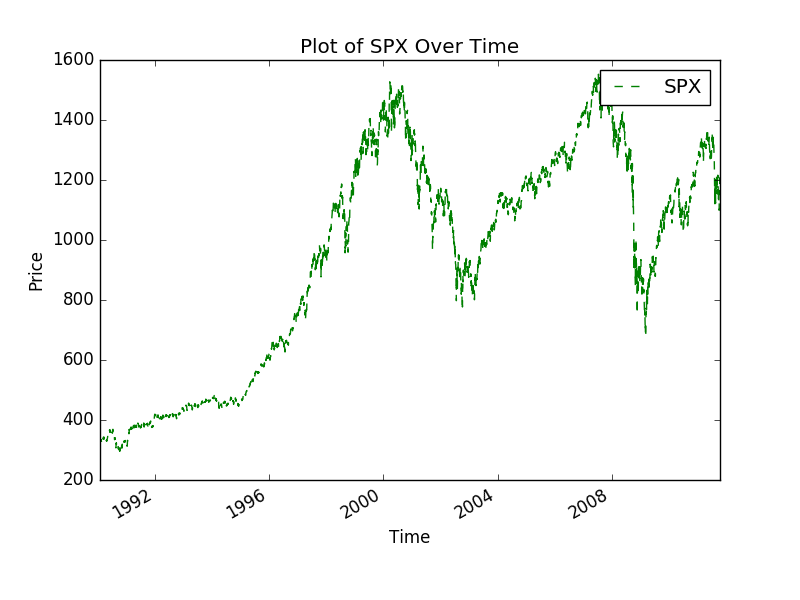
**plt.xlim()**

**plt.ylim()**

**plt.legend()**

**plt.savefig(<filename>)**

Sample Plot

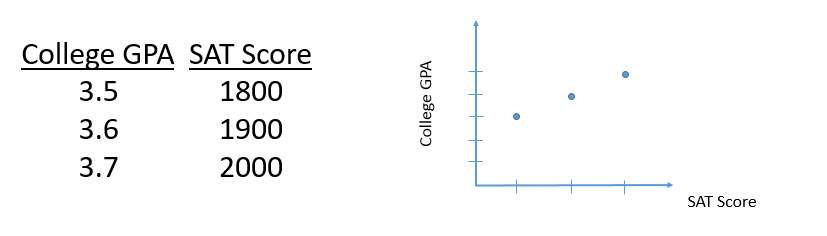


[Linear Regression]

Linear regression is one of the simplest models in machine learning and is a very natural model to build and explain. The latter point is especially important in a business setting because it is very essential to be able to talk about how you make your conclusions.

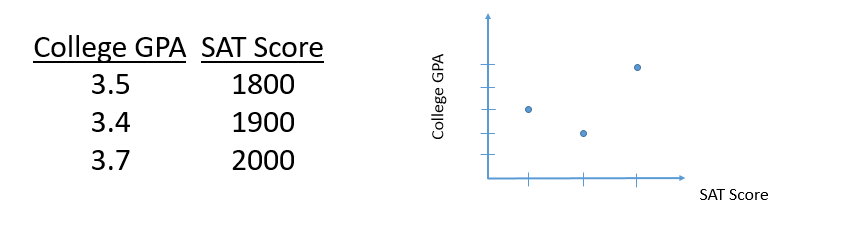
Machine learning is the art of teaching computers to learn about data for us (and often to teach it to predict into the future!).

Imagine that our data looks like the following:



What does the data tell you? Can we draw a line through all the dots? *What does this line even mean?*

Now what if our data looks like this:

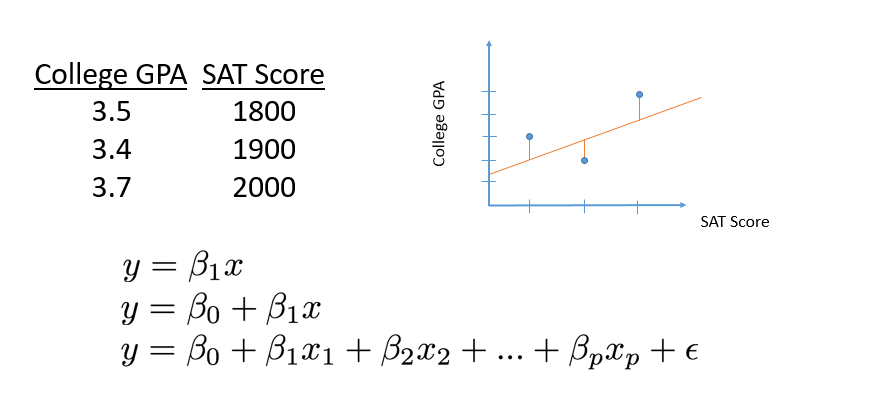


How can we draw a line through the dots? *What does this line mean now?*

We can write this line mathematically. A line can be written into an equation of something like:

**y = ax + b**

The next page of notes will show this in a slightly different way (but trust me it’s the same thing).



The important coefficient we want to find is the BETA. This is because BETA is the slope of your line and we can use this slope to answer certain questions like:

- If I scored 100 more points on the SAT, what would my college GPA be?

Solving for BETA1 is simple for one feature (First equation).

Solving for BETA1 and BETA0 requires at least two rows of data. *Why?*

Solving for more BETA’s requires even more rows of data and more time.

To solve for multiple BETA’s, we use matrix notation and calculate the following equation:



--- Here I go over a 2x2 matrix example with fake numbers ---

[The API Library]

--- Here I use the wine.csv data set found in the lecture 3 folder for real-time coding examples ---

Luckily, we do not need to handwrite these calculations ourselves. There is a library in python that does everything for us!

**import statsmodels.api as sm**

Recall that by typing **as** we are nicknaming this library with a long name to just two letters, **sm**.

The linear regression model is a function that takes in two pieces of data: 1) your X feature matrix and 2) your y response variable.

**clf = sm.OLS(<y data>, <X data>)**

The above function will initiate your linear regression model. This means that the model is given all the information it needs to run BUT HAS NOT RUN YET.

To run the model, we call its fit function using the dot notation:

**result = clf.fit()**

To print the BETA coefficients that were calculated we call the summary function using the dot notation:

**result.summary()**

[Quiz Instructions]

Quizzes and homework require you to download the appropriate files from my github repository. A repository is a storage space that you can dump all your files and data in, much like your own computer’s file system. In your case, you are going to download my files. To do so, in a separate terminal window (launch another Git Bash terminal in Windows or Command+t on Mac) type:

**git clone** [**https://github.com/jc003/datascience\_yeah.git**](https://github.com/jc003/datascience_yeah.git)

This will copy my repository/folder named “datascience\_yeah” into your computer. Explore the repository. You will find folders for homework, quizzes, and my notes. Your homework that is due next week is already there.

The quiz is now part of your homework!!! Please submit your quiz in the same way you submit your homework. You may also collaborate with other students on how to solve the quiz problems.

--------------------------------

In the future if you want to update your repository (because I made some changes to the quiz or homework file) or to grab new homework assignments type in your terminal:

**git pull**

\*If you get a merge error, that means git does not know how to update your files. This happens for example when you start writing solutions for hw1.py, but then I make changes to the blank hw1.py template to the repository. When you do “git pull”, git (the program) will look at my hw1.py file and your hw1.py file and be confused because it won’t know which lines to add or delete in the correct order. To fix, make a copy of your hw1.py (e.g. hw1\_copy.py), delete your hw1.py, type git pull, then copy your solutions from hw1\_copy.py to the fresh copy of hw1.py.

[Homework Instructions]

You have already downloaded the homework for this week. Navigate to the appropriate folder and open up **hw3.py**. Having done the readings might help for this particular assignment. In general, if you don’t know what a function does, go look it up! Doing google searches on your own should be a natural habit. Homework 3 is due before start of class next week 4/29. Please email your python code file to [jerryc@berkeley.edu](mailto:jerryc@berkeley.edu) once you are done.

Note: Any time you see “FILL ME” that is a problem worth 2 points. I give 2 points for a correct answer. 1 point for a wrong answer but good progress was made, and 0 points for leaving it blank.

Note: The homework problems are intended to be more challenging than the in-class examples or the quiz problems. I highly recommend working with other students together when doing these problems.