Matt Kennedy

CS 678

Project 1

**Using the regression equation, how many downloads would you expect at Noon on the fifth day of the next month?**

Noon of fifth day of month 2 = (24 \* 31) + (24 \* 4) + 12 = 852  
Using the equation found for regression line, y = 2.62x + 983.23, where x is 852:  
 y = (2.62 \* 852) + 983.23 = 3214.86

**Note: apparently a well-read blogger made a favorable mention of your book towards the end of the month. Do you think this simple linear analysis captures the expected popularity of yourbook? Explain. What other analytical approaches might produce a better model?**

This simple linear analysis would technically capture the unexpected popularity, but not nearly in its full magnitude; the slope of the regression line may increase, but the difference won’t be much from the original.

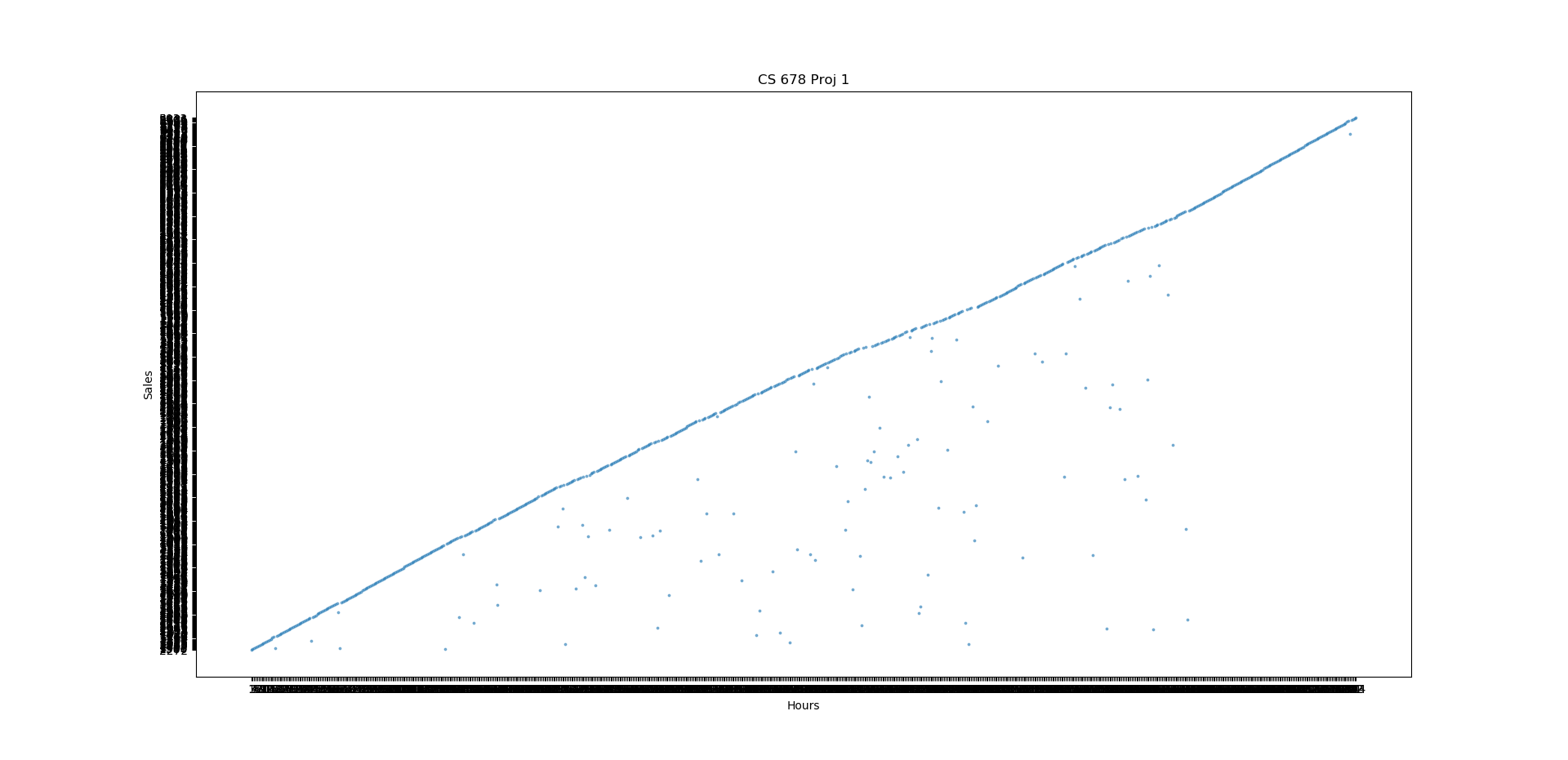
One suggestion may be to use a box-plot to view outliers better. Additionally, a box plot shows clearly the first quartile, third quartile, minimum, median, and maximum of a given dataset.

Another example, from the book, is to use a higher-order polynomial model to better fit a curve. Instead of using a poor fit linear model that doesn’t account for outliers the way we want it to, a model that is more adaptive to data and outliers is recommended. As shown by the output graphs, the higher the degree of polynomial, the more adaptive to change the curve is. (Note: I didn’t print the polynomial equation because printing a 15 degree polynomial is excessive to me).

**Design choices:**  
For this first project, I wanted to get more comfortable with Python and with MatPlotLib - I have limited experience in the former, and none in the latter. It took all of 5 minutes to get the data set loaded into a scatter plot, but I spent some additional time configuring the graph, trying to adjust axes, etc. Eventually, when I had solved the issues I was having, my design theory was to load the data into lists, then load those lists into scatter plots or use the data to create a regression model or polynomial fit. Overall, it was a good learning experience.

A few problems I faced were due to my lack of experience with Python. For example, it took me awhile to figure out why MatPlotLib was plotting all sale values out-of-order (i.e. the first value is >2000, and it put the values in the order they were available). Eventually, I figured out that python will implicitly type values read from a file to a string, which I then had to cast to an int.

Another issue was that, when loading a list of strings into a default scatter plot in MatPlotLib, every single data point will get a tick and a label on the side. Because I didn’t know at the time I was loading strings instead of integers into the scatter plot, it took some time to figure out how to adjust the axes to not be cluttered. Now I know that MatPlotLib does this automatically given a list of integers. Below is a picture of what this looks like:



For the NaN values, I decided to just remove them. If no data was available, then there is nothing to add to the graph. I did this by reverse-iterating through the list and removing all months that contain the NaN value. If the best guess needed to be calculated for a given month, the regression model could be used to do this.