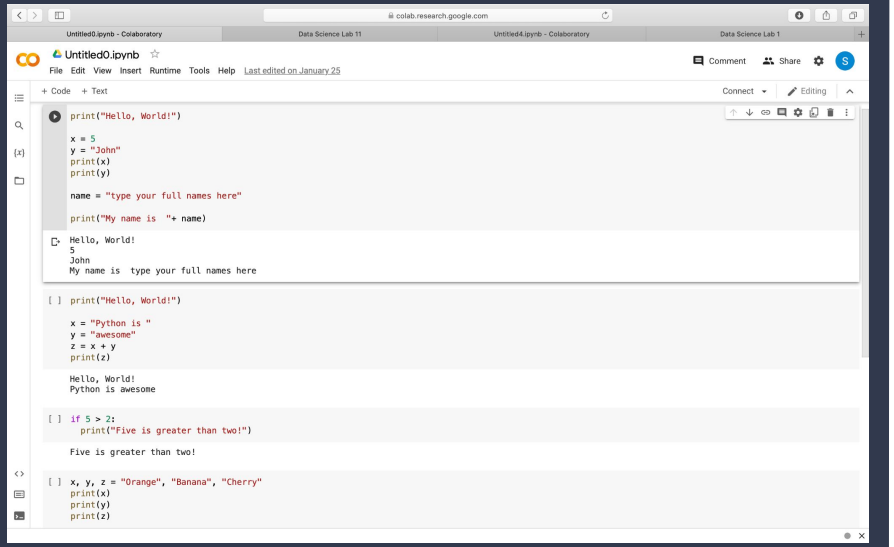
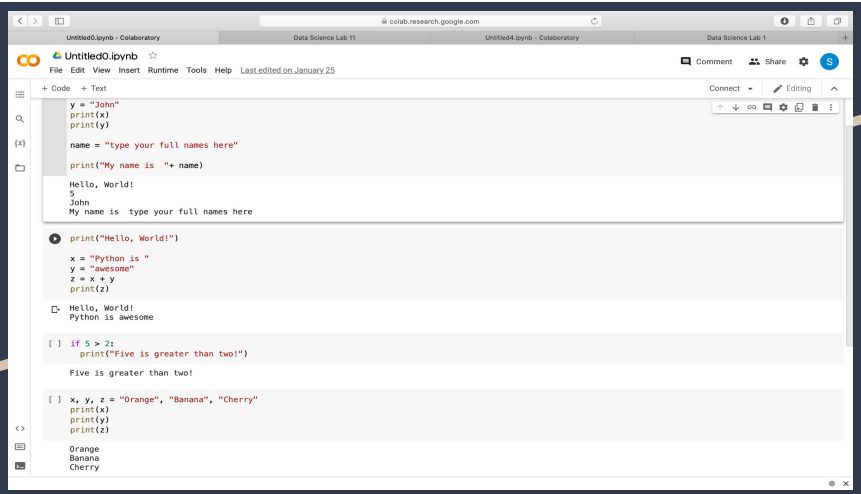
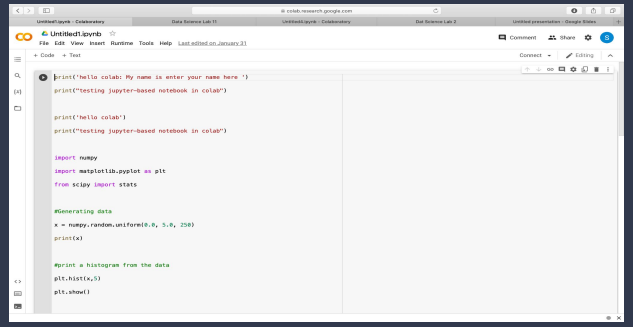
Data Science Presentation

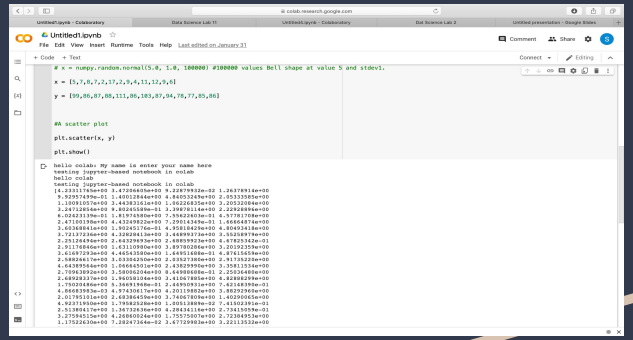
By:

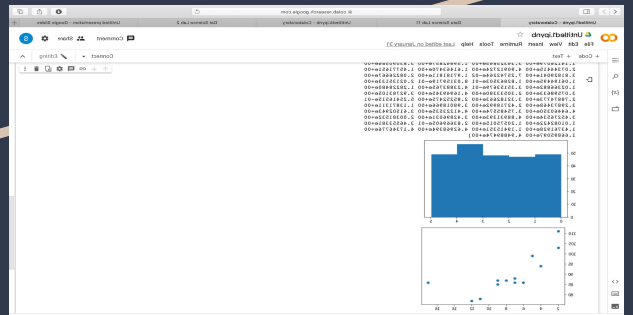




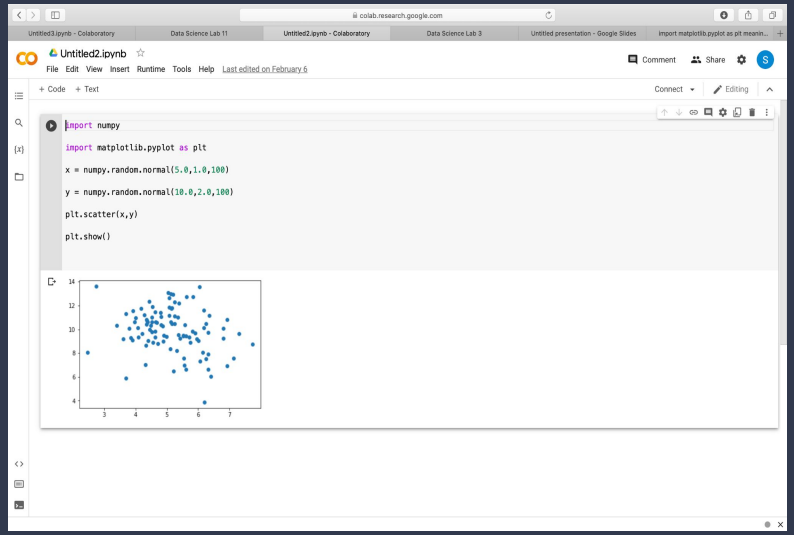
1. For lab 1, we used an introductory code to test put the environment of google colab and to make sure that we were comfortable using the platform.
2. I was able to print and copy three codes, one given, and two found on my own to ensure that I was comfortable using google colab as well as comfortable using and understanding python code.



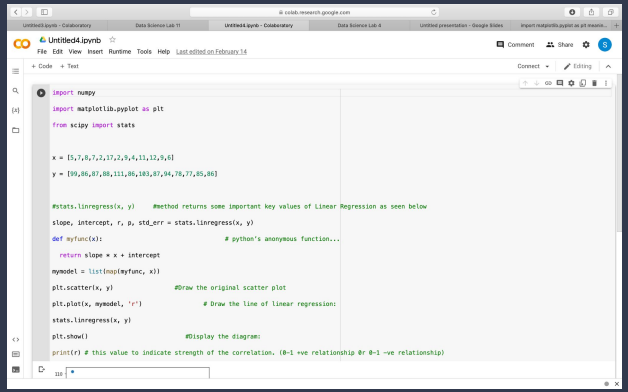


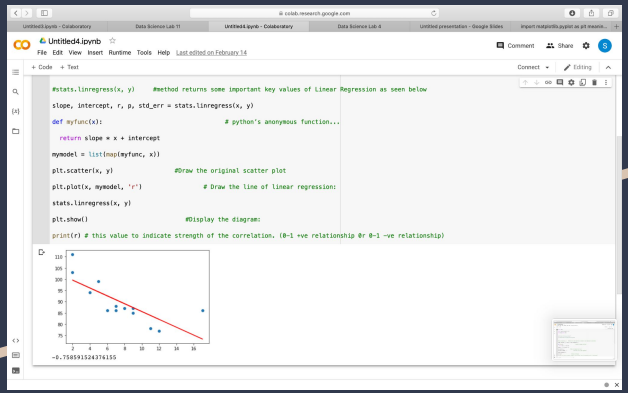


1. For data science lab 2, we were required to print hello colab and our name, print what we were testing, and print hello colab again.
2. The import numpy portion of the code tells Python to bring the NumPy library into your current environment.
3. import matplotlib. pyplot as plt gives an unfamiliar reader a hint that pyplot is a module, rather than a function which could be incorrectly assumed from the first form.
4. The code created random numbers from the numbers given, it created histogram from that data as well as a scatter plot.

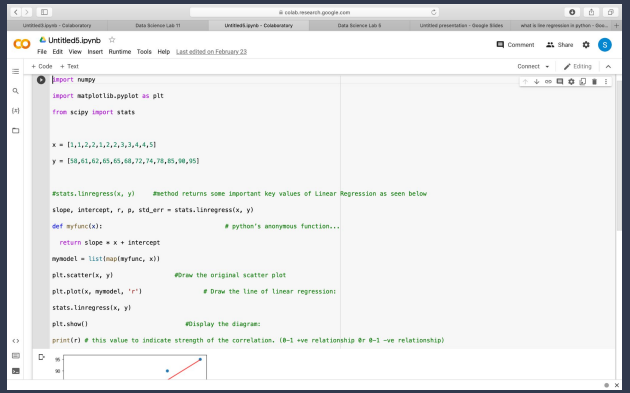


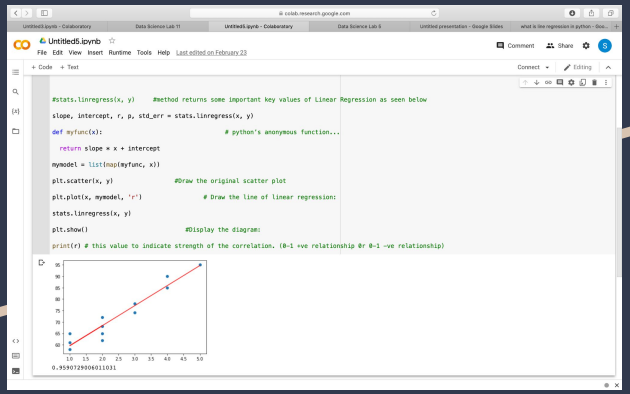
1. For lab 3, we were comparing the data generation to lab 2. The only differences were that the data was only generated through a scatter plot and.
2. The python libraries involved are numpy, matplotlib, and stats.



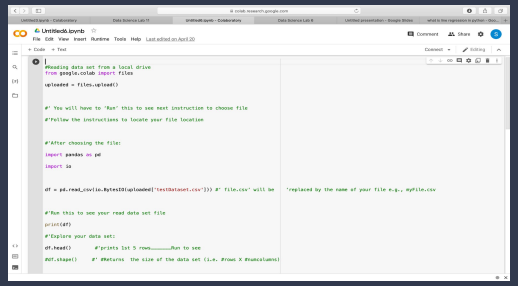


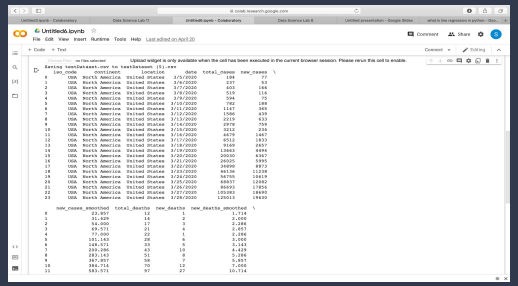
1. The code used for lab 4 is very similar to the code used for lab 2. The numbers are the same but that data was generated differently.
2. Stats.linregress(x, y) :Calculate a linear least-squares regression for two sets of measurements.
3. A line of regression is formed using the slope generated from the data.

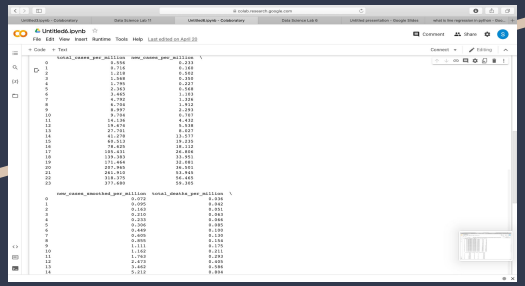




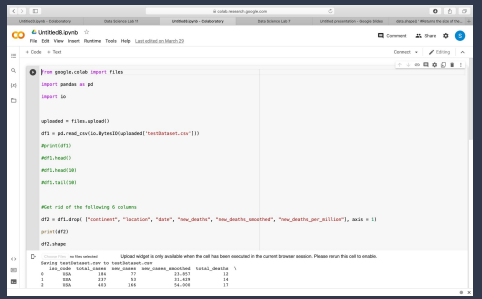
1. For lab 5 we used data (numbers) from a table of data to generate data in the form on a scatter graph.This is similar to what we did in lab 4.
2. In this lab we are strictly focusing on line regression which could help us predict the next set of numbers as well as compute strength.
3. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x).

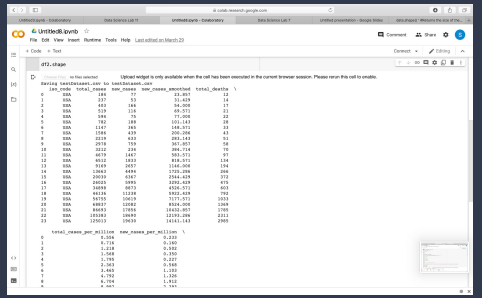


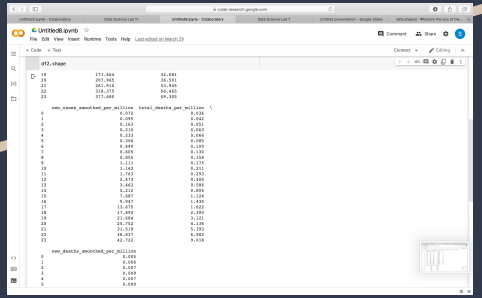




1. Lab 6 was an introductory to importing our own files and using the data from those files.
2. df = pd.read_csv(io.BytesIO(uploaded['file.csv'])) ‘ file.csv’ will be : This reads the file that you imported.
3. data.head() ‘prints 1st 5 rows……Run to see: This prints only the first 5 rows of the data
4. data.shape() ' #Returns the size of the data set (i.e. #rows X #numcolumns): This returns the shape of the array.

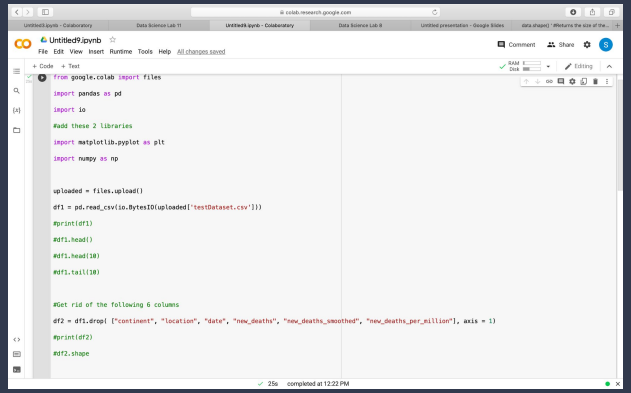


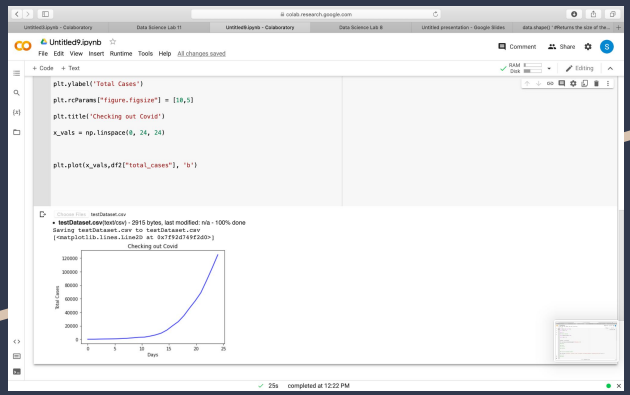




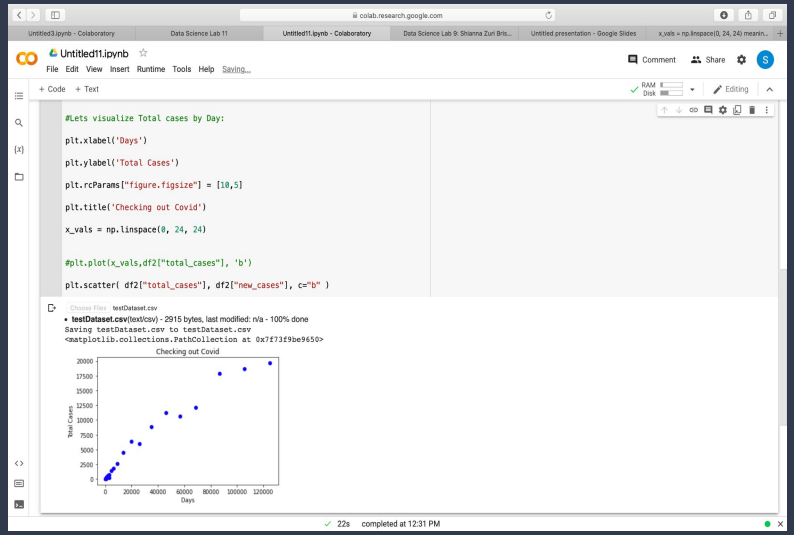
1. Lab 7 is similar to lab 6. Instead of reading the first five rows we are dropping 6 columns from the data that we want to read.
2. #Get rid of the following 6 columns

   df2 = df1.drop(["continent", "location", "date", "new_deaths", "new_deaths_smoothed", "new_deaths_per_million"], axis = 1)

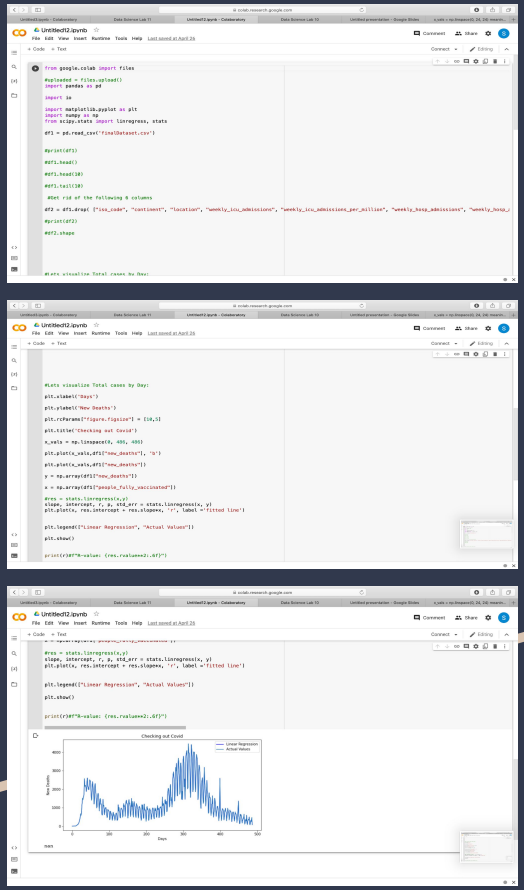




1. For lab 8, we also imported our own data sheet to read from. 6 columns from the data were also dropped. In addition to that we generated data using two of the columns which allowed us to see the amount of total cases per day.
2. plt.xlabel('Days'): Days would be our x values and plt.ylabel('Total Cases') would be our y values in the graph.
3. plt.rcParams["figure.figsize"] = [10,5]: This makes the figures width and height.
4. The NumPy linspace function (sometimes called np. linspace) is a tool in Python for creating numeric sequences.



1. For lab 9, we are doing the exact same thing we did for lab 8. The difference is that we are generating the data using a scatter plot.



1. For lab 10 we did new deaths by day instead of total cases by day. We did the exact same thing that we did in lab 8, we just used two different columns.
2. We imported the data ourselves, displayed the size of the data, and generated a graph.