To begin data analysis on the data set, our group plotted the sea level, which is what we wanted to predict, versus every other variable in our data set. We were able to have a visual representation of which machine learning algorithms we wanted to utilize from these plotted scatter graphs. After examination of the data plots, the conclusion was to use linear regression since there were some noticeable trends in the data that looked hopeful for lines could be drawn. The code for the plotting is identical to the linear regression plot and will be explained later in this paper.

When implementing the linear regression on our data, we first needed to read in our data files. Using the pandas library in Python, we read in each comma separated value sheets. Pandas is popular with big data implementation because it pulls data into one big data frame instead of pulling data individually by columns and adding them into an array. In addition, pandas is particularly useful for our data because we are only pulling from one variable comma separated value sheet at a time. This is beneficial as it allowed our group to easily remove columns, rows, and headers we did not want in our data. Once the data was in our code, we had a set of nested loops that would first loop through all the four seasons followed by a looping through all regions. The linear regression code is then embedded inside of the inner loop.

We first needed to remove the the header columns from our data so that we did not include this information in our model. In order to do this, we created a new list variable that took the old pandas data frames of sea level and the other variable we used against sea level. Using the pandas function call toList(), the new list variables contained every “i+2” column of the data frame as a python list rather than the pandas data frame we initially put the data into. The “i” columns contained the headers so we added two, making it “i+2” to start at the actual data excluding the years and season. **We then took these two new lists and split them from our outer loop of seasons. Starting at zero, we split each of those by four using list comprehension to get all four seasons. When the data is at index zero (or season one) we iterate every fourth which gives us every first season of every year. If the outer loop index is at one, then the data iterates through every second season and so on.**

The new arrays of every season were then reshaped using numpy. The new result will be a 1-D array of that length. The shape dimension of negative one is inferred from the length of the array and remaining dimensions. The reshaped data is used to fit our linear regression. Next, the reshaped data was split into training sets and test sets. The split we used was 80:20. This means 80% of the data goes to the training set and the other 20% goes to the test set. In order to do this we calculated the number of years multiplied by 0.8 (54 years x 0.8) which then gave us the 80%(approximately 43) slice mark. The colon notation refers to a slice. A number in front of the colon is the starting location while the number after is the ending location inclusive. Since we wanted the training set to be 80% of the data, we started at zero and iterated 43 rows. The notion looked like x[:43]. For the test set, we wanted the other 20% of the data thus we started at the 43 row and then finished at the end. The notation is x[43:]. These splices were done for both of our sea level and variable lists.

Now that our data is split into test and training sets, we were able to run the sklearn linear regression algorithm and implement the regression fit for our training sets. From this regression, we calculated the R squared value of the linear line we had just produced. By doing the R squared value, it allowed us too understand how well our model is predicted which will be discussed later in this paper. Sklearn’s linear regression has a function score() that returns the coefficient of determination R^2 of the prediction. In addition to the R-squared value, a graph was constructed for each region of every season for every variable. The graph presented us with an understanding of how the linear regression line looked in terms of increasing or decreasing.

Plotting the data with the regression line was fairly simple. The import we used to plot was matplotlib.pyplot. The matplot library allowed us to add titles, x-axis labels, and y-labels. From there, we created a scatter plot of sea level versus the other variable used. Finally, we plotted the predicted linear regression line.