CS 171 - Programming Assignment 3

Mark Boady - Drexel University

October 27, 2017

1 Overview

In this assignment, you will create a program to analyze data and make predictions about future results.

2 Best Fit Lines

If we have a set of data points, we can try to fit a line through the data points. If the line fits the data well, then we can make predictions about the data. Figure 1 shows a line that fits a set of data points.

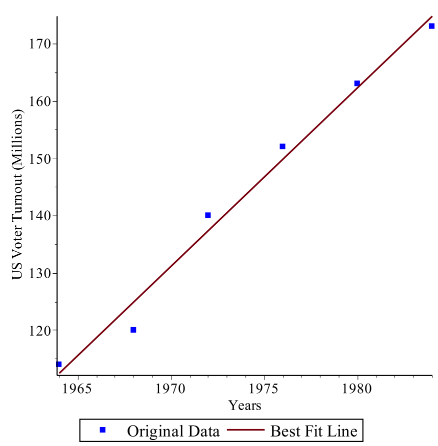

Figure 1: Example Best Fit Line

The data presented in Figure 1 is from US Voter turnout. The data that was used to make this plot is given below.

Year	Turnout (in Millions)
1964	114
1968	120
1972	140
1976	152
1980	163
1984	173

We can use this data to make predictions about future voter turnout. The accuracy of the predictions will depend on a number of factors. Using a line as an approximation only works if the data is roughly linear to begin with. The data may not be well represented by a line, it may not even be meaningful. If we analyze the relationship between number of pear trees in the state and the number of cats, out predictions will be worthless.

We will use Linear Regression to generate a Least Squares Line to fit the data. We want to create a line that minimized the error with all the data points.

The formula for a line is

$$y = mx + b \tag{1}$$

Our table of values gives us x and y. We need to determine m and b. We want to use all of our data points to generate a line.

We could just use the first and last points, but this would not take all our data into account. The slope of a line between to points in simple to compute.

$$\begin{aligned} m =& \frac{y_2 - y_1}{x_2 - x_1} && (2) \\ =& \frac{173 - 114}{1984 - 1964} && (3) \\ =& 2.95 && (4) \end{aligned}$$

Next, we can use the point (x_1, y_1) to compute b.

$$\begin{aligned} y_1 =& mx_1 + b && (5) \\ b =& -(mx_1 - y_1) && (6) \\ =& -(2.95 * 1964 - 114) && (7) \\ =& -5679.80 && (8) \end{aligned}$$

This gives us a simple approximation for the line.

$$y = 2.95x - 5679.80 \tag{9}$$

This approximation only took into account two data points. We can compare its predictions with the known data.

Year	Turnout (in Millions)	Estimated Value	Error
1964	114	114	0
1968	120	125.8	5.8
1972	140	137.6	2.4
1976	152	149.4	2.6
1980	163	161.2	1.8
1984	173	173.0	0

The first and last values will be prefect. The average error on the other 4 data points in 3.15 million voters.

A linear regression technique known as Least Squares can use all our data. Let

- x_a is the average of all x values
- x_i be the ith known x-axis value
- y_a is the average of all y values
- y_i be the ith known y-axis value
- n be the number of known values

The slope can be computed as

$$m = \frac{\sum_{i=0}^{n-1} ((x_i - x_a)(y_i - y_a))}{\sum_{i=0}^{n-1} ((x_i - x_a)^2)} \tag{10}$$

The base b is still based on the same formula but now uses the averages.

$$b = y_a - m x_a \tag{11}$$

Using the example data,
The two averages are

$$x_a = \frac{(1964 + 1968 + 1972 + 1976 + 1980 + 1984)}{6} \tag{12}$$

$$= 1974 \tag{13}$$

$$y_a = \frac{(114 + 120 + 140 + 152 + 163 + 173)}{6} \tag{14}$$

$$= 143.666667 \tag{15}$$

The slope is

$$\begin{aligned}
m_{\text{num}} =&(1964 - x_a)(114 - y_a) + (1968 - x_a)(120 - y_a) \\
&+ (1972 - x_a)(140 - y_a) + (1976 - x_a)(152 - y_a) \\
&+ (1980 - x_a)(163 - y_a) + (1984 - x_a)(173 - y_a) && (16) \\
=&872 && (17) \\
m_{\text{dem}} =&(1964 - x_a)^2 + (1968 - x_a)^2 + (1972 - x_a)^2 \\
&+ (1976 - x_a)^2 + (1980 - x_a)^2 + (1984 - x_a)^2 && (18) \\
=&280 && (19) \\
m =&\frac{m_{\text{num}}}{m_{\text{dem}}} = \frac{872}{280} = 3.114285714 && (20)
\end{aligned}$$

The intercept b is

$$\begin{aligned}
b =&y_a - mx_a && (21) \\
=&143.666667 - 3.114285714 * 1974 && (22) \\
=&- 6003.933332 && (23)
\end{aligned}$$

We know have a line that uses all our data.

$$y = 3.114285714 * x - 6003.933332 \tag{24}$$

The error with this new line is smaller.

Year	Turnout (in Millions)	Estimated Value	Error
1964	114	112.523810	1.476190
1968	120	124.980952	4.980952
1972	140	137.438095	2.561905
1976	152	149.895238	2.104762
1980	163	162.352381	0.647619
1984	173	174.809524	1.809524

The average error here is 2.26349 million voters.

These averages tell us about how the line compares to our real data points. It does not give us an indication of how our line will predict future results. A value called the regression standard error tells us about how the line will predict values. The smaller this value is, the more likely our predictions will be accurate.

If we treat the line as a mathematical function

$$\text{approx}(x) = 3.114285714 * x - 6003.933332 \tag{25}$$

The regression standard error (S) is the square root of the mean square error (MSE).

$$\text{MSE} = \frac{1}{n-2}\left(\sum_{i=0}^{n-1}(y_i - approx(x_i))^2\right) \quad (26)$$

$$S = \sqrt{\text{MSE}} \quad (27)$$

In our case, we have

$$\text{MSE} = 41.67618754 \quad (28)$$

$$S = 3.227854842 \quad (29)$$

When the data generally follows a straight line, more data will further improve our estimated line and decrease the regression standard error.

The approximation line can be used to approximate the number of voters in 1988.

$$3.114285714 * 1988 - 6003.933332 = 187.2666667 \quad (30)$$

The real value for 1988 was 181 million voters. Our line overestimates the value by about 6 million voters.

3 Programming Project

Develop a Python program `leastsquares.py`.

The program will ask the user for the name of a file. The file will contain comma separated values (CSV). The first row of the file will contains two strings separated by a comma. These will have the names of the x and y axis. The remaining rows will contain the values. The x axis values will always be integers. The y access values will be treated floats. They may be either floats or integers depending on the data, but they may always be treated as floats.

The contents of `voters.csv` is shown below as an example.

```
Year,Voters
1964,114
1968,120
1972,140
1976,152
1980,163
1984,173
```

Once the data has been read by the program, compute the Least Squares line, average error, and regression standard error as described in Section 2. All three values will be printed out for the user.

After the values have been computed, enter a loop asking the user for input. There will be three scenarios.

- The user gives an x-axis value, predict the corresponding y-axis value.
- The user enters "exit" and the program quits.
- The user enters an invalid input and the program asks for another input.

The program should also do error checking and exit gracefully in the event of a bad file name or bad input data.

Once the program works, you will use it to analyze the data sets. See Section 5 for more details.

4 Example Execution Trace

You are not required to exactly match the below layout, but your content and results must be the same.

The below execution traces do not test all possible inputs/output. Only a few examples are shown. You are expected to do additional testing on your own.

4.1 Example 1

```
Welcome to Linear Regression Generator
Enter File Name Containing Data: fake_name.csv
Error: File could not be opened.
```

4.2 Example 2

```
Welcome to Linear Regression Generator
Enter File Name Containing Data: bad_input.csv
Error: A value in the file could not be read.
```

4.3 Example 3

```
Welcome to Linear Regression Generator
Enter File Name Containing Data: voters.csv
The Linear Regression Line is y=3.11429*x-6003.93333.
Average Error for Known Values was +/-2.26349.
Regression Standard Error for Known Values was 3.22785.
System ready to make predictions.
To quit, type 'exit' as the year.
Enter Year: 1984
Prediction when Year = 1984 is Voters = 174.80952.
Enter Year: 1968
Prediction when Year = 1968 is Voters = 124.98095.
Enter Year: 1991
Prediction when Year = 1991 is Voters = 196.60952.
```

```
Enter Year: 2016
Prediction when Year = 2016 is Voters = 274.46667.
Enter Year: 2020
Prediction when Year = 2020 is Voters = 286.92381.
Enter Year: 1776
Prediction when Year = 1776 is Voters = -472.96190.
Enter Year: la la l
Input could not be understood. Please try again.
Enter Year: oranges
Input could not be understood. Please try again.
Enter Year: 2100
Prediction when Year = 2100 is Voters = 536.06667.
Enter Year: exit
```

4.4 Example 4

```
Welcome to Linear Regression Generator
Enter File Name Containing Data: temp.csv
The Linear Regression Line is y=0.01287*x+32.16684.
Average Error for Known Values was +/-0.24363.
Regression Standard Error for Known Values was 0.29321.
System ready to make predictions.
To quit, type 'exit' as the year.
Enter Year: 2018
Prediction when Year = 2018 is Temperature F = 58.14501.
Enter Year: 2017
Prediction when Year = 2017 is Temperature F = 58.13214.
Enter Year: 2016
Prediction when Year = 2016 is Temperature F = 58.11926.
Enter Year: exit
```

5 Analysis

Multiple data files are provided. For each data set, you are required to provide two things.

1. A prediction for an unknown value
2. Your opinion on how accurate the prediction is.

You should submit a single text file (`analysis.txt`) with your opinions for all the data sets. You only need a few sentences for each. You should experiment with your program to see how it predicts known values from the data to get feel for how accurate you think it is.

The following data is provided for you to analyze.

File Name	Contents
ages.csv	Tracks the age of a person born in 1989.
hurricanes.csv	Tracks the number of hurricanes recorded from 1851 to 2017.
temp.csv	Tracks the average global temperature from 1880 to 2016.
voters.csv	Tracks the number of US voters in millions from 1964 to 1984.
weights.csv	Tracks people's weight in relation to their height.

6 Grading

There are no strict guidelines for how to write your code or develop your user interface. You will be graded on the quality of your design and execution.

- Analysis (15 points)
  1. 1 point per data set for future prediction
  2. 2 points per data set for opinion on accuracy.
- Slope Calculations are correct (15 points)
- Average Error Calculation Correct (15 points)
- Regression Standard Error Correct (15 points)
- Future Predictions Calculated Correctly (15 points)
- Exits gracefully if file does not exist. (4 points)
- Exits gracefully if file has bad input. (4 points)
- Prints error and continues on bad input for predictions (4 points)
- Program exits on input command "exit" (3 points)
- User Interface easy to read/understand (4 points)
- File is well commented (3 points)
- Name in Comments (1 point)
- Section Number in Comments (1 point)
- File named correctly (1 point)

If you code has any runtime errors, a 50% deduction will be taken. Only portions of the code that execute without errors will be graded.

7 Resources

Additional Resources

`https://onlinecourses.science.psu.edu/stat501/node/250`
`https://climate.nasa.gov/vital-signs/global-temperature/`
`http://www.aoml.noaa.gov/hrd/tcfaq/E11.html`