Project 1: Collecting and Analyzing Data

Name : Gmon Kuzhiyanikkal

NU ID : 002724506

Overview

The data that I have been working with pertains to the charges paid by individuals for their medical insurance. These charges vary depending on factors such as age and lifestyle. In order to gain a better understanding of the information contained within the dataset, I have undergone a process of overall cleaning. This involved removing any irrelevant or redundant data, and ensuring that the remaining numerical data is as accurate and complete as possible. Additionally, in order to make the information more manageable and easy to understand, I have also reduced the size of the dataset. The final results of my analysis are presented below, along with screenshots of the output for each task. Overall, the cleaning and reduction of the dataset has allowed for a more in-depth and comprehensive understanding of the information contained within, and has made it easier to extract valuable insights and conclusions.

1.Collect a Data Set:

I have collected the medical cost personal Dataset for this project and it predicts the expected insurance charges need to be paid by each individual. It has 4 features and 3 are independent and one dependent variable. The features are mentioned below:

- **Age**: age of primary beneficiary
- **Bmi**: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height
- **Children**: Number of children covered by health insurance / Number of dependents
- **Charges**: Individual medical costs billed by health insurance

The independent variables are ***‘age’, ‘BMI’*** and ‘***children’*** and the dependent variable is ‘**charges**’. It has 100 samples and CSV files are provided along with the report and the name of the csv file is ‘insurance.csv’.

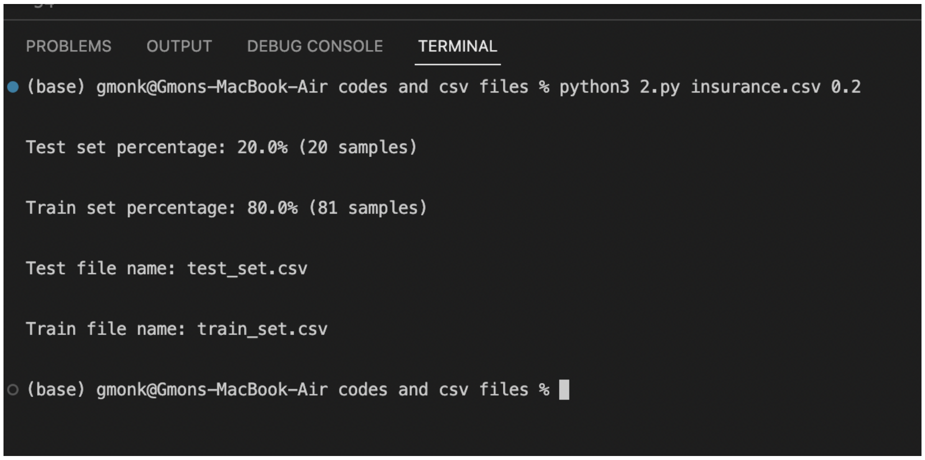
2.Organize your data:

I have used the python file ‘2.py’ to split the data into a training set and test set. I have used the random function for splitting the files into both the training and test set. To run the program, we have given the csv file name and test percentage in the command line.

Python3 2.py ‘any_csv_file_name’ ‘test_percetnage_in_decimal’

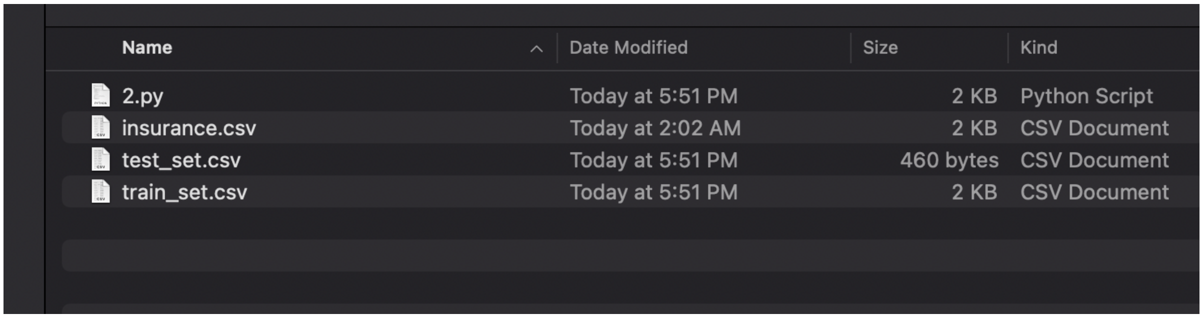
Eg: **python3 2.py insurance.csv 0.2**

Output of the program will be generated as per the below and I have shown the folder, where two separate files called ‘test.csv’ and ‘train.csv’ are created once the program is executed in the same folder where the python program 2.py is executed.



```
(base) gmonk@Gmons-MacBook-Air codes and csv files % python3 2.py insurance.csv 0.2

Test set percentage: 20.0% (20 samples)

Train set percentage: 80.0% (81 samples)

Test file name: test_set.csv

Train file name: train_set.csv

(base) gmonk@Gmons-MacBook-Air codes and csv files %
```

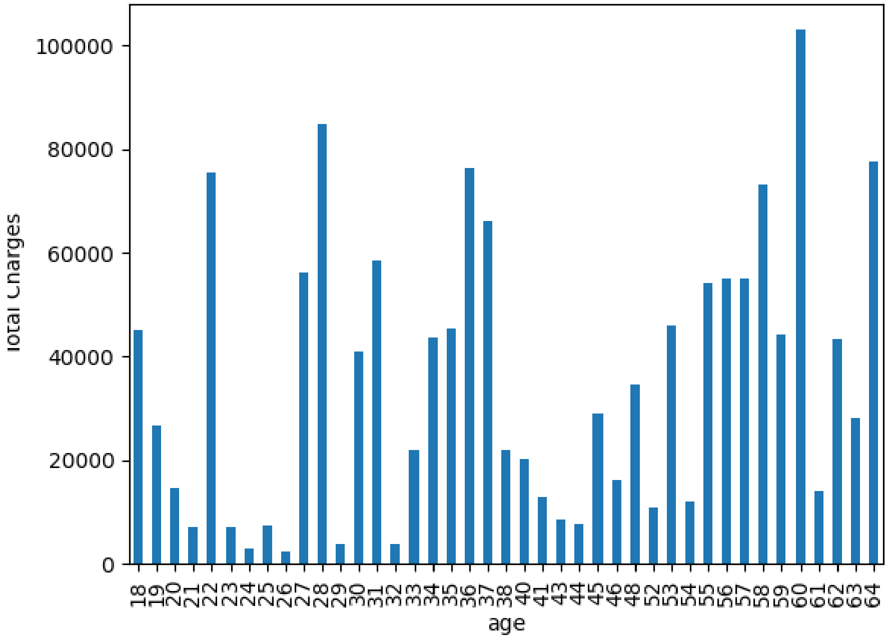
Output of the files in the folder, will now contain ‘test_set.csv’ and ‘train_set.csv’



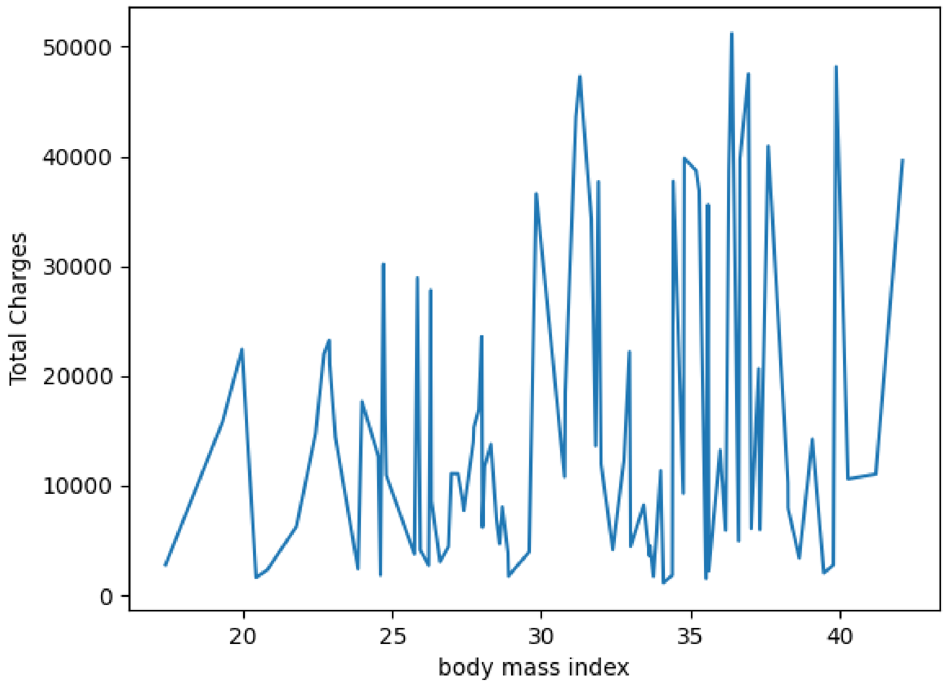
Name	Date Modified	Size	Kind
2.py	Today at 5:51 PM	2 KB	Python Script
insurance.csv	Today at 2:02 AM	2 KB	CSV Document
test_set.csv	Today at 5:51 PM	460 bytes	CSV Document
train_set.csv	Today at 5:51 PM	2 KB	CSV Document

3.Plot your data:

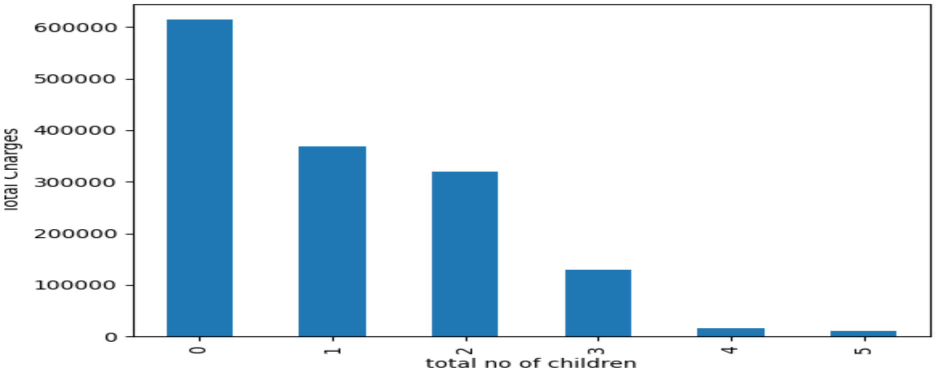
For the ages, I have plotted the Bar graph and name of the python file 3.py and output of the graph of **ages vs insurance charges** is given below:



I could see that, highest insurance is paid by the people at the age of 60. For the body mass index(BMI). People whose bmi is between 35 and 40 has paid the highest medical insurances and I i have used the normal plot and the output of **BMI vs insurance charges** is given below:



I have used a bar graph to plot the **<u>no of children vs insurance charges</u>**. People with zero children paid the highest insurances and the bar graphis given below:

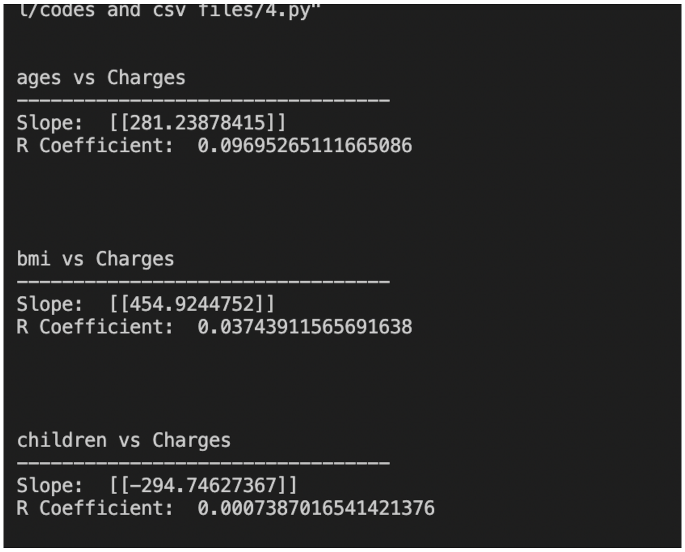


4. Execute a linear regression for each independent variable:

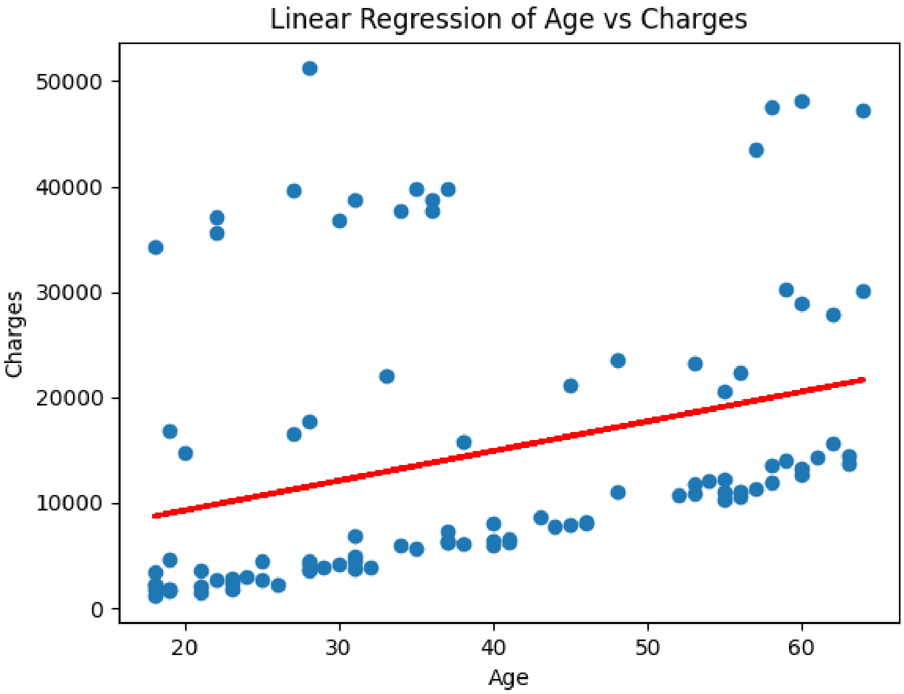
R coefficient and slope of the each independent variable with the dependent variable is given below and their conclusion is also given in the third column:

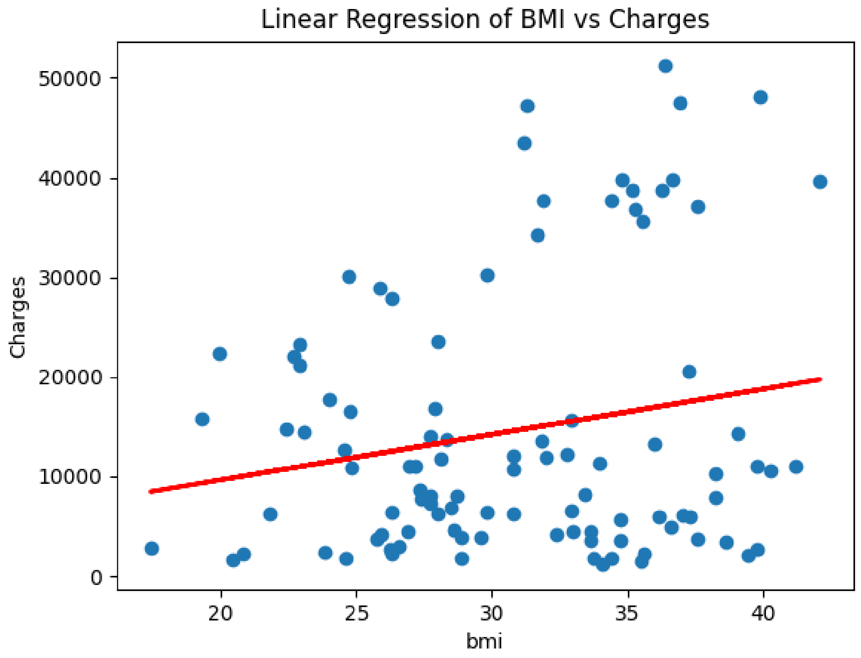
Variables	Slopes vs R-coeff	Conclusion
Age vs charges	Slope: 281.23878415 R Coefficient:0.096	1) As the age increases the insurance charges also increase because of the positive slope. 2) R-coefficient is near to zero and the line isn't a perfect fit. If it is near the 1, it would be a perfect fit.
BMI vs charges	Slope: 454.9244752 R coefficient : 0.03743	1) As the BMI increases the insurance charges also increase because of the positive slope. 2) R-coefficient is near to zero and the line isn't a perfect fit. If it is near the 1, it would be a perfect fit.
Children vs charges	Slope: -294.7462 R coefficient : 0.00073	1) As the number of children increases the insurance charges decrease because of the negative slope. 2) R-coefficient is near to zero and the line isn't a perfect fit. If it is near the 1, it would be a perfect fit.

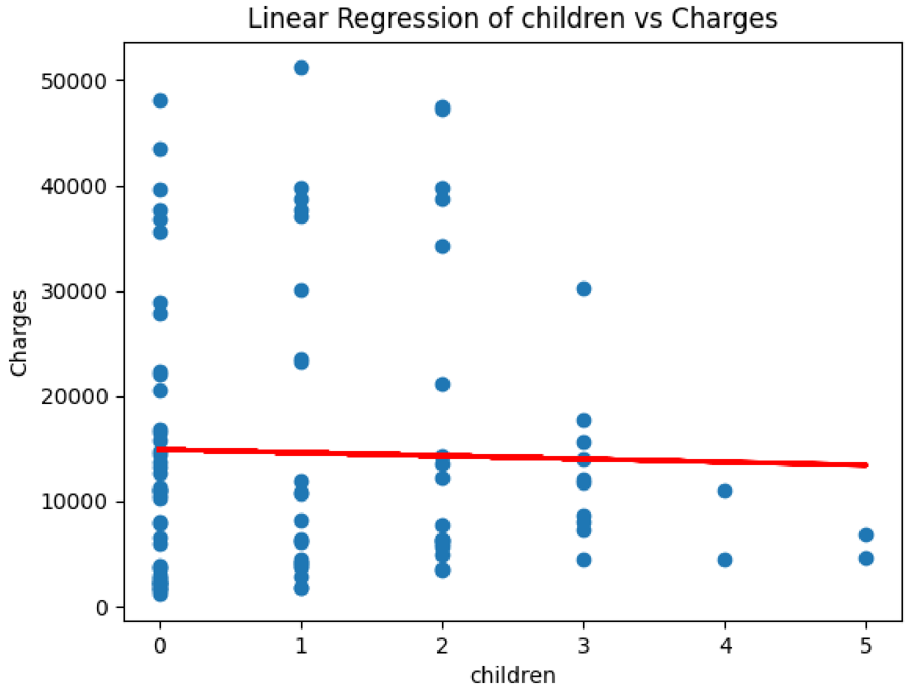
Output of the python file is given below and the output diagram of the each linear regression is also given below:



```
l/codes and csv files/4.py"

ages vs Charges
----------------------------------
Slope:   [[281.23878415]]
R Coefficient:   0.09695265111665086



bmi vs Charges
----------------------------------
Slope:   [[454.9244752]]
R Coefficient:   0.03743911565691638



children vs Charges
----------------------------------
Slope:   [[-294.74627367]]
R Coefficient:   0.0007387016541421376
```




Linear Regression of BMI vs Charges
Charges
bmi
0
10000
20000
30000
40000
50000
20
25
30
35
40


Linear Regression of children vs Charges
Charges
children
0
10000
20000
30000
40000
50000
0
1
2
3
4
5

5) Execute a multiple linear regression:

The individual coefficient of each independent variable with the dependent variable is given below and it is run by the program ‘5.py’.

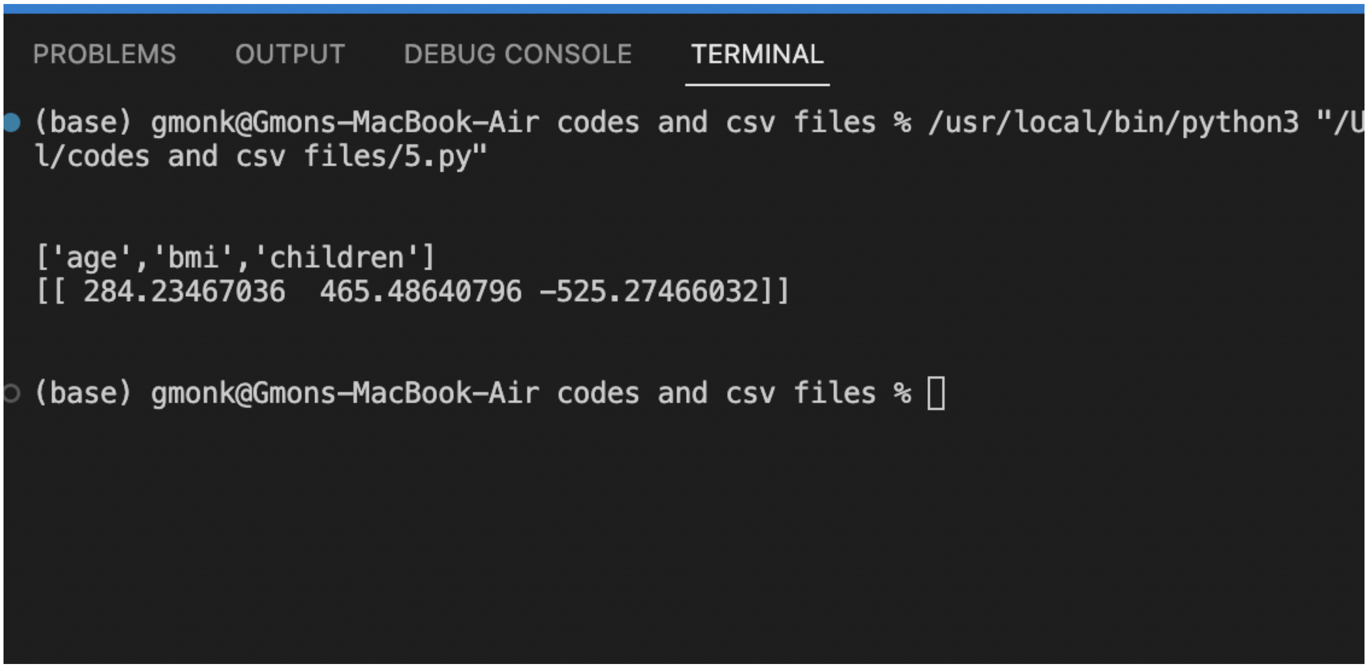
Age: 284.23   (positive correlations)

Bmi: 465.486 (positive correlations)

Children: -525.27 (negative correlations)

From the coefficient, we can conclude that bmi is strongly related to the insurance charges followed by age and children respectively. Children have a negative correlations with the dependent variable and hence the order of the dependency of the independent variable to dependent variable is:

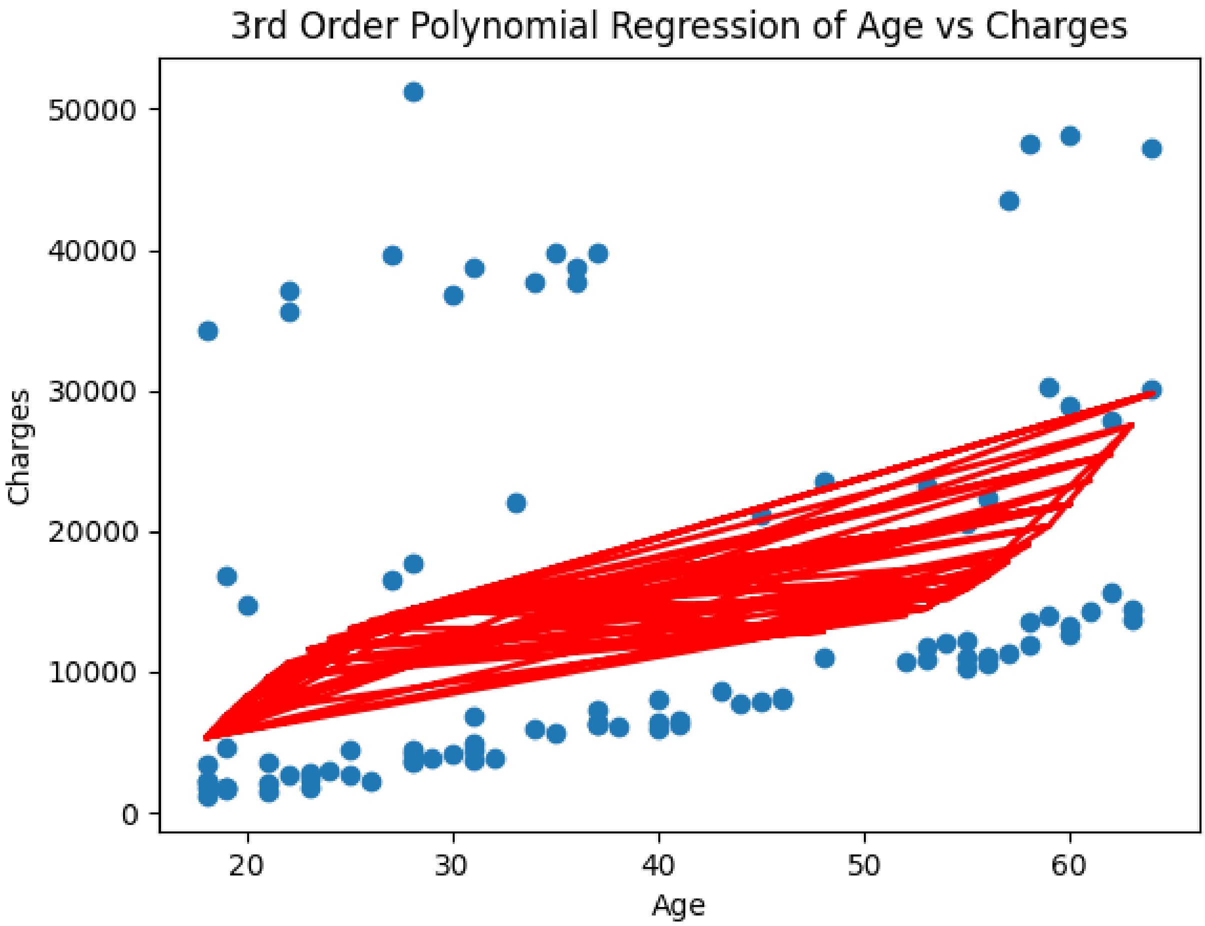
BMI > Age > Children



```
(base) gmonk@Gmons-MacBook-Air codes and csv files % /usr/local/bin/python3 "/U
l/codes and csv files/5.py"


['age','bmi','children']
[[ 284.23467036  465.48640796 -525.27466032]]


(base) gmonk@Gmons-MacBook-Air codes and csv files % 
```

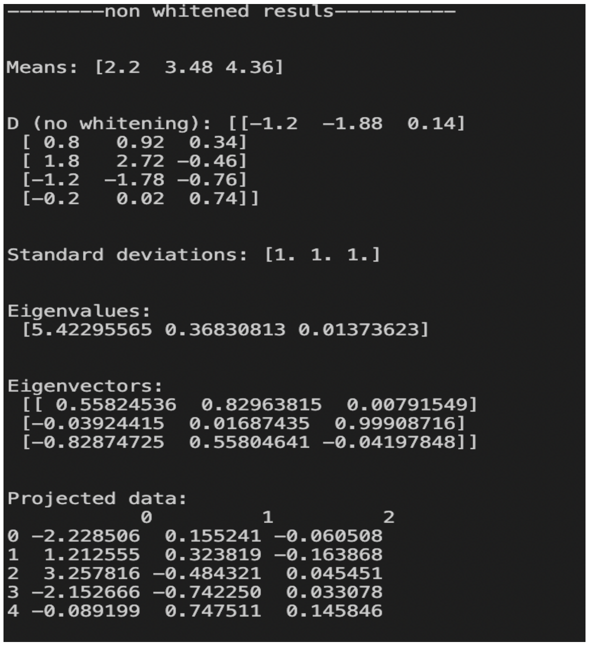
6. Execute a linear regression with a polynomial model:

The Python file for this is ‘6.py’ and I have taken ages vs charges for comparison and x: age_squared, age_cubed and age which represent the linear, squared, and cubic versions of the independent variable respectively. Then I have used the data frame as an input to the Linear Regression model.This has created a scatter plot of the data points with the 3rd order polynomial regression line (in red) superimposed on it. The image is given below:

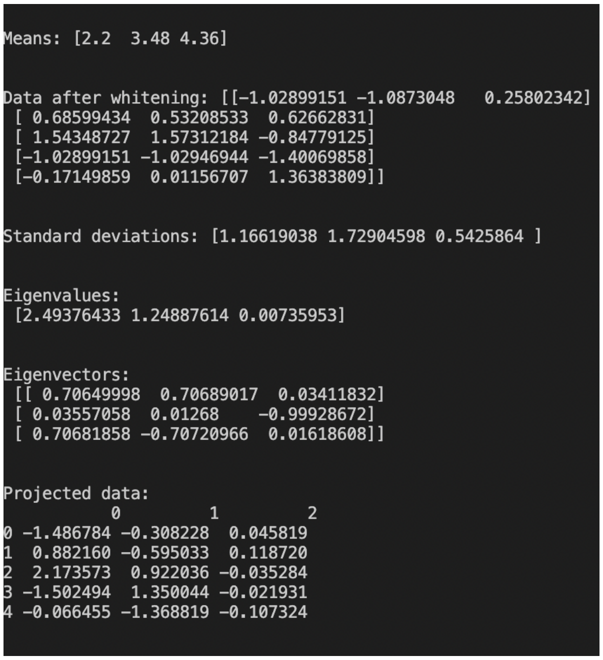


7. Implement Principal Components Analysis

The python file for running the program is ‘7.py’ and both the result are printed below:



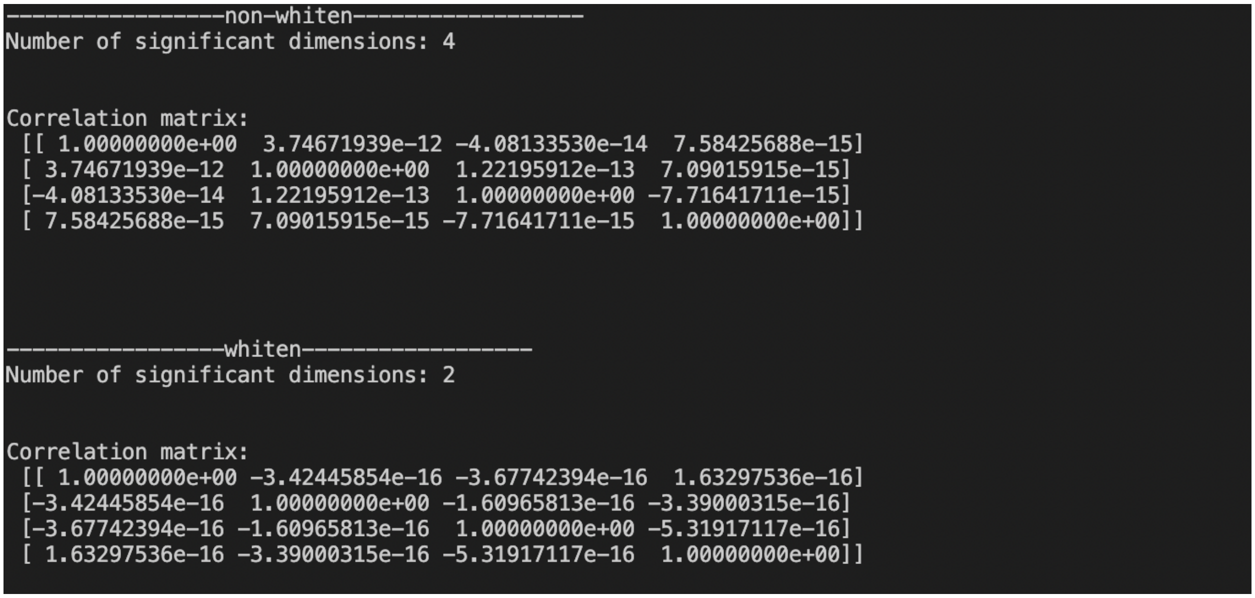
```
---------non whitened resuls-----------


Means: [2.2  3.48 4.36]


D (no whitening): [[-1.2  -1.88  0.14]
 [ 0.8   0.92  0.34]
 [ 1.8   2.72 -0.46]
 [-1.2  -1.78 -0.76]
 [-0.2   0.02  0.74]]


Standard deviations: [1. 1. 1.]


Eigenvalues:
 [5.42295565 0.36830813 0.01373623]


Eigenvectors:
 [[ 0.55824536  0.82963815  0.00791549]
 [-0.03924415  0.01687435  0.99908716]
 [-0.82874725  0.55804641 -0.04197848]]


Projected data:
          0         1         2
0 -2.228506  0.155241 -0.060508
1  1.212555  0.323819 -0.163868
2  3.257816 -0.484321  0.045451
3 -2.152666 -0.742250  0.033078
4 -0.089199  0.747511  0.145846
```



```
Means: [2.2  3.48 4.36]


Data after whitening: [[-1.02899151 -1.0873048   0.25802342]
 [ 0.68599434  0.53208533  0.62662831]
 [ 1.54348727  1.57312184 -0.84779125]
 [-1.02899151 -1.02946944 -1.40069858]
 [-0.17149859  0.01156707  1.36383809]]


Standard deviations: [1.16619038 1.72904598 0.5425864 ]


Eigenvalues:
 [2.49376433 1.24887614 0.00735953]


Eigenvectors:
 [[ 0.70649998  0.70689017  0.03411832]
 [ 0.03557058  0.01268    -0.99928672]
 [ 0.70681858 -0.70720966  0.01618608]]


Projected data:
          0         1         2
0 -1.486784 -0.308228  0.045819
1  0.882160 -0.595033  0.118720
2  2.173573  0.922036 -0.035284
3 -1.502494  1.350044 -0.021931
4 -0.066455 -1.368819 -0.107324
```

8.Apply PCA:

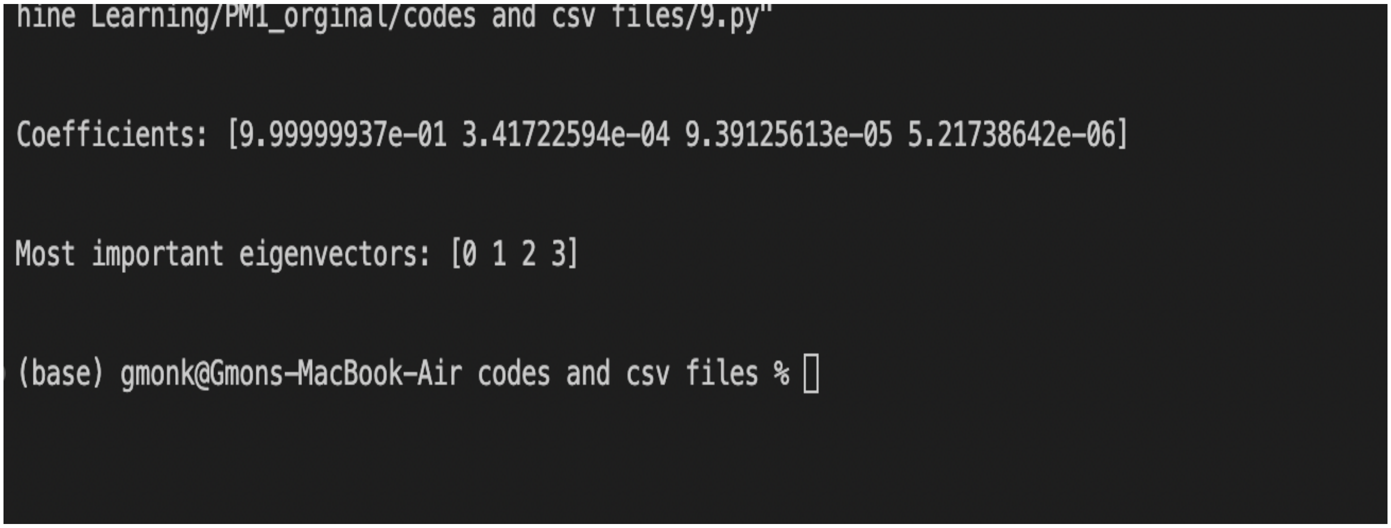
When the non-whiten data is run on the PCA, the number of dimensions I have received is 4 and when the whitened data is run, I have got the number of dimensions is 2. In general, using fewer dimensions is better, it is better to use the data whitened in the ‘insurance.csv’ data set. By retaining only the most important dimensions, you are retaining the most important information in the data. The python program for this is ‘8.py’ and co-variance and number of dimension output is given in the below figure:



```
-----------------non-whiten------------------
Number of significant dimensions: 4


Correlation matrix:
 [[ 1.00000000e+00  3.74671939e-12 -4.08133530e-14  7.58425688e-15]
 [ 3.74671939e-12  1.00000000e+00  1.22195912e-13  7.09015915e-15]
 [-4.08133530e-14  1.22195912e-13  1.00000000e+00 -7.71641711e-15]
 [ 7.58425688e-15  7.09015915e-15 -7.71641711e-15  1.00000000e+00]]




-----------------whiten------------------
Number of significant dimensions: 2


Correlation matrix:
 [[ 1.00000000e+00 -3.42445854e-16 -3.67742394e-16  1.63297536e-16]
 [-3.42445854e-16  1.00000000e+00 -1.60965813e-16 -3.39000315e-16]
 [-3.67742394e-16 -1.60965813e-16  1.00000000e+00 -5.31917117e-16]
 [ 1.63297536e-16 -3.39000315e-16 -5.31917117e-16  1.00000000e+00]]
```

9.Implement multiple linear regression on the projected data

Python program to run is ‘9.py’

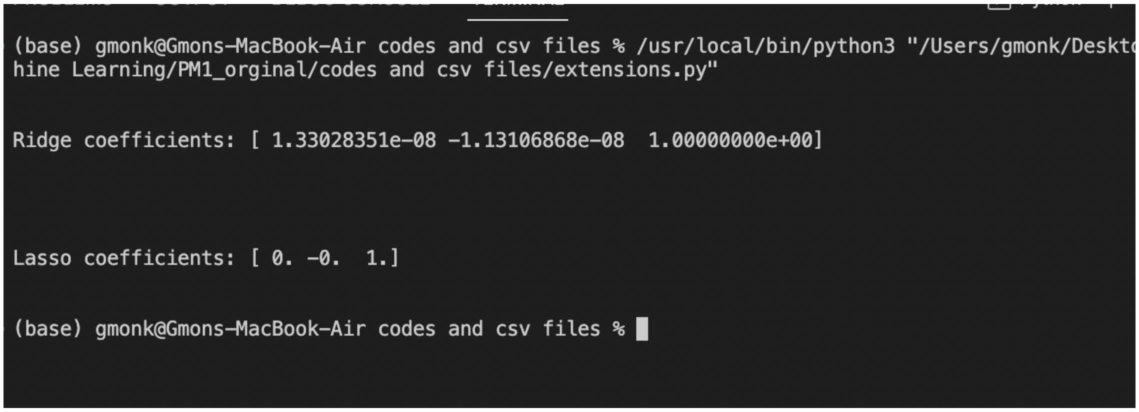


```
hine Learning/PM1_orginal/codes and csv files/9.py"


Coefficients: [9.99999937e-01 3.41722594e-04 9.39125613e-05 5.21738642e-06]


Most important eigenvectors: [0 1 2 3]


(base) gmonk@Gmons-MacBook-Air codes and csv files %
```

It seems that the most important eigenvectors are 1, 2 and 3, as they have the largest absolute value among the coefficients. The eigenvector 0 has a very small coefficient, indicating that it is less important in explaining the variance in the dependent variable 'charges'.

Description and appropriate results for any extensions:

I am trying to use ridge or lasso regression on the ‘insurance.csv’ dataset and discuss the differences in the results. The program to get the output for the extensions is “extensions.py” and the output of ages(independent variable) run on the both the coefficient of the lasso and ridge is given below:



```
(base) gmonk@Gmons-MacBook-Air codes and csv files % /usr/local/bin/python3 "/Users/gmonk/Deskto
hine Learning/PM1_orginal/codes and csv files/extensions.py"


Ridge coefficients: [ 1.33028351e-08 -1.13106868e-08  1.00000000e+00]



Lasso coefficients: [ 0. -0.  1.]

(base) gmonk@Gmons-MacBook-Air codes and csv files % 
```

Although none of the coefficients in a Ridge regression are zero, their magnitudes are reduced. The size of the coefficients in a Lasso regression is reduced, and some of the coefficients are zero. As a result, in the lasso model, certain characteristics are entirely removed.

The size of the coefficients in a Ridge regression indicates how strongly the independent and dependent variables are correlated. However, in Lasso regression, the size of the coefficients not only reflects the proportional weight of each variable in the model but also the strength of the association between the independent and dependent variables.

Reflection of what you learned:

1) I was able to get used to IDE for machine learning and I am also able to collect csv data from various online repositories like kaggle.
2) Implemented a program from scratch for splitting the data by passing as command line(which can be used in the future projects too).
3) I am able to plot the graph using the python library matplotlib and I was able to interrupt different patterns from the graph.
4) Successfully able to execute the linear and multi linear regression. I am able to interpret the correlation between different independent and dependent variables from slope and r-coefficient.
5) I was able to apply principal component algorithm on the data set and I am able to eliminate the features which are less required.
6) By the part of extensions of the project, I am able to learn about lasso and ridge regression and try to find the coefficient, I was able to compare both the result

Bibliography:

1) Website kaggle
https://www.kaggle.com/datasets/mirichoi0218/insurance

2) https://www.geeksforgeeks.org/

3) Muller and Guido(Textbook)