Regression Analysis Applied on Different Datasets with Python and R

Laman Aliyeva

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
September 2023
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Prof. Dr. Nazım Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Asst. Prof. Dr. Mehmet Ali Tut
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Arif Akkeleş
2. Asst. Prof. Dr. Hüseyin Lort
3. Asst. Prof. Dr. Mehmet Ali Tut

ABSTRACT

Data Science is a broad field which includes statistics, data analysis, and machine learning methods to analyze data and extract important information and understanding from the data. In recent years, many data scientists have been arguing about whether Python or R programming languages are better.

This thesis I chose both of them and it explains the differences between two commonly used programming languages, Python and R, for coding algorithms to predict linear regression. My goal is to find coefficients by applying linear and multiple equations with three datasets. Using Python and R languages shows me which variables (in datasets) the regression line is better than the others and I have achieved my goal by showing special codes.

Keywords: Regression Analysis, Linear Regression, Linear Regression Line, Python, R Language

ÖZ

Veri Bilimi, verileri analiz etmek ve verilerden önemli bilgileri ve anlayışı çıkarmak için istatistik, veri analizi ve makine öğrenimi yöntemlerini içeren geniş bir alandır. Son yıllarda, birçok veri bilimcisi Python programlama dillerinin mi yoksa R programlama dillerinin mi daha iyi olduğu konusunda tartışmaktadır.

Bu tezde ikisini de seçtim ve doğrusal regresyonu tahmin eden kodlama algoritmaları için yaygın olarak kullanılan iki programlama dili olan Python ve R arasındaki farkları açıklıyor. Amacım, üç veri seti ile doğrusal ve çoklu denklemler uygulayarak katsayıları bulmak. Python ve R dillerini kullanmak bana hangi değişkenlerin (veri setlerinde) regresyon çizgisinin diğerlerinden daha iyi olduğunu gösteriyor ve özel kodlar göstererek amacıma ulaştım.

Anahtar Kelimeler:Regresyon Analizi, Lineer Regresyon, Lineer Regresyon Doğrusu, Python Dili, R Dili

In the Name of Allah, the Most Gracious, the Most Merciful

"Glory be to Thee! we have no knowledge but that which Thou hast taught us: surely Thou art the Knowing, the Wise" / Qur'án (2:32)/

.... Dedicated to my Family and loved ones.

ACKNOWLEDGMENTS

First and foremost, as I go closer to realising my dream, I want to thank you from the bottom of my heart. I owe God a sincere debt of gratitude for supporting me and directing me towards the successful completion of my thesis. I owe a great deal of gratitude to everyone who has helped me along the way.

I will always be grateful to Prof. Dr. Mehmet Ali Tut, my supervisor, for the priceless lessons he taught me and for his unfailing support. His expertise and patience were crucial to the undeniable success of my work. I have a special place in my heart for the faculty at Eastern Mediterranean University's mathematics department, and I will never forget the impact they had on my development.

I want to express my sincere gratitude to my EMU classmates, especially Walaa Yasin for her insightful observations and LaTex pointers. Her friendship and cooperation have been incredibly valuable.

Last but not least, I want to sincerely thank and appreciate my family, who have been my steadfast supports since I was a child. My accomplishments have been motivated by their ongoing support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ε	Random Error (Residual) Term
exp	the Natural Logarithm
λ	Regularization Parameter
$\sum$	Summation
∂	Partial Derivative
$\log(odds)$	The Natural Logarithm of the Odds
β_0	y- Intercept
β_1	Slope
L1	Lasso
MSE	Mean Squared Error
OLS	Ordinary Least Squares
S	Sum of Squared Errors
SSR	Sum of Squared Residuals

Chapter 1

INTRODUCTION

1.1 Regression Analysis

The word "regression" has been around since the 18th century. However, regression as it relates to correlation was first utilized in 1866 by the 19th century English anthropologist and statistician Francis Galton while examining the heredity of genes involved for height. He published a study titled "Regression to Mediocrity in Hereditary Stature." Since then, the phrase has been well-known as a statistical notion that is commonly utilized in assessing data qualities with different characteristics. Over time, various regression models were developed to approach different types of data and relationships. Which assumes a linear relationship between the variables, linear regression earned stature and became widely used. Polynomial regression, logistic regression, and other decent forms of regression were also developed to analyze non-linear and categorical relationships. Regression analysis became more reachable and efficient with the advent of computers and statistical software. Researchers could analyze large datasets and fit complex regression models with ease, leading to further improvements and applications in various fields. Today, regression analysis is widely used in fields such as economics, social sciences, engineering, finance, and machine learning.

1.1.1 Significance of Regression Analysis

The answer of what is regression is that to build a mathematical model describing the effect of a set of input variables $\{x_1, x_2, \cdots, x_r\}$ on another variable y. There is a relationship between y and $\mathbf{x} = \{x_1, x_2, \cdots, x_r\}$. Regression analysis is used when you want to predict dependent variable from several independent variables.

In general, we can say that regression analysis is used to estimate the relationships

between a dependent variable and one or more independent variables. Since these techniques are practicable in almost every field of study and it is the most used of all data analysis methods. To make the thesis relatively self-contained I have included basic materials from statistics, numerical analysis, and partial differential equations. Linear regression [1] is the most prevalent type of regression analysis since it allows us to discover the line that best fits the data according to a certain mathematical criterion.

- Regression analysis provides insight into the strength of correlations between variables. Regression analysis, which use statistical measures like as R-squared, may tell you how much of the overall variability in the data is explained by your model.
- Regression analysis is more flexible and has wide workability.
- Learning Regression Analysis will improve our general comprehension of statistical inference.
- Learning regression analysis helped me become a better coder (Python AND R), a better statistician, and an overall better model builder.

Overall, regression analysis is a well-rounded and valuable tool for understanding relationships.

1.1.2 Dependent and Independent Variable

In regression analysis, the dependent variable and independent variables play major roles for understanding the relationship between variables. It is important to recognize between the dependent variable and independent variables.

The relationship between the dependent variable and independent variables is known through a mathematical equation or regression model. The goal is to estimate the effect of each independent variable on the dependent variable while accounting for other variables in the calculation.

The dependent variable is known as the response variable or outcome variable and the dependent variable is the variable that the researcher wants to predict, explain, or understand and it is denoted as Y. In the context of regression analysis, the dependent variable is typically a continuous or numeric variable, such as sales income, temperature, or test scores.

The independent variable is known as the predictor variables, explanatory variables. Those are the variables that have an impact on the dependent variable. In regression analysis, independent variables can be continuous or categorical.

It's important to note that the terms "dependent variable" and "independent variables" are specific to regression analysis, and their usage may change in other statistical contexts.

1.1.3 Types of Regression Analysis

Before using the regression technique, there are some notions we have to know them. These methods are different based on the types of dependent and independent variables being studied.

- **Linear regression** [2] is the simplest form of regression that we assume that the dependent variable is directly related to one or more independent variables. We can say it is when you want to predict values of one variable, given values of another variable. The aim is to find the best-fitting straight line that stand for the relationship. There are two models which are simple linear regression and multiple linear regression.When there is only one independent variable, then it is called Simple linear regression. The formula is:

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{1.1}$$

  This formula explains the meaning of some terms used in mathematics to understand relationships between variables where y represents dependent variable, x the independent variable, β_0 the intercept (constant term), β_1 the

coefficient for the independent variable and ε the error term.

On the other hand, when there are more than one independent variable, then it is called **multiple linear regression**. The formula is:

$$y = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots + \beta_r x_r + \varepsilon \tag{1.2}$$

The same definition for this formula where y represents dependent variable, x the independent variable, β_0 the intercept (constant term), $\beta_1 x + \beta_2 x^2 + \cdots + \beta_r x^r$ the coefficients for the polynomial terms and ε the error term.

- Polynomial regression is different from linear regression because the polynomial regression is a technique for fitting a nonlinear equation by taking polynomial functions of independent variable. It is a variant of the multiple linear regression model that includes fitting a polynomial equation to the data, which can take curved or nonlinear types where the dependent and independent variables have a curvilinear relationship, and the polynomial equation is:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_r x^r + \varepsilon$$

where y represents the dependent variable, x the independent variable,β_0 the intercept (constant term), $\beta_1 x + \beta_2 x^2 + \cdots + \beta_r x^r$ the coefficients for the polynomial terms and ε is error term.

Polynomial regression can be categorized to different powers to create polynomial terms. It's important to choose an appropriate degree for the polynomial. For instance, if degree is 2 (quadratic) would include terms like x, x^2, while degree is 3 (cubic) would include terms like x, x^2, x^3, and so on.

Polynomial regression is useful for various fields, including physics, economics, social sciences, and engineering.

- Logistic Regression is that when the dependent variable is discrete, then the technique is applicable. It is mainly created for dependent variables that are binary (having two categories) or ordinal (having ordered categories). Unlike linear regression, it is used for continuous dependent variables. It makes a way to model the relationship between the independent variables and the probability of an event happening and estimates the probability of an event.

  We have the formula for the Logarithm of the odds ratio:

  $$log(odds) = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots + \beta_r x_r$$

  where $x_1, x_2, \cdots, x_r$ represents the independent variables, β_0 the intercept (constant term) and $\beta_1, \beta_2, \cdots, \beta_r$ the coefficients for the independent variables.

  It uses the logistic function which is known as the sigmoid function that changes a linear combination of the independent variables into a probability value between 0 and 1. The general form of logistic regression is:

  $$P(y = 1|x) = 1/(1 + \exp(-z))$$

  where $P(y = 1|x)$ represents the probability of the binary outcome $(y = 1)$ given the values of the independent variables (x) and z represents the linear combination of the independent variables and their respective coefficients.

  There are 3 ways to use logistic regression:

  1. Binary logistic regression - If there are two possible outcomes, like whether someone might have COVID-19 or not.
  2. Multinomial logistic regression – If there are many outcomes, for example, if we extend our initial example to predict if someone has the flu, an allergy, a cold, or COVID-19.
  3. Ordinal logistic regression - If the outcome is ordered, like in our original

example, we can determine the severity of a COVID-19 infection, sorting it into mild, moderate, and severe cases.

Logistic regression is used in many different areas such as medicine, social sciences, marketing and machine learning.

- Ridge Regression (shrinkage regression) is the way to analyze high correlation where the independent variables are strongly correlated with each other in a regression model. If there are not many observations compared to predictor variables, we should use ridge regression.

  Ridge regression adds a squared term to the sum of squared coefficients that minimize the sum of squared errors while also minimizing the sum of squared coefficients. It means that there is the traditional least squares estimation is modified by adding a penalty which controlled by a regulation parameter λ (lambda), reduces the coefficient estimates towards zero that term to the sum of squared coefficients. A larger λ value results in greater decrease and more regularization. The ridge regression equation is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_r x_r + \lambda \sum \beta_i^2 + \varepsilon$$

  where y represents dependent variable, $x_1, x_2, \cdots, x_r$ the independent variables, β_0 the intercept (constant term) and $\beta_1, \beta_2, \cdots, \beta_r$ the coefficients for the independent variables, λ is penalty parameter that controls the amount of regularization and ε is random error term. $\lambda \sum \beta_i^2$ is the ridge term that is added to the traditional least squares equation and penalizes the magnitude of the coefficients, encouraging them to be smaller.

  Ridge regression [3] is commonly applied in fields such as finance, economics, and machine learning.

- Lasso Regression stands for Least Absolute Shrinkage and Selection Operator. Ridge regression and Lasso regression are the same definitions but Lasso regression adds the absolute value of the coefficients. It is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean It makes use of **L1 regularization** technique in the objective function. Additionally, lasso regression is a method that uses absolute values of coefficients to add a penalty in L1 regularization. Rare models with few coefficients may arise from this form of regularization. Some parts might become nothing (zero) and bigger punishments create coefficient values that are closer to zero. The lasso regression equation is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_r x_r + \lambda \sum |\beta_i| + \varepsilon$$

where y represents the dependent variable, $x_1 + x_2 + \cdots + x_r$ the independent variables, β_0 the intercept (constant term) and $\beta_1 + \beta_2 + \cdots + \beta_r$ the coefficients for the independent variables, λ the penalty parameter that controls the amount of regularization, ε the random error term and here the penalty term, $\lambda \sum |\beta_i|$, is the sum of the absolute values of the coefficients multiplied by the tuning parameter λ. The goal is to minimize the sum of squared errors while also minimizing the sum of the absolute values of the coefficients.

Lasso regression use for in various fields, including genetics, economics, machine learning and It's useful for dealing with high-dimensional datasets [4].

Chapter 2

WHY LINEAR REGRESSION

Linear regression is a very important and commonly used statistical analysis method. This is a very simple way to analyze data. One of its main advantages is that it's easy to understand the results. The difference between predicted and actual output values is minimized using linear regression by fitting a straight line or surface. We can find calculators that use a method called "least squares" to figure out the best line for a set of data that comes in pairs. The idea of linear regression says that when we change one thing, another thing changes in a similar way. Linear regression is very beneficial in the following situations [5].

1. Forecasting and prediction are both possible uses for linear regression in predictive modeling. We may create a linear regression model using a collection of independent variables to predict the value of the dependent variable for upcoming observations or future time periods. This makes it useful in fields like financial modeling, demand forecasting, and sales forecasting.
2. Linear regression is used to assess and quantify the connection between variables. It helps us to figure out the level and direction of the relationship, identify important factors, and understand how changes in the independent variables affect the dependent variable.
3. Linear regression can help with variable selection by selecting the most important independent variables. We can evaluate which factors have an important impact on the dependent variable and choose them for future research by examining the size and significance of the coefficients.
4. And so on....

   Overall, linear regression is a flexible and popular method for making

predictions, understanding correlations, and testing hypotheses. It is useful to many different industries, including engineering, finance, healthcare, and the social sciences.

2.1 Math behind in Linear Regression

Linear regression is a mathematical process of finding the most suitable line that goes equally with a set of data points while minimizing the sum of squared between actual and predicted values. There are the mathematical processes necessary for linear regression.

- Representation of a Model- when we use linear regression, we believe that the relationship between x and y can be shown by a linear equation. I showed equation (1.1) is known as the general form of a simple linear regression. For the purpose of review the equation is $y = \beta_0 + \beta_1 x + \varepsilon$, where y represents the dependent variable, x the independent variable, β_0 the intercept or the value of y when y is zero, β_1 the slope or the change in y for a unit change in x and ε the random error term that captures the unexplained variation.

  On the other hand, we must know the role of residual and random error by using linear regression.

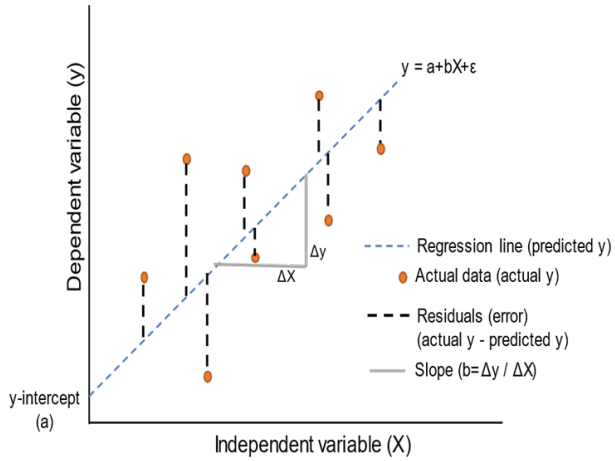

Figure 1: Regression Line

The picture 1 shows that the differences actual y – predicted y are called residuals while an error is the difference between the observed value and the true value. There are r observations $(x_1, y_1), (x_2, y_2) \cdots (x_r, y_r)$ used to build the regression model in regression analysis. Each observation includes two values: a value for one thing that can change (called the independent variable,x) and a value for another thing that is affected by the first thing (called the dependent variable,y).

- Cost Function - the purpose of linear regression is to find the coefficient values $\beta 0, \beta_1$ that minimize the difference between the observed and predicted values of y given by the linear model. The most common cost function used in linear regression is called the mean squared error (MSE).

- Optimization - ordinary least squares (OLS) is a very common math formula used to minimize MSE by finding the best coefficient values. The sum of squared residuals SSR is calculated as the sum of the squared differences between the observed values Y_i and the predicted values $\bar{Y}_i$:

$$SSR = \sum (Y_i - \bar{Y}_i)^2 \tag{2.1}$$

Least squares regression lines are a specific type of model that analysts frequently use to display relationships in their data. We can use least squares regression to mathematically find the best possible line and its equation. Statisticians call it "least squares" because it minimizes the sum of the squared residuals. Let's unpack what that means. If we want to calculate the residual mathematically, it's the simple subtraction which is **Residual = Observed value – Model value.**

Or, equivalently: Let R be the residual. Then,

$$R = y - \hat{y} \tag{2.2}$$

where $\hat{y}$ is the regression model's expected value of y.

Residuals represent the error in a least squares model. We will minimize the total error because it means that the data points are collectively as close to the model's values as possible.

Unfortunately, you can't just sum the residuals to represent the total error because the positive and negative values will cancel each other out even when they tend to be relatively large. Instead, least squares regression takes those residuals and squares them, so they're always positive. In this manner, the process can add them up without canceling each other. Statisticians refer to squared residuals by using equation (2.2) as squared errors and their total as the sum of squared errors (S), shown below mathematically.

$$S = \sum_{i=1}^{n} (y - \hat{y})^2. \tag{2.3}$$

where $\hat{y} = \beta_0 + \beta_1 x$, so to minimize S, we need to differentiate S with respect to β_0 and β_1.

$$\Delta S = 0 \tag{2.4}$$

This (2.4) leads to the following equations,

$$\frac{\partial S}{\partial \beta_0} = \frac{\partial}{\partial \beta_0}\left(\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2\right) = -2\sum_{i=1}^{n}\left(y_i - \beta_0 - \beta_1 x_i\right).$$

This implies

$$\sum_{i=1}^{n} y_i = \beta_0 n + \beta_1 \sum_{i=1}^{n} x_i \tag{2.5}$$

And

$$\frac{\partial S}{\partial \beta_1} = \frac{\partial}{\partial \beta_1}\left(\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \right) = -2 \sum_{i=1}^{n} x_i \left(y_i - \beta_0 - \beta_1 x_i \right).$$

So we get

$$\sum_{i=1}^{n} x_i y_i = \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2. \tag{2.6}$$

from (2.5) and (2.6) we get

$$\beta_0 = \frac{1}{n} \sum_{i=1}^{n} y_i - \frac{\beta_1}{n} \sum_{i=1}^{n} \sum_{i=1}^{n} x_i.$$

But $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$, and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ So,

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \tag{2.7}$$

To find β_1, we substitute (2.7) β_0 in equation (2.6), to get

$$\beta_1 = \frac{\sum_{i=1}^{n} x_i y_i - \bar{y} \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i} \tag{2.8}$$

Chapter 3

PYTHON PROGRAMMING LANGUAGE

3.1 Role of Python

One of the most popular languages is Python that many people used by those interested in data science is Python. We can modify and use any type of data with using libraries.

In this thesis We are going to start implementing linear regression in Python. To do this, we will apply the proper packages and their functions and classes. We can perform the linear regression method in a variety of programs and environments, including [6]:

- **R with linear regression**
- **Python with linear regression**

To do linear regression analysis in Python [7], we can use libraries like as NumPy, pandas, and scikit-learn. Here's an example of how to use scikit-learn which is one of Python's most popular machine learning tools.

$$import \quad numpy \quad as \quad np$$

$$from \quad sklearn.linear\_model \quad import \quad LinearRegression$$

In the above example, we make a model for measuring something using a special code called LinearRegression from scikit-learn.

$$x = np.array([[1,3],[2,4],[3,6],[4,8]])$$

$$y = np.array([6,8,10,12])$$

$$model = LinearRegression()$$

$$model.fit(x,y)$$

To train the model, we give it the input variables x and the output variable y to the fit method.

$$new\_data = np.array([[5,10],[6,12]])$$

$$predictions = model.predict(new\_data)$$

$$print(predictions)$$

After learning, we can use the model to guess what might happen with new information by using the predict function and giving it the new data. The output will be '[16. 18.]' for the new data points.

3.2 Earthquake Parameter Prediction with Linear Regression

For this section we analyze and visualize the earthquake data in python with **Matplotlib** library. There are steps we have to check as following:

- Get the datasets from earthquake in csv file format
- Building a model with Linear Regression
- Visualization with Matplotlib and Seaborn

First of all, we would like to continue using the python packages before read the file [8].

$$Import \quad numpy \quad as \quad np$$

$$Import \quad pandas \quad as \quad pd$$

$$from \quad sklearn.linear\_model \quad import \quad LinearRegression$$

$$from \quad sklearn \quad import \quad metrics$$

$$import \quad matplotlib.pyplot \quad as \quad pltt$$

$$import \quad seaborn \quad as \quad sb$$

$$import \quad warnings$$

$$warnings.filterwarnings('ignore')$$

We could read the data because files with py and csv extensions are on the same file path. Let's read the **csv** file with pandas:

The result in the picture shows that we have a matrix file of 23412 rows and 21

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Depth Seismic Stations	Magnitude
0	1/2/1965	13:44:18	19.2460	145.6160	Earthquake	131.60	NaN	NaN	6.0
1	1/4/1965	11:29:49	1.8630	127.3520	Earthquake	80.00	NaN	NaN	5.8
2	1/5/1965	18:05:58	-20.5790	-173.9720	Earthquake	20.00	NaN	NaN	6.2
3	1/8/1965	18:49:43	-59.0760	-23.5570	Earthquake	15.00	NaN	NaN	5.8
4	1/9/1965	13:32:50	11.9380	126.4270	Earthquake	15.00	NaN	NaN	5.8
...	...	...	...	...	...	...	...	...	...
23407	12/28/2016	8:22:12	38.3917	-118.8941	Earthquake	12.30	1.2	40.0	5.6
23408	12/28/2016	9:13:47	38.3777	-118.8957	Earthquake	8.80	2.0	33.0	5.5
23409	12/28/2016	12:38:51	36.9179	140.4262	Earthquake	10.00	1.8	NaN	5.9
23410	12/29/2016	22:30:19	-9.0283	118.6639	Earthquake	79.00	1.8	NaN	6.3
23411	12/30/2016	20:08:28	37.3973	141.4103	Earthquake	11.94	2.2	NaN	5.5

23412 rows × 24 columns

Figure 2: Earthquake Parameters

columns with Earthquake data from 1965 to 2016. The columns are also distributed according to the title of the data. Let me explain them now.

- Id: order number of the earthquake
- Date: earthquake occurrence date
- Time: time of the earthquake
- Lat: latitude of the earthquake epicenter
- Long: longitude of the earthquake epicenter
- Type: the type of earthquake
- Md: magnitude depending on time
- Azimuthal Gap: the maximum angle separating two adjacent seismic stations
- Horizontal Distance: from the epicenter to the nearest station (in km)

- Dist: district of the occurred earthquake

Then we can look the descriptive statistical measures *df.describe()* that gives us some idea regarding the distribution of the data.

Out[7]:

	Latitude	Longitude	Depth	Depth Error	Depth Seismic Stations	Magnitude
count	23412.000000	23412.000000	23412.000000	4461.000000	7097.000000	23412.000000
mean	1.679033	39.639961	70.767911	4.993115	275.364098	5.882531
std	30.113183	125.511959	122.651898	4.875184	162.141631	0.423066
min	-77.080000	-179.997000	-1.100000	0.000000	0.000000	5.500000
25%	-18.653000	-76.349750	14.522500	1.800000	146.000000	5.600000
50%	-3.568500	103.982000	33.000000	3.500000	255.000000	5.700000
75%	26.190750	145.026250	54.000000	6.300000	384.000000	6.000000
max	86.005000	179.998000	700.000000	91.295000	934.000000	9.100000

Figure 3: Describe of Earthquake Parameters

From the above description 3 of the dataset, we can conclude the maximum magnitude of the Earthquake is 9.1 and the maximum depth at which the earthquake started is 700 km below the ground.

Then, we are going to use $value\_counts()$ method on Magnitude Type and Source to identify the count of each category in that column.
My purpose is to show that how many are there categories of Magnitude Type and Source for using 4 and 5 pictures. We can see that there are 7722 moment magnitude (MW) and 20630 sources and so on.

Secondly, **EDA** (Exploratory Data Analysis) [9] means that looking at the data by using pictures to help understand it. This is a tool used to find trends and patterns by looking at graphs and numbers. By using **plt.figure()** we can follow that the changes caused by earthquakes of greater size are more easily seen during the monsoon season.

```
In [8]: df['Magnitude Type'].value_counts()

Out[8]: MW      7722
        MWC     5669
        MB      3761
        MWB     2458
        MWW     1983
        MS      1702
        ML        77
        MWR       26
        MD         6
        MH         5
        Name: Magnitude Type, dtype: int64
```

Figure 4: Showing the Counts of Magnitude Type

```
In [9]: df['Source'].value_counts()

Out[9]: US           20630
        ISCGEM        2460
        ISCGEMSUP      120
        CI              61
        GCMT            55
        NC              51
        AK              12
        OFFICIAL         8
        UW               6
        NN               4
        ATLAS            3
        SE               1
        PR               1
        Name: Source, dtype: int64
```

Figure 5: Showing the Sources of Each Variable

pltt.figure(figsize=(10, 5))

x1 = df.groupby('Year').mean()['Depth']

x1.plot.bar()

pltt.show()

plt.figure(figsize=(10, 5))

sb.lineplot(data=df, x='Day', y='Magnitude')

plt.show()

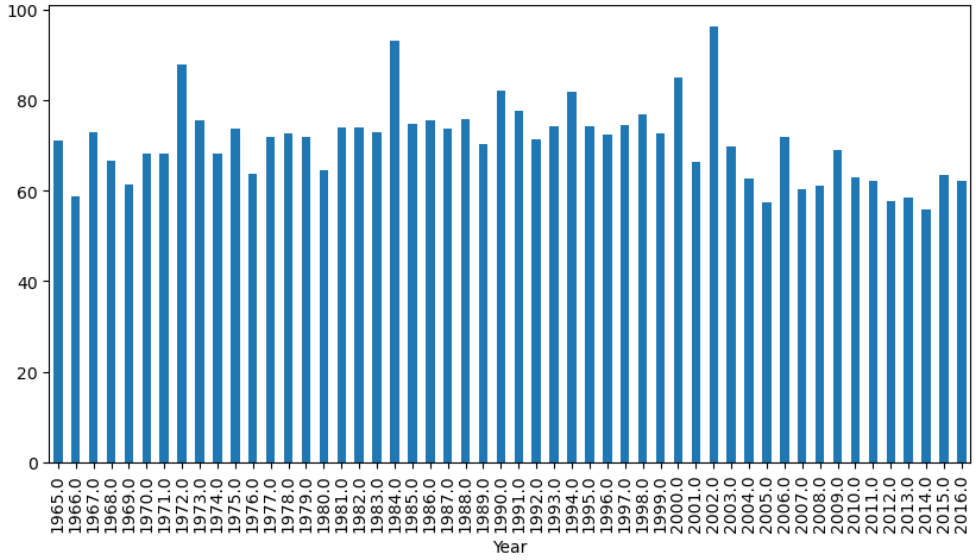

Figure 6: Showing the Depth of Each Year

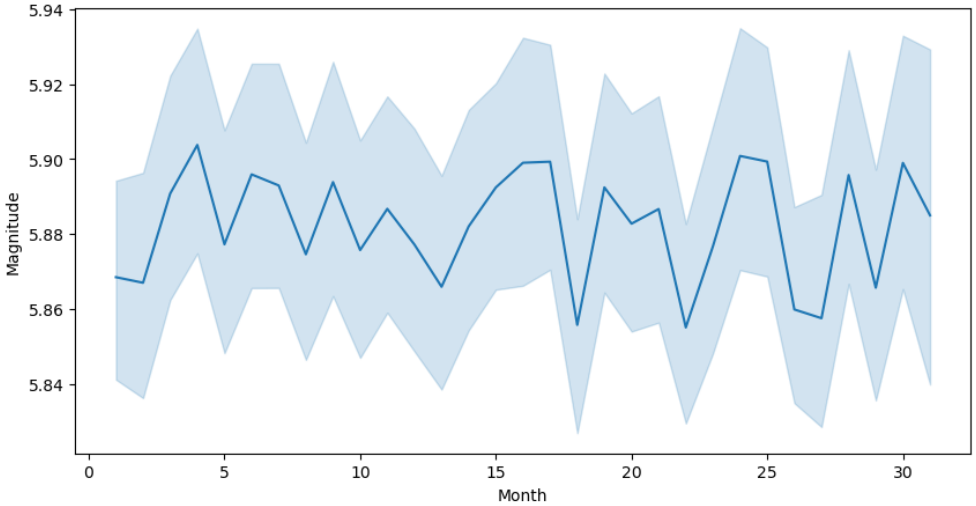

Figure 7: Showing the Magnitude of Each Day

From the table 14 and 7 we see that which earthquakes are reducing or increasing with every passing year and month. The highest depth happened in US on 19 August 2002 and the highest magnitude is 8.7 on 4 February 1965.

In the Python programming language [10], *plt.subplots* stands as a function offered by the *Matplotlib* library, widely utilized to generate visual representations like plots, charts, and graphs. We're going to examine "Depth" and "Magnitude" by using the plt.subplots code.

$$pltt.subplots(figsize = (15,5))$$

$$pltt.subplot(1,2,1)$$

$$sb.distplot(df['Depth'])$$

$$pltt.subplot(1,2,2)$$

$$sb.boxplot(df['Depth'])$$

$$pltt.show()$$

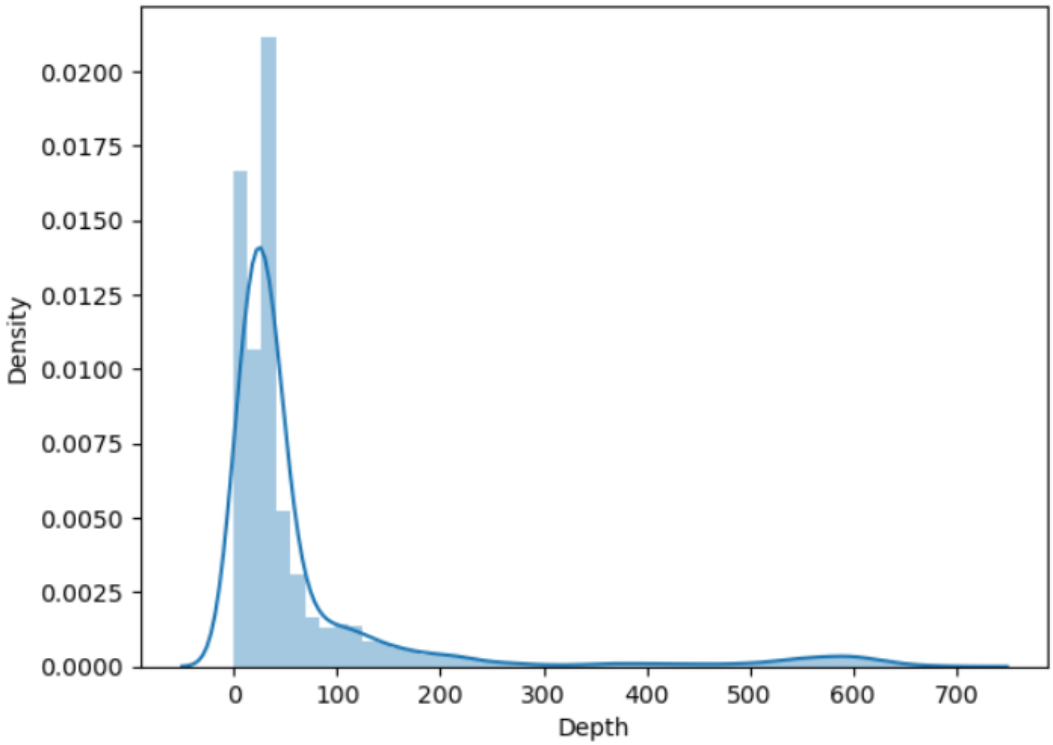

Figure 8: Distribution Graph in Earthquake's Depth

Another code is for "Magnitude":

$$pltt.subplots(figsize = (15,5))$$

$$pltt.subplot(1,2,1)$$

$$sb.distplot(df['Magnitude'])$$

$$pltt.subplot(1,2,2)$$

$$sb.boxplot(df['Magnitude'])$$

$$pltt.show()$$

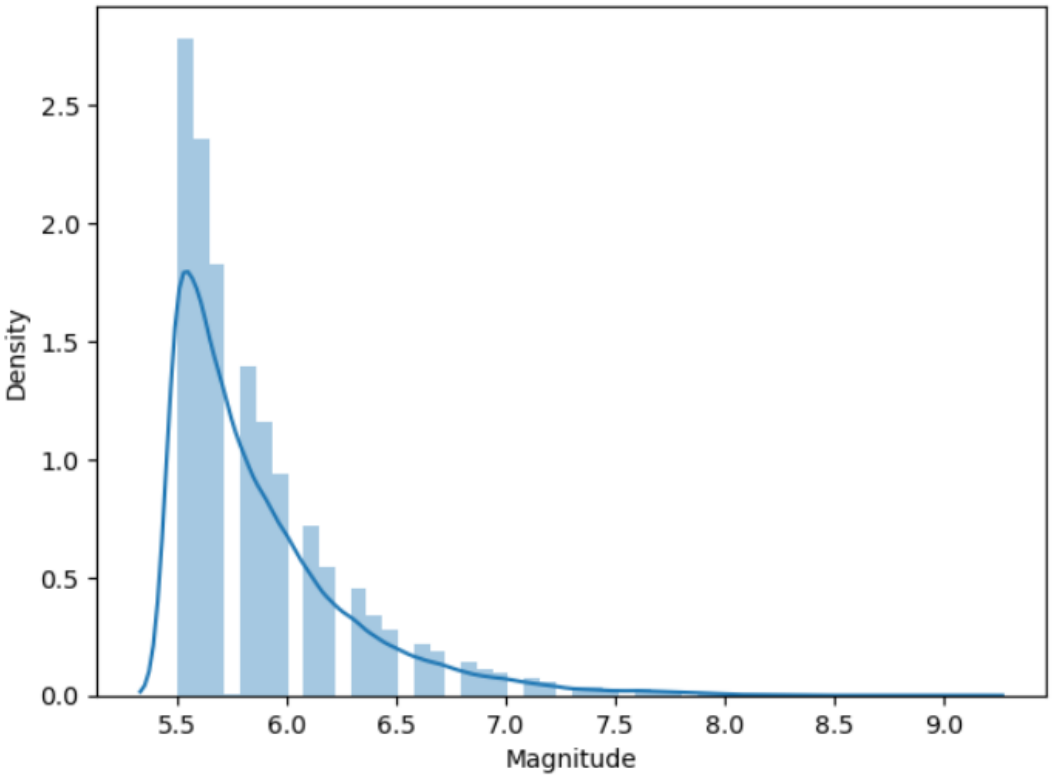

Figure 9: Distribution Graph in Earthquake's Magnitude

From the distribution graph 8 and 9 there are some outliers that we can check them using a different kind of graph called a boxplot. The important thing to notice is that the earthquake's depth and magnitude are more towards one side, which is the left side.

We know that some natural events happen in a predictable pattern, and earthquakes

are one of them. We can see that the size of earthquakes also follows this pattern. So, we're going to use *Plotly* library to make a map showing where earthquakes are more likely to happen based on the latitude and longitude data.

$$pltt.figure(figsize = (10,8))$$

$$sb.scatterplot(data = df, x = 'Latitude', y = 'Longitude', hue = 'Magnitude')$$

$$pltt.show()$$

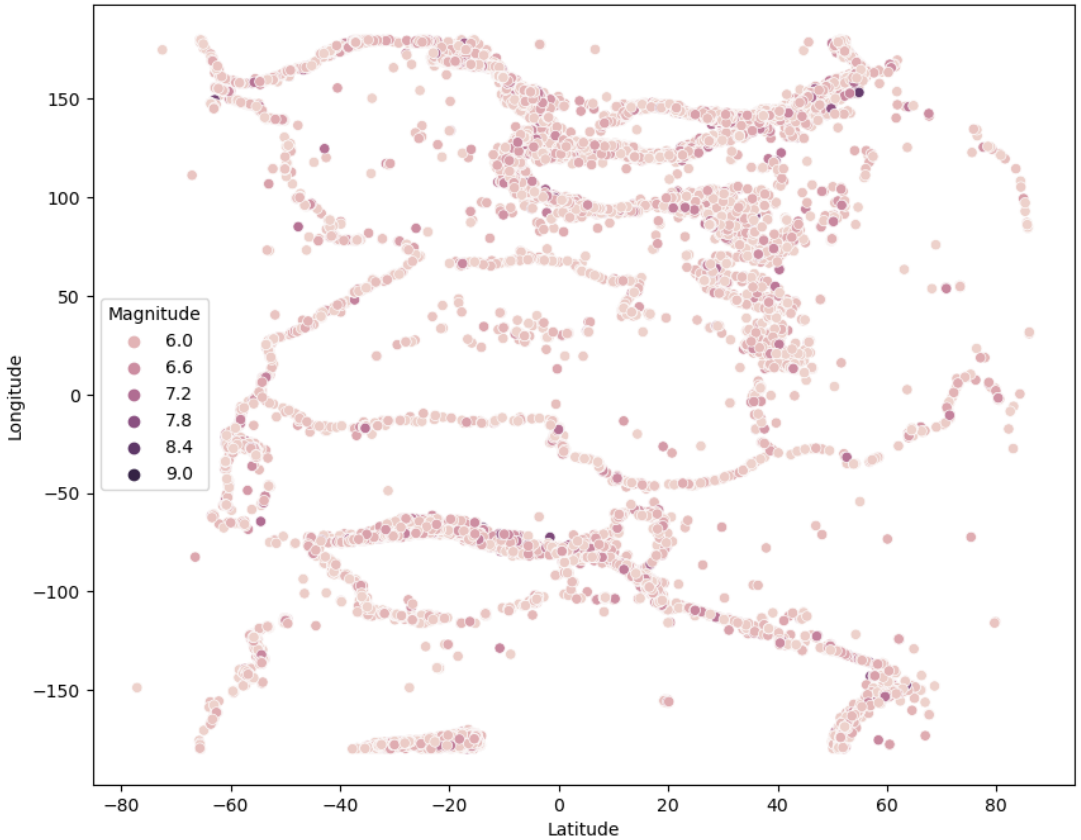

Figure 10: Plot the Data's on the Map

3.3 Parameter Estimate and Correlation

We have to show the relationship between depth and magnitude with the native figure size and this relationship looks linear between them.

$$plt.figure(figsize = (7,5))$$

$$plt.scatter(df['Depth'], df['Magnitude'])$$

$$plt.xlabel("Depth", fontsize = 12)$$

$$plt.ylabel("Magnitude", fontsize = 12)$$

$$plt.title("RelationbetweenDepthandMagnitude")$$

$$plt.show()$$

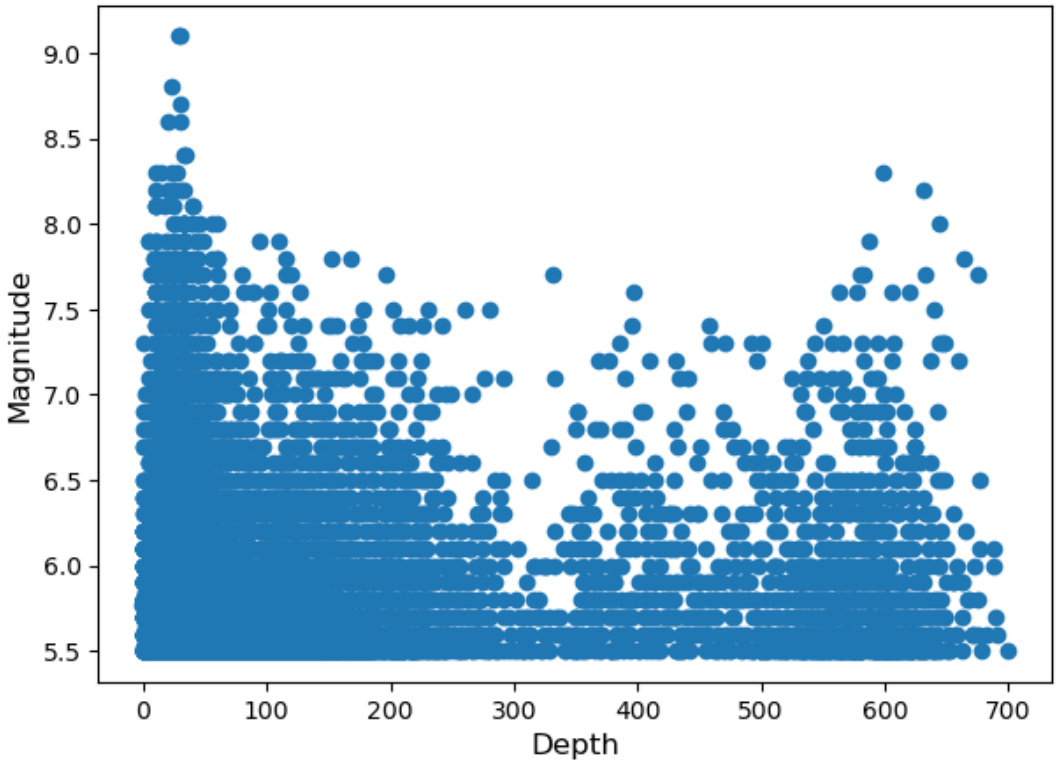

Figure 11: Relation Between Depth and Magnitude

Firstly, we need to figure out two numbers (β_0 and β_1) that we can draw a straight line since we know the connection is straight. We showed how to find β_0 and β_1 with partial equation in (2.7) and (2.8). By using the equations I can show with python code to find the coefficients.

$$x = df['Depth']$$

$$x\_bar = (df['Depth']).mean()$$

$$y = df['Magnitude']$$

$$y\_bar = (df['Magnitude']).mean()$$

$$beta1 = ((x - x\_bar) * (y - y\_bar)).sum()/((x - x\_bar) **2).sum()$$

$$print("Beta\_1 \quad coefficient \quad estimate : " + str(round(beta1,4)))$$

From the code the output is **"Beta_1 coefficient estimate : 0.0001".**

$$beta0 = y\_bar - beta1 * x\_bar$$

$$print("Beta\_0 \quad coefficientestimate : " + str(round(beta0,4)))$$

From the code the output is **"Beta_0 coefficient estimate : 5.8768".**

Then, we can draw a line on a graph to see if it looks right or not.

$$pltt.figure(figsize = (7,5))$$

$$pltt.scatter(df['Depth'], df['Magnitude'])$$

$$pltt.plot(df['Depth'], beta1 * df['Depth'] + beta0, c =' r')$$

$$pltt.xlabel("Depth)", fontsize = 12)$$

$$pltt.ylabel("Magnitude", fontsize = 12)$$

$$pltt.show()$$

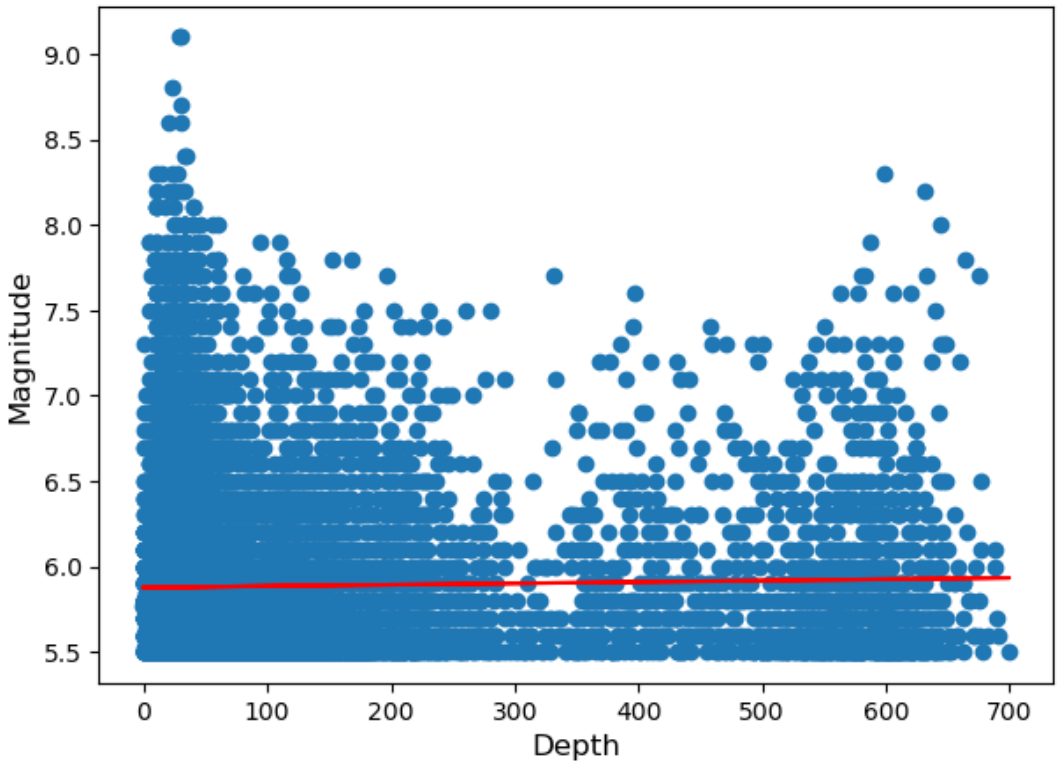

Figure 12: Relation Between Depth and Magnitude

From the picture 12 we don't know that the line is good or not. So, we have to know about correlation before to decide the line is good or not.

Correlation represents a statistical metric that determine how strong the relationship is between two variables using statistics. There are two main kinds of correlation which are positive correlation indicates that when one variable goes up, the other variable also tends to rise and a negative correlation implies that as one variable increases, the other variable typically decreases.

Correlation *heatmaps* are a type of plot that to show how strongly different numbers relate to each other. These plots are used to see how different things are connected and how their connections are strong. A correlation plot typically contains a number of numerical variables, where each number is represented by a column. The rows show how each pair of variables is related to each other. The cell values indicate the strength of the relationship which positive numbers mean a good relationship and

negative numbers mean a bad relationship. Correlation heat-maps can be used to find potential relationships between variables and to understand the strength of these relationships. The different colors used for the cells help us easily see how variables are related to each other. Additionally, correlation heatmaps can uncover both linear and nonlinear associations between variables.

This is a correlation heatmap generated to examine the linear relationship between depth and magnitude. Here is how the correlation *"heatmap"* will look like:

$$sb.heatmap(df.corr(), annot = True, cmap =' magma')$$

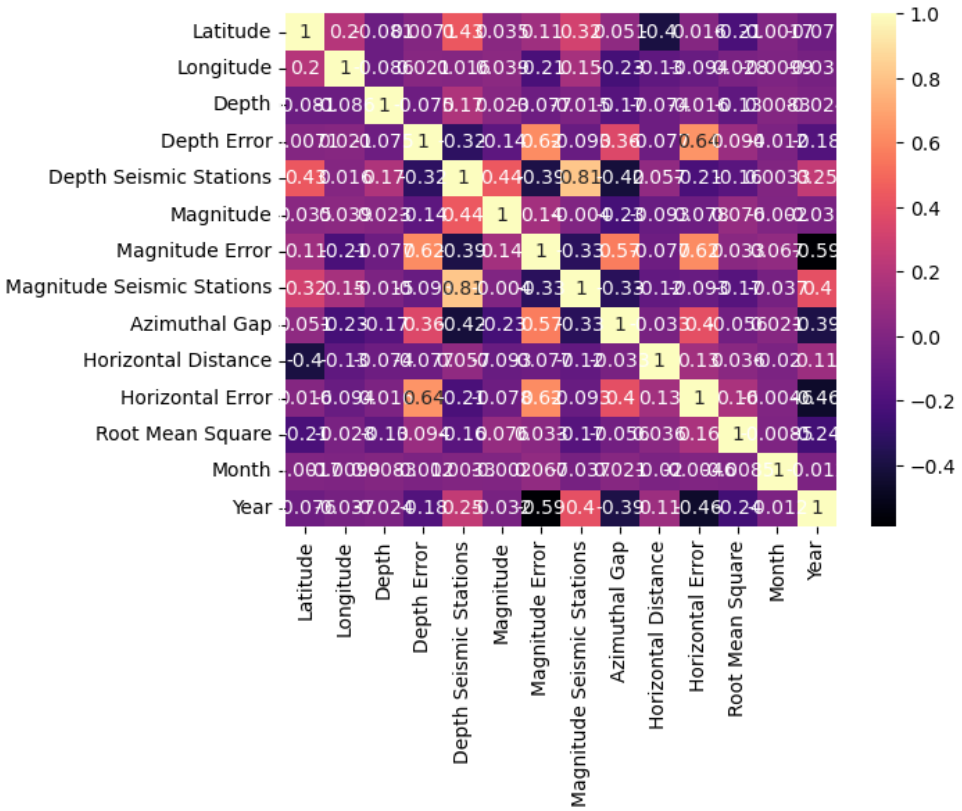

Figure 13: Correlation Heatmap for Earthquake Dataset

From the picture 13 it's not readable for values. So, we can use the code *df.corr()* to understand the variables.

Out[11]:

	Latitude	Longitude	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Error	Magnitude Seismic Stations	Azimuthal Gap	Horizontal Distance	Horizontal Error	Root Mean Square
Latitude	1.000000	0.203546	-0.081020	0.007080	0.433815	0.034987	0.113208	0.315075	0.050794	-0.396768	0.015625	-0.214762
Longitude	0.203546	1.000000	-0.085861	0.020552	0.015924	0.038579	-0.214609	0.148510	-0.233097	-0.131313	-0.093827	-0.028061
Depth	-0.081020	-0.085861	1.000000	-0.074609	0.174663	0.023457	-0.076918	-0.015254	-0.171162	-0.073832	-0.016467	-0.134002
Depth Error	0.007080	0.020552	-0.074609	1.000000	-0.320579	-0.135880	0.618254	-0.093292	0.357704	-0.077423	0.644593	0.094398
Depth Seismic Stations	0.433815	0.015924	0.174663	-0.320579	1.000000	0.440582	-0.385993	0.813374	-0.420556	0.056619	-0.214959	-0.158620
Magnitude	0.034987	0.038579	0.023457	-0.135880	0.440582	1.000000	0.135573	-0.003972	-0.233579	-0.092609	-0.078406	0.075865
Magnitude Error	0.113208	-0.214609	-0.076918	0.618254	-0.385993	0.135573	1.000000	-0.334062	0.567411	-0.076744	0.617721	0.032616
Magnitude Seismic Stations	0.315075	0.148510	-0.015254	-0.093292	0.813374	-0.003972	-0.334062	1.000000	-0.334864	-0.117606	-0.093143	-0.167473
Azimuthal Gap	0.050794	-0.233097	-0.171162	0.357704	-0.420556	-0.233579	0.567411	-0.334864	1.000000	-0.033482	0.396450	-0.056217
Horizontal Distance	-0.396768	-0.131313	-0.073832	-0.077423	0.056619	-0.092609	-0.076744	-0.117606	-0.033482	1.000000	0.126877	0.035778
Horizontal Error	0.015625	-0.093827	-0.016467	0.644593	-0.214959	-0.078406	0.617721	-0.093143	0.396450	0.126877	1.000000	0.157842
Root Mean Square	-0.214762	-0.028061	-0.134002	0.094398	-0.158620	0.075865	0.032616	-0.167473	-0.056217	0.035778	0.157842	1.000000

Figure 14: Correlation for Variables

According to the displayed correlation "heatmap", we can get some of the following information about variables:

- Variables such as Latitude and Longitude, Depth Seismic Stations and Magnitude Error, Magnitude Seismic Stations and Azimuthal Gap and Horizontal Error are having strong positive correlations.
- Variables such as Depth and Depth Error, Magnitude and Horizontal Distance, Root Mean Square are having strong negative correlations.
- There are some variables that are not connected to each other and their connection measure is close to zero.

We see that when correlation is near 0, the linear relationship is weak. So the line between depth and magnitude relationship is not good.

Chapter 4

R PROGRAMMING LANGUAGE

4.1 R Overview and History

In statistics, regression analysis is a way to find the connection between different variables in a dataset using *R programming and statistics*. This method is useful for figuring out what causes a problem. We can easily determine the most important elements, the ones that can be disregarded and how these factors work together when we do a regression. In general, regression analysis is used to find out how the variables in a dataset are related to each other. Regression analysis helps us understand how the dependent variables change when one independent variable changes while keeping the other independent variables constant.

R is a public programming language that it is commonly used for analyzing data and as software for statistics. It was founded by Ross Ihaka and Robert Gentleman at the University of Auckland in New Zealand in the beginning of the 1990s.It has a command-line interface. The interface has a prompt which is usually shown as the '>' symbol. It can be used with Microsoft Windows, Linux, and Apple macOS, which are common operating systems. The R programming language contains a large library that is mostly developed in C, C++, Fortran, and R. Actually, this software is only used for statistical and graphical approaches such as statistical inference, time series analysis, linear regression, and machine learning (ML).

In the modern world, R is becoming very popular because of technology. Almost two million scientists and statisticians worldwide who work with data and statistics use the R coding language to create business applications.

As mentioned before, R is a computer language and software program used for analyzing data, making charts, and creating reports. Here are the main things about R that are important.

- R is a programming language that is well-designed and easy to use. It has features such as input and output capabilities, conditionals, loops, and user-defined recursive functions.
- In R, there are special ways of doing calculations on arrays, lists, vectors, and matrices.
- R has a solid system for managing and saving data.
- R provides tools for analyzing and showing data in pictures, which can be used on a computer or printed on paper.
- R has a large collection of helpful tools for studying and analyzing data.

Overall, R is a strong and versatile language used to analyze statistics and data science. It is popular in academia, companies, and research. Many people like to use it for data analysis and visualization because it has a big community and many useful tools.

4.2 A Case Study on Diamond Prices with R

In R, we will examine the diamonds dataset from the ggplot2 package. This dataset includes over 50,000 entries with ten factors including as price, carat, cut, color, and so on. We are interested in understanding how several variables impact the cost of a diamond. Precisely, our aim to explore how the diamond's carat weight, distinct cuts, and varying levels of depth and table influence its price. The important thing is to check if the calculated coefficients are statistically significant for linear regression. All we need to check out the diamonds dataset and modify it for analysis. We put the excel file in R studio called "Diamonds".

In the figure 15 there are independent (x) variables which are **carat, cut, depth** and **table** to examine their relationships with the **price** (y) of diamonds. We want to focus

```
> head(Diamonds)
# A tibble: 6 × 11
   ...1 carat cut       color clarity depth table price
  <dbl> <dbl> <chr>     <chr> <chr>   <dbl> <dbl> <dbl>
1     1  0.23 Ideal     E     SI2      61.5    55   326
2     2  0.21 Premium   E     SI1      59.8    61   326
3     3  0.23 Good      E     VS1      56.9    65   327
4     4  0.29 Premium   I     VS2      62.4    58   334
5     5  0.31 Good      J     SI2      63.3    58   335
6     6  0.24 Very Good J     VVS2     62.8    57   336
```

Figure 15: Diamonds Dataset in R

just on the carat to test the connection between price. Then, we will only continue using these variables.

The presented simple linear regression equation is $y = \beta_0 + \beta_1 x$ for describe the connection between the response variable (y) and the predictor (x), where β_0 represents the intercept and β_1 represents the slope. The aim of describing the values of β_0 and β_1, where y represents the price and x corresponds to the carat for create the linear model.

```
> x<- Diamonds$carat
> y<- Diamonds$price
> fit<- lm(y~x)
> fit

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      -2256         7756
```

Figure 16: Coefficients of Carat and Price

We got from the figure 16 the intercept β_0 and slope β_1 which is below the x column by using the **fit** variable. If x is 0, it means the value of y is intercept and the price of a diamond is *-$2,256* with 0 carat. Yes, it might not sound logical or understandable,

but that is how it is evaluated. If I want to understand something from it, I can calculate the average of our x variable and subtract it from each value. This way, I can interpret the intercept as *the y value when x is the average carat.*

```
> x <- Diamonds$carat - mean(Diamonds$carat)
> y <- Diamonds$price
> lm(y~x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
       3933         7756
```

Figure 17: The Mean of Carat

We notice form the figure 17 that the slope remains consistent, while the intercept is now **$3,933** by using the price of an average-carat diamond code.We determined the slope is **7756**, indicating that for every additional carat a diamond possesses, its value increases by $7,756. If we have a negative value, it means that for each increase in x, y is decreased by that value. So our equation is $y = -2256 + 7756x$. We can guess how much a diamond may cost based on its size in carats. For instance, we wish to calculate the approximate value of a diamond weighing four carats. By substituting the given carat value into the equation $y = -2256 + (7756)(4)$, the result is $28,768$.

We can also interpreting other coefficients by using **lm** object and the parameter of **summary** function for get more details.

```
> summary(fit)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-18585.4   -804.7    -19.1    537.5  12731.7

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2256.40      13.05  -172.8   <2e-16 ***
x            7756.44      14.07   551.4   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1549 on 53941 degrees of freedom
Multiple R-squared:  0.8493,    Adjusted R-squared:  0.8493
F-statistic: 3.041e+05 on 1 and 53941 DF,  p-value: < 2.2e-16
```

Figure 18: More Details for Interpreting Other Coefficients

There are more consequences by using *lm* object with the *summary* function in the figure 18. We'll examine at any of them to figure out if the coefficients that were estimated and model are appropriate.

We know that *residuals* refer to the variances among the actual and estimated values of the dependent variable.

> Diamonds %>%

select(carat, price) %>%

ggplot(aes(carat, price)) +

geom_point(shape=16, colour = 'purple', size = 2, alpha = 0.5) +

geom_smooth(method = 'lm', colour = 'green') +

theme_minimal()

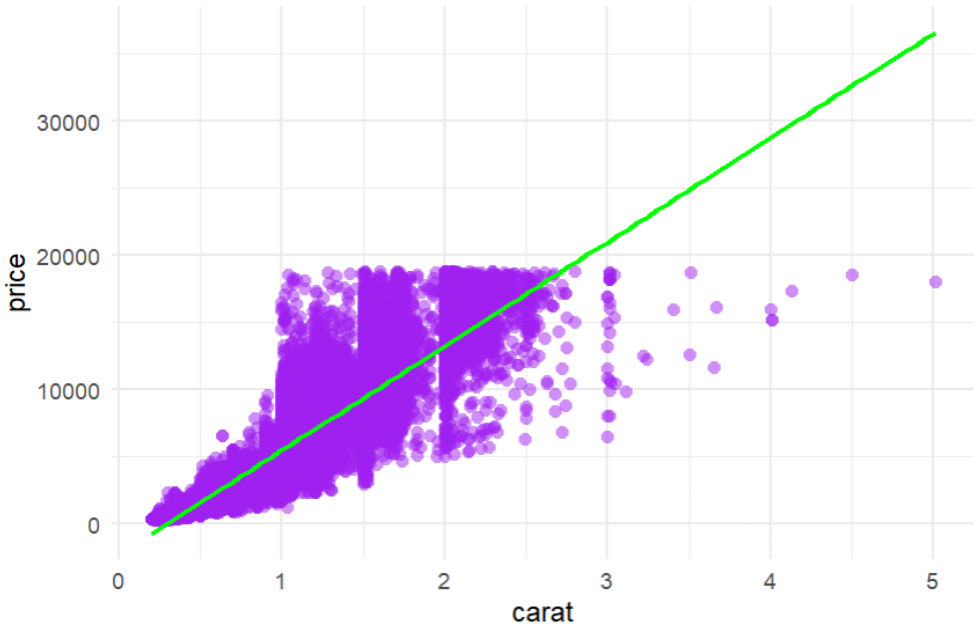

Figure 19: Scatter Plot of Diamonds Carat and Price with Regression Line

The important thing is to analyze the relationship between the variables and determine whether it holds statistical significance, as well as it involves the estimation of the y value using a given x value. That's why we'll show interpretation of linear regression using the diamonds dataset.

Let's give a brief explanation of our basic calculation model using the diamand data from the figure 18. We know our regression equation is represented as $y = -2256 + 7756x$ with coefficient values and the p-value is below the significance cutoff point of 0.05. Both the multiple and adjusted R-squared values stand at 0.8493 and it's quite a few lofty.Thus, we can conclude that this simple linear regression model is robust and reliable. The numbers we found are very important in understanding how much carat affects the price.The model demonstrates its reliability for forecasting diamond prices when using x values that were not present in the witnessed data.

The residual refers to the variation in height between the predicted y represented by the green line and the actual y depicted by the purple dots. We have to focus on the **p-value** among these details because it is important which it demonstrates the

statistical significance of the calculated coefficients.P-values are used in regression analysis to **test hypotheses**. In this situation, our null hypothesis is that both β_0 and β_1 are zero, which means there is no effect fore these variables on our response variable,whereas the alternative hypotheses claim that β_0 and β_1 are not zero, which means that the coefficients do affect the response variable.So,the p-values play a crucial role in determining whether the null hypotheses can be rejected. If the p-value is *lower than 0.05*. So, we have sufficient evidence to reject the null hypothesis and conclude that the coefficients have statistical significance which the coefficient is likely not to equal zero.

4.2.1 Multiple Linear Regression in R

Until now we're done the concepts of statistical technique by using linear regression example, now I will check the multiple linear regression using the similar datasets.

> fit2 <- lm(price carat + cut + depth + table, Diamonds)

> summary(fit2)

```
Call:
lm(formula = price ~ carat + cut + depth + table, data = Diamonds)

Residuals:
     Min       1Q   Median       3Q      Max
-17513.6   -786.5    -40.0    521.5  12596.9

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3290.717    481.090    6.84    8e-12 ***
carat        7890.787     14.047  561.76   <2e-16 ***
cutGood       985.742     44.402   22.20   <2e-16 ***
cutIdeal     1506.180     44.049   34.19   <2e-16 ***
cutPremium   1224.600     42.553   28.78   <2e-16 ***
cutVery Good 1305.137     42.457   30.74   <2e-16 ***
depth         -73.670      5.302  -13.89   <2e-16 ***
table         -41.807      3.881  -10.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1508 on 53935 degrees of freedom
Multiple R-squared:  0.8571,    Adjusted R-squared:  0.8571
F-statistic: 4.62e+04 on 7 and 53935 DF,  p-value: < 2.2e-16
```

Figure 20: Multiple Linear Regression for Other Datasets

In the 20 regression summary, both coefficients yield p-values of 2e-16, which is comparable to 0.0000000000000002, which is significantly less than 0.05. So, we have the grounds to dismiss the null hypothesis this signifies a connection between the "carat" and "price.".

The another way is our goodness of fit measurements are **Multiple R-squared and Adjusted R-squared**.These figures indicate how much of the changes in the y variable can be explained by the model. The numbers go from 0 to 1, and I want these numbers to be lofty. Although they have the equal numbers, our attention should be directed towards the *adjusted R-squared* since it takes into account the presence of irrelevant variables, which the *multiple R-squared* does not consider.

As we see there are many coefficients for *cut* variable because it is considered an ordinal factor variable or a categorical data [11] type in statistics. In regression models, we handle categorical data in a different way. R converts the category variable into sets of binary variables as yes or no options represented by 1's and 0's.In R, additional variables will be generated to represent the values present in the original variable. These new variables will have one value that serves as a reference point for the comparisons.There are 5 categories which are **fair, good, very good, premium, ideal** and four variables were created in R, with the first level of the ordered factor serving as the reference point. The cut variable values are being represented by *cutGood, cutVery Good, cutPremium, cutIdeal*. Selecting *Fair cut* or level 1 as the reference value implies the absence of a column.When all the cut variables have a value of 0, it indicates that the specific observation corresponds to a "fair" type of cut. This approach is known as **dummy coding**.

Regression analysis uses *dummy coding*, which is known as one-hot encoding, to represent categorical variables with binary values (0 or 1). When working with categories in a regression model, we cannot use them directly because the model

needs numbers as input.We change categorical variables into a type that we can easily use in regression models when we employ dummy coding. This means making additional variables (dummy variables) for every category in the original category variable.When dealing with a categorical variable featuring "k" categories, "k-1" dummy variables are produced to represent each category accordingly. For instance, we have a categorical variable called "Color" with three distinct categories: "Green", "Yellow" and "Orange". Through dummy coding, we would create two dummy variables, namely "Yellow" and "Orange".

So, for example:

- If the color is "Yellow," then "Yellow" = 1 and "Orange" = 0.
- If the color is "Orange," then "Yellow" = 0 and "Orange" = 1
- If the color is "Green," then "Yellow" = 0 and "Orange" = 0 (implicitly).

Let's go back our project to look our coefficients. As we see the intercept is 3290.717, signifies the value of "y" when all "x" variables are set to 0. However, there is a supplementary interpretation with the inclusion of a categorical variable in our model.The regression model excludes the "fair cut" category due to dummy coding. As previously explained, when "cutGood" to "cutVery Good" are all set to 0, it signifies that the observation corresponds to a *"fair"* cut.Consequently, the intercept denotes the *diamond's price when it possesses a "fair" cut, assuming that all other variables are set to 0*.

The "carat" variable has a slope of *7890.787*, showing the amount by which the diamond's price increases for each additional carat. Regarding the "cut" variable, its slopes are *985.742, 1506.180, 1224.600,* and *1305.137*, corresponding to the price increases for diamonds with *"good," "very good," "premium,"* and *"ideal"* cuts, respectively. To determine the slope for "fair" cut diamonds, we can infer it from the intercept value.

On the other hand, the slopes for the depth and table variables are **-73.670** and **-41.807**. These coefficients indicate the reduction in the diamond's price for each unit rise in depth and table measurements.Our parameters have values lower than 0.05, confirming that the results are statistically significant.

Chapter 5

PYTHON AND R LANGUAGES

5.1 Diabetes Parameter Prediction with Python

Diabetes is a health condition where the body has high levels of sugar in the blood for a long time. Higher than normal blood sugar levels consist of frequent urination, heightened thirst, and escalated hunger. The dataset's aim is to predict if a patient has diabetes or not with using specific measurements collected in the dataset.

As at 4.2 section I analyze and visualize the diabetes data in python with Matplotlib. Let's read the csv file with pandas:

```
Import numpy as np
Import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import matplotlib.pyplot as pltt
import seaborn as sb
import warnings
warnings.filterwarnings('ignore')
```

```
In [5]:  df = pd.read_csv('Diabetes.csv')
         df.head(768)
```

Out[5]:

	No	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	1	6	148	72	35	0	33.6	0.627	50	1
1	2	1	85	66	29	0	26.6	0.351	31	0
2	3	8	183	64	0	0	23.3	0.672	32	1
3	4	1	89	66	23	94	28.1	0.167	21	0
4	5	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...	...
763	764	10	101	76	48	180	32.9	0.171	63	0
764	765	2	122	70	27	0	36.8	0.340	27	0
765	766	5	121	72	23	112	26.2	0.245	30	0
766	767	1	126	60	0	0	30.1	0.349	47	1
767	768	1	93	70	31	0	30.4	0.315	23	0

768 rows × 10 columns

Figure 21: Diabetes Parameters

The dataset in 21 contains various medical predictor variables, and it includes one specific target variable labeled as "Outcome." These predictor variables encompass different factors, such as the number of pregnancies, BMI, insulin level, age, and other related attributes.

1. **Pregnancies:** Number of times pregnant.
2. **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
3. **BloodPressure:** Diastolic blood pressure.
4. **SkinThickness:** Triceps skin fold thickness.
5. **Insulin:** 2-Hour serum insulin (muU/ml).
6. **BMI:** Body mass index.
7. **DiabetesPedigreeFunction:** Diabetes pedigree function.
8. **Age:** Age (years).
9. **Outcome:** Class variable (0 or 1).

BMI is stands for "Body Mass Index" and it is listed among the medical predictor variables tells that if a person is overweight or not based on body mass in relation to their weight and height. There is a formula to calculate:

$$BMI = weight(kg)/(height(m))^2.$$

The dataset utilizes BMI values in conjunction with other predictor variables such as age, blood pressure, insulin levels, and more, to make predictions regarding the target variable. For instance, I can show the BMI and age variables together.

$$pltt.figure(figsize = (10,5))$$

$$x1 = df.groupby('Age').mean()['BMI']$$

$$x1.plot.bar()$$

$$pltt.show()$$

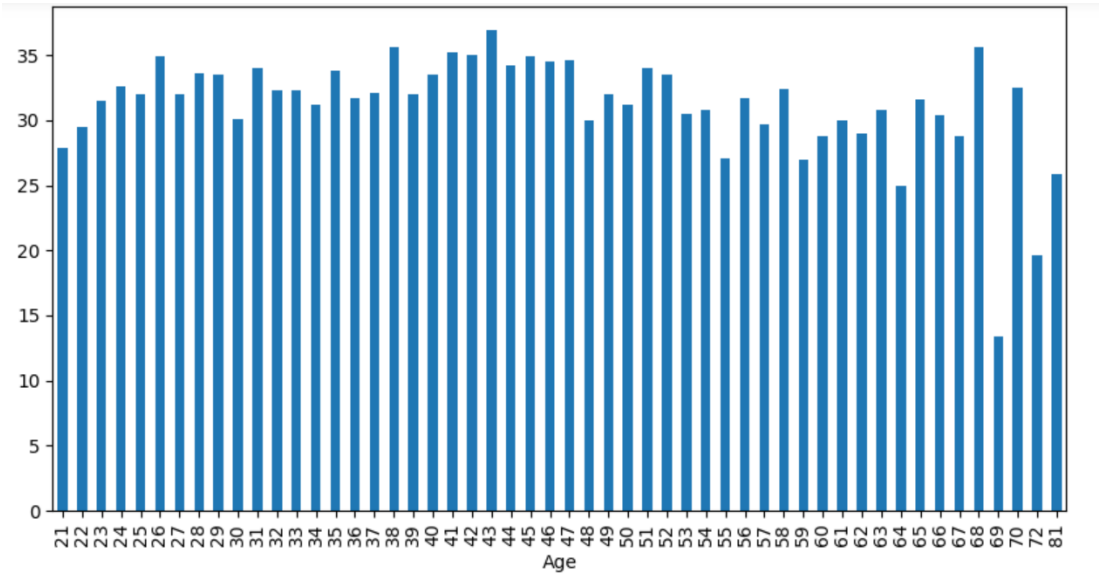

Figure 22: Relation Between BMI and Age

BMI typically rises with age until a certain threshold, after which it may level off or decline, especially in older age groups. This behavior can be affected by various factors, including alterations in metabolism, lifestyle choices, and changes in body composition throughout one's life. We see in the figure 22 that the max age is 81 and the max BMI around 37 in 43 age.

Then, we're going to use $value\_counts()$ method on outcome refers to a binary variable which is often represented as 0 (absence of diabetes) and 1 (presence of diabetes).

```
In [11]: df['Outcome'].value_counts()

Out[11]: 0    500
         1    268
         Name: Outcome, dtype: int64
```

Figure 23: Outcome of the Patients

We see from the figure 23 that there are 500 patients has not diabetes and 268 patients has diabetes.

We can find the **mean (average)** value of the **"Insulin"**, **"Pregnancies"** and **"Glucose"** variables based on different outcomes in the dataset.Here is the code and output:

```
In [15]: df.groupby("Outcome").agg({"Insulin": "mean"})

Out[15]:
```

Outcome	Insulin
0	68.792000
1	100.335821

Figure 24: Mean Value of Insulin

This code shows that display the mean "Insulin" value for patients with no diabetes (Outcome = 0) are 68.792000 and for patients with diabetes (Outcome = 1) are 100.335821 apart in the figure 24. Now we can determine the highest value of the "Insulin" variable for every distinct value in the "Outcome" column. Here is the code and output.

```
In [18]: df.groupby("Outcome").agg({"Insulin": "max"})
```

Out[18]:

Outcome	Insulin
0	744
1	846

Figure 25: The Highest Value of Insulin

Within the dataset in 25, there are two rows: one for Outcome = 0 and the other for Outcome = 1, each displaying the maximum "Insulin" value corresponding to that respective outcome group.We see that the max value of "Insulin" is 744 with no diabetes and 846 with diabetes.

```
In [16]: df.groupby("Outcome").agg({"Pregnancies": "mean"})
```

Out[16]:

Outcome	Pregnancies
0	3.298000
1	4.865672

Figure 26: Outcome of Pregnancies

This code shows in 26 that display the mean "Pregnancies" value for patients with no diabetes (Outcome = 0) are 3.298000 and for patients with diabetes (Outcome = 1) are 4.865672 apart. Now we can determine the highest value of the "Pregnancies" variable for every distinct value in the "Outcome" column. Here is the code and output.

```
In [19]: df.groupby("Outcome").agg({"Pregnancies": "max"})
```

Out[19]:

Outcome	Pregnancies
0	13
1	17

Figure 27: The Highest Value of Pregnancies

There are the max value of "Pregnancies" is 13 with no diabetes and 17 with diabetes in the figure 27.

```
In [17]: df.groupby("Outcome").agg({"Glucose": "mean"})
```

Out[17]:

Outcome	Glucose
0	109.980000
1	141.257463

Figure 28: Outcome of the Glucose

Again this code in 28 shows that display the mean "Glucose" value for patients with no diabetes (Outcome = 0) are 109.980000 and for patients with diabetes (Outcome = 1) are 141.257463 apart. Now we can determine the highest value of the "Glucose" variable for every distinct value in the "Outcome" column. Here is the code and output.

```
In [20]:  df.groupby("Outcome").agg({"Glucose": "max"})
```

Out[20]:

Outcome	Glucose
0	197
1	199

Figure 29: The Highest Value of Glucose

There are the max value of "Glucose" is 197 with no diabetes and 199 with diabetes in the figure 29.

By visualizing this data, we can gain a clear understanding of the outcome variable's distribution within the diabetes dataset and observe the number of instances belonging to each category (0 and 1). The code is following that:

$$f, ax = pltt.subplots(1,2, figsize = (18,8))$$

$$df['Outcome'].value_counts().plot.pie(explode = [0,0.1],$$

$$autopct = '\ \%1.1f\%\%', ax = ax[0],$$

$$shadow = True)$$

$$ax[0].set\_title('target')$$

$$ax[0].set\_ylabel('')$$

$$sb.countplot('Outcome', data = df, ax = ax[1])$$

$$ax[1].set\_title('Outcome')$$

$$pltt.show()$$

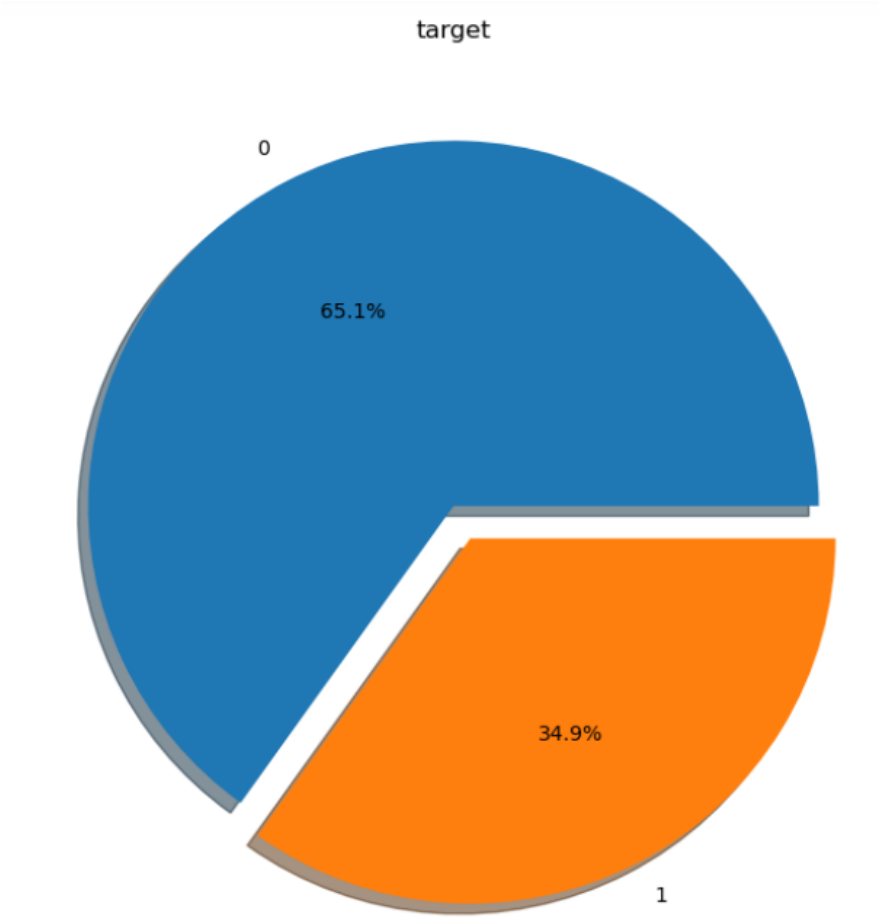

Figure 30: The Distribution of the Outcome Variable

It shows in 30 that there are 65.1% patients have not diabetes and 34.9% patients have diabetes.

Now we can analyze the correlation which to examine the relationship between variables. We have information that if the correlation is greater than 0. In such cases, when one variable increases, the other variable also increases. Conversely, if the correlation is less than 0, it suggests a negative correlation, where one variable increases while the other decreases.

$$sb.heatmap(df.corr(), annot = True, cmap =' magma')$$

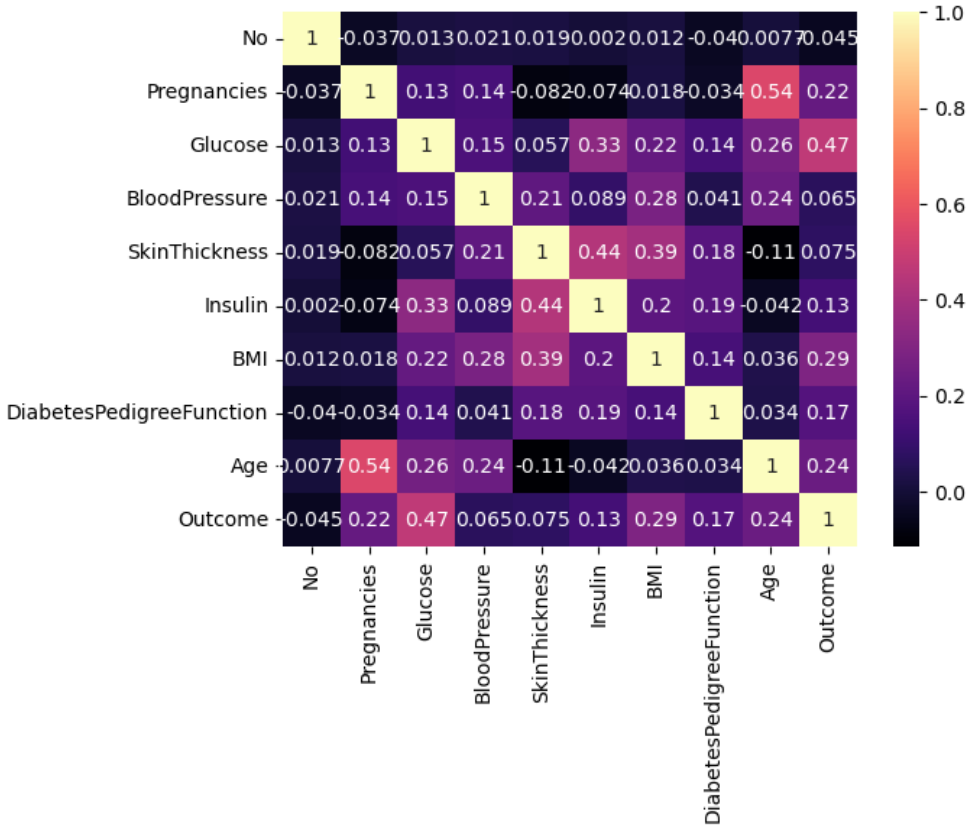

Figure 31: Correlation Matrix with Diabetes Dataset

We can see clearly numbers with **df.corr()** code as following output:

Out[27]:

	No	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
No	1.000000	-0.037201	0.012994	0.020953	0.019006	0.001998	0.012320	-0.040326	0.007714	-0.045184
Pregnancies	-0.037201	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.012994	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.020953	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	0.019006	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	0.001998	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.012320	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
betesPedigreeFunction	-0.040326	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.007714	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	-0.045184	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Figure 32: Correlations of All Datasets

From the figure 32 we see that the values of "BMI" increase, the "Outcome" variable also increases. It means the correlation is bigger than 0, there is a positive correlation 0.292695.

Regarding the diabetes dataset and linear regression we can choose **age, BMI, blood pressure**, and others (we chose BMI) which represents the measure of disease progression one year after baseline as the independent variable to predict the dependent variable. The dataset labels this dependent variable as the **"target."** The term **"outcome"** pertains to the target variable. So, in conclusion:
Independent Variable: BMI (Body Mass Index)
Dependent Variable: Diabetes Outcome (target variable)

The aim of linear regression is to determine the relationship between BMI and outcome of diabetes so that we can predict the outcome of diabetes based on BMI values.

Finally we can find the coefficients (β_0 and β_1) for analyzing the straight line. The code is following:

$$x = df['BMI']$$
$$x\_bar = (df['BMI']).mean()$$
$$y = df['Outcome']$$
$$y\_bar = (df['Outcome']).mean()$$
$$beta1 = ((x - x\_bar) * (y - y\_bar)).sum()/((x - x\_bar) **2).sum()$$
$$print("Beta\_1 \quad coefficientestimate : " + str(round(beta1,4)))$$

From the code the output is **"Beta_1 coefficient estimate : 0.0177."**

$$beta0 = y\_bar - beta1 * x\_bar$$
$$print("Beta_0 coefficientestimate : " + str(round(beta0,4)))$$

From the code the output is **"Beta_0 coefficient estimate : -0.2175."**

The following code gave us the linear line between relationship of "MBI" and "Outcome".

$$pltt.figure(figsize = (7,5))$$

$$pltt.scatter(df['BMI'], df['Outcome'])$$

$$pltt.plot(df['BMI'], beta1 * df['BMI'] + beta0, c =' r')$$

$$pltt.xlabel("BMI)", fontsize = 12)$$

$$pltt.ylabel("Outcome", fontsize = 12)$$

$$pltt.show()$$

So, I can show two types of code that to analyze the line which is good or not good between relationship of "BMI" and "Outcome".

$$pltt.figure(figsize = (7,5))$$

$$pltt.scatter(df['BMI'], df['Outcome'])$$

$$pltt.plot(df['BMI'], beta1 * df['BMI'] + beta0, c =' r')$$

$$pltt.xlabel("BMI)", fontsize = 12)$$

$$pltt.ylabel("Outcome", fontsize = 12)$$

$$pltt.show()$$

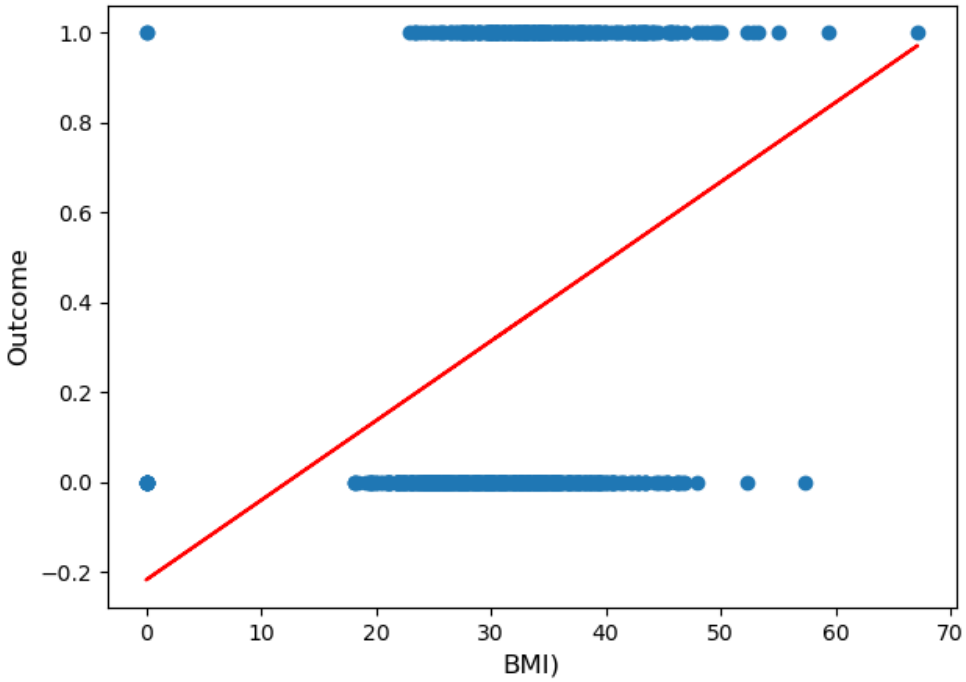

Figure 33: The Line Between BMI and Outcome

From the figure 33 there is no distance between observations. So the code is:

$$predictors = ['BMI']$$

$$outcome =' Outcome'$$

$$model = LinearRegression()$$

$$model.fit(df[predictors], df[outcome])$$

$$print(f'Intercept : model.intercept_.3f')$$

$$print(f'CoefficientExposure : model.coef_[0] : .3f')$$

$$fitted = model.predict(df[predictors])$$

$$residuals = df[outcome] - fitted$$

$$ax = df.plot.scatter(x =' BMI', y =' Outcome', figsize = (15, 10))$$

$$ax.plot(df.BMI, fitted, linewidth = 5, color = 'k',$$

$$label = f'simple \quad linear \quad regression:$$

$$Outcome = model.intercept_{:}.3f + model.coef_{-}[0] : .3fBMI')$$

$$for \quad x, yactual, yfitted \quad in \quad zip(df.BMI, df.Outcome, fitted):$$

$$ax.plot((x,x),(yactual,yfitted),$$

$$'--', color = 'C1')$$

$$pltt.tight_layout()$$

$$pltt.legend()$$

$$pltt.show()$$

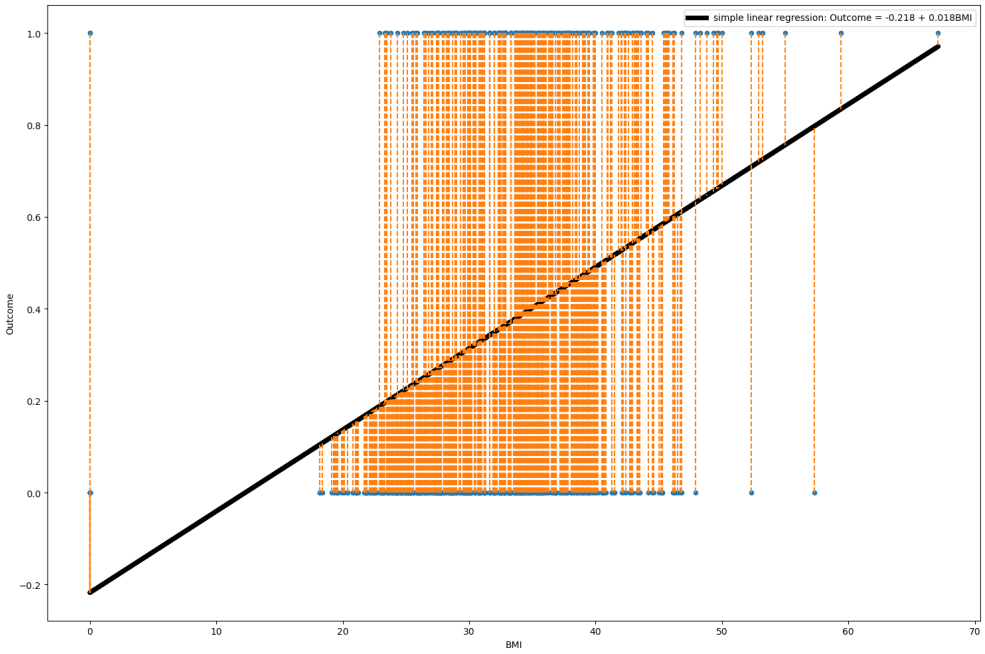

Figure 34: Linear Regression of Outcome and BMI

We know that the intercept overall is -0.218 and slope overall is 0.018, respectively. The orange vertical dashed line represents the residuals, which represent the discrepancies between the predictions made by the regression line and the actual

observed values. From the figure 34 we know that o the line between "BMI" and "Outcome" relationship is good. Because it's not near to 0.

5.2 Diabetes Parameter Prediction with R Language

We use the ggplot2 package to examine the "Diabetes" dataset. We put the excel file in R studio called Diabetes.

```
> head(Diabetes)
# A tibble: 6 × 10
     No Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
  <dbl>       <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
1     1           6     148            72            35       0  33.6
2     2           1      85            66            29       0  26.6
3     3           8     183            64             0       0  23.3
4     4           1      89            66            23      94  28.1
5     5           0     137            40            35     168  43.1
6     6           5     116            74             0       0  25.6
# i 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <dbl>
```

Figure 35: Diabetes Dataset in R

Now we need to find the coefficients β_0 and β_1 to analyze the linear line is good or not by using R language code.

```
> x <- Diabetes$BMI
> y <- Diabetes$Outcome
> fit <- lm(y~x)
> fit

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -0.21752      0.01771
```

Figure 36: Coefficients

We got β_0 is -0.21752 and β_1 is 0.01771 in the figure 36. If x is 0, it means the intercept is -0.21752 with 0 (no diabetes). But it's not understandable, so I can calculate the average of our x. The following following code is:

The output shows that the slope is the same but the intercept is now 0.34896. So our

```
> x <- Diabetes$BMI - mean(Diabetes$BMI)
> y <- Diabetes$Outcome
> lm(y~x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    0.34896      0.01771
```

Figure 37: The Mean of BMI

equation is y = 0.34896 + 7756x, it means that for each increase in x, y is also increased by that value.

On the other hand we can use *summary()* code to check the linear line has a good fit or not by looking the p-value, multiple R-squared and adjusted R-squared.

```
> summary(fit)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7971 -0.3579 -0.2278  0.5451  1.2175

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.21752    0.06886  -3.159  0.00165 **
x            0.01771    0.00209   8.472  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4564 on 766 degrees of freedom
Multiple R-squared:  0.08567,   Adjusted R-squared:  0.08448
F-statistic: 71.77 on 1 and 766 DF,  p-value: < 2.2e-16
```

Figure 38: Summary of BMI and Outcome

The output in 38 shows that the *p-value* are highly significant, so we can reject the null hypothesis which means that it's not equal to zero.

The *R-squared* and *Adjusted R-squared* numbers are 0.08567 and 0.08448 which close

to 1. Commonly, it implies the following that there is a strong linear relationship and the model extremely well fits the observed data values and the model has strong predictive power.

Now we can analyze the linear line if it is good or not between relationship of "BMI" and "Outcome". The following code is:

$$
\begin{aligned}
&Diabetes\%>\% \\
&+select(BMI, Outcome)\%>\% \\
&+ggplot(aes(BMI, Outcome))+ \\
&+geom\_point(colour='pink', size=2, alpha=0.5)+ \\
&+geom\_smooth(method='lm', colour='green')+ \\
&+labs(x='BMI', y='OutcomeinDiabetes', \\
&title='ScatterPlotofDiabetesBMIandOutcomewithRegressionLine')+ \\
&+ylim(0,1)+ \\
&+theme\_minimal()
\end{aligned}
$$

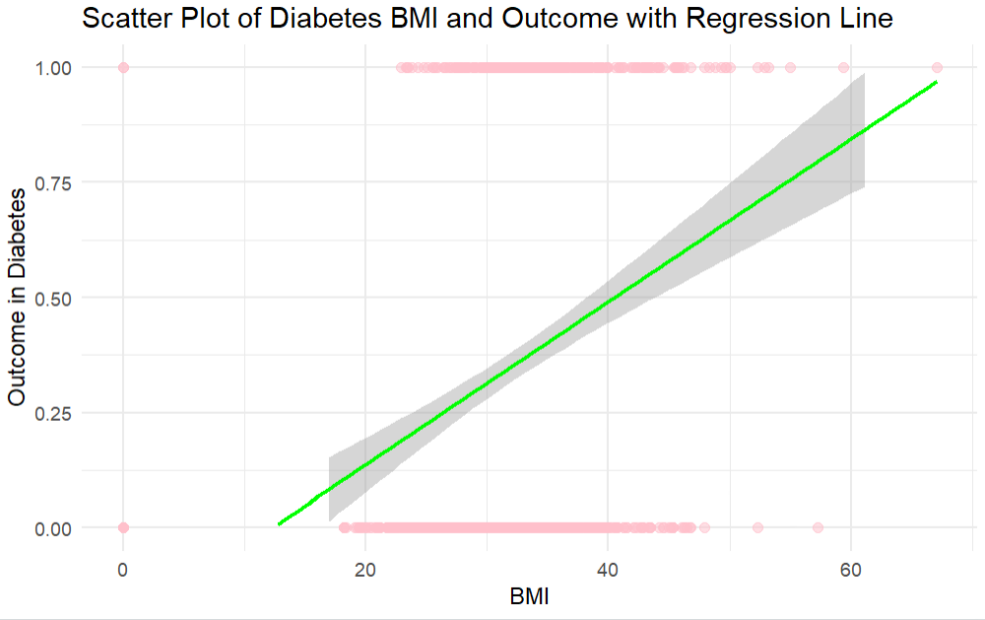

Figure 39: Scatter Plot of BMI and Outcome

From the picture 39 we easily say that the line is best fit because of the residuals.

In conclusion, there are some syntax variations between R and Python [12] when fitting a linear regression model. So, I choosed both of them. However, these differences have minimal impact on the output since the results were nearly same for our both datasets.

REFERENCES

[1] Elizabeth A. Peck Douglas C. and G. Geoffrey Vining. *Introduction to Linear Regression Analysis.* Montgomery, 2012.

[2] Johannes Ledolter Bovas Abraham. *Introduction To Regression Modeling*. Cengage Learning, 2005.

[3] Terry E. Dielman. *Applied Regression Analysis*. Brooks/Cole, 2004.

[4] Jogesh Dhiman. Data Analysis using R and Python. Master's thesis, Central University Of Haryana, Mahendergarh, Haryana, 2018.

[5] Greta M. Linse David J. Lilja. *Linear Regression Using R*. University of Minnesota Libraries Publishing Minneapolis, Minnesota, USA, 2022.

[6] Peter Gedeck Peter Bruce, Andrew Bruce. *Practical Statistics for Data Scientists*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017.

[7] L. Massaron and A. Boschetti. *Regression analysis with python: Learn the art of regression analysis with python.* Birmingham: Packt Publishing., 2016.

[8] Chakraborty R. and Hasija Y. Python for data analysis. hands-on data science for biologists using python. page 71–90, 2021.

[9] Elankovan A.Sundararajan3 Mahathir Rahmany1, Abdullah Mohd Zin2. Comparing tools provided by python and r for exploratory data analysis.

International Journal Information System and Computer Science (IJISCS), 4 issue 3:131–142, oct 2020.

[10] C. Borjigin. Data analysis with python. python data science. 4 issue 3:295–342, 2023.

[11] amp; Loughin T. Bilder, C. *Analysis of categorical data with R.* CRC Press, Taylor amp; Francis Group., 2015.

[12] amp; Costa V. Sarmento, R. *Comparative approaches to using R and Python for Statistical Data Analysis*. Information Science Reference., 2017.