Practical Business Analytics

EARTHQUAKE DETECTION

Group name	Members
Team Metaverse	Harpreet Singh
	Sohom Das
	Himanshu Saxena
	Abdelkader Bouregag
	Taukeer Ahmad

TABLE OF CONTENTS

1. Introduction:

Earthquakes are frequent phenomena that happen in different parts of the world resulting from a sudden movement along faults within the Earth. This disaster has caused humanity a lot of losses every year, the 2011 disastrous 9.0 earthquake as an example hit Japan's east coast. Having the historical data of earthquakes is a chance to understand this phenomenon and its factors.

In this report, we will investigate the geographical factors that led to massive earthquakes and what is the correlation between these factors that are causing high magnitude earthquakes in some particular regions of the world, regions that are ranked as highly earthquake-prone areas. Determining this will help us understand the factors contributing to higher earthquakes mankind is facing which cause lives and millions of dollars.

2. Data description:

The USGS dataset we are using is a real-time earthquakes dataset. It contains 14995 rows and 21 attributes:

Source: https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php

Data Dictionary:

time	The time when the event occurred. Times are reported in milliseconds since the epoch
latitude	Decimal degrees latitude. Negative values for southern latitudes.
longitude	Decimal degrees longitude. Negative values for western longitudes
depth	Depth of the event in kilometers.
mag	Magnitude of the event occured.
magType	The method or algorithm used to calculate the preferred magnitude
nst	The total number of seismic stations used to determine earthquake location.

gap	The largest azimuthal gap between azimuthally adjacent stations (in degrees).
dmin	Horizontal distance from the epicenter to the nearest station (in degrees).
rms	The root-mean-square (RMS) travel time residual, in sec, using all weights.
net	The ID of a data source contributor for the event occurred.
type	Type of seismic event.
place	named geographic region near to the event.

3. Problem definition:

The purpose of this project is to analyze the dataset and understand which are the most factors affecting the higher earthquakes in different regions around the globe. We will be using the CRISP-DM methodology (Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment.) to apply unsupervised algorithms to identify the regions with the highest trend of earthquakes which are the hotspots of earthquakes around the world. Next, we apply prediction models to predict the magnitude of earthquakes in these hotspots. The project focuses on the following three objectives:

- Identify the hotspots with the highest density of earthquakes around the world and provide a clustering of these earthquakes.

- Identify what factors are causing earthquakes with higher magnitude. We will take earthquakes with higher magnitude

- Predict the future values of magnitude of the future earthquakes in the places or cities where the likelihood for earthquakes is more and provide them in graphical representation.

- Test the hypothesis of earthquakes, high magnitudes and CO2 emissions.

4. Data exploration:

As a part of Exploratory data analysis, the data is analyzed based on three parameters (Depth, Latitude, and Longitude) that contribute to the occurrence of the Earthquake and affect the magnitude of the Earthquakes.

1. Country

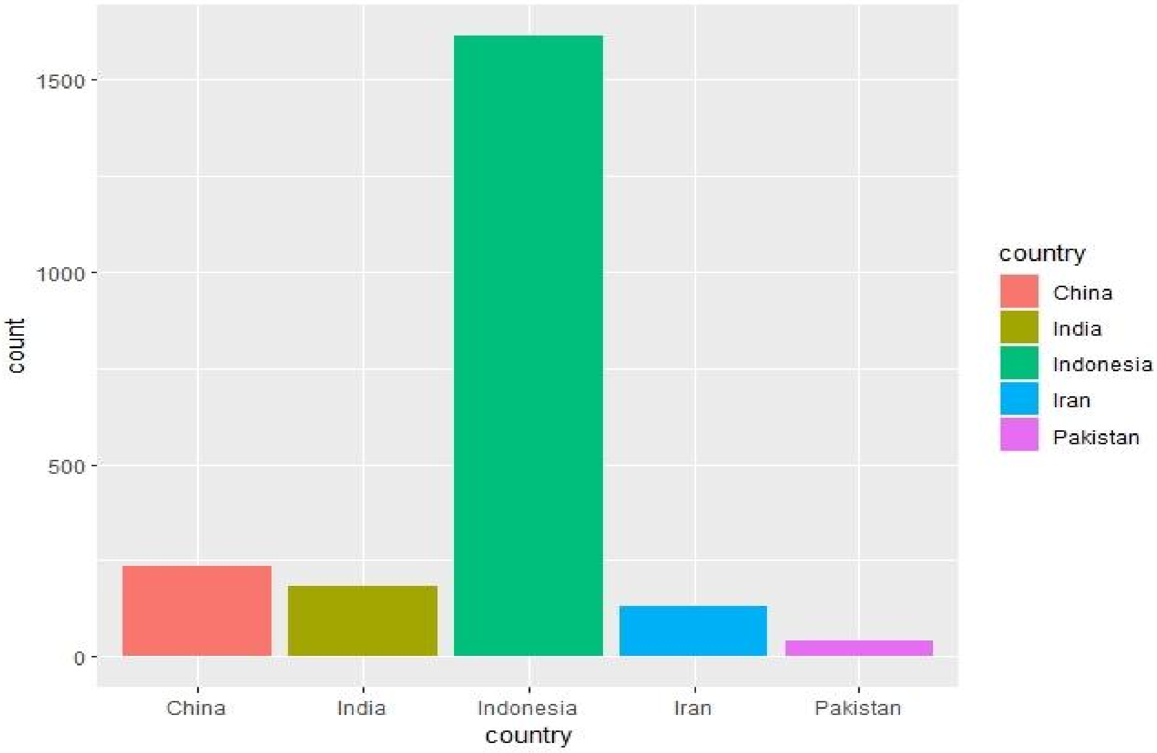


Fig. 1

The original dataset consists of the data of earthquakes for 155 countries. For computing the Linear model we are choosing 5 countries from the dataset to work upon. The countries China and India are taken for observation because the emission rate of CO2 is the highest for both of these countries. Indonesia is taken because it has the highest earthquake rate and the raiding two other countries are taken randomly.

2. Magnitude

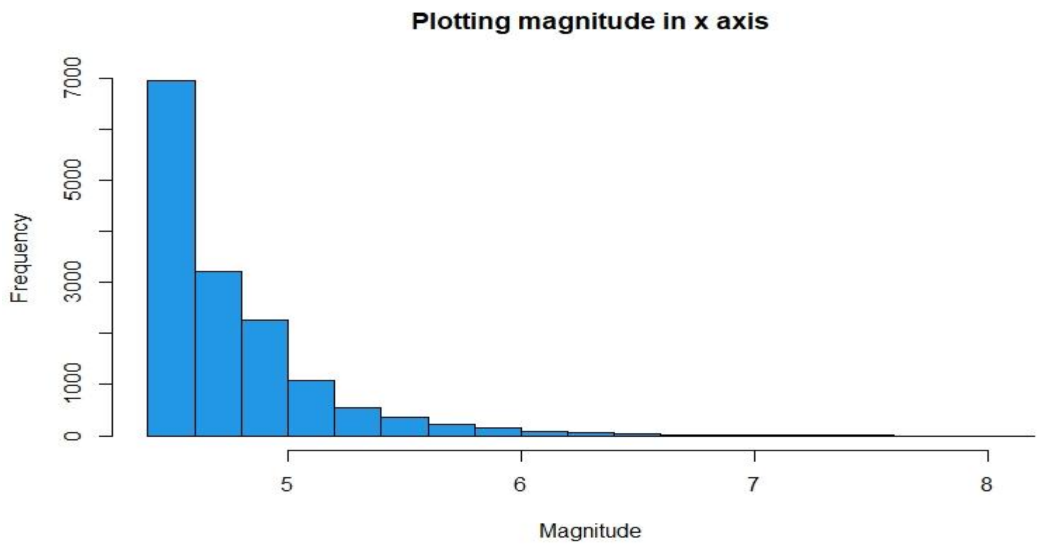


Fig. 2

Here the magnitude attribute is being mapped. It is observed that the majority of the earthquakes had a magnitude below 5.

3. Depth

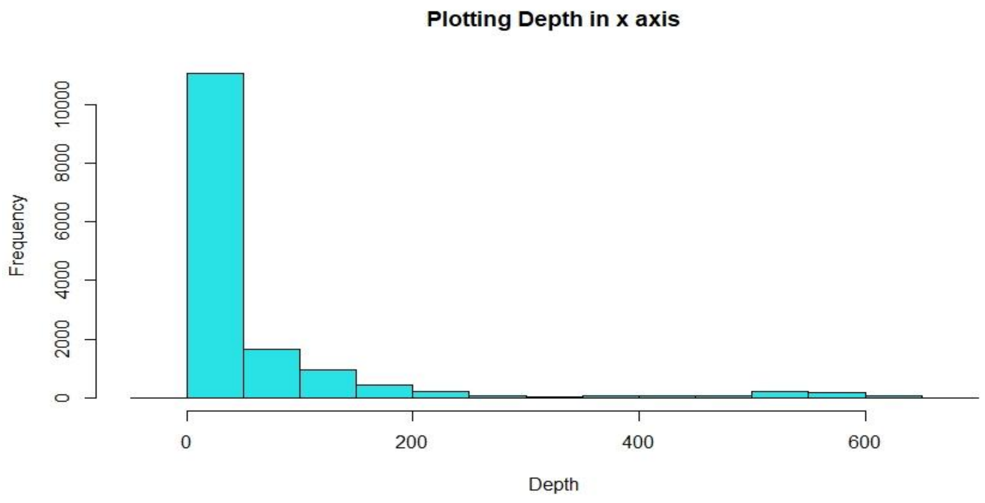


Fig.3

From the graph of depth, we get that most of the earthquake has occurred in the range of 0-50. Almost 90% of the earthquake from the entire Dataset has originated in the depth of 0-200. At the end of the graph, it can be seen that a chunk of earthquake data is plotted in the range of 500-600.

4. Depth VS Magnitude

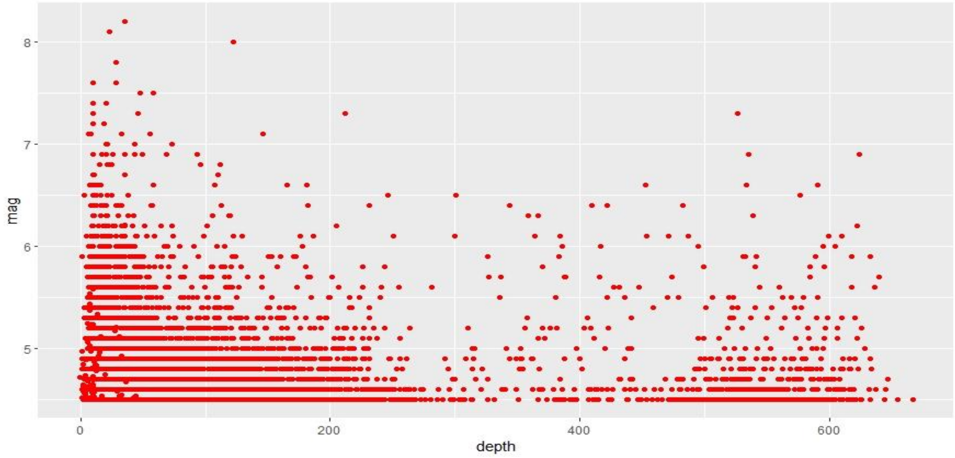
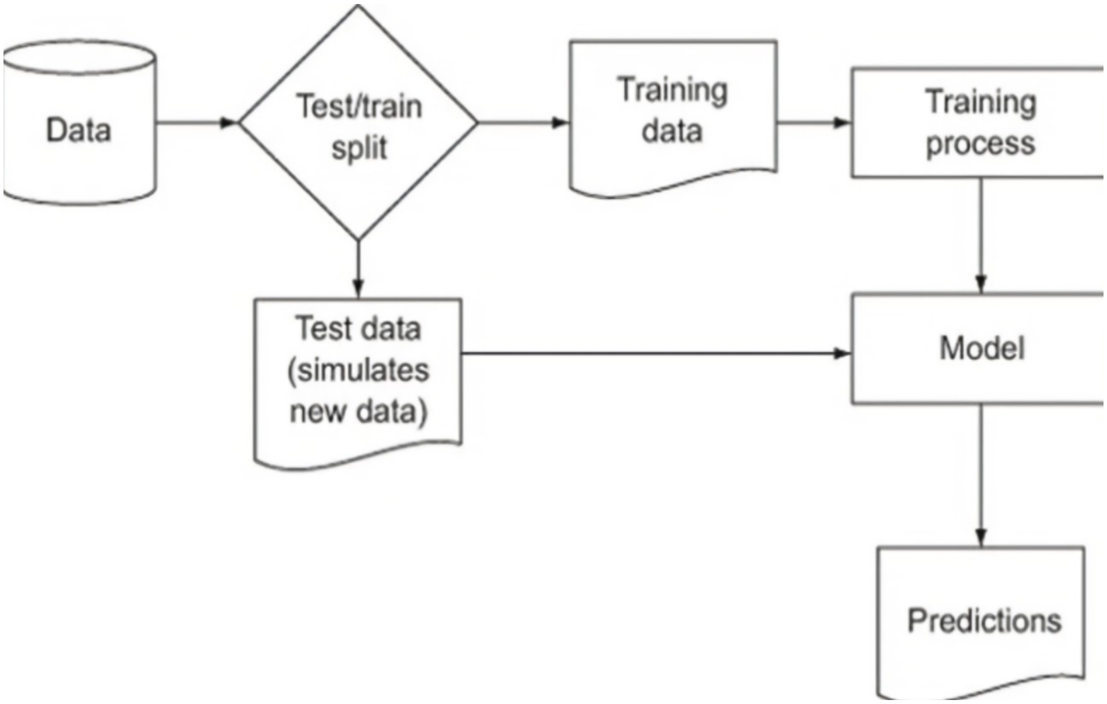


Fig. 4

A scatter plot of the Depth VS Magnitude graph is plotted. Analyzing the graph it is seen that the majority of the earthquake with magnitude ranging from 0-6 has occurred in the depth of 0-200 and a small part of the earthquake of the specified magnitude has occurred in the region of 450-600. It is observed the earthquake with the highest magnitude in the dataset has occurred in the region of 0-100 depth.

5. Data preparation:

In a the life-cycle of a data science project, we usually pass by the following steps explained in this chat:



Data Cleaning of the Earthquake_Final_dataset:

We have used Library(Janitor)
First we clean names, if it carries any special characters by using a function called cleaned names

```
Data_cleaned<-Clean_names(dataset_name)
```

Then we remove the empty using a function called remove_empty as stated below

```
Data_cleaned<-remove_empty(dataset_name, which =c(“rows”,”cols”),quiet=FALSE)
library(dplyr)
```

Removing the special character from specific column by suing a function called gsub:

```
data_cleaned$place <- gsub("[[:punct:]]", "", data_cleaned$place)
data_cleaned %>% filter(data_cleaned$location_source!='us')->notus
View(notus)
```

Removing the specific column:

```
data_cleaned$location_source<-NULL
View(data_cleaned)
```

Exporting the cleaned data set using write.csv function:

```
write.csv(data_cleaned, file = "Earthquake_Data_Cleaned_01.csv")
```

Once the data has been cleaned we need to split Data into Training and Testing data
The previous module introduced the idea of dividing the data set into two subsets:

- **training set**—a subset to train a model.
- **test set**—a subset to test the trained model.

Slicing a single data set into a training set and test set.

Making sure that the test set meets the following two conditions:

- Is large enough to yield statistically meaningful results.
- Is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.

Assuming that our test set meets the preceding two conditions, our goal is to create a model that generalizes well to new data. Our test set serves as a proxy for new data. For example, consider the following figure. Notice that the model learned for the training data is very simple. This model doesn't do a perfect job—a few predictions are wrong. However, this model does about as well on the test data as it does on the training data. In other words, this simple model does not overfit the training data.

We are using the library (CaTools) in which we have used the function sample.split to divide the data into different subset.

```
library(caTools)
 sample.split(Earthquake$mag , SplitRatio = 0.70)->Split_Tag
 subset(Earthquake , Split_Tag==T)->Training_Data
 subset(India_Data , Split_Tag==F)->Testing_Data
```

6. Model selection:

In this section, we will apply the different supervised and unsupervised learning methods in order to better understand our dataset and predict earthquakes magnitude. Starting first with supervised learning:

6.1. Linear Model:

6.1.1. Linear Model Implemented for Earthquake in R

Linear Model : A statistical or mathematical model that is used to formulate a relationship between a dependent variable and single or multiple independent variables called as, linear model in R. The criteria is that the variables involved in the formation of model meet certain assumptions as necessary prerequisites prior model building and that the model has certain important elements as its parts, which are formula, data, subset, weights, method, model, offset etc. It is not necessary that all have to be used every time, but only those that are sufficient and essential in the given context.

6.1.2. Why we choose Linear Model :

As Per our Graphs and Dataset we see some Linearity between Data as shown in Graphs below

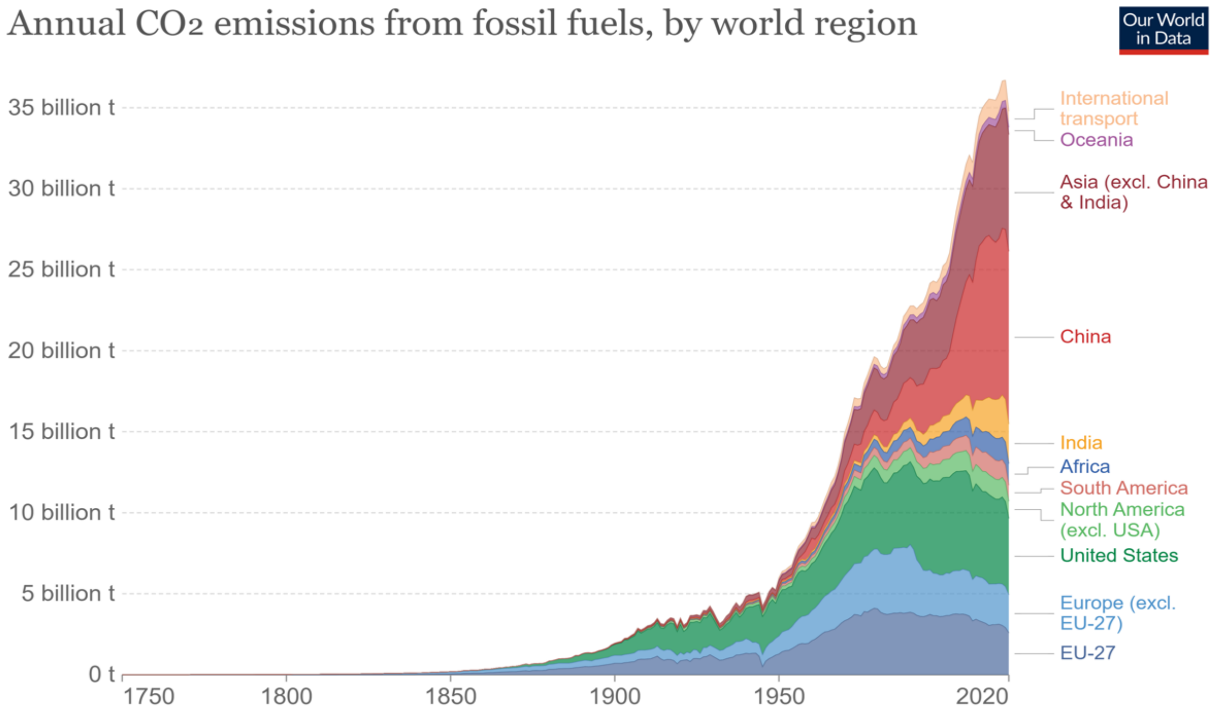


Fig 5

This chart shows the division of global CO2 emissions country wise.

We can see in the year 1700 to 1800 , There is very little or no Global Co2 emission . In 1900, 90 percent + of emissions were produced in Asian countries each year. But recently this has changed.
After the first half of the Twenty(20th) century we see a significant rise of CO2 emission in various parts of the world , Mainly Asia, and mostly China and India.

6.1.3. Graphical representation of C02 vs Countries

6.1.3.1. China:

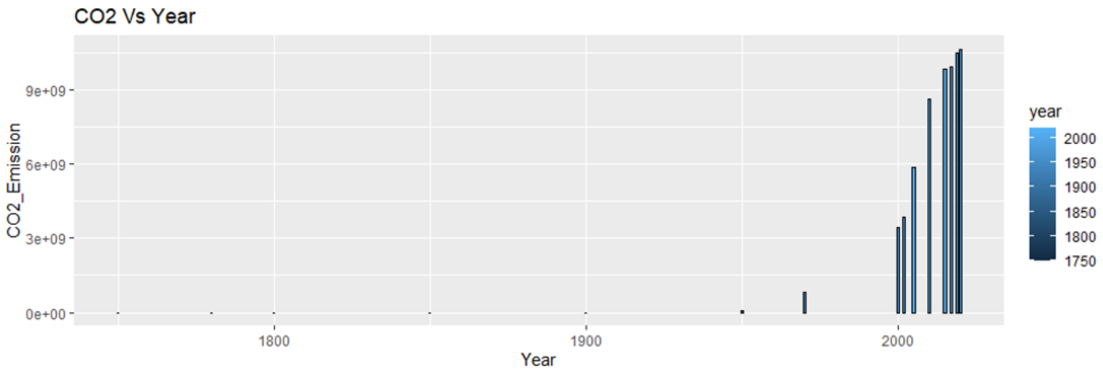


Fig 6

6.1.3.2. India:

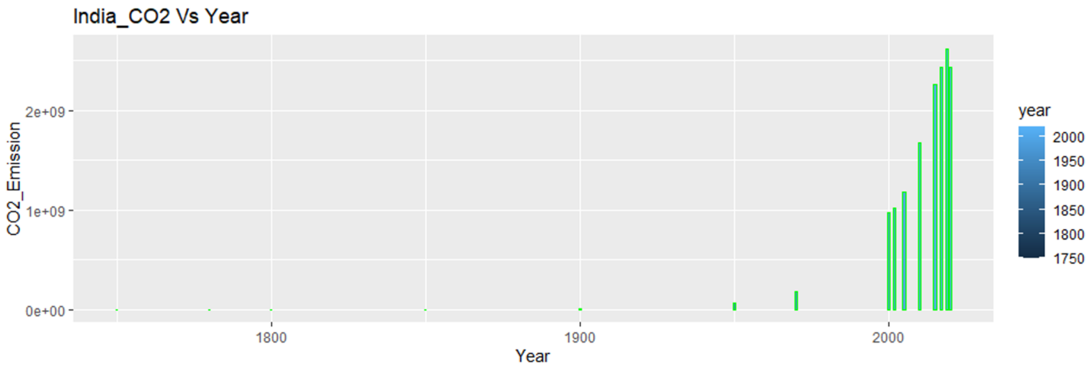


Fig 7

6.1.3.3. Pakistan:

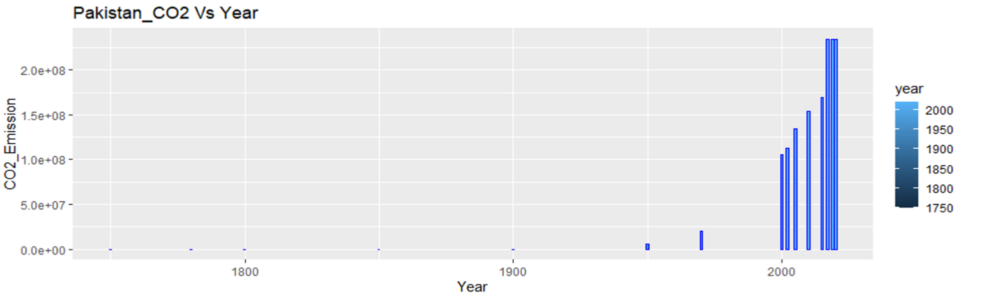


Fig 8

6.1.3.4. Iran:

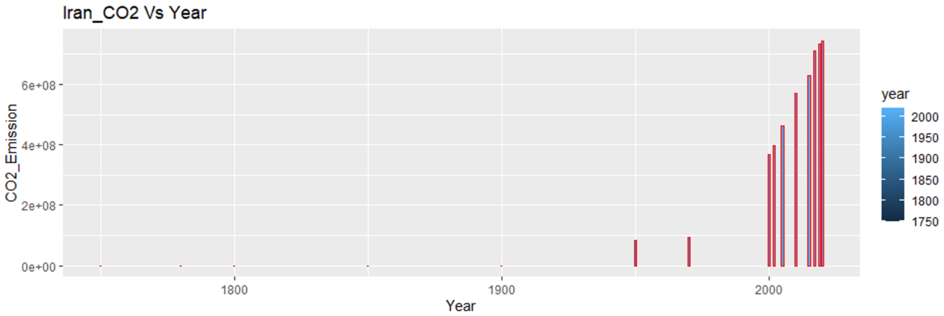


Fig 9

6.1.3.5. Indonesia:

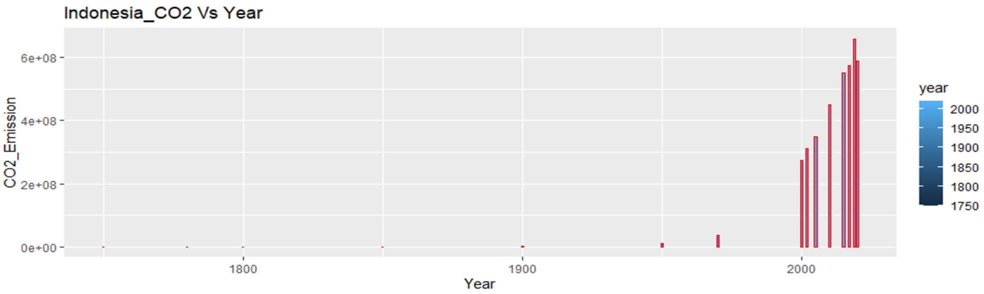


Fig 10

6.1.3.6. Combine Data of Co2 of Selected Countries:

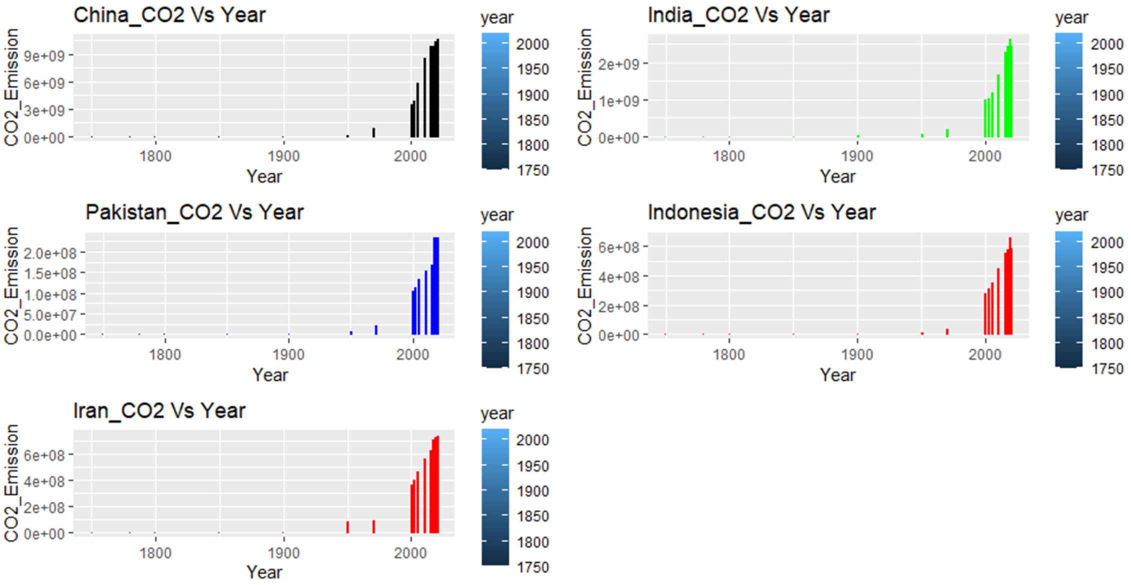


Fig 11

6.1.3.7. Comparison B/w Co2 and Magnitude of selected Countries

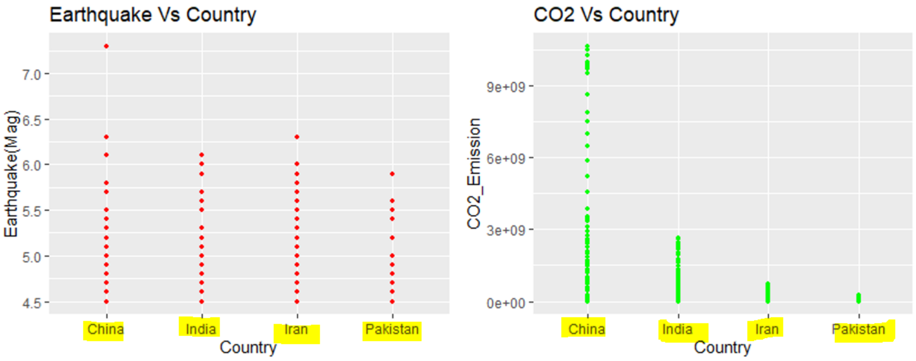


Fig 12

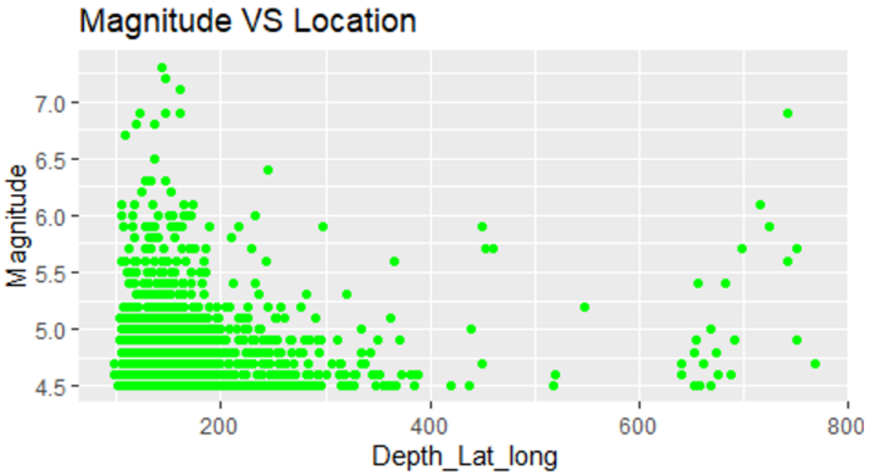


Fig 13

In the above Fig 8 we can see that as our Magnitude is tend to increase as our Depth , Longitude and Latitude Increases

Here we have done Visualization using GGPLOT 2 Library ,We have randomly picked 5 countries(China,India,Pakistan,Indonesia and Iran) showing Co2 emission in below graph (Fig 9)

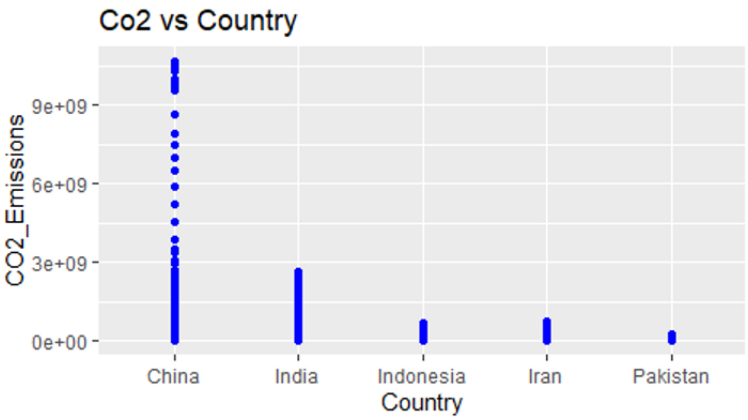


Fig 14

6.1.4. The 3DScatter View of Countries:

he 3DScatter View of Countries which we have selected ,Shown in Fig 10(a) and Fig 10(b)

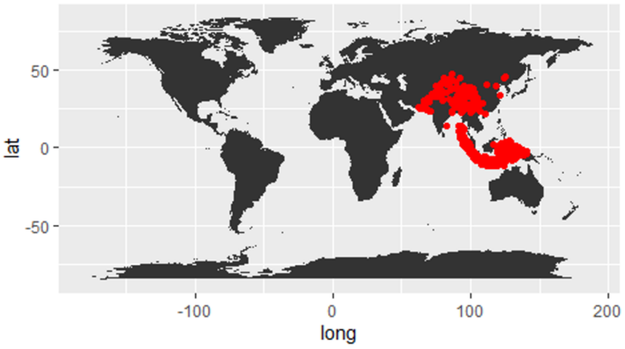


Fig 15(a)

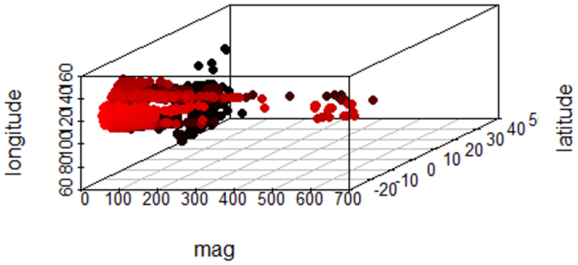


Fig 15(b)

6.1.5. Hypothesis:

About Null Hypothesis and How model is calculating the values in our multiple Linear Regression Model:

- T Stats and PValue

To Calculate the T Stats and PValue we must have coefficient and SE(Standard error) to be known and the formula or equation for calculating T Statistic and P Value is stated below:

$$t - Statistic = \frac{\beta - coefficient}{Std. Error}$$

- RS(RSquared) and Adjacent R Squared

R Squared is mainly used to determine the proportion of variation in the dependent (response) variable that has been explained in this model. Formula for calculation the R Squared value ais attested below.

$$R^2 = 1 - \frac{SSE}{SST}$$

- **Adjusted R-Squared:**

In Adjusted R Squared if we add more variable in x then, the R-Squared value of the new bigger subset will always be greater than that of the smaller subset. This is because, since all the variables in the main subset are present, Therefore, when comparing nested subsets, it is a good practice to look at Adj R squared value over R-squared. The formula/equation is stated below.

$$R^2_{adj} = 1 - \left(\frac{\left(1 - R^2\right)(n-1)}{n-q} \right)$$

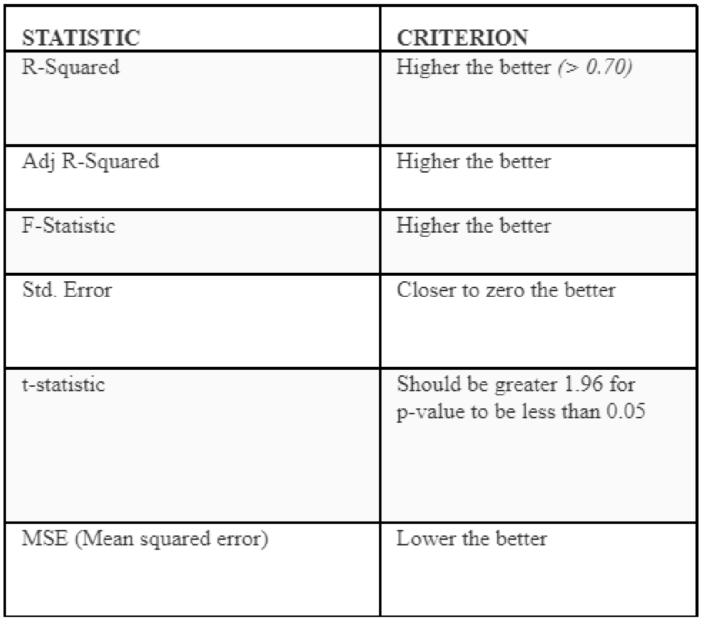
- **Standard Error Values and F Statistic Values :**

Both standard errors values and F statistic values are measures of goodness of fit or best fit value anbd formula for calculation the same has been satiated below.

$$Std.\,Error = \sqrt{MSE} = \sqrt{\frac{SSE}{n-q}}$$

$$F - statistic = \frac{MSR}{MSE}$$

- **Criteria we choose to predict the best values of the stats:**



STATISTIC	CRITERION
R-Squared	Higher the better *(> 0.70)*
Adj R-Squared	Higher the better
F-Statistic	Higher the better
Std. Error	Closer to zero the better
t-statistic	Should be greater 1.96 for p-value to be less than 0.05
MSE (Mean squared error)	Lower the better

Country	**Residual-Standard error**	**Multiple R-squared**	**F-Statistic**	**p-value**	**Adjusted R-squared**
INDIA	0.2956 on 125 degrees of freedom	0.0427	1.858 on 3 and 125 DF	0.1401	0.01972
CHINA	0.3699 on 162 degrees of freedom	0.01692	0.9293 on 3 and 162 DF	0.428	-0.001287
INDONESIA	0.3687 on 1060 degrees of freedom	0.008785	3.132 on 3 and 1060	0.02489	0.00598
PAKISTAN	0.3504 on 25 degrees of freedom	0.2208	2.362 on 3 and 25 DF	0.0954	0.1273
IRAN	0.383 on 84 degrees of freedom	0.01378	0.3913 on 3 and 84 DF	0.7596	-0.02144

Table 1

- **Null Hypothesis Rejection and Support Criteria**

To check if there is a jointly valid relationship between the two predictor variables (Latitude, Longitude and Depth) and the response variable (Mag), we need to analyse the overall F value of the model and the corresponding p-value:
For Example, for various Countries we have implemented the Linear model as stated below in the table

As we can see for Country India F Value and P-value are:

a. F-Value: 1.858
b. P-value: 0.0425

Since this p value is less than .05, we can reject the null hypothesis. In other words, depth, Longitude and Latitude taken have a jointly statistically significant relationship with earthquake(magnitude).

6.1.6. RMSE Values for different countries:

The root mean squared error (**RMSE**) is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Where,

n = Total number of sample data
yj = predictive value for the j^{th}
y^j= observed value for j^th

RMSE is the good way to check the SD(Standard Deviation) of the observed values from our of our predicted data set.

We will be using library available in R tool to calculate mean squared error, later we can simply use lm function to calculate the linear model.

We will be using SQRT,MEAN functions to generate actual and predication array.

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

Also Here We used Multiple Regression taking Depth , Longitude and Latitude as Independent Variable and Magnitude(Mag) as independent Variable

Multiple Linear Regression

Simple linear regression can be further extended to include multiple regression features. This is called multiple linear regression:

y=C+β1x1+...+βn
Each x represents a different values of the features that will calculated by our model(LM()), and each feature has its own value of
coefficient. In this case:
y=C+β1×Depth+β2×Latitude+β3×Longitude

6.1.7. Model Evaluation Metrics for Regression

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Squared Error (MSE) is the mean of the squared errors:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

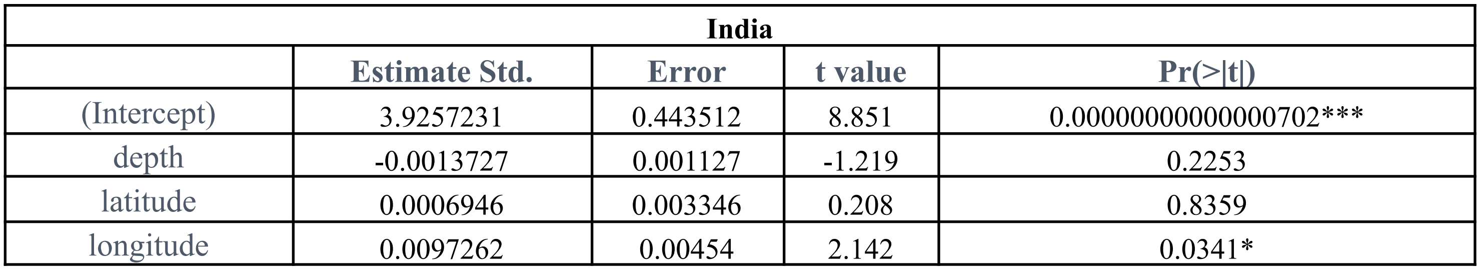
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Here we have used RMSE value to find the best fit or the accuracy of the model

Evaluation for Selected Countries:

1. RMSE India

```
Coefficients:
             Estimate Std.        Error        t value    Pr(>|t|)
(Intercept)  3.9257231  0.4435119   8.851 7.02e-15 ***
depth       -0.0013727  0.0011265  -1.219   0.2253
latitude     0.0006946  0.0033455   0.208   0.8359
longitude    0.0097262  0.0045398   2.142   0.0341 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2956 on 125 degrees of freedom
Multiple R-squared:  0.0427,  Adjusted R-squared:  0.01972
F-statistic: 1.858 on 3 and 125 DF,  p-value: 0.1401
```

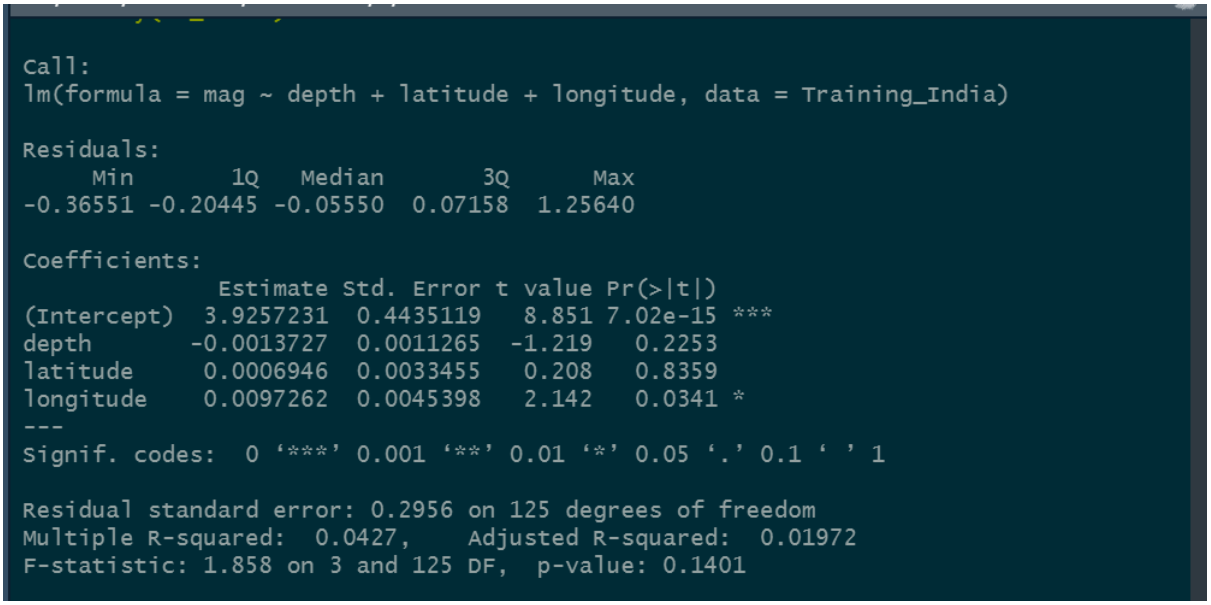


India				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	3.9257231	0.443512	8.851	0.00000000000000702***
depth	-0.0013727	0.001127	-1.219	0.2253
latitude	0.0006946	0.003346	0.208	0.8359
longitude	0.0097262	0.00454	2.142	0.0341*

Table 2

Here we have only a longitude Predictor(Pr) value less than 0.05 so we can say it supports Null Hypothesis(H0). So here we cant reject Null Hypothesis .

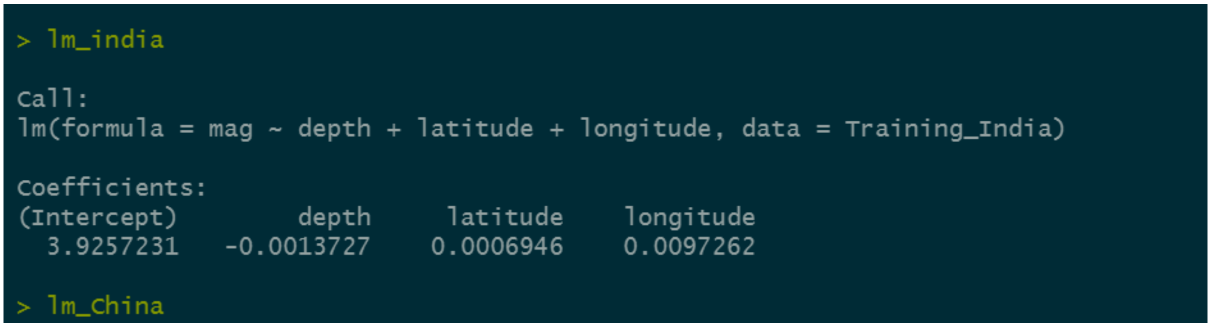
Coefficients:



```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_India)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.36551 -0.20445 -0.05550  0.07158  1.25640 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.9257231  0.4435119   8.851 7.02e-15 ***
depth       -0.0013727  0.0011265  -1.219   0.2253    
latitude     0.0006946  0.0033455   0.208   0.8359    
longitude    0.0097262  0.0045398   2.142   0.0341 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2956 on 125 degrees of freedom
Multiple R-squared:  0.0427,	Adjusted R-squared:  0.01972 
F-statistic: 1.858 on 3 and 125 DF,  p-value: 0.1401
```

Fig 16

Linear Model for India



```
> lm_india

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_India)

Coefficients:
(Intercept)        depth     latitude    longitude  
  3.9257231   -0.0013727    0.0006946    0.0097262  

> lm_China
```

Fig 17

```
> lm_india

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_India)

Coefficients:
(Intercept)       depth     latitude    longitude
  3.9257231   -0.0013727    0.0006946    0.0097262
```

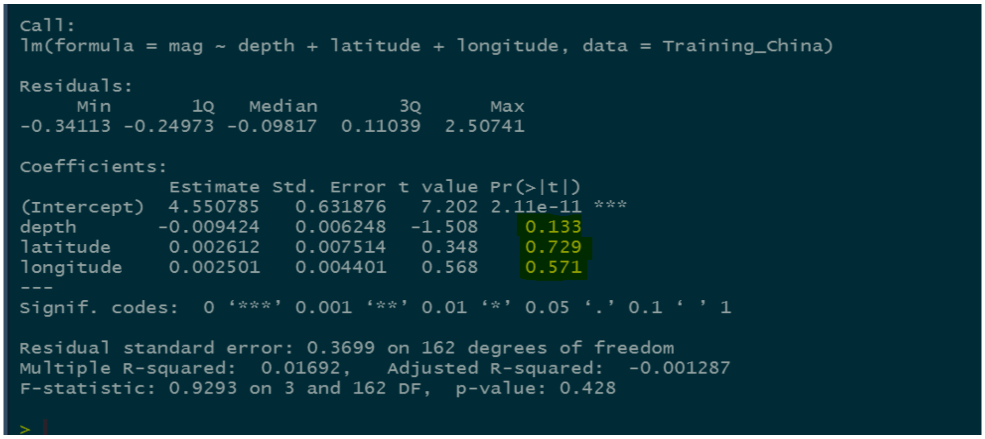
2. RMSE China

```
lm(formula = mag ~ depth + latitude + longitude, data = Training_China)

Residuals:
    Min       1Q   Median       3Q      Max
-0.34113 -0.24973 -0.09817  0.11039  2.50741

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.550785   0.631876   7.202 2.11e-11 ***
depth       -0.009424   0.006248  -1.508    0.133
latitude     0.002612   0.007514   0.348    0.729
longitude    0.002501   0.004401   0.568    0.571
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3699 on 162 degrees of freedom
Multiple R-squared:  0.01692,	Adjusted R-squared:  -0.001287
F-statistic: 0.9293 on 3 and 162 DF,  p-value: 0.428
```

China				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	4.550785	0.631876	7.202	0.0000000000211***
depth	-0.009424	0.006248	-1.508	0.133
latitude	0.002612	0.007514	0.348	0.729
longitude	0.002501	0.004401	0.568	0.571

Table 3

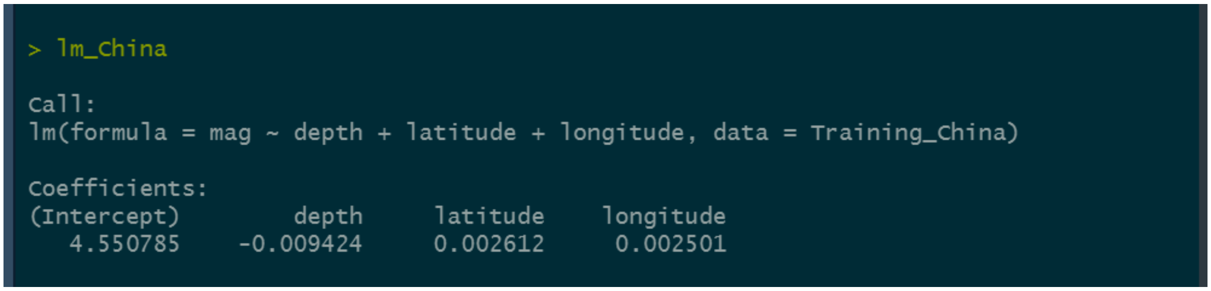
Here we have P value value more than 0.05 and F-statistic<1 so we can say it support Null Hypothesis(H0)



```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_China)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.34113 -0.24973 -0.09817  0.11039  2.50741 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.550785   0.631876   7.202 2.11e-11 ***
depth       -0.009424   0.006248  -1.508    0.133    
latitude     0.002612   0.007514   0.348    0.729    
longitude    0.002501   0.004401   0.568    0.571    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3699 on 162 degrees of freedom
Multiple R-squared:  0.01692,	Adjusted R-squared:  -0.001287 
F-statistic: 0.9293 on 3 and 162 DF,  p-value: 0.428

> 
```

Fig 18

Linear Model for China



```
> lm_China

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_China)

Coefficients:
(Intercept)        depth     latitude    longitude
   4.550785    -0.009424     0.002612     0.002501
```

Fig 19

```
lm(formula = mag ~ depth + latitude + longitude, data = Training_China)

Coefficients:
(Intercept)       depth     latitude    longitude
4.550785    -0.009424     0.002612     0.002501
```

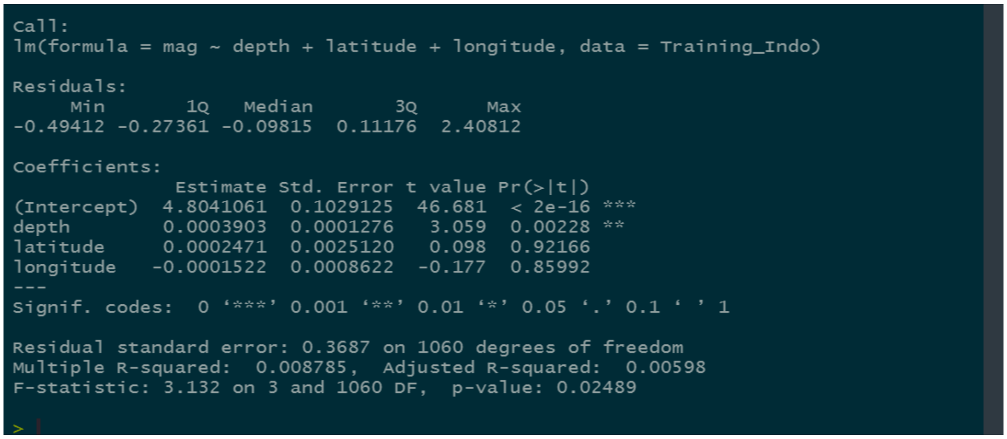
3. RMSE Indonesia

```
lm(formula = mag ~ depth + latitude + longitude, data = Training_Indo)

Residuals:
     Min       1Q   Median       3Q      Max
-0.49412 -0.27361 -0.09815  0.11176  2.40812

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.8041061  0.1029125  46.681  < 2e-16 ***
depth        0.0003903  0.0001276   3.059  0.00228 **
latitude     0.0002471  0.0025120   0.098  0.92166
longitude   -0.0001522  0.0008622  -0.177  0.85992
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3687 on 1060 degrees of freedom
Multiple R-squared:  0.008785,	Adjusted R-squared:  0.00598
F-statistic: 3.132 on 3 and 1060 DF,  p-value: 0.02489
```

Indonesia				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	4.8041061	0.102913	46.681	<0.0000000000000002***
depth	0.0003903	0.000128	3.059	0.00228**
latitude	0.0002471	0.002512	0.098	0.92166
longitude	-0.0001522	0.000862	-0.177	0.85992

Table 4

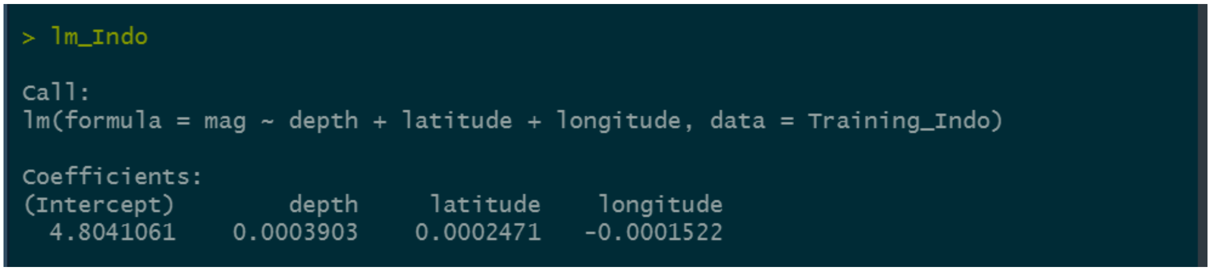
Here we have P value less than 0.05 and F-statistic: 3.132 >1 so we can say it Rejects Null Hypothesis(H0)



```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Indo)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.49412 -0.27361 -0.09815  0.11176  2.40812 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.8041061  0.1029125  46.681  < 2e-16 ***
depth        0.0003903  0.0001276   3.059  0.00228 ** 
latitude     0.0002471  0.0025120   0.098  0.92166    
longitude   -0.0001522  0.0008622  -0.177  0.85992    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3687 on 1060 degrees of freedom
Multiple R-squared:  0.008785,  Adjusted R-squared:  0.00598 
F-statistic: 3.132 on 3 and 1060 DF,  p-value: 0.02489

> 
```

Fig 20

Linear Model for Indonesia :



```
> lm_Indo

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Indo)

Coefficients:
(Intercept)        depth     latitude    longitude  
  4.8041061    0.0003903    0.0002471   -0.0001522  
```

Fig 21

```
lm(formula = mag ~ depth + latitude + longitude, data = Training_Indo)

Coefficients:
(Intercept)        depth     latitude    longitude  
  4.8041061    0.0003903    0.0002471   -0.0001522
```

4. RMSE Pakistan

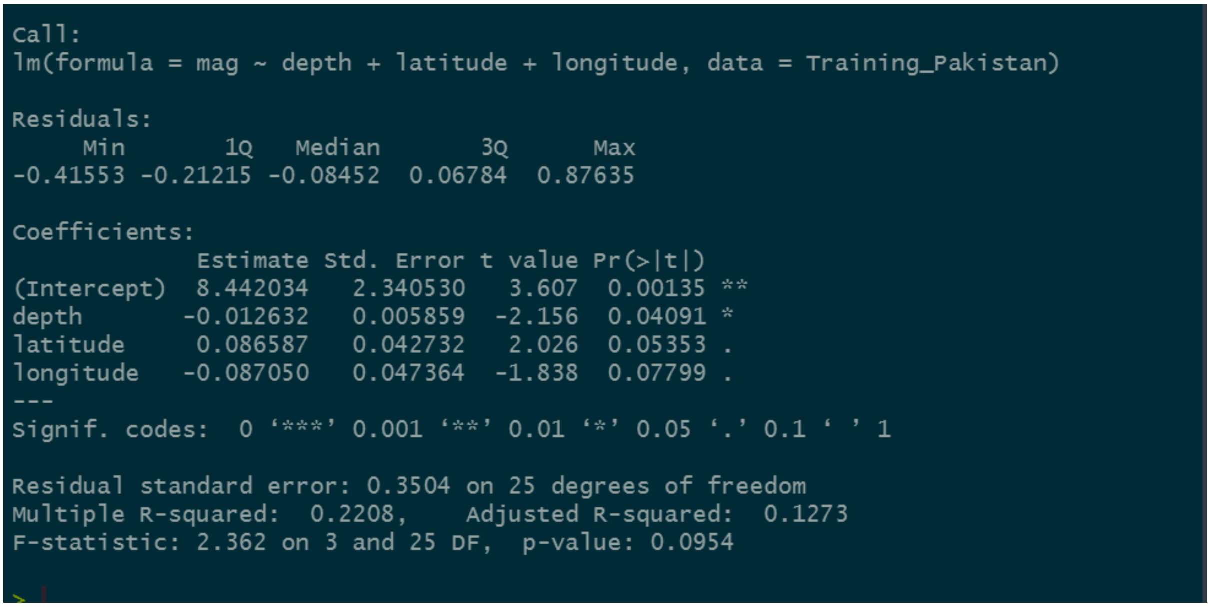
```
lm(formula = mag ~ depth + latitude + longitude, data = Training_Pakistan)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.41553 -0.21215 -0.08452  0.06784  0.87635 



Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.442034   2.340530   3.607  0.00135 **
depth       -0.012632   0.005859  -2.156  0.04091 *
latitude     0.086587   0.042732   2.026  0.05353 .
longitude   -0.087050   0.047364  -1.838  0.07799 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3504 on 25 degrees of freedom
Multiple R-squared:  0.2208,  Adjusted R-squared:  0.1273
F-statistic: 2.362 on 3 and 25 DF,  p-value: 0.0954
```

Pakistan				
	Estimate Std.	Error	t value	Pr(>|t|)
(Intercept)	8.442034	2.34053	3.607	0.00135**
depth	-0.012632	0.005859	-2.156	0.04091*
latitude	0.086587	0.042732	2.026	0.05353
longitude	-0.08705	0.047364	-1.838	0.07799

Table 5

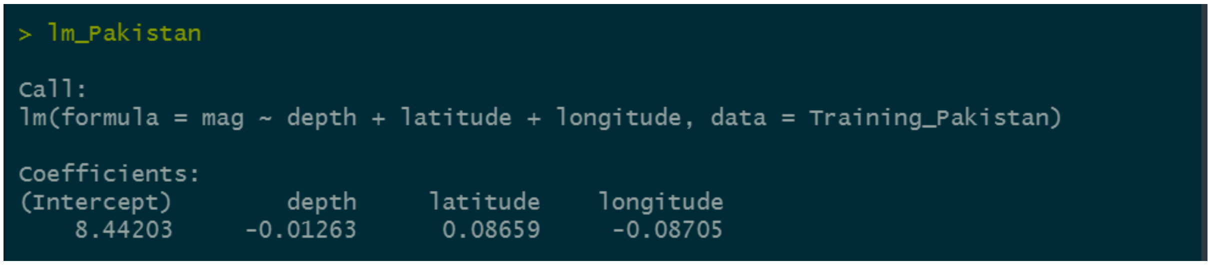
Here we have P value more than 0.05 so we can say it support Null Hypothesis(H0)



```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Pakistan)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.41553 -0.21215 -0.08452  0.06784  0.87635 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept)  8.442034   2.340530   3.607  0.00135 **
depth       -0.012632   0.005859  -2.156  0.04091 * 
latitude     0.086587   0.042732   2.026  0.05353 . 
longitude   -0.087050   0.047364  -1.838  0.07799 . 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3504 on 25 degrees of freedom
Multiple R-squared:  0.2208,	Adjusted R-squared:  0.1273 
F-statistic: 2.362 on 3 and 25 DF,  p-value: 0.0954

> 
```

Fig 22

Linear Model for Pakistan:



```
> lm_Pakistan

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Pakistan)

Coefficients:
(Intercept)        depth     latitude    longitude  
    8.44203     -0.01263      0.08659     -0.08705  
```

Fig 23

```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Pakistan)

Coefficients:
(Intercept)        depth     latitude    longitude  
    8.44203     -0.01263      0.08659     -0.08705  
```

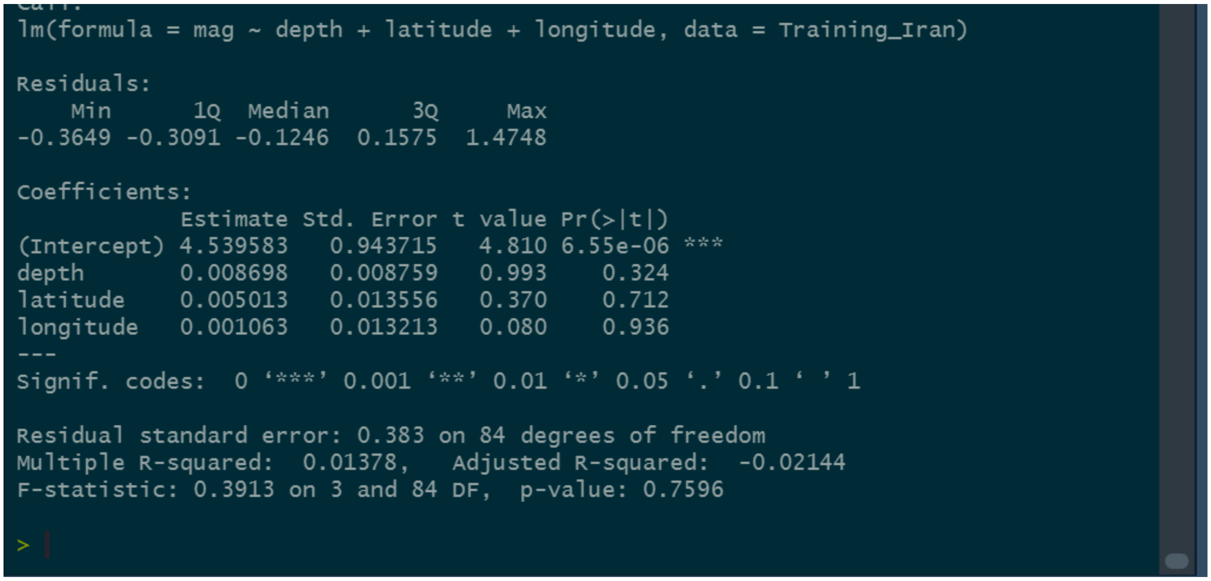
5. RMSE Iran

```
lm(formula = mag ~ depth + latitude + longitude, data = Training_Iran

Residuals:
    Min      1Q  Median      3Q     Max 
-0.3649 -0.3091 -0.1246  0.1575  1.4748 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.539583   0.943715   4.810 6.55e-06 ***
depth       0.008698   0.008759   0.993    0.324    
latitude    0.005013   0.013556   0.370    0.712    
longitude   0.001063   0.013213   0.080    0.936    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.383 on 84 degrees of freedom
Multiple R-squared:  0.01378,	Adjusted R-squared:  -0.02144 
F-statistic: 0.3913 on 3 and 84 DF,  p-value: 0.7596
```

Iran				
	Estimate Std.	Error	t value	Pr(>|t|)
(Intercept)	4.539583	0.943715	4.81	0.00000655***
depth	0.008698	0.008759	0.993	0.324
latitude	0.005013	0.013556	0.37	0.712
longitude	0.001063	0.013213	0.08	0.936

Table 7

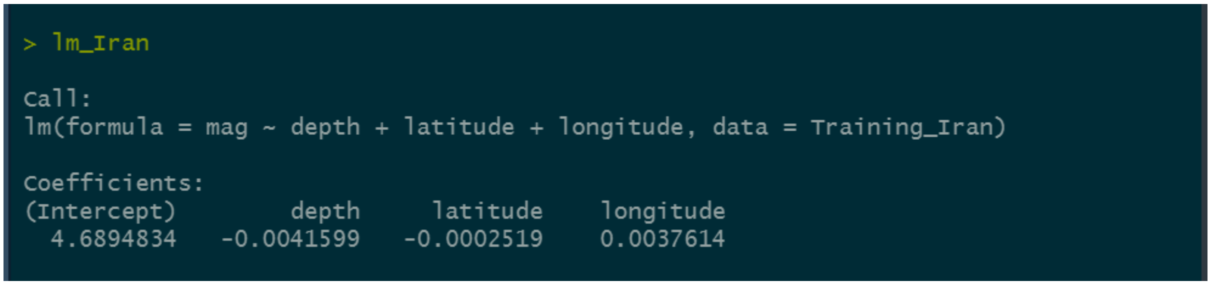
Here we have P value more than 0.05 and F-statistic: 0.3913 <1 so we can say it support Null Hypothesis(H0)



```
Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Iran)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.3649 -0.3091 -0.1246  0.1575  1.4748 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.539583   0.943715   4.810 6.55e-06 ***
depth       0.008698   0.008759   0.993    0.324    
latitude    0.005013   0.013556   0.370    0.712    
longitude   0.001063   0.013213   0.080    0.936    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.383 on 84 degrees of freedom
Multiple R-squared:  0.01378,	Adjusted R-squared:  -0.02144 
F-statistic: 0.3913 on 3 and 84 DF,  p-value: 0.7596

> 
```

Fig 24

Linear Model for Iran



```
> lm_Iran

Call:
lm(formula = mag ~ depth + latitude + longitude, data = Training_Iran)

Coefficients:
(Intercept)        depth     latitude    longitude  
  4.6894834   -0.0041599   -0.0002519    0.0037614  
```

Call:

lm(formula = mag ~ depth + latitude + longitude, data = Training_Iran)

Coefficients:

(Intercept) depth latitude longitude

4.6894834 -0.0041599 -0.0002519 0.0037614

6.1.8. Error Comparisons Between Different Countries:

Here we can see that in terms of Error Pakistan has maximum Error value **Pakistan>Iran>China>India>Indonesia**

India_Error	China_Error	Indonesia_Error	Pakistan_Error	Iran_Error
0.4435119	0.631876	0.1029125	2.34053	0.943715
0.0011265	0.006248	0.0001276	0.005859	0.008759
0.0033455	0.007514	0.002512	0.042732	0.013556
0.0045398	0.004401	0.0008622	0.047364	0.013213

Table 8

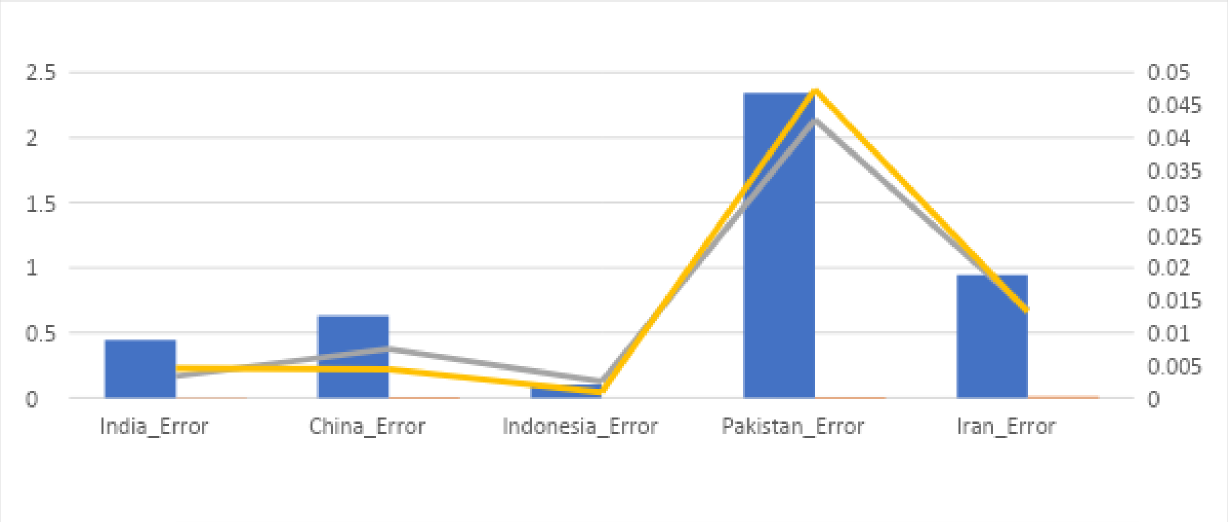
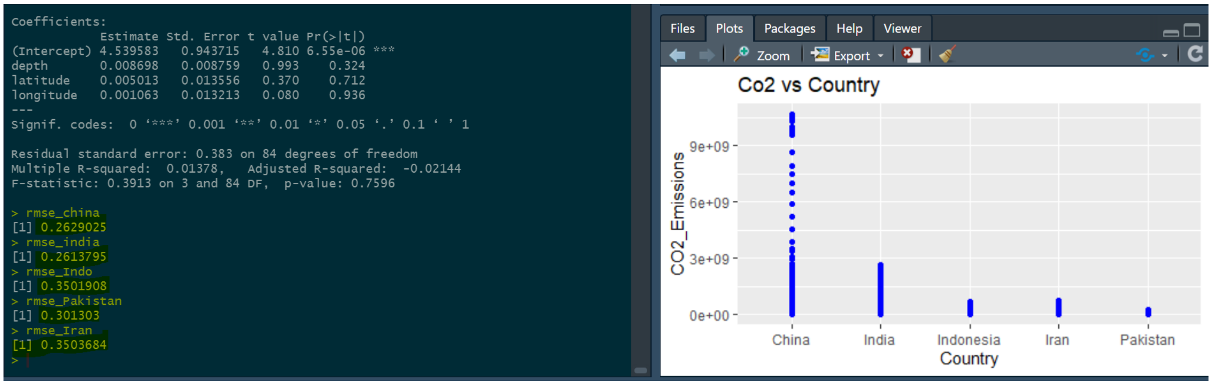


Fig 25

RMSE Value Comparison



```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.539583   0.943715   4.810 6.55e-06 ***
depth       0.008698   0.008759   0.993    0.324    
latitude    0.005013   0.013556   0.370    0.712    
longitude   0.001063   0.013213   0.080    0.936    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.383 on 84 degrees of freedom
Multiple R-squared:  0.01378,	Adjusted R-squared:  -0.02144 
F-statistic: 0.3913 on 3 and 84 DF,  p-value: 0.7596

> rmse_china
[1] 0.2629025
> rmse_india
[1] 0.2613795
> rmse_Indo
[1] 0.3501908
> rmse_Pakistan
[1] 0.301303
> rmse_Iran
[1] 0.3503684
> 
```

Fig 26

Country	RMSE
INDIA	**0.26290**
CHINA	0.26
INDONESIA	0.3501908
PAKISTAN	0.301303
IRAN	0.3503684

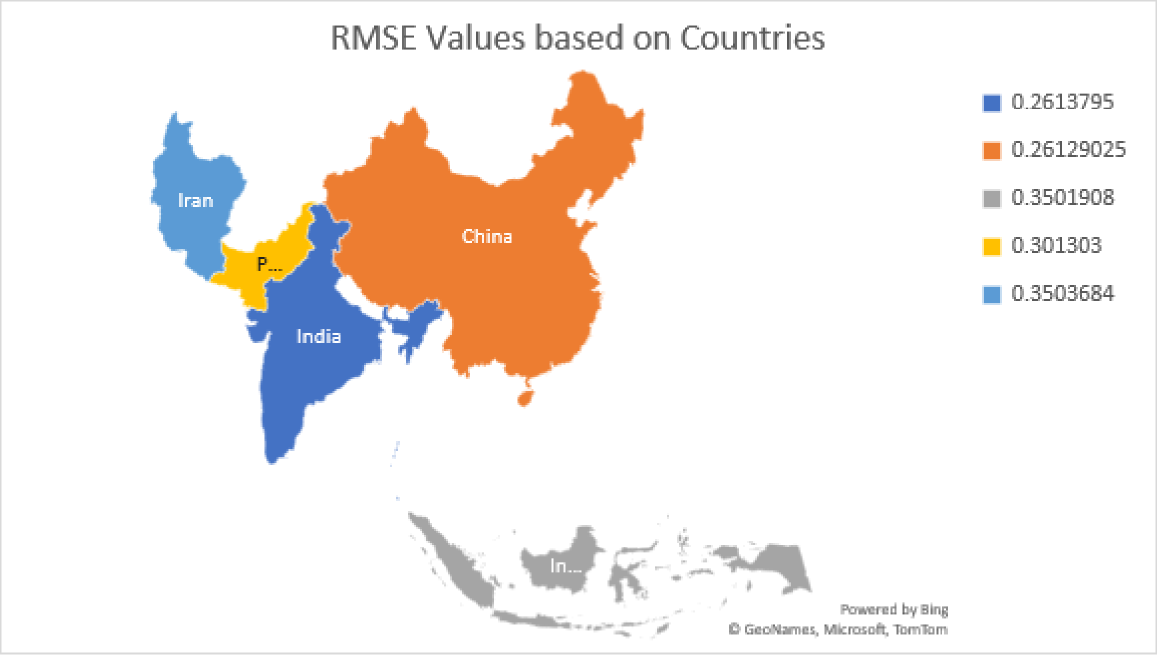


Fig 27

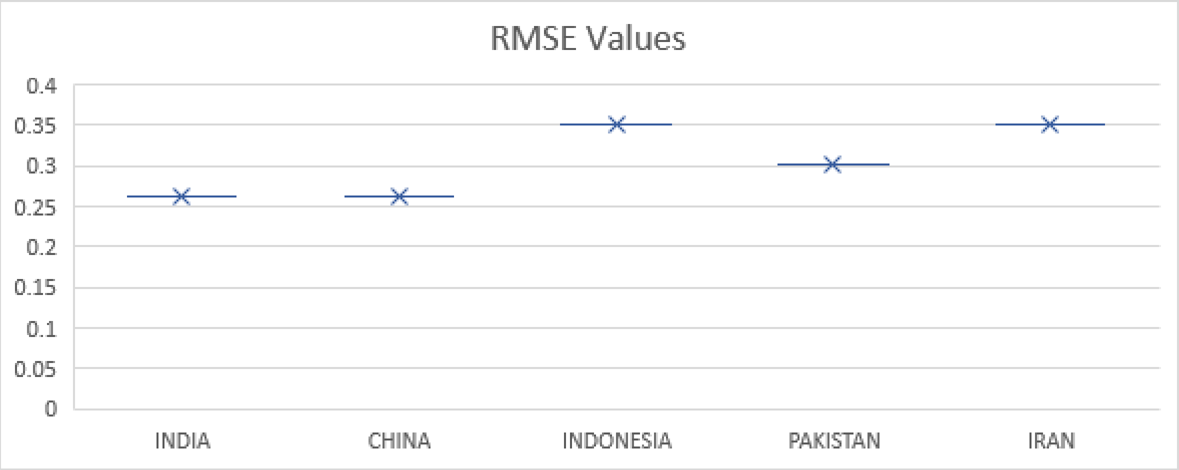

RMSE Values
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
INDIA
CHINA
INDONESIA
PAKISTAN
IRAN

Fig 28

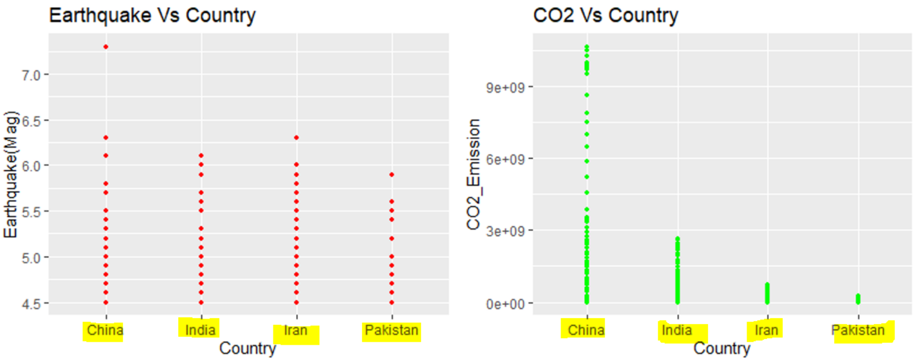

Earthquake Vs Country
Earthquake(Mag)
7.0
6.5
6.0
5.5
5.0
4.5
China
India
Iran
Pakistan
Country
CO2 Vs Country
CO2_Emission
9e+09
6e+09
3e+09
0e+00
China
India
Iran
Pakistan
Country

Fig 29

6.1.9. Summarized

As per our output based on Training and Testing Data, We can see the same pattern as we can see the rmse model/value, Where the earthquake is more we compare their magnitude with Location, Latitude and depth factor and we observed that China and India are best Fitted/Accurate Model as compare to other countries and When we checked the Co2 Emission dataset It found that China and India are the same countries where we have more C02 Emission in the world . So what we can say is that the Earthquake (Magnitude) likely depends or varies on the Environmental factors such as Co2.

Conclusion

Hence, we can conclude that, As shown in the Dataset and Model that we executed, it shows where we have more earthquakes …Are the same countries where we have more Co2 emission. So, Likely we can say or can estimate that the Co2/Environmental factor may depend on earthquake (Mag) and Our Mag depends on Depth, Longitude and Latitude.

6.2. Logistic Model

Logistic Regression is used when the dependent variable(Magnitude) is categorical.
Logistic regression is useful when you are predicting a binary outcome from a set of continuous predictor variables values.

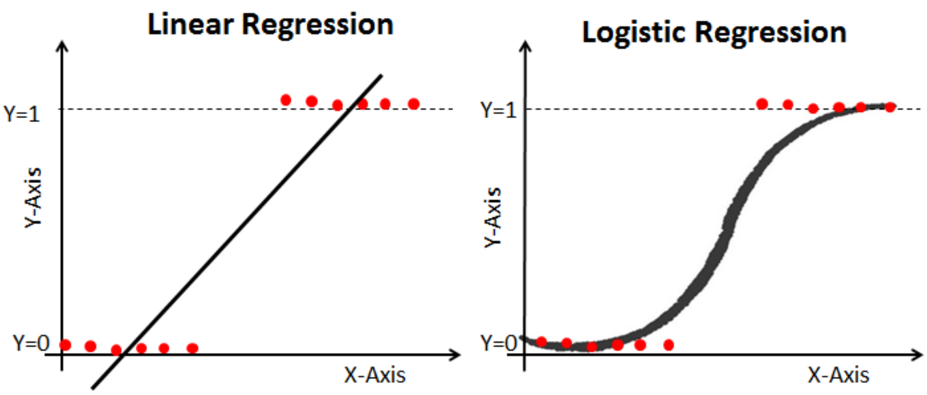
Types of Logistic Regression

1. Binary Logistic Regression(BLR)-
The categorical response has only two 2 possible outcomes. Example: Major or Minor.
2. Multinomial Logistic Regression(MLR)
Three or more categories without ordering.
3. Ordinal Logistic Regression(OLR)

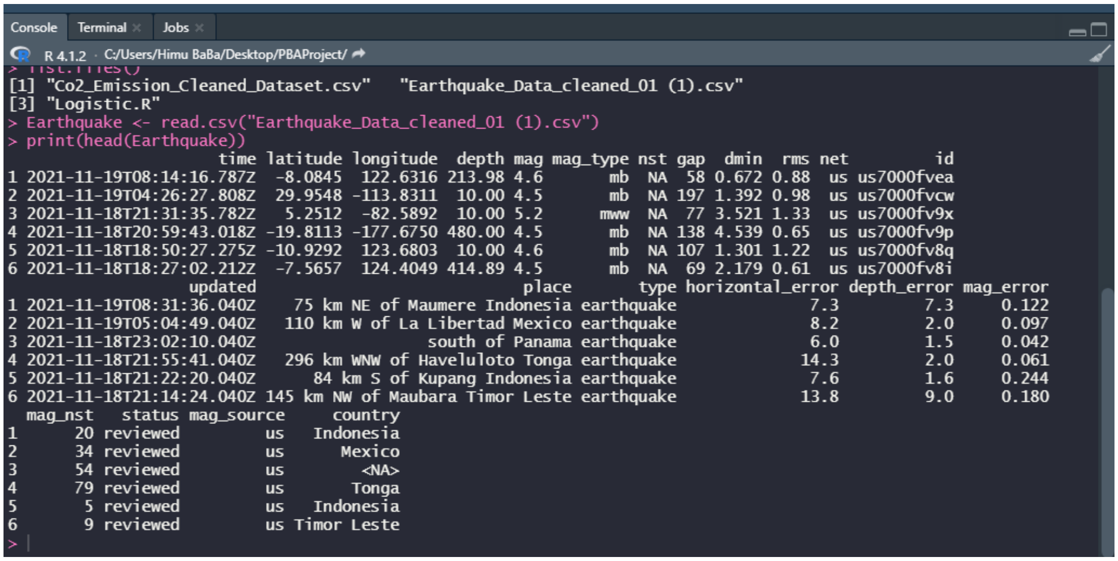
6.2.1. Linear Regression vs Logistic Regression:

Linear Regression is a widely used monitoring algorithm for a Learning Machine that predicts continuous values. Linear Regression assumes that there is a linear relationship that exists between dependent and independent variations. In simple terms, it finds the most relevant line / plane that describes two or more variables.

On the other hand, Logistic Regression is another supervised machine learning algorithm that is very helpful in binary separation (dividing intelligent values).
Linear Regression is used to solve the setback problem On the other hand, Logistic Regression is used to solve the editing problem.
Linear Regression output should be continuous values like price, age etc.
Although, the Outflow Output should be a category value such as 0 or 1, Yes or No etc.
Yes or No etc.



Data Modelling:



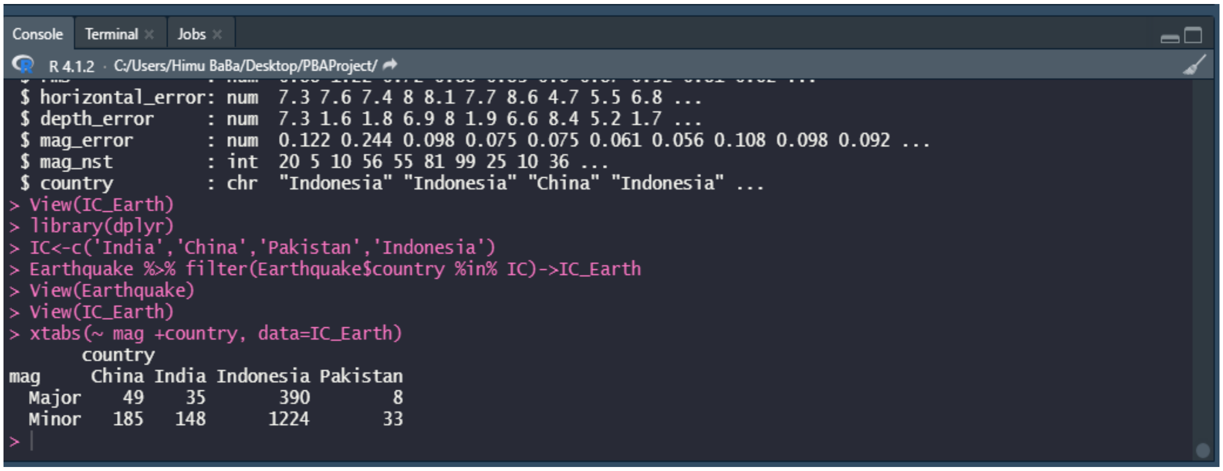
```
> list.files()
[1] "Co2_Emission_Cleaned_Dataset.csv"   "Earthquake_Data_cleaned_01 (1).csv"
[3] "Logistic.R"
> Earthquake <- read.csv("Earthquake_Data_cleaned_01 (1).csv")
> print(head(Earthquake))
                      time latitude longitude  depth mag mag_type nst gap  dmin  rms net         id
1 2021-11-19T08:14:16.787Z  -8.0845  122.6316 213.98 4.6       mb  NA  58 0.672 0.88  us us7000fvea
2 2021-11-19T04:26:27.808Z  29.9548 -113.8311  10.00 4.5       mb  NA 197 1.392 0.98  us us7000fvcw
3 2021-11-18T21:31:35.782Z   5.2512  -82.5892  10.00 5.2      mww  NA  77 3.521 1.33  us us7000fv9x
4 2021-11-18T20:59:43.018Z -19.8113 -177.6750 480.00 4.5       mb  NA 138 4.539 0.65  us us7000fv9p
5 2021-11-18T18:50:27.275Z -10.9292  123.6803  10.00 4.6       mb  NA 107 1.301 1.22  us us7000fv8q
6 2021-11-18T18:27:02.212Z  -7.5657  124.4049 414.89 4.5       mb  NA  69 2.179 0.61  us us7000fv8i
                   updated                            place       type horizontal_error depth_error mag_error
1 2021-11-19T08:31:36.040Z    75 km NE of Maumere Indonesia earthquake              7.3         7.3     0.122
2 2021-11-19T05:04:49.040Z   110 km W of La Libertad Mexico earthquake              8.2         2.0     0.097
3 2021-11-18T23:02:10.040Z                  south of Panama earthquake              6.0         1.5     0.042
4 2021-11-18T21:55:41.040Z    296 km WNW of Haveluloto Tonga earthquake             14.3         2.0     0.061
5 2021-11-18T21:22:20.040Z      84 km S of Kupang Indonesia earthquake              7.6         1.6     0.244
6 2021-11-18T21:14:24.040Z 145 km NW of Maubara Timor Leste earthquake             13.8         9.0     0.180
  mag_nst   status mag_source     country
1      20 reviewed         us   Indonesia
2      34 reviewed         us      Mexico
3      54 reviewed         us        <NA>
4      79 reviewed         us       Tonga
5       5 reviewed         us   Indonesia
6       9 reviewed         us Timor Leste
>
```

Fig 30

Xtabs-

The xtabs() function in R allows you to quickly calculate frequencies for one or more variables. It uses the following basic syntax:
xtabs(~variable_nam, data=data)
where:

- **variable_name:** The variable that you'd like to calculate the frequencies for. I.e. magnitude and country.
- **data:** The name of the data frame that the variable comes from. I.e. IC_Earth.

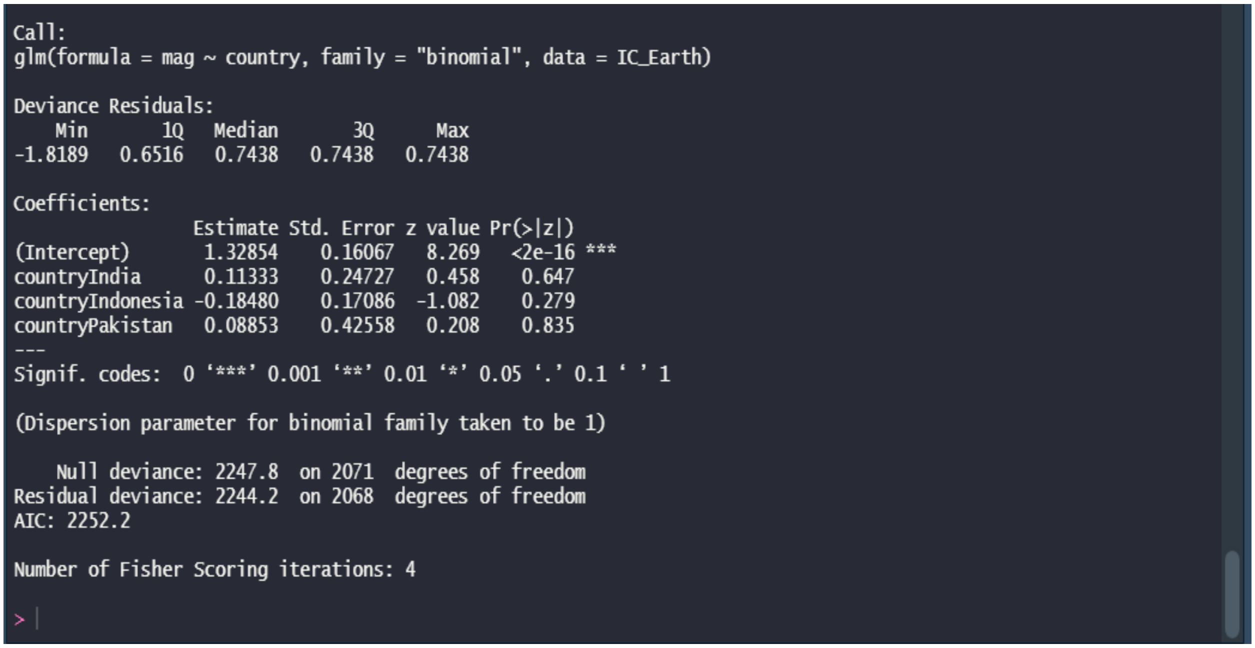


```
 $ horizontal_error: num  7.3 7.6 7.4 8 8.1 7.7 8.6 4.7 5.5 6.8 ...
 $ depth_error     : num  7.3 1.6 1.8 6.9 8 1.9 6.6 8.4 5.2 1.7 ...
 $ mag_error       : num  0.122 0.244 0.098 0.075 0.075 0.061 0.056 0.108 0.098 0.092 ...
 $ mag_nst         : int  20 5 10 56 55 81 99 25 10 36 ...
 $ country         : chr  "Indonesia" "Indonesia" "China" "Indonesia" ...
> View(IC_Earth)
> library(dplyr)
> IC<-c('India','China','Pakistan','Indonesia')
> Earthquake %>% filter(Earthquake$country %in% IC)->IC_Earth
> View(Earthquake)
> View(IC_Earth)
> xtabs(~ mag +country, data=IC_Earth)
       country
mag     China India Indonesia Pakistan
  Major    49    35       390        8
  Minor   185   148      1224       33
> 
```

Fig 31

GLM call:

Generalized linear models (GLM) address these conditions by allowing flexible responses with random distribution (except for normal distribution only), and by using the absurd function of response variance (so-called link function) to vary in sequence of predicted values. (rather than assuming that the answer itself should vary according to the forecast). Thus, in the standard line model (GLM), each Y-effect of dependency variation is assumed to be produced from an exponential distribution family (including distributions such as normal distribution, binomial, Poisson and gamma, among others). GLM thus expands the scenario where linear regression can work by increasing the chances of outcome fluctuations. GLM uses the maximum possible parameter of the descriptive family model and the small squares of standard line models.



```
Call:
glm(formula = mag ~ country, family = "binomial", data = IC_Earth)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.8189   0.6516   0.7438   0.7438   0.7438  

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)       1.32854    0.16067   8.269   <2e-16 ***
countryIndia      0.11333    0.24727   0.458    0.647    
countryIndonesia -0.18480    0.17086  -1.082    0.279    
countryPakistan   0.08853    0.42558   0.208    0.835    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2247.8  on 2071  degrees of freedom
Residual deviance: 2244.2  on 2068  degrees of freedom
AIC: 2252.2

Number of Fisher Scoring iterations: 4

> 
```

Fig 32

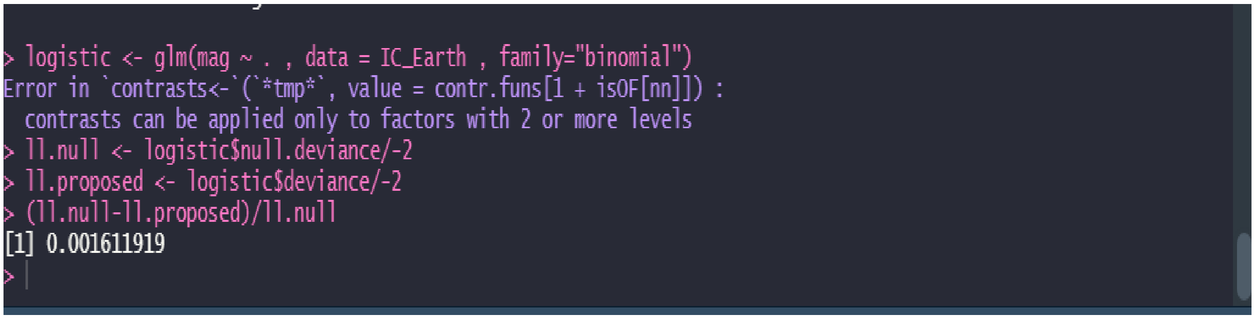
AIC value:

The Akaike information criterion (AIC) is a mathematical formula to measure the accuracy of a model with its generated data. In the calculations, the AIC is used to compare different possible models and determine which data is appropriate. AIC is calculated from:

- the number of independent variables used to construct the model.
- maximum model capability (how the model generates data)

R Squared value:

This value uses the input of the specified model and the corresponding “cross only” model and determines its value. This rating is then subtracted from 1 to determine the reported value. The small scale (as well as the final value close to 1) indicates that the specified model is better than the cross-sectional model only.



```
> logistic <- glm(mag ~ . , data = IC_Earth , family="binomial")
Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) : 
  contrasts can be applied only to factors with 2 or more levels
> ll.null <- logistic$null.deviance/-2
> ll.proposed <- logistic$deviance/-2
> (ll.null-ll.proposed)/ll.null
[1] 0.001611919
> 
```

Fig 33.

P value:

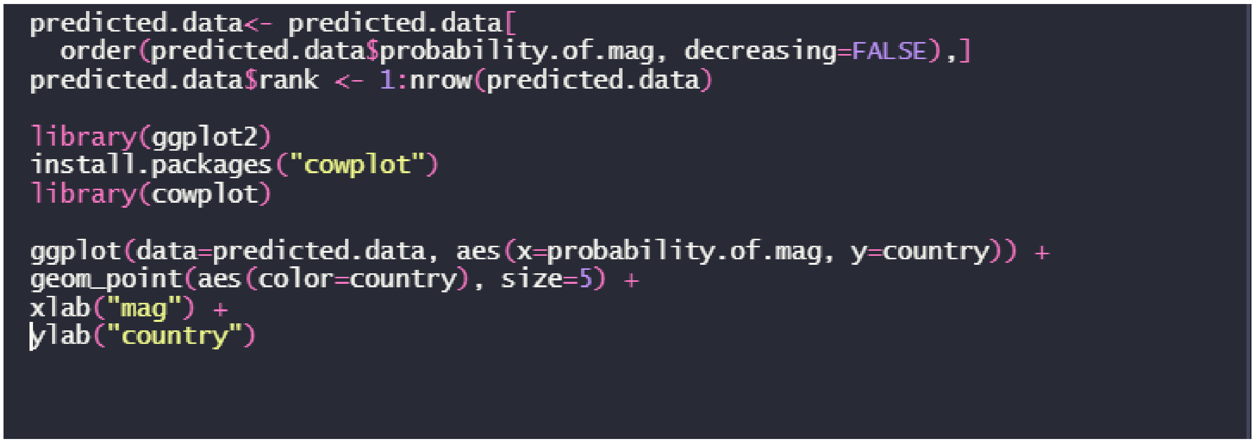
The p value of each term tests the null hypothesis that the coefficient is equal to zero (no result). A low p value (<0.05) indicates that you can reject the null hypothesis. In other words, a prediction with a low p value may be a reasonable addition to your model because changes in the forecast value are related to changes in response variability.

Formula:
ll.null <- logistic$null.deviance/-2
ll.proposed <- logistic$deviance/-2
(ll.null-ll.proposed)/ll.null

1 - pchisq(2*(ll.proposed - ll.null), df=(length(logistic$coefficients)-1))

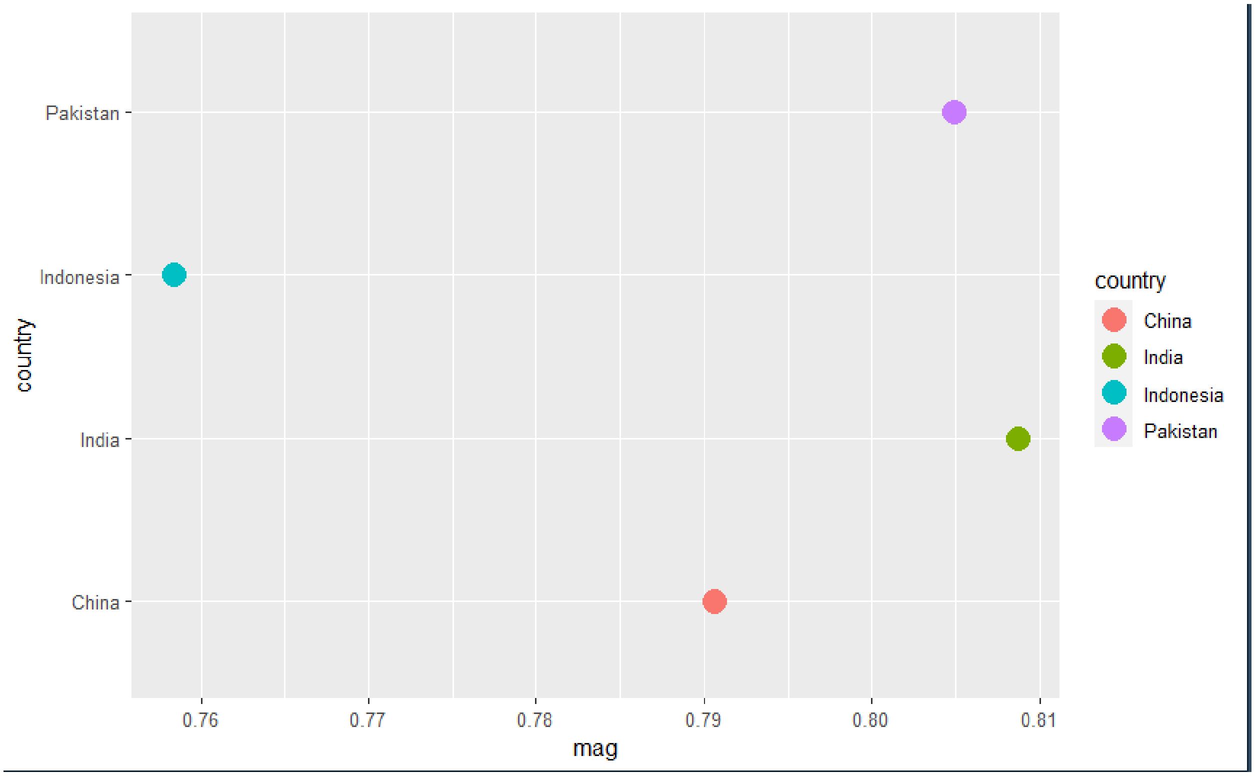
P value of our model- 0.3051171

6.2.2. Visualization :



```
predicted.data<- predicted.data[
  order(predicted.data$probability.of.mag, decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)

library(ggplot2)
install.packages("cowplot")
library(cowplot)

ggplot(data=predicted.data, aes(x=probability.of.mag, y=country)) +
geom_point(aes(color=country), size=5) +
xlab("mag") +
ylab("country")
```

Fig 34



6.3. Decision tree

6.3.1. Decision tree for earthquake dataset

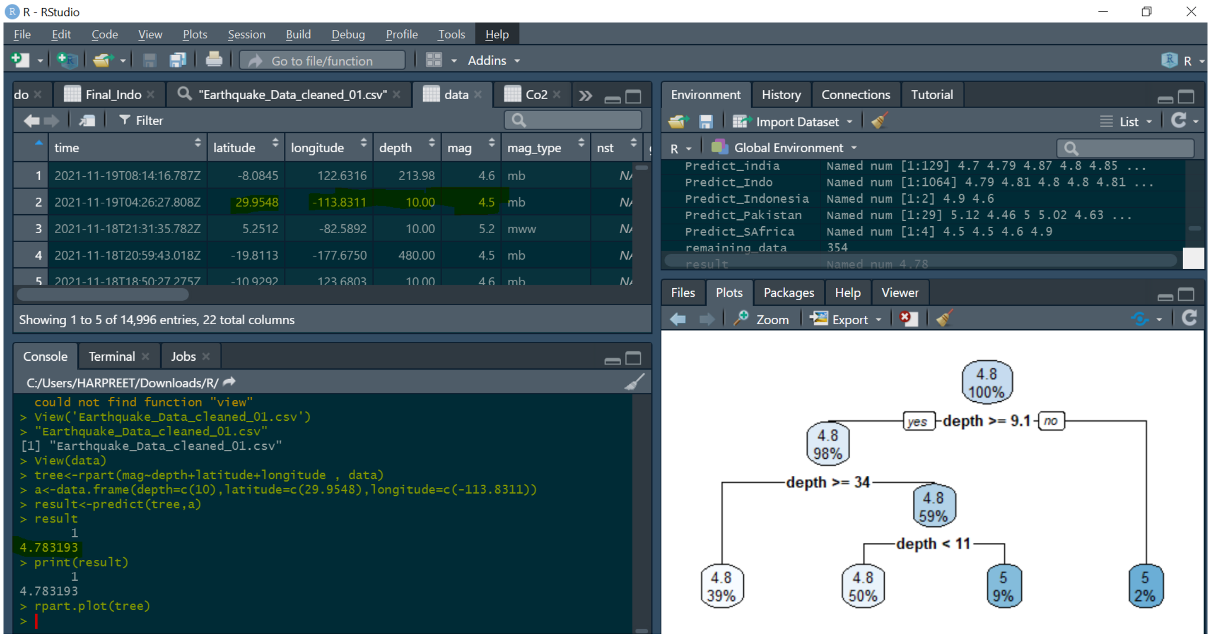
Decision Trees , It is a algorithm in Machine Learning that can perform classification as well as regression tasks. It is more capable of fitting or adjusting complex datasets. Besides, decision trees are fundamental components of random forests.
In this we are predicting magnitude by taking depth + Latitude and Longitude. So we have used Predict() function as mentioned below and Try to predict Magnitude value for different location (Latitude longitude and depth)

Values taken :
depth=c(10),latitude=c(29.9548),longitude=c(-113.8311) as per our dataset magnitude is 4.5 and we from our model prediction we are getting somewhat around 4.7(approx)

Code Explanation
rpart(): Function use to train the model and argument used
mag ~.: Formula used in DT(Decision Tree)

data = Training_Data

rpart.plot(tree): Plot the tree.



	time	latitude	longitude	depth	mag	mag_type	nst
1	2021-11-19T08:14:16.787Z	-8.0845	122.6316	213.98	4.6	mb	NA
2	2021-11-19T04:26:27.808Z	29.9548	-113.8311	10.00	4.5	mb	NA
3	2021-11-18T21:31:35.782Z	5.2512	-82.5892	10.00	5.2	mww	NA
4	2021-11-18T20:59:43.018Z	-19.8113	-177.6750	480.00	4.5	mb	NA
5	2021-11-18T18:50:27.275Z	-10.9292	123.6803	10.00	4.6	mb	NA

Showing 1 to 5 of 14,996 entries, 22 total columns

```
Predict_india      Named num [1:129] 4.7 4.79 4.87 4.8 4.85 ...
Predict_Indo       Named num [1:1064] 4.79 4.81 4.8 4.8 4.81 ...
Predict_Indonesia  Named num [1:2] 4.9 4.6
Predict_Pakistan   Named num [1:29] 5.12 4.46 5 5.02 4.63 ...
Predict_SAfrica    Named num [1:4] 4.5 4.5 4.6 4.9
remaining_data     354
result             Named num 4.78
```

C:/Users/HARPREET/Downloads/R/

```
  could not find function "view"
> View('Earthquake_Data_cleaned_01.csv')
> "Earthquake_Data_cleaned_01.csv"
[1] "Earthquake_Data_cleaned_01.csv"
> View(data)
> tree<-rpart(mag~depth+latitude+longitude , data)
> a<-data.frame(depth=c(10),latitude=c(29.9548),longitude=c(-113.8311))
> result<-predict(tree,a)
> result
       1
4.783193
> print(result)
       1
4.783193
> rpart.plot(tree)
> 
```

Fig. 35

Algo Used:

> View(data)
> tree<-rpart(mag~depth+latitude+longitude , data)
> a<-data.frame(depth=c(10),latitude=c(29.9548),longitude=c(-113.8311))
> result<-predict(tree,a)
> result

4.783193

Visualization output1 is 4.7

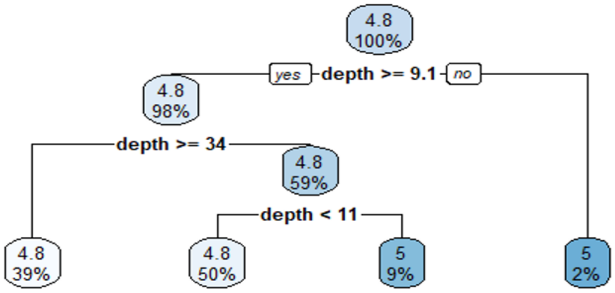
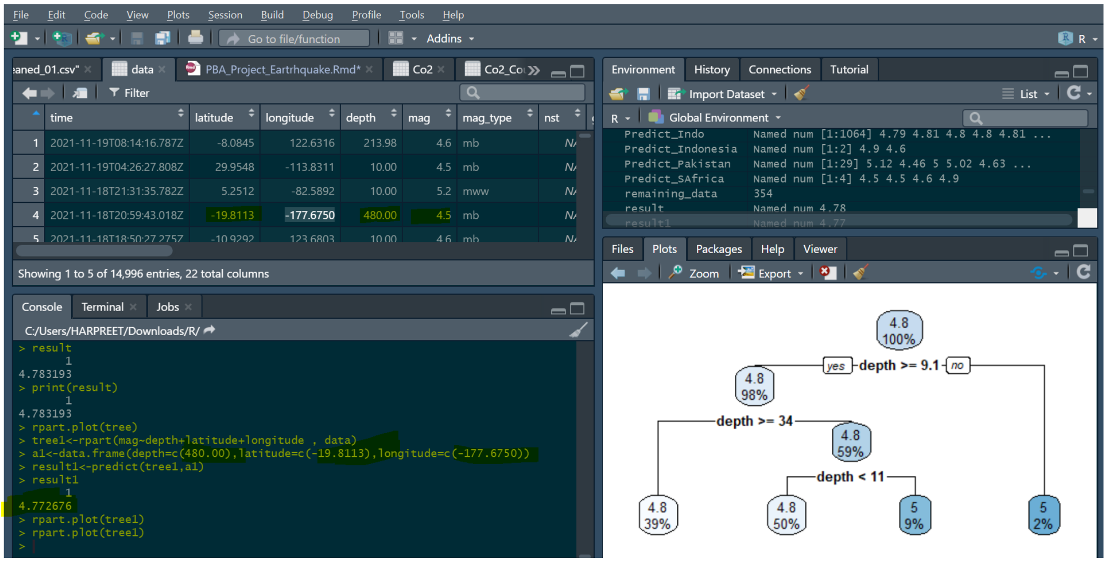


Fig. 36

Tried with different Values

Predicted Value is 4.77 actual is 4.5 hence we are somewhat nearby



	time	latitude	longitude	depth	mag	mag_type	nst
1	2021-11-19T08:14:16.787Z	-8.0845	122.6316	213.98	4.6	mb	NA
2	2021-11-19T04:26:27.808Z	29.9548	-113.8311	10.00	4.5	mb	NA
3	2021-11-18T21:31:35.782Z	5.2512	-82.5892	10.00	5.2	mww	NA
4	2021-11-18T20:59:43.018Z	-19.8113	-177.6750	480.00	4.5	mb	NA
5	2021-11-18T18:50:27.275Z	-10.9292	123.6803	10.00	4.6	mb	NA

Showing 1 to 5 of 14,996 entries, 22 total columns

```
Predict_Indo          Named num [1:1064] 4.79 4.81 4.8 4.8 4.81 ...
Predict_Indonesia     Named num [1:2] 4.9 4.6
Predict_Pakistan      Named num [1:29] 5.12 4.46 5 5.02 4.63 ...
Predict_SAfrica       Named num [1:4] 4.5 4.5 4.6 4.9
remaining_data        354
result                Named num 4.78
result1               Named num 4.77
```

```
> result
       1
4.783193
> print(result)
       1
4.783193
> rpart.plot(tree)
> tree1<-rpart(mag~depth+latitude+longitude , data)
> a1<-data.frame(depth=c(480.00),latitude=c(-19.8113),longitude=c(-177.6750))
> result1<-predict(tree1,a1)
> result1
       1
4.772676
> rpart.plot(tree1)
> rpart.plot(tree1)
>
```

Fig. 37

Visualization :

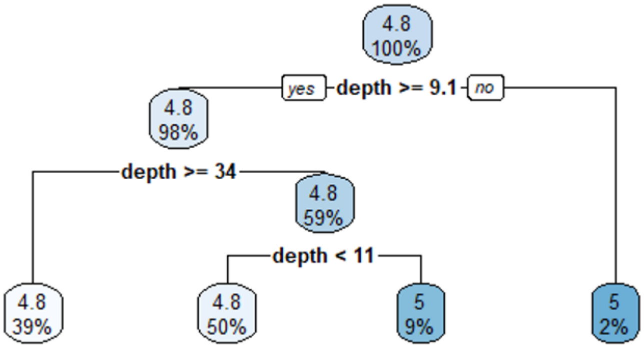


Fig. 38

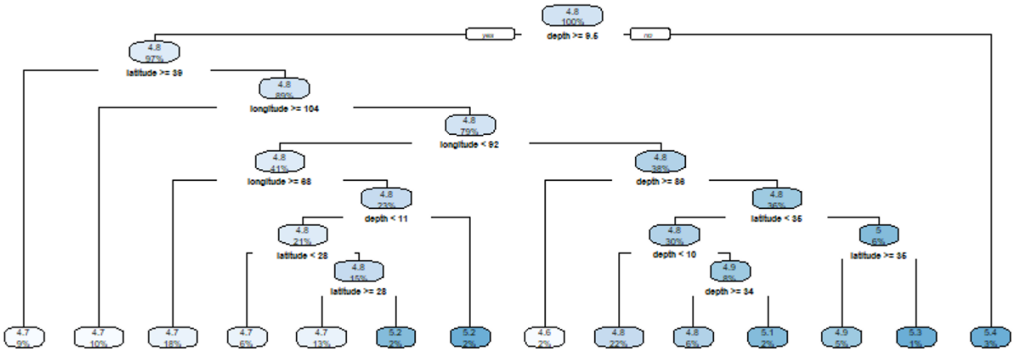


Fig. 39

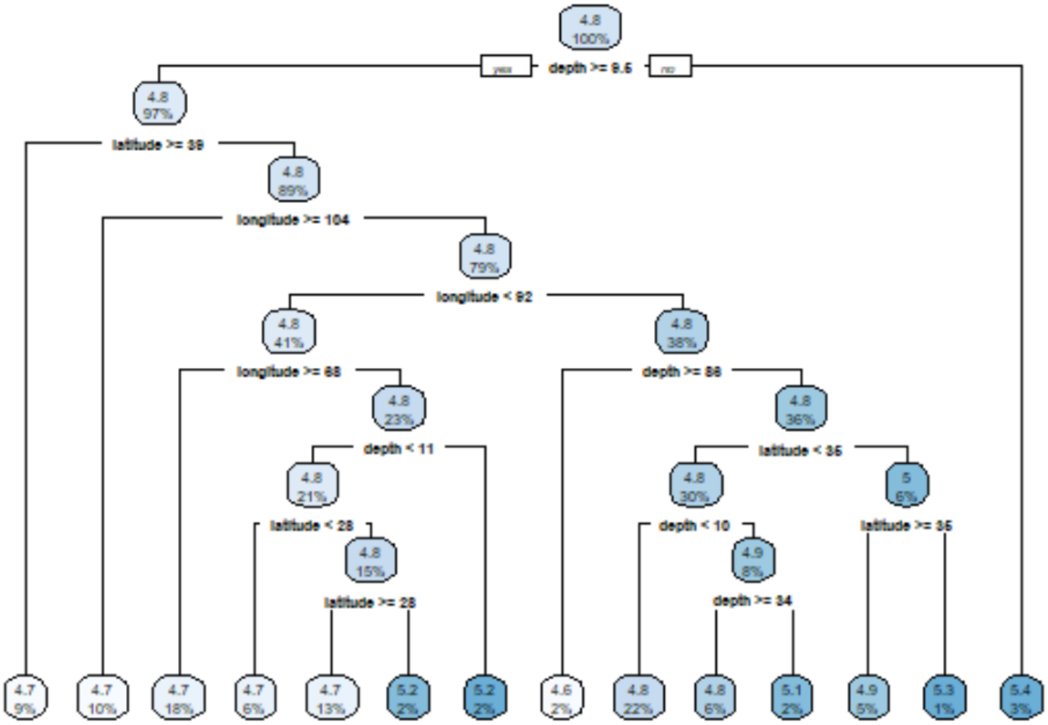
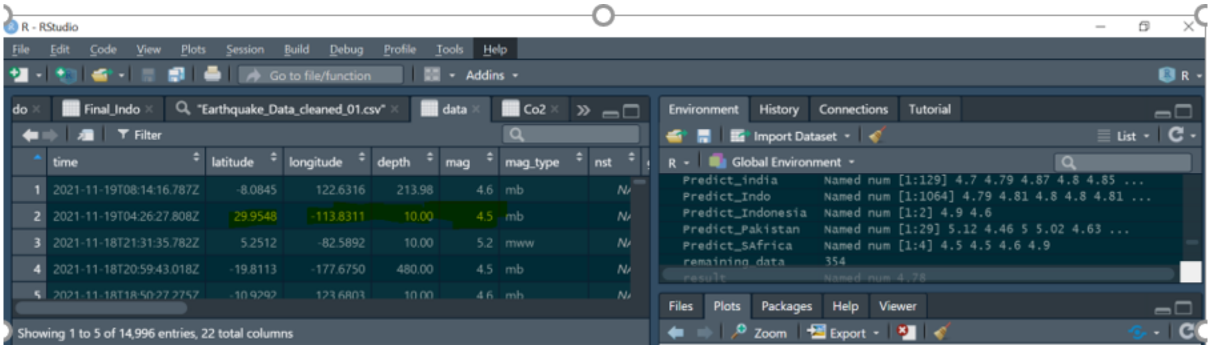

4.8
100%
yes
depth >= 9.5
no
4.8
97%
latitude >= 39
4.8
89%
longitude >= 104
4.8
79%
longitude < 92
4.8
41%
longitude >= 88
4.8
23%
depth < 11
4.8
21%
latitude < 28
4.8
15%
latitude >= 28
4.8
38%
depth >= 86
4.8
36%
latitude < 35
4.8
30%
depth < 10
4.9
8%
depth >= 34
5
6%
latitude >= 35
4.7
9%
4.7
10%
4.7
18%
4.7
6%
4.7
13%
5.2
2%
5.2
2%
4.6
2%
4.8
22%
4.8
6%
5.1
2%
4.9
5%
5.3
1%
5.4
3%

Fig. 40

Predicting Value

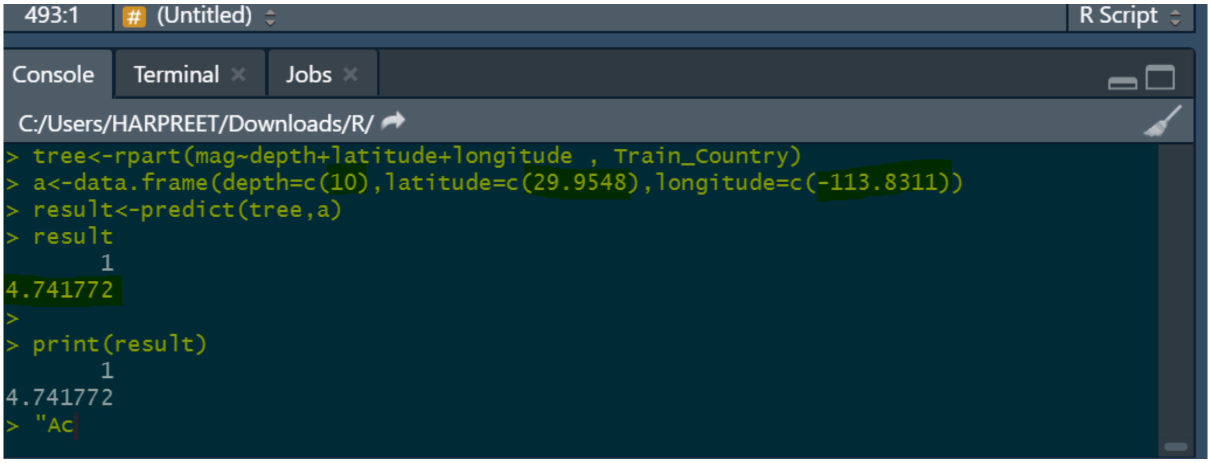
DataSet Value:


R - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
do
Final_Indo
"Earthquake_Data_cleaned_01.csv"
data
Co2
Filter
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
Files Plots Packages Help Viewer
Zoom
Export

	time	latitude	longitude	depth	mag	mag_type	nst
1	2021-11-19T08:14:16.787Z	-8.0845	122.6316	213.98	4.6	mb	NA
2	2021-11-19T04:26:27.808Z	29.9548	-113.8311	10.00	4.5	mb	NA
3	2021-11-18T21:31:35.782Z	5.2512	-82.5892	10.00	5.2	mww	NA
4	2021-11-18T20:59:43.018Z	-19.8113	-177.6750	480.00	4.5	mb	NA
5	2021-11-18T18:50:27.275Z	-10.9292	123.6803	10.00	4.6	mb	NA

Showing 1 to 5 of 14,996 entries, 22 total columns

```
Predict_india       Named num [1:129] 4.7 4.79 4.87 4.8 4.85 ...
Predict_Indo        Named num [1:1064] 4.79 4.81 4.8 4.8 4.81 ...
Predict_Indonesia   Named num [1:2] 4.9 4.6
Predict_Pakistan    Named num [1:29] 5.12 4.46 5 5.02 4.63 ...
Predict_SAfrica     Named num [1:4] 4.5 4.5 4.6 4.9
remaining_data      354
result              Named num 4.78
```



```
> tree<-rpart(mag~depth+latitude+longitude , Train_Country)
> a<-data.frame(depth=c(10),latitude=c(29.9548),longitude=c(-113.8311))
> result<-predict(tree,a)
> result
       1
4.741772
>
> print(result)
       1
4.741772
> "Ac
```

Fig. 41

Calculating Countries Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

This true positive and true negative over the sum of the matrix. With R, you can code as follow:

accuracy_Test1 <-
(diag(table_mat)) / sum(table_mat)

Also Explained:

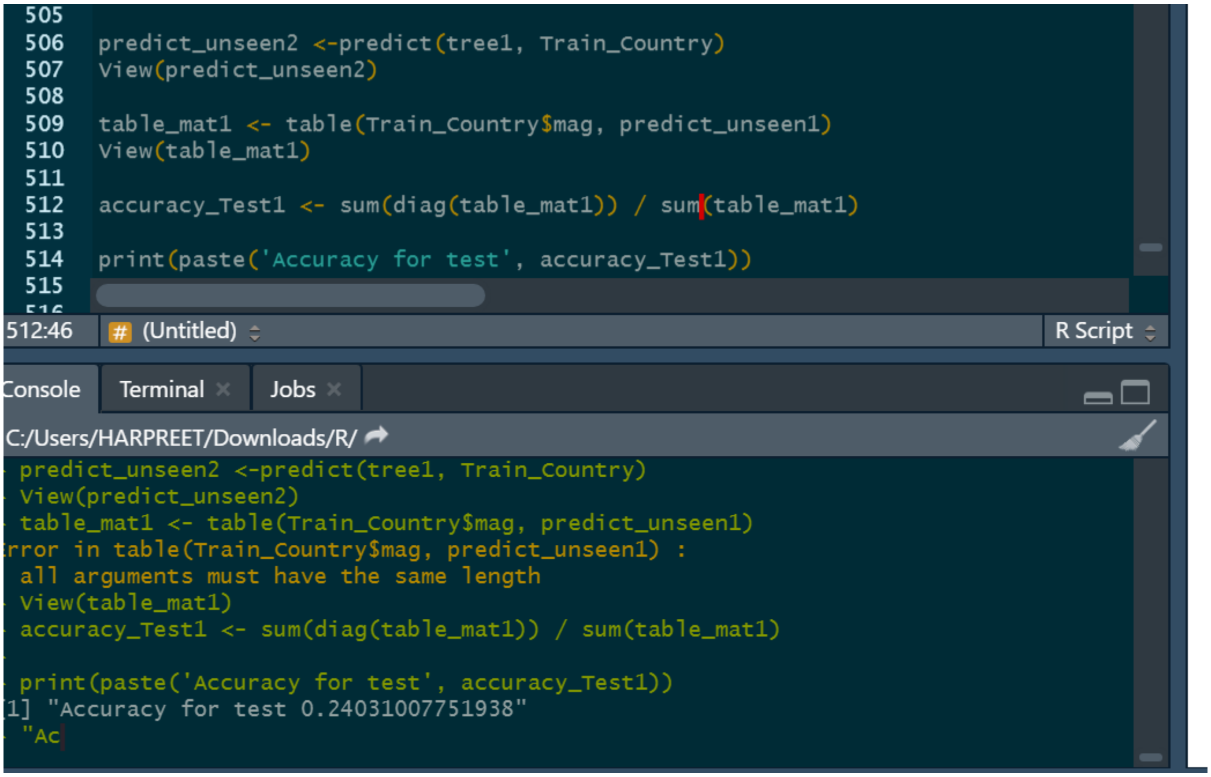
- Sum of the (diagonal (table_mat is our variable)): Sum of the diagonal
- Sum of the variable(table_mat): Sum of the matrix.

You can print the accuracy of the test set:

print(paste('Test_Accuracy', accuracy_Test))

Output:

"Accuracy of Test 24%"



```
predict_unseen2 <-predict(tree1, Train_Country)
View(predict_unseen2)

table_mat1 <- table(Train_Country$mag, predict_unseen1)
View(table_mat1)

accuracy_Test1 <- sum(diag(table_mat1)) / sum(table_mat1)

print(paste('Accuracy for test', accuracy_Test1))
```

```
 predict_unseen2 <-predict(tree1, Train_Country)
 View(predict_unseen2)
 table_mat1 <- table(Train_Country$mag, predict_unseen1)
rror in table(Train_Country$mag, predict_unseen1) :
 all arguments must have the same length
 View(table_mat1)
 accuracy_Test1 <- sum(diag(table_mat1)) / sum(table_mat1)

 print(paste('Accuracy for test', accuracy_Test1))
1] "Accuracy for test 0.24031007751938"
 "Ac
```

Fig 42

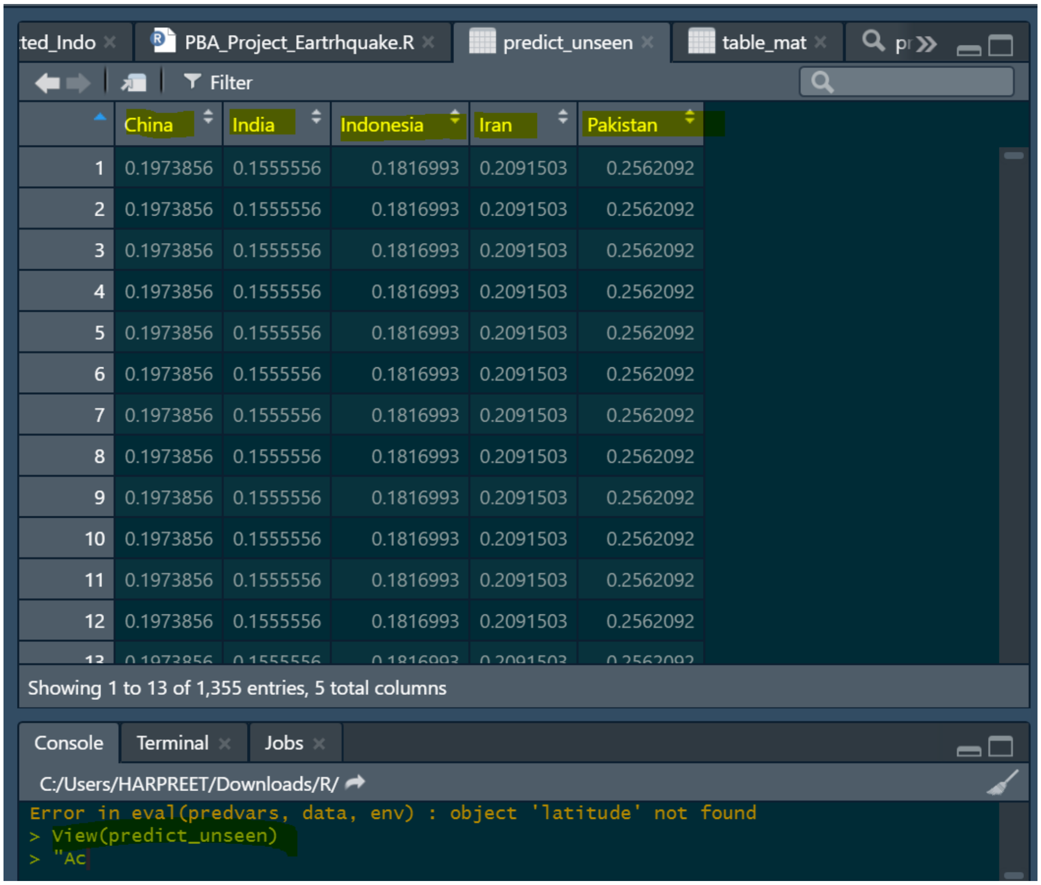
6.3.2. Decision Tree for C02 DataSet

In this we are again predicting the values for different Countries as mentioned in our Train_Country Dataset .

Making Prediction:

To make a prediction, we use the function predict() function. The basic syntax of predict for R decision tree is:

predict(fitted_model, Data_Frame)
arguments:
- fitted_model: This is the value that we stored after estimating the model.
- Data Frame Data frame used to make the prediction
- type: Type of prediction



	China	India	Indonesia	Iran	Pakistan
1	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
2	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
3	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
4	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
5	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
6	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
7	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
8	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
9	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
10	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
11	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
12	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092
13	0.1973856	0.1555556	0.1816993	0.2091503	0.2562092

Showing 1 to 13 of 1,355 entries, 5 total columns

```
C:/Users/HARPREET/Downloads/R/
Error in eval(predvars, data, env) : object 'latitude' not found
> View(predict_unseen)
> "Ac
```

Predicted Values(Fig 9)

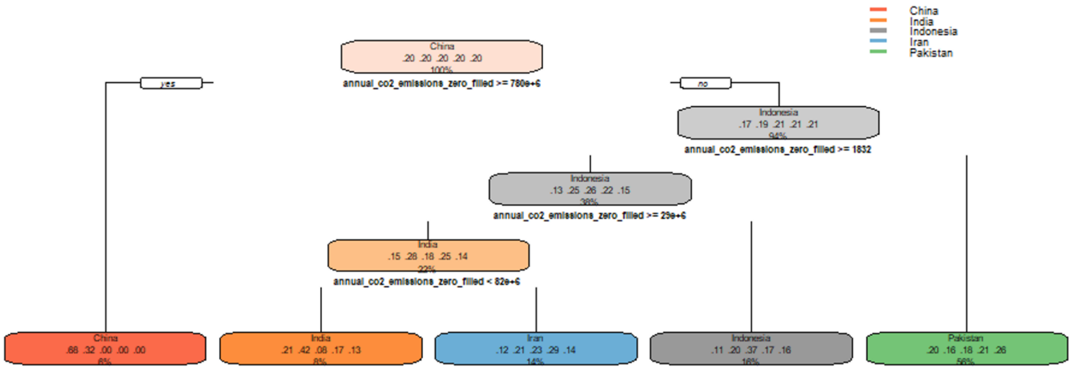


Fig. 43

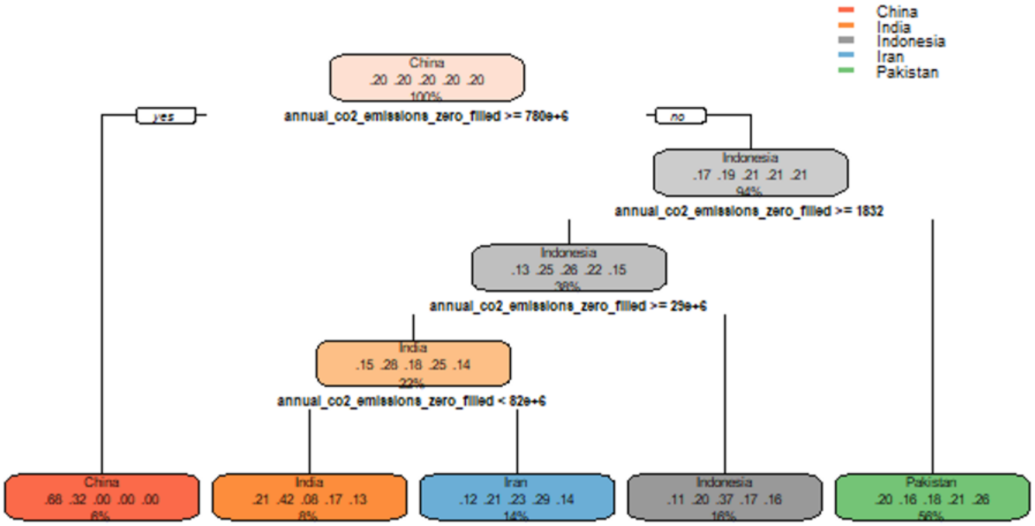
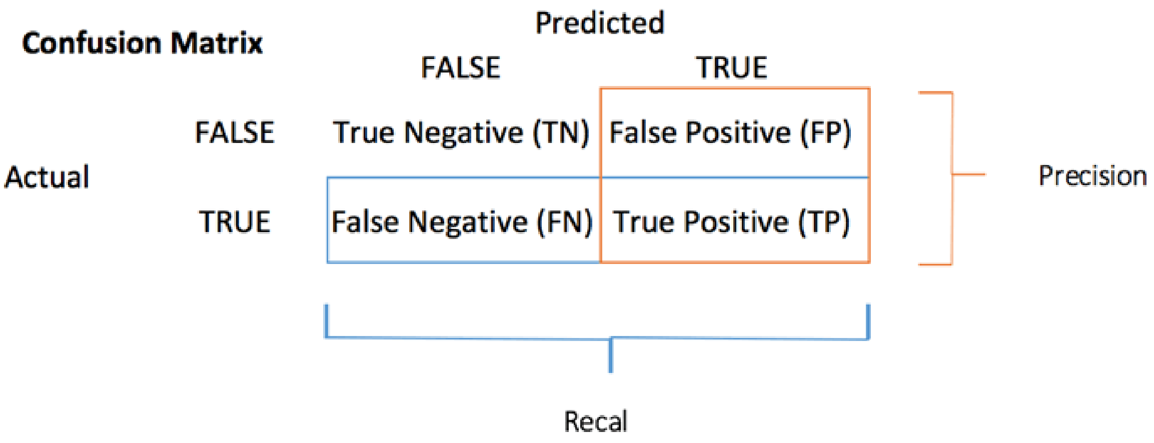


Fig 44

You can compute an accuracy measure for classification task with the confusion matrix:

The confusion matrix is a better choice to evaluate the classification performance. The general idea is to count the number of times True instances are classified are False.



Confusion Matrix

		Predicted	
		FALSE	TRUE
Actual	FALSE	True Negative (TN)	False Positive (FP)
	TRUE	False Negative (FN)	True Positive (TP)

You can compute the accuracy test from the confusion matrix:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

This true positive and true negative over the sum of the matrix. With R, you can code as follow:

accuracy_Test1 <- sum(diag(table_mat)) / sum(table_mat)

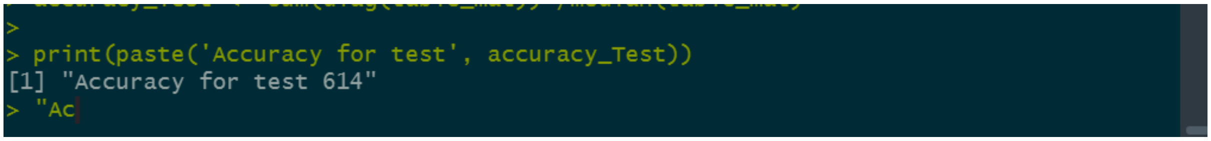
Code Explanation5

- Sum of the (diagonal (table_mat is our variable)): Sum of the diagonal
- Sum of the variable(table_mat): Sum of the matrix.

You can print the accuracy of the test set:

print(paste('Test_Accuracy', accuracy_Test)1)

Output: "Accuracy of Test 61%"



```
>
> print(paste('Accuracy for test', accuracy_Test))
[1] "Accuracy for test 614"
> "Ac
```

6.4. Random Forest

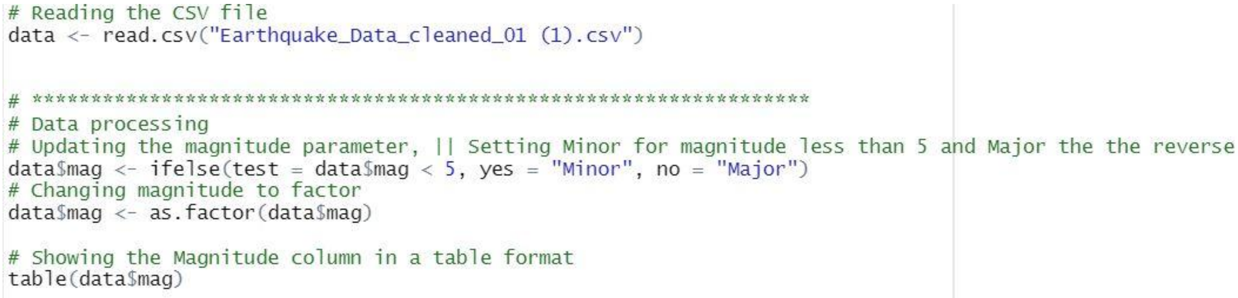
Random forest is a tree-based machine learning algorithm that is implemented to resolve regression and classification problems. As the name suggests, it is a technique that consists of many decision trees. Each node in the decision trees contributes to generating the final output of the random forest.

Advantages of RF:

1. Can be used for Classification or Regression.
2. Avoids Overfitting.
3. Can deal with a large number of features.
4. Works importance based feature selection.

6.4.1. Data processing

The first step is to read the CSV file and make the required changes in the data before sending it to the next process.

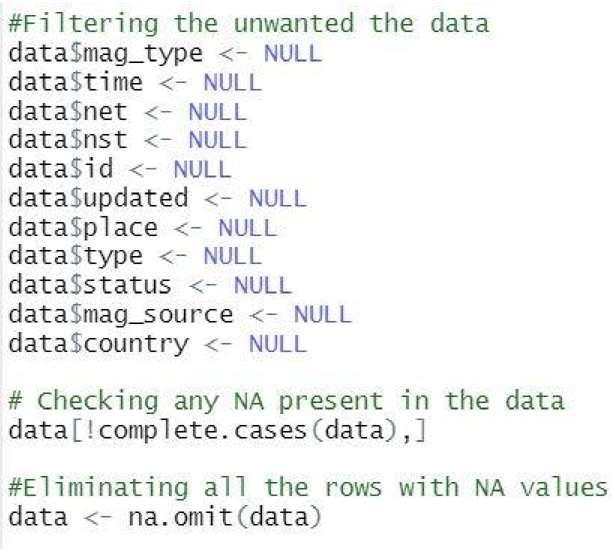


```
# Reading the CSV file
data <- read.csv("Earthquake_Data_cleaned_01 (1).csv")


# ******************************************************************
# Data processing
# Updating the magnitude parameter, || Setting Minor for magnitude less than 5 and Major the the reverse
data$mag <- ifelse(test = data$mag < 5, yes = "Minor", no = "Major")
# Changing magnitude to factor
data$mag <- as.factor(data$mag)

# Showing the Magnitude column in a table format
table(data$mag)
```

Output:

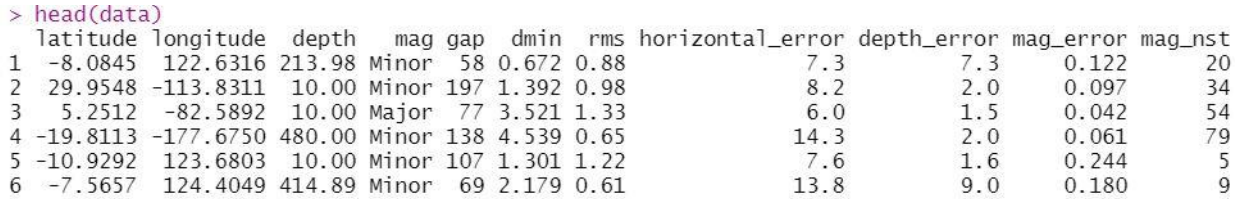
```
Major   Minor
3365    11383
```

As the random forest function does not allow using string values and (NA) for processing, so filtering out the unwanted data.



```
#Filtering the unwanted the data
data$mag_type <- NULL
data$time <- NULL
data$net <- NULL
data$nst <- NULL
data$id <- NULL
data$updated <- NULL
data$place <- NULL
data$type <- NULL
data$status <- NULL
data$mag_source <- NULL
data$country <- NULL

# Checking any NA present in the data
data[!complete.cases(data),]

#Eliminating all the rows with NA values
data <- na.omit(data)
```

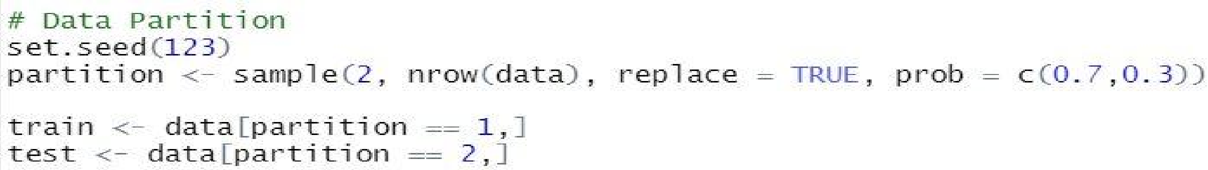
Output after filtering:



```
> head(data)
  latitude longitude  depth   mag gap  dmin  rms horizontal_error depth_error mag_error mag_nst
1  -8.0845  122.6316 213.98 Minor  58 0.672 0.88              7.3         7.3     0.122      20
2  29.9548 -113.8311  10.00 Minor 197 1.392 0.98              8.2         2.0     0.097      34
3   5.2512  -82.5892  10.00 Major  77 3.521 1.33              6.0         1.5     0.042      54
4 -19.8113 -177.6750 480.00 Minor 138 4.539 0.65             14.3         2.0     0.061      79
5 -10.9292  123.6803  10.00 Minor 107 1.301 1.22              7.6         1.6     0.244       5
6  -7.5657  124.4049 414.89 Minor  69 2.179 0.61             13.8         9.0     0.180       9
```

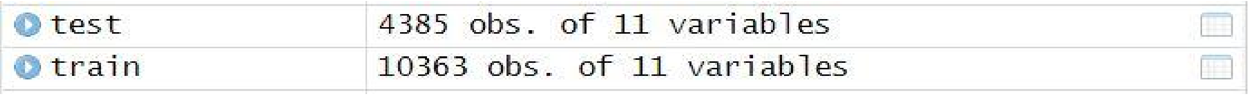
6.4.2. Data Partition

Splitting the processed data set into two data sets in a 70:30 format. 70% of the data goes to the training dataset and the remaining goes in 2nd dataset which will be used for prediction, it is named as testing dataset.



```
# Data Partition
set.seed(123)
partition <- sample(2, nrow(data), replace = TRUE, prob = c(0.7,0.3))

train <- data[partition == 1,]
test <- data[partition == 2,]
```

Output:



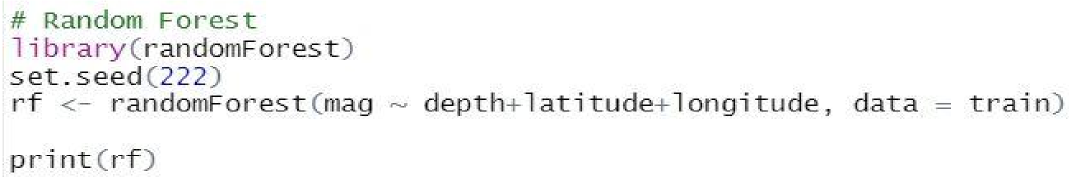
test	4385 obs. of 11 variables
train	10363 obs. of 11 variables

6.4.3. STEPs of Algorithm

1. Draw ntree bootstrap samples.
2. For each bootstrap sample, grown-pruned tree by choosing best split based on a random sample of mtry predictors at each node.
3. Predict new data using majority votes for classification and average for regression based on ntree trees.

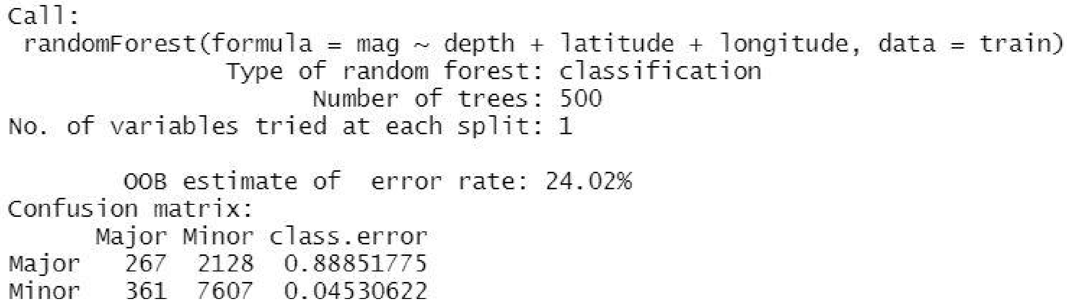
6.4.4. Random Forest Algorithm

For performing Random Forest in R, an inbuilt library is present called randomForest. The Train dataset is used to perform the RF. From the training dataset Magnitude(mag) is considered as a dependent variable and in this case depth, latitude and longitude are considered as the independent variables.



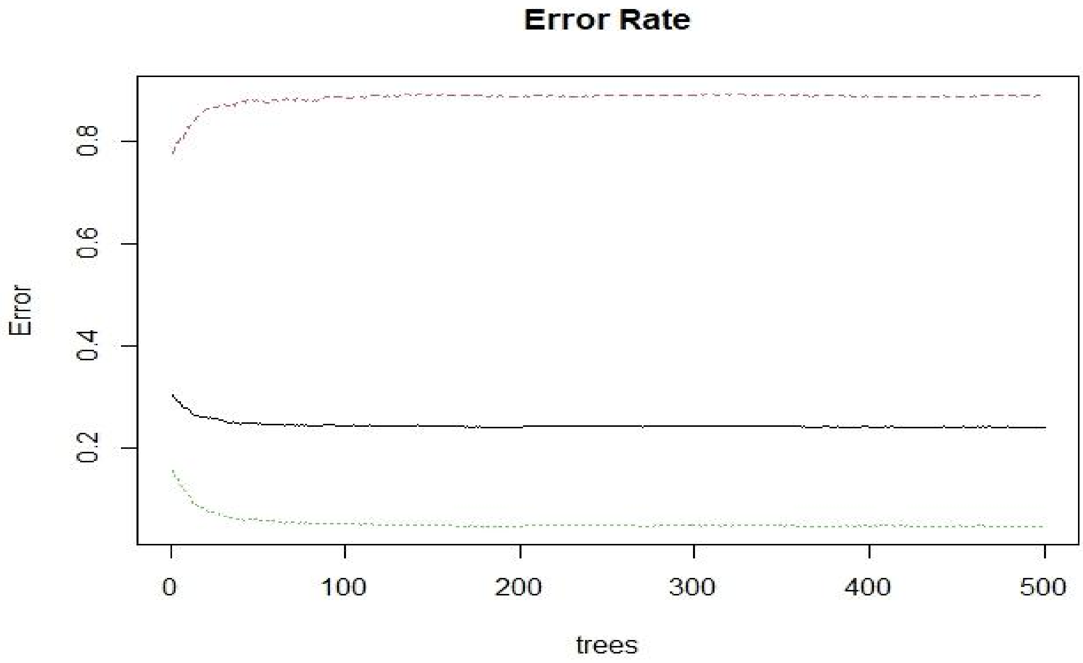
```
# Random Forest
library(randomForest)
set.seed(222)
rf <- randomForest(mag ~ depth+latitude+longitude, data = train)

print(rf)
```

Output:
By default the number of trees is set to 500



```
Call:
 randomForest(formula = mag ~ depth + latitude + longitude, data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 24.02%
Confusion matrix:
      Major Minor class.error
Major   267  2128  0.88851775
Minor   361  7607  0.04530622
```

Plotting the graph for the Error rate of the Random Forest model, here the in X-axis there are trees and in Y-axis there is error.



From the graph it is observed that the error is not improving after the 300 trees.

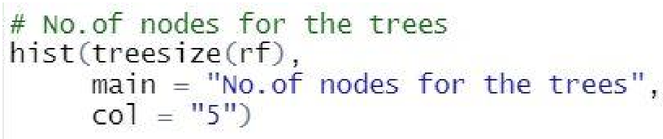
Random Forest Model Comparison

	model	TP	FN	TN	FP	Class. error 1	Class. error 2	OOB error rate
1.	100	7551	417	281	2114	0.88267223	0.05233434	24.42%
2.	**300**	**7606**	**362**	**264**	**2131**	**0.88977035**	**0.04543173**	**24.06%**
3.	500	7607	361	267	2128	0.88851775	0.04530622	24.02%
4.	900	7616	352	252	2143	0.89478079	0.04417671	24.08%

From the graph it can be seen that there is hardly any change in the data after 300 trees. Therefore the number of trees is set to 300.

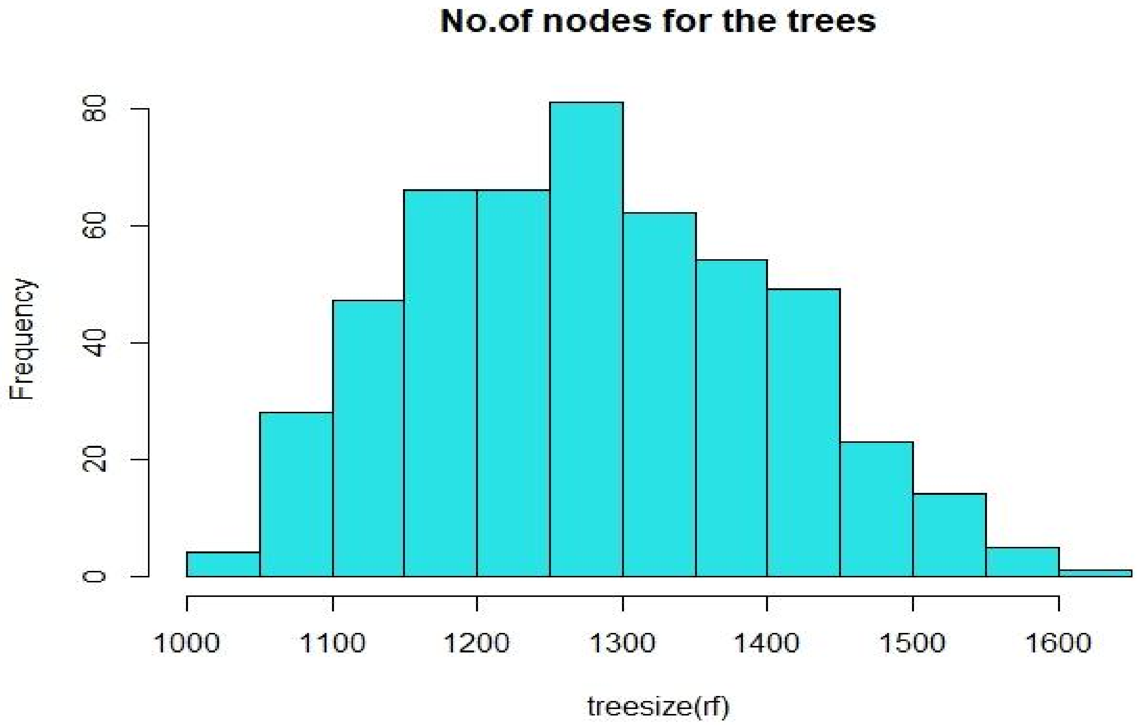
Tree Size VS Frequency

Here a geometric plot is computed to visualise the size of the trees in the RF. According to the graph it is observed that the majority of the trees lies between 1150 and 1350 size.



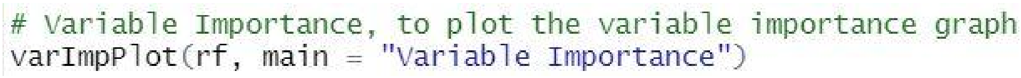
```
# No.of nodes for the trees
hist(treesize(rf),
     main = "No.of nodes for the trees",
     col = "5")
```

Output:



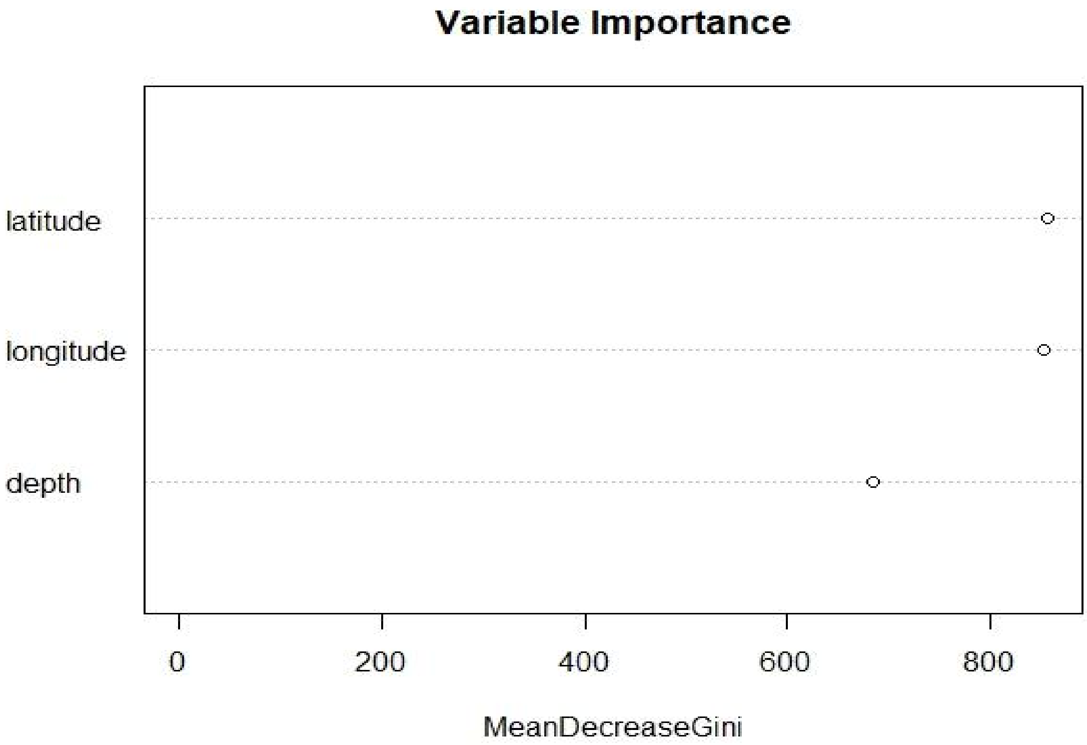
Variable Importance Graph

In this section the main aim is to find out the importance of each variable predicting the n desired output. The graph measures how pure the nodes are at the end of the tree without each variable.



```
# Variable Importance, to plot the variable importance graph
varImpPlot(rf, main = "Variable Importance")
```

Output:

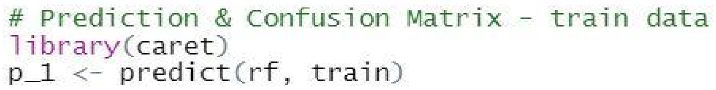


The observation from the graph is that the Latitude variable gets the highest importance for predicting the magnitude, and the Depth variable gets the least importance.

Prediction and Confusion Matrix

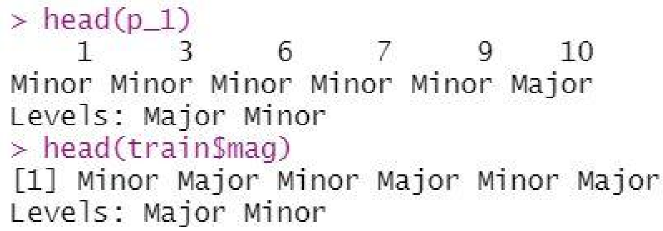
Prediction

After computing the Random Forest(RF), it is time to predict data to evaluate the accuracy of the RF. For appling the prediction algorithm, an inbuilt library is used, called Caret.



```
# Prediction & Confusion Matrix - train data
library(caret)
p_1 <- predict(rf, train)
```

For testing the accuracy of the model, it is observed against the original training dataset.

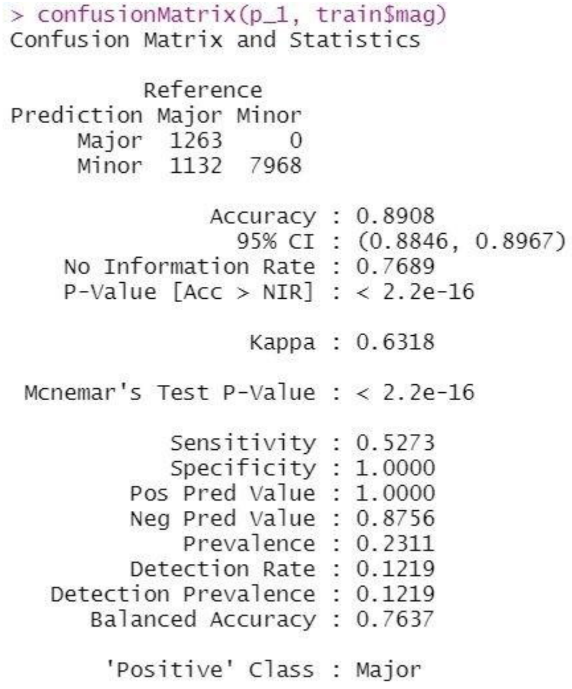


```
> head(p_1)
    1     3     6     7     9    10
Minor Minor Minor Minor Minor Major
Levels: Major Minor
> head(train$mag)
[1] Minor Major Minor Major Minor Major
Levels: Major Minor
```

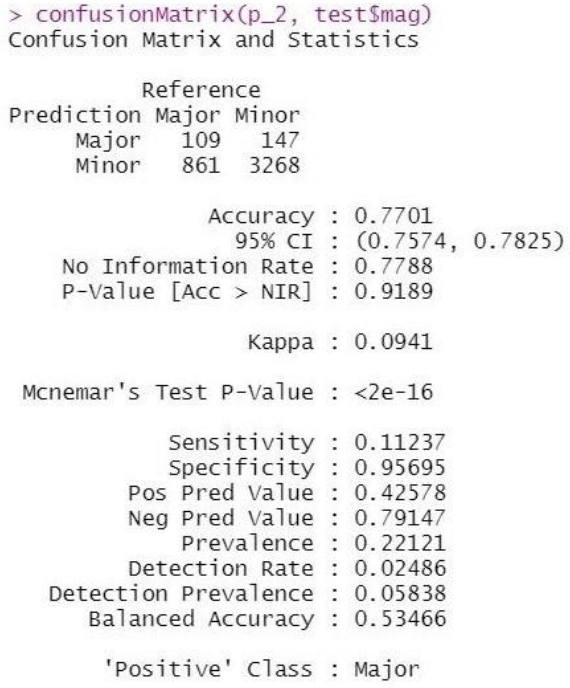
Here **p_1** is the predicted values and **train$mag** is the original values of the magnitude in the training dataset. It is observed that that 4 out of 6 initial data is being predicted correctly by the RF model.

Confusion Matrix

In addition to the prediction, the confusion matrix is computed for both the training and the testing datasets. And the data are as follows:



```
> confusionMatrix(p_1, train$mag)
Confusion Matrix and Statistics

          Reference
Prediction Major Minor
     Major  1263     0
     Minor  1132  7968

               Accuracy : 0.8908
                 95% CI : (0.8846, 0.8967)
    No Information Rate : 0.7689
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6318

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5273
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8756
             Prevalence : 0.2311
         Detection Rate : 0.1219
   Detection Prevalence : 0.1219
      Balanced Accuracy : 0.7637

       'Positive' Class : Major
```



```
> confusionMatrix(p_2, test$mag)
Confusion Matrix and Statistics

          Reference
Prediction Major Minor
     Major   109   147
     Minor   861  3268

               Accuracy : 0.7701
                 95% CI : (0.7574, 0.7825)
    No Information Rate : 0.7788
    P-Value [Acc > NIR] : 0.9189

                  Kappa : 0.0941

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.11237
            Specificity : 0.95695
         Pos Pred Value : 0.42578
         Neg Pred Value : 0.79147
             Prevalence : 0.22121
         Detection Rate : 0.02486
   Detection Prevalence : 0.05838
      Balanced Accuracy : 0.53466

       'Positive' Class : Major
```

Clearly, the values of accuracy for the training dataset is much higher than the testing dataset.

6.5. Unsupervised learning method:

In this part of the report, we will go through unsupervised learning method applied on the earthquake data-set in order to explore the geographical distribution of the data. First, we will explore the geographic components longitude and latitude of each observation in the data-set. Next, we will apply the K-means clustering method in order to cluster the earthquakes geographically to identify the hot-spots where we have high seismic activity in the world.

In this part we will consider the geographic location of each earthquakes in our data-set: taking in consideration the longitude and latitude of each event, we visualize all the earthquakes that happened in a map:

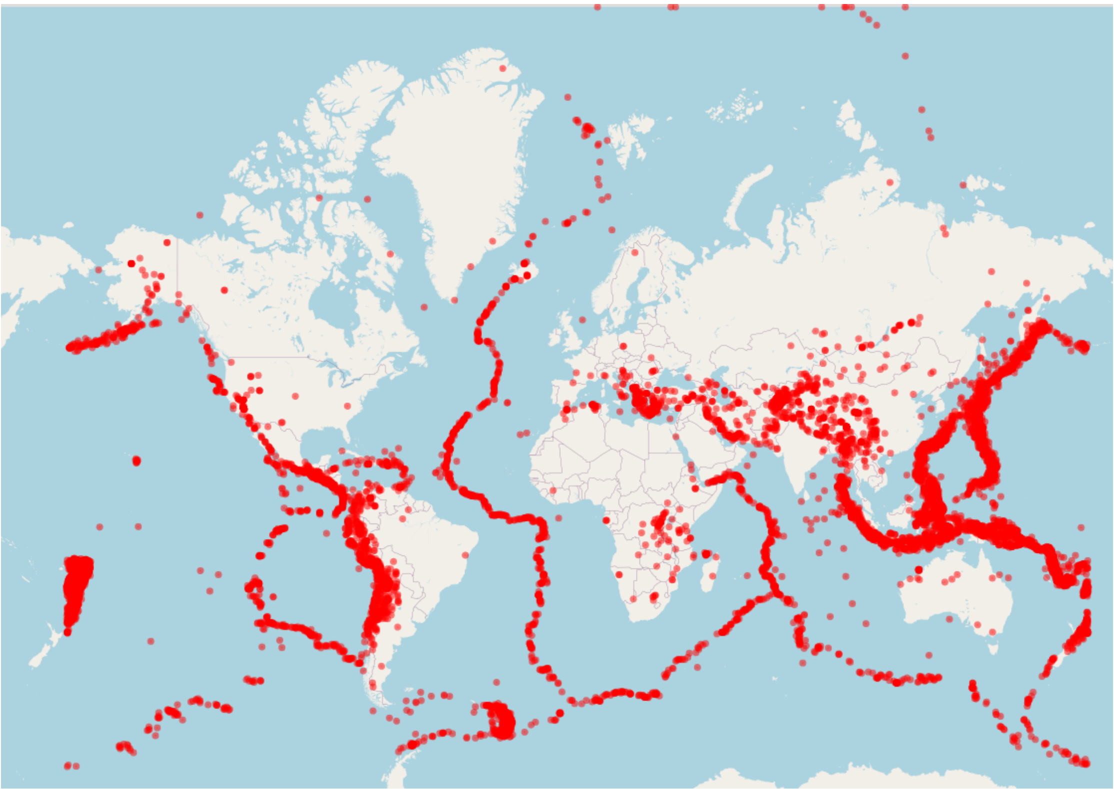


Fig 45

From this visualization, we can see that most of the earthquakes are located in south-east of Asia and South America, we observe an important seismic activity where tectonic plates meet.

This has been justified by Dr Ken Rubin who explained "Where plate pull apart, slide by each other or collide, there is tectonic activity manifested as earthquakes. The great majority of seismicity on the planet occurs at plate boundaries, although intraplate seismicity can occur as well when stresses build up in the plate."

6.5.1. K-means clustering:

Now, we want to study the geographical distribution of the earthquakes to find the hot spots in the globe with the highest number of seismic activities. For this, we are using the clustering algorithm K-means.

To have a first idea about the number of clusters we need to use, we first aggregate geographically our observations in a map using a visualization package. This will allow us to see globally where the most earthquakes are placed and to define the optimum number of clusters to consider for the k-means algorithm.

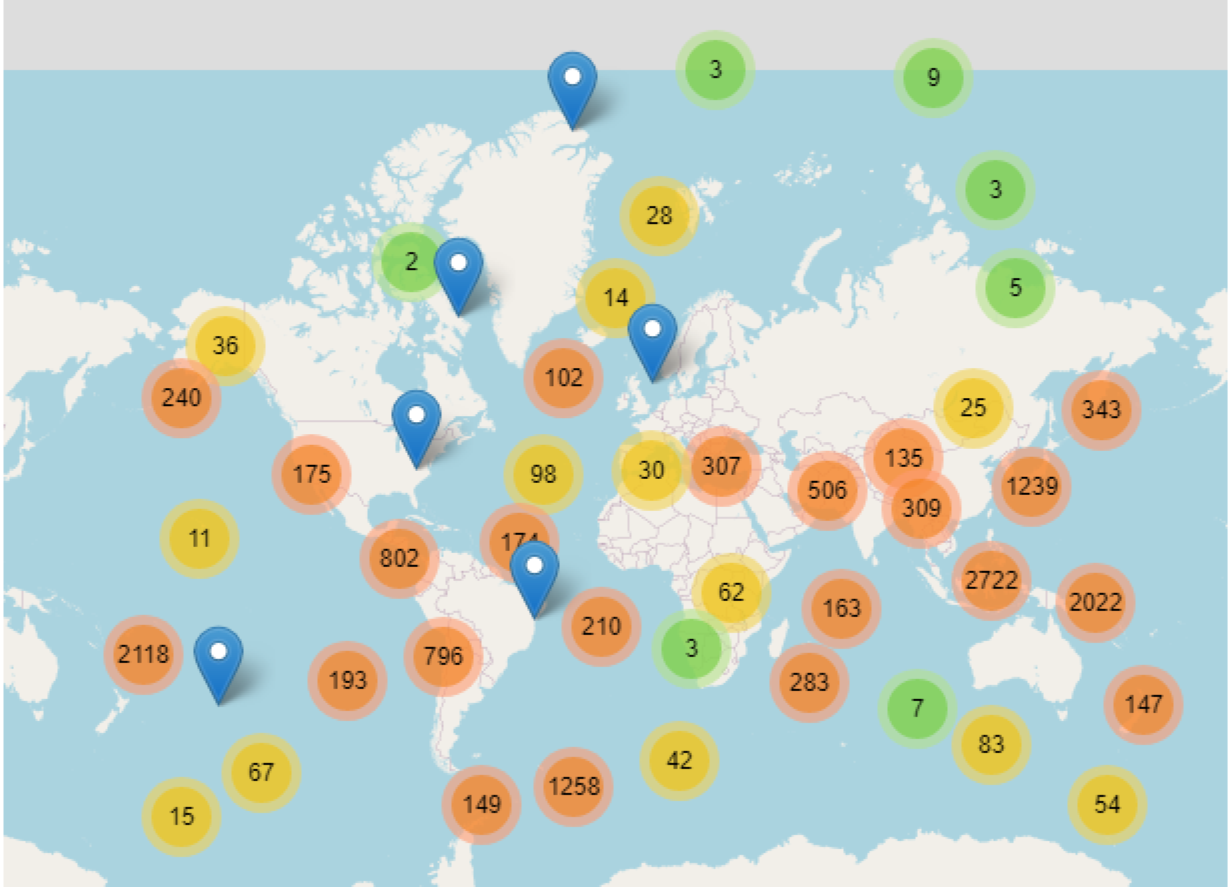


Fig 46

From our last visualization, we see that we ideally have around 50 clusters that can be observed around the globe in different parts.

Next, we conduct the K-means method using the longitude and latitude components as factors in the same scale. Following that we visualize the clusters centers on the map.

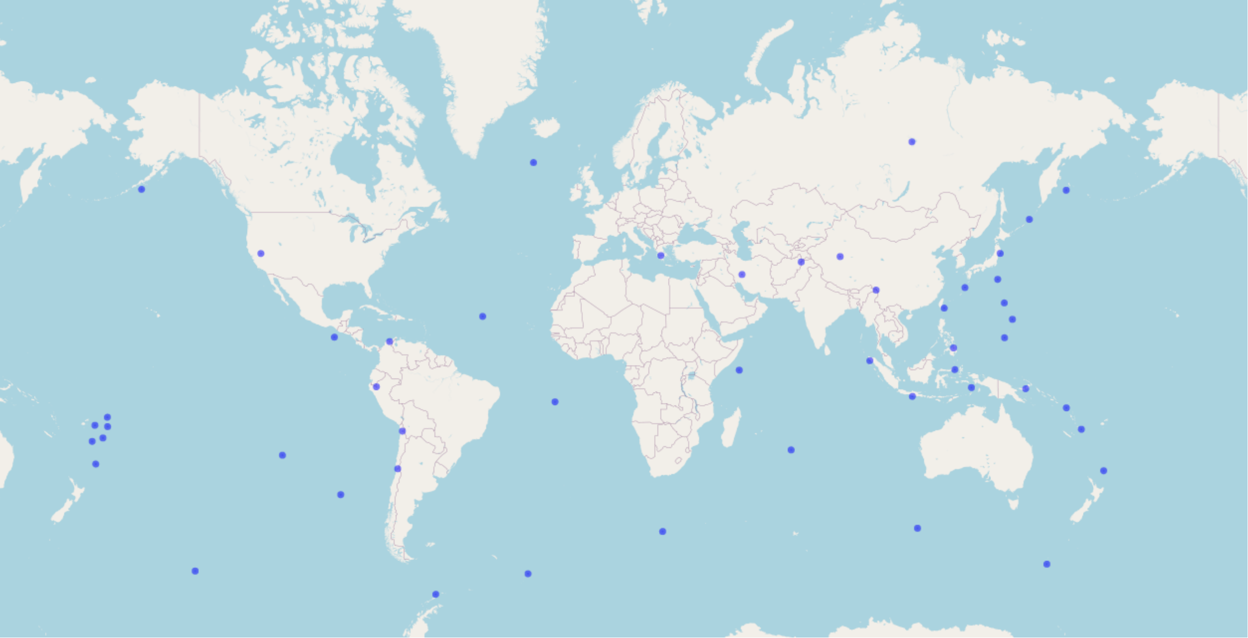
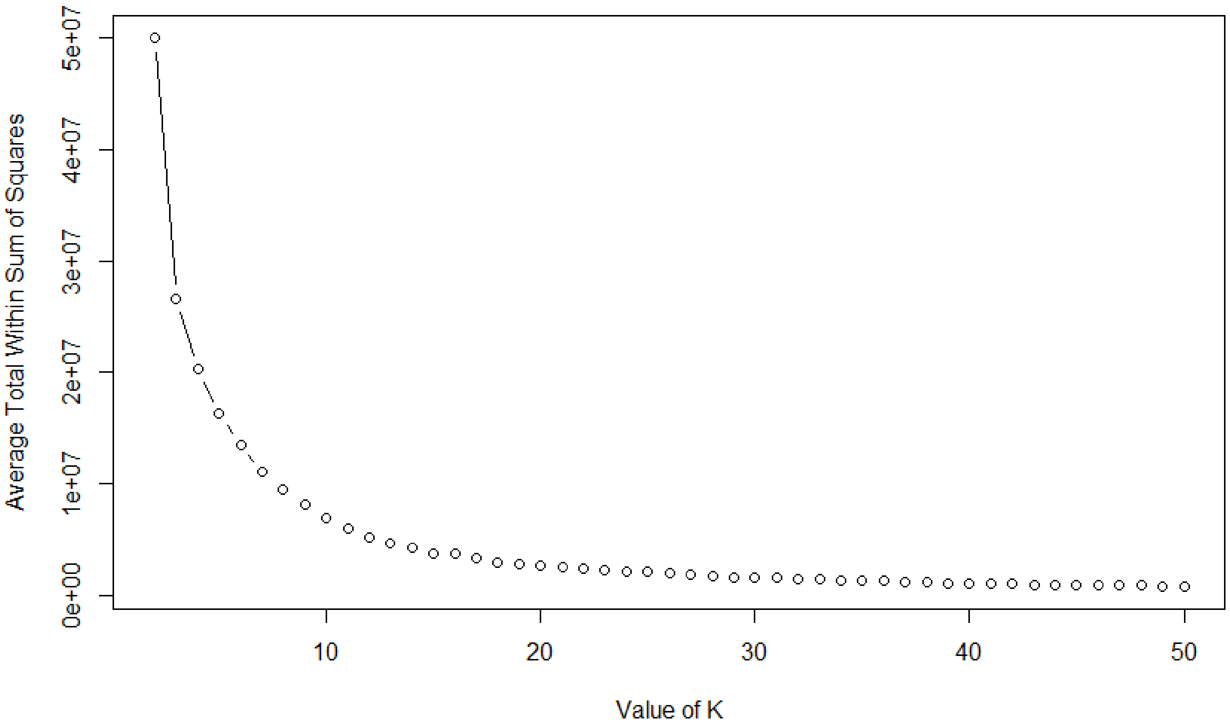


Fig 47

K-means clustering with 50 clusters of sizes 114, 216, 258, 232, 70, 204, 234, 196, 155, 282, 945, 103, 304, 239, 150, 147, 70, 638, 187, 401, 453, 130, 343, 407, 1290, 115, 246, 529, 104, 210, 337, 182, 281, 210, 473, 215, 786, 712, 304, 407, 46, 210, 375, 308, 209, 184, 144, 203, 183, 255.

In order to select the number of clusters k that better cluster our values (Minimize the Intra-clusters distance) we applied the k-means using a range of values for k values and plot the within cluster sum of squares.



We observe that taking the value of K equals 50 will give us minimum inter-cluster distance while keeping the number of clusters relatively manageable according to the visualization within the map.

7. Evaluation of Models

A. Linear Model Output:

India				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	3.9257231	0.443512	8.851	0.00000000000000702***
depth	-0.0013727	0.001127	-1.219	0.2253
latitude	0.0006946	0.003346	0.208	0.8359
longitude	0.0097262	0.00454	2.142	0.0341*
China				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	4.550785	0.631876	7.202	0.0000000000211***
depth	-0.009424	0.006248	-1.508	0.133
latitude	0.002612	0.007514	0.348	0.729
longitude	0.002501	0.004401	0.568	0.571
Indonesia				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	4.8041061	0.102913	46.681	<0.0000000000000002***
depth	0.0003903	0.000128	3.059	0.00228**
latitude	0.0002471	0.002512	0.098	0.92166
longitude	-0.0001522	0.000862	-0.177	0.85992
Pakistan				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	8.442034	2.34053	3.607	0.00135**
depth	-0.012632	0.005859	-2.156	0.04091*
latitude	0.086587	0.042732	2.026	0.05353
longitude	-0.08705	0.047364	-1.838	0.07799
Iran				
	Estimate Std.	**Error**	**t value**	**Pr(>|t|)**
(Intercept)	4.539583	0.943715	4.81	0.00000655***
depth	0.008698	0.008759	0.993	0.324
latitude	0.005013	0.013556	0.37	0.712
longitude	0.001063	0.013213	0.08	0.936

RMSE Values for Linear Model

Country	RMSE
INDIA	**0.2613795**
CHINA	0.26129025
INDONESIA	0.3501908
PAKISTAN	0.301303
IRAN	0.3503684

RMSE for Linear Model

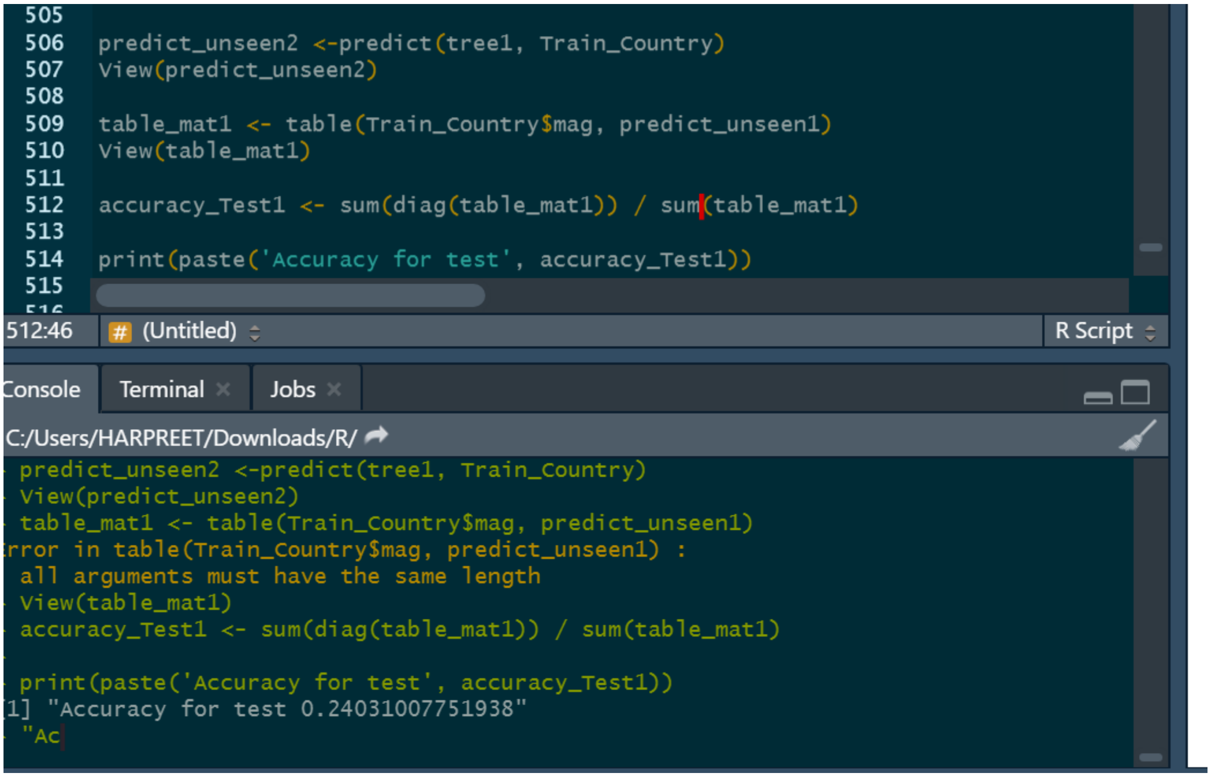
Multiple R-squared	F-statistic	p-value	Adjusted R-squared
0.0427	1.858 on 3 and 125 DF	0.1401	0.01972
0.01692	0.9293 on 3 and 162 DF	0.428	-0.001287
0.008785	3.132 on 3 and 1060	0.02489	0.00598
0.2208	2.362 on 3 and 25 DF	0.0954	0.1273
0.01378	0.3913 on 3 and 84 DF	0.7596	-0.02144

B. Logistic Model

Logistic Regression.					
Mean	**1Q**	**Median**	**3Q**	**Mode**	
-1.8189	0.6516	0.7438	0.7438	0.7438	
	Estimate Std.	**Error**	**zvalue**	**Pr(>|t|)**	
(Intercept)	1.32854	0.16067	8.269	<2e-16***	
countryIndia	0.11333	0.24727	0.458	0.647	
countryIndonesia	-0.1848	0.17086	-1.082	0.279	
countryPakistan	0.08853	0.42558	0.208	0.835	
Null devaince	**Residuals devaince**	**AIC**	**Number of Fisher Scoring iterations**	**P value of the model**	**R square value:**
2247.8 on 2071 degrees of freedom	2244.2 on 2068 degrees of freedom	2252.2	4	0.3051171	0.001611919

C. Decision Tree model:

Accuracy is 24%

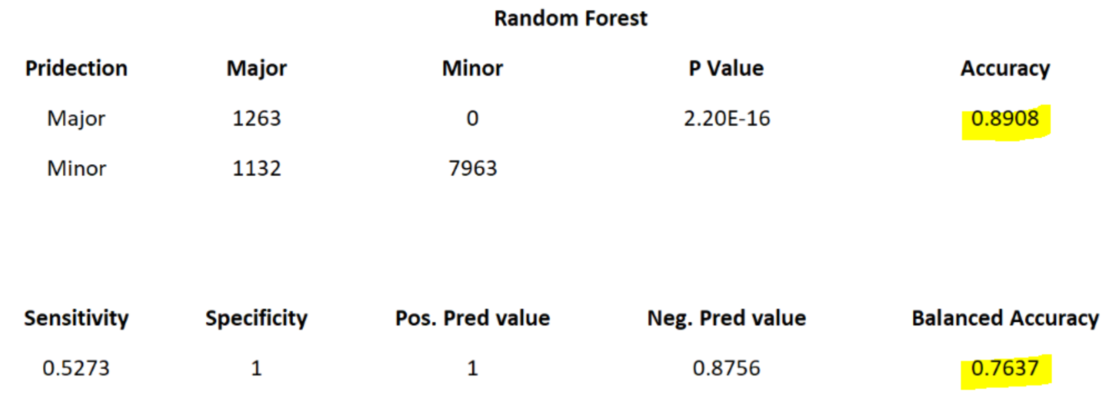


```
predict_unseen2 <-predict(tree1, Train_Country)
View(predict_unseen2)

table_mat1 <- table(Train_Country$mag, predict_unseen1)
View(table_mat1)

accuracy_Test1 <- sum(diag(table_mat1)) / sum(table_mat1)

print(paste('Accuracy for test', accuracy_Test1))
```

```
 predict_unseen2 <-predict(tree1, Train_Country)
 View(predict_unseen2)
 table_mat1 <- table(Train_Country$mag, predict_unseen1)
rror in table(Train_Country$mag, predict_unseen1) :
 all arguments must have the same length
 View(table_mat1)
 accuracy_Test1 <- sum(diag(table_mat1)) / sum(table_mat1)

 print(paste('Accuracy for test', accuracy_Test1))
1] "Accuracy for test 0.24031007751938"
 "Ac
```

D. Random Forest :

Prediction Major Minor P Value Accuracy
Major 1263 0 2.20E-16 0.8908
Minor 1132 7963
Sensitivity Specificity Pos. Pred value Neg. Pred value Balanced Accuracy
0.5273 1 1 0.8756 0.7637
Accuracy:



Random Forest

Pridection	Major	Minor	P Value	Accuracy
Major	1263	0	2.20E-16	0.8908
Minor	1132	7963		

Sensitivity	Specificity	Pos. Pred value	Neg. Pred value	Balanced Accuracy
0.5273	1	1	0.8756	0.7637

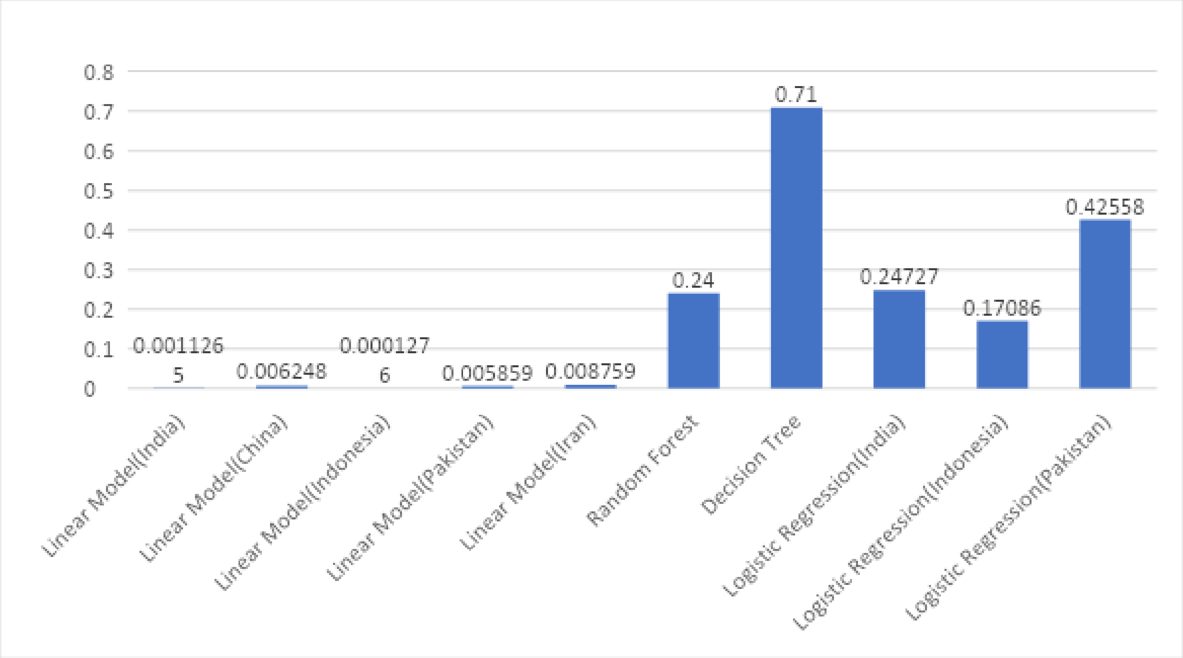
As Per our conclusion and above Graph representation that Linear Model is best fitted for our Dataset that we have selected. Although if we compare Random Forest and decision tree . Random Forest (89%) has more accuracy than Decision Tree(24%).

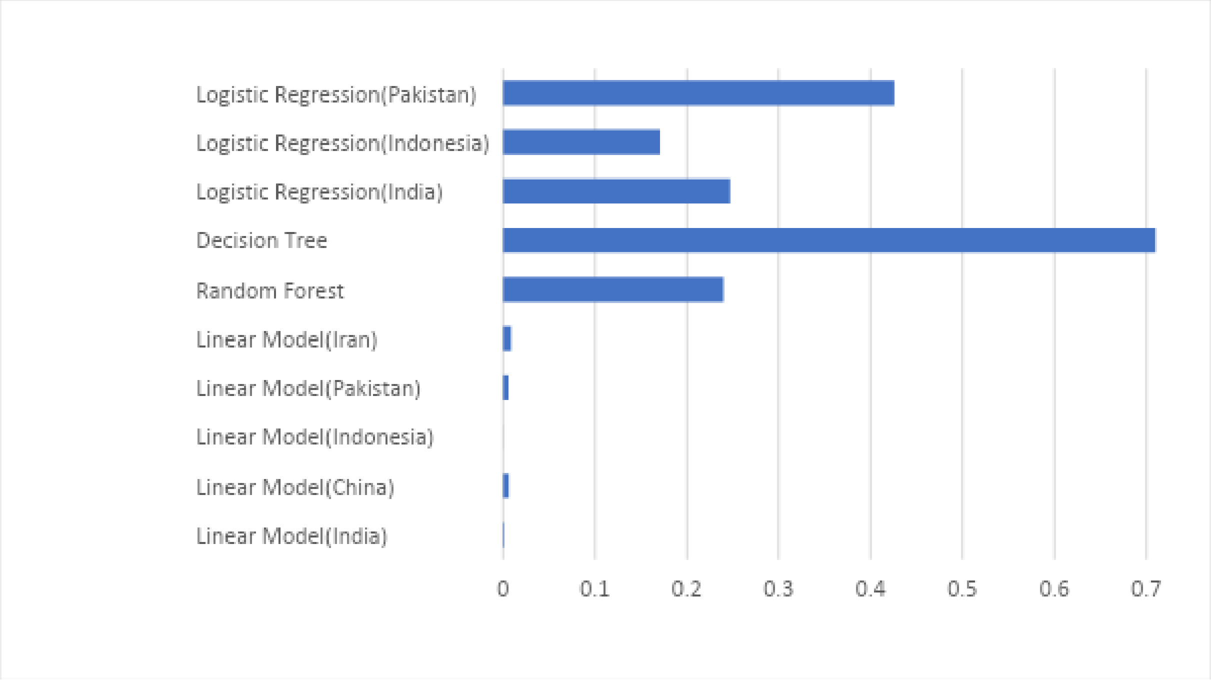
9. Conclusion and Future Prediction

After Comparing all the models we have found out that the Linear Regression model is the best suited for the Dataset and in terms of accuracy Random Forest suits the best. In the below table we have performed the error comparison and found out that the best fit model will be the Linear Model.

Error Comparison Between Models Implemented

Models Implemented	**Error**
Linear Model(India)	0.001127
Linear Model(China)	0.006248
Linear Model(Indonesia)	0.000128
Linear Model(Pakistan)	0.005859
Linear Model(Iran)	0.008759
Random Forest	0.24
Decision Tree	0.71
Logistic Regression(India)	0.24727
Logistic Regression(Indonesia)	0.17086
Logistic Regression(Pakistan)	0.42558





We can say that the magnitude of the Earthquake depends on various independent factors such as depth, Longitude, Latitude, CO_2 Emission. As per our analysis we have seen the strong correlation between the Earthquake(magnitude) with environmental factors such as CO_2, as shown in the different countries that we have taken. The countries that have the maximum CO_2 emission have more risk of getting earthquakes or high rate magnitudes.