Earthquake Sample Model

Nishant

Sample Data Collection Details

The sample data is taken by randomly selecting the region of south east asia covering the countries `Malaysia`, `Indomesia` etc.. The sample data taken over a period of one year from **10 Feb 2017** to **10 Feb 2018**. The data model used here may or may not be used in final model as it all depends on the structure and

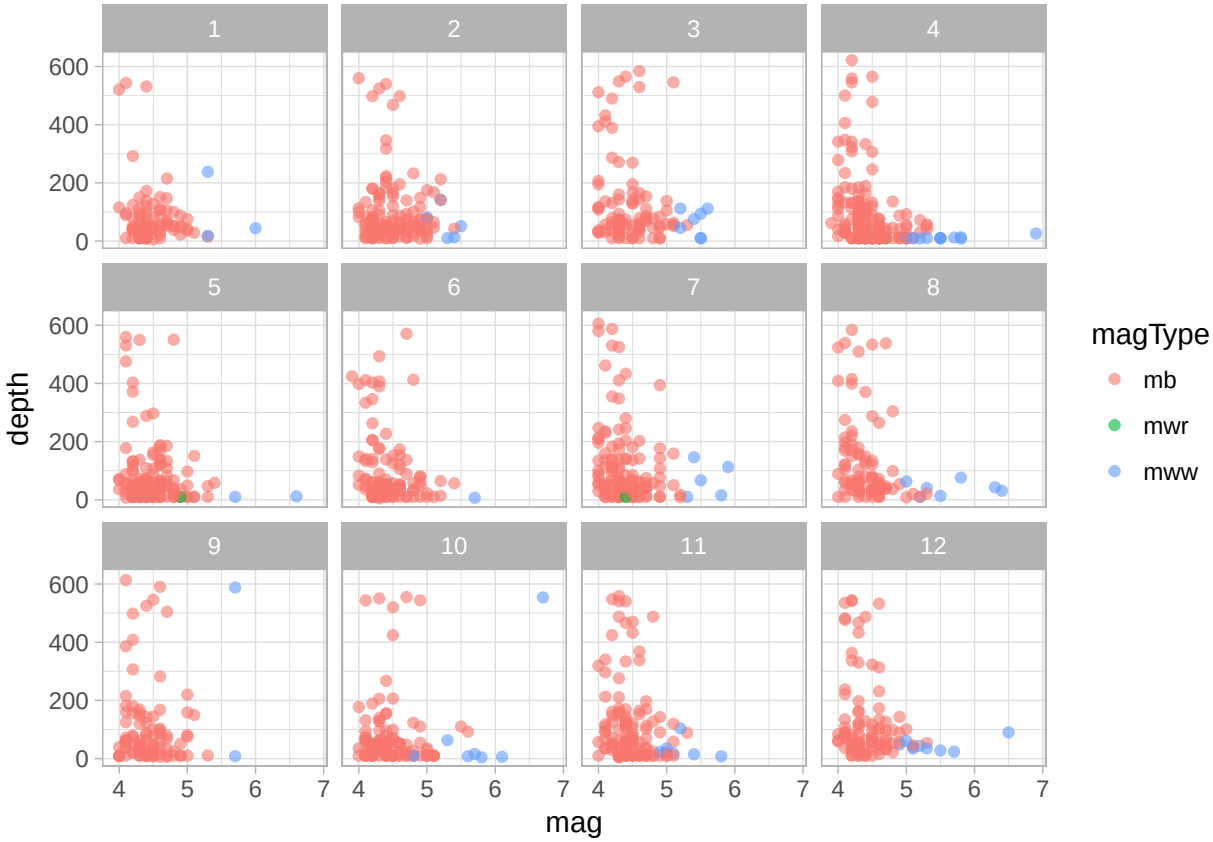
Data Overview

We will overview our data and remove unwanted or low variance features.

```
##      time              latitude          longitude          depth
##  Length:1369        Min.   :-11.7622   Min.   : 95.11   Min.   :  4.00
##  Class :character   1st Qu.: -7.0262   1st Qu.:112.94   1st Qu.: 24.61
##  Mode  :character   Median :  0.2684   Median :123.96   Median : 53.86
##                     Mean   : -1.1173   Mean   :119.21   Mean   :102.07
##                     3rd Qu.:  3.5413   3rd Qu.:126.79   3rd Qu.:121.11
##                     Max.   :  9.2681   Max.   :128.32   Max.   :621.70
##
##       mag          magType            nst               gap
##  Min.   :3.900   Length:1369        Mode:logical   Min.   : 13
##  1st Qu.:4.200   Class :character   NA's:1369      1st Qu.: 71
##  Median :4.400   Mode  :character                  Median : 95
##  Mean   :4.496                                     Mean   :101
##  3rd Qu.:4.600                                     3rd Qu.:128
##  Max.   :6.900                                     Max.   :324
##
##       dmin             rms             net                 id
##  Min.   : 0.008   Min.   :0.1300   Length:1369        Length:1369
##  1st Qu.: 1.310   1st Qu.:0.6700   Class :character   Class :character
##  Median : 2.063   Median :0.8300   Mode  :character   Mode  :character
##  Mean   : 2.327   Mean   :0.8528
##  3rd Qu.: 2.928   3rd Qu.:1.0300
##  Max.   :56.265   Max.   :1.6400
##
##    updated             place               type           horizontalError
##  Length:1369        Length:1369        Length:1369        Min.   : 2.400
##  Class :character   Class :character   Class :character   1st Qu.: 6.500
##  Mode  :character   Mode  :character   Mode  :character   Median : 7.900
##                                                           Mean   : 8.228
##                                                           3rd Qu.: 9.500
##                                                           Max.   :29.300
##
##    depthError        magError          magNst          status
##  Min.   : 0.600   Min.   :0.0340   Min.   :  1.00   Length:1369
##  1st Qu.: 3.600   1st Qu.:0.0850   1st Qu.: 13.00   Class :character
##  Median : 7.000   Median :0.1160   Median : 20.00   Mode  :character
##  Mean   : 6.678   Mean   :0.1214   Mean   : 30.99
##  3rd Qu.: 8.900   3rd Qu.:0.1480   3rd Qu.: 37.00
```

```
##  Max.   :33.700   Max.   :0.5320   Max.   :269.00
##                   NA's   :1        NA's   :1
##  locationSource      magSource
##  Length:1369        Length:1369
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

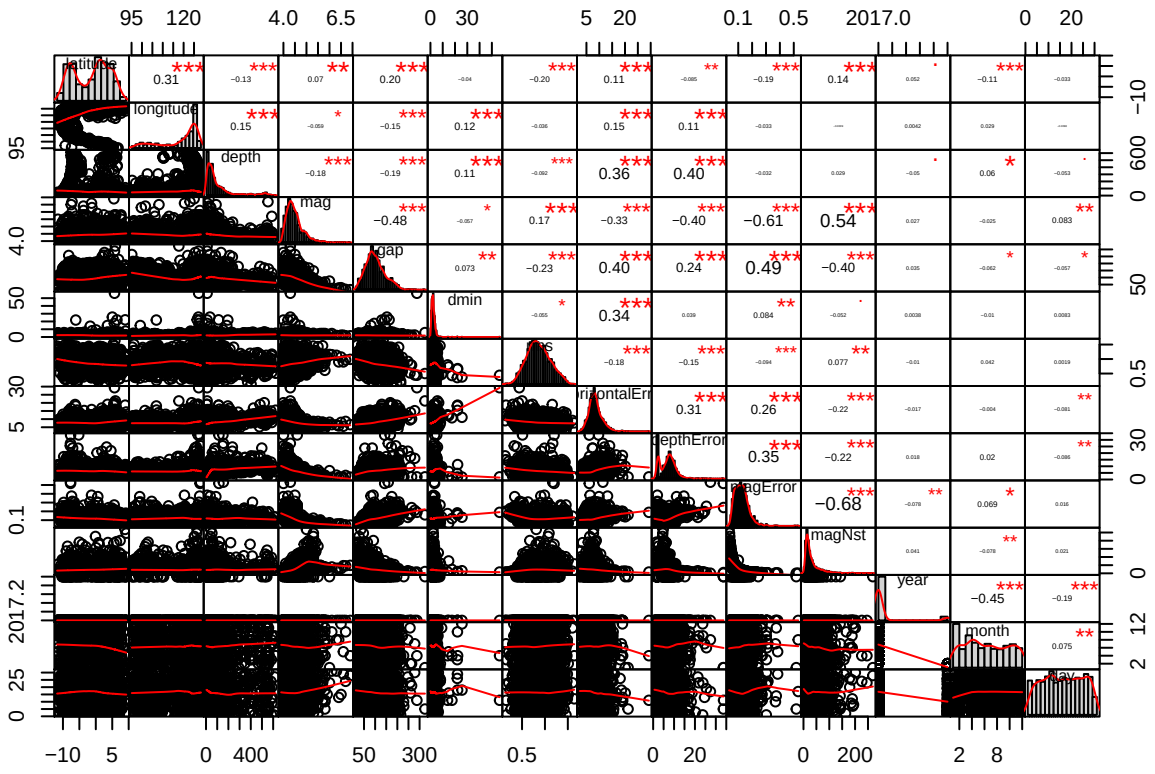
```
## [1] "character"
```



Correlation of numeric Data

We observe the correlation of numeric data.

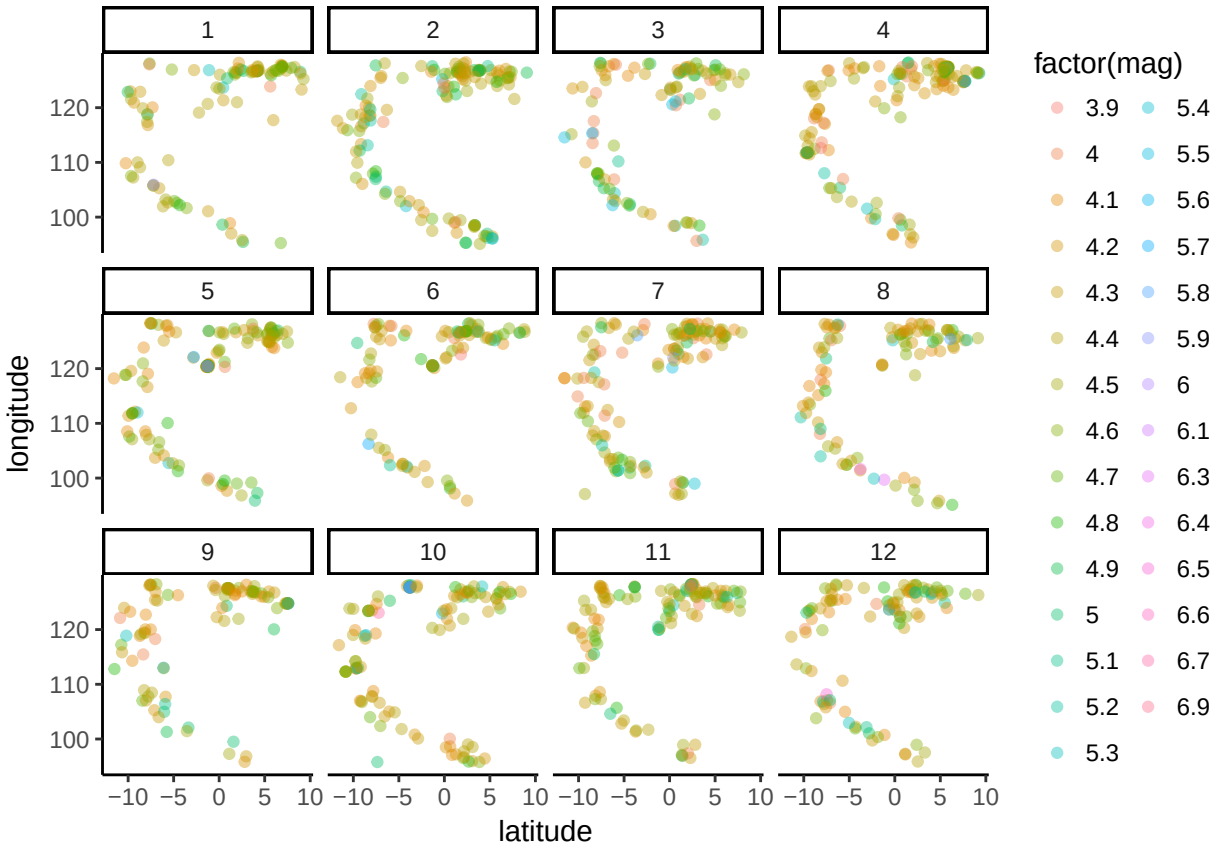
```
numClean = eqClean[, c(-1, -6, -10, -11)]

chart.Correlation(numClean)
```



Some random plots

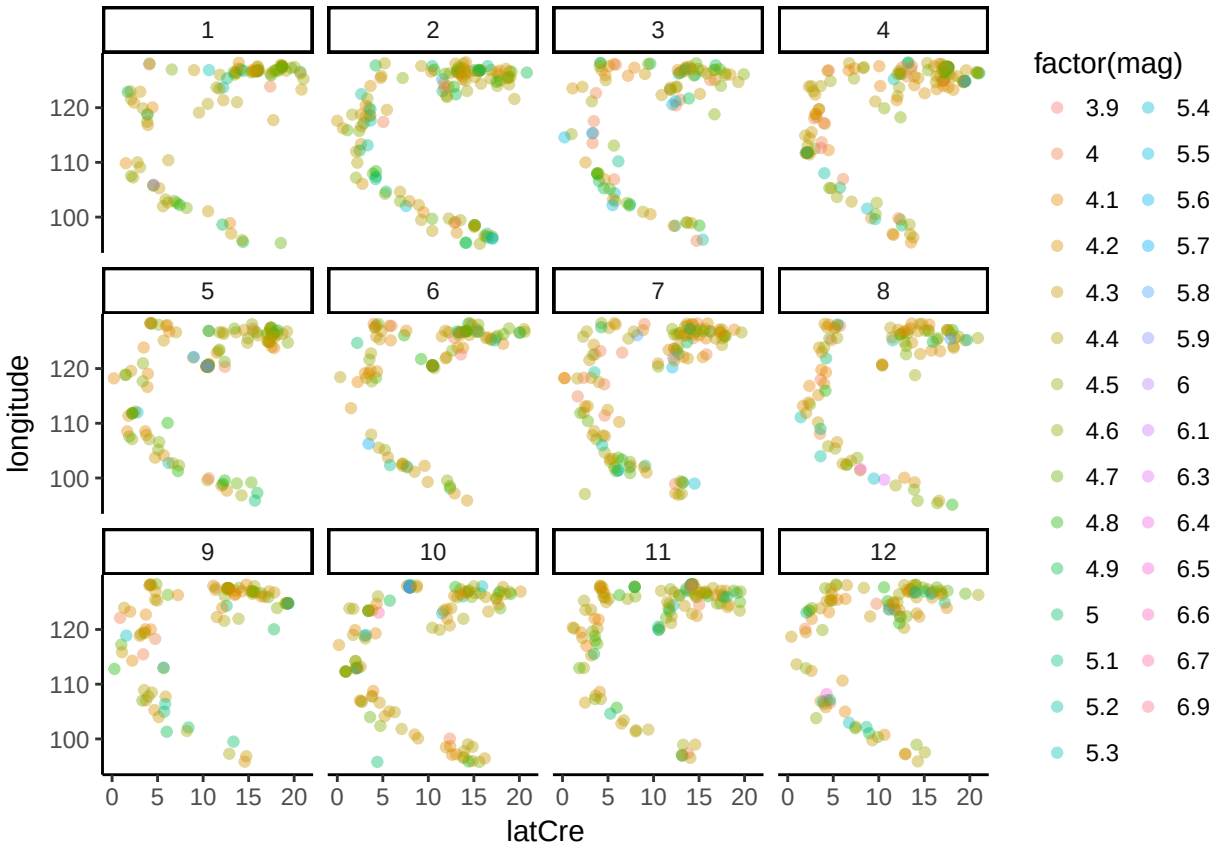
Lets plot some random scatter plots with some fetaure engineering for better understanding of the for now I will go with latitude and longitude and see the behaviour.Some feature engineering is encorporated but it soes not change the characterstics of the plot.

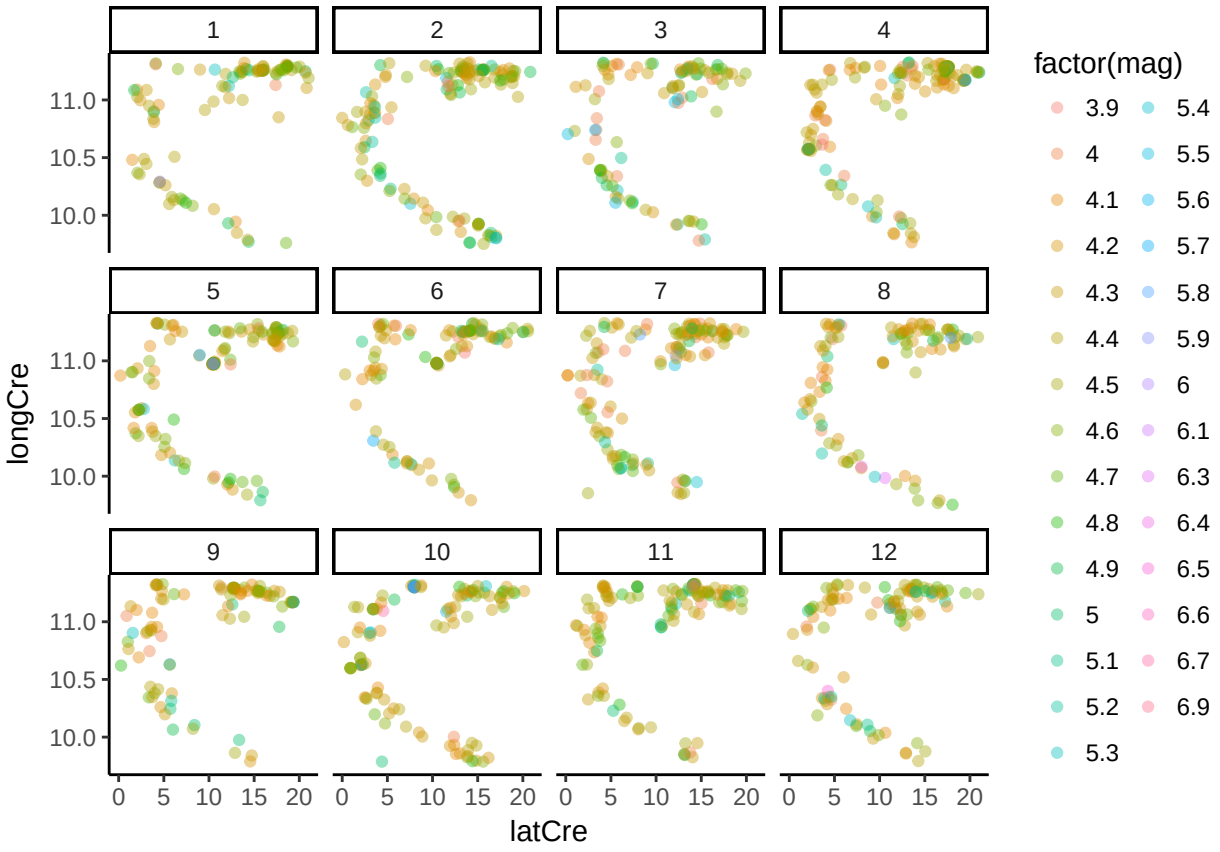
> *Note: In every plot some changes applied on one of the plotted variable but it does not change the behaviour.*



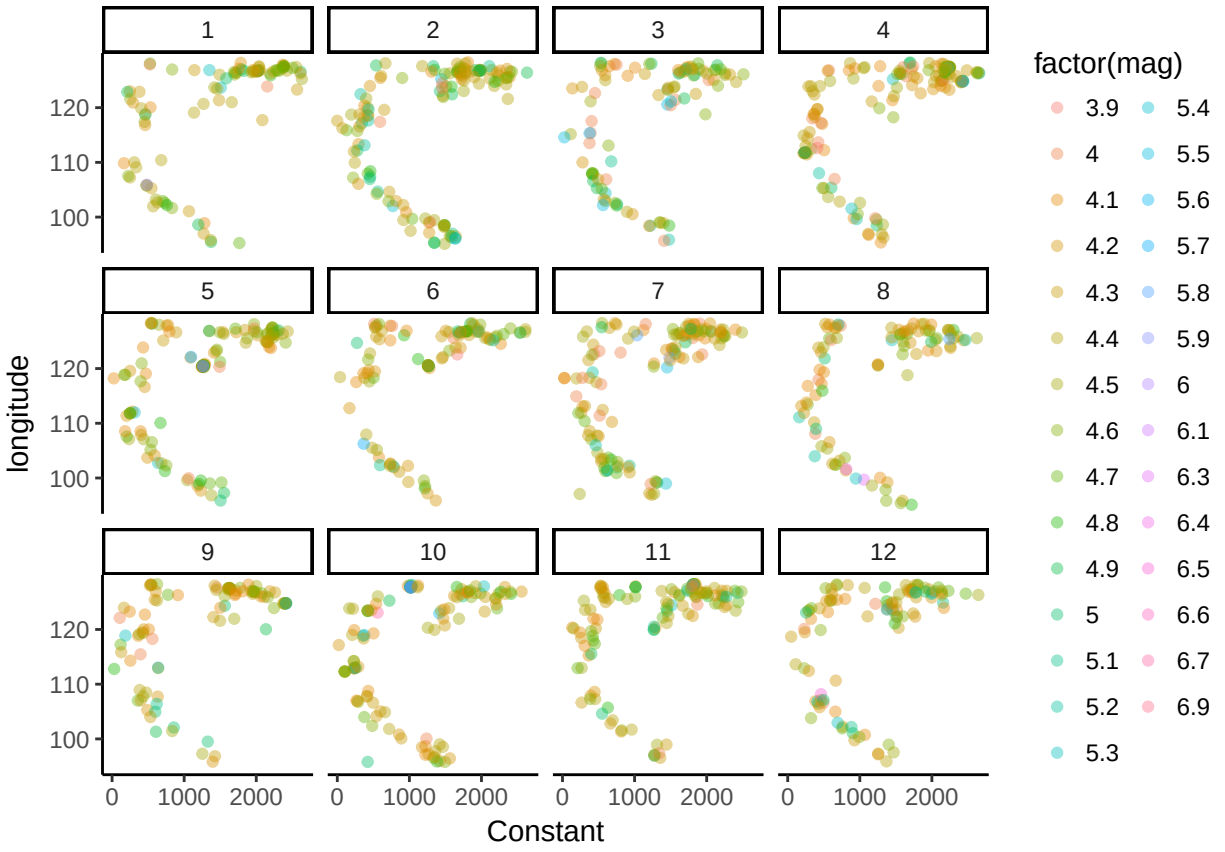

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -11.7622  -7.0262    0.2684   -1.1173    3.5413    9.2681
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.736  12.031  10.645  15.303  21.030
```


1
2
3
4
5
6
7
8
9
10
11
12
longitude
120
110
100
0
5
10
15
20
latCre
factor(mag)
3.9
4
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
5.5
5.6
5.7
5.8
5.9
6
6.1
6.3
6.4
6.5
6.6
6.7
6.9




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   560.9  1351.2  1286.5  1889.7  2655.9
```



Fitting Data in SVR(Support Vector Regression)

From the plots we can visualise the fault line and clearly see the occurences of earthquakes in the region, the fiiting is to find the line with higest risk.

```
#data for svr
dataset_all = eqClean[ , c(2, 3, 4, 5, 17)]
dataset = dataset_all[ , c(1, 2)]

#fitting the fault Line

library(e1071)
regressor = svm(formula = latitude ~ .,
                data = dataset,
                type = 'eps-regression',
                kernel = 'radial')

summary(regressor)
```

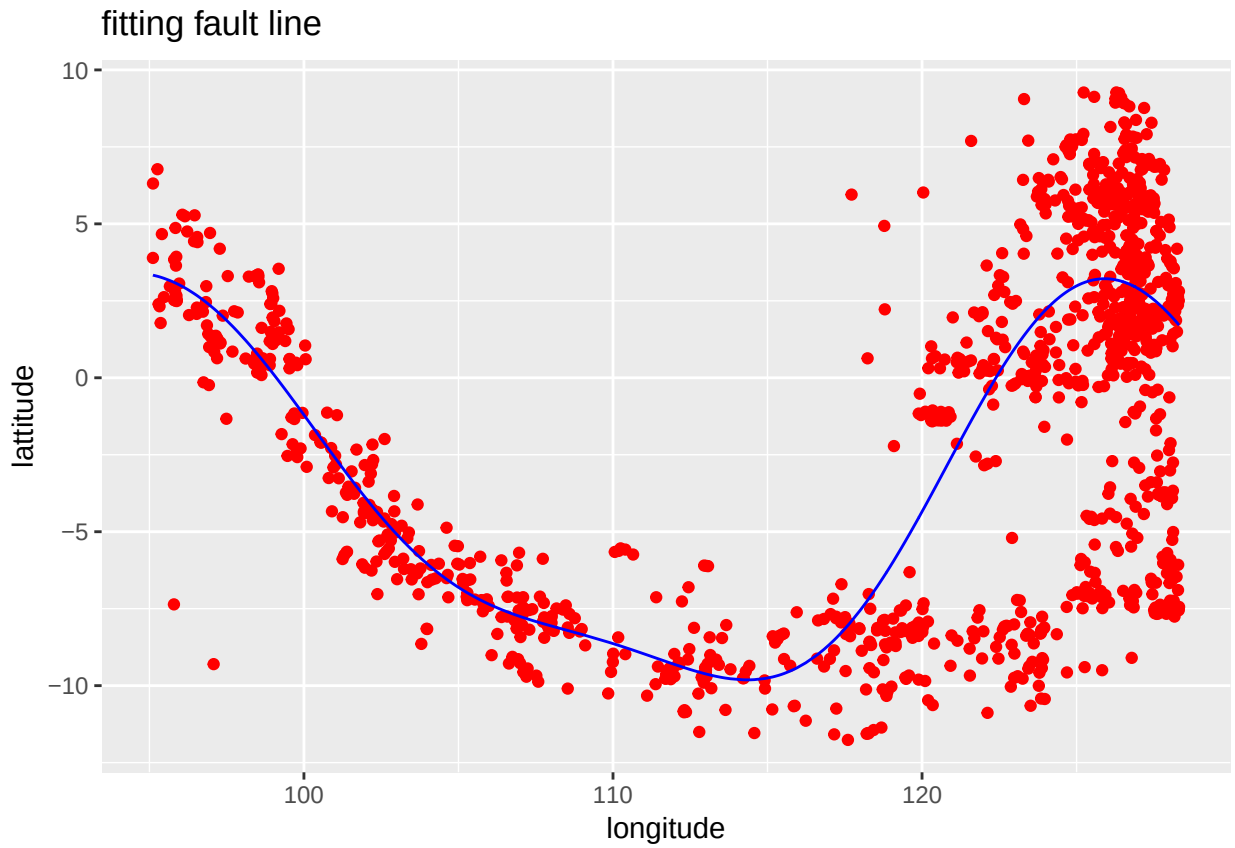
```
##
## Call:
## svm(formula = latitude ~ ., data = dataset, type = "eps-regression",
##     kernel = "radial")
##
##
## Parameters:
```

```
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  1120
```

```
names(regressor)
```

```
##  [1] "call"            "type"            "kernel"
##  [4] "cost"            "degree"          "gamma"
##  [7] "coef0"           "nu"              "epsilon"
## [10] "sparse"          "scaled"          "x.scale"
## [13] "y.scale"         "nclasses"        "levels"
## [16] "tot.nSV"         "nSV"             "labels"
## [19] "SV"              "index"           "rho"
## [22] "compprob"        "probA"           "probB"
## [25] "sigma"           "coefs"           "na.action"
## [28] "fitted"          "decision.values" "residuals"
## [31] "terms"
```

```
y_pred = predict(regressor, newdata = dataset)

# plotting real and predicted

ggplot() +
  geom_point(aes(x = dataset$longitude, y = dataset$latitude),
             colour = 'red') +
  geom_line(aes(x = dataset$longitude, y = predict(regressor, newdata = dataset)),
            colour = 'blue') +
  ggtitle('fitting fault line') +
  xlab('longitude') +
  ylab('lattitude')
```



The curve fitted accurately upto a certain point. In the plot we can see requirement of two models.

Dividing the data in two subsets

The dataset will be divided into two subsets, the point is taken from where the model does not fit.

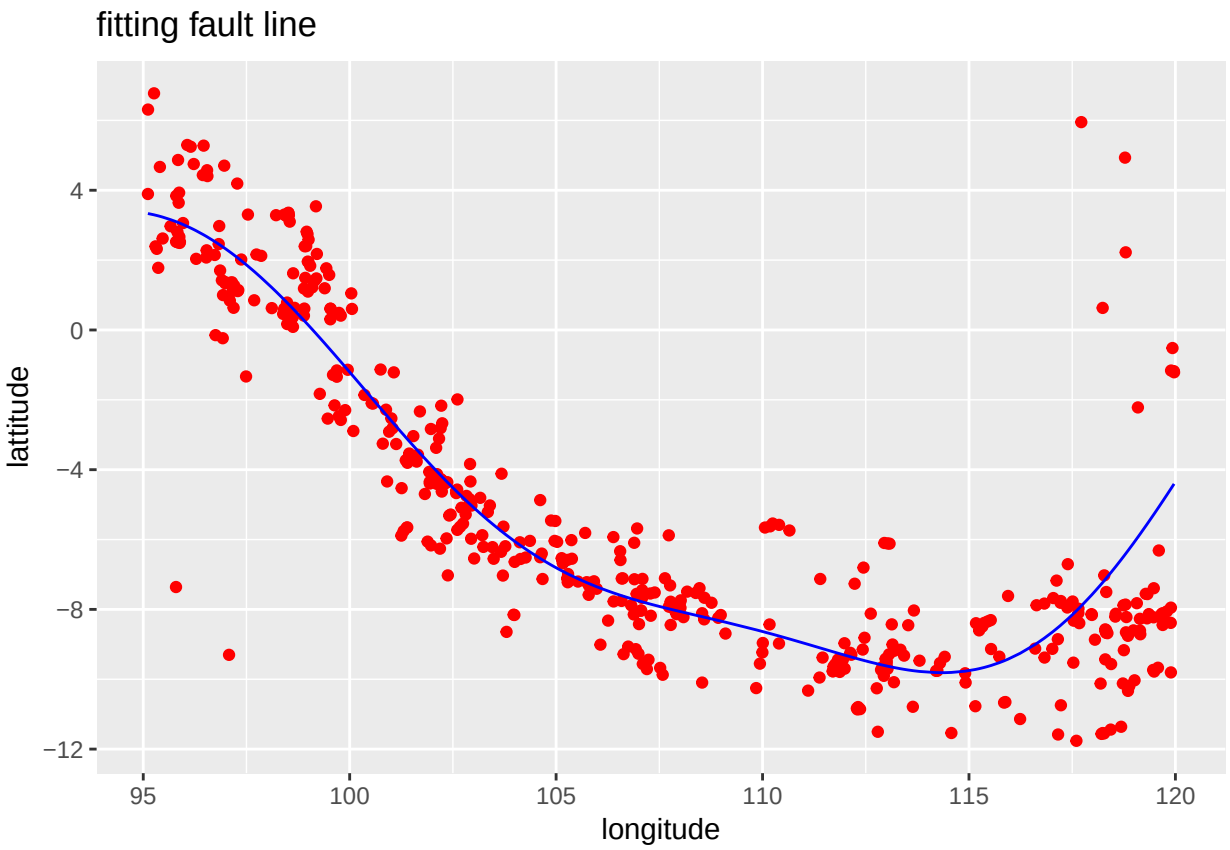
```
# dividing the data to have a better view of the fault line

lon_indx = which(dataset$longitude <= 120)
dataset_cut = dataset[lon_indx, ]

#looking into second index of the data

dataset_cut1 = dataset[-lon_indx, ]
```

Fitting data in First Dataset

The data now fitted more accurately few data points are again observed deviating the curve, right now the points will be considered for better understanding over large dataset.

```
#Plotting the cut dataset

ggplot() +
  geom_point(aes(x = dataset_cut$longitude, y = dataset_cut$latitude),
             colour = 'red') +
```

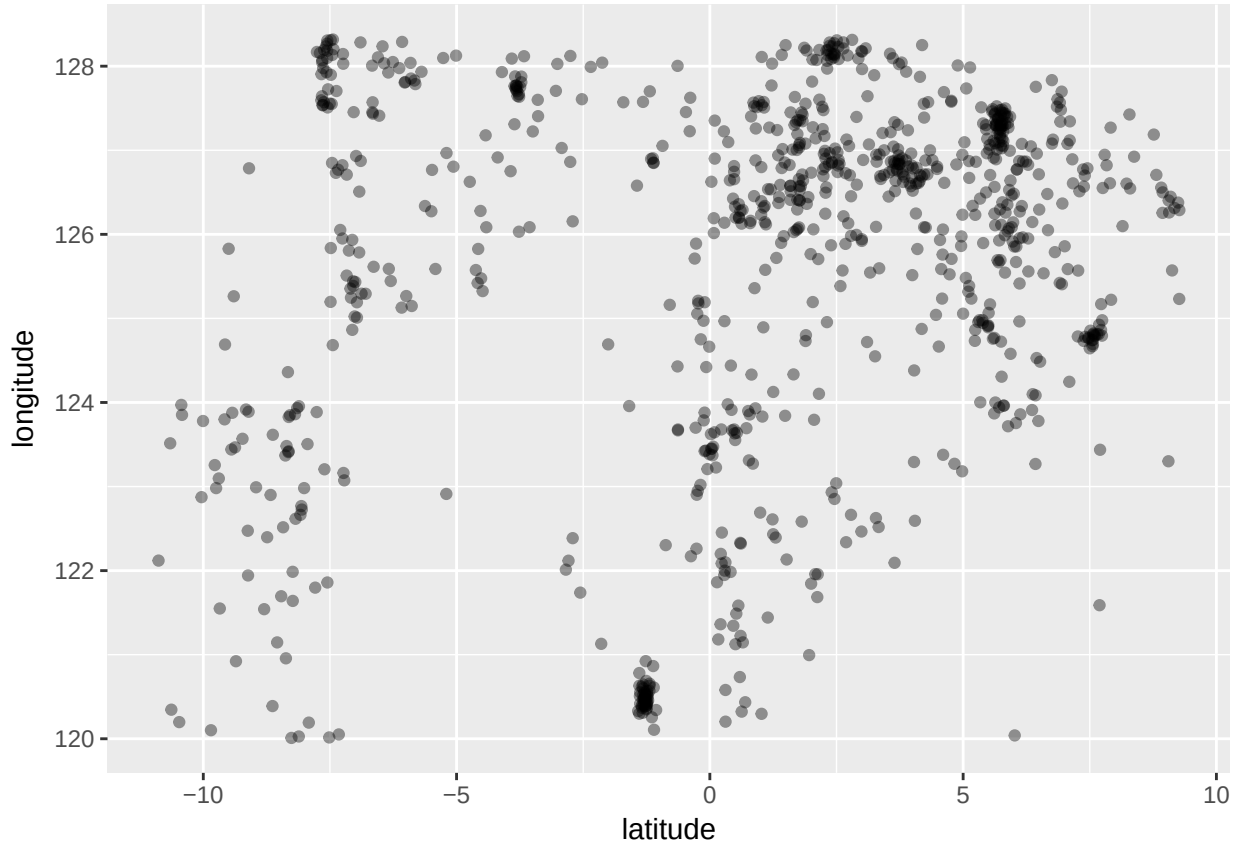
```
  geom_line(aes(x = dataset_cut$longitude, y = predict(regressor, newdata = dataset_cut)),
            colour = 'blue') +
  ggtitle('fitting fault line') +
  xlab('longitude') +
  ylab('lattitude')
```



Fitting data in Second Dataset

The data forming clusters on first look it feels to form 5 clusters so we tend to use kmean to cluster the data and identify the centers of the cluster

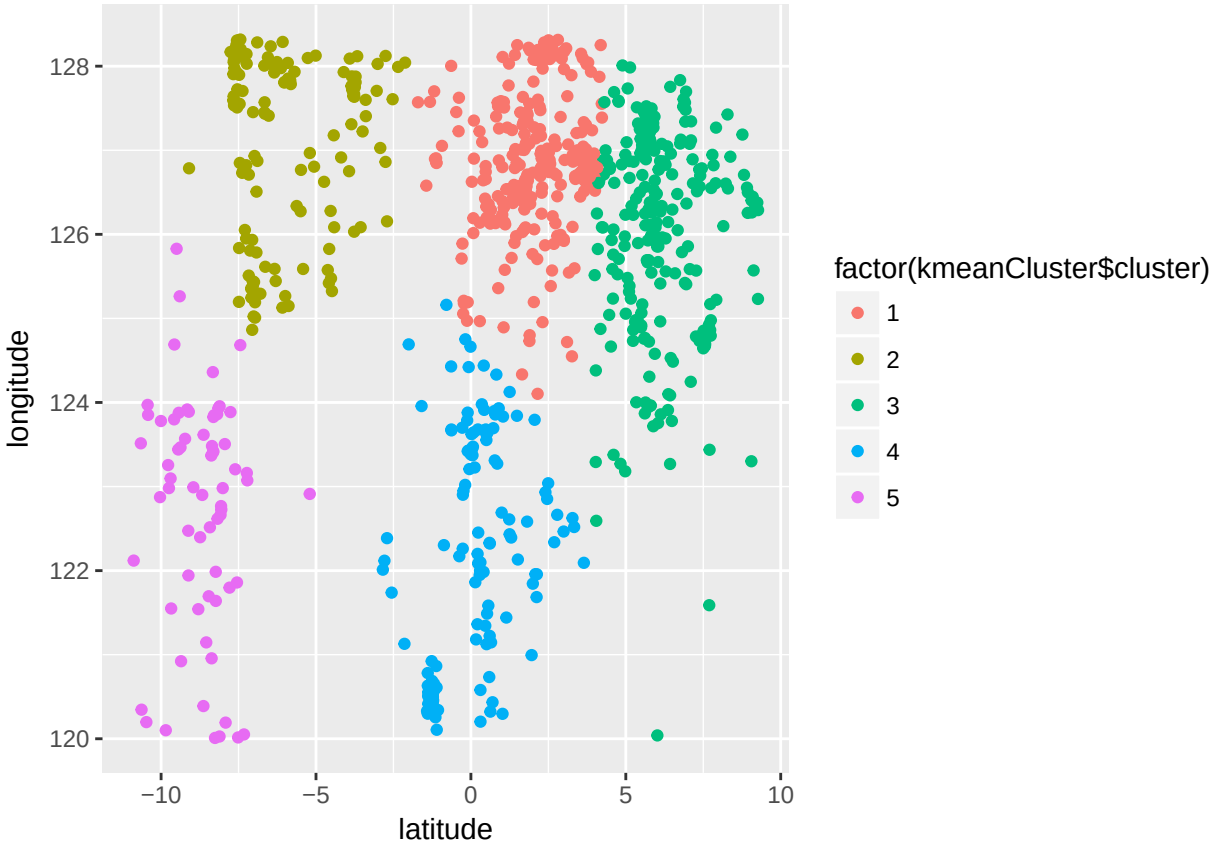
```
#looking into second index of the data

ggplot(dataset_cut1, aes(x = latitude, y = longitude)) +
  geom_point(alpha = 0.4)
```




```
#applying cluster analysis

kmeanCluster = kmeans(dataset_cut1, centers = 5, nstart = 30)
summary(kmeanCluster)
```

```
##              Length Class  Mode
## cluster      900    -none- numeric
## centers       10    -none- numeric
## totss          1    -none- numeric
## withinss       5    -none- numeric
## tot.withinss   1    -none- numeric
## betweenss      1    -none- numeric
## size           5    -none- numeric
## iter           1    -none- numeric
## ifault         1    -none- numeric
```

```
#centers of the cluster

clusterCenters = data.frame(kmeanCluster$centers)

#plotting the cluster

ggplot(dataset_cut1, aes(x = latitude, y = longitude)) +
        geom_point(aes(col = factor(kmeanCluster$cluster)))
```




```
clusterCenters
```

```
##      latitude longitude
## 1  2.02563382  126.8692
## 2 -5.87828603  127.0940
## 3  6.10997040  126.0886
## 4 -0.08687027  122.0011
## 5 -8.72136418  122.7177
```

Through these two models we can estimate a point with maximum risk and through the variance between the input and high risk points we can categorise the zone.

** The model is further enhanced by encorporating few more models and ideas like the latest earthquake centers or the model dveloped by you. **

Data Exploration For Magnitude and Month

The data volume is not giving a clear picture into the relation. I kept the scenario for further analysis and work on larger data chunk to generate a probability between earthquake occurences and time.

```
magClean = eqClean[, c(5, 17, 18)]
magClean = magClean%>%
  mutate(mgIntensity = ifelse(magClean$mag < 4, "low",
                              ifelse(magClean$mag <= 5.5, "medium", "high")))

indx_high = which(magClean$mgIntensity == "high")
indx_medium = which(magClean$mgIntensity == "medium")
```

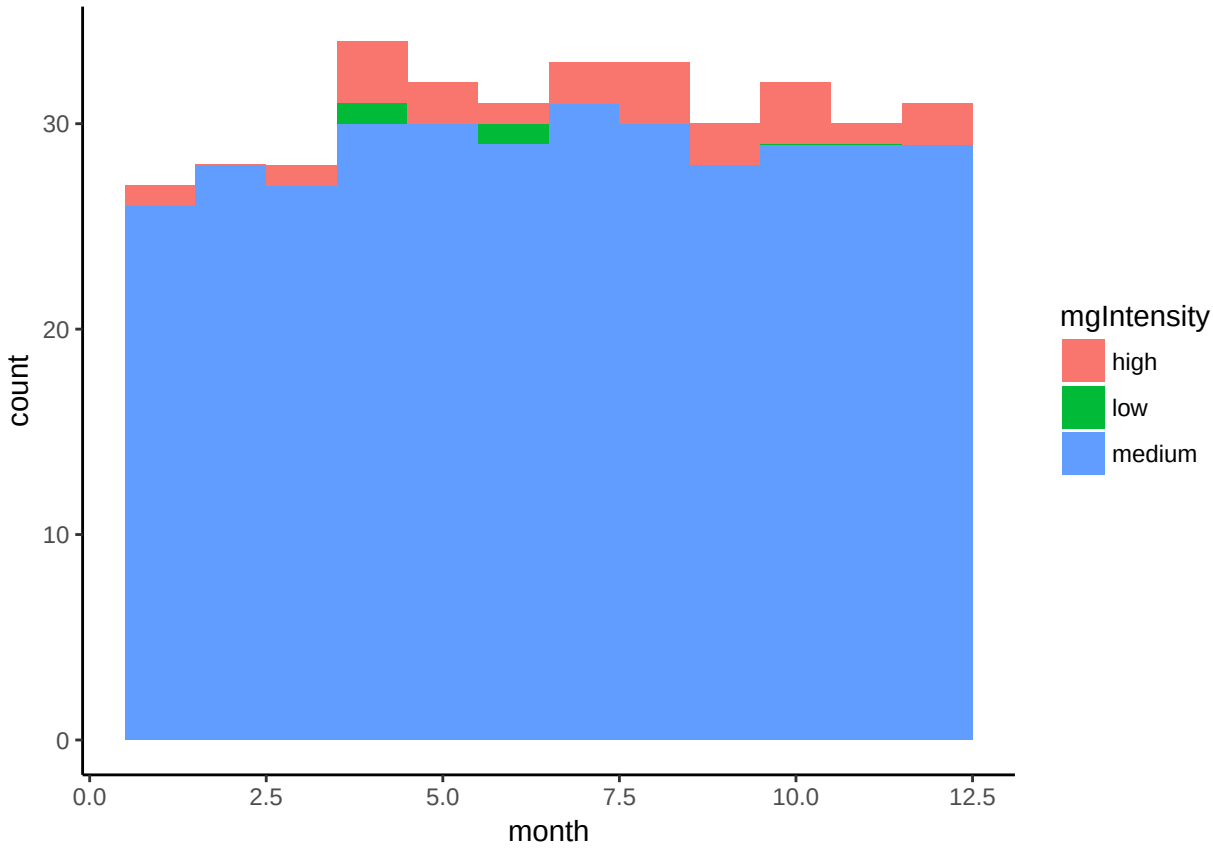
```
summary(magClean[indx_high, ])
```

```
##       mag            month            day        mgIntensity
##  Min.   :5.600   Min.   : 1.00   Min.   : 5.00   Length:25
##  1st Qu.:5.700   1st Qu.: 5.00   1st Qu.:15.00   Class :character
##  Median :5.800   Median : 8.00   Median :23.00   Mode  :character
##  Mean   :5.956   Mean   : 7.48   Mean   :21.16
##  3rd Qu.:6.100   3rd Qu.:10.00   3rd Qu.:28.00
##  Max.   :6.900   Max.   :12.00   Max.   :31.00
```

```
summary(magClean[indx_medium, ])
```

```
##       mag            month             day        mgIntensity
##  Min.   :4.00   Min.   : 1.000   Min.   : 1.0   Length:1342
##  1st Qu.:4.20   1st Qu.: 4.000   1st Qu.: 9.0   Class :character
##  Median :4.40   Median : 6.000   Median :16.0   Mode  :character
##  Mean   :4.47   Mean   : 6.358   Mean   :16.2
##  3rd Qu.:4.60   3rd Qu.: 9.000   3rd Qu.:24.0
##  Max.   :5.50   Max.   :12.000   Max.   :31.0
```

```
summ_month_mag = magClean %>%
  group_by(month, mgIntensity, day)%>%
  summarise(no = n())


ggplot(summ_month_mag, aes(x = month, fill = mgIntensity))+
  geom_histogram(bins = 12, binwidth =  1)+
  scale_x_continuous()+
  scale_y_continuous()+
  theme_classic()
```


count
30
20
10
0
0.0
2.5
5.0
7.5
10.0
12.5
month
mgIntensity
high
low
medium