BDM300 PROJECT

Covid_19 Data Analysis with R

NOVEMBER 8, 2021

EFAT GORJI, ALI SATARI KASBI, SHREYKUMAR RAJESHBHAI PATEL
Asem Omari

Introduction:

Covid-19 has been one of the most important issues during the last 2 years. As a result, many attempts have been taken to deal with this world-wide pandemic. similarly, numerous datasets related to covid-19 have been collected to be analysed and used for decision making. The dataset used in this project is a complete set of the country leveled numbers of cases. In the dataset, the number of cases from 187 countries have been gathered through 14 features. The variables of the dataset include the name of the country, the number of the confirmed cases, the number of death, the number of recovered cases, the number of active cases, the number of new cases (cases in the last 24 hours from the date of data integration) , the number of new death, the number of new recovered, the deaths per 100 cases, the number of recovered per 100 cases, the number of deaths per 100 recovered, the number of confirmed cases in the week before the date of data collection, the number of changes in the number of cases in the last week of data collection which is in fact the difference between the number of confirmed cases and the number of confirmed cases in the week before the collection time, the percentage of increase in the week before the data collection time, and the continent where the country is located.

Data Resource:

The dataset has been retrieved from https://www.kaggle.com/imdevskp/corona-virus-report?select=usa_county_wise.csv

Business Problem:

One considerable business problem that could be answered via analysing this dataset is modeling the data and generate the best model according to which the number of new deaths of a given country can be predicted with a high level of confidence. Thus, in this project, we will first take some exploratory steps on data such as descriptive statistics and data visualization to have an overview of data. Then, we will try to inspect the correlation between the different variables in order to decide which of the variables would make the best predictive model on the data. finally, we will try different algorithms of generating a predictive model along with evaluating the result models to find which algorithm gives the most accurate model on our data.

Descriptive statistics:

```
library(tidyverse)

data<-read.csv("C:/Users/Efat/Desktop/seneca/semester3/BDM300/project/country_wise_latest.csv")

glimpse(data)
```

```
Rows: 187

Columns: 15

$ Country.Region       <chr> "Afghanistan", "Albania", "Algeria~

$ Confirmed          <int> 36263, 4880, 27973, 907, 950, 86, ~

$ Deaths             <int> 1269, 144, 1163, 52, 41, 3, 3059, ~
$ Recovered            <int> 25198, 2745, 18837, 803, 242, 65, ~
$ Active               <int> 9796, 1991, 7973, 52, 667, 18, 917~
$ New.cases            <int> 106, 117, 616, 10, 18, 4, 4890, 73~
$ New.deaths           <int> 10, 6, 8, 0, 1, 0, 120, 6, 6, 1, 6~
$ New.recovered        <int> 18, 63, 749, 0, 0, 5, 2057, 187, 1~
$ Deaths...100.Cases     <dbl> 3.50, 2.95, 4.16, 5.73, 4.32, 3.49~
$ Recovered...100.Cases  <dbl> 69.49, 56.25, 67.34, 88.53, 25.47,~
$ Deaths...100.Recovered <dbl> 5.04, 5.25, 6.17, 6.48, 16.94, 4.6~
$ Confirmed.last.week    <int> 35526, 4171, 23691, 884, 749, 76, ~
$ X1.week.change         <int> 737, 709, 4282, 23, 201, 10, 36642~
$ X1.week...increase     <dbl> 2.07, 17.00, 18.07, 2.60, 26.84, 1~
$ WHO.Region             <chr> "Eastern Mediterranean", "Europe",~
```

Here we see the number of observations, the number of variables, and the name of each variable as well their types and some instances of the data stored under each variable. Also, we can see the dimension, the variables' names, and the structure of our dataset using the codes below.

```
> dim(data)

[1] 187  15

> names(data)

 [1] "Country.Region"         "Confirmed"

 [3] "Deaths"                 "Recovered"

 [5] "Active"                 "New.cases"

 [7] "New.deaths"             "New.recovered"

 [9] "Deaths...100.Cases"     "Recovered...100.Cases"

[11] "Deaths...100.Recovered" "Confirmed.last.week"

[13] "X1.week.change"         "X1.week...increase"

[15] "WHO.Region"

> attributes(data)

$names

 [1] "Country.Region"         "Confirmed"
 [3] "Deaths"                 "Recovered"
 [5] "Active"                 "New.cases"
 [7] "New.deaths"             "New.recovered"
 [9] "Deaths...100.Cases"     "Recovered...100.Cases"
[11] "Deaths...100.Recovered" "Confirmed.last.week"
[13] "X1.week.change"         "X1.week...increase"
[15] "WHO.Region"
```

The following lines show a summary of the data.

```
> summary(data)
 Country.Region       Confirmed          Deaths
 Length:187         Min.   :     10   Min.   :     0.0
 Class :character   1st Qu.:   1114   1st Qu.:    18.5
 Mode  :character   Median :   5059   Median :   108.0
                    Mean   :  88131   Mean   :  3497.5
                    3rd Qu.:  40460   3rd Qu.:   734.0
                    Max.   :4290259   Max.   :148011.0
   Recovered             Active             New.cases
 Min.   :      0.0   Min.   :      0.0   Min.   :    0.0
 1st Qu.:    626.5   1st Qu.:    141.5   1st Qu.:    4.0
 Median :   2815.0   Median :   1600.0   Median :   49.0
 Mean   :  50631.5   Mean   :  34001.9   Mean   : 1223.0
 3rd Qu.:  22606.0   3rd Qu.:   9149.0   3rd Qu.:  419.5
 Max.   :1846641.0   Max.   :2816444.0   Max.   :56336.0
   New.deaths       New.recovered      Deaths...100.Cases
 Min.   :   0.00   Min.   :    0.0   Min.   : 0.000
 1st Qu.:   0.00   1st Qu.:    0.0   1st Qu.: 0.945
 Median :   1.00   Median :   22.0   Median : 2.150
 Mean   :  28.96   Mean   :  933.8   Mean   : 3.020
 3rd Qu.:   6.00   3rd Qu.:  221.0   3rd Qu.: 3.875
 Max.   :1076.00   Max.   :33728.0   Max.   :28.560

 Recovered...100.Cases Deaths...100.Recovered Confirmed.last.week

 Min.   :  0.00       Min.   :0.00          Min.   :     10

 1st Qu.: 48.77       1st Qu.:1.45          1st Qu.:   1052

 Median : 71.32       Median :3.62          Median :   5020

 Mean   : 64.82       Mean   : Inf          Mean   :  78682

 3rd Qu.: 86.89       3rd Qu.:6.44          3rd Qu.:  37080

 Max.   :100.00       Max.   : Inf          Max.   :3834677

 X1.week.change   X1.week...increase  WHO.Region

 Min.   :   -47   Min.   : -3.840    Length:187

 1st Qu.:    49   1st Qu.:  2.775    Class :character

 Median :   432   Median :  6.890    Mode  :character

 Mean   :  9448   Mean   : 13.606

 3rd Qu.:  3172   3rd Qu.: 16.855

 Max.   :455582   Max.   :226.320
```

finding distinct value in region colum:

```
unique(coviddataset[c("WHO Region")])

> unique(coviddataset[c("WHO Region")])

# A tibble: 6 x 1

  `WHO Region`

  <chr>

1 Eastern Mediterranean

2 Europe

3 Africa

4 Americas

5 Western Pacific

6 South-East Asia
```

As we can see there are 6 regions where the data is presented from.

showing range of all columns:

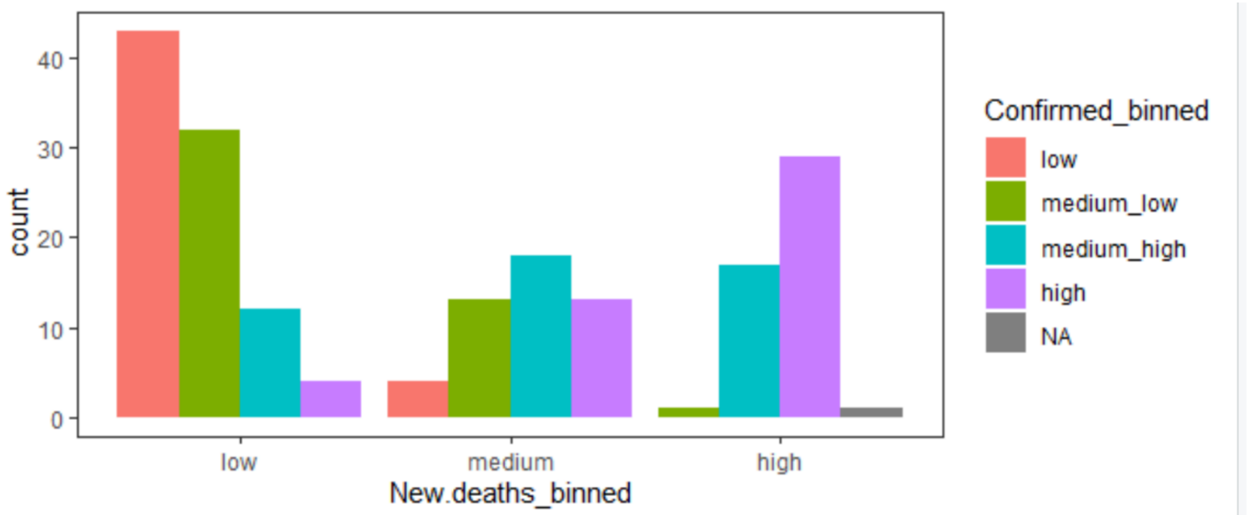
```
> range(coviddataset$Confirmed)
[1]      10 4290259
> range(coviddataset$Deaths)
[1]      0 148011
> range(coviddataset$Recovered)
[1]       0 1846641
> range(coviddataset$Active)
[1]       0 2816444
> range(coviddataset$`New cases`)
[1]     0 56336
> range(coviddataset$`New deaths`)
[1]    0 1076
> range(coviddataset$`New recovered`)
[1]     0 33728
> range(coviddataset$`Deaths / 100 Cases`)
[1]  0.00 28.56
> range(coviddataset$`Deaths / 100 Recovered`)
[1]   0 Inf
> range(coviddataset$`Recovered / 100 Cases`)
[1]   0 100
> range(coviddataset$`Confirmed last week`)
[1]      10 3834677
> range(coviddataset$ '1 week change')
[1]    -47 455582
> range(coviddataset$ '1 week % increase')
[1]  -3.84 226.32
```

So this gives us an idea about what are the minimum and maximum numbers in the dataset. The value on the left is the minimum value in the column and right value gives us maximum value in column.

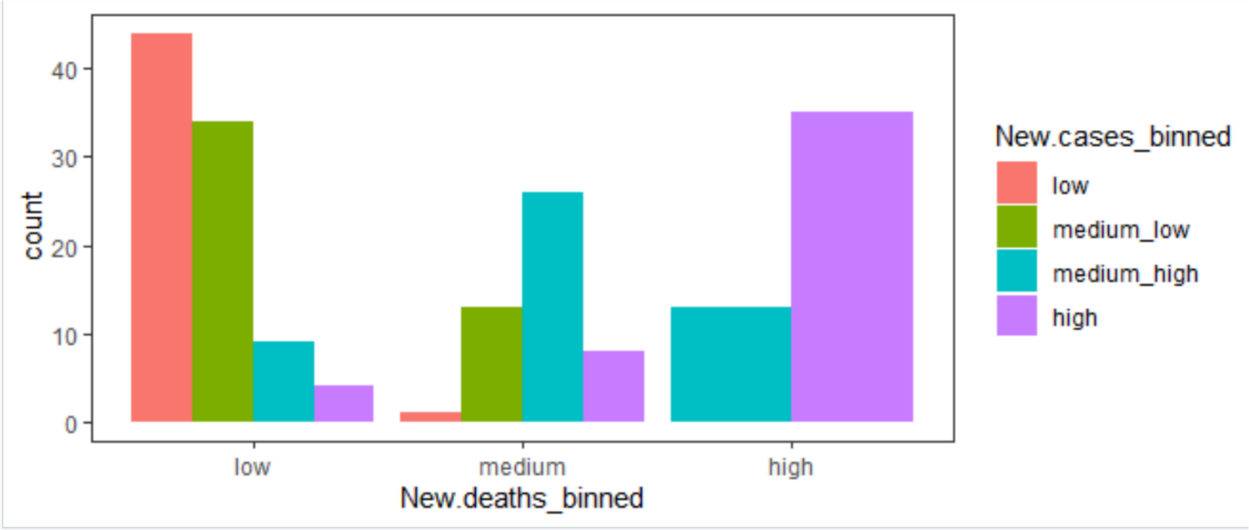
Data visualization:

To have a graphical view of the data, we draw some graphs to compare our target variable (new deaths) versus some other variables.

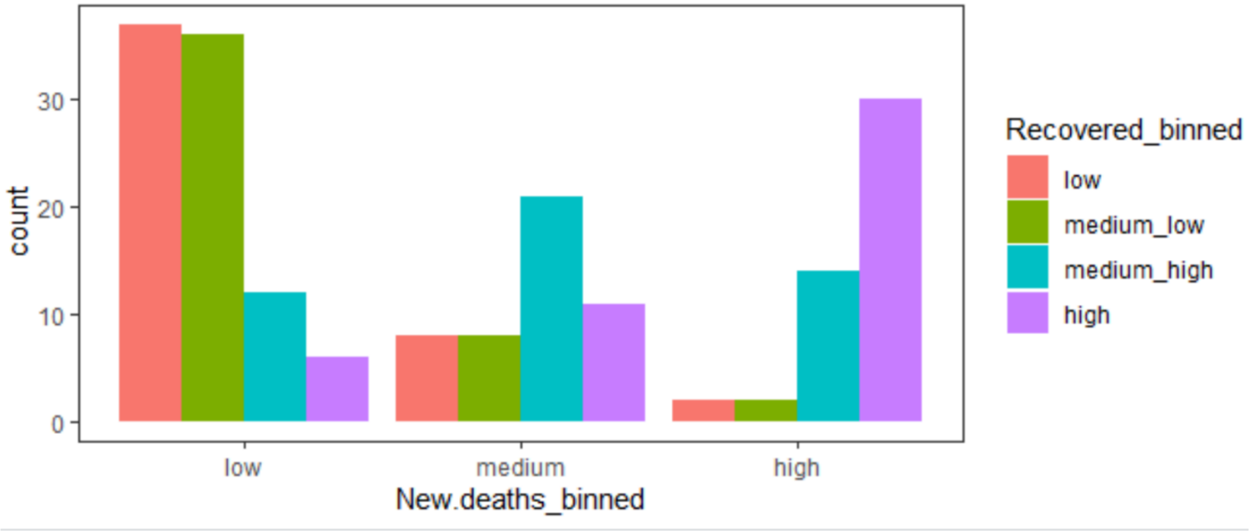
```
data %>%
+  ggplot(aes(New.deaths_binned, fill = Confirmed_binned))+
+  geom_bar(position = "dodge",
+          alpha = 1)+
+  theme_bw()+
+  theme(panel.grid.major = element_blank(),
+        panel.grid.minor = element_blank())
```



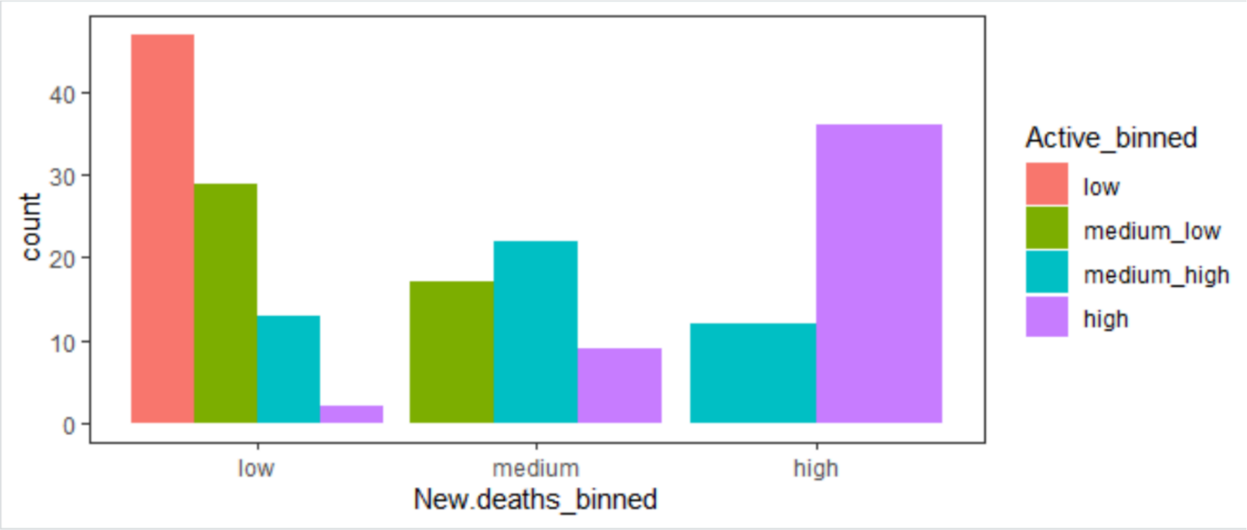
```
data %>%
+  ggplot(aes(New.deaths_binned, fill = New.cases_binned))+
+  geom_bar(position = "dodge",
+          alpha = 1)+
+  theme_bw()+
+  theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
```



```
data %>%
+  ggplot(aes(New.deaths_binned, fill = Recovered_binned))+
+  geom_bar(position = "dodge",
+          alpha = 1)+
+  theme_bw()+
+  theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
```



```
data %>%
+  ggplot(aes(New.deaths_binned, fill = Active_binned))+
+  geom_bar(position = "dodge",
+          alpha = 1)+
+  theme_bw()+
+  theme(panel.grid.major = element_blank(),
+        panel.grid.minor = element_blank())
```



Data modeling:

Since all the variables in our dataset are numeric, we need to apply a binning process on them in order to be able to use Decision Tree, Random Forest, Naïve Bayes, and KNN algorithms on data and build a predictive model.

We will convert the data under some of the variables to categorical data by binning each of them to 4 categories based on their quantiles named as 'low','medium_low','medium_high', and 'high'.

```
library('Hmisc')

install.packages('psych')

install.packages('Hmisc')

labels=c('low','medium_low','medium_high','high')

#binning confirmed

quantile(data$Confirmed)

breaks_Confirmed=c(10.0,1114.0,5059.0,40460.5,4290259.0)
```

```
data$Confirmed_binned<-cut(data$Confirmed,breaks=breaks_Confirmed,
                lower=TRUE, right=FALSE,labels=labels)
data$Confirmed_binned

#binning Active
quantile(data$Active)
breaks_Active=c(0.0,141.5,1600.0,9149.0,2816444.0)
data$Active_binned<-cut(data$Active,breaks =breaks_Active,include.lowest=TRUE,
              right=FALSE,labels=labels )
data$Active_binned

#binnig Deaths
quantile(data$Deaths)
breaks_Death=c(0.0,18.5,108.0,734.0,148011.0 )
data$Deaths_binned<-cut(data$Deaths,breaks =breaks_Death,include.lowest=TRUE,
              right=FALSE,labels=labels )
data$Deaths_binned

#binnig Recovered
quantile(data$Recovered)
breaks_Recovered=c(0.0,626.5,2815.0,22606.0,1846641.0 )
data$Recovered_binned<-cut(data$Recovered,breaks =breaks_Recovered,include.lowest=TRUE,
              right=FALSE,labels=labels)
data$Recovered_binned

#binning New cases
quantile(data$New.cases)
breaks_New.cases=c(0.0,4.0,49.0,419.5,56336.0)
data$New.cases_binned<-cut(data$New.cases,breaks =breaks_New.cases,include.lowest=TRUE,
                    right=FALSE,labels=labels )

data$New.cases_binned


#binning Recovered/100cases

quantile(data$Recovered...100.Cases)

breaks_Recovered...100.Cases=c(0.000,48.770,71.320,86.885,100.000 )

data$Recovered...100.Cases_binned<-cut(data$Recovered...100.Cases,breaks
=breaks_Recovered...100.Cases,include.lowest=TRUE,

                     right=FALSE,labels=labels)

data$Recovered...100.Cases_binned


#binning Deaths/100cases

quantile(data$Deaths...100.Cases)

breaks_Deaths...100.Cases=c(0.000,0.945,2.150,3.875,28.560)

data$Deaths...100.Cases_binned<-cut(data$Deaths...100.Cases,breaks
=breaks_Deaths...100.Cases,include.lowest=TRUE,

                             right=FALSE,labels=labels)

data$Deaths...100.Cases_binned

#binning 1Week%increase

quantile(data$X1.week...increase)

breaks_X1.week...increase=c(-3.840,2.775,6.890,16.855,226.320)

data$X1.week...increase_binned<-cut(data$X1.week...increase,breaks
=breaks_X1.week...increase,include.lowest=TRUE,

                           right=FALSE,labels=labels)

data$X1.week...increase_binned
```

Target variable:

New death which is the target variable is binned into 3 bins: low, medium, and high.

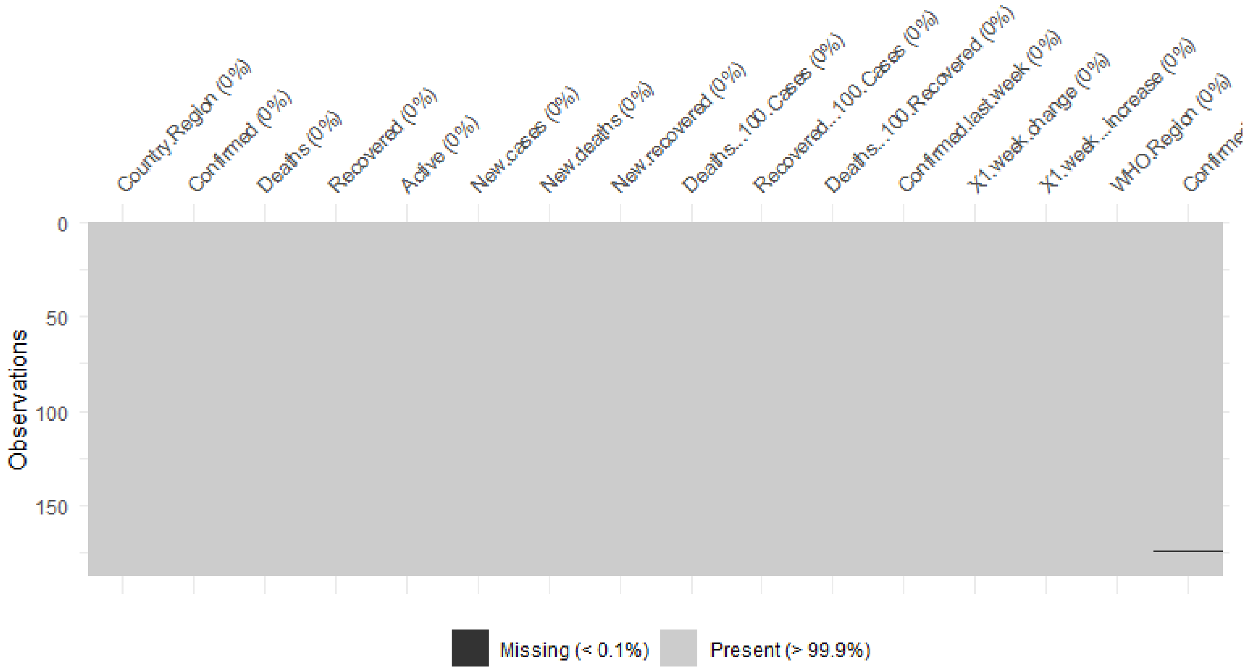
```
#binning New deaths (target variable)

quantile(data$New.deaths)
```

```
breaks_New.deaths=c(0,1,6,1076)

data$New.deaths_binned<-cut(data$New.deaths,breaks =breaks_New.deaths,include.lowest=TRUE,

                    right=FALSE,labels=c('low','medium','high'))

data$New.deaths_binned
```

missing data:

before working on correlations and finding the relationships between the variables, we need to check for the missing values and deal with the missing observations if there is any. We use vis_miss function for this purpose.

```
install.packages('naniar')

library(naniar)

vis_miss(data)
```



As we see in the graph, the percentage of missing value is almost 0, and we do not need to take any steps in this regard.

correlations:

now, it's time to inspect correlations between the variables and decide which variables should be brought in the predictive models to predict the number of new deaths.

```
#correlation between Deaths & Recovered

cor(data$Deaths, data$Recovered)

[1] 0.8320983

table(data$Deaths_binned, data$Recovered_binned)


             low medium_low medium_high high
  low         37        10           0    0
  medium_low    6        29          10    1
  medium_high   1         6          29   11
  high          3         1           8   35

> chisq.test(data$Deaths_binned, data$Recovered_binned,correct=FALSE)


        Pearson's Chi-squared test


data:  data$Deaths_binned and data$Recovered_binned

X-squared = 218.32, df = 9, p-value < 2.2e-16
```

As we expected, the correlation between the number of death and recovered is very high, and we cannot bring both in the models. Also, the cross tabulation table of these variables as well as the very small amount of p-value of the chi-square test shows a high correlation between death and recovered.

```
#correlation between Deaths & New.cases

cor(data$Deaths, data$New.cases)

[1] 0.8069745

table(data$Deaths_binned, data$New.cases_binned)


             low medium_low medium_high high
  low         28        14           5    0
  medium_low   14        19          12    1
  medium_high   2        13          20   12
  high          1         1          11   34
```

```
> chisq.test(data$Deaths_binned, data$New.cases_binned,correct=FALSE)

        Pearson's Chi-squared test

data:  data$Deaths_binned and data$New.cases_binned
X-squared = 130.26, df = 9, p-value < 2.2e-16
#correlation between Active & New.cases
cor(data$Active, data$New.cases)
[1] 0.8511905
> table(data$Active_binned, data$New.cases_binned)

              low medium_low medium_high high
  low          31         16           0    0
  medium_low   11         25          10    0
  medium_high   2          6          28   11
  high          1          0          10   36
> chisq.test(data$Active_binned, data$New.cases_binned,correct=FALSE)

        Pearson's Chi-squared test

data:  data$Active_binned and data$New.cases_binned
X-squared = 189.82, df = 9, p-value < 2.2e-16
```

The same situation applies for death and new cases as well as active and new cases.

```
#correlation between Death/100 cases & New.cases
> cor(data$Deaths...100.Cases,data$New.cases)
[1] 0.02010425
> table(data$Deaths...100.Cases_binned, data$New.cases_binned)

              low medium_low medium_high high
```

```
 low         15         10         13    9
 medium_low   13         11         12   10
 medium_high   8         11         14   14
 high         9         15          9   14
> chisq.test(data$Deaths...100.Cases_binned, data$New.cases_binned,correct=FALSE)


        Pearson's Chi-squared test


data:  data$Deaths...100.Cases_binned and data$New.cases_binned
X-squared = 7.0576, df = 9, p-value = 0.6311
```

As the amount of the correlation between death per 100 cases and the p-value of the chi-square test show, there is no statistically significant linear relationship between these variables and we consider them as independent variables once we want to bring them in the predictive models.

#correlation between Death/100 cases & Recovered/100 cases

```
> cor(data$Deaths...100.Cases,data$Recovered...100.Cases)
[1] -0.1689198
> table(data$Deaths...100.Cases_binned, data$Recovered...100.Cases_binned)


        low medium_low medium_high high
 low          9          9         12   17
 medium_low   13         10         12   11
 medium_high  12         15         12    8
 high        13         12         11   11
> chisq.test(data$Deaths...100.Cases_binned, data$Recovered...100.Cases_binned,correct=FALSE)


        Pearson's Chi-squared test


data:  data$Deaths...100.Cases_binned and data$Recovered...100.Cases_binned
X-squared = 6.4009, df = 9, p-value = 0.6992
```

Similarly, the number of death per 100 cases and the number of recovered per 100 cases seem to be independent.

#correlation between New cases & Recovered/100 cases

```
> cor(data$Recovered...100.Cases,data$New.cases)

[1] -0.07866557

> table(data$Recovered...100.Cases_binned, data$New.cases_binned)


             low medium_low medium_high high

  low          8         8         15   16

  medium_low    5         9         15   17

  medium_high  15        15          10    7

  high        17        15          8    7

> chisq.test(data$Recovered...100.Cases_binned, data$New.cases_binned,correct=FALSE)


              Pearson's Chi-squared test


data:  data$Recovered...100.Cases_binned and data$New.cases_binned

X-squared = 23.145, df = 9, p-value = 0.005879
```

Here for the number of recovered per 100 cases versus new cases, although the amount of pvalue shows dependency between these two variables, the amount of correlation is considerably low and the table also shows no special trends between the amounts of new cases based on recovered per 100 cases. Thus, we will consider these variable independent, and consequently bring them both in the predictive models.

#correlation between X1.week...increase& New.cases

```
> cor(data$X1.week...increase,data$New.cases)

[1] 0.03079057

> table(data$X1.week...increase_binned, data$New.cases_binned)


             low medium_low medium_high high

  low         23        13          9    2
  medium_low   12        17          6   10
  medium_high   6        11         16   15
  high          4         6         17   20
> chisq.test(data$X1.week...increase_binned, data$New.cases_binned,correct=FALSE)


        Pearson's Chi-squared test


data:  data$X1.week...increase_binned and data$New.cases_binned
X-squared = 46.793, df = 9, p-value = 4.289e-07
```

In this case, despite the p-value of the independency test that shows the independency of new cases with the amount of increase in one week, we will bring both in the model since the amount of correlation between these variables is very low.

#relationship between WHO.Region & New.deaths

```
> table(data$WHO.Region,data$New.deaths_binned)


                     low medium high
  Africa                29     15    4
  Americas                14      5   16
  Eastern Mediterranean   6      9    7
  Europe                  24     16   16
  South-East Asia        6      1    3
  Western Pacific       12      2    2
> chisq.test(data$WHO.Region, data$New.deaths_binned,correct=FALSE)


        Pearson's Chi-squared test


data:  data$WHO.Region and data$New.deaths_binned
X-squared = 26.226, df = 10, p-value = 0.003448
```

According to what we observe regarding dependency between the continents and the number of new deaths, we will bring continent in the predictive model.

Finally, the variables that will participate in the predictive models are new cases, deaths per 100 cases, recovered per 100 cases, 1 week increase, and continent.

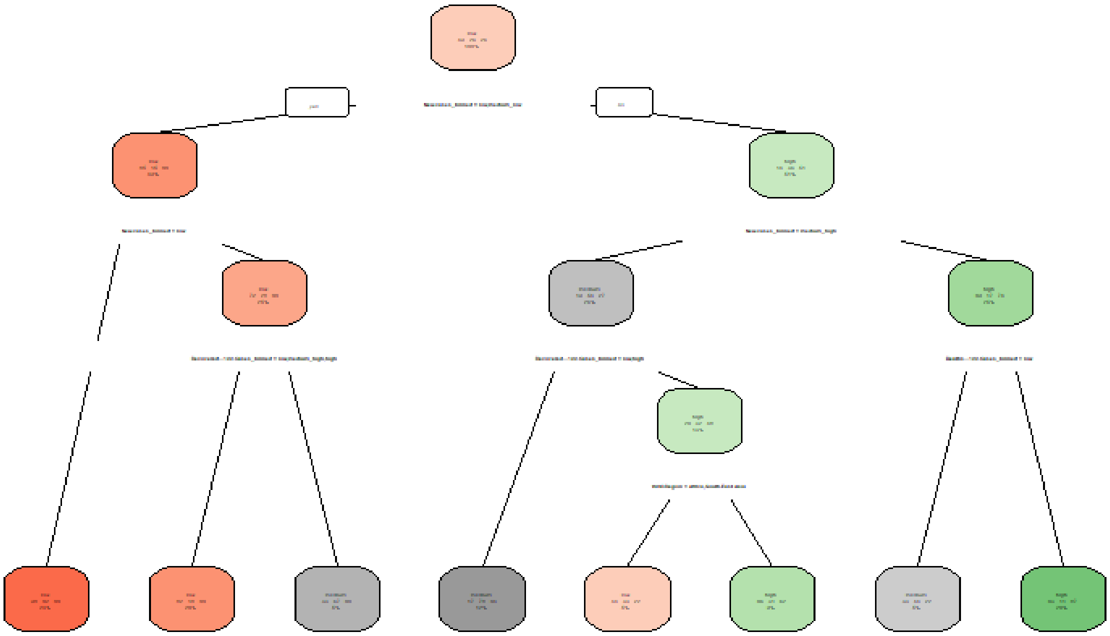
Decision Tree:

The first algorithm we will work on is Decision tree. However, we first need to split our data into training set and testing set. The codes are brought below.

```
#Building a Decision tree with the binned variables

#split data into training and test data sets

indexes = sample(1:nrow(data), size = 0.3*nrow(data))


#Test dataset 30%

Test_data = data[indexes,]

dim(Test_data)  #56 24


#Train dataset 70%

Train_data = data[-indexes,]

dim(Train_data)  #131 24
```

131 observations are training data and the remaining 56 observations are the test data.

```
library(rpart)

des_tree=rpart(New.deaths_binned~New.cases_binned+Deaths...100.Cases_binned+

            Recovered...100.Cases_binned+X1.week...increase_binned+WHO.Region,
data=data,method='class')

install.packages('rattle')

install.packages('rpart.plot')

install.packages('RColorBrewer')

library(rpart.plot)

library(RColorBrewer)

# plot des_tree

fancyRpartPlot(des_tree,main='decision tree on variables')

rpart.plot(des_tree,clip.right.labs = FALSE,branch = .3)
```



The decision tree has been generated, but since it is not very clear, we need to see the rules extracted from the tree.

```
#extracting the rules of the decision tree

rpart.rules(des_tree,cover=TRUE)
```

here is a part of the output:

```
New.deaths_binned  low med hig
cover

            low [.44 .33 .22] when New.cases_binned is medium_high & Recovered...100.Cases_binned is
medium_low or medium_high & WHO.Region is                              Africa or South-East Asia
5%

            low [.82 .18 .00] when New.cases_binned is  medium_low & Recovered...100.Cases_binned is
low or medium_high or high
20%
```

two of the extracted rules are brough here. For example, the second rule represents that when New.cases_binned is medium_low & Recovered...100.Cases_binned is low or medium_high or high, then the number of new deaths is low with the probability of .82. Also, 20% of the observations in the training set have this situation.

model evaluation:

after we built our model, we need to check the accuracy of the model using the test data through the confusion matrix and the MCnemar test. the null hypotheses of this test says the predicted values and the expected values are the same.

```
#Model Evaluation

Predicted_New.deaths <- predict(des_tree,newdata = Test_data,type='class')

Predicted_New.deaths  #the predicted levels of new death for the test set will be the output of this code.


#confusion matrix

install.packages('caret')

library('caret')

con_matrix<-confusionMatrix(data=Predicted_New.deaths,reference=Test_data$New.deaths_binned)

con_matrix  #80.36% accuracy

#Mcnemar's Test P-Value : 0.4235

#(the null hypothesis: the predicted values and the expected values are the same / accepted)
```

```
Confusion Matrix and Statistics


          Reference
Prediction low medium high
    low     24      4    1
    medium   0      9    0
    high     2      2   14


Overall Statistics


               Accuracy : 0.8393
                 95% CI : (0.7167, 0.9238)
    No Information Rate : 0.4643
    P-Value [Acc > NIR] : 7.141e-09
```

```
                  Kappa : 0.7451

 Mcnemar's Test P-Value : 0.09647

Statistics by Class:

                     Class: low Class: medium Class: high
Sensitivity              0.9231        0.6000      0.9333
Specificity              0.8333        1.0000      0.9024
Pos Pred Value           0.8276        1.0000      0.7778
Neg Pred Value           0.9259        0.8723      0.9737
Prevalence               0.4643        0.2679      0.2679
Detection Rate           0.4286        0.1607      0.2500
Detection Prevalence     0.5179        0.1607      0.3214
Balanced Accuracy        0.8782        0.8000      0.9179
```

Thus,we have built a model with almost 84% accuracy.

We can also build the decision tree model based on non-binned numeric data based on anova model using the codes below. However, the content of this model is beyond this project.

```
#building the model with the unbinned variables (anova model)
anova_model=rpart(New.deaths~New.cases+Deaths...100.Cases+
                Recovered...100.Cases+X1.week...increase+WHO.Region, data=data,method='class')
rpart.plot(anova_model)
rpart.rules(anova_model)
```

Random forest:

Now we will try Random Forest on our data to see how the model will look like using this algorithm.

```
#random forest
install.packages('randomForest')
library('randomForest')
```

```
random_forest<-randomForest(New.deaths_binned~New.cases_binned+Deaths...100.Cases_binned+
              Recovered...100.Cases_binned+X1.week...increase_binned+WHO.Region,data = data,
            method='class', ntree=50, mtry=2, maxnodes = NULL)

random_forest
```

here, we have tried a random forest of 50 trees and 2 variables in each split. Using the codes below we can predict the level of new deaths for any arbitrary set of countries. Here we used the test set to find the accuracy of the model.

```
prediction <-predict(random_forest,Test_data)

prediction

69     57    185     38     19    130     18     88    178
  low    low medium   high    low   high    low   high   high
   61     54     59     13    136     68    118      1    129
  low   high medium medium medium    low    low   high   high
  141     58     23    124     32      8     35    177    109
  low medium medium    low   high   high    low medium    low
  122    103    160     89    162     71    147    182      5
  low   high medium    low medium   high medium    low medium
  127    168    187     12    183     77    120     14     36
  low    low medium    low medium   high medium   high   high
  108    101     92     98    165     24     28     51    142
  low    low   high    low    low   high    low   high    low
    4     49
  low    low

Levels: low medium high

confu_matrix<-confusionMatrix(prediction,Test_data$New.deaths_binned)

confu_matrix  #94.55% accuracy

Confusion Matrix and Statistics


         Reference
```

```
Prediction low medium high

    low     24      1    0

    medium   0     14    0

    high     2      0   15

Overall Statistics

               Accuracy : 0.9464

                 95% CI : (0.8513, 0.9888)

    No Information Rate : 0.4643

    P-Value [Acc > NIR] : 9.779e-15

                  Kappa : 0.9169

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: low Class: medium Class: high

Sensitivity              0.9231        0.9333      1.0000

Specificity              0.9667        1.0000      0.9512

Pos Pred Value           0.9600        1.0000      0.8824

Neg Pred Value           0.9355        0.9762      1.0000

Prevalence               0.4643        0.2679      0.2679

Detection Rate           0.4286        0.2500      0.2679

Detection Prevalence     0.4464        0.2500      0.3036

Balanced Accuracy        0.9449        0.9667      0.9756
```

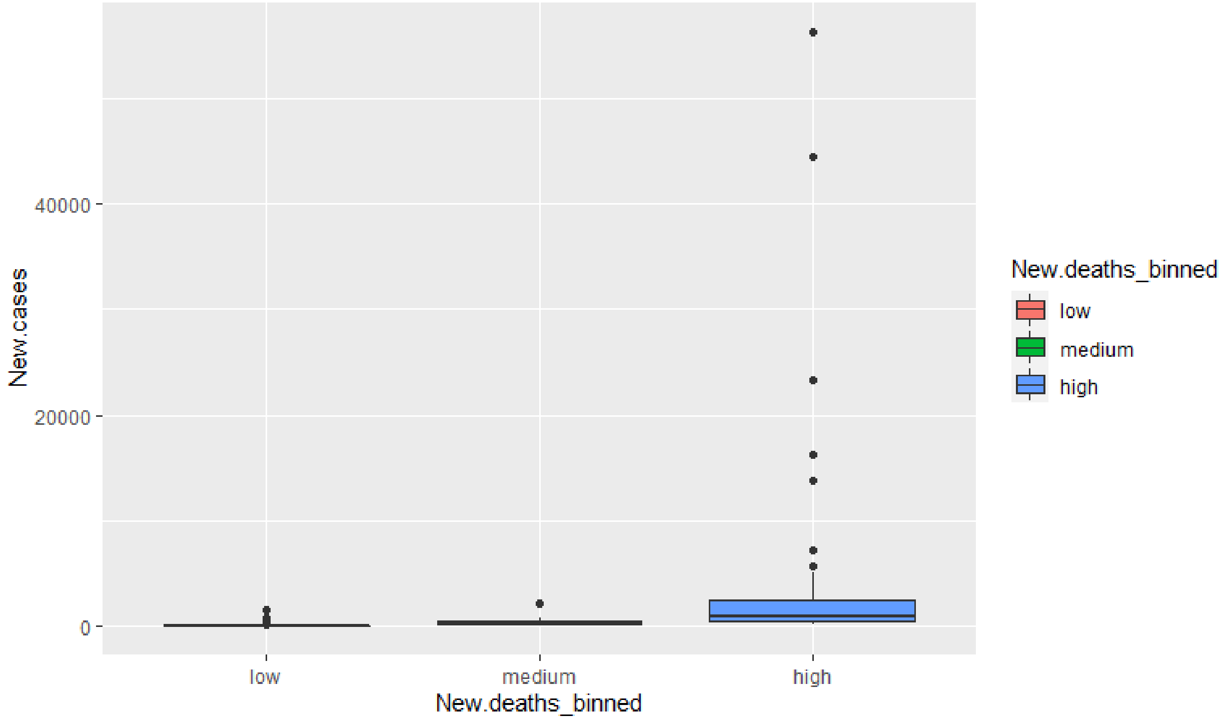
As we see, the accuracy of the model is almost 95%, so we can use this model for prediction purposes.

Naïve Bayes:

to apply naïve bayes algorithm, we need to first do some visualizations on data to see if there is any overlapping between different levels of the target variable based on the independent variables.

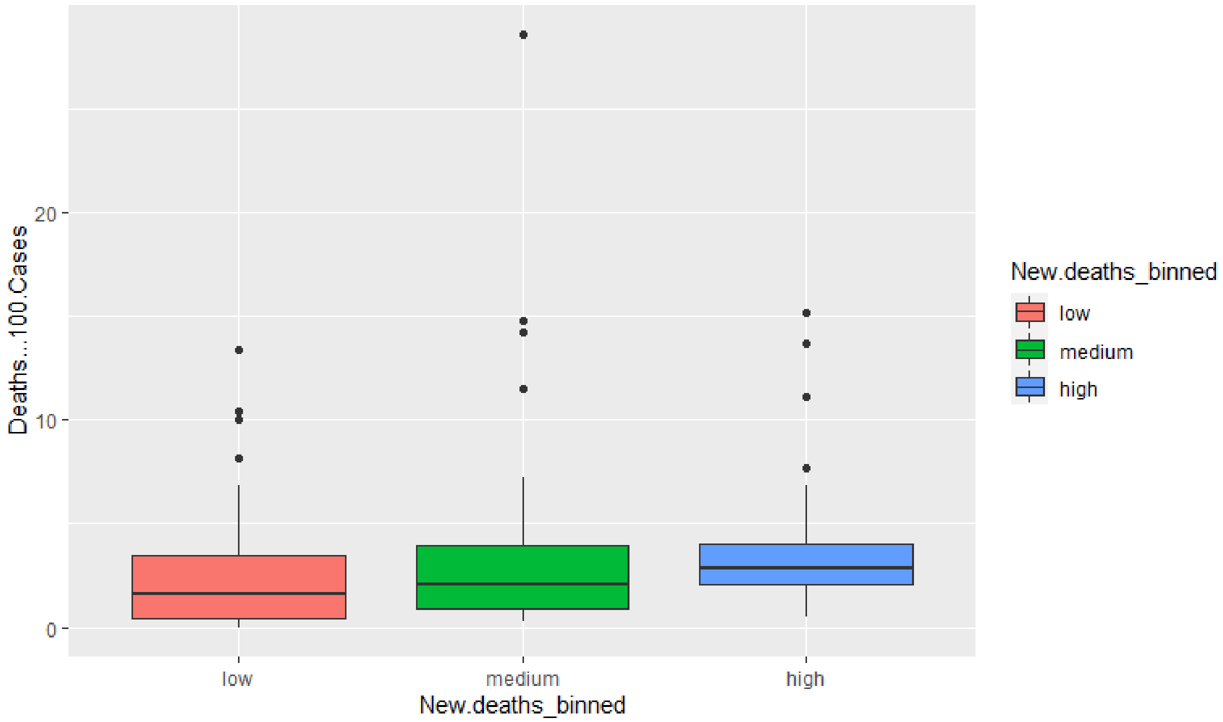
#visualization

```
data%>%
  ggplot(aes(x=New.deaths_binned,y=New.cases,fill=New.deaths_binned))+
  geom_boxplot()
```



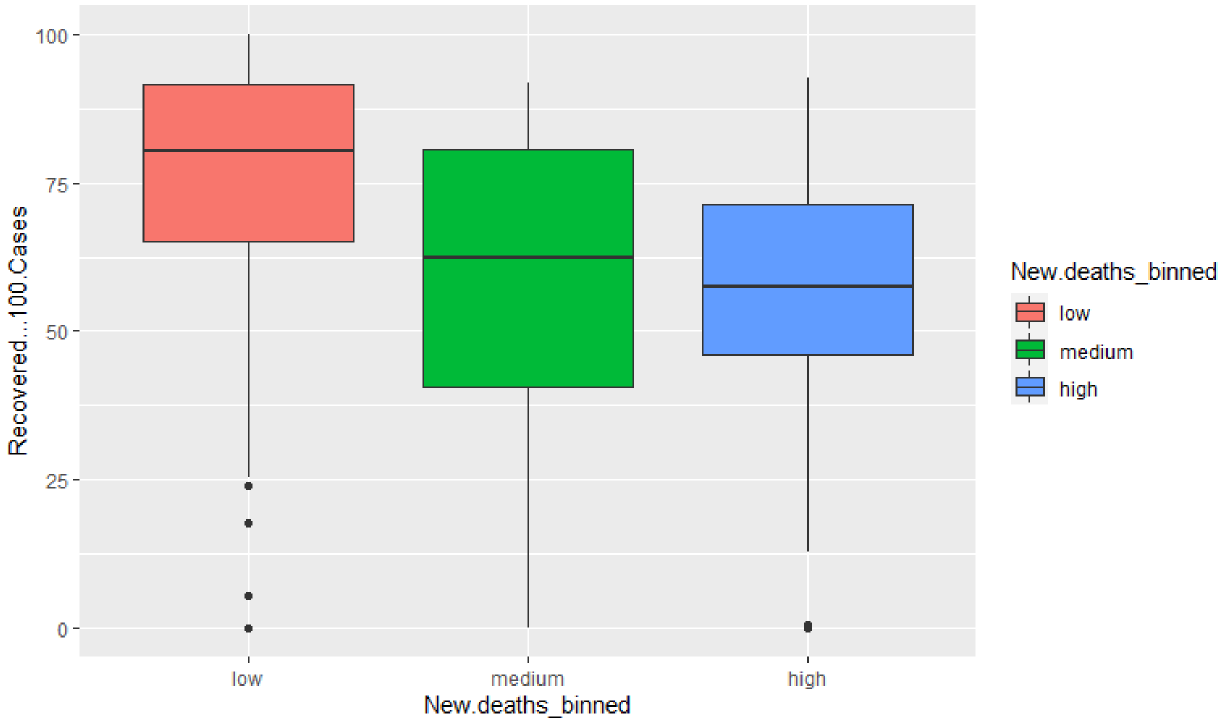
The variability of new cases is higher for high new deaths.

```
data%>%
  ggplot(aes(x=New.deaths_binned,y=Deaths...100.Cases,fill=New.deaths_binned))+
  geom_boxplot()
```

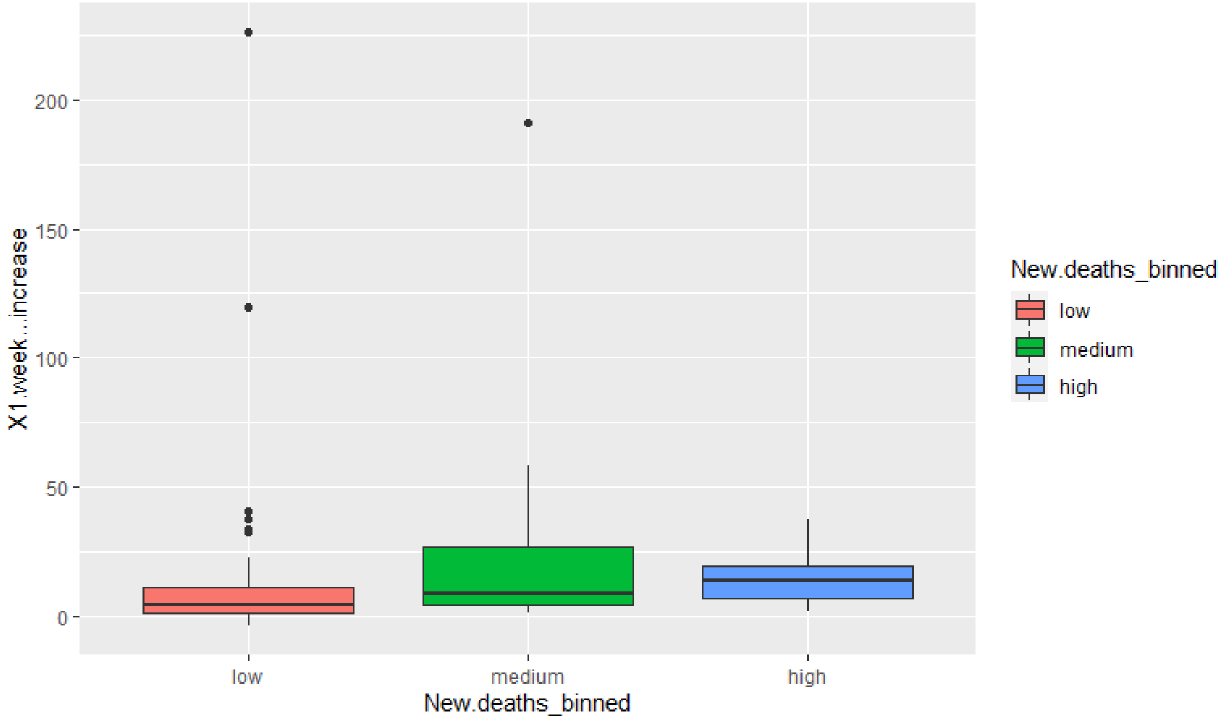


There are overlaps of the numbers of deaths per 100 cases for the levels of the target variable which might affect the accuracy of the model.

```
data%>%
  ggplot(aes(x=New.deaths_binned,y=Recovered...100.Cases,fill=New.deaths_binned))+
  geom_boxplot()
```



```
data%>%
  ggplot(aes(x=New.deaths_binned,y=X1.week...increase,fill=New.deaths_binned))+
  geom_boxplot()
```



Also, we see the same situation for recovered per 100 cases and 1 week increase.

Model building:

We will build the model with the laplace=1 to resolve the problem of the 0 probability.

```
#setting the model
naive_model<-naive_bayes(New.deaths_binned~New.cases_binned+Deaths...100.Cases_binned+

Recovered...100.Cases_binned+X1.week...increase_binned+WHO.Region,data=Train_data,laplace = 1)
#looking at the model
naive_model
```

the probabilities are given. So we can predict the level of the new death for a given country.

```
=========================== Naive Bayes ============================


 Call:
naive_bayes.formula(formula = New.deaths_binned ~ New.cases_binned +
    Deaths...100.Cases_binned + Recovered...100.Cases_binned +
    X1.week...increase_binned + WHO.Region, data = Train_data,
    laplace = 1)


---------------------------------------------------------------------


Laplace smoothing: 1


---------------------------------------------------------------------


 A priori probabilities:


      low    medium      high
0.4961832 0.2519084 0.2519084
```

```
--------------------------------------------------------------------

 Tables:

--------------------------------------------------------------------
 ::: New.cases_binned (Categorical)
--------------------------------------------------------------------

New.cases_binned       low     medium       high
    low        0.47826087 0.02702703 0.02702703
    medium_low  0.33333333 0.29729730 0.02702703
    medium_high 0.13043478 0.45945946 0.24324324
    high       0.05797101 0.21621622 0.70270270

--------------------------------------------------------------------
 ::: Deaths...100.Cases_binned (Categorical)
--------------------------------------------------------------------

Deaths...100.Cases_binned       low    medium      high
           low        0.3913043 0.2162162 0.1081081
           medium_low  0.2173913 0.2702703 0.1891892
           medium_high 0.1739130 0.2972973 0.3243243
           high       0.2173913 0.2162162 0.3783784

--------------------------------------------------------------------
 ::: Recovered...100.Cases_binned (Categorical)
--------------------------------------------------------------------

Recovered...100.Cases_binned       low     medium      high
```

```
          low        0.18840580 0.29729730 0.32432432
          medium_low  0.15942029 0.24324324 0.43243243
          medium_high 0.37681159 0.24324324 0.16216216
          high       0.27536232 0.21621622 0.08108108

---------------------------------------------------------------------
 ::: X1.week...increase_binned (Categorical)
---------------------------------------------------------------------

X1.week...increase_binned       low     medium      high
         low        0.36231884 0.18918919 0.02702703
         medium_low  0.28985507 0.21621622 0.21621622
         medium_high 0.23188406 0.24324324 0.35135135
         high       0.11594203 0.35135135 0.40540541

---------------------------------------------------------------------
 ::: WHO.Region (Categorical)
---------------------------------------------------------------------

WHO.Region                   low     medium      high
  Africa              0.35211268 0.23076923 0.10256410
  Americas              0.15492958 0.15384615 0.23076923
  Eastern Mediterranean 0.07042254 0.12820513 0.15384615
  Europe               0.22535211 0.38461538 0.35897436
  South-East Asia      0.07042254 0.02564103 0.07692308
  Western Pacific      0.12676056 0.07692308 0.07692308
```

Example:

We want to predict the level of new death for a country called ‘x’ with the below properties:

X=(new cases=medium-high, death/100=medium-low, recovered/100=high,1.week increase=high, region=Europe)

P(new death=low)=0.52

P(new death=medium)=0.25

p(new death=high)=0.23

P(new cases=medium-high|new death=low)=0.125

P(death/100=medium-low|new death=low)=0.263

P(recovered=high|new death=low)=0.416

P(1.week increase =high|new death=low)=0.111

P(region=Europe |new death=low)=0.297

=>p(new death=low|x)=0.0002344455

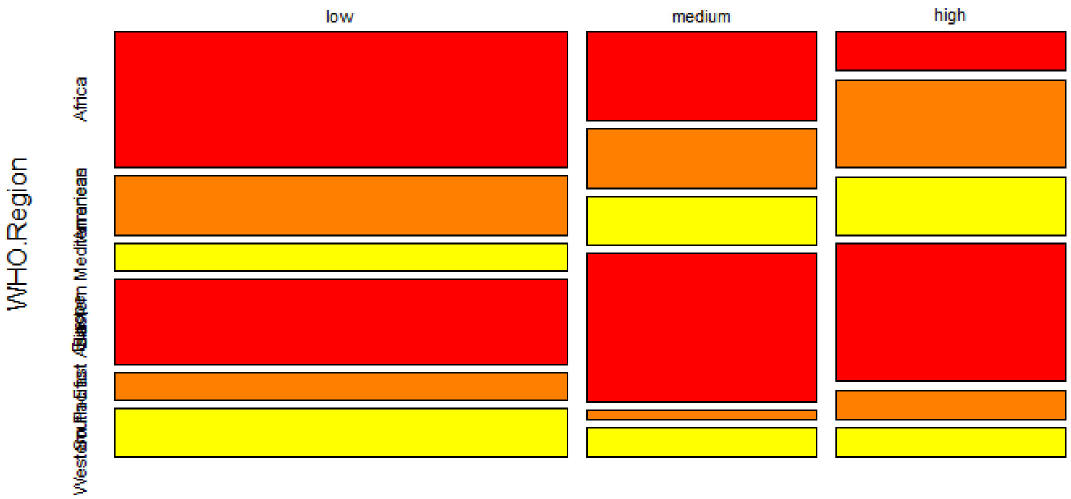
and similarly:

p(new death=medium|x)=0.0005469086

p(new death=high|x)=0.0001405501

thus, for country ‘x’, the level of new deaths will be ‘medium’ according to this model.

```
#plotting the model
plot(naive_model)
```



In this plot we see that for example, in Africa, the level of most observations would be 'low' according to this model.

Prediction:

```
#prediction

prediction1<-predict(naive_model, Train_data,type="prob")

head(cbind(prediction1, Train_data))

#the first country, has 0.5149 chance to have high new.deaths, and the

#actual new.deaths level is also high.

         low    medium       high      Country.Region Confirmed
2  0.01930447 0.4657298 0.51496573             Albania      4880
3  0.02754844 0.1572227 0.81522885             Algeria     27973
6  0.62694049 0.3263147 0.04674481 Antigua and Barbuda        86
7  0.01661260 0.1857041 0.79768334           Argentina    167416
9  0.08346614 0.5207043 0.39582954           Australia     15303
```

#model evaluation

```
prediction2<-predict(naive_model, Test_data)

confu_matrix<-confusionMatrix(prediction2,Test_data$New.deaths_binned)

confu_matrix #96% accuracy

Confusion Matrix and Statistics


          Reference

Prediction low medium high

    low     23      4    1

    medium   2     10    3

    high     1      1   11


Overall Statistics


               Accuracy : 0.7857

                 95% CI : (0.6556, 0.8841)

    No Information Rate : 0.4643

    P-Value [Acc > NIR] : 8.772e-07


                  Kappa : 0.662


 Mcnemar's Test P-Value : 0.6444


Statistics by Class:


                     Class: low Class: medium Class: high

Sensitivity              0.8846        0.6667      0.7333

Specificity              0.8333        0.8780      0.9512

Pos Pred Value           0.8214        0.6667      0.8462

Neg Pred Value           0.8929        0.8780      0.9070
```

```
Prevalence             0.4643       0.2679     0.2679

Detection Rate          0.4107       0.1786      0.1964

Detection Prevalence     0.5000       0.2679      0.2321

Balanced Accuracy       0.8590    0.7724      0.8423
```

We see 78.5% of accuracy and accepted null hypotheses of MCnemar test.

KNN Algorithm:

For this algorithm, we first need to create a separate dataframe of the variables that we are going to use for the model, and second, we need to normalize our data to have them in one scale since the algorithm calculates the Euclidean distance of the observations.

Lets create the new dataframe:

```
#choosing the subset of the dataset to run the algorithm on

data.subset<-
data[c('New.deaths_binned','New.cases','Deaths...100.Cases','Recovered...100.Cases','X1.week...increase','WHO.Region')]

head(data.subset)
```

```
New.deaths_binned New.cases Deaths...100.Cases

1           high      106            3.50

2           high      117            2.95

3           high      616            4.16

4            low       10            5.73

5         medium       18            4.32

6            low        4            3.49

Recovered...100.Cases X1.week...increase          WHO.Region

1              69.49             2.07 Eastern Mediterranean

2              56.25            17.00              Europe

3              67.34            18.07              Africa

4              88.53             2.60             Europe

5              25.47            26.84              Africa

6              75.58            13.16            Americas
```

Data normalization:

Now, lets normalize our data:

```
#normalizing the numerical variables

normalize<-function(x) {

  return((x-mean(x))/(max(x)-min(x)))

}

data.subset.norm<-as.data.frame(lapply(data.subset[2:5],normalize))

head(data.subset.norm)
```

```
New.cases Deaths...100.Cases Recovered...100.Cases

1 -0.01982670       0.016823574          0.04669465

2 -0.01963145      -0.002434129         -0.08570535

3 -0.01077388       0.039932818          0.02519465

4 -0.02153077       0.094904807          0.23709465

5 -0.02138876       0.045535059         -0.39350535

6 -0.02163727       0.016473434          0.10759465

  X1.week...increase

1      -0.050122537

2       0.014745381

3       0.019394320

4      -0.047819791

5       0.057498248

6      -0.001938665
```

#spliting the subset into training and testing datasets

```
indexes = sample(1:nrow(data.subset.norm), size = 0.3*nrow(data))

Train_data<-data.subset.norm[-indexes,] #70% training data

Test_data<-data.subset.norm[indexes,] #30% testing data
```

#creating a dataframe for 'New.deaths_binned' which is our target variable

```
Train_data_labels<-data.subset[-indexes,1]
```

```
Test_data_labels<-data.subset[indexes,1]

#installing the needed package

install.packages('class')

library(class)


#number of observations

NROW(Train_data_labels)  #131
```

Now, it’s time to build the KNN model. We try this model using k=1 and k=2 since the square root of 131 is between 11 and 12.

```
#building the KNN model

knn.11<-knn(train=Train_data, test=Test_data,cl=Train_data_labels, k=11)

knn.12<-knn(train=Train_data, test=Test_data,cl=Train_data_labels, k=12)

knn.11

knn.12

knn.11

 [1] low    low    high   low    low    low    high   low    low

[10] low    low    low    medium low    low    medium low    low

[19] low    medium high   medium low    low    low    low    low

[28] low    low    low    low    low    low    low    low    low

[37] low    low    medium low    high   low    low    high   low

[46] low    medium high   low    low    high   low    high   low

[55] low    low

Levels: low medium high

> knn.12

 [1] low    low    high   low    low    low    high   low    low

[10] low    low    low    medium low    low    low    low    low

[19] low    medium high   medium low    low    low    low    low

[28] low    low    low    low    low    medium low    low    low
```

```
[37] low    low    medium low    high   low    low    high   low
[46] low    medium high   low    low    high   low    high   low
[55] low    low
Levels: low medium high
```

We see that the levels for the test data have been predicted.

Model evaluation:

#model evaluation

```
> Acc.11<-sum(Test_data_labels==knn.11)/NROW(Test_data_labels)*100
> Acc.12<-sum(Test_data_labels==knn.12)/NROW(Test_data_labels)*100
> Acc.11  #Accuracy of the model with k=11  62.5%
[1] 62.5
> Acc.12  #Accuracy of the model with k==12  67.86%
[1] 67.86
```

Let's try the model with k=13 to see if the accuracy increases:

```
knn.13<-knn(train=Train_data, test=Test_data,cl=Train_data_labels, k=13)
Acc.13<-sum(Test_data_labels==knn.13)/NROW(Test_data_labels)*100
Acc.13  #Accuracy of the model with k==13  62.5%
```

So, the most accuracy of the model accurse with k=11.

#the confusion matrix

```
> table(knn.12,Test_data_labels)
       Test_data_labels
knn.12   low medium high
  low     19    14    9
  medium   1     3    2
  high     1     2    5
>
> library(caret)
> confusionMatrix(table(knn.12,Test_data_labels)) #Accuracy: 67.86%
Confusion Matrix and Statistics
```

```
       Test_data_labels
knn.12   low medium high
  low     19     14    9
  medium   1      3    2
  high     1      2    5

Overall Statistics

               Accuracy : 0.6786
                 95% CI : (0.3466, 0.6197)
    No Information Rate : 0.375
    P-Value [Acc > NIR] : 0.0659476

                  Kappa : 0.1928

 Mcnemar's Test P-Value : 0.67

Statistics by Class:

                     Class: low Class: medium Class: high
Sensitivity              0.9048       0.15789     0.31250
Specificity              0.3429       0.91892     0.92500
Pos Pred Value           0.4524       0.50000     0.62500
Neg Pred Value           0.8571       0.68000     0.77083
Prevalence               0.3750       0.33929     0.28571
Detection Rate           0.3393       0.05357     0.08929
Detection Prevalence     0.7500       0.10714     0.14286
Balanced Accuracy        0.6238       0.53841     0.61875
```

The accuracy of the model is 67.86% which is acceptable.

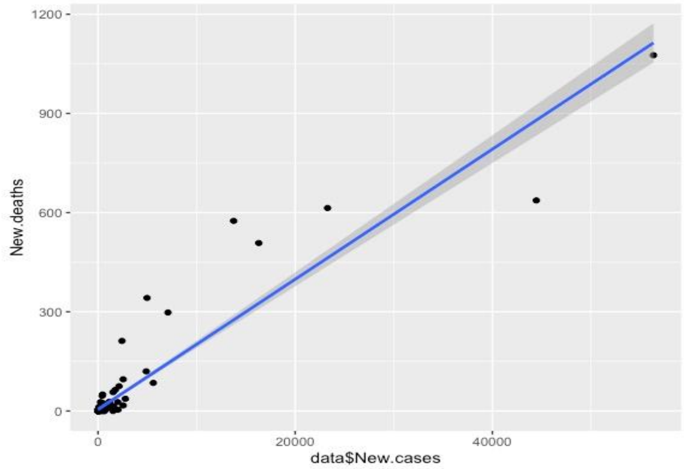
Regression:

Regression:

In this section we will find the relation between variables to see with which variables we can predict our target variable which is New deaths.

First, we will check that if there is a relationship exists between New.cases and New.deaths.

```
ggplot(data, aes(x=data$New.cases, y=New.deaths)) + geom_point()+geom_smooth(method=lm)

cor(data$New.cases, data$New.deaths)

rsq(data$New.cases,data$New.deaths)
```



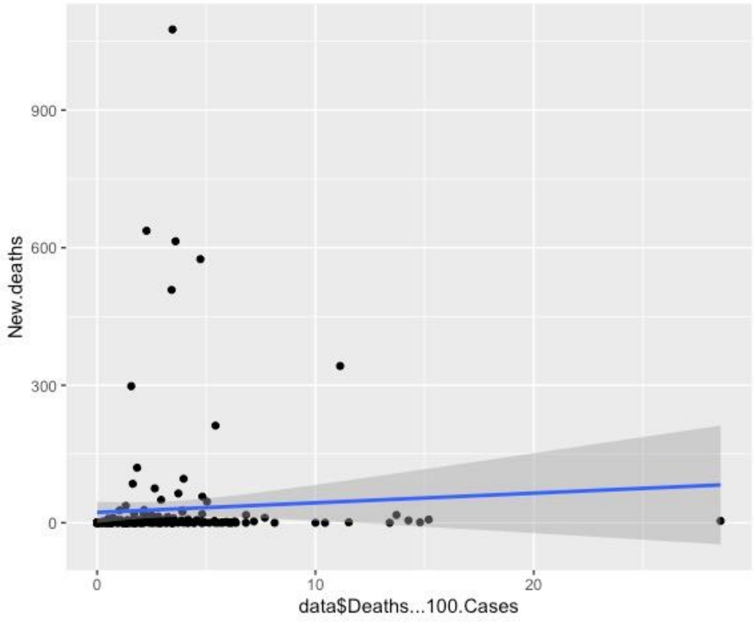
correlation is 0.935. as we can see there is a strong correlation between them.

R2 is 0.875. it means New.cases can explain variability in New.deaths by 87%

So, there is linear relationship exists between New.cases and New.deaths. When New.cases go up the New.deaths also goes up.

Then we will check if there is a relationship exists between Deaths...100.Cases and New.deaths

```
ggplot(data, aes(x=data$Deaths...100.Cases, y=New.deaths)) +
geom_point()+geom_smooth(method=lm)

cor(data$Deaths...100.Cases, data$New.deaths)

rsq(data$Deaths...100.Cases,data$New.deaths)
```



correlation is 0.060. there is no relationship between them.

R2 is 0.00364 it means Deaths...100.Cases can explain variability in New.deaths by 0036%.

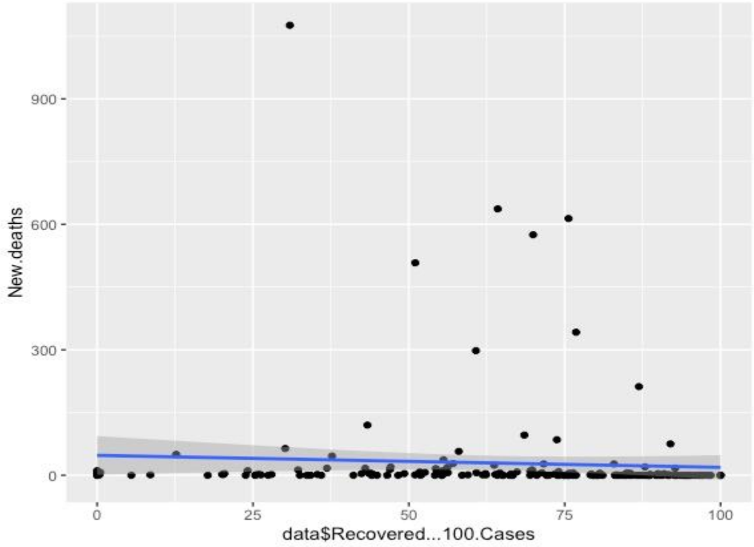
the correlation and R2 between Deaths...100.Cases and New.deaths show, there is no statistically significant linear relationship between them.

Then we will check if there is a relationship exists between Recovered...100.Cases,New.deaths

```
ggplot(data, aes(x=data$Recovered...100.Cases, y=New.deaths)) +
geom_point()+geom_smooth(method=lm)
```

```
cor(data$Recovered...100.Cases, data$New.deaths)
```

```
rsq(data$Recovered...100.Cases,data$New.deaths)
```



correlation is -0.0627. there is no relationship between them.

R2 is 0.00394. it means Recoverd...100.Cases cases can explain variability in New.deaths by 0039%.

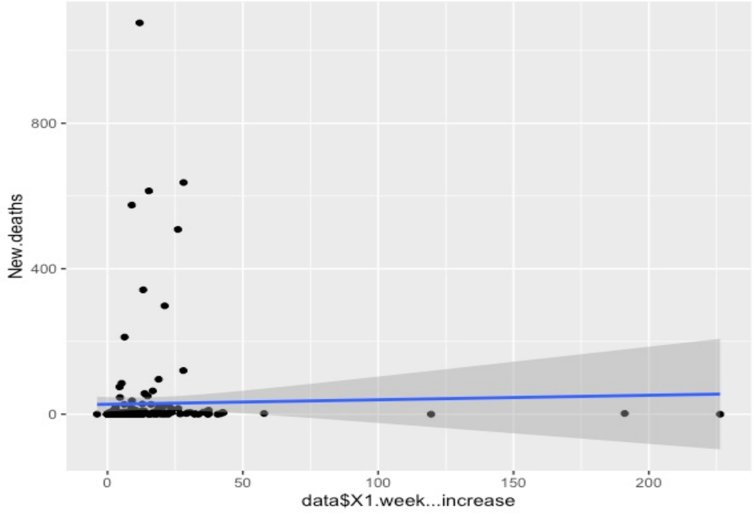
the correlation and R2 between Recovered...100.Cases,New.deaths show, there is no statistically significant linear relationship between them.

so we will check if there is a relationship exists between X1.week...increase,New.deaths.

```
ggplot(data, aes(x=data$X1.week...increase, y=New.deaths)) +
geom_point()+geom_smooth(method=lm)
```

```
cor(data$X1.week...increase, data$New.deaths)
```

```
rsq(data$X1.week...increase,data$New.deaths)
```



correlation is 0.0252. there is no relationship between them.

R2 is 0.000639. it means X1.week...increase cases can explain variability in New.deaths by 00064%.

the correlation and R2 between X1.week...increase,New.deaths show, there is no statistically significant linear relationship between them.

Let’s see if we can find any relationship between target variable and two independent variable.

First we will try relationship between New.deaths and Deaths...100.Cases + X1.week...increase.

```
multi_reg<- lm(New.deaths ~ Deaths...100.Cases + X1.week...increase, data = data)

summary(multi_reg)
```

```
Coefficients:

                      Estimate Std.   Error       t value    Pr(>|t|)

(Intercept)           19.8713         13.1525    1.511       0.133

Deaths...100.Cases    2.2580          2.5791     0.875       0.382

X1.week...increase    0.1667          0.3635      0.459      0.647


Residual standard error: 120.4 on 184 degrees of freedom

Multiple R-squared:  0.004785,  Adjusted R-squared:  -0.006032

F-statistic: 0.4424 on 2 and 184 DF,  p-value: 0.6432
```

Multiple R2 is 0.004785. there is no statistically significant linear relationship between them.

Here is the linear regression equation:

$y = y0 + m1x1 + m2x2$

$y = 19.8713 + 2.2580 x1 + 0.1667 x2$

Now let’s find relationship between New.deaths and New.cases + X1.week...increase.

```
multi_reg_2<- lm(New.deaths ~ New.cases + X1.week...increase, data = data)

summary(multi_reg_2)
```

```
Coefficients:

                    Estimate Std.  Error       t value   Pr(>|t|)

(Intercept)          5.1285753    3.6093612   1.421     0.157

New.cases            0.0196767    0.0005459   36.042    <2e-16 ***

X1.week...increase -0.0172817     0.1271945   -0.136      0.892

---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 42.5 on 184 degrees of freedom

Multiple R-squared:  0.876,       Adjusted R-squared:  0.8747

F-statistic:   650 on 2 and 184 DF,  p-value: < 2.2e-16
```

Multiple R2 is 0.876. there is linear relationship exists between New.deaths and New.cases + X1.week...increase.

Here is the linear regression equation:

y = y0 + m1x1 + m2x2

y = 5.1285753+ 0.0196767x1 -0.0172817x2

Also, relationship between New.deaths and New.cases + Deaths...100.Cases.

```
multi_reg_3<- lm(New.deaths ~ New.cases +  Deaths...100.Cases, data = data)

summary(multi_reg_3)
```

```
Coefficients:
```

```
                    Estimate Std.  Error      t value   Pr(>|t|)
(Intercept)         0.552795       4.148580   0.133     0.894
New.cases           0.019657       0.000542   36.267    <2e-16 ***
Deaths...100.Cases  1.445573       0.895987   1.613     0.108
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42.2 on 184 degrees of freedom
Multiple R-squared:  0.8777,     Adjusted R-squared:  0.8764
F-statistic: 660.4 on 2 and 184 DF,  p-value: < 2.2e-16
```

Multiple R2 is 0.8777. there is linear relationship exists between New.deaths and New.cases + Deaths...100.Cases.

Here is the linear regression equation:

y = y0 + m1x1 + m2x2

y = 0.552795+ 0.019657x1 +1.445573x2

Let's add one more variable and see the relation between New.deaths and New.cases + Deaths...100.Cases + Recovered...100.Cases

```
multi_reg_4<- lm(New.deaths ~ New.cases +  Deaths...100.Cases + Recovered...100.Cases , data = data)
summary(multi_reg_4)
```

```
Coefficients:
                       Estimate Std.  Error       t value     Pr(>|t|)
(Intercept)            -5.2605926    9.2700703    -0.567      0.5711
New.cases              0.0196860     0.0005443    36.165      <2e-16 ***
Deaths...100.Cases     1.5527667     0.9101454    1.706       0.0897 .
Recovered...100.Cases  0.0841407     0.1199441    0.701       0.4839
```

```
---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 42.26 on 183 degrees of freedom

Multiple R-squared:  0.8781,	Adjusted R-squared:  0.8761

F-statistic: 439.2 on 3 and 183 DF,  p-value: < 2.2e-16
```

Multiple R2 is 0.8781. there is linear relationship exists between New.deaths and New.cases + Deaths...100.Cases + Recovered...100.Cases.

Here is the linear regression equation:

y = (-5.2605926) + 0.0196860x1 +1.5527667x2 + 0.0841407x3

Also, we will add one more variable. New.deaths and New.cases + Deaths...100.Cases + Recovered...100.Cases + X1.week...increase

```
multi_reg_5<- lm(New.deaths ~ New.cases +  Deaths...100.Cases + Recovered...100.Cases + X1.week...increase , data = data)
```

```
Coefficients:

                        Estimate Std.     Error         t value    Pr(>|t|)

(Intercept)             -7.8713216        11.1797006    -0.704     0.4823

New.cases               0.0196857         0.0005456     36.083     <2e-16 ***

Deaths...100.Cases      1.6399919         0.9355573     1.753      0.0813 .

Recovered...100.Cases   0.1078967         0.1328641     0.812      0.4178

X1.week...increase      0.0593734         0.1414092     0.420      0.6751

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 42.36 on 182 degrees of freedom

Multiple R-squared:  0.8782,	Adjusted R-squared:  0.8755
```

```
F-statistic:   328 on 4 and 182 DF,  p-value: < 2.2e-16
```

```
summary(multi_reg_5)
```

Multiple R2 is 0.8782 there is linear relationship exists between them but is not good enough.

Here is the linear regression equation:

y = (-7.8713216)+.0196857x1 +1.6399919x2 + 0.1078967x3 +0.0593734x4

We will try another one to see if we can find any good relation or not.

Let’s try New.deaths and Deaths...100.Cases + Confirmed.last.week ,Active,Recovered...100.Cases

```
multi_reg_6<- lm(New.deaths ~  Deaths...100.Cases + Confirmed.last.week
,Active,Recovered...100.Cases, data = data)

summary(multi_reg_6)
```

```
Coefficients:

                        Estimate Std.  Error        t value   Pr(>|t|)

(Intercept)             -4.179e+00     4.273e+00    -0.978    0.334

Deaths...100.Cases      -7.492e-02     1.269e+00    -0.059    0.953

Confirmed.last.week     2.880e-04      5.527e-06    52.113    <2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 138.3 on 42 degrees of freedom

  (137 observations deleted due to missingness)

Multiple R-squared:  0.985,        Adjusted R-squared:  0.9843

F-statistic:  1376 on 2 and 42 DF,  p-value: < 2.2e-16
```

Yes, as we can see R2 increased. Multiple R2 is 0.98. there is linear relationship exists between them.

Here is the linear regression equation:

y = (-4.179)-0.07492x1 +0.0002880x2

We try another one to see any relation between New.deaths and Deaths + Confirmed.last.week ,Active,Recovered...100.Cases.

```
multi_reg_7<- lm(New.deaths ~  Deaths  + Confirmed.last.week ,Active,Recovered...100.Cases, data = data)

summary(multi_reg_7)
```

```
Coefficients:

                      Estimate Std.  Error        t value   Pr(>|t|)

(Intercept)           -2.226e+00    2.531e+00   -0.879    0.38416

Deaths                 5.115e-03    1.701e-03    3.007    0.00444 **

Confirmed.last.week   7.696e-05     7.035e-05   1.094    0.28021

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 125.4 on 42 degrees of freedom

  (137 observations deleted due to missingness)

Multiple R-squared:  0.9876,     Adjusted R-squared:  0.987

F-statistic:  1677 on 2 and 42 DF,  p-value: < 2.2e-16
```

Multiple R2 is 0.99. there is linear relationship exists between New.deaths and Deaths + Confirmed.last.week ,Active,Recovered...100.Cases.

Here is the linear regression equation:

y = (-2.226)+ 0.005115 x1 +0.00007696x2

We will pick this model to predict our target variable.

We also tried to find a relation between other variables to find a best model to predict our target variable, like:

Deaths & Recovered : R2 = 0.69

Deaths & New.cases: R2 = 0.65

Active & New.cases: R2 = 0.72

Death/100 cases & New.cases: R2 = 0.00040

Death/100 cases & Recovered/100 cases: R2 = 0.028

New.cases & Confirmed.last.week: R2 = 0.80

New cases & Recovered/100 cases: R2 = 0.0061

X1.week...increase& New.cases: R2 = 0.00094

After some explore base on R2 and p-value we found out that the best model to predict New Death is New.deaths and Deaths + Confirmed.last.week ,Active,Recovered...100.Cases.

With this equation: **y = (-2.226)+ 0.005115 x1 +0.00007696x2**

So we can predict New.deaths with Deaths and Confirmed.last.week and Active and Recovered...100.Cases variables.