Analyzing COVID-19 Data to predict Death Rate, in the Future if We have to face similar crisis like COVID-19.

STAT 31631 – Statistical Modeling

Department of Statistics & Computer
Science University of Kelaniya
Academic Year 2023/2024

By
Group - 09

Group Details

PS/2021/068 MDDI Senadheera
PS/2021/050 JANN Jayasinghe
PS/2021/169 GLL Hansamali
PS/2021/025 AJMVS Jayasinghe
PS/2021/176 MDRD Rupasighe
PS/2021/009 DSY Dissanayaka
PS/2021/111 LMIK Thilakasiri
PS/2021/069 SASC Ariyarathna
PS/2021/112 GJE Amarasinghe
PS/2021/121 SD Hendavitharana

Introduction

We know that in December 2019, the world faced a major crisis due to the COVID-19 Virus. Many people lost their lives during that time. Recently, new COVID-19 cases have also been reported in India. Our main objective is to fit a linear regression model to predict the total Deaths in case we face a similar pandemic in the future.

Objectives

1. Primary Objective:
    - Develop a linear regression model to predict the Total deaths in case we face a similar pandemic in the future.
2. Secondary Objectives:
    - To identify the most significant predictors of pandemic mortality rates.
    - Develop continent/region-specific regression model.
    - To analyze regional disparities (by continent or WHO Region) in death rates.
3. Exploratory Objectives**:**
    - To provide actionable insights for policymakers to mitigate mortality risks in Future pandemics.
    -

Novelty

Develop continent/region-specific regression model that accounts for:

- Different baseline healthcare systems (Asia vs Europe).
- Cultural factors affecting spread.

Compare coefficient differences across regions to identify the most impactful factors in different contexts.

Advantages

Identify the continent/regional health care differences and provide insights to improve healthcare system.
Provide actionable insights for policymakers to mitigate mortality risks in future pandemics.

Dataset link

https://www.kaggle.com/datasets/imdevskp/corona-virus-report

Gannt Chart

Progress	May				June				July	
	2	3	4	5	1	2	3	4	1	2
Finding data set suitable for fit a linear regression model & confirm data set under the guidance of lectures and demonstrators	🟦	🟦	🟦	🟦						
Choose a suitable topic and Objectives for the data set and select novelty for the data set discussing with group members				🟦						
Submission of the 1st report.				🟦						

Methodology

Research Design

This study employs quantitative research design using secondary data analysis of COVID-19 pandemic statistics to develop predictive models for mortality rates to if, we have to face similar pandemics like COVID-19 in the future.

Data Collection

The dataset was sourced from Kaggle: Worldometer COVID-19 data, that containing comprehensive statistics across 213 countries/regions.

Key variables include:

- Demographic data (population)
- Infection metrics (total cases, new cases)
- Outcome metrics (total deaths, new deaths)
- Recovery metrics (total recovered)
- Healthcare capacity indicators (total test conducted)
- Geographic classifications (continent, WHO region)

Data Preparation

1. Data cleaning:
    - Handling missing values
        - We decided to remove some variables (NewCases, NewRecovered, NewDeaths, ActiveCases) which cases lot of missing values and there are variables for these variables with totals (TotalCases, TotalDeaths & TotalRecovered). So, removing NewCases, NewRecovered, NewDeaths & ActiveCases make no impact.
        - But we have removed the variable Series.Critical only because of the missing values.
        - After we remove data with missing values.
    - We didn’t remove outliers because of real world healthcare data.
    -
2. Data Splitting:
    - 80% training set for model development
    - 20% test set for model validation

Analytical Approach for the Objectives

1. Global Model Development (Primary Objective):
   - Multiple linear regression to predict total deaths.
   - Identify potential predictors: total cases, population, test/million, healthcare indicators.
   - Identify model diagnostics: multicollinearity, heteroscedasticity, normality of residuals.
2. Secondary Objective:
   - Separate regression models for region/continent/WHO region-Specific Models
   - Interaction terms between key predictors and region indicators.
3. Model Evaluation:
   - R-squared & adjusted R-squared for Goodness-of-fit
   - Comparison of global & regional/continent model performance
   - Cross-validation to assess generalizability
4. Secondary Analysis:
   - ANOVA to examine regional/continent disparities in death rates
   - Correlation analysis to identify significant predictors
   - Visualization of reginal patterns in mortality

Ethical Consideration

1. Proper attribution to original data sources
2. Avoidance of causal claims where inappropriate
3. Transparent reporting of limitations

Descriptive analysis

Overview of Dataset

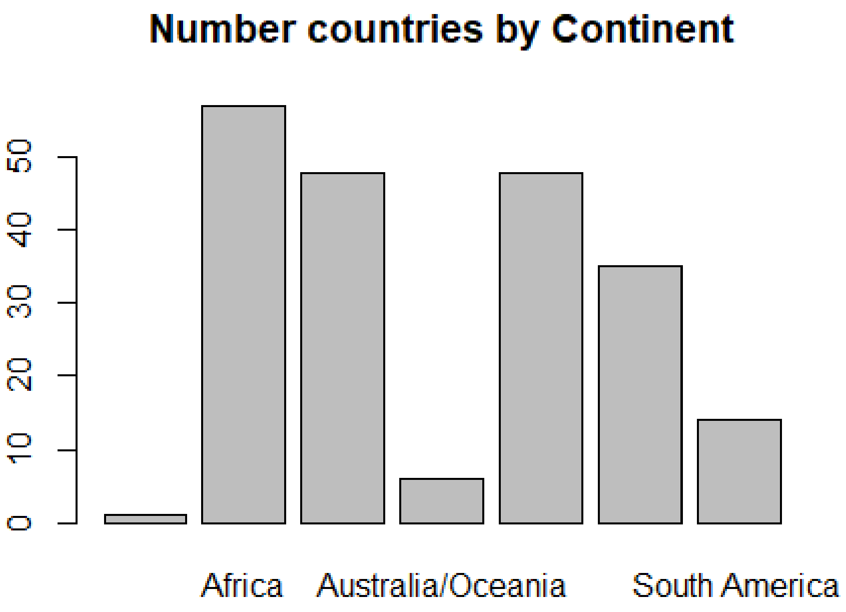
The data set contains records for 209 countries/regions with 15 variables each. Preliminary examination reveals:

```
worldometer_data <- read.csv("worldometer_data.csv")
head(worldometer_data)
```

```
##   Country.Region     Continent Population TotalCases NewCases TotalDeaths
## 1            USA North America  331198130    5032179       NA      162804
## 2         Brazil South America  212710692    2917562       NA       98644
## 3          India          Asia 1381344997    2025409       NA       41638
## 4         Russia        Europe  145940924     871894       NA       14606
## 5   South Africa        Africa   59381566     538184       NA        9604
## 6         Mexico North America  129066160     462690     6590       50517
##   NewDeaths TotalRecovered NewRecovered ActiveCases Serious.Critical
## 1        NA        2576668           NA     2292707            18296
## 2        NA        2047660           NA      771258             8318
## 3        NA        1377384           NA      606387             8944
## 4        NA         676357           NA      180931             2300
## 5        NA         387316           NA      141264              539
## 6       819         308848         4140      103325             3987
##   Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop      WHO.Region
## 1            15194           492   63139605       190640        Americas
## 2            13716           464   13206188        62085        Americas
## 3             1466            30   22149351        16035 South-EastAsia
## 4             5974           100   29716907       203623          Europe
## 5             9063           162    3149807        53044          Africa
## 6             3585           391    1056915         8189        Americas
```

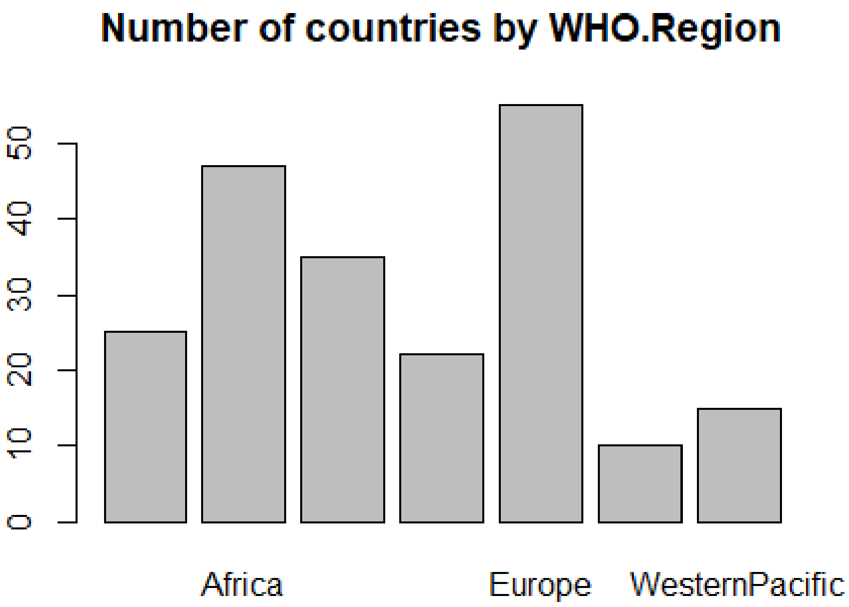
Geographic Distribution

```
barplot(table(worldometer_data$Continent), main = "Number countries by Continent")
```



```
table(worldometer_data$Continent)

##
##




      Africa          Asia Australia/Oceania
##            1           57            48            6


##         Europe    North America    South America
##            48            35            14
```

```
barplot(table(worldometer_data$WHO.Region),main = "Number of countries by WHO.Region")
```

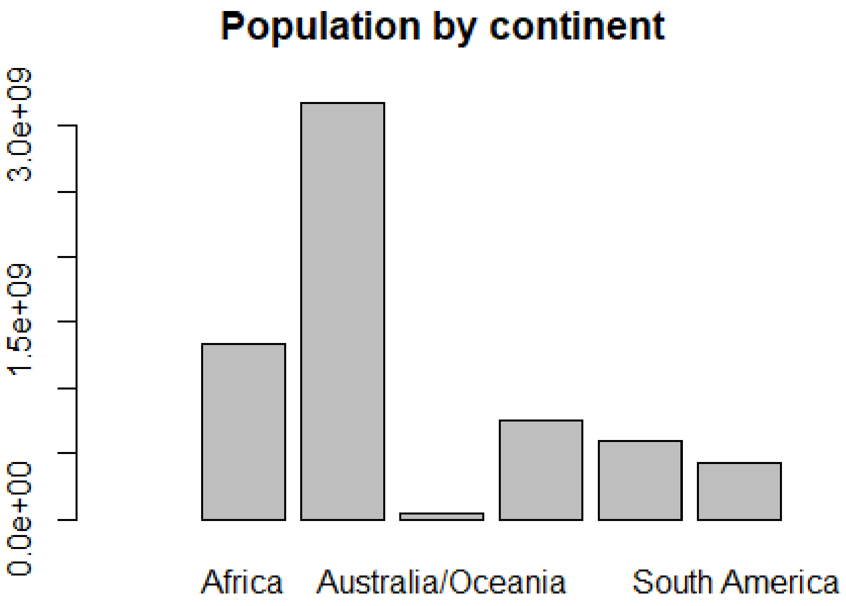


```
table(worldometer_data$WHO.Region)

##
##                          Africa           Americas
##                 25            47                 35

## EasternMediterranean       Europe      South-EastAsia
##                 22              55              10

##      WesternPacific
##                 15
```

Population Coverage

population by continent

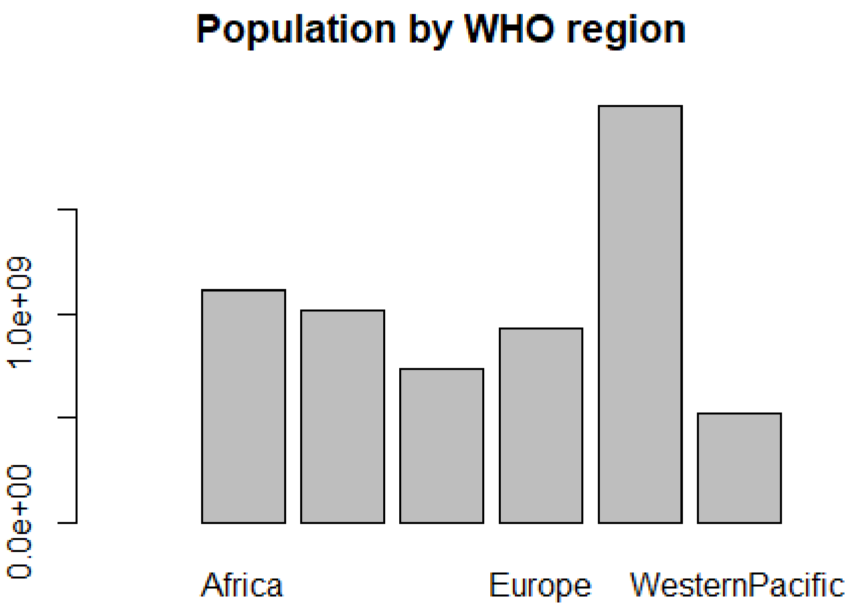
```
pop  by  continent <- tapply(worldometer  data$Population, worldometer_data$Continent, sum)
barplot(pop_by_continent,main = "Population by continent")
```



```
pop_by_continent

##                  Africa          Asia         Australia/Oceania
##         NA    1343515489    3173656415       40957909


##       Europe       North America           South America
##     747677546      589503467             431110464
```

Population by WHO region

```
barplot(tapply(worldometer_data$Population,worldometer_data$WHO.Region,sum),main = "Popula
tion by WHO region")
```



```
tapply(worldometer_data$Population,worldometer_data$WHO.Region,sum)

##                              Africa                   Americas
##                 NA         1118461393             1018879504


## EasternMediterranean              Europe                  South-EastAsia
##           732007690              927733876                 1997512597


##       WesternPacific
##           522144861

population  <- na.omit(worldometer_data$Population)
range(population)

## [1]       801 1381344997
```

population range is 802(Vatican City) to 1.38 billion (India)

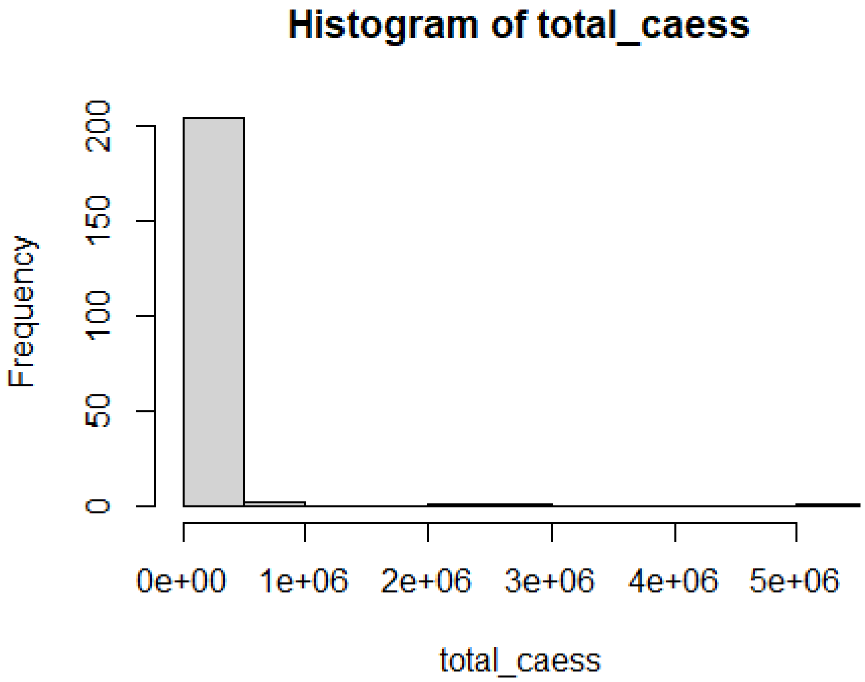
Key Variable Distribution

Total Cases:

```
total_caess <- na.omit(worldometer_data$TotalCases)
range(total_caess)

## [1]      10 5032179
```

Global range : 10(western Sahara) to 5032179 (USA)

```
mean(total_caess)

## [1] 91718.5

hist(total_caess)
```



we can see highly skewed distribution, most countries below 100000 cases

Total deaths

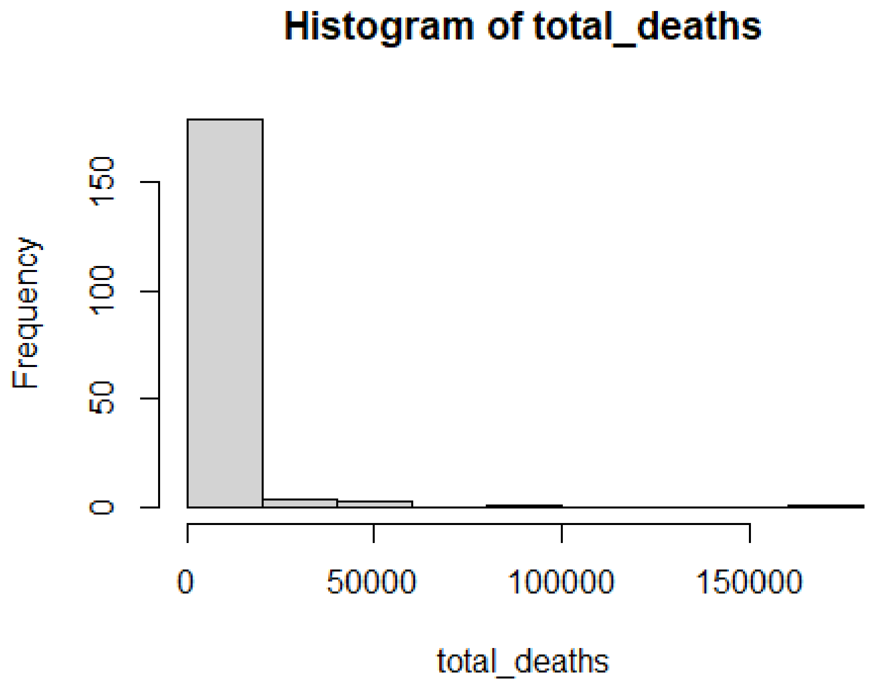
```
total  deaths <- na.omit(worldometer_data$TotalDeaths)
range(total_deaths)

## [1]      1 162804
```

Global death rage in between 0 to 162804

```
mean(worldometer_data$TotalDeaths,na.rm = TRUE)

## [1] 3792.59
```

mean death in country is 3792

```
hist(total_deaths)
```



#

```
count = (total_deaths[total_deaths<=1000])

(nrow(as.matrix(count))/209)*100

## [1] 68.42105
```

Nearly 68.4% of countries reported <1000 deaths

Case Fatality Rate

```
corona <- worldometer_data[,-c(5,7,9,10,11)]
df <- na.omit(corona)
CFR <-(df$TotalDeaths / df$TotalCases) * 100
mean(CFR)

## [1] 3.036697
```

Global Case Fatality Rate is 3.03%

```
range(CFR)

## [1]  0.04949134 28.73303167
```

range of Case Fatality Rate is 0 to 28.7 %

Regional Disparities

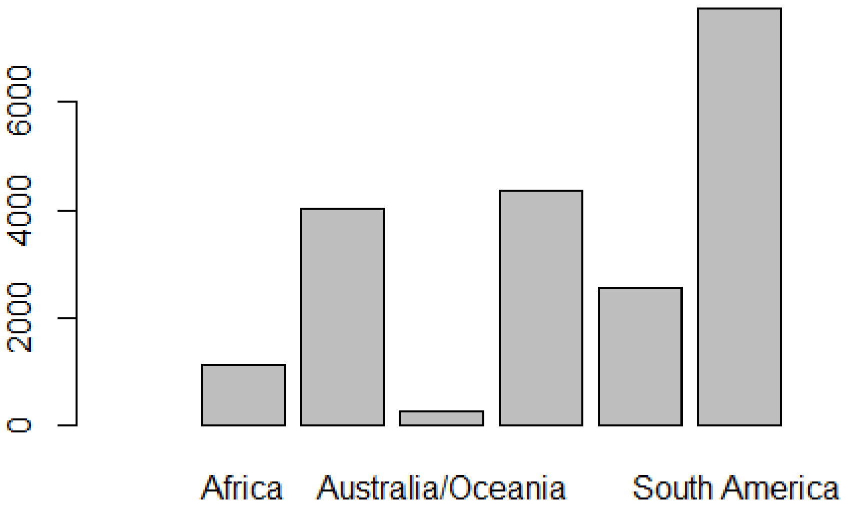
Cases per Million Population

```
cases <- worldometer_data$Tot.Cases.1M.pop
range(cases,na.rm = TRUE)

## [1]     3 39922
```

Cases per Million Population Highest is Quater (39922) and lowest is Laos (3)

Continent Averages

```
tapply(worldometer_data$Tot.Cases.1M.pop,worldometer_data$Continent,mean)

##                           Africa              Asia Australia/Oceania
##               NA        1130.807          4008.938           241.000
##           Europe   North America     South America
##         4363.625        2529.914          7745.786

barplot(tapply(worldometer_data$Tot.Cases.1M.pop,worldometer_data$Continent,mean))
```



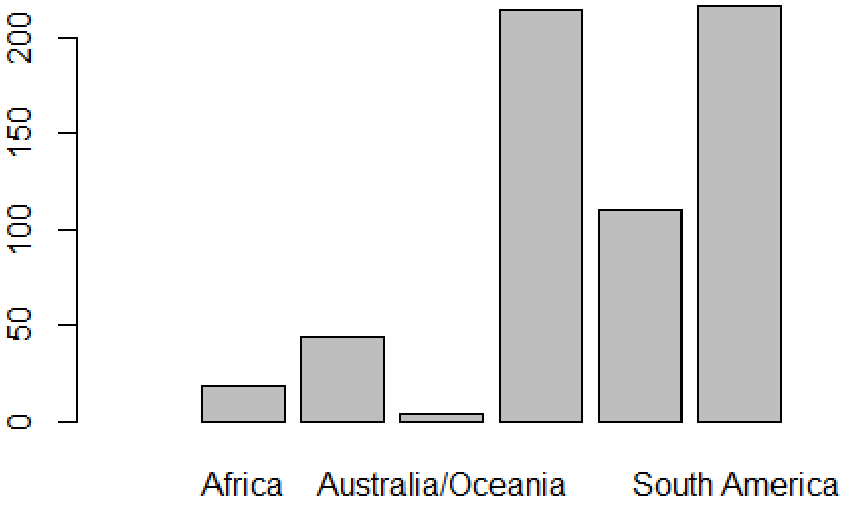
Deaths per Million Population

```
deaths <- worldometer_data$Deaths.1M.pop
range(deaths,na.rm = TRUE)

## [1]    0.08 1238.00
```

Deaths per Million Population Highest is San Marino(1238) and lowest is Burundi(0.08)

Continental averages

```
tapply(worldometer_data$Deaths.1M.pop,worldometer_data$Continent,mean,na.rm = TRUE)

##                       Africa          Asia Australia/Oceania 
##           NaN       18.24145       43.97143          3.82500 
##        Europe    North America     South America 
##     214.95556       110.60714       216.76923

barplot(tapply(worldometer_data$Deaths.1M.pop,worldometer_data$Continent,mean,na.rm = TRU
E))
```



Healthcare Capacity Indicators

Total testing

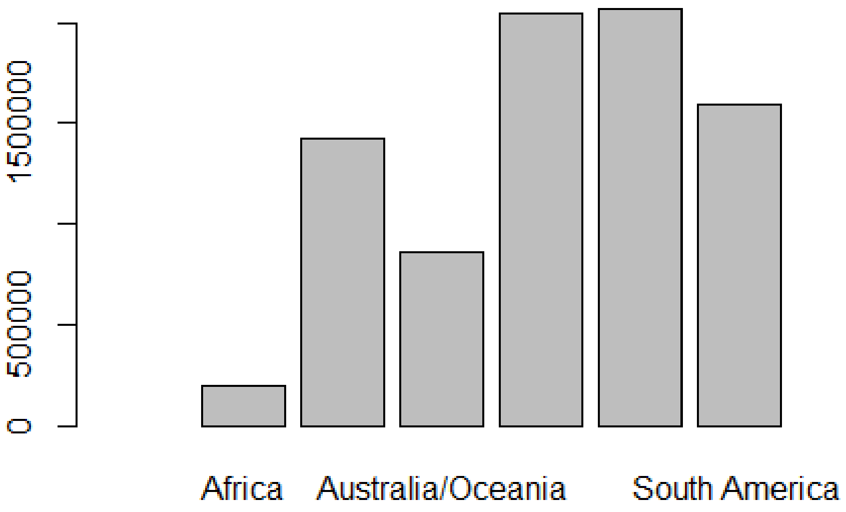
```
test <- worldometer  data$TotalTests
range(test,na.rm = TRUE)

## [1]       61 63139605
```

USA has the highest Healthcare Capacity over 63139605 tests

Montserrat has the lowest Healthcare Capacity which is only 61 test

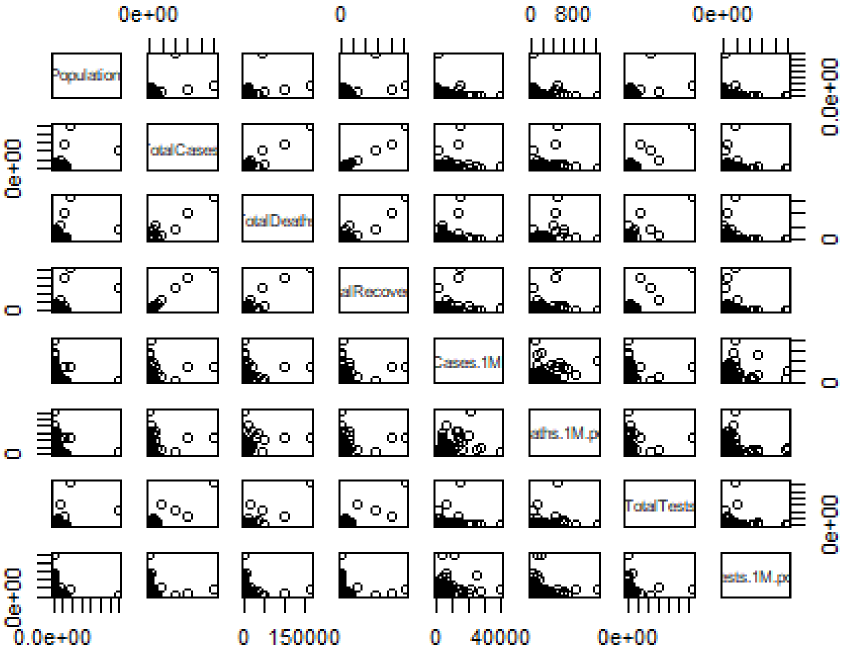
Continent averages

```
tapply(worldometer_data$TotalTests,worldometer_data$Continent,mean,na.rm = TRUE)

##                         Africa            Asia Australia/Oceania
##              NaN        197133.0        1420735.2         858801.8
##           Europe     North America     South America
##        2045225.8         2063928.9         1598544.1

barplot(tapply(worldometer_data$TotalTests,worldometer_data$Continent,mean,na.rm = TRUE))
```



Correlations

```
correlation <- cor(na.omit(corona[,-c(1,2,11)]))
correlation

##                  Population TotalCases TotalDeaths TotalRecovered
## Population       1.00000000 0.54338501  0.43795429     0.58785407
## TotalCases       0.54338501 1.00000000  0.95496099     0.98568247
## TotalDeaths      0.43795429 0.95496099  1.00000000     0.93513544
## TotalRecovered   0.58785407 0.98568247  0.93513544     1.00000000
## Tot.Cases.1M.pop -0.02410002 0.24560317  0.22719400     0.26220930
## Deaths.1M.pop    0.02043201 0.28430834  0.38474649     0.29052272
## TotalTests       0.50173948 0.90460050  0.84343006     0.86933559
## Tests.1M.pop    -0.07844237 0.03969952  0.02976143     0.03722018
##                 Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop
## Population          -0.02410002   0.02043201  0.5017395  -0.07844237
## TotalCases           0.24560317   0.28430834  0.9046005   0.03969952
## TotalDeaths          0.22719400   0.38474649  0.8434301   0.02976143
## TotalRecovered       0.26220930   0.29052272  0.8693356   0.03722018
## Tot.Cases.1M.pop     1.00000000   0.50610859  0.1848900   0.31844108
## Deaths.1M.pop        0.50610859   1.00000000  0.2224010   0.13493032
## TotalTests           0.18489000   0.22240101  1.0000000   0.10382784
## Tests.1M.pop         0.31844108   0.13493032  0.1038278   1.00000000

pairs(na.omit(corona[,-c(1,2,11)]))
```



Correlation of population

```
correlation[,1]

##       Population       TotalCases      TotalDeaths   TotalRecovered
##       1.00000000       0.54338501       0.43795429       0.58785407
## Tot.Cases.1M.pop    Deaths.1M.pop       TotalTests     Tests.1M.pop
##      -0.02410002       0.02043201       0.50173948      -0.07844237
```

Correlation of TotalCases

```
correlation[,2]

##       Population       TotalCases      TotalDeaths   TotalRecovered
##       0.54338501       1.00000000       0.95496099       0.98568247
## Tot.Cases.1M.pop    Deaths.1M.pop       TotalTests     Tests.1M.pop
##       0.24560317       0.28430834       0.90460050       0.03969952
```

Correlation of TotalDeaths

```
correlation[,3]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##      0.43795429      0.95496099      1.00000000      0.93513544 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##      0.22719400      0.38474649      0.84343006      0.02976143
```

Correlation of TotalRecovered

```
correlation[,4]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##      0.58785407      0.98568247      0.93513544      1.00000000 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##      0.26220930      0.29052272      0.86933559      0.03722018
```

Correlation of Tot.Cases.1M.pop

```
correlation[,5]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##     -0.02410002      0.24560317      0.22719400      0.26220930 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##      1.00000000      0.50610859      0.18489000      0.31844108
```

Correlation of Deaths.1M.pop

```
correlation[,6]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##      0.02043201      0.28430834      0.38474649      0.29052272 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##      0.50610859      1.00000000      0.22240101      0.13493032
```

Correlation of TotalTests

```
correlation[,7]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##       0.5017395       0.9046005       0.8434301       0.8693356 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##       0.1848900       0.2224010       1.0000000       0.1038278
```

Correlation of Tests.1M.pop

```
correlation[,8]
```

```
##      Population      TotalCases     TotalDeaths  TotalRecovered 
##     -0.07844237      0.03969952      0.02976143      0.03722018 
## Tot.Cases.1M.pop   Deaths.1M.pop      TotalTests    Tests.1M.pop 
##      0.31844108      0.13493032      0.10382784      1.00000000
```

Results and discussion

```
#install.packages("caTools")
#install.packages("car")
#install.packages("quantmod")
#install.packages("MASS")
#install.packages("corrplot")
#install.packages("leaps")
#install.packages("Metrics")
#install.packages("tidyr")
#install.packages("dplyr")
#install.packages("randtests")
#install.packages("lmtest")
#install.packages("nlme")

library(caTools)

## Warning: package 'caTools' was built under R version 4.4.3

library(car)

## Warning: package 'car' was built under R version 4.4.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.4.3

library(quantmod)

## Warning: package 'quantmod' was built under R version 4.4.3

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.4.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.4.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.4.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(MASS)

## Warning: package 'MASS' was built under R version 4.4.3
```

```r
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.4.3

## corrplot 0.95 loaded

library(Metrics)

## Warning: package 'Metrics' was built under R version 4.4.3

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.4.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.3

## 
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################

## 
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
## 
##     select

## The following objects are masked from 'package:xts':
## 
##     first, last

## The following object is masked from 'package:car':
## 
##     recode

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

```
library(randtests)
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.4.3
```

```
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 4.4.3
```

```
## 
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
## 
##     collapse
```

```
worldometer_data <- read.csv("worldometer_data.csv")
head(worldometer_data)
```

```
##   Country.Region     Continent Population TotalCases NewCases TotalDeaths
## 1            USA North America  331198130    5032179       NA      162804
## 2         Brazil South America  212710692    2917562       NA       98644
## 3          India          Asia 1381344997    2025409       NA       41638
## 4         Russia        Europe  145940924     871894       NA       14606
## 5   South Africa        Africa   59381566     538184       NA        9604
## 6         Mexico North America  129066160     462690     6590       50517
##   NewDeaths TotalRecovered NewRecovered ActiveCases Serious.Critical
## 1        NA        2576668           NA     2292707            18296
## 2        NA        2047660           NA      771258             8318
## 3        NA        1377384           NA      606387             8944
## 4        NA         676357           NA      180931             2300
## 5        NA         387316           NA      141264              539
## 6       819         308848         4140      103325             3987
##   Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop     WHO.Region
## 1            15194           492   63139605       190640       Americas
## 2            13716           464   13206188        62085       Americas
## 3             1466            30   22149351        16035 South-EastAsia
## 4             5974           100   29716907       203623         Europe
## 5             9063           162    3149807        53044         Africa
## 6             3585           391    1056915         8189       Americas
```

```
names(worldometer_data)
```

```
##  [1] "Country.Region"   "Continent"        "Population"       "TotalCases"      
##  [5] "NewCases"         "TotalDeaths"      "NewDeaths"        "TotalRecovered"  
##  [9] "NewRecovered"     "ActiveCases"      "Serious.Critical" "Tot.Cases.1M.pop"
## [13] "Deaths.1M.pop"    "TotalTests"       "Tests.1M.pop"     "WHO.Region"
```

```
str(worldometer_data)
```

```
## 'data.frame':    209 obs. of  16 variables:
##  $ Country.Region  : chr  "USA" "Brazil" "India" "Russia" ...
##  $ Continent       : chr  "North America" "South America" "Asia" "Europe" ...
##  $ Population      : int  331198130 212710692 1381344997 145940924 59381566 129066160 3301
6319 19132514 50936262 46756648 ...
```

```
##  $ TotalCases     : int  5032179 2917562 2025409 871894 538184 462690 455409 366671 357710
354530 ...
##  $ NewCases       : int  NA NA NA NA NA 6590 NA NA NA NA ...
##  $ TotalDeaths    : int  162804 98644 41638 14606 9604 50517 20424 9889 11939 28500 ...
##  $ NewDeaths      : int  NA NA NA NA NA 819 NA NA NA NA ...
##  $ TotalRecovered : int  2576668 2047660 1377384 676357 387316 308848 310337 340168 1923
55 NA ...
##  $ NewRecovered   : int  NA NA NA NA NA 4140 NA NA NA NA ...
##  $ ActiveCases    : int  2292707 771258 606387 180931 141264 103325 124648 16614 153416 N
A ...
##  $ Serious.Critical: int  18296 8318 8944 2300 539 3987 1426 1358 1493 617 ...
##  $ Tot.Cases.1M.pop: int  15194 13716 1466 5974 9063 3585 13793 19165 7023 7582 ...
##  $ Deaths.1M.pop  : num  492 464 30 100 162 391 619 517 234 610 ...
##  $ TotalTests     : int  63139605 13206188 22149351 29716907 3149807 1056915 2493429 17606
15 1801835 7064329 ...
##  $ Tests.1M.pop   : int  190640 62085 16035 203623 53044 8189 75521 92022 35374 151087 ...
##  $ WHO.Region     : chr  "Americas" "Americas" "South-EastAsia" "Europe" ...
```

We dicided to remove NewCases, NewDeaths, NewRecovered because of they have lot of missing values and there are variables with totals (TotalCases, TotalDeaths ,TotalRecovered).

```r
corona <- worldometer_data[,-c(1,5,7,9)]
head(corona)
```

```
##       Continent Population TotalCases TotalDeaths TotalRecovered ActiveCases
## 1 North America  331198130    5032179      162804        2576668     2292707
## 2 South America  212710692    2917562       98644        2047660      771258
## 3          Asia 1381344997    2025409       41638        1377384      606387
## 4        Europe  145940924     871894       14606         676357      180931
## 5        Africa   59381566     538184        9604         387316      141264
## 6 North America  129066160     462690       50517         308848      103325
##   Serious.Critical Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop
## 1            18296            15194           492   63139605       190640
## 2             8318            13716           464   13206188        62085
## 3             8944             1466            30   22149351        16035
## 4             2300             5974           100   29716907       203623
## 5              539             9063           162    3149807        53044
## 6             3987             3585           391    1056915         8189
##       WHO.Region
## 1       Americas
## 2       Americas
## 3 South-EastAsia
## 4         Europe
## 5         Africa
## 6       Americas
```

Total number of missing values

```
sum(is.na(corona))

## [1] 176

n = nrow(corona)
```

missing values percentage of Serious.critical cases

```
(sum(is.na(corona$Serious.Critical)) / n) * 100

## [1] 41.62679
```

missing values percentage of ActiveCases cases

```
(sum(is.na(corona$ActiveCases)) / n) * 100

## [1] 1.913876
```

missing values percentage of Population cases

```
(sum(is.na(corona$Population)) / n) * 100

## [1] 0.4784689
```

missing values percentage of TotalCases

```
(sum(is.na(corona$TotalCases)) / n) * 100

## [1] 0
```

missing values percentage of TotalDeaths

```
(sum(is.na(corona$TotalDeaths)) / n) * 100

## [1] 10.04785
```

missing values percentage of TotalRecovered

```
(sum(is.na(corona$TotalRecovered)) / n) * 100

## [1] 1.913876
```

missing values percentage of Tot.Cases.1M.pop

```
(sum(is.na(corona$Tot.Cases.1M.pop)) / n) * 100

## [1] 0.4784689
```

missing values percentage of Deaths.1M.pop

```r
(sum(is.na(corona$Deaths.1M.pop)) / n) * 100
```

```
## [1] 10.52632
```

missing values percentage of TotalTests

```r
(sum(is.na(corona$TotalTests)) / n) * 100
```

```
## [1] 8.61244
```

missing values percentage of Tests.1M.pop

```r
(sum(is.na(corona$Tests.1M.pop)) / n) * 100
```

```
## [1] 8.61244
```

missing values percentage of WHO.Region

```r
(sum(is.na(corona$WHO.Region)) / n) * 100
```

```
## [1] 0
```

missing values percentage of Continent

```r
(sum(is.na(corona$Continent)) / n) * 100
```

```
## [1] 0
```

```r
set.seed(68)
```

We can see that Serious.Critical has high missing value percentage(over 41.%), So we cannot simply remove the missing values. So we have to use imputation method like,

1. Mean Imputation

2. Median Imputation

Lets first use mean Imputation for Serious.Critical

```
corona1 <- corona
corona1$Serious.Critical[is.na(corona1$Serious.Critical)] <- mean(corona1$Serious.Critical, na.rm =
TRUE)
(sum(is.na(corona1$Serious.Critical)) / n) * 100

## [1] 0

sum(is.na(corona1))

## [1] 89

df1 <- na.omit(corona1)
sum(is.na(df1))

## [1] 0

nrow(df1)

## [1] 169

names(df1)

##  [1] "Continent"       "Population"      "TotalCases"      "TotalDeaths"
##  [5] "TotalRecovered"   "ActiveCases"      "Serious.Critical" "Tot.Cases.1M.pop"
##  [9] "Deaths.1M.pop"    "TotalTests"       "Tests.1M.pop"     "WHO.Region"
```

set split data set (80/20)

```
split.ratio <- 0.8
```

Split the dataset into training and test sets

```
df1_bound <- ceiling(nrow(df1)*split.ratio)
df1_bound

## [1] 136
```

training data part

```
train1 = df1 %>% slice_sample(n = df1_bound, replace = FALSE)
head(train1)
```

```
##   Continent Population TotalCases TotalDeaths TotalRecovered ActiveCases
## 1    Europe    5794279      14306         617          12787         902
## 2    Africa   59381566     538184        9604         387316      141264
## 3      Asia   40306025     140603        5161         101025       34417
## 4    Europe    3278650      13396         384           7042        5970
## 5      Asia   69817894       3330          58           3148         124
## 6    Europe     174022        597          47            533          17
##   Serious.Critical Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop
## 1           2.0000             2469         106.0    1654512       285542
## 2         539.0000             9063         162.0    3149807        53044
## 3         517.0000             3488         128.0    1092741        27111
## 4         534.3934             4086         117.0     147021        44842
## 5           1.0000               48           0.8     749213        10731
## 6         534.3934             3431         270.0      30721       176535
##            WHO.Region
## 1              Europe
## 2              Africa
## 3 EasternMediterranean
## 4              Europe
## 5      South-EastAsia
## 6
```

```
nrow(train1)
```

```
## [1] 136
```

Test data part

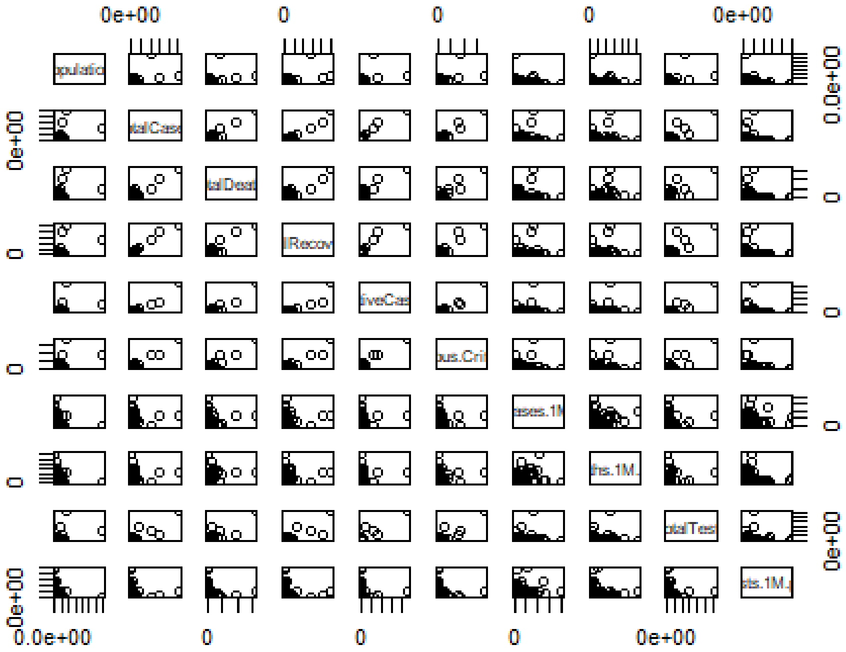
```
test1 <- df1[-as.numeric(rownames(train1)),]
head(test1)
```

```
##         Continent Population TotalCases TotalDeaths TotalRecovered ActiveCases
## 152        Africa     219544        878          15            797          66
## 153        Africa    2356075        804           2             63         739
## 154 North America     393616        761          14             91         656
## 155          Asia   97425470        747          10            392         345
## 156        Africa    2143943        742          23            175         544
## 158        Europe      33938        699          42            657           0
##     Serious.Critical Tot.Cases.1M.pop Deaths.1M.pop TotalTests Tests.1M.pop
## 152         534.3934             3999          68.0       3079        14025
## 153           1.0000              341           0.8      68423        29041
## 154           1.0000             1933          36.0       4814        12230
## 155         534.3934                8           0.1     482456         4952
## 156         534.3934              346          11.0       8771         4091
## 158         534.3934            20596        1238.0       6068       178797
##         WHO.Region
## 152         Africa
## 153         Africa
## 154       Americas
## 155 WesternPacific
## 156        Africa
## 158        Europe
```

```
names(df1)
```

```
##  [1] "Continent"        "Population"       "TotalCases"       "TotalDeaths"
##  [5] "TotalRecovered"   "ActiveCases"      "Serious.Critical" "Tot.Cases.1M.pop"
##  [9] "Deaths.1M.pop"    "TotalTests"       "Tests.1M.pop"     "WHO.Region"
```

```
pairs(df1[-c(1,12)])
```



fit the model for mean imputation using Stepwise regression procedure

```
model1_step <- lm(TotalDeaths ~ ., data = train1)

step_model1 <- stepAIC(model1_step, direction = "both", trace = TRUE)

## Start:  AIC=-6457.13
## TotalDeaths ~ Continent + Population + TotalCases + TotalRecovered +
##     ActiveCases + Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop +
##     TotalTests + Tests.1M.pop + WHO.Region

## Warning: attempting model selection on an essentially perfect fit is nonsense

##                    Df  Sum of Sq       RSS     AIC
## - WHO.Region        6          0         0 -6460.7
## - Tests.1M.pop      1          0         0 -6458.7
## - Serious.Critical  1          0         0 -6458.7
## - TotalTests        1          0         0 -6458.1
## <none>                                   0 -6457.1
## - Tot.Cases.1M.pop  1          0         0 -6455.4
## - Deaths.1M.pop     1          0          0 -6454.9
## - Continent         5          0          0 -6148.1
## - Population        1          0          0 -6080.5
## - ActiveCases       1 1526605407 1526605407  2247.8
## - TotalRecovered    1 1600606326 1600606326  2254.2
## - TotalCases        1 1681982098 1681982098  2261.0
## 
## Step:  AIC=-6460.74
## TotalDeaths ~ Continent + Population + TotalCases + TotalRecovered + 
##     ActiveCases + Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + 
##     TotalTests + Tests.1M.pop
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                    Df  Sum of Sq        RSS     AIC
## - Tests.1M.pop      1          0          0 -6462.5
## - Serious.Critical  1          0          0 -6462.3
## - TotalTests        1          0          0 -6462.1
## <none>                                    0 -6460.7
## - Tot.Cases.1M.pop  1          0          0 -6460.3
## - Deaths.1M.pop     1          0          0 -6459.3
## + WHO.Region        6          0          0 -6457.1
## - Continent         5          0          0 -6149.0
## - Population        1          0          0 -6090.0
## - ActiveCases       1 1546654973 1546654973  2237.6
## - TotalRecovered    1 1625228201 1625228201  2244.3
## - TotalCases        1 1707028664 1707028664  2251.0
## 
## Step:  AIC=-6462.45
## TotalDeaths ~ Continent + Population + TotalCases + TotalRecovered + 
##     ActiveCases + Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + 
##     TotalTests
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                    Df  Sum of Sq        RSS     AIC
## - Serious.Critical  1          0          0 -6464.1
## - TotalTests        1          0          0 -6464.0
## <none>                                    0 -6462.5
## - Tot.Cases.1M.pop  1          0          0 -6462.2
## - Deaths.1M.pop     1          0          0 -6461.2
## + Tests.1M.pop      1          0          0 -6460.7
## + WHO.Region        6          0          0 -6458.7
## - Continent         5          0          0 -6150.7
## - Population        1          0          0 -6091.8
## - ActiveCases       1 1556761966 1556761966  2236.4
## - TotalRecovered    1 1632407359 1632407359  2242.9
## - TotalCases        1 1716995134 1716995134  2249.8
## 
## Step:  AIC=-6464.07
## TotalDeaths ~ Continent + Population + TotalCases + TotalRecovered + 
##     ActiveCases + Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests
## Warning: attempting model selection on an essentially perfect fit is nonsense
## Warning: attempting model selection on an essentially perfect fit is nonsense

##                     Df  Sum of Sq        RSS     AIC
## - TotalTests         1          0          0 -6465.7
## <none>                                     0 -6464.1
## - Tot.Cases.1M.pop   1          0          0 -6463.9
## - Deaths.1M.pop      1          0          0 -6462.8
## + Serious.Critical   1          0          0 -6462.5
## + Tests.1M.pop       1          0          0 -6462.3
## + WHO.Region         6          0          0 -6460.4
## - Continent          5          0          0 -6152.2
## - Population         1          0          0 -6091.9
## - ActiveCases        1 1824219901 1824219901  2256.0
## - TotalRecovered     1 1839046908 1839046908  2257.1
## - TotalCases         1 1998614544 1998614544  2268.4
##
## Step:  AIC=-6465.73
## TotalDeaths ~ Continent + Population + TotalCases + TotalRecovered +
##     ActiveCases + Tot.Cases.1M.pop + Deaths.1M.pop

## Warning: attempting model selection on an essentially perfect fit is nonsense
## Warning: attempting model selection on an essentially perfect fit is nonsense

##                     Df  Sum of Sq        RSS     AIC
## <none>                                     0 -6465.7
## - Tot.Cases.1M.pop   1          0          0 -6465.7
## - Deaths.1M.pop      1          0          0 -6464.5
## + TotalTests         1          0          0 -6464.1
## + Serious.Critical   1          0          0 -6464.0
## + Tests.1M.pop       1          0          0 -6463.8
## + WHO.Region         6          0          0 -6461.7
## - Continent          5          0          0 -6154.0
## - Population         1          0          0 -6087.7
## - ActiveCases        1 1824750340 1824750340  2254.0
## - TotalRecovered     1 1849163447 1849163447  2255.8
## - TotalCases         1 2002111751 2002111751  2266.7
```

```
summary(step_model1)
```

```
##
## Call:
## lm(formula = TotalDeaths ~ Continent + Population + TotalCases +
##     TotalRecovered + ActiveCases + Tot.Cases.1M.pop + Deaths.1M.pop,
##     data = train1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.308e-10 -2.005e-11 -7.400e-14  7.921e-12  2.855e-10
##
## Coefficients:
##                             Estimate Std. Error    t value Pr(>|t|)
## (Intercept)               -7.299e-12  7.805e-12 -9.350e-01 0.351517
## ContinentAsia              6.415e-11  1.173e-11  5.469e+00 2.39e-07 ***
## ContinentAustralia/Oceania  8.585e-11  2.404e-11  3.572e+00 0.000506 ***
## ContinentEurope            6.120e-11  1.201e-11  5.095e+00 1.26e-06 ***
## ContinentNorth America     -3.319e-11  1.266e-11 -2.622e+00 0.009841 **
## ContinentSouth America     -6.751e-11  1.688e-11 -3.999e+00 0.000109 ***
## Population                 -1.007e-18  5.199e-20 -1.938e+01  < 2e-16 ***
## TotalCases                  1.000e+00  1.017e-15  9.832e+14  < 2e-16 ***
## TotalRecovered             -1.000e+00  1.058e-15 -9.449e+14  < 2e-16 ***
## ActiveCases                -1.000e+00  1.065e-15 -9.387e+14  < 2e-16 ***
## Tot.Cases.1M.pop            1.336e-15  9.714e-16  1.375e+00 0.171521
## Deaths.1M.pop              -5.628e-14  3.251e-14 -1.731e+00 0.085869 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.551e-11 on 124 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 6.603e+29 on 11 and 124 DF,  p-value: < 2.2e-16
```

Overall Significance and Goodness of Fit

Key Observations:

1. Residuals are extremely small, suggesting near-perfect prediction.

2. R-squared = 1(both Multiple and Adjusted), meaning the model explains 100% of the variance in TotalDeaths

3. Extremely high F-statistic (6.603e+29) with p-value (2.2e-16) indication model is statistically significant.

Suspicious Coefficients:

TotalCases, TotalRecovered, and ActiveCases have exact coefficients (1.0, -1.0, -1.0) with near-zero standard errors (1.017e-15,1.058e-15,1.065e-15) and huge t-values (9.449e+14, 9.449e+14, 9.387e+14).

This suggest perfect multicollinearity or a mathematical relationship between these predictors and TotalDeaths

Goodness of fit test

Since the model fits data too perfectly, traditional goodness-of-fit tests are meaningless

vif values

```
vif(step_model1)

##                      GVIF Df GVIF^(1/(2*Df))
## Continent        1.705422  5        1.054832
## Population         2.664970  1        1.632474
## TotalCases      6584.867535  1       81.147197
## TotalRecovered   3494.402380  1       59.113470
## ActiveCases       553.363435  1       23.523678
## Tot.Cases.1M.pop    1.745077  1        1.321014
## Deaths.1M.pop       1.963243  1        1.401158
```

TotalDeaths may be directly computed from TotalCases, TotalRecovered, and ActiveCases

If true, the model is not meaningful for statistical inference—it's just an algebraic identity.

This recomend simplyfy the model:

so we can keep only TotalCases or TotalRecovered, So lets drop TotalCases

Lets remove TotalRecovered and fit the model

```
names(train1)

##  [1] "Continent"        "Population"       "TotalCases"       "TotalDeaths"
##  [5] "TotalRecovered"   "ActiveCases"      "Serious.Critical" "Tot.Cases.1M.pop"
##  [9] "Deaths.1M.pop"    "TotalTests"       "Tests.1M.pop"     "WHO.Region"

model1  step <- lm(TotalDeaths ~ ., data = train1[-c(5)])
step_model1 <- stepAIC(model1_step, direction = "both", trace = TRUE)

## Start:  AIC=2254.22
## TotalDeaths ~ Continent + Population + TotalCases + ActiveCases +
##     Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests +
##     Tests.1M.pop + WHO.Region
##
##                    Df Sum of Sq        RSS    AIC
## - WHO.Region        6  24621875 1625228201 2244.3
## - Continent         5  18973480 1619579806 2245.8
## - TotalTests        1    401043 1601007368 2252.2
## - ActiveCases       1    720037 1601326362 2252.3
## - Tests.1M.pop      1   5911480 1606517805 2252.7
## <none>                          1600606326 2254.2
## - Tot.Cases.1M.pop  1  97434564 1698040890 2260.2
## - TotalCases        1 195398906 1796005232 2267.9
## - Deaths.1M.pop     1 195969869 1796576195 2267.9
## - Serious.Critical  1 216455860 1817062185 2269.5
## - Population        1 235124343 1835730669 2270.9
##
## Step:  AIC=2244.29
## TotalDeaths ~ Continent + Population + TotalCases + ActiveCases +
##     Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests +
##     Tests.1M.pop
##
##                    Df Sum of Sq        RSS    AIC
## - Continent         5  11986697 1637214897 2235.3
## - ActiveCases       1    310255 1625538455 2242.3
## - TotalTests        1    508597 1625736797 2242.3
## - Tests.1M.pop      1   7179158 1632407359 2242.9
## <none>                          1625228201 2244.3
## - Tot.Cases.1M.pop  1 107528833 1732757033 2251.0
## + WHO.Region        6  24621875 1600606326 2254.2
## - TotalCases        1 194295684 1819523885 2257.7
## - Deaths.1M.pop     1 199625916 1824854117 2258.1
## - Serious.Critical  1 212517525 1837745725 2259.0
## - Population        1 253532463 1878760664 2262.0
##
## Step:  AIC=2235.29
## TotalDeaths ~ Population + TotalCases + ActiveCases + Serious.Critical +
##     Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests + Tests.1M.pop
##
##                    Df Sum of Sq        RSS    AIC
## - ActiveCases       1    196293 1637411191 2233.3
## - TotalTests        1    291022 1637505920 2233.3
## - Tests.1M.pop      1   3990538 1641205435 2233.6
## <none>                          1637214897 2235.3
## - Tot.Cases.1M.pop  1 128294975 1765509873 2243.6
## + Continent         5  11986697 1625228201 2244.3
## + WHO.Region        6  17635092 1619579806 2245.8
## - TotalCases        1 198959028 1836173926 2248.9
## - Serious.Critical  1 221241661 1858456559 2250.5
## - Population        1 267579908 1904794805 2253.9
## - Deaths.1M.pop     1 279032722 1916247619 2254.7
##
## Step:  AIC=2233.31
## TotalDeaths ~ Population + TotalCases + Serious.Critical + Tot.Cases.1M.pop +
##     Deaths.1M.pop + TotalTests + Tests.1M.pop
##
##                    Df  Sum of Sq        RSS    AIC
## - TotalTests        1     491118 1637902309 2231.3
## - Tests.1M.pop      1    4155568 1641566758 2231.7
## <none>                           1637411191 2233.3
## + ActiveCases       1     196293 1637214897 2235.3
## - Tot.Cases.1M.pop  1  132310154 1769721345 2241.9
## + Continent         5   11872736 1625538455 2242.3
## + WHO.Region        6   17128467 1620282724 2243.9
## - Serious.Critical  1  243238617 1880649808 2250.1
## - Deaths.1M.pop     1  281868250 1919279441 2252.9
## - Population        1  334913901 1972325092 2256.6
## - TotalCases        1 1098412775 2735823966 2301.1
##
## Step:  AIC=2231.35
## TotalDeaths ~ Population + TotalCases + Serious.Critical + Tot.Cases.1M.pop +
##     Deaths.1M.pop + Tests.1M.pop
##
##                    Df  Sum of Sq        RSS    AIC
## - Tests.1M.pop      1    3692098 1641594407 2229.7
## <none>                           1637902309 2231.3
## + TotalTests        1     491118 1637411191 2233.3
## + ActiveCases       1     396389 1637505920 2233.3
## - Tot.Cases.1M.pop  1  133963795 1771866104 2240.0
## + Continent         5   11548285 1626354024 2240.4
## + WHO.Region        6   17523784 1620378525 2241.9
## - Serious.Critical  1  247446512 1885348821 2248.5
## - Deaths.1M.pop     1  283226642 1921128951 2251.0
## - Population        1  583416039 2221318349 2270.8
## - TotalCases        1 1666491511 3304393820 2324.8
## 
## Step:  AIC=2229.65
## TotalDeaths ~ Population + TotalCases + Serious.Critical + Tot.Cases.1M.pop + 
##     Deaths.1M.pop
## 
##                    Df  Sum of Sq        RSS    AIC
## <none>                            1641594407 2229.7
## + Tests.1M.pop      1    3692098 1637902309 2231.3
## + ActiveCases       1     388838 1641205569 2231.6
## + TotalTests        1      27648 1641566758 2231.7
## + Continent         5    8485002 1633109405 2238.9
## - Tot.Cases.1M.pop  1  151640386 1793234793 2239.7
## + WHO.Region        6   16722085 1624872322 2240.3
## - Serious.Critical  1  246045312 1887639719 2246.7
## - Deaths.1M.pop     1  283687737 1925282144 2249.3
## - Population        1  581089964 2222684371 2268.9
## - TotalCases        1 1679568737 3321163144 2323.5
```

```
summary(step_model1)
```

```
## 
## Call:
## lm(formula = TotalDeaths ~ Population + TotalCases + Serious.Critical + 
##     Tot.Cases.1M.pop + Deaths.1M.pop, data = train1[-c(5)])
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9188.4 -1118.0     1.9   494.9 26415.7 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      -2.283e+02  3.996e+02  -0.571 0.568696    
## Population       -2.537e-05  3.740e-06  -6.784 3.73e-10 ***
## TotalCases        2.691e-02  2.333e-03  11.533  < 2e-16 ***
## Serious.Critical  3.005e+00  6.807e-01   4.414 2.11e-05 ***
## Tot.Cases.1M.pop -2.396e-01  6.915e-02  -3.465 0.000718 ***
## Deaths.1M.pop     1.032e+01  2.176e+00   4.740 5.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3554 on 130 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8867 
## F-statistic: 212.2 on 5 and 130 DF,  p-value: < 2.2e-16
```

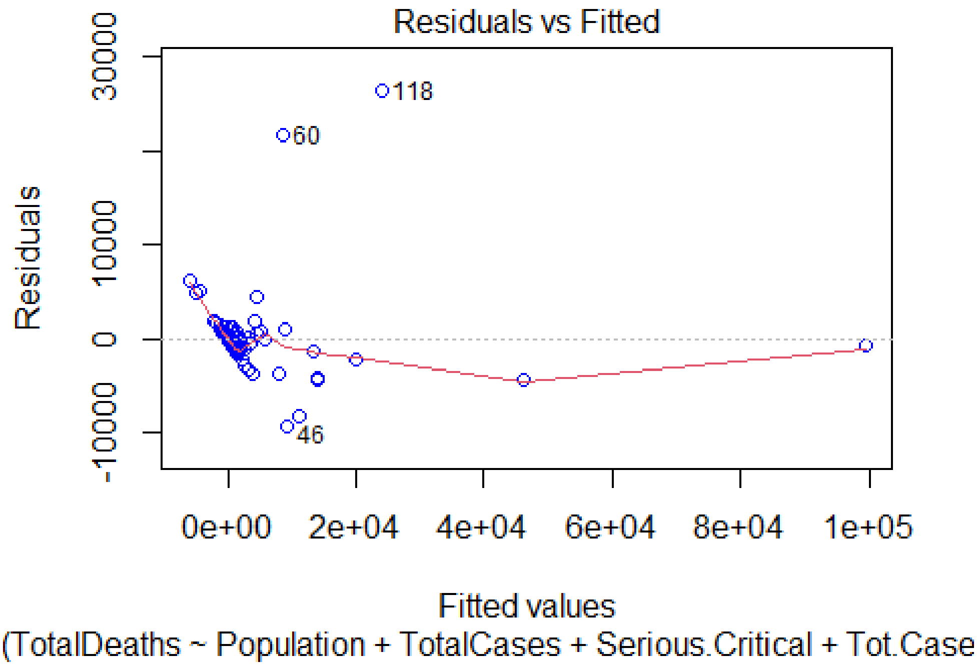
Step_model1 all predictors are significance at 5% significance level

Residual Analysis

1. Check Homoscedasticity (Constant Variance)

Residual vs. Fitted Plot:

```
plot(step_model1, which = 1, col ="blue")
```



# Breusch-Pagan Test

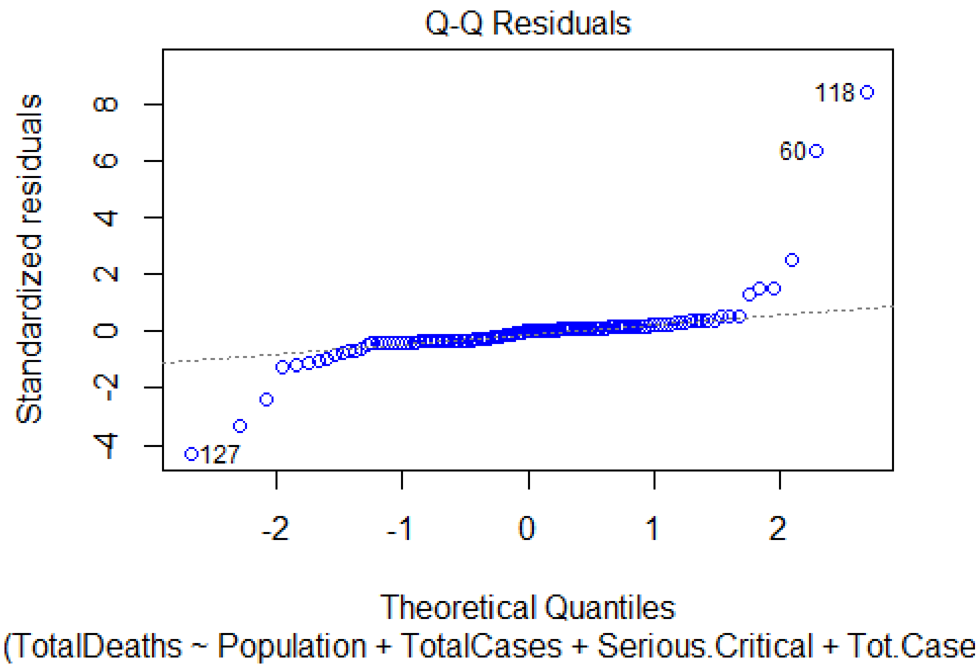
```
bptest(step_model1)

##
##  studentized Breusch-Pagan test
##
## data:  step_model1
## BP = 26.412, df = 5, p-value = 7.423e-05
```

since $p < 0.05$ assumption homoscedasticity is vialated

2. Check Normality of Residuals

Normal Q-Q plot

```
plot(step_model1, which = 2, col ="blue")
```



# Shapiro-Wilk Test

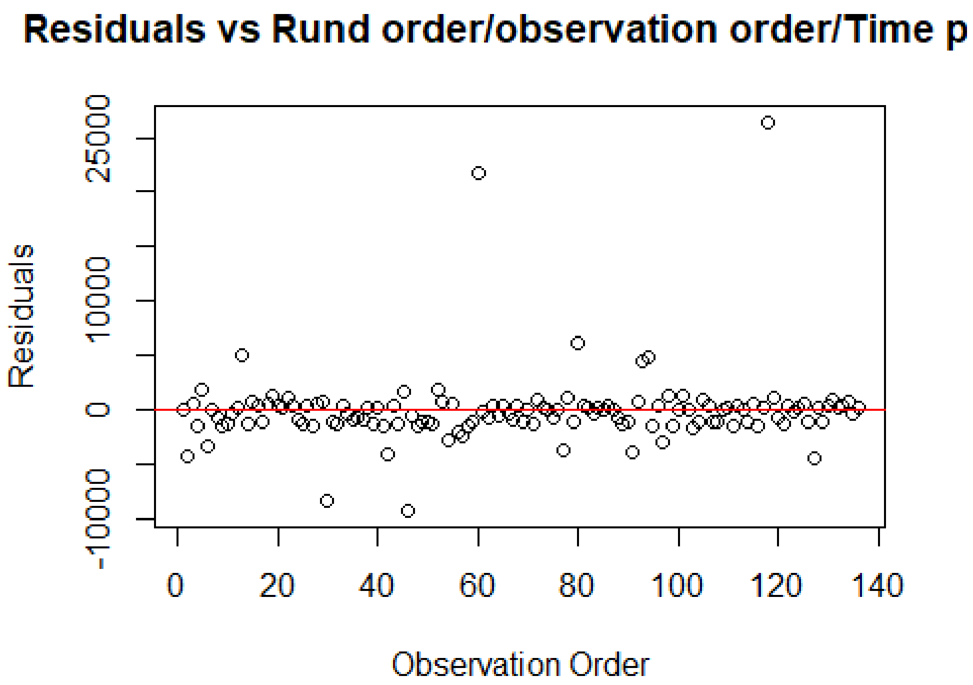
```
shapiro.test(step_model1$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  step_model1$residuals
## W = 0.52154, p-value < 2.2e-16
```

Since p-value < 0.05 normality assumption is vialated.

3. Independence of Residuals

Residual vs Rund order

```
plot(residuals(step_model1),
xlab = "Observation Order",
ylab = "Residuals",
main = "Residuals vs Rund order/observation order/Time plot")
abline(h = 0, col = "red")
```



# Durbin-Watson Test

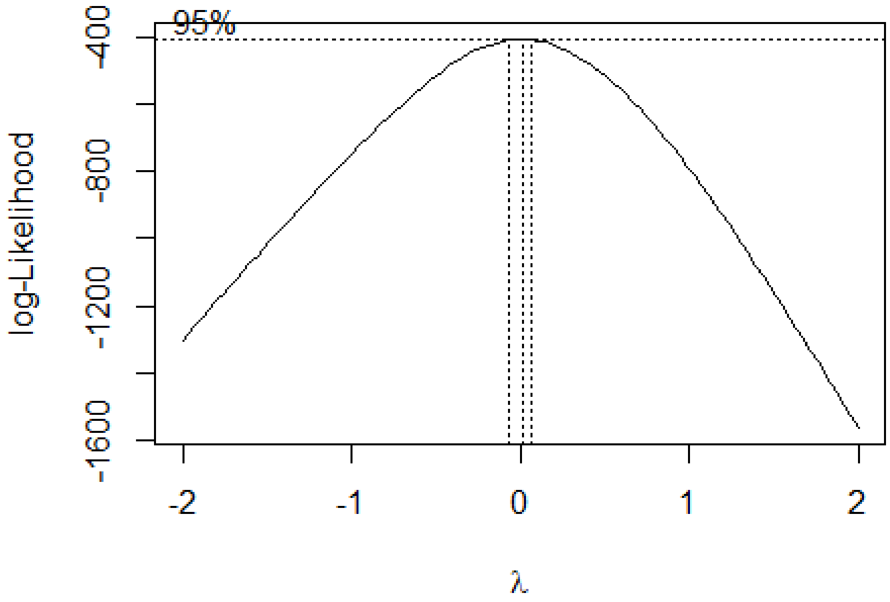
```
dwtest(step_model1)

##
##  Durbin-Watson test
##
## data:  step_model1
## DW = 2.0075, p-value = 0.5235
## alternative hypothesis: true autocorrelation is greater than 0
```

since p > 0.05 suggests no autocorrelation

Now we can see that two assumptions (Normality & Homoscedasticity (Constant Variance) of Residuals) are vialated.

lets try use box-cox tranformation

```
b <- boxcox(lm(TotalDeaths ~ ., data = train1[-c(5)]))
```



```
lambda <- b$x[which.max(b$y)]

lambda

## [1] 0.02020202
```

lambda = 0.02020202, this mean log transformation is nearly optimal

```
model1_step <- lm(log(TotalDeaths) ~ ., data = train1[-c(5)])
step_model1 <- stepAIC(model1_step, direction = "both", trace = TRUE)

## Start:  AIC=184.08
## log(TotalDeaths) ~ Continent + Population + TotalCases + ActiveCases +
##     Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests +
##     Tests.1M.pop + WHO.Region
##
##                    Df Sum of Sq    RSS    AIC
## - Continent         5    14.727 407.04 179.09
## - ActiveCases       1     0.213 392.53 182.15
## - Population        1     0.770 393.08 182.35
## - Tot.Cases.1M.pop  1     0.820 393.13 182.36
## - TotalCases        1     1.912 394.23 182.74
## <none>                          392.31 184.08
## - Serious.Critical  1     9.579 401.89 185.36
## - TotalTests        1    18.418 410.73 188.32
## - Tests.1M.pop      1    27.540 419.85 191.31
## - Deaths.1M.pop     1    33.773 426.09 193.31
## - WHO.Region        6   110.376 502.69 205.79
##
## Step:  AIC=179.09
## log(TotalDeaths) ~ Population + TotalCases + ActiveCases + Serious.Critical +
##     Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests + Tests.1M.pop +
##     WHO.Region
## 
##                    Df Sum of Sq    RSS    AIC
## - ActiveCases       1     0.479 407.52 177.25
## - Population        1     1.508 408.55 177.59
## - Tot.Cases.1M.pop  1     1.605 408.65 177.63
## - TotalCases        1     3.146 410.19 178.14
## <none>                          407.04 179.09
## - Serious.Critical  1    12.712 419.75 181.27
## - TotalTests        1    16.669 423.71 182.55
## + Continent         5    14.727 392.31 184.08
## - Tests.1M.pop      1    27.022 434.06 185.83
## - Deaths.1M.pop     1    36.108 443.15 188.65
## - WHO.Region        6   139.085 546.13 207.07
## 
## Step:  AIC=177.25
## log(TotalDeaths) ~ Population + TotalCases + Serious.Critical + 
##     Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests + Tests.1M.pop + 
##     WHO.Region
## 
##                    Df Sum of Sq    RSS    AIC
## - Population        1     1.080 408.60 175.61
## - Tot.Cases.1M.pop  1     2.020 409.54 175.92
## <none>                          407.52 177.25
## - TotalCases        1     7.366 414.89 177.69
## + ActiveCases       1     0.479 407.04 179.09
## - Serious.Critical  1    12.279 419.80 179.29
## - TotalTests        1    19.430 426.95 181.59
## + Continent         5    14.993 392.53 182.15
## - Tests.1M.pop      1    27.700 435.22 184.19
## - Deaths.1M.pop     1    35.654 443.17 186.66
## - WHO.Region        6   141.386 548.91 205.76
## 
## Step:  AIC=175.61
## log(TotalDeaths) ~ TotalCases + Serious.Critical + Tot.Cases.1M.pop + 
##     Deaths.1M.pop + TotalTests + Tests.1M.pop + WHO.Region
## 
##                    Df Sum of Sq    RSS    AIC
## - Tot.Cases.1M.pop  1     1.991 410.59 174.27
## <none>                          408.60 175.61
## - TotalCases        1     6.292 414.89 175.69
## + Population        1     1.080 407.52 177.25
## - Serious.Critical  1    11.386 419.99 177.35
## + ActiveCases       1     0.051 408.55 177.59
## + Continent         5    15.499 393.10 180.35
## - Tests.1M.pop      1    30.244 438.84 183.32
## - Deaths.1M.pop     1    34.908 443.51 184.76
## - TotalTests        1    46.478 455.08 188.26
## - WHO.Region        6   140.713 549.31 203.86
## 
## Step:  AIC=174.27
## log(TotalDeaths) ~ TotalCases + Serious.Critical + Deaths.1M.pop + 
##     TotalTests + Tests.1M.pop + WHO.Region
## 
##                    Df Sum of Sq    RSS    AIC
## <none>                              410.59 174.27
## - TotalCases        1     8.436 419.03 175.04
## + Tot.Cases.1M.pop  1     1.991 408.60 175.61
## + Population        1     1.051 409.54 175.92
## + ActiveCases       1     0.222 410.37 176.20
## - Serious.Critical  1    13.452 424.04 176.66
## + Continent         5    16.472 394.12 178.70
## - Tests.1M.pop      1    28.253 438.84 181.32
## - TotalTests        1    45.083 455.67 186.44
## - Deaths.1M.pop     1    57.331 467.92 190.05
## - WHO.Region        6   143.623 554.21 203.07
```

```
summary(step_model1)
```

```
## 
## Call:
## lm(formula = log(TotalDeaths) ~ TotalCases + Serious.Critical + 
##     Deaths.1M.pop + TotalTests + Tests.1M.pop + WHO.Region, data = train1[-c(5)])
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.5123 -1.0450  0.1229  1.1657  5.2208 
## 
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                     1.825e+00  5.567e-01   3.278 0.001357 ** 
## TotalCases                      2.197e-06  1.377e-06   1.596 0.112997    
## Serious.Critical               -6.605e-04  3.277e-04  -2.016 0.046002 *  
## Deaths.1M.pop                   4.348e-03  1.045e-03   4.161 5.87e-05 ***
## TotalTests                      4.758e-07  1.290e-07   3.690 0.000334 ***
## Tests.1M.pop                   -3.390e-06  1.161e-06  -2.921 0.004146 ** 
## WHO.RegionAfrica                2.133e+00  6.365e-01   3.351 0.001068 ** 
## WHO.RegionAmericas              3.226e+00  6.528e-01   4.943 2.44e-06 ***
## WHO.RegionEasternMediterranean  4.010e+00  7.136e-01   5.619 1.20e-07 ***
## WHO.RegionEurope                2.997e+00  6.231e-01   4.810 4.29e-06 ***
## WHO.RegionSouth-EastAsia        1.660e+00  9.118e-01   1.821 0.071023 .  
## WHO.RegionWesternPacific        1.707e+00  7.974e-01   2.140 0.034283 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.82 on 124 degrees of freedom
## Multiple R-squared:  0.5475, Adjusted R-squared:  0.5074 
## F-statistic: 13.64 on 11 and 124 DF,  p-value: < 2.2e-16
```

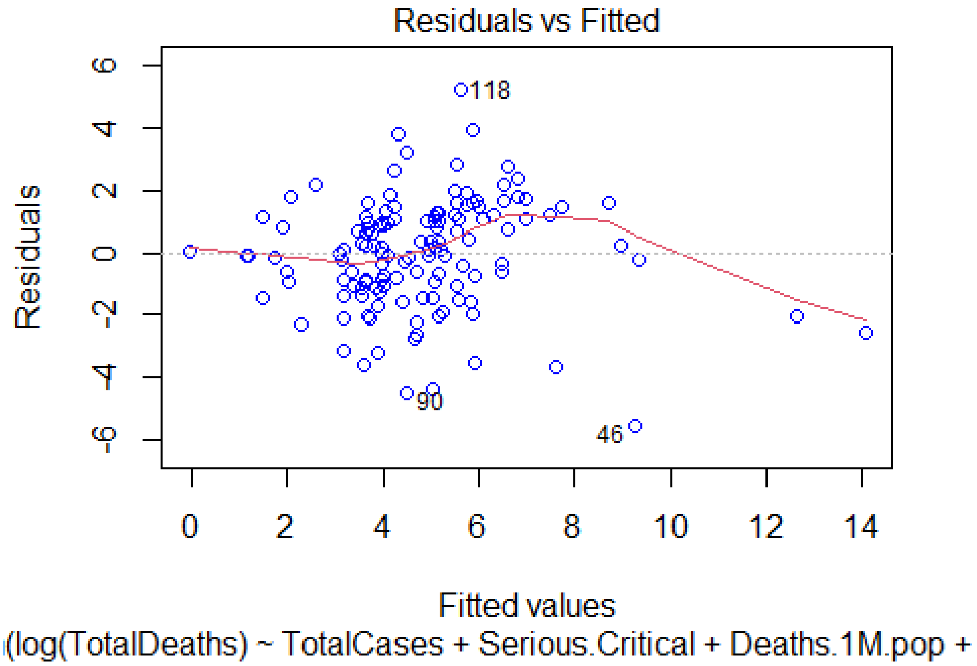
log tranfomation Step_model2 is at 5% significance level.

Residual Analysis

1. Check Homoscedasticity (Constant Variance)

Residual vs. Fitted Plot:

```
plot(step_model1, which = 1, col ="blue")
```



Breusch-Pagan Test

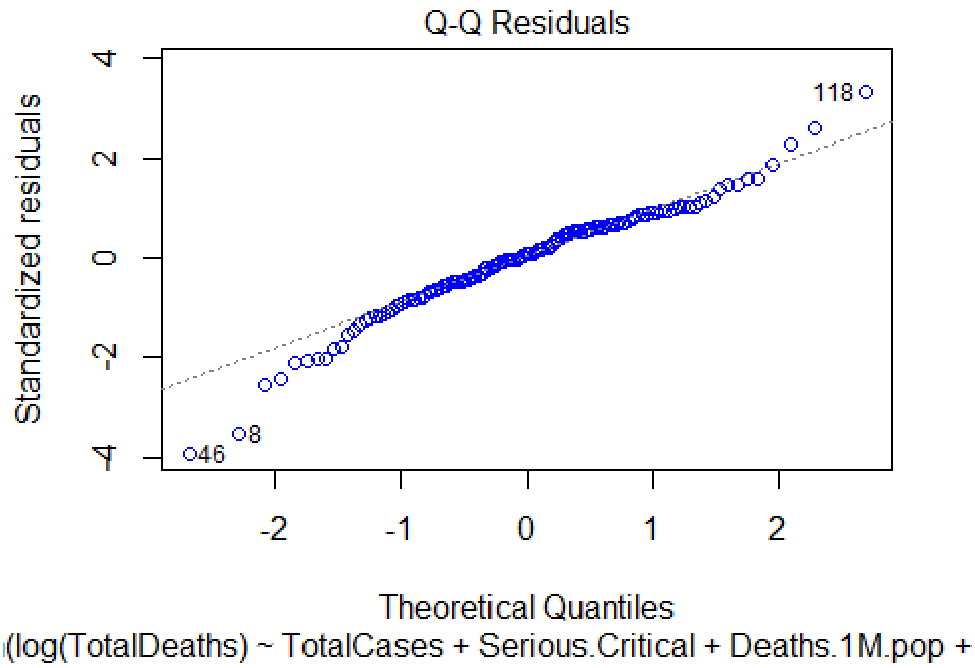
```
bptest(step_model1)

##
##  studentized Breusch-Pagan test
##
## data:  step_model1
## BP = 45.972, df = 11, p-value = 3.27e-06
```

since p < 0.05 assumption homoscedasticity is vialated

2. Check Normality of Residuals

Normal Q-Q plot

```
plot(step_model1,which = 2, col ="blue")
```



# Shapiro-Wilk Test

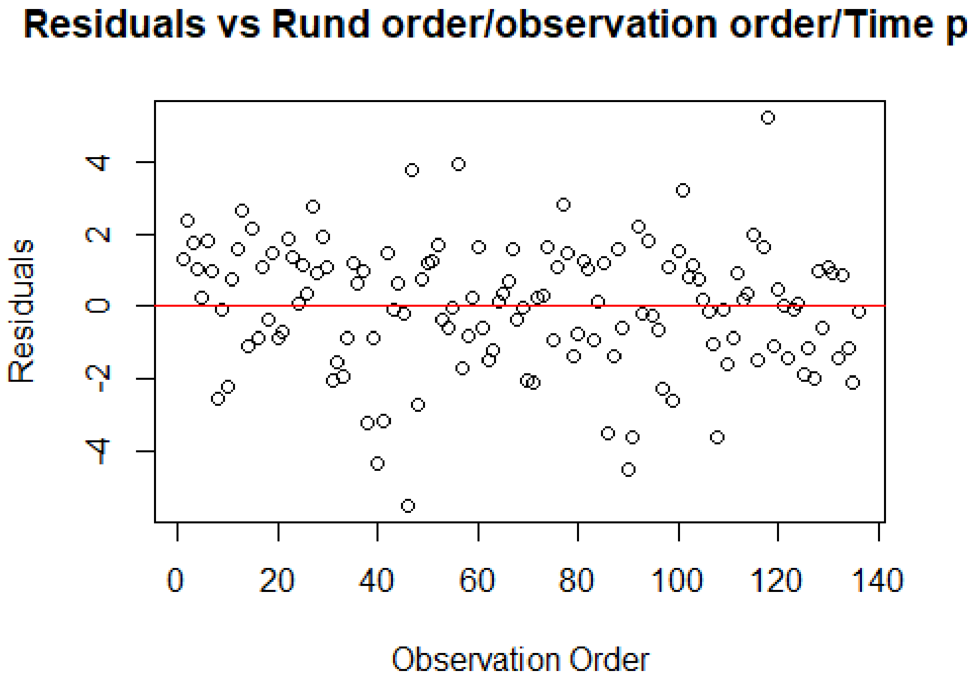
```
shapiro.test(step_model1$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  step_model1$residuals
## W = 0.98484, p-value = 0.1377
```

Since p-value > 0.05 normality assumption is not vialated.

3. Independence of Residuals

Residual vs Rund order

```
plot(residuals(step_model1),
xlab = "Observation Order",
ylab = "Residuals",
main = "Residuals vs Rund order/observation order/Time plot")
abline(h = 0, col = "red")
```



# Durbin-Watson Test

```
dwtest(step_model1)

##
##  Durbin-Watson test
##
## data:  step_model1
## DW = 1.895, p-value = 0.2714
## alternative hypothesis: true autocorrelation is greater than 0
```

since p > 0.05 suggests no autocorrelation between residuals.

Independence of residuals is not vialated.

still we can see that two assumptions (Normality & Homoscedasticity (Constant Variance) of Residuals) are vialated.

let’s use Weighted Least Squares (WLS) try remove Homoscedasticity vialation

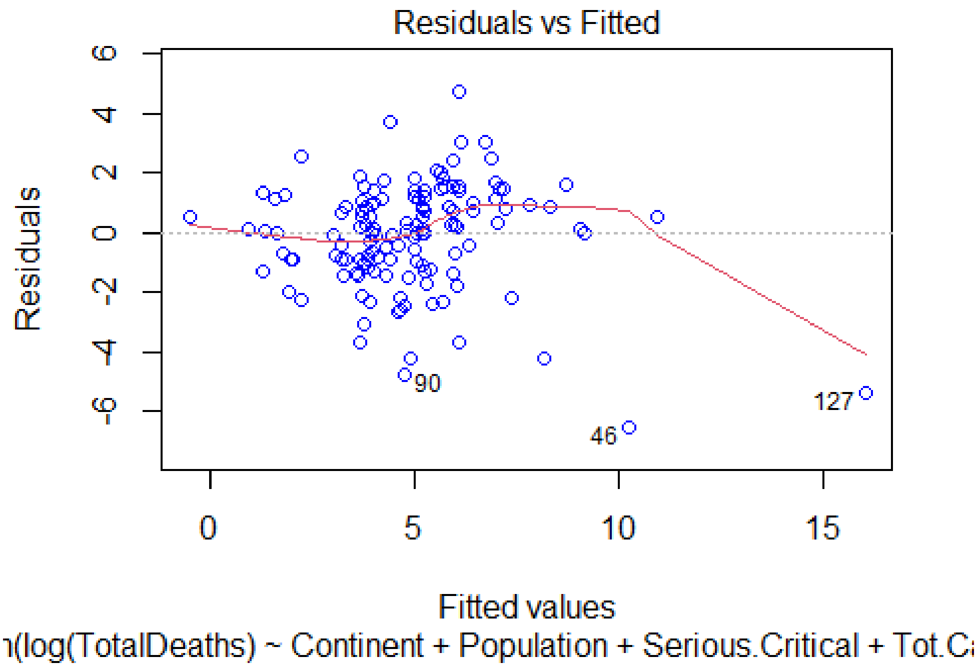
```
model1_step <- lm(log(TotalDeaths) ~ ., data = train1[-c(5)])
weights <- 1 / residuals(model1  step)^2
wls <- lm(log(TotalDeaths) ~ ., data = train1[-c(5)],weights = weights)
step_model1 <- stepAIC(wls, direction = "both", trace = TRUE)

## Start:  AIC=25.99
## log(TotalDeaths) ~ Continent + Population + TotalCases + ActiveCases +
##     Serious.Critical + Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests +
##     Tests.1M.pop + WHO.Region
```

```
##
##                    Df Sum of Sq     RSS    AIC
## - ActiveCases       1      0.69  123.38  24.75
## - TotalCases        1      1.40  124.09  25.54
## <none>                           122.69  25.99
## - Tot.Cases.1M.pop  1      2.73  125.42  26.98
## - Population        1      5.13  127.81  29.56
## - Serious.Critical  1      6.70  129.39  31.22
## - Continent         5     22.36  145.05  38.76
## - TotalTests        1     52.98  175.66  72.80
## - Deaths.1M.pop     1     71.61  194.30  86.51
## - Tests.1M.pop      1    104.04  226.73 107.51
## - WHO.Region        6   1248.32 1371.01 342.25
##
## Step:  AIC=24.75
## log(TotalDeaths) ~ Continent + Population + TotalCases + Serious.Critical +
##     Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests + Tests.1M.pop +
##     WHO.Region
##
##                    Df Sum of Sq     RSS    AIC
## - TotalCases        1      0.83  124.21  23.67
## <none>                           123.38  24.75
## + ActiveCases       1      0.69  122.69  25.99
## - Tot.Cases.1M.pop  1      3.26  126.64  26.30
## - Population        1      4.50  127.87  27.62
## - Serious.Critical  1      6.72  130.10  29.97
## - Continent         5     35.95  159.33  49.53
## - Deaths.1M.pop     1     72.48  195.86  85.60
## - Tests.1M.pop      1    112.53  235.91 110.91
## - TotalTests        1    214.28  337.66 159.67
## - WHO.Region        6   1292.55 1415.93 344.63
##
## Step:  AIC=23.67
## log(TotalDeaths) ~ Continent + Population + Serious.Critical +
##     Tot.Cases.1M.pop + Deaths.1M.pop + TotalTests + Tests.1M.pop +
##     WHO.Region
##
##                    Df Sum of Sq     RSS    AIC
## <none>                           124.21  23.67
## + TotalCases        1      0.83  123.38  24.75
## + ActiveCases       1      0.12  124.09  25.54
## - Tot.Cases.1M.pop  1      4.48  128.69  26.49
## - Population        1      5.79  130.01  27.87
## - Serious.Critical  1      7.09  131.30  29.21
## - Continent         5     64.61  188.82  70.63
## - Deaths.1M.pop     1     76.39  200.60  86.86
## - Tests.1M.pop      1    119.42  243.63 113.29
## - TotalTests        1    389.82  514.03 214.83
## - WHO.Region        6   1318.74 1442.95 345.20
```

1. Check Homoscedasticity (Constant Variance)

Residual vs. Fitted Plot:

```
plot(step_model1, which = 1, col ="blue")
```



```
bptest(step_model1)

##
##  studentized Breusch-Pagan test
##
## data:  step_model1
## BP = 6513.2, df = 17, p-value < 2.2e-16
```

p-value < 0.05, still it is same, Homoscedasticity is vialated

since p > 0.05 suggests no autocorrelation between residuals.

Independence of residuals is not vialated.

still we can see that two assumptions (Normality & Homoscedasticity (Constant Variance) of Residuals) are vialated.

Before we use mean imputation for Serious.Critical, next we use median imputation but the result is still same Normality & Homoscedasticity (Constant Variance) of Residuals are vialated.

After all since we are using real life data, sometimes normality can be vialated. So we can neglect that, but the case we face is we cann't neglect Homoscedasticity (Constant Variance) of Residuals) assumption vialation.

So our final conclusion is, the model is not suitable for predict deathrate.