MIDTERM PROJECT

1. Exploratory Data Analysis
   a. What variables look most promising for predicting cancer mortality from exploratory data analysis? Why?

Ans-The variables that look promising for predicting cancer are incidenceRate, medIncome, PctHS18_24, PctBachDeg18_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace. This can be seen by comparing the p-values that are calculated by fitting a linear regression model using the Cancer Data dataset. This can also be seen for the co-relation plot and co-relation matrix.

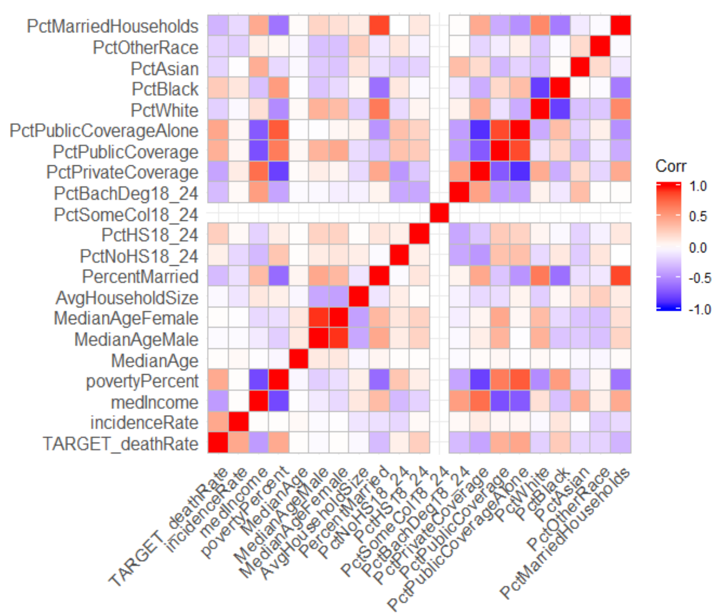


Figure 1-Co-relation graph

   b. Are there any outliers? Can they be detected and addressed? How does addressing outliers affect model performance?

Ans- MedIncome has outliers, this can be seen by plotting a box plot. Outliers can be treated by replacing them with the column mean or mode. They can also be treated using capping technique. In percentile capping, the value at 1st percentile, and values that are greater than the value at 99th percentile are replaced by the value at 99th percentile.

Code-

```
#finding outliers

OutVals = boxplot(train, plot=FALSE)$out

OutVals1 = boxplot(medIncome, plot=FALSE)$out

plot(OutVals1)

plot(OutVals)

boxplot(train)

library(outliers)

outlier(medIncome)

#treating outlies- by using capping

x <- train$medIncome

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x

boxplot(train$medIncome)
```

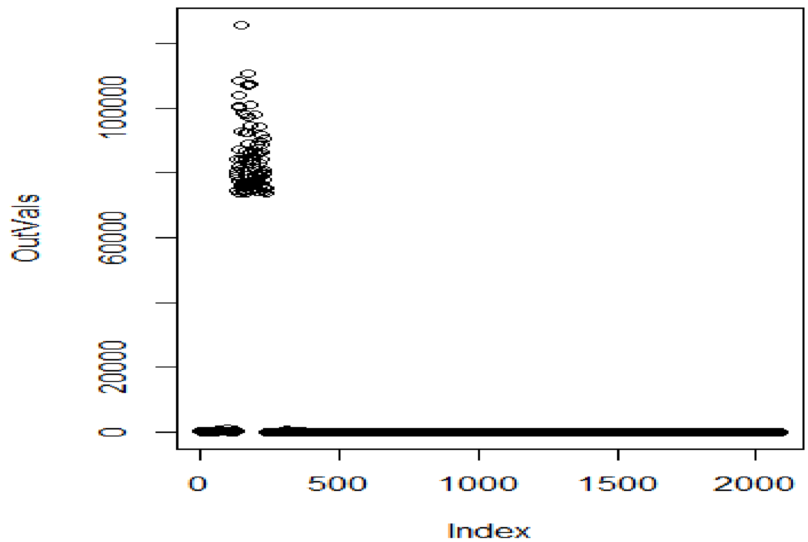


Figure 2- Outliers Plot

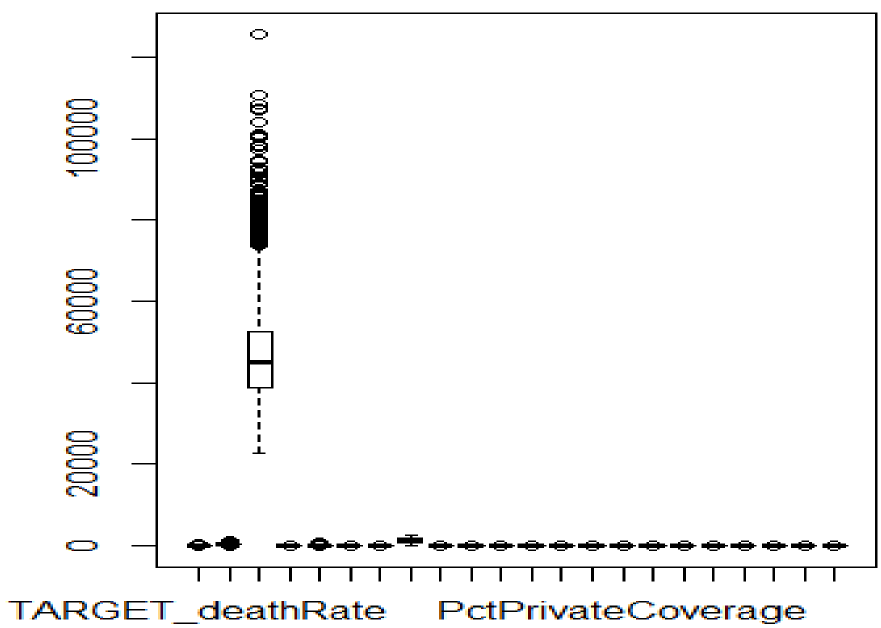


Figure 3-Box plot

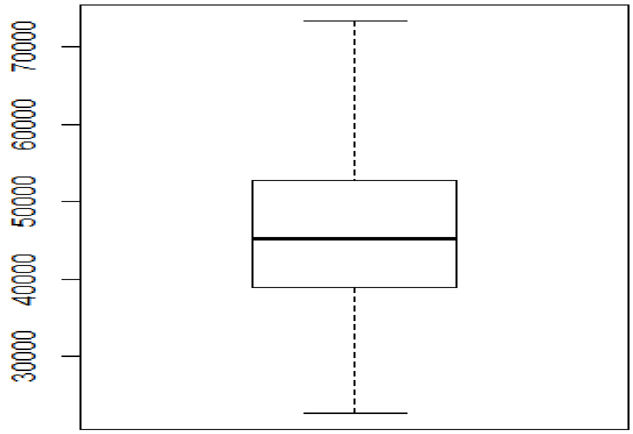


Figure 4- Box pot of medIncome after capping

c. Are there any missing values? Research and explore techniques to handle missing values. Note that the approach to handle missing data might be different for different variables. Document model performance improvement obtained by missing data handling.

   Ans – By observing the test and the train dataset, one can observe that there are a lot of missing values in PctSomeCol18_24. This can also be observed by plotting a Missing Map of the datasets. There are a total of 1938 missing values in PctSomeCol18_24. Missing values can be treated by replacing them with mean, median or mode of that column or ignoring the column if there are a lot of missing values. In this case, since there are a lot of missing values PctSomeCol18_24 can be neglected from model fitting. Documentation of model performance improvement obtained by missing data handling is done in question 2.

Code-

#missing values

library(Amelia)

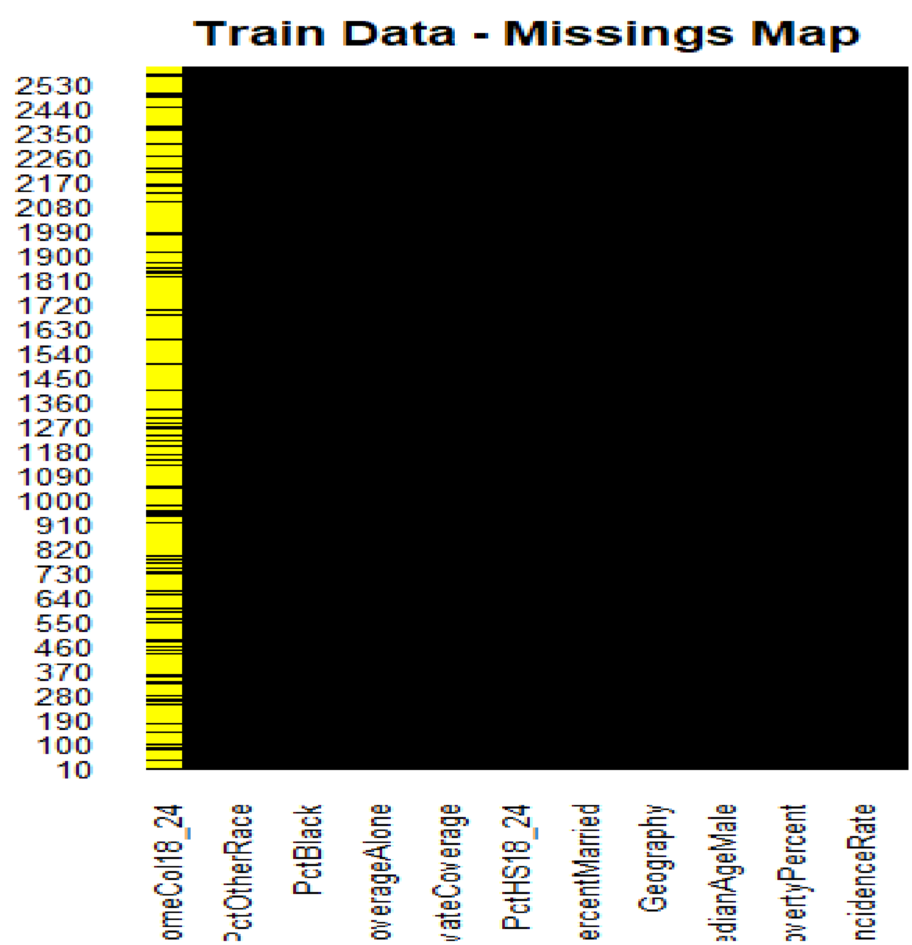
sum(is.na(train$PctSomeCol18_24))

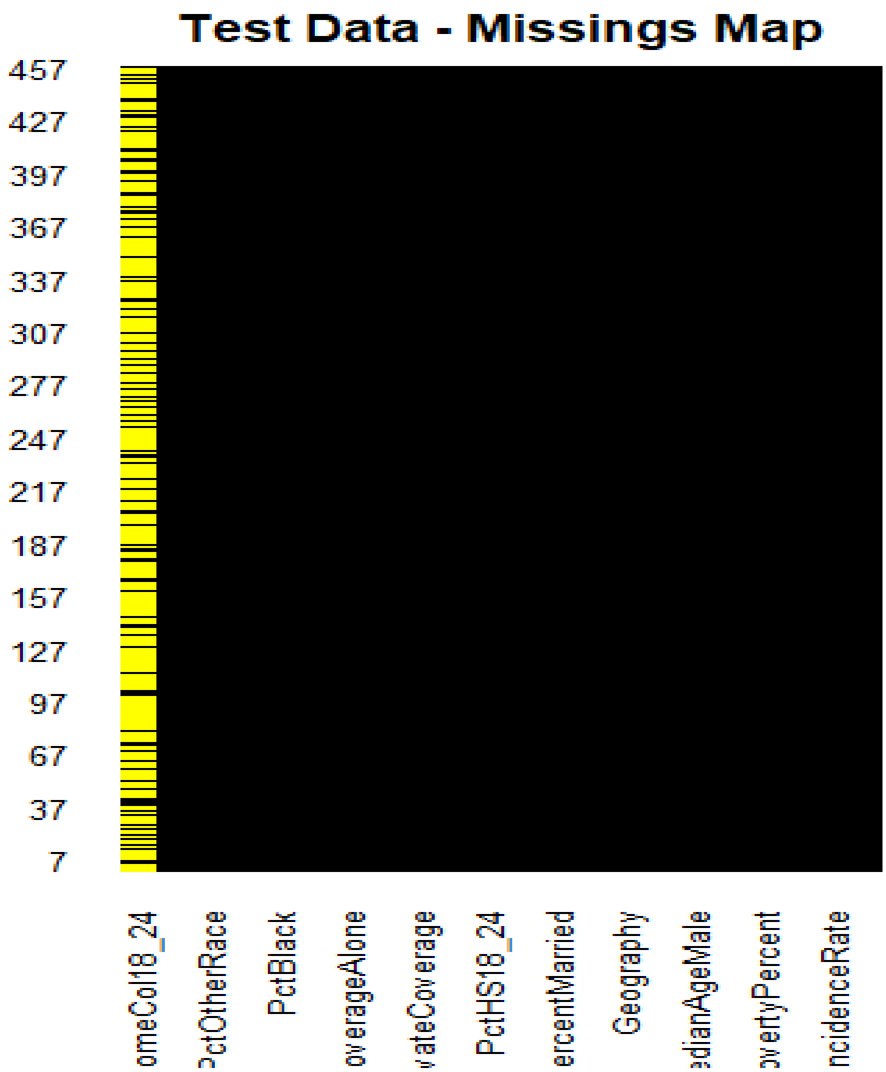
missmap(train, main="Train Data - Missings Map",

col=c("yellow", "black"), legend=FALSE)

Output-

```
> sum(is.na(train$PctSomeCol18_24))
[1] 1938
```





d. Is there any collinearity between variables? Can it be detected? Document how addressing collinearity affects model performance?
e. Ans- There is collinearity between the variables. It can be detected by observing the VIF values of the variables after fitting them in linear regression using the olsrr library. Documentation on how addressing collinearity affects the model performance is discussed in question 2. Any variable with VIF value above 4 should be neglected because of collinearity.

Code-

#finding collinearity

install.packages('olsrr')

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24)]= median(train$PctSomeCol18_24, na.rm= TRUE)

test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]= median(test$PctSomeCol18_24, na.rm= TRUE)

library(olsrr)

ols_vif_tol(LR3)

Output-

```
> ols_vif_tol(LR3)
                 Variables  Tolerance       VIF
1            incidenceRate 0.82900494  1.206265
2                medIncome 0.17124026  5.839748
3           povertyPercent 0.13454213  7.432616
4                MedianAge 0.98189629  1.018437
5            MedianAgeMale 0.11254315  8.885481
6          MedianAgeFemale 0.09976550 10.023505
7         AvgHouseholdSize 0.68497668  1.459904
8           PercentMarried 0.13710069  7.293909
9             PctNoHS18_24 0.63776768  1.567969
10              PctHS18_24 0.72733607  1.374880
11         PctBachDeg18_24 0.54450499  1.836530
12      PctPrivateCoverage 0.10992638  9.096997
13       PctPublicCoverage 0.05668669 17.640825
14 PctPublicCoverageAlone 0.05545882 18.031395
15                PctWhite 0.14668757  6.817210
16                PctBlack 0.19471208  5.135788
17                PctAsian 0.57515070  1.738675
18            PctOtherRace 0.71566391  1.397304
19   PctMarriedHouseholds 0.15894287  6.291569
```

2. Linear Regression
   a. Develop a linear regression model.

Ans- Multiple linear regression models are developed after refining the data at each step.

<u>After treating the missing values-</u>

- Three models are built after treating the missing by replacing them with mean, median and by neglecting the column. The train MSE obtained by neglecting the column, replacing it with median and mean is 411.3217, 411.3189, 411.3217 respectively and the test MSEs obtained are 414.5908, 414.54, 414.5908. Neglecting the column is a better choice in this situation since there are a lot of missing values even though the test MSE and train MSE are better when replaced by median.

<u>After treating the outliers-</u>

- The fourth model is developed after replacing the outliers using percentile capping and neglecting the PctSomeCol18_24 column. The train and test MSE are 409.5991 and 416.1014 respectively. This model performs well on the training data set since the train MSE is lower than the other three models but it has a higher test MSE compared to the other models.

<u>After treating collinearity-</u>

- The fifth model is developed after removing the collinear variables and neglecting the PctSomeCol18_24 column. The train and test MSE are 459.1824 and 460.9086 respectively. This model doesn’t perform as good as the other models because it has a higher test and train MSE.

<u>After treating everything -</u>

- The last model is developed after treating the missing values, outliers and collinearity. The train and test MSE are 409.59 and 416.10 respectively.

Test MSE vs Train MSE plot-

The train and test MSEs of all the models is plotted to give a good summary of the models and also helps in choosing the optimum model.

Code-

```
#missing values

library(Amelia)

sum(is.na(train$PctSomeCol18_24))

missmap(train, main="Train Data - Missings Map",

       col=c("yellow", "black"), legend=FALSE)

missmap(test, main="Test Data - Missings Map",

       col=c("yellow", "black"), legend=FALSE)


#treating missing values

#method1 - neglecting the coloumn

LR3                                                                    =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

        , data =train)

summary(LR3)

LR3.pred= predict(LR3 ,newdata= test )
```

```
msetrain_n=mean((train$TARGET_deathRate-fitted(LR3))^2)

msetrain_n

msetest_n=mean(((test$TARGET_deathRate) - (LR3.pred))^2)

msetest_n


#Method2 - inputing median

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24) ]= median(train$PctSomeCol18_24, na.rm= TRUE)

test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]=
median(test$PctSomeCol18_24, na.rm= TRUE)

LR2 = lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctSomeCol18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

    , data =train)

summary(LR2)

LR2.pred= predict(LR2 ,newdata= test)

LR2.pred
```

```
msetrain_median=mean((train$TARGET_deathRate-fitted(LR2))^2)

msetrain_median

msetest_median=mean(((test$TARGET_deathRate) - (LR2.pred))^2)

msetest_median


#method3- Inputing the mean

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24) ]= mean(train$PctSomeCol18_24, na.rm= TRUE)

test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]=
mean(test$PctSomeCol18_24, na.rm= TRUE)

LR1 = lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

    , data =train)

summary(LR1)

LR1.pred= predict(LR1 ,newdata= test)

msetrain1=mean((train$TARGET_deathRate-fitted(LR1))^2)
```

```
msetrain1

msetest1=mean(((test$TARGET_deathRate) - (LR1.pred))^2)

msetest1

#finding outliers

OutVals = boxplot(train, plot=FALSE)$out

OutVals1 = boxplot(medIncome, plot=FALSE)$out

plot(OutVals1)

plot(OutVals)

boxplot(train)

library(outliers)

outlier(medIncome)


#treating outlies- by using capping

x <- train$medIncome

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x

boxplot(train$medIncome)
```

```
LR5 = lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds
        , data =train)

summary(LR5)

LR5.pred= predict(LR5 ,newdata= test)

msetrain2=mean((train$TARGET_deathRate-fitted(LR5))^2)

msetrain2 #optimum msetrain

msetest2=mean(((test$TARGET_deathRate) - (LR5.pred))^2)

msetest2   #optimum msetest


#finding collinearity


#install.packages('olsrr')

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24) ]= median(train$PctSomeCol18_24, na.rm= TRUE)
```

```
test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]=
median(test$PctSomeCol18_24, na.rm= TRUE)

library(olsrr)

ols_vif_tol(LR3)

#treating collinearity - neglecting the variables


LR6 =
lm(TARGET_deathRate~incidenceRate+medIncome+MedianAge+AvgHousehol
dSize+PctBlack+PctAsian+PctOtherRace, data =train)

summary(LR5)

LR6.pred= predict(LR6 ,newdata= test)

msetrain3=mean((train$TARGET_deathRate-fitted(LR6))^2)

msetrain3

msetest3=mean(((test$TARGET_deathRate) - (LR6.pred))^2)

msetest3




#optimummodel

train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

x <- train$TARGET_deathRate
```

```
qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$TARGET_deathRate = x

LR7 =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

      , data =train)

summary(LR7)

LR7.pred= predict(LR7 ,newdata= test)

msetrain4=mean((train$TARGET_deathRate-fitted(LR7))^2)

msetrain4 #optimum msetrain

msetest4=mean(((test$TARGET_deathRate) - (LR7.pred))^2)

msetest4   #optimum msetest

#trainmse vs testmse

trainMSE= c(459,411,371,367)

testMSE= c(460,414,416,409)

#1= collinearity,2= neglecting, 3= outliers, 4= optimum in x
```

x= c(1,2,3,4)

plot(x,trainMSE, ylab='trainMSE and testMSE')

lines(testMSE, col = 'red')

lines(trainMSE, col='blue')

Output-

```
> summary(LR3)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent +
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 +
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-86.338 -12.160  -0.137  11.656 127.254

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.057e+02  1.422e+01   7.435 1.42e-13
incidenceRate           2.177e-01  8.218e-03  26.494  < 2e-16
medIncome              -2.648e-04  7.983e-05  -3.317 0.000922
povertyPercent          3.093e-01  1.697e-01   1.823 0.068467
MedianAge               2.215e-03  9.630e-03   0.230 0.818095
MedianAgeMale          -2.048e-01  2.292e-01  -0.893 0.371682
MedianAgeFemale        -1.382e-01  2.392e-01  -0.578 0.563459
AvgHouseholdSize        6.104e-01  1.201e+00   0.508 0.611419
PercentMarried          1.748e-01  1.565e-01   1.117 0.264197
PctNoHS18_24           -4.513e-02  6.158e-02  -0.733 0.463691
PctHS18_24              4.582e-01  5.217e-02   8.782  < 2e-16
PctBachDeg18_24        -3.448e-01  1.182e-01  -2.918 0.003553
PctPrivateCoverage     -2.744e-01  1.135e-01  -2.417 0.015711
PctPublicCoverage       2.896e-02  2.136e-01   0.136 0.892171
PctPublicCoverageAlone  5.627e-01  2.780e-01   2.024 0.043095
PctWhite               -4.835e-02  6.361e-02  -0.760 0.447280
PctBlack                3.708e-02  6.232e-02   0.595 0.551899
PctAsian               -2.683e-01  1.989e-01  -1.349 0.177477
PctOtherRace           -9.938e-01  1.293e-01  -7.687 2.12e-14
PctMarriedHouseholds   -2.982e-01  1.531e-01  -1.947 0.051613

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent         .
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
PctHS18_24             ***
PctBachDeg18_24        **
```

```
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2570 degrees of freedom
Multiple R-squared:  0.4728,	Adjusted R-squared:  0.4689
F-statistic: 121.3 on 19 and 2570 DF,  p-value: < 2.2e-16

> summary(LR2)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent +
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctSomeCol18_24 +
    PctBachDeg18_24 + PctPrivateCoverage + PctPublicCoverage +
    PctPublicCoverageAlone + PctWhite + PctBlack + PctAsian +
    PctOtherRace + PctMarriedHouseholds, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-86.368 -12.179  -0.142  11.648 127.281

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.051e+02  1.497e+01   7.022 2.79e-12
incidenceRate           2.177e-01  8.222e-03  26.484  < 2e-16
medIncome              -2.647e-04  7.985e-05  -3.314 0.000931
povertyPercent          3.094e-01  1.697e-01   1.823 0.068440
MedianAge               2.211e-03  9.632e-03   0.230 0.818497
MedianAgeMale          -2.055e-01  2.293e-01  -0.896 0.370160
MedianAgeFemale        -1.377e-01  2.393e-01  -0.576 0.564976
AvgHouseholdSize        6.071e-01  1.202e+00   0.505 0.613465
PercentMarried          1.758e-01  1.568e-01   1.122 0.262074
PctNoHS18_24           -4.241e-02  6.492e-02  -0.653 0.513636
PctHS18_24              4.610e-01  5.636e-02   8.181 4.39e-16
PctSomeCol18_24         1.112e-02  8.397e-02   0.132 0.894639
PctBachDeg18_24        -3.423e-01  1.197e-01  -2.860 0.004277
PctPrivateCoverage     -2.747e-01  1.136e-01  -2.419 0.015653
PctPublicCoverage       2.916e-02  2.137e-01   0.136 0.891447
PctPublicCoverageAlone  5.624e-01  2.781e-01   2.022 0.043271
PctWhite               -4.830e-02  6.362e-02  -0.759 0.447871
PctBlack                3.724e-02  6.234e-02   0.597 0.550318
PctAsian               -2.683e-01  1.990e-01  -1.348 0.177667
PctOtherRace           -9.937e-01  1.293e-01  -7.685 2.17e-14
PctMarriedHouseholds   -2.991e-01  1.533e-01  -1.951 0.051200

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent         .
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
PctHS18_24             ***
```

```
PctSomeCol18_24
PctBachDeg18_24        **
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2569 degrees of freedom
Multiple R-squared:  0.4728,	Adjusted R-squared:  0.4687 
F-statistic: 115.2 on 20 and 2569 DF,  p-value: < 2.2e-16

> summary(LR1)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent + 
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize + 
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 + 
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone + 
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds, 
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-86.338 -12.160  -0.137  11.656 127.254 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.057e+02  1.422e+01   7.435 1.42e-13
incidenceRate           2.177e-01  8.218e-03  26.494  < 2e-16
medIncome              -2.648e-04  7.983e-05  -3.317 0.000922
povertyPercent          3.093e-01  1.697e-01   1.823 0.068467
MedianAge               2.215e-03  9.630e-03   0.230 0.818095
MedianAgeMale          -2.048e-01  2.292e-01  -0.893 0.371682
MedianAgeFemale        -1.382e-01  2.392e-01  -0.578 0.563459
AvgHouseholdSize        6.104e-01  1.201e+00   0.508 0.611419
PercentMarried          1.748e-01  1.565e-01   1.117 0.264197
PctNoHS18_24           -4.513e-02  6.158e-02  -0.733 0.463691
PctHS18_24              4.582e-01  5.217e-02   8.782  < 2e-16
PctBachDeg18_24        -3.448e-01  1.182e-01  -2.918 0.003553
PctPrivateCoverage     -2.744e-01  1.135e-01  -2.417 0.015711
PctPublicCoverage       2.896e-02  2.136e-01   0.136 0.892171
PctPublicCoverageAlone  5.627e-01  2.780e-01   2.024 0.043095
PctWhite               -4.835e-02  6.361e-02  -0.760 0.447280
PctBlack                3.708e-02  6.232e-02   0.595 0.551899
PctAsian               -2.683e-01  1.989e-01  -1.349 0.177477
PctOtherRace           -9.938e-01  1.293e-01  -7.687 2.12e-14
PctMarriedHouseholds   -2.982e-01  1.531e-01  -1.947 0.051613

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent         .
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
```

```
PctHS18_24             ***
PctBachDeg18_24        **
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2570 degrees of freedom
Multiple R-squared:  0.4728,	Adjusted R-squared:  0.4689 
F-statistic: 121.3 on 19 and 2570 DF,  p-value: < 2.2e-16

> summary(LR5)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent + 
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize + 
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 + 
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone + 
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds, 
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-85.035 -11.981  -0.135  11.704 129.847 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.206e+02  1.488e+01   8.104 8.11e-16
incidenceRate           2.177e-01  8.197e-03  26.562  < 2e-16
medIncome              -4.684e-04  1.002e-04  -4.675 3.09e-06
povertyPercent          8.939e-02  1.818e-01   0.492  0.62298
MedianAge               2.477e-03  9.610e-03   0.258  0.79663
MedianAgeMale          -1.668e-01  2.289e-01  -0.729  0.46635
MedianAgeFemale        -1.615e-01  2.387e-01  -0.676  0.49880
AvgHouseholdSize        6.984e-01  1.198e+00   0.583  0.56005
PercentMarried          1.130e-01  1.570e-01   0.720  0.47148
PctNoHS18_24           -4.484e-02  6.138e-02  -0.731  0.46515
PctHS18_24              4.587e-01  5.206e-02   8.811  < 2e-16
PctBachDeg18_24        -3.704e-01  1.159e-01  -3.195  0.00142
PctPrivateCoverage     -2.289e-01  1.140e-01  -2.009  0.04467
PctPublicCoverage      -8.077e-02  2.144e-01  -0.377  0.70641
PctPublicCoverageAlone  6.787e-01  2.786e-01   2.436  0.01493
PctWhite               -7.036e-02  6.368e-02  -1.105  0.26933
PctBlack                2.203e-02  6.236e-02   0.353  0.72390
PctAsian               -2.956e-01  1.949e-01  -1.516  0.12954
PctOtherRace           -9.799e-01  1.291e-01  -7.593 4.35e-14
PctMarriedHouseholds   -2.674e-01  1.513e-01  -1.767  0.07727

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
```

```
PctHS18_24             ***
PctBachDeg18_24        **
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.32 on 2570 degrees of freedom
Multiple R-squared:  0.475,	Adjusted R-squared:  0.4711
F-statistic: 122.4 on 19 and 2570 DF,  p-value: < 2.2e-16

> summary(LR6)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + MedianAge +
    AvgHouseholdSize + PctBlack + PctAsian + PctOtherRace, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-79.812 -13.020  -0.769  12.190 122.468

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.074e+02  4.924e+00  21.813  < 2e-16 ***
incidenceRate     2.263e-01  8.180e-03  27.666  < 2e-16 ***
medIncome        -9.349e-04  4.053e-05 -23.067  < 2e-16 ***
MedianAge        -2.352e-03  1.008e-02  -0.233    0.815
AvgHouseholdSize  5.421e+00  1.096e+00   4.948 7.98e-07 ***
PctBlack          2.004e-01  3.082e-02   6.504 9.33e-11 ***
PctAsian         -1.709e-02  1.804e-01  -0.095    0.925
PctOtherRace     -6.210e-01  1.231e-01  -5.044 4.87e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.46 on 2582 degrees of freedom
Multiple R-squared:  0.4115,	Adjusted R-squared:  0.4099
F-statistic: 257.9 on 7 and 2582 DF,  p-value: < 2.2e-16

> summary(LR7)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent +
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 +
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-101.36  -11.48   -0.04   11.57   87.20

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.445e+02  1.343e+01  10.754  < 2e-16
incidenceRate           1.799e-01  7.764e-03  23.168  < 2e-16
medIncome              -3.549e-04  7.542e-05  -4.706 2.65e-06
povertyPercent         -8.541e-02  1.603e-01  -0.533  0.59429
```

```
MedianAge              -9.394e-04  9.098e-03  -0.103  0.91777
MedianAgeMale          -3.632e-01  2.165e-01  -1.677  0.09364
MedianAgeFemale        -1.653e-02  2.260e-01  -0.073  0.94170
AvgHouseholdSize        2.603e-01  1.135e+00   0.229  0.81857
PercentMarried          8.696e-02  1.479e-01   0.588  0.55655
PctNoHS18_24           -3.582e-02  5.818e-02  -0.616  0.53811
PctHS18_24              4.347e-01  4.929e-02   8.820  < 2e-16
PctBachDeg18_24        -2.282e-01  1.116e-01  -2.044  0.04103
PctPrivateCoverage     -3.494e-01  1.073e-01  -3.258  0.00114
PctPublicCoverage       2.631e-02  2.018e-01   0.130  0.89628
PctPublicCoverageAlone  4.421e-01  2.627e-01   1.683  0.09246
PctWhite               -1.739e-02  6.010e-02  -0.289  0.77229
PctBlack                9.119e-02  5.887e-02   1.549  0.12153
PctAsian               -2.758e-01  1.879e-01  -1.467  0.14241
PctOtherRace           -1.006e+00  1.221e-01  -8.238 2.76e-16
PctMarriedHouseholds   -3.051e-01  1.447e-01  -2.109  0.03504

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent +
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 +
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-85.035 -11.981  -0.135  11.704 129.847

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.206e+02  1.488e+01   8.104 8.11e-16
incidenceRate           2.177e-01  8.197e-03  26.562  < 2e-16
medIncome              -4.684e-04  1.002e-04  -4.675 3.09e-06
povertyPercent          8.939e-02  1.818e-01   0.492  0.62298
MedianAge               2.477e-03  9.610e-03   0.258  0.79663
MedianAgeMale          -1.668e-01  2.289e-01  -0.729  0.46635
MedianAgeFemale        -1.615e-01  2.387e-01  -0.676  0.49880
AvgHouseholdSize        6.984e-01  1.198e+00   0.583  0.56005
PercentMarried          1.130e-01  1.570e-01   0.720  0.47148
PctNoHS18_24           -4.484e-02  6.138e-02  -0.731  0.46515
PctHS18_24              4.587e-01  5.206e-02   8.811  < 2e-16
PctBachDeg18_24        -3.704e-01  1.159e-01  -3.195  0.00142
PctPrivateCoverage     -2.289e-01  1.140e-01  -2.009  0.04467
PctPublicCoverage      -8.077e-02  2.144e-01  -0.377  0.70641
PctPublicCoverageAlone  6.787e-01  2.786e-01   2.436  0.01493
PctWhite               -7.036e-02  6.368e-02  -1.105  0.26933
PctBlack                2.203e-02  6.236e-02   0.353  0.72390
PctAsian               -2.956e-01  1.949e-01  -1.516  0.12954
PctOtherRace           -9.799e-01  1.291e-01  -7.593 4.35e-14
PctMarriedHouseholds   -2.674e-01  1.513e-01  -1.767  0.07727

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
PctHS18_24             ***
PctBachDeg18_24        **
```

```
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.32 on 2570 degrees of freedom
Multiple R-squared:  0.475,	Adjusted R-squared:  0.4711
F-statistic: 122.4 on 19 and 2570 DF,  p-value: < 2.2e-16
```

```
> msetrain_n
[1] 411.3217
> msetest_n
[1] 414.5908
> msetrain_median
[1] 411.3189
> msetest_median
[1] 414.54
> msetrain1
[1] 411.3217
> msetest1
[1] 414.5908
> msetrain2 #optimum msetrain
[1] 409.5991
> msetest2   #optimum msetest
[1] 416.1014
> msetrain3
[1] 459.1824
> msetest3
[1] 460.9086
> msetrain4 #optimum msetrain
[1] 409.5991
> msetest4   #optimum msetest
[1] 416.1014
```

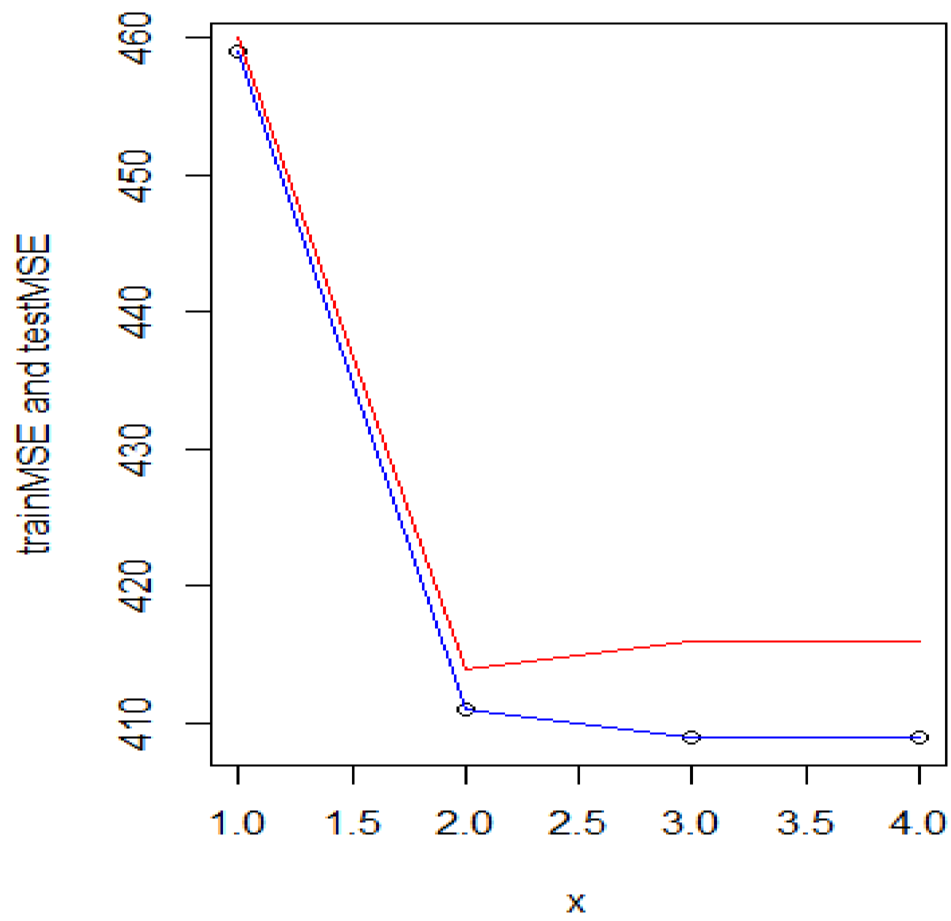


Figure 5Test MSE vs Train MSE (red line- test MSE, blue line- train MSE)

b. What variables are significant? Insignificant? How does removing insignificant variables affect model performance?

Ans- The variables incidenceRate, medIncome, PctHS18_24, PctBachDeg18_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace are significant.

The model performs better than the original model because it has a lower test MSE compared to the original model (test MSE – 413.59).

Code-

#removing insignificant variables

fix(train)

LR4 = lm(TARGET_deathRate~incidenceRate+medIncome+PctHS18_24+PctOtherRace+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)

LR4.pred= predict(LR4 ,newdata= test)

msetrain_sign=mean((train$TARGET_deathRate - fitted(LR4))^2)

msetrain_sign

msetest_sign=mean(((test$TARGET_deathRate) - (LR4.pred))^2)

msetest_sign

Output-

```
msetrain_sign
[1] 415.7994
> msetest_sign
[1] 413.5129

>
```

c. Present and interpret model diagnosis. What insights did you obtain to improve the model from diagnosis?

Ans-

Code-

model diagnosis

par(mfrow=c(2,2))

plot(LR3)

Output-

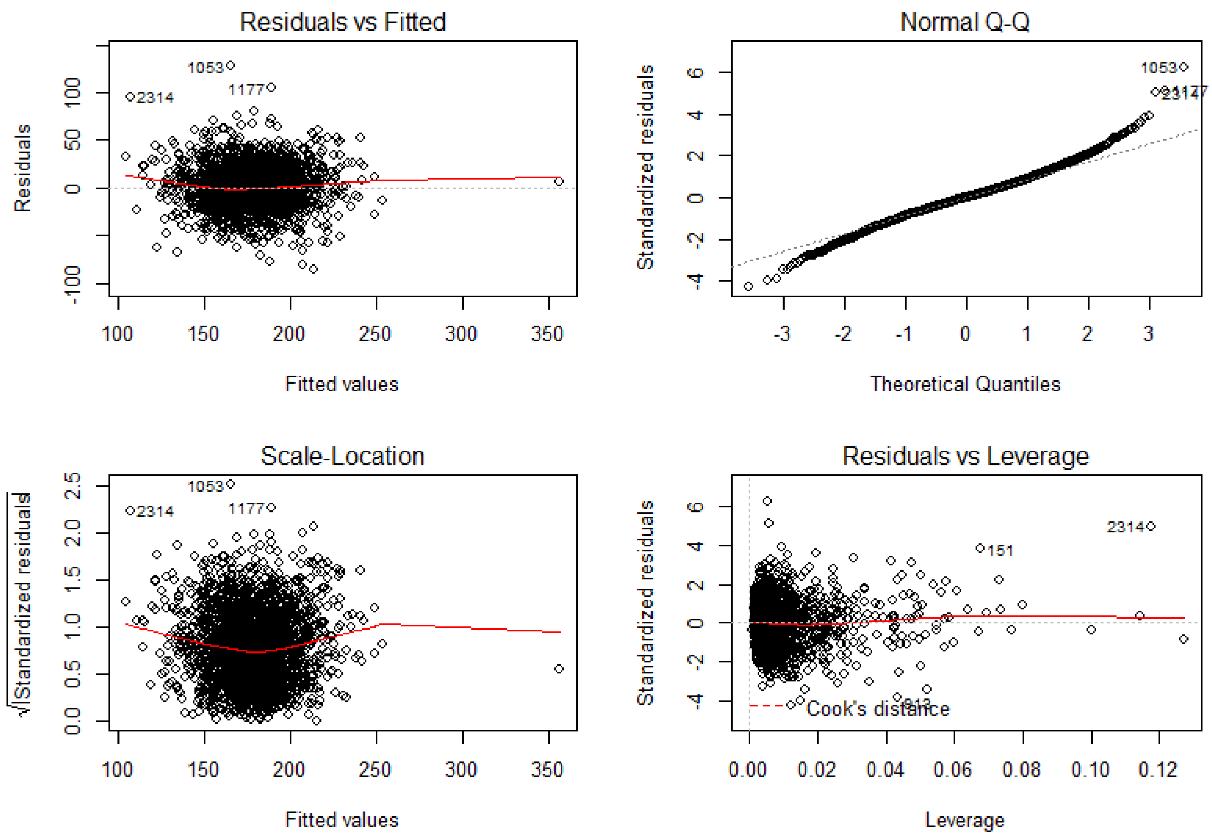


Figure 6-Diagnostics Plot

1. Plot-1 (Residual vs Fitted)- - This plot is used to check the linear relationship assumptions (This plot shows if residuals have non-linear patterns). From the diagnostic plot drawn, the residuals have a linear relationship.
2. Plot-2 (Normal Q-Q)- This plot shows if the residuals are normally distributed. They are normally distributed if all the points fall on a straight line. From the plot obtained, the residuals are normally distributed.
3. Plot-3 (Scale- Location)- It's also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors (homogeneity of variance of the residuals). From the plot obtained the residuals are spread equally along the range of the predictors.

4. Plot-4(Residuals vs leverage)- This plot helps us to find if the outliers are influential in linear regression analysis. This can be found out by Cook's distance. From our plot we can infer that the outliers are not influential since we can't see cook's distance line (since its well inside the Cook's distance line).

d. Include few non-linear and interaction terms and evaluate how they affect model performance and diagnosis.

Ans-

Code-

```
#inputing non-linear terms

attach(train)

LR8 = lm(TARGET_deathRate~incidenceRate+sqrt(medIncome)+povertyPercent+MedianAge+sqrt(MedianAgeMale)+MedianAgeFemale+AvgHouseholdSize+(PercentMarried)^2+PctNoHS18_24^3+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

        :medIncome, data =train)

summary(LR8)

LR8.pred= predict(LR8 ,newdata= test)

msetrain5=mean((train$TARGET_deathRate-fitted(LR8))^2)

msetrain5 #optimum msetrain

msetest5=mean(((test$TARGET_deathRate) - (LR8.pred))^2)

msetest5   #optimum msetest
```

par(mfrow=c(2,2))

plot(LR8)

Output-

```
Residuals:
    Min      1Q  Median      3Q     Max 
-109.06  -11.57    0.16   11.37   83.58 

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    1.985e+02  2.414e+01   8.222 3.13e-16 ***
incidenceRate                  1.821e-01  7.740e-03  23.527  < 2e-16 ***
sqrt(medIncome)               -3.023e-01  7.953e-02  -3.801 0.000147 ***
povertyPercent                -3.734e-01  1.820e-01  -2.052 0.040264 *  
MedianAge                     -7.476e-04  9.085e-03  -0.082 0.934427    
sqrt(MedianAgeMale)           -6.145e-01  2.743e+00  -0.224 0.822760    
MedianAgeFemale               -2.337e-01  2.311e-01  -1.012 0.311838    
AvgHouseholdSize              -3.703e-01  1.123e+00  -0.330 0.741644    
PercentMarried                -2.704e-01  1.338e-01  -2.021 0.043418 *  
PctNoHS18_24                  -3.960e-02  5.803e-02  -0.682 0.495035    
PctHS18_24                     4.253e-01  4.916e-02   8.651  < 2e-16 ***
PctBachDeg18_24               -1.749e-01  1.097e-01  -1.594 0.111054    
PctPrivateCoverage            -3.102e-01  1.075e-01  -2.885 0.003945 ** 
PctPublicCoverage             -7.788e-02  2.012e-01  -0.387 0.698712    
PctPublicCoverageAlone         5.502e-01  2.618e-01   2.102 0.035692 *  
PctWhite                      -6.322e-02  5.972e-02  -1.059 0.289886    
PctBlack                       6.550e-02  5.947e-02   1.101 0.270800    
PctAsian                      -2.396e-01  1.871e-01  -1.281 0.200349    
PctOtherRace                  -9.940e-01  1.220e-01  -8.151 5.59e-16 ***
PctMarriedHouseholds:medIncome  1.868e-06  2.182e-06   0.856 0.391962    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> msetrain5
[1] 366.141
> msetest5
[1] 410.0572
```

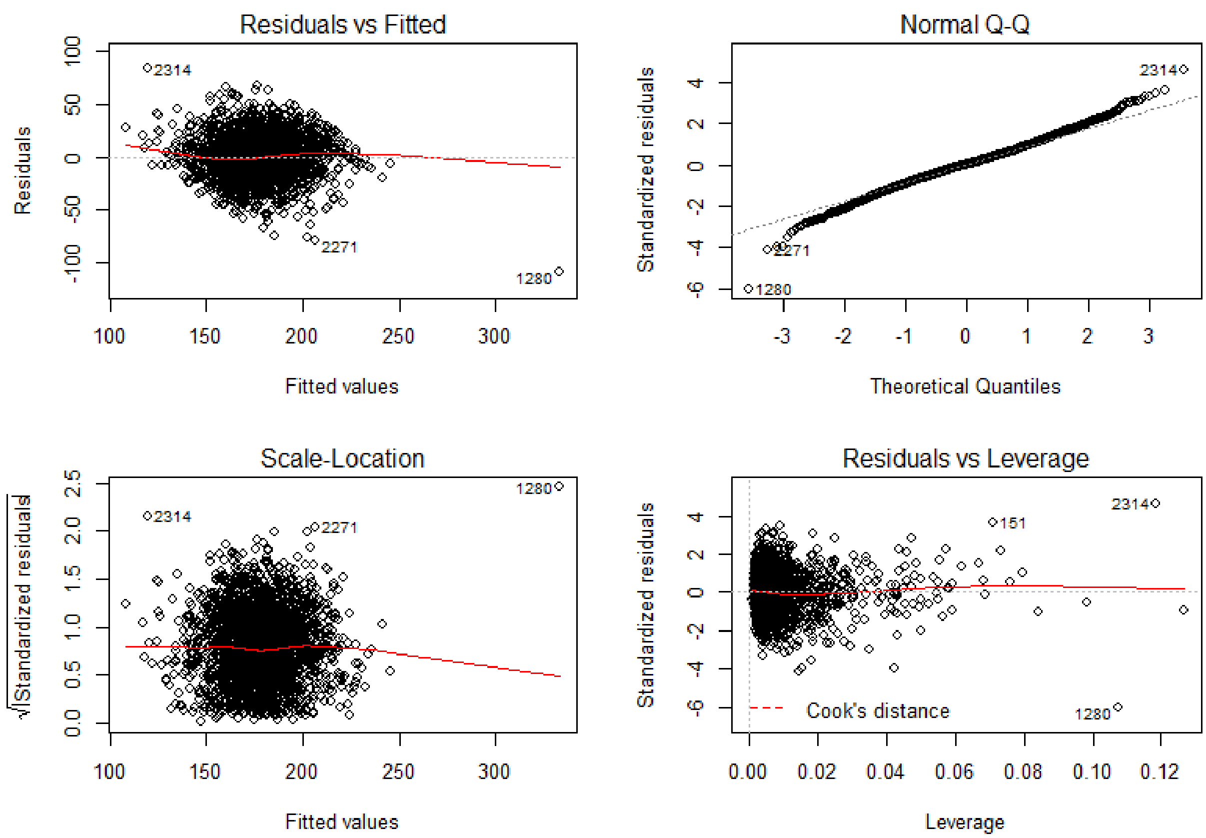


Figure 7-Diagnostics Plot

Summary- The model performance increases due to the addition of interaction terms and non-linear terms since the test and train MSE (410.0572 and 366.141 respectively) are lower when compared to the test and train MSE of the original model.

1.Plot-1 (Residual vs Fitted)- - This plot is used to check the linear relationship assumptions (This plot shows if residuals have non-linear patterns). From the diagnostic plot drawn, the residuals have a linear relationship.

2.Plot-2 (Normal Q-Q)- This plot shows if the residuals are normally distributed. They are normally distributed if all the points fall on a straight line. From the plot obtained, the residuals are normally distributed.

3.Plot-3 (Scale- Location)- It's also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors (homogeneity of variance of the residuals). From the plot obtained the residuals are spread equally along the range of the predictors.

4.Plot-4(Residuals vs leverage)- This plot helps us to find if the outliers are influential in linear regression analysis. This can be found out by Cook's distance. From our plot we can infer that the outliers are not influential since we can't see cook's distance line (since its well inside the Cook's distance line).

3. KNN-
    a. Split CanverData.csv data into 70% training and 30% testing.

Code-

```
library(FNN)

library(class)

set.seed(132)

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

x <- train$medIncome

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x
```

n <- nrow(train) * 0.7

T <- sample(nrow(train), size = n)

train1 <- train[T,-c(8,13)]

test1 <- train[-T,-c(8,13)]

train.Y = train1$TARGET_deathRate

b. Develop KNN model for predicting Cancer Mortality. Evaluate test MSE for at least 5 different values of K and find the K that minimizes test MSE.

   Ans- The test MSE for the 5 different values of K (K=1,2,3,4,5) are477.8506, 402.0091, 393.5481, 407.6580 and 418.2897. K= 3 minimizes the test MSE since it has the lowest test MSE of 393.5481 among the other K values

Code-

```
error = c(0,0,0,0,0)

for(i in 1:5)

{

 knn <- knn.reg(train1, test1, train.Y, k=i)

 knntestmse =mean(((test1$TARGET_deathRate) - (knn$pred))^2)

 error[i] = knntestmse

}

Error
```

Output-

```
error
[1] 477.8506 402.0091 393.5481 407.6580 418.2897
```

c. KNN is a non-linear technique, but does not work well with high dimensional data. Try to identify important variables from Linear Regression model and use only a subset of important features in the KNN model. Document impact on test performance.

Ans- The variables incidenceRate, medIncome, PctHS18_24, PctBachDeg18_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace are significant. This can be found out using the p-values obtained from the linear regression. The test MSE for K = 1,2,3,4,5 is 447.0869, 394.2024, 366.6698, 389.5896, 425.7973.
Using significant variables improved the performance of the KNN model since the test MSE for the same seed is less when compared to the KNN model when all the variables are used. K=3 is the optimum K values since the test MSE is 366.6698.

Code-

```
#significant variables

train2 <- train[T,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

test2 <- train[-T,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

train.Y1 = train2$TARGET_deathRate

fix(train2)

knn3 <- knn.reg(train2, test2, train.Y1, k=1)

error2 = c(0,0,0,0,0)

for(i in 1:5)

{
```

knn3 <- knn.reg(train2, test2, train.Y1,k=i)

knntestmse3 =mean(((test2$TARGET_deathRate) - (knn3$pred))^2)

error2[i] = knntestmse3

}

error2

error2

Output-

```
error2
[1] 456.5744 398.1046 389.2878 398.5234 418.7392
```

4. Feature Selection
   a. Write an “Executive Summary” section documenting your interpretation of the i mportant features impacting cancer mortality and how they influence cancer mortality.

Ans – The feature selection from a data set in R can be done creating a correlation matrix. Visually it can also be done by plotting a correlation plot from the matrix.

In this project correlation matrix and correlation plot are used to select the features in the initial stages. Later the p-values obtained from the linear regression are used to select the significant features that are used to improve the model performance.

Interpreting Correlation plot-

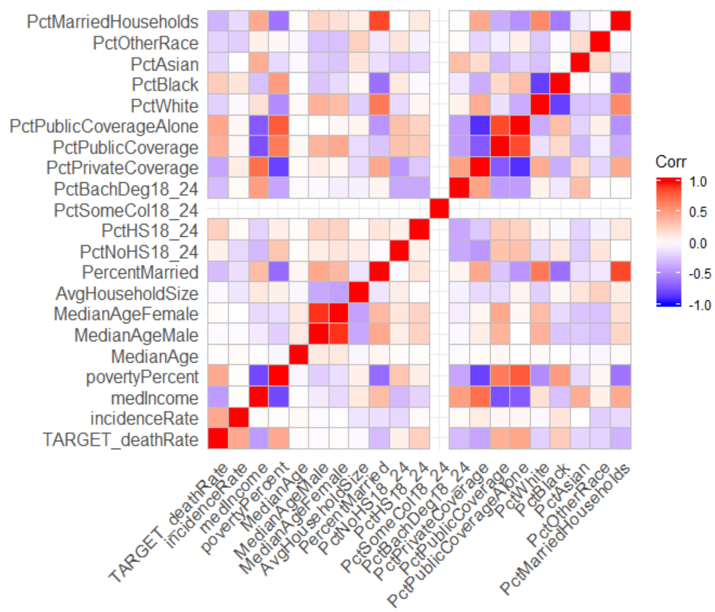


Figure 8-Correlation graph

In this correlation plot, correlation scale which is displayed in the right hand corner of the plot is used to select a feature(darker the color, the higher the correlation).

Interpreting Linear Regression-

```
> summary(LR3)

Call:
lm(formula = TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent +
    MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +
    PercentMarried + PctNoHS18_24 + PctHS18_24 + PctBachDeg18_24 +
    PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +
    PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-86.338 -12.160  -0.137  11.656 127.254

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.057e+02  1.422e+01   7.435 1.42e-13
incidenceRate           2.177e-01  8.218e-03  26.494  < 2e-16
medIncome              -2.648e-04  7.983e-05  -3.317 0.000922
povertyPercent          3.093e-01  1.697e-01   1.823 0.068467
MedianAge               2.215e-03  9.630e-03   0.230 0.818095
MedianAgeMale          -2.048e-01  2.292e-01  -0.893 0.371682
MedianAgeFemale        -1.382e-01  2.392e-01  -0.578 0.563459
AvgHouseholdSize        6.104e-01  1.201e+00   0.508 0.611419
PercentMarried          1.748e-01  1.565e-01   1.117 0.264197
PctNoHS18_24           -4.513e-02  6.158e-02  -0.733 0.463691
PctHS18_24              4.582e-01  5.217e-02   8.782  < 2e-16
```

```
PctBachDeg18_24        -3.448e-01  1.182e-01  -2.918 0.003553
PctPrivateCoverage     -2.744e-01  1.135e-01  -2.417 0.015711
PctPublicCoverage       2.896e-02  2.136e-01   0.136 0.892171
PctPublicCoverageAlone  5.627e-01  2.780e-01   2.024 0.043095
PctWhite               -4.835e-02  6.361e-02  -0.760 0.447280
PctBlack                3.708e-02  6.232e-02   0.595 0.551899
PctAsian               -2.683e-01  1.989e-01  -1.349 0.177477
PctOtherRace           -9.938e-01  1.293e-01  -7.687 2.12e-14
PctMarriedHouseholds   -2.982e-01  1.531e-01  -1.947 0.051613

(Intercept)            ***
incidenceRate          ***
medIncome              ***
povertyPercent         .
MedianAge
MedianAgeMale
MedianAgeFemale
AvgHouseholdSize
PercentMarried
PctNoHS18_24
PctHS18_24             ***
PctBachDeg18_24        **
PctPrivateCoverage     *
PctPublicCoverage
PctPublicCoverageAlone *
PctWhite
PctBlack
PctAsian
PctOtherRace           ***
PctMarriedHouseholds   .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Features can be selected from linear regression by observing the corresponding p-values. Smaller the p-value, higher the significance. Significance code displayed at the bottom can be used to interpret the p-value.

5. Performance reporting on Holdout data-
   a. Summarize and compare the model performance (MSE) of LR and KNN on holdout dataset as a table.

Code-

#question-5

set.seed(132)

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

```
test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train <- train[,-c(8,13)]

test <- test[,-c(8,13)]

error1 = c(0,0,0,0,0)

for(i in 1:5)

{

  knn1<- knn.reg(train, test, train$TARGET_deathRate, k=i)

  knntestmse1 =mean(((test$TARGET_deathRate) - (knn1$pred))^2)

  error1[i] = knntestmse1

}

error1

?knn.reg


#significant features

train3 <- train[,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

test3 <- test[,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

train.Y2 = train3$TARGET_deathRate

error3 = c(0,0,0,0,0)

for(i in 1:5)

{

  knn4 <- knn.reg(train3, test3, train.Y2,k=i)
```

knntestmse4 =mean(((test$TARGET_deathRate) - (knn4$pred))^2)

error3[i] = knntestmse4

}

error3

Output-

```
> error1
[1] 514.7173 424.0900 414.3958 421.1690 411.4077
> error3
[1] 509.1923 409.7158 393.3701 411.5611 414.4351
```

SR. No	Model name	Test MSE
1	Linear Regression	414.5908
2	Linear Regression with significant variables	413.9023
3	KNN	411.4077
4	KNN with significant variables	367.113

Summary-

The Test MSE for Linear Regression, Linear Regression with significant variables, KNN, KNN with significant variables are 414.5908, 413.9023, 411.4077, 367.113 respectively. KNN perform better than Linear Regression and Linear Regression with significant variables since KNN is a no linear method but KNN with significant variables performs better than KNN since KNN does not work well with high dimensional data.

R-CODE(FULL)-

#question 2

#promising variables

```
train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

library(ggplot2)

mydata <- train[, -c(8)]

cormat<-signif(cor(mydata),2)

cormat

install.packages("ggcorrplot")

library(ggcorrplot)

ggcorrplot(cormat)

#missing values

library(Amelia)

sum(is.na(train$PctSomeCol18_24))

missmap(train, main="Train Data - Missings Map",

    col=c("yellow", "black"), legend=FALSE)

missmap(test, main="Test Data - Missings Map",

    col=c("yellow", "black"), legend=FALSE)


#treating missing values

#method1 - neglecting the coloumn
```

LR3 =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR3)

LR3.pred= predict(LR3 ,newdata= test)

msetrain_n=mean((train$TARGET_deathRate-fitted(LR3))^2)

msetrain_n

msetest_n=mean(((test$TARGET_deathRate) - (LR3.pred))^2)

msetest_n

#Method2 - inputing median

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24)]= median(train$PctSomeCol18_24, na.rm= TRUE)

test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]= median(test$PctSomeCol18_24, na.rm= TRUE)

```
LR2 =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianA
ge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+P
ctNoHS18_24+PctHS18_24+PctSomeCol18_24+PctBachDeg18_24+PctPrivate
Coverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+P
ctAsian+PctOtherRace+PctMarriedHouseholds
    , data =train)

summary(LR2)

LR2.pred= predict(LR2 ,newdata= test)

LR2.pred

msetrain_median=mean((train$TARGET_deathRate-fitted(LR2))^2)

msetrain_median

msetest_median=mean(((test$TARGET_deathRate) - (LR2.pred))^2)

msetest_median


#method3- Inputing the mean

train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24) ]=
mean(train$PctSomeCol18_24, na.rm= TRUE)

test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]=
mean(test$PctSomeCol18_24, na.rm= TRUE)
```

```
LR1 =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds
    , data =train)
summary(LR1)
LR1.pred= predict(LR1 ,newdata= test)
msetrain1=mean((train$TARGET_deathRate-fitted(LR1))^2)
msetrain1
msetest1=mean(((test$TARGET_deathRate) - (LR1.pred))^2)
msetest1

attach(train)
#removing insignificant variables
fix(train)
LR4 =
lm(TARGET_deathRate~incidenceRate+medIncome+PctHS18_24+PctOtherRace+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)
LR4.pred= predict(LR4 ,newdata= test)
msetrain_sign=mean((train$TARGET_deathRate - fitted(LR4))^2)
msetrain_sign
```

```
msetest_sign=mean(((test$TARGET_deathRate) - (LR4.pred))^2)
msetest_sign


#finding outliers
OutVals = boxplot(train, plot=FALSE)$out
OutVals1 = boxplot(medIncome, plot=FALSE)$out
plot(OutVals1)
plot(OutVals)
boxplot(train)
library(outliers)
outlier(medIncome)


#treating outlies- by using capping
x <- train$medIncome
qnt <- quantile(x, probs=c(.25, .75))
caps <- quantile(x, probs=c(.05, .95))
H <- 1.5 * IQR(x)
x[x < (qnt[1] - H)] <- caps[1]
x[x > (qnt[2] + H)] <- caps[2]
train$medIncome = x
boxplot(train$medIncome)
```

```
LR5  =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds
        , data =train)

summary(LR5)

LR5.pred= predict(LR5 ,newdata= test)

msetrain2=mean((train$TARGET_deathRate-fitted(LR5))^2)

msetrain2 #optimum msetrain

msetest2=mean(((test$TARGET_deathRate) - (LR5.pred))^2)

msetest2   #optimum msetest


#finding collinearity


#install.packages('olsrr')

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

train$PctSomeCol18_24[is.na(train$PctSomeCol18_24) ]= median(train$PctSomeCol18_24, na.rm= TRUE)
```

```
test$PctSomeCol18_24[is.na(test$PctSomeCol18_24)]=
median(test$PctSomeCol18_24, na.rm= TRUE)

library(olsrr)

ols_vif_tol(LR3)

#treating collinearity - neglecting the variables


LR6  =
lm(TARGET_deathRate~incidenceRate+medIncome+MedianAge+AvgHousehol
dSize+PctBlack+PctAsian+PctOtherRace, data =train)

summary(LR6)

LR6.pred= predict(LR6 ,newdata= test)

msetrain3=mean((train$TARGET_deathRate-fitted(LR6))^2)

msetrain3

msetest3=mean(((test$TARGET_deathRate) - (LR6.pred))^2)

msetest3



#optimummodel

train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

x <- train$medIncome
```

```
qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x

LR7 =
lm(TARGET_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18_24+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

        , data =train)

summary(LR7)

LR7.pred= predict(LR7 ,newdata= test)

msetrain4=mean((train$TARGET_deathRate-fitted(LR7))^2)

msetrain4 #optimum msetrain

msetest4=mean(((test$TARGET_deathRate) - (LR7.pred))^2)

msetest4   #optimum msetest


#inputing non-linear terms

attach(train)
```

```
LR8 =
lm(TARGET_deathRate~incidenceRate+sqrt(medIncome)+povertyPercent+MedianAge+sqrt(MedianAgeMale)+MedianAgeFemale+AvgHouseholdSize+(PercentMarried)^2+PctNoHS18_24^3+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds
        :medIncome, data =train)
summary(LR8)
LR8.pred= predict(LR8 ,newdata= test)
msetrain5=mean((train$TARGET_deathRate-fitted(LR8))^2)
msetrain5 #optimum msetrain
msetest5=mean(((test$TARGET_deathRate) - (LR8.pred))^2)
msetest5   #optimum msetest

# model diagnosis
par(mfrow=c(2,2))
plot(LR1)
plot(LR5)
plot(LR6)

#trainmse vs testmse
trainMSE= c(459,411,409,409)
testMSE= c(460,414,416,416)
```

```
#1= collinearity,2= neglecting, 3= outliers, 4= optimum in x

x= c(1,2,3,4)

plot(x,trainMSE, ylab='trainMSE and testMSE')

lines(testMSE, col = 'red')

lines(trainMSE, col='blue')


#question 3

library(FNN)

library(class)

set.seed(132)

train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

x <- train$medIncome

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 * IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x

n <- nrow(train) * 0.7
```

```
{

T <- sample(nrow(train), size = n)

train1 <- train[T,-c(8,13)]

test1 <- train[-T,-c(8,13)]

train.Y = train1$TARGET_deathRate

fix(train1)

error = c(0,0,0,0,0)

for(i in 1:5)

{

  knn <- knn.reg(train1, test1, train.Y, k=i)

  knntestmse =mean(((test1$TARGET_deathRate) - (knn$pred))^2)

  error[i] = knntestmse

}

error


train2 <- train[T,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

test2 <- train[-T,-c(4,5,6,7,8,9,10,13,12,15,17,18)]

train.Y1 = train2$TARGET_deathRate

fix(train2)

knn3 <- knn.reg(train2, test2, train.Y1, k=1)

error2 = c(0,0,0,0,0)

for(i in 1:5)

{
```

```
  knn3 <- knn.reg(train2, test2, train.Y1,k=i)

  knntestmse3 =mean(((test2$TARGET_deathRate) - (knn3$pred))^2)

  error2[i] = knntestmse3

}

error2
```

```
#question-6

set.seed(132)

train = read.csv("C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerData.csv ")

test = read.csv('C:¥¥Users¥¥KIRAN
KONDISETTI¥¥Desktop¥¥CancerHoldoutData.csv ')

x  = train$medIncome

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train$medIncome = x

train <- train[,-c(8,13)]

test <- test[,-c(8,13)]
```

```
y = test$TARGET_deathRate

error1 = c(0,0,0,0,0)

for(i in 1:5)

{

  knn1<- knn.reg(train, test, train$TARGET_deathRate, k=i)

  knntestmse1 =mean(((test$TARGET_deathRate) - (knn1$pred))^2)

  error1[i] = knntestmse1

}

error1

?knn.reg
```