Regression_analysis_cancerdata

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(leaps)
```

```
cancer<-read.csv('cancer_reg.csv')
print(names(cancer))
```

```
##  [1] "avganncount"              "avgdeathsperyear"
##  [3] "target_deathrate"         "incidencerate"
##  [5] "medincome"                "popest2015"
##  [7] "povertypercent"           "studypercap"
##  [9] "binnedinc"                "medianage"
## [11] "medianagemale"            "medianagefemale"
## [13] "geography"                "percentmarried"
## [15] "pctnohs18_24"             "pcths18_24"
## [17] "pctsomecol18_24"          "pctbachdeg18_24"
## [19] "pcths25_over"             "pctbachdeg25_over"
## [21] "pctemployed16_over"       "pctunemployed16_over"
## [23] "pctprivatecoverage"       "pctprivatecoveragealone"
## [25] "pctempprivcoverage"       "pctpubliccoverage"
## [27] "pctpubliccoveragealone"   "pctwhite"
## [29] "pctblack"                 "pctasian"
## [31] "pctotherrace"             "pctmarriedhouseholds"
## [33] "birthrate"
```

```
#dropping geography variable
mydata <-cancer[c(-13,-9)]
print("Dimension")
```

```
## [1] "Dimension"
```

```
print(dim(mydata))
```

```
## [1] 3047   31
```

```
# Split the data into training and test set
set.seed(123)
training.samples <- createDataPartition(mydata$target_deathrate, p = 0.8, list = FALSE)
train.data  <- mydata[training.samples, ]
test.data <- mydata[-training.samples, ]
```

```r
# check for NA's
na_s <- apply(train.data,2,function(x) any(is.na(x)))
print("Sum of NAs")
```

```
## [1] "Sum of NAs"
```

```r
print(sum(!na_s))
```

```
## [1] 28
```

```r
# removing NA's
# to get rid of any column that has one or more NAs
train.data2<-train.data[,colSums(is.na(train.data))==0]
print(dim(train.data2))
```

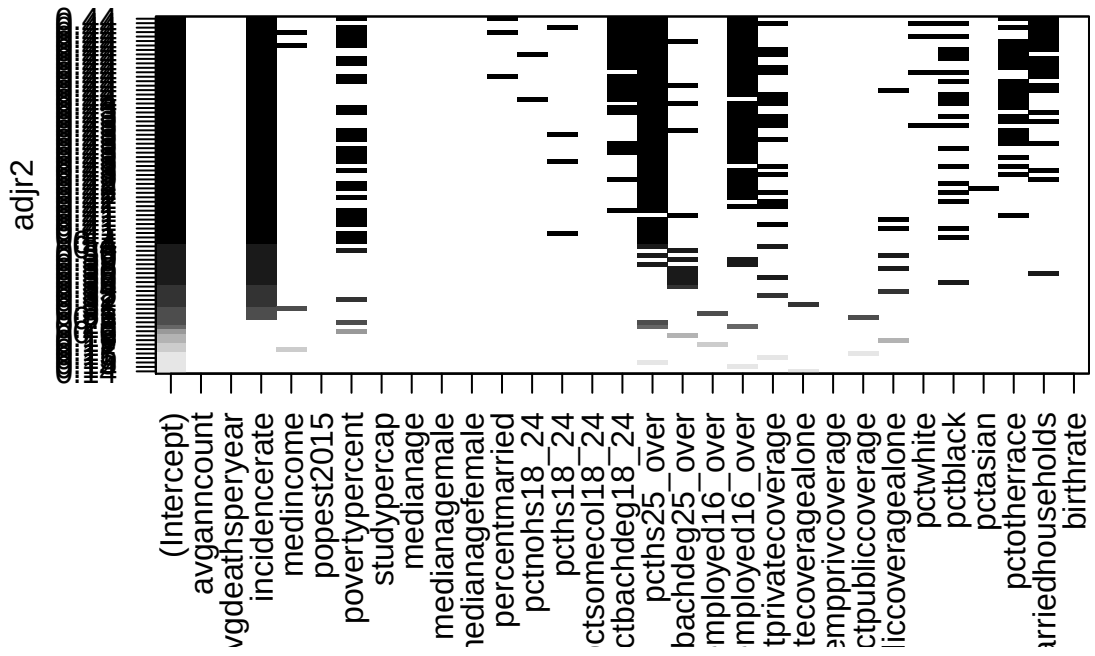
```
## [1] 2439   28
```

```r
#Build the model
model1 <- lm(target_deathrate ~. , data = train.data2)
#Make predictions
predictions <- predict(model1,test.data)

# Model performance
performance1 <- data.frame(
  RMSE = RMSE(predictions, test.data$target_deathrate),
  R2 = R2(predictions, test.data$target_deathrate)
)
```

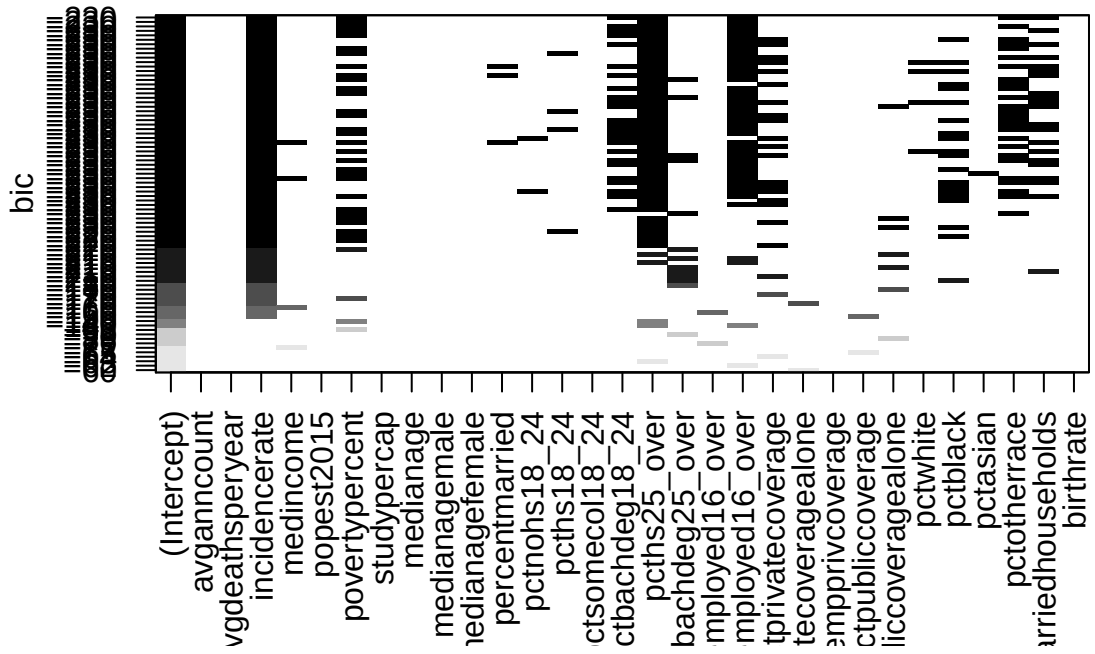
```r
# Checking for Multicolinearity

# Correlation
corr_train <- cor(train.data2[-c(9)],use="pairwise.complete.obs")

#finding highly correlated variables
highlyCorrelatedVars <- findCorrelation(corr_train, cutoff=(0.7),verbose = FALSE)
#print(highlyCorrelatedVars)

important_var=colnames(train.data2[,-highlyCorrelatedVars])
#print(important_var)


df <- train.data2[,-highlyCorrelatedVars]
df<-df[,colSums(is.na(df))==0]
#print(dim(df))

#Build the model
model2 <- lm(target_deathrate ~. , data = df)
#Make predictions
predictions <- predict(model2,test.data)

# Model performance
performance2 <- data.frame(
  RMSE = RMSE(predictions, test.data$target_deathrate),
  R2 = R2(predictions, test.data$target_deathrate)
)
```

```
#Variance Inflation factor
vif<-ols_vif_tol(model1)
vif2 <- vif[3]<4
vifVars <- colnames(train.data2)[vif2]
#print(vifVars)


df <- train.data2[,vifVars]
df<-df[,colSums(is.na(df))==0]
#print(dim(df))

#Build the model
model3 <- lm(target_deathrate ~. , data = df)
#Make predictions
predictions <- predict(model3,test.data)

# Model performance
performance3 <- data.frame(
  RMSE = RMSE(predictions, test.data$target_deathrate),
  R2 = R2(predictions, test.data$target_deathrate)
)
```

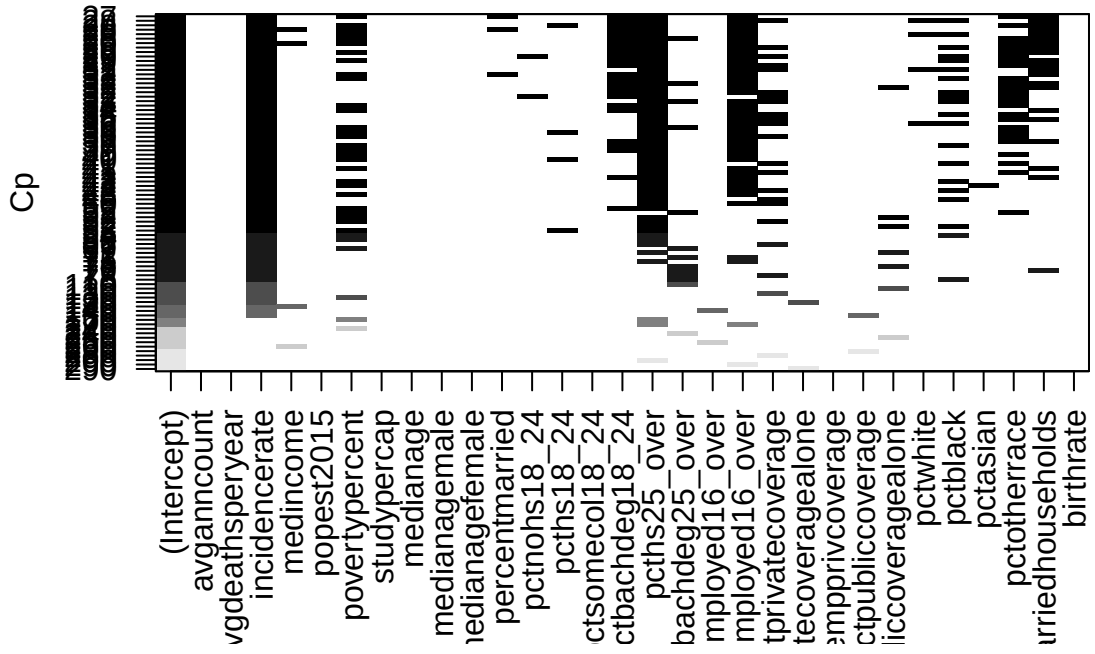
```
# Variable selection
var.selection <- regsubsets(target_deathrate~., data=train.data, nbest = 10)

plot(var.selection,scale="adjr2")
```




```
plot(var.selection,scale="bic")
```




```
plot(var.selection,scale="Cp")
```




```
model4 = lm(target_deathrate ~ incidencerate+medianagemale+percentmarried +pctbachdeg18_24+
              pctbachdeg25_over+pctunemployed16_over+pctunemployed16_over +pctmarriedhouseholds,data=tra

summary(model4)
```

```
##
## Call:
## lm(formula = target_deathrate ~ incidencerate + medianagemale +
##     percentmarried + pctbachdeg18_24 + pctbachdeg25_over + pctunemployed16_over +
##     pctunemployed16_over + pctmarriedhouseholds, data = train.data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -110.345  -11.427   -0.043   11.453  112.075
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          135.833194   6.620908  20.516  < 2e-16 ***
## incidencerate          0.206147   0.007462  27.626  < 2e-16 ***
## medianagemale         -0.211868   0.098149  -2.159   0.0310 *
## percentmarried         0.389777   0.152660   2.553   0.0107 *
## pctbachdeg18_24       -0.262544   0.116218  -2.259   0.0240 *
## pctbachdeg25_over     -2.005087   0.099534 -20.145  < 2e-16 ***
## pctunemployed16_over   1.126441   0.155332   7.252 5.50e-13 ***
## pctmarriedhouseholds  -0.820377   0.138842  -5.909 3.93e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.14 on 2431 degrees of freedom
## Multiple R-squared:  0.4744, Adjusted R-squared:  0.4729
## F-statistic: 313.5 on 7 and 2431 DF,  p-value: < 2.2e-16
```

```
# Prediciting the Test set results
predictions = predict(model4,test.data)

# Model performance
performance4 <- data.frame(
  RMSE = RMSE(predictions, test.data$target_deathrate),
  R2 = R2(predictions, test.data$target_deathrate)
)
```

```
## [1] "Model1 performance - without removing correlated variables"
```

```
##       RMSE        R2
## 1 20.35137 0.4647917
```

```
## [1] "Model2 performance -  after removing highly correlated variables"
```

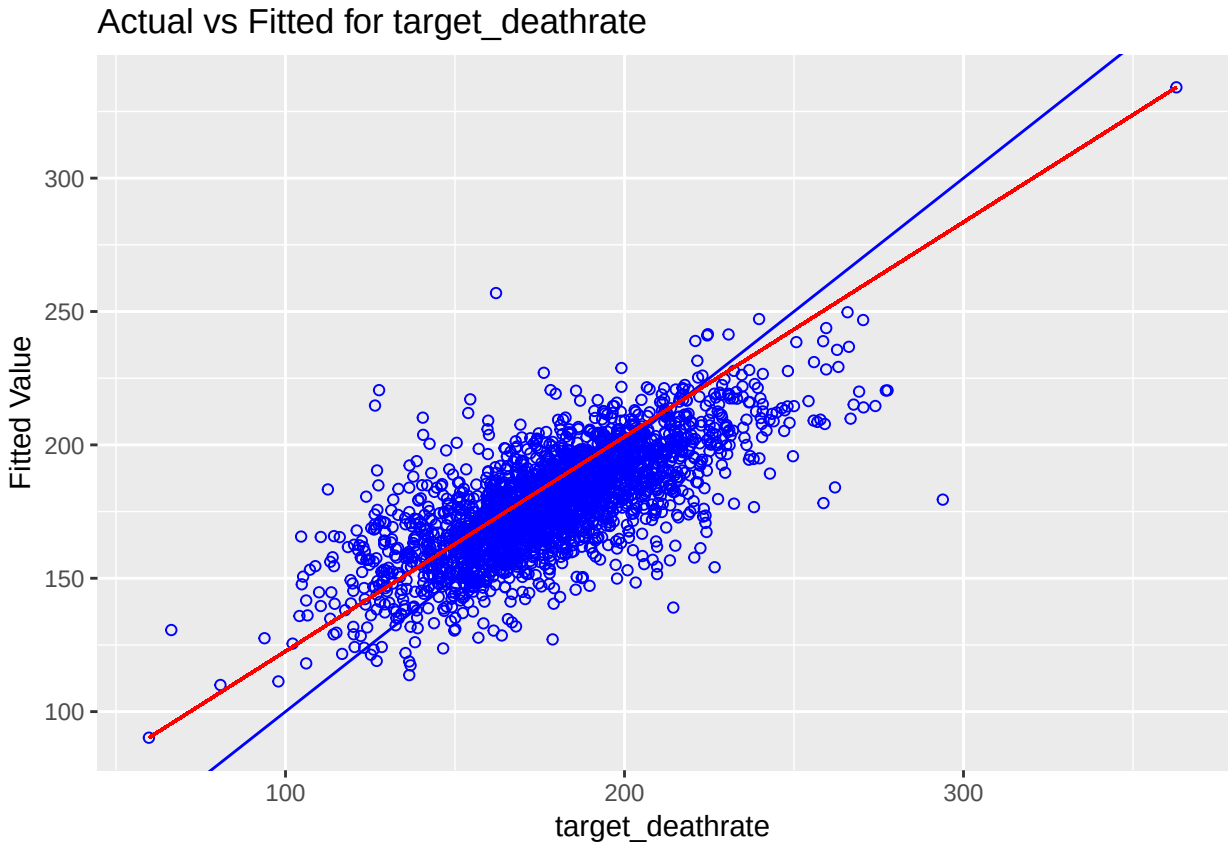
```
##       RMSE       R2
## 1 20.12843 0.475912
```

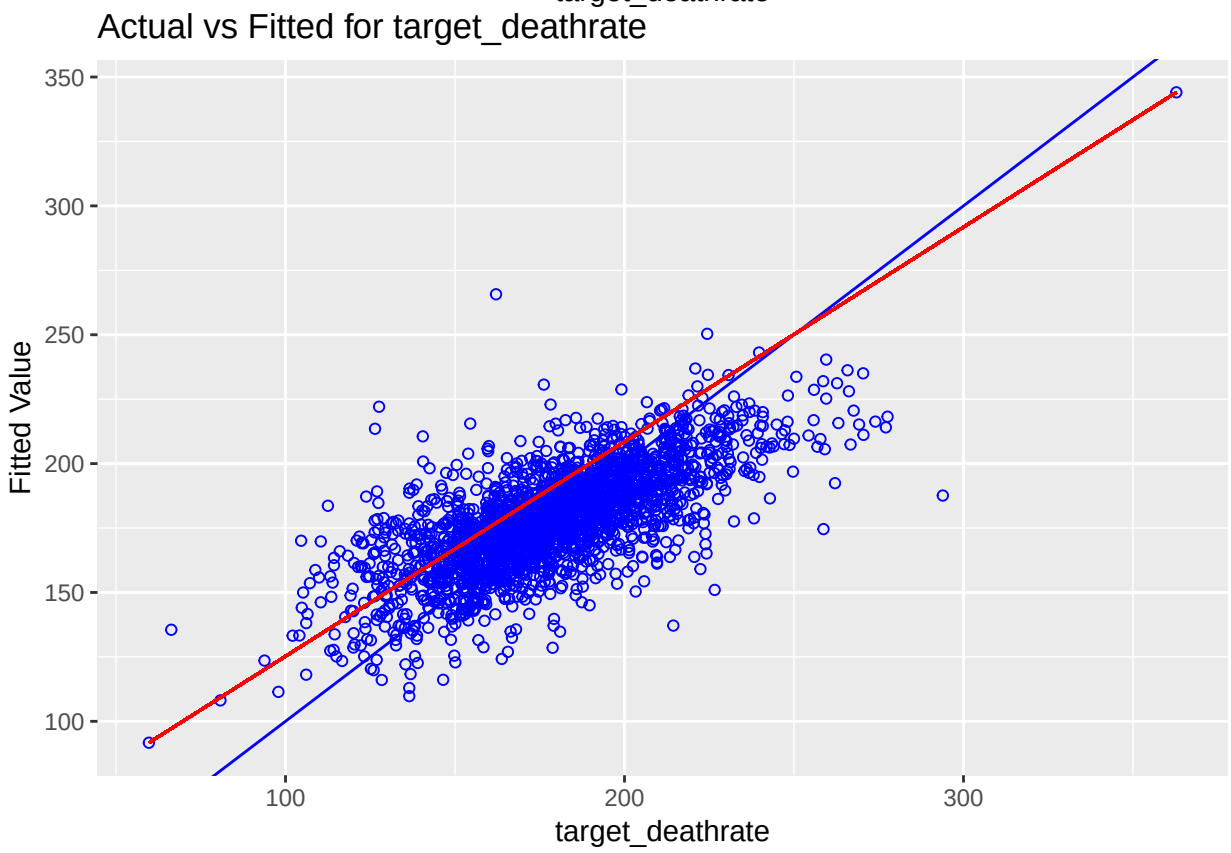
```
## [1] "Model3 performance -  after removing high VIF variables"
```

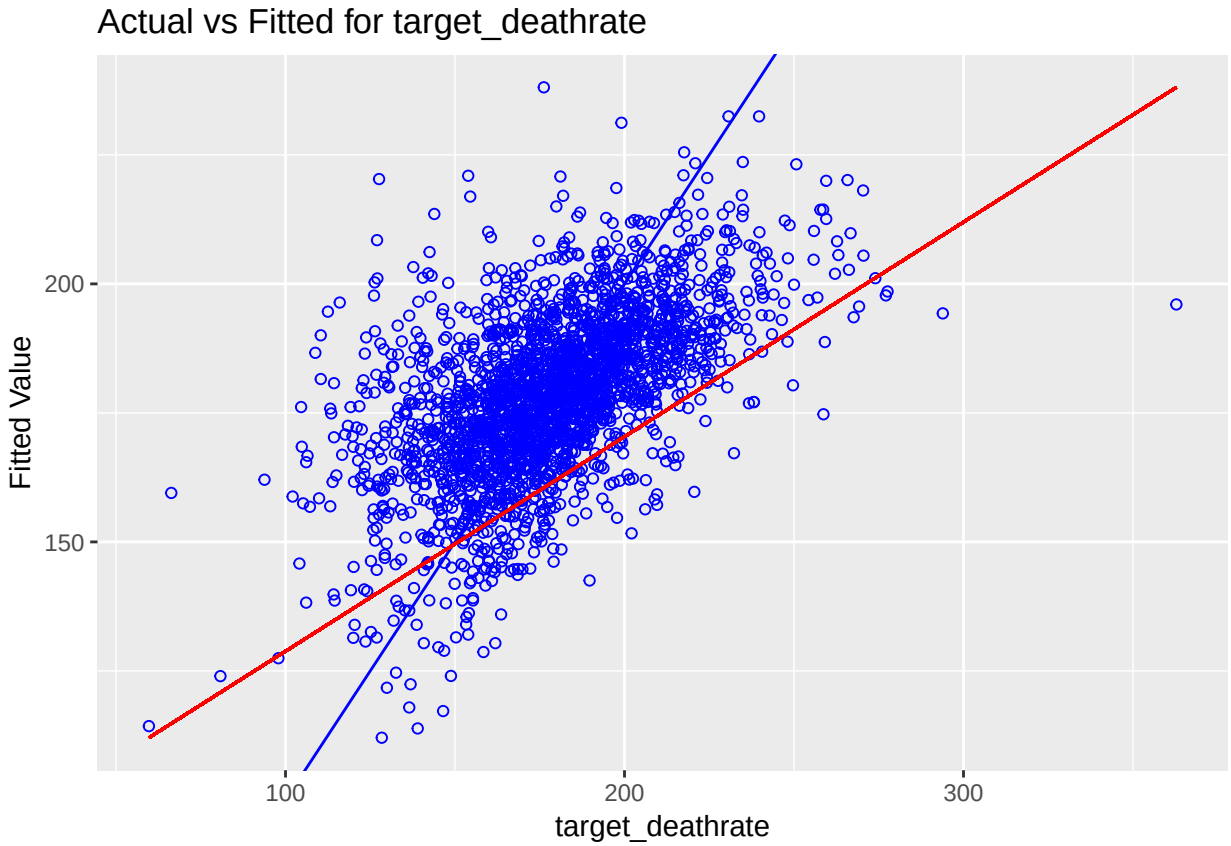
```
##      RMSE        R2
## 1 23.1221 0.3092514
```

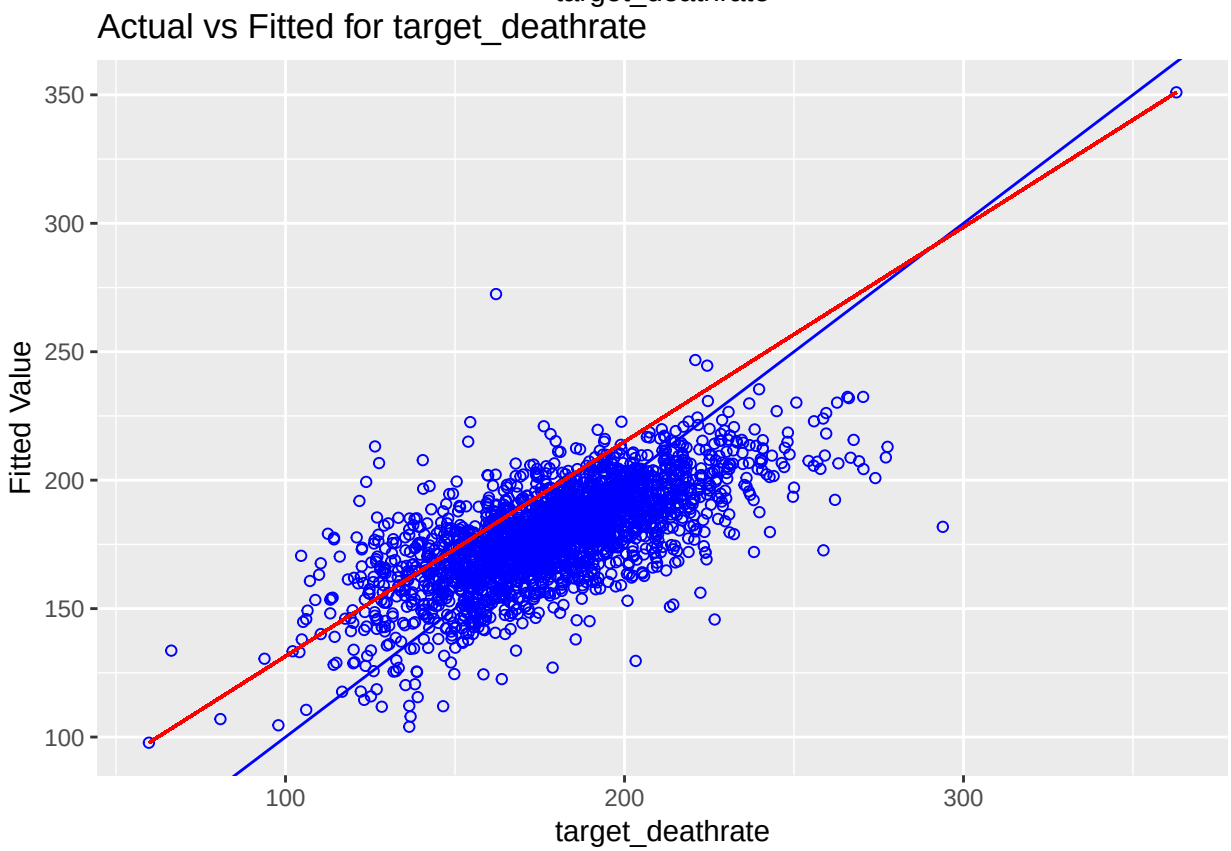
```
## [1] "Model4 performance -  Choosing variables based on adjr2"
```

```
##       RMSE        R2
## 1 20.76644 0.4422523
```

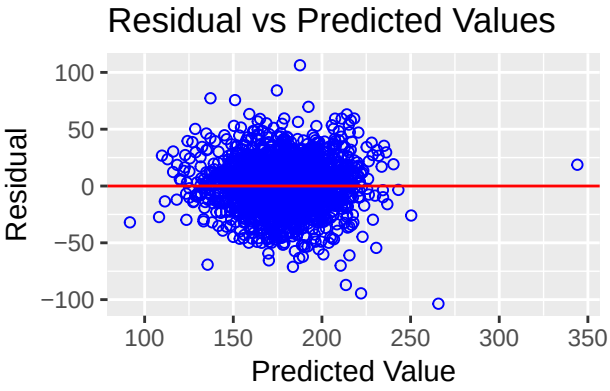

Actual vs Fitted for target_deathrate
Fitted Value
300
250
200
150
100
100
200
300
target_deathrate

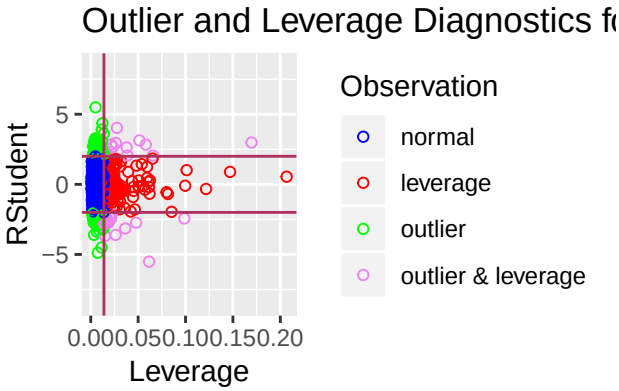

Actual vs Fitted for target_deathrate
Fitted Value
350
300
250
200
150
100
100
200
300
target_deathrate

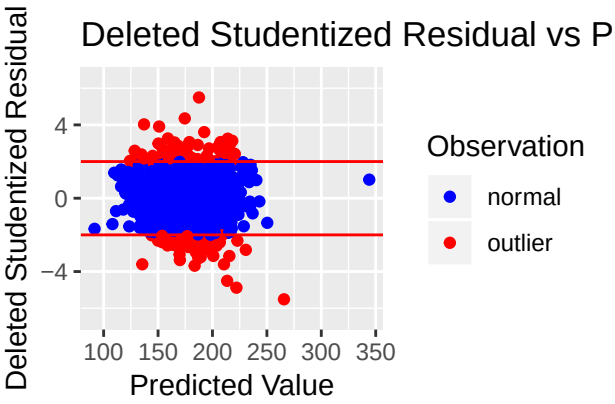

Actual vs Fitted for target_deathrate
Fitted Value
200
150
100
200
300
target_deathrate

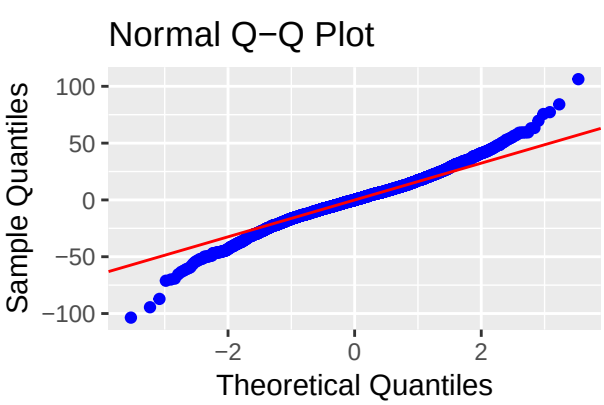

Actual vs Fitted for target_deathrate
Fitted Value
350
300
250
200
150
100
100
200
300
target_deathrate

page 1 of 3

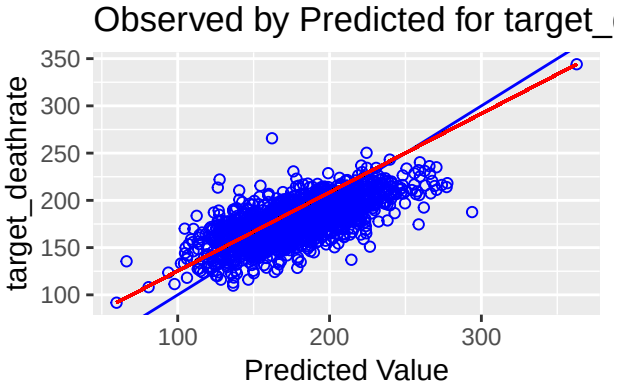


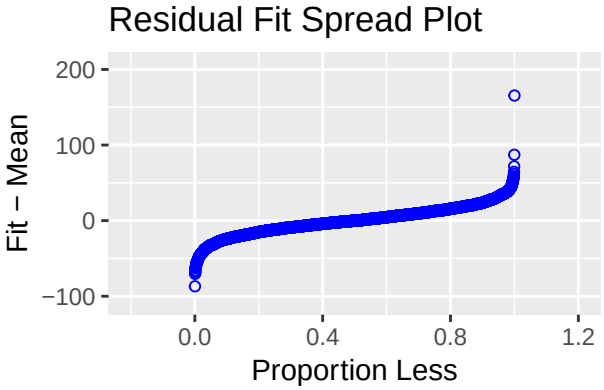


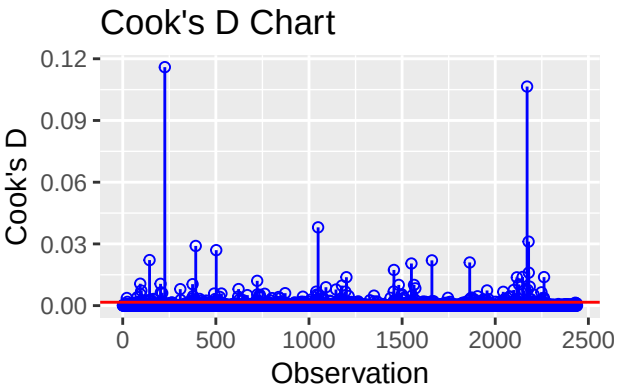


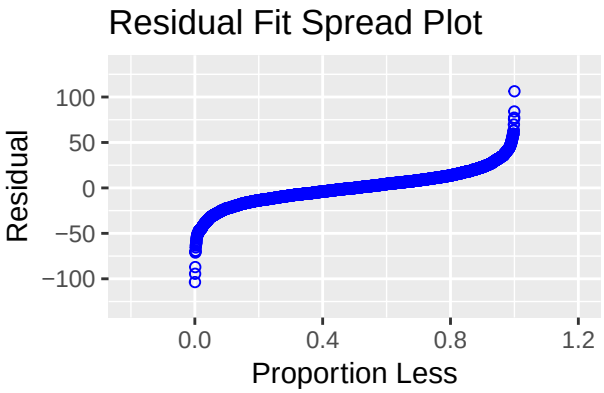


page 2 of 3

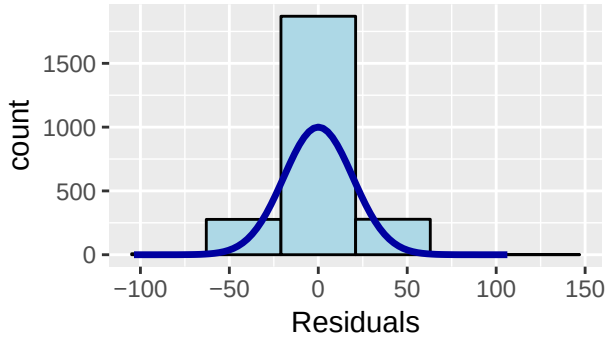








Residual Histogram



Residual Box Plot