Regression_analysis_cancerData

Cancer Data Analysis:

1. Checking for highly correlated variables
2. Checking for Multi collinearity
3. Plots in Olsrr toolbox
4. Checking for linear vs non linear using Residual plot

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(olsrr)
```

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

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

```
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)
```

```r
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(
```

```r
  RMSE = RMSE(predictions, test.data$target_deathrate),
  R2 = R2(predictions, test.data$target_deathrate)
)



#print("*****")
#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)
)
```

```r
print("Model1 performance - without removing correlated variables")
```

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

```r
print(performance1)
```

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

```r
print("Model2 performance -  after removing highly correlated variables")
```

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

```r
print(performance2)
```

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

```r
print("Model3 performance -  after removing variables with high VIF")
```

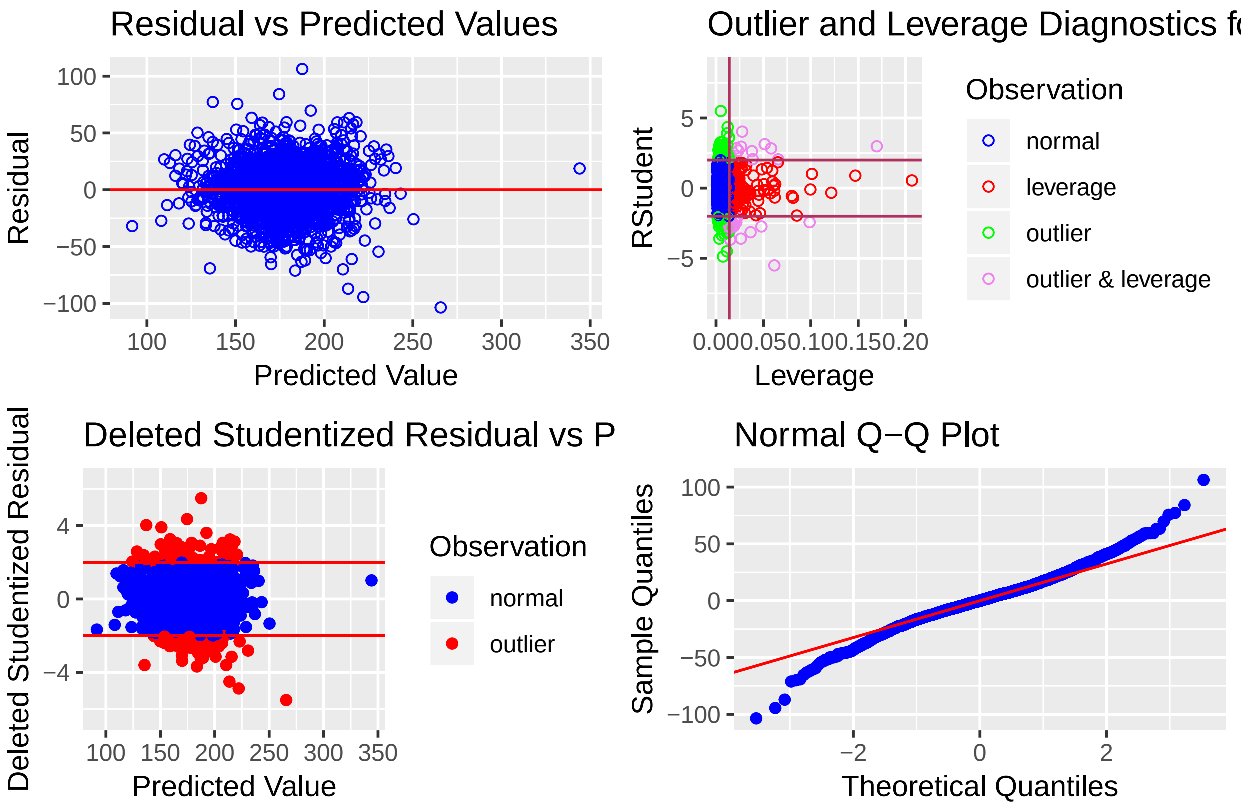
```
## [1] "Model3 performance -  after removing variables with high VIF"
```

```r
print(performance3)
```

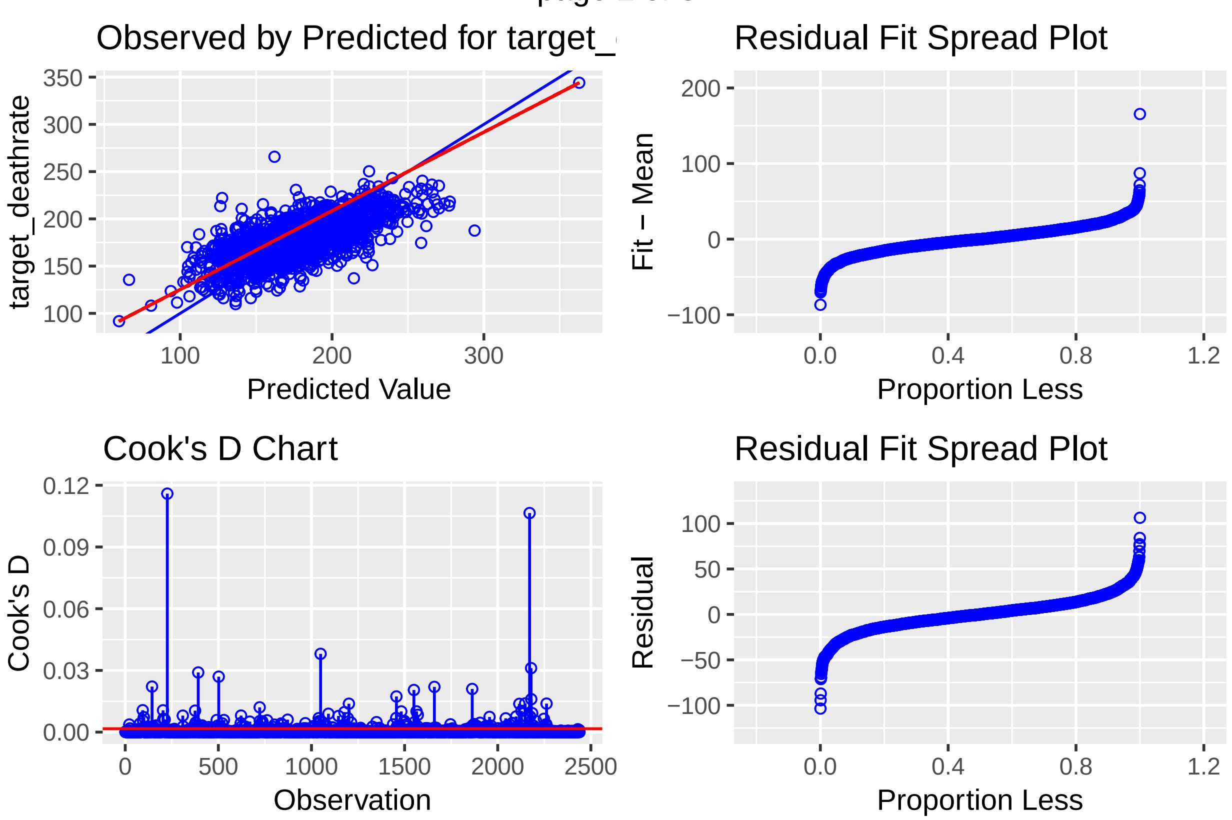
```
##      RMSE        R2
## 1 23.1221 0.3092514
```

```r
print(ols_plot_diagnostics(model2))
```

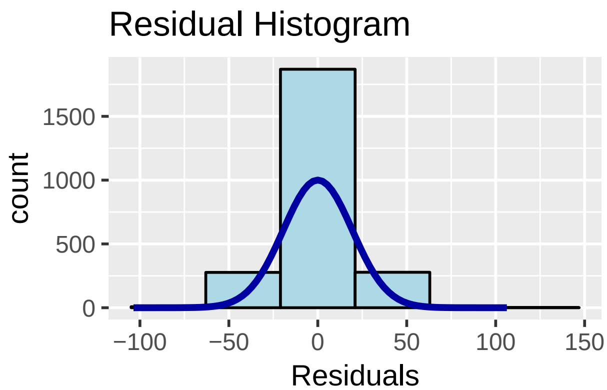
page 1 of 3

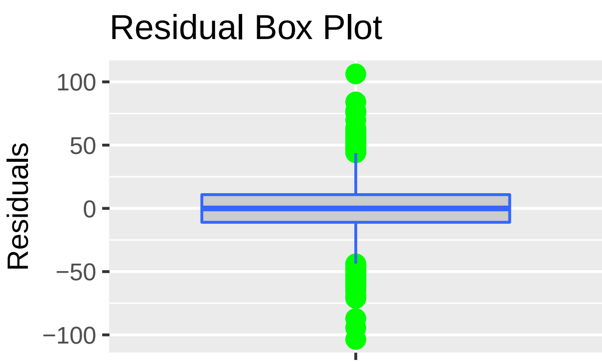


page 2 of 3



page 3 of 3






```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```