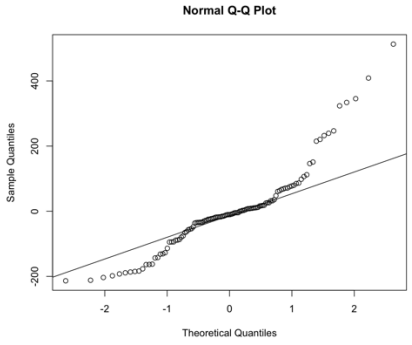
```
library(car)
library(carData)
library(zoo)
library(MASS)
library(lmtest)
library(boot)
library(fmsb)
library(leaps)
library(caret)
library(lmridge)

# Question 4:
# Does the number of vaccines have the same impact on infected cases
# with death cases before and after when mask mandate was lifted in
Indiana?

# Reading Data
covid<-read.csv("~/Desktop/covid.csv")

# Getting predictive factors:
data<-covid[c('TotalVac', 'death', 'Mask')]

model<-lm(death~TotalVac+TotalVac*factor(Mask), data)

# Checking assumptions:
# 1. Checking for constant variance
bptest(model) # Result showing for non-constant variance
```

```
        studentized Breusch-Pagan test

data:  model
BP = 19.313, df = 3, p-value = 0.0002355
```

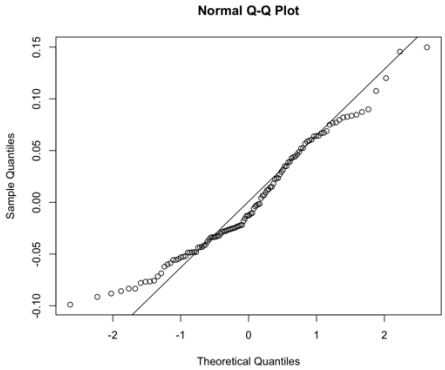
```
# 2. Checking for normality
shapiro.test(residuals(model))
qqnorm(residuals(model))
qqline(residuals(model)) # Violation of normality assumption
```

```
        Shapiro-Wilk normality test

data:  residuals(model)
W = 0.89634, p-value = 1.89e-07
```



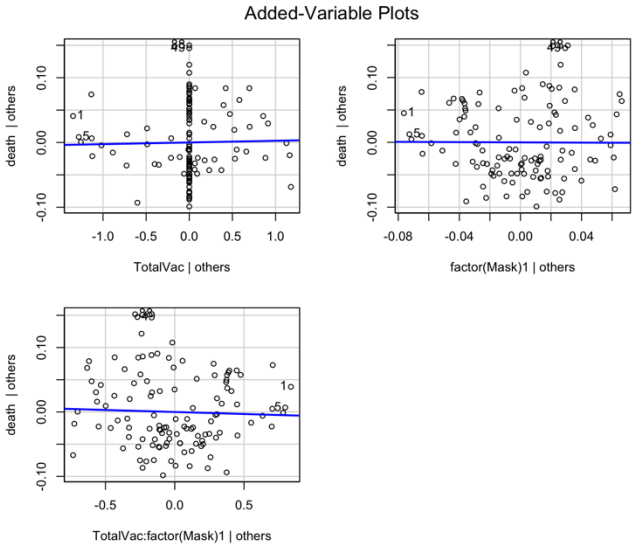
```
# Transformation
# On X:
data$TotalVac<-log(data$TotalVac)
newModel<-lm(death~TotalVac+TotalVac*factor(Mask), data)
# Box-cox transformation on Y
bcmle<-boxcox(newModel, lambda = seq(-3,3,by=0.1))
lambda<-bcmle$x[which.max(bcmle$y)]
data$death<-data$death^lambda
newModel<-lm(death~TotalVac+TotalVac*factor(Mask), data) # new OLS model

# Rechecking non-constant variance
bptest(newModel)

# Rechecking normality
shapiro.test(residuals(newModel))
qqnorm(residuals(newModel))
qqline(residuals(newModel)) # Improvement
```

```
        studentized Breusch-Pagan test

data:  newModel
BP = 15.613, df = 3, p-value = 0.001361
```



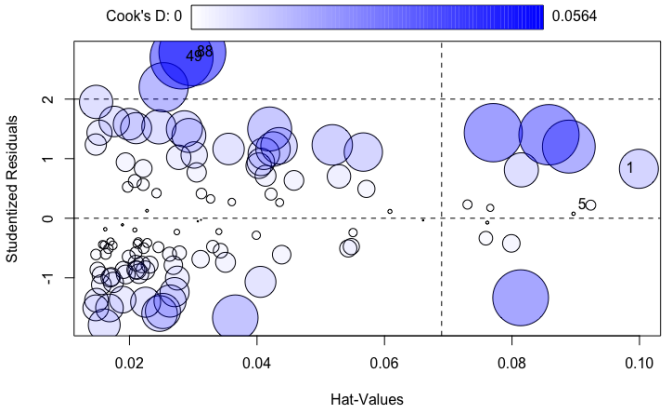

```
summary(newModel)
> summary(newModel)

Call:
lm(formula = death ~ TotalVac + TotalVac * factor(Mask), data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.09887 -0.04228 -0.01266  0.04392  0.14983

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               0.315235   0.117902   2.674  0.00862 **
TotalVac                  0.002616   0.011738   0.223  0.82406
factor(Mask)1            -0.009388   0.158832  -0.059  0.95297
TotalVac:factor(Mask)1 -0.006298   0.014875  -0.423  0.67282
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0562 on 112 degrees of freedom
Multiple R-squared:  0.3292,    Adjusted R-squared:  0.3113
F-statistic: 18.32 on 3 and 112 DF,  p-value: 9.64e-10
```

```
# Advanced diagnostic measurement
# 1. Checking marginal effects
avPlots(newModel) # Don't have much added-on effect
```




```
# 2. Checking for X and Y outliers
h<-2*4/116
hii<-lm.influence(newModel)$hat
length(which(hii[] > h)) # X Has 13 outliers
 > length(which(hii[] > h)) # X Has 13 outliers
 [1] 13
BonfCV<-qt(1-0.05/(2*116), 116-1-4) # Bonferroni critical value
rstudent<-rstudent(newModel)
length(which(rstudent[] > BonfCV)) # Y has no outlier
> length(which(rstudent[] > BonfCV)) # Y has no outlier
[1] 0

# 3. Checking for influential points
dfbetas<-dfbetas(newModel)
sum(dfbetas[which(abs(dfbetas[, 2]) > 1 & abs(dfbetas[, 3])> 1 &
                    abs(dfbetas[, 4]) > 1)]) # DFBETAS No influential point
> sum(dfbetas[which(abs(dfbetas[, 2]) > 1 & abs(dfbetas[, 3])> 1 &
+                 abs(dfbetas[, 4]) > 1)]) # DFBETAS No influential point
 [1] 0
dff<-dffits(newModel)
length(dff[dff > 1]) # DFFITS No influential point
 > length(dff[dff > 1]) # DFFITS No influential point
 [1] 0
CookDis<-cooks.distance(newModel)
minor<-qf(0.2, df1 = 4, df2 = 116 - 4)
major<-qf(0.5, df1 = 4, df2 = 116 - 4) # Two threholds
MinInf<-sum(CookDis > minor & CookDis > major)
MajInf<-sum(CookDis > major) # Cook's distance No influential point
```

```
influencePlot(newModel) # Rechecking outliers and influential points
```




```
# Multicollinearity remedial: Ridge Regression
RModel<-lmridge(death~TotalVac+TotalVac*factor(Mask), data,
K=seq(0,1,0.02))
# plot(RModel)
vif(RModel) # 0.06
```

	TotalVac	factor(Mask)1	TotalVac:factor(Mask)1
k=0	5.80334	224.74425	268.97498
k=0.02	2.11490	2.56865	2.66031
k=0.04	1.87480	1.14226	0.99427
k=0.06	1.68544	0.80927	0.63227
k=0.08	1.52610	0.66171	0.48709

```
RModel<-lmridge(death~TotalVac+TotalVac*factor(Mask), data, K=0.06)
summary(RModel)
```

```
Call:
lmridge.default(formula = death ~ TotalVac + TotalVac * factor(Mask),
    data = data, K = 0.06)


Coefficients: for Ridge parameter K= 0.06
                     Estimate Estimate (Sc) StdErr (Sc) t-value (Sc) Pr(>|t|)
Intercept              0.3492        1.8637      0.8494       2.1943   0.0303 *
TotalVac              -0.0009       -0.0104      0.0724      -0.1441   0.8857
factor(Mask)1         -0.0379       -0.2010      0.0502      -4.0081   0.0001 ***
TotalVac:factor(Mask)1 -0.0032       -0.1961      0.0443      -4.4234   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Ridge Summary
       R2     adj-R2   DF ridge          F        AIC        BIC
   0.31040    0.29820    1.85010   18.62174 -668.11663 -111.60575
Ridge minimum MSE= 0.07193067 at K= 0.06
P-value for F-test ( 1.8501 , 113.9623 ) = 2.272328e-07
-----------------------------------------------------------------
```

```
# Bootstrapping
boot.ridgecoef <- function(data, indices, maxit=100) {
  data <- data[indices,]
  colnames(data)<-c('x1', 'y', 'x2')
  mod <- lmridge(y~x1+x1*factor(x2), data=data, K=0.06)
  return(coef(mod))
}
RModel<-boot(data = data, statistic = boot.ridgecoef, R=100, maxit=100)
```

```
boot.ci(RModel, index = 2, type="perc")# 95% CI of X1
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = RModel, type = "perc", index = 2)

Intervals :
Level     Percentile
95%   (-0.0120,  0.0117 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
boot.ci(RModel, index = 3, type="perc")# 95% CI of X2
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = RModel, type = "perc", index = 3)

Intervals :
Level     Percentile
95%   (-0.0580, -0.0188 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
boot.ci(RModel, index = 4, type="perc")# 95% CI of interaction factor
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = RModel, type = "perc", index = 4)

Intervals :
Level     Percentile
95%   (-0.0043, -0.0020 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
```

```
#K-Folder
set.seed(123)
train.control<-trainControl(method = 'cv', number = 5)
step.model1<-train(death~TotalVac+TotalVac*factor(Mask), data = data,
                   method="leapBackward", tuneGrid = data.frame(nvmax =
4),
                   trControl = train.control)
step.model1$results # To check the value of RMSE
```

```
> step.model1$results
  nvmax       RMSE  Rsquared        MAE      RMSESD RsquaredSD       MAESD
1     4 0.05637614 0.3503281 0.04750515 0.008679268  0.1750909 0.006109299
```