

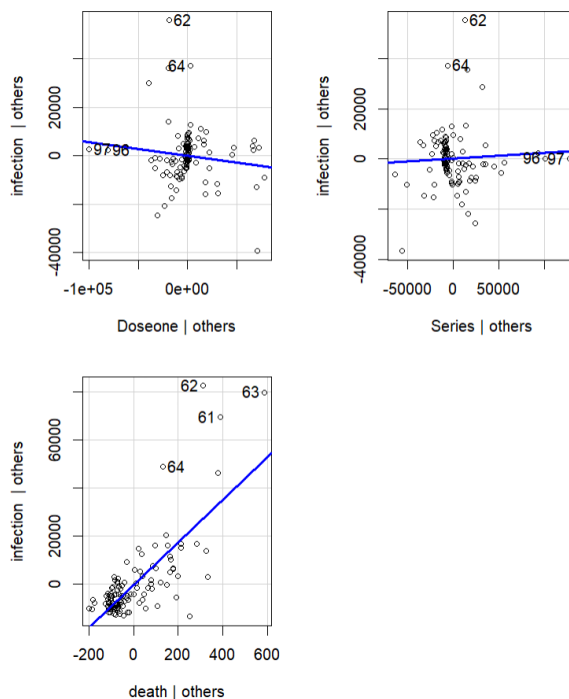
```

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

covid <- read.csv("C:/Users/74406/Desktop/covid_project/covid.csv",
stringsAsFactors=TRUE)
model<-lm(infection~Doseone+Series+death, covid)
avPlots(model) #added-variable plots

```

Added-Variable Plots



```

new<-subset(covid, select = -c(TotalVac, Booster,
Mask)) #delete totalvac, booster and mask column
plot(new) #plot the correlation plot

```



```
#shapiro test
```

```
res<-residuals(model)
```

```
shapiro.test(res)
```

Shapiro-wilk normality test

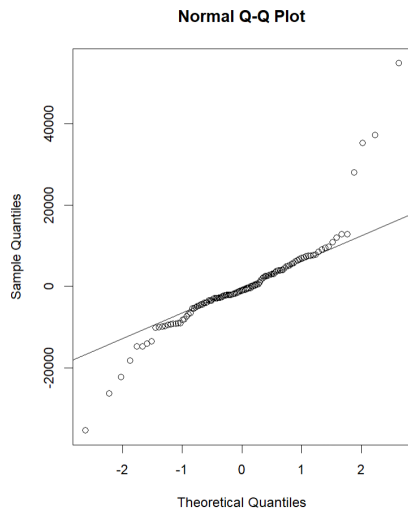
```
data: res
```

```
W = 0.85227, p-value = 2.153e-09
```

```
#bp test and qqplot
```

```
qqnorm(res)
```

```
qqline(res)
```



```
bptest(model)
```

studentized Breusch-Pagan test

```
data: model
```

```
BP = 35.205, df = 3, p-value = 1.103e-07
```

#logx trans

```
new$Doseone<-log(new$Doseone)
```

```
new$Series<-log(new$Series)
```

```
new$death<-log(new$death)
```

```
modelNew<-lm(infection~Doseone+Series+death, new)
```

#box-cox trans

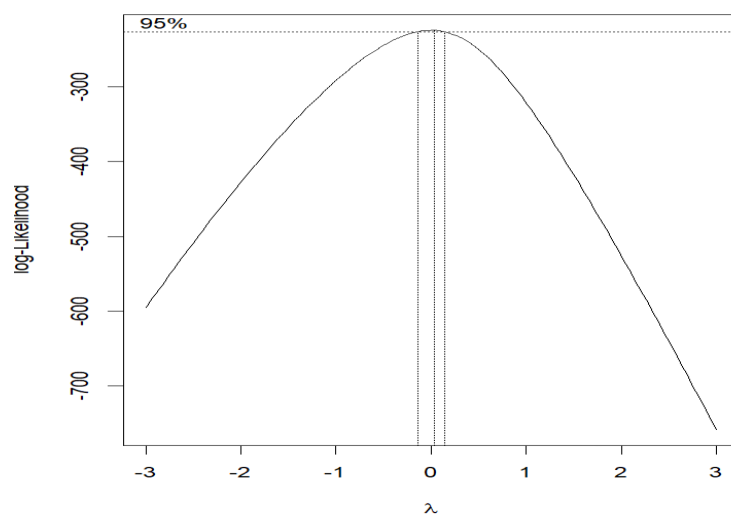
```
bc<-boxcox(modelNew, lambda = seq(-3,3,by=0.1))
```

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

```
new$newY<-new$infection^lambda
```

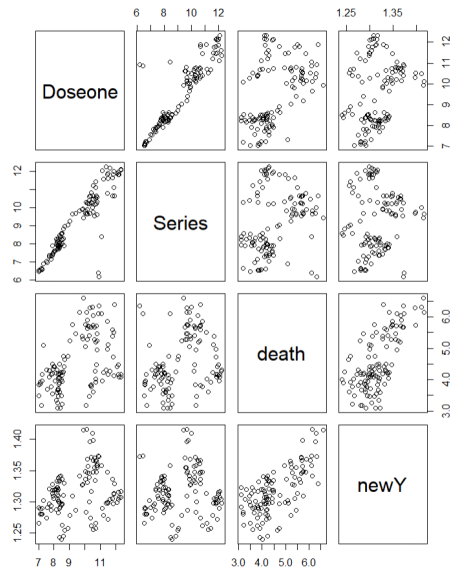
```
modelNew<-lm(newY~Doseone+Series+death, new)
```

```
new<-new[, -3]
```



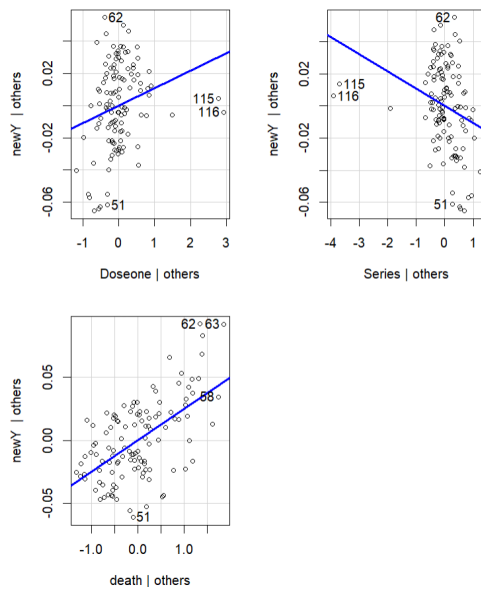
#diagnostic

```
plot(new)
```



```
avPlots(modelNew)
```

Added-Variable Plots

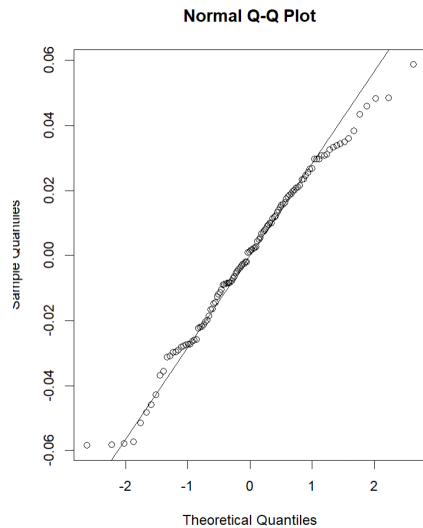


```
res<-residuals(modelNew)
shapiro.test(res)
```

shapiro-wilk normality test

```
data:  res
W = 0.98715, p-value = 0.3407
```

```
qqnorm(res)
qqline(res)
```



```
bptest(modelNew)
```

studentized Breusch-Pagan test

```
data: modelNew
```

```
BP = 7.5327, df = 3, p-value = 0.05672
```

```
#advanced diagnostic
```

```
standard = 2*(4/116)
```

```
infl<-lm.influence(modelNew)$hat
```

```
length(which(infl[] > standard)) #X influential pts
```

```
[1] 3
```

```
#dfbetas
```

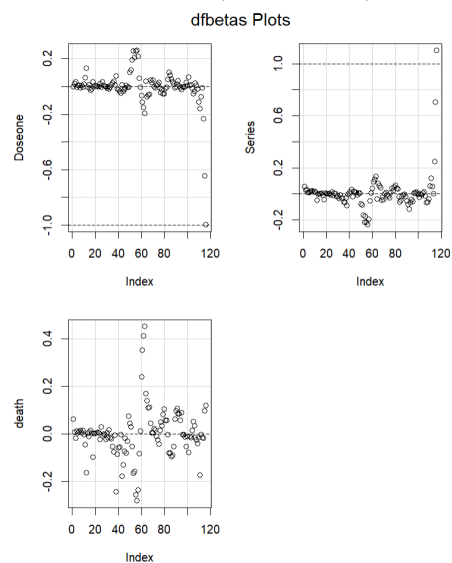
```
d<-dfbetas(modelNew)
```

```
dfbetasPlots(modelNew)
```

```
d[which(abs(d[,2])>1 & abs(d[,3])>1 & abs(d[,4]) > 1)]
```

```
0
```

```
dfbetasPlots(modelNew)
```



```
#dffits
```

```
dff<-dffits(modelNew)
```

```
length(dff[dff>1])
```

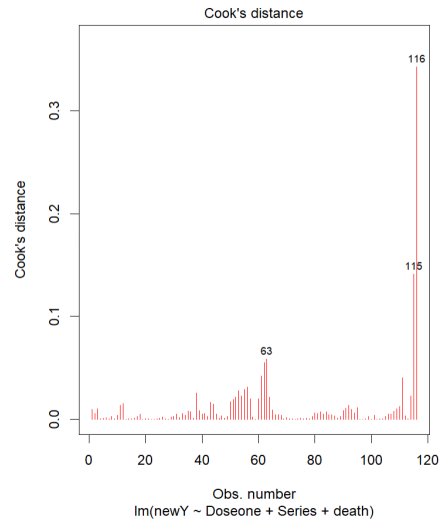
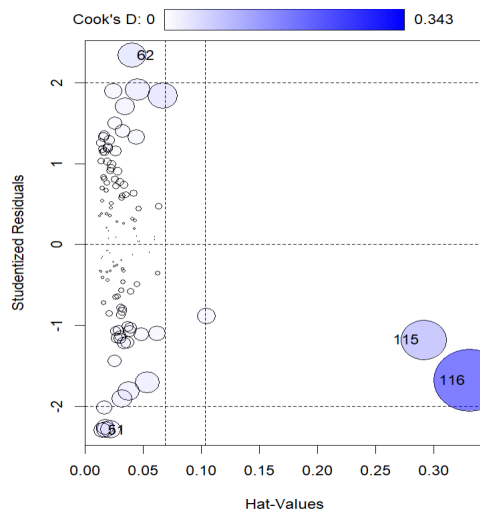
```
0
```

```
influencePlot(modelNew)
```

	StudRes	Hat	CookD
51	-2.290805	0.01498624	0.01923091
62	2.339097	0.04024583	0.05515646
115	-1.174938	0.29115063	0.14127359
116	-1.678955	0.33084514	0.34286239

#cooks distance

```
plot(lm(newY~Doseone+Series+death, new), pch = 18,
col="red", which = c(4))
```



```
qf(0.2,4,112-4)
```

```
qf(0.5,4,112-4)
```

```
[1] 0.4115393
```

```
> qf(0.5,4,112-4)
```

```
[1] 0.8444662
```

#VIF

```
VIF(lm(Doseone~Series+death, new))
```

```
VIF(lm(Series~Doseone+death, new))
```

```
VIF(lm(death~Doseone+Series, new))
```

```
> VIF(lm(Doseone~Series+death, new))
```

```
[1] 7.278238
```

```
> VIF(lm(Series~Doseone+death, new))
```

```
[1] 6.08119
```

```
> VIF(lm(death~Doseone+Series, new))
```

```
[1] 1.530908
```

#Robust

```
modelR<- rlm(newY~Doseone+Series+death, data = new, psi =
```

```
psi.bisquare)
```

```
summary(modelR)
```

```
Call: rlm(formula = newY ~ Doseone + Series + death, data = new, psi = psi.b
isquare)
Residuals:
      Min       1Q   Median       3Q      Max
-0.0593507 -0.0196333  0.0006057  0.0189734  0.0586918
```

```
Coefficients:
      Value Std. Error t value
(Intercept)  1.2014   0.0173   69.5361
Doseone       0.0097   0.0045    2.1478
Series      -0.0098   0.0037   -2.6316
death         0.0248   0.0035    7.1164
```

```
Residual standard error: 0.02852 on 112 degrees of freedom
```

#bootstrapping on Robust

```
boot.R<-function(data, indices, maxit = 100){
  data<-data[indices,]
  mod<-rlm(newY~Doseone+Series+death, data = data,
maxit=maxit)
  return(coefficients(mod))
}
modelBR<-boot(data=new, statistic = boot.R, R=100,
maxit=100)
boot.ci(modelBR, index = 2, type="perc")
boot.ci(modelBR, index = 3, type="perc")
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates
```

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

```
Intervals :
Level      Percentile
95%    ( 0.0011,  0.0340 )
Calculations and Intervals on Original scale
Some percentile intervals may be unstable
> boot.ci(modelBR, index = 3, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates
```

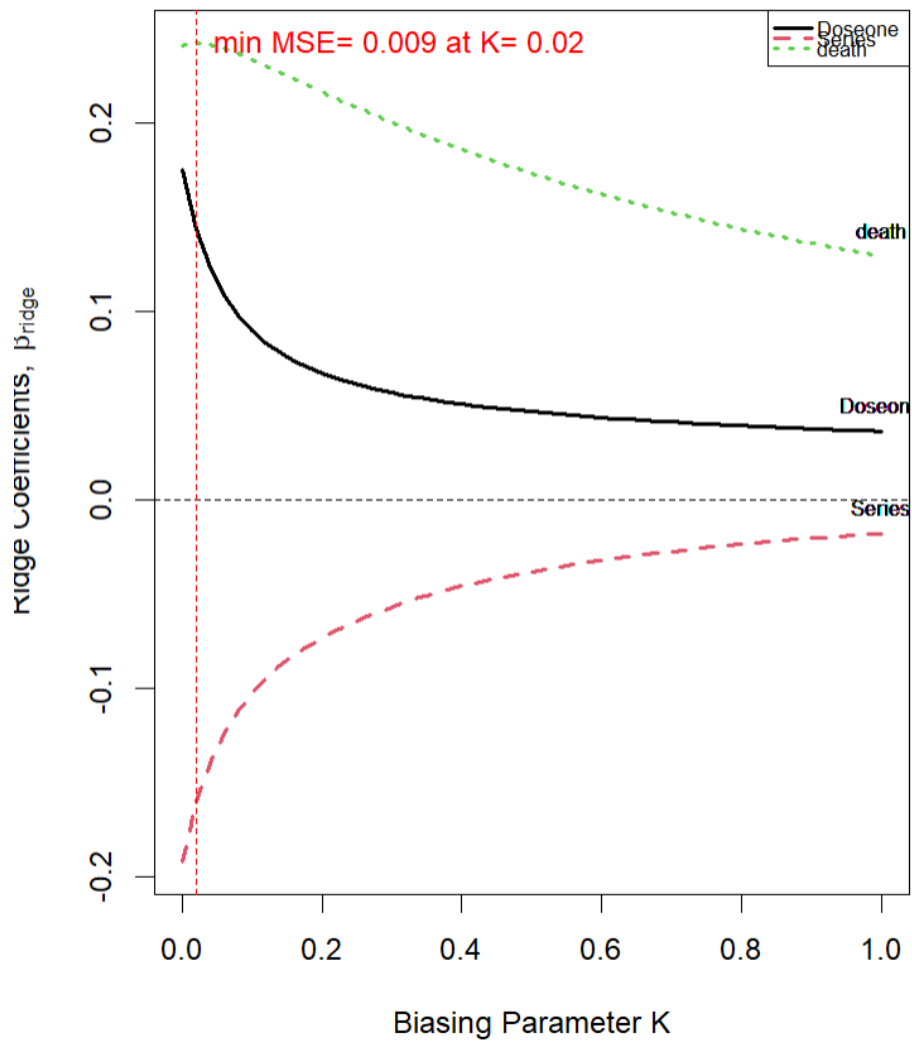
```
CALL :
boot.ci(boot.out = modelBR, type = "perc", index = 3)
```

```
Intervals :
Level      Percentile
95%    (-0.0317, -0.0040 )
Calculations and Intervals on Original scale
Some percentile intervals may be unstable
```

#Ridge

```
rid<-lmridge(newY~Doseone+Series+death, data = new, K =
seq(0,1,0.02))
plot(rid)
```

Ridge Trace Plot



```
vif(rid)
```

```
      Doseone  Series  death
k=0      7.27824 6.08119 1.53091
k=0.02   4.64893 3.95529 1.31858
k=0.04   3.25399 2.82313 1.18529
k=0.06   2.42526 2.14714 1.08994
k=0.08   1.89241 1.70981 1.01570
k=0.1    1.52914 1.40948 0.95459
k=0.12   1.27003 1.19346 0.90238
k=0.14   1.07843 1.03223 0.85661
k=0.16   0.93252 0.90820 0.81575
k=0.18   0.81864 0.81034 0.77880
k=0.2    0.72789 0.73146 0.74506
k=0.22   0.65426 0.66670 0.71401
k=0.24   0.59359 0.61269 0.68528
k=0.26   0.54292 0.56700 0.65856
```

```
summary(lmridge(newY~Doseone+Series+death,data = new, K = 0.14))
```



```
Call:
lmridge.default(formula = newY ~ Doseone + Series + death, data = new,
  K = 0.14)
```

Coefficients: for Ridge parameter K= 0.14

	Estimate	Estimate (Sc)	StdErr (Sc)	t-value (Sc)	Pr(> t)
Intercept	1.2083	0.3492	0.3767	0.9270	0.3559
Doseone	0.0048	0.0783	0.0274	2.8541	0.0051 **
Series	-0.0049	-0.0882	0.0268	-3.2848	0.0014 **
death	0.0235	0.2268	0.0245	9.2744	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ridge Summary

	R2	adj-R2	DF ridge	F	AIC	BIC
	0.37610	0.36500	2.13782	38.13064	-841.32571	-284.02256

Ridge minimum MSE= 0.02239581 at K= 0.14
P-value for F-test (2.13782 , 113.4468) = 5.919348e-14

#bootstrapping on Ridge

```
boot.Rid<-function(data, indices, maxit = 100){
  data<-data[indices,]
  mod<-lmridge(newY~Doseone+Series+death, data = data,
maxit=maxit, K = 0.14)
  return(coefficients(mod))
}
modelrid<-boot(data=new, statistic = boot.Rid, R=100,
maxit=100)
boot.ci(modelrid, index = 2, type="perc")
boot.ci(modelrid, index = 3, type="perc")
> boot.ci(modelrid, index = 2, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates
```

CALL :

```
boot.ci(boot.out = modelrid, type = "perc", index = 2)
```

Intervals :

```
Level Percentile
95% ( 0.0019, 0.0074 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
> boot.ci(modelrid, index = 3, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates
```

CALL :

```
boot.ci(boot.out = modelrid, type = "perc", index = 3)
```

Intervals :

```
Level Percentile
95% (-0.0069, -0.0026 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
```

#k-fold for prediction

```
set.seed(123)
train.control<-trainControl(method = 'cv', number = 5)
step.modell<-train(newY~Doseone+Series+death, data = new,
method="leapBackward", tuneGrid = data.frame(nvmax = 4),
trControl = train.control)
```

```

step.modell$results
      nvmax      RMSE Rsquared      MAE      RMSESD RsquaredSD
1       4 0.02725769 0.4864292 0.02210709 0.002422338 0.181374
      MAESD
1 0.002059341

#without advanced method
reduced<-lm(newY~Doseone+death, new)
full<-lm(newY~Doseone+Series+death, new)
MSR<-(sum(reduced$residuals^2) - sum(full$residuals^2))/
(reduced$df.residual - full$df.residual)
MSE<-sum(full$residuals^2) / full$df.residual
FS = MSR/MSE
qf(0.95, reduced$df.residual-full$df.residual,
full$df.residual)
p = 1-pf(FS,reduced$df.residual-full$df.residual,
full$df.residual)
FS
p
summary(full)
> qf(0.95, reduced$df.residual-full$df.residual, full$df.residual)
[1] 3.925834
> p = 1-pf(FS,reduced$df.residual-full$df.residual, full$df.residual)
> FS
[1] 8.854264
> p
[1] 0.003582887
> summary(full)

Call:
lm(formula = newY ~ Doseone + Series + death, data = new)

Residuals:
      Min       1Q   Median       3Q      Max
-0.058280 -0.018959  0.001353  0.019180  0.058684

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.197918   0.016555  72.360 < 2e-16 ***
Doseone      0.010766   0.004326   2.489  0.01429 *
Series      -0.010661   0.003583  -2.976  0.00358 **
death       0.024937   0.003344   7.458 2.01e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02612 on 112 degrees of freedom
Multiple R-squared:  0.511,    Adjusted R-squared:  0.498
F-statistic: 39.02 on 3 and 112 DF,  p-value: < 2.2e-16

cv = qt(0.975, 112)
CI_lower = -.010661 - 0.003583*cv
CI_higher = -.010661 + 0.003583*cv
CI_higher
CI_lower
DCl = 0.010766 + 0.004326*cv
DCh = 0.010766 - 0.004326*cv
DCl
DCh

```

```

> CI_higher
[1] -0.003561745
> CI_lower
[1] -0.01776026
> DCI = 0.010766 + 0.004326*cv
> DCh = 0.010766 - 0.004326*cv
> DCI
[1] 0.01933741
> DCh
[1] 0.002194586

#Bootstrapping on OLS
boot.ols<-function(data, indices, maxit = 100){
  data<-data[indices,]
  data.mod1<-lm(newY~Doseone+Series+death, data = data,
maxit = 100)
  return(coef(data.mod1))
}
modelols<-boot(data = new, statistic = boot.ols, R = 100,
maxit = 100)
boot.ci(modelols, index = 2, type="perc")
boot.ci(modelols, index = 3, type="perc")
> boot.ci(modelols, index = 2, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

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

Intervals :
Level      Percentile
95%      ( 0.0033,  0.0314 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
> boot.ci(modelols, index = 3, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

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

Intervals :
Level      Percentile
95%      (-0.0293, -0.0052 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable

#WLS
wts<-1/fitted(lm(abs(residuals(full))~Doseone+Series+death
, new))^2
modelM<-lm(newY~Doseone+Series+death, weights = wts,data =
new)
summary(modelM)

```

```

Call:
lm(formula = newY ~ Doseone + Series + death, data = new, weights = wts)

Weighted Residuals:
    Min       1Q   Median       3Q      Max
-2.8481 -0.9096  0.1437  0.8997  2.3628

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.213278   0.015851  76.543 < 2e-16 ***
Doseone       0.012437   0.004337   2.868 0.004939 **
Series       -0.012719   0.003661  -3.474 0.000729 ***
death        0.022217   0.003334   6.664 1.04e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.22 on 112 degrees of freedom
Multiple R-squared:  0.485,    Adjusted R-squared:  0.4712
F-statistic: 35.16 on 3 and 112 DF,  p-value: 4.336e-16

#bootstrapping on WLS
boot.wls<-function(data, indices, maxit = 100){
  data<-data[indices,]
  data.mod1<-lm(newY~Doseone+Series+death, data = data,
maxit = maxit)

  wts<-1/fitted(lm(abs(residuals(full))~Doseone+Series+death
, data))^2
  data.mod2<-lm(newY~Doseone+Series+death, weights = wts,
data = data)
  return(coef(data.mod2))
}
modelwls<-boot(data = new, statistic = boot.wls, R = 100,
maxit = 100)
boot.ci(modelwls, index = 2, type="perc")
boot.ci(modelwls, index = 3, type="perc")
> boot.ci(modelwls, index = 2, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

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

Intervals :
Level      Percentile
95%      ( 0.0010,  0.0345 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
> boot.ci(modelwls, index = 3, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

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

Intervals :
Level      Percentile
95%      (-0.0302, -0.0021 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable

```