STAT656projectKM

Ken Marciel

11/13/2020

COVID-19 and Unemployment in the United States

Data preprocessing step 1: U.S. COVID-19 aggregate data

```
# read in data
covid19 = read.csv('C:\\Users\\keoka\\OneDrive - Texas A&M University\\Courses\\STAT_656\\Project\\Data'

# sort rows alphabetically by state
covid19 = covid19[order(covid19$state),]

# subset columns
covid19 = covid19[,c(1,2,3,6,8,11)]

# subset rows
remObs  = c("AS", "FSM","GU","MP","NYC","PR","PW","RMI","VI")
covid19 = covid19[!covid19$state %in% remObs,]
covid19 = covid19[covid19$submission_date > "01/31/2020",]
unique(covid19[,2]) # state should have 51 values
```

```
##  [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DC" "DE" "FL" "GA" "HI" "IA" "ID" "IL"
## [16] "IN" "KS" "KY" "LA" "MA" "MD" "ME" "MI" "MN" "MO" "MS" "MT" "NC" "ND" "NE"
## [31] "NH" "NJ" "NM" "NV" "NY" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT"
## [46] "VA" "VT" "WA" "WI" "WV" "WY"
```

Data preprocessing step 2: U.S. COVID-19 monthly data

```
# monthly totals

covidJan    = covid19[covid19$submission_date > "12/31/2019" &
                      covid19$submission_date < "02/01/2020",]
newCaseJan  = sum(covidJan[,4])
totCaseJan  = newCaseJan
newDeathJan = sum(covidJan[,6])
totDeathJan = newDeathJan

covidFeb    = covid19[covid19$submission_date > "01/31/2020" &
                      covid19$submission_date < "03/01/2020",]
newCaseFeb  = sum(covidFeb[,4])
totCaseFeb  = sum(newCaseFeb, totCaseJan)
newDeathFeb = sum(covidFeb[,6])
totDeathFeb = sum(newDeathFeb, totDeathJan)
```

```
covidMar    = covid19[covid19$submission_date > "02/29/2020" &
                      covid19$submission_date < "04/01/2020",]
newCaseMar  = sum(covidMar[,4])
totCaseMar  = sum(newCaseMar, totCaseFeb)
newDeathMar = sum(covidMar[,6])
totDeathMar = sum(newDeathMar, totDeathFeb)

covidApr    = covid19[covid19$submission_date > "03/31/2020" &
                      covid19$submission_date < "05/01/2020",]
newCaseApr  = sum(covidApr[,4])
totCaseApr  = sum(newCaseApr, totCaseMar)
newDeathApr = sum(covidApr[,6])
totDeathApr = sum(newDeathApr, totDeathMar)

covidMay    = covid19[covid19$submission_date > "04/30/2020" &
                      covid19$submission_date < "06/01/2020",]
newCaseMay  = sum(covidMay[,4])
totCaseMay  = sum(newCaseMay, totCaseApr)
newDeathMay = sum(covidMay[,6])
totDeathMay = sum(newDeathMay, totDeathApr)

covidJun    = covid19[covid19$submission_date > "05/31/2020" &
                      covid19$submission_date < "07/01/2020",]
newCaseJun  = sum(covidJun[,4])
totCaseJun  = sum(newCaseJun, totCaseMay)
newDeathJun = sum(covidJun[,6])
totDeathJun = sum(newDeathJun, totDeathMay)

covidJul    = covid19[covid19$submission_date > "06/30/2020" &
                      covid19$submission_date < "08/01/2020",]
newCaseJul  = sum(covidJul[,4])
totCaseJul  = sum(newCaseJul, totCaseJun)
newDeathJul = sum(covidJul[,6])
totDeathJul = sum(newDeathJul, totDeathJun)

covidAug    = covid19[covid19$submission_date > "07/31/2020" &
                      covid19$submission_date < "09/01/2020",]
newCaseAug  = sum(covidAug[,4])
totCaseAug  = sum(newCaseAug, totCaseJul)
newDeathAug = sum(covidAug[,6])
totDeathAug = sum(newDeathAug, totDeathJul)

covidSep    = covid19[covid19$submission_date > "08/31/2020" &
                      covid19$submission_date < "10/01/2020",]
newCaseSep  = sum(covidSep[,4])
totCaseSep  = sum(newCaseSep, totCaseAug)
newDeathSep = sum(covidSep[,6])
totDeathSep = sum(newDeathSep, totDeathAug)

covidOct    = covid19[covid19$submission_date > "09/30/2020" &
                      covid19$submission_date < "11/01/2020",]
newCaseOct  = sum(covidOct[,4])
totCaseOct  = sum(newCaseOct, totCaseSep)
```

```
newDeathOct = sum(covidOct[,6])
totDeathOct = sum(newDeathOct, totDeathSep)

newCase     = c(newCaseFeb, newCaseMar, newCaseApr, newCaseMay,
                newCaseJun, newCaseJul, newCaseAug, newCaseSep,
                newCaseOct)

totCase     = c(totCaseFeb, totCaseMar, totCaseApr, totCaseMay,
                totCaseJun, totCaseJul, totCaseAug, totCaseSep,
                totCaseOct)

newDeath    = c(newDeathFeb, newDeathMar, newDeathApr, newDeathMay,
                newDeathJun, newDeathJul, newDeathAug, newDeathSep,
                newDeathOct)

totDeath    = c(totDeathFeb, totDeathMar, totDeathApr, totDeathMay,
                totDeathJun, totDeathJul, totDeathAug, totDeathSep,
                totDeathOct)

month       = 1:9

covidMonthly  = cbind(month, newCase, totCase, newDeath, totDeath)
```

Data preprocessing step 3: U.S. unemployment monthly data

```
uRate = read.csv('C:\\Users\\keoka\\OneDrive - Texas A&M University\\Courses\\STAT_656\\Project\\Data\\
uRate = as.numeric(t(uRate[84,3:11]))
#colnames(uRate) = 'uRate'
```

Data preprocessing step 4: U.S. monthly COVID-19 and unemployment

```
# Combined data set
covidUnemployment  = data.frame(cbind(uRate, month, newCase, totCase, newDeath, totDeath))
str(covidUnemployment)
```

```
## 'data.frame':    9 obs. of  6 variables:
##  $ uRate   : num  3.5 4.4 14.7 13.3 11.1 10.2 8.4 7.9 6.9
##  $ month   : num  1 2 3 4 5 6 7 8 9
##  $ newCase : num  19 143625 754009 679064 823725 ...
##  $ totCase : num  19 143644 897653 1576717 2400442 ...
##  $ newDeath: num  1 2495 41941 38203 21389 ...
##  $ totDeath: num  1 2496 44437 82640 104029 ...
```

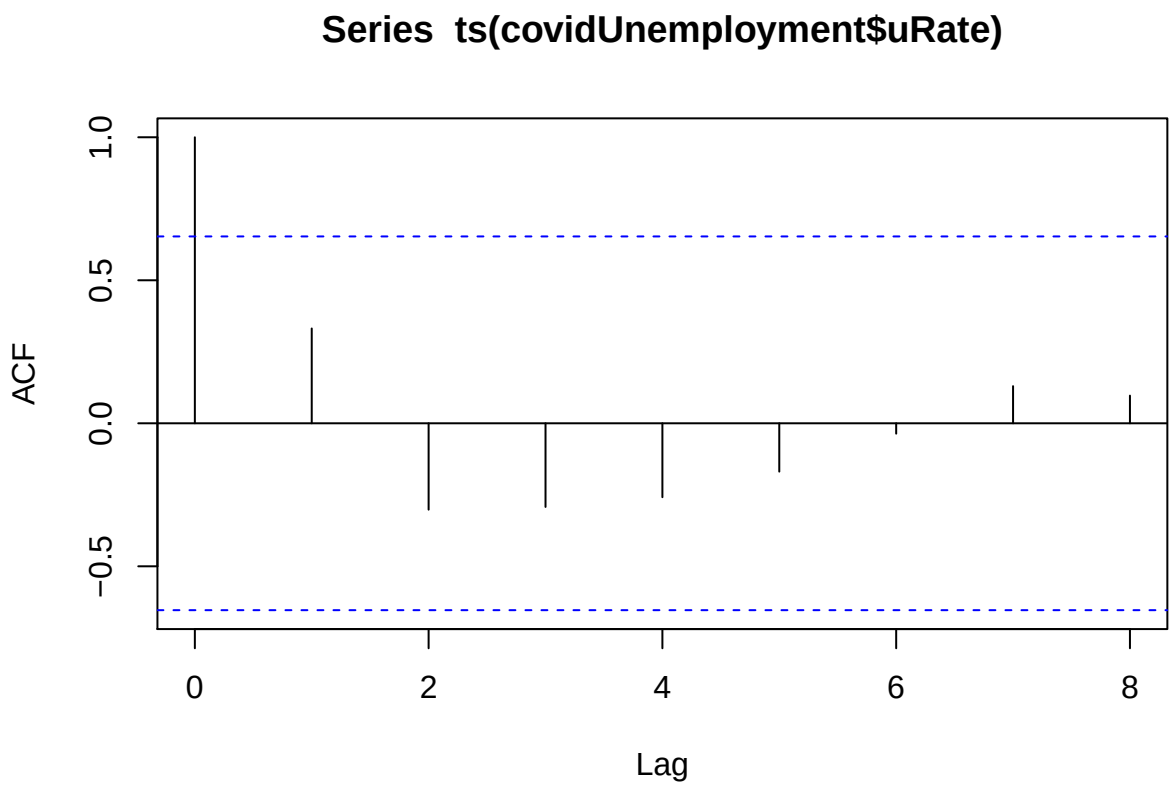
```
write.csv(covidUnemployment,'C:\\Users\\keoka\\OneDrive - Texas A&M University\\Courses\\STAT_656\\Proj
```

Time series: check for autocorrelation

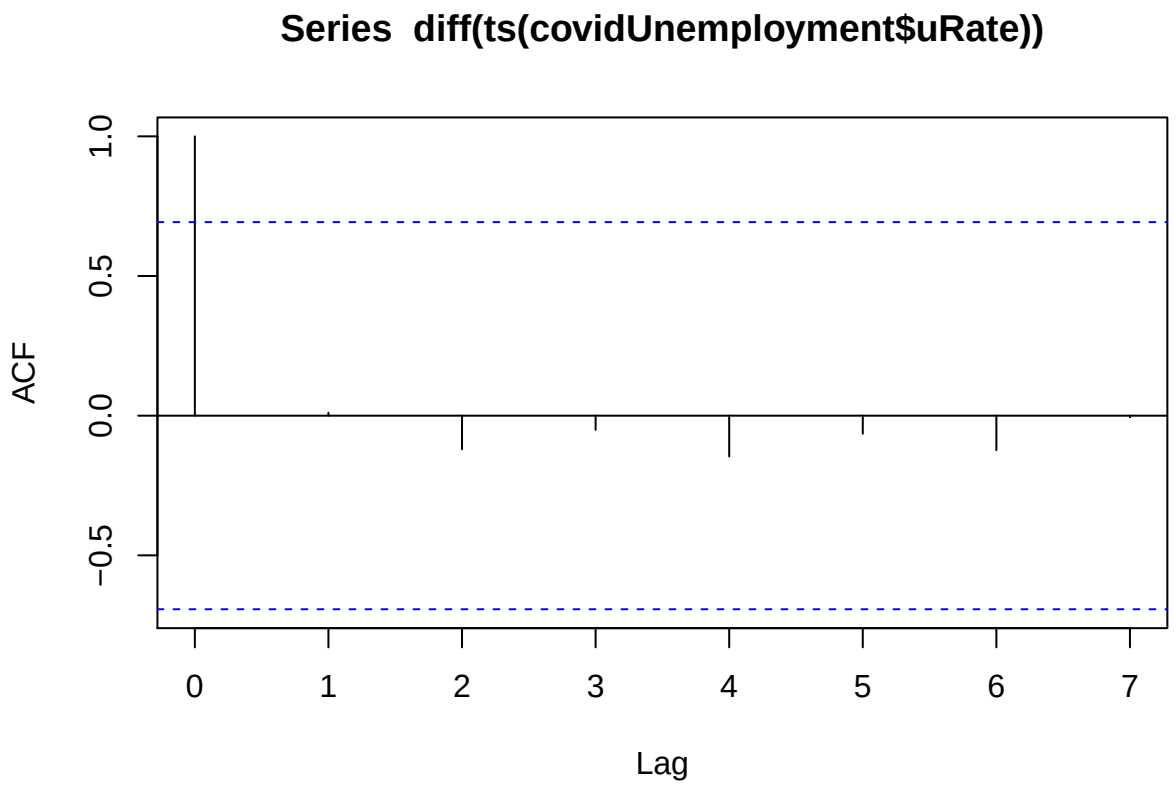
```
ts(covidUnemployment$uRate)
```

```
## Time Series:
## Start = 1
## End = 9
## Frequency = 1
## [1]  3.5  4.4 14.7 13.3 11.1 10.2  8.4  7.9  6.9
```

```
acf(ts(covidUnemployment$uRate))
```



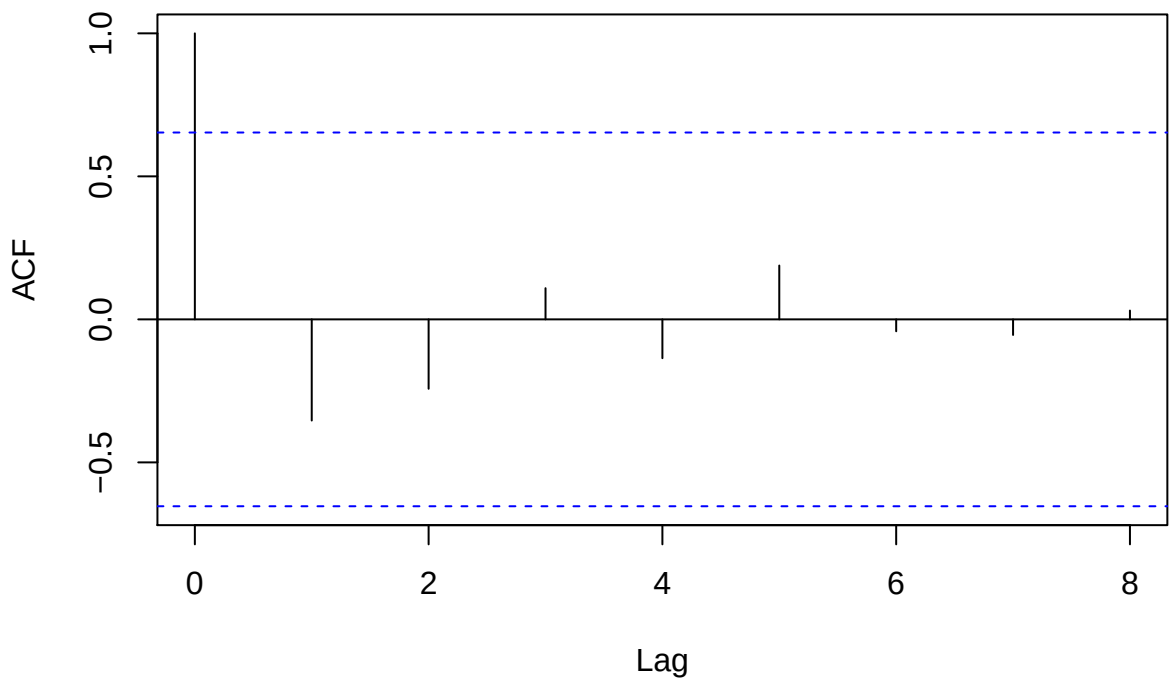

```
acf(diff(ts(covidUnemployment$uRate)))
```



The residuals for unemployment rate do not show evidence of significant autocorrelation. This justifies fitting a regression model using ordinary least squares instead of generalized least squares.

Regression: Ordinary Least Squares

```
# Full model that includes all features in the data set
m1 = lm(uRate ~ month + newCase + totCase + newDeath + totDeath,
        data = covidUnemployment)

# Check for autocorrelation
acf(m1$residuals)
```

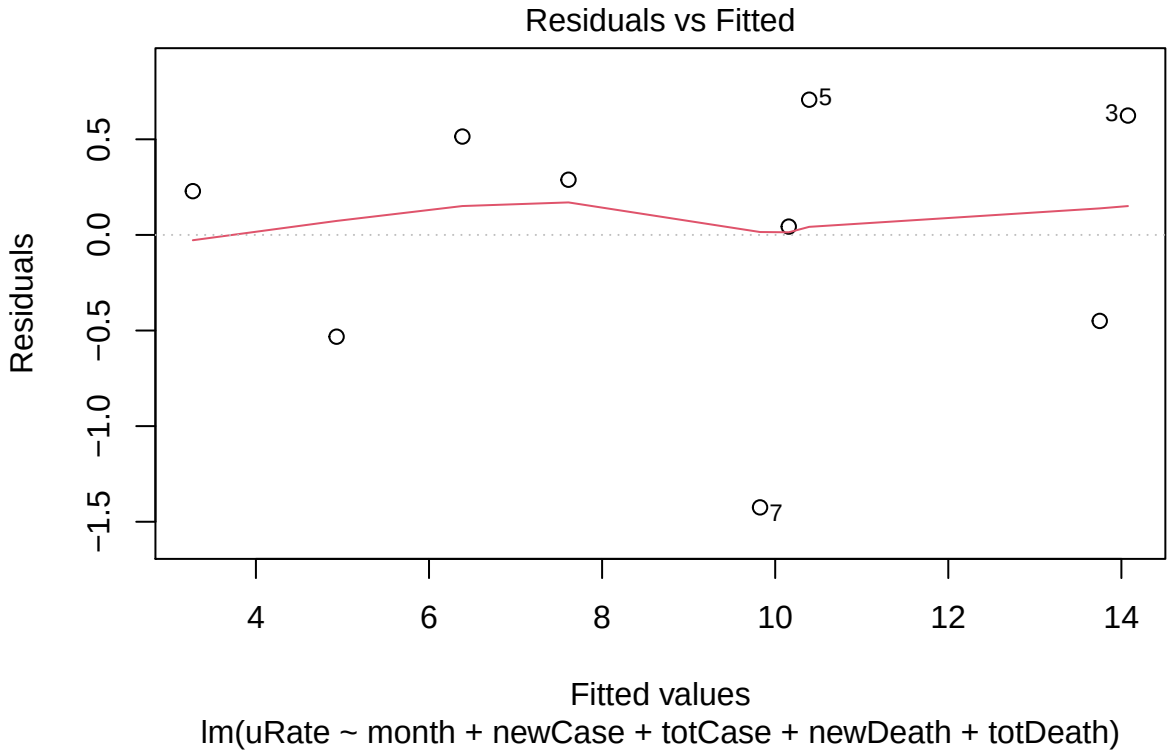


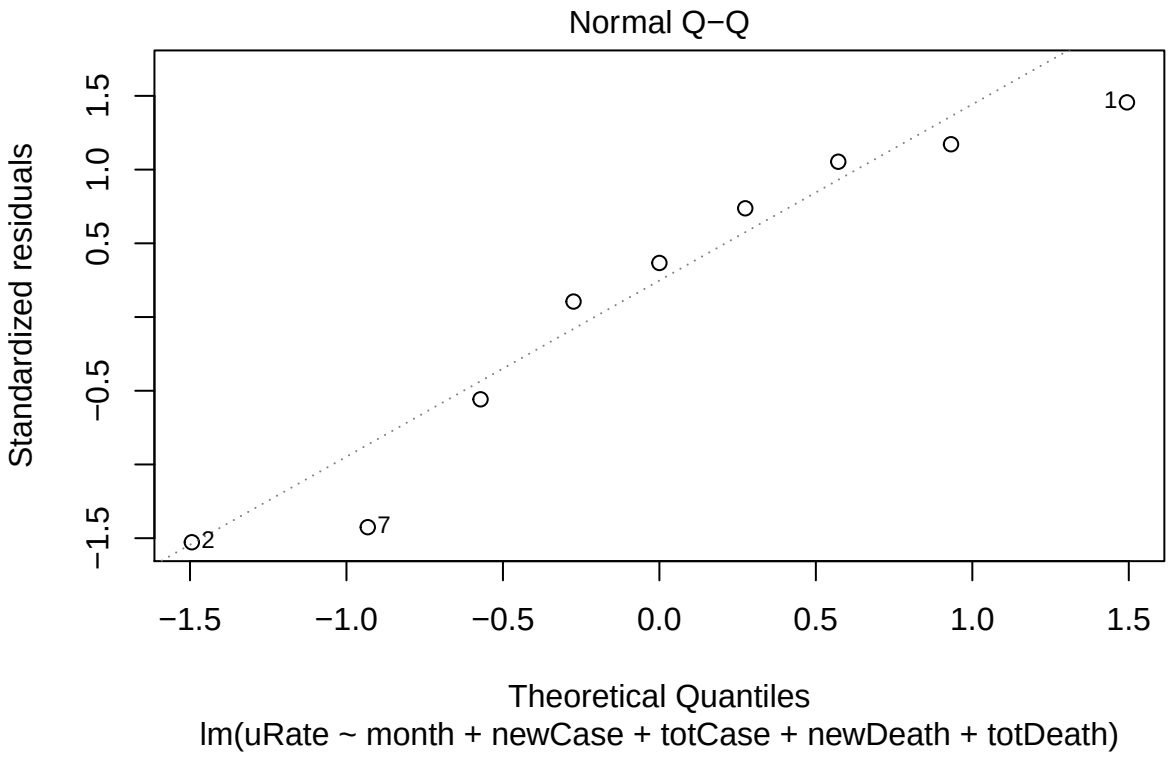

```
summary(m1)
```

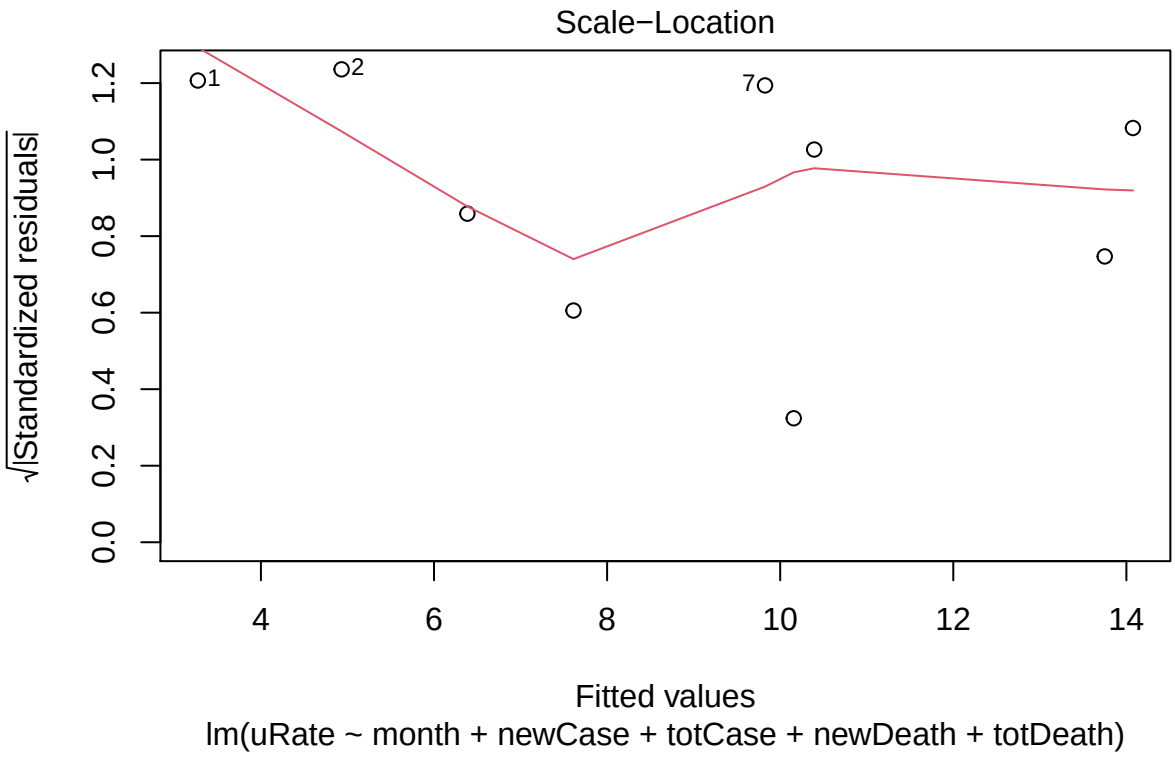
```
##
## Call:
## lm(formula = uRate ~ month + newCase + totCase + newDeath + totDeath,
##     data = covidUnemployment)
##
## Residuals:
##       1       2       3       4       5       6       7       8
##  0.22883 -0.53215  0.62423 -0.44979  0.70710  0.04353 -1.42484  0.28862
##       9
##  0.51446
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.016e+00  2.585e+00   0.780   0.4923
## month        1.255e+00  1.692e+00   0.742   0.5121
## newCase      7.768e-07  1.322e-06   0.588   0.5981
## totCase     -1.792e-06  7.385e-07  -2.427   0.0936 .
## newDeath     2.090e-04  4.216e-05   4.957   0.0158 *
## totDeath     1.246e-05  5.526e-05   0.225   0.8361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.127 on 3 degrees of freedom
## Multiple R-squared:  0.9667, Adjusted R-squared:  0.9111
```

```
## F-statistic: 17.39 on 5 and 3 DF,  p-value: 0.02006
```

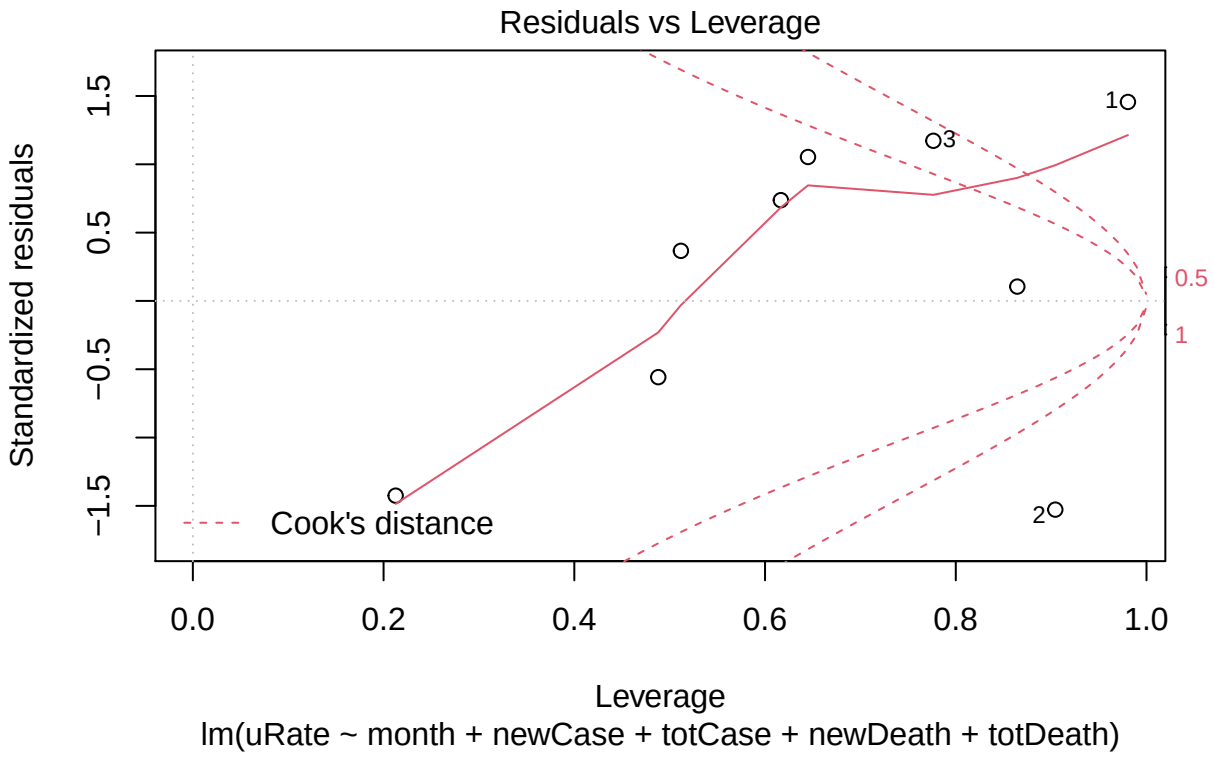
```
# Residual plots
plot(m1)
```



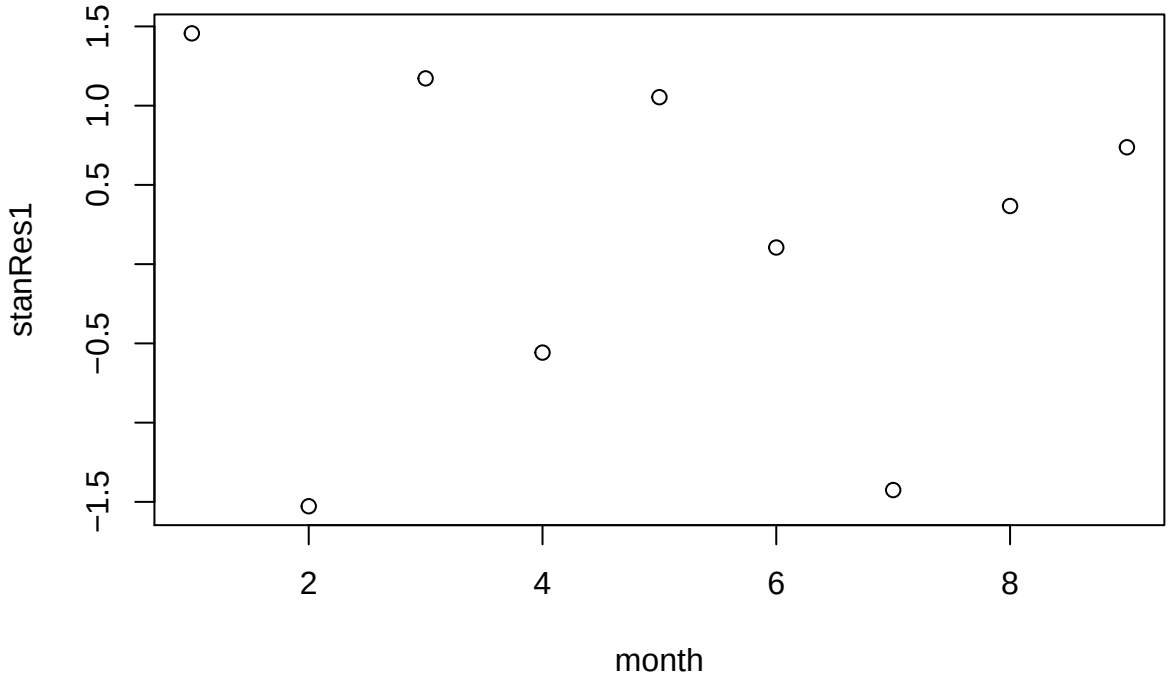

Normal Q−Q
Standardized residuals
−1.5
−0.5
0.5
1.0
1.5
2
7
1
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(uRate ~ month + newCase + totCase + newDeath + totDeath)



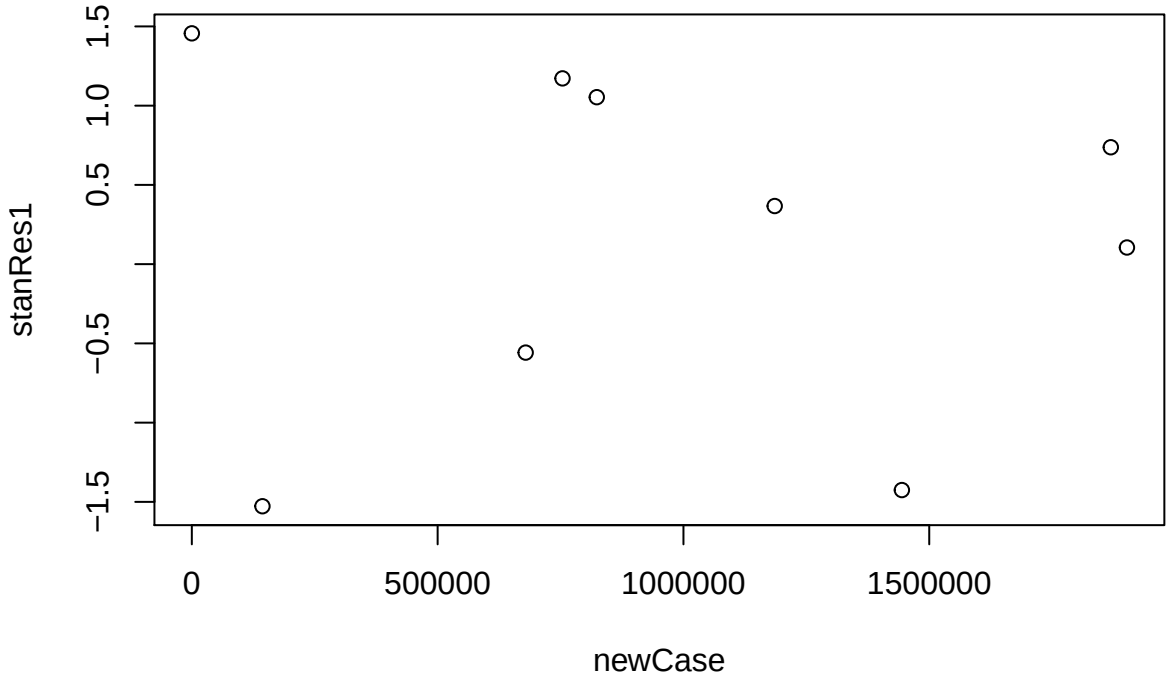

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



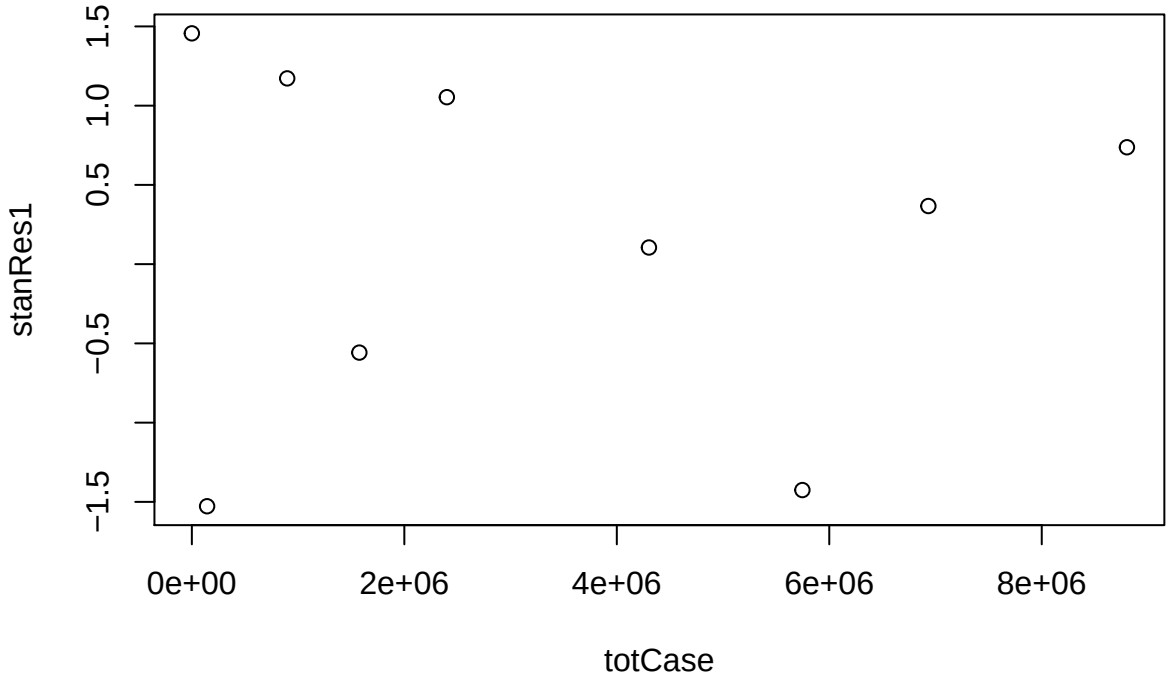

```
stanRes1 = rstandard(m1)
plot(month,    stanRes1)
```



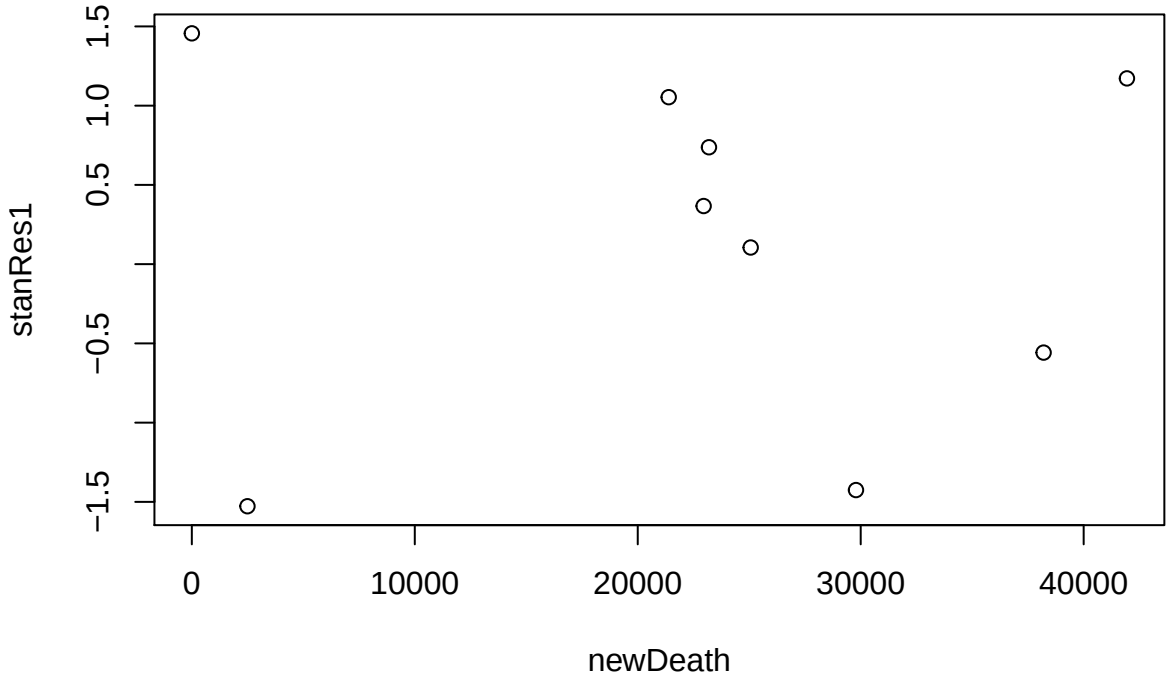

```
plot(newCase,  stanRes1)
```



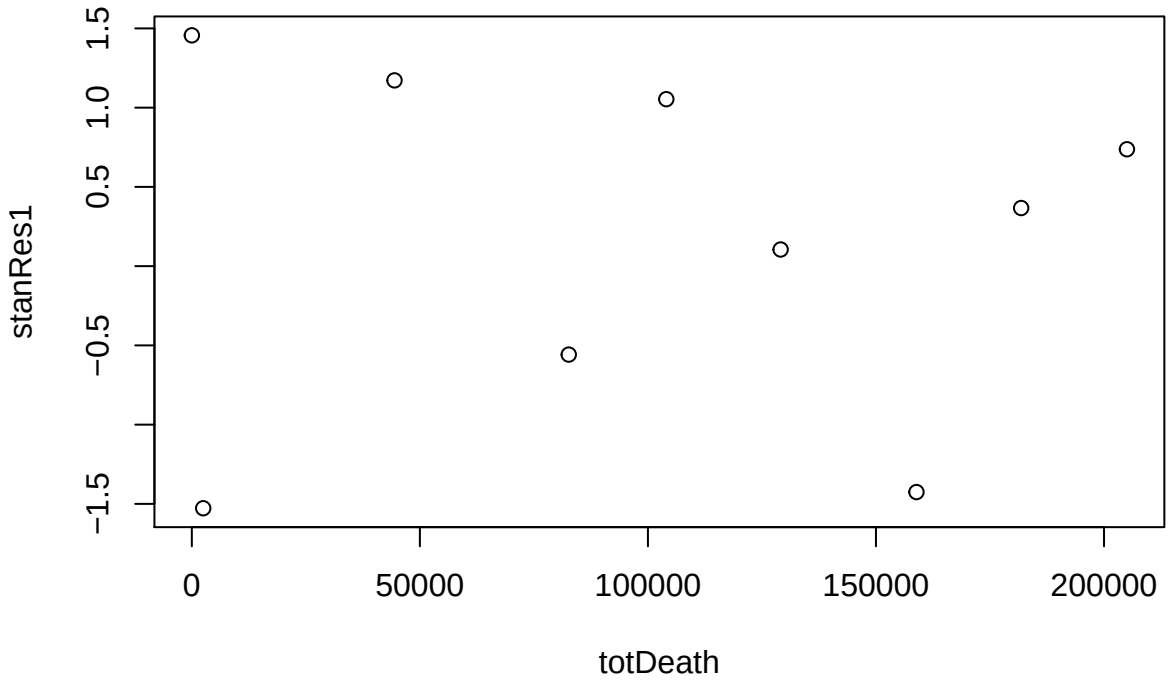

```
plot(totCase,  stanRes1)
```




```
plot(newDeath, stanRes1)
```



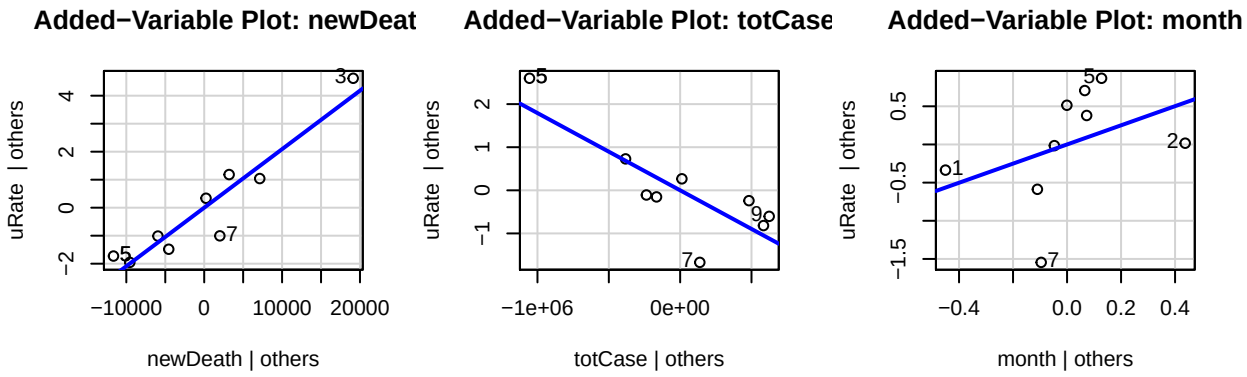

```
plot(totDeath, stanRes1)
```

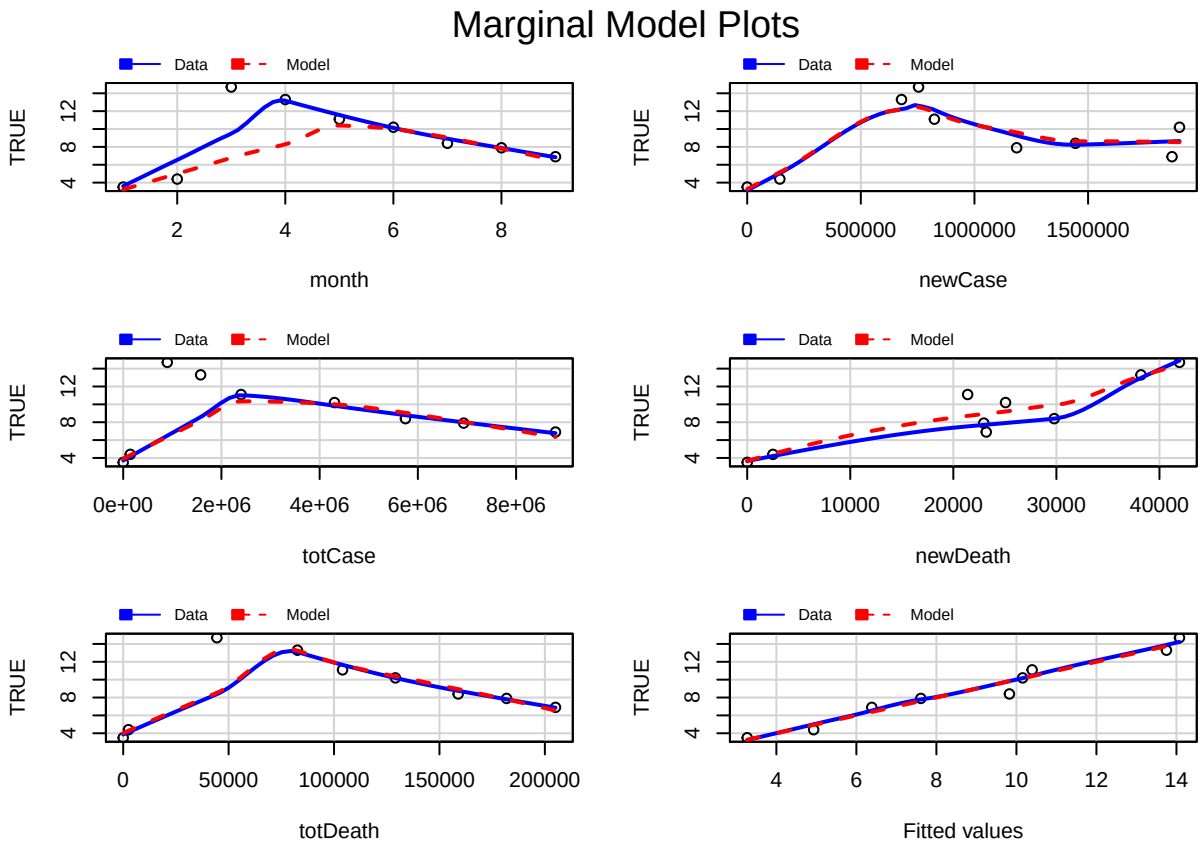



```
# Added variable plots
library(car)
```

```
## Loading required package: carData
```

```
par(mfrow=c(2,3))
avPlot(m1,variable="newDeath",ask=FALSE)
avPlot(m1,variable="totCase",ask=FALSE)
avPlot(m1,variable="month",ask=FALSE)

# Marginal model plots
mmps(m1)
```


Added−Variable Plot: newDeat
uRate | others
newDeath | others
−10000
0
10000
20000
−2
0
2
4
3
5
7
Added−Variable Plot: totCase
uRate | others
totCase | others
−1e+06
0e+00
−1
0
1
2
5
7
9
Added−Variable Plot: month
uRate | others
month | others
−0.4
0.0
0.2
0.4
−1.5
−0.5
0.5
5
1
2
7



Using the autocorrelation function, the plot of residuals for the full model does not show evidence of sigificant autocorrelation.

The plot of residuals versus fitted values does not show evidence of nonconstant variance, suggesting that the model is a valid fit for the data. The normal Q-Q plot suggests that the residuals follow a symmetric distribution albeit with heavier tails than a normal distribution.

The first two values in the data set, corresponding to February and March, are bad leverage points. During these months, the SARS-CoV-2 virus had not yet completed its initial spread across the United States.

For the month of February, the national unemployment rate was at a 50-year low of 3.5%. For COVID-19, there were 19 cases and 1 death. These low figures compared to subsequent months, explains why February is a bad leverage point.

The national unemployment rate increased to 4.4% in March, then to 14.7% in April. The number of COVID-19 cases and deaths followed a similar trajectory for these two months. These sharp increases from March to April, explain why March was the other bad leverage point.

For the remaining months in the data set, May through October, a plot of the square root of the absolute value of the standardized residuals against fitted values lacks evidence of nonconstant variance. Furthermore, the marginal model plots show that the fitted curves match well with the nonparametric curves. This suggests that the full multilinear regression model may provide an adequate fit for the data.

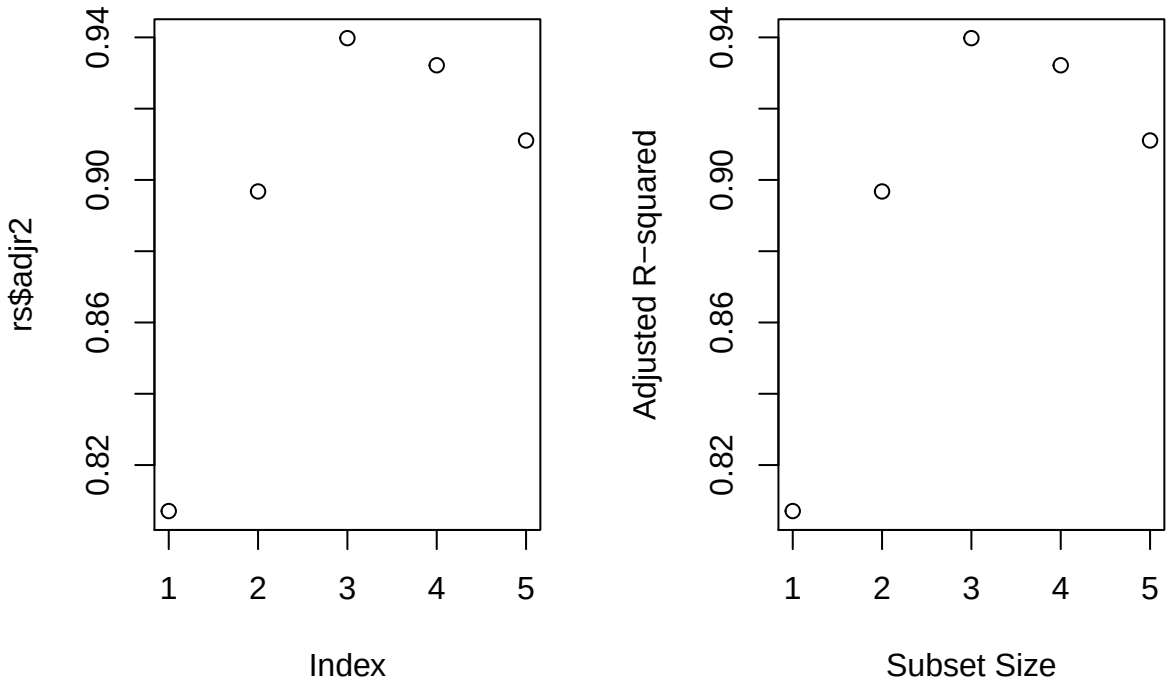
The overall F-test for the full model is statistically significant, with a p-value less than 0.05. However, only one of the estimated regression coefficients is statistically significant. This is the number of new deaths, with a p-value less than 0.05. This is also evident in the marginal model plots. Thus, the next step was to choose a subset of the predictors using variable selection.

Variable selection using all possible subsets

```
#install.packages("leaps")
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.5
```

```
b    = regsubsets(as.matrix(covidMonthly), uRate)
rs   = summary(b)
adjRsq = rs$adjr2

# Plots of adjusted R-squared against the number of predictors
par(mfrow=c(1,2))
plot(rs$adjr2)
plot(1:5,rs$adjr2,xlab="Subset Size",ylab="Adjusted R-squared")
```




```
library(car)
#subsets(b,statistic=c("adjr2"))

# Calculate adjusted R-squared
om1  = lm(uRate ~ newDeath)
om2  = lm(uRate ~ newDeath + totCase)
om3  = lm(uRate ~ newDeath + totCase + month)
om4  = lm(uRate ~ newDeath + totCase + month + newCase)
om5  = m1

# Subset size = 1
```

```
n        = length(om1$residuals)
npar     = length(om1$coefficients) + 1
sub1AIC  = (extractAIC(om1,k=2))[2]
sub1AICc = (extractAIC(om1,k=2) + 2*npar*(npar+1)/(n-npar-1))[2]
sub1BIC  = (extractAIC(om1,k=log(n)))[2]

# Subset size = 2
npar     = length(om2$coefficients) + 1
sub2AIC  = (extractAIC(om2,k=2))[2]
sub2AICc = (extractAIC(om2,k=2) + 2*npar*(npar+1)/(n-npar-1))[2]
sub2BIC  = (extractAIC(om2,k=log(n)))[2]

# Subset size = 3
npar     = length(om3$coefficients) +1
sub3AIC  = (extractAIC(om3,k=2))[2]
sub3AICc = (extractAIC(om3,k=2) + 2*npar*(npar+1)/(n-npar-1))[2]
sub3BIC  = (extractAIC(om3,k=log(n)))[2]

# Subset size = 4
npar     = length(om4$coefficients) +1
sub4AIC  = (extractAIC(om4,k=2))[2]
sub4AICc = (extractAIC(om4,k=2) + 2*npar*(npar+1)/(n-npar-1))[2]
sub4BIC  = (extractAIC(om4,k=log(n)))[2]

# Subset size = 5
npar     = length(om5$coefficients) +1
sub5AIC  = (extractAIC(om5,k=2))[2]
sub5AICc = (extractAIC(om5,k=2) + 2*npar*(npar+1)/(n-npar-1))[2]
sub5BIC  = (extractAIC(om5,k=log(n)))[2]

# Table with four criteria for evaluating subsets of predictors
AIC   = c(sub1AIC, sub2AIC, sub3AIC, sub4AIC, sub5AIC)
AICc  = c(sub1AICc, sub2AICc, sub3AICc, sub4AICc, sub5AICc)
BIC   = c(sub1BIC, sub2BIC, sub3BIC, sub4BIC, sub5BIC)
SubSize = 1:5
cbind(SubSize, adjRsq, AIC, AICc, BIC)
```

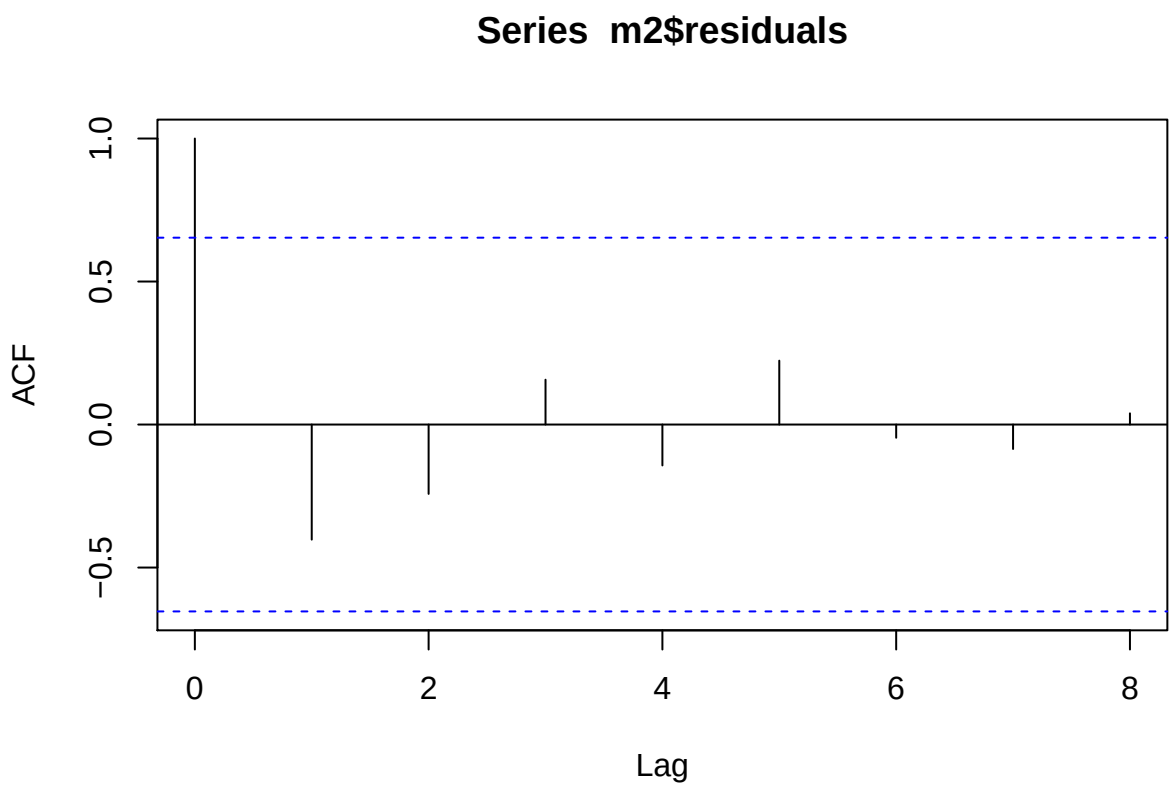
```
##      SubSize    adjRsq       AIC      AICc       BIC
## [1,]       1 0.8070927 10.853273  15.65327 11.247722
## [2,]       2 0.8967661  5.839004  15.83900  6.430678
## [3,]       3 0.9397833  1.346686  21.34669  2.135585
## [4,]       4 0.9321760  2.409091  44.40909  3.395214
## [5,]       5 0.9110751  4.257834 116.25783  5.441181
```

The table above gives the values of adjusted R squared, AIC, AICc and BIC for the best subset of each size. The predictor subset of size three had the best score for each of the four criteria. It had the maximum value for adjusted R squared, and the minumum values for AIC, AICc, and BIC.

Linear model using best subset

```
# Preferred three-variable model
m2 = om3
```

```r
# Check for autocorrelation
acf(m2$residuals)
```

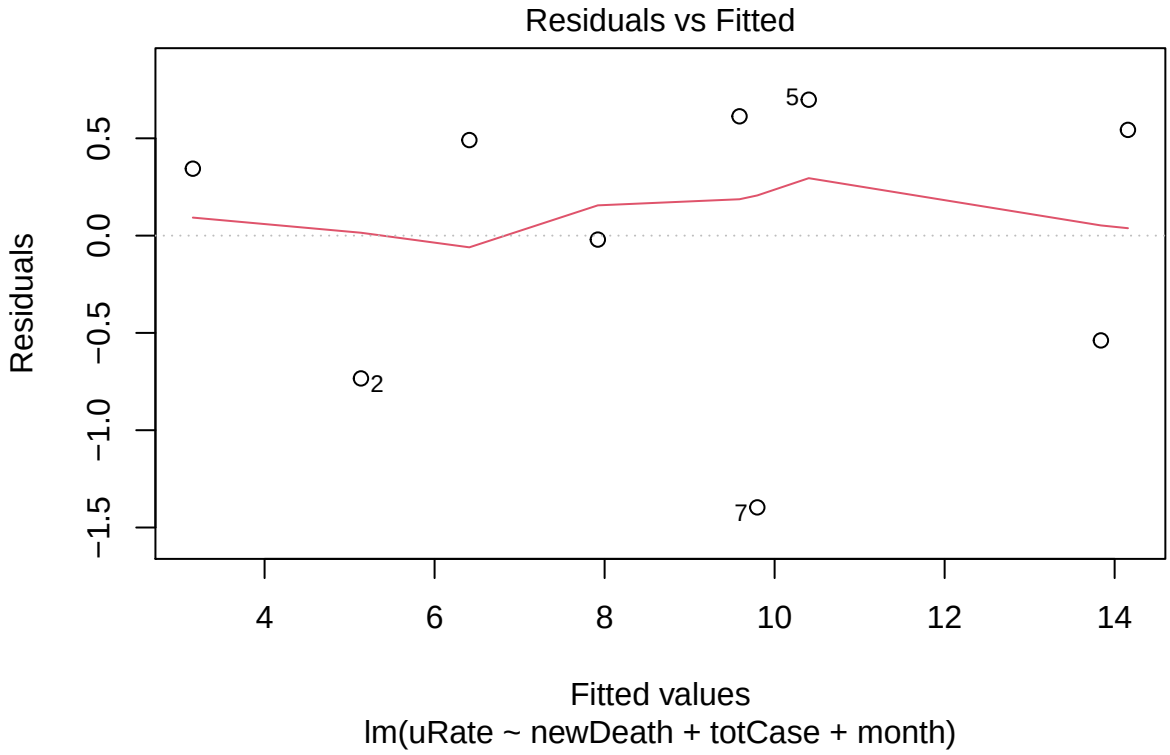


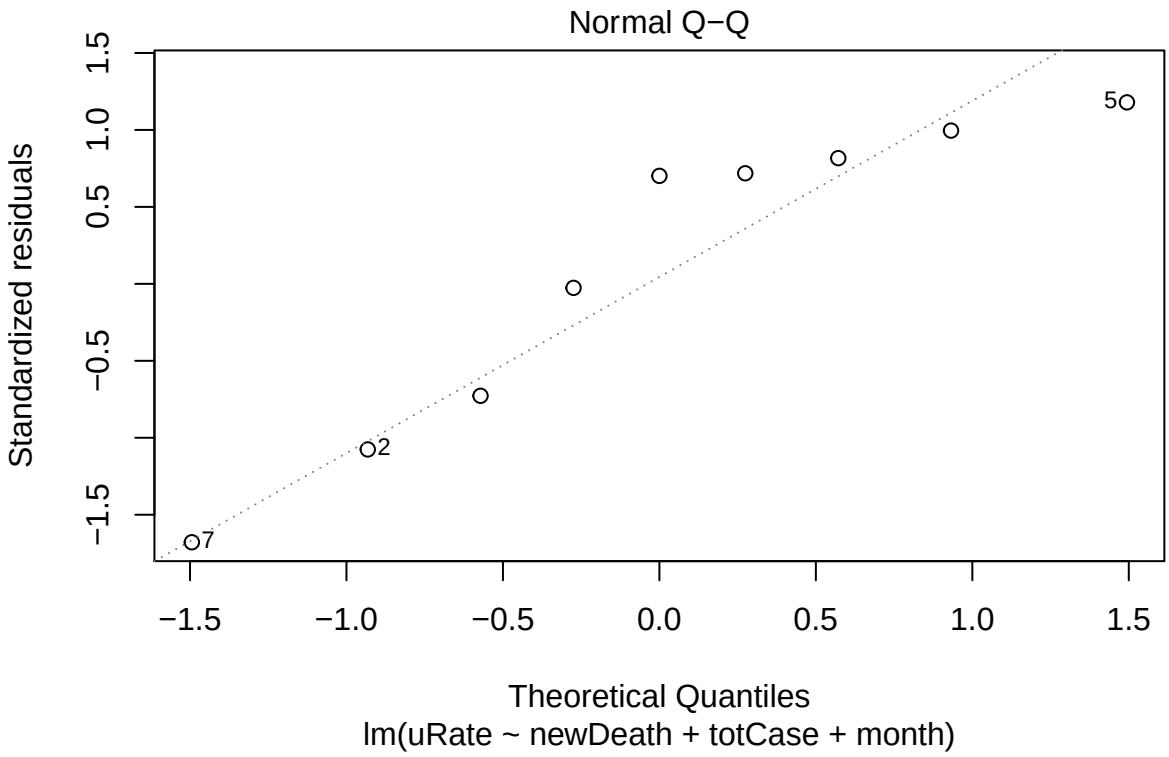

```r
summary(m2)
```

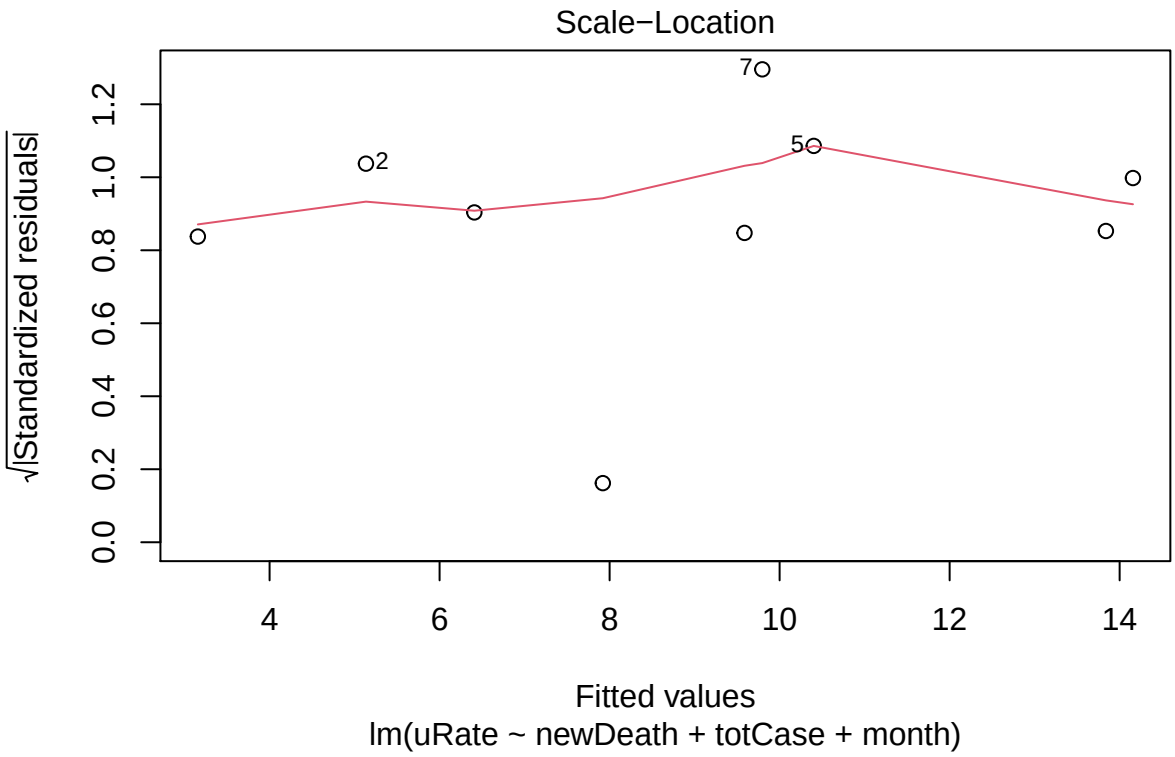
```
##
## Call:
## lm(formula = uRate ~ newDeath + totCase + month)
##
## Residuals:
##       1       2       3       4       5       6       7       8
##  0.34393 -0.73392  0.54333 -0.53856  0.69834  0.61298 -1.39649 -0.02054
##       9
##  0.49092
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.476e+00  1.339e+00   1.102 0.320613
## newDeath     2.193e-04  3.122e-05   7.024 0.000903 ***
## totCase     -1.735e-06  6.027e-07  -2.879 0.034621 *
## month        1.680e+00  7.307e-01   2.299 0.069844 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9271 on 5 degrees of freedom
## Multiple R-squared:  0.9624, Adjusted R-squared:  0.9398
```

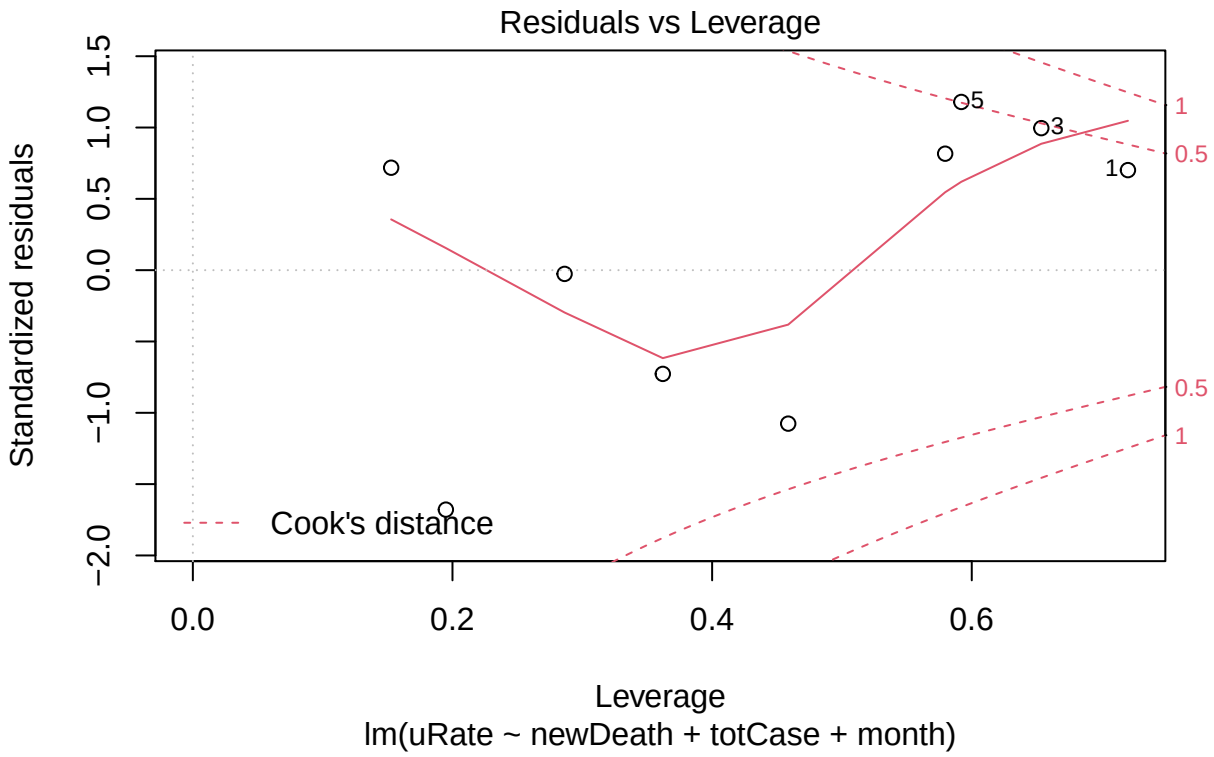
```
## F-statistic: 42.62 on 3 and 5 DF,  p-value: 0.0005522
```

```
# Residual plots
plot(m2)
```

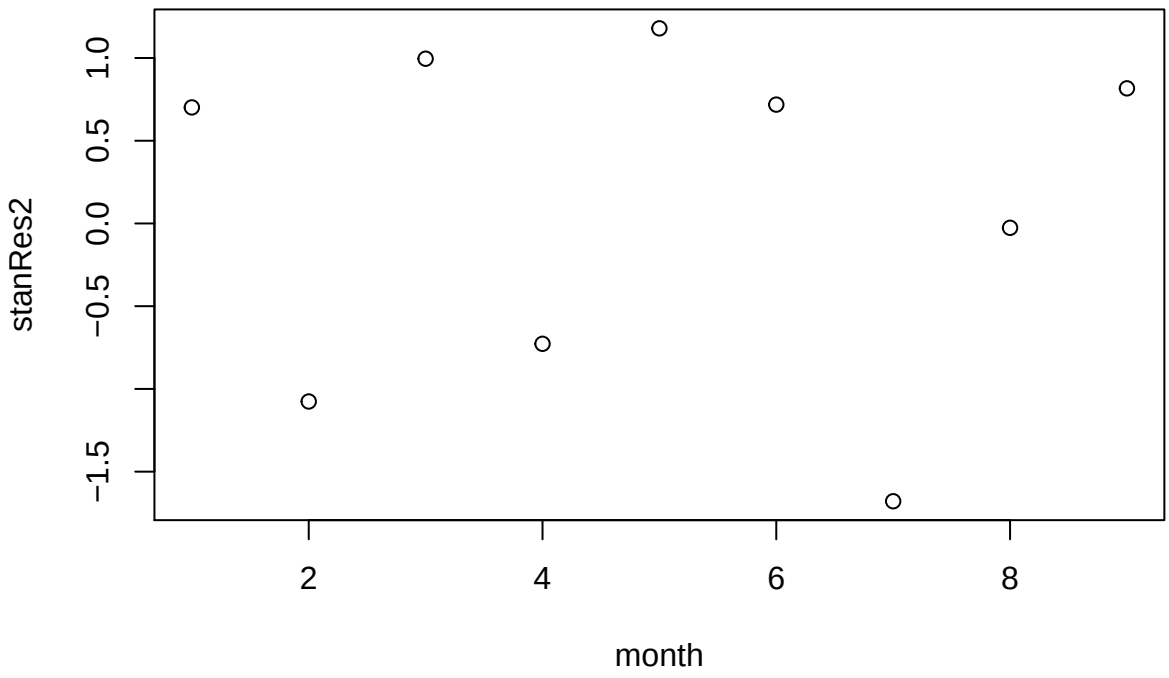



Normal Q−Q
Standardized residuals
1.5
1.0
0.5
−0.5
−1.5
5
2
7
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(uRate ~ newDeath + totCase + month)

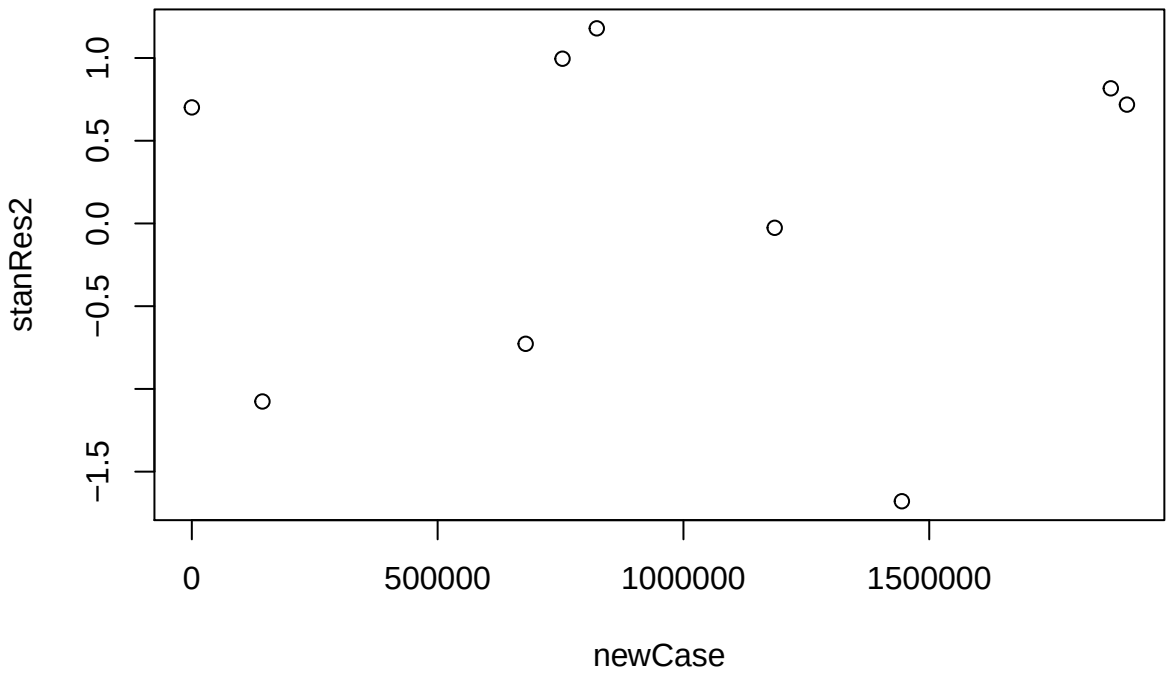

Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
7
2
5
4
6
8
10
12
14
Fitted values
lm(uRate ~ newDeath + totCase + month)



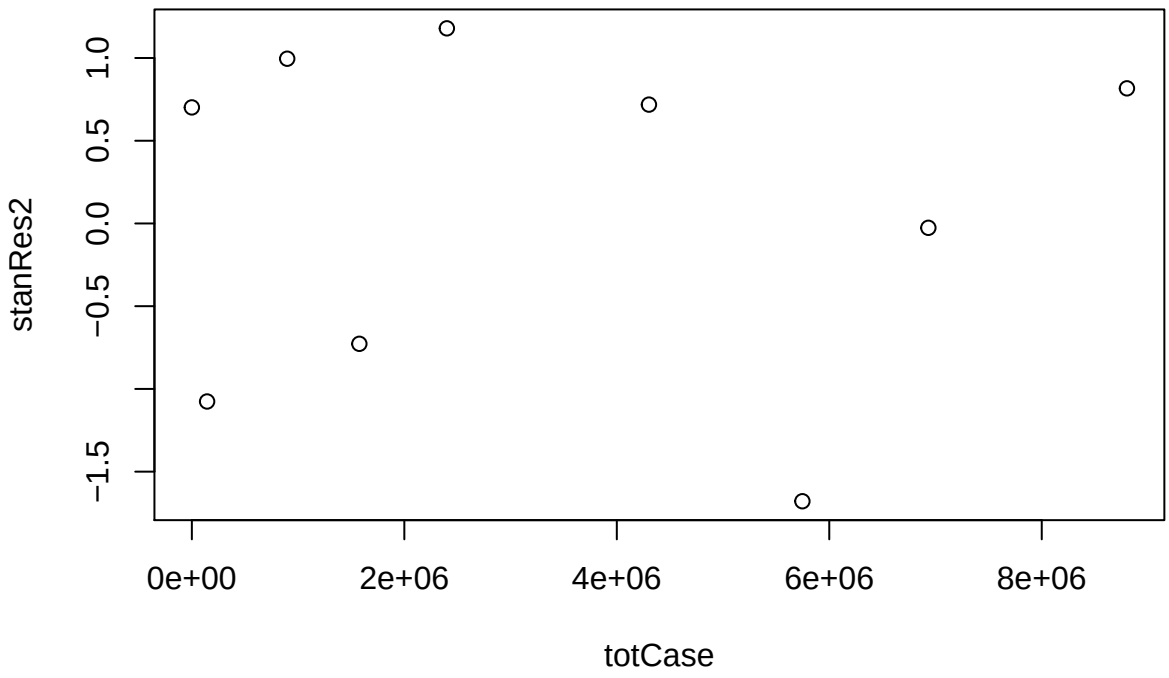

```
stanRes2 = rstandard(m2)
plot(month,    stanRes2)
```



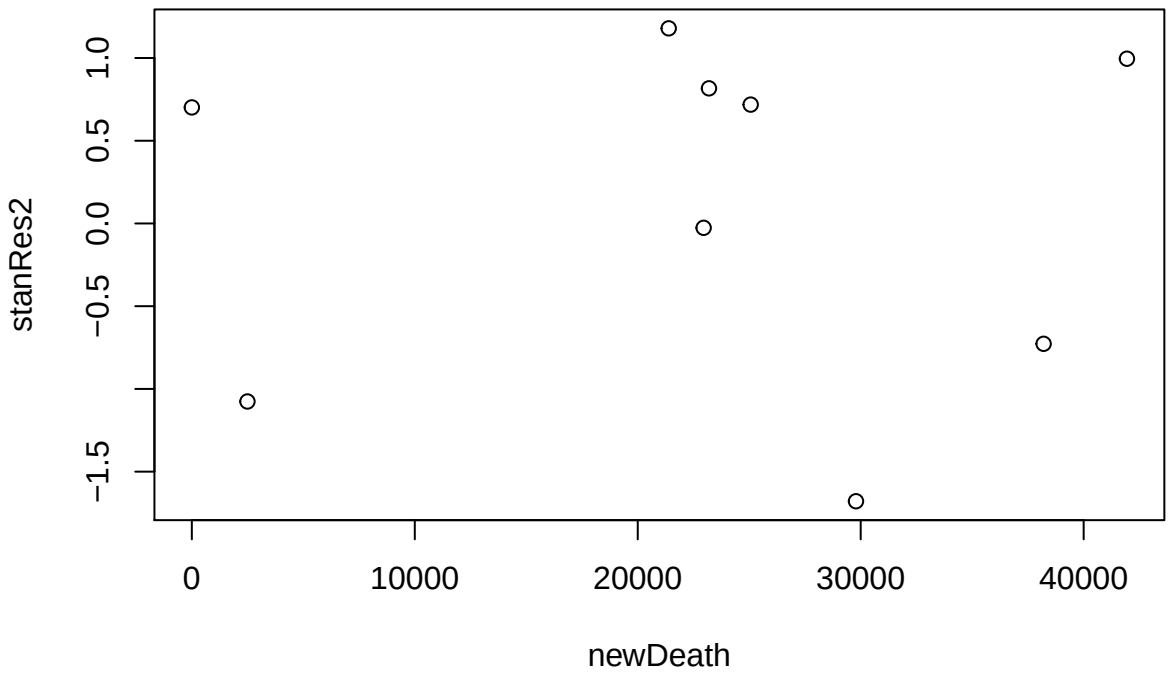

```
plot(newCase,  stanRes2)
```



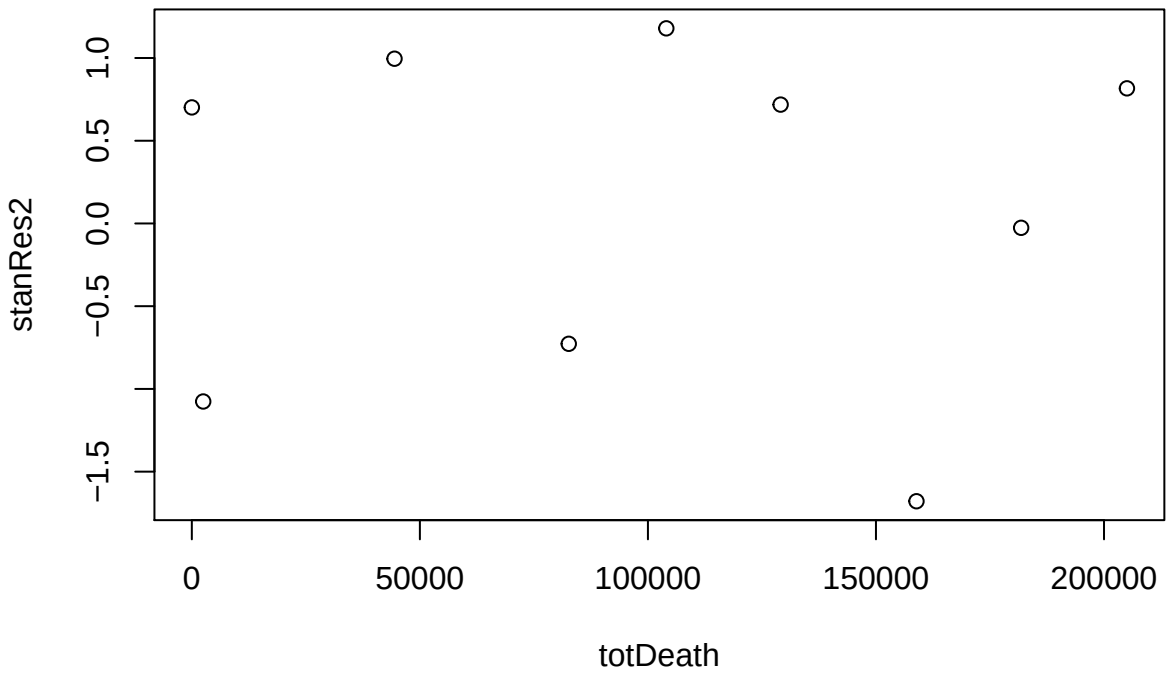

```
plot(totCase,  stanRes2)
```



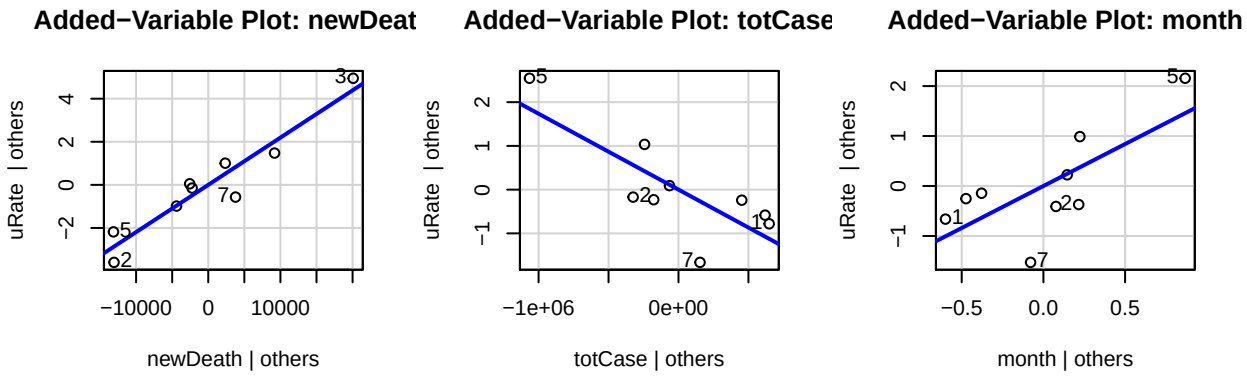

```
plot(newDeath, stanRes2)
```

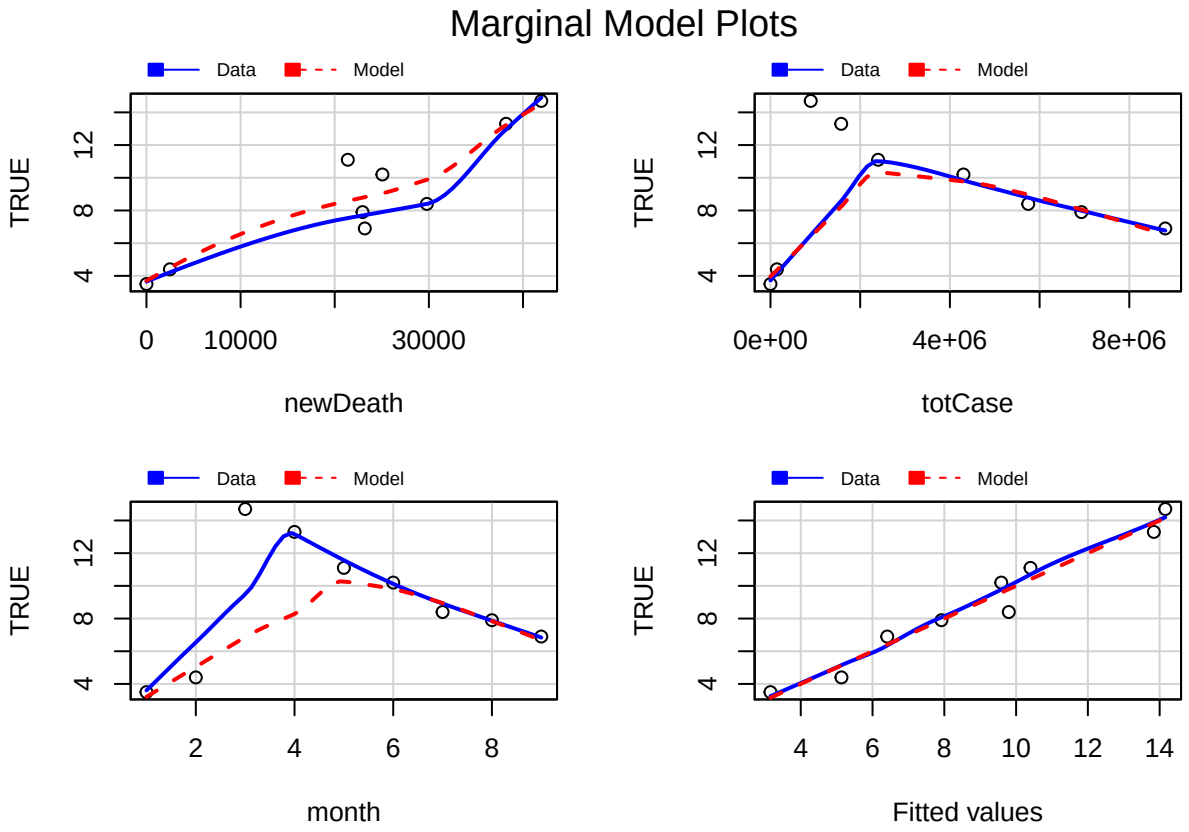



```
plot(totDeath, stanRes2)
```




```
# Added variable plots
library(car)
par(mfrow=c(2,3))
avPlot(m2,variable="newDeath",ask=FALSE)
avPlot(m2,variable="totCase",ask=FALSE)
avPlot(m2,variable="month",ask=FALSE)

# Marginal model plots
mmps(m2)
```


Added−Variable Plot: newDeat
Added−Variable Plot: totCase
Added−Variable Plot: month
uRate | others
newDeath | others
totCase | others
month | others
−10000
0
10000
−1e+06
0e+00
−0.5
0.0
0.5



Using the autocorrelation function, the plot of residuals for the chosen three-variable model does not show evidence of sigificant autocorrelation.

The overall F-statistic for the full model is highly statistically significant, with a p-value less than 0.001. Of the three estimated regression coefficients, new deaths is highly statistically significant with a p-value less than 0.001, total cases is statistically significant with a p-value less than 0.05, and month is less significantly significant with a p-value below 0.10. The significance of these three variables are also evident in the added variable plots.

The plot of residuals versus fitted values does not show evidence of nonconstant variance, suggesting that the model is a valid fit for the data. The normal Q-Q plot suggests that a straight-line regression fit is appropriate due to the lack of bad leverage points. The plot of the square root of the absolute value of the standardized residuals against fitted values shows evidence of constant variance due to the mostly horizontal loess fit.

In the plot of Residuals vs. Leverage, none of the standardized residuals exceed a Cook's distance of 1. However, the value for time 5 warrants investigation because its residual slightly exceeds a Cook's distance of 0.5.

The plots of standardized residuals for each of the predictors indicate nonconstant variance.

The marginal model plots show that the fitted values match well with the curvature of the nonparametric estimates.

Altogether, the diagnostic plots suggest that the three-variable model provides a valid fit for the data, and a better fit than the full model.

Prediction interval for November 2020

```
bestOut = predict(m2, interval = "prediction",
                  newdata = data.frame(newDeath = newDeath[9],
                                       totCase = totCase[9],
                                       month = 10))
bestOut
```

```
##        fit      lwr      upr
## 1 8.089205 4.954087 11.22432
```

To predict the national unemployment rate for November, using the numbers of new deaths and total cases in October, the three-variable model generates a prdiction interval of (4.9540869, `bestOut[3]`), with the middle of the interval being 8.0892048%.