hw2

Aldo Sanjoto

September 27, 2017

1a)

```
library("alr4")
```

```
## Loading required package: car
```

```
## Loading required package: effects
```

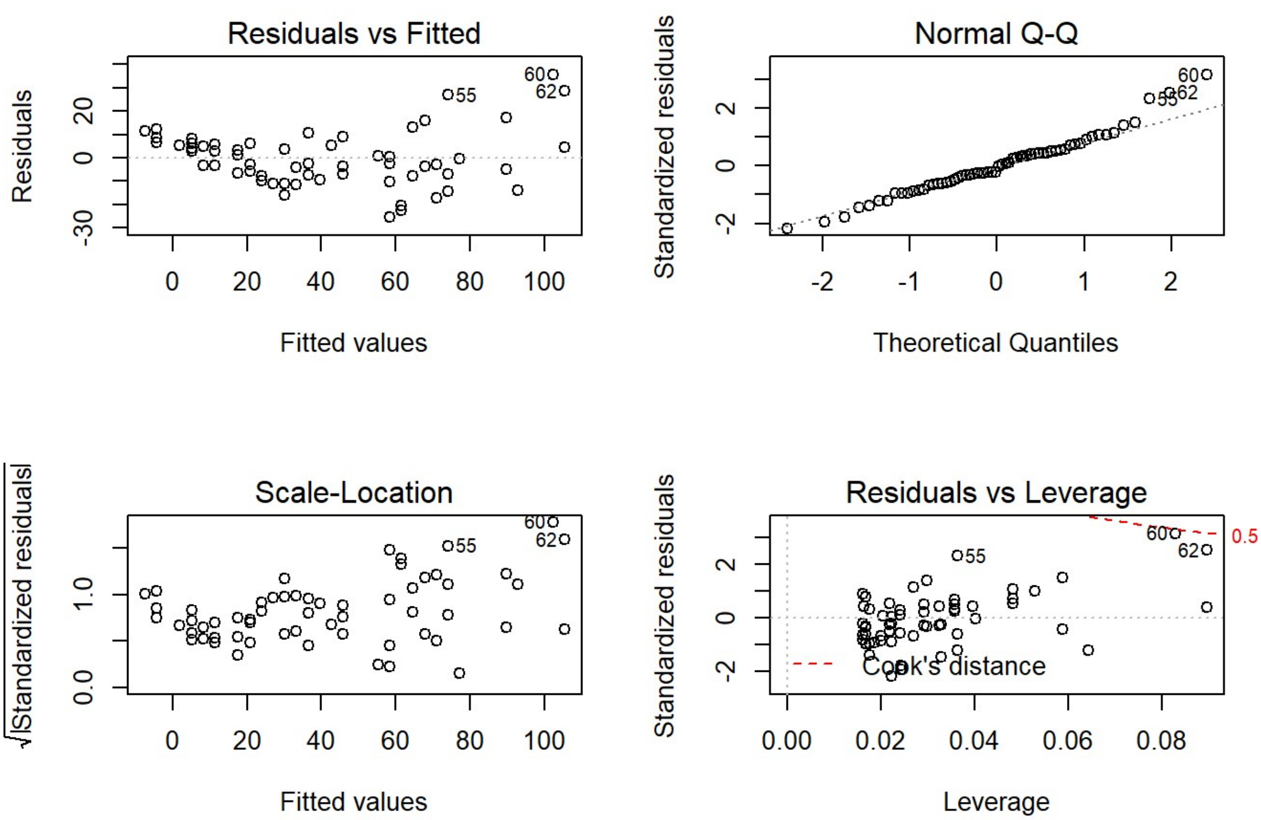
```
##
## Attaching package: 'effects'
```

```
## The following object is masked from 'package:car':
##
##     Prestige
```

```
data("stopping")
#head(stopping)
#??stopping
model = lm(formula = Distance ~ Speed, data = stopping)
summary(model)
```

```
##
## Call:
## lm(formula = Distance ~ Speed, data = stopping)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.410  -7.343  -1.334   5.927  35.608
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.1309     3.2308  -6.231 5.04e-08 ***
## Speed         3.1416     0.1514  20.751  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.77 on 60 degrees of freedom
## Multiple R-squared:  0.8777, Adjusted R-squared:  0.8757
## F-statistic: 430.6 on 1 and 60 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model, add.smooth = FALSE)
```



1b) There is slightly a curvature in the Residuals vs Fitted plot however the points are not evenly distributed. Thus, mean function might not be appropriate.

1c) There's no problem with the constant variance since in the Scale vs Location plot, trend is roughly flat (constant).

```
model$fitted.values[which.max(model$residuals)]
```

```
##       60
## 102.3922
```

```
model$fitted.values[which.min(model$residuals)]
```

```
##       41
## 58.40952
```

```
residuals(model)
```

```
##           1           2           3           4           5           6
##  11.5644657   6.4228475   8.4228475  12.4228475  12.4228475   5.1396110
##           7           8           9          10          11          12
##   5.1396110   2.9979927   3.9979927   5.9979927   7.9979927  -3.1436255
##          13          14          15          16          17          18
##  -3.1436255   4.8563745  -3.2852437   2.7147563   5.7147563  -6.5684802
##          19          20          21          22          23          24
##   1.4315198   3.4315198  -5.7100985  -2.7100985   6.2899015  -9.8517167
##          25          26          27          28          29          30
##  -7.8517167 -10.9933350 -16.1349532 -11.1349532   3.8650468 -11.2765714
##          31          32          33          34          35          36
##  -4.2765714  -7.4181897  -2.4181897  10.5818103  -9.5598079   5.2985738
##          37          38          39          40          41          42
##  -6.8430444  -3.8430444   9.1569556   0.7321009 -25.4095174 -10.4095174
##          43          44          45          46          47          48
##  -2.4095174   0.5904826 -22.5511356 -20.5511356  -7.6927539  13.3072461
##          49          50          51          52          53          54
##  -3.8343721  16.1656279 -16.9759903  -2.9759903 -14.1176086  -7.1176086
##          55          56          57          58          59          60
##  26.8823914  -0.2592268  -4.8256998  17.1743002 -13.9673180  35.6078272
##          61          62
##   4.4662090  28.4662090
```

1d) Largest residual: 60th value: 35.6078272, Smallest residual: 41st value: -25.4095174

```
#hatvalues(model)
hatvalues(model)[which.max(hatvalues(model))]
```

```
##         61
## 0.08967251
```

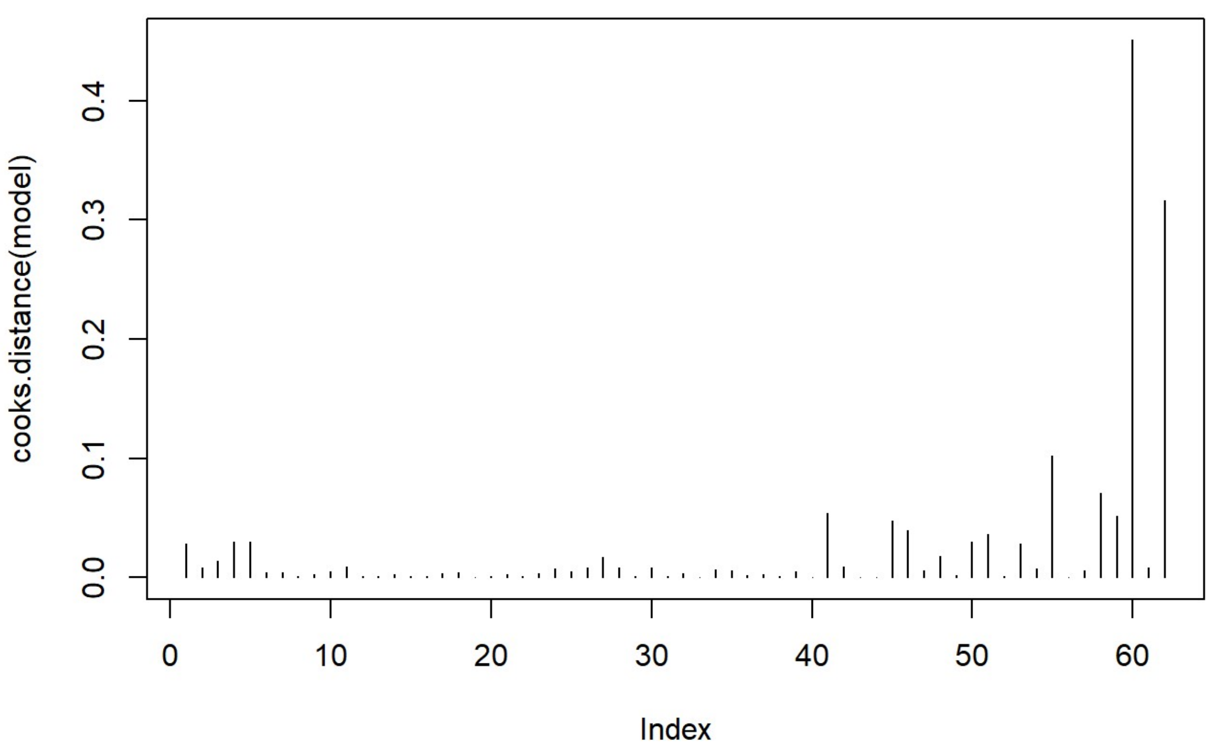
1e) 61th has the largest leverage value with 0.08967251

1f) 60th value might be the outlier since it is the furthest from the other points in the plot.

```
#cooks distance
which(cooks.distance(model) >= 1)
```

```
## named integer(0)
```

```
plot(cooks.distance(model), type = "h")
```



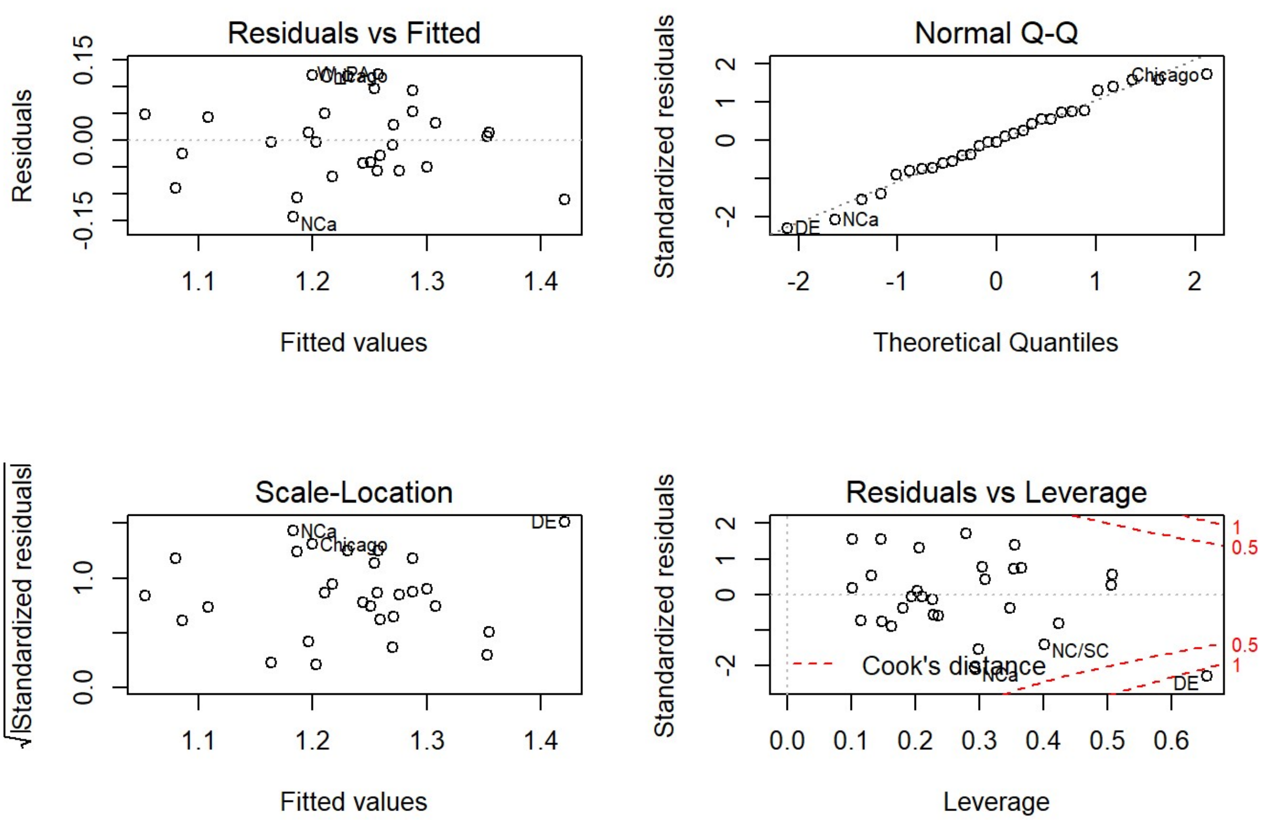
1g) Based on the plot above, no value is greater than 1 thus no influential points.

2a)

```
data("drugcost")
head(drugcost)
```

```
##     COST RXPM GS   RI COPAY  AGE    F      MM
## MN1 1.34  4.2 36 45.6 10.87 29.7 52.3 1158096
## MN2 1.34  5.4 37 45.6  8.66 29.7 52.3 1049892
## MN3 1.38  7.0 37 45.6  8.12 29.7 52.3   96168
## GA  1.22  7.1 40 23.6  5.89 28.7 53.4  407268
## GA2 1.08  3.5 40 23.6  6.05 28.7 53.4   13224
## AZ1 1.16  7.2 46 22.3  5.05 29.1 52.2  303312
```

```
model2 = lm(formula = COST ~ RXPM + GS + RI + COPAY + AGE + F + MM, data = drugcost)
par(mfrow=c(2,2))
plot(model2, add.smooth = FALSE)
```



2b) The trend is roughly flat in the Residual vs Fitted plot. Thus, mean function might be appropriate.

2c) There's no problem with the constant variance since in the Scale vs Location plot, trend is roughly flat (constant).

```
match(max(residuals(model2)), residuals(model2))
```

```
## [1] 29
```

```
fitted.values(model2)[match(max(residuals(model2)), residuals(model2))]
```

```
##     W_PA
## 1.257477
```

```
match(min(residuals(model2)), residuals(model2))
```

```
## [1] 10
```

```
fitted.values(model2)[match(min(residuals(model2)), residuals(model2))]
```

```
##      NCa
## 1.182888
```

```
residuals(model2)
```

```
##          MN1          MN2          MN3           GA          GA2
##  0.032441427  0.052605956  0.092430041 -0.056359558 -0.106567981
##          AZ1          AZ2           TN    San_Diego          NCa
## -0.003963415 -0.050521007 -0.003367492  0.047232131 -0.142888039
##         SoCA        NC/SC           LA           FL       Dallas
## -0.025326259 -0.089613792 -0.043788286  0.041763224 -0.010061894
##      Chicago      Houston           NJ           DE Mid-Atlantic
##  0.120153296 -0.040578271  0.119462661 -0.111105245  0.006631769
##     Richmond           NY        C/E_PA         S_NE    St._Louis
## -0.067659763  0.095783925 -0.029092217 -0.056938336  0.028642207
##           OH   Cincinnati     Columbus         W_PA
##  0.013954188  0.049449241  0.014758960  0.122522528
```

2d) Largest residual: 29th value: 0.122522528, Smallest residual: 10th value: -0.142888039

```
hatvalues(model2)[which.max(hatvalues(model2))]
```

```
##        DE
## 0.6553194
```

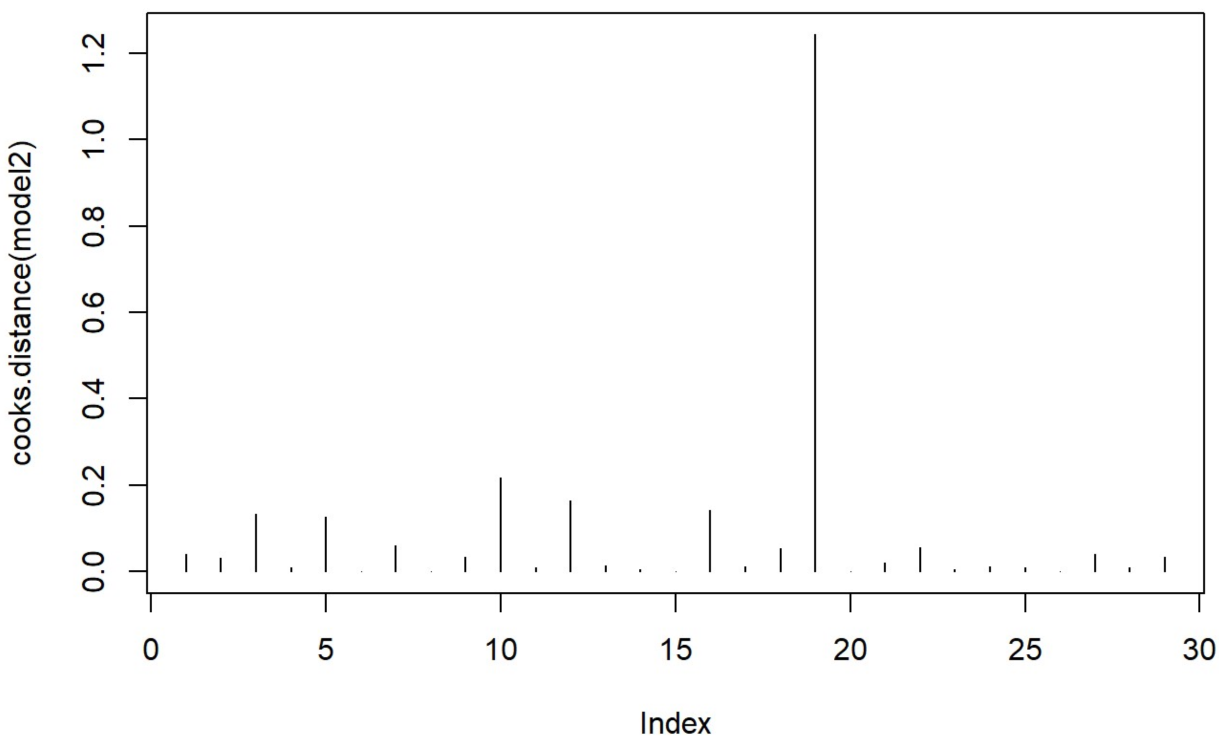
2e) DE has the largest leverage value with 0.6553194

2f) DE is the outlier because it is the furthest from the other of the points in the plot.

```
which(cooks.distance(model2) >= 1)
```

```
## DE
## 19
```

```
plot(cooks.distance(model2), type = "h")
```



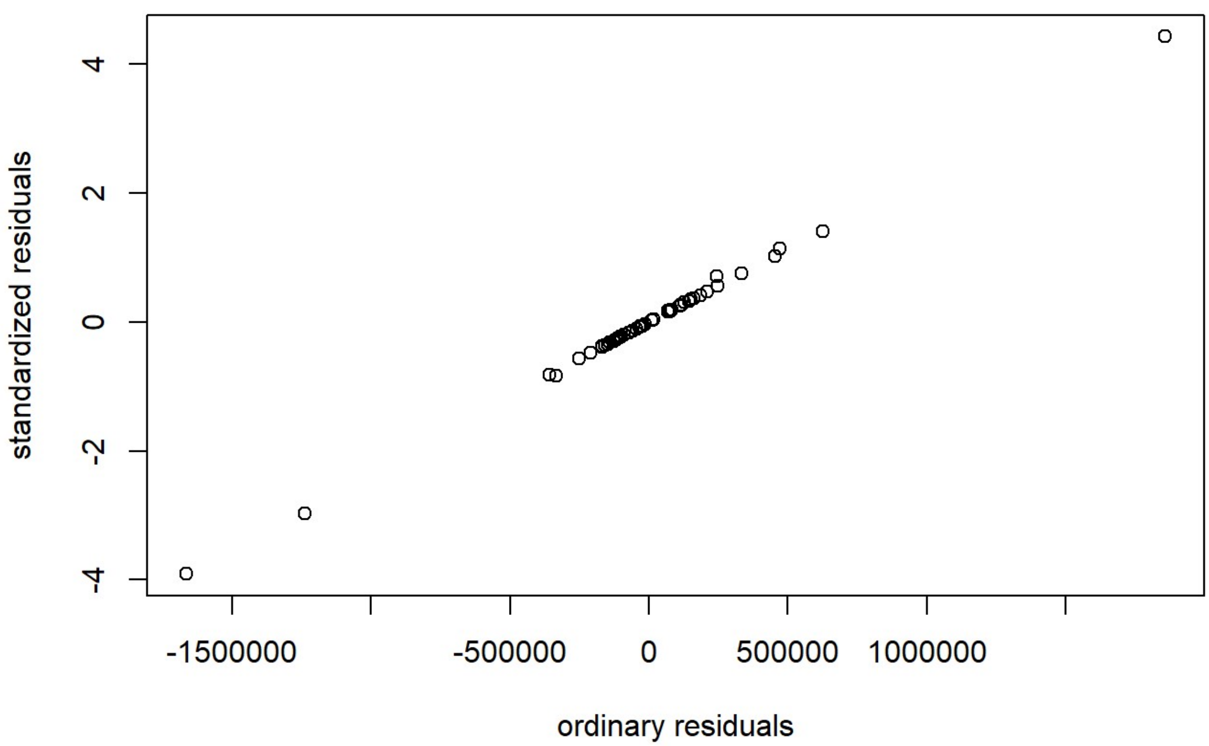
2g) Based on the plot above, there exist a value that is greater than 1 thus there is a influential point.

3a)

```
data("fuel2001")
head(fuel2001)
```

```
##      Drivers    FuelC Income  Miles      MPC      Pop  Tax
## AL   3559897  2382507  23471  94440 12737.00  3451586 18.0
## AK    472211   235400  30064  13628  7639.16   457728  8.0
## AZ   3550367  2428430  25578  55245  9411.55  3907526 18.0
## AR   1961883  1358174  22257  98132 11268.40  2072622 21.7
## CA  21623793 14691753  32275 168771  8923.89 25599275 18.0
## CO   3287922  2048664  32949  85854  9722.73  3322455 22.0
```

```
model3 = lm(formula = FuelC ~ Tax + Drivers + Income, data = fuel2001)
y = rstandard(model3)
x = model3$residuals
par(mfrow=c(1,1))
plot(x, y, xlab="ordinary residuals",ylab="standardized residuals")
```



3b) Points in the plot do not exactly fall on a straightline indicates there might be an error in the independent and identically distributed normal.

3c)

```r
rstudent(model3)
```

```
##          AL          AK          AZ          AR          CA          CO
## -0.46986746 -0.82491434 -0.22552020 -0.26084510  0.70545119 -0.05630035
##          CT          DE          DC          FL          GA          HI
##  0.19227116  0.24295695  0.30391038 -3.26347861  1.14716565 -0.56496830
##          ID          IL          IN          IA          KS          KY
## -0.06721422 -0.37602240  0.41402139  0.15750910 -0.10053959  0.04140851
##          LA          ME          MD          MA          MI          MN
##  0.36507551 -0.23761917  0.75825045 -0.29709900  0.46684060  1.42180335
##          MS          MO          MT          NE          NV          NH
## -0.07658889  0.55145654 -0.06313120  0.02320195  0.26378418  0.25899886
##          NJ          NM          NY          NC          ND          OH
##  0.16981675 -0.38521666 -4.70659975  0.32209073 -0.16259656 -0.31755855
##          OK          OR          PA          RI          SC          SD
##  0.15315873 -0.35581807 -0.82241372  0.34623857  0.17893242 -0.03417366
##          TN          TX          UT          VT          VA          WA
## -0.12936141  5.74349084 -0.25316822 -0.19927711  1.02258646 -0.22841001
##          WV          WI          WY
## -0.33722937  0.18520584 -0.28225433
```

```r
values = qt(0.05, df = df.residual(model3) - 1, lower = FALSE)
studentized = rstudent(model3)
which(abs(studentized) > values)
```

```
## FL NY TX
## 10 33 44
```

3d) FL, NY, TX states are considered as outliers.

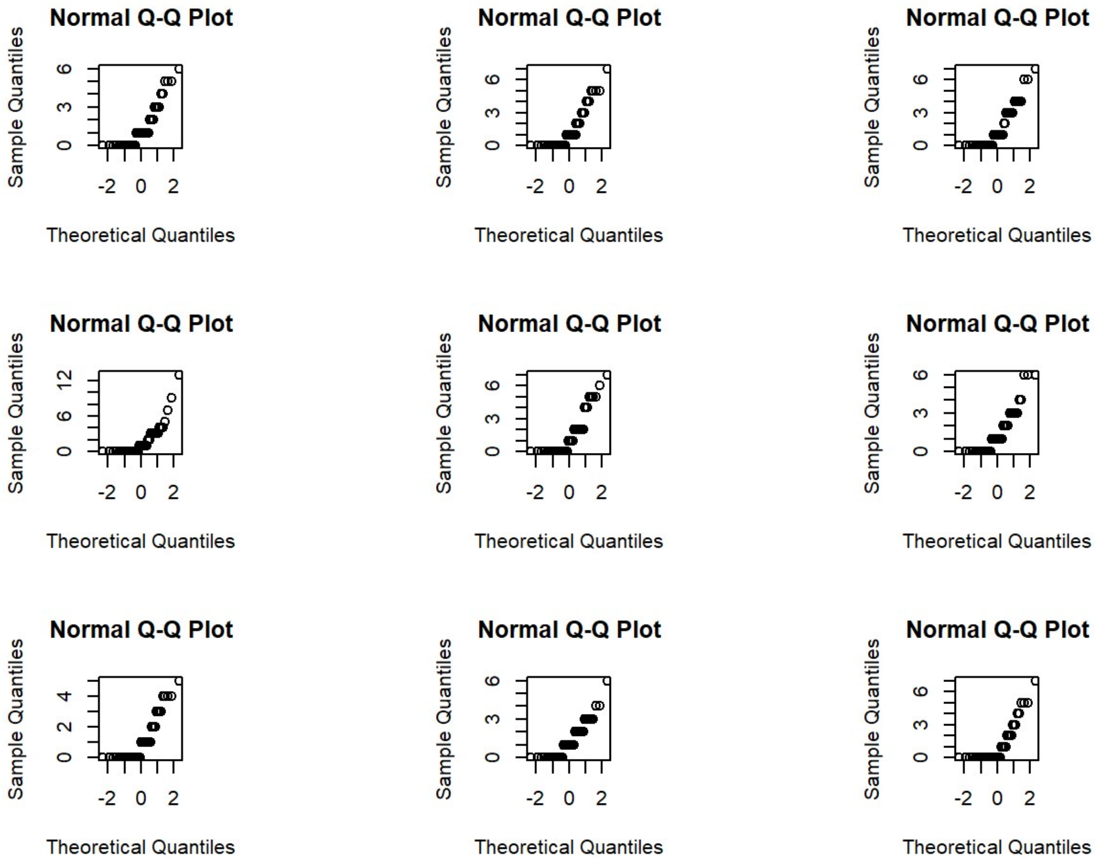
```r
values2 = qt(0.05/(2*nobs(model3)), df = df.residual(model3) - 1, lower = FALSE)
which(abs(studentized) > values2)
```

```
## NY TX
## 33 44
```

3e) NY, TX states are considered as outliers.

4a)

```r
par(pty = "s")
par(mfrow = c(3,3))
for(i in 1:9){qqnorm(rgeom(50,0.4))}
```



4b) What makes the above plots different with normally-distributed data:

First difference: The shape is kind of curvy.

Second difference: All plots have gap between the points which means it is step graphs.