R output for A2

Question 2. a)

```
y = c(5.5,5.9,6.5,5.9,8,9,10,10.8)
X = matrix(c(rep(1,8),7.2,10,9,5.5,9,9.8,14.5,8,8.7,9.4,
             10,9,12,11,12,13.7,5.5,4.4,4,7,5,6.2,5.8,3.9),8,4)
b = solve(t(X)%*%X,t(X)%*%y)
e = y - X%*%b
n = dim(X)[1]
p = dim(X)[2]
s2 = sum(e^2)/(n-p)

b
```

```
##            [,1]
## [1,] -7.4044796
## [2,]  0.1207646
## [3,]  1.1174846
## [4,]  0.3861206
```

```
s2
```

```
## [1] 0.3955368
```

Question 2. b)

```
c = solve(t(X)%*%X)
c
```

```
##             [,1]         [,2]        [,3]         [,4]
## [1,] 13.49743324 -0.054817613 -0.69854293 -1.029731987
## [2,] -0.05481761  0.024498395 -0.01478859 -0.001937333
## [3,] -0.69854293 -0.014788594  0.06226378  0.031714790
## [4,] -1.02973199 -0.001937333  0.03171479  0.135362495
```

Question 2. c)

```
s = sqrt(sum(e^2)/(n-p))
alpha = 0.01
ta = qt(1-alpha/2, df=(n-p))
t = c(1,8,9,5)
ttb = t(t)%*%b
CI = c(ttb) + c(-1,1)*c(ta*s*sqrt(t(t)%*%solve(t(X)%*%X)%*%t))

CI
```

```
## [1] 3.926075 7.173129
```

Question 2. d)

```
for (alpha in seq(0.01, 0.15, by = 0.0005)) {
  # t_alpha given an alpha value
  ta = qt(1-alpha/2, df=(n-p))
  # Generate Prediction Interval given alpha
  PI = c(ttb) + c(-1,1)*c(ta*s*sqrt(1+t(t)%*%solve(t(X)%*%X)%*%t))
  if (round(PI[1],3) == 4.012 && round(PI[2],3) == 7.087) {
    print(alpha)
```

```r
    print(round(PI,3))
  }
}
```

```
## [1] 0.1
## [1] 4.012 7.087
```

Question 2. e)

```r
SSRes = t(y-X%*%b)%*%(y-X%*%b)
CorrectedSSReg = t(y)%*%X%*%b - sum(y)^2/n
k = 3 # num parameters
Fstat = (CorrectedSSReg/k)/(SSRes/(n-k-1))
Fval = qf(0.95,k,n-k-1)

Fstat
```

```
##          [,1]
## [1,] 23.47683
```

```r
Fval
```

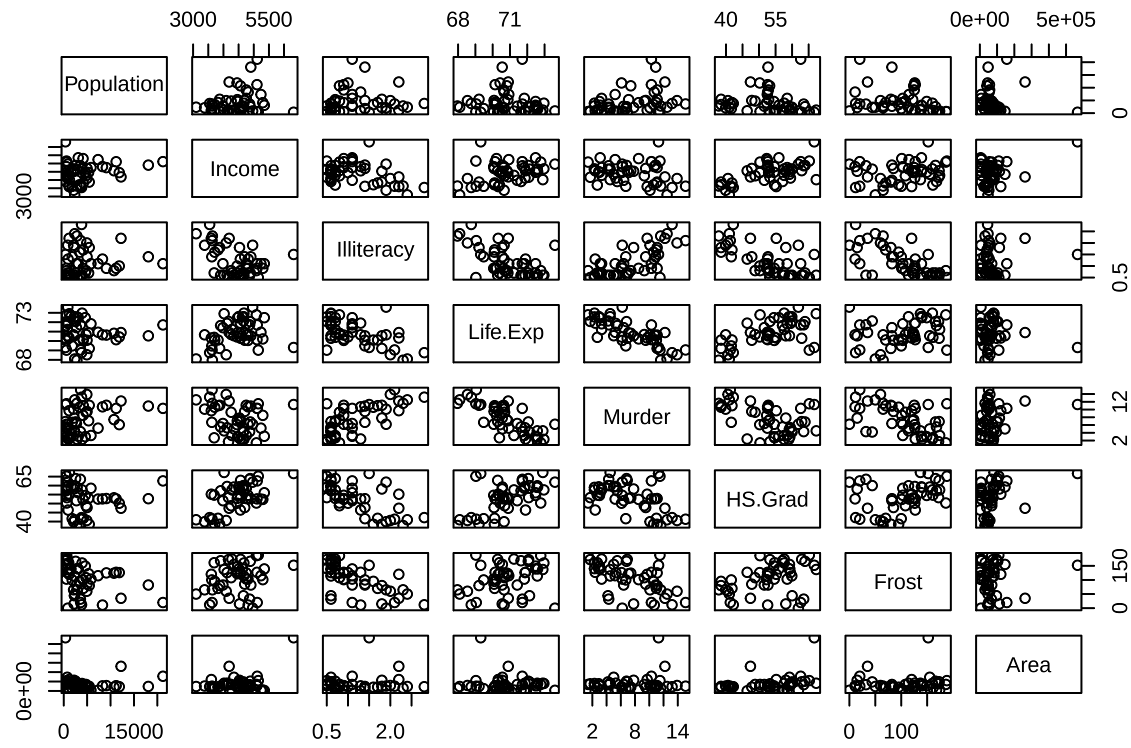
```
## [1] 6.591382
```

```r
Fstat > Fval
```

```
##      [,1]
## [1,] TRUE
```

Question 4. a)

```r
data(state)
statedata <- data.frame(state.x77, row.names=state.abb, check.names=TRUE)

pairs(statedata)
```




```r
statedata$Area = log(statedata$Area) # log Area
statedata$Illiteracy = log(statedata$Illiteracy) # log Illiteracy
```

Question 4. b)

```r
basemodel = lm(Murder ~ 1, data=statedata)

add1(basemodel, scope= ~ . + Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + Area, data=
```

```
## Single term additions
##
## Model:
## Murder ~ 1
##            Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                  667.75 131.59
## Population  1     78.85 588.89 127.31  6.4273 0.0145504 *
## Income      1     35.35 632.40 130.88  2.6829 0.1079683
## Illiteracy  1    322.29 345.46 100.64 44.7810 2.183e-08 ***
## Life.Exp    1    407.14 260.61  86.55 74.9887 2.260e-11 ***
## HS.Grad     1    159.00 508.75 120.00 15.0017 0.0003248 ***
## Frost       1    193.91 473.84 116.44 19.6433 5.405e-05 ***
## Area        1     58.63 609.12 129.00  4.6201 0.0366687 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# we add Life.Exp
model1 = lm(Murder ~ Life.Exp, data=statedata)
add1(model1, scope= ~ . + Population + Income + Illiteracy + HS.Grad + Frost + Area, data=statedata, te
```

```
## Single term additions
##
## Model:
## Murder ~ Life.Exp
##            Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                  260.61 86.550
## Population  1    56.615 203.99 76.303 13.0442 0.0007374 ***
## Income      1     0.958 259.65 88.366  0.1733 0.6790605
## Illiteracy  1    61.648 198.96 75.054 14.5629 0.0003952 ***
## HS.Grad     1     1.124 259.48 88.334  0.2035 0.6539823
## Frost       1    80.104 180.50 70.187 20.8575 3.576e-05 ***
## Area        1    30.223 230.38 82.386  6.1656 0.0166517 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# we add Frost
model2 = lm(Murder ~ Life.Exp + Frost, data=statedata)
add1(model2, scope= ~ . + Population + Income + Illiteracy + HS.Grad +  Area, data=statedata, test="F")
```

```
## Single term additions
##
## Model:
## Murder ~ Life.Exp + Frost
##            Df Sum of Sq    RSS    AIC F value   Pr(>F)
## <none>                  180.50 70.187
## Population  1   23.7098 156.79 65.146  6.9559 0.011358 *
## Income      1    5.5598 174.94 70.622  1.4619 0.232807
## Illiteracy  1    6.4775 174.03 70.359  1.7122 0.197204
## HS.Grad     1    2.0679 178.44 71.610  0.5331 0.469015
## Area        1   30.9733 149.53 62.774  9.5283 0.003422 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# we add Area
model3 = lm(Murder ~ Life.Exp + Frost + Area, data=statedata)
add1(model3, scope= ~ . + Population + Income + Illiteracy + HS.Grad, data=statedata, test="F")
```

```
## Single term additions
##
## Model:
## Murder ~ Life.Exp + Frost + Area
##            Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                  149.53 62.774
## Population  1    16.347 133.18 58.985  5.5235 0.02321 *
## Income      1     4.786 144.75 63.147  1.4879 0.22889
## Illiteracy  1    13.479 136.05 60.050  4.4584 0.04032 *
## HS.Grad     1     0.190 149.34 64.710  0.0572 0.81200
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# we add Population
model4 = lm(Murder ~ Life.Exp + Frost + Area + Population, data=statedata)
add1(model4, scope= ~ . + Income + Illiteracy + HS.Grad, data=statedata, test="F")
```

```
## Single term additions
```

```
##
## Model:
## Murder ~ Life.Exp + Frost + Area + Population
##            Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                  133.18 58.985
## Income      1    0.9201 132.26 60.639  0.3061 0.58289
## Illiteracy  1   14.2593 118.92 55.323  5.2757 0.02644 *
## HS.Grad     1    0.0829 133.10 60.954  0.0274 0.86929
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# we add Illiteracy
model5 = lm(Murder ~ Life.Exp + Frost + Area + Population + Illiteracy, data=statedata)
add1(model5, scope= ~ . + Income + HS.Grad, data=statedata, test="F")
```

```
## Single term additions
##
## Model:
## Murder ~ Life.Exp + Frost + Area + Population + Illiteracy
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>               118.92 55.323
## Income   1    2.2064 116.72 56.387  0.8129 0.3723
## HS.Grad  1    2.0227 116.90 56.465  0.7440 0.3932
```

```
# Our final model keeps Life.Exp, Frost, Area, Population, Illiteracy

model5
```

```
##
## Call:
## lm(formula = Murder ~ Life.Exp + Frost + Area + Population +
##     Illiteracy, data = statedata)
##
## Coefficients:
## (Intercept)     Life.Exp        Frost         Area   Population
##   1.104e+02   -1.550e+00   -1.173e-02    6.936e-01    1.422e-04
##  Illiteracy
##   1.785e+00
```

Question 4. c)

```
fullmodel = lm(Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + Area, data=state
model = step(fullmodel, scope= ~ . + Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + Ar
```

```
## Start:  AIC=58.2
## Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad +
##     Frost + Area
##
##              Df Sum of Sq    RSS    AIC
## - HS.Grad     1     0.432 116.72 56.387
## - Income      1     0.616 116.90 56.465
## <none>                    116.29 58.201
## - Frost       1     8.555 124.84 59.751
## - Population  1    12.255 128.54 61.211
## - Illiteracy  1    14.806 131.09 62.194
## - Area        1    23.755 140.04 65.496
## - Life.Exp    1   124.645 240.93 92.624
```

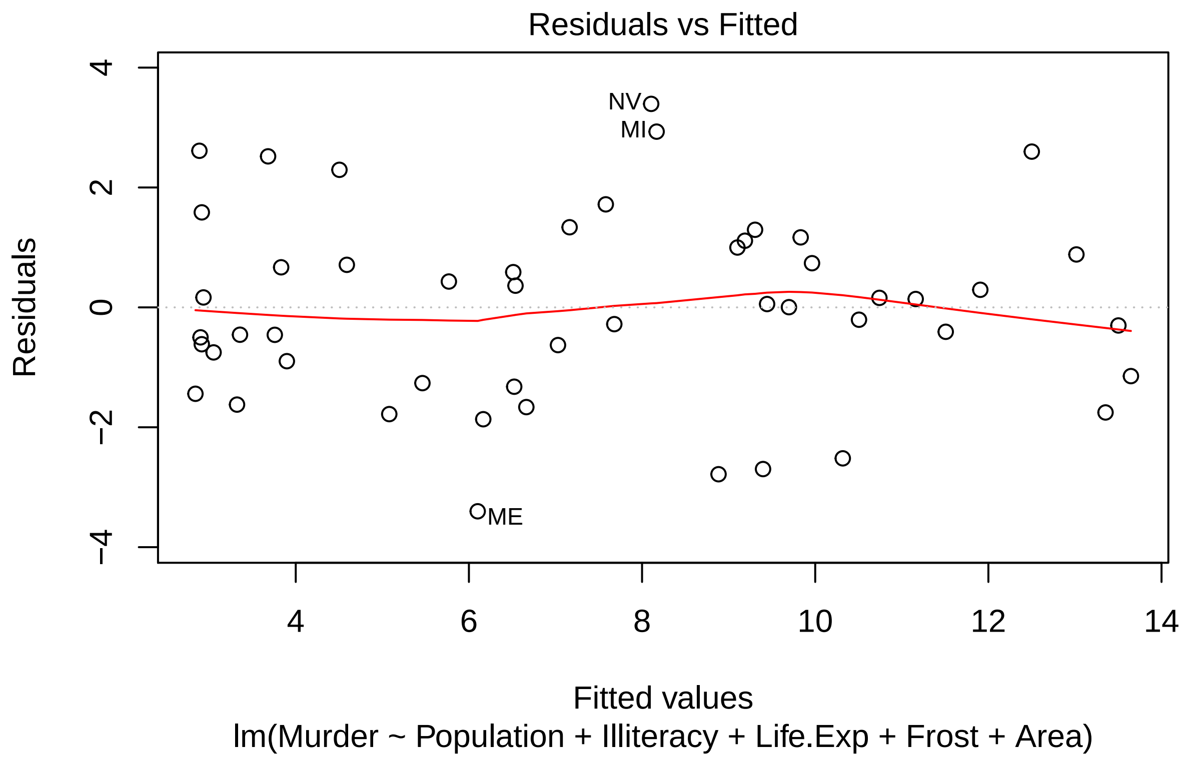
```
##
## Step:  AIC=56.39
## Murder ~ Population + Income + Illiteracy + Life.Exp + Frost +
##     Area
##
##              Df Sum of Sq    RSS    AIC
## - Income      1     2.206 118.92 55.323
## <none>                    116.72 56.387
## + HS.Grad     1     0.432 116.29 58.201
## - Frost       1     9.542 126.26 58.316
## - Population  1    11.960 128.68 59.264
## - Illiteracy  1    15.546 132.26 60.639
## - Area        1    30.621 147.34 66.035
## - Life.Exp    1   133.825 250.54 92.580
##
## Step:  AIC=55.32
## Murder ~ Population + Illiteracy + Life.Exp + Frost + Area
##
##              Df Sum of Sq    RSS    AIC
## <none>                    118.92 55.323
## + Income      1     2.206 116.72 56.387
## + HS.Grad     1     2.023 116.90 56.465
## - Frost       1     8.663 127.59 56.839
## - Illiteracy  1    14.259 133.18 58.985
## - Population  1    17.127 136.05 60.050
## - Area        1    29.940 148.86 64.551
## - Life.Exp    1   132.043 250.97 90.665
```

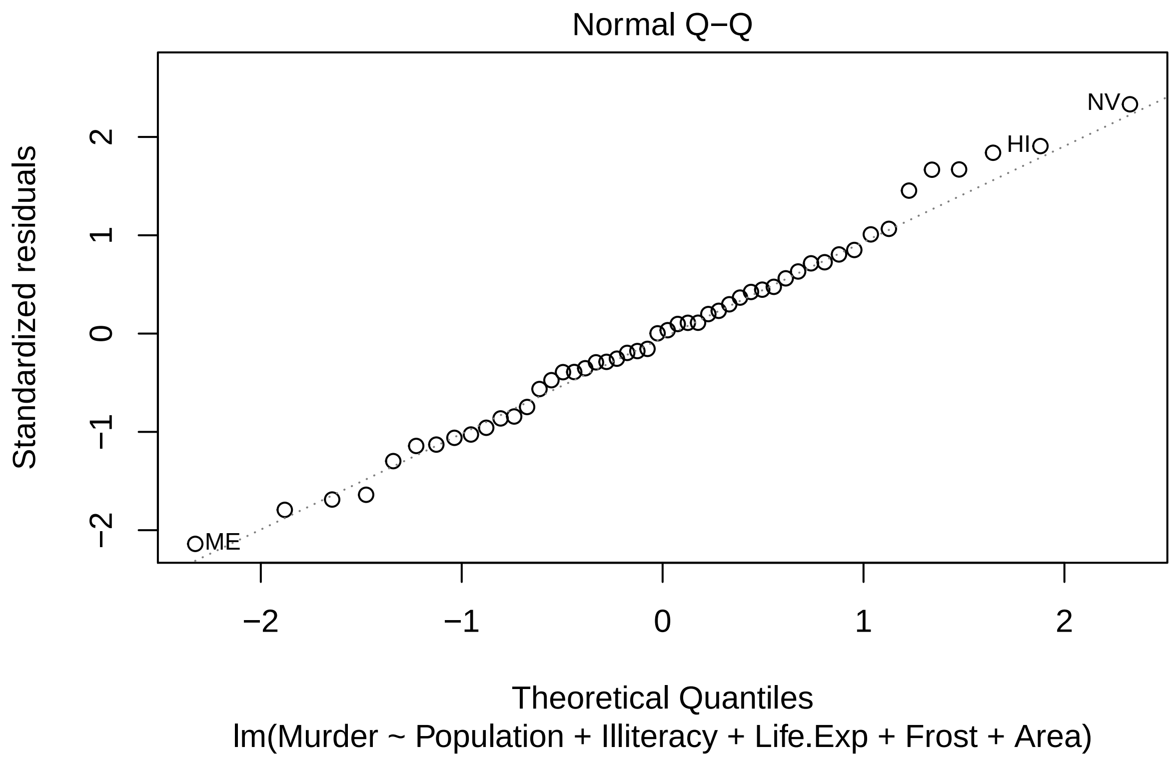
```
# Our final model keeps Life.Exp, Area, Illiteracy, Population, Frost at a significance level of alpha=

model
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + Life.Exp + Frost +
##     Area, data = statedata)
##
## Coefficients:
## (Intercept)   Population   Illiteracy     Life.Exp        Frost
##   1.104e+02    1.422e-04    1.785e+00   -1.550e+00   -1.173e-02
##        Area
##   6.936e-01
```

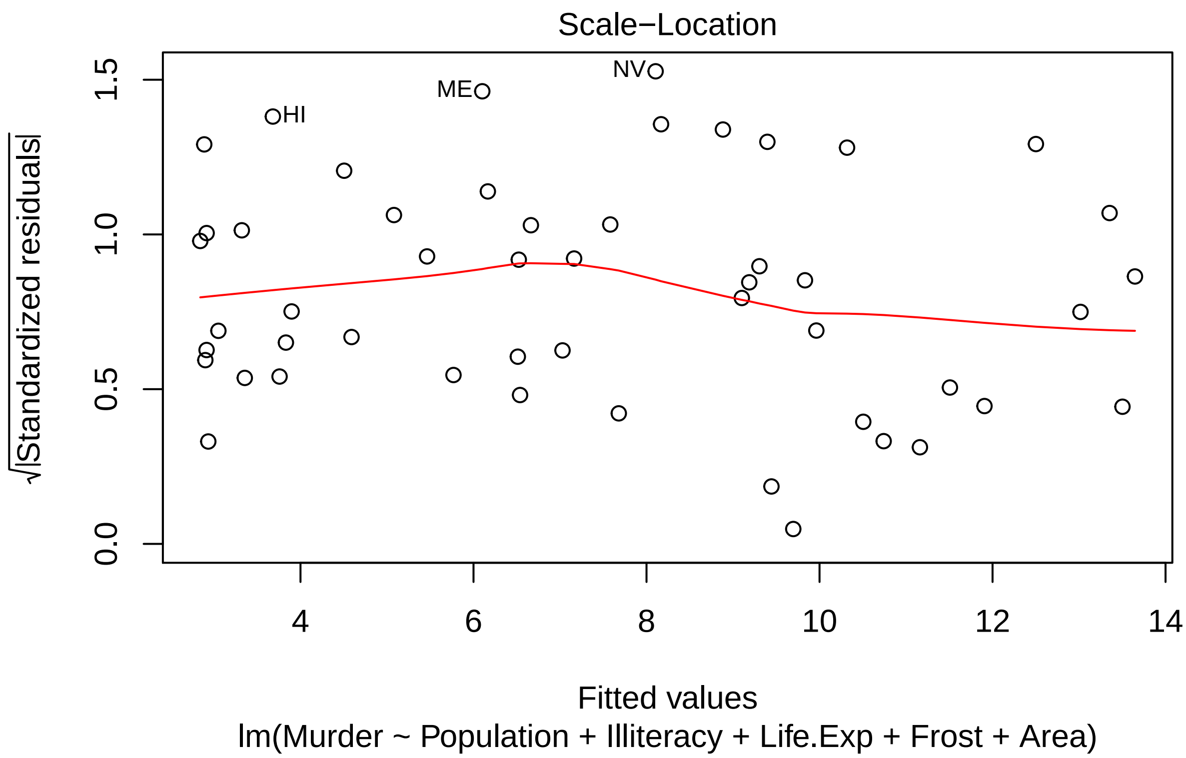
Question 4. e)

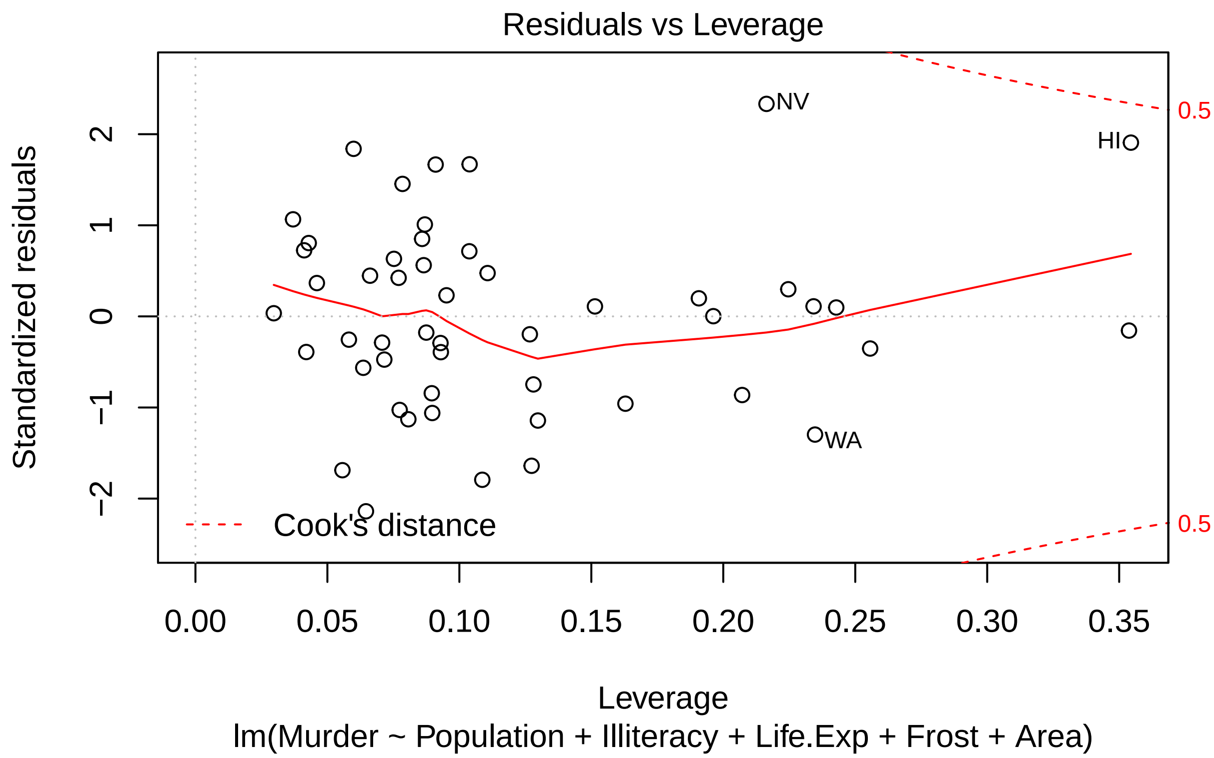
```
plot(model)
```


Residuals vs Fitted
Residuals
4
2
0
−2
−4
NV
MI
ME
4
6
8
10
12
14
Fitted values
lm(Murder ~ Population + Illiteracy + Life.Exp + Frost + Area)



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Murder ~ Population + Illiteracy + Life.Exp + Frost + Area)
NV
HI
ME
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
HI
ME
NV
4
6
8
10
12
14
Fitted values
lm(Murder ~ Population + Illiteracy + Life.Exp + Frost + Area)



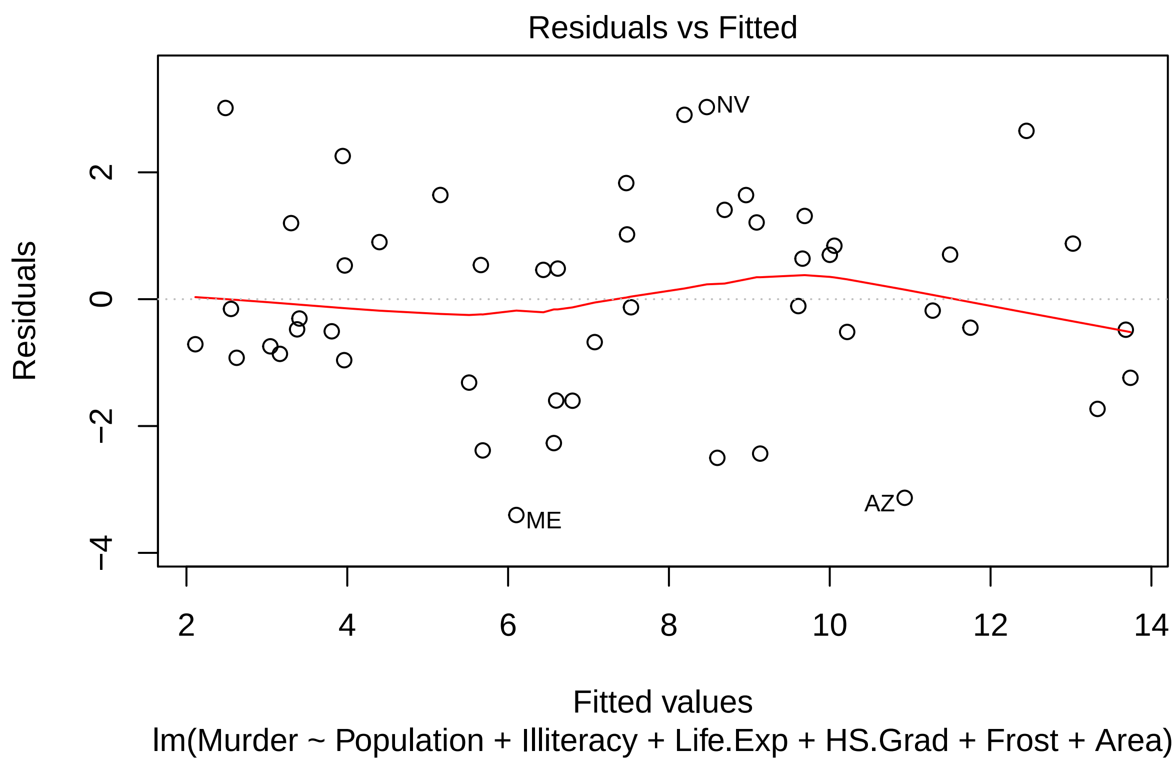
Compare this to another final model that also used a log transformation on Population.

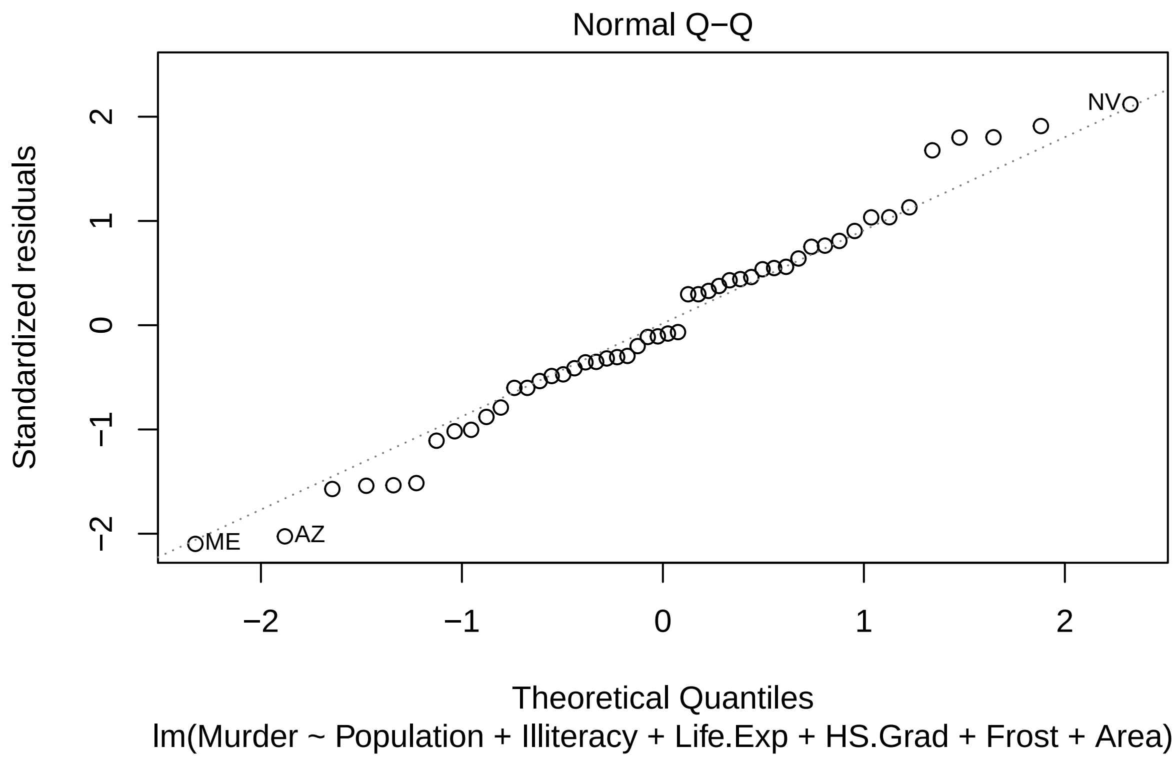
```
statedata$Population = log(statedata$Population)

fullmodel = lm(Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + Area, data=state
model = step(fullmodel, scope= ~ . + Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + Are
```

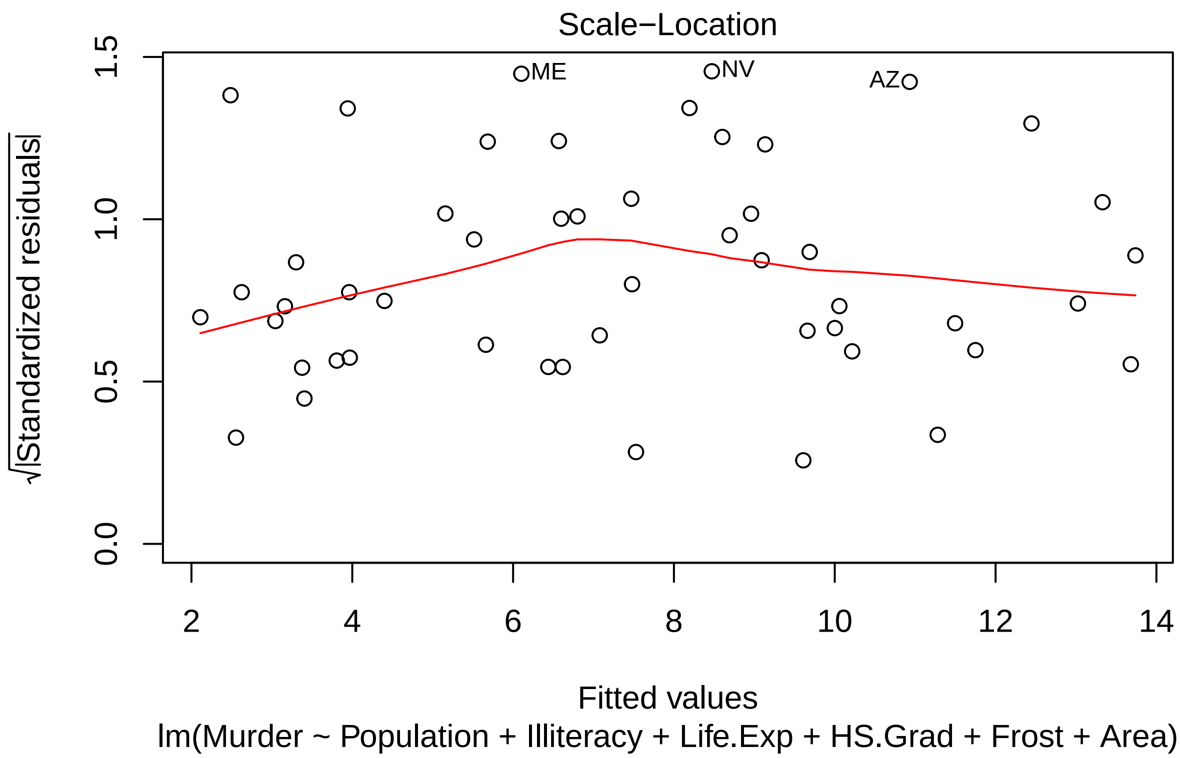
```
## Start:  AIC=59.82
## Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad +
##     Frost + Area
##
##              Df Sum of Sq    RSS    AIC
## - Income      1     0.991 121.11 58.233
## - HS.Grad     1     1.219 121.34 58.327
## <none>                    120.12 59.822
## - Frost       1     6.267 126.38 60.365
## - Population  1     8.424 128.54 61.211
## - Illiteracy  1    16.539 136.66 64.272
## - Area        1    24.459 144.57 67.089
## - Life.Exp    1   127.765 247.88 94.046
##
## Step:  AIC=58.23
## Murder ~ Population + Illiteracy + Life.Exp + HS.Grad + Frost +
##     Area
##
##              Df Sum of Sq    RSS    AIC
## <none>                    121.11 58.233
```

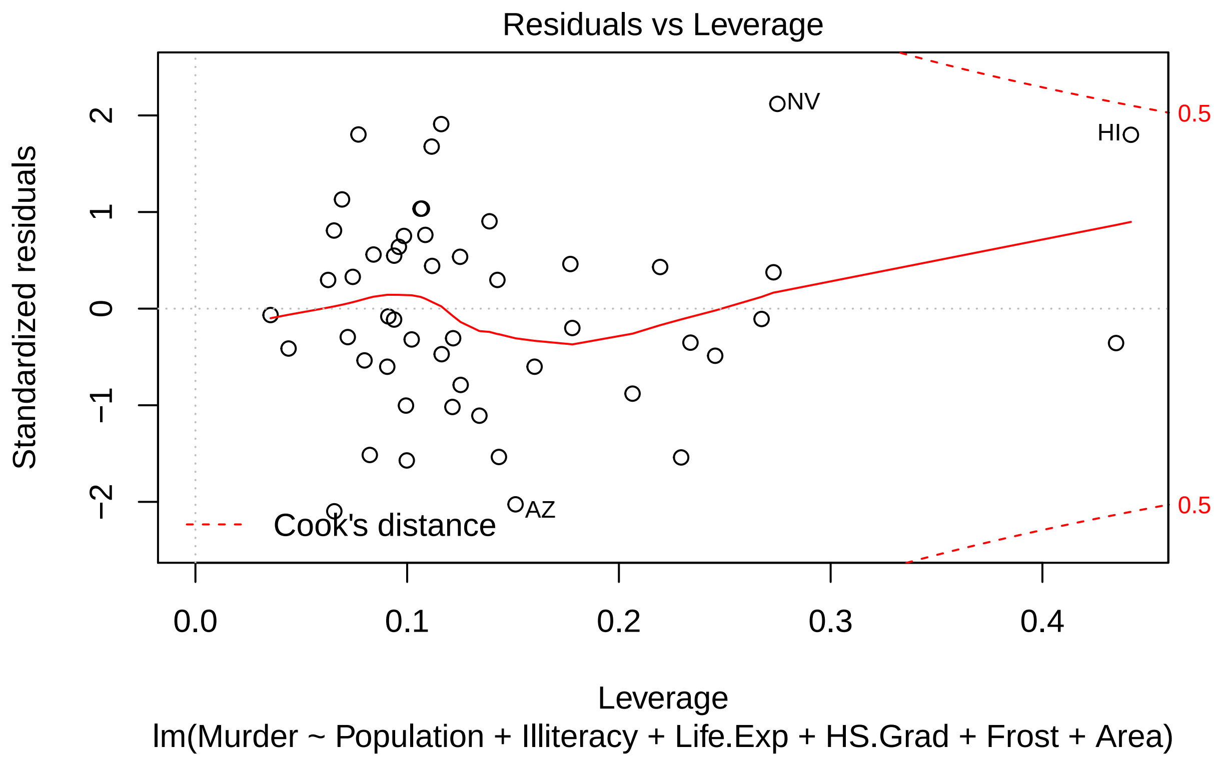
```
## - HS.Grad      1     5.275 126.38 58.364
## - Frost        1     5.426 126.53 58.424
## + Income       1     0.991 120.12 59.822
## - Population   1    13.493 134.60 61.514
## - Illiteracy   1    19.123 140.23 63.563
## - Area         1    23.594 144.70 65.132
## - Life.Exp     1   131.223 252.33 92.936
```

```r
plot(model)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
NV
ME
AZ
−2
−1
0
1
2
Theoretical Quantiles
lm(Murder ~ Population + Illiteracy + Life.Exp + HS.Grad + Frost + Area)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
4
6
8
10
12
14
ME
NV
AZ
Fitted values
lm(Murder ~ Population + Illiteracy + Life.Exp + HS.Grad + Frost + Area)



Not taking a transformation is arguably better even though it looks like it should have. Residuals vs Fitted are not as spread, QQ-Plot suggests that the tails follow another distribution, Scale-Location seems to have a negative quadratic trend and the Residuals vs Leverage has points with much larger leverage compared to the previous final model.

Question 5. b)

```
Xscaled = scale(X[,-1]) # No intercept parameter (Piazza)
yscaled = scale(y, scale=FALSE) # Only centering, no scale (Piazza)
r = dim(t(Xscaled)%*%Xscaled)
lambda = diag(0.5, r)
b = solve(t(Xscaled)%*%Xscaled + lambda, t(Xscaled)%*%yscaled)

b
```

```
##            [,1]
## [1,] 0.3494789
## [2,] 1.7899861
## [3,] 0.3432961
```

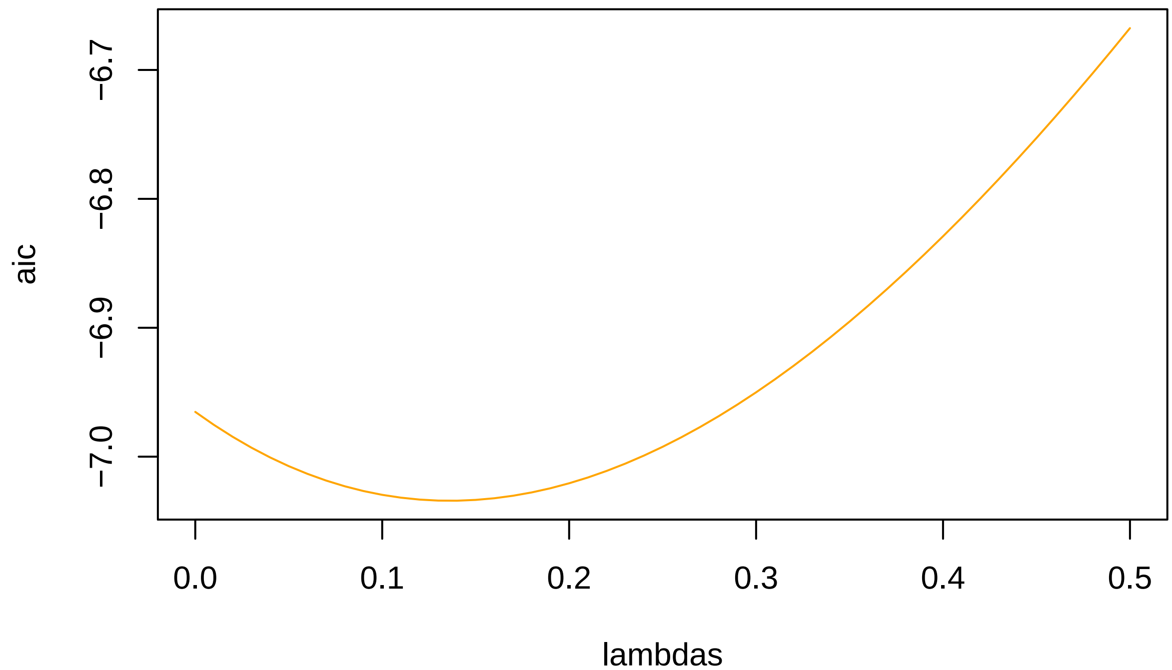
Question 5. c)

```
library(matrixcalc)
```

```
## Warning: package 'matrixcalc' was built under R version 3.5.2
```

```
aic = c()
lambdas = seq(0, 0.5, by=0.01)

for (i in lambdas){
```

```r
  lambda = diag(i, r)

  ridgeb = solve(t(Xscaled)%*%Xscaled + lambda, t(Xscaled)%*%yscaled)
  SSRes = t(yscaled-Xscaled%*%ridgeb)%*%(yscaled-Xscaled%*%ridgeb)

  H = Xscaled%*%solve(t(Xscaled)%*%Xscaled + lambda)%*%t(Xscaled)

  aic = c(aic, n*log(SSRes/n) + 2*matrix.trace(H))
}

plot(lambdas, aic, col='orange', type='l')
```




```r
lambda_aic = lambdas[which.min(aic)]

lambda_aic
```

```
## [1] 0.14
```