S20 PSTAT126 Final Project

Karen Zhao

6/6/2020

Regression Analysis on U.S. Life Expectancy

1. Introduction

This project focuses on studying the prediction of life expectancy in the U.S. states based on the dataset 'state.x77' in R library, which is derived from the U.S. Department of Commerce,Bureau of the Census (1977) Statistical Abstract of the United States. We will examine the effects of the following 7 variables on life expectancy: population, income, illiteracy, murder rate(Murder), high school graduate rate (HS Grad), land area, and mean number of days with minimum temperature below freezing (Frost). We find that 'Murder', 'HS Grad', 'Frost', and "Population' are the most related predictors.

```
dat=as.data.frame(state.x77)
attach(dat)
names(dat)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"    "Frost"      "Area"
```

2. Questions of Interest

Can we predict life expectancy of a region given its population, income, illiteracy, murder rate (Murder), high school graduate percent (HS Grad), land area, and mean number of days with minimum temperature below freezing (Frost) as predictors?

3. Regression Method

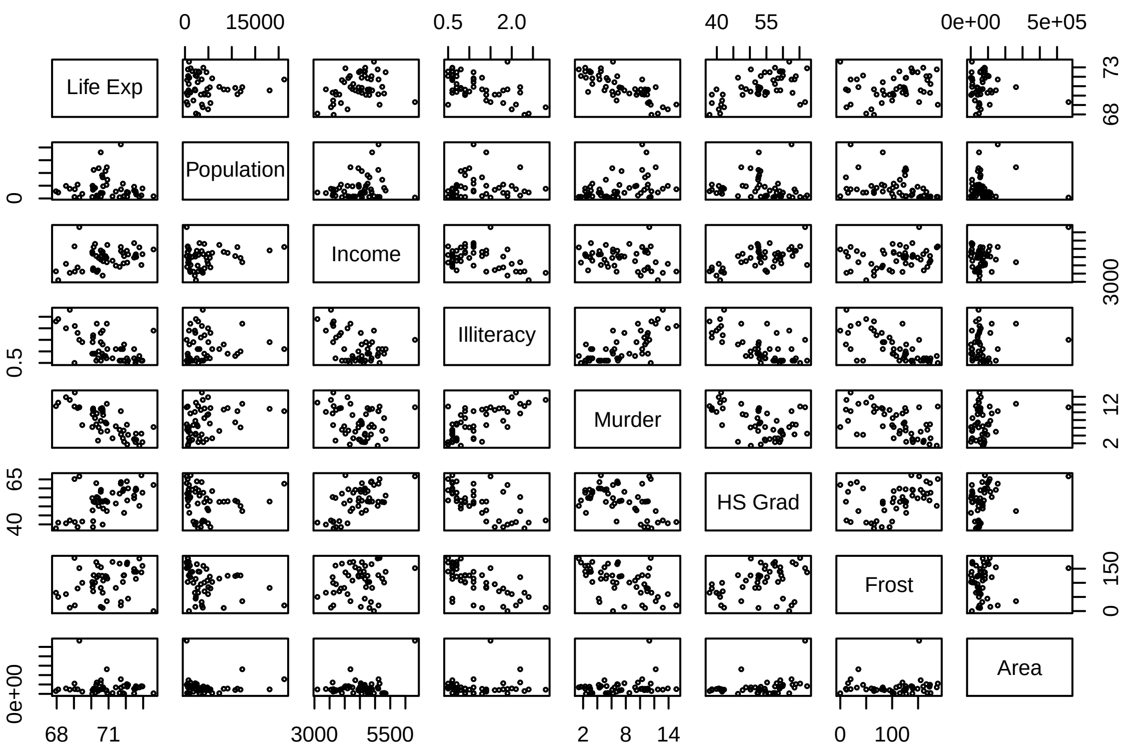
We will approach this question first by applying stepwise and best subsets regression on the 7 potential predictors to determine the best model. Then we will check LINE conditions on this model using residual analysis. If any of the assumptions are not met, we will transform the data and check LINE conditions for the new model. After fitting the model with transformed data, we will interpret our model and summarize our findings.

4. Regression Analysis, Results and Interpretation

Variable Selection

First, we look at the scatterplot matrix to gain some insight on the relationships between the variables in the data. From the scatterplot below, we can tell that there are some predictors like 'Murder' seems to be strongly related to 'Life Expectancy'. Others like 'Area' and 'Income' seem to be moderately or weakly related.

```
pairs(dat[c(4,1,2,3,5,6,7,8)], cex=0.4) #scatterplot matrix
```




```
cor(dat)
```

```
##             Population     Income  Illiteracy    Life Exp     Murder
## Population  1.00000000  0.2082276  0.10762237 -0.06805195  0.3436428
## Income      0.20822756  1.0000000 -0.43707519  0.34025534 -0.2300776
## Illiteracy  0.10762237 -0.4370752  1.00000000 -0.58847793  0.7029752
## Life Exp   -0.06805195  0.3402553 -0.58847793  1.00000000 -0.7808458
## Murder      0.34364275 -0.2300776  0.70297520 -0.78084575  1.0000000
## HS Grad    -0.09848975  0.6199323 -0.65718861  0.58221620 -0.4879710
## Frost      -0.33215245  0.2262822 -0.67194697  0.26206801 -0.5388834
## Area        0.02254384  0.3633154  0.07726113 -0.10733194  0.2283902
##                HS Grad      Frost        Area
## Population -0.09848975 -0.3321525  0.02254384
## Income      0.61993232  0.2262822  0.36331544
```

```
## Illiteracy -0.65718861 -0.6719470  0.07726113
## Life Exp    0.58221620  0.2620680 -0.10733194
## Murder     -0.48797102 -0.5388834  0.22839021
## HS Grad     1.00000000  0.3667797  0.33354187
## Frost       0.36677970  1.0000000  0.05922910
## Area        0.33354187  0.0592291  1.00000000
```

Secondly, we perform variable selection using stepwise regression, including AIC and partial F test, and the best subsets regression to determine the predictors. The results of our AIC test, partial F test, and adjusted R2 criterion chooses four predictors: "Murder", "HS Grad", "Frost", and "Population" . The Mallows' Cp criterion gives similar result except excluding the fourth predictor "Population". Therefore, We decide our model to be `Life Exp` ~ Murder + `HS Grad` + Frost + Population.

```
# Stepwise regression using AIC
mod0=lm(`Life Exp`~1)
mod.all = lm(`Life Exp`~., data=dat) # including all predictors in lm()
step(mod0, scope = list(lower = mod0, upper = mod.all))
```

```
## Start:  AIC=30.44
## `Life Exp` ~ 1
##
##                Df Sum of Sq    RSS     AIC
## + Murder        1    53.838 34.461 -14.609
## + Illiteracy    1    30.578 57.721  11.179
## + `HS Grad`     1    29.931 58.368  11.737
## + Income        1    10.223 78.076  26.283
## + Frost         1     6.064 82.235  28.878
## <none>                      88.299  30.435
## + Area          1     1.017 87.282  31.856
## + Population    1     0.409 87.890  32.203
##
## Step:  AIC=-14.61
## `Life Exp` ~ Murder
##
##                Df Sum of Sq    RSS     AIC
## + `HS Grad`     1     4.691 29.770 -19.925
## + Population    1     4.016 30.445 -18.805
## + Frost         1     3.135 31.327 -17.378
## + Income        1     2.405 32.057 -16.226
## <none>                      34.461 -14.609
## + Area          1     0.470 33.992 -13.295
## + Illiteracy    1     0.273 34.188 -13.007
## - Murder        1    53.838 88.299  30.435
##
## Step:  AIC=-19.93
## `Life Exp` ~ Murder + `HS Grad`
##
##                Df Sum of Sq    RSS     AIC
## + Frost         1    4.3987 25.372 -25.920
## + Population    1    3.3405 26.430 -23.877
## <none>                      29.770 -19.925
## + Illiteracy    1    0.4419 29.328 -18.673
## + Area          1    0.2775 29.493 -18.394
```

```
## + Income      1    0.1022 29.668 -18.097
## - `HS Grad`   1    4.6910 34.461 -14.609
## - Murder      1   28.5974 58.368  11.737
##
## Step:  AIC=-25.92
## `Life Exp` ~ Murder + `HS Grad` + Frost
##
##               Df Sum of Sq    RSS     AIC
## + Population  1     2.064 23.308 -28.161
## <none>                    25.372 -25.920
## + Income      1     0.182 25.189 -24.280
## + Illiteracy  1     0.172 25.200 -24.259
## + Area        1     0.026 25.346 -23.970
## - Frost       1     4.399 29.770 -19.925
## - `HS Grad`   1     5.955 31.327 -17.378
## - Murder      1    32.756 58.128  13.531
##
## Step:  AIC=-28.16
## `Life Exp` ~ Murder + `HS Grad` + Frost + Population
##
##               Df Sum of Sq    RSS     AIC
## <none>                    23.308 -28.161
## + Income      1     0.006 23.302 -26.174
## + Illiteracy  1     0.004 23.304 -26.170
## + Area        1     0.001 23.307 -26.163
## - Population  1     2.064 25.372 -25.920
## - Frost       1     3.122 26.430 -23.877
## - `HS Grad`   1     5.112 28.420 -20.246
## - Murder      1    34.816 58.124  15.528
```

```
##
## Call:
## lm(formula = `Life Exp` ~ Murder + `HS Grad` + Frost + Population)
##
## Coefficients:
## (Intercept)       Murder    `HS Grad`        Frost   Population
##   7.103e+01   -3.001e-01    4.658e-02   -5.943e-03    5.014e-05
```

```r
mod.AIC = lm(`Life Exp` ~ Murder + `HS Grad` + Frost + Population, data=dat)

# Stepwise regression using F-test
mod0=lm(`Life Exp`~1)
add1(mod0, ~.+Population+Income+Illiteracy+Murder+`HS Grad`+Frost+Area, test = 'F')
```

```
## Single term additions
##
## Model:
## `Life Exp` ~ 1
##            Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                  88.299 30.435
## Population  1     0.409 87.890 32.203  0.2233   0.63866
## Income      1    10.223 78.076 26.283  6.2847   0.01562 *
## Illiteracy  1    30.578 57.721 11.179 25.4289 6.969e-06 ***
```

```
## Murder      1    53.838 34.461 -14.609 74.9887 2.260e-11 ***
## `HS Grad`   1    29.931 58.368  11.737 24.6146 9.196e-06 ***
## Frost       1     6.064 82.235  28.878  3.5397   0.06599 .
## Area        1     1.017 87.282  31.856  0.5594   0.45815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#choose Murder, which has the smallest p-value or largest F-statistic
mod1 = update(mod0, ~.+Murder)
add1(mod1, ~.+Population+Income+Illiteracy+`HS Grad`+Frost+Area, test = 'F')
```

```
## Single term additions
##
## Model:
## `Life Exp` ~ Murder
##             Df Sum of Sq    RSS     AIC F value   Pr(>F)
## <none>                   34.461 -14.609
## Population  1    4.0161 30.445 -18.805  6.1999 0.016369 *
## Income      1    2.4047 32.057 -16.226  3.5257 0.066636 .
## Illiteracy  1    0.2732 34.188 -13.007  0.3756 0.542910
## `HS Grad`   1    4.6910 29.770 -19.925  7.4059 0.009088 **
## Frost       1    3.1346 31.327 -17.378  4.7029 0.035205 *
## Area        1    0.4697 33.992 -13.295  0.6494 0.424375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#choose HS Grad, which has the smallest p-value or largest F-statistic
mod2 = update(mod1, ~.+`HS Grad`)
#check if Murder is still significant after adding HS Grad
summary(mod2)
```

```
##
## Call:
## lm(formula = `Life Exp` ~ Murder + `HS Grad`)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66758 -0.41801  0.05602  0.55913  2.05625
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.29708    1.01567  69.213  < 2e-16 ***
## Murder      -0.23709    0.03529  -6.719 2.18e-08 ***
## `HS Grad`    0.04389    0.01613   2.721  0.00909 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7959 on 47 degrees of freedom
## Multiple R-squared:  0.6628, Adjusted R-squared:  0.6485
## F-statistic:  46.2 on 2 and 47 DF,  p-value: 8.016e-12
```

```r
#both predictors have very small p-value: significant
add1(mod2, ~.+Population+Income+Illiteracy+Frost+Area, test = 'F')
```

```
## Single term additions
##
## Model:
## `Life Exp` ~ Murder + `HS Grad`
##            Df Sum of Sq    RSS     AIC F value   Pr(>F)
## <none>                  29.770 -19.925
## Population  1    3.3405 26.430 -23.877  5.8141 0.019949 *
## Income      1    0.1022 29.668 -18.097  0.1585 0.692418
## Illiteracy  1    0.4419 29.328 -18.673  0.6931 0.409421
## Frost       1    4.3987 25.372 -25.920  7.9751 0.006988 **
## Area        1    0.2775 29.493 -18.394  0.4329 0.513863
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#choose Frost, which has the smallest p-value or largest F-statistic
mod3 = update(mod2, ~.+Frost)
#check if Murder and HS Grad are still significant after adding Frost
summary(mod3)
```

```
##
## Call:
## lm(formula = `Life Exp` ~ Murder + `HS Grad` + Frost)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5015 -0.5391  0.1014  0.5921  1.2268
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 71.036379   0.983262  72.246  < 2e-16 ***
## Murder      -0.283065   0.036731  -7.706 8.04e-10 ***
## `HS Grad`    0.049949   0.015201   3.286  0.00195 **
## Frost       -0.006912   0.002447  -2.824  0.00699 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7427 on 46 degrees of freedom
## Multiple R-squared:  0.7127, Adjusted R-squared:  0.6939
## F-statistic: 38.03 on 3 and 46 DF,  p-value: 1.634e-12
```

```r
#all predictors have very small p-value: significant
add1(mod3, ~.+Population+Income+Illiteracy+Area, test = 'F')
```

```
## Single term additions
##
## Model:
## `Life Exp` ~ Murder + `HS Grad` + Frost
##            Df Sum of Sq    RSS     AIC F value  Pr(>F)
## <none>                  25.372 -25.920
```

```
## Population  1   2.06358 23.308 -28.161  3.9841 0.05201 .
## Income      1   0.18232 25.189 -24.280  0.3257 0.57103
## Illiteracy  1   0.17184 25.200 -24.259  0.3069 0.58236
## Area        1   0.02573 25.346 -23.970  0.0457 0.83173
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#choose Pop, which has the smallest p-value or largest F-statistic
mod4 = update(mod3, ~.+Population)
#check if Murder, HS Grad, and Frost are still significant after adding Pop
summary(mod4)
```

```
##
## Call:
## lm(formula = `Life Exp` ~ Murder + `HS Grad` + Frost + Population)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47095 -0.53464 -0.03701  0.57621  1.50683
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.103e+01  9.529e-01  74.542  < 2e-16 ***
## Murder      -3.001e-01  3.661e-02  -8.199 1.77e-10 ***
## `HS Grad`    4.658e-02  1.483e-02   3.142  0.00297 **
## Frost       -5.943e-03  2.421e-03  -2.455  0.01802 *
## Population   5.014e-05  2.512e-05   1.996  0.05201 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7197 on 45 degrees of freedom
## Multiple R-squared:  0.736,  Adjusted R-squared:  0.7126
## F-statistic: 31.37 on 4 and 45 DF,  p-value: 1.696e-12
```

```r
#all predictors have very small p-value: significant
add1(mod4, ~.+Income+Illiteracy+Area, test = 'F')
```

```
## Single term additions
##
## Model:
## `Life Exp` ~ Murder + `HS Grad` + Frost + Population
##            Df Sum of Sq    RSS     AIC F value Pr(>F)
## <none>                  23.308 -28.161
## Income      1 0.0060582 23.302 -26.174  0.0114 0.9153
## Illiteracy  1 0.0039221 23.304 -26.170  0.0074 0.9318
## Area        1 0.0007900 23.307 -26.163  0.0015 0.9694
```

```r
#no more significant predictors, p-values > 0.15
#same model as what we found in AIC

#Best subset regression
library(leaps)
```

```r
mod = regsubsets(cbind(Population, Income, Illiteracy, Murder, `HS Grad`, Frost, Area), `Life Exp`)
summary.mod = summary(mod)
summary.mod$which
```

```
##   (Intercept) Population Income Illiteracy Murder HS Grad Frost  Area
## 1        TRUE      FALSE  FALSE      FALSE   TRUE   FALSE FALSE FALSE
## 2        TRUE      FALSE  FALSE      FALSE   TRUE    TRUE FALSE FALSE
## 3        TRUE      FALSE  FALSE      FALSE   TRUE    TRUE  TRUE FALSE
## 4        TRUE       TRUE  FALSE      FALSE   TRUE    TRUE  TRUE FALSE
## 5        TRUE       TRUE   TRUE      FALSE   TRUE    TRUE  TRUE FALSE
## 6        TRUE       TRUE   TRUE       TRUE   TRUE    TRUE  TRUE FALSE
## 7        TRUE       TRUE   TRUE       TRUE   TRUE    TRUE  TRUE  TRUE
```

```r
names(summary.mod)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```r
summary.mod$adjr2
```

```
## [1] 0.6015893 0.6484991 0.6939230 0.7125690 0.7061129 0.6993268 0.6921823
```

```r
# from 3rd to 4th, increased almost 2%
# from 4th to fifith, dropping
# so we choose 4 predictors, look back at matrix, find that same as what we found in stepwise regressio
summary.mod$cp
```

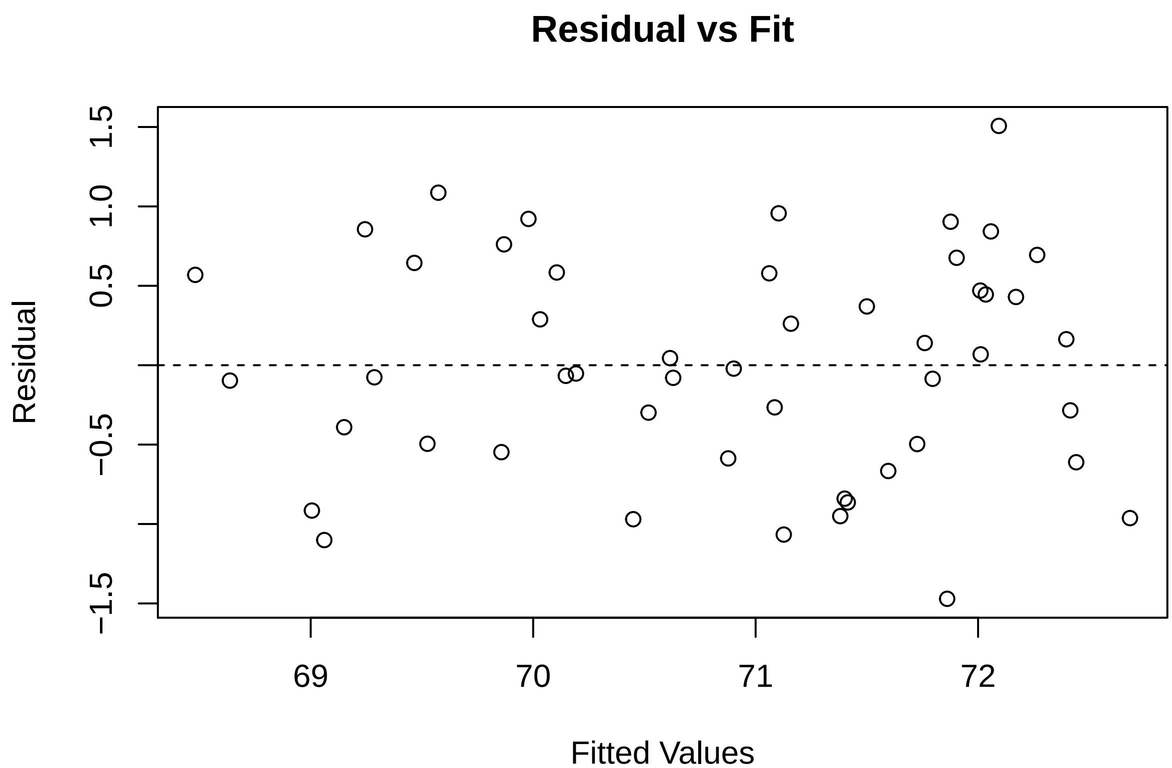
```
## [1] 16.126760  9.669894  3.739878  2.019659  4.008737  6.001959  8.000000
```

```r
# only C_p close to p is the third one, 3.74 close to p=4, three predictors
```

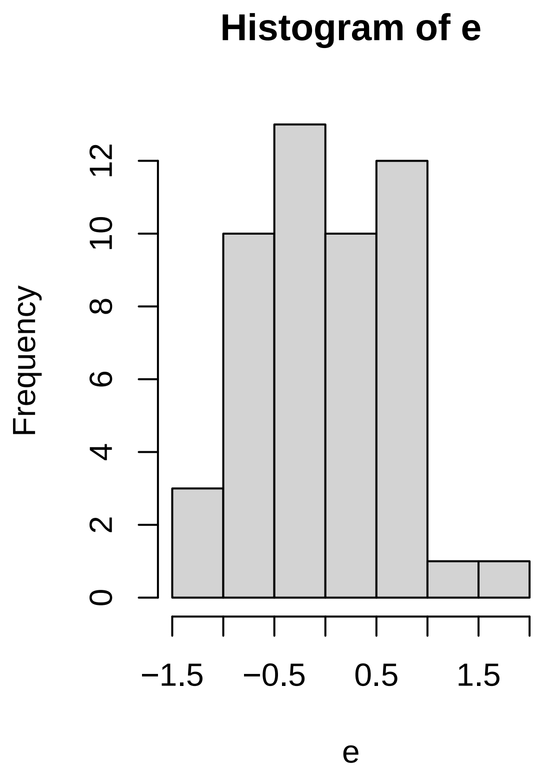
Diagnostic Checks and Transformation

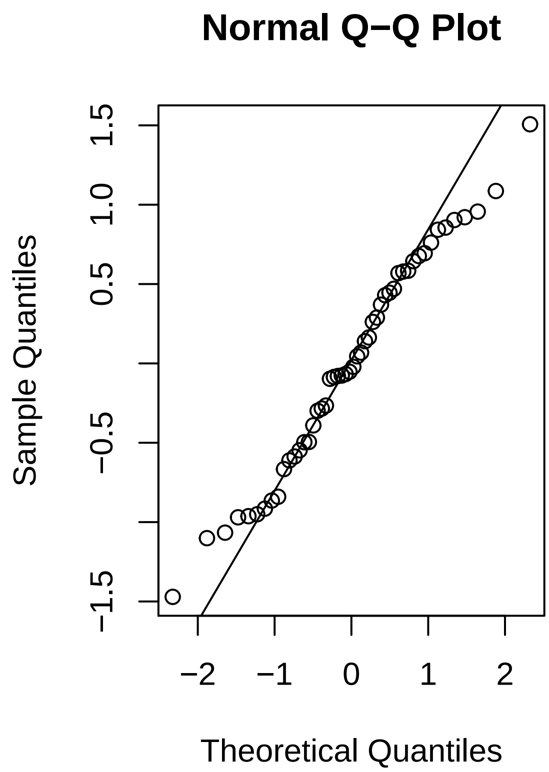
Thirdly, we check the LINE conditions for this model. We will not be checking the independence assumption, since we are not given data related to time order.

```
# Residuals Analysis
yhat=mod.AIC$fitted.values
e=mod.AIC$residuals
plot(yhat, e, xlab = 'Fitted Values', ylab = 'Residual', main = 'Residual vs Fit')
abline(h = 0, lty = 2)
```



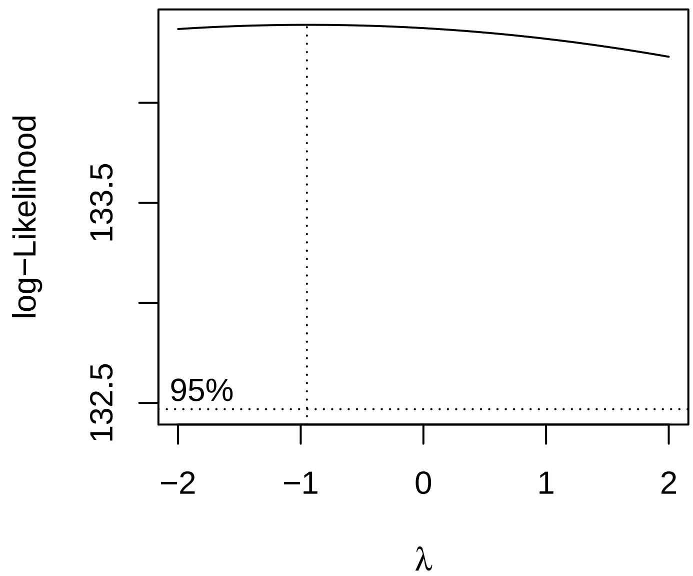

```
par(mfrow=c(1,2))
hist(e)
qqnorm(e)
qqline(e)
```





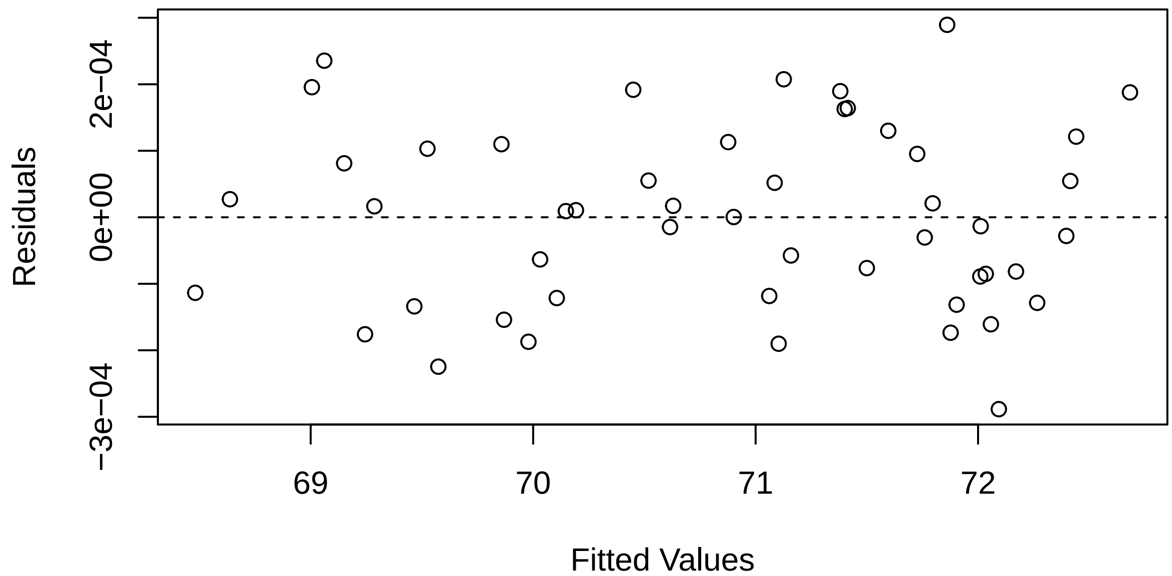
The Residual v.s. Fitted plot shows that residuals "bounce randomly" and roughly form a "horizontal band" around the y=0 line. However, when looking at the "Residuals vs Predictor" plot, and see a strong funneling effect for the "Residuals v.s. Population" plot. Since a log function has the ability to "spread out" smaller values and bring in larger ones, we will perform log transformation on "Population". Our model is now `Life Exp` ~ Murder + `HS Grad` + Frost + log(Population). Then we check our LINE conditions again.

```
library(MASS)
boxcox(`Life Exp`~Murder+`HS Grad`+Frost+Population, data=dat)
```

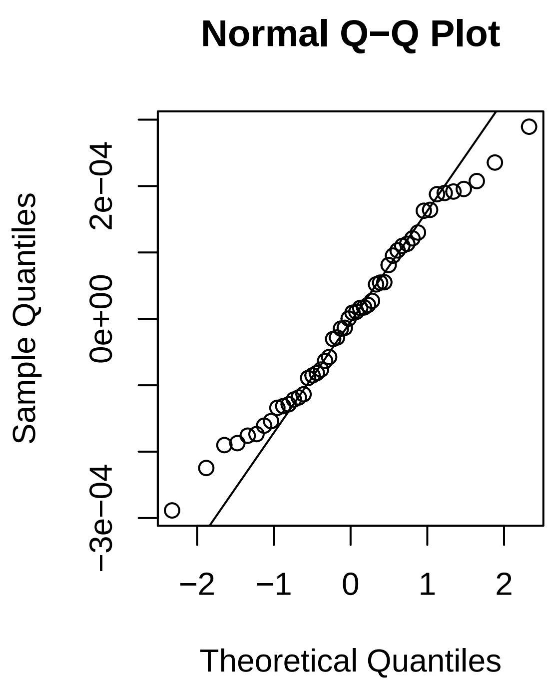


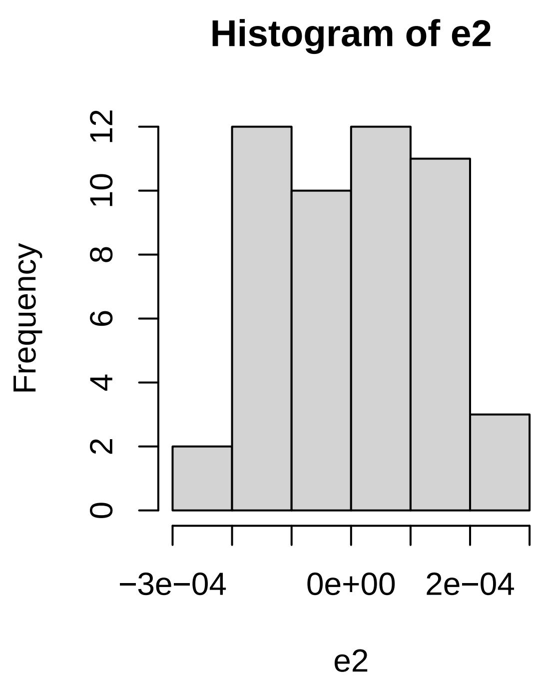

```r
# choose lambda -1

y <- 1/`Life Exp` #transform y
mod.trans <- lm(y ~ Murder + `HS Grad` + Frost + Population) #fit new model
```

```r
#Residuals Analysis again
e2 = resid(mod.trans)
yhat2 = fitted(mod.trans)
plot(yhat, e2, xlab = 'Fitted Values', ylab = 'Residuals' )
abline(h = 0, lty = 2)
```




```
par(mfrow=c(1,2))
qqnorm(e2)
qqline(e2)
hist(e2)
```





The "Residuals vs Predictor" plot for log(Population) is well-behaved now. The Residual v.s. Fit plot and Normal Q-Q plot are both well-behaved. There are no unequal variance or nonlinearity problems.

Our final step is checking for outliers and leverage. After computing for both internally studentized residuals and studentized deleted (or externally studentized) residuals, none of them are larger than 3 in absolute value. Thus, there are no unusual Y observations. After computing the hat values, we find that none of the points has higher hat value than 3pn=0.3. Therefore, there are no outliers or leverage points. And we will not need to investigate for any potentially influential points. Our model has met the LINE conditions.

```
rs=rstandard(mod.trans) # internally studentized residuals
sort(rs)
```

```
##          11           4           6          43          34          17
## -2.312397145 -1.642096199 -1.392666838 -1.380232508 -1.299924684 -1.292403731
##          44          22          42          16          23          25
## -1.178380512 -1.127632159 -0.967624764 -0.938663562 -0.931440707 -0.862638779
##           1          45          49           7          27          12
## -0.856274572 -0.851708177 -0.642991480 -0.610231618 -0.585185596 -0.545616584
##          31          36          39          15           9          41
## -0.457293342 -0.408572247 -0.225683154 -0.202064790 -0.107968537 -0.098731576
##          14          46          13          33          32           5
##  0.002777054  0.065610866  0.078390689  0.120827440  0.136776727  0.186842593
##          10          35          20          37          18          29
##  0.200003870  0.378433684  0.390104057  0.409146962  0.602970158  0.687516284
##          50          28          21           2          30          26
##  0.833693070  0.854206385  0.876780085  0.885184531  0.933028492  1.166504902
##           3          48          38          24          47           8
##  1.240330690  1.400678620  1.415709048  1.438322963  1.441925978  1.478156382
##          40          19
##  1.737232362  2.090548516
```

```
rsd=rstudent(mod.trans) # studentized deleted
sort(rsd)
```

```
##          11           4           6          43          34          17
## -2.435856650 -1.674698690 -1.407777903 -1.394650408 -1.310235780 -1.302362064
##          44          22          42          16          23          25
## -1.183618594 -1.131128046 -0.966925070 -0.937397750 -0.930042275 -0.860141598
##           1          45          49           7          27          12
## -0.853690309 -0.849062901 -0.638748015 -0.605925449 -0.580861352 -0.541313619
##          31          36          39          15           9          41
## -0.453238095 -0.404758491 -0.223287874 -0.199897720 -0.106775978 -0.097638971
##          14          46          13          33          32           5
##  0.002746025  0.064880864  0.077520081  0.119496756  0.135276570  0.184826607
##          10          35          20          37          18          29
##  0.197857079  0.374802119  0.386399132  0.405329966  0.598656145  0.683433153
##          50          28          21           2          30          26
##  0.830818915  0.851594337  0.874485022  0.883015295  0.931658924  1.171316298
##           3          48          38          24          47           8
##  1.247989909  1.416244694  1.432146649  1.456116427  1.459940479  1.498471455
##          40          19
##  1.778494656  2.175531013
```

```r
n=length(e2)
p=4+1 # four predictors + 1
3*p/n # rules of thumb, 3 times the mean leverage value
```

```
## [1] 0.3
```

```r
hv=hatvalues(mod.trans)
sort(hv)
```

```
##         20         14         46         25         36          8         16
## 0.02251734 0.02574946 0.03054924 0.03207145 0.03526037 0.03735911 0.04264019
##         12         26          7         30         27         45         15
## 0.04280306 0.04851763 0.04944598 0.05097477 0.05189556 0.05722013 0.05932553
##         29         31         49         42         19         21         23
## 0.06221607 0.06286777 0.06355888 0.06417731 0.06424817 0.06542733 0.06818938
##         35         48          4         22         33          6         44
## 0.08138412 0.08498652 0.08623296 0.08844258 0.08927508 0.08960146 0.09012184
##         41         17          9         24         10         43         50
## 0.09208789 0.09506497 0.09648760 0.09685602 0.10033898 0.10172016 0.10198735
##         40         13         18         39         38         34         37
## 0.10289140 0.10541465 0.11572004 0.11735640 0.12395238 0.12949804 0.13125063
##          1          3         47         32         11          2         28
## 0.14061825 0.14434012 0.17168830 0.22522744 0.23979244 0.24727915 0.28860921
##          5
## 0.38475924
```

```r
# 0.385 > .3

2*sqrt((p+1)/(n-p-1))
```

```
## [1] 0.7385489
```

```r
diff=dffits(mod.trans) # Difference in Fits (DFFITS)
sort(diff)
```

```
##            11             4            34            43             6
## -1.3680544508 -0.5144647137 -0.5053545527 -0.4693135104 -0.4416476273
##            17            44            22             1            42
## -0.4221174414 -0.3725076355 -0.3523306753 -0.3453250862 -0.2532134711
##            23            45            16            49            25
## -0.2515922280 -0.2091746895 -0.1978317089 -0.1664092449 -0.1565696529
##             7            27            31            12            39
## -0.1381962579 -0.1358968880 -0.1173924079 -0.1144684296 -0.0814190137
##            36            15             9            41            14
## -0.0773809949 -0.0502006076 -0.0348933176 -0.0310958595  0.0004464299
##            46            13            33            20            10
##  0.0115173861  0.0266105475  0.0374134383  0.0586462342  0.0660764923
##            32            35             5            37            29
##  0.0729367020  0.1115590770  0.1461626706  0.1575477328  0.1760337515
##            30            18            21            26            50
##  0.2159214150  0.2165642925  0.2313797959  0.2644987420  0.2799872837
```

```
##            8           48           24            2            3
##  0.2951988159  0.4316180200  0.4768490402  0.5061100998  0.5125709942
##           38           28           19           40           47
##  0.5387050822  0.5424175546  0.5700531269  0.6023093405  0.6646738595
```

```r
# no abs val greater than .739

ck=cooks.distance(mod.trans) # Cook's distance measure
sort(ck)
```

```
##           14           46           13           41            9           33
## 4.076583e-08 2.713040e-05 1.448232e-04 1.977429e-04 2.489785e-04 2.862228e-04
##           15           20           10           32           36           39
## 5.150075e-04 7.011306e-04 8.922726e-04 1.087681e-03 1.220238e-03 1.354409e-03
##           35           12           31           27            7            5
## 2.537554e-03 2.662433e-03 2.805737e-03 3.748792e-03 3.874125e-03 4.366422e-03
##           25           37           49           29           16           45
## 4.931320e-03 5.058195e-03 5.612240e-03 6.271852e-03 7.848631e-03 8.805422e-03
##           30           18           21           23           42           26
## 9.351846e-03 9.515693e-03 1.076360e-02 1.269783e-02 1.284198e-02 1.387720e-02
##           50            8            1           22           44           17
## 1.578724e-02 1.695911e-02 2.399450e-02 2.467415e-02 2.750730e-02 3.509373e-02
##           48            6           43           24           34            4
## 3.644429e-02 3.817754e-02 4.314494e-02 4.437235e-02 5.027590e-02 5.089382e-02
##            2            3           38           28           19           40
## 5.148150e-02 5.190281e-02 5.671595e-02 5.920489e-02 6.001373e-02 6.922770e-02
##           47           11
## 8.619118e-02 3.373325e-01
```

```r
#not influential
```

Interpretation

We are now able to observe our model with 4 predictors: Murder, HS Grad, Frost, log(Population).
$LifeExpectancy = -0.29Murder + 0.0546HSGrad - 0.051Frost + 0.24log(Population)$

```
summary(mod.trans)
```

```
##
## Call:
## lm(formula = y ~ Murder + ‘HS Grad‘ + Frost + Population)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.885e-04 -1.172e-04  4.819e-06  1.082e-04  2.894e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.409e-02  1.895e-04  74.366  < 2e-16 ***
## Murder       5.999e-05  7.280e-06   8.241 1.54e-10 ***
## ‘HS Grad‘   -9.329e-06  2.948e-06  -3.164  0.00279 **
## Frost        1.158e-06  4.814e-07   2.406  0.02029 *
## Population  -1.053e-08  4.995e-09  -2.107  0.04068 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.0001431 on 45 degrees of freedom
## Multiple R-squared:  0.7387, Adjusted R-squared:  0.7155
## F-statistic: 31.81 on 4 and 45 DF,  p-value: 1.352e-12
```

From the above summary table of our model, the adjusted R2is 0.7173, telling us that about 71.73% percent variation in life expectancy is explained by our model. Also, the associated p-value 1.17e-12of the whole model is very small, indicating our model is significant.

"Murder" has negative coefficients -0.29, meaning that we predict a 1 percent increase in murder rate would result in -0.29 year decrease in the mean life expectancy. Similarly, "Frost" has a coefficient -0.00517, indicating that we expect a 1 unit increase in the mean number of days under freezing would bring 0.00517 year decrease in the mean life expectancy. On the other hand, the positive coefficient of "HS Grad" indicates that 1 percentage increase in high school graduation increases mean life expectancy by 0.0546 years. And we expect mean life expectancy to increase 0.5684 years for each ten-fold increase in population . (0.56836 = 0.246836 ln(10))

5. Conclusion

In conclusion, we are able to predict the meanlife expectancy of people in a U.S. state given its population, local murder rate, high school graduation percentage, and the mean number of days with minimum temperature below freezing. In general, states with higher population and high school graduation percentage would have longer life expectancy, while higher murder rates and more days in freezing temperatures would result in shorter life expectancy.

Given that the size of the dataset is limited (including only statistics from each state), the accuracy could be improved if we are able to draw more data by smaller region, for example, census by county. It would also be helpful if we could draw more possible related predictors into the dataset, for example, the elevation of the region, unemployment rate, healthcare coverage, air quality, etc. We should also note that the data we draw is from the US census in 1977, which means that necessary adjustment is needed with updated data for contemporary prediction.