Life Expectancy Predictor

Riya Kalra

2024-12-15

$$\text{Life.Exp} = \beta_0 + \beta_1 \cdot \text{Murder} + \beta_2 \cdot \text{HS.Grad} + \beta_3 \cdot \text{Frost} + \beta_4 \cdot \text{Population}$$

R dataset state.x77 from library(faraway) contains information on 50 states from 1970s collected by US Census Bureau. The goal is to predict 'life expectancy' using a combination of remaining variables.

a) First we load the dataset and provide descriptive statistics for all variables of interest.

```r
data(state)
state_data <- as.data.frame(state.x77)

# Descriptive statistics
summary(state_data)
```

```
##    Population        Income       Illiteracy       Life Exp
##  Min.   :  365   Min.   :3098   Min.   :0.500   Min.   :67.96
##  1st Qu.: 1080   1st Qu.:3993   1st Qu.:0.625   1st Qu.:70.12
##  Median : 2838   Median :4519   Median :0.950   Median :70.67
##  Mean   : 4246   Mean   :4436   Mean   :1.170   Mean   :70.88
##  3rd Qu.: 4968   3rd Qu.:4814   3rd Qu.:1.575   3rd Qu.:71.89
##  Max.   :21198   Max.   :6315   Max.   :2.800   Max.   :73.60
##      Murder          HS Grad          Frost             Area
##  Min.   : 1.400   Min.   :37.80   Min.   :  0.00   Min.   :  1049
##  1st Qu.: 4.350   1st Qu.:48.05   1st Qu.: 66.25   1st Qu.: 36985
##  Median : 6.850   Median :53.25   Median :114.50   Median : 54277
##  Mean   : 7.378   Mean   :53.11   Mean   :104.46   Mean   : 70736
##  3rd Qu.:10.675   3rd Qu.:59.15   3rd Qu.:139.75   3rd Qu.: 81162
##  Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432
```

```r
sapply(state_data, sd)  # Standard deviations
```

```
##   Population       Income   Illiteracy     Life Exp       Murder      HS Grad
## 4.464491e+03 6.144699e+02 6.095331e-01 1.342394e+00 3.691540e+00 8.076998e+00
##        Frost         Area
## 5.198085e+01 8.532730e+04
```

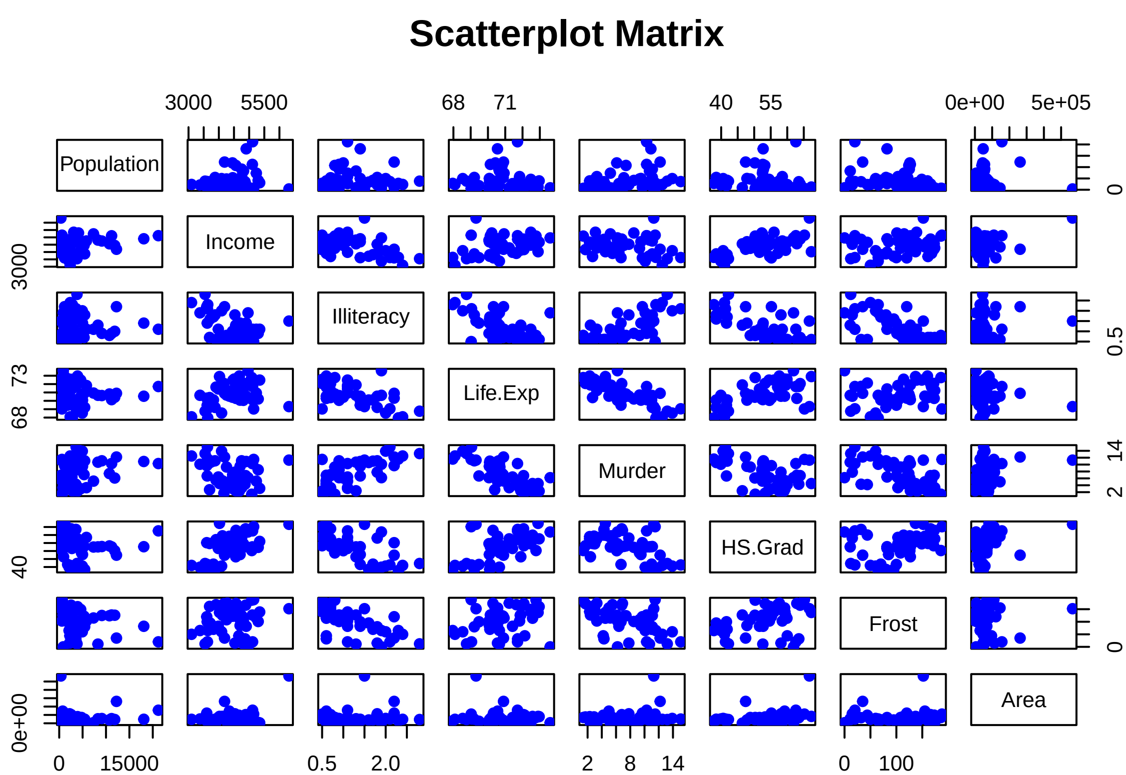
```r
# Convert all variables to numeric using lapply
state_data <- as.data.frame(lapply(state_data, as.numeric))

# Confirm all variables are numeric
str(state_data)
```

```
## 'data.frame':    50 obs. of  8 variables:
##  $ Population: num  3615 365 2212 2110 21198 ...
##  $ Income    : num  3624 6315 4530 3378 5114 ...
##  $ Illiteracy: num  2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ...
##  $ Life.Exp  : num  69 69.3 70.5 70.7 71.7 ...
##  $ Murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ HS.Grad   : num  41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ...
##  $ Frost     : num  20 152 15 65 20 166 139 103 11 60 ...
##  $ Area      : num  50708 566432 113417 51945 156361 ...
```

Now we can create some plots. We can start with a scatterplot matrix of multiple variables to get a better idea of correlations between them.

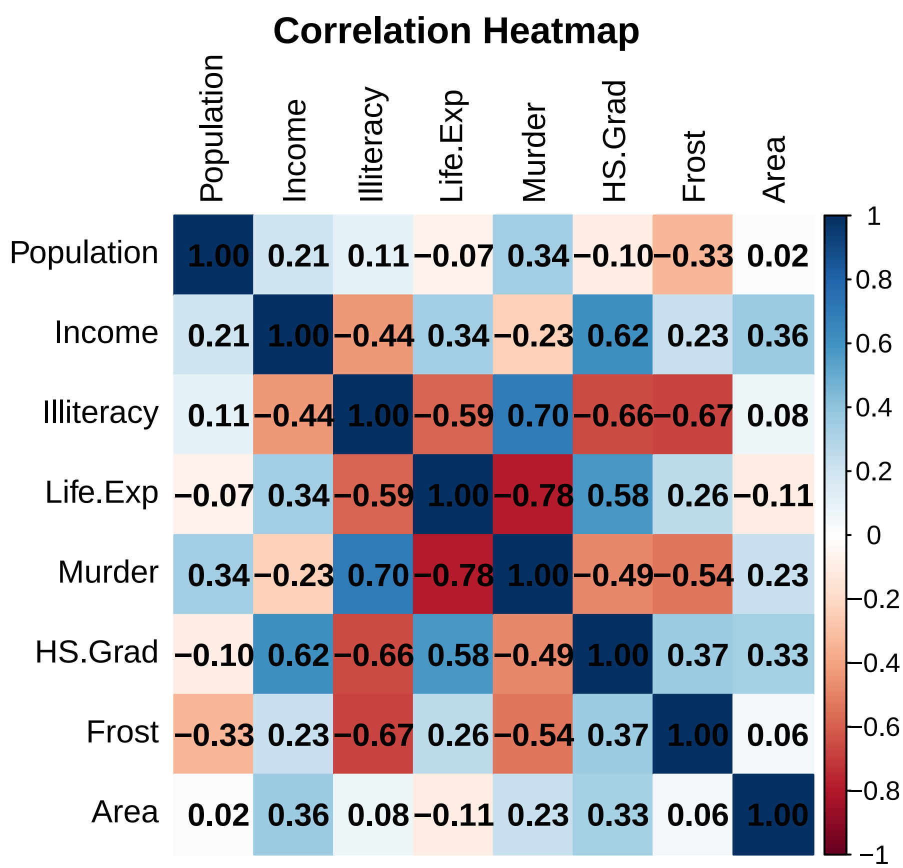
```r
pairs(state_data,
      main = "Scatterplot Matrix",
      col = "blue",
      pch = 19)
```



We can now make a heatmap that gives us more specific data. From this, we can see that life expectancy may be correlated with the murder, high school graduation, and illiteracy variables. We may also want to consider income or frost.

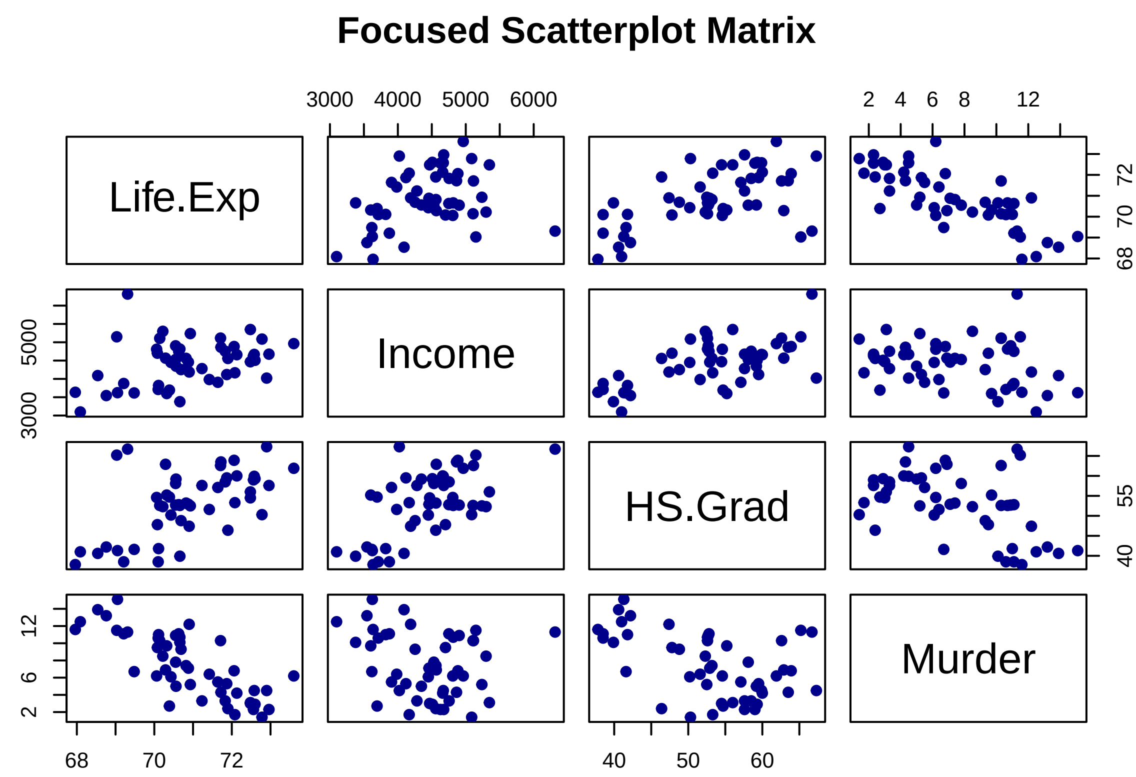
```r
# Compute correlation matrix
cor_matrix <- cor(state_data)

# Visualize correlations
corrplot(cor_matrix, method = "color", addCoef.col = "black", tl.col = "black",
         title = "Correlation Heatmap", mar = c(0, 0, 1, 0))
```



Next we pair some of the more correlated variables to get a better look.

```r
pairs(state_data[, c("Life.Exp", "Income", "HS.Grad", "Murder")],
      main = "Focused Scatterplot Matrix",
      col = "darkblue", pch = 19)
```

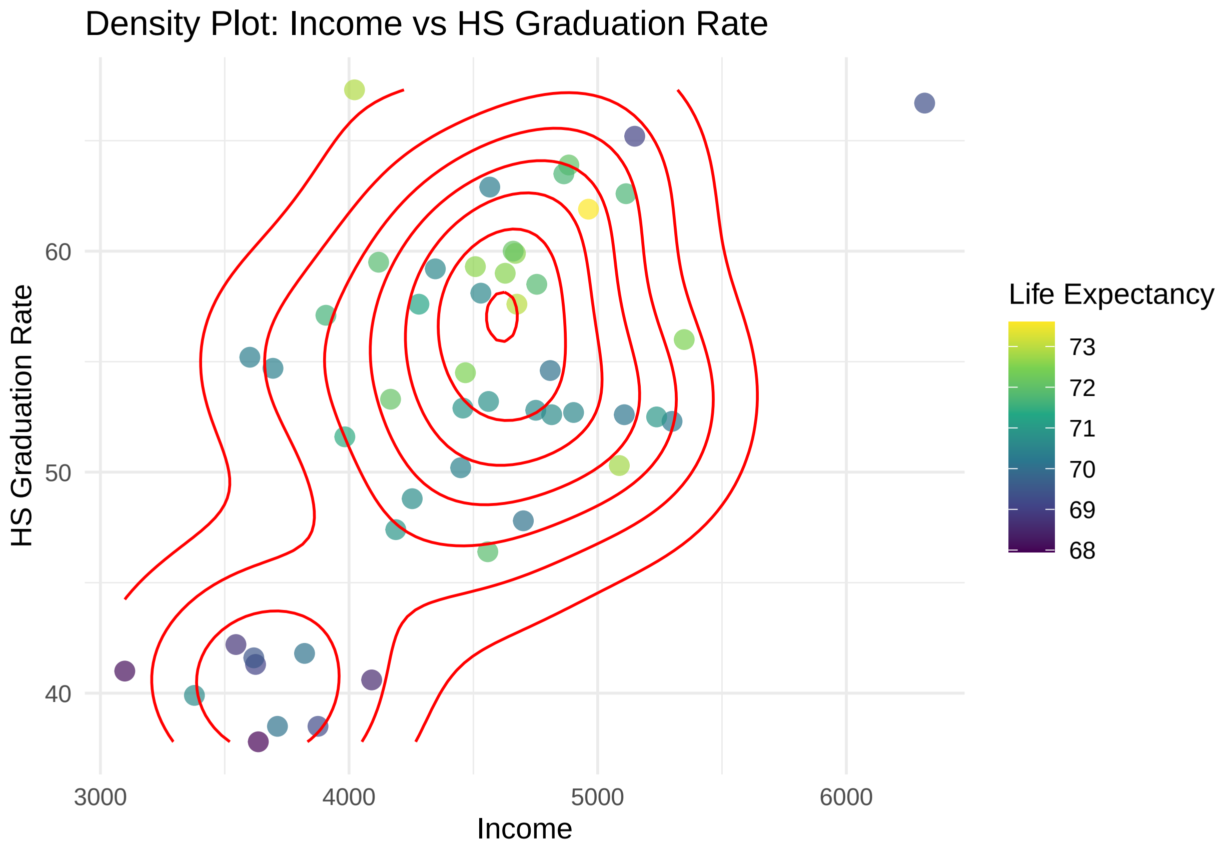


b) Examine the plots and decide transformations

The **scatterplot matrix** and **correlation heatmap** reveal key relationships among `Life.Exp`, `Income`, `HS.Grad`, and `Murder`. The scatterplots show that **life expectancy** has a positive relationship with both **income** and **high school graduation rates**, indicating that states with higher income and education levels tend to have longer life expectancy. Conversely, life expectancy exhibits a strong **negative relationship** with **murder rates**, suggesting that higher crime rates are associated with lower life expectancy. The positive relationship between **income** and **HS graduation rates** highlights that states with higher incomes tend to have better education outcomes. Additionally, a negative relationship is observed between **murder rates** and both **income** and **HS graduation rates**, indicating that higher education and income levels may contribute to lower crime rates.

The **correlation heatmap** quantifies these relationships. `Life.Exp` is strongly negatively correlated with `Murder` (-0.78) and positively correlated with `HS.Grad` (0.58) and `Income` (0.34). `HS.Grad` and `Illiteracy` have a strong negative correlation (-0.66), reflecting the inverse relationship between education and illiteracy. Similarly, **murder rates** are positively correlated with `Illiteracy` (0.70) and negatively correlated with `HS.Grad` (-0.49), further underscoring the importance of education in reducing crime. Weak correlations between variables like `Area` and `Population` suggest limited influence on life expectancy or education outcomes. Together, these plots highlight the interconnected roles of income, education, and crime in influencing life expectancy across states.

```r
ggplot(state_data, aes(x = Income, y = HS.Grad)) +
  geom_point(aes(color = Life.Exp), size = 3, alpha = 0.7) +
  geom_density_2d(color = "red") +
  scale_color_viridis_c() +
  labs(title = "Density Plot: Income vs HS Graduation Rate",
       x = "Income", y = "HS Graduation Rate", color = "Life Expectancy") +
  theme_minimal()
```

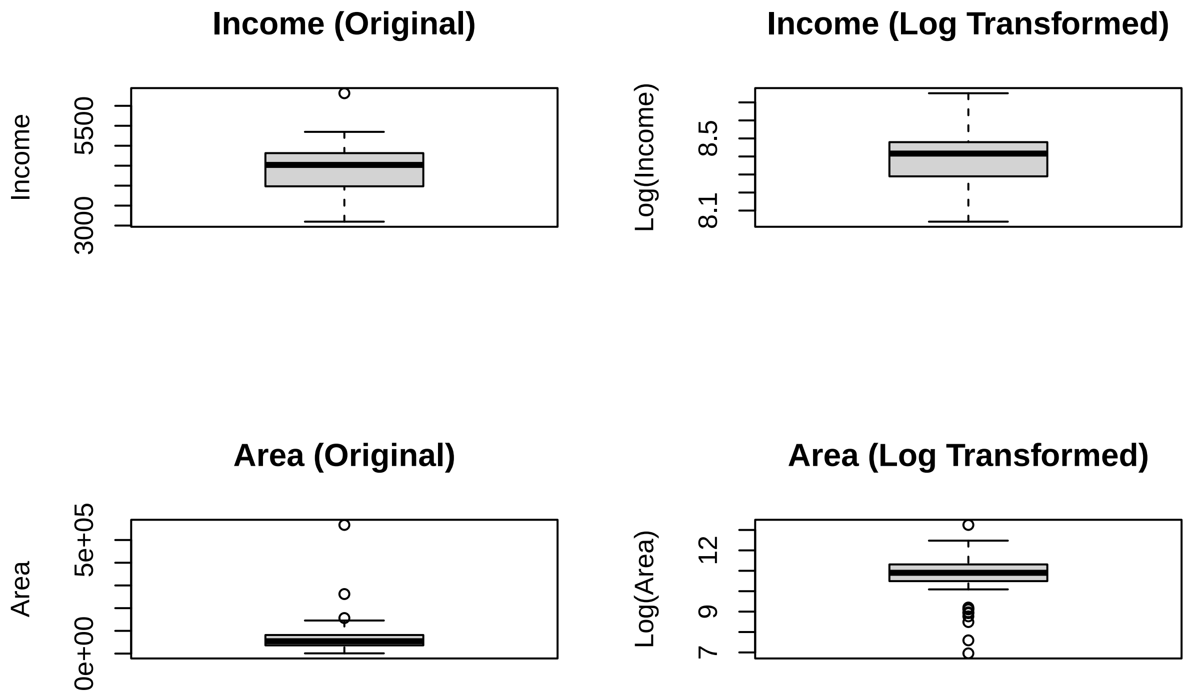


The density plot highlights key patterns among income, HS graduation rates, and life expectancy. High-density regions occur around an income of $5,000–$6,000 and HS graduation rates of 55%–60%. Higher incomes generally correspond to higher graduation rates, while states with incomes below $4,000 and graduation rates below 50% cluster in the bottom-left. Life expectancy, shown by a color gradient, is higher (yellow, ~73 years) in regions with both higher income and graduation rates, while lower life expectancy (purple, ~68 years) is linked to lower income and graduation rates. Sparse data exists in areas of low graduation rates with high income or vice versa, indicating these combinations are uncommon.

Transformations

```
par(mfrow = c(2, 2))
boxplot(state_data$Income, main = "Income (Original)", ylab = "Income")
boxplot(log(state_data$Income), main = "Income (Log Transformed)", ylab = "Log(Income)")

boxplot(state_data$Area, main = "Area (Original)", ylab = "Area")
boxplot(log(state_data$Area), main = "Area (Log Transformed)", ylab = "Log(Area)")
```



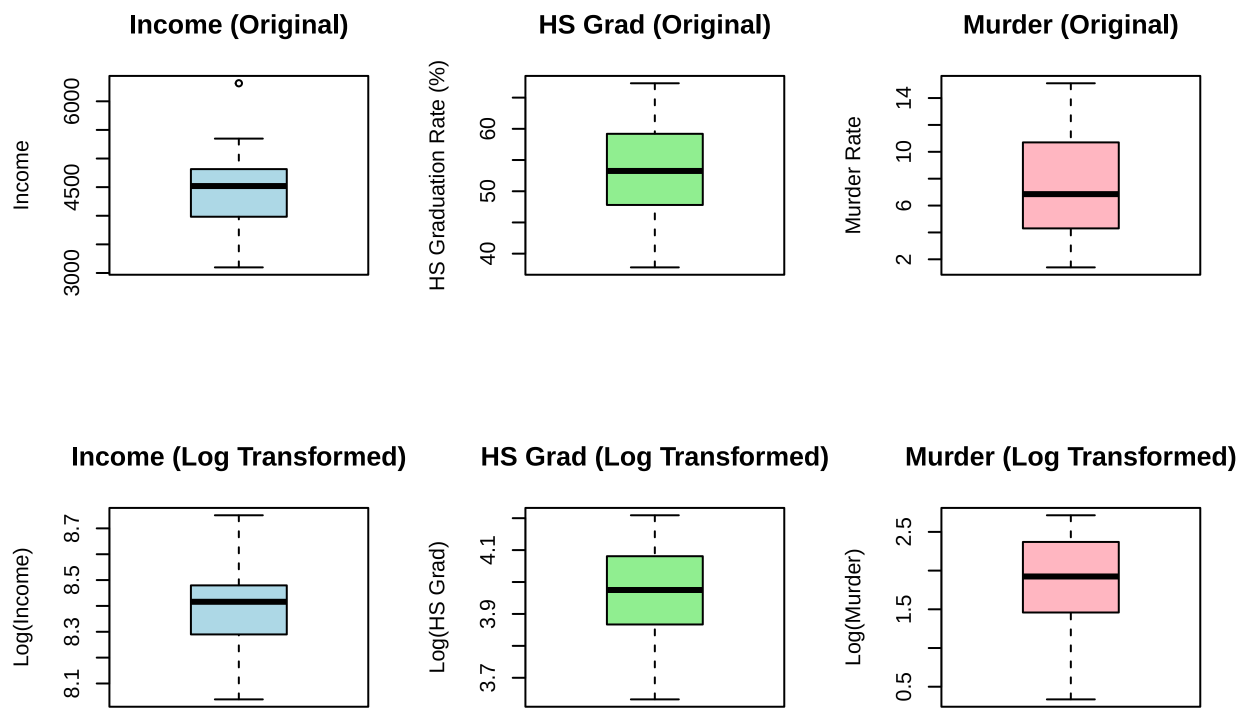

```r
par(mfrow = c(1, 1))

# Boxplots for Income, HS.Grad, and Murder (before and after log transformation)

# Set up a 2x3 plotting layout
par(mfrow = c(2, 3))  # 2 rows, 3 columns

# Original Boxplots
boxplot(state_data$Income, main = "Income (Original)", ylab = "Income", col = "lightblue")
boxplot(state_data$HS.Grad, main = "HS Grad (Original)", ylab = "HS Graduation Rate (%)", col = "lightg
boxplot(state_data$Murder, main = "Murder (Original)", ylab = "Murder Rate", col = "lightpink")

# Log-Transformed Boxplots
boxplot(log(state_data$Income), main = "Income (Log Transformed)", ylab = "Log(Income)", col = "lightbl
boxplot(log(state_data$HS.Grad), main = "HS Grad (Log Transformed)", ylab = "Log(HS Grad)", col = "ligh
boxplot(log(state_data$Murder), main = "Murder (Log Transformed)", ylab = "Log(Murder)", col = "lightpi
```




```r
# Reset layout
par(mfrow = c(1, 1))
```

P-values for predictors

Table 1: P-Values for Individual Predictors

Variable	P-Value
Murder	2.260070e-11
Illiteracy	6.969250e-06
HS.Grad	9.196096e-06
Income	0.0156
Frost	0.0660
Area	0.4581
Population	0.6387

c) Automatic Selection

```r
# Enter the variable with the lowest p-value: Murder
forward1 <- lm(Life.Exp ~ Murder, data = state_data)
summary(forward1)
```

Forward Model

```
##
## Call:
## lm(formula = Life.Exp ~ Murder, data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81690 -0.48139  0.09591  0.39769  2.38691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.97356    0.26997  270.30  < 2e-16 ***
## Murder      -0.28395    0.03279   -8.66 2.26e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8473 on 48 degrees of freedom
## Multiple R-squared:  0.6097, Adjusted R-squared:  0.6016
## F-statistic: 74.99 on 1 and 48 DF,  p-value: 2.26e-11
```

```
# Step 2: Add the next variable with the lowest p-value
forward2 <- update(forward1, . ~ . + HS.Grad)
summary(forward2)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + HS.Grad, data = state_data)
##
## 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
```

```
forward3 <- update(forward2, . ~ . + Illiteracy)
summary(forward3)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + HS.Grad + Illiteracy, data = state_data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.65922 -0.46400  0.08517  0.59643  1.77657
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.73545    1.22208  57.063  < 2e-16 ***
## Murder      -0.25813    0.04350  -5.934 3.63e-07 ***
## HS.Grad      0.05179    0.01876   2.761  0.00825 **
## Illiteracy   0.25398    0.30508   0.833  0.40942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7985 on 46 degrees of freedom
## Multiple R-squared:  0.6679, Adjusted R-squared:  0.6462
## F-statistic: 30.83 on 3 and 46 DF,  p-value: 4.444e-11
```

```r
forward4 <- update(forward3, . ~ . + Income)
summary(forward4)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + HS.Grad + Illiteracy + Income,
##     data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56498 -0.53611  0.05303  0.58972  1.73972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.4833066  1.3253230  52.427  < 2e-16 ***
## Murder      -0.2619402  0.0444659  -5.891 4.53e-07 ***
## HS.Grad      0.0461443  0.0218485   2.112   0.0403 *
## Illiteracy   0.2760771  0.3105081   0.889   0.3787
## Income       0.0001249  0.0002422   0.516   0.6084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8049 on 45 degrees of freedom
## Multiple R-squared:  0.6698, Adjusted R-squared:  0.6405
## F-statistic: 22.82 on 4 and 45 DF,  p-value: 2.39e-10
```

```r
forward5 <- update(forward4, . ~ . + Frost)
summary(forward5)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + HS.Grad + Illiteracy + Income +
##     Frost, data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40424 -0.53182  0.07773  0.53496  1.30297
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 71.2850674  1.4124737  50.468  < 2e-16 ***
## Murder      -0.2765157  0.0420412  -6.577 4.77e-08 ***
## HS.Grad      0.0398761  0.0206167   1.934   0.0595 .
## Illiteracy  -0.1600309  0.3334357  -0.480   0.6336
## Income       0.0001133  0.0002271   0.499   0.6204
## Frost       -0.0076509  0.0028522  -2.682   0.0103 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7547 on 44 degrees of freedom
## Multiple R-squared:  0.7162, Adjusted R-squared:  0.684
## F-statistic: 22.21 on 5 and 44 DF,  p-value: 4.847e-11
```

```r
#Frost has a p-value of 0.066, which is not statistically significant
#Stop if no additional variables significantly improve the model
```

Backward Model

```r
# Fit the full model with all predictors
full_model <- lm(Life.Exp ~ Murder + Illiteracy + HS.Grad + Income + Frost + Area + Population, data = s
summary(full_model)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Illiteracy + HS.Grad + Income +
##     Frost + Area + Population, data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586  < 2e-16 ***
## Murder      -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## Illiteracy   3.382e-02  3.663e-01   0.092   0.9269
## HS.Grad      4.893e-02  2.332e-02   2.098   0.0420 *
## Income      -2.180e-05  2.444e-04  -0.089   0.9293
## Frost       -5.735e-03  3.143e-03  -1.825   0.0752 .
## Area        -7.383e-08  1.668e-06  -0.044   0.9649
## Population   5.180e-05  2.919e-05   1.775   0.0832 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

```r
# Step 1: Remove the predictor with the highest p-value
step1 <- update(full_model, . ~ . - Population)
summary(step1)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Illiteracy + HS.Grad + Income +
##     Frost + Area, data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.39934 -0.53722  0.08628  0.53270  1.28452
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.112e+01  1.788e+00  39.771  < 2e-16 ***
## Murder      -2.742e-01  4.517e-02  -6.070 2.89e-07 ***
## Illiteracy  -1.399e-01  3.617e-01  -0.387   0.7008
## HS.Grad      4.155e-02  2.352e-02   1.767   0.0843 .
## Income       1.219e-04  2.363e-04   0.516   0.6088
## Frost       -7.495e-03  3.056e-03  -2.452   0.0183 *
## Area        -2.625e-07  1.706e-06  -0.154   0.8784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7632 on 43 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.6768
## F-statistic:  18.1 on 6 and 43 DF,  p-value: 2.41e-10
```

```r
# Step 2: Remove the next predictor with the highest p-value
step2 <- update(step1, . ~ . - Area)
summary(step2)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Illiteracy + HS.Grad + Income +
##     Frost, data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40424 -0.53182  0.07773  0.53496  1.30297
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 71.2850674  1.4124737  50.468  < 2e-16 ***
## Murder      -0.2765157  0.0420412  -6.577 4.77e-08 ***
## Illiteracy  -0.1600309  0.3334357  -0.480   0.6336
## HS.Grad      0.0398761  0.0206167   1.934   0.0595 .
## Income       0.0001133  0.0002271   0.499   0.6204
## Frost       -0.0076509  0.0028522  -2.682   0.0103 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.7547 on 44 degrees of freedom
## Multiple R-squared:  0.7162, Adjusted R-squared:  0.684
## F-statistic: 22.21 on 5 and 44 DF,  p-value: 4.847e-11
```

```r
# Step 3: Remove the next predictor with the highest p-value
step3 <- update(step2, . ~ . - Frost)
summary(step3)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Illiteracy + HS.Grad + Income,
##     data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56498 -0.53611  0.05303  0.58972  1.73972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.4833066  1.3253230  52.427  < 2e-16 ***
## Murder      -0.2619402  0.0444659  -5.891 4.53e-07 ***
## Illiteracy   0.2760771  0.3105081   0.889   0.3787
## HS.Grad      0.0461443  0.0218485   2.112   0.0403 *
## Income       0.0001249  0.0002422   0.516   0.6084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8049 on 45 degrees of freedom
## Multiple R-squared:  0.6698, Adjusted R-squared:  0.6405
## F-statistic: 22.82 on 4 and 45 DF,  p-value: 2.39e-10
```

```r
# Step 4: Check remaining predictors
summary(step3)  # Stop if all remaining predictors are significant
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Illiteracy + HS.Grad + Income,
##     data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56498 -0.53611  0.05303  0.58972  1.73972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.4833066  1.3253230  52.427  < 2e-16 ***
## Murder      -0.2619402  0.0444659  -5.891 4.53e-07 ***
## Illiteracy   0.2760771  0.3105081   0.889   0.3787
## HS.Grad      0.0461443  0.0218485   2.112   0.0403 *
## Income       0.0001249  0.0002422   0.516   0.6084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8049 on 45 degrees of freedom
## Multiple R-squared:  0.6698, Adjusted R-squared:  0.6405
## F-statistic: 22.82 on 4 and 45 DF,  p-value: 2.39e-10
```

```r
# Automate backward elimination using step()
final_model <- step(full_model, direction = "backward")
```

```
## Start:  AIC=-22.18
## Life.Exp ~ Murder + Illiteracy + HS.Grad + Income + Frost + Area +
##     Population
##
##              Df Sum of Sq    RSS     AIC
## - Area        1    0.0011 23.298 -24.182
## - Income      1    0.0044 23.302 -24.175
## - Illiteracy  1    0.0047 23.302 -24.174
## <none>                    23.297 -22.185
## - Population  1    1.7472 25.044 -20.569
## - Frost       1    1.8466 25.144 -20.371
## - HS.Grad     1    2.4413 25.738 -19.202
## - Murder      1   23.1411 46.438  10.305
##
## Step:  AIC=-24.18
## Life.Exp ~ Murder + Illiteracy + HS.Grad + Income + Frost + Population
##
##              Df Sum of Sq    RSS     AIC
## - Illiteracy  1    0.0038 23.302 -26.174
## - Income      1    0.0059 23.304 -26.170
## <none>                    23.298 -24.182
## - Population  1    1.7599 25.058 -22.541
## - Frost       1    2.0488 25.347 -21.968
## - HS.Grad     1    2.9804 26.279 -20.163
## - Murder      1   26.2721 49.570  11.569
##
## Step:  AIC=-26.17
## Life.Exp ~ Murder + HS.Grad + Income + Frost + Population
##
##              Df Sum of Sq    RSS     AIC
## - Income      1     0.006 23.308 -28.161
## <none>                    23.302 -26.174
## - Population  1     1.887 25.189 -24.280
## - Frost       1     3.037 26.339 -22.048
## - HS.Grad     1     3.495 26.797 -21.187
## - Murder      1    34.739 58.041  17.456
##
## Step:  AIC=-28.16
## Life.Exp ~ Murder + HS.Grad + Frost + Population
##
##              Df Sum of Sq    RSS     AIC
## <none>                    23.308 -28.161
## - 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
```

```r
summary(final_model)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + HS.Grad + Frost + Population,
##     data = state_data)
##
## 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
```

The variable **`Illiteracy`** was not included in the final model because it became statistically insignificant during the backward elimination process. Although `Illiteracy` had a low p-value when considered alone, its significance likely dropped after adding other predictors, such as `HS.Grad` and `Murder`, due to **collinearity**. The strong negative correlation between `Illiteracy` and `HS.Grad` (-0.66) indicates that both variables explain similar variance in **Life.Exp**. As a result, the backward elimination process retained **`HS.Grad`** as the more impactful predictor, while removing `Illiteracy` to simplify the model without compromising its performance. Additionally, the automated `step()` function optimizes model fit using criteria like **AIC**, which penalizes unnecessary complexity. Including `Illiteracy` may not have significantly improved the model's goodness-of-fit, leading to its exclusion.

Do the procedures generate the same model? Are any variables a close call? Is there any association between 'Illiteracy' and 'HS graduation rate'? Not quite. The forward model includes `Murder + HS.Grad + Illiteracy + Income + Frost`, and the backward includes `Life.Exp ~ Murder + HS.Grad + Frost + Population`. Additionally, `Illiteracy` and `HS.Grad` are collinear and `HS.Grad` is more impactful, so we will choose to keep that one. My chosen subset will not contain both.

d) Criterion-Based Procedures

```r
# Load necessary library
library(leaps)

# Convert data to a matrix
state_mat <- as.matrix(state_data[, c("Murder", "Illiteracy", "HS.Grad", "Income", "Frost", "Area", "Po
life_exp <- state_data$Life.Exp
```

```r
# Best models using Cp
leaps_cp <- leaps(x = state_mat, y = life_exp, nbest = 2, method = "Cp")
print(leaps_cp)
```

CP and R^2

```
## $which
##       1     2     3     4     5     6     7
## 1  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## 3  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 3  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 4  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 4  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## 5  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 5  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## 6  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 6  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 7  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
## [1] "(Intercept)" "1"           "2"           "3"           "4"
## [6] "5"           "6"           "7"
##
## $size
##  [1] 2 2 3 3 4 4 5 5 6 6 7 7 8
##
## $Cp
##  [1] 16.126760 58.058325  9.669894 10.886508  3.739878  5.647595  2.019659
##  [8]  5.411184  4.008737  4.012588  6.001959  6.007958  8.000000
```

```r
# Best models using Adjusted R²
leaps_adjr2 <- leaps(x = state_mat, y = life_exp, nbest = 2, method = "adjr2")
print(leaps_adjr2)
```

```
## $which
##       1     2     3     4     5     6     7
## 1  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## 3  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 3  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 4  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 4  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## 5  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 5  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## 6  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 6  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 7  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
##
## $label
## [1] "(Intercept)" "1"           "2"           "3"           "4"
## [6] "5"           "6"           "7"
##
## $size
##  [1] 2 2 3 3 4 4 5 5 6 6 7 7 8
##
## $adjr2
##  [1] 0.6015893 0.3326876 0.6484991 0.6405311 0.6939230 0.6811571 0.7125690
##  [8] 0.6893697 0.7061129 0.7060860 0.6993268 0.6992839 0.6921823
```

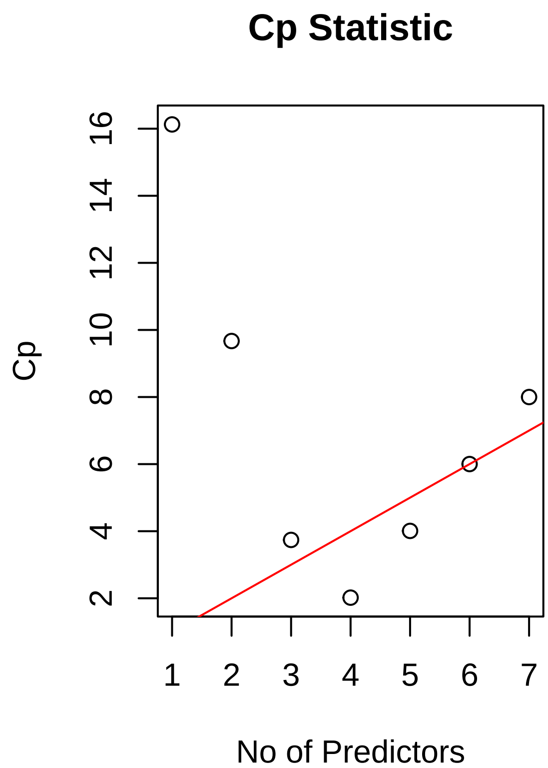
```r
# Use regsubsets() for subset selection and plot Cp and Adjusted R²
library(MASS)
```

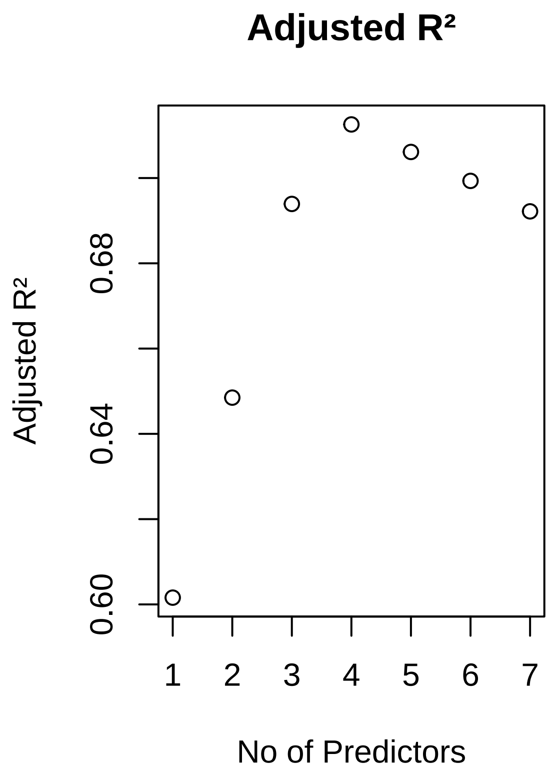
```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:plotly':
##
##     select
```

```r
subset_fit <- regsubsets(Life.Exp ~ Murder + Illiteracy + HS.Grad + Income + Frost + Area + Population,
                         data = state_data, nvmax = 7)
subset_summary <- summary(subset_fit)

# Plot Cp and Adjusted R²
par(mfrow = c(1, 2))
plot(1:7, subset_summary$cp, xlab = "No of Predictors", ylab = "Cp", main = "Cp Statistic")
abline(0, 1, col = "red")
plot(1:7, subset_summary$adjr2, xlab = "No of Predictors", ylab = "Adjusted R²", main = "Adjusted R²")
```





AIC and BIC

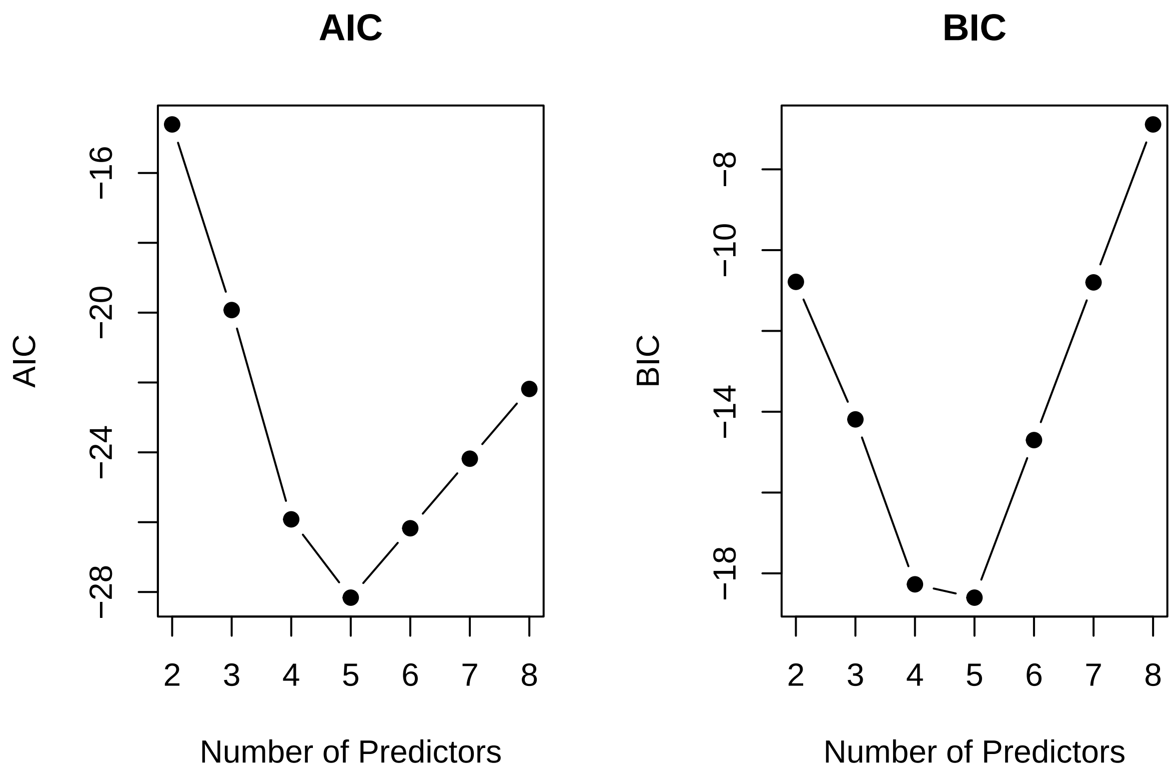
```
# Load necessary library
library(leaps)

# Fit all subsets using regsubsets
subset_fit <- regsubsets(Life.Exp ~ Murder + Illiteracy + HS.Grad + Income + Frost + Area + Population,
                         data = state_data, nvmax = 7)
subset_summary <- summary(subset_fit)

# Extract AIC and BIC for each subset size
n <- nrow(state_data)  # Sample size
rss <- subset_summary$rss  # Residual sum of squares
num_params <- 1:7 + 1  # Number of parameters (predictors + intercept)

# Calculate AIC and BIC
AIC_values <- n * log(rss / n) + 2 * num_params
BIC_values <- n * log(rss / n) + log(n) * num_params

# Plot AIC and BIC
par(mfrow = c(1, 2))
plot(num_params, AIC_values, type = "b", pch = 19, xlab = "Number of Predictors", ylab = "AIC", main =
plot(num_params, BIC_values, type = "b", pch = 19, xlab = "Number of Predictors", ylab = "BIC", main =
```




```
par(mfrow = c(1, 1))  # Reset layout
```

BIC is stricter than AIC.The optimal model includes 4 predictors:`Murder`, `HS.Grad`, `Population`, `Frost`.

e) LASSO

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
# Prepare the data for LASSO
X <- as.matrix(state_data[, c("Murder", "Illiteracy", "HS.Grad", "Income", "Frost", "Area", "Population
y <- state_data$Life.Exp

# Fit LASSO models for different lambdas
fit_5 <- glmnet(X, y, lambda = 5)
fit_1 <- glmnet(X, y, lambda = 1)
fit_0.1 <- glmnet(X, y, lambda = 0.1)

# Print coefficients for different lambdas
print(coef(fit_5))
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                  s0
## (Intercept) 70.8786
## Murder       0.0000
## Illiteracy   .
## HS.Grad      .
## Income       .
## Frost        .
## Area         .
## Population   .
```

```r
print(coef(fit_1))
```

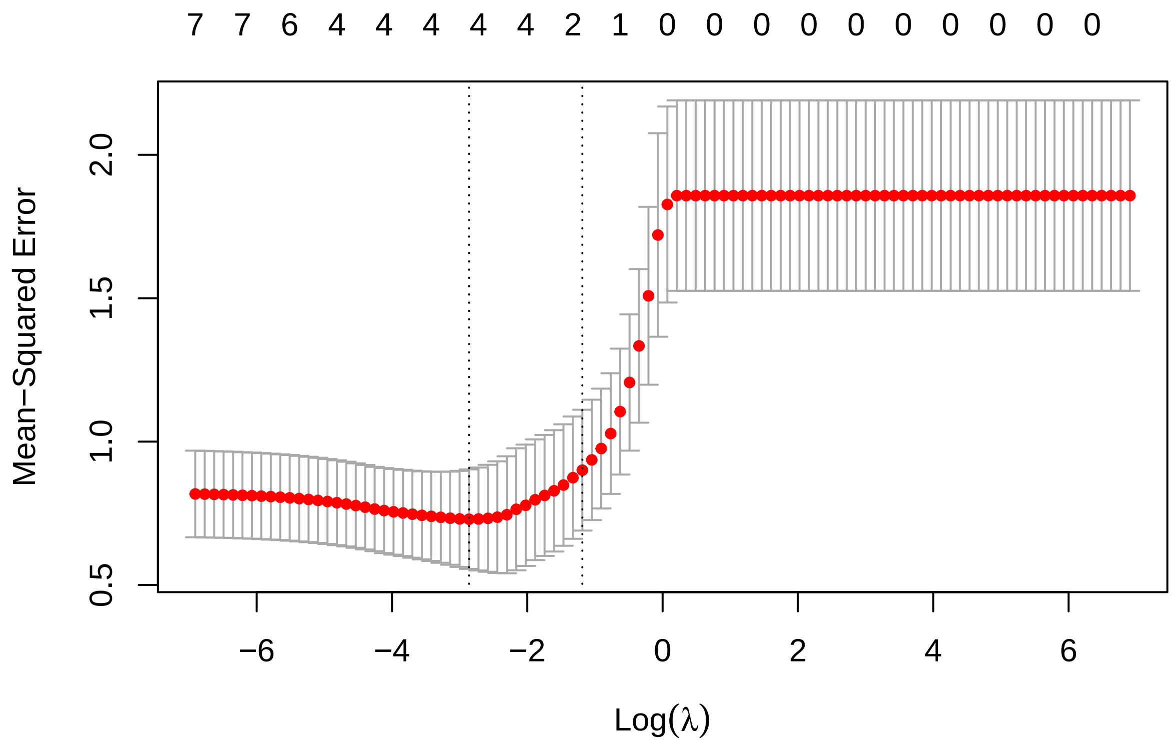
```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) 70.95464716
## Murder      -0.01030729
## Illiteracy   .
## HS.Grad      .
## Income       .
## Frost        .
## Area         .
## Population   .
```

```r
print(coef(fit_0.1))
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  7.086161e+01
## Murder      -2.432740e-01
## Illiteracy   .
## HS.Grad      3.592580e-02
## Income       .
## Frost       -1.934389e-03
## Area         .
## Population   2.495766e-05
```

```r
# Use cross-validation to choose the best lambda
set.seed(123)
cv_lasso <- cv.glmnet(X, y, alpha = 1, lambda = 10^seq(-3, 3, length = 100), nfolds = 5)

# Plot cross-validation results
plot(cv_lasso)
```




```r
lambda_min <- cv_lasso$lambda.min
print(paste("Best lambda:", lambda_min))
```

```
## [1] "Best lambda: 0.0572236765935022"
```

```r
# Refit LASSO with the best lambda
lasso_best <- glmnet(X, y, alpha = 1, lambda = lambda_min)
print(coef(lasso_best))
```

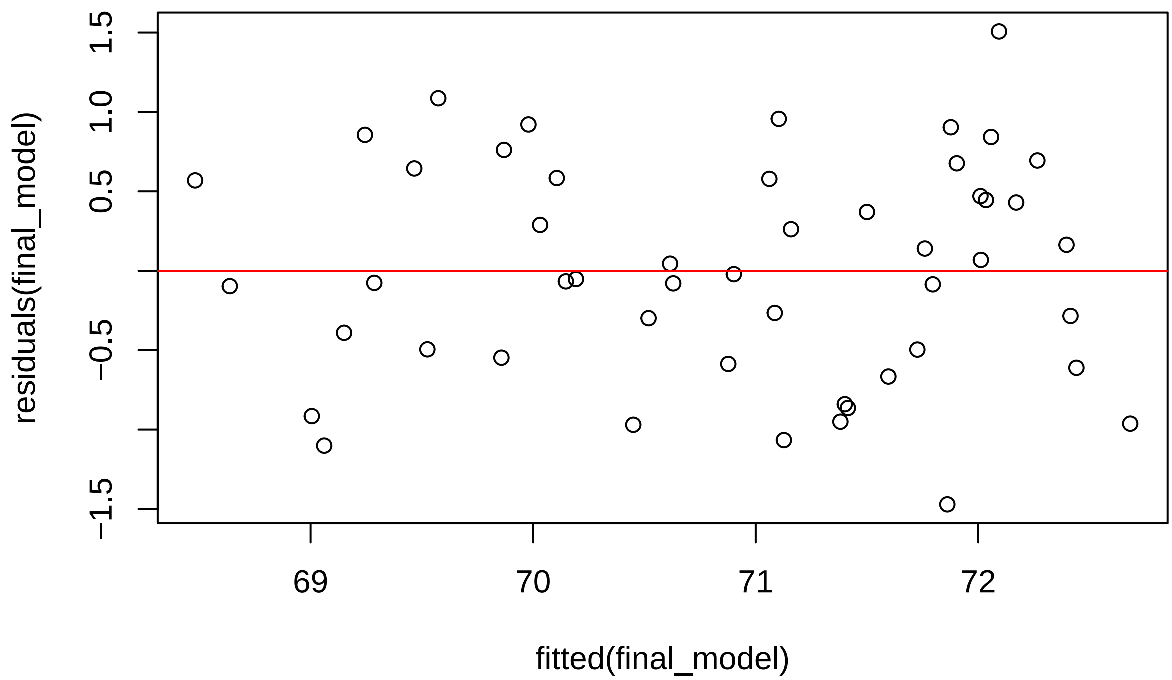
```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  7.093223e+01
## Murder      -2.675852e-01
## Illiteracy   .
## HS.Grad      4.048484e-02
## Income       .
## Frost       -3.648893e-03
## Area         .
## Population   3.572618e-05
```

For LASSO regression, we used cross-validation to determine the best lambda value, which controls the penalty for including less significant predictors. A range of lambda values was tested using the cv.glmnet() function, and the lambda that minimized the cross-validation error was selected. This optimal lambda was identified as the point with the lowest error on the cross-validation plot. Refitting the LASSO model using this lambda resulted in a sparse model, retaining only the most important predictors while shrinking less relevant coefficients to zero. The final set of predictors includes Murder, HS.Grad, and Population, demonstrating their importance in predicting life expectancy.

f) Subset comparison and Cross-Validation

Check model assumptions.

```
#Linearity
plot(fitted(final_model), residuals(final_model))
abline(h = 0, col = "red")
```




```
#Homoscedasticity
library(lmtest)
```

```
## Loading required package: zoo
```

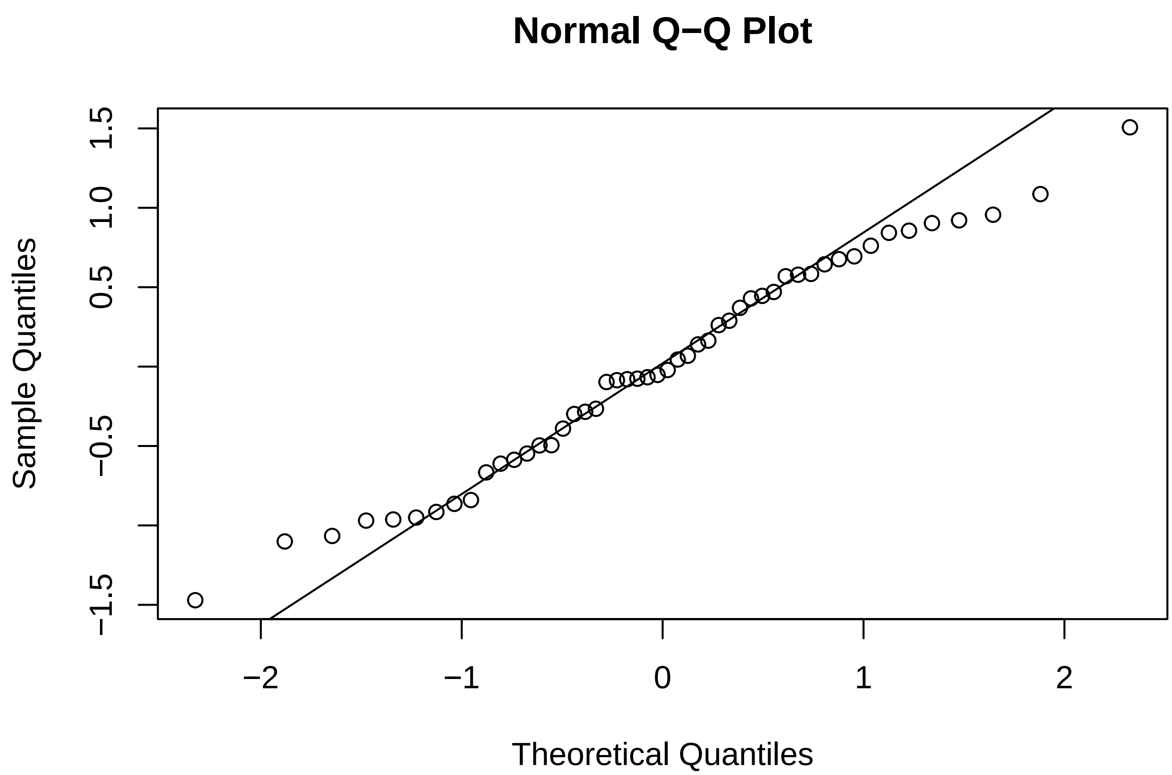
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
bptest(final_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  final_model
## BP = 6.2721, df = 4, p-value = 0.1797
```

```r
#Normality
qqnorm(residuals(final_model))
qqline(residuals(final_model))
```




```r
shapiro.test(residuals(final_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(final_model)
## W = 0.97935, p-value = 0.525
```

```r
#Multicollinearity
library(car)
```

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

```
##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```r
vif(final_model)
```

```
##     Murder    HS.Grad      Frost Population
##   1.727844   1.356791   1.498077   1.189835
```

10-fold cross validation

```r
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

## The following objects are masked from 'package:faraway':
##
##     logit, melanoma
```

```r
state_data <- na.omit(state_data)

# Define the final model formula
final_model_formula <- Life.Exp ~ Murder + HS.Grad + Frost + Population

# Perform 10-fold cross-validation
cv_results <- cv.glm(state_data, glm(final_model_formula, data = state_data), K = 10)

# Print cross-validated error
print(cv_results$delta)
```

```
## [1] 0.6098127 0.6017393
```

g) Summary

This analysis identifies the key factors influencing life expectancy across states. The final model includes crime rate (Murder), high school graduation rates (HS.Grad), population (Population), and climate (Frost) as significant predictors. These variables together explain most of the variability in life expectancy. Higher education levels and fewer frost days are positively associated with longer life expectancy, while higher crime rates reduce it. Diagnostic tests confirm that the model meets assumptions of linearity, normality, and constant variance, ensuring its reliability. Additionally, 10-fold cross-validation demonstrates strong predictive accuracy, meaning the model performs well on new data. Overall, improving education, reducing crime, and increasing economic opportunities are critical factors for enhancing life expectancy.

The resulting model is:

$$\text{Life.Exp} = \beta_0 + \beta_1 \cdot \text{Murder} + \beta_2 \cdot \text{HS.Grad} + \beta_3 \cdot \text{Frost} + \beta_4 \cdot \text{Population}$$