Homework 5

Pei Tian, pt2632

2023-12-10

Problem 1

```
library(faraway)
library(tidyverse)
library(patchwork)
library(corrplot)

theme_set(
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
)

life_expectency = state.x77 |> as_tibble() |> janitor::clean_names()
```

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) Provide descriptive statistics for all variables of interest (continuous and categorical) - no test required.

Variables:

- `Population:` population estimate as of July 1, 1975
- `Income:` per capita income (1974)
- `Illiteracy:` illiteracy (1970, percent of population)
- `Life Exp:` life expectancy in years (1969–71)
- `Murder:` murder and non-negligent manslaughter rate per 100,000 population (1976)
- `HS Grad:` percent high-school graduates (1970)
- `Frost:` mean number of days with minimum temperature below freezing (1931–1960) in capital or large city

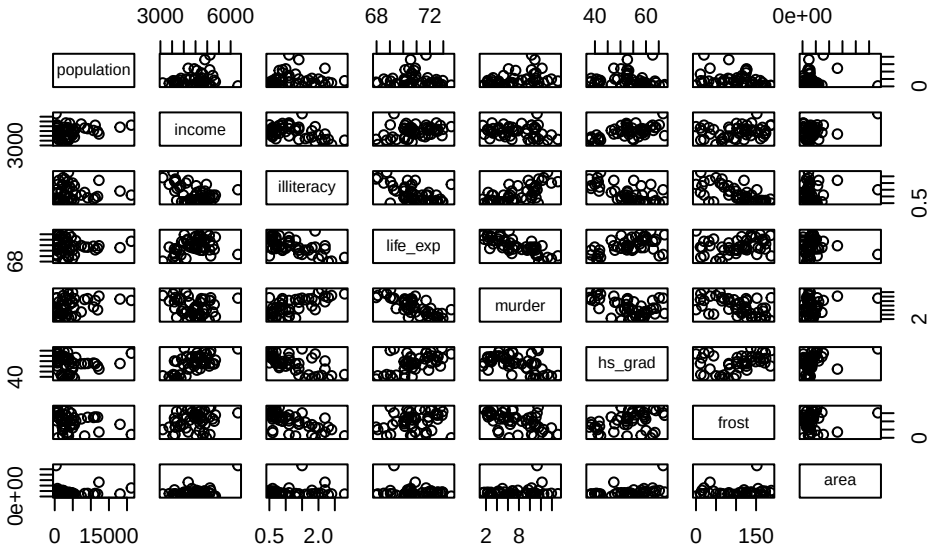
- `Area:` land area in square miles

All variables are continuous.

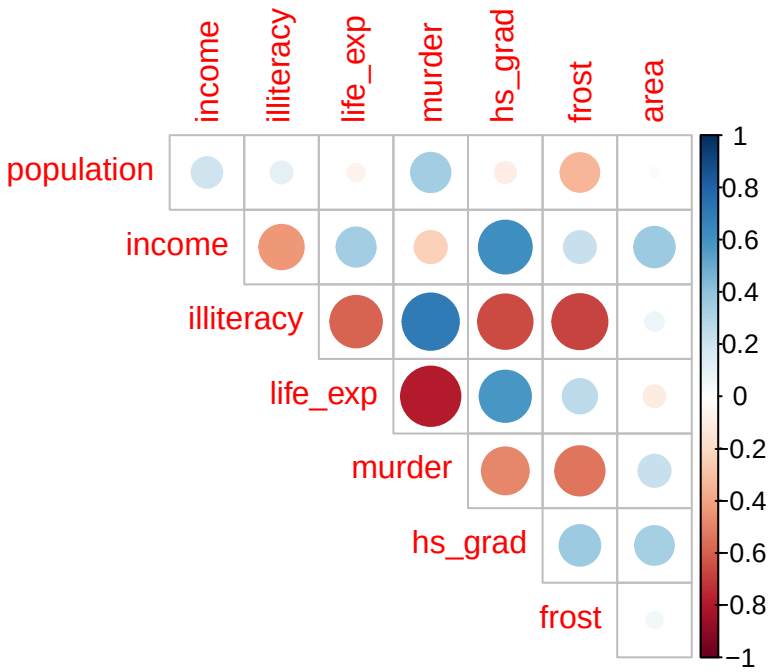
```
life_expectency |> summary()
```

```
   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
```

```
pairs(life_expectency)
```



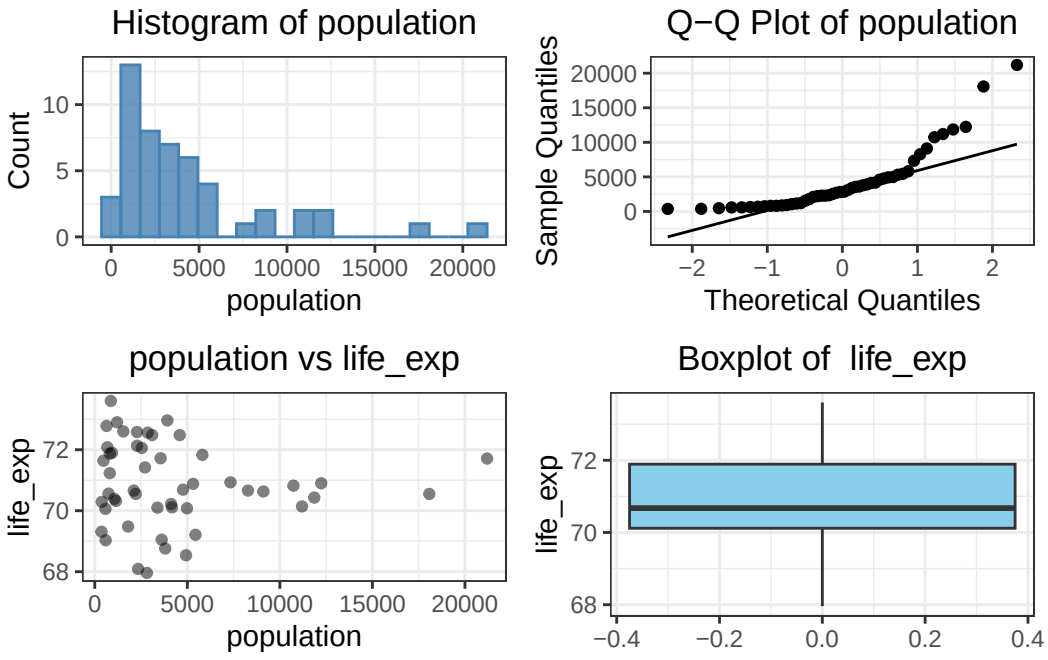
```r
corrplot(cor(life_expectency), type = "upper", diag = FALSE)
```

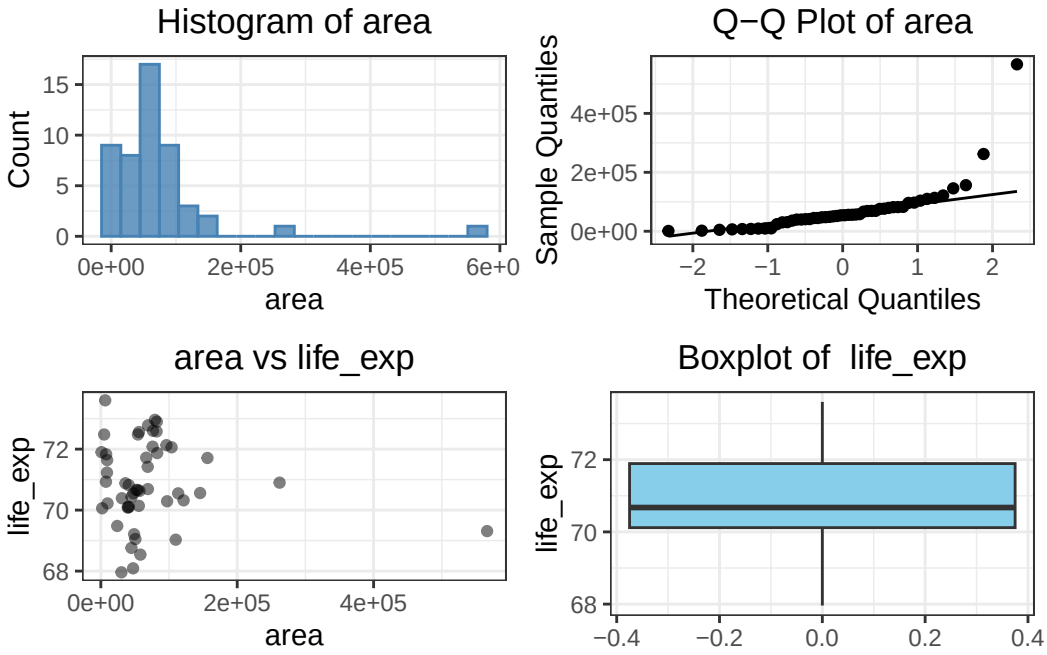


b) Examine exploratory plots, e.g., scatter plots, histograms, box-plots to get a sense of the data and possible variable transformations. (Be selective! Even if you create 20 plots, you don't want to show them all). If you find a transformation to be necessary or recommended, perform the transformation and use it through the rest of the problem.

```r
create_panel <- function(df, var1, var2) {
  # Histogram for var1
  histogram_var1 <- ggplot(df, aes_string(x = var1)) +
    geom_histogram(bins = 20, fill = "steelblue", color = "steelblue", alpha = .8) +
    labs(title = paste("Histogram of", var1), x = var1, y = "Count")

  # Q-Q plot for var1
  qqplot_var1 <- ggplot(data = df, aes_string(sample = var1)) +
    geom_qq() +
    geom_qq_line() +
    labs(title = paste("Q-Q Plot of", var1), x = "Theoretical Quantiles", y = "Sample Quan

  # Scatter plot
  scatter_plot <- ggplot(df, aes_string(x = var1, y = var2)) +
```

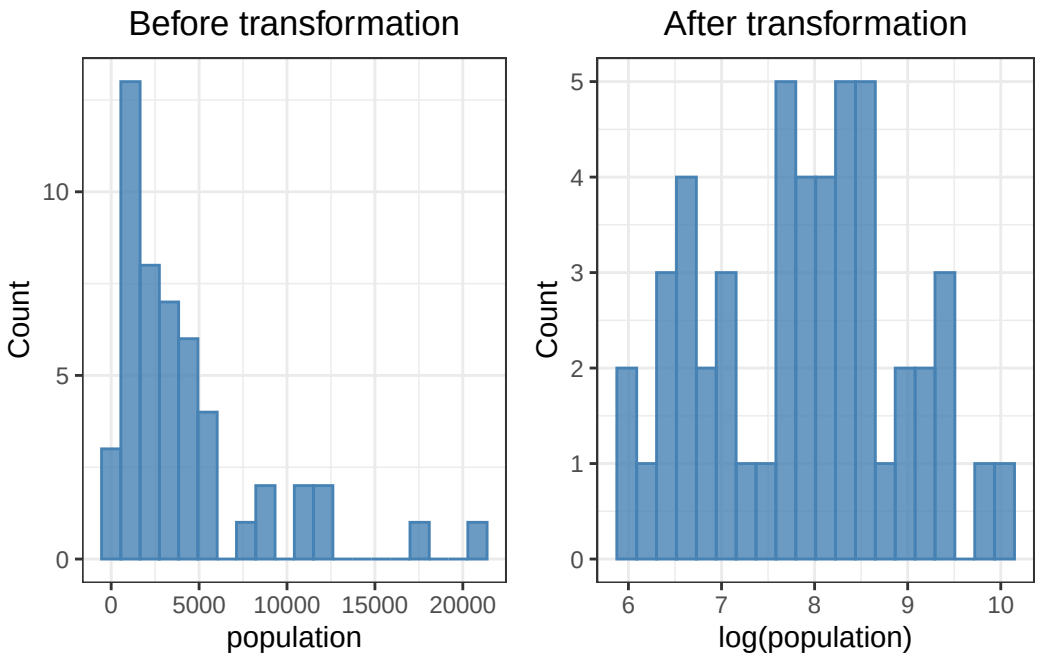
```
    geom_point(alpha = .5) +
    labs(title = paste(var1, "vs", var2),
         x = var1, y = var2)

  box_plot = ggplot(df, aes_string(y = var2)) +
    geom_boxplot(fill = "skyblue") +
    labs(title = paste("Boxplot of ", var2))


  # Arrange plots in a grid
  panel <- gridExtra::grid.arrange(histogram_var1, qqplot_var1, scatter_plot, box_plot, nc
}

target_var = "life_exp"
for(v in c("population", "area")){
  optm = create_panel(life_expectency, v, target_var)
}
```



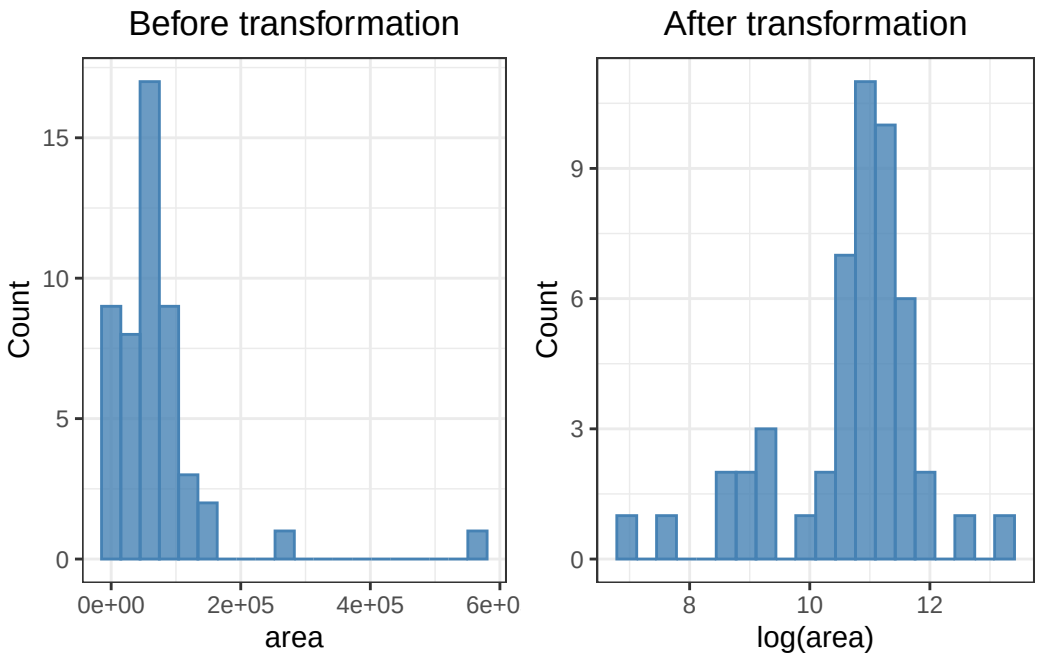


After observing the distributions of different variables, I choose to perform logarithm transformation to `population` and `area` variable to make the distribution of them more close to the normal distribution.

```
hist1 <-
  ggplot(life_expectency, aes_string(x = "population")) +
    geom_histogram(bins = 20, fill = "steelblue", color = "steelblue", alpha = .8) +
    labs(title = "Before transformation", x = "population", y = "Count")
hist2 <-
  ggplot(life_expectency, aes(x = population |> log1p())) +
    geom_histogram(bins = 20, fill = "steelblue", color = "steelblue", alpha = .8) +
    labs(title = "After transformation", x = "log(population)", y = "Count")
gridExtra::grid.arrange(hist1, hist2, ncol = 2, nrow = 1)
```




```
hist1 <-
  ggplot(life_expectency, aes_string(x = "area")) +
    geom_histogram(bins = 20, fill = "steelblue", color = "steelblue", alpha = .8) +
    labs(title = "Before transformation", x = "area", y = "Count")
hist2 <-
  ggplot(life_expectency, aes(x = area |> log1p())) +
    geom_histogram(bins = 20, fill = "steelblue", color = "steelblue", alpha = .8) +
    labs(title = "After transformation", x = "log(area)", y = "Count")
gridExtra::grid.arrange(hist1, hist2, ncol = 2, nrow = 1)
```




```
life_expectency = life_expectency |>
  mutate(log_population = log1p(population),
         log_area = log1p(area)) |>
  select(-population, -area)
```

c) Use automatic procedures to find a 'best subset' of the full model. Present the results and comment on the following:

Result:

- backward stepwise selection: $life\_exp = \beta_0 + \beta_1 * murder + \beta_2 * hs\_grad + \beta_3 * frost + \beta_4 * log(population)$
- forward stepwise selection: $life\_exp = \beta_0 + \beta_1 * murder + \beta_2 * hs\_grad + \beta_3 * frost + \beta_4 * log(population)$

```
mult.fit = lm(life_exp ~ ., data = life_expectency)
step(mult.fit, direction = "backward") |> summary()
```

```
Start:  AIC=-23.6
life_exp ~ income + illiteracy + murder + hs_grad + frost + log_population +
    log_area
```

```
                Df Sum of Sq    RSS      AIC
- income         1    0.0017 22.650 -25.5929
- illiteracy     1    0.0556 22.704 -25.4741
- log_area       1    0.2107 22.859 -25.1338
<none>                      22.648 -23.5968
- frost          1    1.2379 23.886 -22.9360
- log_population 1    1.8851 24.533 -21.5992
- hs_grad        1    2.4373 25.086 -20.4864
- murder         1   23.2771 45.926   9.7499

Step:  AIC=-25.59
life_exp ~ illiteracy + murder + hs_grad + frost + log_population +
    log_area

                Df Sum of Sq    RSS      AIC
- illiteracy     1    0.0556 22.706 -27.4704
- log_area       1    0.2198 22.870 -27.1100
<none>                      22.650 -25.5929
- frost          1    1.2607 23.911 -24.8847
- log_population 1    2.1907 24.841 -22.9768
- hs_grad        1    4.0368 26.687 -19.3925
- murder         1   24.2136 46.864   8.7611

Step:  AIC=-27.47
life_exp ~ murder + hs_grad + frost + log_population + log_area

                Df Sum of Sq    RSS     AIC
- log_area       1    0.2158 22.922 -28.997
<none>                      22.706 -27.470
- log_population 1    2.2790 24.985 -24.688
- frost          1    2.3768 25.082 -24.493
- hs_grad        1    4.9482 27.654 -19.613
- murder         1   29.2319 51.938  11.901

Step:  AIC=-29
life_exp ~ murder + hs_grad + frost + log_population

                Df Sum of Sq    RSS     AIC
<none>                      22.922 -28.997
- frost          1     2.215 25.136 -26.385
- log_population 1     2.450 25.372 -25.920
- hs_grad        1     6.958 29.880 -17.742
- murder         1    34.111 57.033  14.580
```

```
Call:
lm(formula = life_exp ~ murder + hs_grad + frost + log_population,
    data = life_expectency)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4176 -0.4390  0.0254  0.5207  1.6304

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.720122   1.417117  48.493  < 2e-16 ***
murder         -0.290028   0.035441  -8.183 1.87e-10 ***
hs_grad         0.054546   0.014758   3.696 0.000592 ***
frost          -0.005175   0.002482  -2.085 0.042748 *
log_population  0.246955   0.112601   2.193 0.033502 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7137 on 45 degrees of freedom
Multiple R-squared:  0.7404,    Adjusted R-squared:  0.7173
F-statistic: 32.09 on 4 and 45 DF,  p-value: 1.171e-12
```

```
intercept.fit = lm(life_exp ~ 1, data = life_expectency)
step(intercept.fit, direction = "forward", scope = formula(mult.fit)) |> summary()
```

```
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
+ log_population 1     1.055 87.244  31.834
+ log_area       1     1.042 87.257  31.842

Step:  AIC=-14.61
life_exp ~ murder
```

```
                 Df Sum of Sq    RSS     AIC
+ hs_grad         1    4.6910 29.770 -19.925
+ frost           1    3.1346 31.327 -17.378
+ log_population  1    2.9858 31.476 -17.141
+ income          1    2.4047 32.057 -16.226
+ log_area        1    1.4583 33.003 -14.771
<none>                       34.461 -14.609
+ illiteracy      1    0.2732 34.188 -13.007

Step:  AIC=-19.93
life_exp ~ murder + hs_grad

                 Df Sum of Sq    RSS     AIC
+ log_population  1    4.6339 25.136 -26.385
+ frost           1    4.3987 25.372 -25.920
<none>                       29.770 -19.925
+ illiteracy      1    0.4419 29.328 -18.673
+ log_area        1    0.1236 29.647 -18.134
+ income          1    0.1022 29.668 -18.097

Step:  AIC=-26.39
life_exp ~ murder + hs_grad + log_population

             Df Sum of Sq    RSS     AIC
+ frost       1   2.21489 22.922 -28.997
+ illiteracy  1   1.10777 24.029 -26.639
<none>                    25.136 -26.385
+ income      1   0.11832 25.018 -24.621
+ log_area    1   0.05391 25.082 -24.493

Step:  AIC=-29
life_exp ~ murder + hs_grad + log_population + frost

             Df Sum of Sq    RSS     AIC
<none>                    22.922 -28.997
+ log_area    1  0.215823 22.706 -27.470
+ illiteracy  1  0.051581 22.870 -27.110
+ income      1  0.010701 22.911 -27.021


Call:
lm(formula = life_exp ~ murder + hs_grad + log_population + frost,
    data = life_expectency)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.4176 -0.4390  0.0254  0.5207  1.6304

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.720122   1.417117  48.493  < 2e-16 ***
murder         -0.290028   0.035441  -8.183 1.87e-10 ***
hs_grad         0.054546   0.014758   3.696 0.000592 ***
log_population  0.246955   0.112601   2.193 0.033502 *
frost          -0.005175   0.002482  -2.085 0.042748 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7137 on 45 degrees of freedom
Multiple R-squared:  0.7404,    Adjusted R-squared:  0.7173
F-statistic: 32.09 on 4 and 45 DF,  p-value: 1.171e-12
```

- Do the procedures generate the same model?

  Yes.

- Are any variables a close call? What was your decision: keep or discard? Provide arguments for your choice. (Note: this question might have more or less relevance depending on the 'subset' you choose).

  When I manually deliver "backward" stepwise selection procedure, I found the `frost` variable is a close call with p-value as 0.043. As for the decision, I finally choose to keep this variable because the model's adjusted r-square will decrease after I remove `frost` from predictors subset.

```r
mult.fit = lm(life_exp ~ ., data = life_expectency)
summary(mult.fit)
```

```
Call:
lm(formula = life_exp ~ ., data = life_expectency)

Residuals:
     Min       1Q   Median       3Q      Max
-1.43084 -0.45557  0.02759  0.49621  1.70216

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.795e+01  2.093e+00  32.472  < 2e-16 ***
income          1.392e-05  2.444e-04   0.057   0.9549
illiteracy      1.126e-01  3.507e-01   0.321   0.7497
murder         -3.092e-01  4.706e-02  -6.570 6.01e-08 ***
hs_grad         5.278e-02  2.482e-02   2.126   0.0394 *
frost          -4.870e-03  3.214e-03  -1.515   0.1372
log_population  2.528e-01  1.352e-01   1.870   0.0685 .
log_area        6.863e-02  1.098e-01   0.625   0.5353
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7343 on 42 degrees of freedom
Multiple R-squared:  0.7435,    Adjusted R-squared:  0.7008
F-statistic: 17.39 on 7 and 42 DF,  p-value: 1.434e-10
```

```r
step = update(mult.fit, . ~ . - income)
step = update(step, . ~ . - illiteracy)
step = update(step, . ~ . - log_area)
summary(step)
```

```
Call:
lm(formula = life_exp ~ murder + hs_grad + frost + log_population,
    data = life_expectency)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4176 -0.4390  0.0254  0.5207  1.6304

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.720122   1.417117  48.493  < 2e-16 ***
murder         -0.290028   0.035441  -8.183 1.87e-10 ***
hs_grad         0.054546   0.014758   3.696 0.000592 ***
frost          -0.005175   0.002482  -2.085 0.042748 *
log_population  0.246955   0.112601   2.193 0.033502 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7137 on 45 degrees of freedom
Multiple R-squared:  0.7404,    Adjusted R-squared:  0.7173
```

```
F-statistic: 32.09 on 4 and 45 DF,  p-value: 1.171e-12
```

- Is there any association between 'Illiteracy' and 'HS graduation rate'? Does your 'subset' contain both?

  From the correlation heatmap, we can see that 'illteracy' and 'HS graduation rate' are negatively related (correlation coefficient = -0.65)

  All models don't contain both of them.

d) Use criterion-based procedures to guide your selection of the 'best subset'. Summarize your results (tabular or graphical).

```
X = life_expectency |> select(-life_exp)
y = life_expectency |> pull(life_exp)

leaps::leaps(
  x = X,
  y = y,
  nbest = 2,
  method = "Cp"
)
```

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

$label
[1] "(Intercept)" "1"           "2"           "3"           "4"
[6] "5"           "6"           "7"

$size
```

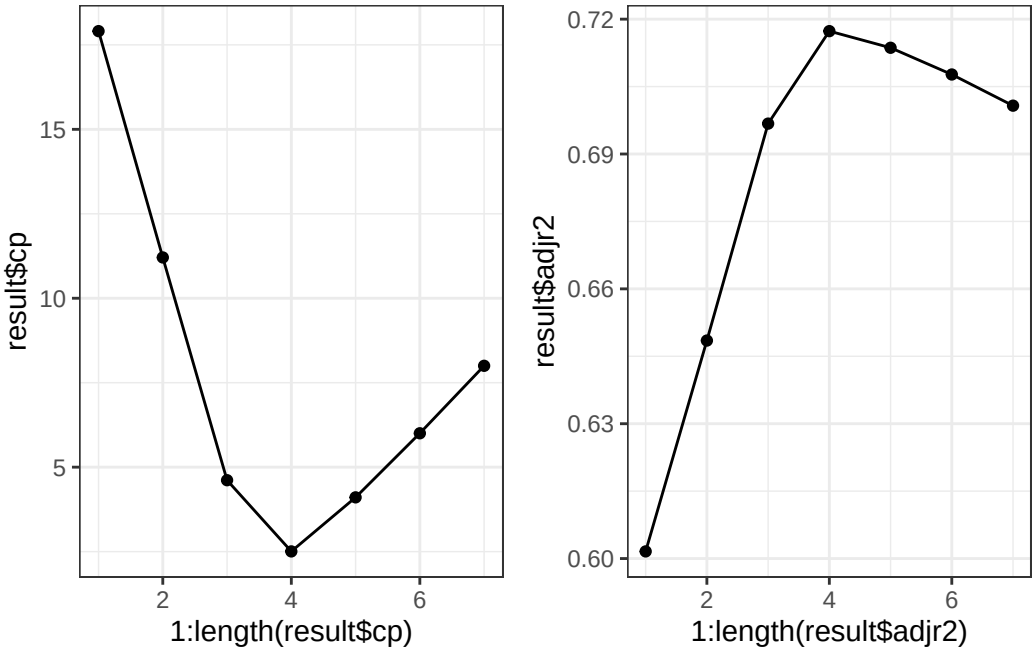
```
 [1] 2 2 3 3 4 4 5 5 6 6 7 7 8
```

```
$Cp
 [1] 17.906417 61.039134 11.207299 14.093450  4.613928  5.050122  2.506544
 [8]  4.559645  4.106313  4.410890  6.003242  6.103129  8.000000
```

```
leaps::leaps(
  x = X,
  y = y,
  nbest = 2,
  method = "adjr2"
)
```

```
$which
      1     2     3     4     5     6     7
1 FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
1 FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
2 FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
2 FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
3 FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE
3 FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
4 FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
4 FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
5 FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
5 FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
6 FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
6  TRUE FALSE  TRUE  TRUE  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 0.6301232 0.6967606 0.6939230 0.7173356
 [8] 0.7036827 0.7136334 0.7115620 0.7076910 0.7069959 0.7007544
```

```
result = leaps::regsubsets(
  x = X,
  y = y,
  nbest = 1
) |> summary()
gridExtra::grid.arrange(
  ggplot(aes(x = 1:length(result$cp), y = result$cp), data = NULL) + geom_point() + g
  ggplot(aes(x = 1:length(result$adjr2), y = result$adjr2), data = NULL) + geom_point
)
```



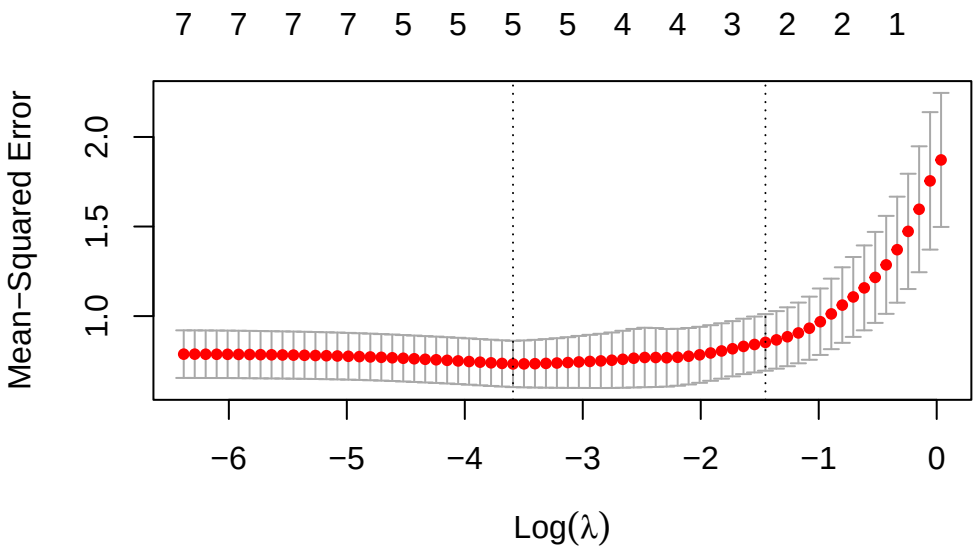

```
colnames(result$which)[result$which[4,]]
```

```
[1] "(Intercept)"    "murder"         "hs_grad"        "frost"
[5] "log_population"
```

According to the Cp's criterion and adjusted R-square, the final model is $life\_exp = \beta_0 + \beta_1 * murder + \beta_2 * hs\_grad + \beta_3 * frost + \beta_4 * log(population)$, which is the same as model selected by procedure-based procedure.

e) Use the LASSO method to perform variable selection. Make sure you choose the 'best lambda' to use and show how you determined this.

```
cv.lasso = glmnet::cv.glmnet(x = X |> as.matrix(), y = y, alpha = 1)
plot(cv.lasso)
```




```
lasso = glmnet::glmnet(X |> as.matrix(), y, alpha = 1, lambda = cv.lasso$lambda.min)
coef(lasso)
```

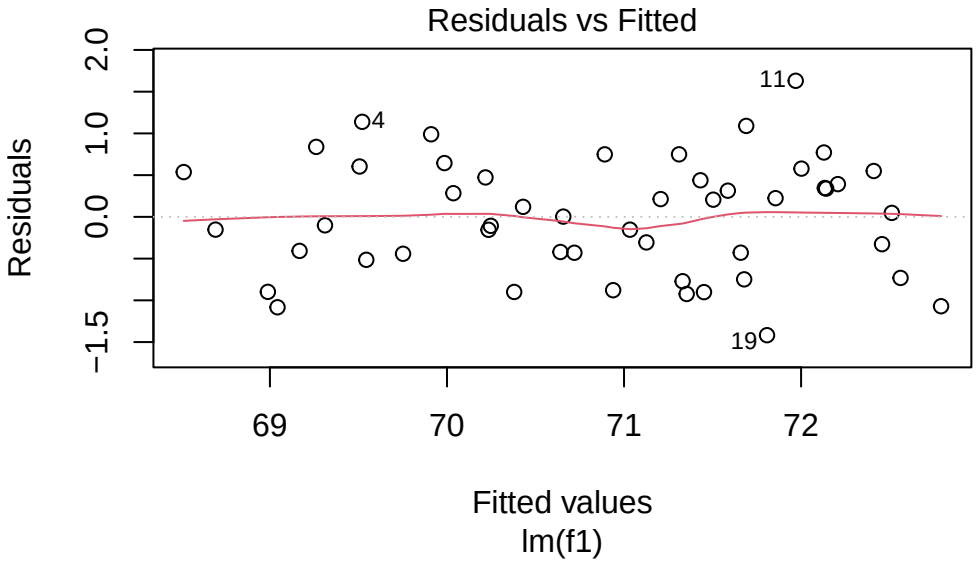
```
8 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept)    68.84063960
income          .
illiteracy      .
murder         -0.28078379
hs_grad         0.04877422
frost          -0.00426217
log_population  0.21221922
log_area        0.02772341
```

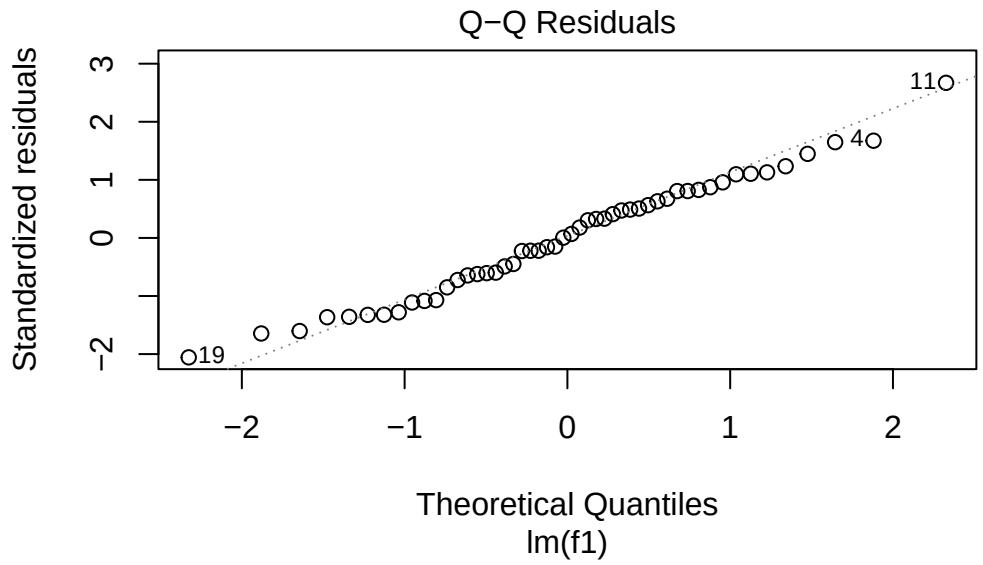
Given the result of lasso model with optimized λ parameter, the final model is $life\_exp = \beta_0 + \beta_1 * murder + \beta_2 * hs\_grad + \beta_3 * frost + \beta_4 * log(population) + \beta_5 * log(area)$

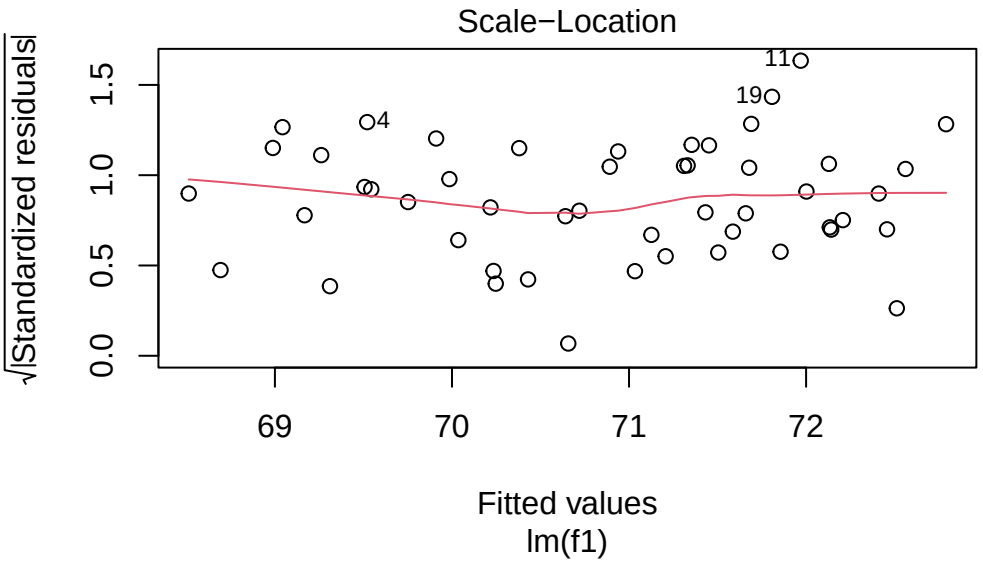
f) Compare the 'subsets' from parts c, d, and e and recommend a 'final' model. Using this 'final' model do the following:

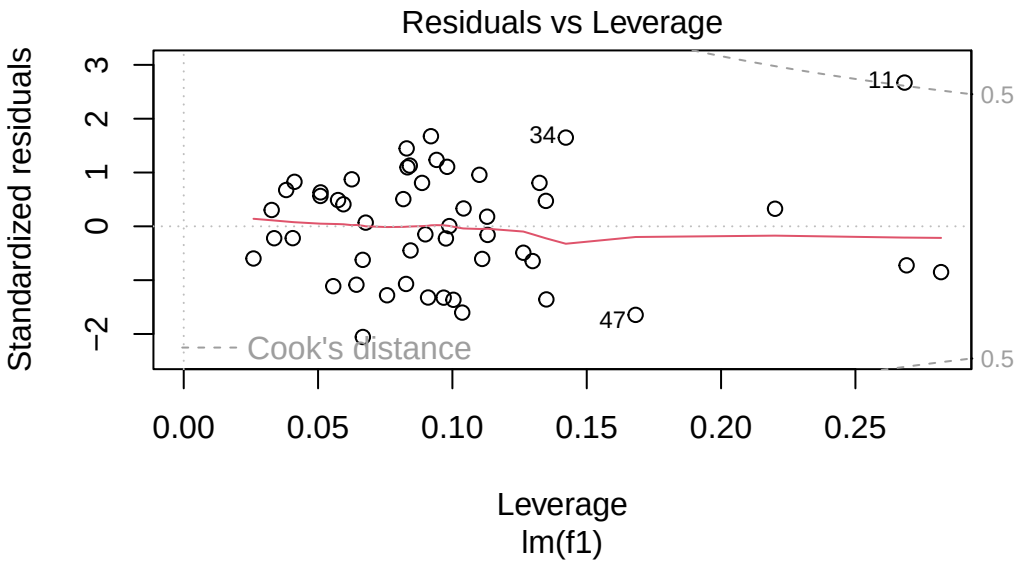
- Check the model assumptions.

```
f1 = formula(life_exp ~ murder + frost + hs_grad + log_population)
f2 = formula(life_exp ~ murder + frost + hs_grad + log_population + log_area)
model1 = lm(f1, data = life_expectency)
model2 = lm(f2, data = life_expectency)
plot(model1)
```

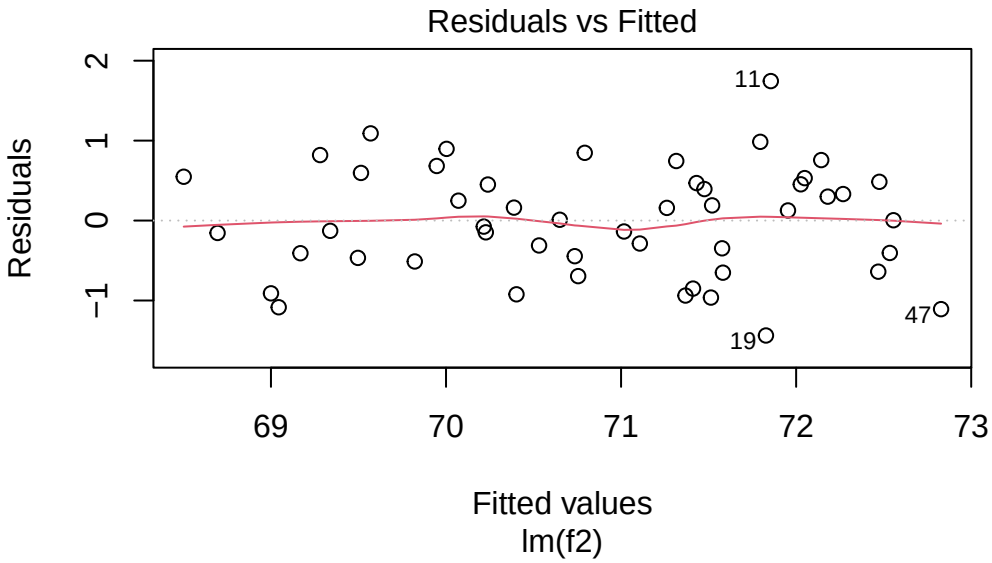


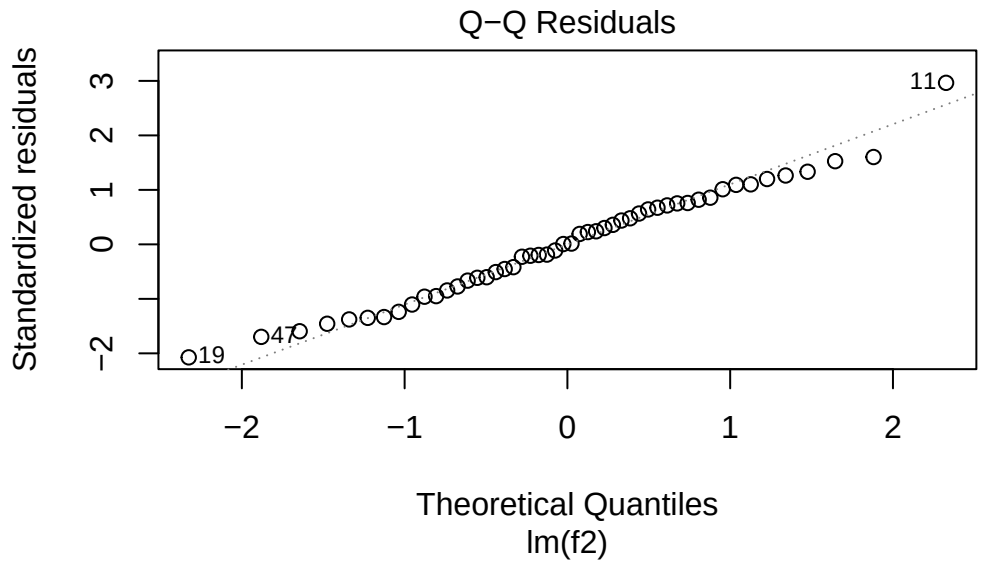

Q−Q Residuals
Standardized residuals
−2
0
1
2
3
19
4
11
−2
−1
0
1
2
Theoretical Quantiles
lm(f1)

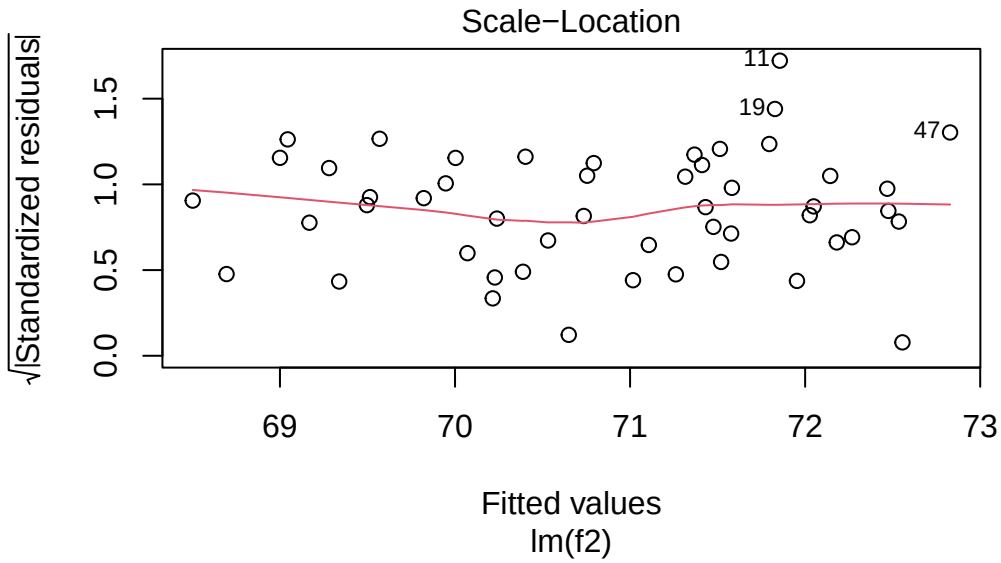

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
11
19
4
69
70
71
72
Fitted values
lm(f1)

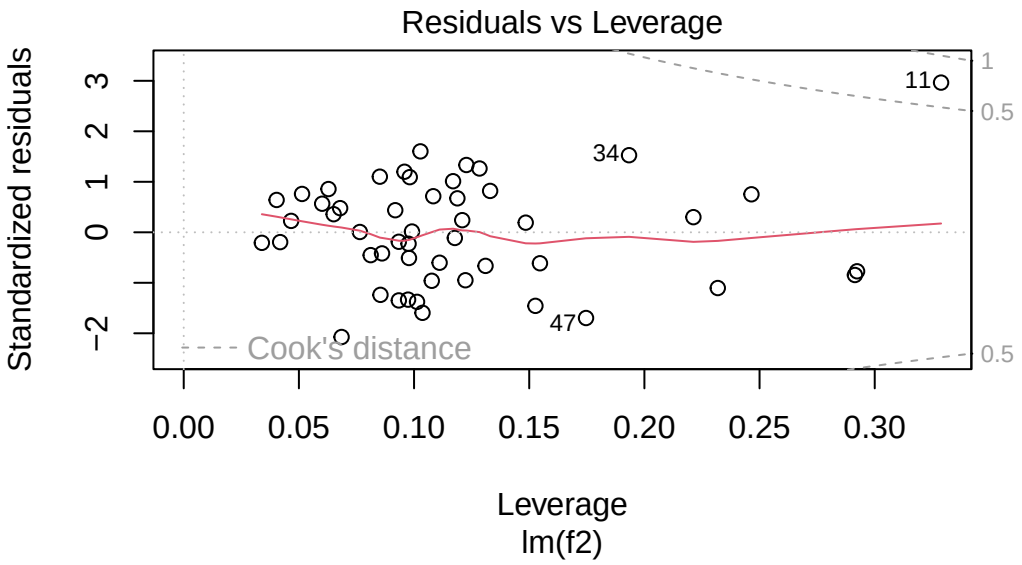

Residuals vs Leverage
Standardized residuals
3
2
1
0
−2
11
34
47
0.5
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(f1)

```
plot(model2)
```


Residuals vs Fitted
Residuals
2
1
0
−1
11
47
19
69
70
71
72
73
Fitted values
lm(f2)


Q−Q Residuals
Standardized residuals
−2
0
1
2
3
19
47
11
−2
−1
0
1
2
Theoretical Quantiles
lm(f2)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
11
19
47
69
70
71
72
73
Fitted values
lm(f2)



From the diagnostic plots, I conclude that both models satisify the assumptions related to the residuals. Given the residual vs fitted value plots consist of points bounce around 0, so the assumption about homoscedasticity is met. In addition, the points QQ-plot is nearly fitting to a line, so the assumption about normality is met. Finally, the

- Test the model predictive ability using a 10-fold cross-validation.

```
mse = function(train, test, formula, slot = "life_exp"){
  model = lm(formula, data = train)
  true = test[, slot]
  pred = predict(model, data = test)
  sum((pred - true)^2) / length(true)
}
cross_data = modelr::crossv_kfold(life_expectency, k = 10) |>
  mutate(train = map(train, as_tibble),
         test = map(test, as_tibble),
         result1 = map2(train, test, mse, formula = f1),
         result2 = map2(train, test, mse, formula = f2)) |>
  unnest(result1, result2)
```

From the cross-validation result, 2 models have nearly same performance according to the evaluation result on test datasets, given close mean MSE of different model.

- Mean MSE for model selected by stepwise-based/criterion-based procedure: 18.5934199
- Mean MSE for model selected by lasso: 18.3938063

g) In a paragraph, summarize your findings to address the primary question posed by the investigator (that has limited statistical knowledge).

For the goal of predicting `life_exp` using a combination of variables, I contend that `life_exp` could be predicted by variables including `population`, `hs_grad`, `murder` and `frost`, given its good performance in validation of linear model with them as predictors. Also, the criterion and significance level also indicate this linear model is the most reasonable and satisfying one among all linear models.