Homework 5 SOLUTIONS

P8130 Fall 2022

Due: December 5, 2022 at midnight Eastern

P8130 Guidelines for Submitting Homework

- Your homework must be submitted through Courseworks. No email submissions!
- Only one PDF file should be submitted, including all derivations, graphs, output, and interpretations. When handwriting is allowed (this will be specified), scan the derivations and merge ALL PDF files (http: //www.pdfmerge.com/).
- You are encouraged to use R for calculations, but you must show all mathematical formulas and derivations. Please include the important parts of your R code in the PDF file but also submit your full, commented code as a separate R/RMD file.
- To best follow these guidelines, we suggest using Word (built in equation editor), R Markdown, Latex, or embedding a screenshot or scanned picture to compile your work.

DO NOT FORGET: You are encouraged to collaborate on homeworks, explain things to each other, and test each other's knowledge. But Do NOT hand out answers to someone who has not done any work. Everyone ought to have ideas about the possible answers or at least some thoughts about how to probe the problem further. Write your own solutions!

```r
library(tidyverse)
library(GGally)
library(patchwork)
library(gt)
library(leaps)
library(caret)
```

Problem 1 (30 points)

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.

```r
library(faraway)
dat.state <- as.data.frame(state.x77)
head(dat.state)
```

```
           Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

a) Provide descriptive statistics for all variables of interest – no test required.

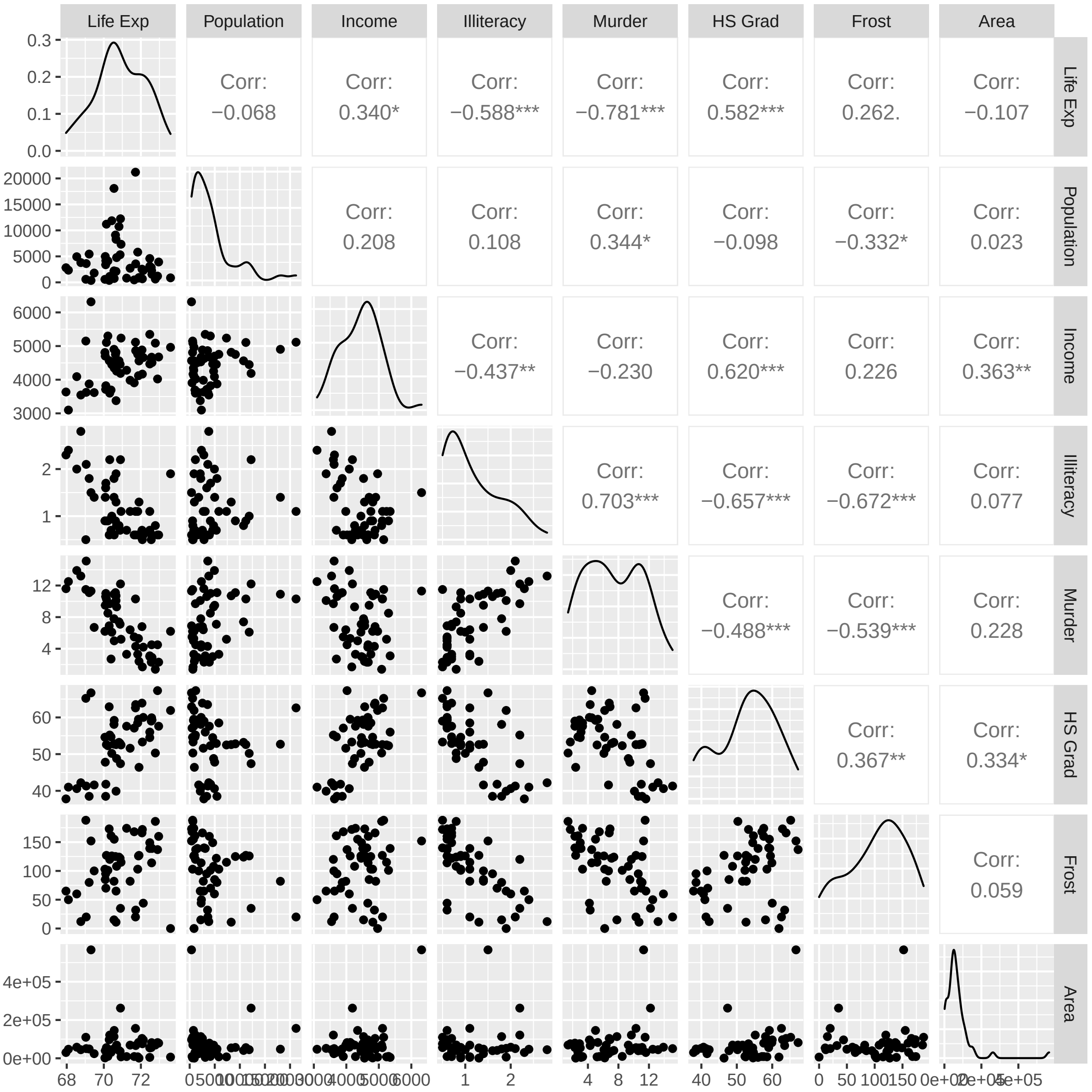
```r
dat.state |>
  gtsummary::tbl_summary() |>
  gtsummary::bold_labels()
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 50**
Population	2,838 (1,080, 4,968)
Income	4,519 (3,993, 4,814)
Illiteracy	0.95 (0.63, 1.58)
Life Exp	70.67 (70.12, 71.89)
Murder	6.8 (4.3, 10.7)
HS Grad	53 (48, 59)
Frost	114 (66, 140)
Area	54,277 (36,985, 81,162)

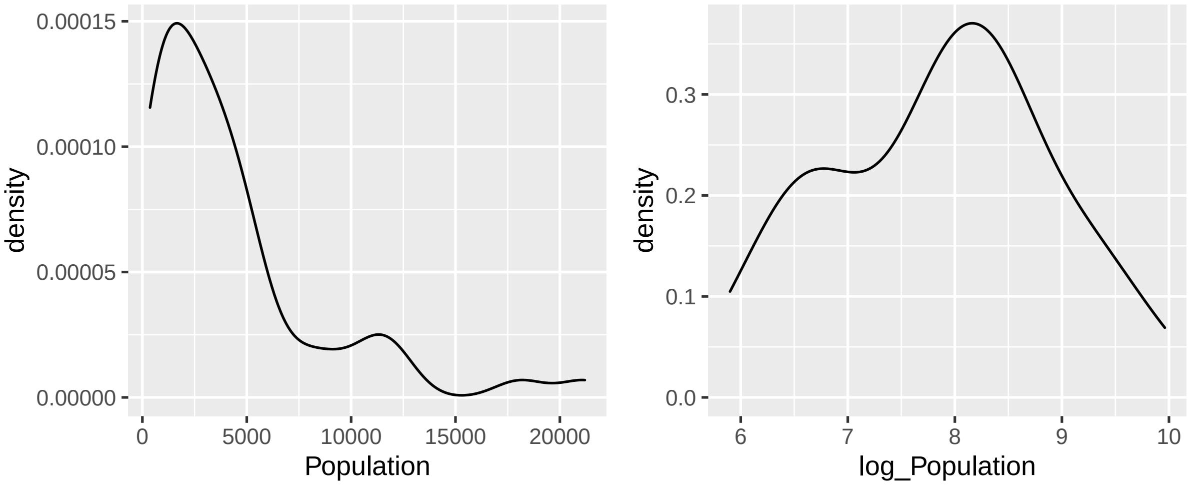
b) Examine exploratory plots, e.g., scatter plots, histograms, boxplots to get a sense of the data and possible variable transformations. (Be selective! Even if you create 20 plots, you don't want to show them all). If you find a transformation to be necessary or recommended, perform the transformation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

```
# plot scatterplots between each pair variables,
# density plots for each varaible, and correlations between each pair
dat.state |>
  relocate(`Life Exp`) |>
  ggpairs()
```



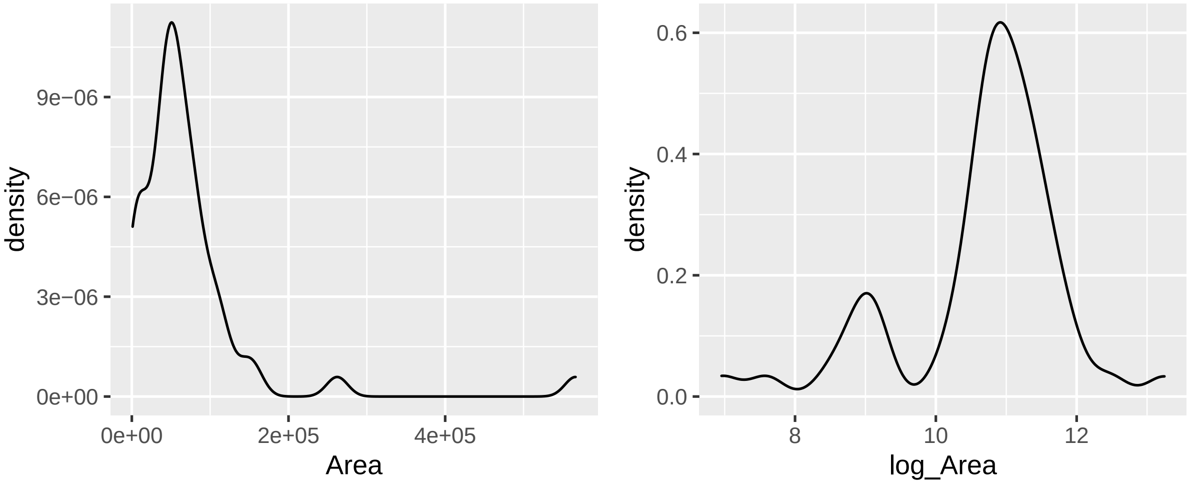
From the plots above, we can see that population and area are severely right skewed. We should look into transforming these variables. (Illiteracy has some skew, but not enough for me to warrant a transformation.) This matrix of plots also shows the correlation between each variable. From this, we can see that life expectancy is moderately correlated with illiteracy, murder, and HS grad.

```
## look for appropriate transformations

density_plot_population <-
  dat.state |>
  ggplot(aes(x = Population)) +
  geom_density()

density_plot_logpopulation <-
  dat.state |>
  mutate(log_Population = log(Population)) |>
  ggplot(aes(x = log_Population)) +
  geom_density()

density_plot_population + density_plot_logpopulation
```



```
density_plot_area <-
  dat.state |>
  ggplot(aes(x = Area)) +
  geom_density()

density_plot_logarea <-
  dat.state |>
  mutate(log_Area = log(Area)) |>
  ggplot(aes(x = log_Area)) +
  geom_density()
```

```
density_plot_area + density_plot_logarea
```



From the plots above, we can see that the log transformation helps to “normalize” the variables Population and Area. We will use these transformations instead of the raw variable in our model building.

```
# add log population and log area to the data frame
dat.state <-
  dat.state |>
  mutate(log_Population = log(Population)) |>
  mutate(log_Area = log(Area))
```

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

- **Do the procedures generate the same model?**
- **Is there any variable 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).**
- **Is there any association between ‘Illiteracy’ and ‘HS graduation rate’? Does your ‘subset’ contain both?**

First: Forward Selection

Since I want to use p-values to determine if a variable gets added, I must do this manually.

```r
variables <- c("log_Population", "Income", "Illiteracy", "Murder",
               "HS Grad", "Frost", "log_Area")

## fit SLR for each variable - keep the variable with the lowest p-value
df_results1 <- data.frame(variable = NA,
                          estimate = NA,
                          std.error = NA,
                          statistic = NA,
                          p.val = NA)

# for loop to run univariate LS for each variable and save results
for(var in variables){
  fit <- lm(dat.state$`Life Exp` ~ dat.state[[var]])
  fit_sum <- summary(fit)

  temp_results <- c(var, fit_sum$coefficients[2,])

  df_results1 <- rbind(df_results1, temp_results)
}

df_results1 <- df_results1[-1,] #remove initial rows of NAs

# print table of results
df_results1 |>
  mutate(across(-variable, ~as.numeric(.))) |>
  arrange(p.val) |>
  gt() |>
  fmt_number(
    columns = 2:5,
    decimals = 4
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = p.val,
      rows = p.val < 0.05
    )
  )
```

variable	estimate	std.error	statistic	p.val
Murder	−0.2839	0.0328	−8.6596	0.0000
Illiteracy	−1.2960	0.2570	−5.0427	0.0000
HS Grad	0.0968	0.0195	4.9613	0.0000
Income	0.0007	0.0003	2.5069	0.0156
Frost	0.0068	0.0036	1.8814	0.0660
log_Population	−0.1408	0.1849	−0.7616	0.4501
log_Area	−0.1248	0.1649	−0.7571	0.4527

Since this table is sorted by p-value (smallest at the top), the first variable we will add to the model is murder.

```
# fit models with 2 variables, keeping murder in the model each time
variables <- c("log_Population", "Income", "Illiteracy",
               "HS Grad", "Frost", "log_Area")

df_results2 <- data.frame(variable = NA,
                          estimate = NA,
                          std.error = NA,
                          statistic = NA,
                          p.val = NA)

# for loop to run univariate LS for each variable and save results
for(var in variables){
  fit <- lm(dat.state$`Life Exp` ~ dat.state$Murder + dat.state[[var]])
  fit_sum <- summary(fit)

  temp_results <- c(var, fit_sum$coefficients[3,])

  df_results2 <- rbind(df_results2, temp_results)
}

df_results2 <- df_results2[-1,] #remove initial rows of NAs

# print table of results
df_results2 |>
```

```r
  mutate(across(-variable, ~as.numeric(.))) |>
  arrange(p.val) |>
  gt() |>
  fmt_number(
    columns = 2:5,
    decimals = 4
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = p.val,
      rows = p.val < 0.05
    )
  )
```

variable	estimate	std.error	statistic	p.val
HS Grad	0.0439	0.0161	2.7214	0.0091
Frost	−0.0058	0.0027	−2.1686	0.0352
log_Population	0.2540	0.1203	2.1114	0.0401
Income	0.0004	0.0002	1.8777	0.0666
log_Area	0.1546	0.1073	1.4411	0.1562
Illiteracy	−0.1723	0.2811	−0.6129	0.5429

HS graduation has the smallest p-value (and is still less than 0.05), so it will be the next variable we add to the model.

```r
# fit models with 3 variables, keeping murder and HS grad in the model each time
variables <- c("log_Population", "Income", "Illiteracy",
               "Frost", "log_Area")

df_results3 <- data.frame(variable = NA,
                          estimate = NA,
                          std.error = NA,
                          statistic = NA,
                          p.val = NA)

# for loop to run univariate LS for each variable and save results
```

```
for(var in variables){
  fit <- lm(dat.state$`Life Exp` ~ dat.state$Murder + dat.state$`HS Grad` + dat.state[[var
  fit_sum <- summary(fit)

  temp_results <- c(var, fit_sum$coefficients[4,])

  df_results3 <- rbind(df_results3, temp_results)
}

df_results3 <- df_results3[-1,] #remove initial rows of NAs

# print table of results
df_results3 |>
  mutate(across(-variable, ~as.numeric(.))) |>
  arrange(p.val) |>
  gt() |>
  fmt_number(
    columns = 2:5,
    decimals = 4
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = p.val,
      rows = p.val < 0.05
    )
  )
```

variable	estimate	std.error	statistic	p.val
log_Population	0.3217	0.1105	2.9125	0.0055
Frost	−0.0069	0.0024	−2.8240	0.0070
Illiteracy	0.2540	0.3051	0.8325	0.4094
log_Area	0.0494	0.1127	0.4380	0.6634
Income	0.0001	0.0002	0.3981	0.6924

Population (the log-transformed version) is the variable with the smallest p-value when considering 3-variable models. We will now include it in the next step of forward selection.

```r
# fit models with 4 variables, keeping murder, HS grad, log_Population in the
# model each time
variables <- c("Income", "Illiteracy", "Frost", "log_Area")

df_results4 <- data.frame(variable = NA,
                          estimate = NA,
                          std.error = NA,
                          statistic = NA,
                          p.val = NA)

# for loop to run univariate LS for each variable and save results
for(var in variables){
  fit <- lm(dat.state$`Life Exp` ~ dat.state$Murder + dat.state$`HS Grad` +
              dat.state$log_Population + dat.state[[var]])
  fit_sum <- summary(fit)

  temp_results <- c(var, fit_sum$coefficients[5,])

  df_results4 <- rbind(df_results4, temp_results)
}

df_results4 <- df_results4[-1,] #remove initial rows of NAs

# print table of results
df_results4 |>
  mutate(across(-variable, ~as.numeric(.))) |>
  arrange(p.val) |>
  gt() |>
  fmt_number(
    columns = 2:5,
    decimals = 4
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = p.val,
      rows = p.val < 0.05
    )
  )
```

variable	estimate	std.error	statistic	p.val
Frost	−0.0052	0.0025	−2.0849	0.0428
Illiteracy	0.4082	0.2834	1.4402	0.1567
Income	−0.0001	0.0002	−0.4611	0.6470
log_Area	0.0326	0.1049	0.3109	0.7573

Frost is the variable with the lowest p-value, and the only p-value that is less than 0.05 in this step. We will now include frost in our models.

```
# fit models with 5 variables, keeping murder, HS grad, log_Population, Frost
# in the model each time
variables <- c("Income", "Illiteracy", "log_Area")

df_results5 <- data.frame(variable = NA,
                          estimate = NA,
                          std.error = NA,
                          statistic = NA,
                          p.val = NA)

# for loop to run univariate LS for each variable and save results
for(var in variables){
  fit <- lm(dat.state$`Life Exp` ~ dat.state$Murder + dat.state$`HS Grad` +
              dat.state$log_Population + dat.state$Frost + dat.state[[var]])
  fit_sum <- summary(fit)

  temp_results <- c(var, fit_sum$coefficients[6,])

  df_results5 <- rbind(df_results5, temp_results)
}

df_results5 <- df_results5[-1,] #remove initial rows of NAs

# print table of results
df_results5 |>
  mutate(across(-variable, ~as.numeric(.))) |>
  arrange(p.val) |>
  gt() |>
  fmt_number(
```

```r
    columns = 2:5,
    decimals = 4
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = p.val,
      rows = p.val < 0.05
    )
  )
```

variable	estimate	std.error	statistic	p.val
log_Area	0.0661	0.1022	0.6466	0.5213
Illiteracy	0.1085	0.3443	0.3151	0.7542
Income	0.0000	0.0002	−0.1432	0.8868

At this point, no other variables have a p-value below our 0.05 threshold, so the forward selection process terminates. The final model chose via forward selection is:

$$\widehat{\text{Life Exp}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{Murder} + \hat{\beta}_2 \cdot \text{HS Grad} + \hat{\beta}_3 \cdot \log\left(\text{Population}\right) + \hat{\beta}_4 \cdot \text{Frost}$$

Next: Backward Selection

Start with all variables in the model and remove the variable with the largest p-value until all p-values are significant.

```r
# fit full model
lm(`Life Exp` ~ log_Population + Income + Illiteracy + Murder + `HS Grad` +
     Frost + log_Area,
   data = dat.state) |>
  summary()
```

```

Call:
lm(formula = `Life Exp` ~ log_Population + Income + Illiteracy + 
    Murder + `HS Grad` + Frost + log_Area, data = dat.state)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-1.43084 -0.45559  0.02759  0.49618  1.70215 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     6.795e+01  2.092e+00  32.478  < 2e-16 ***
log_Population  2.527e-01  1.351e-01   1.870   0.0685 .  
Income          1.396e-05  2.444e-04   0.057   0.9547    
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  2.483e-02   2.126   0.0394 *  
Frost          -4.869e-03  3.215e-03  -1.515   0.1373    
log_Area        6.862e-02  1.098e-01   0.625   0.5354    
---
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: 1.433e-10
```

Income has the largest p-value (and is greater than 0.05), so we will remove it from the model.

```
# fit model without income
lm(`Life Exp` ~ log_Population + Illiteracy + Murder + `HS Grad` +
     Frost + log_Area,
   data = dat.state) |>
  summary()
```

```
Call:
lm(formula = `Life Exp` ~ log_Population + Illiteracy + Murder + 
    `HS Grad` + Frost + log_Area, data = dat.state)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.44005 -0.45856  0.02945  0.49580  1.70521 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
```

```
(Intercept)    67.960656   2.063512  32.934  < 2e-16 ***
log_Population  0.255371   0.125215   2.039  0.04758 *
Illiteracy      0.112615   0.346646   0.325  0.74685
Murder         -0.308609   0.045518  -6.780 2.68e-08 ***
`HS Grad`       0.053650   0.019378   2.769  0.00828 **
Frost          -0.004837   0.003127  -1.547  0.12926
log_Area        0.066687   0.103250   0.646  0.52179
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7258 on 43 degrees of freedom
Multiple R-squared:  0.7435,    Adjusted R-squared:  0.7077
F-statistic: 20.77 on 6 and 43 DF,  p-value: 2.978e-11
```

Illiteracy has the largest p-value in this model (and is greater than 0.05), so we will remove it from the model.

```r
# fit model without income, illiteracy
lm(`Life Exp` ~ log_Population + Murder + `HS Grad` + Frost + log_Area,
   data = dat.state) |>
  summary()
```

```

Call:
lm(formula = `Life Exp` ~ log_Population + Murder + `HS Grad` +
    Frost + log_Area, data = dat.state)

Residuals:
     Min       1Q   Median       3Q      Max
-1.43763 -0.46147  0.00721  0.48073  1.74473

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.414712   1.502603  45.531  < 2e-16 ***
log_Population  0.239312   0.113870   2.102   0.0413 *
Murder         -0.301875   0.040110  -7.526 1.95e-09 ***
`HS Grad`       0.050302   0.016243   3.097   0.0034 **
Frost          -0.005424   0.002528  -2.146   0.0374 *
log_Area        0.066067   0.102178   0.647   0.5213
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.7184 on 44 degrees of freedom
Multiple R-squared:  0.7429,    Adjusted R-squared:  0.7136
F-statistic: 25.42 on 5 and 44 DF,  p-value: 5.833e-12
```

Area (the log version) has the largest p-value in this model (and is greater than 0.05), so we will remove it from the model.

```
# fit model without income, illiteracy, log_Area
lm(`Life Exp` ~ log_Population + Murder + `HS Grad` + Frost,
   data = dat.state) |>
  summary()
```

```
Call:
lm(formula = `Life Exp` ~ log_Population + Murder + `HS Grad` +
    Frost, data = dat.state)

Residuals:
     Min       1Q   Median       3Q      Max
-1.41760 -0.43880  0.02539  0.52066  1.63048

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.720810   1.416828  48.503  < 2e-16 ***
log_Population  0.246836   0.112539   2.193 0.033491 *
Murder         -0.290016   0.035440  -8.183 1.87e-10 ***
`HS Grad`       0.054550   0.014758   3.696 0.000591 ***
Frost          -0.005174   0.002482  -2.085 0.042779 *
---
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: 1.17e-12
```

All variables are significant in this model, so the backwards elimination process terminates. The final model chose via backward selection is:

$$\widehat{\text{Life Exp}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{Murder} + \hat{\beta}_2 \cdot \text{HS Grad} + \hat{\beta}_3 \cdot \log\left(\text{Population}\right) + \hat{\beta}_4 \cdot \text{Frost}$$

Next: Stepwise Selection

We want to see which variables are selected in a stepwise selection procedure.

```
step(
 lm(`Life Exp` ~ log_Population + Income + Illiteracy + Murder + `HS Grad` +
     Frost + log_Area,
   data = dat.state),
 direction = "both"
)
```

```
Start:  AIC=-23.6
`Life Exp` ~ log_Population + Income + Illiteracy + Murder +
    `HS Grad` + Frost + log_Area

                 Df Sum of Sq    RSS      AIC
- Income          1    0.0018 22.650 -25.5934
- Illiteracy      1    0.0556 22.704 -25.4746
- log_Area        1    0.2106 22.859 -25.1344
<none>                        22.648 -23.5973
- Frost           1    1.2374 23.886 -22.9374
- log_Population  1    1.8854 24.533 -21.5992
- `HS Grad`       1    2.4375 25.086 -20.4864
- Murder          1   23.2760 45.924   9.7483

Step:  AIC=-25.59
`Life Exp` ~ log_Population + Illiteracy + Murder + `HS Grad` +
    Frost + log_Area

                 Df Sum of Sq    RSS      AIC
- Illiteracy      1    0.0556 22.705 -27.4708
- log_Area        1    0.2197 22.870 -27.1107
<none>                        22.650 -25.5934
- Frost           1    1.2602 23.910 -24.8862
+ Income          1    0.0018 22.648 -23.5973
- log_Population  1    2.1909 24.841 -22.9768
- `HS Grad`       1    4.0374 26.687 -19.3918
- Murder          1   24.2130 46.863   8.7601

Step:  AIC=-27.47
`Life Exp` ~ log_Population + Murder + `HS Grad` + Frost + log_Area
```

```
                 Df Sum of Sq    RSS     AIC
- log_Area        1    0.2157 22.921 -28.998
<none>                        22.705 -27.471
+ Illiteracy      1    0.0556 22.650 -25.593
+ Income          1    0.0017 22.704 -25.475
- log_Population  1    2.2792 24.985 -24.688
- Frost           1    2.3760 25.082 -24.495
- `HS Grad`       1    4.9491 27.655 -19.612
- Murder          1   29.2296 51.935  11.899

Step:  AIC=-29
`Life Exp` ~ log_Population + Murder + `HS Grad` + Frost

                 Df Sum of Sq    RSS     AIC
<none>                        22.921 -28.998
+ log_Area        1     0.216 22.705 -27.471
+ Illiteracy      1     0.052 22.870 -27.111
+ Income          1     0.011 22.911 -27.021
- Frost           1     2.214 25.135 -26.387
- log_Population  1     2.450 25.372 -25.920
- `HS Grad`       1     6.959 29.881 -17.741
- Murder          1    34.109 57.031  14.578



Call:
lm(formula = `Life Exp` ~ log_Population + Murder + `HS Grad` +
    Frost, data = dat.state)

Coefficients:
   (Intercept)  log_Population          Murder       `HS Grad`           Frost
     68.720810        0.246836       -0.290016        0.054550       -0.005174
```

Thus, the final model selected via stepwise selection is:

$$\widehat{\text{Life Exp}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{Murder} + \hat{\beta}_2 \cdot \text{HS Grad} + \hat{\beta}_3 \cdot \log\left(\text{Population}\right) + \hat{\beta}_4 \cdot \text{Frost}$$

FINAL THOUGHTS:

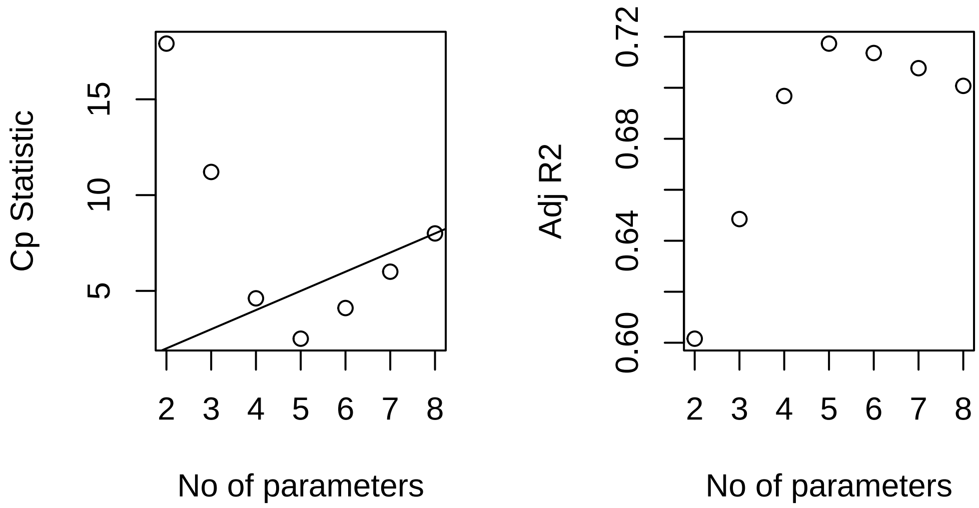
All three of these methods agree on the same suggested model. This might give good evidence that we should explore this model further.

From the outputs above, we see that frost had a p-value close to 0.05. This might mean that the relationship between life expectancy and frost is weak.

To determine if there is an association between illiteracy and HS graduation rate, we can refer to our matrix of plots in part a. The correlation between these two variables is quite strong (approx. 0.7), implying that including that including both of these variables in the model could result in poor coefficient estimation and inflated standard errors (due to multicollinearity). Our model selection techniques were able to "figure this out" and only HS gradation rate was included in the final models.

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

```
b = regsubsets(`Life Exp` ~ log_Population + Income + Illiteracy + Murder +
                  `HS Grad` + Frost + log_Area, data = dat.state)
rs = summary(b)

par(mfrow=c(1,2))
plot(2:8, rs$cp, xlab = "No of parameters", ylab = "Cp Statistic")
abline(0,1)

plot(2:8, rs$adjr2, xlab = "No of parameters", ylab = "Adj R2")
```



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

The plots above show the Cp index and Adjusted R^2 for various numbers of parameters.

When choosing a model based on Cp criterion, we want to choose a model for which $Cp \leq p$, where p is the number of parameters. From the Cp plot above, we should have either 4 parameters (3 predictors), 5 parameters (4 predictors), or 6 parameters (5 predictors). If we consider the principle of parsimony as well, this would suggest the 4 parameters (3 predictors) model. But if we choose the model with 5 parameters (4 predictors), which has the lowest value for Cp, we would include the following variables in the model:

```r
cp_model <- leaps(x = state.x77[,c(1:3, 5:8)], y = state.x77[,4], nbest = 1, method = "Cp"
colnames(state.x77[,c(1:3, 5:8)])[which(cp_model$which[4,])]
```

```
[1] "Population" "Murder"     "HS Grad"    "Frost"
```

When choosing a model based on Adjusted R^2, we want to choose a model that maximizes Adj R^2. According to our model above, this criterion would suggest the 5 parameter (4 predictor) model.

```r
adjr2_model <- leaps(x = state.x77[,c(1:3, 5:8)], y = state.x77[,4], nbest = 1, method = "
colnames(state.x77[,c(1:3, 5:8)])[which(adjr2_model$which[4,])]
```

```
[1] "Population" "Murder"     "HS Grad"    "Frost"
```

Both of these criterion-based techniques suggest the same model.

We can also select a model based on the one that minimizes the AIC:

```r
b = regsubsets(`Life Exp` ~ log_Population + Income + Illiteracy + Murder +
                 `HS Grad` + Frost + log_Area, data = dat.state)
rs = summary(b)

vec_aic <- vector()

for(i in 1:7){
  temp_dat.state <-
    dat.state |>
    select(-`Life Exp`) |>
    select(which(rs$which[i,-1]))

  temp_dat.state <-
    temp_dat.state |>
    bind_cols(dat.state |> select(`Life Exp`))

  fit <- lm(`Life Exp` ~ . , data = temp_dat.state)
  vec_aic <- c(vec_aic, AIC(fit))
}

df_results_aic <-
  data.frame(
    num_parameters = 2:8,
    AIC = vec_aic
  )

df_results_aic |>
  gt() |>
  fmt_number(
    columns = AIC,
    decimals = 2
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "lightcyan"),
      cell_text(weight = "bold")
    ),
    locations = cells_body(
      columns = AIC,
      rows = which(AIC == min(AIC))
    )
  )
```

num_parameters	AIC
2	129.28
3	123.97
4	120.02
5	115.73
6	117.73
7	119.72
8	121.71

These results indicate that the model with 5 parameters (4 predictors) is the “best” model. Again this would be the model that contains log population, Murder, HS Grad, Frost.

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

```
# supply sequence of lambda values for the lasso cross validation for lambda
lambda_seq <- 10^seq(-3, 0, by = .1)
set.seed(2022)

# save matrix of predictors to pass to the lasso function
predictors_dat.state <-
  dat.state |>
  select(log_Population, Income, Illiteracy, Murder,
         `HS Grad`, Frost, log_Area) |>
  as.matrix()

response_dat.state <-
  dat.state |>
  select(`Life Exp`) |>
  as.matrix()

cv_lasso_fit <- glmnet::cv.glmnet(x = predictors_dat.state,
                                  y = response_dat.state,
                                  lambda = lambda_seq,
                                  nfolds = 5)
cv_lasso_fit
```

```
Call:  glmnet::cv.glmnet(x = predictors_dat.state, y = response_dat.state,      lambda = lamb

Measure: Mean-Squared Error

     Lambda Index Measure     SE Nonzero
min 0.03162    16  0.6467 0.1813       5
1se 0.31623     6  0.7919 0.1511       2
```

Min. lambda from CV is 0.0316. We can now run a LASSO regression using this value for lambda.

```
lasso_fit <- glmnet::glmnet(x = predictors_dat.state,
                            y = response_dat.state,
                            lambda = cv_lasso_fit$lambda.min)
coef(lasso_fit)
```

```
8 x 1 sparse Matrix of class "dgCMatrix"
                         s0
(Intercept)    68.904248035
log_Population  0.208096315
Income          .
Illiteracy      .
Murder         -0.277654038
HS Grad         0.048555590
Frost          -0.004090183
log_Area        0.022049521
```

The model selected via the minimum lambda and the LASSO technique includes the variables: log Population, Murder, HS graduation rate, Frost, and 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.**
- **Test the model predictive ability using a 10-fold cross-validation (10 repeats).**

Since the stepwise selection techniques and the criterion techniques all chose the same model with 4 predictors, we recommend this as our final model. (The LASSO gave a very similar suggested model, with the addition of log area.)

Our final model:

```r
final_model <- lm(`Life Exp` ~ Murder + `HS Grad` +
     log_Population + Frost, data = dat.state)

summary(final_model)
```

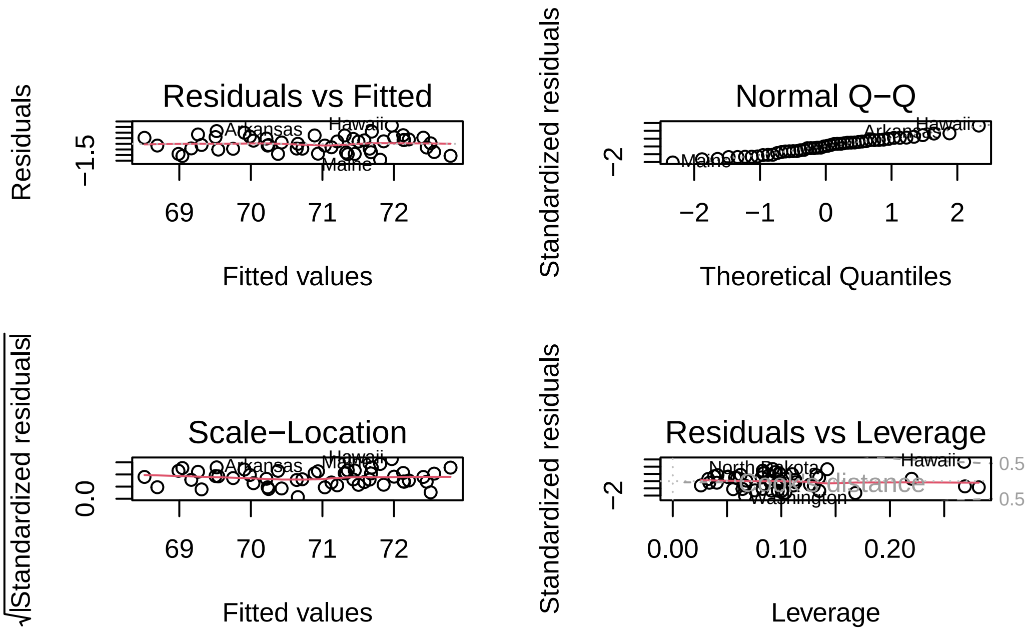
```
Call:
lm(formula = `Life Exp` ~ Murder + `HS Grad` + log_Population +
    Frost, data = dat.state)

Residuals:
     Min       1Q   Median       3Q      Max
-1.41760 -0.43880  0.02539  0.52066  1.63048

Coefficients:
```

```
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    68.720810   1.416828  48.503  < 2e-16 ***
Murder         -0.290016   0.035440  -8.183 1.87e-10 ***
`HS Grad`       0.054550   0.014758   3.696 0.000591 ***
log_Population  0.246836   0.112539   2.193 0.033491 *  
Frost          -0.005174   0.002482  -2.085 0.042779 *  
---
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: 1.17e-12
```

$$\widehat{\text{Life Exp}} = 68.7 - 0.29 \cdot \text{Murder} + 0.05 \cdot \text{HS Grad} + 0.25 \cdot \log\left(\text{Population}\right) - 0.005 \cdot \text{Frost}$$

Checking the model diagnostic plots:

```
par(mfrow = c(2,2))
plot(final_model)
```



Overall, these plots look like this model is fitting the data well. Hawaii seems like it could be an influential point, so it may be worth investigating further.

10-fold CV

```
set.seed(2022)
# use 10-fold validation and create the training sets
train = trainControl(method = "cv", number = 10)

# fit the 4-variables model that we selected as our final model
model_caret = train(`Life Exp` ~ Murder + `HS Grad` + log_Population + Frost,
                   data = dat.state,
                   trControl = train,
                   method = 'lm',
                   na.action = na.pass)

model_caret
```

```
Linear Regression

50 samples
 4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 46, 46, 44, 45, 46, 44, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.7586994  0.6691637  0.6417845

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
model_caret$resample
```

```
        RMSE  Rsquared       MAE Resample
1  0.7621059 0.7544187 0.4880670   Fold01
2  0.6811317 0.9758260 0.5559302   Fold02
3  0.6906081 0.2186818 0.6212361   Fold03
4  0.7283781 0.8715378 0.5776364   Fold04
5  0.8152456 0.3656340 0.8015345   Fold05
6  1.0641139 0.4170131 0.8115803   Fold06
7  0.6877440 0.7541411 0.5763108   Fold07
```

```
8  1.0169469 0.7007803 0.9807082   Fold08
9  0.5071120 0.9009624 0.4523930   Fold09
10 0.6336083 0.7326414 0.5524481   Fold10
```

From the output above, the overall RMSE (root mean squared error) is 0.76, which would mean our MSE is 0.58. Our MAE (mean absolute error) is 0.64.

These measures show that this model is doing a good job at predicting responses for "new" data points. Additionally, the variance for these measures is relatively small, showing that these estimates are probably pretty close to the true predictive ability.

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

We employed automatic search procedures, criterion based approaches, and the LASSO technique to select a final model:

$$\widehat{\text{Life Exp}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{Murder} + \hat{\beta}_2 \cdot \text{HS Grad} + \hat{\beta}_3 \cdot \log\left(\text{Population}\right) + \hat{\beta}_4 \cdot \text{Frost}$$

$$\widehat{\text{Life Exp}} = 68.7 - 0.29 \cdot \text{Murder} + 0.05 \cdot \text{HS Grad} + 0.25 \cdot \log\left(\text{Population}\right) - 0.005 \cdot \text{Frost}$$

From this model, we can see that as the murder rate and average number of freezing days (frost) increase, the predicted life expectancy decreases, while increases in high school graduations and (log) population were associated with an increase in expected life expectancy.

Overall, from our 10-fold cross validation, we see that our model has pretty good predictive ability for new points. However, this model was only built on data from the US, so it should not be used to predict life expectancy in other locations. Additionally, we noticed that Hawaii was a potential influential point – by exploring Hawaii's role in our model, we may have slightly different conclusions.