P8130 Homework 5

Xiaoni Xu

2024-12-15

```r
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.4.2
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(knitr)
library(ggplot2)
library(tidyr)
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.2
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.4.2

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8
```

```r
library(caret)
```

```
## Warning: package ’caret’ was built under R version 4.4.2

## Loading required package: lattice

## 
## Attaching package: ’lattice’

## The following object is masked from ’package:faraway’:
## 
##     melanoma
```

a)

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

```r
data(state)

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

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 ...
```

```r
# Generate summary statistics for all variables
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.: 81163  
##  Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432  
```

```r
# Clean the variables' names
state_data = state_data |> janitor::clean_names()


# Calculate additional descriptive statistics (mean, median, standard deviation, etc.)
descriptive_stats <- data.frame(
  Mean = apply(state_data, 2, mean),    # Calculate the mean for each variable
  Median = apply(state_data, 2, median), # Calculate the median for each variable
  SD = apply(state_data, 2, sd),        # Calculate the standard deviation for each variable
  Min = apply(state_data, 2, min),      # Calculate the minimum value for each variable
  Max = apply(state_data, 2, max)       # Calculate the maximum value for each variable
)

# Show the descriptive statistics table
descriptive_stats |> knitr::kable(caption = "Descriptive Statistics for state.x77 Dataset")
```

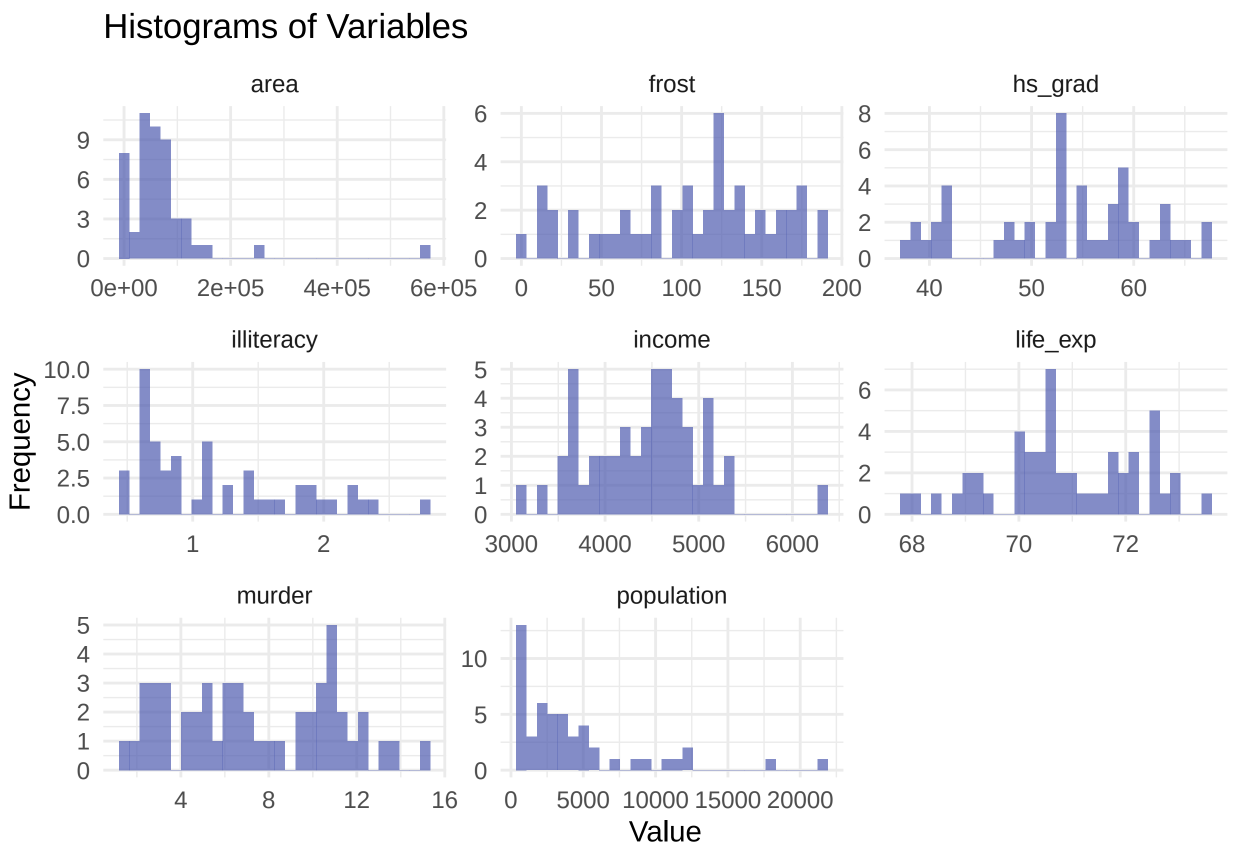
Table 1: Descriptive Statistics for state.x77 Dataset

	Mean	Median	SD	Min	Max
population	4246.4200	2838.500	4.464491e+03	365.00	21198.0
income	4435.8000	4519.000	6.144699e+02	3098.00	6315.0
illiteracy	1.1700	0.950	6.095331e-01	0.50	2.8
life_exp	70.8786	70.675	1.342394e+00	67.96	73.6
murder	7.3780	6.850	3.691540e+00	1.40	15.1
hs_grad	53.1080	53.250	8.076998e+00	37.80	67.3
frost	104.4600	114.500	5.198085e+01	0.00	188.0
area	70735.8800	54277.000	8.532730e+04	1049.00	566432.0

b)

Examine exploratory plots (histograms) to get a sense of the data and possible variable transformations. If you find a transformation to be necessary or recommended, perform the transformation and use it through the rest of the problem.

```r
# Create histograms to explore variable distributions
state_data %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "#4d5aaf", alpha = 0.7) +
  facet_wrap(~ variable, scales = "free") +
  theme_minimal() +
  labs(title = "Histograms of Variables", x = "Value", y = "Frequency")
```

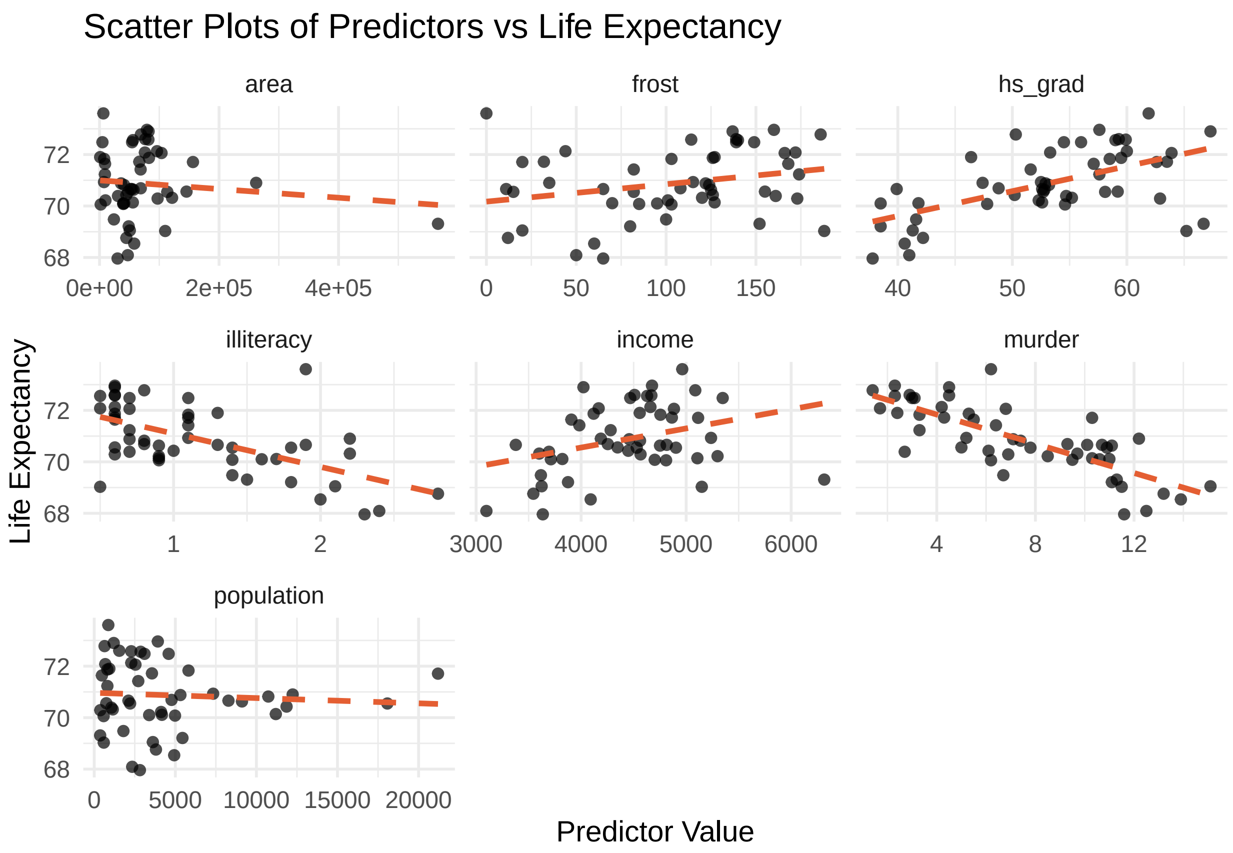


```
# Check column names
names(state_data)
```

```
## [1] "population" "income"     "illiteracy" "life_exp"   "murder"
## [6] "hs_grad"    "frost"      "area"
```

```
# Corrected code
state_data %>%
  pivot_longer(cols = -life_exp, names_to = "predictor", values_to = "value") %>%
  ggplot(aes(x = value, y = life_exp)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "#e45e32", linetype = "dashed") +
  facet_wrap(~ predictor, scales = "free_x") +
  theme_minimal() +
  labs(title = "Scatter Plots of Predictors vs Life Expectancy",
       x = "Predictor Value",
       y = "Life Expectancy")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

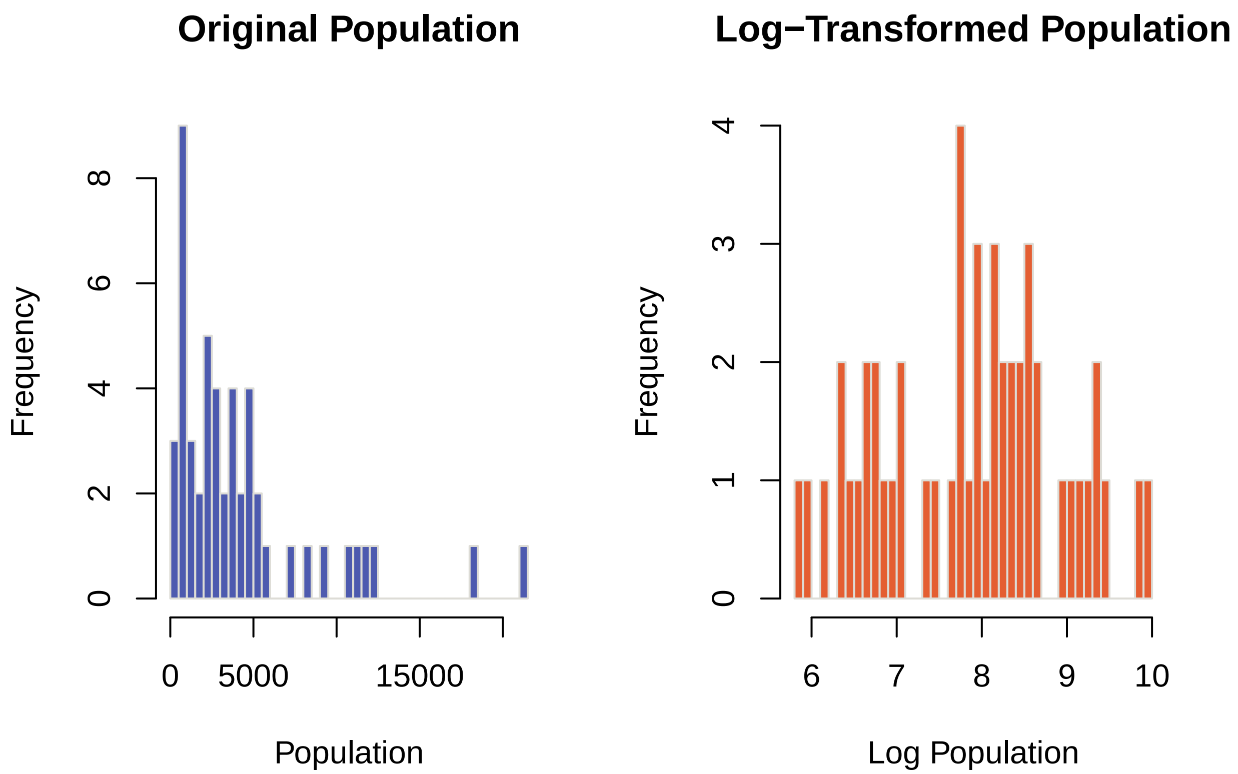


Scatter Plots of Predictors vs Life Expectancy

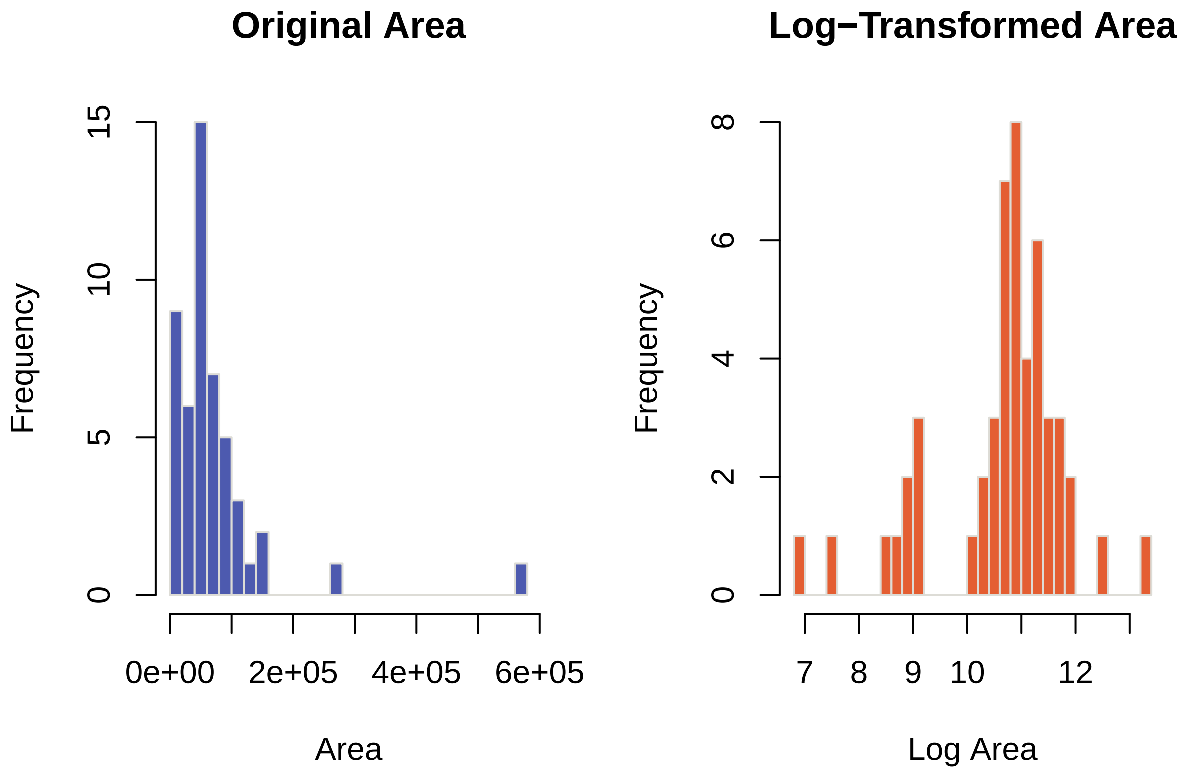
Perform log transformation for `area`, `population`, and `illiteracy`.

```
# Define a function for side-by-side plotting
plot_side_by_side <- function(original, transformed, variable_name) {
  par(mfrow = c(1, 2)) # Set up the plotting area for side-by-side plots

  # Plot original data
  hist(original, main = paste("Original", variable_name), xlab = variable_name,
       col = "#4d5aaf", border = "#dddcd6", breaks = 30)

  # Plot transformed data
  hist(transformed, main = paste("Log-Transformed", variable_name), xlab = paste("Log", variable_name),
       col = "#e45e32", border = "#dddcd6", breaks = 30)

  par(mfrow = c(1, 1)) # Reset the plotting area
}

# Perform log transformations
state_data <- state_data %>%
  mutate(
    log_population = log(population),
    log_area = log(area),
    log_illiteracy = log(illiteracy)
  )

# Plot histograms for population
```

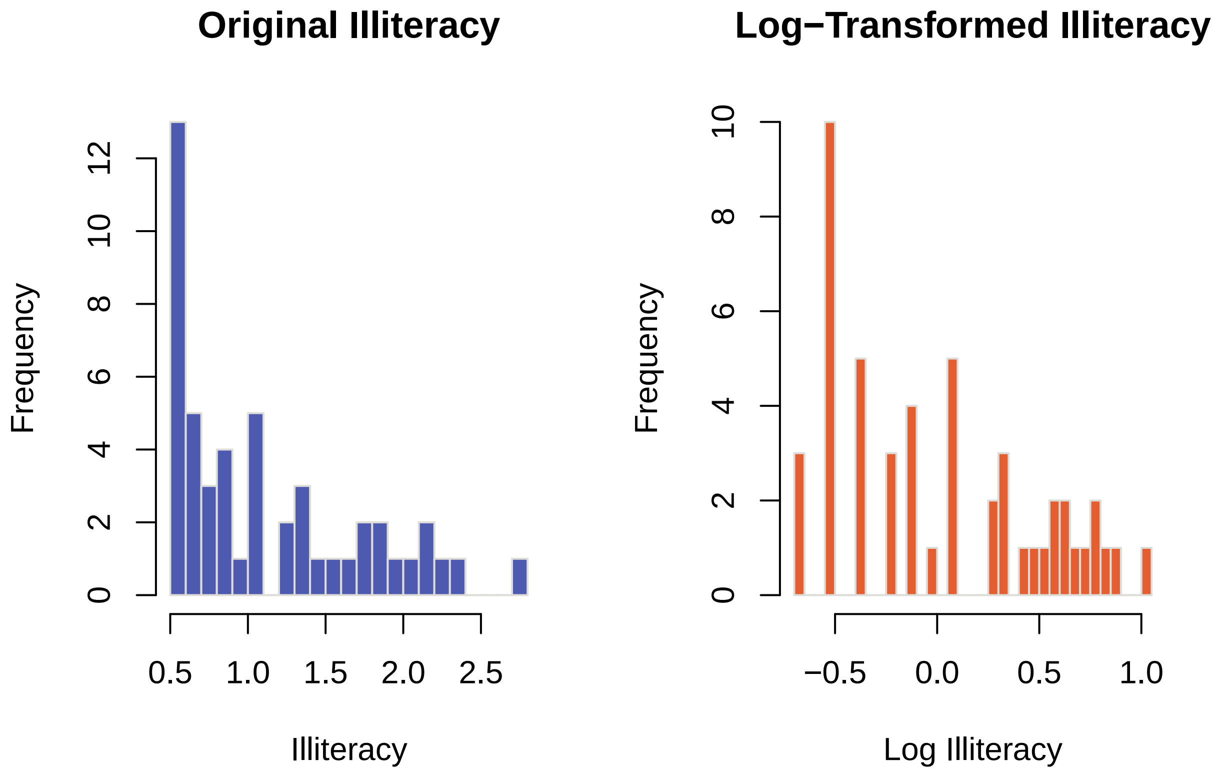
```r
plot_side_by_side(state_data$population, state_data$log_population, "Population")
```



```r
# Plot histograms for area
plot_side_by_side(state_data$area, state_data$log_area, "Area")
```



```r
# Plot histograms for illiteracy
plot_side_by_side(state_data$illiteracy, state_data$log_illiteracy, "Illiteracy")
```



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?

- 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).

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

Perform Best Subset Selection

```
# Perform best subset selection
best_subset <- regsubsets(life_exp ~ ., data = state_data, nbest = 1, method = "exhaustive")

# Summarize the results
best_subset_summary <- summary(best_subset)

# Display the subset selection results
best_subset_summary$outmat
```

```
##          population income illiteracy murder hs_grad frost area log_population
## 1  ( 1 ) " "        " "    " "        "*"    " "     " "   " "  " "
## 2  ( 1 ) " "        " "    " "        "*"    "*"     " "   " "  " "
## 3  ( 1 ) " "        " "    " "        "*"    "*"     " "   " "  "*"
## 4  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 5  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 6  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 7  ( 1 ) " "        " "    " "        "*"    "*"     "*"   "*"  "*"
## 8  ( 1 ) " "        " "    "*"        "*"    "*"     "*"   "*"  "*"
##          log_area log_illiteracy
## 1  ( 1 ) " "      " "
## 2  ( 1 ) " "      " "
## 3  ( 1 ) " "      " "
## 4  ( 1 ) " "      " "
## 5  ( 1 ) "*"      " "
## 6  ( 1 ) "*"      "*"
## 7  ( 1 ) "*"      "*"
## 8  ( 1 ) "*"      "*"
```

```r
# Extract model performance metrics
model_metrics <- data.frame(
  Num_Variables = 1:length(best_subset_summary$adjr2),
  Adj_R2 = best_subset_summary$adjr2,
  Cp = best_subset_summary$cp,
  BIC = best_subset_summary$bic
)

# Display the model metrics
model_metrics |> knitr::kable(caption = "Model Performance Metrics")
```

Table 2: Model Performance Metrics

Num_Variables	Adj_R2	Cp	BIC
1	0.6015893	14.4508946	-39.22051
2	0.6484991	8.2221519	-42.62472
3	0.6967729	2.0916426	-47.17452
4	0.7173392	0.2076419	-47.87315
5	0.7136360	1.8291971	-44.43397
6	0.7083894	3.6371377	-40.76364
7	0.7036378	5.3461738	-37.22000
8	0.6981631	7.1189035	-33.59762

```r
best_subset_summary$outmat
```

```
##          population income illiteracy murder hs_grad frost area log_population
## 1  ( 1 ) " "        " "    " "        "*"    " "     " "   " "  " "
## 2  ( 1 ) " "        " "    " "        "*"    "*"     " "   " "  " "
## 3  ( 1 ) " "        " "    " "        "*"    "*"     " "   " "  "*"
## 4  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 5  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 6  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "  "*"
## 7  ( 1 ) " "        " "    " "        "*"    "*"     "*"   "*"  "*"
## 8  ( 1 ) " "        " "    "*"        "*"    "*"     "*"   "*"  "*"
##          log_area log_illiteracy
## 1  ( 1 ) " "      " "
## 2  ( 1 ) " "      " "
## 3  ( 1 ) " "      " "
## 4  ( 1 ) " "      " "
## 5  ( 1 ) "*"      " "
## 6  ( 1 ) "*"      "*"
## 7  ( 1 ) "*"      "*"
## 8  ( 1 ) "*"      "*"
```

1. Do the procedures generate the same model?

From the `best_subset_summary$outmat`, the models selected vary depending on the number of predictors included. The optimal model depends on the evaluation criterion:

- Adjusted R-squared:
    - Best model: 4 predictors (`murder`, `hs_grad`, `frost`, `log_population`).
    - Achieves the highest Adj R^2 of 0.717.
- Cp:
    - Best model: 4 predictors, as it minimizes Mallows' C_p, which is close to the number of predictors $(p + 1)$.
- BIC:
    - Best model: 4 predictors, as it achieves the lowest BIC of -47.873.

The best subset models are consistent across all three criteria, selecting a 4-predictor model that includes `murder`, `hs_grad`, `frost`, and `log_population`.

2. Are any variables a close call?

From the `best_subset_summary$outmat`, examine the inclusion patterns:

- Close Call: `log_area`:
    - Appears in 5-predictor and larger models but is excluded in the top 4-predictor model.
    - Discard the variable. Its contribution is minor (marginal increase in Adj R^2) and leads to overfitting.
- Close Call: `log_illiteracy`:
    - Appears in 6-predictor and larger models but not the top 4-predictor model.
    - Discard the variable. The improvement in model fit is negligible compared to the added complexity.

3. Is there any association between Illiteracy and HS Graduation Rate?

From the correlation and scatterplot analysis:

```r
# Correlation and test
correlation <- cor(state_data$log_illiteracy, state_data$hs_grad)
correlation_test <- cor.test(state_data$log_illiteracy, state_data$hs_grad)

# Results
cat("Correlation:", correlation, "\n")
```

```
## Correlation: -0.6688091
```

```r
print(correlation_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  state_data$log_illiteracy and state_data$hs_grad
## t = -6.2328, df = 48, p-value = 1.105e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7985058 -0.4797775
## sample estimates:
##        cor
## -0.6688091
```

Correlation Results

- **Correlation coefficient**: -0.6688, indicating a **moderate to strong negative relationship** between `log_illiteracy` and `hs_grad`.
- **Statistical significance**:
  - t-value = -6.2328,
  - p-value = 1.104677e-07.

  The p-value is highly significant ($p < 0.001$).
- **95% Confidence Interval**: [-0.7985, -0.4798].

Interpretation: States with higher `log_illiteracy` tend to have lower `hs_grad` rates. This suggests a **strong inverse relationship**, likely driven by shared socioeconomic or educational factors.

The best subset model does **not** include both `log_illiteracy` and `hs_grad`. Only `hs_grad` is retained in the model.

Justification:

- **Multicollinearity**: The strong correlation ($r = -0.6688$) between `log_illiteracy` and `hs_grad` indicates potential redundancy if both are included in the model.
- **Explanatory Power**: `hs_grad` likely captures sufficient information to explain its relationship with `life_exp`. Including `log_illiteracy` may add unnecessary complexity without significantly improving model performance.

d)

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

```r
best_subset <- regsubsets(life_exp ~ ., data = state_data, nbest = 1, method = "exhaustive")

best_subset_summary <- summary(best_subset)
```

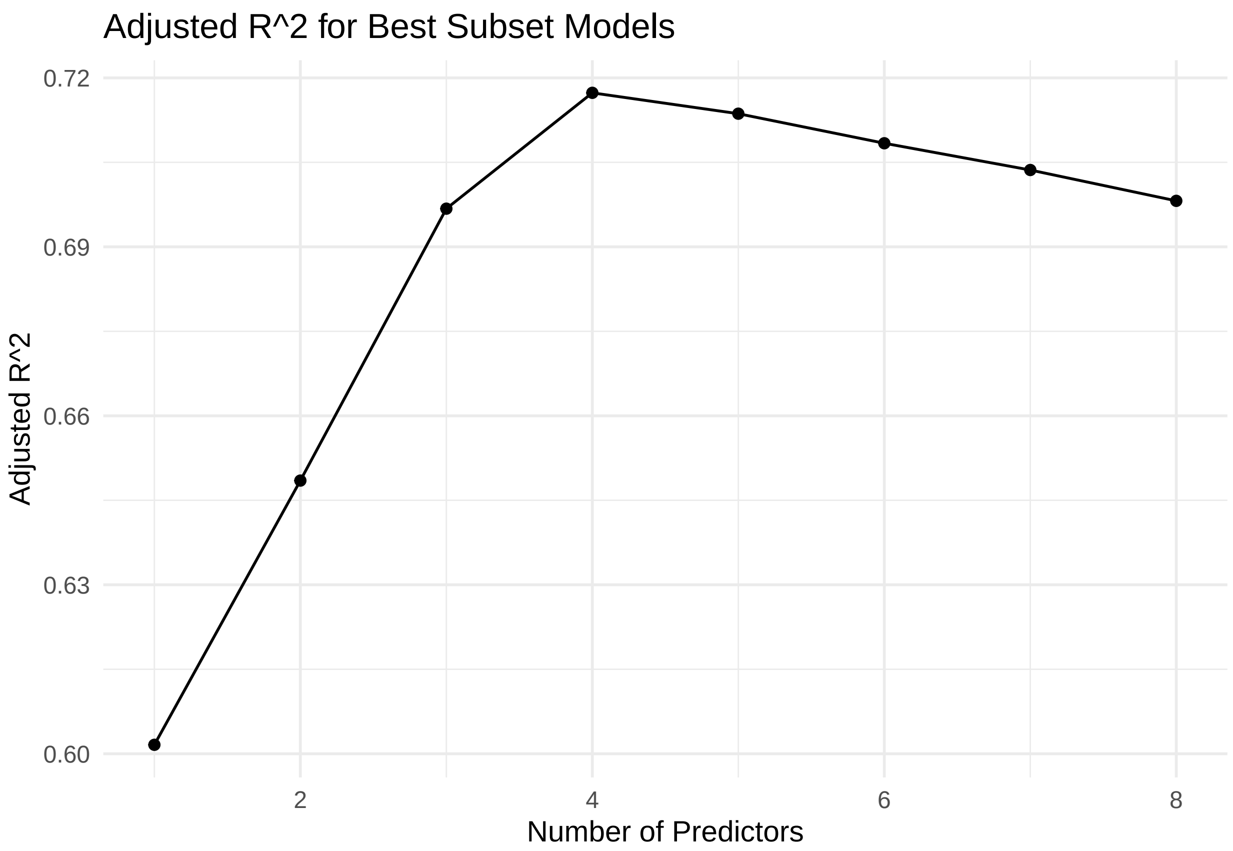
```r
# Extract performance metrics
model_metrics <- data.frame(
  Num_Variables = 1:length(best_subset_summary$adjr2),
  Adj_R2 = best_subset_summary$adjr2,
  Cp = best_subset_summary$cp,
  BIC = best_subset_summary$bic
)

# Display the model metrics
model_metrics |> kable(caption = "Model Performance Metrics")
```

Table 3: Model Performance Metrics

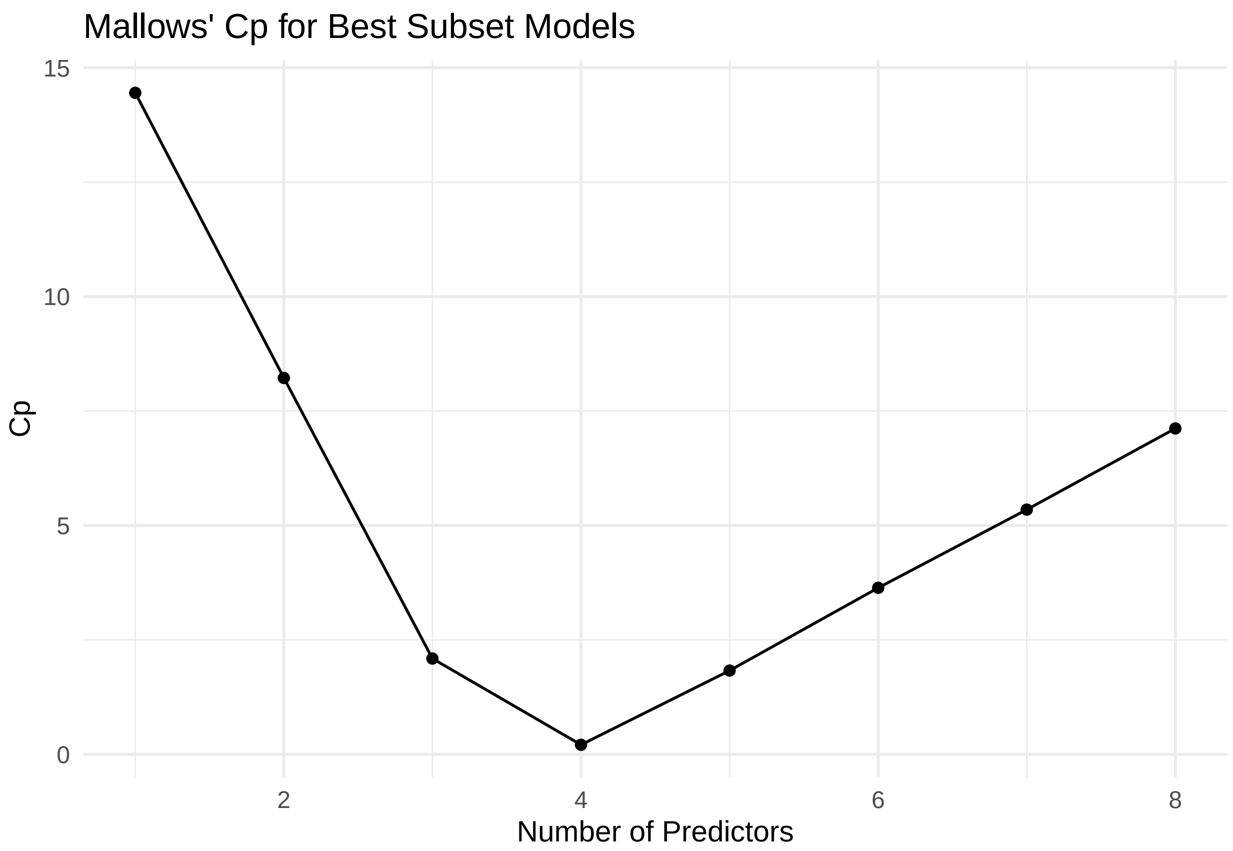
Num_Variables	Adj_R2	Cp	BIC
1	0.6015893	14.4508946	-39.22051
2	0.6484991	8.2221519	-42.62472
3	0.6967729	2.0916426	-47.17452
4	0.7173392	0.2076419	-47.87315
5	0.7136360	1.8291971	-44.43397
6	0.7083894	3.6371377	-40.76364
7	0.7036378	5.3461738	-37.22000
8	0.6981631	7.1189035	-33.59762

The best subset model includes three predictors: `murder`, `hs_grad`, `frost`, and `log_population`.

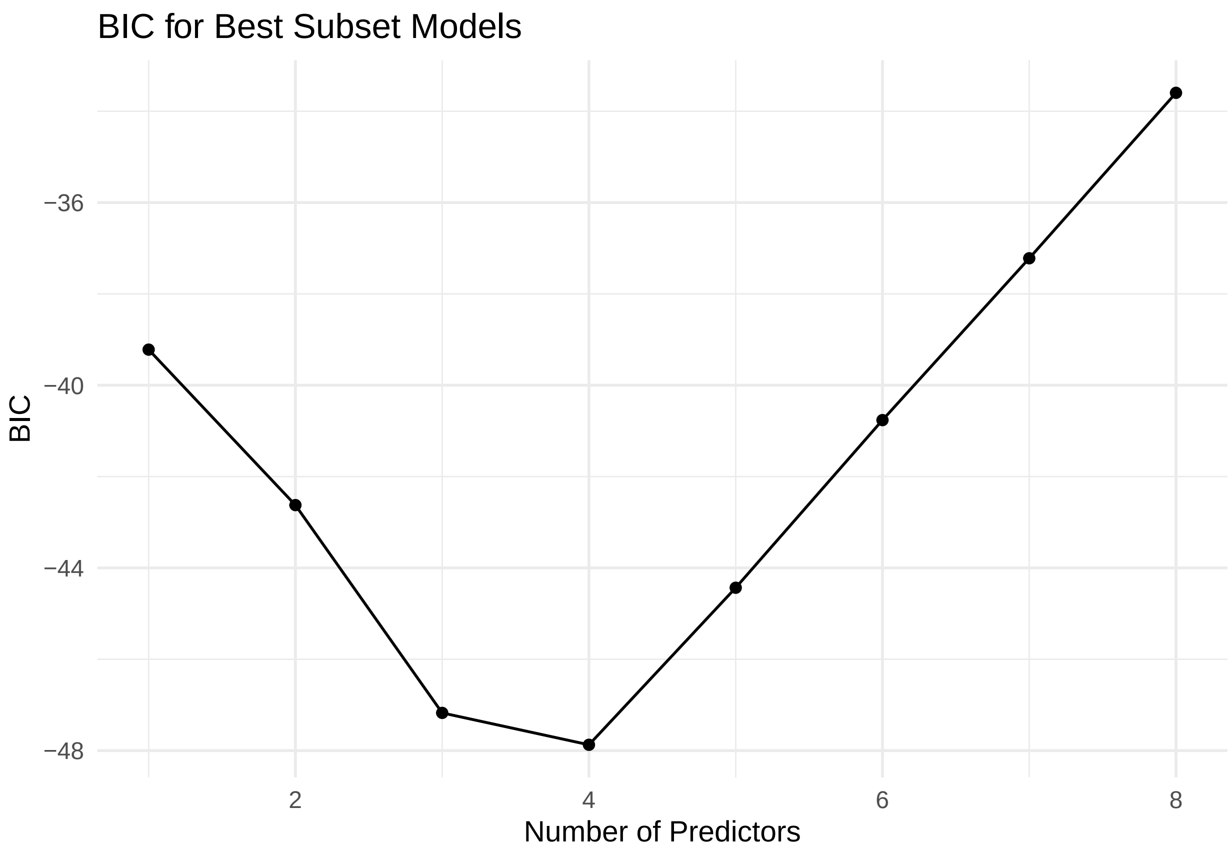
```r
# Adjusted R^2
ggplot(model_metrics, aes(x = Num_Variables, y = Adj_R2)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "Adjusted R^2 for Best Subset Models",
       x = "Number of Predictors",
       y = "Adjusted R^2")
```



```r
# Cp
ggplot(model_metrics, aes(x = Num_Variables, y = Cp)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "Mallows' Cp for Best Subset Models",
       x = "Number of Predictors",
       y = "Cp")
```



```r
# BIC
ggplot(model_metrics, aes(x = Num_Variables, y = BIC)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "BIC for Best Subset Models",
       x = "Number of Predictors",
       y = "BIC")
```



The criterion-based procedures (Adjusted R^2, Mallows' C_p, and BIC) consistently selected the best subset model containing the following predictors:

- `murder`
- `hs_grad`
- `frost`
- `log_population`
- **Adjusted** R^2: The 4-predictor model achieves the highest R^2 of 0.7173.
- **Cp**: The 4-predictor model minimizes C_p at 0.2076, closest to $p + 1$.
- **BIC**: The 4-predictor model achieves the lowest BIC of -47.8732.

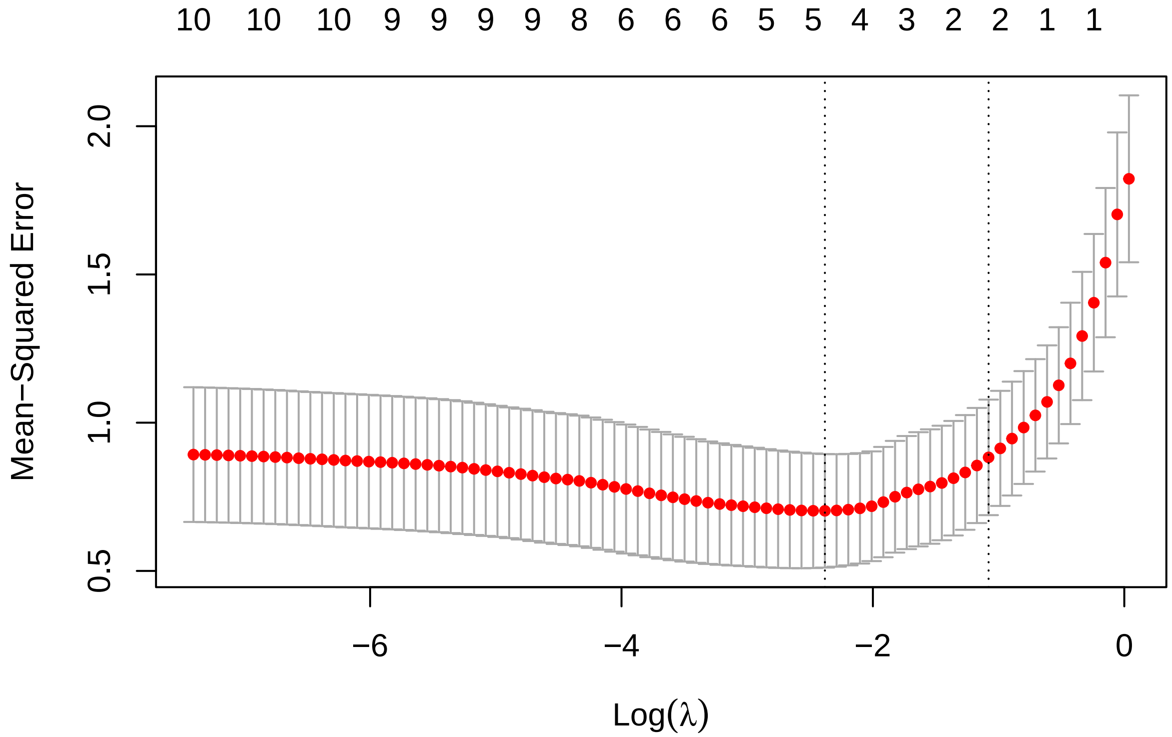
e)

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

LASSO with Cross-Validation

```
# Prepare data
X <- as.matrix(state_data %>% select(-life_exp))  # Predictor variables
Y <- state_data$life_exp  # Response variable
```

```
# Perform LASSO with cross-validation
set.seed(123)  # For reproducibility
lasso_cv <- cv.glmnet(X, Y, alpha = 1, standardize = TRUE)

# Extract the best lambda
best_lambda <- lasso_cv$lambda.min
lambda_1se <- lasso_cv$lambda.1se

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



```
# Display best lambda
cat("Best Lambda (Min):", best_lambda, "\n")
```

```
## Best Lambda (Min): 0.09237471
```

```
cat("Best Lambda (1SE):", lambda_1se, "\n")
```

```
## Best Lambda (1SE): 0.3397893
```

Fit LASSO Model at Best Lambda

```
# Fit the LASSO model using the best lambda
lasso_model <- glmnet(X, Y, alpha = 1, lambda = best_lambda, standardize = TRUE)
```

LASSO Coefficients at the Best Lambda

```
# Extract non-zero coefficients
lasso_coefs <- as.matrix(coef(lasso_model))  # Convert to a standard matrix
selected_vars <- rownames(lasso_coefs)[lasso_coefs[, 1] != 0]  # Identify non-zero coefficients
selected_vars <- selected_vars[-1]  # Remove the intercept (if present)

# Display selected variables
cat("Selected Variables:", paste(selected_vars, collapse = ", "), "\n")
```

```
## Selected Variables: murder, hs_grad, frost, log_population
```

```
# Create a data frame of selected coefficients
lasso_coefs_df <- data.frame(
  Variable = rownames(lasso_coefs),
  Coefficient = as.vector(lasso_coefs)
) %>% filter(Coefficient != 0)  # Filter non-zero coefficients
lasso_coefs_df |> knitr::kable(caption = "LASSO Selected Variables and Coefficients")
```

Table 4: LASSO Selected Variables and Coefficients

Variable	Coefficient
(Intercept)	69.5557584
murder	-0.2424378
hs_grad	0.0411712
frost	-0.0017660
log_population	0.1410969

- **Intercept**:
  - The intercept (β_0) is approximately 69.556. This represents the predicted value of `life_exp` when all predictors are at zero. While this value is not meaningful in isolation, it serves as a baseline for predictions.
- **Murder**:
  - The coefficient for `murder` is -0.242, indicating that for each unit increase in the murder rate (per 100,000 people), the predicted life expectancy decreases by -0.242 years, assuming all other variables are held constant. This highlights a significant negative association between crime rates and life expectancy.
- **HS Graduation Rate (`hs_grad`)**:
  - The coefficient for `hs_grad` is 0.041, suggesting that for each percentage point increase in the high school graduation rate, life expectancy increases by 0.041 years, holding other variables constant. This reflects the positive impact of education on health outcomes.
- **Frost**:
  - The coefficient for `frost` is -0.002, indicating that for each additional day of frost per year, life expectancy decreases by approximately -0.002 years. This suggests a slight negative association between colder climates and life expectancy.

- **Log of Population (`log_population`)**:
  - The coefficient for `log_population` is 0.141, meaning that for each unit increase in the natural log of population, life expectancy increases by 0.141 years, holding other factors constant. This could indicate that larger population sizes (log-transformed) are associated with improved access to resources or infrastructure that positively affect life expectancy.

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.

1. **Subset from Part (c)**:
   - The best subset model selected using exhaustive search includes the predictors:
     - `murder`, `hs_grad`, `frost`, `log_population`.
2. **Subset from Part (d)**:
   - Criterion-based procedures (Adjusted R^2, C_p, BIC) also selected the same predictors:
     - `murder`, `hs_grad`, `frost`.
3. **Subset from Part (e)**:
   - The LASSO method selected the following predictors:
     - `murder`, `hs_grad`, `frost`, `log_population`.

Final Model Recommendation

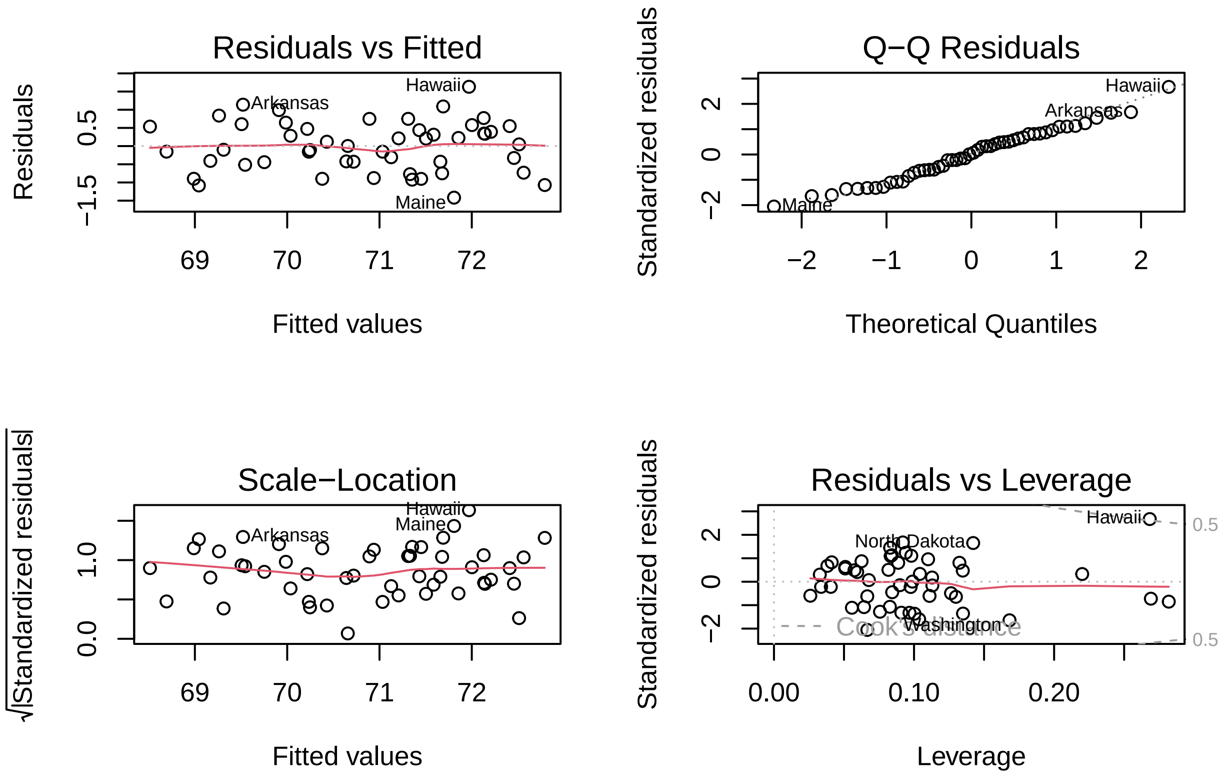
- The subsets from parts (c) and (e) are the same, making it a majority in number of variables selected. The final model will include:
  - `murder`, `hs_grad`, `frost`, `log_population`.

```r
# Fit the final model
final_model <- lm(life_exp ~ murder + hs_grad + frost + log_population, data = state_data)

# Summary of the final model
summary(final_model)
```

```
##
## Call:
## lm(formula = life_exp ~ murder + hs_grad + frost + log_population,
##     data = state_data)
##
## 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 ***
## frost          -0.005174   0.002482  -2.085 0.042779 *
## log_population  0.246836   0.112539   2.193 0.033491 *
## ---
## 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
```

```
# Residual diagnostics
par(mfrow = c(2, 2))
plot(final_model)
```



```
# Check normality of residuals
shapiro_test <- shapiro.test(final_model$residuals)
cat("Shapiro-Wilk test p-value:", shapiro_test$p.value, "\n")
```

```
## Shapiro-Wilk test p-value: 0.6820349
```

The residual vs. fitted plot shows no clear pattern, indicating homoscedasticity (equal variance of residuals).

The normal Q-Q plot shows points roughly along the diagonal, indicating residuals are normally distributed.

Shapiro-Wilk Test:

Given p>0.05, the residuals are normally distributed.

Test Predictive Ability Using 10-Fold Cross-Validation

```r
train_control <- trainControl(method = "cv", number = 10)

# Train the model using cross-validation
cv_model <- train(life_exp ~ murder + hs_grad + frost + log_population,
                  data = state_data,
                  method = "lm",
                  trControl = train_control)

# Display cross-validation results
cv_model_results <- cv_model$results
cv_model_results |> knitr::kable(caption = "10-Fold Cross-Validation Results")
```

Table 5: 10-Fold Cross-Validation Results

intercept	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
TRUE	0.7713868	0.7239287	0.6495825	0.1850696	0.1223869	0.1437972

Cross-Validation Results

Metric	Value	Description
RMSE	0.7714	The **Root Mean Squared Error** reflects the average prediction error of the model. A lower RMSE indicates better predictive accuracy. Here, the model's average error is approximately **0.771 years** in predicting life expectancy.
R-squared	0.724	The **R-squared** value shows that **72.4%** of the variation in life expectancy is explained by the model during cross-validation, indicating strong predictive ability.
MAE	0.650	The **Mean Absolute Error** measures the average absolute difference between predicted and observed values. The model's average prediction error is **0.650 years**, which is slightly lower than the RMSE and reflects consistent predictions.
RMSESD	0.185	The **standard deviation of RMSE** across the 10 folds indicates the variability in prediction errors across different data splits. The variability in RMSE is relatively low, suggesting a stable model.
R-squaredSD	0.122	The **standard deviation of R-squared** shows low variability in the proportion of variance explained across folds, supporting model robustness.
MAESD	0.144	The **standard deviation of MAE** is also small, indicating consistent predictive performance.

The model is fitted using the formula:

$$\text{life\_exp} = \beta_0 + \beta_1 \cdot \text{murder} + \beta_2 \cdot \text{hs\_grad} + \beta_3 \cdot \text{frost} + \beta_4 \cdot \text{log\_population}$$

The regression model for predicting `life_exp` is given by:

$$\text{life}\hat{\_}\text{exp} = 68.721 - 0.290 \cdot \text{murder} + 0.055 \cdot \text{hs\_grad} - 0.005 \cdot \text{frost} + 0.247 \cdot \text{log\_population}$$

g)

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

Our analysis examined the factors influencing life expectancy across U.S. states. We identified four key predictors: the murder rate, high school graduation rate, annual frost days, and population size (log-transformed). States with higher murder rates or more annual frost days tend to have lower life expectancy, while states with higher high school graduation rates and larger populations generally have longer life expectancy. The model we developed explains approximately 74% of the variation in life expectancy, indicating strong predictive power. These findings suggest that education, public safety, and environmental factors play significant roles in shaping life expectancy across regions.