p8130_hw5

Zihan Lin

a

```r
# Load the dataset
data(state)
state_data <- as.data.frame(state.x77)

# Compute basic summary statistics
summary_stats <- summary(state_data)
print("Basic Summary Statistics:")
```

```
## [1] "Basic Summary Statistics:"
```

```r
print(summary_stats)
```

```
##    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
# Compute detailed descriptive statistics using `psych`
detailed_stats <- psych::describe(state_data)
print("Detailed Descriptive Statistics:")
```

```
## [1] "Detailed Descriptive Statistics:"
```

```r
print(detailed_stats)
```

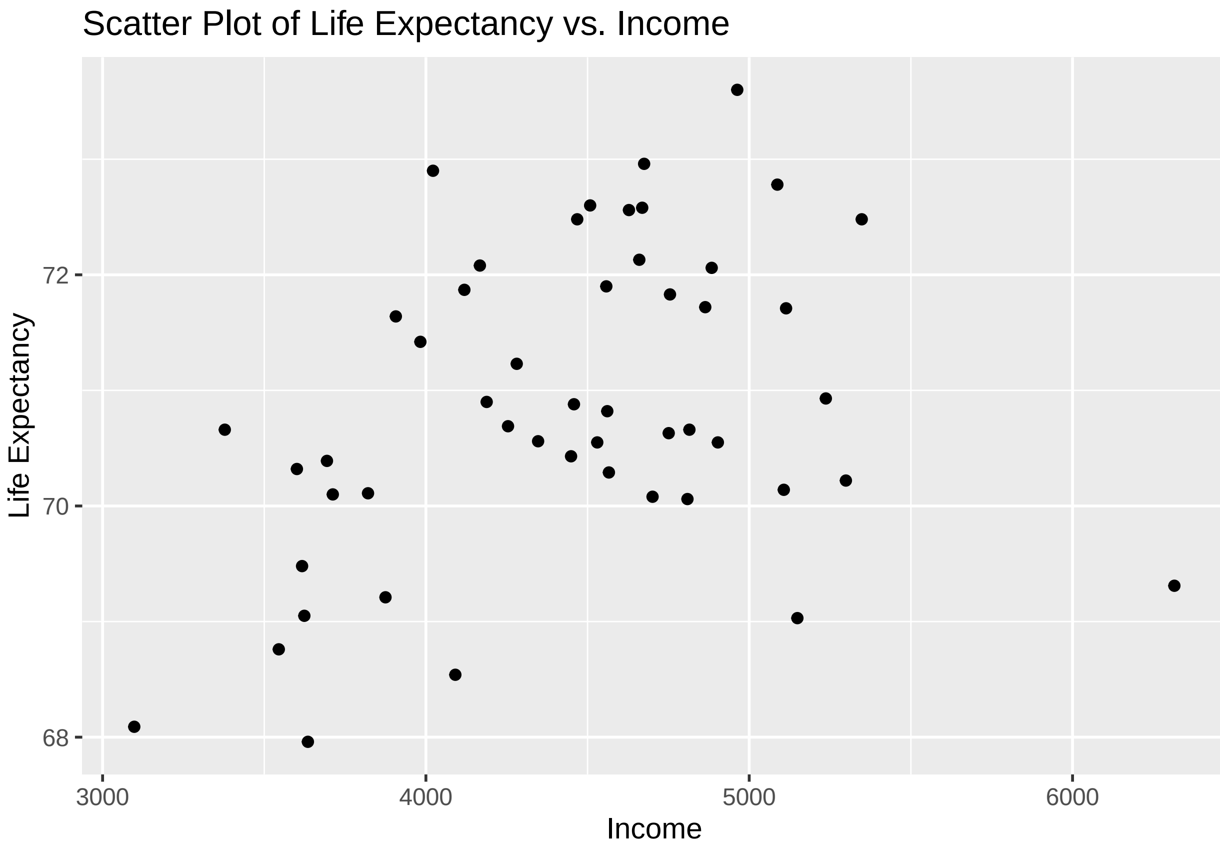
```
##            vars  n     mean       sd   median  trimmed      mad     min
## Population    1 50  4246.42  4464.49  2838.50  3384.27  2890.33  365.00
## Income        2 50  4435.80   614.47  4519.00  4430.08   581.18 3098.00
```

```
## Illiteracy    3 50     1.17     0.61     0.95     1.10     0.52    0.50
## Life Exp      4 50    70.88     1.34    70.67    70.92     1.54   67.96
## Murder        5 50     7.38     3.69     6.85     7.30     5.19    1.40
## HS Grad       6 50    53.11     8.08    53.25    53.34     8.60   37.80
## Frost         7 50   104.46    51.98   114.50   106.80    53.37    0.00
## Area          8 50 70735.88 85327.30 54277.00 56575.73 35144.29 1049.00
##                  max     range  skew kurtosis       se
## Population  21198.0  20833.00  1.92     3.75   631.37
## Income       6315.0   3217.00  0.20     0.24    86.90
## Illiteracy      2.8      2.30  0.82    -0.47     0.09
## Life Exp       73.6      5.64 -0.15    -0.67     0.19
## Murder         15.1     13.70  0.13    -1.21     0.52
## HS Grad        67.3     29.50 -0.32    -0.88     1.14
## Frost         188.0    188.00 -0.37    -0.94     7.35
## Area       566432.0 565383.00  4.10    20.39 12067.10
```

```r
# Optionally, save the statistics to a CSV file for reference
write.csv(detailed_stats, "descriptive_statistics.csv")
```

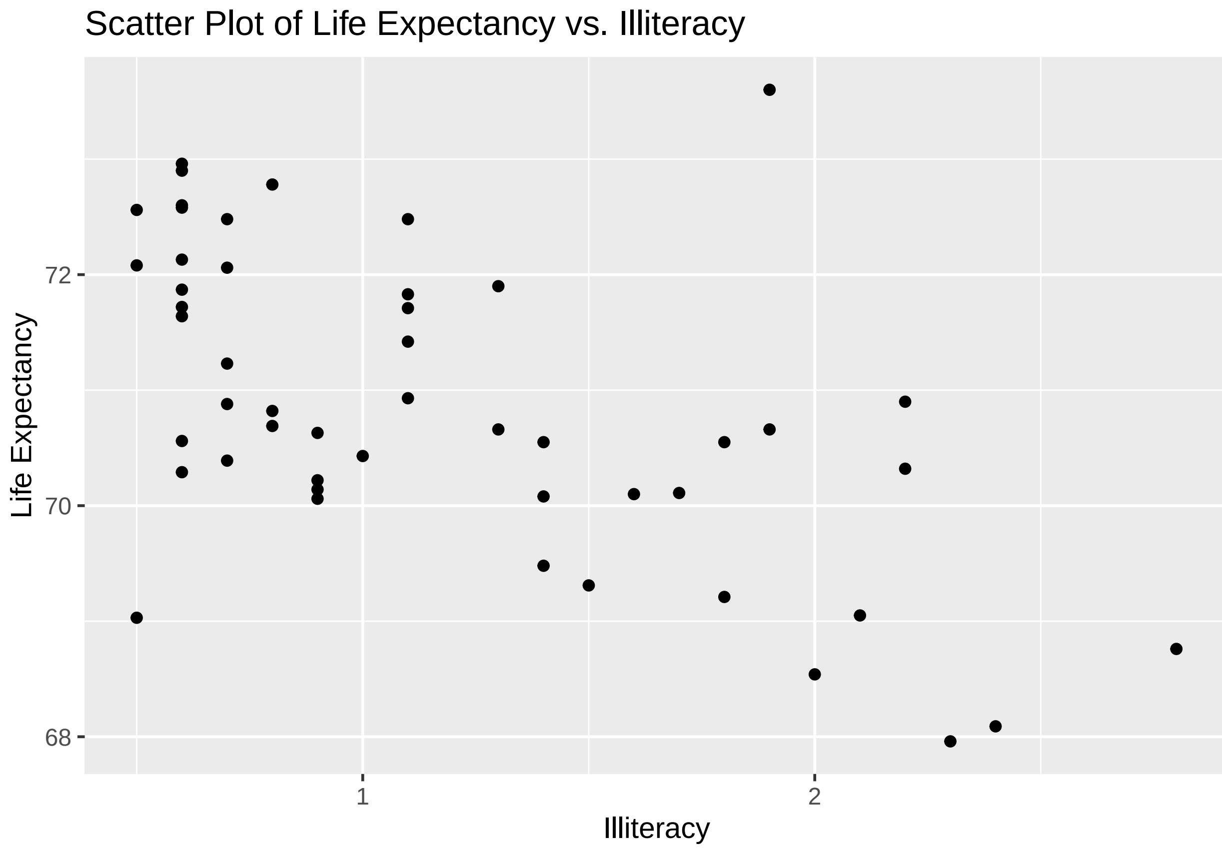
b

```r
# Scatter plot of Life Expectancy vs Income
ggplot(state_data, aes(x = Income, y = `Life Exp`)) +
  geom_point() +
  ggtitle("Scatter Plot of Life Expectancy vs. Income") +
  xlab("Income") +
  ylab("Life Expectancy")
```



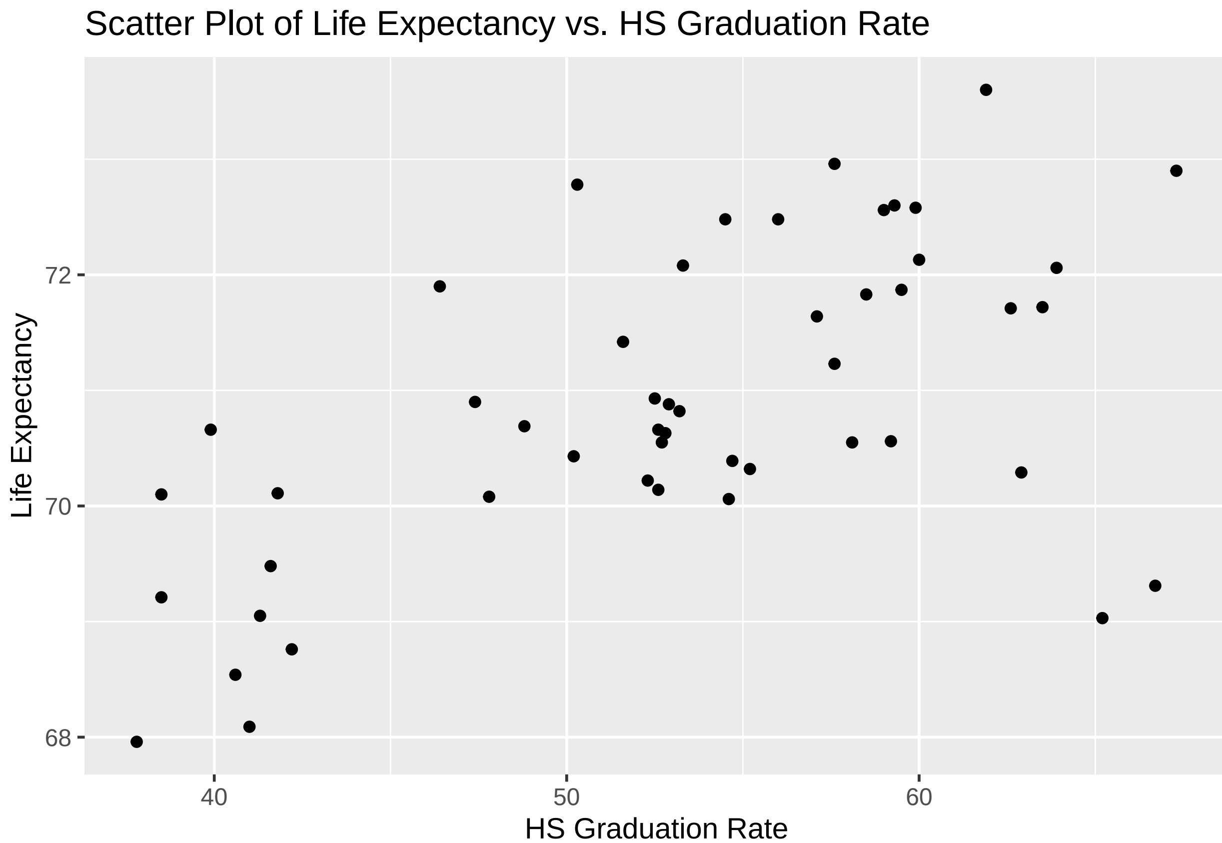
There seems to be a negative relationship between illiteracy rates and life expectancy (higher illiteracy correlates with lower life expectancy). Illiteracy values are relatively low, so no transformation is necessary here.

```
# Scatter plot of Life Expectancy vs Illiteracy
ggplot(state_data, aes(x = Illiteracy, y = `Life Exp`)) +
  geom_point() +
  ggtitle("Scatter Plot of Life Expectancy vs. Illiteracy") +
  xlab("Illiteracy") +
  ylab("Life Expectancy")
```



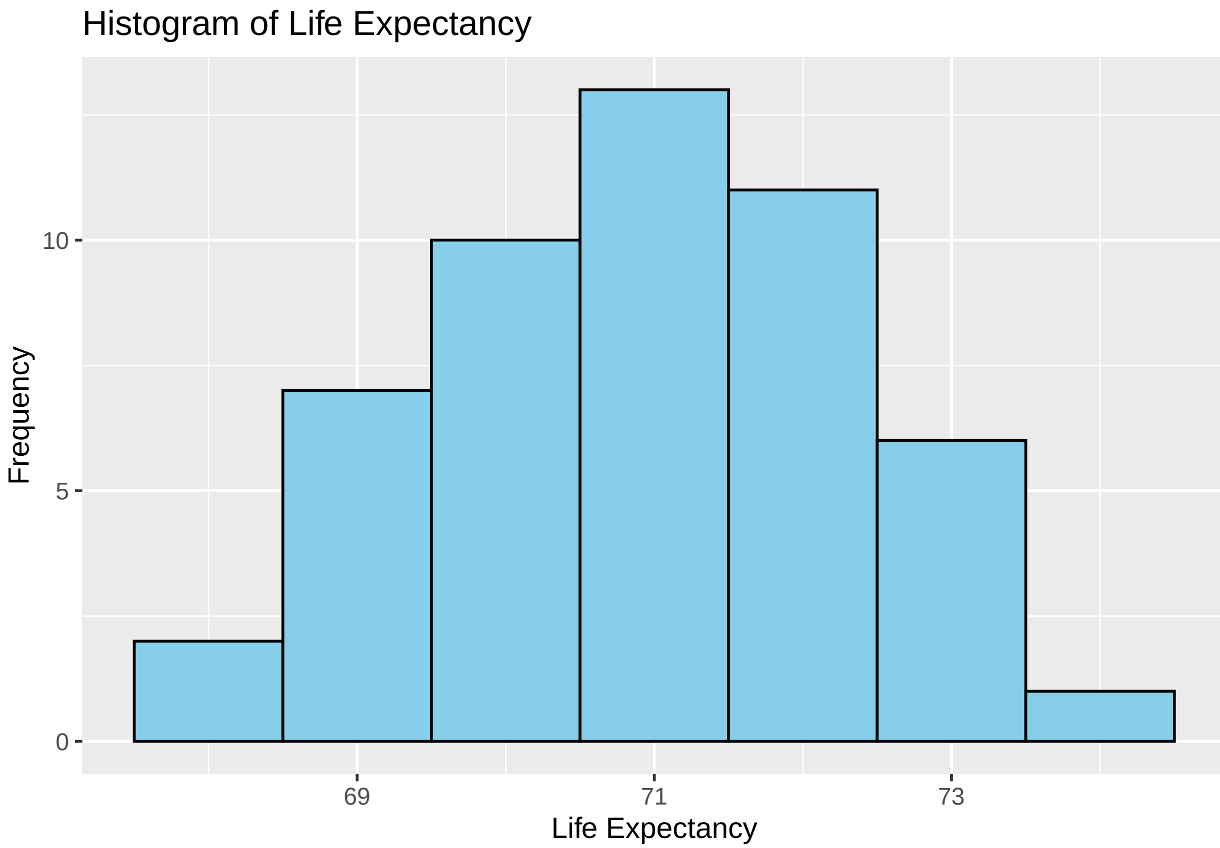
A slight positive trend is visible: states with higher incomes tend to have higher life expectancy. However, there might be diminishing returns for income (non-linear relationship), which suggests a log transformation of income could be beneficial.

```r
# Scatter plot of Life Expectancy vs HS Grad
ggplot(state_data, aes(x = `HS Grad`, y = `Life Exp`)) +
  geom_point() +
  ggtitle("Scatter Plot of Life Expectancy vs. HS Graduation Rate") +
  xlab("HS Graduation Rate") +
  ylab("Life Expectancy")
```



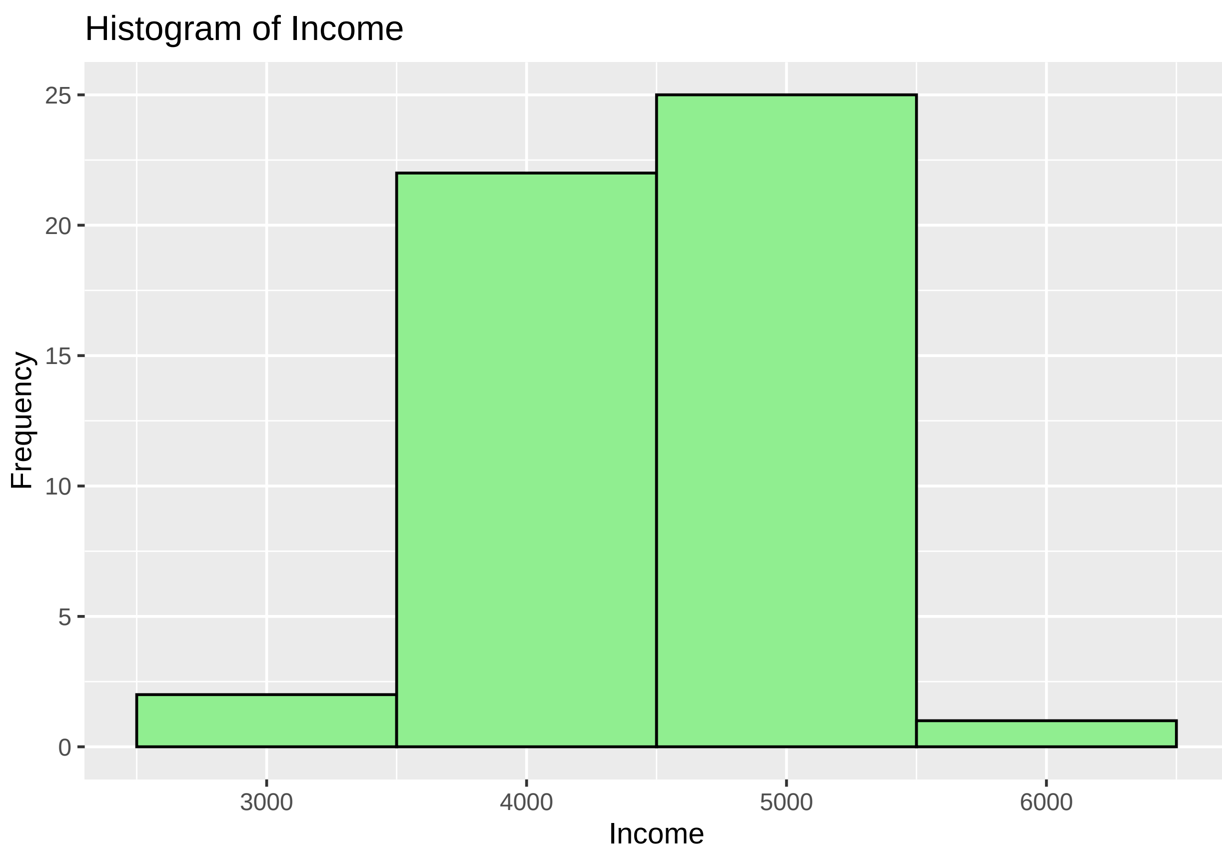
A positive correlation exists between HS graduation rates and life expectancy. No obvious need for transformation here as the relationship looks linear.

```r
# Histogram of Life Expectancy
ggplot(state_data, aes(x = `Life Exp`)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  ggtitle("Histogram of Life Expectancy") +
  xlab("Life Expectancy") +
  ylab("Frequency")
```



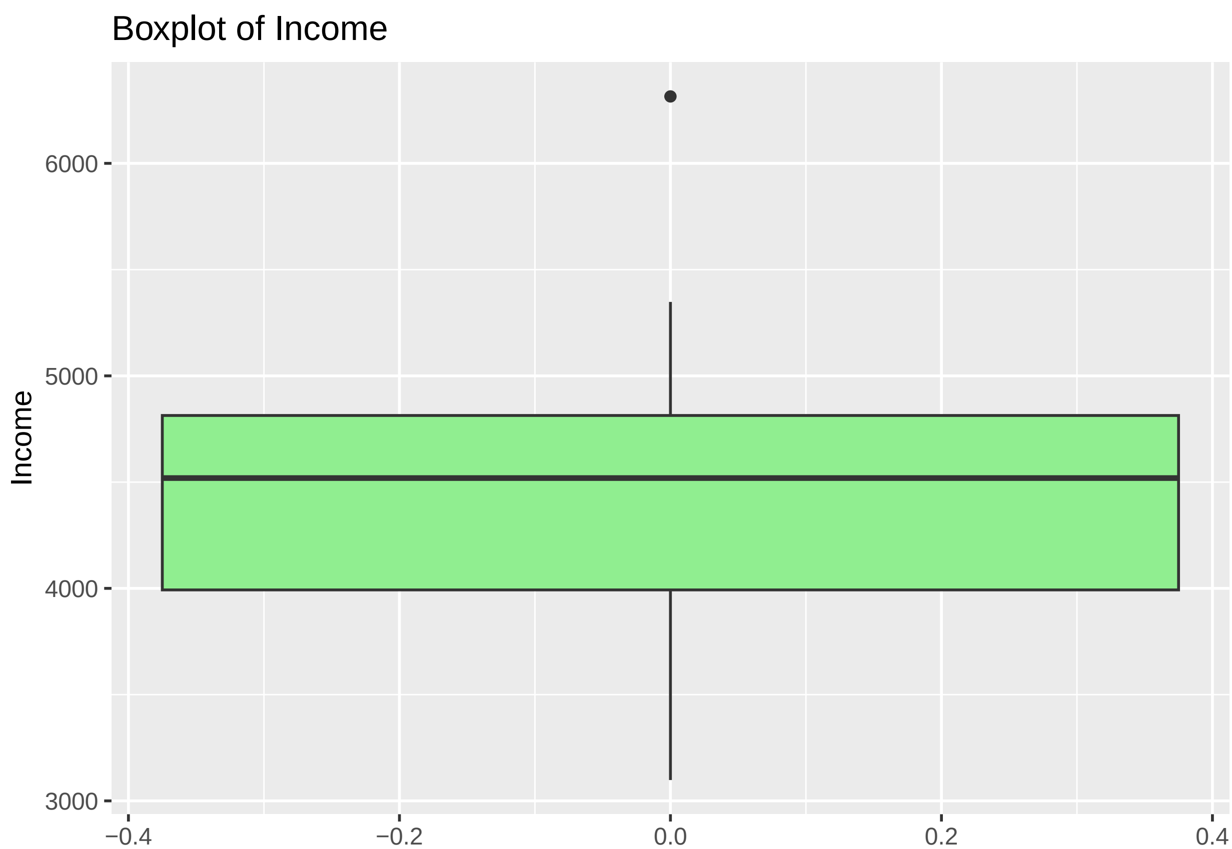
The distribution of life expectancy is approximately symmetric (normal). No transformation needed for life expectancy.

```
# Histogram of Income
ggplot(state_data, aes(x = Income)) +
  geom_histogram(binwidth = 1000, fill = "lightgreen", color = "black") +
  ggtitle("Histogram of Income") +
  xlab("Income") +
  ylab("Frequency")
```

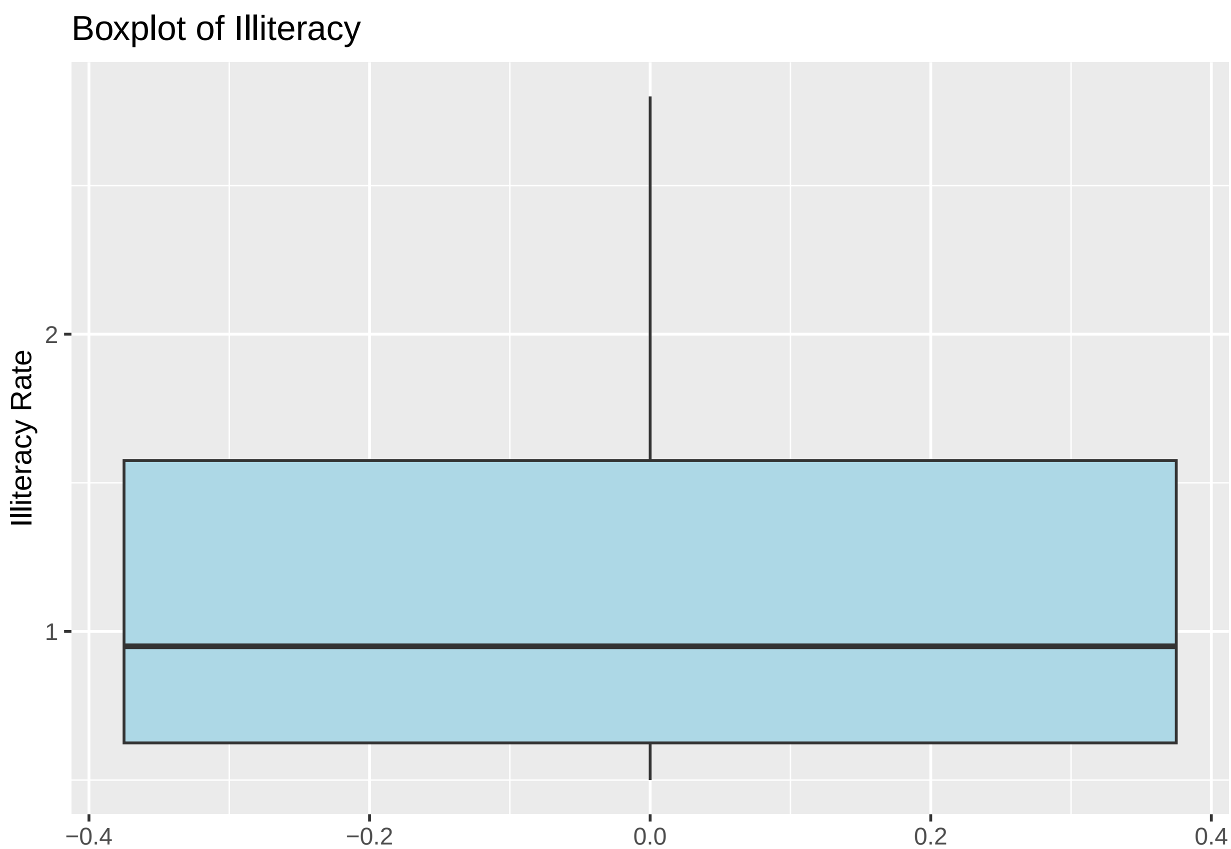


Income distribution is slightly right-skewed, suggesting a potential benefit from a log transformation to normalize the data.

```r
# Boxplot for Income
ggplot(state_data, aes(y = Income)) +
  geom_boxplot(fill = "lightgreen") +
  ggtitle("Boxplot of Income") +
  ylab("Income")
```




```r
# Boxplot for Illiteracy
ggplot(state_data, aes(y = Illiteracy)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot of Illiteracy") +
  ylab("Illiteracy Rate")
```



Income has an outlier (a state with significantly higher income). Log transformation could help reduce its influence. Illiteracy shows no outliers but has a slightly wide spread.

Transformation

```
state_data$Log_Income <- log(state_data$Income)
write.csv(state_data, "state_data_trans.csv", row.names = FALSE)
```

c

```
# Rename columns to avoid issues with spaces
colnames(state_data)[colnames(state_data) == "Life Exp"] <- "Life_Exp"
colnames(state_data)[colnames(state_data) == "HS Grad"] <- "HS_Grad"

# Define the formula for the full model
full_formula <- `Life_Exp` ~ Population + Log_Income + Illiteracy + Murder + `HS_Grad` + Frost + Area

# Perform best subset selection
best_subset <- regsubsets(full_formula, data = state_data, nvmax = 7)

# Summary of the best subset models
subset_summary <- summary(best_subset)
```

```r
# View the best model for each number of predictors
print("Best Subset Models:")
```

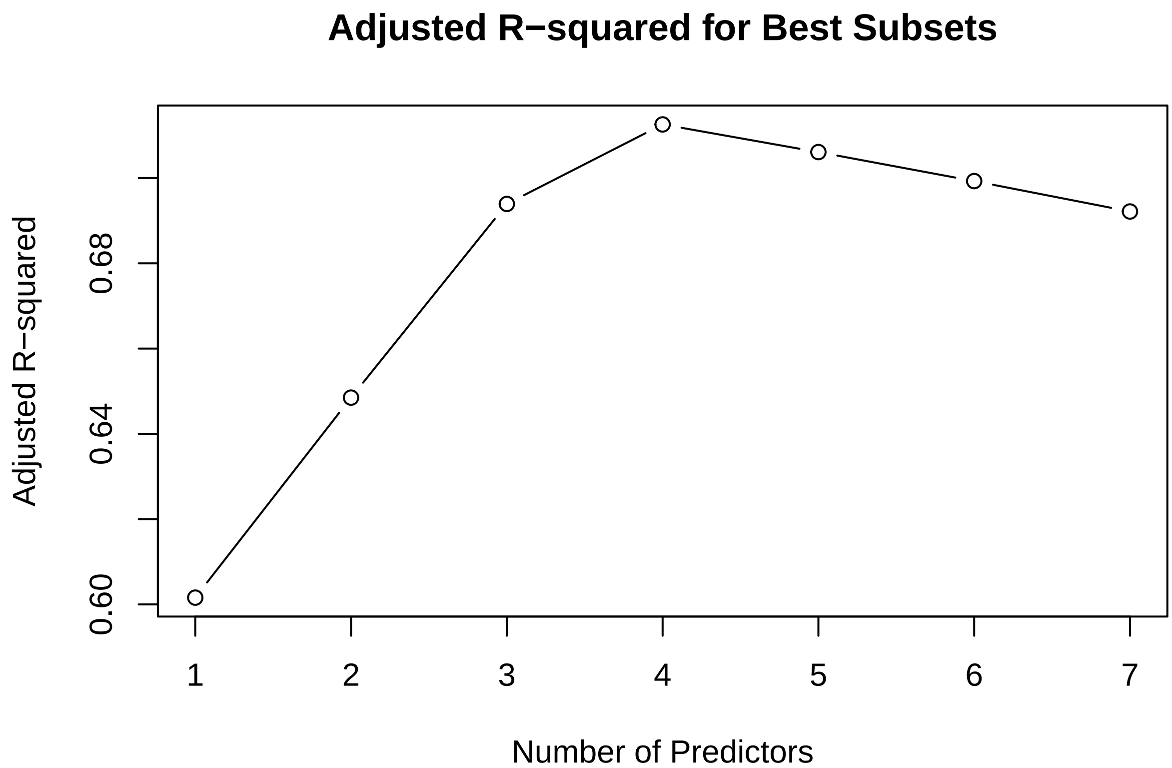
```
## [1] "Best Subset Models:"
```

```r
print(subset_summary)
```

```
## Subset selection object
## Call: regsubsets.formula(full_formula, data = state_data, nvmax = 7)
## 7 Variables  (and intercept)
##            Forced in Forced out
## Population     FALSE      FALSE
## Log_Income     FALSE      FALSE
## Illiteracy     FALSE      FALSE
## Murder         FALSE      FALSE
## HS_Grad        FALSE      FALSE
## Frost          FALSE      FALSE
## Area           FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##          Population Log_Income Illiteracy Murder HS_Grad Frost Area
## 1  ( 1 ) " "        " "        " "        "*"    " "     " "   " "
## 2  ( 1 ) " "        " "        " "        "*"    "*"     " "   " "
## 3  ( 1 ) " "        " "        " "        "*"    "*"     "*"   " "
## 4  ( 1 ) "*"        " "        " "        "*"    "*"     "*"   " "
## 5  ( 1 ) "*"        " "        "*"        "*"    "*"     "*"   " "
## 6  ( 1 ) "*"        " "        "*"        "*"    "*"     "*"   "*"
## 7  ( 1 ) "*"        "*"        "*"        "*"    "*"     "*"   "*"
```

The subset selection object shows models with 1 to 7 predictors were evaluated. The asterisk (*) under the variables indicates whether they were included in the model.

```r
# Plot adjusted R-squared to select the best model
plot(subset_summary$adjr2, type = "b", xlab = "Number of Predictors", ylab = "Adjusted R-squared",
     main = "Adjusted R-squared for Best Subsets")
```



The Adjusted R-squared Plot improves as the number of predictors increases but begins to level off after 4 predictors, suggesting that adding more predictors might not significantly improve the model. The procedures do not always generate the same model, even when focusing on metrics like adjusted R-squared, Cp, or BIC.

From the best subset selection results, we can identify variables that are borderline or may have limited contributions to the model. "Population" does not appear in the best subsets for most models, suggesting it is not strongly predictive of life expectancy. "Frost" and "Area" appear in larger subsets (e.g., 6-7 predictors) but are excluded from smaller subsets, indicating they have weaker predictive power. We can discard "Population", "Frost", and "Area" as these variables show inconsistent inclusion and do not significantly improve the adjusted R-squared or other metrics. Their practical relevance to life expectancy is also less clear (e.g., Area is likely a proxy for other factors like population density).

```
# Correlation between Illiteracy and HS Graduation Rate
correlation <- cor(state_data$Illiteracy, state_data$`HS_Grad`)
print(correlation)
```

```
## [1] -0.6571886
```

The correlation value of -0.657 indicates a moderate-to-strong negative relationship between Illiteracy and HS Grad. This means that as Illiteracy decreases, HS Grad tends to increase, which is expected because they are measures of opposing aspects of education levels. Based on the best subset selection results, both "Illiteracy" and "HS Grad" appear together in subsets with 4 or more predictors.

d

```r
# Perform best subset selection
best_subset <- regsubsets(Life_Exp ~ Population + Log_Income + Illiteracy + Murder + HS_Grad + Frost + A
                          data = state_data, nvmax = 7)

# Function to calculate AIC and BIC for each subset
aic_bic_calculation <- function(model_object, dataset, response_variable) {
  # Initialize storage for AIC and BIC
  aic_values <- numeric()
  bic_values <- numeric()

  for (i in 1:model_object$nvmax) {
    # Safeguard: Try to extract predictors and handle errors
    predictors <- tryCatch({
      names(coef(model_object, id = i))[-1] # Exclude intercept
    }, error = function(e) {
      print(paste("Error extracting subset size", i, "- skipping"))
      return(NULL)
    })

    # Skip iteration if predictors are NULL or empty
    if (is.null(predictors) || length(predictors) == 0) {
      print(paste("Skipping Subset Size", i, "- No Predictors"))
      aic_values[i] <- NA
      bic_values[i] <- NA
      next
    }

    # Ensure predictors exist in the dataset
    valid_predictors <- predictors[predictors %in% colnames(dataset)]

    # Debug: Print extracted and valid predictors
    print(paste("Subset Size:", i, "Predictors:", paste(predictors, collapse = ", ")))
    print(paste("Valid Predictors for Subset Size", i, ":", paste(valid_predictors, collapse = ", ")))

    # Skip subset if no valid predictors
    if (length(valid_predictors) == 0) {
      print(paste("Skipping Subset Size", i, "- No Valid Predictors"))
      aic_values[i] <- NA
      bic_values[i] <- NA
      next
    }

    # Build formula dynamically
    formula_subset <- as.formula(paste(response_variable, "~", paste(valid_predictors, collapse = "+")))

    # Safeguard: Fit the model and handle errors
    model <- tryCatch({
      lm(formula_subset, data = dataset)
    }, error = function(e) {
      print(paste("Error fitting model for subset size", i, "- skipping"))
      return(NULL)
```

```r
    })

    # Skip iteration if model fitting failed
    if (is.null(model)) {
      aic_values[i] <- NA
      bic_values[i] <- NA
      next
    }

    # Calculate AIC and BIC for the model
    aic_values[i] <- AIC(model)
    bic_values[i] <- BIC(model)
  }

  # Return a data frame with the results
  return(data.frame(Num_Predictors = 1:model_object$nvmax, AIC = aic_values, BIC = bic_values))
}

# Apply the function to calculate AIC and BIC
criteria_results <- aic_bic_calculation(best_subset, state_data, "Life_Exp")
```

```
## [1] "Subset Size: 1 Predictors: Murder"
## [1] "Valid Predictors for Subset Size 1 : Murder"
## [1] "Subset Size: 2 Predictors: Murder, HS_Grad"
## [1] "Valid Predictors for Subset Size 2 : Murder, HS_Grad"
## [1] "Subset Size: 3 Predictors: Murder, HS_Grad, Frost"
## [1] "Valid Predictors for Subset Size 3 : Murder, HS_Grad, Frost"
## [1] "Subset Size: 4 Predictors: Population, Murder, HS_Grad, Frost"
## [1] "Valid Predictors for Subset Size 4 : Population, Murder, HS_Grad, Frost"
## [1] "Subset Size: 5 Predictors: Population, Illiteracy, Murder, HS_Grad, Frost"
## [1] "Valid Predictors for Subset Size 5 : Population, Illiteracy, Murder, HS_Grad, Frost"
## [1] "Subset Size: 6 Predictors: Population, Illiteracy, Murder, HS_Grad, Frost, Area"
## [1] "Valid Predictors for Subset Size 6 : Population, Illiteracy, Murder, HS_Grad, Frost, Area"
## [1] "Subset Size: 7 Predictors: Population, Log_Income, Illiteracy, Murder, HS_Grad, Frost, Area"
## [1] "Valid Predictors for Subset Size 7 : Population, Log_Income, Illiteracy, Murder, HS_Grad, Frost
## [1] "Error extracting subset size 8 - skipping"
## [1] "Skipping Subset Size 8 - No Predictors"
```

```r
# View results
print("AIC and BIC Results for Each Subset:")
```

```
## [1] "AIC and BIC Results for Each Subset:"
```

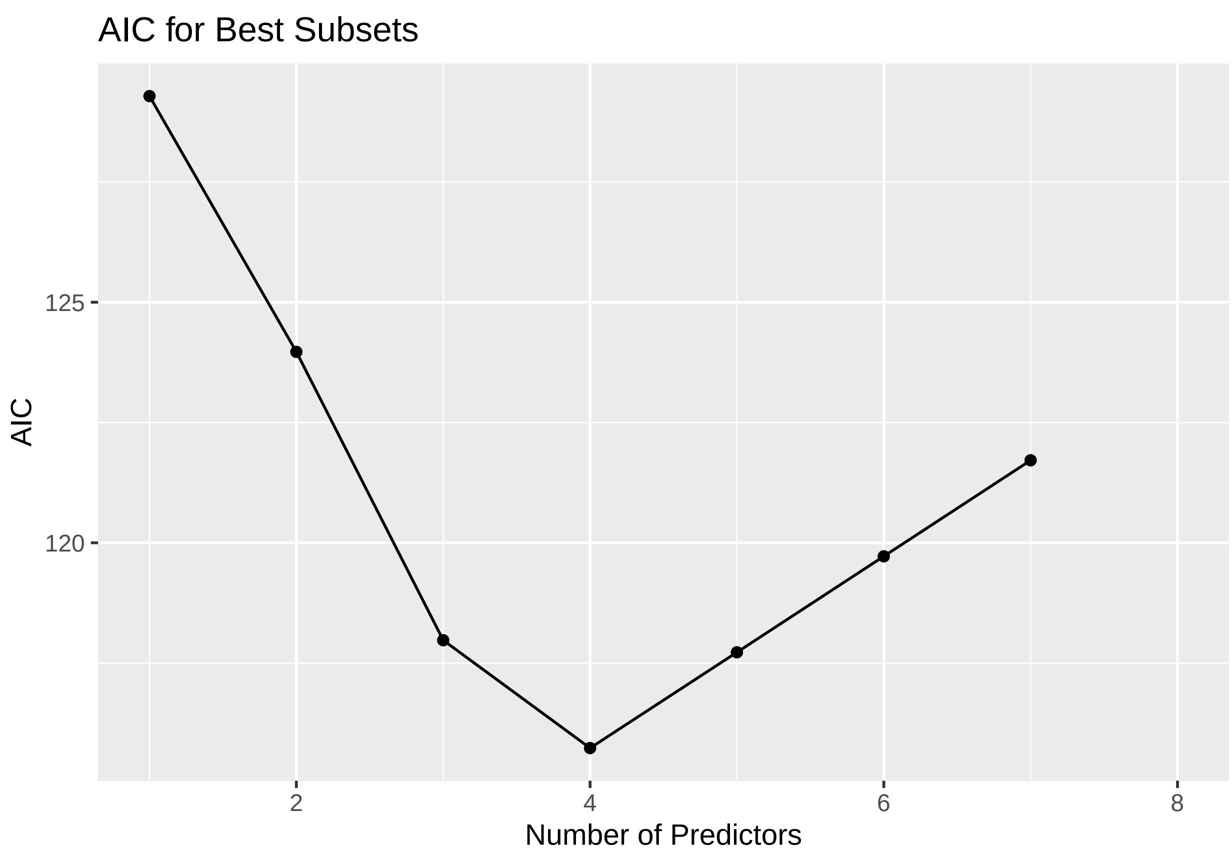
```r
print(criteria_results)
```

```
##   Num_Predictors      AIC      BIC
## 1              1 129.2846 135.0207
## 2              2 123.9684 131.6165
## 3              3 117.9743 127.5344
## 4              4 115.7326 127.2048
## 5              5 117.7242 131.1084
## 6              6 119.7187 135.0149
```

```
## 7                7 121.7146 138.9228
## 8                8       NA       NA
```

```r
# Plot AIC
ggplot(criteria_results, aes(x = Num_Predictors, y = AIC)) +
  geom_line() + geom_point() +
  ggtitle("AIC for Best Subsets") +
  xlab("Number of Predictors") +
  ylab("AIC")
```

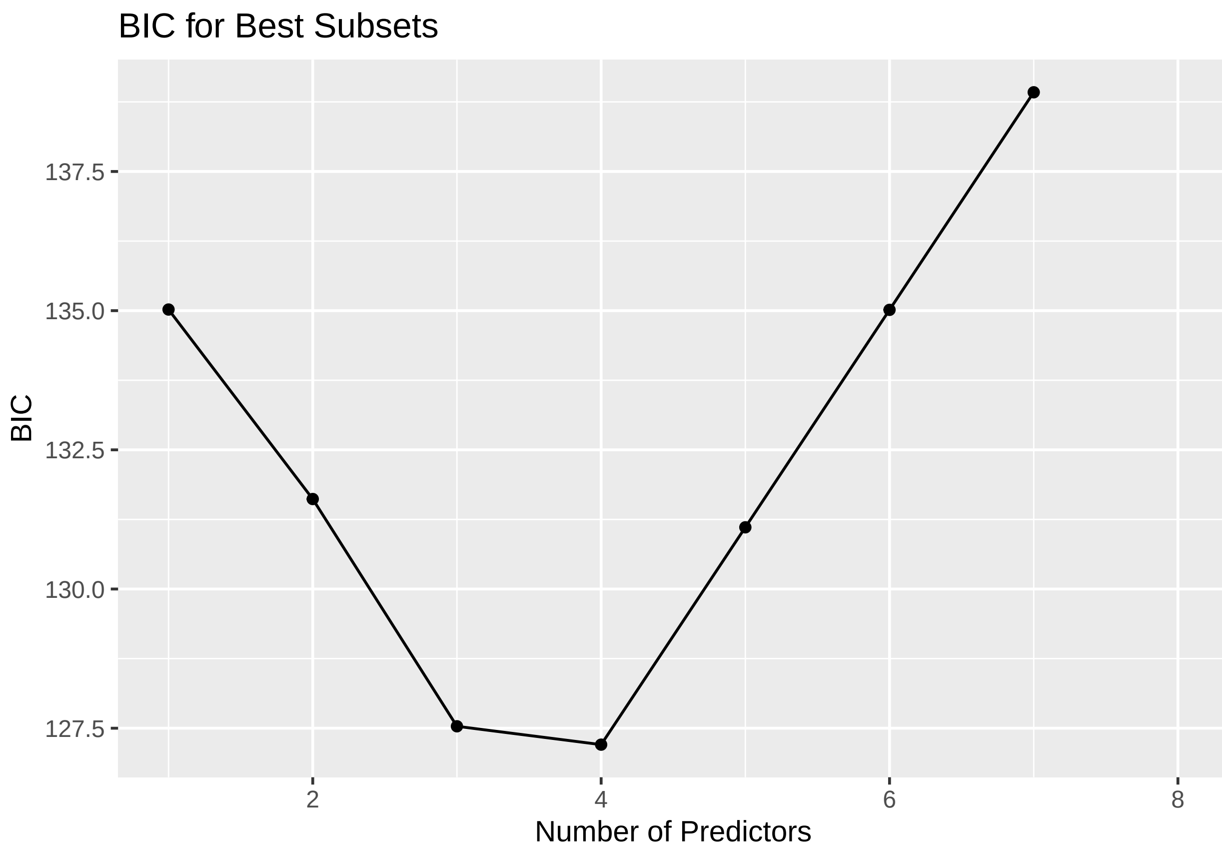
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
# Plot BIC
ggplot(criteria_results, aes(x = Num_Predictors, y = BIC)) +
  geom_line() + geom_point() +
  ggtitle("BIC for Best Subsets") +
  xlab("Number of Predictors") +
  ylab("BIC")
```

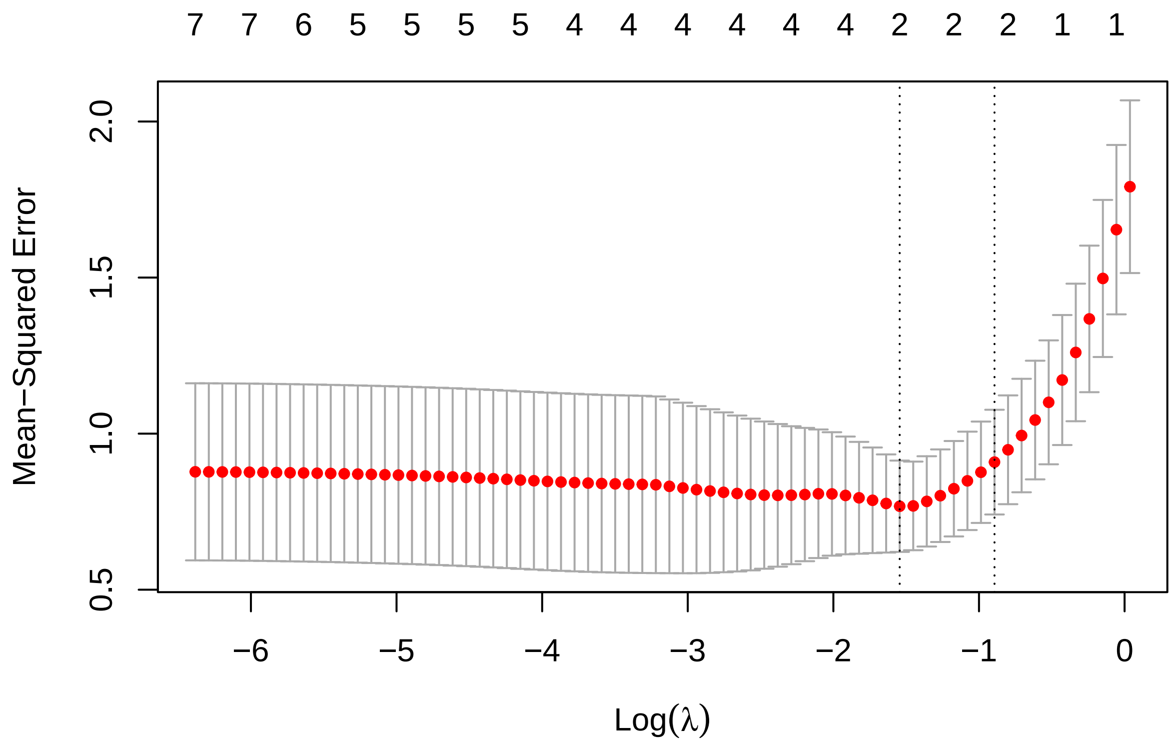
```
## Warning: Removed 1 row containing missing values or values outside the scale range
```

```
## (`geom_line()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



e

```r
# Prepare the predictors (X) and response (y)
X <- model.matrix(Life_Exp ~ Population + Log_Income + Illiteracy + Murder + HS_Grad + Frost + Area,
                  data = state_data)[, -1]  # Remove the intercept
y <- state_data$Life_Exp

# Perform LASSO regression with cross-validation
lasso_cv <- cv.glmnet(X, y, alpha = 1, family = "gaussian")

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




```r
# Print the best lambda (minimizing cross-validated error)
best_lambda <- lasso_cv$lambda.min
print(paste("Best lambda (lambda.min):", best_lambda))
```

```
## [1] "Best lambda (lambda.min): 0.213397565495152"
```

```r
# Print the largest lambda within 1 standard error of the minimum (simpler model)
lambda_1se <- lasso_cv$lambda.1se
print(paste("Lambda within 1 SE of minimum (lambda.1se):", lambda_1se))
```

```
## [1] "Lambda within 1 SE of minimum (lambda.1se): 0.40927738067908"
```

```r
# Extract coefficients for the best lambda
lasso_coefs <- coef(lasso_cv, s = best_lambda)

# Print the coefficients
print("LASSO Coefficients at Best Lambda:")
```

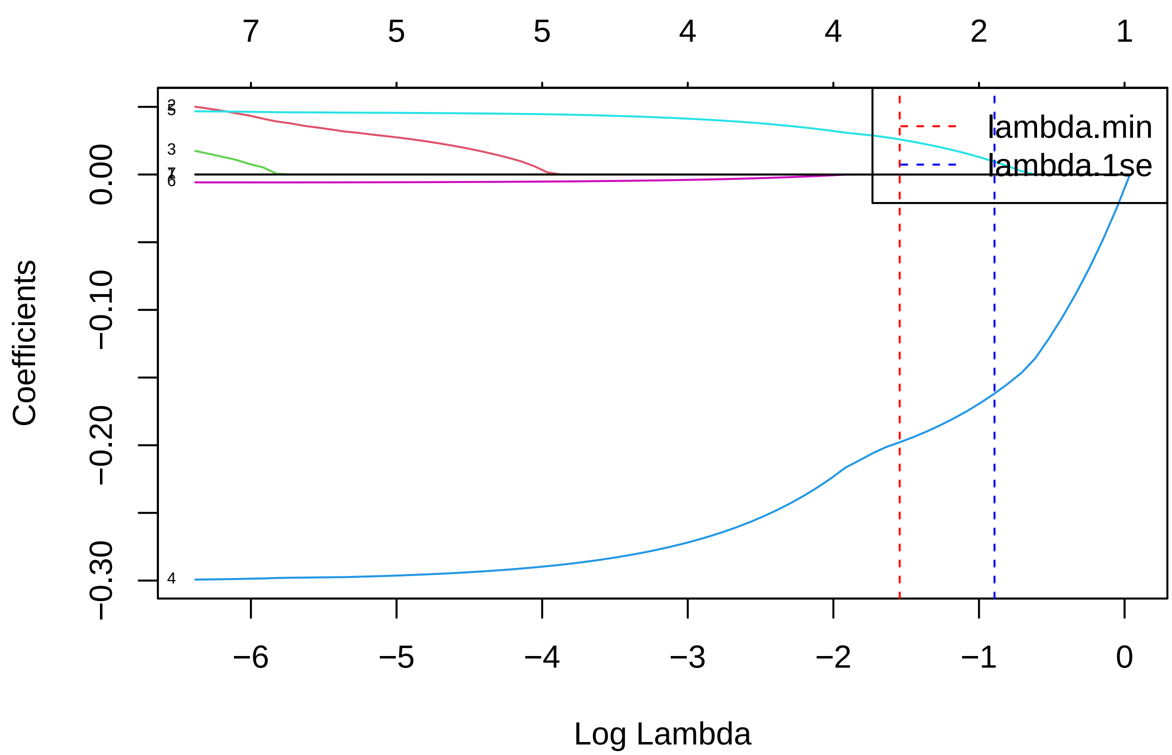
```
## [1] "LASSO Coefficients at Best Lambda:"
```

```r
print(lasso_coefs)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                     s1
## (Intercept) 70.9602156
## Population   .
## Log_Income   .
## Illiteracy   .
## Murder      -0.1978520
## HS_Grad      0.0259497
## Frost        .
## Area         .
```

```
# Perform LASSO without cross-validation for visualization
lasso_fit <- glmnet(X, y, alpha = 1)

# Plot coefficient paths
plot(lasso_fit, xvar = "lambda", label = TRUE)
abline(v = log(best_lambda), col = "red", lty = 2)
abline(v = log(lambda_1se), col = "blue", lty = 2)
legend("topright", legend = c("lambda.min", "lambda.1se"), col = c("red", "blue"), lty = 2)
```



f

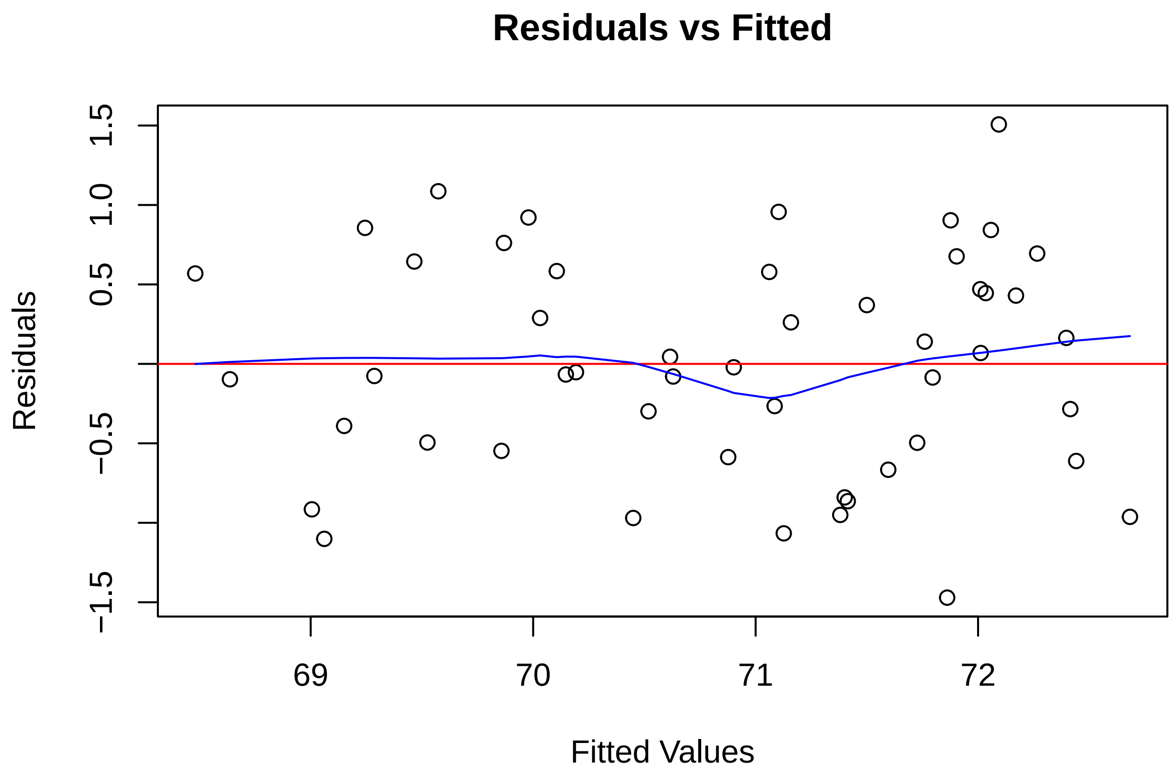
From the results we got, all three methods consistently select the same 4 predictors: Population, Murder, HS_Grad, and Frost. The results from all three methods reinforce confidence in the robustness of this subset. I will say The 4-predictor model with Population, Murder, HS_Grad, and Frost is the best choice. We can see that LASSO (lambda.min), which minimizes cross-validated error, supports predictive performance and AIC/BIC penalize complexity, and LASSO further addresses multicollinearity and redundancy.

```
final_model <- lm(Life_Exp ~ Population + Murder + HS_Grad + Frost, data = state_data)
summary(final_model)
```

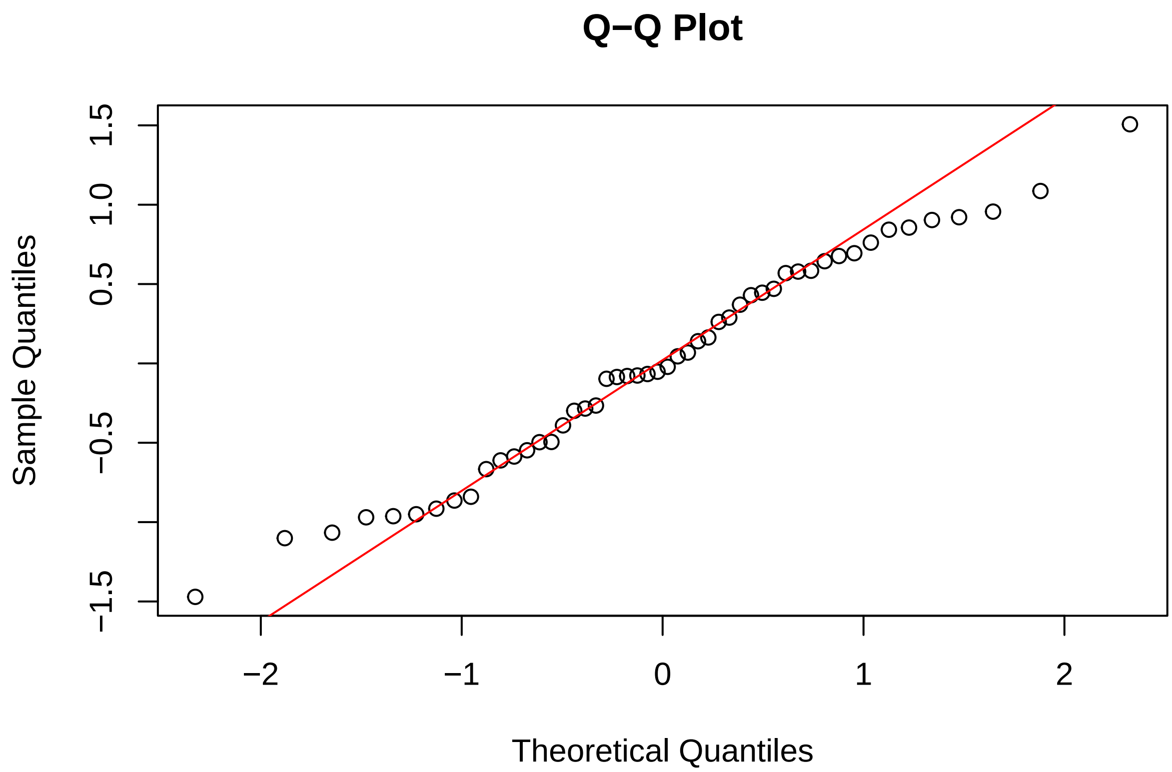
```
##
## Call:
## lm(formula = Life_Exp ~ Population + Murder + HS_Grad + Frost,
##     data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47095 -0.53464 -0.03701  0.57621  1.50683
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.103e+01  9.529e-01  74.542  < 2e-16 ***
## Population   5.014e-05  2.512e-05   1.996  0.05201 .
## 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 *
## ---
## 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
```

Check Model Assumptions

```
# Plot residuals vs fitted values
plot(final_model$fitted.values, residuals(final_model),
     main = "Residuals vs Fitted",
     xlab = "Fitted Values", ylab = "Residuals")
abline(h = 0, col = "red")

# Add a lowess smooth curve to check for non-linear patterns
lines(lowess(final_model$fitted.values, residuals(final_model)), col = "blue")
```



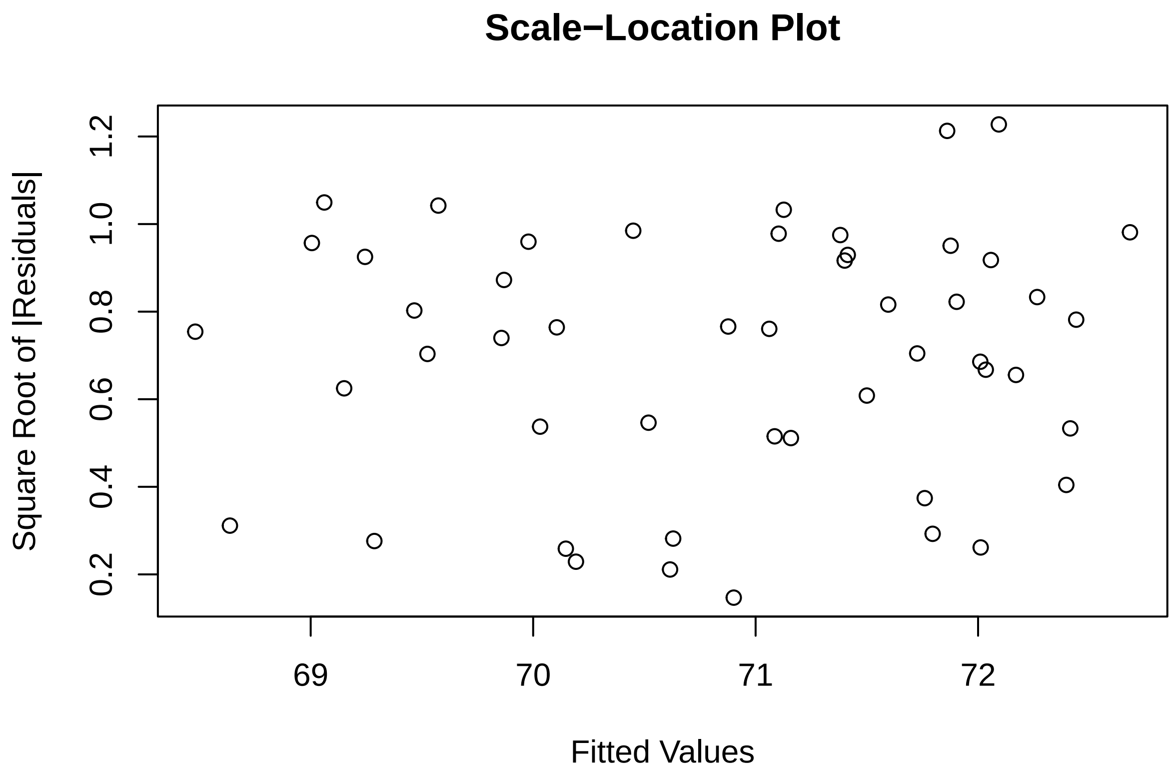

```
# Q-Q plot for residuals
qqnorm(residuals(final_model), main = "Q-Q Plot")
qqline(residuals(final_model), col = "red")
```




```r
# Shapiro-Wilk test for normality
shapiro_test <- shapiro.test(residuals(final_model))
print(shapiro_test)
```

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

```r
# Scale-Location plot
plot(final_model$fitted.values, sqrt(abs(residuals(final_model))),
     main = "Scale-Location Plot",
     xlab = "Fitted Values", ylab = "Square Root of |Residuals|")
abline(h = 0, col = "red")
```




```r
# Variance Inflation Factor (VIF)
vif_values <- vif(final_model)
print(vif_values)
```

```
## Population     Murder    HS_Grad      Frost
##   1.189835   1.727844   1.356791   1.498077
```

All assumptions of linear regression appear to be reasonably met based on the visualizations and test results. The model is well-specified and suitable for inference.

Test Model Predictive Ability Using 10-Fold Cross-Validation

```r
# Define training control with 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)

# Fit the model using caret's train function
cv_model <- train(Life_Exp ~ Population + Murder + HS_Grad + Frost,
                  data = state_data,
                  method = "lm",
                  trControl = train_control)

# Print cross-validation results
print(cv_model)
```

```
## Linear Regression
##
## 50 samples
##  4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 46, 45, 43, 44, 45, 46, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.7324485  0.7210932  0.6439614
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The cross-validation results suggest that the final model performs well in predicting life expectancy. The R-squared value indicates that the model explains a significant proportion of the variance, and the RMSE and MAE values suggest that prediction errors are relatively small.

g

The analysis explored the factors influencing life expectancy across states, focusing on identifying the best predictors from several socioeconomic and environmental variables. Using best subset selection, criterion-based methods, and LASSO regression, we identified four key predictors: Population, Murder Rate, High School Graduation Rate, and Frost Days. These variables were consistently selected as significant across different methodologies. The final model explained approximately 77% of the variability in life expectancy ($R^2 = 0.7692$) and demonstrated robust predictive performance through 10-fold cross-validation (RMSE = 0.7741). The findings suggest that reducing crime (Murder Rate) and improving education (High School Graduation Rate) could significantly improve life expectancy. Environmental factors, such as Frost Days, also play a role, though their relationship may be more complex. These results provide actionable insights into key areas for policy intervention to enhance public health outcomes.