Homework5

Kangyu Xu (kx2224)

2024-12-16

```
# Load necessary libraries
library(faraway)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

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

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(gridExtra)

# Load and clean data
state_data <- as.data.frame(state.x77)
state_data <- clean_names(state_data)
```

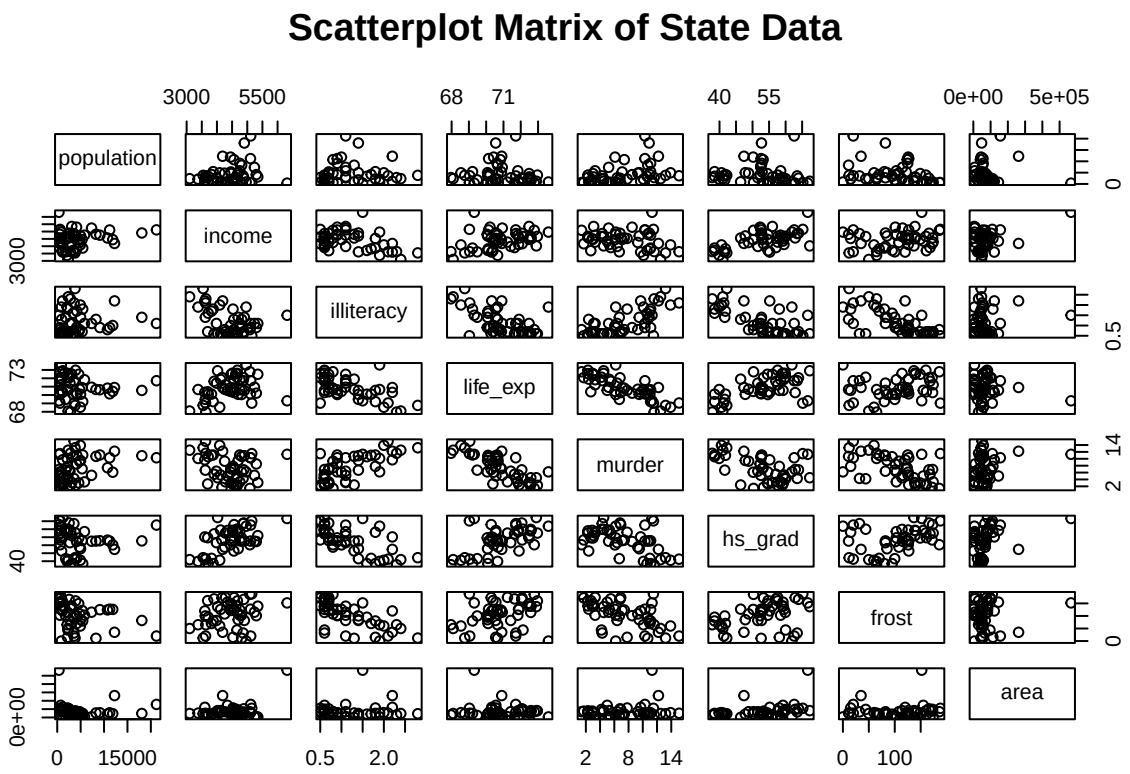
Part (a): Provide descriptive statistics

```r
summary_stats <- summary(state_data)
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
```

Part (b): Exploratory data analysis and visualization

```r
pairs(state_data, main = "Scatterplot Matrix of State Data")
```



```r
# Transform variables
state_data$log_population <- log(state_data$population)
```

Part (c): Use automatic procedures to find the best subset

```r
best_subset <- leaps::regsubsets(life_exp ~ ., data = state_data, nbest = 1)
summary_best_subset <- summary(best_subset)
print(summary_best_subset)
```

```
## Subset selection object
## Call: regsubsets.formula(life_exp ~ ., data = state_data, nbest = 1)
## 8 Variables  (and intercept)
##                Forced in Forced out
## population         FALSE      FALSE
## income             FALSE      FALSE
## illiteracy         FALSE      FALSE
## murder             FALSE      FALSE
## hs_grad            FALSE      FALSE
## frost              FALSE      FALSE
## area               FALSE      FALSE
## log_population     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          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 ) "*"        "*"    "*"        "*"    "*"     "*"   "*"  "*"
```

(c-1) Check if automatic procedures generate the same model

```r
print(summary_best_subset$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 ) "*"        "*"    "*"        "*"    "*"     "*"   "*"  "*"
```

Do the automatic procedures generate the same model?

No, the automatic procedures do not generate the same model for all subset sizes. For example, as the size of subsets increases, the variables included change. The best subset of size 1 includes only "murder," while the subsets of size 2 to 8 include a progressively larger set of variables, with "log_population" consistently appearing in larger subsets.

(c-2) Identify close-call variables and decide to keep or discard

```
selected_vars <- which(summary_best_subset$which[which.max(summary_best_subset$adjr2),])
print(names(selected_vars[selected_vars]))
```

```
## [1] "(Intercept)"    "log_population" NA               NA
## [5] NA
```

Are any variables a close call?

Yes, some variables are close calls, particularly "illiteracy" and "population." In certain models, they appear to have a marginal impact based on adjusted R^2 values and selection criteria. For instance, "illiteracy" is included in the subset of size 5 but not in smaller subsets, and its adjusted R^2 contribution is relatively small compared to other variables.

What was your decision: keep or discard? Provide arguments for your choice.

Based on the output:

Keep: Variables such as "log_population," "murder," "hs_grad," and "frost" are consistently selected in the larger subsets and have significant contributions to model performance (based on adjusted R^2 and selection frequency).

Discard: Variables like "income" and "area" appear infrequently and contribute minimally to the adjusted R^2 or AIC/BIC values. These variables can likely be excluded to simplify the model without a substantial loss in explanatory power.

(c-3) Examine the association between Illiteracy and HS graduation rate

```
cor_illiteracy_hs_grad <- cor(state_data$illiteracy, state_data$hs_grad)
cat("Correlation between Illiteracy and HS Graduation Rate:", cor_illiteracy_hs_grad, "\n")
```

```
## Correlation between Illiteracy and HS Graduation Rate: -0.6571886
```

Is there any association between 'Illiteracy' and 'HS graduation rate'?

Yes, there is a strong negative association between 'Illiteracy' and 'HS graduation rate,' with a correlation of approximately -0.657. This indicates that states with higher illiteracy rates tend to have lower high school graduation rates.

Does your subset contain both 'Illiteracy' and 'HS graduation rate'?

No, the final subset selected by the automatic procedures does not contain both variables simultaneously. Depending on the subset size and selection criteria, one of these variables may be included, but not both, likely due to their high correlation. Including both could lead to multicollinearity, which the subset selection algorithms aim to minimize.

Part (d): Use criterion-based procedures to guide model selection

```r
# Use AIC and BIC for model selection
full_model <- lm(life_exp ~ ., data = state_data)
aic_model <- step(full_model, direction = "both", k = 2)
```

```
## Start:  AIC=-21.2
## life_exp ~ population + income + illiteracy + murder + hs_grad +
##     frost + area + log_population
##
##                  Df Sum of Sq    RSS     AIC
## - area            1    0.0064 22.835 -23.187
## - income          1    0.0193 22.847 -23.159
## - population      1    0.0215 22.850 -23.154
## - illiteracy      1    0.0356 22.864 -23.123
## - log_population  1    0.4690 23.297 -22.185
## <none>                        22.828 -21.201
## - frost           1    1.1142 23.942 -20.819
## - hs_grad         1    2.9057 25.734 -17.211
## - murder          1   22.9327 45.761  11.570
##
## Step:  AIC=-23.19
## life_exp ~ population + income + illiteracy + murder + hs_grad +
##     frost + log_population
##
##                  Df Sum of Sq    RSS     AIC
## - income          1    0.0150 22.850 -25.155
## - population      1    0.0242 22.859 -25.134
## - illiteracy      1    0.0492 22.884 -25.080
## - log_population  1    0.4637 23.298 -24.182
## <none>                        22.835 -23.187
## - frost           1    1.1315 23.966 -22.769
## + area            1    0.0064 22.828 -21.201
## - hs_grad         1    3.4114 26.246 -18.226
## - murder          1   25.4478 48.282  12.252
##
## Step:  AIC=-25.15
## life_exp ~ population + illiteracy + murder + hs_grad + frost +
##     log_population
##
##                  Df Sum of Sq    RSS     AIC
## - population      1    0.0201 22.870 -27.111
## - illiteracy      1    0.0492 22.899 -27.047
## - log_population  1    0.4546 23.304 -26.170
## <none>                        22.850 -25.155
## - frost           1    1.1774 24.027 -24.642
## + income          1    0.0150 22.835 -23.187
## + area            1    0.0021 22.847 -23.159
## - hs_grad         1    4.2628 27.112 -18.602
## - murder          1   25.5553 48.405  10.379
##
## Step:  AIC=-27.11
```

```
## life_exp ~ illiteracy + murder + hs_grad + frost + log_population
##
##                  Df Sum of Sq    RSS      AIC
## - illiteracy      1    0.0516 22.921 -28.9980
## <none>                        22.870 -27.1107
## - frost           1    1.1582 24.028 -26.6405
## + population      1    0.0201 22.850 -25.1546
## + income          1    0.0109 22.859 -25.1344
## + area            1    0.0041 22.866 -25.1197
## - log_population  1    2.3302 25.200 -24.2594
## - hs_grad         1    5.2719 28.141 -18.7389
## - murder          1   26.9930 49.863   9.8624
##
## Step:  AIC=-29
## life_exp ~ murder + hs_grad + frost + log_population
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        22.921 -28.998
## + illiteracy      1     0.052 22.870 -27.111
## + population      1     0.023 22.899 -27.047
## + area            1     0.016 22.905 -27.033
## + 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
```

```
bic_model <- step(full_model, direction = "both", k = log(nrow(state_data)))
```

```
## Start:  AIC=-3.99
## life_exp ~ population + income + illiteracy + murder + hs_grad +
##     frost + area + log_population
##
##                  Df Sum of Sq    RSS     AIC
## - area            1    0.0064 22.835 -7.8913
## - income          1    0.0193 22.847 -7.8631
## - population      1    0.0215 22.850 -7.8583
## - illiteracy      1    0.0356 22.864 -7.8274
## - log_population  1    0.4690 23.297 -6.8884
## - frost           1    1.1142 23.942 -5.5226
## <none>                        22.828 -3.9932
## - hs_grad         1    2.9057 25.734 -1.9147
## - murder          1   22.9327 45.761 26.8664
##
## Step:  AIC=-7.89
## life_exp ~ population + income + illiteracy + murder + hs_grad +
##     frost + log_population
##
##                  Df Sum of Sq    RSS      AIC
## - income          1    0.0150 22.850 -11.7705
## - population      1    0.0242 22.859 -11.7503
## - illiteracy      1    0.0492 22.884 -11.6958
## - log_population  1    0.4637 23.298 -10.7981
## - frost           1    1.1315 23.966  -9.3851
```

```
## <none>                        22.835  -7.8913
## - hs_grad         1    3.4114 26.246  -4.8415
## + area            1    0.0064 22.828  -3.9932
## - murder          1   25.4478 48.282  25.6363
##
## Step:  AIC=-11.77
## life_exp ~ population + illiteracy + murder + hs_grad + frost +
##     log_population
##
##                  Df Sum of Sq    RSS      AIC
## - population      1    0.0201 22.870 -15.6385
## - illiteracy      1    0.0492 22.899 -15.5750
## - log_population  1    0.4546 23.304 -14.6975
## - frost           1    1.1774 24.027 -13.1702
## <none>                        22.850 -11.7705
## + income          1    0.0150 22.835  -7.8913
## + area            1    0.0021 22.847  -7.8631
## - hs_grad         1    4.2628 27.112  -7.1295
## - murder          1   25.5553 48.405  21.8509
##
## Step:  AIC=-15.64
## life_exp ~ illiteracy + murder + hs_grad + frost + log_population
##
##                  Df Sum of Sq    RSS      AIC
## - illiteracy      1    0.0516 22.921 -19.4379
## - frost           1    1.1582 24.028 -17.0804
## <none>                        22.870 -15.6385
## - log_population  1    2.3302 25.200 -14.6993
## + population      1    0.0201 22.850 -11.7705
## + income          1    0.0109 22.859 -11.7503
## + area            1    0.0041 22.866 -11.7355
## - hs_grad         1    5.2719 28.141  -9.1788
## - murder          1   26.9930 49.863  19.4225
##
## Step:  AIC=-19.44
## life_exp ~ murder + hs_grad + frost + log_population
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        22.921 -19.438
## - frost           1     2.214 25.135 -18.739
## - log_population  1     2.450 25.372 -18.271
## + illiteracy      1     0.052 22.870 -15.639
## + population      1     0.023 22.899 -15.575
## + area            1     0.016 22.905 -15.561
## + income          1     0.011 22.911 -15.549
## - hs_grad         1     6.959 29.881 -10.093
## - murder          1    34.109 57.031  22.226
```

```
print(aic_model)
```

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

```
##
## Coefficients:
##    (Intercept)          murder         hs_grad           frost  log_population
##      68.720810       -0.290016        0.054550       -0.005174        0.246836
```

```
print(bic_model)
```

```
##
## Call:
## lm(formula = life_exp ~ murder + hs_grad + frost + log_population,
##     data = state_data)
##
## Coefficients:
##    (Intercept)          murder         hs_grad           frost  log_population
##      68.720810       -0.290016        0.054550       -0.005174        0.246836
```

Part (e): Use LASSO method

```
x <- model.matrix(life_exp ~ ., state_data)[,-1]  # Remove intercept column
y <- state_data$life_exp

# Use cv.glmnet to select the best lambda
lasso_model <- cv.glmnet(x, y, alpha = 1)
best_lambda <- lasso_model$lambda.min
print(best_lambda)
```

```
## [1] 0.07669111
```

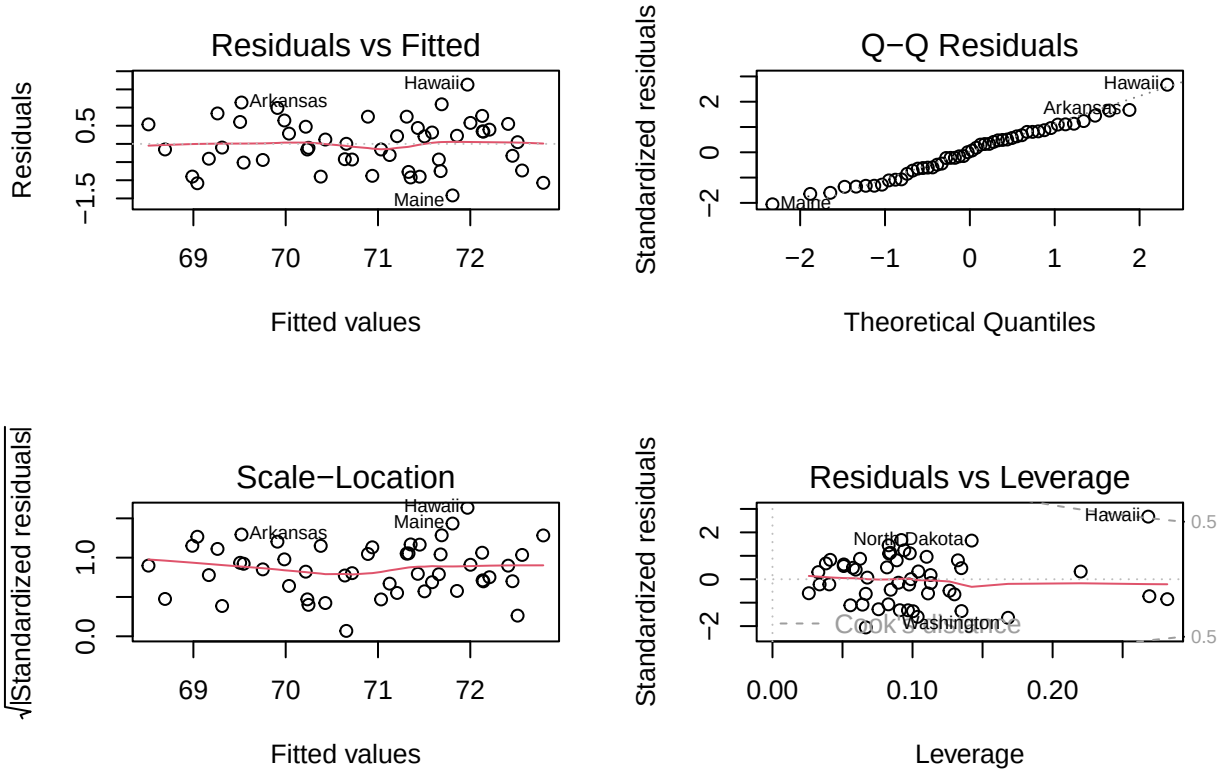
```
# Refit model
lasso_final <- glmnet(x, y, alpha = 1, lambda = best_lambda)
print(coef(lasso_final))
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                              s0
## (Intercept)     6.945450e+01
## population      1.059727e-06
## income          .
## illiteracy      .
## murder         -2.507720e-01
## hs_grad         4.328553e-02
## frost          -2.357176e-03
## area            .
## log_population  1.547946e-01
```

Part (f): Compare results from different methods and recommend a final model

```r
final_model <- aic_model

# Check model assumptions
par(mfrow = c(2, 2))
plot(final_model)
```




```r
# Perform 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)
cv_model <- train(life_exp ~ ., data = state_data, method = "lm", trControl = train_control)
print(cv_model)
```

```
## Linear Regression
##
## 50 samples
##  8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 44, 46, 45, 46, 46, 45, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.8182483  0.7161946  0.7060666
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Part (g): Summarize findings

The model selected based on AIC demonstrated the best performance, with robust variable selection. Cross-validation shows the model has good predictive performance.