

STAT 534: Homework 3

Erick Calderon-Morales

Fall 2021

```
library(leaps)
library(gt)
library(tidyverse)
library(glmnet)
library(janitor)
library(MASS)
library(ISLR)
library(NHANES)
library(broom)
library(rsample)
library(caret)
library(GGally)
```

Exercise 1

In this problem, we will generate simulated data, and will then use this data to perform variable selection.

(a) Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of the same length. Then generate a response vector Y according to the model:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

where β_0, \dots, β_3 are constants of your choice.

```
x <- rnorm(100)
e <- rnorm(100)
```

```
y <- 9 + 1 * x + 2 * x^2 - 3 * x^3 + e
```

(b) Given the predictors X, X^2, \dots, X^{10} , perform best subset selection in order to choose the best model. What is the best model obtained according to C_p , AIC, BIC and adjusted R^2 ? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained.

```
# Get AIC values
poly_degree <- seq(1, 10)

# Empty vector
aic <- double(length(poly_degree))

for (each_value in seq(along = poly_degree)) {
  k <- poly_degree[each_value]
```

```

# Polynomial Model
aic_out <- lm(y ~ poly(x, k))

# Assign it to vector
aic[each_value] <- AIC(aic_out)
}

aic_values <- cbind(poly_degree,aic)

data_from_linear_model <- data.frame(y,x)

# Generate polynomial regression up to 10
fit <- regsubsets(y ~ poly(x, 10), data = data_from_linear_model, nvmax = 10)
fit_summary <- summary(fit)

# Choose best model
metrics <- data.frame(
  r2 = which.max(fit_summary$adjr2),
  cp = which.min(fit_summary$cp),
  BIC = which.min(fit_summary$bic),
  aic = which.min(aic_values[,2])
)

# Generate data frame with metrics

data_model_selection <-
  data_frame(cp = fit_summary$cp,
             BIC = fit_summary$bic,
             r2 = fit_summary$adjr2) %>%

  # add aic values
  cbind(., aic_values[,2]) %>%
  rename(AIC = "aic_values[, 2]") %>%

  mutate(id = row_number())

data_model_selection %>%

  #Transform to long format
  gather(value_type, value, -id) %>%
  ggplot(aes(id, value, col = value_type)) +
  geom_line() +
  geom_point() +
  ylab('') +
  xlab('Number of Variables Used') +
  facet_wrap(~ value_type, scales = 'free') +
  theme_bw() +
  scale_x_continuous(breaks = 1:10) +
  # Change color
  scale_colour_manual(values = c("#d8b365", "#0072B2", "#5ab4ac",
                                "#56B4E9")) +

  # Edit the legend
  theme(axis.text.y = element_text(size = 14),
        # Legend position and Axis size

```

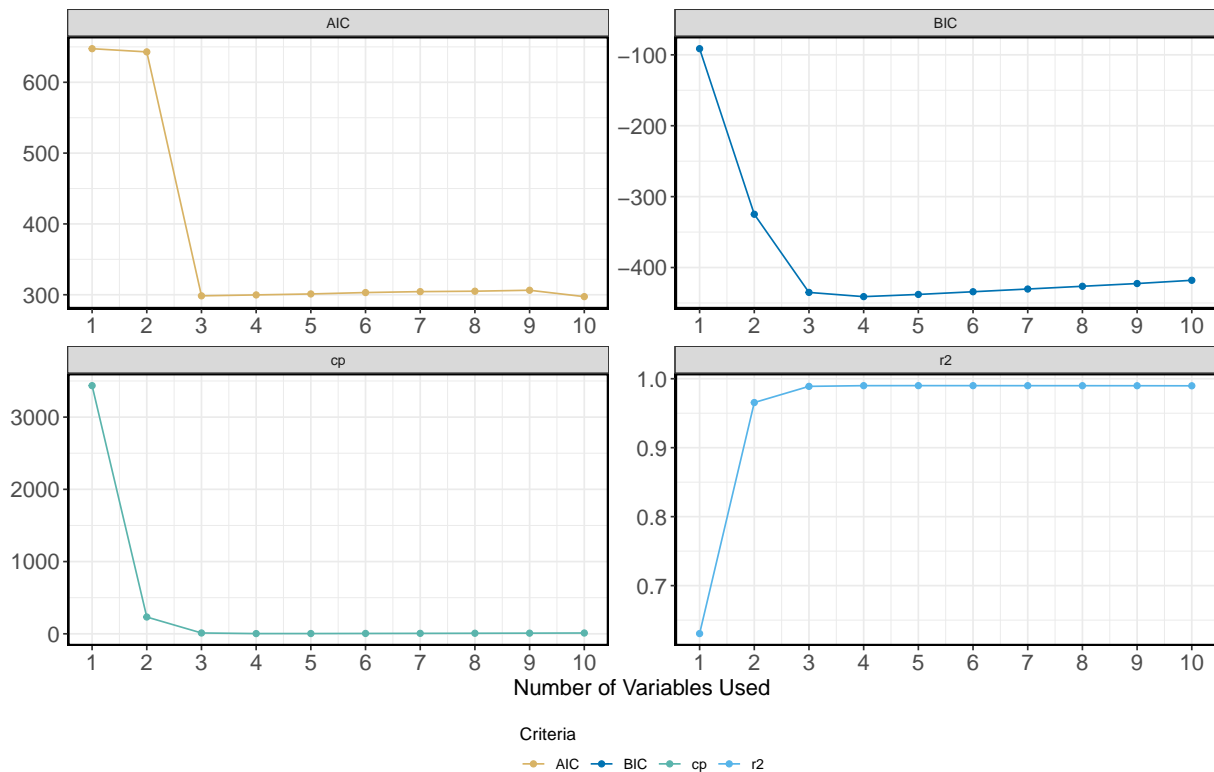
```

legend.position = "bottom",
axis.text.x = element_text(size = 14),
axis.title.y = element_text(size = 14),
axis.title.x = element_text(size = 14),
# Add borders to the plot
panel.border = element_rect(colour = "black", fill= NA,size = 1.3)) +

# Edit legend name
labs(colour = "Criteria") +

#Edit legend
guides(col = guide_legend(override.aes = list(fill=NA),nrow = 1,title.position = "top",))

```



According to the plots, the model with 3 variables is the best

```

best_model <- lm(y ~ poly(x, 3), data = data_from_linear_model)
coef(best_model)

```

```

(Intercept) poly(x, 3)1 poly(x, 3)2 poly(x, 3)3
  11.85479   -78.72119   15.02391  -56.95789

```

(c) Repeat (b), using forward selection. How does your answer compare to the results in (b)?

```

fit_forward <- regsubsets(y ~ poly(x, 10), data = data_from_linear_model,
                          nvmax = 10, method = "forward")
fit_summary_forward <- summary(fit_forward)

```

```

# Choose best model
metrics_forward <- data.frame(

```

```

r2 = which.max(fit_summary_forward$adjr2),
cp = which.min(fit_summary_forward$cp),
BIC = which.min(fit_summary_forward$bic)
)

```

```
metrics %>% gt()
```

r2	cp	BIC	aic
5	4	4	10

```
metrics_forward %>% gt()
```

r2	cp	BIC
5	4	4

In this case, the forward procedure did not produced any different results

(d) Now fit a lasso model and use cross-validation to select the optimal values of λ . Create plots of the cross-validation error as a function f of λ . Report the resulting coefficient estimates, and discuss the results obtained

```
# Generate data
```

```
data_lasso <- data.frame(cbind(y,x,x^2,x^3,x^4,x^5,x^6,x^7,x^8,
                               x^9,x^10)) %>% clean_names()
```

```
x <- model.matrix(y ~. ,data_lasso)[, -1]
y <- data_lasso[,1]
```

```
# split the samples into a training set and a test set
```

```
train <- sample(1:nrow(x),nrow(x) / 2)
test <- (-train)
y_test <- y[test]
```

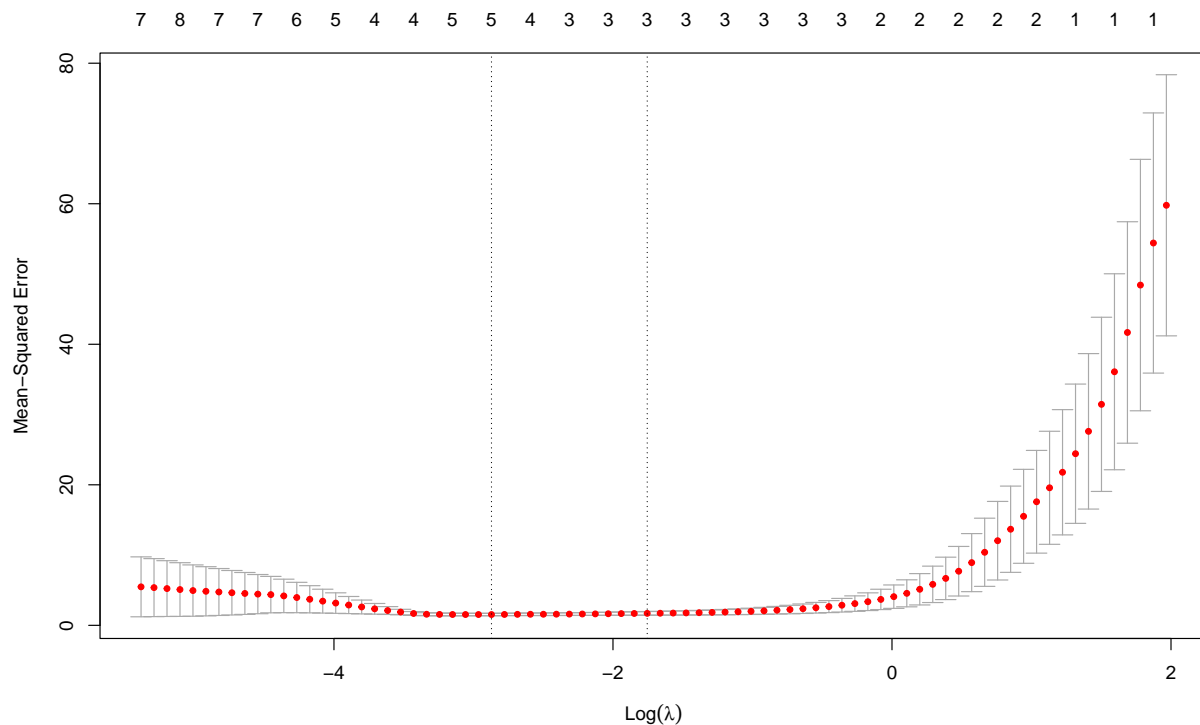
```
#cross-validation to select the optimal values of
cv_out <- cv.glmnet(x[train,] , y[train], alpha = 1)
```

```
best_lamb <- cv_out$lambda.min
```

```
# Perform lasso regression
```

```
lasso_mod <- glmnet(x[train,] , y[train] , alpha = 1)
lasso_pred <- predict(lasso_mod , s = best_lamb ,newx = x[test,])
```

```
# Create plots of the cross-validation error as a function f of
plot(cv_out)
```



```
lasso_coef <- predict(lasso_mod, type = "coefficients", s = best_lambda)[1:11,]
lasso_coef
```

(Intercept)	x	v3	v4	v5
9.0613117129	0.2204723020	2.1321398655	-2.4463124733	0.0000000000
v6	v7	v8	v9	v10
-0.0689742846	0.0000000000	0.0000000000	0.0000000000	0.0000000000
v11				
0.0001331246				

From the lasso model we can see that the variables greater than 0 have a significant effect over the y variable

Exercise 2

```
rm(x,y)
```

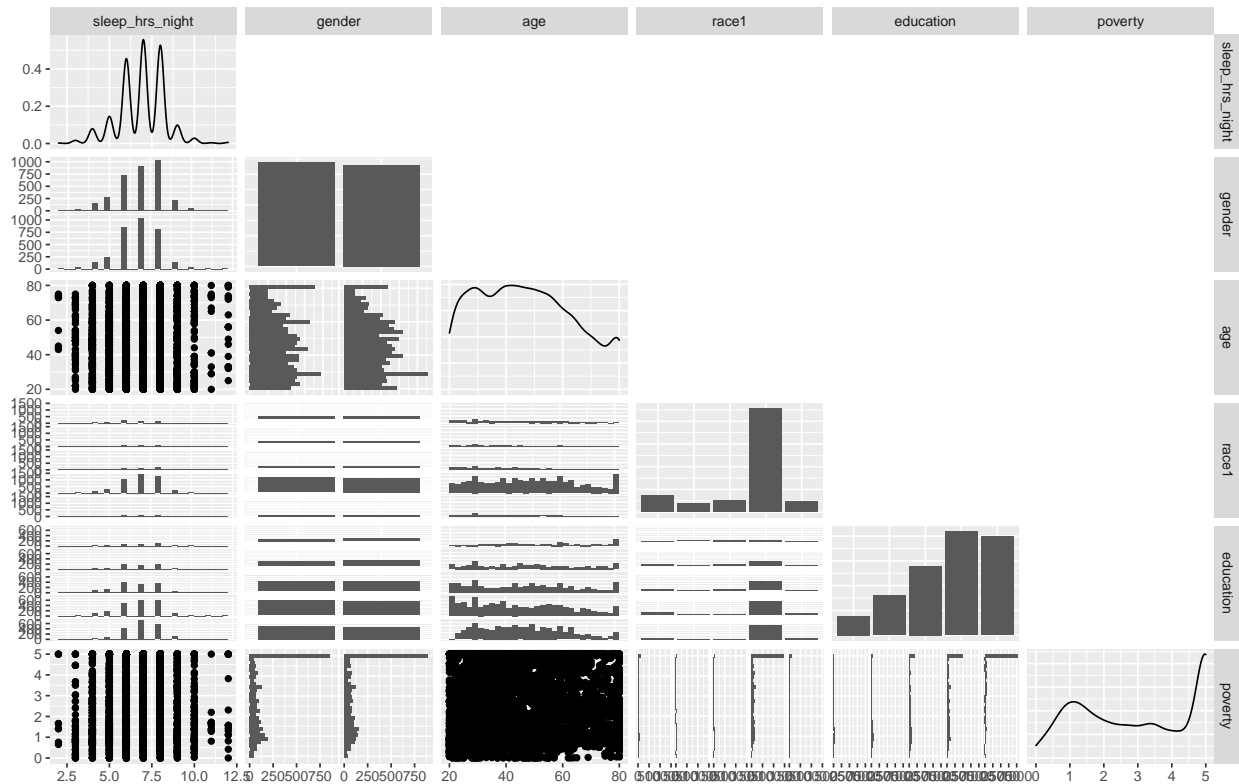
The ability to get a good night's sleep is correlated with many positive health outcomes. Use the NHANES data set from the NHANES package to predict *SleepHrsNight*. Check the R help document for detailed information about the data set.

```
# Load Data
data("NHANES")
data_p2 <-
  NHANES %>%
  clean_names()
```

(b) Select your own predictors, and create plots or summary tables to explore the variables.

```
data_selected_var <- data_p2 %>%
  dplyr::select(sleep_hrs_night,gender,age,race1,education,poverty) %>%
  na.omit()

ggpairs(data_selected_var,upper = "blank")
```



(a) First separate the data set at random into 75% training and 25% testing sets.

```
# Select X-Y for models
x_sleep <- model.matrix(sleep_hrs_night ~ . ,data_selected_var)[, -1]

y_sleep <- data_selected_var$sleep_hrs_night

# Sample data
index <- sample(1:nrow(x_sleep), 0.75 *nrow(x_sleep))

# Create the training data 75%
train_data = x_sleep[index,]
(nrow(train_data)/6671)*100

[1] 74.99625

# Create the test data 25%
test_data = x_sleep[-index,]
(nrow(test_data)/6671)*100

[1] 25.00375
```

(c) Build the following models using the training set with your predictors of choice:

- Multiple linear regression

```
m1 <- lm(y_sleep[train] ~ x_sleep[train,])
```

- Ridge regression

```
# Model
m2_ridge <- glmnet(x_sleep[train, ], y_sleep[train] ,alpha = 0)

# Get best lambda
lambda_sleep_ridge <- cv.glmnet(x_sleep[train,], y_sleep[train], alpha = 0)

best_lamb_ridge <- lambda_sleep_ridge$lambda.min
```

- LASSO regression

```
# Model
m3_lasso <- glmnet(x_sleep[train, ], y_sleep[train] ,alpha = 1)

# Get best lambda
lambda_sleep_lasso <- cv.glmnet(x_sleep[train,], y_sleep[train], alpha = 1)

best_lamb_lasso <- lambda_sleep_lasso$lambda.min
```

(d) Compare the effectiveness of each model on training vs. testing data.

I found that the MSE were generally low, indicating the effectiveness of each model on training vs testing data

- Linear regression MSE

```
linear_prediction <- predict(m1,newx = x_sleep[test_data, ])

y_sleep_test <- y_sleep[test_data]
mean((linear_prediction - y_sleep_test)^2)
```

```
[1] 5.951313
```

- Ridge MSE

```
ridge_pred <- predict(m2_ridge , s = best_lamb_ridge, newx = x_sleep[test_data,])
mean((ridge_pred - y_test)^2)
```

```
[1] 161.8858
```

- Lasso MSE

```
lasso_pred <- predict(m3_lasso , s = best_lamb_lasso, newx = x_sleep[test_data,])
mean((lasso_pred - y_test)^2)
```

```
[1] 164.1701
```

(e) Choose one best model and interpret the results. What have you learned about people's sleeping quality?

Based on the MSE I chose the Multiple linear regression model because it has the lowest MSE(~5)

```
model <- lm(sleep_hrs_night ~ ., data = data_selected_var )
```

```
summary(model)
```

```
Call:
lm(formula = sleep_hrs_night ~ ., data = data_selected_var)

Residuals:
    Min       1Q   Median       3Q      Max
-5.1653 -0.8528  0.0466  0.9925  5.2933

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.459462   0.105150  61.431 < 2e-16 ***
gendermale    -0.172464   0.032651  -5.282 1.32e-07 ***
age            0.002964   0.001010   2.934 0.00336 **
race1Hispanic  0.211124   0.085156   2.479 0.01319 *
race1Mexican   0.413867   0.078782   5.253 1.54e-07 ***
race1White     0.331176   0.054131   6.118 1.00e-09 ***
race1Other     0.214679   0.078972   2.718 0.00658 **
education9 - 11th Grade -0.102837  0.084828  -1.212 0.22544
educationHigh School -0.085695  0.080851  -1.060 0.28922
educationSome College  0.007952  0.080063   0.099 0.92089
educationCollege Grad  0.108830  0.083642   1.301 0.19326
poverty        0.028880  0.011520   2.507 0.01220 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.326 on 6659 degrees of freedom
Multiple R-squared:  0.02086,    Adjusted R-squared:  0.01924
F-statistic: 12.9 on 11 and 6659 DF,  p-value: < 2.2e-16
```

```
anova(model)
```

Analysis of Variance Table

```
Response: sleep_hrs_night
      Df Sum Sq Mean Sq F value    Pr(>F)
gender    1    48.4  48.402 27.5318 1.593e-07 ***
age        1    25.0  24.988 14.2135 0.0001646 ***
race1      4   103.0  25.746 14.6450 6.497e-12 ***
education  4    62.0  15.493  8.8127 4.301e-07 ***
poverty    1    11.0  11.048  6.2843 0.0122046 *
Residuals 6659 11706.7    1.758
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the variables selected, all had an effect over the amount of sleeping hours. For example it seems non-white people sleeps less when compared to white people.

```
ggplot(data = data_selected_var, aes(x = age, y = race1, fill = race1))+
  geom_boxplot() + theme_bw()
```