BIOSTAT 705 - Homework 2

Austin Allen

February 20, 2024

Problem 1

Consider data on corn yield Y (bushels/acre) and rainfall X (inches/yr) in six Midwestern states recorded from 1890 to 1927. Data are given in corn yield and rainfall.csv.

```r
# Load data
rainfall <- read.csv("corn_yield_and_rainfall.csv")
# Fit models
simple_model <- lm(Yield ~ Rainfall, data = rainfall)
rainfall_model_a <- lm(Yield ~ Rainfall + I(Rainfall^2), data = rainfall)
# Print summaries
summary(simple_model)
```

a) Fit a multiple regression model with rainfall and rainfall^2 as predictors in the model. Assess model fit with diagnostic Plots of residuals vs. fitted values and normal Q-Q plot of the standardized residuals. Is this model an improvement over the simple regression model? why?

```
## 
## Call:
## lm(formula = Yield ~ Rainfall, data = rainfall)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10.2014  -2.3530  -0.2577   3.8929   5.7515 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  23.5521     3.2365   7.277 1.43e-08 ***
## Rainfall      0.7755     0.2939   2.639   0.0122 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.049 on 36 degrees of freedom
## Multiple R-squared:  0.1621, Adjusted R-squared:  0.1388 
## F-statistic: 6.965 on 1 and 36 DF,  p-value: 0.01221
```

```r
summary(rainfall_model_a)
```

```
## 
## Call:
## lm(formula = Yield ~ Rainfall + I(Rainfall^2), data = rainfall)
## 
```

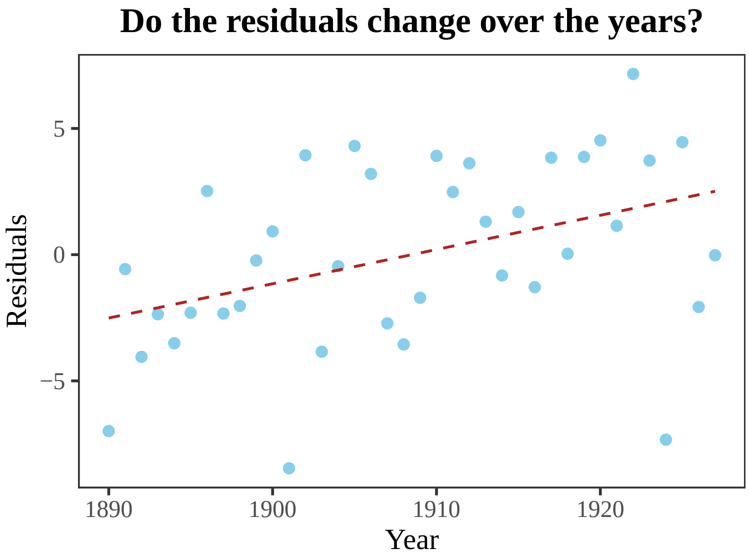
```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4642 -2.3236 -0.1265  3.5151  7.1597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.01467   11.44158  -0.438  0.66387
## Rainfall       6.00428    2.03895   2.945  0.00571 **
## I(Rainfall^2) -0.22936    0.08864  -2.588  0.01397 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.763 on 35 degrees of freedom
## Multiple R-squared:  0.2967, Adjusted R-squared:  0.2565
## F-statistic: 7.382 on 2 and 35 DF,  p-value: 0.002115
```

Thoughts:

- The QQ plot looks slightly better in the quadratic model
- Mostly, the plots all look the same
-

```
# Create plot of Year vs Residuals
ggplot(data.frame(residuals = rainfall_model_a$residuals, year = rainfall$Year), aes(x = year, y = resi
  geom_point(col = "skyblue") +
  geom_smooth(method = "lm", se = FALSE, col = "firebrick", lty = 2, lwd = .5) +
  labs(title = "Do the residuals change over the years?", x = "Year", y = "Residuals") +
  theme_bw() +
  theme(text = element_text(family = "serif"),
        panel.grid = element_blank(),
        plot.title = element_text(hjust = .5, face = "bold"))
```

b) Plot the residuals from the above model vs. year. Is there a pattern in this plot? ie, yield increases with years after adjusting for rainfall.



```r
# Fit model
rainfall_model_c <- lm(Yield  ~ Rainfall + I(Rainfall^2) + Year, data = rainfall)
summary(rainfall_model_c)
```

c) If there is a pattern, then fit a multiple regression model with rainfall, rainfall^2 and year as predictors. What is the interpretation of estimated coeficient for year? Is this model better than the model in part a? Why?

```
##
## Call:
## lm(formula = Yield ~ Rainfall + I(Rainfall^2) + Year, data = rainfall)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3995 -1.8086 -0.0479  2.4050  5.1839
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -263.30324   98.24094  -2.680  0.01126 *
## Rainfall         5.67038    1.88824   3.003  0.00499 **
## I(Rainfall^2)   -0.21550    0.08207  -2.626  0.01286 *
## Year             0.13634    0.05156   2.644  0.01229 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.477 on 34 degrees of freedom
## Multiple R-squared:  0.4167, Adjusted R-squared:  0.3652
## F-statistic: 8.095 on 3 and 34 DF,  p-value: 0.0003339
```

d) Is there a multicollinearity issue in part (c)? if yes, which type?

> Yes, there's a structural multicollinearity because Rainfall^2 is obviously correlated with Rainfall.

e) Examine the data in part c for any influential cases and check for multicollinearity.

f) Refit the model in part c by adding an interaction term rainfallyear. Does the interaction exist? How would you interpret this interaction term and the main effects coeficient?

Problem 2

Data on Major League Baseball from the 1986 and 1987 seasons are given in (Hitters.csv).

Description

- Major League Baseball Data from the 1986 and 1987 seasons.

Format

- A data frame with 322 observations of major league players on the following 19 variables.
  - AtBat: Number of times at bat in 1986, Hits: Number of hits in 1986
  - HmRun: Number of home runs in 1986, Runs: Number of runs in 1986
  - RBI: Number of runs batted in in 1986, Walks: Number of walks in 1986
  - Years: Number of years in the major leagues, CAtBat: Number of times at bat during his career
  - CHits: Number of hits during his career, CHmRun: Number of home runs during his career
  - CRuns: Number of runs during his career, CRBI: Number of runs batted in during his career

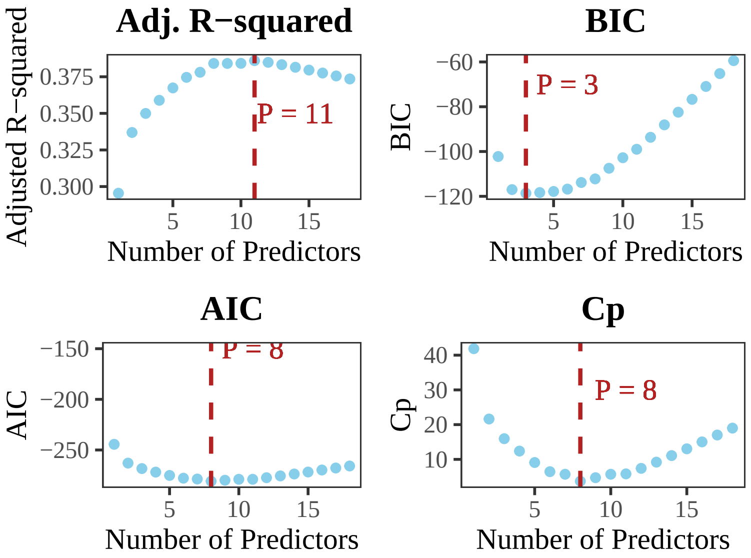
  - CWalks: Number of walks during his career, League: A factor with levels A and N indicating player's league at the end of 1986
  - Division: A factor with levels E and W indicating player's division at the end of 1986
  - PutOuts: Number of put outs in 1986
  - Assists: Number of assists in 1986
  - Errors: Number of errors in 1986
  - Salary (outcome): 1987 annual salary on opening day in thousands of dollars,

Note, there are 59 players with missing salary. Impute missing salary by the mean of non-missing salaries. Also, due to large difference in salaries, would be more appropriate to model log(Salary) as an outcome.

```
# Load in the data
hitters <- read.csv("Hitters.csv")

# Format salary, league, and division
hitters_clean <- hitters %>%
  mutate(Salary_imputed = ifelse(is.na(Salary), mean(hitters$Salary, na.rm = TRUE), Salary),
         log_salary = log(Salary_imputed),
         League_A = ifelse(League == "A", 1,0),
         Division_E = ifelse(Division == "E", 1,0)) %>%
  select(log_salary, !c(League, Division, Salary, Salary_imputed))
```

```
# Create full model
model_full_a <- regsubsets(log_salary~.,data=hitters_clean,nvmax =18)
summary_model_a <- summary(model_full_a)
n <- nrow(hitters_clean)

# Add AIC
summary_model_a$aic <- (n*log((summary_model_a$rss/n))) + (2*c(1:18))

# Create a dataframe to plot values
model_df <- data.frame(adjr2 = summary_model_a$adjr2, bic = summary_model_a$bic, aic = summary_model_a$a

par(mfrow =c(2,2))

# Adjusted R-squared plot
p1 <- ggplot(model_df, aes(x = n_predictors, y = adjr2)) +
  geom_point(col = "skyblue", size = 1.25) +
  labs(x = "Number of Predictors", y = "Adjusted R-squared", title = "Adj. R-squared") +
  geom_vline(xintercept = which.max(summary_model_a$adjr2), lty = 2, lwd = .75, col = "firebrick") +
  geom_text(aes(x = 14, y = .35), label = "P = 11", family = "serif", col = "firebrick") +
  theme_bw()+
  theme(text = element_text(family = "serif"),
      panel.grid = element_blank(),
      plot.title = element_text(hjust = .5, face= "bold"))

# BIC plot
p2 <- ggplot(model_df, aes(x = n_predictors, y = bic)) +
  geom_point(col = "skyblue", size = 1.25) +
  labs(x = "Number of Predictors", y = "BIC", title = "BIC") +
  geom_vline(xintercept = which.min(summary_model_a$bic), lty = 2, lwd = .75, col = "firebrick") +
  geom_text(aes(x = 6, y = -70), label = "P = 3", family = "serif", col = "firebrick") +
  theme_bw()+
  theme(text = element_text(family = "serif"),
      panel.grid = element_blank(),
      plot.title = element_text(hjust = .5, face= "bold"))

# AIC plot
p3 <- ggplot(model_df, aes(x = n_predictors, y = aic)) +
  geom_point(col = "skyblue", size = 1.25) +
  labs(x = "Number of Predictors", y = "AIC", title = "AIC") +
  geom_vline(xintercept = which.min(summary_model_a$aic), lty = 2, lwd = .75, col = "firebrick") +
  geom_text(aes(x = 11, y = -150), label = "P = 8", family = "serif", col = "firebrick") +
  theme_bw()+
  theme(text = element_text(family = "serif"),
      panel.grid = element_blank(),
      plot.title = element_text(hjust = .5, face= "bold"))

# Cp plot
p4 <- ggplot(model_df, aes(x = n_predictors, y = cp)) +
  geom_point(col = "skyblue", size = 1.25) +
  labs(x = "Number of Predictors", y = "Cp", title = "Cp") +
  geom_vline(xintercept = which.min(summary_model_a$cp), lty = 2, lwd = .75, col = "firebrick") +
  geom_text(aes(x = 11, y = 30), label = "P = 8", family = "serif", col = "firebrick") +
  theme_bw()+
  theme(text = element_text(family = "serif"),
      panel.grid = element_blank(),
      plot.title = element_text(hjust = .5, face= "bold"))

# Arrange plots neatly
gridExtra::grid.arrange(p1,p2,p3,p4)
```

a) Indicate which subset of predictor variables you would recommend as a 'best' for predicting player salary by examining plots of the following criteria (vs p number of model predictors): 1) adjusted R^2 2) C_p, 3) AIC$_p$ and 4) BIC$_p$.



This is interesting. Let's pull out the variables used at each point in the function.

```r
# Create matrix with included variables
model_variables <- summary_model_a$which

# Adjusted R-squared: p = 11:
adjr2_variables <- which(model_variables[11,])

# Print variables
print("Variables included in best model for adjusted R-squared:")
```

```
## [1] "Variables included in best model for adjusted R-squared:"
```

```
##  [1] "AtBat"      "Hits"       "HmRun"      "Walks"      "Years"
##  [6] "CRuns"      "CWalks"     "PutOuts"    "Assists"    "Errors"
## [11] "Division_E"
```

When looking only at adjusted R^2, these predictors would be included in the model that is considered "best."

```r
# Cp: p = 8:
cp_variables <- which(model_variables[8,])

# Print variables
print("Variables included in best model for Cp:")
```

```
## [1] "Variables included in best model for Cp:"
```

```
## [1] "AtBat"      "Hits"       "Walks"      "Years"      "CRuns"
## [6] "CWalks"     "PutOuts"    "Division_E"
```

When looking only at C_p, these predictors would be included in the model that is considered "best."

```r
# BIC: p = 3:
bic_variables <- which(model_variables[3,])

# Print variables
print("Variables included in best model for BIC:")
```

```
## [1] "Variables included in best model for BIC:"
```

```
## [1] "Runs"       "CHits"      "Division_E"
```

When looking only at BIC, these predictors would be included in the model that is considered "best."

```r
# AIC: p = 8:
aic_variables <- which(model_variables[8,])

# Print variables
print("Variables included in best model for AIC:")
```

```
## [1] "Variables included in best model for AIC:"
```

```
## [1] "AtBat"      "Hits"       "Walks"      "Years"      "CRuns"
## [6] "CWalks"     "PutOuts"    "Division_E"
```

When looking only at BIC, these predictors would be included in the model that is considered "best." These are the same as those included in the model chosen for C_p.

b) Do the four criteria in part (a) identify the same 'best' subset? What is the total subset models, one would expect?

No, they don't agree. We're going to have to use our thinking caps to use this information to determine what the "best" means for us.

When talking about how many subset models there are in our model, there's a simple formula for calculating how many subsets there are:

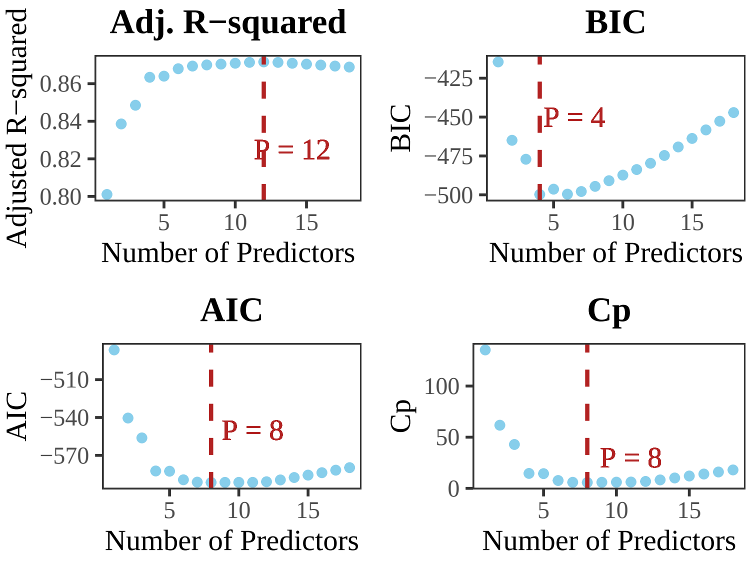
$$\text{Total Subsets} = 2^p - 1$$

$>$ In our case, there are 18 predictors in the model, which means that there are $2^18 - 1$ different models, which equates to 262,143 different possible models given our data. That's a ton.

c) Repeat parts a and b above using data without imputing the salary. How do these differ on selecting a 'best' subset?

I'm going to remove an NA's from the salary column.

```r
# Removing NA's
hitters_rm.na <- hitters %>% filter(!is.na(Salary)) %>%
  mutate(log_salary = log(Salary))
# Create full model
model_full_c <- regsubsets(log_salary~.,data=hitters_rm.na,nvmax =18)
summary_model_c <- summary(model_full_c)
n <- nrow(hitters_rm.na)

# Add AIC
summary_model_c$aic <- (n*log((summary_model_c$rss/n))) + (2*c(1:18))

# Create a dataframe to plot values
model_df <- data.frame(adjr2 = summary_model_c$adjr2, bic = summary_model_c$bic, aic = summary_model_c$
# Plot code omitted for brevity
```



d) Instead of imputing missing salary using mean, use multiple-imputation (MI) method in R via "mice" package with m=25 number of imputations, method="pmm" (Predictive mean matching) and maxit=20. Repeat parts a and b with MI data.

e) Now, you analyzed the Hitters dataset using i) complete case data, ie no imputation; ii) used mean as a single imputation method; and iii) using MI method, which one would you recommend? Why?

f) Perform forward, backward and stepwise procedure to identify the 'best' subset of regression model, using an entry to the model level of significance $\alpha = 0.05$ and remained in the model at $\alpha = 0.05$. Summarize your results. Use MI dataset for this part as well as for the remaining parts below.

g) Perform ridge and LASSO shrinkage procedures to identify the 'best' regression model. For each shrinkage procedure, show plots of estimated coeficients vs. $\log(\lambda)$ (where λ is a tuning parameter) and deviance as well as cross-validation (cv) plots. What is 'best' estimated λ from the cv method? How the estimated regression model coeficients in LASSO differ for those estimated by ridge regression? Provide Which shrinkage procedure would you recommend? why?

h) Perform cv by splitting the dataset into 2/3 as training and 1/3 as validation set, ie 215 vs. 107. How the estimated "best" λ compares the one produced in part g by LASSO? Is the model selected by cv differ from the LASSO in part d? If yes, which one you recommend? Note: for parts g and h, you need to download "glmnet" in R.