MRA - Assignment 2

Q3. Model Selection

```
library(tidyverse)
library(dplyr)
library(olsrr)
```

```
crime_data = read_csv("Crimes.csv", show_col_types = FALSE)
crime_data
```

```
# A tibble: 51 x 4
      VR    MR     M     P
   <dbl> <dbl> <dbl> <dbl>
 1   761   9    41.8   9.1
 2   780  11.6  67.4  17.4
 3   593  10.2  44.7  20
 4   715   8.6  84.7  15.4
 5  1078  13.1  96.7  18.2
 6   567   5.8  81.8   9.9
 7   456   6.3  95.7   8.5
 8   686   5    82.7  10.2
 9  1206   8.9  93    17.8
10   723  11.4  67.7  13.5
# i 41 more rows
```

a) Use R to fit all possible models and compute AIC, BIC and R^2_{adj} for each model. Report a table with your results.

```r
crime_model = lm(VR ~ ., data = crime_data)
tab = ols_step_all_possible(crime_model)
results = tab$result[,c("predictors","aic","sbc","adjr")]
knitr::kable(results)
```

predictors	aic	sbc	adjr
MR	692.3529	698.1484	0.7809632
M	752.9293	758.7248	0.2816104
P	755.4989	761.2944	0.2444882
MR M	671.5556	679.2829	0.8569988
MR P	694.3304	702.0577	0.7764989
M P	727.1934	734.9207	0.5742764
MR M P	669.7249	679.3840	0.8645242

b) Indicate the best model overall according to each of AIC, BIC and R^2_{adj}.

```r
print(c("AIC",results$predictors[which.min(results$aic)],
        results$aic[which.min(results$aic)]))
```

```
[1] "AIC"              "MR M P"           "669.724867314439"
```

```r
print(c("BIC",results$predictors[which.min(results$sbc)],
        results$sbc[which.min(results$sbc)]))
```

```
[1] "BIC"              "MR M"             "679.282922911935"
```

```r
print(c("$R2_Adj",results$predictors[which.max(results$adjr)],
        results$adjr[which.max(results$adjr)]))
```

```
[1] "$R2_Adj"           "MR M P"            "0.864524166055133"
```

c) Implement a forward stepwise regression that uses BIC. You will start by fitting the null model (the model with no covariates) and computing its BIC. Then, consider all possible one covariate models and compute their BICs. Iterate until there is no improvement in your criteria.

```r
combinations = list(c("1"),
                    c("P","M","MR"),
                    c("M + P","MR + P","MR + M"),
                    c("MR + M + P"))
min = 100000000
min_combination = ""

for(i in 1:length(combinations)){
  flag = FALSE
  for(j in 1:length(combinations[[i]])){
    f = as.formula(paste("VR ~ ",paste(combinations[[i]][j])))
    crime_model = lm(f,data = crime_data)
    lm = sum(log(dnorm(crime_data$VR,fitted.values(crime_model),
                       sd = summary(crime_model)$sigma)))
    bic = -2 * lm + (i+1) * log(nrow(crime_data))
    if(bic < min){
      min = bic
      min_combination = combinations[[i]][j]
      flag = TRUE
    }
  }
  if(flag == FALSE){
    break
  }
}

print(c("Best Model" = min_combination, "Minimum BIC" = min))
```

```
        Best Model        Minimum BIC
          "MR + M" "679.374778624573"
```

d) Does the forward stepwise method find the best possible subset? Compare the solutions to item (a) and (c). Explain why solutions from stepwise regression might differ from the all possible regressions method in (a).

By looking at the BIC values, the best possible subset found using the all possible regressions was MR + M with a BIC value of 679.282922. The forward selection algorithm using the BIC found the same subset MR + M with a BIC value of 679.3748.

Forward and backward selection algorithms are greedy algorithms, meaning they explore a sequence of local improvements and stop when no further improvement is found. However, this greedy nature can prevent them from finding the global optimum.

For example, suppose there are four explanatory variables. The forward selection algorithm might identify a model with two variables as optimal because adding a third variable does not improve the model. As a result, it stops searching and does not evaluate the model with all four variables, which could potentially be the global optimum.

While these methods are efficient for large datasets with many explanatory variables—where evaluating all possible subsets is computationally infeasible—they may miss the global optimum. In this particular case, the forward selection algorithm did identify the global optimum subset but this cannot be guaranteed in every scenario.

e) Explain how you could implement a backward selection algorithm using the F test as a decision rule. You can assume that the confidence level is $\alpha = 0.05$.

A backward selection algorithm using F-test as a decision rule can be implemented the same way we implement with BIC as the decision rule. Start with the complete model with all the explanatory variables and do the F-test with $\alpha = 0.05$. This is our base.

...write more

Q4) Multiple regression

```
football = read_csv("football.csv", show_col_types = FALSE)
football
```

```
# A tibble: 28 x 10
       y    x1    x2    x3    x4    x5    x6    x7    x8    x9
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1    10  2113  1985  38.9  64.7     4   868  59.7  2205  1917
 2    11  2003  2855  38.8  61.3     3   615  55    2096  1575
 3    11  2957  1737  40.1  60      14   914  65.6  1847  2175
 4    13  2285  2905  41.6  45.3    -4   957  61.4  1903  2476
 5    10  2971  1666  39.2  53.8    15   836  66.1  1457  1866
 6    11  2309  2927  39.7  74.1     8   786  61    1848  2339
 7    10  2528  2341  38.1  65.4    12   754  66.1  1564  2092
 8    11  2147  2737  37    78.3    -1   761  58    1821  1909
 9     4  1689  1414  42.1  47.6    -3   714  57    2577  2001
10     2  2566  1838  42.3  54.2    -1   797  58.9  2476  2254
# i 18 more rows
```

a) Fit a linear regression model relating the number of games won (y) to the team's passing yardage ($x2$), the percentage of rushing plays ($x7$), and the opponents' yards rushing ($x8$).

```
football_model = lm(y ~ x2 + x7 + x8,data=football)
summary(football_model)
```

```
Call:
lm(formula = y ~ x2 + x7 + x8, data = football)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0370 -0.7129 -0.2043  1.1101  3.7049 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.808372   7.900859  -0.229 0.820899    
x2           0.003598   0.000695   5.177 2.66e-05 ***
x7           0.193960   0.088233   2.198 0.037815 *  
x8          -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.706 on 24 degrees of freedom
Multiple R-squared:  0.7863,    Adjusted R-squared:  0.7596 
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

b) Compute the sum of squares.

```
y_bar = mean(football$y)

sst = sum((football$y - y_bar)^2)
sse = sum((football$y - fitted.values(football_model))^2)
ssr = sum((fitted.values(football_model) - y_bar)^2)

cat("sse = ",sse," , ","ssr = ",ssr," , ","sst = ",sst)
```

```
sse =  69.87  ,  ssr =  257.0943  ,  sst =  326.9643
```

The relationship between SS_t, SS_e, and SS_r:

$SS_t = SS_e + SS_r$.

```
sse + ssr
```

```
[1] 326.9643
```

```
p = 3
n = nrow(football)
df_sse = n - p - 1
df_ssr = p
df_sst = df_sse + df_ssr

cat("DF of sse = ",df_sse," , ",
    "DF of ssr = ",df_ssr," , ",
    "DF of sst = ",df_sst)
```

```
DF of sse =  24  ,  DF of ssr =  3  ,  DF of sst =  27
```

```
mst = sst / df_sst
mse = sse / df_sse
msr = ssr / df_ssr
cat("MSE = ",mse," , ","MSR = ",msr," , ","MST = ",mst)
```

```
MSE =  2.91125  ,  MSR =  85.69809  ,  MST =  12.10979
```

c) Calculate the t statistics for testing the hypothesis $H0: \beta_j = 0$ versus $H1: \beta_j \neq 0$ for $j = 1,2,3$ where β_1, β_2, β_3 are the coefficients of x_2, x_7 and x_8.

```
X = model.matrix(football_model)
beta = solve(t(X)%*%X)%*%t(X)%*%football$y
sigma_2 = sse / (n - p - 1)
sigma = sigma_2 * solve(t(X)%*%X)

t_b1 = beta[2,1] / sqrt(sigma[2,2])
t_b2 = beta[3,1] / sqrt(sigma[3,3])
t_b3 = beta[4,1] / sqrt(sigma[4,4])

print(c("T_" = t_b1, "T_" = t_b2, "T_" = t_b3))
```

```
    T_.x2     T_.x7     T_.x8
 5.177090  2.198262 -3.771036
```

d) Calculate R^2 and R^2_{adj} using (b).

```
r2 = 1 - (sse/sst)
r2_adj = 1 - (mse/mst)

print(c(r2 = r2,r2_adj = r2_adj))
```

```
       r2    r2_adj
0.7863069 0.7595953
```

e) Use item (b) to test the significance of the regression (F test). Outline the hypothesis of this test, compute the test statistic and conclude using the critical regions approach.

```
f_stat = msr/mse
c(f_stat = f_stat,"95% CI F distribution" = qf(0.95,p, n - p - 1))
```

```
               f_stat 95% CI F distribution
            29.436870              3.008787
```

The F-statistic we have calculated is 29.4368. The 95% value of the F distribution for df = 3 and df = 24 is 3.008787. Since, the F is greater than the 0.95 quantile, we reject $H0$.

f) Show numerically that R^2 is equal to the square of the correlation coefficient between Y_i and $\hat{Y}_i$.

```
c("Square of correlation coefficent" =
    cor(football$y, fitted.values(football_model))^2, "R2" = r2)
```

```
Square of correlation coefficent                                   R2
                       0.7863069                            0.7863069
```

We see that the Square of the correlation coefficient is equal to the R^2 value.

g) Find a 95% CI on the mean number of games won by a team when x_2 = 2300, x_7= 56, x_8 = 2100.

```r
X_Star = matrix(c(1,2300,56,2100),ncol=1)

Y_Star_mean = t(X_Star)%*%matrix(beta,ncol=1)

Y_Star_mean_upper = Y_Star_mean + qt(0.95,n-p-1) *
  sqrt(sigma_2 * t(X_Star)%*%solve(t(X)%*%(X))%*%X_Star)
Y_Star_mean_lower = Y_Star_mean - qt(0.95,n-p-1) *
  sqrt(sigma_2 * t(X_Star)%*%solve(t(X)%*%(X))%*%X_Star)

c(Lower = Y_Star_mean_lower,Upper = Y_Star_mean_upper)
```

```
   Lower    Upper
6.569655 7.863193
```

```r
Y_Star_mean_upper - Y_Star_mean_lower
```

```
         [,1]
[1,] 1.293539
```

The length of confidence interval is 1.29353.

h) Fit the model using x_7 and x_8 only and compute its error sums of squares.

```r
football_model_2 = lm(y ~ x7 + x8,data=football)
summary(football_model_2)
```

```
Call:
lm(formula = y ~ x7 + x8, data = football)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7985 -1.5166 -0.5792  1.9927  4.5248

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.944319   9.862484   1.819  0.08084 .
x7           0.048371   0.119219   0.406  0.68839
x8          -0.006537   0.001758  -3.719  0.00102 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.432 on 25 degrees of freedom
Multiple R-squared:  0.5477,	Adjusted R-squared:  0.5115
F-statistic: 15.13 on 2 and 25 DF,  p-value: 4.935e-05
```

```r
y_bar_2 = mean(football$y)

sst_2 = sum((football$y - y_bar)^2)
sse_2 = sum((football$y - fitted.values(football_model_2))^2)
ssr_2 = sum((fitted.values(football_model_2) - y_bar)^2)

cat("sse = ",sse_2," , ","ssr = ",ssr_2," , ","sst = ",sst_2)
```

```
sse =  147.8981  ,  ssr =  179.0662  ,  sst =  326.9643
```

```r
p_2 = 2
n = nrow(football)
df_sse_2 = n - p_2 - 1
df_ssr_2 = p_2
df_sst_2 = df_sse_2 + df_ssr_2

cat("DF of sse = ",df_sse_2," , ",
    "DF of ssr = ",df_ssr_2," , ",
    "DF of sst = ",df_sst_2)
```

```
DF of sse =  25  ,  DF of ssr =  2  ,  DF of sst =  27
```

```r
mst_2 = sst_2 / df_sst_2
mse_2 = sse_2 / df_sse_2
msr_2 = ssr_2 / df_ssr_2
cat("MSE = ",mse_2," , ","MSR = ",msr_2," , ","MST = ",mst_2)
```

```
MSE =  5.915924  ,  MSR =  89.53309  ,  MST =  12.10979
```

i) Perform an F test for hypothesis $H0 : \beta_2 = \beta_3 = 0$ versus $H1 :$ at least one of β_2 or β_3 is different than zero.

```r
f_stat_2 = msr_2/mse_2
c(f_stat = f_stat_2,"95% CI F distribution" = qf(0.95,p_2,n -p_2 -1))
```

```
               f_stat 95% CI F distribution 
             15.13425               3.38519 
```

The F-statistic we have calculated is 29.4368. The 95% value of the F distribution for df = 2 and df = 25 is 3.008787. Since, the F is greater than the 0.95 quantile, we reject $H0$.

j) Recompute R^2 and R^2_{adj} for the new model. How do these quantities compute to those in (d) ?

```r
r2_2 = 1 - (sse_2/sst_2)
r2_adj_2 = 1 - (mse_2/mst_2)

print(c(r2 = r2_2,r2_adj = r2_adj_2))
```

```
       r2    r2_adj 
0.5476628 0.5114759 
```

The R^2 and R^2_{adj} value are lower than that of the previous model.

k) Recompute the 95% confidence interval on the mean number of games won by a team with the new model, using x_7 = 56, x_8 = 2100. Compare the length of the interval to (g).

```r
X = model.matrix(football_model_2)

beta = solve(t(X)%*%X)%*%t(X)%*%football$y
sigma_2 = sse / (n - p_2 - 1)
sigma = sigma_2 * solve(t(X)%*%X)

X_Star = matrix(c(1,56,2100),ncol=1)
Y_Star_mean = t(X_Star)%*%matrix(beta,ncol=1)

Y_Star_mean_upper = Y_Star_mean + qt(0.95,n - p_2 -1) *
  sqrt(sigma_2 * t(X_Star)%*%solve(t(X)%*%(X))%*%X_Star)
Y_Star_mean_lower = Y_Star_mean - qt(0.95,n - p_2 - 1) *
  sqrt(sigma_2 * t(X_Star)%*%solve(t(X)%*%(X))%*%X_Star)
```

```r
c(lower = Y_Star_mean_lower,upper = Y_Star_mean_upper)
```

```
   lower    upper 
6.300549 7.551936 
```

```r
Y_Star_mean_upper - Y_Star_mean_lower
```

```
         [,1]
[1,] 1.251387
```

The length of the confidence interval is 1.25138. It is slightly lower than that of the previous model.

l) Comment on how removing x_2 from the model changed the model adequacy and its predictions.

On removing x_2, the model started performing worse. We can very clearly see from the SS_e value and R^2 values. The initial model had a lesser SS_e and a higher R^2 value which signifies a better model.

Q5) Types of variables

```r
bike_sharing = read.csv("bikesharing.csv")
```

a) Explain how to code the month of the year mnth using indicator variables.

```r
bike_sharing$mnth = as.factor(bike_sharing$mnth)
```

We use the as.factor() function to make the mnth variable as an indicator variable. January is reference category by default. We can print the levels for a factor with the levels() function. We can then pass this variable to the lm() function and it will automatically recognize it as an indicator variable.

```r
levels(bike_sharing$mnth)
```

```
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"
```

b) Using R, fit a linear regression model to cnt using hum: the normalised measure of humidity, windspeed: the normalised wind speed on the day, temp: the normalised temperature, and mnth. Make sure to use mnth as a categorical variable using January as the reference category. Include the summary of the fitted model.

```r
bike_model = lm(cnt ~ hum + windspeed + temp + mnth, data = bike_sharing)
summary(bike_model)
```

```
Call:
lm(formula = cnt ~ hum + windspeed + temp + mnth, data = bike_sharing)

Residuals:
    Min      1Q  Median      3Q     Max 
-5345.2  -997.8  -162.3  1115.5  3411.3 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3914.561    352.443  11.107  < 2e-16 ***
hum         -4205.372    379.303 -11.087  < 2e-16 ***
windspeed   -4807.145    674.576  -7.126 2.52e-12 ***
temp         7262.317    673.126  10.789  < 2e-16 ***
mnth2          -9.116    247.020  -0.037 0.970572    
mnth3         486.786    259.701   1.874 0.061281 .  
mnth4         757.273    287.355   2.635 0.008588 ** 
mnth5         892.579    336.832   2.650 0.008229 ** 
mnth6         202.492    385.904   0.525 0.599940    
mnth7        -524.788    422.957  -1.241 0.215101    
mnth8         117.049    395.509   0.296 0.767357    
mnth9        1178.208    348.989   3.376 0.000775 ***
mnth10       1522.113    290.321   5.243 2.08e-07 ***
mnth11       1162.629    255.896   4.543 6.50e-06 ***
mnth12        785.963    246.182   3.193 0.001472 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1325 on 716 degrees of freedom
Multiple R-squared:  0.5415,	Adjusted R-squared:  0.5325 
F-statistic:  60.4 on 14 and 716 DF,  p-value: < 2.2e-16
```

c) Is there an indication that month of the year is an important variable?

```r
anova(bike_model)
```

```
Analysis of Variance Table

Response: cnt
           Df     Sum Sq    Mean Sq F value    Pr(>F)
hum         1   27757373   27757373  15.822 7.666e-05 ***
windspeed   1  196708994  196708994 112.127 < 2.2e-16 ***
temp        1 1038171824 1038171824 591.776 < 2.2e-16 ***
mnth       11  220794669   20072243  11.441 < 2.2e-16 ***
Residuals 716 1256102532    1754333
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The P value for the mnth variable is < 2.2e-16. A low p-value (typically < 0.05) indicates that the variable is statistically significant and contributes meaningfully to explaining the variance in the response variable.

d) What months of the year have a different average number of bike rentals in comparison to January, given hum, temp and windspeed? Use $\alpha = 0.05$.

By looking at the summary chart of the model, we can see the following.

Months **with** significant difference compared to January, given Hum, Temp, and Windspeed:

April, May, September, October, November, December.

Months **without** significant difference compared to January, given Hum, Temp, and Windspeed

February, March, June, July, August.

e) Given hum = 0.4, temp = 0.3, windspeed = 0.65, what is the average number of rentals in each month? Report a table with your results.

```r
X_Star = matrix(nrow = 0,ncol = 15)
for(i in 1:12){
  x = c(1,0.4,0.3,0.65,0,0,0,0,0,0,0,0,0,0,0)
  if(i!=1){
    x[3+i] = 1
  }
  X_Star = rbind(X_Star,x)
```

```
}


result = X_Star%*%matrix(coef(bike_model))
dimnames(result) = list(c("Jan","Feb","Mar","Apr","May","Jun",
                          "Jul","Aug","Sep","Oct","Nov","Dec"),
                        c("Predicted value"))
knitr::kable(result)
```

	Predicted value
Jan	5510.775
Feb	5501.659
Mar	5997.560
Apr	6268.048
May	6403.354
Jun	5713.266
Jul	4985.987
Aug	5627.824
Sep	6688.983
Oct	7032.888
Nov	6673.404
Dec	6296.737