Module 10 Assignment

JARED YU

1. Use any math/stat software (e.g., www.numbergenerator.org/randomnumbergenerator) of your choice to find a random number generator to randomly select 20 rows of Table B.2. Then do Problem 10.4 (a), (b), (c), (d), (e), page 367 of Textbook, using your generated data.
    a. Use forward selection to specify a subset regression model.

Ans:
The randomly chosen rows for this problem are: 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 14, 17, 18, 19, 21, 22, 23, 25, 26, and 28.

The "stepAIC()" function in R was used for this task. It calculated from an intercept-only model the AIC value, and then proceeded to add a regressor one-at-a-time to the model. The results can be seen below in Table 1. It can be seen that the regressors are added iteratively one-by-one, and the AIC continues to decrease, and so regressors are continually added until the final model is the full model that includes all five original regressors.

Table 1 The table below shows the result of using the stepAIC() function in R to calculate the forward selection model.

Step	Model	AIC
1	$y \sim 1$	131.07
2	$y \sim x_4$	107.72
3	$y \sim x_4 + x_3$	95.75
4	$y \sim x_4 + x_3 + x_2$	95.35
5	$y \sim x_4 + x_3 + x_2 + x_1$	91.52
6	$y \sim x_4 + x_3 + x_2 + x_1 + x_5$	91.16

    b. Use backward elimination to specify a subset regression model.

Ans:
The same "stepAIC()" function was used again in R to calculate the backward elimination model for the data. It starts with the full model and then calculates the sum of squares, residual sum of squares, and the AIC for each case where it potentially removes any of the regressors. The result can be seen below in Table 2 (*Note: This function seems to prefer AIC, and so this metric will be used continuously from here on.*). It can be seen that starting from the full model with all regressors (marked None), it continuously had the lowest AIC (and hence the best) score amongst all other possible models with one regressor removed. Removing x_5 would bring the AIC to a close level, but not low enough to remove it from the current model. Therefore, the backward elimination method has chosen the same model as in part a) (i.e., the full model with all regressors).

Table 2 The below table shows the results of using the stepAIC() function in R to calculate the backward elimination model.

Regressor to remove	Resulting AIC
None	91.161
x_5	91.522
x_2	93.528
x_3	94.677
x_1	97.176
x_4	118.759

c. Use stepwise regression to specify a subset regression model.

Ans:

The same "stepAIC()" function was used again in R to calculate the stepwise regression model for the data. The result is identical to what is seen in part a), so the above Table 1 shows the same output. Table 1 however doesn't show the steps during stepwise regression, where the function checks first to see if all regressors should be retained. These can be seen in section 1.3 of the Code Appendix. As with part a), the final model chosen is the full model with all five regressors.

Check appendix

d. Apply all possible regressions to the data. Evaluate R_p^2, C_p, and MS_{Res} for each model. Which subset model do you recommend?

Ans:

Using the leaps package and its "regsubsets()" function, the all possible regressions method was used to find the "best" subset. The package only provides by default the R_p^2 and C_p values, so the MS_{Res} is calculated based on the SS_{Res} values which is by default given. Based on these three metrics, two have selected the full model, and one has selected a reduced model that excludes x_4. Based on the majority vote and considering the choices previously by the forward and backward methods, I would go with the full model. There is a nuance though, for example R_p^2 is utilized here, rather than $R_{Adj,p}^2$. However, when using that method also, it likewise chooses the full model over any other model.

Table 3 The table below shows the results based on the regsubsets() function in R. It shows the best performing model and corresponding metric when utilizing R_p^2, C_p, and MS_{Res}.

Metric	Best Model	Best value
R_p^2	$y \sim x_1 + x_2 + x_3 + x_4 + x_5$	0.9175502
C_p	$y \sim x_1 + x_2 + x_3 + x_4$	5.754262
MS_{Res}	$y \sim x_1 + x_2 + x_3 + x_4 + x_5$	74.79415

e. Compare and contrast the models produced by the variable selection strategies in part a-d.

Ans:
There was a roughly unanimous decision amongst the different subset selection methods. Using forward selection, backward elimination, stepwise regression, and all possible regressions method, the chosen model has continuously been the full model, that is, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 x_5$. The only exception is with regard to the C_p metric in part d). This is the only time that an alternative model was recommended. Doing some additional investigation, the BIC metric also chooses the same model as C_p, while AIC also chooses the full model. Therefore, it is possible that the model excluding x_5 is also viable, but it is uncertain since this model was not chosen nearly unanimously.

2. Use any math/stat software (e.g., www.numbergenerator.org/randomnumbergenerator) of your choice to find a random number generator to randomly select 20 rows of Table B.1. Then do Problem 11.1 (a), (b), page 386 of Textbook, using your generated data.
   a. Calculate the PRESS statistic for this model. What comments can you make about the likely predictive performance of this model?

Ans:
The randomly chosen rows for this problem are: 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 14, 17, 18, 19, 21, 22, 23, 25, 26, and 28.

To calculate the PRESS statistic, the following formula was used,

$$PRESS = \sum_{i=1}^{n} [y_i - \hat{y}_{(i)}]^2 = \sum_{i=1}^{n} \left(\frac{e_i}{1 - h_{ii}} \right)^2.$$

Using R, this evaluates to $PRESS = 199.7497$. Calculating the statistic itself is fairly straightforward, since it simply involves finding a few variables and then taking the sum of squares. However, it alone does not say much. For example, what is 199.7497 relative to? Ideally, we want PRESS to be as small as possible. If we for example compared it to different models that utilized subsets of the regressors, we can then look at their corresponding PRESS statistics. If for example a model with one fewer regressor had only a slightly larger PRESS (e.g., $PRESS = 200$), then we may possibly want to choose that model, since it likely has better generalizability while only having a slightly worse performance on the training data.

Another possibility is to look at the SS_{Res}, which is supposed to be a similar metric, since it is doing the same calculation, but without removing the i'th observation for each calculation of $\hat{y}_{(i)}$. A comparison of the residuals e_i to the corresponding $PRESS$ residuals can be seen in Section 2.1.1 of the Appendix. Looking at these residuals, some of them are comparatively closer to each other, while others have a larger difference. Taking the sums of squares of the ordinary residuals leads to 38.56744. This value is much smaller than the $PRESS$ calculation of 199.7497. This could possibly indicate that there are some serious issues with the model as is. If they were similar, it would indicate that the model is fitting the observations fine, without being too reliant on any set of outliers. However, since there's a noticeable difference as is, it could instead indicate that there are outliers or that the model is not properly fitting all the observations.

Without doing more extensive modeling, it would be difficult to draw too much of a conclusion. This could involve doing residual analysis, looking for outliers visually using plots, etc. It's also worth noting that this dataset is not too large, it is based on 20 observations out of an already small sized 28 total observations. Therefore, it could be unrealistic to have too high hopes for a seemingly appropriate model, for example if there were hundreds or thousands of observations. However, with more observations come further problems, such as if a subset of outliers would make the $PRESS$ statistic more difficult to use, since that method removes only single observations at a time, while a subset of outliers could prevent that from being as apparent.

b. Delete half the observations (chosen at random), and refit the regression model. Have the regression coefficients changed dramatically? How well does this model predict the number of games won for the deleted observations?

Ans:
After deleting half of the observations, the remaining rows are: 1, 2, 4, 5, 9, 14, 18, 19, 23, and 26.

The two sets of regression coefficients for part (a) and part (b) can be seen below in Table 4. Model 1 represents the fitted model from part (a), and Model 2 represents the model using half the observations from part (b). Looking at some of the regressions, there are some minor and major differences. For example, with $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_5$, $\hat{\beta}_6$, $\hat{\beta}_8$, and $\hat{\beta}_9$ the differences seem relatively minor (*Note: The variables are not scaled or normalized in anyway, so the initial range of the variables may have some unseen influences.*).

However, other regressors have some larger changes. The biggest change can be seen in $\hat{\beta}_0$, where it goes from -10.3272 to -179.5886. This is quite a large jump in comparison to every other coefficient. Often times the difference is less than 1, and other times it is less than 5 or 10. However, in this one case, the change is quite dramatic. It is hard to tell if it is due for example due to the observations showing a different distribution, or if there is just too few observations to make a strong model.

Table 4 The table below shows the regression coefficients for the model from part (a) (Model 1) and the model from part (b) (Model 2).

Regression Coefficients	Model 1	Model 2
$\hat{\beta}_0$	-10.3272	-179.5886
$\hat{\beta}_1$	-0.0015	0.0112
$\hat{\beta}_2$	0.0032	-0.0047
$\hat{\beta}_3$	0.4581	8.3696
$\hat{\beta}_4$	0.0775	1.4701
$\hat{\beta}_5$	0.0452	-0.0935
$\hat{\beta}_6$	0.0018	0.0196
$\hat{\beta}_7$	0.1062	-3.3489
$\hat{\beta}_8$	-0.0058	-0.0346
$\hat{\beta}_9$	-0.0019	0.0073

This time, the $PRESS = 3{,}675{,}576$, which is significantly larger than before. At the same time, the $SS_{Res} = 2.2605 \times 10^{-19}$. This is quite a dramatic change and seems to indicate that the predictive powers from before have since significantly decreased. Before, there was already a noticeable difference between the two metrics. However, now the difference seems to have significantly increased. Doing a closer analysis of the individual residuals in comparison to the $PRESS$ residuals can help to explain why the numbers are now so different. Looking at section 2.2.1 of the Code Appendix, it shows that certain $PRESS$ residuals are suddenly quite large, for example observation 4 and 14 have $PRESS$ residuals of -1820.0988 and -571.7961 respectively. Taking the sums of squares with such values could easily explain how the $PRESS$ could explode so much. On the other hand, the ordinary residuals have all become extremely small decimals that are roughly close to zero. Taking the sums of squares of these would decrease the value even more. This result indicates that at least with Model 2, the model is greatly overfitting to the training data, since it's such a small sample size.

The question also asks us to do a simple cross-validation, where half the observations are used for modeling and half are used for prediction. Using the observations that were removed before fitting Model 2 from Model 1, these were then used for prediction based on the fitted model of Model 2. The resulting $SS_{Res} = 5{,}389.1$. This is clearly much larger than the same calculation based on only the Model 2 data (i.e., 2.2605×10^{-19}). At the same time, it is much smaller than the $PRESS$ statistic for Model 2 (i.e., 3,675,576). However, given the difference in the testing SS_{Res} versus the training SS_{Res}, it is apparent that Model 2 has an inability to generalize to data outside the training set. This is not too surprising, given what was seen previously, in addition to the size of the sample data.

Code Appendix

1

```
### Problem 1
df <- MPV::table.b2
n <- 20
set.seed(1); chosen_rows <- sort(sample(seq(1, nrow(df)), n))
df1 <- df[chosen_rows,]
```

1.1

```
# Reference: https://stats.stackexchange.com/questions/347652/default-stepaic-in-r
# (a) forward selection
model_0 <- lm(y~1, data = df1)
model_1 <- lm(y~., data = df1)
forward1 <- MASS::stepAIC(model_0,
              scope = list(upper=model_1, lower=model_0),
              direction = c('forward'))

## Start:  AIC=131.07
## y ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + x4    1    9125.4  3574.7 107.72
## + x1    1    5965.9  6734.1 120.38
## + x5    1    1707.9 10992.2 130.18
## <none>               12700.1 131.07
## + x3    1     253.6 12446.4 132.67
## + x2    1       0.2 12699.9 133.07
##
## Step:  AIC=107.72
## y ~ x4
##
##        Df Sum of Sq    RSS     AIC
## + x3    1   1796.39 1778.3  95.753
## + x5    1   1308.09 2266.6 100.606
## + x1    1    397.82 3176.8 107.358
## <none>              3574.7 107.718
## + x2    1      1.41 3573.3 109.710
##
## Step:  AIC=95.75
## y ~ x4 + x3
##
##        Df Sum of Sq    RSS    AIC
## + x2    1   201.056 1577.2 95.354
## + x1    1   173.115 1605.2 95.705
## <none>              1778.3 95.753
## + x5    1    90.108 1688.2 96.713
##
## Step:  AIC=95.35
## y ~ x4 + x3 + x2
##
##        Df Sum of Sq    RSS    AIC
## + x1    1    398.90 1178.3 91.522
## <none>              1577.2 95.354
## + x5    1     13.93 1563.3 97.176
##
## Step:  AIC=91.52
## y ~ x4 + x3 + x2 + x1
##
##        Df Sum of Sq    RSS    AIC
## + x5    1    131.21 1047.1 91.161
## <none>              1178.3 91.522
##
## Step:  AIC=91.16
## y ~ x4 + x3 + x2 + x1 + x5
```

1.2

```
# (b) backward elimination
backward1 <- MASS::stepAIC(model_1,
                           # scope = list(upper=model_1, lower=model_0),
                           direction = c('backward'))

## Start:  AIC=91.16
## y ~ x1 + x2 + x3 + x4 + x5
##
##        Df Sum of Sq    RSS     AIC
## <none>              1047.1  91.161
## - x5    1     131.2 1178.3  91.522
## - x2    1     255.5 1302.6  93.528
## - x3    1     332.5 1379.7  94.677
## - x1    1     516.2 1563.3  97.176
## - x4    1    3550.1 4597.2 118.750
```

1.3

```
# (c) stepwise regression
step1 <- MASS::stepAIC(model_0,
                           scope = list(upper=model_1, lower=model_0),
                           direction = c('both'))

## Start:  AIC=131.07
## y ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + x4    1    9125.4  3574.7 107.72
## + x1    1    5965.9  6734.1 120.38
## + x5    1    1707.9 10992.2 130.18
## <none>              12700.1 131.07
## + x3    1     253.6 12446.4 132.67
## + x2    1       0.2 12699.9 133.07
##
## Step:  AIC=107.72
## y ~ x4
##
##        Df Sum of Sq     RSS     AIC
## + x3    1    1796.4  1778.3  95.753
## + x5    1    1308.1  2266.6 100.606
## + x1    1     397.8  3176.8 107.358
## <none>               3574.7 107.718
## + x2    1       1.4  3573.3 109.710
## - x4    1    9125.4 12700.1 131.073
##
## Step:  AIC=95.75
## y ~ x4 + x3
##
##        Df Sum of Sq     RSS     AIC
## + x2    1     201.1  1577.2  95.354
## + x1    1     173.1  1605.2  95.705
## <none>               1778.3  95.753
## + x5    1      90.1  1688.2  96.713
## - x3    1    1796.4  3574.7 107.718
## - x4    1   10668.1 12446.4 132.669
##
## Step:  AIC=95.35
## y ~ x4 + x3 + x2
##
##        Df Sum of Sq     RSS     AIC
## + x1    1     398.9  1178.3  91.522
## <none>               1577.2  95.354
## - x2    1     201.1  1778.3  95.753
## + x5    1      13.9  1563.3  97.176
## - x3    1    1996.0  3573.3 109.710
## - x4    1   10843.4 12420.6 134.628
```

```
##
## Step:  AIC=91.52
## y ~ x4 + x3 + x2 + x1
##
##        Df Sum of Sq    RSS     AIC
## + x5    1     131.2 1047.1  91.161
## <none>              1178.3  91.522
## - x1    1     398.9 1577.2  95.354
## - x2    1     426.8 1605.2  95.705
## - x3    1    1896.7 3075.0 108.707
## - x4    1    4035.6 5213.9 119.267
##
## Step:  AIC=91.16
## y ~ x4 + x3 + x2 + x1 + x5
##
##        Df Sum of Sq    RSS     AIC
## <none>              1047.1  91.161
## - x5    1     131.2 1178.3  91.522
## - x2    1     255.5 1302.6  93.528
## - x3    1     332.5 1379.7  94.677
## - x1    1     516.2 1563.3  97.176
## - x4    1    3550.1 4597.2 118.750
```

1.4

```
# (d) all possible regressions
best1 <- leaps::regsubsets(x = y~., data = df1, nvmax = 5)
best1_sum <- summary(best1)
p.m <- 2:6
aic <- n * log(best1_sum$rss / n) + 2 * p.m
data.frame(
  Adj.R2 = which.max(best1_sum$adjr2),
  CP = which.min(best1_sum$cp),
  BIC = which.min(best1_sum$bic),
  AIC = which.min(aic)
)
```

```
##   Adj.R2 CP BIC AIC
## 1      5  4   4   5
```

```
### Problem 2
df <- MPV::table.b1
n <- 20
set.seed(1); chosen_rows <- sort(sample(seq(1, nrow(df)), n))
df2 <- df[chosen_rows,]
```

2.1

```
# (a)
# PRESS residuals
beta_hat_calc <- function(X, y) {
  X <- as.matrix(X)
  beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
  return(beta_hat)
}
H_calc <- function(X) {
  X <- as.matrix(X)
  H <- X %*% solve(t(X) %*% X) %*% t(X)
  return(H)
}
y_hat_calc <- function(H, y) {
  y_hat <- H %*% y
  return(y_hat)
}
e_calc <- function(y, y_hat) {
  e <- y - y_hat
  return(e)
}
```

```
X <- df2[,2:ncol(df2)]
ones <- rep(1, n)
X <- cbind(ones, X)
y <- df2$y
beta_hat <- beta_hat_calc(X=X, y=y)
H <- H_calc(X)
y_hat <- y_hat_calc(H=H, y=y)
e <- e_calc(y=y, y_hat=y_hat)
H_diag <- diag(H)

PRESS_res <- sapply(1:n, function(x) e[x] / (1 - H_diag[x]))
PRESS <- sum(PRESS_res^2)
sum(e^2)

## [1] 38.56744
```

2.1.1

```
# 2.1.1
data.frame(e=e,PRESS=PRESS_res)

##               e        PRESS
## 1   2.603555099  3.326864455
## 2   0.689629058  1.748870047
## 4   2.530430030  8.877899013
## 5   0.084051420  0.204607506
## 6  -1.636850024 -3.257775145
## 7  -1.802280589 -2.611010953
## 9   0.832731273  2.736284742
## 10 -2.353290395 -4.971565865
## 11  1.197279727  3.116432957
## 12  1.169352737  1.656639103
## 14 -0.849504725 -1.593749179
## 17 -0.657157608 -1.301927252
## 18 -0.725923065 -1.792462885
## 19  0.789040347  1.454843630
## 21 -1.682754382 -4.522626657
## 22  1.160479603  2.323335875
## 23  0.695241265  1.231994527
## 25  0.001838846  0.004286539
## 26 -0.357409797 -0.600428788
## 28 -1.688458818 -2.716326614
```

2.2

```
# (b)
set.seed(1)
chosen_row_subset <- sort(sample(chosen_rows, size = length(chosen_rows)/2))
df2b <- df[chosen_row_subset,]
n <- 10
X <- df2b[,2:ncol(df2)]
ones <- rep(1, n)
X <- cbind(ones, X)
y <- df2b$y
beta_hat <- beta_hat_calc(X=X, y=y)
H <- H_calc(X)
y_hat <- y_hat_calc(H=H, y=y)
e <- e_calc(y=y, y_hat=y_hat)
H_diag <- diag(H)

PRESS_res_b <- sapply(1:n, function(x) e[x] / (1 - H_diag[x]))
PRESS_b <- sum(PRESS_res_b^2)
sum(e^2)

## [1] 2.260452e-19
```

2.2.1

```
# 2.2.1
data.frame(e=e,PRESS=PRESS_res_b)
```

```
##               e        PRESS
## 1  6.255441e-11    -6.081513
## 2  1.152571e-10    75.408150
## 4  1.309424e-10 -1820.098765
## 5  1.915428e-10    84.497208
## 9  1.492748e-10    76.014699
## 14 1.444853e-10  -571.796134
## 18 1.536167e-10   -54.872145
## 19 1.609957e-10    72.296341
## 23 1.667431e-10    80.964313
## 26 1.858318e-10   -49.305526
```

2.2.2

```
# predictive powers
deleted_rows <- chosen_rows[!(chosen_rows %in% chosen_row_subset)]
df2c <- df[deleted_rows,]
X2 <- df2c[,2:ncol(df2c)]
X2 <- cbind(ones, X2)
y_hat2 <- as.matrix(X2) %*% as.matrix(beta_hat)
y2 <- df2c$y
e2 <- e_calc(y=y2, y_hat=y_hat2)
sum(e2^2)
```

```
## [1] 5389.1
```