

# Midterm - STAT460

Brianna Humphries

10/16/2020

## Import Data

```
library(readxl)
data <- read_excel("Midtest_Data.xlsx")
names(data)[3] <- "win"
names(data)[4] <- "runs"
names(data)[5] <- "ba"
names(data)[6] <- "dp"
names(data)[7] <- "walk"
names(data)[8] <- "so"
attach(data)
```

## Question 10.1

Use the forward selection method to fit the best multiple linear regression model (AICc) for the response variable percentage of winning. Do not take the variable year into your predictor variable.

## Solution

The forward selection below for R uses AIC to find the best multiple linear regression model for the response variable percentage of winning.

```
fit10.1 <- lm(win~1, data=data)
step.for <- step(fit10.1, direction="forward", scope=~runs+ba+dp+walk+so)
```

```
## Start:  AIC=-211.82
## win ~ 1
##
##           Df Sum of Sq      RSS      AIC
## + ba      1  0.050075  0.14069 -222.00
## + runs    1  0.049713  0.14105 -221.90
## + so      1  0.034203  0.15656 -217.73
## + walk    1  0.013187  0.17758 -212.69
## <none>                0.19076 -211.82
## + dp      1  0.000044  0.19072 -209.83
##
## Step:  AIC=-222
## win ~ ba
##
##           Df Sum of Sq      RSS      AIC
## + walk    1  0.042053  0.098636 -234.21
## + so      1  0.027546  0.113143 -228.72
## <none>                0.140689 -222.00
```

```

## + runs  1  0.005240 0.135449 -221.52
## + dp    1  0.004305 0.136383 -221.25
##
## Step: AIC=-234.21
## win ~ ba + walk
##
##          Df Sum of Sq      RSS      AIC
## + so      1 0.0195773 0.079059 -241.06
## + runs    1 0.0092226 0.089413 -236.13
## <none>                0.098636 -234.21
## + dp      1 0.0004366 0.098199 -232.38
##
## Step: AIC=-241.06
## win ~ ba + walk + so
##
##          Df Sum of Sq      RSS      AIC
## + runs    1 0.0062069 0.072852 -242.33
## + dp      1 0.0043958 0.074663 -241.35
## <none>                0.079059 -241.06
##
## Step: AIC=-242.33
## win ~ ba + walk + so + runs
##
##          Df Sum of Sq      RSS      AIC
## + dp      1 0.0053229 0.067529 -243.36
## <none>                0.072852 -242.33
##
## Step: AIC=-243.36
## win ~ ba + walk + so + runs + dp
summary(step.for)



##
## Call:
## lm(formula = win ~ ba + walk + so + runs + dp, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.108803 -0.020586  0.007429  0.022087  0.083116
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2776752  0.1913151  -1.451  0.15583
## ba           1.7419995  0.9284706   1.876  0.06923 .
## walk        -0.0005897  0.0001292  -4.566 6.23e-05 ***
## so           0.0003461  0.0001044   3.315  0.00219 **
## runs         0.0002778  0.0001466   1.895  0.06659 .
## dp           0.0007370  0.0004502   1.637  0.11084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04457 on 34 degrees of freedom
## Multiple R-squared:  0.646, Adjusted R-squared:  0.594
## F-statistic: 12.41 on 5 and 34 DF, p-value: 6.856e-07



```

The forward selection starts with no predictors in the model and adds the best predictors until the addition of another predictor is not statistically significant. The output shows that the model with the minimum AIC is the last model with all the predictors (ba, walk, so, runs, and dp). This model has an AIC value of -243.36. Since the R function uses AIC instead of AICc, I included the JMP output as well, since JMP uses minimum AICc. Here is the output for JMP:

**Stepwise Fit for win: The team's winning percentage**

**Stepwise Regression Control**

Stopping Rule: Minimum AICc   Enter All Make Model

Direction: Forward   Remove All Run Model

Go Stop Step

SSE	DFE	RMSE	RSquare	RSquare Adj	Cp	p	AICc	BIC
0.0675288	34	0.0445661	0.6460	0.5940	6	6	-124.348	-116.026

**Current Estimates**

Lock	Entered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Intercept	-0.2776752	1	0	0.000	1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	runs: The number of runs scored by the team	0.00027783	1	0.007134	3.592	0.06659
<input type="checkbox"/>	<input checked="" type="checkbox"/>	ba: The team's overall batting average	1.74199948	1	0.006991	3.520	0.06923
<input type="checkbox"/>	<input checked="" type="checkbox"/>	dp: The total number of double plays	0.00073702	1	0.005323	2.680	0.11084
<input type="checkbox"/>	<input checked="" type="checkbox"/>	walk: The number of walks given to the other team	-0.0005897	1	0.041404	20.847	6.23e-5
<input type="checkbox"/>	<input checked="" type="checkbox"/>	so: The number of strikeouts by the team's pitchers	0.00034611	1	0.021824	10.988	0.00219

**Step History**

Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p	AICc	BIC	
1	ba: The team's overall batting average	Entered	0.0007	0.050075	0.2625	34.835	2	-105.82	-101.42	<input type="radio"/>
2	walk: The number of walks given to the other team	Entered	0.0003	0.042053	0.4829	15.662	3	-117.55	-111.94	<input type="radio"/>
3	so: The number of strikeouts by the team's pitchers	Entered	0.0051	0.019577	0.5856	7.8051	4	-123.78	-117.1	<input type="radio"/>
4	runs: The number of runs scored by the team	Entered	0.0930	0.006207	0.6181	6.68	5	-124.27	-116.68	<input type="radio"/>
5	dp: The total number of double plays	Entered	0.1108	0.005323	0.6460	6	6	-124.35	-116.03	<input type="radio"/>
6	Best	Specific	.	.	0.6460	6	6	-124.35	-116.03	<input checked="" type="radio"/>

The best model in JMP also included all the predictors and had an AICc value of -124.35.

## Question 10.2

Discuss the significance of your fitted model. Your interpretation should include the test statistic and the p-value.

### Solution

The model below represents the model chosen in the forward selection process in question 10.1.

```
model <- lm(formula = win ~ ba + walk + so + runs + dp, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = win ~ ba + walk + so + runs + dp, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.108803 -0.020586  0.007429  0.022087  0.083116
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2776752  0.1913151  -1.451  0.15583
## ba           1.7419995  0.9284706   1.876  0.06923 .
## walk        -0.0005897  0.0001292  -4.566 6.23e-05 ***
## so           0.0003461  0.0001044   3.315  0.00219 **
## runs         0.0002778  0.0001466   1.895  0.06659 .
## dp           0.0007370  0.0004502   1.637  0.11084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04457 on 34 degrees of freedom
## Multiple R-squared:  0.646, Adjusted R-squared:  0.594
## F-statistic: 12.41 on 5 and 34 DF, p-value: 6.856e-07
```

The equation for the model is  $y_{win} = -0.2777 + 1.7420(ba) - 0.0006(walk) + 0.0003(so) + 0.0003(runs) + 0.0007(dp)$ .

The null hypothesis for the test of significance of a linear regression model is that all the coefficients equal 0. The alternative hypothesis is that at least one of the coefficients does not equal 0. Rejection of the null hypothesis implies that at least one predictor contributes to the model.

**F Test Statistic** The summary of the model shows that the calculated F statistic is 12.41 with degrees of freedom 5 and 34. The critical F value for those degrees of freedom on a significance level of 0.05 is 2.49. Since  $12.41 > 2.49$ , then we reject the null hypothesis at significance level 0.05. This means that at least one of the predictors in the model contributes significantly to a team's winning percentage.

**P-Value** The p-value of the model is shown above as  $6.86e-07$ . Since this value is significantly less than 0.05, then we reject the null hypothesis. This means that we have further proof that at least one of the predictors in the model contributes significantly to a team's winning percentage.

## Question 10.3

Use t-tests to assess the contribution of each predictor to the model. Discuss your findings.

### Solution

The null hypothesis for testing any individual regression coefficient is that the coefficient equals 0. The alternative hypothesis that the coefficient does not equal 0. If the null hypothesis is not rejected, then that predictor is not statistically significant to the model. The value from the t-test of each predictor is shown in the “t value” column in the output shown in Question 10.2. The critical t-value that each value will be compared to from a significance level of 0.05 is  $t_{0.025,34} = \pm 2.0322$ .

Therefore, the rejection region is  $-2.0322 > x$  or  $x > 2.0322$ .

#### Predictors:

**ba:** The t-value of the batting average predictor (ba) is 1.876. Since  $-2.0322 < 1.876 < 2.0322$ , then we fail to reject the null hypothesis, and the predictor is not statistically significant.

**walk:** The t-value of the number of walks given predictor (walk) is -4.566. Since  $-4.566 < -2.0322$ , then we reject the null hypothesis, and the predictor is statistically significant.

**so:** The t-value of the number of strikeouts predictor (so) is 3.315. Since  $3.315 > 2.0322$ , then we reject the null hypothesis, and the predictor is statistically significant.

**runs:** The t-value of the number of runs scored predictor (runs) is 1.895. Since  $-2.0322 < 1.895 < 2.0322$ , then we fail to reject the null hypothesis, and the predictor is not statistically significant.

**dp:** The t-value of the number of number of double plays predictor (dp) is 1.637.  $-2.0322 < 1.637 < 2.0322$ , then we fail to reject the null hypothesis, and the predictor is not statistically significant.

The intercept coefficient is also not statistically significant. Therefore, the only statistically significant predictors in the model are the number of walks given (walk) and the number of strikeouts (so). We could have also used the “Pr(>|t|)” value in the output of Question 10.2. If the value is less than the significance level 0.05, then it is statistically significant. This also shows that the only statistically significant predictors in the model are walk and so.

## Question 10.4

Give the 95% confidence interval (CI) on the mean percentage of winning for the runs=707, ba=0.254, dp=152, walk=467, sa=916. Your answer must include the formula for calculating the CI. (You do not have to use all the values if your best model does not contain the corresponding variables.)

### Solution

The estimate of winning percentage for the specified data and its 95% confidence interval are calculated below.

```
new.data <- data.frame(runs=c(707),ba=c(0.254), dp=c(152), walk=c(467), so=c(916))
predict(model, newdata=new.data, interval="confidence", se=T, level=0.95)
```

```
## $fit
##          fit          lwr          upr
## 1 0.5148921 0.4904402 0.5393441
##
## $se.fit
## [1] 0.01203197
##
## $df
## [1] 34
##
## $residual.scale
## [1] 0.04456613
```

The 95% confidence interval on the mean percentage of winning for the given data is shown in the output above as (0.490, 0.539). This can be calculated by using the confidence interval formula,

$$\left( \hat{y}(x_0) - t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 x_0' (\mathbf{X}'\mathbf{X})^{-1} x_0}, \hat{y}(x_0) + t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 x_0' (\mathbf{X}'\mathbf{X})^{-1} x_0} \right).$$

Since the standard error for the confidence interval is  $se(\hat{y}(x_0)) = \sqrt{\hat{\sigma}^2 x_0' (\mathbf{X}'\mathbf{X})^{-1} x_0}$  then the confidence interval equation can be simplified to  $\left( \hat{y}(x_0) - t_{\alpha/2, n-p} se(\hat{y}(x_0)), \hat{y}(x_0) + t_{\alpha/2, n-p} se(\hat{y}(x_0)) \right)$ .

The estimate of the winning percentage for the given data is shown above as 0.5149. This can be calculated by plugging the new data into the equation for the model,

$$\hat{y}(x_0) = -0.2777 + 1.7420(\text{ba}) - 0.0006(\text{walk}) + 0.0003(\text{so}) + 0.0003(\text{runs}) + 0.0007(\text{dp}).$$

The standard error of the estimate is shown in the output above as 0.0120. The t-value for the model is  $t_{\alpha/2, n-p} = t_{0.025, 34} = 2.032$ . If we substitute all the values in the confidence interval equation, we get

$$\text{CI} = (0.5149 - 2.032 * 0.01203, 0.5149 + 2.032 * 0.01203)$$

$$\text{CI} = (0.490, 0.539).$$

## Question 10.5

Give the 95% prediction interval (PI) on a new observation of the percentage of winning for the runs=707, ba=0.254, dp=152, walk=467, sa=916. Your answer must include the formula for calculating the PI. (You do not have to use all the values if your best model does not contain the corresponding variables.)

### Solution

The 95% prediction interval and the model's  $\sigma^2$  value are calculated below.

```
predict(model, newdata=new.data, interval="predict", level=0.95)
```

```
##           fit           lwr           upr
## 1 0.5148921 0.4210802 0.6087041
```

```
summary(model)$sigma^2
```

```
## [1] 0.00198614
```

The 95% prediction interval of the percentage of winning for the given data is shown in the output above as (0.421, 0.609). This can be calculated by using the prediction interval formula,

$$\left( \hat{y}(x_0) - t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 [1 + x_0'(\mathbf{X}'\mathbf{X})^{-1}x_0]}, \hat{y}(x_0) + t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 [1 + x_0'(\mathbf{X}'\mathbf{X})^{-1}x_0]} \right).$$

Since the standard error for the prediction interval is  $se(\hat{y}(x_0)) = \sqrt{\hat{\sigma}^2 [1 + x_0'(\mathbf{X}'\mathbf{X})^{-1}x_0]}$  then the prediction interval equation can be simplified to  $\left( \hat{y}(x_0) - t_{\alpha/2, n-p} se(\hat{y}(x_0)), \hat{y}(x_0) + t_{\alpha/2, n-p} se(\hat{y}(x_0)) \right)$ .

The estimate of the winning percentage for the given data is calculated in Question 10.5 as 0.5149. The t-value for the model is  $t_{\alpha/2, n-p} = t_{0.025, 34} = 2.032$ .

The standard error for the prediction interval can be calculated using the standard error from the confidence interval, which was 0.0120. Using the  $\sigma^2$  value of the model of 0.001986 from the output above, the prediction interval standard error can be calculated by  $se(\hat{y}(x_0)) = \sqrt{0.001986 + 0.0120^2} = 0.0462$ .

If we substitute all the values in the prediction interval equation, we get

$$PI = (0.5149 - 2.032 * 0.0462, 0.5149 + 2.032 * 0.0462)$$

$$PI = (0.421, 0.609).$$

## Question 10.6

Compare the results from part 4 and part 5. Which interval has a longer interval length? Explain the reason.

### Solution

Recall that the confidence interval from Question 10.4 is (0.490, 0.539), and the prediction interval from Question 10.5 is (0.421, 0.609). The prediction interval appears to have a longer interval length, where the upper and lower limits between the two intervals have a difference of 0.07. The reason that the PI's interval is wider than the CI's interval is because the standard error for the PI is larger. The standard error for the PI is larger because the PI is predicting an individual value while a CI is predicting a mean value. Therefore, there is more uncertainty in a PI's standard error, so a larger standard error, and thus a wider interval.

## Question 10.7

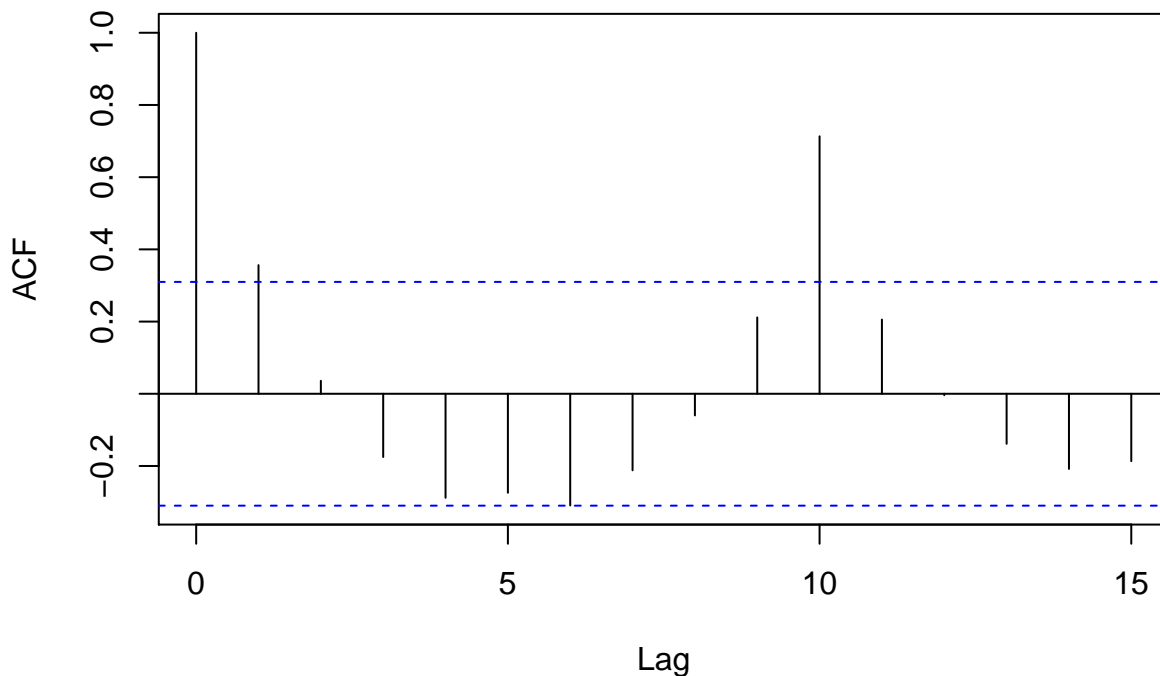
Let lag=15, calculate the Sample ACF, the corresponding Z-Statistic, and Ljung-Box Statistic of the percentage of winning. The output should be similar to the Table 2.3 on page 71 in the textbook. It will be fine if your outputs include all information contained in Table 2.3. The values of ACF, Z-statistic, and Ljung-Box statistic are needed. Is there an indication of non-stationary behavior in the residuals?

### Solution

The table above shows the Sample ACF, corresponding Z-Statistic, and Ljung-Box Statistic of the percentage of winning for each lag 1 to 15.

```
ts <- ts(data$win, frequency=10, start=c(1965,1))
y <- acf(data$win, lag.max = 15, type="correlation", main="ACF of Winning Percentage")
```

### ACF of Winning Percentage



```
lag <- c(1:15)
acf.values <- y$acf[2:16]
tab <- data.frame(Lag=lag, Sample.ACF= acf.values)
T <- dim(data)[1]
K <- 15
Z.Statistic <- (tab$Sample.ACF)*sqrt(T)
tab <- cbind(tab, Z.Statistic)
# ljung <-function(ts) {
#   for(i in c(1:K)) {
#     box<-(Box.test(data[3], lag=i, type="Ljung-Box"))
#     LjungBox.Statistic <- c(LjungBox.Statistic, box$statistic)
#     return(LjungBox.Statistic)}
# }
#couldn't get function to work so I wrote out manually
LjungBox.Statistic = c()
```



```

box<-(Box.test(data[3], lag=1, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=2, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=3, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=4, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=5, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=6, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=7, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=8, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=9, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=10, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=11, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=12, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=13, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=14, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
box<-(Box.test(data[3], lag=15, type="Ljung-Box"))
LjungBox.Statistic <- c(LjungBox.Statistic, box$Statistic)
tab <- cbind(tab, LjungBox.Statistic)
print.data.frame(tab)

```

```

##      Lag   Sample.ACF Z.Statistic LjungBox.Statistic
## 1      1  0.356340819  2.25369722      5.469855
## 2      2  0.035976390  0.22753467      5.527077
## 3      3 -0.175206014 -1.10810013      6.920893
## 4      4 -0.288136126 -1.82233287     10.795273
## 5      5 -0.274255101 -1.73454156     14.405634
## 6      6 -0.309450420 -1.95713630     19.137283
## 7      7 -0.212047346 -1.34110517     21.426364
## 8      8 -0.060011323 -0.37954493     21.615435
## 9      9  0.211607012  1.33832025     24.042088
## 10    10  0.713216330  4.51077614     52.528030
## 11    11  0.205604831  1.30035913     54.976969
## 12    12 -0.003936801 -0.02489852     54.977899
## 13    13 -0.138590090 -0.87652069     56.173014
## 14    14 -0.208194418 -1.31673712     58.973763
## 15    15 -0.186628504 -1.18034230     61.314352

```

```

#Critical Z Statistic
zc <- qnorm(1-0.05/2)
zc

```

```
## [1] 1.959964
```

**Z-Statistic** The null hypothesis for the Z statistic is that  $\rho_k = 0$ , and the alternative hypothesis is that  $\rho_k \neq 0$ . We reject the null hypothesis if the absolute value of the Z Statistic in the table above is greater than the critical Z statistic 1.96 (calculated above using significance level 0.05). The individual ACFs that we reject based on their Z statistics are of lags 1, 6, and 10. These values are an indication of non-stationary behavior

### Ljung-Box Statistic

The Ljung-Box goodness-of-fit statistic gives the group test of ACF. If the value at row j is bigger than the critical value of chi-squared at j, then we reject the null hypothesis that all ACF values up to j are 0. At lag 15,  $Q_{LB,15} = 61.3146$ . The critical QB value at lag 15 is shown below as 24.99.

```
#Critical QB Statistic at 15
```

```
qb_15 <- qchisq(1-0.05,15)
```

```
qb_15
```

```
## [1] 24.99579
```

Since the critical value at lag 15 is less than the Ljung-Box value at lag 15, then we reject the null hypothesis. Therefore we cannot define all the ACF values up to j as 0, which is another indication of non-stationary behavior.

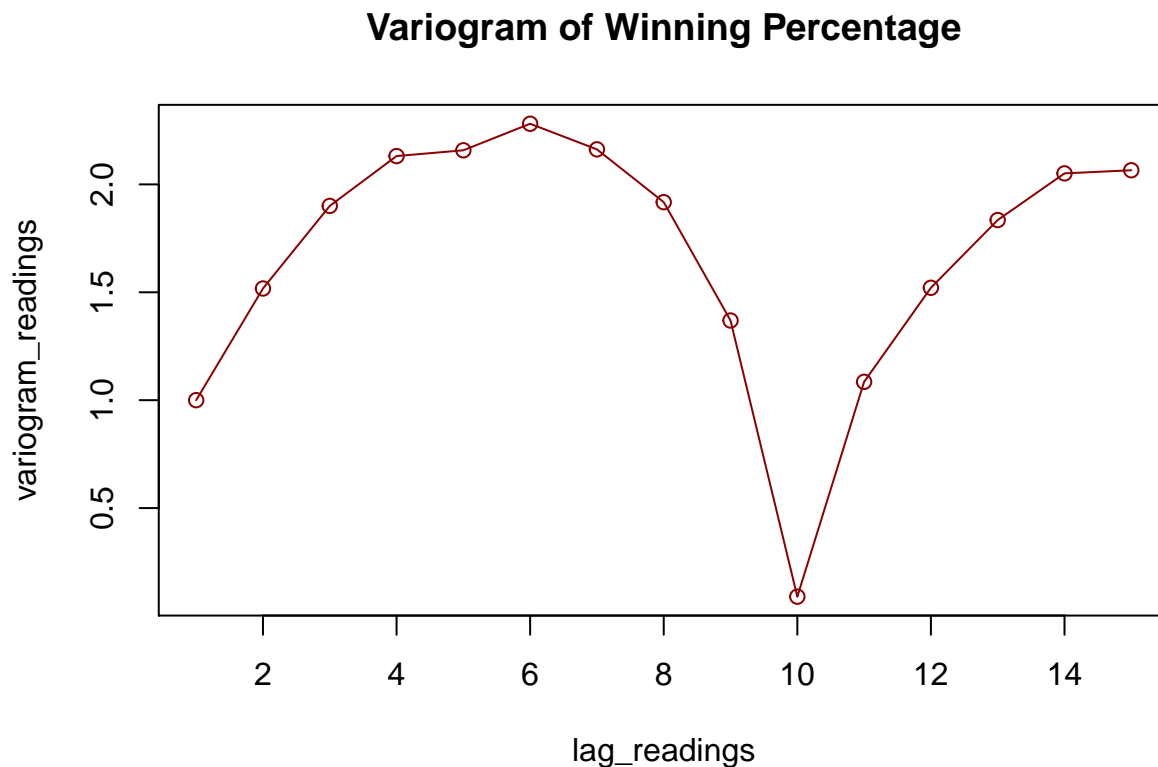
## Question 10.8

Let the lag=15, calculate the variogram of the percentage of winning. What can you tell from the variogram?

### Solution

The variogram of the percentage of winning for lag = 15 is shown below.

```
variogram_func <- function(x, lag) {  
  x <- as.matrix(x) # Make sure the x is a vector. It represents the observations of y_t.  
  Lag <- NULL  
  var_k <- NULL  
  vario <- NULL  
  for (k in 1:lag) {  
    Lag[k] <- k  
    var_k[k] <- sd(diff(x, k))^2  
    vario[k] <- var_k[k] / var_k[1]  
  }  
  return(as.data.frame(cbind(Lag, vario)))  
}  
  
x <- ts  
lag <- 15  
lag_readings <- 1:lag  
z <- variogram_func(x, lag)  
variogram_readings <- z$vario  
plot(lag_readings, variogram_readings, type="o", col="dark red",  
     main="Variogram of Winning Percentage")
```



The variogram shows that the data increases but then dips very low at lag 10. Since the data does not converge to a level and then fluctuate around it, then it is not stationary.

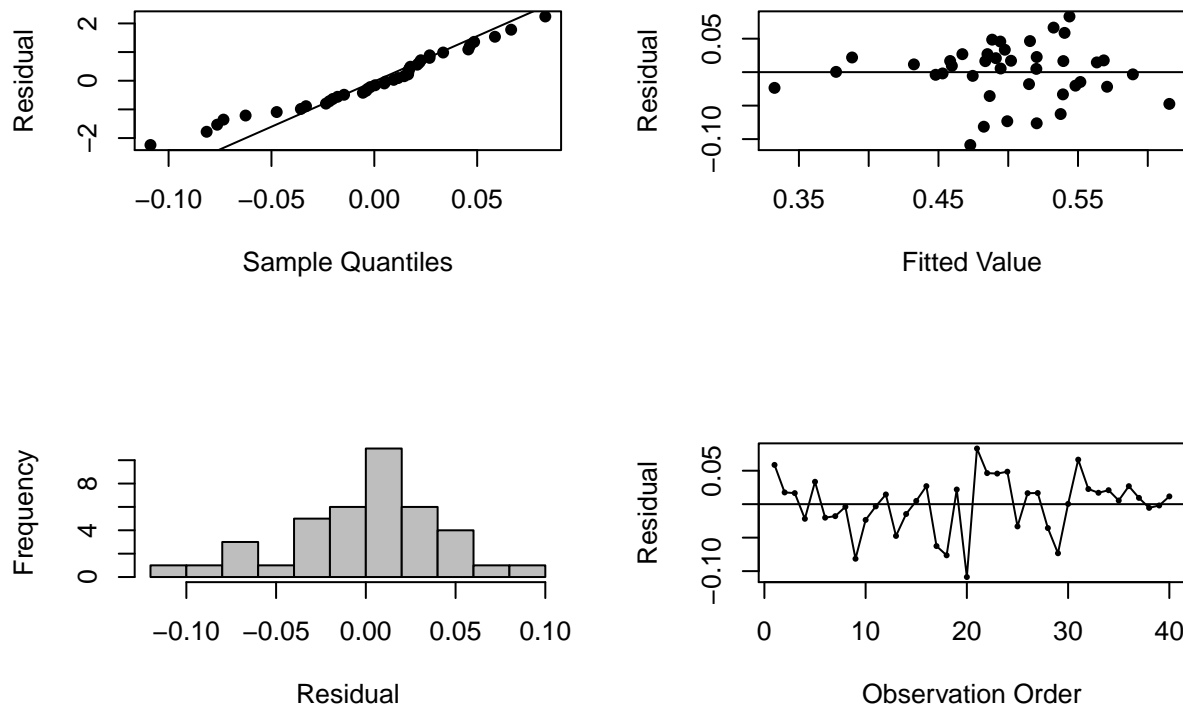
## Question 10.9

Plot the 4 in 1 residual plots (QQ plot, Fitted value vs Residual, Histogram of Residual, and Observation order vs Residual) and interpret the graphs. The graphs should be similar to Figure 3.1 on page 138 in the textbook.

### Solution

The plots below are the residual plots from the model created in the first half of the questions.

```
par(mfrow = c(2, 2), oma = c(0, 0, 0, 0))
qqnorm(model$residuals, datax = T, pch = 16, xlab = "Residual", main = "")
qqline(model$residuals, data = T)
plot(model$fitted.values, model$residuals, pch = 16,
xlab = "Fitted Value", ylab = "Residual")
abline(h = 0)
hist(model$residuals, col = "grey", xlab = "Residual", main = "")
plot(model$residuals, type = "l", xlab = "Observation Order", ylab = "Residual")
points(model$residuals, pch = 16, cex = .5)
abline(h = 0)
```



**QQ Plot** The first plot is the QQ Plot or Normal Probability Plot. The residuals generally follow along a straight line, so the normality assumption still stands. The lower half of the line strays a little bit from the line, but I don't think it's any reason to be too concerned.

**Fitted Value vs. Residuals** The second plot is the Fitted value vs. Residual plot. The plot at first glance might not appear to be randomly scattered since the lower end of the x-axis doesn't have many points, but the scatter does not have a funnel shape since both ends of the x-axis have less points. Therefore, the plot has a random scatter, and the constant variance assumption is satisfied.

**Histogram of Residuals** The third graph is the histogram of the residuals. The histogram shows a normal distribution since the frequencies follow a bell curve-like shape. Therefore, the histogram does not give any

serious indication of non-normality.

**Observation Order vs. Residuals** The last graph is the plot of observation order vs. residuals. The plot appears to be randomly scattered, therefore there are not any model inadequacies.

## Question 10.10

Discuss the model adequate by analyzing the residuals. Your output should be similar to the table 3.7 on page 141 in the textbook. Based on your outputs, answer the following questions: Are there any outliers? High leverage observations? High influential observations? Use criteria given in the textbook.

### Solution

The table below shows the Residuals, Studentized Residuals, R-students,  $h_{ii}$  values, and Cook's Distance values for each observation in our model.

```
studres <- rstandard(model)
res <- data.frame(Residuals= model$residuals, Studentized.Residuals=studres)
library(MASS)
rstudent <- studres(model)
res <- cbind(res, R.Student=rstudent)
hi <- hatvalues(model)
res <- cbind(res, hii=hi)
cookd <- cooks.distance(model)
res <- cbind(res, CooksDistance=cookd)
print.data.frame(res)
```

##	Residuals	Studentized.Residuals	R.Student	hii	CooksDistance
## 1	0.0586367515	1.387399642	1.407260144	0.10065408	3.590513e-02
## 2	0.0175415992	0.416145738	0.411028387	0.10538210	3.399922e-03
## 3	0.0165138903	0.404247423	0.399218790	0.15977733	5.179216e-03
## 4	-0.0218233319	-0.598084152	-0.592347357	0.32964069	2.931612e-02
## 5	0.0334544390	0.795106460	0.790712179	0.10865249	1.284375e-02
## 6	-0.0201730597	-0.486481646	-0.480950916	0.13423356	6.115642e-03
## 7	-0.0178921444	-0.416957368	-0.411834153	0.07288928	2.278055e-03
## 8	-0.0039659363	-0.096445692	-0.095029788	0.14863477	2.706568e-04
## 9	-0.0815013512	-1.903723300	-1.984251257	0.07719044	5.052517e-02
## 10	-0.0235065303	-0.623902020	-0.618207520	0.28528254	2.589538e-02
## 11	-0.0033084114	-0.081909228	-0.080703654	0.17858343	2.431040e-04
## 12	0.0145114056	0.351016484	0.346444261	0.13949369	3.328927e-03
## 13	-0.0474504725	-1.420512338	-1.442939720	0.43820072	2.623196e-01
## 14	-0.0147060048	-0.343639387	-0.339137607	0.07790928	1.662915e-03
## 15	0.0048640351	0.122558159	0.120769061	0.20695275	6.532890e-04
## 16	0.0269028890	0.630751320	0.625074219	0.08405030	6.084610e-03
## 17	-0.0625661946	-1.466713521	-1.492987115	0.08382379	3.280406e-02
## 18	-0.0763946652	-1.846187458	-1.917483908	0.13788621	9.085662e-02
## 19	0.0219121823	0.546800345	0.541083507	0.19145569	1.179969e-02
## 20	-0.1088030141	-2.561463641	-2.809064979	0.09156140	1.102153e-01
## 21	0.0831155213	1.986874652	2.082036655	0.11892385	8.880659e-02
## 22	0.0464095289	1.078779732	1.081466511	0.06816529	1.418857e-02
## 23	0.0456884908	1.085776561	1.088731872	0.10849698	2.391247e-02
## 24	0.0485233263	1.163753741	1.170052150	0.12467554	3.215016e-02
## 25	-0.0332014293	-0.822847943	-0.818851265	0.18028194	2.481851e-02
## 26	0.0164213895	0.441841072	0.436550027	0.30453103	1.424737e-02
## 27	0.0166344226	0.393604722	0.388659715	0.10074061	2.892604e-03
## 28	-0.0356975584	-0.835843106	-0.832052459	0.08163015	1.034978e-02
## 29	-0.0733184866	-1.743314710	-1.799808507	0.10943522	6.224321e-02
## 30	0.0003818821	0.009584036	0.009442055	0.20062267	3.842145e-06
## 31	0.0665092999	1.561280185	1.596440656	0.08632198	3.838298e-02
## 32	0.0226105727	0.546194083	0.540478269	0.13718187	7.905333e-03

## 33	0.0170034524	0.397150740	0.392177428	0.07710224	2.196205e-03
## 34	0.0208706614	0.534926182	0.529232644	0.23356588	1.453353e-02
## 35	0.0054505343	0.144584945	0.142486638	0.28447943	1.385236e-03
## 36	0.0268635254	0.626628269	0.620940379	0.07467102	5.281103e-03
## 37	0.0094069702	0.227821116	0.224617311	0.14157628	1.426676e-03
## 38	-0.0055556019	-0.132367212	-0.130439721	0.11306556	3.722617e-04
## 39	-0.0019554284	-0.047778783	-0.047072490	0.15665786	7.067524e-05
## 40	0.0115928510	0.281423131	0.277577149	0.14562006	2.249772e-03

If the studentized residual does not fall within the range -3 to 3, then the observation is an outlier. Since all of the values fall within that range, then there are no outliers. An observation is considered influential if its Cook's Distance value is greater than 1. Since none of the Cook's Distance values in the table above are greater than 1, then none of the observations are influential. An observation is considered high-leverage if the  $h_{ii}$  value exceeds  $\frac{2p}{n}$ . This value is calculated below as 0.30.

```
high <- 2*6/40
high
```

```
## [1] 0.3
```

Observations 4, 13, and 26 are all considered high leverage because their  $h_{ii}$  value in the table above is greater than 0.30. This means that they have “extreme” predictor x values.