STAT 560: Time Series Analysis

Md Mominul Islam, ID: 101009250

Date: 10/15/2020

Problem 1:

1. Using forward selection method to fit the best multiple linear regression model (AICc) for the response variable win. Do not take the variable year into your predictor variable.

Answer:

```
#forward selection method to fit the best multiple linear regression model
baseball.fit <- lm (Win ~ 1, data= baseball.data)
step.for <- step(baseball.fit, direction = "forward", scope = ~ Runs+Batting+
Doubleplays+Walk+Strikeout)
```

```
## Start:  AIC=-211.82
## Win ~ 1
##
##               Df Sum of Sq     RSS     AIC
## + Batting      1  0.050075 0.14069 -222.00
## + Runs         1  0.049713 0.14105 -221.90
## + Strikeout    1  0.034203 0.15656 -217.73
## + Walk         1  0.013187 0.17758 -212.69
## <none>                     0.19076 -211.82
## + Doubleplays  1  0.000044 0.19072 -209.83
##
## Step:  AIC=-222
## Win ~ Batting
##
##               Df Sum of Sq      RSS     AIC
## + Walk         1  0.042053 0.098636 -234.21
## + Strikeout    1  0.027546 0.113143 -228.72
## <none>                     0.140689 -222.00
## + Runs         1  0.005240 0.135449 -221.52
## + Doubleplays  1  0.004305 0.136383 -221.25
##
## Step:  AIC=-234.21
## Win ~ Batting + Walk
##
##               Df Sum of Sq      RSS     AIC
## + Strikeout    1 0.0195773 0.079059 -241.06
## + Runs         1 0.0092226 0.089413 -236.13
## <none>                     0.098636 -234.21
## + Doubleplays  1 0.0004366 0.098199 -232.38
##
## Step:  AIC=-241.06
## Win ~ Batting + Walk + Strikeout
##
```

```
##                Df Sum of Sq      RSS     AIC
## + Runs          1 0.0062069 0.072852 -242.33
## + Doubleplays   1 0.0043958 0.074663 -241.35
## <none>                      0.079059 -241.06
## 
## Step:  AIC=-242.33
## Win ~ Batting + Walk + Strikeout + Runs
## 
##                Df Sum of Sq      RSS     AIC
## + Doubleplays   1 0.0053229 0.067529 -243.36
## <none>                      0.072852 -242.33
## 
## Step:  AIC=-243.36
## Win ~ Batting + Walk + Strikeout + Runs + Doubleplays
```

```
# linear regression model
baseball.fit1 <- lm (Win ~ Runs + Batting + Doubleplays + Walk + Strikeout, d
ata= baseball.data)
# where runs=The number of runs scored by the team, ba =  The team's overall
batting average
# dp=The total number of double plays, walk = The number of walks given to th
e other teamArpma, so = The number of strikeouts by the team's pitchers
summary(baseball.fit1)
```

```
## 
## Call:
## lm(formula = Win ~ Runs + Batting + Doubleplays + Walk + Strikeout, 
##     data = baseball.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    
## Runs         0.0002778  0.0001466   1.895  0.06659 .  
## Batting      1.7419995  0.9284706   1.876  0.06923 .  
## Doubleplays  0.0007370  0.0004502   1.637  0.11084    
## Walk        -0.0005897  0.0001292  -4.566 6.23e-05 ***
## Strikeout    0.0003461  0.0001044   3.315  0.00219 ** 
## ---
## 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
```

```
coefficients(baseball.fit1)
```

```
##   (Intercept)          Runs       Batting   Doubleplays          Walk 
## -0.2776752183  0.0002778312  1.7419994841  0.0007370206 -0.0005897050 
##     Strikeout 
##  0.0003461112
```

Discussion:

Base on the above steps and we can conclude that regression model with predictor variables Batting, Walk, Strikeout, Runs and Doubleplays will generates less AIC value compare to other combinations. Therefore, the appropriate model for the 'Win' will be $Win = \beta_o + \beta_1(Runs) + \beta_2(Batting) + \beta_3(Doubleplays) + \beta_4(Walk) + \beta_5(Strikeout) + \varepsilon$.

Finally, the appropriate fit model base on this data is

Win = -0.2776752+ 0.0002778312(Runs)+ 1.7419994841(Batting) + 0.0007370206(Doubleplays) -0.0005897050(Walk)+ 0.0003461112(Strikeout)

Here, multiple R-squared: `0.646` and Adjusted R-squared: `0.594`

Problem 2

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

Answer:

```
summary(baseball.fit1)

## 
## Call:
## lm(formula = Win ~ Runs + Batting + Doubleplays + Walk + Strikeout, 
##     data = baseball.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    
## Runs         0.0002778  0.0001466   1.895  0.06659 .  
## Batting      1.7419995  0.9284706   1.876  0.06923 .  
## Doubleplays  0.0007370  0.0004502   1.637  0.11084    
## Walk        -0.0005897  0.0001292  -4.566 6.23e-05 ***
## Strikeout    0.0003461  0.0001044   3.315  0.00219 ** 
## ---
## 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
```

From F-statistic, we got 12.41 on 5 and 34 DF and the p-value:6.856e-07 which is less than 0.05.So that the null hypothesis is rejected. Therefore there is a significant relation with win and predictors.So, the linear regression model is significant. Moreover, the value of Multiple R-squared: 0.646 and Adjusted R-squared: 0.594 values are pretty close which indicates a good liner relation with predicors.

Problem 3

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

Answer:

Coefficients with t-values are mentioned below:

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2776752 0.1913151 -1.451 0.15583
Runs 0.0002778 0.0001466 1.895 0.06659 .
Batting 1.7419995 0.9284706 1.876 0.06923 .
Doubleplays 0.0007370 0.0004502 1.637 0.11084
Walk -0.0005897 0.0001292 -4.566 6.23e-05 * **Strikeout 0.0003461 0.0001044 3.315 0.00219**

From the the t-tests results as listed above for each predictor, it can be concluded that the predictr 'Walk' and 'Strikeout' coefficients has produced the p-value less than 0.05. So that, there is a significant relation with predictor 'Walk' and 'Strikeout' with the 'Win'. For other predictor shows non significant relation with 'Win' as the p-values are greater than 0.05.

Problem 4

4. Give the 95% confidence interval (CI) for the wins for the year=1965, 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.)

Answer:

```
# Confidence intervals (by default 95%)
confint(baseball.fit1, level= 0.95 ) # confidence interval for full model's c
oefficient

##                     2.5 %        97.5 %
## (Intercept) -6.664742e-01  0.1111238100
## Runs        -2.008678e-05  0.0005757493
## Batting     -1.448798e-01  3.6288787395
## Doubleplays -1.779084e-04  0.0016519497
## Walk        -8.521836e-04 -0.0003272264
## Strikeout    1.339200e-04  0.0005583025
```

```
# confidence interval for a specific model's coefficient:
new <- data.frame(Year=1965, Runs= 707,Batting = 0.254,Doubleplays=152,Walk=4
67,Strikeout = 916)

pred.win1.clim<-predict(baseball.fit1, newdata = new, se.fit = TRUE, interval
= "confidence")
pred.win1.clim$fit

##         fit       lwr       upr
## 1 0.5148921 0.4904402 0.5393441

#prediction interval for a specific set of data:
pred.win1.plim<-predict(baseball.fit1, newdata = new, se.fit = TRUE, interval
= "prediction")
pred.win1.plim$fit

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

With 95% confidence interval, we can be 95% confident that the confidence interval contains the population mean for the specified values of the variables in the model where the lower limit is 0.4904402 and the upper limit is 0.5393441.

Problem 5

5. Give the 95% prediction interval (PI) for the wins for the year=1965, 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.)

Answer:

```
#prediction interval for a specific set of data:
pred.win1.plim<-predict(baseball.fit1, newdata = new, se.fit = TRUE, interval
= "prediction")
pred.win1.plim$fit

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

With 95% prediction bands, the prediction interval indicates that we can be 95% confident that the actual value will be between approximately 0.4210802 and 0.6087041.

Problem 6

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

Answer:

From the results, we can see that the prediction interval is wider than the corresponding confidence interval. The reason is, prediction intervals must account for both the uncertainty in estimating the population mean, plus the random variation of the individual values. As we have tried to predict where a new observation will be & that new observation has an additional standard deviation of the error term. Confidence intervals are based on only the fitted values & do not involve making a prediction. It represents the uncertainty in the "fitted' value. So, a prediction interval is always wider than a confidence interval. Also, the prediction interval will not converge to a single value as the sample size increases.

Problem 7

6. Let lag=15, calculate the Sample ACF, the corresponding Z-Statistic, and Ljung-Box Statistic. The output should be similar to the Table 2.3 on page 71 in the textbook. The values of ACF, Z-statistic, and Ljung-Box statistic are needed. Is there an indication of non-stationary behavior in the residuals?

Answer:

```
ACF<-acf(baseball.data[,3],lag.max=15,type="correlation", plot =FALSE)
QLB <- 0
for (K in 0:16){
  T <- dim(baseball.data)[1]
  QLB[K] <- T*(T+2)*sum((1/(T-1:K))*(ACF$acf[2:(K+1)]^2))
}
QLB.mat <- as.matrix(QLB)

Table <- as.data.frame(ACF$lag)
Table$Lag <- ACF$acf

z.test <- ACF$acf[0:16]
z.test.mat <-as.matrix(z.test)

#colnames(Table)
colnames(Table) <- c("Lag", "ACF")
#colnames(Table)
Table$LjungTest <- QLB.mat[,1]
Table$Ztest <- z.test.mat[,1]
Table

##    Lag          ACF LjungTest        Ztest
## 1    0  1.000000000  5.469855  1.000000000
## 2    1  0.356340819  5.527077  0.356340819
## 3    2  0.035976390  6.920893  0.035976390
## 4    3 -0.175206014 10.795273 -0.175206014
## 5    4 -0.288136126 14.405634 -0.288136126
## 6    5 -0.274255101 19.137283 -0.274255101
## 7    6 -0.309450420 21.426364 -0.309450420
## 8    7 -0.212047346 21.615435 -0.212047346
## 9    8 -0.060011323 24.042088 -0.060011323
## 10   9  0.211607012 52.528030  0.211607012
## 11  10  0.713216330 54.976969  0.713216330
## 12  11  0.205604831 54.977899  0.205604831
## 13  12 -0.003936801 56.173014 -0.003936801
## 14  13 -0.138590090 58.973763 -0.138590090
## 15  14 -0.208194418 61.314352 -0.208194418
## 16  15 -0.186628504        NA -0.186628504
```

As from the table, we have seen that, time series can be said as weakly stationary.

Problem 8

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

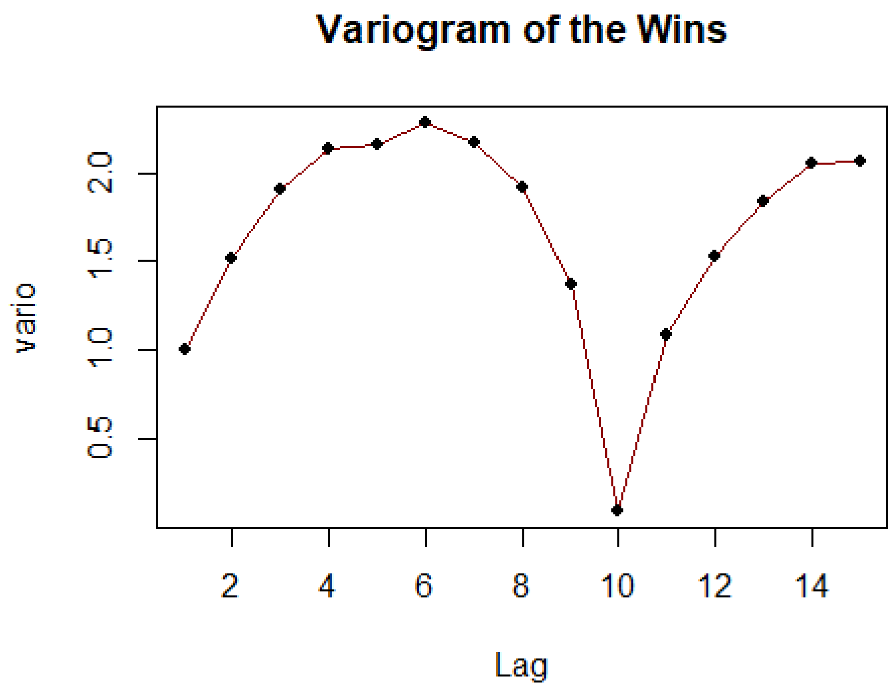
Answer:

```
#Defining variogram function
variogram <- function(x, lag){
  Lag <- NULL
vark <- NULL
vario <- NULL
for (k in 1:lag){
Lag[k] <- k
vark[k] = sd(diff(x,k))^2
vario[k] = vark[k]/vark[1]
}
return(as.data.frame(cbind(Lag,vario)))
}

#Variogram of  original baseball data
y_win <- as.matrix(baseball.data[, 3])
lag_length <-15
lag_readings <- 1:lag_length
vario_win <- variogram(y_win, lag_length)
vario_win
```

```
##    Lag      vario
## 1    1 1.00000000
## 2    2 1.51762654
## 3    3 1.90029081
## 4    4 2.13129676
## 5    5 2.15802734
## 6    6 2.28100369
## 7    7 2.16245512
## 8    8 1.91762196
## 9    9 1.36922530
## 10  10 0.08956979
## 11  11 1.08501226
## 12  12 1.52092217
## 13  13 1.83479333
## 14  14 2.05105860
## 15  15 2.06550364
```

```
plot(vario_win, type="o",col = 'dark red', main="Variogram of the Wins", pch=
19,cex=.8 )
points(vario_win ,pch=16,cex=.8)
```



Interpretation: As from the variogram data, we may say that variogram values varies almost around a constant number except 10th observation. So we can say, the data is weakly stationary.

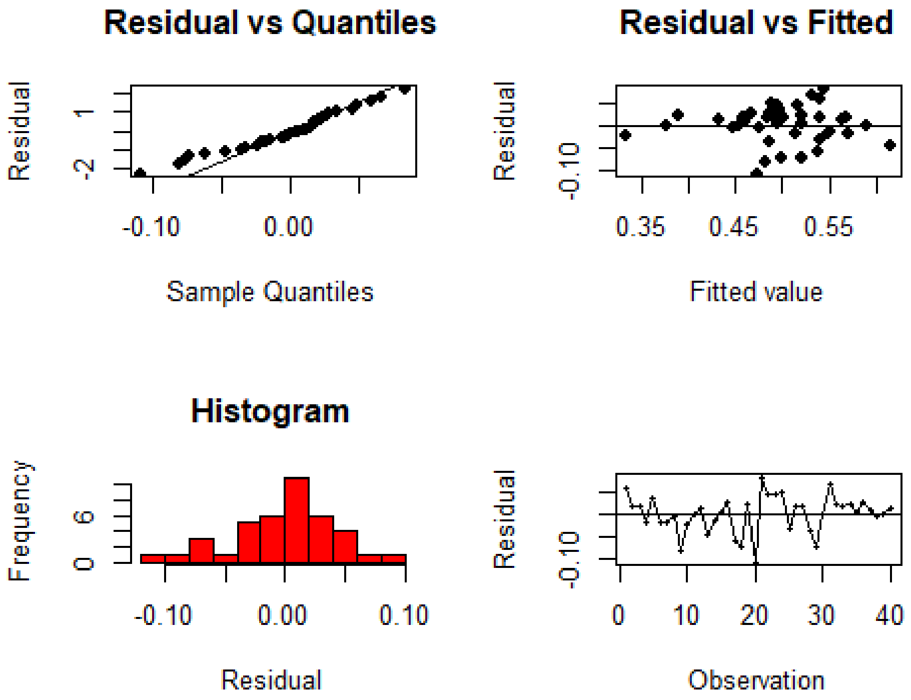
Problem 9

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.

Answer:

```
par(mfrow=c(2,2), oma = c(0,0,0,0))
qqnorm(baseball.fit1$res, datax = TRUE,pch=16, xlab='Residual',main='Residual
vs Quantiles')
qqline(baseball.fit1$res, datax = TRUE)
plot(baseball.fit1$fit, baseball.fit1$res, pch =16, xlab='Fitted value', ylab
='Residual', main = 'Residual vs Fitted')
```

```
abline(h=0)
hist(baseball.fit1$res,col='red', xlab = 'Residual', main='Histogram')
plot(baseball.fit1$res, type='l',xlab='Observation', ylab = 'Residual' )
points(baseball.fit1$res, pch=16, cex=0.5)
abline(h=0)
```



```
#part(a): normal probability plot from which we are looking for a straight li
ne
#part(c): histogram: Looking for a normal shape
#part(b):Residual vs Fitted: used to detect nonlinearity  between the variabl
es
#part(d): residuals vs observations order: where we are looking for no patter
ns
```

Interpretation: From the above plot of sample quantiles, we can see that the residuals follow normal distribution. Similarly,scatter plot of residuals are described by fitted value and observation order plot. Histogram also shows that, residuals are normally distributed. From the Residual vs Fitted curve: we can say that there is no linearity between the variables. Furthermore, the residual plot above shows the residuals' distribution, noramlity and skewneses of model.Also, from the residuals vs observations order: we have seen that there is no patterns.

Problem 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 criterions given in the textbook.

```r
library(styler)
nrow <- dim(baseball_data)[1]
baseball.mat <- as.matrix(baseball_data)

baseball.residual <- cbind(matrix(1, nrow, 1), baseball.mat[, 3:8])
res.fun <- function(data) {
data <- as.matrix(data)
n <- nrow(data)
p <- ncol(data) - 1
X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variables
hat.mat <- X %*% solve(t(X) %*% X) %*% t(X) # hat matrix
y <- data[, ncol(data)] # Response variable / dependant variable
reg <- lm(y ~ X) # Linear regression model
sum.reg <- summary(reg)
CI.coef <- confint(reg, level = .95)
e.i <- reg$residuals # Residuals
sigma.hat <- sigma(reg) # estimator of standard deviation/ sqrt(MSE)
d.i <- e.i / sigma.hat # d_i / Standardized Residuals
h.ii <- diag(hat.mat)
r.i <- e.i / sigma.hat / sqrt(1 - h.ii) # r_i / Studentized residuals
PRESS <- sum((e.i / (1 - h.ii))^2) # Prediction Error Sum of Squares
anova.sat <- anova(reg) # ANOVA to obtain SST, SSE.
R.2.pred <- 1 - PRESS / sum(anova.sat$`Sum Sq`) # R squared for prediction
S.2.i <- (((n - p) * anova.sat$`Mean Sq`[2]) - e.i^2 / (1 - h.ii)) / (n - p - 1)
t.i <- e.i / sqrt(S.2.i * (1 - h.ii)) # R-Student
index.leverage <- which(h.ii > 2 * p / n) # Return index of the high leverage observations
high.leverage <- h.ii[h.ii > 2 * p / n] # Return the h.ii values for high leverage
cook.dis <- cooks.distance(reg) # Return the cook's distance
index.inf <- which(cook.dis > 1)
inf.out <- cook.dis[cook.dis > 1]
Sati <- as.data.frame(cbind(
round(e.i, 3), # Combine all statistics into a data frame
round(r.i, 3),
round(t.i, 3),
round(h.ii, 3),
round(cook.dis, 3)
))
names(Sati) <- c(
"Residuals",
"Studentized Residuals", "R-Student", "h_ii", "Cook's Distance"
)#
print(cbind(e.i, r.i, t.i, h.ii,cook.dis))
names(PRESS) <- c("PRESS")
names(R.2.pred) <- c("Prediction R-squared")
leverage <- as.data.frame(cbind(index.leverage, high.leverage))
names(leverage) <- c("High-leverage-index", "h_ii")
influ <- as.data.frame(cbind(index.inf, inf.out))
names(influ) <- c("Influential-outlier-index", "cook's distance")
mylist <- list(sum.reg$coefficients, Sati, PRESS, R.2.pred, leverage, influ,
CI.coef)
return(mylist)
}
res.fun(baseball.residual)
```

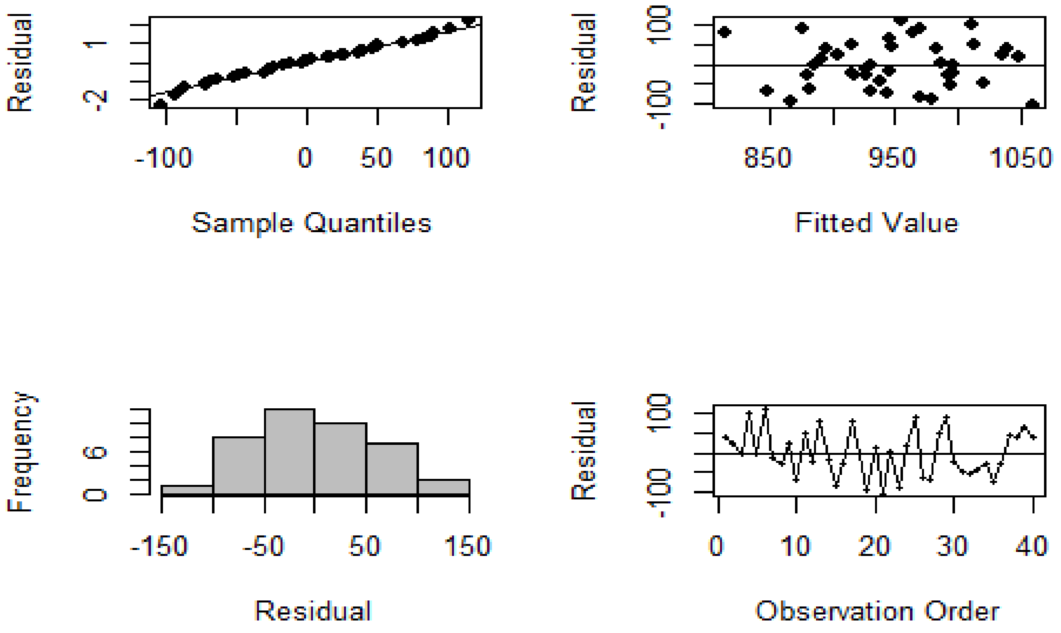
```
##              e.i          r.i         t.i       h.ii     cook.dis
## 1    40.4580587  0.685443181  0.68000259 0.13968130 1.271366e-02
## 2    25.0827968  0.416725590  0.41160405 0.10536933 3.408941e-03
## 3    -3.0471655 -0.052362894 -0.05158919 0.16374830 8.948203e-05
## 4   102.2367333  1.868597441  1.94340540 0.26077843 2.052945e-01
## 5     0.1353632  0.002274305  0.00224061 0.12522602 1.234084e-07
## 6   115.4565125  1.855065143  1.92774001 0.04344324 2.604825e-02
## 7   -11.6870880 -0.191124512 -0.18839411 0.07663788 5.053043e-04
## 8   -25.6038146 -0.434901485 -0.42965488 0.14410642 5.307550e-03
## 9    26.7856901  0.462114980  0.45670497 0.17034451 7.307675e-03
## 10  -71.4331764 -1.301750227 -1.31566936 0.25640441 9.738516e-02
## 11   50.8462144  0.871779989  0.86862707 0.15996833 2.412131e-02
## 12  -23.6813123 -0.400944307 -0.39594120 0.13853898 4.308769e-03
## 13   83.2067069  1.718748044  1.77201349 0.42125875 3.583750e-01
## 14  -17.0714724 -0.279534960 -0.27571048 0.07899518 1.117017e-03
## 15  -86.3611399 -1.474711857 -1.50168347 0.15313420 6.554215e-02
## 16  -25.7724253 -0.424538747 -0.41936193 0.08994400 2.968845e-03
## 17   79.3537350  1.311574589  1.32612583 0.09605717 3.046657e-02
## 18   -2.0467935 -0.036518904 -0.03597856 0.22428036 6.426443e-05
## 19  -93.7982039 -1.584523416 -1.62208731 0.13466569 6.512071e-02
## 20   15.8432570  0.290408653  0.28646157 0.26504283 5.068995e-03
## 21 -103.4168311 -1.756043874 -1.81425574 0.14354622 8.614048e-02
## 22    3.9642409  0.065663184  0.06469444 0.09994637 7.979786e-05
## 23  -90.9314953 -1.489238623 -1.51750279 0.07935481 3.186092e-02
## 24   17.1445847  0.293504171  0.28952273 0.15740751 2.682163e-03
## 25   90.6142151  1.532779012  1.56511272 0.13697022 6.214529e-02
## 26  -62.7692631 -1.162383050 -1.16861697 0.27990843 8.753375e-02
## 27  -68.3259793 -1.113930523 -1.11801776 0.07093152 1.578907e-02
## 28   50.4665097  0.827711059  0.82378998 0.08200307 1.019988e-02
## 29   89.8183295  1.513606862  1.54411209 0.13044717 5.728140e-02
## 30  -22.9215245 -0.401911244 -0.39690063 0.19680890 6.596824e-03
## 31  -49.1103586 -0.829445895 -0.82555222 0.13431022 1.778982e-02
## 32  -51.5853725 -0.866810683 -0.86356352 0.12542545 1.795916e-02
## 33  -44.0454294 -0.716678624 -0.71145490 0.06729351 6.176264e-03
## 34  -28.9117066 -0.519086719 -0.51343466 0.23394522 1.371459e-02
## 35  -72.4931969 -1.312574503 -1.32719078 0.24675062 9.406242e-02
## 36  -29.8207564 -0.488271165 -0.48273256 0.07889878 3.403563e-03
## 37   47.3904354  0.796845334  0.79247432 0.12657517 1.533627e-02
## 38   39.5796203  0.656393981  0.65080579 0.10214486 8.169363e-03
## 39   68.6738465  1.152005183  1.15775767 0.12246169 3.086681e-02
## 40   37.7776555  0.639127253  0.63347503 0.13724490 1.083009e-02
```

```
## [[1]]
##                                                        Estimate   Std. Err
or
## (Intercept)                                         945.1170618  230.17135
06
## Xwin: The team's winning percentage                 705.6894545  212.88713
77
## Xruns: The number of runs scored by the team         -0.1285373    0.21899
70
## Xba: The team's overall batting average            -459.3582375 1390.47746
27
## Xdp: The total number of double plays                -2.2737340    0.54201
22
## Xwalk: The number of walks given to the other team    0.3961517    0.22416
58
##                                                       t value     Pr(>|t|)
## (Intercept)                                         4.1061455 0.0002386243
## Xwin: The team's winning percentage                 3.3148525 0.0021868785
## Xruns: The number of runs scored by the team       -0.5869365 0.5611237844
## Xba: The team's overall batting average            -0.3303601 0.7431550224
## Xdp: The total number of double plays              -4.1949867 0.0001845368
## Xwalk: The number of walks given to the other team  1.7672259 0.0861660438
## 
## [[2]]
##    Residuals Studentized Residuals R-Student  h_ii Cook's Distance
## 1     40.458                 0.685     0.680 0.140           0.013
## 2     25.083                 0.417     0.412 0.105           0.003
## 3     -3.047                -0.052    -0.052 0.164           0.000
## 4    102.237                 1.869     1.943 0.261           0.205
## 5      0.135                 0.002     0.002 0.125           0.000
## 6    115.457                 1.855     1.928 0.043           0.026
## 7    -11.687                -0.191    -0.188 0.077           0.001
## 8    -25.604                -0.435    -0.430 0.144           0.005
## 9     26.786                 0.462     0.457 0.170           0.007
## 10   -71.433                -1.302    -1.316 0.256           0.097
## 11    50.846                 0.872     0.869 0.160           0.024
## 12   -23.681                -0.401    -0.396 0.139           0.004
## 13    83.207                 1.719     1.772 0.421           0.358
## 14   -17.071                -0.280    -0.276 0.079           0.001
## 15   -86.361                -1.475    -1.502 0.153           0.066
## 16   -25.772                -0.425    -0.419 0.090           0.003
## 17    79.354                 1.312     1.326 0.096           0.030
## 18    -2.047                -0.037    -0.036 0.224           0.000
## 19   -93.798                -1.585    -1.622 0.135           0.065
## 20    15.843                 0.290     0.286 0.265           0.005
## 21  -103.417                -1.756    -1.814 0.144           0.086
## 22     3.964                 0.066     0.065 0.100           0.000
## 23   -90.931                -1.489    -1.518 0.079           0.032
## 24    17.145                 0.294     0.290 0.157           0.003
## 25    90.614                 1.533     1.565 0.137           0.062
## 26   -62.769                -1.162    -1.169 0.280           0.088
## 27   -68.326                -1.114    -1.118 0.071           0.016
## 28    50.467                 0.828     0.824 0.082           0.010
## 29    89.818                 1.514     1.544 0.130           0.057
## 30   -22.922                -0.402    -0.397 0.197           0.007
## 31   -49.110                -0.829    -0.826 0.134           0.018
## 32   -51.585                -0.867    -0.864 0.125           0.018
## 33   -44.045                -0.717    -0.711 0.067           0.006
## 34   -28.912                -0.519    -0.513 0.234           0.014
## 35   -72.493                -1.313    -1.327 0.247           0.094
## 36   -29.821                -0.488    -0.483 0.079           0.003
## 37    47.390                 0.797     0.792 0.127           0.015
## 38    39.580                 0.656     0.651 0.102           0.008
## 39    68.674                 1.152     1.158 0.122           0.031
## 40    37.778                 0.639     0.633 0.137           0.011
## 
## [[3]]
##    PRESS 
## 201492.5 
## 
## [[4]]
## Prediction R-squared 
##            0.2466781 
## 
## [[5]]
##   High-leverage-index      h_ii
## 1                  13 0.4212587
## 
## [[6]]
## [1] Influential-outlier-index cook's distance         
## <0 rows> (or 0-length row.names)
## 
## [[7]]
##                                                             2.5 %       97.
5 %
## (Intercept)                                         4.773526e+02 1412.8815
252
## X                                                              NA         
NA
## Xwin: The team's winning percentage                 2.730507e+02 1138.3281
711
## Xruns: The number of runs scored by the team       -5.735928e-01    0.3165
181
```

```
## Xba: The team's overall batting average            -3.285148e+03 2366.4319
514
## Xdp: The total number of double plays              -3.375235e+00   -1.1722
327
## Xwalk: The number of walks given to the other team -5.940811e-02    0.8517
114

# Residual Plots.
# To use this function, make sure your data satisfies the following condition
s:
# 1. The first column is 1's vector.
# 2. The last column is the response variable (y).
# 3. The other columns are independant variables (X) which are involved in th
e regression model.
resid.plot <- function(data) {
data <- as.matrix(data)
X <- data[, -ncol(data)] # Predictors/ regressor matrix / independant variabl
es
y <- data[, ncol(data)] # Response variable / dependant variable
fit.data <- lm(y ~ X)
par(mfrow = c(2, 2), oma = c(0, 0, 0, 0))
qqnorm(fit.data$residuals, datax = T, pch = 16, xlab = "Residual", main = "")
qqline(fit.data$residuals, data = T)
plot(fit.data$fitted.values, fit.data$residuals, pch = 16, xlab = "Fitted Val
ue", ylab = "Residual")
abline(h = 0)
hist(fit.data$residuals, col = "grey", xlab = "Residual", main = "")
plot(fit.data$residuals, type = "l", xlab = "Observation Order", ylab = "Resi
dual")
points(fit.data$residuals, pch = 16, cex = .5)
abline(h = 0)
}

win.mat <- as.matrix(baseball_data)
resid.plot(win.mat[ ,3:8])
```



Interpretation:

- From the above plot of sample quantiles, we can see that the residuals follow normal distribution.The residuals lie generally along a straight line, so there is no obvious reason to be concerned with the normality assumption.Any observation with a standardized residual outside this interval $-3 \le di \le 3$ is a potential outlier.From the table, we did not see any outliers.

- There is one high leverage point above 100th sample which can be defined as an extreme predictor X-values.Absolute values of the studentized residuals that are larger than three or four indicate potentially problematic observations.But form the table, we have seen no presence of such values.

- Most of the time, t_i will be closed to r_i. However, for influential observation, they can differ significantly. As we see that all of the values of ti are much closer to the values of ri. So, there is no influential points.

The Normal probability plots of residuals does not violate the normality assumption. Here, the value of Prediction R-squared is 0.2466781. So the model would be adequate to explain 24.667% of the variability in the new data.

These outputs indicate that the model is not adequate.