Project IV

Isaiah Thompson Ocansey

04/01/2022

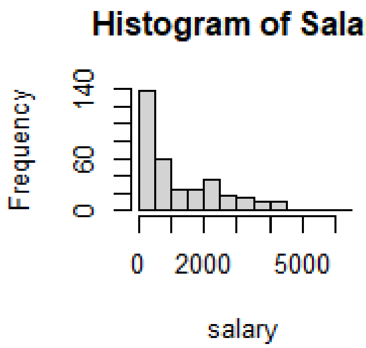
```
baseball <- read.table(file=
                         "http://jse.amstat.org/datasets/baseball.dat.txt",
                       header = F, col.names=c("salary", "batting.avg",
"OBP", "runs", "hits",
                                               "doubles", "triples",
"homeruns", "RBI", "walks", "strike.outs",
                                               "stolen.bases", "errors",
"free.agency.elig", "free.agent.91",
                                               "arb.elig", "arb.91", "name"))
head(baseball);dim(baseball)

##   salary batting.avg   OBP runs hits doubles triples homeruns RBI walks
## 1   3300       0.272 0.302   69  153      21       4       31 104    22
## 2   2600       0.269 0.335   58  111      17       2       18  66    39
## 3   2500       0.249 0.337   54  115      15       1       17  73    63
## 4   2475       0.260 0.292   59  128      22       7       12  50    23
## 5   2313       0.273 0.346   87  169      28       5        8  58    70
## 6   2175       0.291 0.379  104  170      32       2       26 100    87
##   strike.outs stolen.bases errors free.agency.elig free.agent.91 arb.elig
## 1          80            4      3                1             0        0
## 2          69            0      3                1             1        0
## 3         116            6      5                1             0        0
## 4          64           21     21                0             0        1
## 5          53            3      8                0             0        1
## 6          89           22      4                1             0        0
##   arb.91           name
## 1      0 Andre Dawson
## 2      0 Steve Buchele
## 3      0 Kal Daniels
## 4      0 Shawon Dunston
## 5      0 Mark Grace
## 6      0 Ryne Sandberg

## [1] 337  18
```

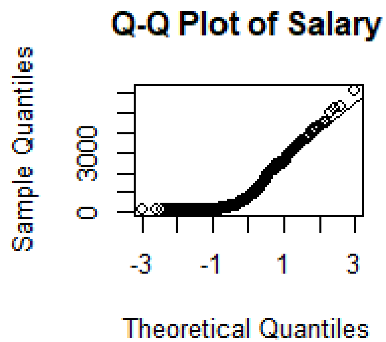
The data has 337 observations and 18 variables

EXPLORATORY DATA ANALYSIS (EDA)

```
par(mfrow=c(2,2),mar=c(4, 4, 4, 4))
hist(baseball$salary, xlab="salary", main="Histogram of Salary")
```

```
qqnorm(baseball$salary, main="Q-Q Plot of Salary")
qqline(baseball$salary)
```

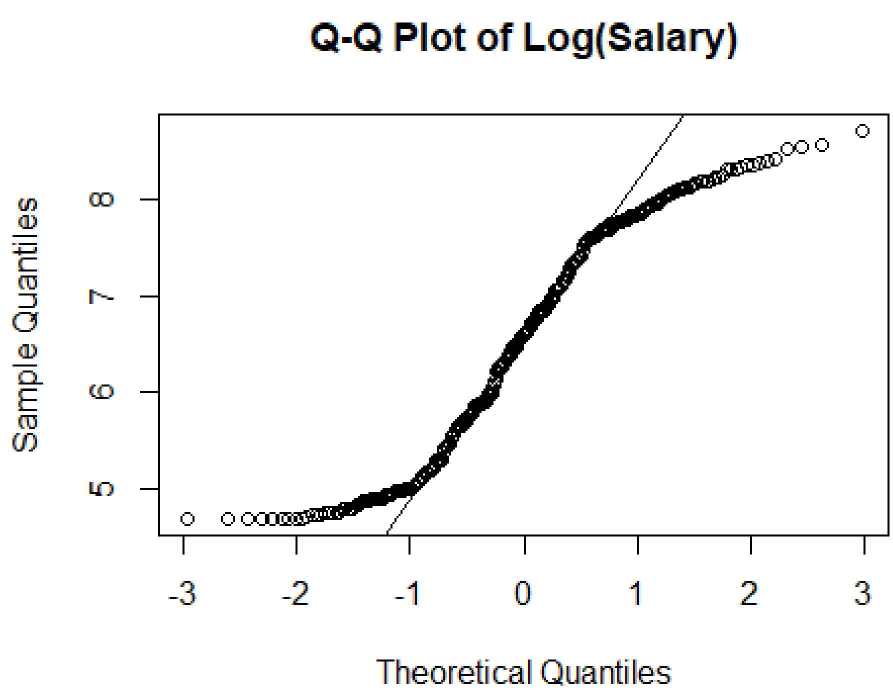




It is observed from the histogram above that the 'Salary' is skewed to the right, an indication of a violation of normality

```
logsalary<-log(baseball$salary)

hist(logsalary, xlab="log-salary", main="Histogram of Log(Salary)")
```


```
qqnorm(logsalary, main="Q-Q Plot of Log(Salary)")
qqline(logsalary)
```



The plot for log salary transformation does not appear to be normal.

```r
colMeans(is.na(baseball))
```

```
##           salary      batting.avg              OBP             runs 
##                0                0                0                0 
##             hits          doubles          triples         homeruns 
##                0                0                0                0 
##              RBI            walks      strike.outs     stolen.bases 
##                0                0                0                0 
##           errors free.agency.elig    free.agent.91         arb.elig 
##                0                0                0                0 
##           arb.91             name 
##                0                0
```

From the output above, There are no missing values in the data.

```r
str(baseball)
```

```
## 'data.frame':    337 obs. of  18 variables:
##  $ salary          : int  3300 2600 2500 2475 2313 2175 600 460 240 200 
...
##  $ batting.avg     : num  0.272 0.269 0.249 0.26 0.273 0.291 0.258 0.228 
0.25 0.203 ...
##  $ OBP             : num  0.302 0.335 0.337 0.292 0.346 0.379 0.37 0.279 
0.327 0.24 ...
##  $ runs            : int  69 58 54 59 87 104 34 16 40 39 ...
##  $ hits            : int  153 111 115 128 169 170 86 38 61 64 ...
##  $ doubles         : int  21 17 15 22 28 32 14 7 11 10 ...
##  $ triples         : int  4 2 1 7 5 2 1 2 0 1 ...
##  $ homeruns        : int  31 18 17 12 8 26 14 3 1 10 ...
##  $ RBI             : int  104 66 73 50 58 100 38 21 18 33 ...
##  $ walks           : int  22 39 63 23 70 87 15 11 24 14 ...
##  $ strike.outs     : int  80 69 116 64 53 89 45 32 26 96 ...
##  $ stolen.bases    : int  4 0 6 21 3 22 0 2 14 13 ...
##  $ errors          : int  3 3 5 21 8 4 10 3 2 6 ...
##  $ free.agency.elig: int  1 1 1 0 0 1 1 0 0 0 ...
##  $ free.agent.91   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ arb.elig        : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ arb.91          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ name            : chr  "Andre Dawson     " "Steve Buchele    " "Kal 
Daniels      " "Shawon Dunston   " ...
```

```r
#checking for variable types in the data
# number of continuous variables
print("Number of continuous variables")
```

```
## [1] "Number of continuous variables"
```

```r
sum(sapply(baseball, FUN = is.double))
```

```
## [1] 2
```

```
# number of integer variables
print("Number of integer variables")

## [1] "Number of integer variables"

sum(sapply(baseball, FUN = is.integer))

## [1] 15

# number of categorical variables
print("Number of categorical variables")

## [1] "Number of categorical variables"

sum(sapply(baseball, FUN = is.factor))

## [1] 0
```

From the output above, the number of continuous variables are 2, number of interger counts are 15 and 0 categorical variable.

Question 2a)

```
#Define variables
attach(baseball)
y<-log(baseball$salary)
x1<-batting.avg
x2<-OBP
x3<-runs
x4<-hits
x5<-doubles
x6<-triples
x7<-homeruns
x8<-RBI
x9<-walks
x10<-strike.outs
x11<-stolen.bases
x12<-errors
x13<-free.agency.elig
x14<-free.agent.91/2
x15<-arb.elig
x16<-arb.91/2
dat <- data.frame(y,x1=x1, x2=x2, x3=x3,
x4=x4,x5=x5,x6=x6,x7=x7,x8=x8,x9=x9,x10=x10,x11=x11,x12=x12,x13=x13,x14=x14,x
15=x15,x16=x16)
detach()
dim(dat)

## [1] 337  17
```

The dataset for the defined variables has 337 observations and 17

Linear Model with Variable Selection

```
#Train-Test-Split
dt = sort(sample(nrow(dat), nrow(dat)*.667))
D<-dat[dt,]#Training Data
D_prime<-dat[-dt,] #Test Data
dim(D);dim(D_prime)

## [1] 224  17

## [1] 113  17
```

We randomly Split the data into train and test data in the ration 2:1 respectively.

```
#Full Model with all variables included
fit.full <- lm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16 -1,
data = D)
BIC(fit.full)

## [1] 603.4976

summary(fit.full)

##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11 + x12 + x13 + x14 + x15 + x16 - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5609 -0.3522  0.0322  0.4747  4.3556
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x1   3.1850509  3.9095786   0.815 0.416188
## x2  12.2948188  3.1268517   3.932 0.000115 ***
## x3  -0.0004157  0.0077882  -0.053 0.957480
## x4   0.0031144  0.0045592   0.683 0.495301
## x5  -0.0093956  0.0119953  -0.783 0.434360
## x6  -0.0093050  0.0325211  -0.286 0.775069
## x7  -0.0257143  0.0167054  -1.539 0.125256
## x8   0.0165185  0.0069663   2.371 0.018643 *
## x9  -0.0144418  0.0061555  -2.346 0.019909 *
## x10  0.0085196  0.0027036   3.151 0.001866 **
## x11  0.0004103  0.0070549   0.058 0.953677
## x12  0.0059261  0.0102827   0.576 0.565023
## x13  1.9206298  0.1442063  13.319  < 2e-16 ***
## x14 -0.4812287  0.3808856  -1.263 0.207843
## x15  1.5826088  0.1666278   9.498  < 2e-16 ***
## x16 -0.7105340  0.7072022  -1.005 0.316203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

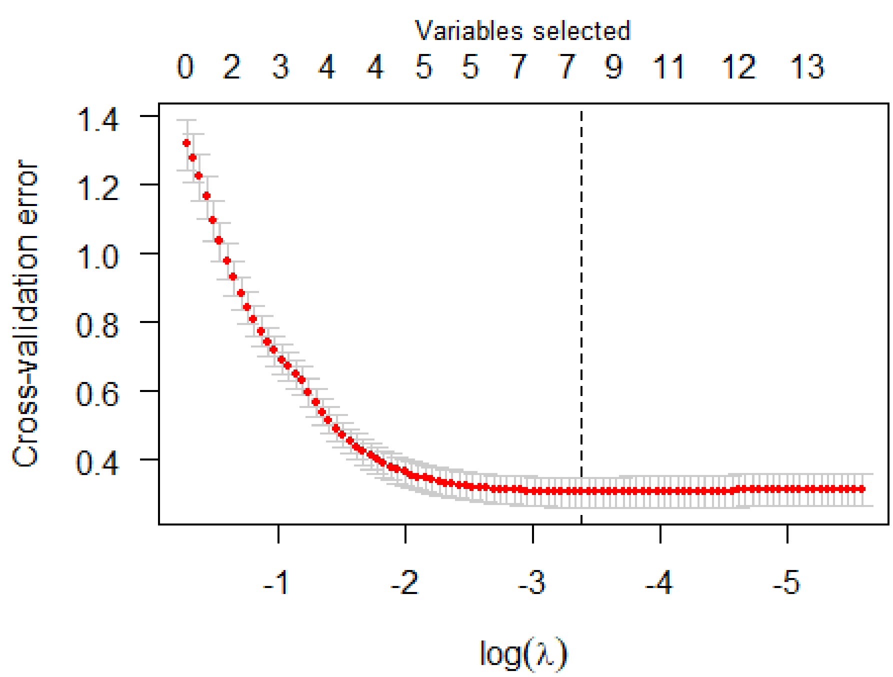
```
##
## Residual standard error: 0.7865 on 208 degrees of freedom
## Multiple R-squared:  0.987,  Adjusted R-squared:  0.986
## F-statistic: 987.7 on 16 and 208 DF,  p-value: < 2.2e-16
```

From the above, the full model is found to be statistically significant given the F-Values and P-values from the above with an R-Square of 98.38%. The variables x2,x8,x9,x10,x13 and x15 were found to be statistically significant with p_values <0.05.

2(b) Using the training data D, apply three variable selection methods of your choice and identify your 'best' models accordingly

Method One (LASSO)

```
library(ncvreg)
formula0 <- y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16 -1
y <- D[, all.vars(formula0)[1]]
X <- as.data.frame(model.matrix(as.formula(formula0),D))
#10-FOLD CV FOR SELECTING THE TUNING PARAMETER
cvfit.L1 <- cv.ncvreg(X=X,y=y, nfolds=10, family="gaussian",
    penalty="lasso", lambda.min=.005, nlambda=100, eps=.001, max.iter=1000)
plot(cvfit.L1)
```




```
names(cvfit.L1)

## [1] "cve"         "cvse"        "fold"        "lambda"      "fit"
## [6] "min"         "lambda.min"  "null.dev"    "Bias"
```

```r
beta.hat <- coef(cvfit.L1)  # THE LASSO COEFFICIENTS WITH MINIMUM CV ERROR
```

We used the Least absolute shrinkage and selection operator (LASSO) method . From the plot above, one model was found to be the best model with 11 variables.

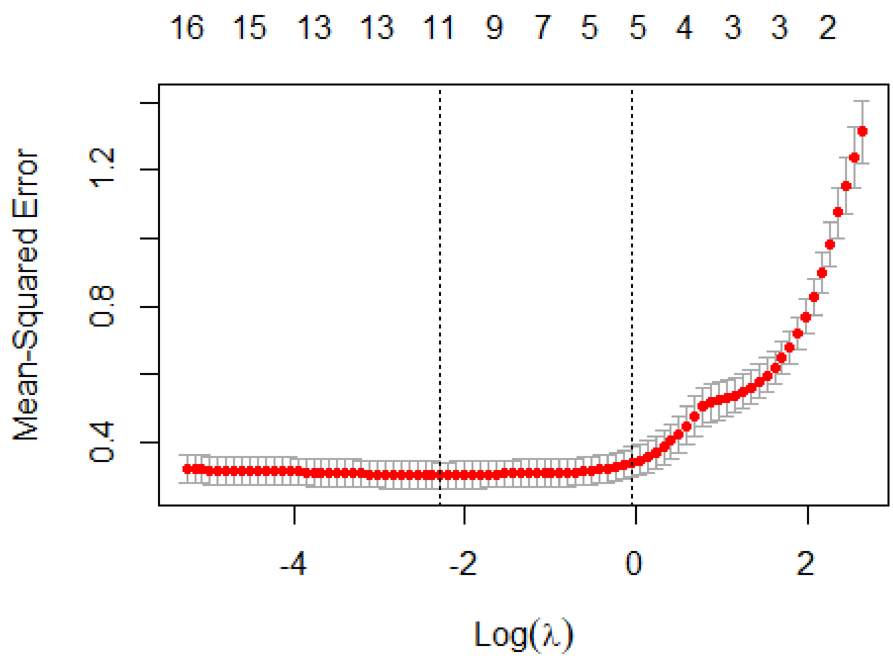
```r
 #NEXT, WE REFIT THE MODEL USING OLS WITH VARIABLES SELECTED BY LASSO
cutoff <- 0.0001
terms <- names(beta.hat)[abs(beta.hat) > cutoff]
formula.LASSO <- as.formula(paste(c("y ~ ", terms[-1]), collapse=" + "))
fit.L1 <- lm(formula.LASSO, data = D)
summary(fit.L1)

## 
## Call:
## lm(formula = formula.LASSO, data = D)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.61872 -0.26900 -0.04758  0.32423  1.18154 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  4.7991847  0.0789086  60.820  < 2e-16 ***
## x3          -0.0006776  0.0039227  -0.173  0.86302    
## x4           0.0055774  0.0019683   2.834  0.00504 ** 
## x8           0.0074080  0.0023865   3.104  0.00216 ** 
## x9           0.0009429  0.0026751   0.352  0.72484    
## x13          1.6405950  0.0943939  17.380  < 2e-16 ***
## x14         -0.6364802  0.2486074  -2.560  0.01114 *  
## x15          1.2902001  0.1065728  12.106  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5315 on 216 degrees of freedom
## Multiple R-squared:  0.7926, Adjusted R-squared:  0.7859 
## F-statistic: 117.9 on 7 and 216 DF,  p-value: < 2.2e-16
```

After we applied LASSO to the training data, the variables; intercept, x8,x10,x13,x14 and x15 are found to be statistically significant with an Adjusted R-squared value of 75.97%.

Method Two (Adaptive LASSO)

```r
set.seed(125)
library(MESS)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-3
```

```
wt <- adaptive.weights(x=X, y=y, weight.method="univariate")
cv.ALASSO <- cv.glmnet(x=as.matrix(X), y=y, family="gaussian", alpha=1,
nlambda=100,
                       penalty.factor=as.numeric(wt$weights),
standardize=FALSE)
plot(cv.ALASSO)
```




```
beta.hat.alasso <- coef(cv.ALASSO, s="lambda.1se")
```

From the plot for the adaptive LASSO,two models were found to be the best;one with 11 variables and the other with 5 variables.However, due to the law of parsimony, we choose the model with 5 variables.

```
# AGAIN, LET'S FIT OLS MODEL WITH ALASSO SELECTED VARIABLES
beta.hat.alasso <- coef(cv.ALASSO, s="lambda.1se")
cutoff <- 0
terms3 <- names(X)[abs(as.vector(beta.hat.alasso[-1])) > cutoff]
formula.ALASSO <- as.formula(paste(c("y ~ ", terms3),
                                   collapse=" + "))
fit.ALASSO <- lm(formula.ALASSO, data =D)
summary(fit.ALASSO)

##
## Call:
## lm(formula = formula.ALASSO, data = D)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max 
## -2.51240 -0.26459 -0.03114  0.33379  1.21783 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.786076   0.078667  60.839   <2e-16 ***
## x3          0.001585   0.003206   0.494   0.6215    
## x4          0.004962   0.001942   2.555   0.0113 *  
## x8          0.007512   0.002390   3.144   0.0019 ** 
## x13         1.542587   0.085619  18.017   <2e-16 ***
## x15         1.275470   0.107318  11.885   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5374 on 218 degrees of freedom
## Multiple R-squared:  0.786,  Adjusted R-squared:  0.7811 
## F-statistic: 160.1 on 5 and 218 DF,  p-value: < 2.2e-16
```

After we apply LASSO to the training data D, the variables intercept,x8, x13 and x15 were found to be statistically significant. A decrease in the number of variables as compared to LASSO with an adjusted R-square to 74.74% as compared to LASSO.*

Method 3 (Stepwise Regression)

```
library(MASS)
fit.back <- stepAIC(fit.full, direction="backward", k=log(nrow(dat)))
```

```
## Start:  AIC=-31.06
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + 
##     x12 + x13 + x14 + x15 + x16 - 1
## 
##        Df Sum of Sq    RSS     AIC
## - x3    1     0.002 128.67 -36.881
## - x11   1     0.002 128.67 -36.880
## - x6    1     0.051 128.72 -36.795
## - x12   1     0.205 128.88 -36.526
## - x4    1     0.289 128.96 -36.382
## - x5    1     0.380 129.05 -36.224
## - x1    1     0.411 129.08 -36.170
## - x16   1     0.624 129.29 -35.799
## - x14   1     0.987 129.66 -35.171
## - x7    1     1.466 130.13 -34.346
## <none>              128.67 -31.064
## - x9    1     3.405 132.07 -31.033
## - x8    1     3.478 132.15 -30.909
## - x10   1     6.143 134.81 -26.437
## - x2    1     9.564 138.23 -20.823
## - x15   1    55.804 184.47  43.814
## - x13   1   109.732 238.40 101.259
## 
```

```
## Step:  AIC=-36.88
## y ~ x1 + x2 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 +
##     x13 + x14 + x15 + x16 - 1
##
##        Df Sum of Sq    RSS     AIC
## - x11   1     0.001 128.67 -42.699
## - x6    1     0.060 128.73 -42.597
## - x12   1     0.204 128.88 -42.346
## - x4    1     0.340 129.01 -42.110
## - x5    1     0.402 129.07 -42.002
## - x1    1     0.414 129.09 -41.982
## - x16   1     0.644 129.31 -41.583
## - x14   1     0.988 129.66 -40.988
## - x7    1     1.853 130.52 -39.498
## <none>              128.67 -36.881
## - x8    1     3.580 132.25 -36.553
## - x9    1     4.258 132.93 -35.408
## - x10   1     6.511 135.18 -31.643
## - x2    1     9.572 138.24 -26.628
## - x15   1    55.803 184.47  37.995
## - x13   1   109.937 238.61  95.633
##
## Step:  AIC=-42.7
## y ~ x1 + x2 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x12 + x13 +
##     x14 + x15 + x16 - 1
##
##        Df Sum of Sq    RSS     AIC
## - x6    1     0.065 128.74 -48.407
## - x12   1     0.206 128.88 -48.160
## - x4    1     0.355 129.03 -47.902
## - x5    1     0.405 129.08 -47.816
## - x1    1     0.413 129.09 -47.802
## - x16   1     0.653 129.32 -47.386
## - x14   1     1.014 129.69 -46.761
## - x7    1     1.853 130.53 -45.317
## <none>              128.67 -42.699
## - x8    1     3.684 132.36 -42.196
## - x9    1     4.268 132.94 -41.210
## - x10   1     6.586 135.26 -37.338
## - x2    1     9.591 138.26 -32.415
## - x15   1    55.903 184.57  32.297
## - x13   1   109.940 238.61  89.817
##
## Step:  AIC=-48.41
## y ~ x1 + x2 + x4 + x5 + x7 + x8 + x9 + x10 + x12 + x13 + x14 +
##     x15 + x16 - 1
##
##        Df Sum of Sq    RSS     AIC
## - x12   1     0.243 128.98 -53.804
## - x4    1     0.292 129.03 -53.720
```

```
## - x5    1     0.378 129.12 -53.570
## - x1    1     0.433 129.17 -53.474
## - x16   1     0.652 129.39 -53.095
## - x14   1     1.035 129.77 -52.432
## - x7    1     1.791 130.53 -51.133
## <none>               128.74 -48.407
## - x8    1     3.861 132.60 -47.607
## - x9    1     4.204 132.94 -47.029
## - x10   1     6.800 135.54 -42.696
## - x2    1     9.536 138.27 -38.219
## - x15   1    55.901 184.64  26.553
## - x13   1   112.090 240.83  86.066
##
## Step:  AIC=-53.8
## y ~ x1 + x2 + x4 + x5 + x7 + x8 + x9 + x10 + x13 + x14 + x15 +
##     x16 - 1
##
##        Df Sum of Sq    RSS     AIC
## - x1    1     0.418 129.40 -58.900
## - x4    1     0.442 129.42 -58.859
## - x5    1     0.450 129.43 -58.845
## - x16   1     0.639 129.62 -58.517
## - x14   1     0.910 129.89 -58.050
## - x7    1     1.991 130.97 -56.194
## <none>               128.98 -53.804
## - x8    1     3.908 132.89 -52.938
## - x9    1     4.401 133.38 -52.108
## - x10   1     7.818 136.80 -46.443
## - x2    1     9.639 138.62 -43.481
## - x15   1    58.047 187.03  23.613
## - x13   1   112.349 241.33  80.713
##
## Step:  AIC=-58.9
## y ~ x2 + x4 + x5 + x7 + x8 + x9 + x10 + x13 + x14 + x15 + x16 -
##     1
##
##        Df Sum of Sq    RSS    AIC
## - x5    1      0.48 129.88 -63.89
## - x16   1      0.67 130.06 -63.57
## - x14   1      0.91 130.30 -63.16
## - x4    1      1.38 130.78 -62.34
## - x7    1      1.90 131.30 -61.46
## <none>              129.40 -58.90
## - x8    1      3.94 133.34 -58.00
## - x10   1      7.44 136.84 -52.19
## - x9    1     16.84 146.24 -37.31
## - x15   1     57.68 187.08  17.86
## - x13   1    112.26 241.66  75.20
## - x2    1    840.12 969.52 386.40
##
```

```
## Step:  AIC=-63.89
## y ~ x2 + x4 + x7 + x8 + x9 + x10 + x13 + x14 + x15 + x16 - 1
##
##        Df Sum of Sq    RSS    AIC
## - x16   1      0.77 130.65 -68.38
## - x14   1      0.88 130.76 -68.20
## - x4    1      0.90 130.78 -68.15
## - x7    1      2.04 131.92 -66.22
## <none>              129.88 -63.89
## - x8    1      3.65 133.53 -63.50
## - x10   1      7.75 137.63 -56.73
## - x9    1     16.56 146.44 -42.82
## - x15   1     57.38 187.26  12.25
## - x13   1    112.62 242.50  70.16
## - x2    1    839.70 969.59 380.59
##
## Step:  AIC=-68.38
## y ~ x2 + x4 + x7 + x8 + x9 + x10 + x13 + x14 + x15 - 1
##
##        Df Sum of Sq    RSS    AIC
## - x4    1      0.80 131.45 -72.84
## - x14   1      0.89 131.54 -72.68
## - x7    1      2.16 132.81 -70.52
## <none>              130.65 -68.38
## - x8    1      3.81 134.46 -67.76
## - x10   1      8.02 138.68 -60.85
## - x9    1     16.49 147.14 -47.57
## - x15   1     58.26 188.91   8.40
## - x13   1    113.04 243.69  65.44
## - x2    1    839.40 970.05 374.88
##
## Step:  AIC=-72.84
## y ~ x2 + x7 + x8 + x9 + x10 + x13 + x14 + x15 - 1
##
##        Df Sum of Sq     RSS    AIC
## - x14   1      1.06  132.51 -76.86
## <none>               131.45 -72.84
## - x7    1      5.49  136.94 -69.48
## - x10   1      9.22  140.67 -63.48
## - x9    1     15.70  147.15 -53.39
## - x8    1     15.82  147.27 -53.20
## - x15   1     65.49  196.94  11.90
## - x13   1    119.54  250.98  66.22
## - x2    1    884.97 1016.42 379.52
##
## Step:  AIC=-76.86
## y ~ x2 + x7 + x8 + x9 + x10 + x13 + x15 - 1
##
##        Df Sum of Sq     RSS    AIC
## <none>               132.51 -76.86
```

```
## - x7    1      4.84  137.35 -74.64
## - x10   1      9.13  141.64 -67.76
## - x9    1     14.77  147.28 -59.01
## - x8    1     14.95  147.46 -58.73
## - x15   1     65.17  197.68   6.92
## - x13   1    137.62  270.13  76.86
## - x2    1    883.92 1016.42 373.70
```

```
fit.back$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 +
##     x12 + x13 + x14 + x15 + x16 - 1
##
## Final Model:
## y ~ x2 + x7 + x8 + x9 + x10 + x13 + x15 - 1
##
##
##     Step Df     Deviance Resid. Df Resid. Dev       AIC
## 1                              208   128.6698 -31.06355
## 2   - x3  1 0.0017626853       209   128.6715 -36.88056
## 3  - x11  1 0.0008674855       210   128.6724 -42.69913
## 4   - x6  1 0.0647127419       211   128.7371 -48.40659
## 5  - x12  1 0.2428987865       212   128.9800 -53.80443
## 6   - x1  1 0.4176282921       213   129.3977 -58.90039
## 7   - x5  1 0.4829350552       214   129.8806 -63.88602
## 8  - x16  1 0.7717201519       215   130.6523 -68.37909
## 9   - x4  1 0.7971067748       216   131.4494 -72.83670
## 10 - x14  1 1.0594876660       217   132.5089 -76.85857
```

```
summary(fit.back)
```

```
##
## Call:
## lm(formula = y ~ x2 + x7 + x8 + x9 + x10 + x13 + x15 - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5936 -0.3986  0.0588  0.5140  4.2163
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x2  14.898275   0.391583  38.046  < 2e-16 ***
## x7  -0.033887   0.012038  -2.815 0.005325 **
## x8   0.021328   0.004310   4.949 1.50e-06 ***
## x9  -0.016291   0.003312  -4.918 1.72e-06 ***
## x10  0.009076   0.002348   3.866 0.000146 ***
## x13  1.853139   0.123442  15.012  < 2e-16 ***
```

```
## x15  1.568766   0.151852  10.331  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7814 on 217 degrees of freedom
## Multiple R-squared:  0.9866, Adjusted R-squared:  0.9862
## F-statistic:  2286 on 7 and 217 DF,  p-value: < 2.2e-16
```

We performed step-wise regression. From the iterations above we see that AIC reduces after each iteration. This selection criteria produce a model with 6 variables (x2,x8,x9,x10,x13 and x15) all found to be statistically significant with an R-squared of 98.42%

2(c) Report the essential steps and/or key quantities involved in the variable selection procedure that you choose.
………………………………………………………………………………………………………

i)LASSO: The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so, the method apply a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero.
ii) ALASSO: Adaptive LASSO selection is a modification of LASSO selection. In adaptive LASSO selection, weights are applied to each of the parameters in forming the LASSO constraint.Adaptive LASSO enjoys the oracle properties; namely, it performs as well as if the true underlying model were given in advance.

iii)Step-wise: Step-wise regression is a combination of the forward and backward selection techniques. Step-wise regression is a modification of the forward selection so that after each step in which a variable was added, all candidate variables in the model are checked to see if their significance has been reduced below the specified tolerance level. If a non significant variable is found, it is removed from the model. Stepwise regression requires two significance levels: one for adding variables and one for removing variables. The cutoff probability for adding variables should be less than the cutoff probability for removing variables so that the procedure does not get into an infinite loop.

2(d) Output the necessary fitting results for each 'best' model, e.g., in particular, selected variables and their corresponding slope parameter estimates.
………………………………………………………………………………………………………..

```
#Outputting  the best fit for the LASSO selections method
fit1<- lm(y ~ x8+x10+x13+x14+x15 -1, data=D )
summary(fit1)

##
## Call:
## lm(formula = y ~ x8 + x10 + x13 + x14 + x15 - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -4.3472 -0.3721  1.0190  2.8498  5.0114
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## x8  0.027798   0.008375   3.319  0.00106 **
## x10 0.037271   0.006199   6.013 7.56e-09 ***
## x13 3.260204   0.402453   8.101 3.80e-14 ***
## x14 1.004756   1.099729   0.914  0.36191
## x15 3.539593   0.442470   8.000 7.19e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.4 on 219 degrees of freedom
## Multiple R-squared:  0.8727, Adjusted R-squared:  0.8698
## F-statistic: 300.3 on 5 and 219 DF,  p-value: < 2.2e-16
```

The OLS model of the best fits for LASSO variables is statistically significant given the F-Values and P-values from the above output with adjusted R-Square of 87.03%

```
# Outputting  the best fit for the ALASSO selections method.

fit1<- lm(y ~ x8+x13+x15 -1, data=D )
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ x8 + x13 + x15 - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4163 -0.3308  1.2814  3.2247  5.2396
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## x8  0.068743   0.005144  13.363  < 2e-16 ***
## x13 3.685390   0.381080   9.671  < 2e-16 ***
## x15 3.641531   0.474669   7.672 5.37e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 221 degrees of freedom
## Multiple R-squared:  0.8515, Adjusted R-squared:  0.8495
## F-statistic: 422.5 on 3 and 221 DF,  p-value: < 2.2e-16
```

The OLS model of the best fits for ALASSO variables is statistically significant given the F-Values and P-values from the above output with adjusted R-Square of 84.65%

The OLS model of best-fit for Step-wise variables is found to be statistically significant given the F-Values and P-values above with an adjusted R-Square of 98.42%

2(e) Apply your 'best' models to the test data D0 Output the sum of squared prediction error (SSPE). Let's consider the one yielding the minimum SSPE as the final model.

```
#LASSO fit with test data
fit1.D_prime<- lm(y ~ x7+x8++x10+x11+x13+x15 -1, data=D_prime )
summary(fit1)

##
## Call:
## lm(formula = y ~ x8 + x13 + x15 - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4163 -0.3308  1.2814  3.2247  5.2396
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## x8  0.068743   0.005144  13.363  < 2e-16 ***
## x13 3.685390   0.381080   9.671  < 2e-16 ***
## x15 3.641531   0.474669   7.672 5.37e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 221 degrees of freedom
## Multiple R-squared:  0.8515, Adjusted R-squared:  0.8495
## F-statistic: 422.5 on 3 and 221 DF,  p-value: < 2.2e-16

pred1.D_prime<-predict(fit1.D_prime,newdata = D_prime)

#Adaptive ALASSO with Test data
fit2.D_prime<- lm(y~x8+x13+x15, data=D_prime )
summary(fit2.D_prime)

##
## Call:
## lm(formula = y ~ x8 + x13 + x15, data = D_prime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.82444 -0.34500 -0.00774  0.40221  1.20095
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.997373   0.100423  49.763  < 2e-16 ***
## x8          0.014698   0.002145   6.853 4.50e-10 ***
## x13         1.665919   0.143673  11.595  < 2e-16 ***
## x15         1.423348   0.158840   8.961 9.97e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5876 on 109 degrees of freedom
```

```
## Multiple R-squared:  0.7796, Adjusted R-squared:  0.7735
## F-statistic: 128.5 on 3 and 109 DF,  p-value: < 2.2e-16
```

```
pred2.D_prime<-predict(fit2.D_prime, newdata = D_prime)
```

```
fit3.D_prime<-lm(formula = y ~ x2 + x8 + x9 + x10 + x13 + x15 - 1, data =
D_prime)
summary(fit3.D_prime)
```

```
##
## Call:
## lm(formula = y ~ x2 + x8 + x9 + x10 + x13 + x15 - 1, data = D_prime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9627 -0.3070  0.1358  0.5292  2.8090
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x2  16.586201   0.594753  27.888  < 2e-16 ***
## x8   0.008608   0.005768   1.492    0.139
## x9  -0.015030   0.005806  -2.589    0.011 *
## x10  0.006327   0.004241   1.492    0.139
## x13  1.609493   0.227973   7.060 1.74e-10 ***
## x15  1.538497   0.243156   6.327 5.94e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8978 on 107 degrees of freedom
## Multiple R-squared:  0.9826, Adjusted R-squared:  0.9816
## F-statistic:  1007 on 6 and 107 DF,  p-value: < 2.2e-16
```

```
pred3.D_prime<-predict(fit3.D_prime, newdata= D_prime)
```

Estimating the SSPE

```
sum((D_prime$y-pred1.D_prime)*2)
```

```
## [1] 227.0976
```

```
sum((D_prime$y-pred2.D_prime)*2)
```

```
## [1] 9.769963e-14
```

```
sum((D_prime$y-pred3.D_prime)*2)
```

```
## [1] 18.52055
```

From the above SSPE, ALASSO has the least number (8.437695e-13) followed by the stepwise (16.12466) then LASSO (222.3409). Since the ALASSO yields the minimum SSPE it is considered the best model from these three.

3). Refit your final model using the entire data, i.e., D ∪D0 . Call it fit.final. Provide the output from your final model with summary(fit.final). Interpret the results.

```
fit.final<- lm(formula0, data=dat)
summary(fit.final)

##
## Call:
## lm(formula = formula0, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1415 -0.3165  0.0940  0.4784  4.3657
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x1   2.6777258  3.2238395   0.831  0.40682
## x2  13.2225785  2.5900614   5.105 5.68e-07 ***
## x3   0.0024554  0.0067120   0.366  0.71474
## x4   0.0004297  0.0039250   0.109  0.91289
## x5  -0.0020711  0.0102425  -0.202  0.83988
## x6  -0.0315463  0.0257256  -1.226  0.22100
## x7  -0.0282370  0.0147752  -1.911  0.05688 .
## x8   0.0162860  0.0060105   2.710  0.00710 **
## x9  -0.0162213  0.0051447  -3.153  0.00177 **
## x10  0.0092750  0.0023184   4.001 7.85e-05 ***
## x11  0.0029501  0.0056059   0.526  0.59907
## x12 -0.0001796  0.0089033  -0.020  0.98392
## x13  1.8370748  0.1278943  14.364  < 2e-16 ***
## x14 -0.3844369  0.3272867  -1.175  0.24102
## x15  1.5704458  0.1394797  11.259  < 2e-16 ***
## x16 -0.4631937  0.5748378  -0.806  0.42097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8258 on 321 degrees of freedom
## Multiple R-squared:  0.9853, Adjusted R-squared:  0.9845
## F-statistic:  1342 on 16 and 321 DF,  p-value: < 2.2e-16
```

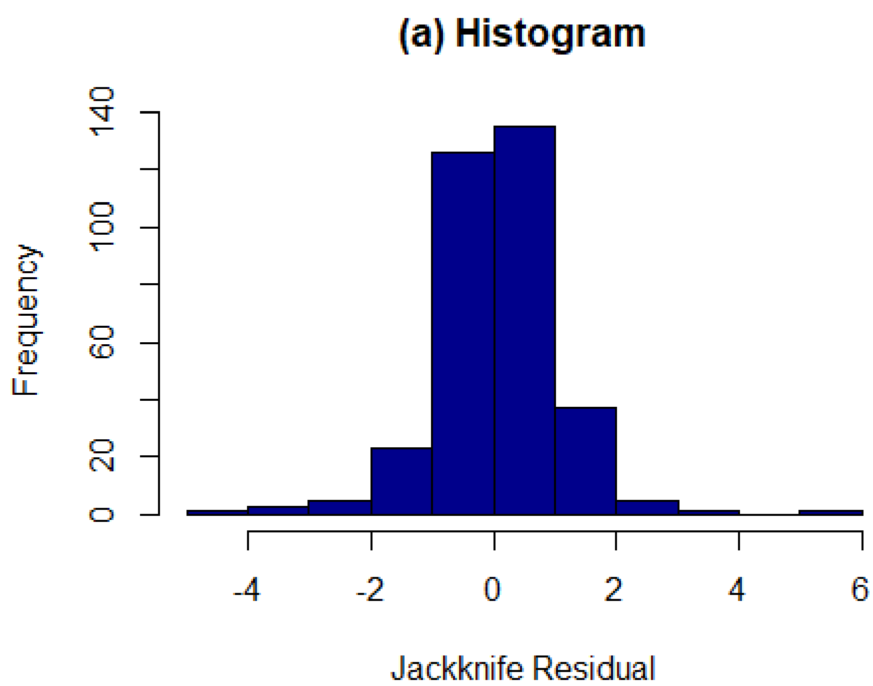
The full model (fit.final) is found to be statistically significant given its P-value from the above with an Adjusted R-Squared of 98.45%. Six(6) variables; x2,x8,x9,x10,x13,and x15 are statistically significant with p_values <0.05.

4) Check Model Assumption

a) Checking for Normality ………………………………………………………

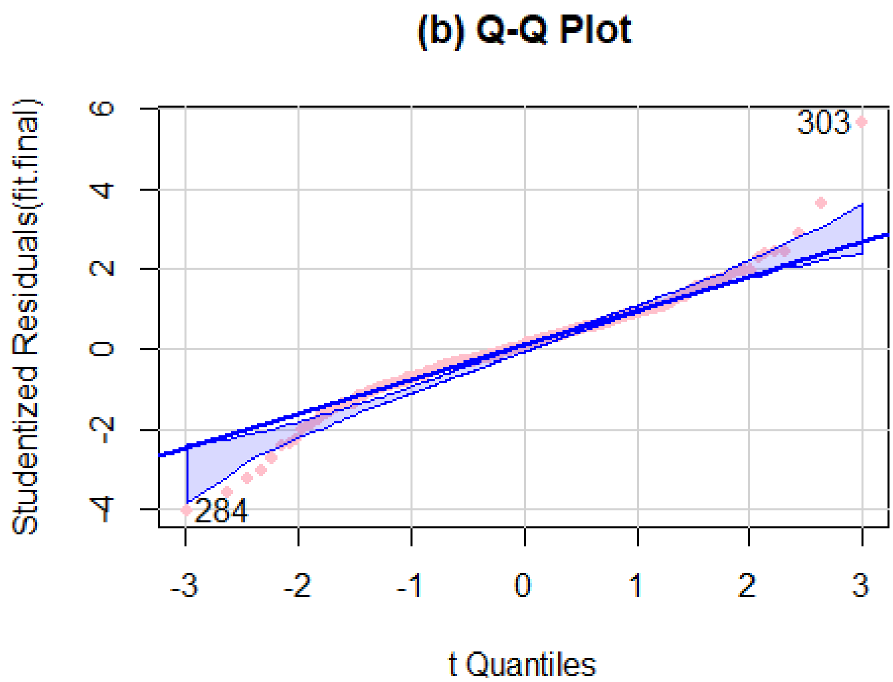
```
r.jack <- rstudent(fit.final)
# The plot of Histogram
```

```
hist(r.jack, xlab="Jackknife Residual", col="blue4",
    main="(a) Histogram")
```



We can observed from the histogram that the plot is slightly normal, which is an indication of normality.

```
set.seed(125)
library(car)

## Loading required package: carData

# A qq plot for studentized jackknife residuals
qqPlot(fit.final, pch=19, cex=.8, col="pink", main="(b) Q-Q Plot")

## Warning in rlm.default(x, y, weights, method = method, wt.method =
wt.method, :
## 'rlm' failed to converge in 20 steps
```




```
## [1] 284 303
```

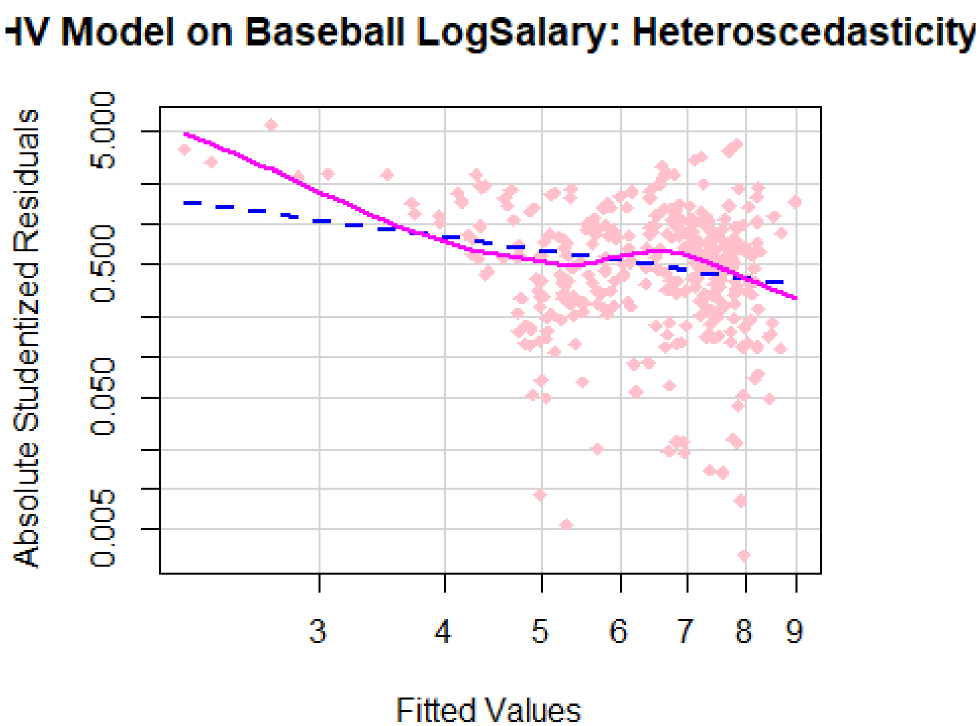
Fom the Q-Q plot above, there is a deviation of the plot from the line in the graph thus, normality assumption is violated

```
shapiro.test(r.jack)

##
##  Shapiro-Wilk normality test
##
## data:  r.jack
## W = 0.95121, p-value = 3.953e-09
```

From the Shapiro test, since the p-value is small, it is an indication of a violation of the normality assumption.

b) Check Homoscedasticity ………………………………………………………………………..

```
# Plot Absolute Jackknife Residuals vs. Fitted values
par(mfrow=c(1,1),mar=c(4, 4, 4, 4))
spreadLevelPlot(fit.final, pch=18, cex=0.5, col="pink",
    main="HV Model on Baseball LogSalary: Heteroscedasticity")
```




```
##
## Suggested power transformation:  2.022572

# IF THE LINES ARE FLAT, THEN EQUAL VARIANCE IS JUSTIFIED.
```

From the above plot since the lines are not flat, it's an indication of heteroscedasticity'

```
library(car)
# homoscedasticity
# The Breusch-Pagan Test for Non-Constant Error Variance
ncvTest(fit.final)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 65.99886, Df = 1, p = 4.5118e-16
```

From the p_value above, The Breusch-Pagan Test for Non-Constant Error Variance indicates unequal variances and this confirms the result from the graph of the unequal variance

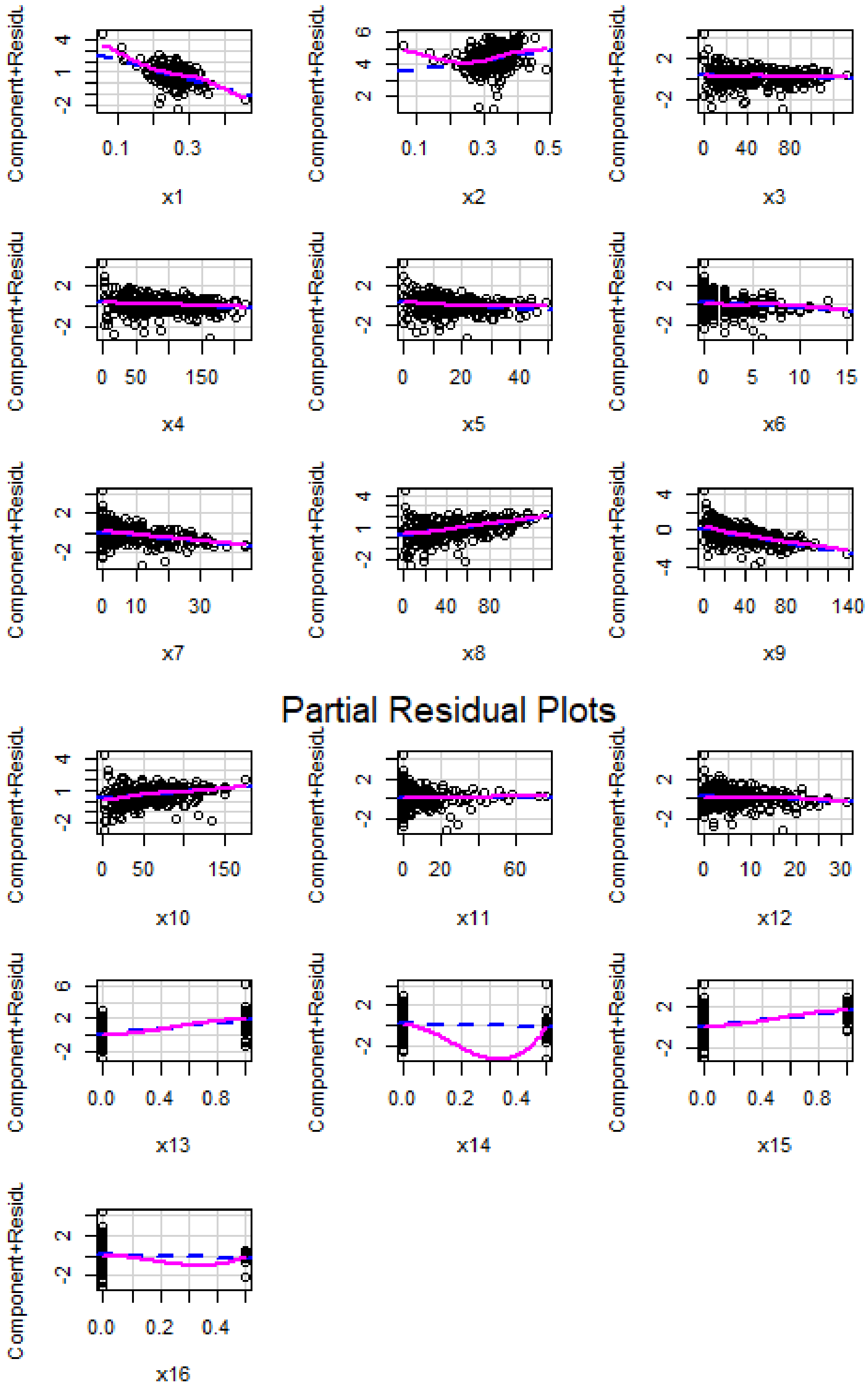
C) Checking for Independence ……………………………………………………………..

```
library(car)
# Test for Autocorrelated Errors
durbinWatsonTest(fit.final)

##  lag Autocorrelation D-W Statistic p-value
##    1       0.1064993      1.779008   0.034
##  Alternative hypothesis: rho != 0
```

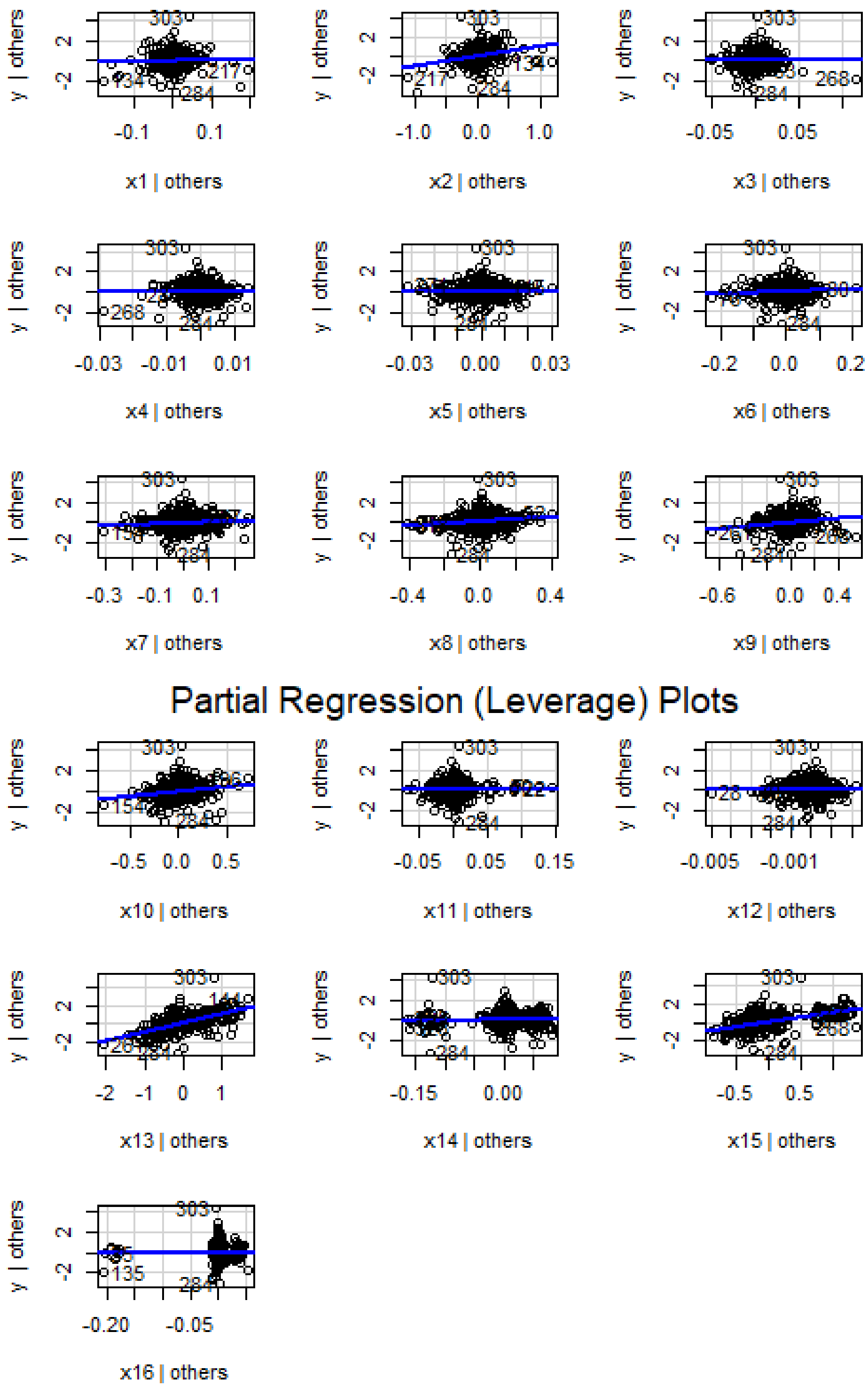
From the p_value above there is a violation of the independence assumption since the p_values < 0.05.

D) Checking for Linearity ………………………………………………………………………………..

```
library(car)
crPlots(fit.final, main="Partial Residual Plots")
```




```
library(car)
#leverage plots or partial regression plot
leveragePlots(fit.final, main="Partial Regression (Leverage) Plots")
```



From the plots above, there is no linearity in the model

d)Oulier Detection ………………………………………………………………………………………..

```r
infl <- influence.measures(fit.final);
infl.mat <- as.data.frame(infl$infmat)
n <- NROW(dat);
p <- length(coef(fit.final))-1 # NUMBER OF SLOPES
```

Cook's Distance

```r
# Cook's Distance
cook.d <- infl.mat$cook.d
infl <- summary(influence.measures(fit.final)); infl
```

```
## Potentially influential observations of
##   lm(formula = formula0, data = dat) :
##
##     dfb.x1 dfb.x2 dfb.x3  dfb.x4 dfb.x5 dfb.x6 dfb.x7 dfb.x8 dfb.x9
dfb.x10
## 21  -0.02   0.02   0.02    0.00   0.01   0.01   0.00   0.00  -0.04  -0.03
## 22   0.00   0.00   0.00    0.00   0.00   0.00   0.00   0.00   0.00   0.00
## 23  -0.07   0.03   0.16    0.01  -0.12  -0.15  -0.11   0.07  -0.14   0.13
## 28  -0.02   0.02  -0.02    0.08  -0.03  -0.02  -0.03  -0.03  -0.01   0.04
## 35   0.00   0.00  -0.02    0.00   0.00   0.01   0.01   0.00   0.01   0.00
## 52   0.03  -0.03  -0.07   -0.06   0.01   0.03  -0.01   0.11   0.11  -0.09
## 54   0.00   0.00  -0.02    0.04  -0.10   0.05  -0.06   0.06   0.00   0.02
## 57  -0.01   0.01  -0.03    0.03  -0.01  -0.05  -0.02   0.03   0.00  -0.01
## 76   0.00   0.00   0.01    0.04   0.00  -0.11   0.03  -0.05   0.01  -0.03
## 80   0.00   0.00   0.01    0.00  -0.02  -0.05  -0.01   0.00  -0.01   0.00
## 98   0.00   0.00   0.00    0.00  -0.01  -0.01   0.00   0.00   0.00   0.00
## 115  0.04  -0.04   0.04    0.05  -0.09  -0.04  -0.02  -0.05   0.05   0.02
## 126 -0.01   0.01  -0.01    0.03  -0.06  -0.03  -0.04   0.05  -0.05   0.06
## 134  0.62  -0.67   0.05   -0.24   0.04   0.00  -0.10   0.00   0.42   0.20
## 135 -0.01  -0.04   0.07    0.10   0.03  -0.09  -0.05  -0.04  -0.03   0.08
## 148 -0.22   0.23   0.09    0.00  -0.01  -0.07  -0.05   0.04  -0.21   0.04
## 151 -0.03   0.05  -0.02   -0.02   0.01   0.02   0.02   0.02  -0.02  -0.04
## 152 -0.23   0.25  -0.02    0.09  -0.02  -0.01   0.02   0.00  -0.16  -0.03
## 169  0.00   0.00   0.00    0.00   0.00   0.00   0.00   0.00   0.00   0.00
## 173  0.00   0.00   0.01    0.00   0.02  -0.02  -0.01  -0.01   0.00   0.00
## 179 -0.96   0.86   0.03    0.51  -0.06  -0.29  -0.13   0.01  -0.61   0.17
## 183 -0.01   0.03  -0.08    0.09   0.07  -0.04   0.06  -0.12   0.00   0.02
## 196  0.04  -0.05   0.05   -0.12   0.02   0.02  -0.04   0.00   0.06   0.16
## 205  0.15  -0.14   0.24    0.06  -0.03  -0.19   0.16  -0.07  -0.11  -0.31
## 206  0.00   0.00   0.02    0.00  -0.01   0.02   0.01  -0.01   0.00  -0.01
## 233  0.00   0.00  -0.04    0.00   0.04   0.04   0.02  -0.01   0.02  -0.01
## 234  0.00   0.01  -0.02    0.01  -0.04  -0.02  -0.01   0.05   0.02  -0.04
## 246  0.00   0.00   0.01   -0.01   0.00   0.00   0.00   0.00   0.00   0.00
## 261 -0.04   0.04   0.04   -0.02   0.01  -0.01  -0.04   0.03  -0.10   0.03
## 264 -0.07   0.09   0.12   -0.03  -0.01  -0.01  -0.03  -0.03  -0.11   0.07
## 268  0.22  -0.26  -1.19_*  0.98  -0.39   0.14   0.48  -0.23   0.66  -0.10
## 284 -0.04   0.09   0.11   -0.26   0.14  -0.06  -0.11   0.30   0.02  -0.49
## 297 -0.01   0.00   0.03   -0.03   0.02  -0.01  -0.02   0.04  -0.03  -0.01
## 298  0.01   0.00   0.02   -0.01  -0.01  -0.05   0.02  -0.01   0.02  -0.03
```

```
## 303  0.37  -0.37   0.05   -0.28   0.08   0.03   0.02  -0.03   0.18   0.06
## 318  0.00   0.00   0.02    0.01   0.01  -0.02   0.02  -0.05   0.01  -0.01
## 322  0.04  -0.03   0.09   -0.22   0.16   0.14  -0.10   0.18   0.09  -0.42
## 324  0.00   0.00   0.04    0.01   0.07  -0.01   0.08  -0.10   0.00  -0.05
## 333  0.02   0.00   0.07   -0.05  -0.04  -0.03  -0.02   0.01  -0.04   0.02
##     dfb.x11 dfb.x12 dfb.x13 dfb.x14 dfb.x15 dfb.x16 dffit   cov.r   cook.d
## 21  -0.05   -0.04    0.02    0.00    0.02   -0.01   -0.10    1.23_*  0.00
## 22   0.01    0.00    0.00    0.00    0.00    0.00    0.01    1.25_*  0.00
## 23  -0.01   -0.03    0.10    0.01    0.12   -0.02   -0.40    0.82_*  0.01
## 28  -0.05   -0.12   -0.03    0.02    0.02    0.00   -0.17    1.24_*  0.00
## 35   0.00    0.01    0.00    0.00    0.00    0.06    0.06    1.20_*  0.00
## 52   0.14   -0.02   -0.03    0.02    0.05   -0.02    0.25    1.23_*  0.00
## 54  -0.03   -0.02    0.01   -0.02    0.01    0.28    0.34    1.15_*  0.01
## 57   0.05   -0.07    0.00    0.00    0.01   -0.15   -0.20    1.21_*  0.00
## 76  -0.04    0.01    0.02    0.00    0.03    0.00   -0.17    1.18_*  0.00
## 80   0.06   -0.02    0.00    0.03    0.00    0.00    0.09    1.30_*  0.00
## 98   0.01    0.00    0.00    0.00    0.00    0.04    0.05    1.19_*  0.00
## 115 -0.01   -0.07    0.01   -0.01    0.01    0.00    0.19    1.21_*  0.00
## 126  0.01    0.00    0.00   -0.01    0.00    0.17    0.22    1.21_*  0.00
## 134 -0.03    0.07    0.16   -0.01    0.10    0.00   -0.76_*  0.91    0.04
## 135 -0.05    0.07   -0.13    0.03   -0.12   -0.79   -0.94_*  0.91    0.05
## 148 -0.01    0.01   -0.06    0.00   -0.04   -0.01    0.29    0.80_*  0.01
## 151  0.02    0.06   -0.02   -0.02   -0.04    0.01    0.18    0.70_*  0.00
## 152  0.01   -0.02   -0.06    0.00   -0.04    0.00    0.28    0.83_*  0.00
## 169  0.00    0.00    0.00    0.00    0.00    0.00    0.00    1.18_*  0.00
## 173 -0.01    0.00    0.00    0.00    0.00    0.06    0.08    1.22_*  0.00
## 179 -0.01   -0.05   -0.01    0.06    0.10   -0.07   -1.23_*  0.63_*  0.09
## 183 -0.01   -0.02   -0.07   -0.01   -0.07    0.00    0.26    0.80_*  0.00
## 196 -0.06    0.02    0.07   -0.03    0.04    0.01    0.25    1.16_*  0.00
## 205 -0.31   -0.03   -0.35    0.22   -0.08   -0.06   -0.85_*  0.67_*  0.04
## 206 -0.02    0.00    0.00    0.00   -0.01    0.00    0.05    1.20_*  0.00
## 233  0.05    0.01    0.00    0.00    0.02   -0.01    0.09    1.18_*  0.00
## 234  0.01    0.03   -0.02    0.01    0.00    0.01   -0.10    1.17_*  0.00
## 246 -0.02    0.01    0.00    0.00    0.00   -0.05   -0.06    1.23_*  0.00
## 261  0.02    0.03    0.07   -0.02    0.03    0.00   -0.15    1.21_*  0.00
## 264 -0.10   -0.07   -0.06    0.01   -0.03   -0.01    0.24    0.81_*  0.00
## 268  0.20    0.06   -0.18   -0.05   -0.46    0.25   -1.56_*  0.98    0.15
## 284 -0.08   -0.11    0.05   -0.53    0.07    0.01   -1.05_*  0.51_*  0.07
## 297  0.00   -0.03   -0.02    0.01   -0.01    0.14    0.18    1.19_*  0.00
## 298  0.08    0.01    0.01   -0.01    0.00    0.00    0.13    1.18_*  0.00
## 303  0.08   -0.05    0.39    0.72    0.31    0.04    1.12_*  0.24_*  0.07
## 318 -0.03    0.01   -0.03    0.01   -0.02   -0.01    0.08    1.16_*  0.00
## 322  0.06    0.18   -0.23    0.15   -0.09   -0.02   -0.67_*  0.70_*  0.03
## 324 -0.05    0.00   -0.04    0.03   -0.02    0.18    0.27    1.23_*  0.00
## 333 -0.03   -0.04   -0.03   -0.01   -0.02    0.00    0.21    0.55_*  0.00
##     hat
## 21   0.15_*
## 22   0.16_*
## 23   0.03
## 28   0.16_*
```

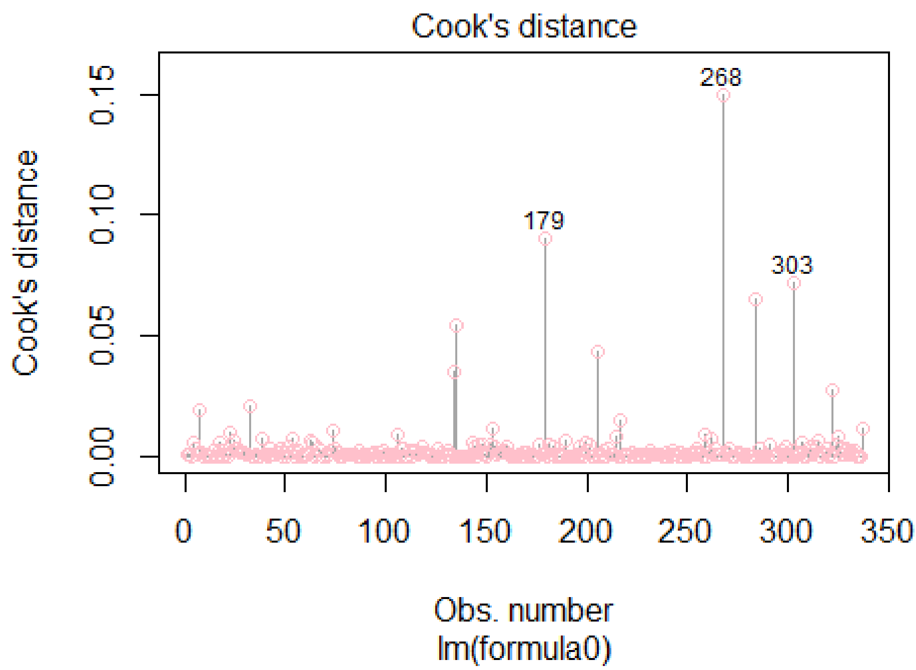
```
## 35   0.12
## 52   0.16_*
## 54   0.13
## 57   0.14_*
## 76   0.12
## 80   0.19_*
## 98   0.12
## 115  0.14
## 126  0.14_*
## 134  0.10
## 135  0.13
## 148  0.01
## 151  0.00
## 152  0.01
## 169  0.11
## 173  0.14
## 179  0.11
## 183  0.01
## 196  0.12
## 205  0.06
## 206  0.12
## 233  0.11
## 234  0.10
## 246  0.14_*
## 261  0.14
## 264  0.01
## 268  0.25_*
## 284  0.06
## 297  0.13
## 298  0.11
## 303  0.04
## 318  0.10
## 322  0.05
## 324  0.16_*
## 333  0.00

##            dfb.x1        dfb.x2        dfb.x3        dfb.x4       dfb.x5
## 21  -0.0161761215  0.0188447398  2.213061e-02  0.0036127279  0.008603872
## 22  -0.0002659282  0.0002740378 -3.621993e-03  0.0021084240  0.001029766
## 23  -0.0709192492  0.0289277910  1.594976e-01  0.0140484691 -0.121077098
## 28  -0.0186971746  0.0197017486 -1.619899e-02  0.0815758475 -0.033698196
## 35  -0.0001752466  0.0006138459 -1.853166e-02  0.0036672912  0.002468403
## 52   0.0340502177 -0.0315632192 -6.992695e-02 -0.0637168420  0.006124730
## 54   0.0029663822 -0.0042731599 -2.459992e-02  0.0370418296 -0.099111731
## 57  -0.0072711128  0.0104710518 -3.410832e-02  0.0278503379 -0.012222293
## 76  -0.0038329164  0.0049827002  7.565819e-03  0.0391701220 -0.004978486
## 80   0.0028411289 -0.0021824595  1.254355e-02  0.0026921314 -0.016699235
## 98   0.0024747957 -0.0012392197 -2.030974e-03 -0.0013225775 -0.006512607
## 115  0.0353064415 -0.0389513717  3.531794e-02  0.0498705440 -0.090015819
## 126 -0.0116162629  0.0079399072 -1.177554e-02  0.0283135306 -0.064896643
```

```
## 134  0.6171104687 -0.6703706004  5.030970e-02 -0.2359661814  0.042815439
## 135 -0.0072862570 -0.0382045293  6.811633e-02  0.0988148171  0.031655871
## 148 -0.2152488839  0.2288562617  8.819029e-02  0.0026956864 -0.006775489
## 151 -0.0271502826  0.0471508235 -2.460006e-02 -0.0246315488  0.014250662
## 152 -0.2274445136  0.2453474148 -2.358406e-02  0.0887530842 -0.024114458
## 169 -0.0004981907  0.0005279753  1.745592e-05 -0.0002118437 -0.001255445
## 173 -0.0023716903  0.0007628150  1.028331e-02 -0.0039762531  0.024649437
## 179 -0.9615650219  0.8551859434  3.474152e-02  0.5115878395 -0.060935541
## 183 -0.0145010476  0.0328596075 -8.116558e-02  0.0922560836  0.070228959
## 196  0.0431452682 -0.0485439121  4.803077e-02 -0.1154719974  0.017167531
## 205  0.1465378942 -0.1415970687  2.385536e-01  0.0554566408 -0.034755224
## 206  0.0005195280 -0.0007532497  1.535713e-02  0.0043446987 -0.008163570
## 233 -0.0014680007 -0.0003900825 -4.443791e-02  0.0028280714  0.042582432
## 234 -0.0039288460  0.0068688908 -2.099598e-02  0.0062023388 -0.040195090
## 246 -0.0005440429  0.0011701659  5.294068e-03 -0.0074196713  0.002213893
## 261 -0.0377219743  0.0370894855  3.672259e-02 -0.0199476830  0.010792088
## 264 -0.0672342024  0.0850162567  1.176859e-01 -0.0277705502 -0.005201543
## 268  0.2248920459 -0.2603884090 -1.194100e+00  0.9781847542 -0.391620364
## 284 -0.0373415753  0.0866979851  1.110161e-01 -0.2646129380  0.140952750
## 297 -0.0053419106  0.0037228871  2.940333e-02 -0.0256379857  0.015891045
## 298  0.0058599138 -0.0047515268  1.602217e-02 -0.0149712329 -0.013669441
## 303  0.3656426125 -0.3654920143  4.891033e-02 -0.2832642936  0.081520275
## 318  0.0017950247 -0.0024139939  2.060877e-02  0.0121899060  0.014991161
## 322  0.0355751112 -0.0261436720  9.387116e-02 -0.2183254057  0.162768264
## 324  0.0001601976 -0.0010107362  3.530915e-02  0.0104829948  0.073521349
## 333  0.0199964924  0.0044772422  6.725948e-02 -0.0469148850 -0.043491361
##            dfb.x6        dfb.x7       dfb.x8        dfb.x9       dfb.x10
## 21   9.962379e-03 -0.0002291384  0.003520214 -3.753561e-02 -2.717227e-02
## 22   9.317473e-04  0.0022673308 -0.001330786  2.419842e-04 -1.625286e-05
## 23  -1.483186e-01 -0.1128797434  0.068464528 -1.389500e-01  1.333697e-01
## 28  -2.357580e-02 -0.0252399870 -0.032432519 -1.051895e-02  3.747549e-02
## 35   7.609904e-03  0.0074794952 -0.004518274  1.464579e-02 -3.046269e-03
## 52   3.389162e-02 -0.0112746075  0.113904402  1.084716e-01 -9.029850e-02
## 54   4.604982e-02 -0.0578802257  0.063308104  2.313174e-03  1.798603e-02
## 57  -5.309983e-02 -0.0226581683  0.033938763  4.846884e-06 -6.698138e-03
## 76  -1.067414e-01  0.0288558818 -0.050317092  5.929491e-03 -2.951411e-02
## 80  -4.501741e-02 -0.0070310556  0.004496219 -8.974223e-03 -2.533458e-03
## 98  -6.125902e-03  0.0037302819  0.001812188  7.648659e-04 -7.821521e-04
## 115 -4.146389e-02 -0.0191412541 -0.047007220  5.217168e-02  2.399296e-02
## 126 -2.925374e-02 -0.0351755836  0.054056285 -4.576545e-02  5.876490e-02
## 134  1.142623e-04 -0.0966776598  0.001205448  4.243778e-01  2.036264e-01
## 135 -8.842979e-02 -0.0491521885 -0.038903773 -3.115022e-02  8.245413e-02
## 148 -6.805519e-02 -0.0480891296  0.036568634 -2.050642e-01  3.939422e-02
## 151  1.539133e-02  0.0169502312  0.015504595 -2.216391e-02 -3.896402e-02
## 152 -9.153567e-03  0.0167393796  0.004321231 -1.583861e-01 -2.819745e-02
## 169  8.768555e-05 -0.0005113719  0.001395646 -1.162560e-03  9.010880e-04
## 173 -1.699208e-02 -0.0137967199 -0.005641016  2.810905e-03 -5.819427e-04
## 179 -2.900427e-01 -0.1268047062  0.013715215 -6.077554e-01  1.660816e-01
## 183 -4.316785e-02  0.0577401895 -0.116081020 -1.356095e-03  1.721776e-02
## 196  2.351824e-02 -0.0394561172  0.003388792  5.973221e-02  1.582783e-01
```

```
## 205 -1.933764e-01  0.1604594766 -0.067956363 -1.111634e-01 -3.084556e-01
## 206  2.380209e-02  0.0104206608 -0.014671984 -1.345775e-03 -1.409265e-02
## 233  3.769010e-02  0.0177592454 -0.007233761  1.967127e-02 -1.412818e-02
## 234 -2.426588e-02 -0.0127218137  0.046109291  1.523044e-02 -4.178121e-02
## 246  6.935054e-04  0.0033060664  0.002059481 -2.022603e-03 -4.222160e-03
## 261 -1.437505e-02 -0.0407672835  0.028257210 -1.024385e-01  3.008305e-02
## 264 -1.232521e-02 -0.0331708713 -0.029402631 -1.135251e-01  6.805981e-02
## 268  1.403839e-01  0.4762090058 -0.234682826  6.555148e-01 -9.562078e-02
## 284 -6.265219e-02 -0.1108648299  0.302664840  2.046093e-02 -4.921638e-01
## 297 -1.217442e-02 -0.0239581973  0.043917551 -2.767473e-02 -9.661444e-03
## 298 -4.600949e-02  0.0173528011 -0.007235870  1.955489e-02 -2.695689e-02
## 303  2.523734e-02  0.0179265276 -0.028764885  1.848772e-01  6.361097e-02
## 318 -1.648591e-02  0.0177508380 -0.045366578  1.276269e-02 -7.150764e-03
## 322  1.414688e-01 -0.0952070561  0.175917438  9.457514e-02 -4.165563e-01
## 324 -1.478046e-02  0.0806546267 -0.098419729  4.976702e-03 -5.460796e-02
## 333 -2.778567e-02 -0.0209512241  0.012101725 -4.159693e-02  1.627638e-02
##           dfb.x11       dfb.x12       dfb.x13       dfb.x14       dfb.x15
## 21  -0.0545417492 -0.0430937105  0.0157628538 -6.273093e-04  0.0166800367
## 22   0.0072551340 -0.0007778508 -0.0015037989  6.997621e-04 -0.0016562891
## 23  -0.0101288445 -0.0329095182  0.1030228944  1.081739e-02  0.1159288072
## 28  -0.0501898182 -0.1230620921 -0.0328902325  2.322426e-02  0.0175172143
## 35   0.0032596092  0.0125077370  0.0040842132 -2.656313e-03  0.0025132382
## 52   0.1399066813 -0.0177490710 -0.0339291327  2.143606e-02  0.0548760226
## 54  -0.0319800769 -0.0218110714  0.0129988160 -2.168835e-02  0.0053869408
## 57   0.0467143682 -0.0714763013 -0.0036989935  4.205415e-03  0.0092861502
## 76  -0.0376037093  0.0087707691  0.0211639092 -1.274127e-03  0.0329913968
## 80   0.0622251691 -0.0166988466 -0.0031974989  3.472221e-02  0.0004844998
## 98   0.0085957254 -0.0017991598  0.0028414964  3.279399e-04  0.0040271892
## 115 -0.0124887909 -0.0665237907  0.0064254197 -1.386486e-02  0.0055167736
## 126  0.0082170171 -0.0012708303  0.0038035961 -1.175812e-02 -0.0012955790
## 134 -0.0264135753  0.0735631854  0.1620080294 -1.326565e-02  0.1026740147
## 135 -0.0537893766  0.0676372698 -0.1309670292  2.546727e-02 -0.1190682436
## 148 -0.0098329358  0.0062239877 -0.0585466113  1.518064e-03 -0.0445726179
## 151  0.0216885052  0.0634444404 -0.0211336620 -1.705861e-02 -0.0396456236
## 152  0.0059634797 -0.0206892436 -0.0582319678 -7.059405e-04 -0.0385480419
## 169  0.0008356741 -0.0003222250 -0.0002613801  3.603944e-04  0.0002674750
## 173 -0.0124097358  0.0015773595 -0.0027232560  6.257773e-05 -0.0034506469
## 179 -0.0100647943 -0.0541339865 -0.0061862271  5.872105e-02  0.1015960880
## 183 -0.0068258723 -0.0165214539 -0.0739104670 -6.171600e-03 -0.0730837527
## 196 -0.0555535474  0.0163802402  0.0671450821 -3.345176e-02  0.0351478973
## 205 -0.3065199137 -0.0304934712 -0.3494434475  2.188365e-01 -0.0815275148
## 206 -0.0206995107 -0.0035627975  0.0014999519 -4.082570e-03 -0.0050323539
## 233  0.0549702090  0.0112012810 -0.0035388111  3.149292e-03  0.0162153404
## 234  0.0060469036  0.0321282516 -0.0180593742  6.683839e-03  0.0023899240
## 246 -0.0198058737  0.0141474471  0.0020551624 -2.305737e-03  0.0009343482
## 261  0.0203341689  0.0294165772  0.0698932750 -2.302404e-02  0.0275336318
## 264 -0.0978785314 -0.0745341529 -0.0561649559  6.455410e-03 -0.0347320935
## 268  0.1973768744  0.0611157492 -0.1799688443 -5.415700e-02 -0.4584408735
## 284 -0.0817387344 -0.1093552642  0.0492605635 -5.347622e-01  0.0661756264
## 297  0.0036104695 -0.0280047228 -0.0185351903  9.586735e-03 -0.0127319024
```

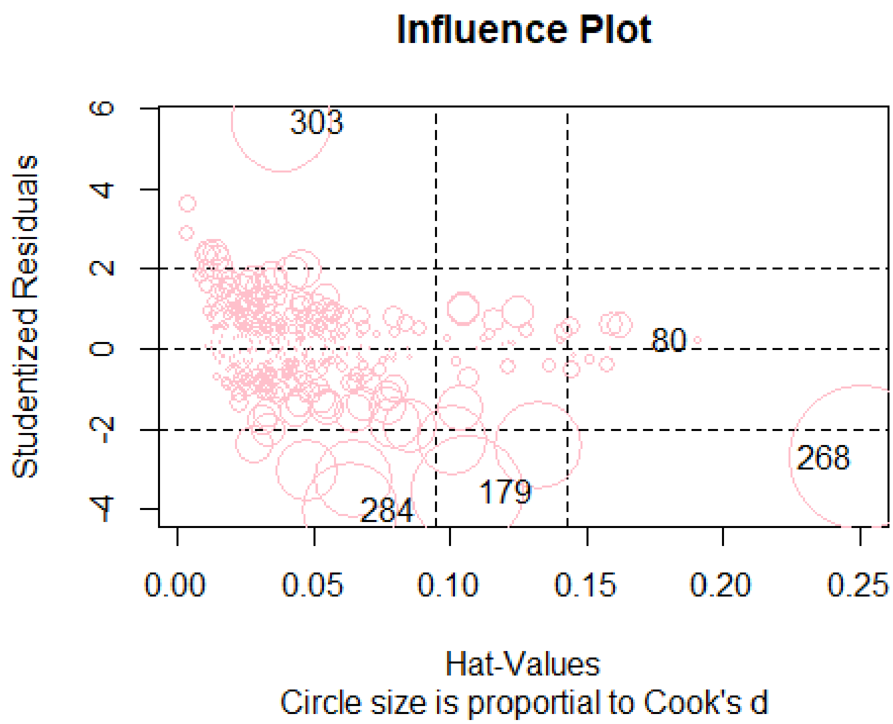
```
## 298  0.0828765023  0.0117029521  0.0143470777 -5.739579e-03 -0.0003906546
## 303  0.0824195362 -0.0522495013  0.3854859627  7.179846e-01  0.3069597508
## 318 -0.0348228040  0.0088828561 -0.0304419677  7.587702e-03 -0.0193158201
## 322  0.0649827552  0.1813358600 -0.2294741932  1.530121e-01 -0.0907178191
## 324 -0.0495809235  0.0010958620 -0.0394618632  2.550775e-02 -0.0196077652
## 333 -0.0307785769 -0.0422702945 -0.0318311018 -5.474564e-03 -0.0248900875
##          dfb.x16        dffit     cov.r       cook.d         hat
## 21  -0.0050736815 -0.103530654 1.2340117 6.718789e-04 0.150721328
## 22   0.0002700130  0.008764271 1.2497883 4.815774e-06 0.158923350
## 23  -0.0205431196 -0.403729087 0.8165334 1.004128e-02 0.027949857
## 28  -0.0035693506 -0.165758199 1.2387460 1.721813e-03 0.157615015
## 35   0.0561128829  0.064617633 1.1960572 2.617559e-04 0.122435430
## 52  -0.0158154773  0.253008721 1.2293902 4.009107e-03 0.159268355
## 54   0.2802434101  0.340538633 1.1542495 7.252222e-03 0.125353023
## 57  -0.1484149574 -0.202814420 1.2134391 2.576924e-03 0.144215264
## 76   0.0001701898 -0.166830315 1.1838351 1.743856e-03 0.120978096
## 80   0.0012760833  0.094425360 1.2967830 5.589347e-04 0.190916950
## 98   0.0402157681  0.045958147 1.1936963 1.324156e-04 0.120064387
## 115 -0.0030642752  0.188111966 1.2106238 2.217050e-03 0.141002164
## 126  0.1714810522  0.224726306 1.2105625 3.163278e-03 0.144528319
## 134 -0.0019019590 -0.757174028 0.9051987 3.537529e-02 0.100267359
## 135 -0.7944441453 -0.942242987 0.9088871 5.467022e-02 0.132618797
## 148 -0.0085318622  0.288806221 0.8007470 5.136564e-03 0.014223525
## 151  0.0103567651  0.178538880 0.6985863 1.947628e-03 0.003800374
## 152  0.0018336956  0.278116998 0.8291840 4.773546e-03 0.014978810
## 169  0.0001492099 -0.002895945 1.1813826 5.257940e-07 0.110207333
## 173  0.0599503217  0.083386052 1.2202430 4.358772e-04 0.140373117
## 179 -0.0729798289 -1.225029422 0.6338842 9.052031e-02 0.106370410
## 183 -0.0001839140  0.261213489 0.7979589 4.201723e-03 0.011630825
## 196  0.0075782969  0.248879435 1.1620382 3.877711e-03 0.116396545
## 205 -0.0589396118 -0.847263502 0.6733670 4.358853e-02 0.064524797
## 206 -0.0034140566  0.047400015 1.1965830 1.408543e-04 0.122216075
## 233 -0.0120577146  0.090485100 1.1802890 5.132174e-04 0.112258157
## 234  0.0066935417 -0.104251986 1.1653698 6.811994e-04 0.102269960
## 246 -0.0453311435 -0.059740032 1.2277095 2.237368e-04 0.144682938
## 261 -0.0026504195 -0.150242767 1.2084443 1.414582e-03 0.136326550
## 264 -0.0113735718  0.238853761 0.8082265 3.516296e-03 0.010248689
## 268  0.2477286415 -1.560844351 0.9789811 1.493435e-01 0.250766142
## 284  0.0111957134 -1.047543434 0.5097581 6.548676e-02 0.063502976
## 297  0.1381536184  0.180180727 1.1927102 2.034007e-03 0.128316970
## 298  0.0044796666  0.133313243 1.1794869 1.113771e-03 0.114855372
## 303  0.0424329759  1.121131163 0.2395324 7.167245e-02 0.037976960
## 318 -0.0067742937  0.079682275 1.1637932 3.979976e-04 0.099351532
## 322 -0.0158181798 -0.671191706 0.7021742 2.745824e-02 0.046881239
## 324  0.1790953289  0.270019558 1.2313426 4.565778e-03 0.162219605
## 333  0.0046123613  0.206540095 0.5523308 2.568554e-03 0.003221287

write.csv(infl, file="Infleunce-Mat.csv", row.names=TRUE)
# Plot of the Cook's Distance
cutoff <- 4/(n-p-2)
```

```r
plot(fit.final, which=4, cook.levels=cutoff, col="gray65", lwd=1.5)
points(1:n, cook.d, pch=1, cex=1, col="pink")
```



With a threshold of (0.0125), It is observed that there is one outlier from the plot above which is observation 268

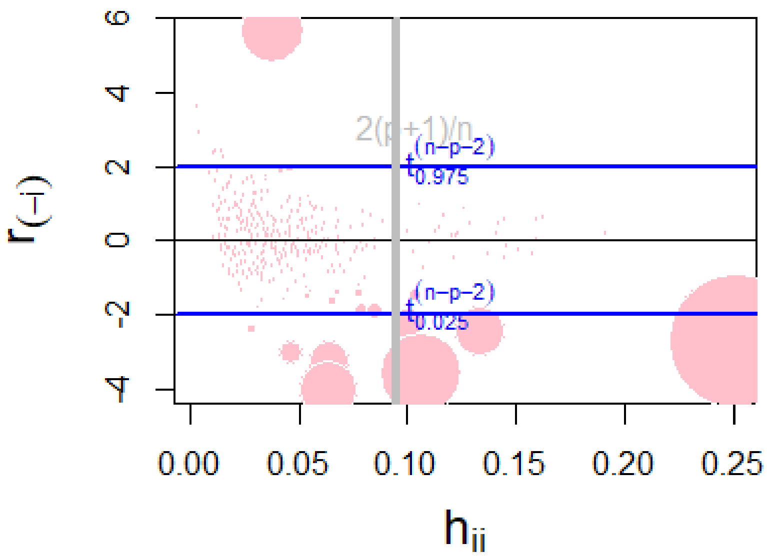
```r
library(car)
# EXTRACT INFLUETIAL POINTS
dat[cook.d > 0.05, ]

##             y    x1    x2 x3  x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16
## 135 4.828314 0.269 0.345  4   7  2  1  0  5  3   4   0   0   0 0.0   1 0.5
## 179 4.976734 0.457 0.486  6  16  4  2  0  7  2   2   0   2   0 0.0   0 0.0
## 268 4.691348 0.225 0.333 71   8 16  4  0  3 14  12  25   0   0 0.0   1 0.0
## 284 4.691348 0.271 0.328 74 161 22  6 12 58 49 133  23  17   1 0.5   0 0.0
## 303 7.047517 0.063 0.063  0   1  0  0  0  1  0   2   0   0   1 0.5   0 0.0

# Interactive Plot for Identifying Influential Points
influencePlot(fit.final,
    col="pink",
    main="Influence Plot",
    sub="Circle size is proportial to Cook's d")
```




```
##         StudRes        Hat        CookD
## 80   0.1943851 0.19091695 0.0005589347
## 179 -3.5507073 0.10637041 0.0905203077
## 268 -2.6979497 0.25076614 0.1493434548
## 284 -4.0227978 0.06350298 0.0654867603
## 303  5.6427273 0.03797696 0.0716724464
```

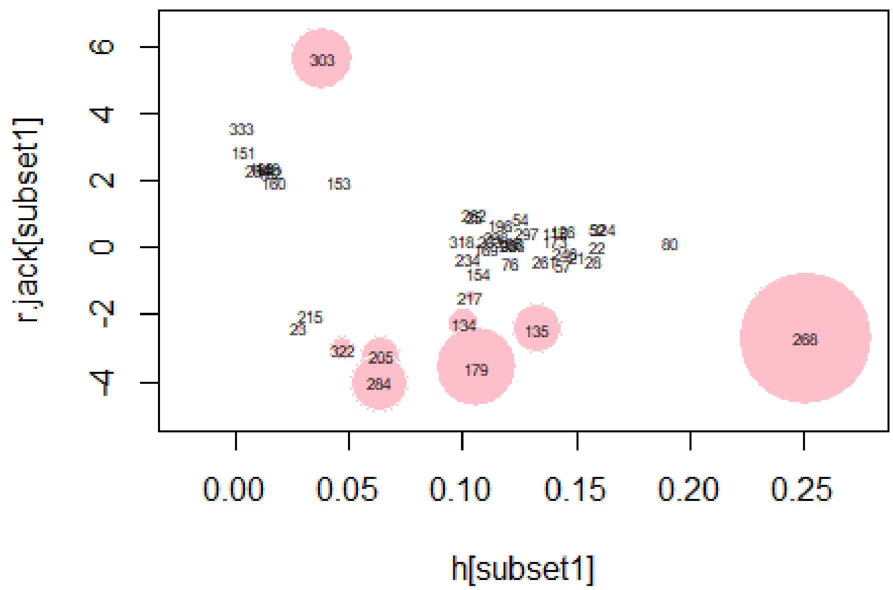
From the Influence Plot above, there are 5 outliers.

```
 #A BuBBLE PLOT OF THREE DIAGNOSTIC MEASURES
h <- infl.mat$hat
cook.d <- infl.mat$cook.d
par(bg="white", mar=c(5, 5, 5, 5), mfrow=c(1, 1), xaxt="s")
plot(x=c(min(h), max(h)), y=c(min(r.jack), max(r.jack)),
xlab=expression(h[ii]),
    ylab=expression(r[(-i)]), cex.lab=1.5, type="n")
symbols(h, r.jack, circles=cook.d, inches=0.35, fg="white", bg="pink", add=T)
abline(h=0, col="black", lwd=1)
abline(h=qt(.975, n-p-2), col="blue", lwd=2)
abline(h=qt(.025, n-p-2), col="blue", lwd=2)
text(x=.12, y=qt(.975, n-p-2)+.3, labels=expression(t[.975]^(n-p-2)),
col="blue")
text(x=.12, y=qt(.025, n-p-2)+.3, labels=expression(t[.025]^(n-p-2)),
col="blue")
abline(v=2*(p+1)/n, lwd=4, col="grey")
text(2*(p+1)/n+.008, 3.0, labels="2(p+1)/n", col="grey")
```



The bubble plot also confirms there are some form of outliers.

```
# IDENTIFYING OUTLIERS
t0 <- qt(.975, n-1-(p+1)); t0

## [1] 1.967405

# subset1 <- (cook.d >=.0325)|(h>0.06)|(abs(r.jack) > 3)
subset1 <- (cook.d >=0.065)|(h> 2*(p+1)/n)|(abs(r.jack) > t0)
symbols(h[subset1], r.jack[subset1], circles=cook.d[subset1], inches=0.35,
    fg="white", bg="pink",)
text(h[subset1], r.jack[subset1], (1:n)[subset1], cex=0.5)
```



```
cbind(id=(1:n)[subset1],  "r.jack"= abs(r.jack[subset1]) > t0, "h"=
h[subset1]> 2*(p+1)/n,
    "cook.d" = cook.d[subset1] >= 0.065)
```

```
##      id r.jack h cook.d
## 21   21      0 1      0
## 22   22      0 1      0
## 23   23      1 0      0
## 25   25      0 1      0
## 28   28      0 1      0
## 35   35      0 1      0
## 52   52      0 1      0
## 54   54      0 1      0
## 57   57      0 1      0
## 76   76      0 1      0
## 80   80      0 1      0
## 98   98      0 1      0
## 115 115      0 1      0
## 126 126      0 1      0
## 134 134      1 1      0
## 135 135      1 1      0
## 148 148      1 0      0
## 151 151      1 0      0
## 152 152      1 0      0
## 153 153      1 0      0
## 154 154      0 1      0
## 160 160      1 0      0
```

```
## 169 169      0 1      0
## 173 173      0 1      0
## 179 179      1 1      1
## 183 183      1 0      0
## 196 196      0 1      0
## 205 205      1 0      0
## 206 206      0 1      0
## 215 215      1 0      0
## 217 217      0 1      0
## 233 233      0 1      0
## 234 234      0 1      0
## 246 246      0 1      0
## 261 261      0 1      0
## 262 262      0 1      0
## 264 264      1 0      0
## 268 268      1 1      1
## 284 284      1 0      1
## 297 297      0 1      0
## 298 298      0 1      0
## 303 303      1 0      1
## 318 318      0 1      0
## 322 322      1 0      0
## 324 324      0 1      0
## 333 333      1 0      0
```

We can see clearly from this plot above that observation 303, 284 together with some other observations are outliers.

(f) Multi collinearity …………………………………………………………………………………………..

```
#CONDITION NUMBER
fit <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 +
    x12 + x13 + x14 + x15 + x16 , data=dat, x=TRUE);
kappa(fit$x)

## [1] 9072.617
```

Since the condition number is greater than 100, there is correlation between two predictor variables in the statistical model, thus, multi collinearity.

```
# COMPUTE VIF USING FUNCTION vif DIRECTLY
vif(fit)

##        x1        x2        x3        x4        x5        x6        x7 
x8 
##  8.020975  8.742184 18.696504 20.587123  5.646413  2.112813  9.524102 
15.644728 
##        x9       x10       x11       x12       x13       x14       x15 
x16 
##  8.801820  3.689858  2.107829  1.377758  1.974733  1.355388  1.522882 
1.177224
```

From the VIF with a threshold of 10 ,it is noticed the variables x3,x4 and x8 should be removed from the model since their VIF is greater than 10 leading to multi collinearity in the model.

5) MODEL DEPLOYMENT

```
test<- read.table(file = "bb92-test.csv",sep=",", header = T, na.strings =
c("NA", "", " "),
stringsAsFactors = T)

dim(test)

## [1] 20 16

head(test)

##   batting.avg   OBP runs hits doubles triples homeruns RBI walks
strike.outs
## 1       0.184 0.296   51   45      19       2        9  50    37
133
## 2       0.218 0.315   67   70      11       0        1   8    25
65
## 3       0.243 0.325   84  102      30       0        4  50    65
107
## 4       0.286 0.138   10  140       4       4        8  11    23
48
## 5       0.194 0.339   38  113      16       3        0  34     5
60
## 6       0.225 0.342   13   55      16       1        2  28    26
49
##   stolen.bases errors free.agency.elig free.agent.91 arb.elig arb.91
## 1           34     10                0             0        0      0
## 2            4      9                1             0        0      0
## 3           41      5                1             0        0      0
## 4            6      0                0             0        0      0
## 5            0      6                0             0        0      0
## 6            0      9                1             0        0      0
```

The data has 16 variables with 20 observations

```
#Predicting the data
pred <- predict(fit.final, test, interval="prediction")

## Warning: 'newdata' had 20 rows but variables found have 337 rows

pred

##           fit       lwr       upr
## 1   7.838913 6.1556207  9.522206
## 2   7.459953 5.8047464  9.115159
## 3   7.858841 6.2033734  9.514308
```

```
## 4   6.815278 5.1469183  8.483637
## 5   7.029187 5.3584203  8.699953
## 6   8.199539 6.5477375  9.851340
## 7   7.875988 6.1878414  9.564134
## 8   4.658668 3.0268139  6.290521
## 9   5.252554 3.6081298  6.896978
## 10  4.743644 3.0903078  6.396981
## 11  4.720062 3.0771557  6.362967
## 12  5.114423 3.4753606  6.753486
## 13  4.315447 2.6795903  5.951304
## 14  5.591786 3.9361741  7.247398
## 15  8.105125 6.4459624  9.764288
## 16  7.368248 5.7129909  9.023505
## 17  6.569617 4.8786251  8.260610
## 18  8.204018 6.5527904  9.855245
## 19  6.969816 5.3132041  8.626428
## 20  4.351390 2.7167771  5.986004
## 21  5.897736 4.1548710  7.640601
## 22  5.688488 3.9394231  7.437554
## 23  6.784425 5.1371560  8.431695
## 24  3.734124 2.0899943  5.378255
## 25  7.953420 6.2453550  9.661484
## 26  7.589585 5.9201945  9.258975
## 27  7.237440 5.5764773  8.898402
## 28  8.039732 6.2916541  9.787809
## 29  6.812756 5.1702895  8.455223
## 30  6.737038 5.0746865  8.399389
## 31  7.543989 5.8752204  9.212758
## 32  7.167012 5.5071491  8.826874
## 33  8.234631 6.5425084  9.926753
## 34  6.702831 5.0503278  8.355335
## 35  6.499282 4.7779707  8.220593
## 36  6.031657 4.3858711  7.677443
## 37  5.382097 3.7387031  7.025492
## 38  5.020727 3.3732130  6.668240
## 39  6.415548 4.7628790  8.068217
## 40  7.333205 5.6783240  8.988087
## 41  8.272082 6.6242625  9.919901
## 42  7.533079 5.8724080  9.193750
## 43  6.812782 5.1439880  8.481576
## 44  6.137079 4.4925693  7.781588
## 45  7.078501 5.4268934  8.730108
## 46  7.123954 5.4749293  8.772978
## 47  6.513039 4.8698410  8.156237
## 48  4.121856 2.4818980  5.761814
## 49  5.293855 3.6586720  6.929039
## 50  5.915312 4.2709402  7.559684
## 51  5.066280 3.4307189  6.701842
## 52  8.106130 6.3568041  9.855455
## 53  7.777332 6.1125894  9.442075
```

```
## 54  6.905957 5.1824100  8.629503
## 55  7.380894 5.7125430  9.049246
## 56  7.836890 6.1856752  9.488105
## 57  7.152109 5.4141780  8.890040
## 58  6.978909 5.3341068  8.623710
## 59  5.795185 4.1561691  7.434201
## 60  5.567248 3.9313900  7.203105
## 61  5.094542 3.4513026  6.737781
## 62  4.625326 2.9885260  6.262126
## 63  6.088938 4.4204548  7.757421
## 64  6.151933 4.5022790  7.801587
## 65  4.390235 2.7440033  6.036467
## 66  7.527389 5.8490394  9.205738
## 67  6.716849 5.0682234  8.365474
## 68  6.934792 5.2642906  8.605294
## 69  7.590159 5.9162393  9.264078
## 70  7.915834 6.2721317  9.559535
## 71  6.818588 5.1667214  8.470454
## 72  7.330710 5.6636576  8.997762
## 73  4.391528 2.7577357  6.025320
## 74  6.872074 5.1858473  8.558302
## 75  6.422989 4.7454710  8.100507
## 76  5.786699 4.0665056  7.506892
## 77  4.802197 3.1548546  6.449540
## 78  5.214035 3.5725791  6.855491
## 79  8.235748 6.5481150  9.923380
## 80  7.787392 6.0143486  9.560436
## 81  7.571803 5.8837494  9.259856
## 82  7.039433 5.3984951  8.680370
## 83  7.998672 6.3547779  9.642567
## 84  7.580492 5.9282932  9.232690
## 85  7.285252 5.6276312  8.942873
## 86  6.392450 4.7376036  8.047297
## 87  6.868355 5.2058380  8.530872
## 88  5.616706 3.9680277  7.265384
## 89  5.208235 3.5617346  6.854736
## 90  4.770079 3.1319446  6.408214
## 91  5.272557 3.6350009  6.910114
## 92  8.066544 6.4112416  9.721847
## 93  7.444051 5.8015720  9.086531
## 94  7.523870 5.8447493  9.202991
## 95  7.960638 6.2973532  9.623923
## 96  7.506324 5.8438761  9.168771
## 97  6.455182 4.8063220  8.104041
## 98  6.811225 5.0917331  8.530717
## 99  5.343905 3.7043866  6.983424
## 100 6.352010 4.6959057  8.008113
## 101 4.690810 3.0530441  6.328576
## 102 4.979297 3.3236154  6.634978
## 103 7.424154 5.7608759  9.087433
```

```
## 104 7.672480 6.0139303  9.331029
## 105 7.290659 5.6215752  8.959742
## 106 5.349084 3.6905909  7.007576
## 107 6.700286 5.0405727  8.360000
## 108 4.913405 3.2624865  6.564323
## 109 6.778069 5.1241120  8.432026
## 110 5.819428 4.1413670  7.497490
## 111 5.104362 3.4498631  6.758861
## 112 3.707689 2.0691085  5.346270
## 113 8.210076 6.5569117  9.863240
## 114 7.580021 5.9264122  9.233630
## 115 7.755819 6.0203300  9.491308
## 116 7.220685 5.5751132  8.866257
## 117 7.827229 6.1760914  9.478367
## 118 7.615880 5.9520511  9.279710
## 119 7.096252 5.4484681  8.744036
## 120 6.681922 5.0234841  8.340360
## 121 6.389273 4.7317478  8.046798
## 122 4.732374 3.0858428  6.378905
## 123 3.945783 2.3105599  5.581006
## 124 5.060991 3.3752710  6.746711
## 125 8.425288 6.7471159 10.103460
## 126 7.683609 5.9454401  9.421778
## 127 7.758866 6.0663392  9.451393
## 128 7.174006 5.5078225  8.840190
## 129 7.348115 5.6899827  9.006247
## 130 6.939943 5.2695742  8.610313
## 131 5.886680 4.2281802  7.545180
## 132 6.832820 5.1749866  8.490653
## 133 5.398482 3.7462988  7.050664
## 134 6.632896 4.9286679  8.337124
## 135 6.667954 4.9388523  8.397056
## 136 4.422668 2.7790792  6.066257
## 137 4.861594 3.2148321  6.508356
## 138 4.869762 3.2338015  6.505721
## 139 8.480303 6.8126118 10.147995
## 140 8.238582 6.5783636  9.898801
## 141 7.724062 6.0350505  9.413074
## 142 7.366415 5.7019692  9.030860
## 143 5.995567 4.3428328  7.648301
## 144 5.770880 4.1173286  7.424431
## 145 6.651268 4.9897985  8.312738
## 146 5.114298 3.4687327  6.759863
## 147 4.593366 2.9595879  6.227144
## 148 3.053769 1.4175345  4.690003
## 149 5.360404 3.7107641  7.010043
## 150 4.899165 3.2656247  6.532705
## 151 2.335560 0.7077549  3.963365
## 152 2.854482 1.2176384  4.491325
## 153 6.543676 4.8820674  8.205284
```

```
## 154 8.206738 6.4975105  9.915966
## 155 7.486602 5.8281901  9.145015
## 156 7.628300 5.9510966  9.305504
## 157 7.345063 5.6614218  9.028704
## 158 5.796343 4.1492283  7.443457
## 159 5.640769 3.9896116  7.291926
## 160 4.326996 2.6884686  5.965524
## 161 5.304984 3.6610974  6.948871
## 162 5.052781 3.4055420  6.700020
## 163 5.050615 3.4040722  6.697157
## 164 5.256228 3.6157587  6.896697
## 165 5.241384 3.5985704  6.884197
## 166 7.783621 6.1102905  9.456952
## 167 7.975015 6.3059428  9.644087
## 168 7.116768 5.4754728  8.758063
## 169 7.907427 6.1955181  9.619336
## 170 6.903777 5.2599224  8.547631
## 171 7.339409 5.6833604  8.995458
## 172 7.086080 5.4361316  8.736029
## 173 7.219525 5.4845142  8.954535
## 174 7.191938 5.5137818  8.870094
## 175 6.098359 4.4415791  7.755139
## 176 4.586651 2.9381444  6.235157
## 177 6.373051 4.7180912  8.028012
## 178 5.619963 3.9831437  7.256783
## 179 7.699875 5.9909266  9.408823
## 180 6.708818 5.0484746  8.369161
## 181 5.152971 3.5058034  6.800138
## 182 6.657622 4.9938067  8.321438
## 183 4.301043 2.6669012  5.935184
## 184 6.219882 4.5409658  7.898797
## 185 5.266194 3.6161373  6.916252
## 186 5.317183 3.6820267  6.952339
## 187 6.117892 4.4611731  7.774611
## 188 5.507286 3.8667533  7.147818
## 189 6.434833 4.7753502  8.094316
## 190 5.150810 3.5131904  6.788429
## 191 5.292314 3.6515037  6.933124
## 192 8.450345 6.7634354 10.137255
## 193 7.984135 6.3128554  9.655415
## 194 7.304844 5.6491795  8.960507
## 195 7.513744 5.8438538  9.183634
## 196 7.076615 5.3599403  8.793289
## 197 7.921592 6.2792454  9.563940
## 198 6.941737 5.2663912  8.617082
## 199 7.025245 5.3643720  8.686119
## 200 6.748545 5.0982201  8.398869
## 201 6.485028 4.7968096  8.173246
## 202 5.689738 4.0275930  7.351882
## 203 5.313585 3.6680507  6.959119
```

```
## 204 5.538189 3.8986498  7.177728
## 205 7.231181 5.5548620  8.907499
## 206 8.048562 6.3274189  9.769704
## 207 7.819689 6.1579765  9.481401
## 208 7.464088 5.7901991  9.137977
## 209 6.626689 4.9851889  8.268190
## 210 7.274513 5.6102660  8.938759
## 211 5.872611 4.2072881  7.537933
## 212 5.912361 4.2478613  7.576861
## 213 6.360875 4.6874473  8.034302
## 214 6.040868 4.3766029  7.705132
## 215 7.265642 5.6143352  8.916949
## 216 5.490525 3.8417711  7.139280
## 217 5.817418 4.1107891  7.524047
## 218 8.668105 6.9973871 10.338823
## 219 7.749680 6.0825102  9.416850
## 220 7.398991 5.7403853  9.057597
## 221 7.121186 5.4777446  8.764628
## 222 7.439586 5.7728263  9.106346
## 223 7.066983 5.4056591  8.728306
## 224 7.407073 5.7388975  9.075248
## 225 5.254424 3.6104700  6.898379
## 226 5.030484 3.3807761  6.680191
## 227 5.379437 3.7082821  7.050593
## 228 4.982968 3.3359023  6.630034
## 229 5.300458 3.6550102  6.945906
## 230 4.995389 3.3165418  6.674236
## 231 8.171807 6.5035360  9.840078
## 232 7.346171 5.6983663  8.993975
## 233 7.802405 6.0889155  9.515895
## 234 8.018140 6.3123612  9.723918
## 235 7.455890 5.7849384  9.126842
## 236 7.694608 6.0512141  9.338001
## 237 6.539782 4.8886311  8.190932
## 238 6.926108 5.2448223  8.607394
## 239 6.682226 5.0327419  8.331709
## 240 7.383956 5.7340680  9.033845
## 241 6.652631 4.9807452  8.324516
## 242 5.768952 4.1172506  7.420653
## 243 5.856754 4.2133470  7.500160
## 244 7.504795 5.8439449  9.165645
## 245 7.063872 5.4246150  8.703130
## 246 7.519636 5.7813504  9.257923
## 247 5.690155 4.0486659  7.331643
## 248 4.929557 3.2822192  6.576894
## 249 4.340837 2.7073070  5.974366
## 250 4.219304 2.5841434  5.854464
## 251 4.741305 3.0862671  6.396343
## 252 5.641134 4.0000675  7.282200
## 253 7.792957 6.1277946  9.458118
```

```
## 254 7.771573 6.0766058  9.466541
## 255 7.498773 5.8205800  9.176965
## 256 7.446118 5.7801176  9.112118
## 257 7.055246 5.3853298  8.725163
## 258 7.320123 5.6580522  8.982194
## 259 7.705626 6.0263554  9.384897
## 260 6.742863 5.0911217  8.394604
## 261 6.720338 4.9884085  8.452268
## 262 5.550793 3.8426273  7.258958
## 263 4.993855 3.3458201  6.641890
## 264 3.501030 1.8680056  5.134055
## 265 5.937689 4.2913199  7.584059
## 266 5.666947 4.0243847  7.309509
## 267 3.950156 2.3136594  5.586652
## 268 6.601316 4.7842664  8.418365
## 269 8.054816 6.3936527  9.715980
## 270 7.310586 5.6621529  8.959019
## 271 7.413826 5.7260686  9.101583
## 272 7.600749 5.9413144  9.260183
## 273 7.286464 5.6197266  8.953202
## 274 6.820774 5.1648107  8.476737
## 275 7.200130 5.5506879  8.849572
## 276 5.585344 3.9379538  7.232734
## 277 6.506367 4.8556025  8.157131
## 278 5.745642 4.0919809  7.399303
## 279 5.445053 3.8030995  7.087007
## 280 5.863727 4.1789148  7.548540
## 281 4.977406 3.3320177  6.622795
## 282 5.341032 3.7047399  6.977323
## 283 4.578497 2.9416380  6.215355
## 284 7.832841 6.1573273  9.508355
## 285 7.943133 6.2851862  9.601080
## 286 8.683831 7.0081086 10.359553
## 287 8.092010 6.4285765  9.755443
## 288 7.842170 6.1690529  9.515286
## 289 7.145855 5.4998961  8.791813
## 290 7.327741 5.6712105  8.984271
## 291 8.060087 6.3995256  9.720647
## 292 7.130395 5.4794922  8.781298
## 293 6.181757 4.5377412  7.825772
## 294 5.684956 4.0232100  7.346701
## 295 5.832821 4.1935765  7.472065
## 296 5.465106 3.8254471  7.104764
## 297 8.154664 6.4288492  9.880479
## 298 7.798481 6.0829916  9.513969
## 299 7.059835 5.3947162  8.724954
## 300 8.164996 6.5053509  9.824641
## 301 7.727647 6.0528458  9.402448
## 302 8.210727 6.5592722  9.862183
## 303 2.681841 1.0265572  4.337125
```

```
## 304 6.898440 5.2456879  8.551191
## 305 6.104891 4.4573098  7.752472
## 306 7.081892 5.4356786  8.728106
## 307 7.222852 5.5552446  8.890460
## 308 5.502384 3.8606814  7.144087
## 309 4.957307 3.3192405  6.595373
## 310 4.262237 2.6239624  5.900511
## 311 7.299152 5.6373963  8.960908
## 312 7.319644 5.6676591  8.971629
## 313 7.022160 5.3488904  8.695430
## 314 6.407602 4.7630463  8.052157
## 315 8.535474 6.8494017 10.221545
## 316 7.440495 5.7683028  9.112687
## 317 7.349223 5.6886282  9.009817
## 318 6.139633 4.4361145  7.843152
## 319 4.866033 3.2182378  6.513828
## 320 4.981343 3.3434184  6.619268
## 321 4.255658 2.6211703  5.890145
## 322 7.100886 5.4385174  8.763255
## 323 8.510207 6.8177701 10.202645
## 324 7.791542 6.0399912  9.543093
## 325 8.965498 7.2970905 10.633905
## 326 7.540392 5.8589540  9.221831
## 327 7.044025 5.3917036  8.696346
## 328 6.929344 5.2493717  8.609316
## 329 6.425452 4.7585435  8.092360
## 330 6.108324 4.4635717  7.753076
## 331 4.159408 2.5213046  5.797511
## 332 4.749519 3.0985283  6.400509
## 333 2.195588 0.5682532  3.822924
## 334 5.229186 3.5909347  6.867437
## 335 4.816614 3.1639271  6.469301
## 336 4.996264 3.3501882  6.642340
## 337 5.987825 4.3109906  7.664660
```

```
test.pred<-data.frame(pred)
test.pred.exp<- exp(test.pred)
test.pred.exp
```

```
##             fit         lwr         upr
## 1   2537.446400  471.359324 13659.71543
## 2   1737.065872  331.871005  9092.08035
## 3   2588.517972  494.414074 13552.25438
## 4    911.669716  171.900922  4835.00414
## 5   1129.111781  212.389170  6002.62912
## 6   3639.271215  697.663896 18983.77579
## 7   2633.286141  486.794160 14244.61605
## 8    105.495432   20.631394   539.43451
## 9    191.053619   36.896982   989.28104
## 10   114.852009   21.983843   600.03084
```

```
## 11   112.175159   21.696603   579.96482
## 12   166.404753   32.309476   857.04089
## 13    74.847078   14.579119   384.25402
## 14   268.214206   51.222255  1404.44540
## 15  3311.395656  630.152837 17401.08200
## 16  1584.854649  302.775299  8295.80309
## 17   713.096911  131.449808  3868.45147
## 18  3655.608302  701.198094 19058.05531
## 19  1064.026774  202.999624  5577.11857
## 20    77.586264   15.131477   397.82161
## 21   364.211924   63.743739  2080.99380
## 22   295.446669   51.388947  1698.58968
## 23   883.971928  170.230935  4590.27243
## 24    41.851367    8.084869   216.64383
## 25  2845.287847  515.612256 15701.06769
## 26  1977.492356  372.484137 10498.36928
## 27  1390.529061  264.139491  7320.26499
## 28  3101.781059  540.045870 17815.23805
## 29   909.373986  175.965764  4699.55648
## 30   843.059850  159.922046  4444.35228
## 31  1889.352141  356.103145 10024.20665
## 32  1295.965880  246.447537  6814.94968
## 33  3769.247343  694.025292 20470.76050
## 34   814.709133  156.073614  4252.80709
## 35   664.664038  118.862902  3716.70451
## 36   416.404563   80.308150  2159.09296
## 37   217.477925   42.043429  1124.94744
## 38   151.521360   29.172106   787.00944
## 39   611.275532  117.082518  3191.40538
## 40  1530.278838  292.458845  8007.12088
## 41  3913.087133  753.148524 20330.98443
## 42  1868.851113  355.103051  9835.46741
## 43   909.397372  171.397940  4825.04970
## 44   462.699850   89.350719  2396.07643
## 45  1186.188958  227.441577  6186.39855
## 46  1241.348749  238.633602  6457.37526
## 47   673.871310  130.300193  3485.04889
## 48    61.673599   11.963951   317.92448
## 49   199.109577   38.809771  1021.51141
## 50   370.669998   71.588912  1919.23921
## 51   158.583344   30.898849   813.90335
## 52  3314.724050  576.401286 19062.05937
## 53  2385.900965  451.506335 12607.84838
## 54   998.202979  178.111541  5594.29883
## 55  1605.024784  302.639702  8512.11703
## 56  2532.317261  485.740813 13201.75398
## 57  1276.795818  224.567873  7259.30892
## 58  1073.745800  207.287507  5561.98518
## 59   328.712950   63.826542  1692.90393
## 60   261.712772   50.977789  1343.59642
```

```
## 61   163.129043   31.541451   843.68613
## 62   102.036011   19.856392   524.33230
## 63   440.952900   83.134086  2338.86566
## 64   469.624213   90.222513  2444.47748
## 65    80.659369   15.549108   418.41202
## 66  1858.246647  346.900980  9954.08144
## 67   826.209904  158.891786  4296.14912
## 68  1027.405917  193.309125  5460.49193
## 69  1978.627858  371.013804 10552.08232
## 70  2740.329822  529.605127 14179.25763
## 71   914.692169  175.339035  4771.67998
## 72  1526.465136  288.200833  8084.97252
## 73    80.763727   15.764108   413.77411
## 74   964.948264  178.724817  5209.82575
## 75   615.840984  115.061986  3296.13741
## 76   325.935267   58.352701  1820.54639
## 77   121.777712   23.449628   632.41138
## 78   183.834355   35.608311   949.07816
## 79  3773.460338  697.927327 20401.84182
## 80  2410.024332  409.259171 14192.02718
## 81  1942.638892  359.153332 10507.61758
## 82  1140.740312  221.073461  5886.22647
## 83  2977.003165  575.234542 15406.84224
## 84  1959.591897  375.513039 10226.01082
## 85  1458.628665  278.002797  7653.15170
## 86   597.318497  114.160302  3125.33675
## 87   961.365615  182.333607  5068.86174
## 88   274.981998   52.880135  1429.93393
## 89   182.771203   35.224243   948.36142
## 90   117.928576   22.918503   606.80877
## 91   194.913768   37.901887  1002.36108
## 92  3186.071943  608.648893 16678.01346
## 93  1709.662660  330.819210  8835.47968
## 94  1851.720100  345.415933  9926.77814
## 95  2865.900599  543.132429 15122.25344
## 96  1819.512208  345.114434  9592.83169
## 97   635.989287  122.281039  3307.80943
## 98   907.982447  162.671544  5068.07832
## 99   209.328570   40.625122  1078.60478
## 100  573.644290  109.497939  3005.24169
## 101  108.941367   21.179719   560.35784
## 102  145.372102   27.760536   761.26226
## 103 1675.981524  317.626428  8843.45200
## 104 2148.402483  409.088010 11282.73896
## 105 1466.536097  276.324312  7783.34742
## 106  210.415364   40.068518  1104.97286
## 107  812.638437  154.558511  4272.69403
## 108  136.102026   26.114391   709.33156
## 109  878.370764  168.024876  4591.79151
## 110  336.779485   62.888733  1803.50940
```

```
## 111  164.738908   31.496082   861.65981
## 112   40.759509    7.917761   209.82416
## 113 3677.820929  704.093900 19211.02681
## 114 1958.670458  374.807375 10235.63095
## 115 2335.121233  411.714442 13244.10956
## 116 1367.425273  263.779417  7088.69516
## 117 2507.970541  481.107811 13073.81856
## 118 2030.181498  384.541257 10718.32175
## 119 1207.432976  232.401883  6273.16085
## 120  797.851155  151.939751  4189.59793
## 121  595.423635  113.493757  3123.77804
## 122  113.564825   21.885904   589.28202
## 123   51.716806   10.080067   265.33831
## 124  157.746750   29.232206   851.25418
## 125 4560.958416  851.599116 24427.38757
## 126 2172.445643  382.007462 12354.52324
## 127 2342.247274  431.099618 12725.88067
## 128 1305.062451  246.613549  6906.30344
## 129 1553.265593  295.888493  8153.86221
## 130 1032.711804  194.333200  5487.96433
## 131  360.207459   68.592294  1891.60336
## 132  927.803256  176.794245  4869.04357
## 133  221.070466   42.363995  1153.62469
## 134  759.679175  138.195304  4176.06410
## 135  786.784175  139.609925  4433.99232
## 136   83.318312   16.104186   431.06440
## 137  129.230005   24.899110   670.72253
## 138  130.289840   25.375941   668.95814
## 139 4818.910800  909.242444 25539.83424
## 140 3784.170992  719.361213 19906.48068
## 141 2262.129861  417.819905 12247.45744
## 142 1581.951890  299.456505  8357.04597
## 143  401.644401   76.925145  2097.08056
## 144  320.819817   61.395011  1676.44492
## 145  773.765075  146.906824  4075.45665
## 146  166.383901   32.096040   862.52393
## 147   98.826498   19.290021   506.30721
## 148   21.195074    4.126933   108.85352
## 149  212.810834   40.885033  1107.70244
## 150  134.177691   26.196471   687.25490
## 151   10.335243    2.029430    52.63412
## 152   17.365436    3.379198    89.23963
## 153  694.835920  131.903078  3660.24027
## 154 3665.566487  663.487800 20251.12996
## 155 1783.980368  339.743212  9367.62190
## 156 2055.553626  384.174392 10998.39239
## 157 1548.532543  287.557215  8339.04667
## 158  329.093738   63.385067  1708.64675
## 159  281.679188   54.033900  1468.39603
## 160   75.716499   14.709133   389.75706
```

```
## 161  201.337825   38.904012  1041.97274
## 162  156.456994   30.130622   812.42237
## 163  156.118374   30.086370   810.09928
## 164  191.756740   37.179542   989.00215
## 165  188.931346   36.545952   976.71702
## 166 2400.953496  450.469569 12796.81934
## 167 2907.400501  547.817832 15430.27113
## 168 1232.460622  238.763325  6361.77765
## 169 2717.390018  490.545510 15053.05492
## 170  996.029326  192.466553  5154.52895
## 171 1539.802241  293.935523  8066.36408
## 172 1195.213728  229.552468  6223.13439
## 173 1365.839869  240.931868  7742.92983
## 174 1328.675868  248.087583  7115.95292
## 175  445.126712   84.908919  2333.53331
## 176   98.165117   18.880778   510.38099
## 177  585.842788  111.954346  3065.64046
## 178  275.879303   53.685540  1417.68881
## 179 2208.071688  399.784863 12195.51070
## 180  819.601185  155.784655  4312.01713
## 181  172.944460   33.308194   897.97081
## 182  778.697176  147.496838  4111.06637
## 183   73.776687   14.395292   378.10970
## 184  502.643720   93.781328  2694.04063
## 185  193.677505   37.193621  1008.53250
## 186  203.808925   39.726827  1045.59263
## 187  453.906928   86.589030  2379.41803
## 188  246.481179   47.786986  1271.32881
## 189  623.178424  118.551820  3275.79406
## 190  172.571136   33.555150   887.51792
## 191  198.802921   38.532564  1025.69353
## 192 4676.686047  865.610800 25267.00495
## 193 2934.038626  551.617800 15606.06394
## 194 1487.487155  284.058313  7789.30923
## 195 1833.063910  345.106750  9736.47515
## 196 1183.953516  212.712247  6589.86939
## 197 2756.156714  533.386019 14241.84280
## 198 1034.565391  193.715624  5525.24121
## 199 1124.670377  213.657008  5920.15900
## 200  852.816605  163.730231  4442.03954
## 201  655.257055  121.123371  3544.83041
## 202  295.816024   56.125655  1559.12873
## 203  203.076902   39.175466  1052.70549
## 204  254.217184   49.335792  1309.92883
## 205 1381.852890  258.491299  7387.16321
## 206 3129.290746  559.710059 17495.59510
## 207 2489.130467  472.471067 13113.54477
## 208 1744.263833  327.078143  9301.92488
## 209  754.978452  146.231195  3897.88557
## 210 1443.047690  273.216912  7621.73404
```

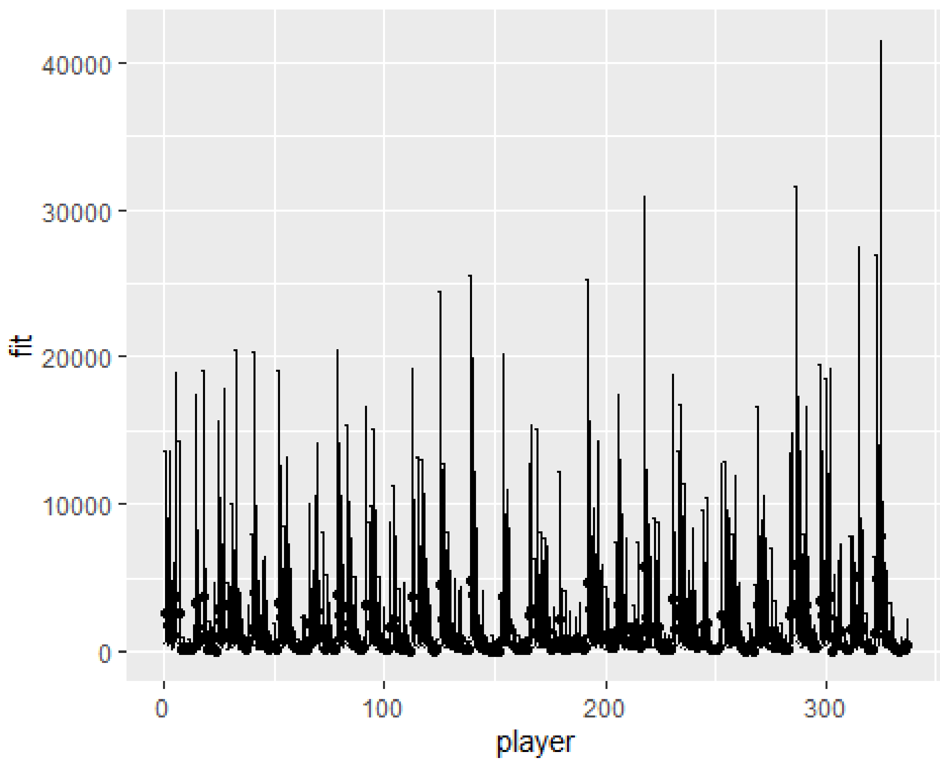
```
## 211  355.174983   67.174120  1877.94448
## 212  369.577673   69.955641  1952.48954
## 213  578.752367  108.575670  3084.98490
## 214  420.257502   79.567275  2219.71115
## 215 1430.303875  274.330954  7457.30346
## 216  242.384505   46.607948  1260.51996
## 217  336.103194   60.994829  1852.04809
## 218 5814.471890 1093.771504 30909.63993
## 219 2320.829717  438.127589 12293.79459
## 220 1634.334746  311.184299  8583.49881
## 221 1237.917951  239.306370  6403.67765
## 222 1702.045360  321.444963  9012.29988
## 223 1172.604674  222.662936  6175.26089
## 224 1647.596715  310.721639  8736.35627
## 225  191.411275   36.983431   990.66732
## 226  153.007006   29.393575   796.47147
## 227  216.900214   40.783683  1153.54227
## 228  145.906779   28.103730   757.50758
## 229  200.428553   38.667914  1038.88731
## 230  147.730375   27.564861   791.74220
## 231 3539.734422  667.497755 18771.17890
## 232 1550.248692  298.379548  8054.40929
## 233 2446.478702  440.942944 13573.76984
## 234 3035.525151  551.345257 16712.60036
## 235 1730.023572  325.361998  9198.92788
## 236 2196.471632  424.628252 11361.67368
## 237  692.135430  132.771701  3608.08402
## 238 1018.522060  189.582114  5471.96760
## 239  798.093384  153.352916  4153.51117
## 240 1609.946665  309.224642  8382.02366
## 241  774.819909  145.582828  4123.74109
## 242  320.201899   61.390223  1670.12353
## 243  349.587392   67.582363  1808.33194
## 244 1816.732298  345.138209  9562.88280
## 245 1168.962977  226.923965  6021.72821
## 246 1843.896873  324.196684 10487.32405
## 247  295.939362   57.320931  1527.89049
## 248  138.318173   26.634815   718.30486
## 249   76.771744   14.988856   393.21885
## 250   67.986128   13.251932   348.78789
## 251  114.583631   21.895193   599.64799
## 252  281.781946   54.601836  1454.18305
## 253 2423.472014  458.424055 12811.75482
## 254 2372.200334  435.548352 12920.11415
## 255 1805.824569  337.167538  9671.75664
## 256 1713.199514  323.797252  9064.47648
## 257 1158.922963  218.182054  6155.87951
## 258 1510.389806  286.589878  7960.07653
## 259 2220.807245  414.202678 11907.17752
## 260  847.984881  162.572123  4423.13446
```

```
## 261  829.097788  146.702765  4685.68635
## 262  257.441500   46.647874  1420.77485
## 263  147.503957   28.383843   766.54233
## 264   33.149594    6.475369   169.70393
## 265  379.058081   73.062840  1966.59517
## 266  289.150326   55.945872  1494.44290
## 267   51.943444   10.111359   266.84064
## 268  736.063015  119.613587  4529.49182
## 269 3148.924623  598.037058 16580.45458
## 270 1496.053683  287.767516  7777.72507
## 271 1658.760166  306.760902  8969.47842
## 272 1999.692311  380.434658 10511.05429
## 273 1460.397789  275.813975  7732.60928
## 274  916.694289  175.004322  4801.75809
## 275 1339.604574  257.414562  6971.40209
## 276  266.491964   51.313496  1384.00173
## 277  669.389904  128.458067  3488.16431
## 278  312.824334   59.858349  1634.84402
## 279  231.609646   44.839950  1196.32221
## 280  352.033848   65.294960  1897.96932
## 281  145.097523   27.994769   752.04377
## 282  208.727943   40.639476  1072.04518
## 283   97.367893   18.946855   500.37361
## 284 2522.084764  472.164450 13471.81381
## 285 2816.170257  536.564215 14780.73769
## 286 5906.629997 1105.561457 31557.06786
## 287 3268.248964  619.291744 17247.85031
## 288 2545.721920  477.733429 13565.51521
## 289 1268.835264  244.666510  6580.15242
## 290 1521.939308  290.385837  7976.62614
## 291 3165.564013  601.559580 16658.02664
## 292 1249.370517  239.724951  6511.32344
## 293  483.841233   93.479408  2504.31987
## 294  294.404747   55.880192  1551.07116
## 295  341.320181   66.259346  1758.23445
## 296  236.300827   45.853295  1217.75503
## 297 3479.570456  619.460688 19545.08298
## 298 2436.896424  438.338583 13547.66478
## 299 1164.253427  220.239644  6154.59605
## 300 3515.707146  668.710265 18483.63540
## 301 2270.253771  425.321704 12118.00889
## 302 3680.218467  705.757890 19190.72837
## 303   14.611972    2.791439    76.48734
## 304  990.727554  189.746305  5172.91276
## 305  448.043711   86.255153  2327.31798
## 306 1190.218449  229.448511  6174.02115
## 307 1370.392031  258.590212  7262.35654
## 308  245.275965   47.497706  1266.59378
## 309  142.210251   27.639350   731.70156
## 310   70.968557   13.790259   365.22419
```

```
## 311 1479.045290  280.730824  7792.42885
## 312 1509.666611  289.356381  7876.42308
## 313 1121.206038  210.374739  5975.54147
## 314  606.437455  117.102116  3140.56141
## 315 5092.242573  943.316348 27489.11803
## 316 1703.592776  319.994193  9069.62818
## 317 1554.987076  295.487992  8183.02224
## 318  463.883455   84.446187  2548.22470
## 319  129.804963   24.984054   674.40330
## 320  145.669924   28.315755   749.39648
## 321   70.503181   13.751808   361.45782
## 322 1213.041533  230.100788  6394.89232
## 323 4965.192828  913.944753 26974.43117
## 324 2420.046575  419.889345 13948.02104
## 325 7828.278960 1475.999225 41518.95912
## 326 1882.568588  350.357477 10115.56687
## 327 1145.990917  219.577140  5981.01962
## 328 1021.823275  190.446578  5482.49707
## 329  617.359605  116.576004  3269.39395
## 330  449.584569   86.796973  2328.72504
## 331   64.033587   12.444821   329.47844
## 332  115.528645   22.165307   602.15127
## 333    8.985286    1.765181    45.73773
## 334  186.640788   36.267961   960.48365
## 335  123.546078   23.663343   645.03285
## 336  147.859761   28.508099   766.88763
## 337  398.547010   74.514267  2131.66856
```

The output above SHOWS the log transform of the predicted values and their corresponding Confidence Intervals, and it is clear that all predicted values lies within each corresponding confidence interval.

```
#Making the error plot
library(ggplot2)
pred <- predict(fit.final, test, interval="prediction");

## Warning: 'newdata' had 20 rows but variables found have 337 rows

dat.plot <- data.frame(player=1:337, exp(pred)); names(dat.plot)

## [1] "player" "fit"    "lwr"    "upr"

ggplot(dat.plot, aes(x=player, y=fit)) +
geom_errorbar(aes(ymin=lwr, ymax=upr)) + geom_point()
```



The dot plot of the error indicates there exist slight error in the model