DATA 621

Charls Joseph, Mary Anna Kivenson, Elina Azrilyan, Sunny Mehta, Vinayak Patel

2020-02-29

# HOMEWORK #1

## Overview:

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season. Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided).

## Deliverables:

* A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
* Assigned predictions (the number of wins for the team) for the evaluation data set.
* Include your R statistical programming code in an Appendix.

1. DATA EXPLORATION

### Data acquisition

First, we need to explore our given data set. I have published the original data sets in my github account

#### Read Data

Here, we read the dataset and shorten the feature names for better readibility in visualizations.

df <- read.csv("https://raw.githubusercontent.com/mkivenson/Business-Analytics-Data-Mining/master/Moneyball%20Regression/moneyball-training-data.csv")[-1]  
names(df) <- sub("TEAM\_", "", names(df))  
names(df) <- sub("BATTING\_", "bt\_", names(df))  
names(df) <- sub("BASERUN\_", "br\_", names(df))  
names(df) <- sub("FIELDING\_", "fd\_", names(df))  
names(df) <- sub("PITCHING\_", "ph\_", names(df))  
names(df) <- sub("TARGET\_", "", names(df))  
head(df)

## WINS bt\_H bt\_2B bt\_3B bt\_HR bt\_BB bt\_SO br\_SB br\_CS bt\_HBP ph\_H ph\_HR  
## 1 39 1445 194 39 13 143 842 NA NA NA 9364 84  
## 2 70 1339 219 22 190 685 1075 37 28 NA 1347 191  
## 3 86 1377 232 35 137 602 917 46 27 NA 1377 137  
## 4 70 1387 209 38 96 451 922 43 30 NA 1396 97  
## 5 82 1297 186 27 102 472 920 49 39 NA 1297 102  
## 6 75 1279 200 36 92 443 973 107 59 NA 1279 92  
## ph\_BB ph\_SO fd\_E fd\_DP  
## 1 927 5456 1011 NA  
## 2 689 1082 193 155  
## 3 602 917 175 153  
## 4 454 928 164 156  
## 5 472 920 138 168  
## 6 443 973 123 149

#### Summary

First, we take a look at a summary of the data. A few things of interest are revealed:

* bt\_SO, br\_SB, br\_CS, bt\_HBP, ph\_SO, and fd\_DP have missing values
* The max values of ph\_H, ph\_BB, ph\_SO, and fd\_E seem abnormally high

summary(df)

## WINS bt\_H bt\_2B bt\_3B   
## Min. : 0.00 Min. : 891 Min. : 69.0 Min. : 0.00   
## 1st Qu.: 71.00 1st Qu.:1383 1st Qu.:208.0 1st Qu.: 34.00   
## Median : 82.00 Median :1454 Median :238.0 Median : 47.00   
## Mean : 80.79 Mean :1469 Mean :241.2 Mean : 55.25   
## 3rd Qu.: 92.00 3rd Qu.:1537 3rd Qu.:273.0 3rd Qu.: 72.00   
## Max. :146.00 Max. :2554 Max. :458.0 Max. :223.00   
##   
## bt\_HR bt\_BB bt\_SO br\_SB   
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 42.00 1st Qu.:451.0 1st Qu.: 548.0 1st Qu.: 66.0   
## Median :102.00 Median :512.0 Median : 750.0 Median :101.0   
## Mean : 99.61 Mean :501.6 Mean : 735.6 Mean :124.8   
## 3rd Qu.:147.00 3rd Qu.:580.0 3rd Qu.: 930.0 3rd Qu.:156.0   
## Max. :264.00 Max. :878.0 Max. :1399.0 Max. :697.0   
## NA's :102 NA's :131   
## br\_CS bt\_HBP ph\_H ph\_HR   
## Min. : 0.0 Min. :29.00 Min. : 1137 Min. : 0.0   
## 1st Qu.: 38.0 1st Qu.:50.50 1st Qu.: 1419 1st Qu.: 50.0   
## Median : 49.0 Median :58.00 Median : 1518 Median :107.0   
## Mean : 52.8 Mean :59.36 Mean : 1779 Mean :105.7   
## 3rd Qu.: 62.0 3rd Qu.:67.00 3rd Qu.: 1682 3rd Qu.:150.0   
## Max. :201.0 Max. :95.00 Max. :30132 Max. :343.0   
## NA's :772 NA's :2085   
## ph\_BB ph\_SO fd\_E fd\_DP   
## Min. : 0.0 Min. : 0.0 Min. : 65.0 Min. : 52.0   
## 1st Qu.: 476.0 1st Qu.: 615.0 1st Qu.: 127.0 1st Qu.:131.0   
## Median : 536.5 Median : 813.5 Median : 159.0 Median :149.0   
## Mean : 553.0 Mean : 817.7 Mean : 246.5 Mean :146.4   
## 3rd Qu.: 611.0 3rd Qu.: 968.0 3rd Qu.: 249.2 3rd Qu.:164.0   
## Max. :3645.0 Max. :19278.0 Max. :1898.0 Max. :228.0   
## NA's :102 NA's :286

#### Dimensions

Let’s see the dimensions of our moneyball training data set.

dim(df)

## [1] 2276 16

The training data has 17 columns and 2,276 rows.

The explanatory columns are broken down into four categories:

* Batting
* Base run
* Pitching
* Fielding

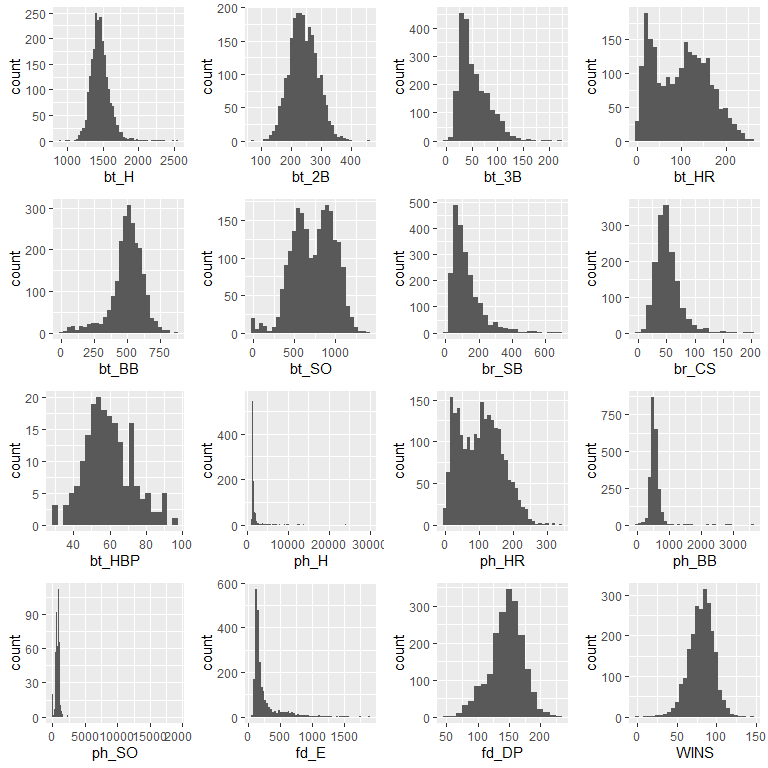
Below you will see a preview of the columns and the first few observations broken down into these four categories.

#### Histogram

Next, we create histograms of each of the features and target variable.

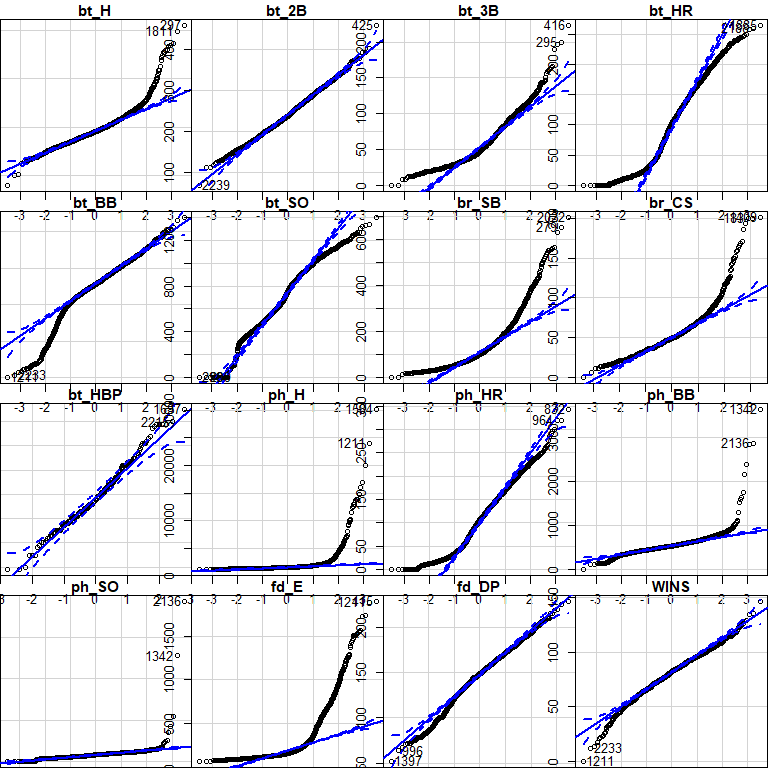
* bt\_H, bt\_2B, bt\_BB, br\_CS, bt\_HBP, fd\_DP, WINS all have normal distributions
* ph\_H, ph\_BB, ph\_SO, and fd\_E are highly right-skewed

grid.arrange(ggplot(df, aes(bt\_H)) + geom\_histogram(binwidth = 30),  
 ggplot(df, aes(bt\_2B)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(bt\_3B)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(bt\_HR)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(bt\_BB)) + geom\_histogram(binwidth = 30),  
 ggplot(df, aes(bt\_SO)) + geom\_histogram(binwidth = 50),  
 ggplot(df, aes(br\_SB)) + geom\_histogram(binwidth = 30),  
 ggplot(df, aes(br\_CS)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(bt\_HBP)) + geom\_histogram(binwidth = 3),  
 ggplot(df, aes(ph\_H)) + geom\_histogram(binwidth = 100),  
 ggplot(df, aes(ph\_HR)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(ph\_BB)) + geom\_histogram(binwidth = 100),  
 ggplot(df, aes(ph\_SO)) + geom\_histogram(binwidth = 30),  
 ggplot(df, aes(fd\_E)) + geom\_histogram(binwidth = 30),  
 ggplot(df, aes(fd\_DP)) + geom\_histogram(binwidth = 10),  
 ggplot(df, aes(WINS)) + geom\_histogram(binwidth = 5),  
 ncol=4)



#### QQ Plots

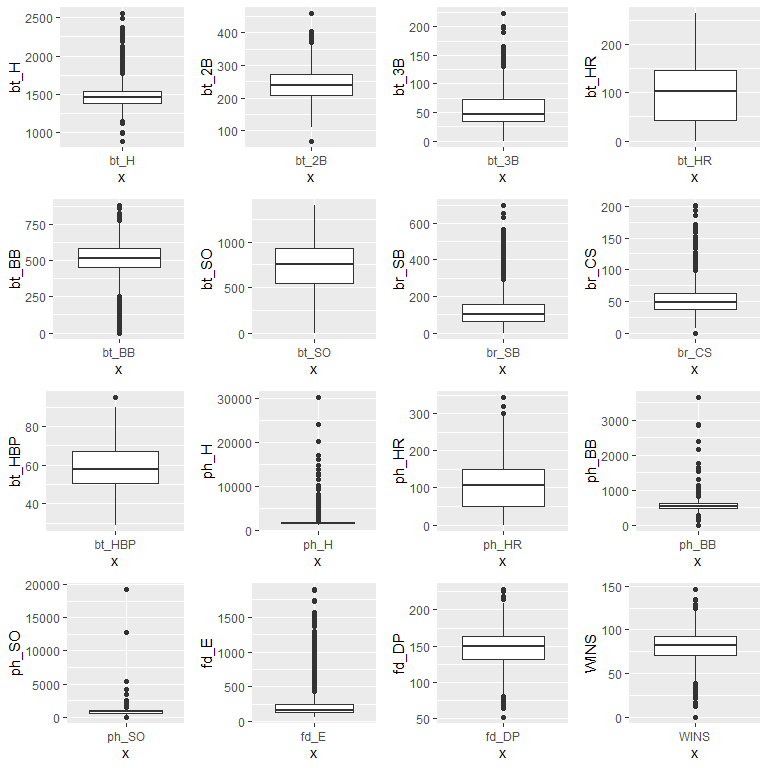
* Most of the features are not lined up with the theoretical QQ plot, however this will be addressed by the models we build.



#### Boxplot

* Most of the boxplots shown below reflect a long right tail with many outliers.

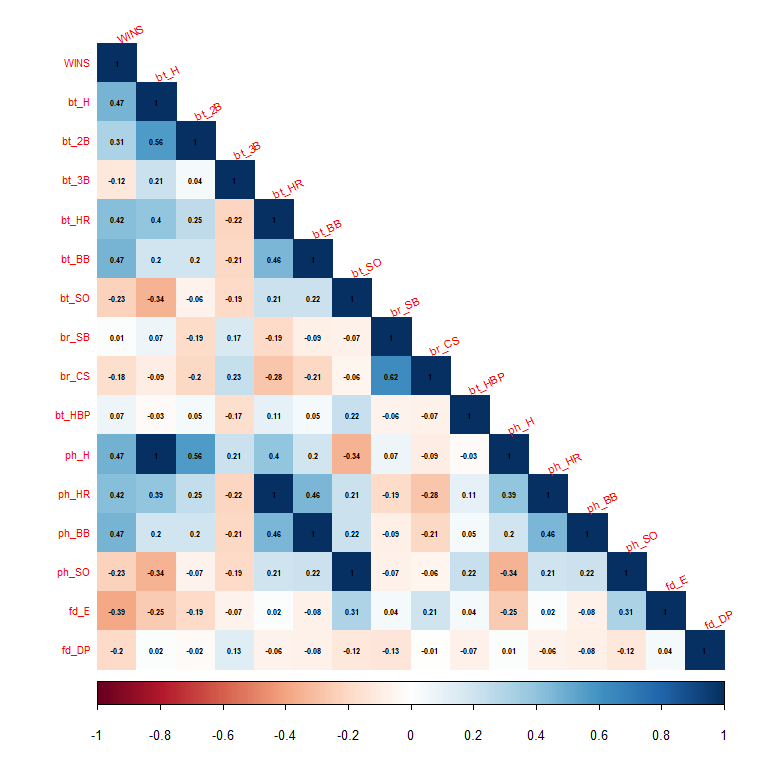
grid.arrange(ggplot(df, aes(x = "bt\_H", y = bt\_H))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_2B", y = bt\_2B))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_3B", y = bt\_3B))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_HR", y = bt\_HR))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_BB", y = bt\_BB))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_SO", y = bt\_SO))+geom\_boxplot(),  
 ggplot(df, aes(x = "br\_SB", y = br\_SB))+geom\_boxplot(),  
 ggplot(df, aes(x = "br\_CS", y = br\_CS))+geom\_boxplot(),  
 ggplot(df, aes(x = "bt\_HBP", y = bt\_HBP))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_H", y = ph\_H))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_HR", y = ph\_HR))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_BB", y = ph\_BB))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_SO", y = ph\_SO))+geom\_boxplot(),  
 ggplot(df, aes(x = "fd\_E", y = fd\_E))+geom\_boxplot(),  
 ggplot(df, aes(x = "fd\_DP", y = fd\_DP))+geom\_boxplot(),  
 ggplot(df, aes(x = "WINS", y = WINS))+geom\_boxplot(),  
 ncol=4)



#### Correlation Plot

* There is a strong positive correlation between ph\_H and bt\_H
* There is a strong positive correlation between ph\_HR and bt\_HR
* There is a strong positive correlation between ph\_BB and bt\_BB
* There is a strong positive correlation between ph\_SO and bt\_SO
* There seems to be a weak correlation between bt\_HBP/br\_SB and Wins

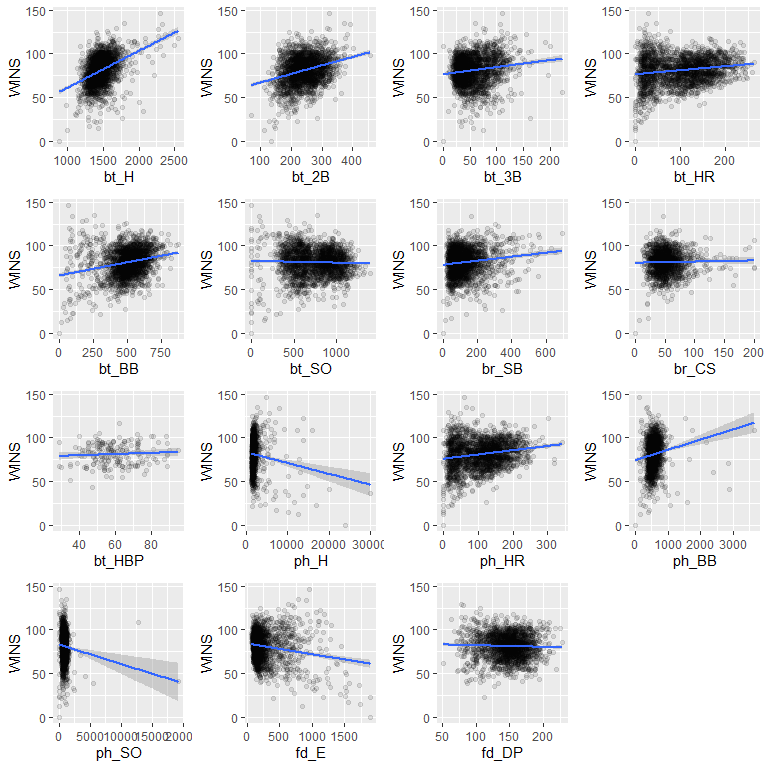
corrplot(cor(df, use = "complete.obs"), method ="color", type="lower", addrect = 1, number.cex = 0.5, sig.level = 0.30,  
 addCoef.col = "black", # Add coefficient of correlation  
 tl.srt = 25, # Text label color and rotation  
 tl.cex = 0.7,  
 diag = TRUE)



#### Scatter Plots

Here, we see a scatter plot of each of the feature variables with the target variable.

grid.arrange(ggplot(df, aes(bt\_H, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_2B, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_3B, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_HR, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_BB, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_SO, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(br\_SB, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(br\_CS, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(bt\_HBP, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(ph\_H, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(ph\_HR, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(ph\_BB, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(ph\_SO, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(fd\_E, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ggplot(df, aes(fd\_DP, WINS)) + geom\_point(alpha = 1/10)+geom\_smooth(method=lm),  
 ncol=4)



1. Data Preparation

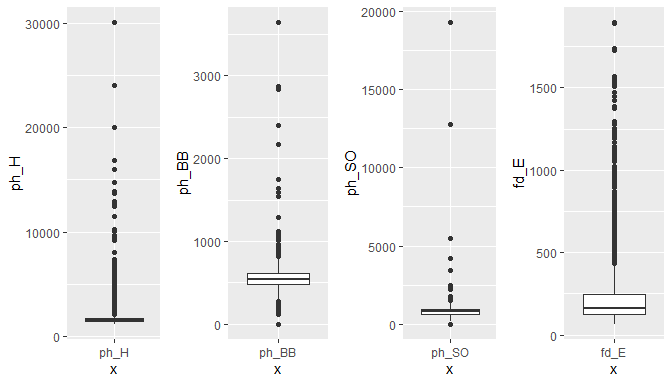
### Outliers

#### Extreme Values

While exploring the data, we noticed that the max values of ph\_H, ph\_BB, ph\_SO, and fd\_E seem abnormally high.

We see that the record for most hits in a season by team (ph\_H) was set at 1,724 in 1921. However, we also know that the datapoints were normalized for 162 games in a season. To take a moderate approach, we will remove the some of the most egggregious outliers that are seen in these variables.

grid.arrange(ggplot(df, aes(x = "ph\_H", y = ph\_H))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_BB", y = ph\_BB))+geom\_boxplot(),  
 ggplot(df, aes(x = "ph\_SO", y = ph\_SO))+geom\_boxplot(),  
 ggplot(df, aes(x = "fd\_E", y = fd\_E))+geom\_boxplot(),  
 ncol=4)

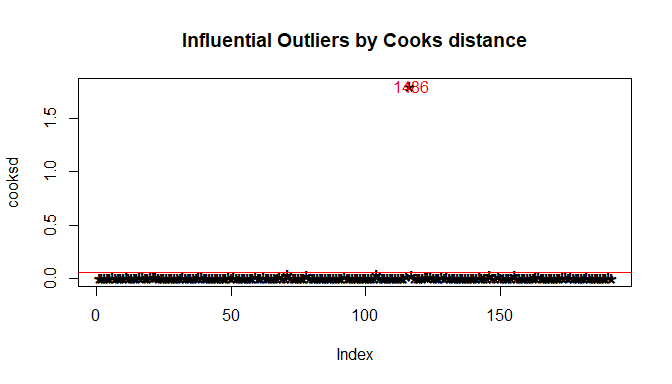


df <- filter(df, ph\_H < 15000 | ph\_BB < 1500 | ph\_SO < 3000 | fd\_E < 1500)

#### Cooks Distance

We will also remove influencial outliers using Cooks distance.

mod <- lm(WINS ~ ., data=df)  
cooksd <- cooks.distance(mod)  
plot(cooksd, pch="\*", cex=2, main="Influential Outliers by Cooks distance")  
abline(h = 4\*mean(cooksd, na.rm=T), col="red") # add cutoff line  
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4\*mean(cooksd, na.rm=T),names(cooksd),""), col="red") # add labels



We remove the influencial outliers.

influential <- as.numeric(names(cooksd)[(cooksd > 4\*mean(cooksd, na.rm=T))])  
df <- df[-influential, ]

### Fill Missing Values

The following features have missing values.

* bt\_SO - Strikeouts by batters
* br\_SB - Stolen bases
* br\_CS - Caught stealing
* bt\_HBP - Batters hit by pitch (get a free base)
* ph\_SO - Strikeouts by pitchers
* fd\_DP - Double Plays

Since most values in bt\_HBP are missing (90%), we will drop this feature.

#### Multivariate Imputation by Chained Equations (mice)

We will use Multivariable Imputation by Chained Equations (mice) to fill the missing variables.

### Address Correlated Features

While exploring the data, we noticed several features had strong positive linear relationships.

Let’s run a Variance Inflation Factor test to detect multicollinearity. Features with a VIF score > 10 will be reviewed.

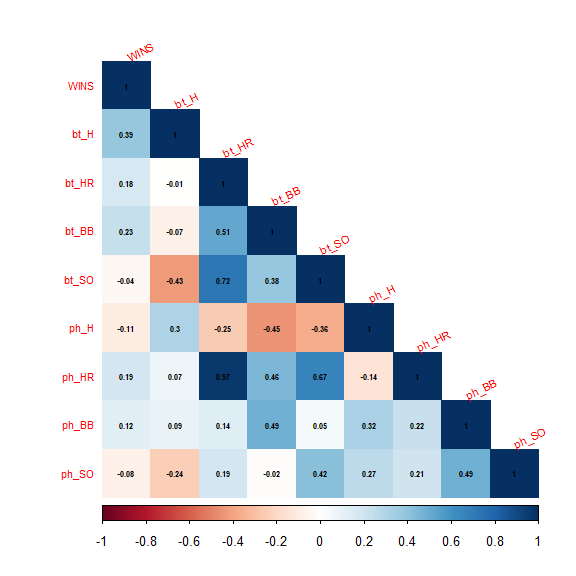
model1 <- lm(WINS ~., data = df)  
car::vif(model1)

## bt\_H bt\_2B bt\_3B bt\_HR bt\_BB bt\_SO br\_SB   
## 3.820596 2.467157 2.989892 36.501400 6.787771 5.279911 3.862460   
## br\_CS ph\_H ph\_HR ph\_BB ph\_SO fd\_E fd\_DP   
## 3.793169 4.073762 29.596294 6.468847 3.369127 4.988328 1.902235

Let’s make another correlation plot with only these features.

* bt\_SO (strikeouts by batters) and bt\_H (base hits by batters) have a strong positive correlation
* bt\_H (base hits by batters) and bt\_BB (walks by batters) have a strong positive correlation
* ph\_BB (walks allowed) and bt\_BB (walks by batters) have a strong negative correlation
* ph\_SO (strikeouts by pitchers) and bt\_SO (strikeouts by batters) have a moderate negative correlation
* ph\_HR (homeruns allowed) and bt\_HR (homeruns by batters) have a strong negative correlation
* ph\_SO (strikeouts by pitchers) and ph\_BB (walks allowed) have a moderate negative correation

corrplot(cor(subset(df, select = c(WINS, bt\_H, bt\_HR, bt\_BB, bt\_SO, ph\_H, ph\_HR, ph\_BB, ph\_SO)), use = "complete.obs"), method ="color", type="lower", addrect = 1, number.cex = 0.5, sig.level = 0.30,  
 addCoef.col = "black", # Add coefficient of correlation  
 tl.srt = 25, # Text label color and rotation  
 tl.cex = 0.7,  
 diag = TRUE)



To fix this, we can remove some correlated features and combine others.

* Remove bt\_HR. It has an extremely strong correlation with ph\_HR.
* Remove bt\_SO. It has an extremely strong correlation with ph\_SO.
* Replace bt\_H (total base hits by batters) with BT\_1B = bt\_H - BT\_2B - BT\_3B - BT\_HR (1B base hits)
* Replace ph\_BB and bt\_BB as a ratio of walks by batters to walks allowed

df$bt\_1B <- df$bt\_H - df$bt\_2B - df$bt\_3B - df$bt\_HR  
df$BB <- df$bt\_BB / df$ph\_BB  
df2 <- subset(df, select = -c(bt\_HR, bt\_SO, bt\_H, bt\_BB, ph\_BB))

These adjustments result in less multicollinearity.

model1 <- lm(WINS ~., data = df2)  
car::vif(model1)

## bt\_2B bt\_3B br\_SB br\_CS ph\_H ph\_HR ph\_SO fd\_E   
## 1.553145 2.338689 3.650821 3.686438 3.628940 2.311793 1.832450 6.805560   
## fd\_DP bt\_1B BB   
## 1.865776 2.664315 5.725045

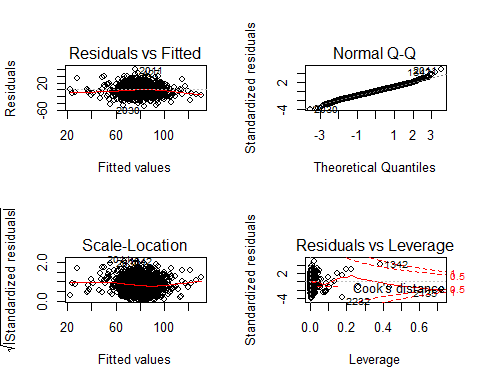
### Create Output

write.csv(df, "baseball\_output.csv")

### Linear Model 1.

We will begin with all independent variables and use the back elimination method to eliminate the non-significant ones.

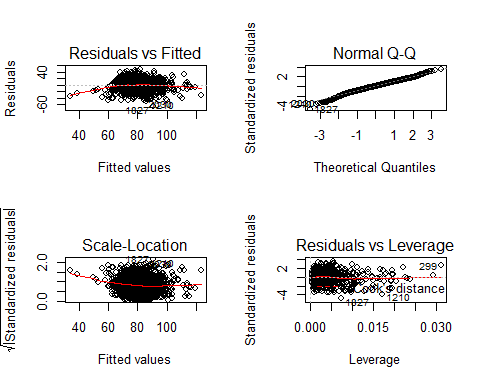
be\_lm1 <- lm(WINS ~., data = df)  
sum\_lm1<- summary(be\_lm1)  
par(mfrow=c(2,2))  
plot(be\_lm1)



We will start by eliminating the variables with high p-values and lowest significance from the model

Let’s take a look at the resulting model:

be\_lm2 <- lm(WINS ~ bt\_H + bt\_BB + br\_SB + br\_CS, data =df)  
sum\_lm2<-summary(be\_lm2)  
par(mfrow=c(2,2))  
plot(be\_lm2)



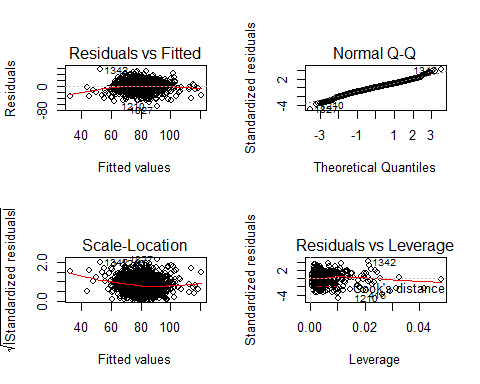
###Linear Model 2.

This Linear Model will be built using the variables we believe would have the highest corelation with WINs.

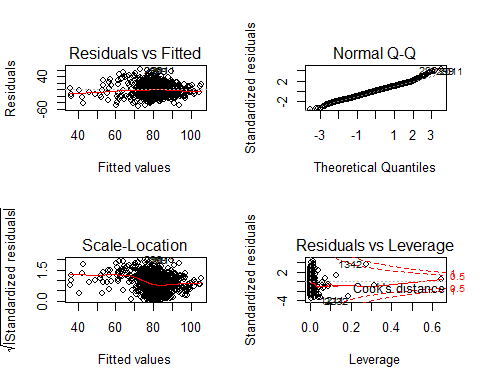
THe following variables will be used: - Base Hits by batters (1B,2B,3B,HR) - Walks by batters - Stolen bases - Strikeouts by batters

Let’s remove the two variables with low significance:

be\_lm3 <- lm(WINS ~ bt\_H + bt\_2B + bt\_3B + bt\_HR + bt\_BB + bt\_SO, data =df)  
sum\_lm3<-summary(be\_lm3)  
par(mfrow=c(2,2))  
plot(be\_lm3)



be\_lm4 <- lm(WINS ~   
 I(bt\_H + bt\_BB   
 - ph\_H - ph\_BB) +   
 I(bt\_HR - ph\_HR) +   
 I(bt\_SO - ph\_SO) +   
 I(br\_SB - br\_CS) +  
 fd\_E + fd\_DP , df)  
sum\_lm4<-summary(be\_lm4)  
par(mfrow=c(2,2))  
plot(be\_lm4)



# list of models and model summaries  
models <- list(be\_lm1, be\_lm2,be\_lm3,be\_lm4)  
modsums <- list(sum\_lm1, sum\_lm2, sum\_lm3, sum\_lm4)  
nmod <- length(modsums)  
  
# storage variables  
nvar <- integer(nmod)  
sigma <- numeric(nmod)  
rsq <- numeric(nmod)  
adj\_rsq <- numeric(nmod)  
fstat <- numeric(nmod)  
fstat\_p <- numeric(nmod)  
mse <- numeric(nmod)  
rmse <- numeric(nmod)  
  
# loop through model summaries  
for (j in 1:nmod) {  
 nvar[j] <- modsums[[j]]$df[1]  
 sigma[j] <- modsums[[j]]$sigma  
 rsq[j] <- modsums[[j]]$r.squared  
 adj\_rsq[j] <- modsums[[j]]$adj.r.squared  
 fstat[j] <- modsums[[j]]$fstatistic[1]  
 fstat\_p[j] <- 1 - pf(modsums[[j]]$fstatistic[1], modsums[[j]]$fstatistic[2],   
 modsums[[j]]$fstatistic[3])  
 mse[j] <- mean(modsums[[j]]$residuals^2)  
 rmse[j] <- sqrt(mse[j])  
}  
  
modnames <- paste0("lm", c(1:nmod))  
  
# evaluation dataframe  
eval <- data.frame(Model = modnames,   
 N\_Vars = nvar,  
 Sigma = sigma,  
 R\_Sq = rsq,  
 Adj\_R\_Sq = adj\_rsq,  
 F\_Stat = fstat,  
 F\_P\_Val = fstat\_p,  
 MSE = mse,  
 RMSE = rmse)  
  
kable(eval, digits = 3, align = 'c')

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | N\_Vars | Sigma | R\_Sq | Adj\_R\_Sq | F\_Stat | F\_P\_Val | MSE | RMSE |
| lm1 | 16 | 12.540 | 0.363 | 0.359 | 85.935 | 0 | 156.134 | 12.495 |
| lm2 | 5 | 13.714 | 0.243 | 0.242 | 182.565 | 0 | 187.659 | 13.699 |
| lm3 | 7 | 13.785 | 0.236 | 0.234 | 116.894 | 0 | 189.444 | 13.764 |
| lm4 | 7 | 14.774 | 0.123 | 0.120 | 52.855 | 0 | 217.602 | 14.751 |