Old_net_rating

Chenjie Li

November 23, 2018

```r
# load data
data <- read.csv('/home/chenjie/Desktop/Math564Project/Net_Rating/old_net_rating.csv')
data$color = "green"
data$color[data$win_ratio>=0.5]="blue"
data$color[data$win_ratio>=0.7317073]="red" #won more than 60 games
```

2014

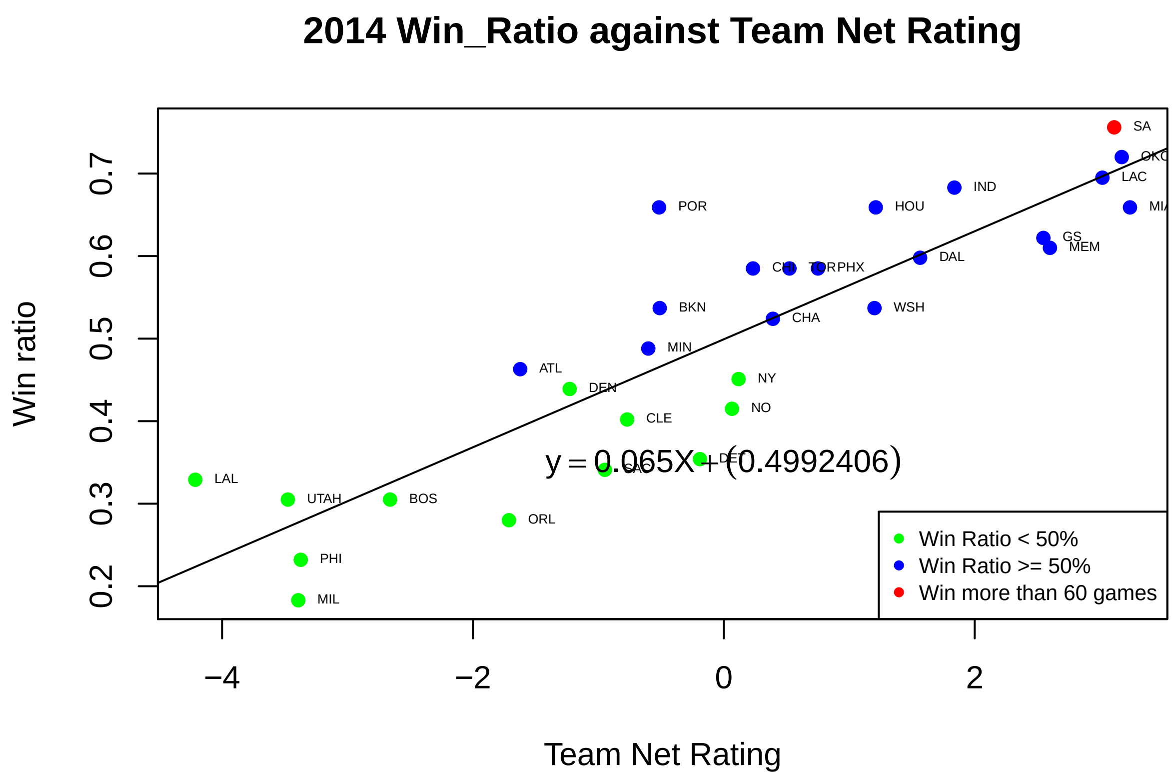
```r
s14 <- data[data$season == 2014,]
mod14 <- lm(win_ratio ~ old_net_rating, data = s14)
summary(mod14)
```

```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = s14)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132778 -0.050782 -0.001273  0.052731  0.193525
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.499241   0.013635   36.62  < 2e-16 ***
## old_net_rating 0.065432   0.006508   10.05 8.54e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07468 on 28 degrees of freedom
## Multiple R-squared:  0.7831, Adjusted R-squared:  0.7754
## F-statistic: 101.1 on 1 and 28 DF,  p-value: 8.535e-11
```

```r
plot(s14$old_net_rating,s14$win_ratio,xlab = 'Team Net Rating', ylab = 'Win ratio', main = '2014 Win_Ra
```

```
## integer(0)
```

```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



2015

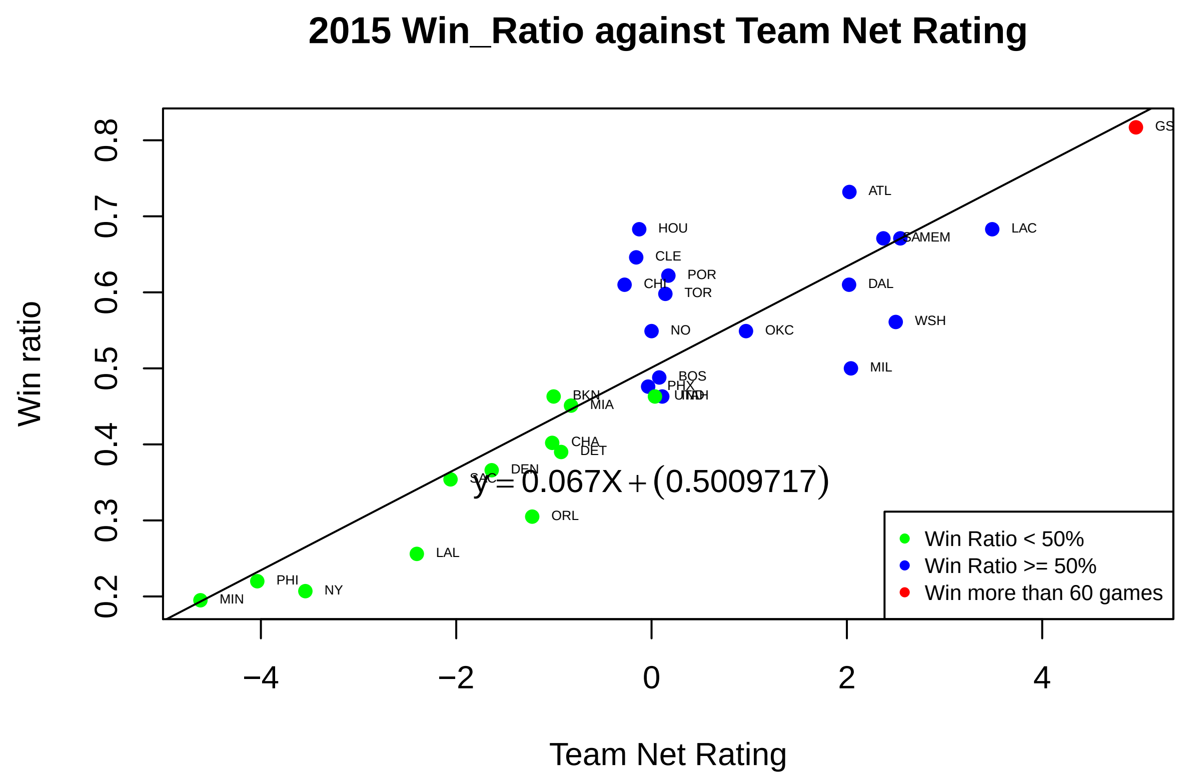
```r
s15 <- data[data$season == 2015,]
mod15 <- lm(win_ratio ~ old_net_rating, data = s15)
summary(mod15)
```

```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = s15)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13697 -0.04404 -0.01539  0.02459  0.19045
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.500972   0.014442  34.689  < 2e-16 ***
## old_net_rating 0.066616   0.006768   9.843 1.36e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0791 on 28 degrees of freedom
## Multiple R-squared:  0.7758, Adjusted R-squared:  0.7678
## F-statistic: 96.88 on 1 and 28 DF,  p-value: 1.364e-10
```

```r
plot(s15$old_net_rating,s15$win_ratio,xlab = 'Team Net Rating', ylab = 'Win ratio', main = '2015 Win_Ra
```

```
## integer(0)
```

```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



2016

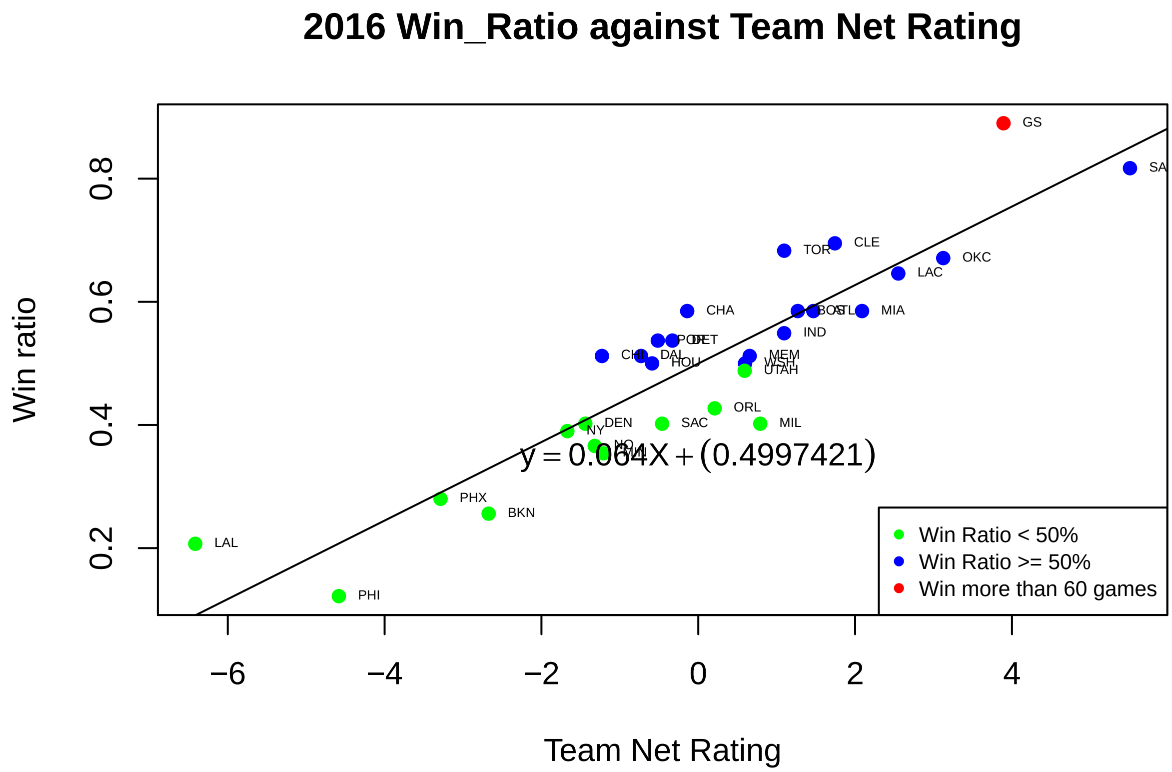
```r
s16 <- data[data$season == 2016,]
mod16 <- lm(win_ratio ~ old_net_rating, data = s16)
summary(mod16)
```

```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = s16)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14813 -0.04903 -0.01327  0.05864  0.14214
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.49974    0.01326   37.69  < 2e-16 ***
## old_net_rating    0.06375    0.00560   11.38  5.1e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07263 on 28 degrees of freedom
## Multiple R-squared:  0.8224, Adjusted R-squared:  0.816
## F-statistic: 129.6 on 1 and 28 DF,  p-value: 5.102e-12
```

```r
plot(s16$old_net_rating,s16$win_ratio,xlab = 'Team Net Rating', ylab = 'Win ratio', main = '2016 Win_Ra
```

```
## integer(0)
```

```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



2017

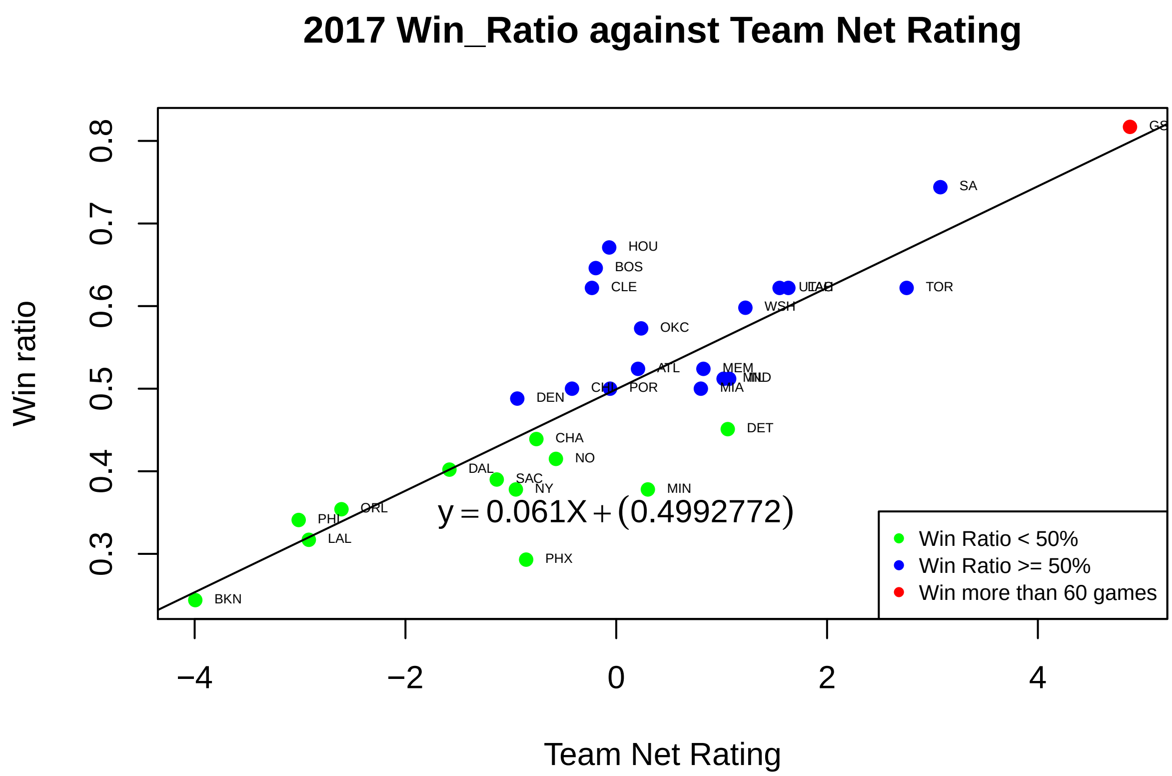
```r
s17 <- data[data$season == 2017,]
mod17 <- lm(win_ratio ~ old_net_rating, data = s17)
summary(mod17)
```

```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = s17)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.153747 -0.048088  0.002158  0.026786  0.175823
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.499277   0.013899  35.922  < 2e-16 ***
## old_net_rating 0.061449   0.007612   8.073 8.64e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07613 on 28 degrees of freedom
## Multiple R-squared:  0.6995, Adjusted R-squared:  0.6887
## F-statistic: 65.17 on 1 and 28 DF,  p-value: 8.644e-09
```

```r
plot(s17$old_net_rating,s17$win_ratio,xlab = 'Team Net Rating', ylab = 'Win ratio', main = '2017 Win_Ra
```

```
## integer(0)
```

```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



4 years as a whole

```r
four_years_total <- data
mod_total <- lm(win_ratio ~ old_net_rating, data = four_years_total)
summary(mod_total)
```

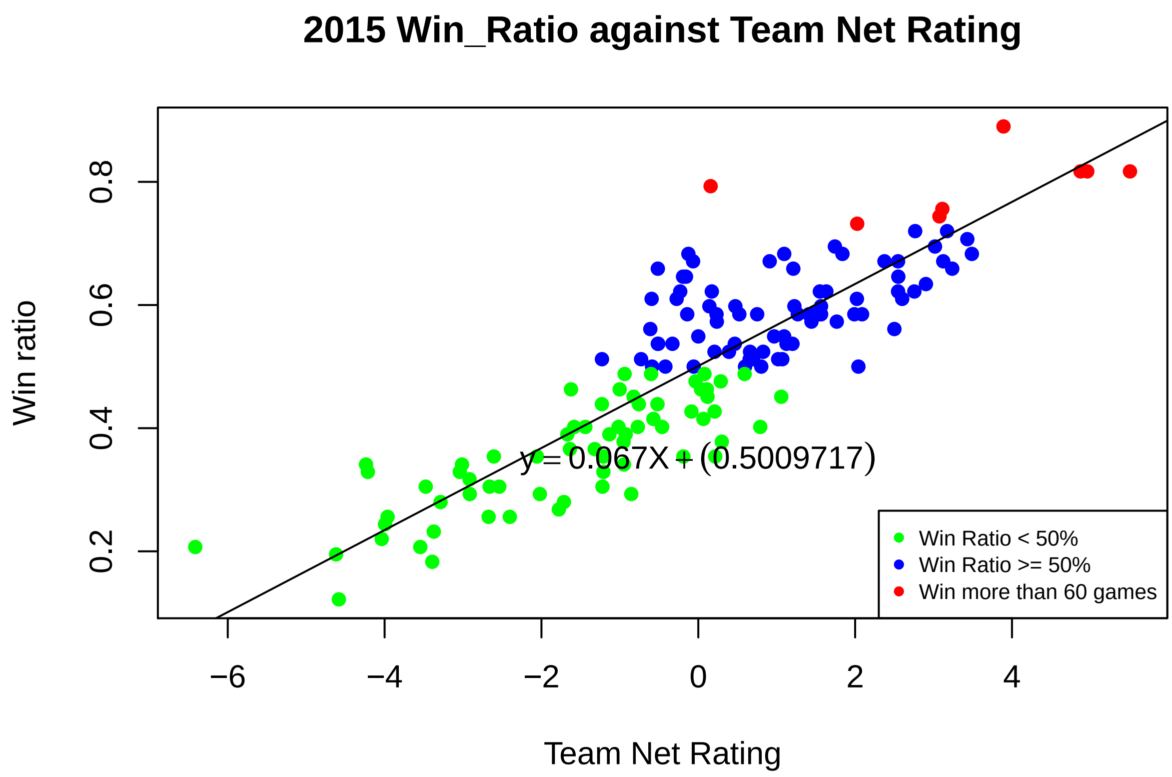
```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = four_years_total)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15948 -0.04813 -0.01204  0.04226  0.28326
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.499708   0.006225   80.27   <2e-16 ***
## old_net_rating 0.063858   0.002987   21.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.07624 on 148 degrees of freedom
## Multiple R-squared:  0.7554, Adjusted R-squared:  0.7538
## F-statistic: 457.1 on 1 and 148 DF,  p-value: < 2.2e-16
```

```
plot(four_years_total$old_net_rating,four_years_total$win_ratio,xlab = 'Team Net Rating', ylab = 'Win ra
```

```
## integer(0)
```

```
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



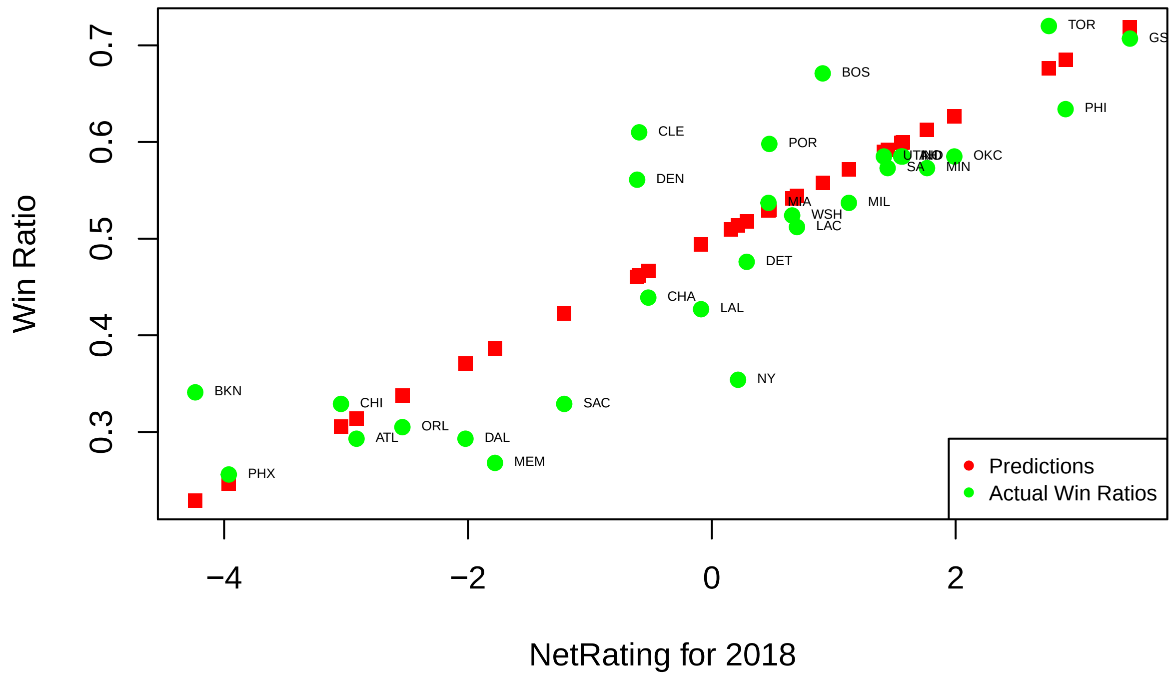
predict 2018 and compare with the actual results

```
x<-subset(data, season==2018, select=c(team,win_ratio,old_net_rating))
s18 <- data[data$season == 2018,]
pred <- predict(mod_total,s18,interval = "confidence")
data1 <-cbind(x,pred)
```

```
plot(data1$old_net_rating,data1$fit,pch=15,col="red",xlab = "NetRating for 2018",ylab = "Win Ratio")+poi
```

```
## integer(0)
```

```
legend("bottomright",legend=c("Predictions","Actual Win Ratios"),
       col=c("red","green"), pch = c(16,16,16), cex = 0.7)
```




```
SSE <-sum((data1$fit-data1$win_ratio)^2)
SSE
```

```
## [1] 0.2169388
```

```
SSTO <- sum((data1$fit - mean(data1$win_ratio))^2)
SSTO
```

```
## [1] 0.4689995
```

```
R_square <- 1 - SSE/SSTO
R_square
```

```
## [1] 0.5374434
```

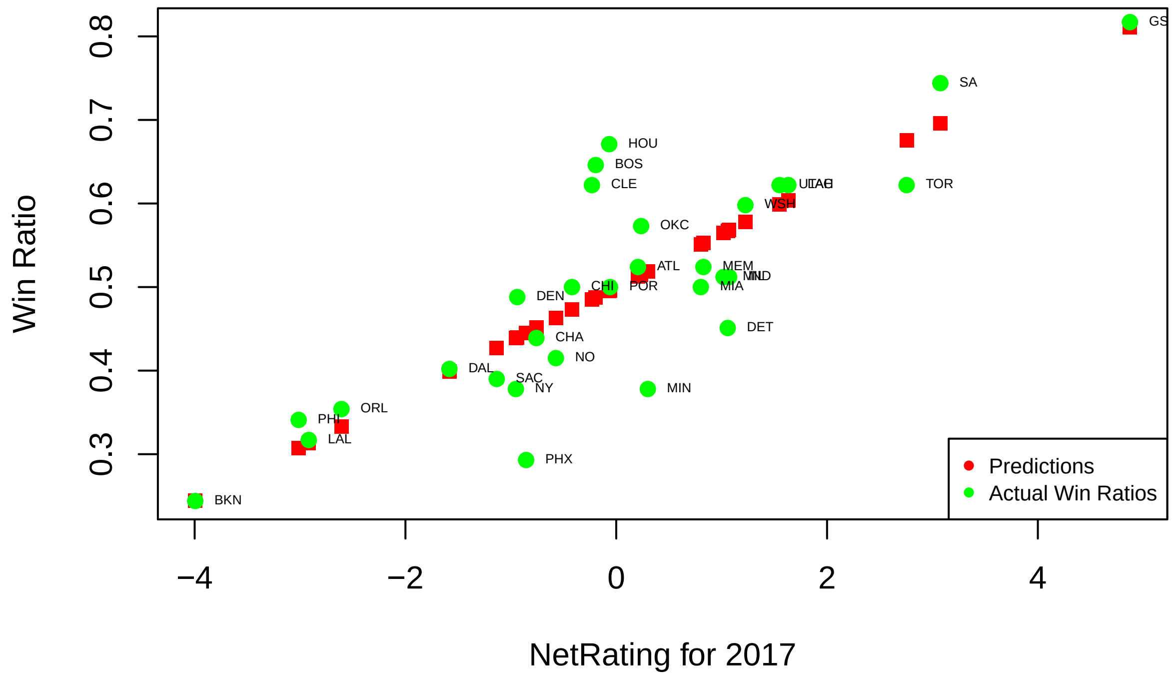
predict 2017 and compare with the actual results

```
x<-subset(data, season==2017, select=c(team,win_ratio,old_net_rating))
s18 <- data[data$season == 2017,]
pred <- predict(mod_total,s18,interval = "confidence")
data1 <-cbind(x,pred)
```

```
plot(data1$old_net_rating,data1$fit,pch=15,col="red",xlab = "NetRating for 2017",ylab = "Win Ratio")+po
```

```
## integer(0)
```

```
legend("bottomright",legend=c("Predictions","Actual Win Ratios"),
       col=c("red","green"), pch = c(16,16,16), cex = 0.7)
```




```
SSE <-sum((data1$fit-data1$win_ratio)^2)
SSE
```

```
## [1] 0.1628556
```

```
SSTO <- sum((data1$fit - mean(data1$win_ratio))^2)
SSTO
```

```
## [1] 0.4078517
```

```
R_square <- 1 - SSE/SSTO
R_square
```

```
## [1] 0.6006991
```

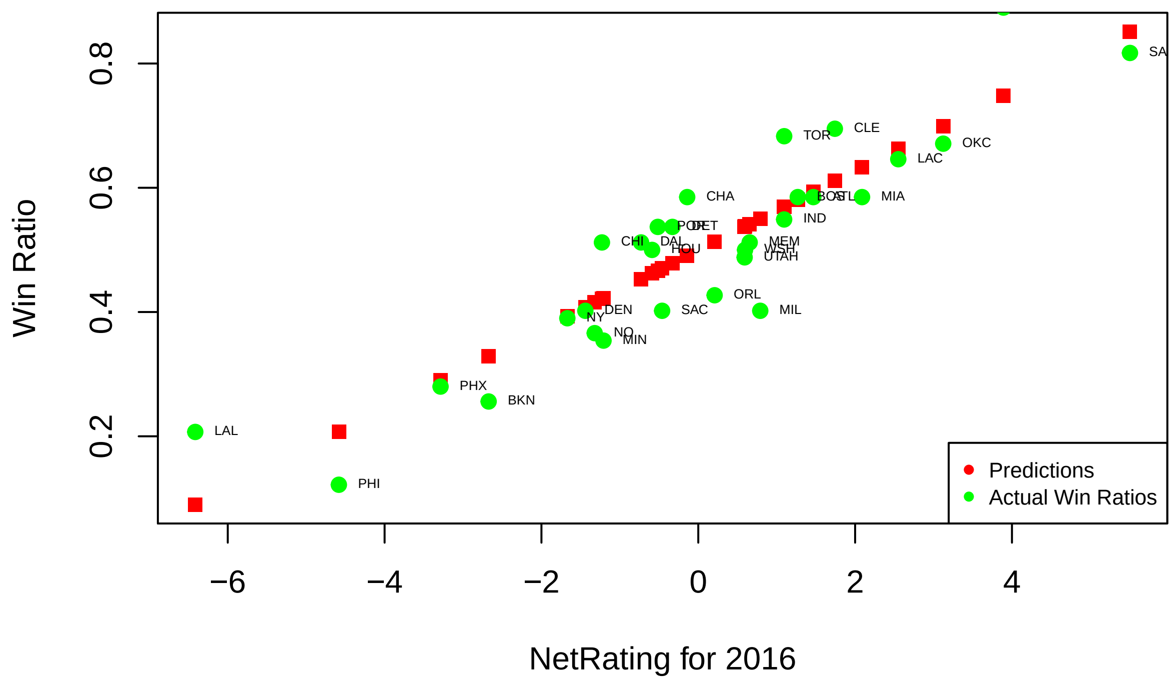
predict 2016 and compare with the actual results

```
x<-subset(data, season==2016, select=c(team,win_ratio,old_net_rating))
s18 <- data[data$season == 2016,]
pred <- predict(mod_total,s18,interval = "confidence")
data1 <-cbind(x,pred)
```

```
plot(data1$old_net_rating,data1$fit,pch=15,col="red",xlab = "NetRating for 2016",ylab = "Win Ratio")+po
```

```
## integer(0)
```

```
legend("bottomright",legend=c("Predictions","Actual Win Ratios"),
       col=c("red","green"), pch = c(16,16,16), cex = 0.7)
```




```r
SSE <-sum((data1$fit-data1$win_ratio)^2)
SSE
```

```
## [1] 0.1477097
```

```r
SSTO <- sum((data1$fit - mean(data1$win_ratio))^2)
SSTO
```

```
## [1] 0.686024
```

```r
R_square <- 1 - SSE/SSTO
R_square
```

```
## [1] 0.7846872
```

```r
mod14$coef
```

```
##   (Intercept) old_net_rating
##    0.49924058     0.06543151
```

```r
mod15$coef
```

```
##   (Intercept) old_net_rating
##    0.50097166     0.06661641
```

```r
mod16$coef
```

```
##   (Intercept) old_net_rating
##    0.49974210     0.06375233
```

```r
mod17$coef
```

```
##   (Intercept) old_net_rating
##    0.49927721     0.06144943
```

```r
summary(mod14)$r.squared
```

```
## [1] 0.7831021
```

```r
summary(mod15)$r.squared
```

```
## [1] 0.7757837
```

```r
summary(mod16)$r.squared
```

```
## [1] 0.8223506
```

```r
summary(mod17)$r.squared
```

```
## [1] 0.6994603
```

```r
sample_data <-subset(data,data$season !=2018)
sample_data
```

```
##     season       team win_ratio old_net_rating color
## 31    2017 GS           0.817     4.87416449   red
## 32    2017 SA           0.744     3.07509655   red
## 33    2017 HOU          0.671    -0.06672436  blue
## 34    2017 BOS          0.646    -0.19448857  blue
## 35    2017 UTAH         0.622     1.54979664  blue
## 36    2017 TOR          0.622     2.75499001  blue
## 37    2017 CLE          0.622    -0.23065336  blue
## 38    2017 LAC          0.622     1.63300677  blue
## 39    2017 WSH          0.598     1.22518933  blue
## 40    2017 OKC          0.573     0.23601872  blue
## 41    2017 MEM          0.524     0.82696758  blue
## 42    2017 ATL          0.524     0.20650965  blue
## 43    2017 IND          0.512     1.07110546  blue
## 44    2017 MIL          0.512     1.01843413  blue
## 45    2017 CHI          0.500    -0.41985526  blue
## 46    2017 POR          0.500    -0.05881372  blue
## 47    2017 MIA          0.500     0.80255128  blue
## 48    2017 DEN          0.488    -0.93920381 green
## 49    2017 DET          0.451     1.05792423 green
## 50    2017 CHA          0.439    -0.75788388 green
## 51    2017 NO           0.415    -0.57241955 green
## 52    2017 DAL          0.402    -1.58271420 green
## 53    2017 SAC          0.390    -1.13388477 green
## 54    2017 MIN          0.378     0.29979316 green
## 55    2017 NY           0.378    -0.95230019 green
## 56    2017 ORL          0.354    -2.60709424 green
## 57    2017 PHI          0.341    -3.01306809 green
## 58    2017 LAL          0.317    -2.91676392 green
## 59    2017 PHX          0.293    -0.85485722 green
## 60    2017 BKN          0.244    -3.99422360 green
## 61    2016 GS           0.890     3.89189189   red
## 62    2016 SA           0.817     5.50484988   red
## 63    2016 CLE          0.695     1.74175978  blue
## 64    2016 TOR          0.683     1.09542942  blue
## 65    2016 OKC          0.671     3.12321305  blue
## 66    2016 LAC          0.646     2.55022375  blue
## 67    2016 MIA          0.585     2.08823373  blue
## 68    2016 BOS          0.585     1.26871121  blue
## 69    2016 CHA          0.585    -0.14134904  blue
## 70    2016 ATL          0.585     1.46694147  blue
```

```
## 71    2016 IND            0.549     1.09477852  blue
## 72    2016 DET            0.537    -0.32984930  blue
## 73    2016 POR            0.537    -0.51635422  blue
## 74    2016 DAL            0.512    -0.72947645  blue
## 75    2016 CHI            0.512    -1.22855479  blue
## 76    2016 MEM            0.512     0.65544886  blue
## 77    2016 HOU            0.500    -0.58925354  blue
## 78    2016 WSH            0.500     0.59617172  blue
## 79    2016 UTAH           0.488     0.59101655 green
## 80    2016 ORL            0.427     0.20840349 green
## 81    2016 MIL            0.402     0.79033002 green
## 82    2016 DEN            0.402    -1.44013964 green
## 83    2016 SAC            0.402    -0.46113307 green
## 84    2016 NY             0.390    -1.66973358 green
## 85    2016 NO             0.366    -1.32086084 green
## 86    2016 MIN            0.354    -1.20855783 green
## 87    2016 PHX            0.280    -3.28673416 green
## 88    2016 BKN            0.256    -2.67339397 green
## 89    2016 LAL            0.207    -6.41453162 green
## 90    2016 PHI            0.122    -4.58317644 green
## 91    2015 GS             0.817     4.96003497   red
## 92    2015 ATL            0.732     2.02599388   red
## 93    2015 HOU            0.683    -0.12638075  blue
## 94    2015 LAC            0.683     3.48837209  blue
## 95    2015 MEM            0.671     2.54734723  blue
## 96    2015 SA             0.671     2.37393870  blue
## 97    2015 CLE            0.646    -0.15712652  blue
## 98    2015 POR            0.622     0.17224010  blue
## 99    2015 CHI            0.610    -0.27619848  blue
## 100   2015 DAL            0.610     2.02250929  blue
## 101   2015 TOR            0.598     0.14143887  blue
## 102   2015 WSH            0.561     2.50027819  blue
## 103   2015 NO             0.549     0.00000000  blue
## 104   2015 OKC            0.549     0.96707209  blue
## 105   2015 MIL            0.500     2.04155844  blue
## 106   2015 BOS            0.488     0.07883948 green
## 107   2015 PHX            0.476    -0.03495630 green
## 108   2015 IND            0.463     0.10982385 green
## 109   2015 UTAH           0.463     0.03483941 green
## 110   2015 BKN            0.463    -1.00263712 green
## 111   2015 MIA            0.451    -0.82457556 green
## 112   2015 CHA            0.402    -1.01730019 green
## 113   2015 DET            0.390    -0.92592593 green
## 114   2015 DEN            0.366    -1.63700140 green
## 115   2015 SAC            0.354    -2.05840310 green
## 116   2015 ORL            0.305    -1.22207441 green
## 117   2015 LAL            0.256    -2.40332875 green
## 118   2015 PHI            0.220    -4.03691896 green
## 119   2015 NY             0.207    -3.54482672 green
## 120   2015 MIN            0.195    -4.61919979 green
## 121   2014 SA             0.756     3.11206336   red
## 122   2014 OKC            0.720     3.17225923  blue
## 123   2014 LAC            0.695     3.01804424  blue
## 124   2014 IND            0.683     1.83805646  blue
```

```
## 125   2014 MIA           0.659     3.23804940  blue
## 126   2014 POR           0.659    -0.51604636  blue
## 127   2014 HOU           0.659     1.21124850  blue
## 128   2014 GS            0.622     2.54751213  blue
## 129   2014 MEM           0.610     2.59986118  blue
## 130   2014 DAL           0.598     1.56496368  blue
## 131   2014 TOR           0.585     0.52319269  blue
## 132   2014 CHI           0.585     0.23261607  blue
## 133   2014 PHX           0.585     0.75031056  blue
## 134   2014 WSH           0.537     1.20113146  blue
## 135   2014 BKN           0.537    -0.51093643  blue
## 136   2014 CHA           0.524     0.39108616  blue
## 137   2014 MIN           0.488    -0.60222483 green
## 138   2014 ATL           0.463    -1.62328868 green
## 139   2014 NY            0.451     0.11762881 green
## 140   2014 DEN           0.439    -1.22905028 green
## 141   2014 NO            0.415     0.06528047 green
## 142   2014 CLE           0.402    -0.77094836 green
## 143   2014 DET           0.354    -0.19046914 green
## 144   2014 SAC           0.341    -0.94846742 green
## 145   2014 LAL           0.329    -4.21379420 green
## 146   2014 UTAH          0.305    -3.47543976 green
## 147   2014 BOS           0.305    -2.66073715 green
## 148   2014 ORL           0.280    -1.71313837 green
## 149   2014 PHI           0.232    -3.37292503 green
## 150   2014 MIL           0.183    -3.39236717 green
```

```r
# training and testing data using "new_net_rating"

set.seed(1)  # setting seed to reproduce results of random sampling

trainingRowIndex <- sample(1:nrow(sample_data), 0.80*nrow(sample_data))  # row incices for training dat

trainingData <- sample_data[trainingRowIndex, ]  # model training data

testData  <- sample_data[-trainingRowIndex, ]   # test data

train_new_toe<- lm(win_ratio ~ old_net_rating, data=trainingData)  # build the model

predict_new_toe <- predict(train_new_toe, testData)  # predict

summary(train_new_toe) # model summary
```

```
##
## Call:
## lm(formula = win_ratio ~ old_net_rating, data = trainingData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14856 -0.04781 -0.00703  0.04855  0.19052
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.500901   0.007466   67.09   <2e-16 ***
## old_net_rating 0.062829   0.003486   18.02   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07311 on 94 degrees of freedom
## Multiple R-squared:  0.7756, Adjusted R-squared:  0.7732
## F-statistic: 324.8 on 1 and 94 DF,  p-value: < 2.2e-16
```

```
# Calculate: akaike information criterion
AIC(train_new_toe)
```

```
## [1] -225.8254
```

```
actuals_preds_new <- data.frame(cbind(actuals=testData$win_ratio, predicteds=predict_new_toe))
# make actuals_predicteds dataframe.

correlation_accuracy_new <- cor(actuals_preds_new)
correlation_accuracy_new
```

```
##              actuals predicteds
## actuals    1.0000000  0.8879518
## predicteds 0.8879518  1.0000000
```

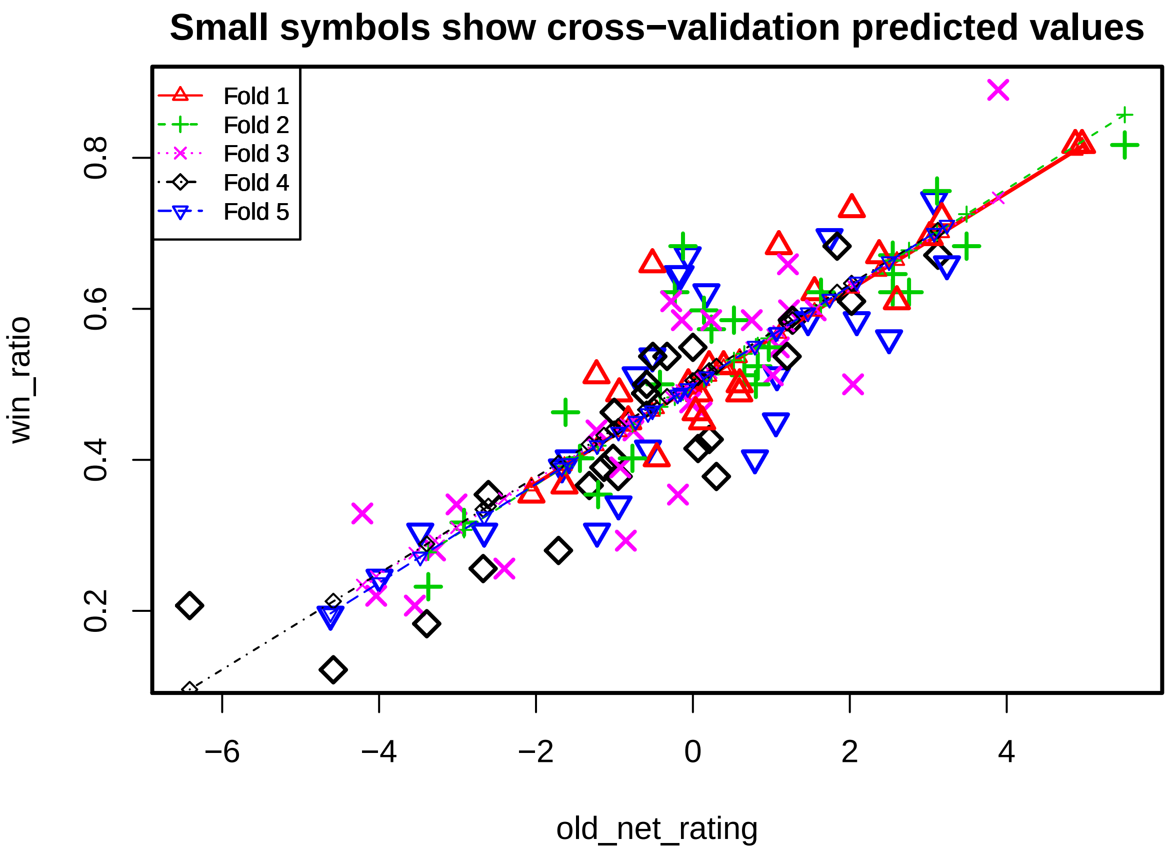
5 - Fold Cross Validation - new net rating

```
library(DAAG)
```

```
## Loading required package: lattice
```

```
sample_data <-subset(data,data$season !=2018)
cv.lm(sample_data, form.lm = formula(win_ratio ~ old_net_rating), m=5, dots = FALSE, seed=123, plotit=TI
```

```
## Analysis of Variance Table
##
## Response: win_ratio
##                 Df Sum Sq Mean Sq F value Pr(>F)
## old_net_rating   1  2.230   2.230     409 <2e-16 ***
## Residuals      118  0.643   0.005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```




```
##
## fold 1
## Observations in test set: 24
##                    31     35     42      46      48    64      75      78
## old_net_rating 4.87416 1.5498 0.2065 -0.0588 -0.9392 1.095 -1.2286  0.5962
## cvpred         0.80961 0.5964 0.5103  0.4933  0.4368 0.567  0.4183  0.5353
## win_ratio      0.81700 0.6220 0.5240  0.5000  0.4880 0.683  0.5120  0.5000
## CV residual    0.00739 0.0256 0.0137  0.0067  0.0512 0.116  0.0937 -0.0353
##                    79     83      91    92     96     106     109
## old_net_rating  0.591 -0.4611 4.96003 2.026 2.3739  0.0788  0.0348
## cvpred          0.535  0.4675 0.81512 0.627 0.6493  0.5021  0.4993
## win_ratio       0.488  0.4020 0.81700 0.732 0.6710  0.4880  0.4630
## CV residual    -0.047 -0.0655 0.00188 0.105 0.0217 -0.0141 -0.0363
##                     111     114     115    122     123    126     129
## old_net_rating -0.82458 -1.6370 -2.0584 3.1723 3.01804 -0.516  2.5999
## cvpred          0.44419  0.3921  0.3651 0.7005 0.69059  0.464  0.6638
## win_ratio       0.45100  0.3660  0.3540 0.7200 0.69500  0.659  0.6100
## CV residual     0.00681 -0.0261 -0.0111 0.0195 0.00441  0.195 -0.0538
##                   136     139
## old_net_rating 0.39109  0.1176
## cvpred         0.52214  0.5046
## win_ratio      0.52400  0.4510
## CV residual    0.00186 -0.0536
##
## Sum of squares = 0.09    Mean square = 0    n = 24
##
## fold 2
```

```
## Observations in test set: 24
##                     36     37     38     40      41      45      47
## old_net_rating  2.7550 -0.231 1.6330 0.2360  0.8270 -0.4199  0.8026
## cvpred          0.6775  0.483 0.6043 0.5131  0.5517  0.4703  0.5501
## win_ratio       0.6220  0.622 0.6220 0.5730  0.5240  0.5000  0.5000
## CV residual    -0.0555  0.139 0.0177 0.0599 -0.0277  0.0297 -0.0501
##                      58     62      66      76       82      86     93
## old_net_rating -2.91676  5.505  2.5502  0.6554 -1.44014 -1.2086 -0.126
## cvpred          0.30726  0.857  0.6642  0.5405  0.40366  0.4188  0.489
## win_ratio       0.31700  0.817  0.6460  0.5120  0.40200  0.3540  0.683
## CV residual     0.00974 -0.040 -0.0182 -0.0285 -0.00166 -0.0648  0.194
##                     94      95    101     104    121    128    131     138
## old_net_rating  3.4884 2.54735 0.1414  0.9671 3.1121  2.548 0.5232 -1.6233
## cvpred          0.7254 0.66396 0.5069  0.5608 0.7008  0.664 0.5318  0.3917
## win_ratio       0.6830 0.67100 0.5980  0.5490 0.7560  0.622 0.5850  0.4630
## CV residual    -0.0424 0.00704 0.0911 -0.0118 0.0552 -0.042 0.0532  0.0713
##                    142     149
## old_net_rating -0.7709 -3.3729
## cvpred          0.4473  0.2775
## win_ratio       0.4020  0.2320
## CV residual    -0.0453 -0.0455
##
## Sum of squares = 0.1    Mean square = 0    n = 24
##
## fold 3
## Observations in test set: 24
##                    39      44      50      57     59    61      69     71
## old_net_rating 1.2252  1.0184 -0.7579 -3.0131 -0.855 3.892 -0.1413  1.095
## cvpred         0.5783  0.5652  0.4528  0.3101  0.447 0.747  0.4918  0.570
## win_ratio      0.5980  0.5120  0.4390  0.3410  0.293 0.890  0.5850  0.549
## CV residual    0.0197 -0.0532 -0.0138  0.0309 -0.154 0.143  0.0932 -0.021
##                     87     99   105     107     108     113     117
## old_net_rating -3.2867 -0.276  2.04 -0.0350  0.1098 -0.9259 -2.4033
## cvpred          0.2928  0.483  0.63  0.4986  0.5077  0.4422  0.3487
## win_ratio       0.2800  0.610  0.50  0.4760  0.4630  0.3900  0.2560
## CV residual    -0.0128  0.127 -0.13 -0.0226 -0.0447 -0.0522 -0.0927
##                    118     119    127      130    132    133    140    143
## old_net_rating -4.0369 -3.5448 1.2112  1.56496 0.2326 0.7503 -1.229 -0.190
## cvpred          0.2454  0.2765 0.5774  0.59977 0.5155 0.5482  0.423  0.489
## win_ratio       0.2200  0.2070 0.6590  0.59800 0.5850 0.5850  0.439  0.354
## CV residual    -0.0254 -0.0695 0.0816 -0.00177 0.0695 0.0368  0.016 -0.135
##                    145
## old_net_rating -4.2138
## cvpred          0.2342
## win_ratio       0.3290
## CV residual     0.0948
##
## Sum of squares = 0.15    Mean square = 0.01    n = 24
##
## fold 4
## Observations in test set: 24
##                     53     54      55      56      65        68      72
## old_net_rating -1.1339  0.300 -0.9523 -2.6071  3.1232  1.268711 -0.3298
## cvpred          0.4325  0.524  0.4441  0.3386  0.7039  0.585647  0.4837
```

```
## win_ratio       0.3900  0.378  0.3780  0.3540  0.6710  0.585000  0.5370
## CV residual    -0.0425 -0.146 -0.0661  0.0154 -0.0329 -0.000647  0.0533
##                     77      80      85      88      89      90     100
## old_net_rating -0.5893  0.2084 -1.3209 -2.6734 -6.4145 -4.5832  2.0225
## cvpred          0.4672  0.5181  0.4206  0.3344  0.0959  0.2126  0.6337
## win_ratio       0.5000  0.4270  0.3660  0.2560  0.2070  0.1220  0.6100
## CV residual     0.0328 -0.0911 -0.0546 -0.0784  0.1111 -0.0906 -0.0237
##                    103     110     112    124     134     135     137
## old_net_rating 0.0000 -1.0026 -1.0173 1.8381  1.2011 -0.5109 -0.6022
## cvpred         0.5048  0.4409  0.4399 0.6219  0.5813  0.4722  0.4664
## win_ratio      0.5490  0.4630  0.4020 0.6830  0.5370  0.5370  0.4880
## CV residual    0.0442  0.0221 -0.0379 0.0611 -0.0443  0.0648  0.0216
##                    141    148    150
## old_net_rating  0.0653 -1.713 -3.392
## cvpred          0.5089  0.396  0.289
## win_ratio       0.4150  0.280  0.183
## CV residual    -0.0939 -0.116 -0.106
##
## Sum of squares = 0.12    Mean square = 0    n = 24
##
## fold 5
## Observations in test set: 24
##                    32      33     34      43     49      51      52
## old_net_rating 3.0751 -0.0667 -0.194  1.0711  1.058 -0.5724 -1.5827
## cvpred         0.7003  0.4946  0.486  0.5691  0.568  0.4615  0.3954
## win_ratio      0.7440  0.6710  0.646  0.5120  0.451  0.4150  0.4020
## CV residual    0.0437  0.1764  0.160 -0.0571 -0.117 -0.0465  0.0066
##                      60    63      67     70      73      74      81
## old_net_rating -3.99422 1.742  2.0882  1.467 -0.5164 -0.7295  0.790
## cvpred          0.23754 0.613  0.6357  0.595  0.4652  0.4513  0.551
## win_ratio       0.24400 0.695  0.5850  0.585  0.5370  0.5120  0.402
## CV residual     0.00646 0.082 -0.0507 -0.010  0.0718  0.0607 -0.149
##                       84     97    98    102    116      120    125
## old_net_rating -1.669734 -0.157 0.172  2.500 -1.222 -4.61920  3.238
## cvpred          0.389702  0.489 0.510  0.663  0.419  0.19663  0.711
## win_ratio       0.390000  0.646 0.622  0.561  0.305  0.19500  0.659
## CV residual     0.000298  0.157 0.112 -0.102 -0.114 -0.00163 -0.052
##                    144     146     147
## old_net_rating -0.9485 -3.4754 -2.6607
## cvpred          0.4369  0.2715  0.3248
## win_ratio       0.3410  0.3050  0.3050
## CV residual    -0.0959  0.0335 -0.0198
##
## Sum of squares = 0.19    Mean square = 0.01    n = 24
##
## Overall (Sum over all 24 folds)
##      ms
## 0.00546
```

predict 2018 and compare with the actual results

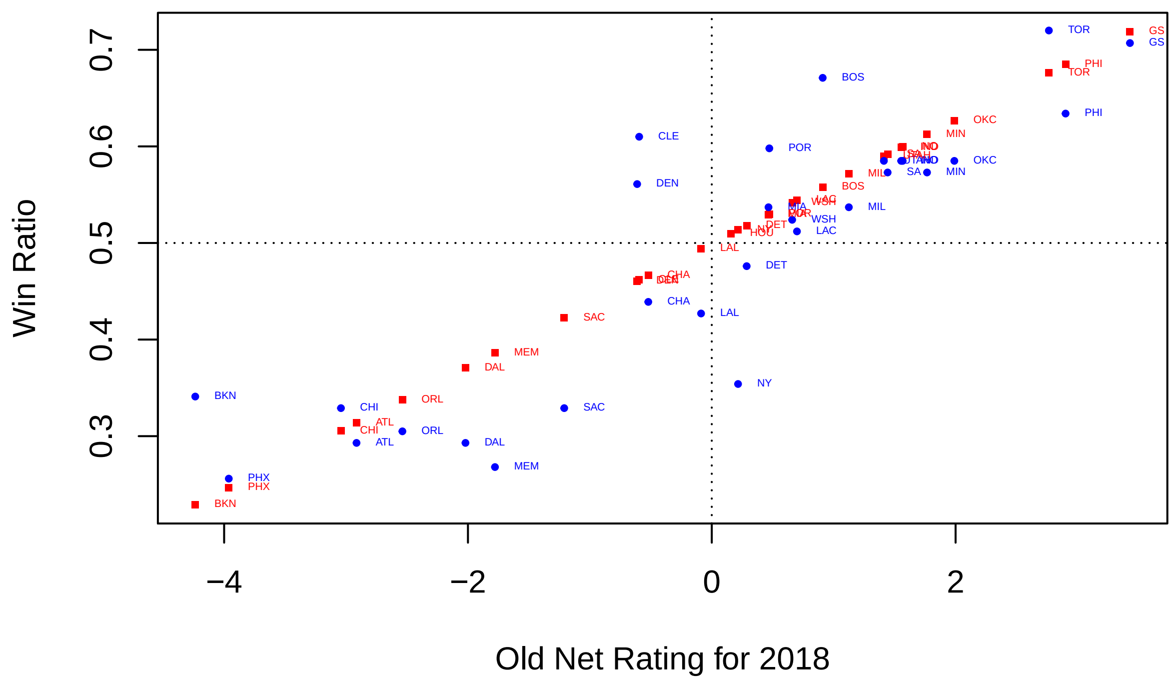
using new net rating

```r
x<-subset(data, season==2018, select=c(team,win_ratio,old_net_rating))
s18 <- data[data$season == 2018,]
pred <- predict(mod_total,s18,interval = "confidence")
data1 <-cbind(x,pred)

ranking <- subset(data1,select=c(team,fit))
ordered_data <- ranking[order(-ranking$fit),]
ordered_data
```

```
##          team   fit
## 3  GS         0.719
## 5  PHI        0.685
## 2  TOR        0.676
## 9  OKC        0.627
## 13 MIN        0.612
## 10 NO         0.600
## 8  IND        0.599
## 12 SA         0.592
## 11 UTAH       0.590
## 16 MIL        0.572
## 4  BOS        0.558
## 18 LAC        0.544
## 17 WSH        0.542
## 7  POR        0.530
## 15 MIA        0.529
## 19 DET        0.518
## 22 NY         0.513
## 1  HOU        0.510
## 21 LAL        0.494
## 20 CHA        0.466
## 6  CLE        0.462
## 14 DEN        0.461
## 24 SAC        0.422
## 29 MEM        0.386
## 27 DAL        0.371
## 26 ORL        0.338
## 28 ATL        0.314
## 25 CHI        0.305
## 30 PHX        0.247
## 23 BKN        0.229
```

```r
plot(data1$old_net_rating,data1$fit,pch=15,col="red",xlab = "Old Net Rating for 2018",ylab = "Win Ratio
```




```
## integer(0)
```

```
SSE <-sum((data1$fit-data1$win_ratio)^2)
SSE
```

```
## [1] 0.217
```

```
SSTO <- sum((data1$win_ratio - mean(data1$win_ratio))^2)
SSTO
```

```
## [1] 0.645
```

```
R_square <- 1 - SSE/SSTO
R_square
```

```
## [1] 0.663
```