TOE

Chenjie Li

November 16, 2018

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

2014 win ratio against Team TOE

Season 2014 our new TOE:

```
s14 <- team_stats[team_stats$season == 2014,]
new_mod14 <- lm(win_ratio ~ new_toe, data = s14)
summary(new_mod14)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = s14)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14514 -0.07483 -0.01923  0.04819  0.24369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7505     0.4995  -5.507 6.95e-06 ***
## new_toe       6.6228     1.0169   6.513 4.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1011 on 28 degrees of freedom
## Multiple R-squared:  0.6023, Adjusted R-squared:  0.5881
## F-statistic: 42.41 on 1 and 28 DF,  p-value: 4.661e-07
```

Season 2014 old TOE:

```
mod14 <- lm(win_ratio ~ toe, data = s14)
summary(mod14)
```

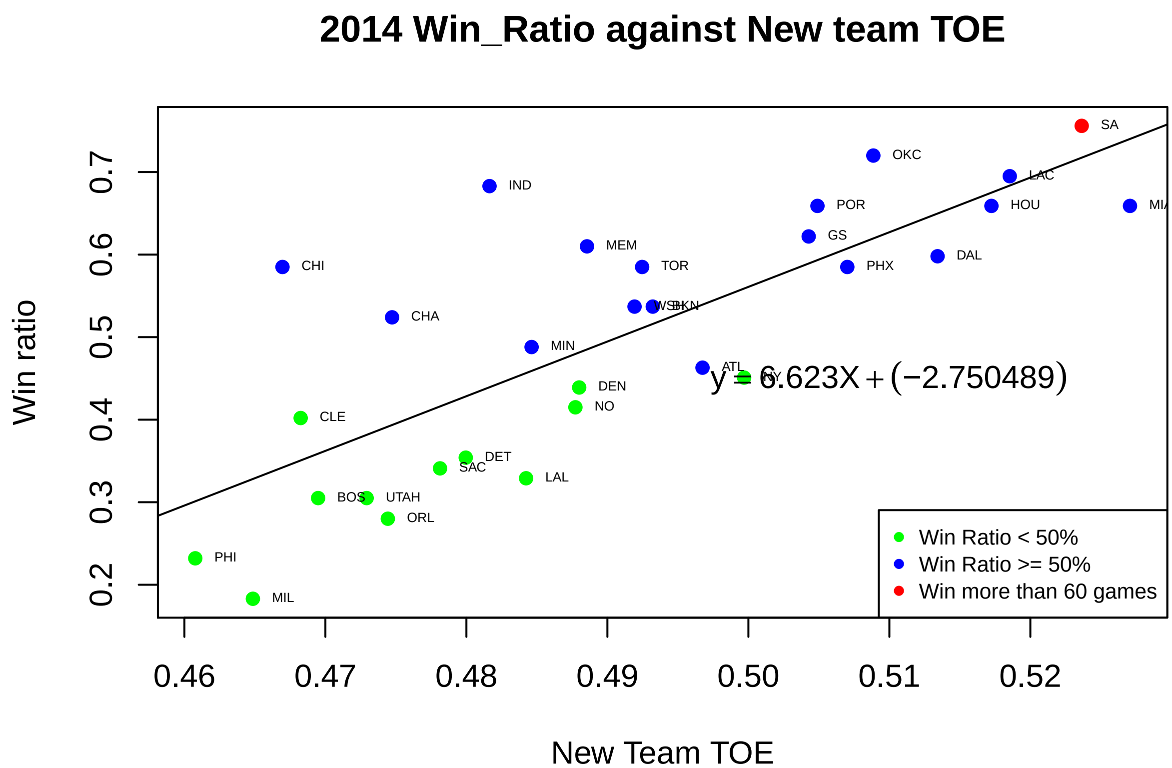
```
##
## Call:
## lm(formula = win_ratio ~ toe, data = s14)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.23709 -0.06599 -0.01423  0.08087  0.27310
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7875     0.7047  -3.956 0.000473 ***
## toe           7.5582     1.6193   4.667 6.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1203 on 28 degrees of freedom
## Multiple R-squared:  0.4376, Adjusted R-squared:  0.4175
## F-statistic: 21.79 on 1 and 28 DF,  p-value: 6.883e-05
```

```
plot(s14$new_toe,s14$win_ratio,xlab = 'New Team TOE', ylab = 'Win ratio', main = '2014 Win_Ratio agains
```

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

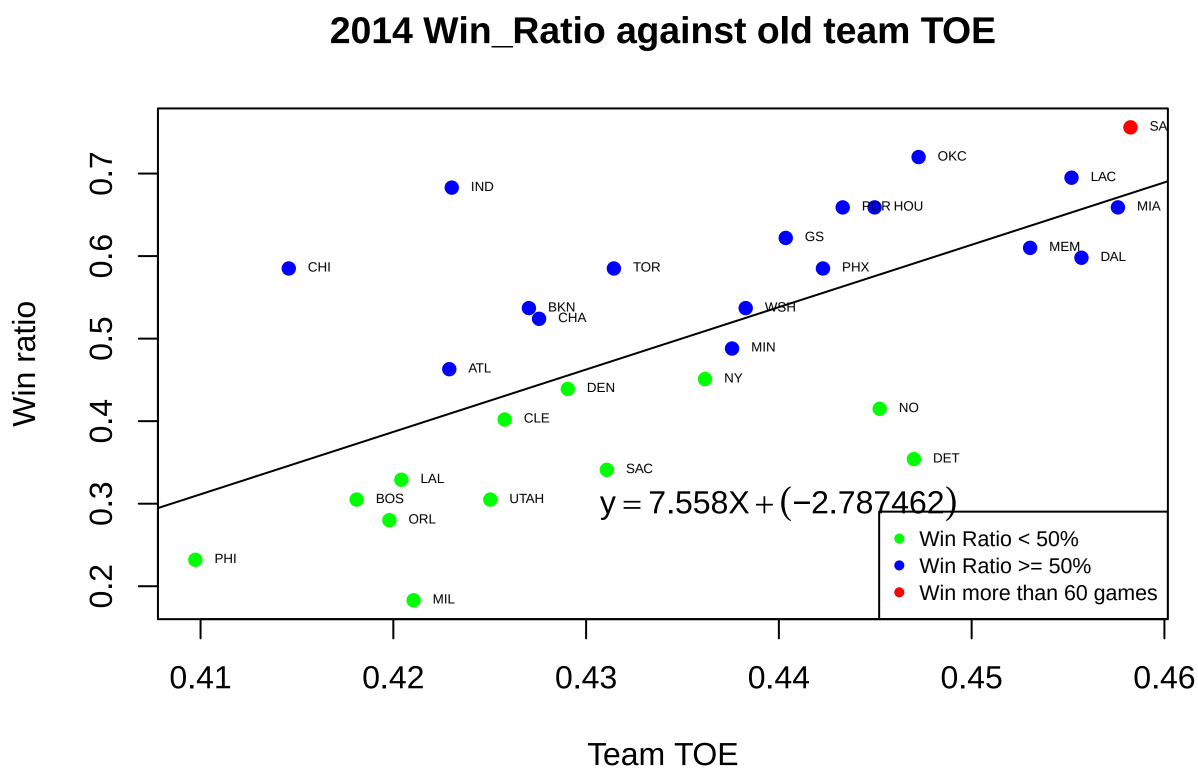
```
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```




```
plot(s14$toe,s14$win_ratio,xlab = 'Team TOE', ylab = 'Win ratio', main = '2014 Win_Ratio against old tea
```

```
## 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)
```



2015 win ratio against Team TOE

Season 2015 our new TOE:

```r
s15 <- team_stats[team_stats$season == 2015,]
new_mod15 <- lm(win_ratio ~ new_toe, data = s15)
summary(new_mod15)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = s15)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.145230 -0.038499 -0.005498  0.041102  0.179241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.9699     0.3353  -8.857 1.31e-09 ***
## new_toe       7.1203     0.6875  10.357 4.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.076 on 28 degrees of freedom
## Multiple R-squared:  0.793,  Adjusted R-squared:  0.7856
## F-statistic: 107.3 on 1 and 28 DF,  p-value: 4.414e-11
```

Season 2015 old TOE:

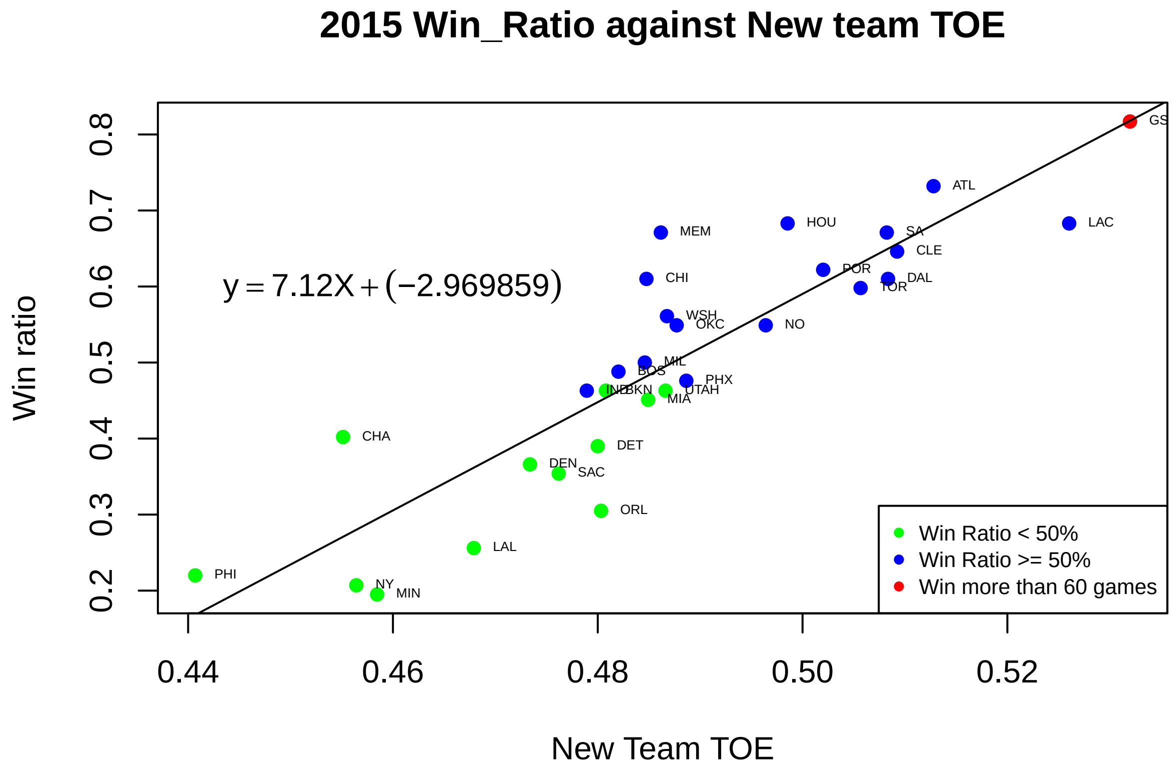
```
mod15 <- lm(win_ratio ~ toe, data = s15)
summary(mod15)
```

```
##
## Call:
## lm(formula = win_ratio ~ toe, data = s15)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.219372 -0.068909 -0.000372  0.063043  0.279014
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.9506     0.5896  -5.005 2.74e-05 ***
## toe           8.0058     1.3671   5.856 2.70e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.112 on 28 degrees of freedom
## Multiple R-squared:  0.5505, Adjusted R-squared:  0.5345
## F-statistic: 34.29 on 1 and 28 DF,  p-value: 2.697e-06
```

```
plot(s15$new_toe,s15$win_ratio,xlab = 'New Team TOE', ylab = 'Win ratio', main = '2015 Win_Ratio agains
```

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

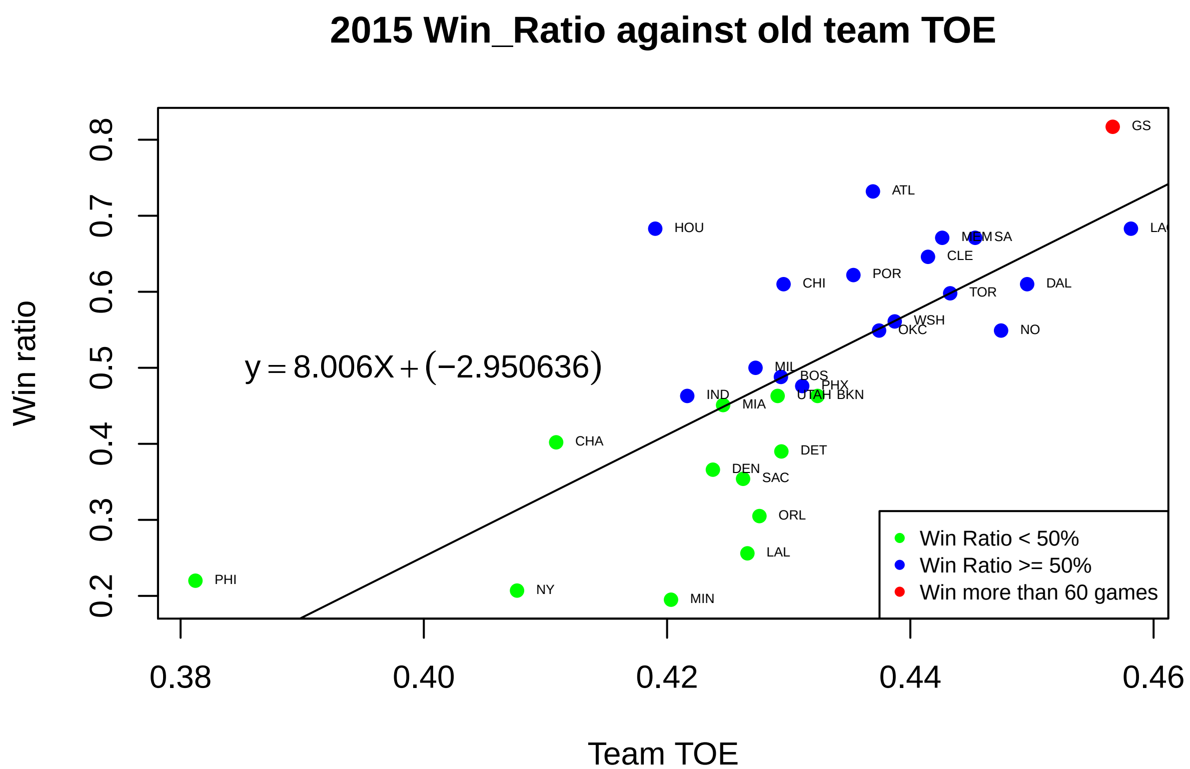
```
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



```
plot(s15$toe,s15$win_ratio,xlab = 'Team TOE', ylab = 'Win ratio', main = '2015 Win_Ratio against old tea
```

```
## 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)
```



2016 win ratio against Team TOE

Season 2016 our new TOE:

```
s16 <- team_stats[team_stats$season == 2016,]
new_mod16 <- lm(win_ratio ~ new_toe, data = s16)
summary(new_mod16)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = s16)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.154603 -0.052252 -0.004685  0.038798  0.180176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.3433     0.3913  -8.544 2.75e-09 ***
## new_toe       7.8113     0.7948   9.828 1.41e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08169 on 28 degrees of freedom
## Multiple R-squared:  0.7753, Adjusted R-squared:  0.7672
## F-statistic:  96.6 on 1 and 28 DF,  p-value: 1.409e-10
```

Season 2016 old TOE:

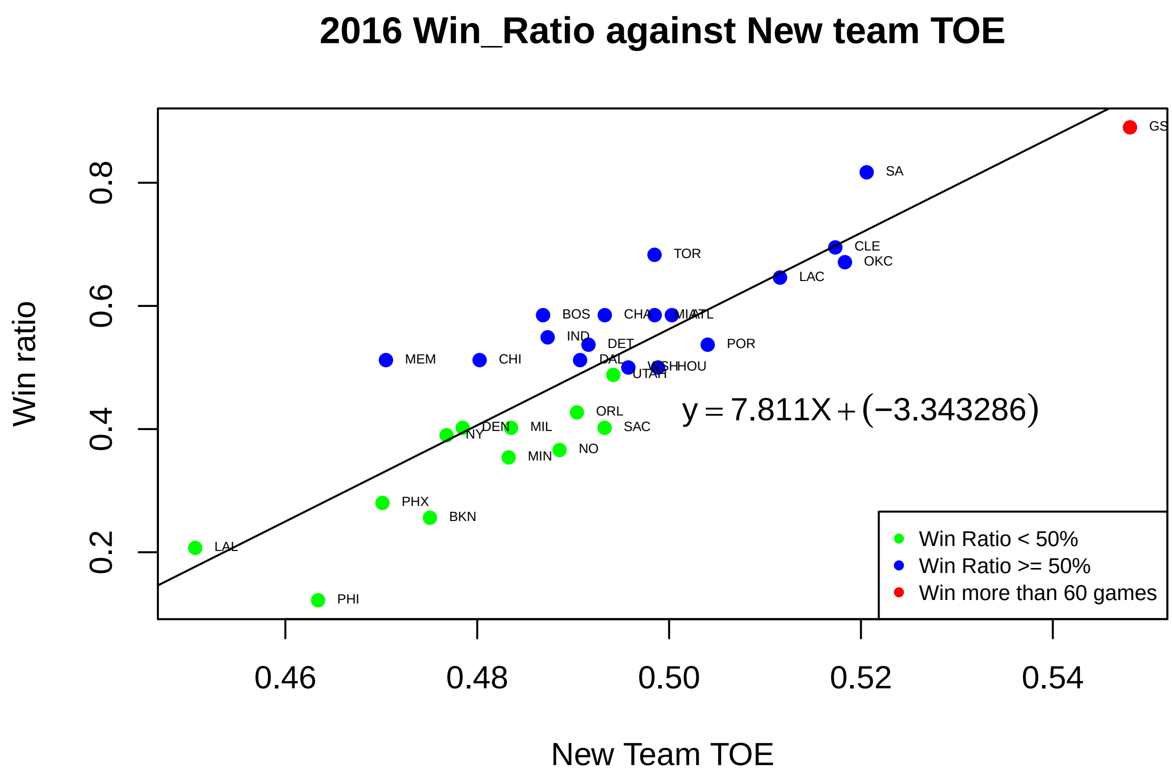
```r
mod16 <- lm(win_ratio ~ toe, data = s16)
summary(mod16)
```

```
##
## Call:
## lm(formula = win_ratio ~ toe, data = s16)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23305 -0.07065  0.01909  0.05874  0.18231
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.333      0.582  -5.727 3.83e-06 ***
## toe            8.874      1.347   6.589 3.81e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1079 on 28 degrees of freedom
## Multiple R-squared:  0.6079, Adjusted R-squared:  0.5939
## F-statistic: 43.42 on 1 and 28 DF,  p-value: 3.806e-07
```

```r
plot(s16$new_toe,s16$win_ratio,xlab = 'New Team TOE', ylab = 'Win ratio', main = '2016 Win_Ratio agains
```

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

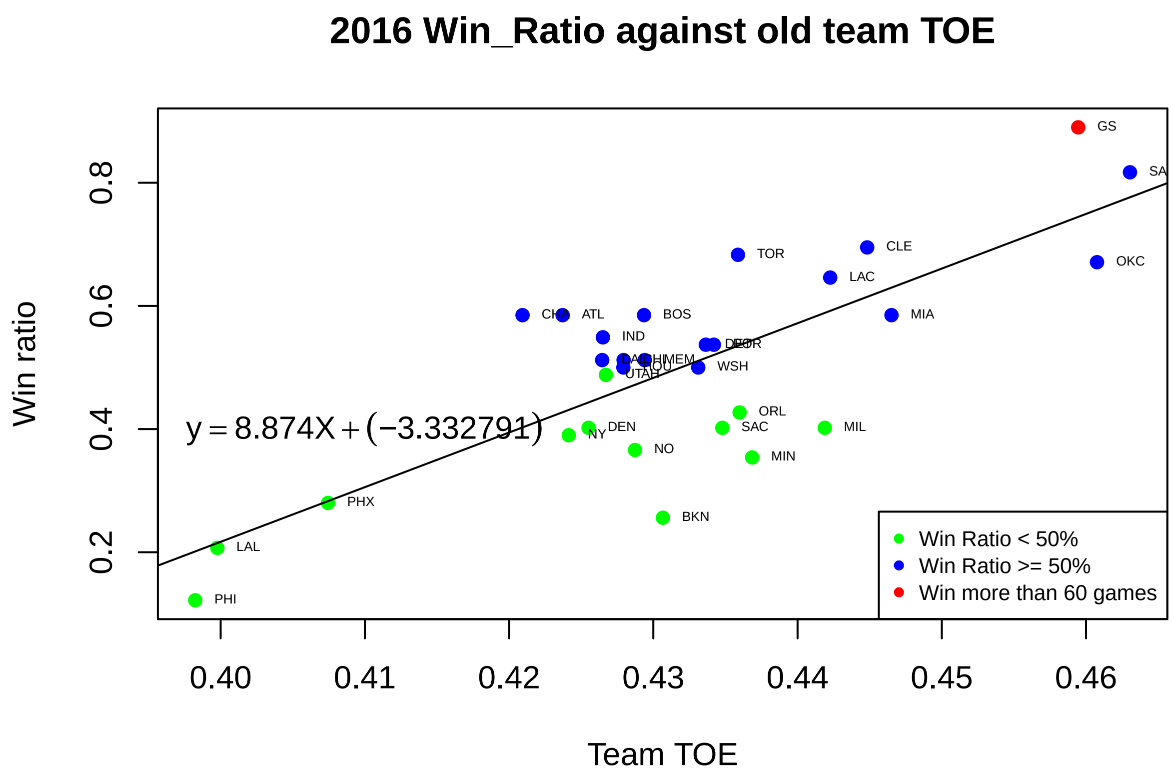
```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```




```
plot(s16$toe,s16$win_ratio,xlab = 'Team TOE', ylab = 'Win ratio', main = '2016 Win_Ratio against old tea
```

```
## 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)
```



2017 win ratio against Team TOE

Season 2017 our new TOE:

```r
s17 <- team_stats[team_stats$season == 2017,]
new_mod17 <- lm(win_ratio ~ new_toe, data = s17)
summary(new_mod17)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = s17)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.133492 -0.046916 -0.009949  0.057958  0.146446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7251     0.4314  -6.318 7.82e-07 ***
## new_toe       6.3994     0.8554   7.481 3.79e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08019 on 28 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6546
## F-statistic: 55.96 on 1 and 28 DF,  p-value: 3.788e-08
```

Season 2017 old TOE:

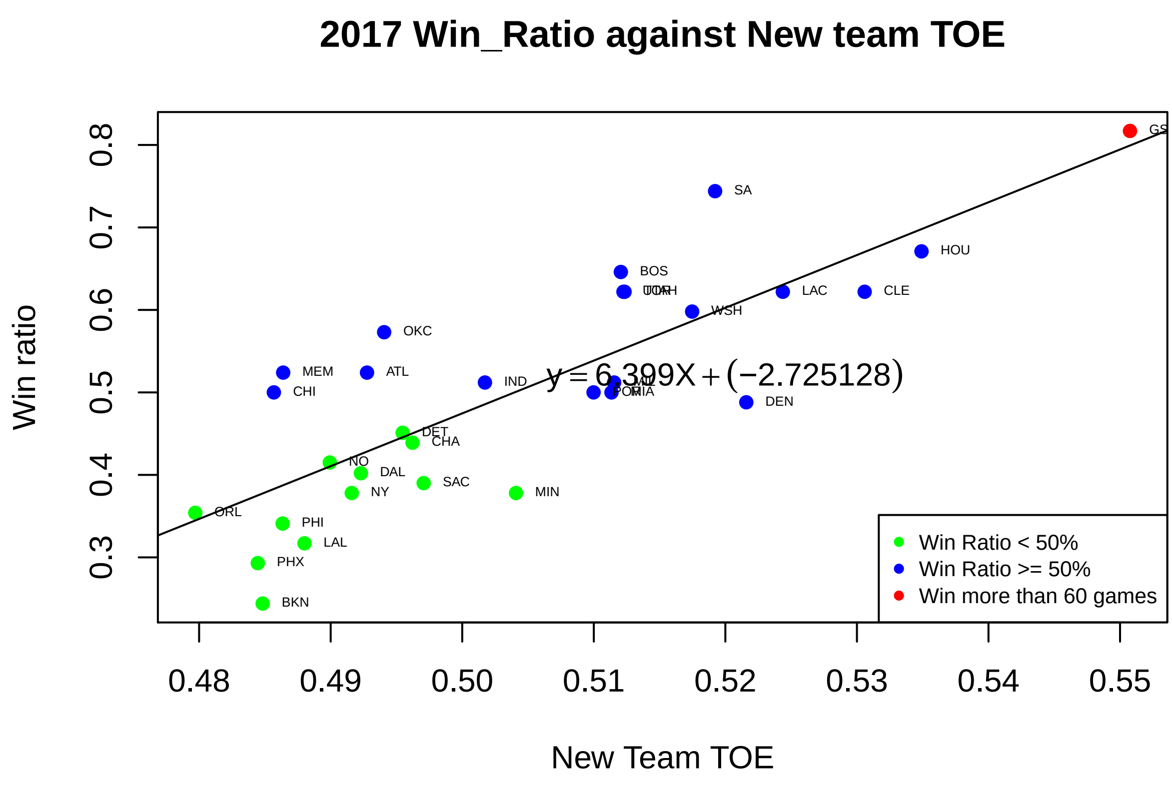
```
mod17 <- lm(win_ratio ~ toe, data = s17)
summary(mod17)
```

```
##
## Call:
## lm(formula = win_ratio ~ toe, data = s17)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.230886 -0.051278 -0.000371  0.068626  0.181794
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.4082     0.6082  -3.959 0.000468 ***
## toe           6.6452     1.3891   4.784 5.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.103 on 28 degrees of freedom
## Multiple R-squared:  0.4497, Adjusted R-squared:  0.4301
## F-statistic: 22.88 on 1 and 28 DF,  p-value: 5.009e-05
```

```
plot(s17$new_toe,s17$win_ratio,xlab = 'New Team TOE', ylab = 'Win ratio', main = '2017 Win_Ratio agains
```

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

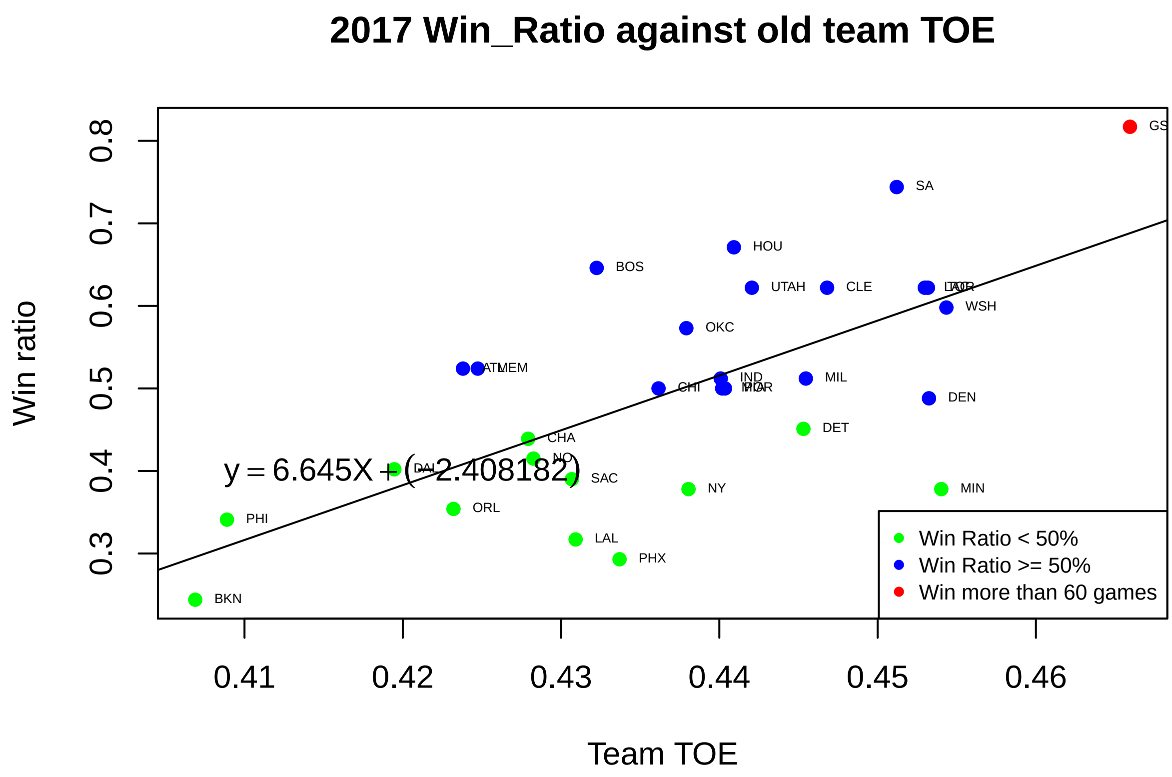
```
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```



```r
plot(s17$toe,s17$win_ratio,xlab = 'Team TOE', ylab = 'Win ratio', main = '2017 Win_Ratio against old tea
```

```
## 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)
```



2018 win ratio against Team TOE

Season 2018 our new TOE:

```r
s18 <- team_stats[team_stats$season == 2018,]
new_mod18 <- lm(win_ratio ~ new_toe, data = s18)
summary(new_mod18)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = s18)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.146570 -0.050708 -0.001863  0.038705  0.188224
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.7317     0.4399  -8.482 3.19e-09 ***
## new_toe       8.3093     0.8634   9.623 2.23e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0731 on 28 degrees of freedom
## Multiple R-squared:  0.7678, Adjusted R-squared:  0.7596
## F-statistic: 92.61 on 1 and 28 DF,  p-value: 2.23e-10
```

Season 2018 old TOE:

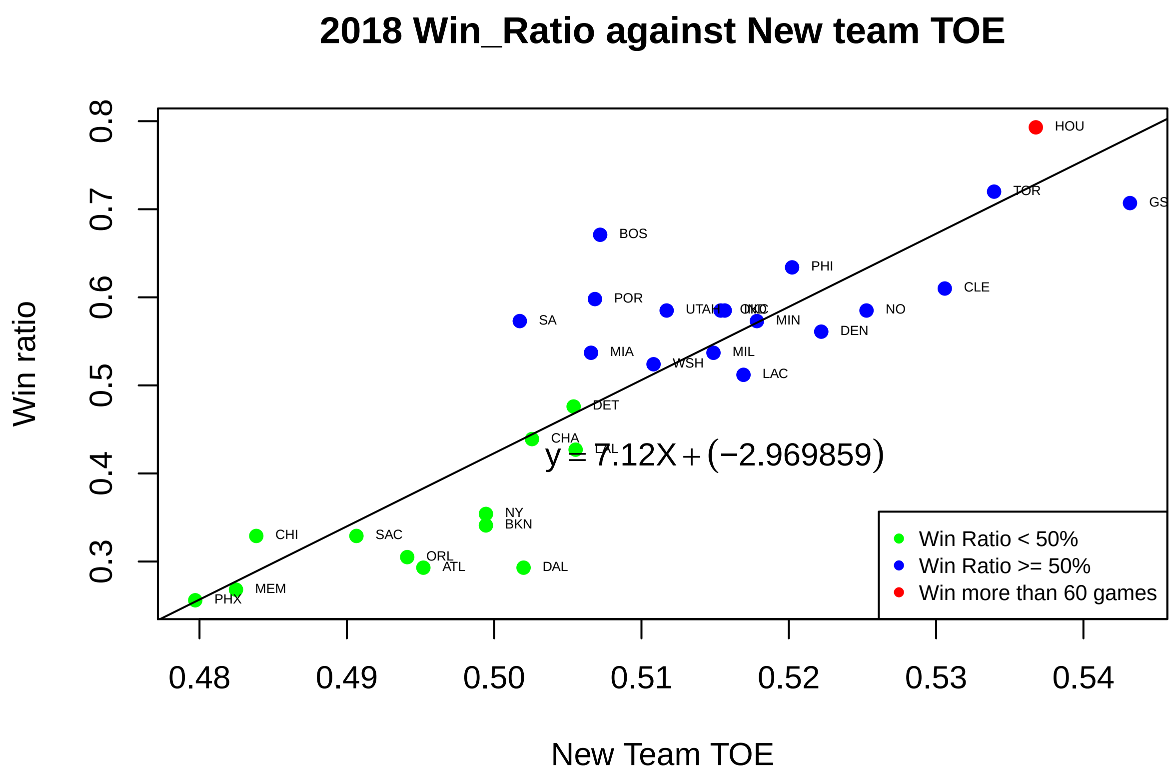
```r
mod18 <- lm(win_ratio ~ toe, data = s18)
summary(mod18)
```

```
##
## Call:
## lm(formula = win_ratio ~ toe, data = s18)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18725 -0.06340 -0.01733  0.03139  0.30990
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.6613     0.6128  -4.343 0.000166 ***
## toe           7.2314     1.4009   5.162 1.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1086 on 28 degrees of freedom
## Multiple R-squared:  0.4876, Adjusted R-squared:  0.4693
## F-statistic: 26.64 on 1 and 28 DF,  p-value: 1.782e-05
```

```r
plot(s18$new_toe,s18$win_ratio,xlab = 'New Team TOE', ylab = 'Win ratio', main = '2018 Win_Ratio agains
```

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

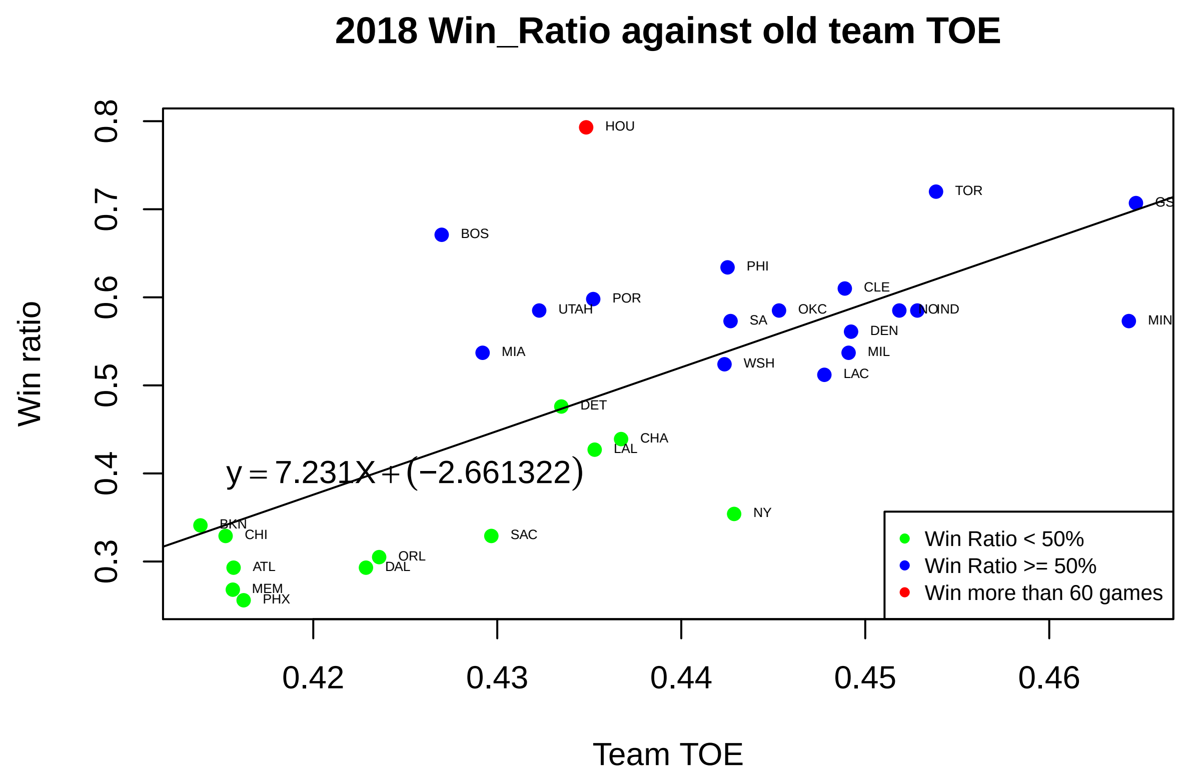
```r
legend("bottomright",legend=c("Win Ratio < 50%", "Win Ratio >= 50%","Win more than 60 games"),
       col=c("green", "blue","red"), pch = c(16,16,16), cex = 0.7)
```




```
plot(s18$toe,s18$win_ratio,xlab = 'Team TOE', ylab = 'Win ratio', main = '2018 Win_Ratio against old tea
```

```
## 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)
```




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

```
## [1] 0.4375821
```

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

```
## [1] 0.5505217
```

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

```
## [1] 0.6079441
```

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

```
## [1] 0.4497334
```

```r
summary(new_mod17)$r.squared
```

```
## [1] 0.6665262
```

```r
summary(new_mod16)$r.squared
```

```
## [1] 0.7752725
```

```r
summary(new_mod15)$r.squared
```

```
## [1] 0.7929965
```

```r
summary(new_mod14)$r.squared
```

```
## [1] 0.6023495
```

```r
mod14$coef
```

```
## (Intercept)         toe
##   -2.787462    7.558199
```

```r
mod15$coef
```

```
## (Intercept)         toe
##   -2.950636    8.005761
```

```r
mod16$coef
```

```
## (Intercept)         toe
##   -3.332791    8.874283
```

```r
mod17$coef
```

```
## (Intercept)         toe
##   -2.408182    6.645190
```

```r
new_mod14$coef
```

```
## (Intercept)     new_toe
##   -2.750489    6.622779
```

```r
new_mod15$coef
```

```
## (Intercept)     new_toe
##   -2.969859    7.120309
```

```r
new_mod16$coef
```

```
## (Intercept)     new_toe
##   -3.343286    7.811340
```

```r
new_mod17$coef
```

```
## (Intercept)     new_toe
##   -2.725128    6.399379
```

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

```
##     season team win_ratio       toe   new_toe color
## 31    2017   GS     0.817 0.4659459 0.5507572   red
## 32    2017   SA     0.744 0.4512055 0.5192194   red
## 33    2017  HOU     0.671 0.4409190 0.5349099  blue
## 34    2017  BOS     0.646 0.4322508 0.5120551  blue
## 35    2017 UTAH     0.622 0.4420550 0.5122549  blue
## 36    2017  TOR     0.622 0.4531792 0.5123384  blue
## 37    2017  CLE     0.622 0.4468085 0.5305889  blue
## 38    2017  LAC     0.622 0.4529817 0.5243688  blue
## 39    2017  WSH     0.598 0.4543454 0.5174746  blue
## 40    2017  OKC     0.573 0.4379157 0.4940644  blue
## 41    2017  MEM     0.524 0.4247375 0.4863905  blue
## 42    2017  ATL     0.524 0.4238042 0.4927620  blue
## 43    2017  IND     0.512 0.4400896 0.5017261  blue
## 44    2017  MIL     0.512 0.4454650 0.5115453  blue
## 45    2017  CHI     0.500 0.4361582 0.4856816  blue
## 46    2017  POR     0.500 0.4403567 0.5099829  blue
## 47    2017  MIA     0.500 0.4401806 0.5113438  blue
## 48    2017  DEN     0.488 0.4532453 0.5215928 green
## 49    2017  DET     0.451 0.4453125 0.4954700 green
## 50    2017  CHA     0.439 0.4279228 0.4962231 green
```

```
## 51    2017   NO     0.415 0.4282585 0.4899329 green
## 52    2017  DAL     0.402 0.4194670 0.4922986 green
## 53    2017  SAC     0.390 0.4306818 0.4970692 green
## 54    2017  MIN     0.378 0.4540230 0.5040888 green
## 55    2017   NY     0.378 0.4380531 0.4916013 green
## 56    2017  ORL     0.354 0.4232044 0.4797069 green
## 57    2017  PHI     0.341 0.4088937 0.4863481 green
## 58    2017  LAL     0.317 0.4309211 0.4880089 green
## 59    2017  PHX     0.293 0.4336957 0.4844617 green
## 60    2017  BKN     0.244 0.4068891 0.4848315 green
## 61    2016   GS     0.890 0.4594595 0.5480447   red
## 62    2016   SA     0.817 0.4630485 0.5205970   red
## 63    2016  CLE     0.695 0.4448276 0.5173224  blue
## 64    2016  TOR     0.683 0.4358670 0.4984802  blue
## 65    2016  OKC     0.671 0.4607623 0.5183276  blue
## 66    2016  LAC     0.646 0.4422633 0.5115590  blue
## 67    2016  MIA     0.585 0.4465116 0.4984967  blue
## 68    2016  BOS     0.585 0.4293538 0.4868642  blue
## 69    2016  CHA     0.585 0.4209329 0.4932945  blue
## 70    2016  ATL     0.585 0.4237102 0.5002872  blue
## 71    2016  IND     0.549 0.4265033 0.4873418  blue
## 72    2016  DET     0.537 0.4336384 0.4915942  blue
## 73    2016  POR     0.537 0.4341957 0.5040230  blue
## 74    2016  DAL     0.512 0.4264538 0.4907193  blue
## 75    2016 CHI      0.512 0.4279379 0.4802483  blue
## 76    2016  MEM     0.512 0.4294049 0.4704841  blue
## 77    2016  HOU     0.500 0.4279228 0.4988413  blue
## 78    2016  WSH     0.500 0.4331140 0.4957555  blue
## 79    2016 UTAH     0.488 0.4267139 0.4941825 green
## 80    2016  ORL     0.427 0.4359823 0.4903955 green
## 81    2016  MIL     0.402 0.4418872 0.4835294 green
## 82    2016  DEN     0.402 0.4255079 0.4784854 green
## 83    2016  SAC     0.402 0.4347826 0.4932735 green
## 84    2016   NY     0.390 0.4241379 0.4768056 green
## 85    2016   NO     0.366 0.4287305 0.4885845 green
## 86    2016  MIN     0.354 0.4368482 0.4832736 green
## 87    2016  PHX     0.280 0.4074480 0.4701240 green
## 88    2016  BKN     0.256 0.4306652 0.4750716 green
## 89    2016  LAL     0.207 0.3997722 0.4506066 green
## 90    2016  PHI     0.122 0.3982398 0.4634146 green
## 91    2015   GS     0.817 0.4566411 0.5319751   red
## 92    2015  ATL     0.732 0.4369266 0.5127900   red
## 93    2015  HOU     0.683 0.4190260 0.4985406  blue
## 94    2015  LAC     0.683 0.4581395 0.5260355  blue
## 95    2015  MEM     0.671 0.4426230 0.4861613  blue
## 96    2015   SA     0.671 0.4453303 0.5082256  blue
## 97    2015  CLE     0.646 0.4414520 0.5092317  blue
## 98    2015  POR     0.622 0.4353206 0.5020150  blue
## 99    2015  CHI     0.610 0.4295775 0.4847579  blue
## 100   2015  DAL     0.610 0.4496036 0.5083477  blue
## 101   2015  TOR     0.598 0.4432749 0.5056716  blue
## 102   2015  WSH     0.561 0.4387171 0.4867569  blue
## 103   2015   NO     0.549 0.4474616 0.4964072  blue
## 104   2015  OKC     0.549 0.4374295 0.4877073  blue
```

```
## 105   2015  MIL     0.500 0.4272727 0.4845972  blue
## 106   2015  BOS     0.488 0.4293598 0.4820225 green
## 107   2015  PHX     0.476 0.4311111 0.4886493 green
## 108   2015  IND     0.463 0.4216590 0.4789318 green
## 109   2015 UTAH     0.463 0.4290909 0.4866210 green
## 110   2015  BKN     0.463 0.4323699 0.4808033 green
## 111   2015  MIA     0.451 0.4246080 0.4849246 green
## 112   2015  CHA     0.402 0.4108796 0.4551358 green
## 113   2015  DET     0.390 0.4293981 0.4800000 green
## 114   2015  DEN     0.366 0.4237668 0.4733862 green
## 115   2015  SAC     0.354 0.4262485 0.4761905 green
## 116   2015  ORL     0.305 0.4275941 0.4803288 green
## 117   2015  LAL     0.256 0.4266055 0.4679005 green
## 118   2015  PHI     0.220 0.3812217 0.4406977 green
## 119   2015   NY     0.207 0.4076655 0.4564315 green
## 120   2015  MIN     0.195 0.4203233 0.4584561 green
## 121   2014   SA     0.756 0.4582393 0.5236427   red
## 122   2014  OKC     0.720 0.4472477 0.5088652  blue
## 123   2014  LAC     0.695 0.4551804 0.5185407  blue
## 124   2014  IND     0.683 0.4230317 0.4816401  blue
## 125   2014  MIA     0.659 0.4575866 0.5270691  blue
## 126   2014  POR     0.659 0.4433107 0.5048991  blue
## 127   2014  HOU     0.659 0.4449649 0.5172414  blue
## 128   2014   GS     0.622 0.4403567 0.5042784  blue
## 129   2014  MEM     0.610 0.4530321 0.4885542  blue
## 130   2014  DAL     0.598 0.4556962 0.5134189  blue
## 131   2014  TOR     0.585 0.4314421 0.4924653  blue
## 132   2014  CHI     0.585 0.4145759 0.4669549  blue
## 133   2014  PHX     0.585 0.4422857 0.5070175  blue
## 134   2014  WSH     0.537 0.4382786 0.4919262  blue
## 135   2014  BKN     0.537 0.4270335 0.4932182  blue
## 136   2014  CHA     0.524 0.4275618 0.4747292  blue
## 137   2014  MIN     0.488 0.4375703 0.4846241 green
## 138   2014  ATL     0.463 0.4229025 0.4967475 green
## 139   2014   NY     0.451 0.4361702 0.4996993 green
## 140   2014  DEN     0.439 0.4290503 0.4880000 green
## 141   2014   NO     0.415 0.4452297 0.4877319 green
## 142   2014  CLE     0.402 0.4257768 0.4682448 green
## 143   2014  DET     0.354 0.4470046 0.4799542 green
## 144   2014  SAC     0.341 0.4310748 0.4781306 green
## 145   2014  LAL     0.329 0.4204171 0.4842342 green
## 146   2014 UTAH     0.305 0.4250295 0.4729242 green
## 147   2014  BOS     0.305 0.4180985 0.4694836 green
## 148   2014  ORL     0.280 0.4197952 0.4744268 green
## 149   2014  PHI     0.232 0.4097297 0.4607679 green
## 150   2014  MIL     0.183 0.4210526 0.4648553 green
```

```r
new_total_mod <-lm(win_ratio ~ new_toe,data = sample_data)
old_mod <- lm(win_ratio ~ toe,data = sample_data)
summary(new_total_mod)
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = sample_data)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.201148 -0.063004  0.001647  0.057389  0.257980 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -2.6127     0.2153  -12.13   <2e-16 ***
## new_toe       6.3069     0.4360   14.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09369 on 118 degrees of freedom
## Multiple R-squared:  0.6394, Adjusted R-squared:  0.6364 
## F-statistic: 209.3 on 1 and 118 DF,  p-value: < 2.2e-16
```

```r
summary(old_mod)
```

```
## 
## Call:
## lm(formula = win_ratio ~ toe, data = sample_data)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.274446 -0.067617 -0.000689  0.075331  0.295383 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -2.7832     0.3038   -9.16 1.94e-15 ***
## toe           7.5672     0.6999   10.81  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1106 on 118 degrees of freedom
## Multiple R-squared:  0.4976, Adjusted R-squared:  0.4934 
## F-statistic: 116.9 on 1 and 118 DF,  p-value: < 2.2e-16
```

```r
AIC(new_total_mod)
```

```
## [1] -223.7303
```

```r
AIC(old_mod)
```

```
## [1] -183.9364
```

```r
BIC(new_total_mod)
```

```
## [1] -215.3679
```

```r
BIC(old_mod)
```

```
## [1] -175.5739
```

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

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

trainingRowIndex <- sample(1:nrow(sample_data), 0.833*nrow(sample_data))  # row incices for training da
```

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

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

train_old_toe<- lm(win_ratio ~ toe, data=trainingData)  # build the model

predict_old_toe <- predict(train_old_toe, testData)  # predict

summary(train_old_toe) # model summary
```

```
##
## Call:
## lm(formula = win_ratio ~ toe, data = trainingData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.269236 -0.062468  0.003937  0.070826  0.288794
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.6354     0.3269  -8.062 1.98e-12 ***
## toe           7.2301     0.7536   9.594 1.01e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1099 on 97 degrees of freedom
## Multiple R-squared:  0.4869, Adjusted R-squared:  0.4816
## F-statistic: 92.05 on 1 and 97 DF,  p-value: 1.008e-15
```

```r
# Calculate: akaike information criterion
AIC(train_old_toe)
```

```
## [1] -152.2682
```

```r
actuals_preds <- data.frame(cbind(actuals=testData$win_ratio, predicteds=predict_old_toe))
# make actuals_predicted dataframe.

correlation_accuracy <- cor(actuals_preds)
correlation_accuracy
```

```
##              actuals predicteds
## actuals    1.0000000  0.7563707
## predicteds 0.7563707  1.0000000
```

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

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

trainingRowIndex <- sample(1:nrow(sample_data), 0.833*nrow(sample_data))  # row incices for training da

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

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

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

```r
predict_new_toe <- predict(train_new_toe, testData)  # predict

summary(train_new_toe) # model summary
```

```
##
## Call:
## lm(formula = win_ratio ~ new_toe, data = trainingData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.200640 -0.061954  0.002038  0.059417  0.258763
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.6550     0.2426  -10.94   <2e-16 ***
## new_toe       6.3931     0.4914   13.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09261 on 97 degrees of freedom
## Multiple R-squared:  0.6357, Adjusted R-squared:  0.632
## F-statistic: 169.3 on 1 and 97 DF,  p-value: < 2.2e-16
```

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

```
## [1] -186.1824
```

```r
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  0.8095832
## predicteds 0.8095832  1.0000000
```

5 - Fold Cross Validation - old toe

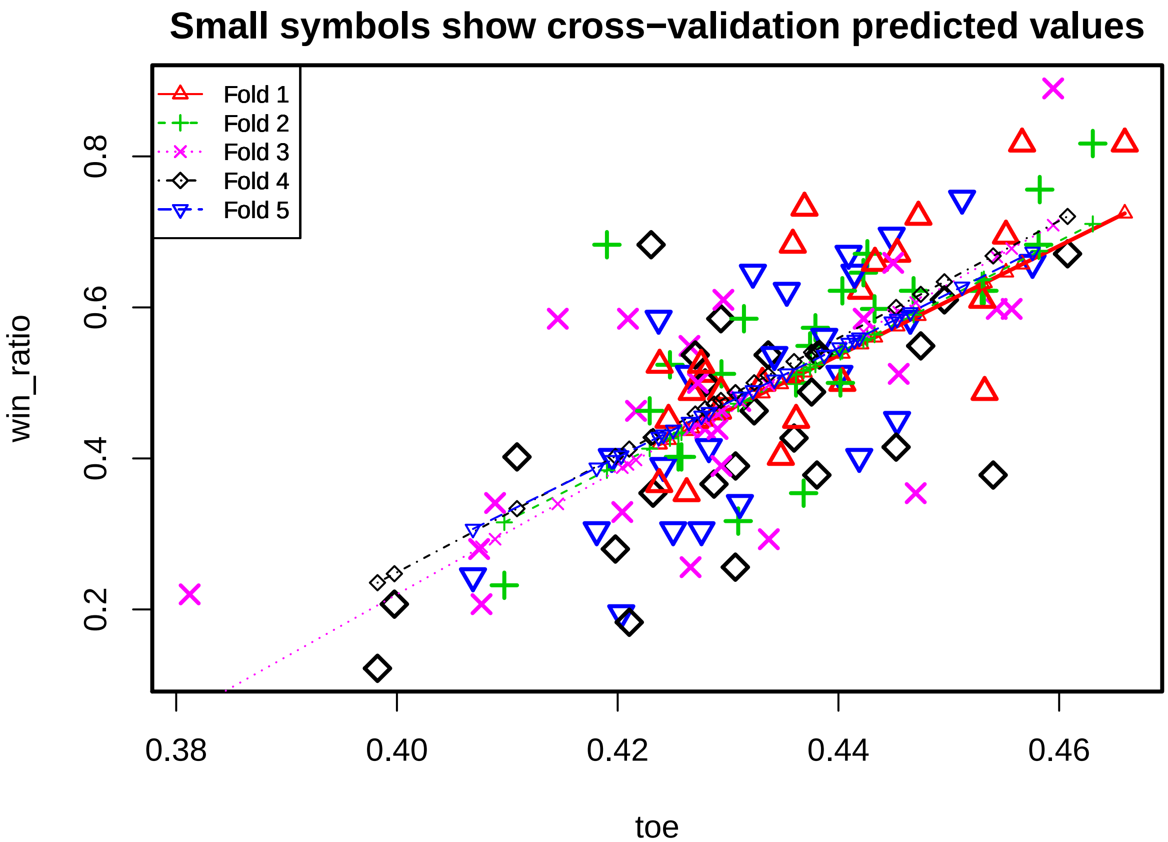
```r
library(DAAG)
```

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

```r
sample_data <-subset(team_stats,team_stats$season !=2018)
cv.lm(sample_data, form.lm = formula(win_ratio ~ toe), m=5, dots = FALSE, seed=123, plotit=TRUE, printi
```

```
## Analysis of Variance Table
##
## Response: win_ratio
##            Df Sum Sq Mean Sq F value Pr(>F)
## toe         1   1.43   1.430     117 <2e-16 ***
## Residuals 118   1.44   0.012
## ---
```

```
## 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    79
## toe           0.4659 0.4421 0.424  0.4404  0.453 0.436 0.4279 0.4331 0.427
## cvpred        0.7246 0.5513 0.419  0.5389  0.632 0.506 0.4489 0.4864 0.440
## win_ratio     0.8170 0.6220 0.524  0.5000  0.488 0.683 0.5120 0.5000 0.488
## CV residual 0.0924 0.0707 0.105 -0.0389 -0.144 0.177 0.0631 0.0136 0.048
##                    83    91    92    96    106     109    111     114
## toe            0.4348 0.457 0.437 0.445 0.4294 0.42909 0.4246  0.4238
## cvpred         0.4985 0.657 0.514 0.575 0.4592 0.45722 0.4247  0.4186
## win_ratio      0.4020 0.817 0.732 0.671 0.4880 0.46300 0.4510  0.3660
## CV residual -0.0965 0.160 0.218 0.096 0.0288 0.00578 0.0263 -0.0526
##                   115   122    123    126     129    136     139
## toe            0.4262 0.447 0.4552 0.4433  0.4530 0.4276  0.4362
## cvpred         0.4366 0.589 0.6465 0.5604  0.6309 0.4461  0.5086
## win_ratio      0.3540 0.720 0.6950 0.6590  0.6100 0.5240  0.4510
## CV residual -0.0826 0.131 0.0485 0.0986 -0.0209 0.0779 -0.0576
##
## Sum of squares = 0.23    Mean square = 0.01    n = 24
##
## fold 2
## Observations in test set: 24
##                    36     37      38     40     41      45      47      58
## toe            0.4532 0.4468  0.4530 0.4379 0.4247  0.4362  0.4402  0.431
## cvpred         0.6374 0.5902  0.6359 0.5243 0.4266  0.5112  0.5411  0.472
```

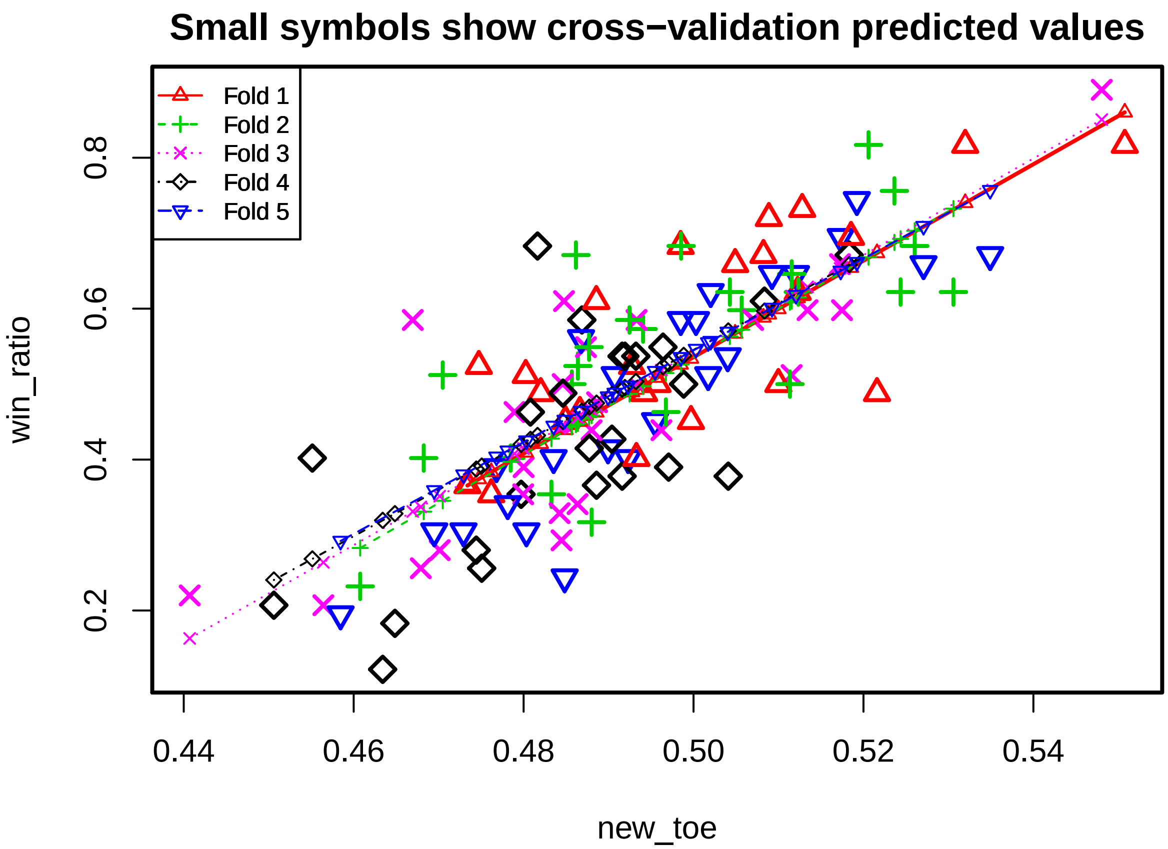
```
## win_ratio    0.6220 0.6220  0.6220 0.5730 0.5240  0.5000  0.5000  0.317
## CV residual -0.0154 0.0318 -0.0139 0.0487 0.0974 -0.0112 -0.0411 -0.155
##                62     66     76      82     86    93      94    95   101
## toe         0.463 0.4423 0.4294  0.4255  0.437 0.419 0.45814 0.443 0.443
## cvpred      0.711 0.5565 0.4612  0.4323  0.516 0.384 0.67418 0.559 0.564
## win_ratio   0.817 0.6460 0.5120  0.4020  0.354 0.683 0.68300 0.671 0.598
## CV residual 0.106 0.0895 0.0508 -0.0303 -0.162 0.299 0.00882 0.112 0.034
##               104    121    128   131   138     142     149
## toe         0.4374 0.4582 0.4404 0.431 0.423  0.4258  0.4097
## cvpred      0.5207 0.6749 0.5424 0.476 0.413  0.4343  0.3153
## win_ratio   0.5490 0.7560 0.6220 0.585 0.463  0.4020  0.2320
## CV residual 0.0283 0.0811 0.0796 0.109 0.050 -0.0323 -0.0833
##
## Sum of squares = 0.23    Mean square = 0.01    n = 24
##
## fold 3
## Observations in test set: 24
##                  39      44      50    57     59    61    69    71
## toe          0.4543  0.4455  0.4279 0.409  0.434 0.459 0.421 0.427
## cvpred       0.6668  0.5938  0.4495 0.293  0.497 0.709 0.392 0.438
## win_ratio    0.5980  0.5120  0.4390 0.341  0.293 0.890 0.585 0.549
## CV residual -0.0688 -0.0818 -0.0105 0.048 -0.204 0.181 0.193 0.111
##                   87    99    105      107   108     113    117    118
## toe          0.40745 0.430 0.4273 0.431111 0.422  0.4294  0.427 0.3812
## cvpred       0.28116 0.463 0.4442 0.475755 0.398  0.4617  0.439 0.0655
## win_ratio    0.28000 0.610 0.5000 0.476000 0.463  0.3900  0.256 0.2200
## CV residual -0.00116 0.147 0.0558 0.000245 0.065 -0.0717 -0.183 0.1545
##                 119    127     130   132    133     140    143     145
## toe          0.4077 0.4450  0.4557 0.415 0.4423  0.4291  0.447  0.4204
## cvpred       0.2829 0.5897  0.6779 0.340 0.5677  0.4588  0.606  0.3878
## win_ratio    0.2070 0.6590  0.5980 0.585 0.5850  0.4390  0.354  0.3290
## CV residual -0.0759 0.0693 -0.0799 0.245 0.0173 -0.0198 -0.252 -0.0588
##
## Sum of squares = 0.37    Mean square = 0.02    n = 24
##
## fold 4
## Observations in test set: 24
##                  53     54     55      56      65    68    72     77
## toe          0.4307  0.454  0.438  0.4232  0.4608 0.429 0.434 0.4279
## cvpred       0.4871  0.668  0.544  0.4291  0.7205 0.477 0.510 0.4657
## win_ratio    0.3900  0.378  0.378  0.3540  0.6710 0.585 0.537 0.5000
## CV residual -0.0971 -0.290 -0.166 -0.0751 -0.0495 0.108 0.027 0.0343
##                 80     85     88      89     90     100     103     110
## toe          0.436  0.429  0.431  0.3998  0.398  0.4496  0.4475  0.4324
## cvpred       0.528  0.472  0.487  0.2473  0.235  0.6339  0.6173  0.5002
## win_ratio    0.427  0.366  0.256  0.2070  0.122  0.6100  0.5490  0.4630
## CV residual -0.101 -0.106 -0.231 -0.0403 -0.113 -0.0239 -0.0683 -0.0372
##               112   124     134    135     137    141    148    150
## toe         0.4109 0.423  0.43828 0.4270  0.4376  0.445  0.420  0.421
## cvpred      0.3335 0.428  0.54603 0.4588  0.5405  0.600  0.403  0.412
## win_ratio   0.4020 0.683  0.53700 0.5370  0.4880  0.415  0.280  0.183
## CV residual 0.0685 0.255 -0.00903 0.0782 -0.0525 -0.185 -0.123 -0.229
##
## Sum of squares = 0.42    Mean square = 0.02    n = 24
```

```
##
## fold 5
## Observations in test set: 24
##                 32    33    34      43     49      51      52      60    63
## toe          0.451 0.441 0.432  0.4401  0.445  0.4283 0.41947  0.4069 0.445
## cvpred       0.627 0.553 0.490  0.5469  0.585  0.4613 0.39763  0.3066 0.581
## win_ratio    0.744 0.671 0.646  0.5120  0.451  0.4150 0.40200  0.2440 0.695
## CV residual 0.117 0.118 0.156 -0.0349 -0.134 -0.0463 0.00437 -0.0626 0.114
##                    67    70     73     74     81      84     97    98   102
## toe           0.44651 0.424 0.4342 0.4265  0.442  0.4241 0.4415 0.435 0.439
## cvpred        0.59337 0.428 0.5042 0.4482  0.560  0.4314 0.5567 0.512 0.537
## win_ratio     0.58500 0.585 0.5370 0.5120  0.402  0.3900 0.6460 0.622 0.561
## CV residual -0.00837 0.157 0.0328 0.0638 -0.158 -0.0414 0.0893 0.110 0.024
##                 116    120     125    144    146     147
## toe           0.428  0.420  0.4576  0.431  0.425  0.4181
## cvpred        0.456  0.404  0.6735  0.482  0.438  0.3877
## win_ratio     0.305  0.195  0.6590  0.341  0.305  0.3050
## CV residual -0.151 -0.209 -0.0145 -0.141 -0.133 -0.0827
##
## Sum of squares = 0.28    Mean square = 0.01    n = 24
##
## Overall (Sum over all 24 folds)
##     ms
## 0.0127
```

5 - Fold Cross Validation - new toe

```
library(DAAG)
sample_data <-subset(team_stats,team_stats$season !=2018)
cv.lm(sample_data, form.lm = formula(win_ratio ~ new_toe), m=5, dots = FALSE, seed=123, plotit=TRUE, pr
```

```
## Analysis of Variance Table
##
## Response: win_ratio
##            Df Sum Sq Mean Sq F value Pr(>F)
## new_toe     1   1.84   1.837     209 <2e-16 ***
## Residuals 118   1.04   0.009
## ---
## 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
## new_toe       0.5508 0.51225 0.4928  0.5100  0.522 0.498 0.480  0.49576
## cvpred        0.8601 0.61406 0.4895  0.5995  0.674 0.526 0.409  0.50861
## win_ratio     0.8170 0.62200 0.5240  0.5000  0.488 0.683 0.512  0.50000
## CV residual -0.0431 0.00794 0.0345 -0.0995 -0.186 0.157 0.103 -0.00861
##                   79      83     91    92     96    106    109    111
## new_toe       0.4942  0.4933 0.5320 0.513 0.5082 0.4820 0.4866 0.4849
## cvpred        0.4986  0.4927 0.7401 0.617 0.5883 0.4208 0.4502 0.4394
## win_ratio     0.4880  0.4020 0.8170 0.732 0.6710 0.4880 0.4630 0.4510
## CV residual -0.0106 -0.0907 0.0769 0.115 0.0827 0.0672 0.0128 0.0116
##                  114     115    122    123    126   129   136     139
## new_toe      0.47339  0.4762 0.509 0.5185 0.5049 0.489 0.475  0.4997
## cvpred       0.36564  0.3836 0.592 0.6542 0.5671 0.463 0.374  0.5338
## win_ratio    0.36600  0.3540 0.720 0.6950 0.6590 0.610 0.524  0.4510
## CV residual  0.00036 -0.0296 0.128 0.0408 0.0919 0.147 0.150 -0.0828
##
## Sum of squares = 0.2    Mean square = 0.01    n = 24
##
## fold 2
## Observations in test set: 24
##                   36      37     38     40     41     45     47     58
## new_toe      0.51234   0.531  0.5244 0.4941 0.4864 0.4857  0.511  0.488
## cvpred       0.61497   0.733  0.6925 0.4973 0.4478 0.4433  0.609  0.458
## win_ratio    0.62200   0.622  0.6220 0.5730 0.5240 0.5000  0.500  0.317
```

```
## CV residual 0.00703 -0.111 -0.0705 0.0757 0.0762 0.0567 -0.109 -0.141
##                62    66    76      82      86    93      94    95   101
## new_toe     0.521 0.512 0.470 0.47849  0.4833 0.499  0.5260 0.486 0.506
## cvpred      0.668 0.610 0.345 0.39693  0.4278 0.526  0.7032 0.446 0.572
## win_ratio   0.817 0.646 0.512 0.40200  0.3540 0.683  0.6830 0.671 0.598
## CV residual 0.149 0.036 0.167 0.00507 -0.0738 0.157 -0.0202 0.225 0.026
##                104    121    128   131     138   142     149
## new_toe     0.4877 0.5236 0.5043 0.492  0.4967 0.468  0.4608
## cvpred      0.4563 0.6878 0.5631 0.487  0.5146 0.331  0.2828
## win_ratio   0.5490 0.7560 0.6220 0.585  0.4630 0.402  0.2320
## CV residual 0.0927 0.0682 0.0589 0.098 -0.0516 0.071 -0.0508
##
## Sum of squares = 0.23    Mean square = 0.01    n = 24
##
## fold 3
## Observations in test set: 24
##                  39     44      50     57     59     61     69     71
## new_toe      0.5175  0.512  0.4962  0.486  0.484 0.5480 0.4933 0.4873
## cvpred       0.6547  0.617  0.5186  0.455  0.443 0.8505 0.4999 0.4617
## win_ratio    0.5980  0.512  0.4390  0.341  0.293 0.8900 0.5850 0.5490
## CV residual -0.0567 -0.105 -0.0796 -0.114 -0.150 0.0395 0.0851 0.0873
##                  87    99    105     107    108     113     117   118
## new_toe      0.4701 0.485 0.4846 0.48865 0.4789  0.4800  0.4679 0.441
## cvpred       0.3515 0.445 0.4442 0.47011 0.4079  0.4147  0.3372 0.163
## win_ratio    0.2800 0.610 0.5000 0.47600 0.4630  0.3900  0.2560 0.220
## CV residual -0.0715 0.165 0.0558 0.00589 0.0551 -0.0247 -0.0812 0.057
##                 119     127     130   132      133    140     143    145
## new_toe      0.4564 0.51724  0.5134 0.467  0.50702  0.488  0.4800  0.484
## cvpred       0.2638 0.65324  0.6288 0.331  0.58776  0.466  0.4144  0.442
## win_ratio    0.2070 0.65900  0.5980 0.585  0.58500  0.439  0.3540  0.329
## CV residual -0.0568 0.00576 -0.0308 0.254 -0.00276 -0.027 -0.0604 -0.113
##
## Sum of squares = 0.21    Mean square = 0.01    n = 24
##
## fold 4
## Observations in test set: 24
##                 53     54     55     56     65    68     72      77     80
## new_toe      0.497  0.504  0.492  0.480 0.5183 0.487 0.4916  0.4988  0.490
## cvpred       0.527  0.571  0.493  0.420 0.6584 0.464 0.4934  0.5381  0.486
## win_ratio    0.390  0.378  0.378  0.354 0.6710 0.585 0.5370  0.5000  0.427
## CV residual -0.137 -0.193 -0.115 -0.066 0.0126 0.121 0.0436 -0.0381 -0.059
##                 85     88      89     90    100    103    110   112   124
## new_toe      0.489  0.475  0.4506  0.463 0.5083 0.4964 0.4808 0.455 0.482
## cvpred       0.475  0.391  0.2405  0.320 0.5968 0.5231 0.4268 0.268 0.432
## win_ratio    0.366  0.256  0.2070  0.122 0.6100 0.5490 0.4630 0.402 0.683
## CV residual -0.109 -0.135 -0.0335 -0.198 0.0132 0.0259 0.0362 0.134 0.251
##                134    135    137     141    148    150
## new_toe     0.4919 0.4932 0.4846  0.4877  0.474  0.465
## cvpred      0.4955 0.5034 0.4504  0.4696  0.387  0.328
## win_ratio   0.5370 0.5370 0.4880  0.4150  0.280  0.183
## CV residual 0.0415 0.0336 0.0376 -0.0546 -0.107 -0.145
##
## Sum of squares = 0.29    Mean square = 0.01    n = 24
##
```

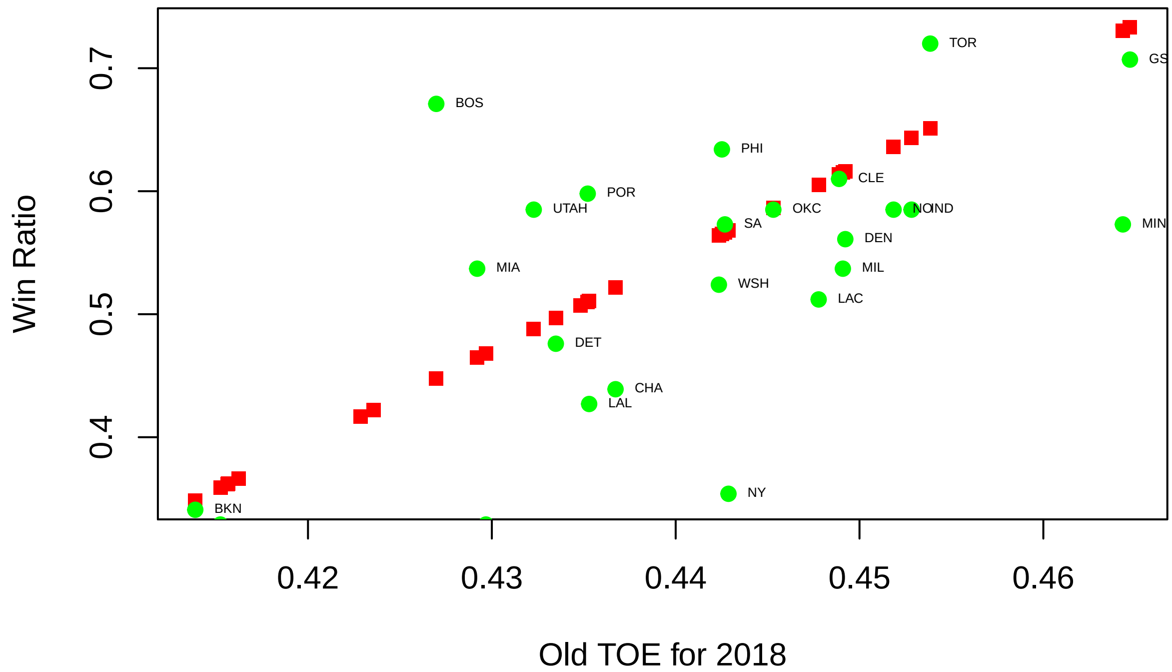
```
## fold 5
## Observations in test set: 24
##                  32      33     34      43      49      51     52     60
## new_toe     0.5192  0.5349 0.5121  0.5017  0.4955  0.4899  0.492  0.485
## cvpred      0.6617  0.7571 0.6181  0.5553  0.5172  0.4836  0.498  0.453
## win_ratio   0.7440  0.6710 0.6460  0.5120  0.4510  0.4150  0.402  0.244
## CV residual 0.0823 -0.0861 0.0279 -0.0433 -0.0662 -0.0686 -0.096 -0.209
##                  63     67     70      73     74      81      84     97
## new_toe     0.5173 0.4985 0.5003  0.5040 0.4907  0.4835  0.4768 0.5092
## cvpred      0.6501 0.5356 0.5465  0.5692 0.4883  0.4446  0.4037 0.6009
## win_ratio   0.6950 0.5850 0.5850  0.5370 0.5120  0.4020  0.3900 0.6460
## CV residual 0.0449 0.0494 0.0385 -0.0322 0.0237 -0.0426 -0.0137 0.0451
##                 98    102    116     120     125     144     146     147
## new_toe     0.502 0.4868  0.480  0.4585  0.5271  0.4781  0.4729  0.4695
## cvpred      0.557 0.4643  0.425  0.2922  0.7094  0.4118  0.3801  0.3592
## win_ratio   0.622 0.5610  0.305  0.1950  0.6590  0.3410  0.3050  0.3050
## CV residual 0.065 0.0967 -0.120 -0.0972 -0.0504 -0.0708 -0.0751 -0.0542
##
## Sum of squares = 0.14    Mean square = 0.01    n = 24
##
## Overall (Sum over all 24 folds)
##      ms
## 0.00894
```

predict 2018 and compare with the actual results

using old toe:

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

```
plot(data1$toe,data1$fit,pch=15,col="red",xlab = "Old TOE for 2018",ylab = "Win Ratio")+points(data1$to
```




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

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

```
## [1] 0.35
```

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

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

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

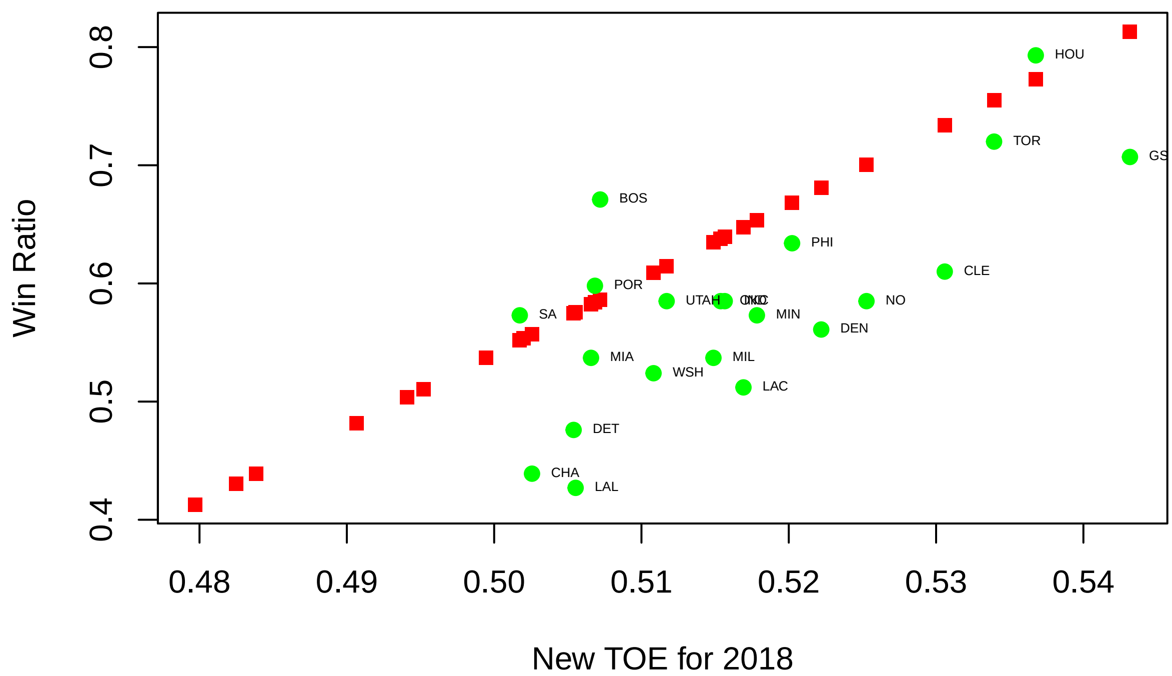
```
## [1] 0.458
```

using new toe:

```r
x1<-subset(team_stats, season==2018, select=c(team,win_ratio,new_toe))
s18 <- team_stats[team_stats$season == 2018,]
pred1 <- predict(new_total_mod,s18,interval = "confidence")
data2 <-cbind(x1,pred1)
mean <- mean(data2$win_ratio)
mean
```

```
## [1] 0.5
```

```r
plot(data2$new_toe,data2$fit,pch=15,col="red",xlab = "New TOE for 2018",ylab = "Win Ratio")+points(data2
```




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

```r
SSE <-sum((data2$fit-data2$win_ratio)^2)
SSE
```

```
## [1] 0.474
```

```r
SSTO <- sum((data2$win_ratio - mean(data2$win_ratio))^2)
SSTO
```

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

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

```
## [1] 0.265
```