Data 621 Homework 1: Code Appendix

Jeff Nieman, Scott Karr, James Topor, Armenoush Aslanian-Persico

Contents

This Appendix contains all of the source R code and associated relevant output from our final writeup and our model building efforts. The R code is organized to match up to the relevant sections of the Writeup document.

However, we begin here by providing the full ouput of our "best" linear model as indicated in Part 4 of the final writeup document.

Full Results of our "Best" Model

The table displayed below presents the results of our selected linear model's TARGET_WINS estimates for the 259 records contained in the Evaluation data set:

INDEX	TARGET_WINS
9	61
10	66
14	72
47	86
60	66
63	73
74	82
83	71
98	69
120	74
123	68
135	81
138	82
140	85
151	90
153	74
171	72
184	79
193	71
213	89
217	86

INDEX	TARGET_WINS
226	85
230	82
241	74
291	83
294	89
300	81
348	75
350	87
357	76
367	93
368	86
372	85
382	84
388	80
396	86
398	77
403	87
407	83
410	89
412	89
414	101
436	77
440	100
476	87
479	94
481	92
501	75
503	69
506	80
519	77
522	86
550	73
554	76
566	72
578	80
596	88
599	74
605	63
607	78
614	86
644	78
692	88
699	88
700	86
716	99
721	70
722	74
729	76
731	83
746	84
763	75
774	78

INDEX	TARGET_WINS
776	90
788	76
789	75
792	83
811	82
835	71
837	75
861	88
862	86
863	95
871	74
879	84
887	81
892	85
904	84
909	93
925	92
940	78
951	101
976	74
981	83
983	83
984	83
989	87
995	98
1000	87
1001	85
1007	79
1016	77
1027	86
1033	86
1070	74
1081	71
1084	61
1098	79
1150	92
1160	59
1169	85
1172	86
1174	89
1176	90
1178	80
1184	80
1193	82
1196	81
1199	74
1207	80
1218	91
1223	69
1226	67
1227	65
1229	66

INDEX	TARGET_WINS
1241	84
1244	88
1246	75
1248	93
1249	88
1253	85
1261	85
1305	76
1314	77
1323	95
1328	80
1353	75
1363	78
1371	93
1372	80
1389	62
1393	71
1421	89
1431	72
1437	78
1442	75
1450	74
1463	80
1464	78
1470	87
1471	82
1484	86
1495	58
1507	65
1514	79
1526	64
1549	92
1552	72
1556	88
1564	73
1585	99
1586	104
1590	96
1591	101
1592	98
1603	95
1612	81
1634	84
1645	71
1647	83
1673	88
1674	86
1687	84
1688	94
1700	84
1708	78
1713	83

INDEX	TARGET_WINS
1717	68
1721	77
1730	78
1737	85
1748	86
1749	88
1763	90
1768	97
1778	97
1780	86
1782	78
1784	69
1794	104
1803	72
1804	81
1819	73
1832	76
1833	78
1844	69
1847	77
1854	92
1855	82
1857	83
1864	70
1865	80
1869	76
1880	90
1881	83
1882	83
1894	81
1896	82
1916	81
1918	74
1921	102
1926	86
1938	79
1979	65
1982	69
1987	81
1997	74
2004	87
2011	78
2015	80
2022	75
2025	70
2027	80
2031	73
2036	82
2066	76
2073	80
2087	76
2092	86

INDEX	TARGET_WINS
2125	77
2148	80
2162	88
2191	80
2203	86
2218	78
2221	75
2225	76
2232	79
2267	90
2291	69
2299	86
2317	92
2318	86
2353	82
2403	60
2411	85
2415	79
2424	83
2441	76
2464	83
2465	82
2472	73
2481	90
2487	50
2500	70
2501	77
2520	82
2521	81
2525	77

Part 1. Data Exploration

```
library(alr3)
library(car)
library(corrplot)
# library(fBasics)
library(knitr)
library(MASS)
```

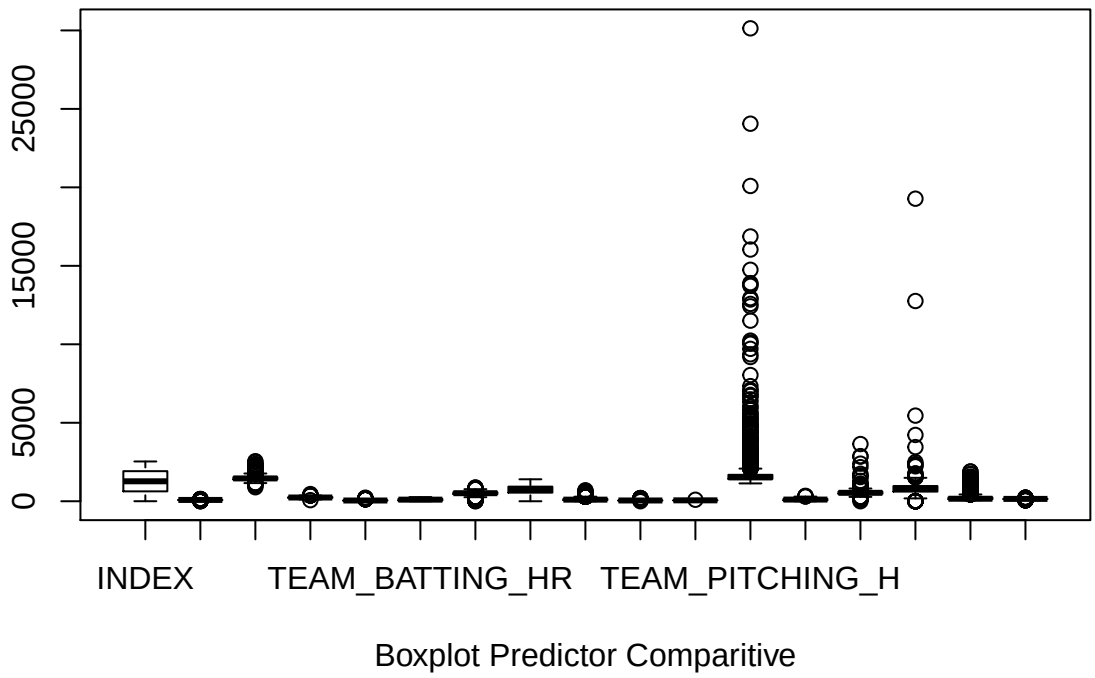
```
library(psych)
library(car)
library(corrplot)
```

Load and explore original data

```
mb_train <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/moneyball-tra
describe(mb_train)
```

```
##                   vars    n    mean      sd median trimmed    mad  min
## INDEX                1 2276 1268.46  736.35 1270.5 1268.57 952.57    1
## TARGET_WINS          2 2276   80.79   15.75   82.0   81.31  14.83    0
## TEAM_BATTING_H       3 2276 1469.27  144.59 1454.0 1459.04 114.16  891
## TEAM_BATTING_2B      4 2276  241.25   46.80  238.0  240.40  47.44   69
## TEAM_BATTING_3B      5 2276   55.25   27.94   47.0   52.18  23.72    0
## TEAM_BATTING_HR      6 2276   99.61   60.55  102.0   97.39  78.58    0
## TEAM_BATTING_BB      7 2276  501.56  122.67  512.0  512.18  94.89    0
## TEAM_BATTING_SO      8 2174  735.61  248.53  750.0  742.31 284.66    0
## TEAM_BASERUN_SB      9 2145  124.76   87.79  101.0  110.81  60.79    0
## TEAM_BASERUN_CS     10 1504   52.80   22.96   49.0   50.36  17.79    0
## TEAM_BATTING_HBP    11  191   59.36   12.97   58.0   58.86  11.86   29
## TEAM_PITCHING_H     12 2276 1779.21 1406.84 1518.0 1555.90 174.95 1137
## TEAM_PITCHING_HR    13 2276  105.70   61.30  107.0  103.16  74.13    0
## TEAM_PITCHING_BB    14 2276  553.01  166.36  536.5  542.62  98.59    0
## TEAM_PITCHING_SO    15 2174  817.73  553.09  813.5  796.93 257.23    0
## TEAM_FIELDING_E     16 2276  246.48  227.77  159.0  193.44  62.27   65
## TEAM_FIELDING_DP    17 1990  146.39   26.23  149.0  147.58  23.72   52
##                     max range  skew kurtosis    se
## INDEX              2535  2534  0.00    -1.22 15.43
## TARGET_WINS         146   146 -0.40     1.03  0.33
## TEAM_BATTING_H     2554  1663  1.57     7.28  3.03
## TEAM_BATTING_2B     458   389  0.22     0.01  0.98
## TEAM_BATTING_3B     223   223  1.11     1.50  0.59
## TEAM_BATTING_HR     264   264  0.19    -0.96  1.27
## TEAM_BATTING_BB     878   878 -1.03     2.18  2.57
## TEAM_BATTING_SO    1399  1399 -0.30    -0.32  5.33
## TEAM_BASERUN_SB     697   697  1.97     5.49  1.90
## TEAM_BASERUN_CS     201   201  1.98     7.62  0.59
## TEAM_BATTING_HBP     95    66  0.32    -0.11  0.94
## TEAM_PITCHING_H   30132 28995 10.33   141.84 29.49
## TEAM_PITCHING_HR    343   343  0.29    -0.60  1.28
## TEAM_PITCHING_BB   3645  3645  6.74    96.97  3.49
## TEAM_PITCHING_SO  19278 19278 22.17   671.19 11.86
## TEAM_FIELDING_E    1898  1833  2.99    10.97  4.77
## TEAM_FIELDING_DP    228   176 -0.39     0.18  0.59
```

```r
boxplot(mb_train, xlab="Boxplot Predictor Comparitive")
```



Look for correlations among values

```
m <- na.omit(mb_train)
cor(m)
```

```
##                       INDEX TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B
## INDEX            1.00000000 -0.04895047    -0.09493748     -0.01841285
## TARGET_WINS     -0.04895047  1.00000000     0.46994665      0.31298400
## TEAM_BATTING_H  -0.09493748  0.46994665     1.00000000      0.56177286
## TEAM_BATTING_2B -0.01841285  0.31298400     0.56177286      1.00000000
## TEAM_BATTING_3B -0.04639454 -0.12434586     0.21391883      0.04203441
## TEAM_BATTING_HR -0.09532668  0.42241683     0.39627593      0.25099045
## TEAM_BATTING_BB  0.04702541  0.46868793     0.19735234      0.19749256
## TEAM_BATTING_SO  0.04132959 -0.22889273    -0.34174328     -0.06415123
## TEAM_BASERUN_SB -0.03159666  0.01483639     0.07167495     -0.18768279
## TEAM_BASERUN_CS -0.07692325 -0.17875598    -0.09377545     -0.20413884
## TEAM_BATTING_HBP  0.07719303  0.07350424    -0.02911218      0.04608475
## TEAM_PITCHING_H -0.08865725  0.47123431     0.99919269      0.56045355
## TEAM_PITCHING_HR -0.09361594  0.42246683     0.39495630      0.24999875
## TEAM_PITCHING_BB  0.04958287  0.46839882     0.19529071      0.19592157
## TEAM_PITCHING_SO  0.04466127 -0.22936481    -0.34445001     -0.06616615
## TEAM_FIELDING_E -0.02004841 -0.38668800    -0.25381638     -0.19427027
## TEAM_FIELDING_DP  0.13168916 -0.19586601     0.01776946     -0.02488808
##                  TEAM_BATTING_3B TEAM_BATTING_HR TEAM_BATTING_BB
## INDEX                -0.04639454     -0.09532668      0.04702541
## TARGET_WINS          -0.12434586      0.42241683      0.46868793
```

```
## TEAM_BATTING_H        0.21391883      0.39627593      0.19735234
## TEAM_BATTING_2B       0.04203441      0.25099045      0.19749256
## TEAM_BATTING_3B       1.00000000     -0.21879927     -0.20584392
## TEAM_BATTING_HR      -0.21879927      1.00000000      0.45638161
## TEAM_BATTING_BB      -0.20584392      0.45638161      1.00000000
## TEAM_BATTING_SO      -0.19291841      0.21045444      0.21833871
## TEAM_BASERUN_SB       0.16946086     -0.19021893     -0.08806123
## TEAM_BASERUN_CS       0.23213978     -0.27579838     -0.20878051
## TEAM_BATTING_HBP     -0.17424715      0.10618116      0.04746007
## TEAM_PITCHING_H       0.21250322      0.39549390      0.19848687
## TEAM_PITCHING_HR     -0.21973263      0.99993259      0.45659283
## TEAM_PITCHING_BB     -0.20675383      0.45542468      0.99988140
## TEAM_PITCHING_SO     -0.19386654      0.20829574      0.21793253
## TEAM_FIELDING_E      -0.06513145      0.01567397     -0.07847126
## TEAM_FIELDING_DP      0.13314758     -0.06182222     -0.07929078
##                  TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## INDEX                 0.04132959     -0.03159666    -0.076923249
## TARGET_WINS          -0.22889273      0.01483639    -0.178755979
## TEAM_BATTING_H       -0.34174328      0.07167495    -0.093775445
## TEAM_BATTING_2B      -0.06415123     -0.18768279    -0.204138837
## TEAM_BATTING_3B      -0.19291841      0.16946086     0.232139777
## TEAM_BATTING_HR       0.21045444     -0.19021893    -0.275798375
## TEAM_BATTING_BB       0.21833871     -0.08806123    -0.208780510
## TEAM_BATTING_SO       1.00000000     -0.07475974    -0.056130355
## TEAM_BASERUN_SB      -0.07475974      1.00000000     0.624737808
## TEAM_BASERUN_CS      -0.05613035      0.62473781     1.000000000
## TEAM_BATTING_HBP      0.22094219     -0.06400498    -0.070513896
## TEAM_PITCHING_H      -0.34145321      0.07395373    -0.092977893
## TEAM_PITCHING_HR      0.21111617     -0.18948057    -0.275471495
## TEAM_PITCHING_BB      0.21895783     -0.08741902    -0.208470154
## TEAM_PITCHING_SO      0.99976835     -0.07351325    -0.055308336
## TEAM_FIELDING_E       0.30814540      0.04292341     0.207701189
## TEAM_FIELDING_DP     -0.12319072     -0.13023054    -0.006764233
##                  TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
## INDEX                  0.07719303     -0.08865725      -0.09361594
## TARGET_WINS            0.07350424      0.47123431       0.42246683
## TEAM_BATTING_H        -0.02911218      0.99919269       0.39495630
## TEAM_BATTING_2B        0.04608475      0.56045355       0.24999875
## TEAM_BATTING_3B       -0.17424715      0.21250322      -0.21973263
## TEAM_BATTING_HR        0.10618116      0.39549390       0.99993259
## TEAM_BATTING_BB        0.04746007      0.19848687       0.45659283
## TEAM_BATTING_SO        0.22094219     -0.34145321       0.21111617
## TEAM_BASERUN_SB       -0.06400498      0.07395373      -0.18948057
## TEAM_BASERUN_CS       -0.07051390     -0.09297789      -0.27547150
## TEAM_BATTING_HBP       1.00000000     -0.02769699       0.10675878
## TEAM_PITCHING_H       -0.02769699      1.00000000       0.39463199
## TEAM_PITCHING_HR       0.10675878      0.39463199       1.00000000
## TEAM_PITCHING_BB       0.04785137      0.19703302       0.45580983
## TEAM_PITCHING_SO       0.22157375     -0.34330646       0.20920115
## TEAM_FIELDING_E        0.04178971     -0.25073028       0.01689330
## TEAM_FIELDING_DP      -0.07120824      0.01416807      -0.06292475
##                  TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E
## INDEX                  0.04958287       0.04466127     -0.02004841
## TARGET_WINS            0.46839882      -0.22936481     -0.38668800
```

```
## TEAM_BATTING_H         0.19529071      -0.34445001     -0.25381638
## TEAM_BATTING_2B        0.19592157      -0.06616615     -0.19427027
## TEAM_BATTING_3B       -0.20675383      -0.19386654     -0.06513145
## TEAM_BATTING_HR        0.45542468       0.20829574      0.01567397
## TEAM_BATTING_BB        0.99988140       0.21793253     -0.07847126
## TEAM_BATTING_SO        0.21895783       0.99976835      0.30814540
## TEAM_BASERUN_SB       -0.08741902      -0.07351325      0.04292341
## TEAM_BASERUN_CS       -0.20847015      -0.05530834      0.20770119
## TEAM_BATTING_HBP       0.04785137       0.22157375      0.04178971
## TEAM_PITCHING_H        0.19703302      -0.34330646     -0.25073028
## TEAM_PITCHING_HR       0.45580983       0.20920115      0.01689330
## TEAM_PITCHING_BB       1.00000000       0.21887700     -0.07692315
## TEAM_PITCHING_SO       0.21887700       1.00000000      0.31008407
## TEAM_FIELDING_E       -0.07692315       0.31008407      1.00000000
## TEAM_FIELDING_DP      -0.08040645      -0.12492321      0.04020581
##                   TEAM_FIELDING_DP
## INDEX                  0.131689160
## TARGET_WINS           -0.195866006
## TEAM_BATTING_H         0.017769456
## TEAM_BATTING_2B       -0.024888081
## TEAM_BATTING_3B        0.133147578
## TEAM_BATTING_HR       -0.061822219
## TEAM_BATTING_BB       -0.079290775
## TEAM_BATTING_SO       -0.123190715
## TEAM_BASERUN_SB       -0.130230537
## TEAM_BASERUN_CS       -0.006764233
## TEAM_BATTING_HBP      -0.071208241
## TEAM_PITCHING_H        0.014168073
## TEAM_PITCHING_HR      -0.062924751
## TEAM_PITCHING_BB      -0.080406452
## TEAM_PITCHING_SO      -0.124923213
## TEAM_FIELDING_E        0.040205814
## TEAM_FIELDING_DP       1.000000000
```

Part 2. Data Preparation

New Variable Creation

Creating a new column for batting singles and eliminating hits for batting

```
mb_e <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/moneyball-traini
#eliminate index column
mb_e1 <- mb_e[,-1]

#add singles column for hitting
mb_e1$TEAM_BATTING_1B <- as.numeric(mb_e1$TEAM_BATTING_H-mb_e1$TEAM_BATTING_2B-mb_e1$TEAM_BATTING_3B-mb
mb_e1 <- mb_e1[,-2]
mb_e1 <- as.data.frame(mb_e1)
```

Building a regression model and filling in NA's for SB.

Note: This approach is suggested in LMAR p. 201. "A more sophisticated alternative to mean imputation is to use regression methods to predict the missing values of the covariates."

```
SB <- lm(data=mb_e1, TEAM_BASERUN_SB~.)
summary(SB)

#eliminate CS as there are no blank SB's with a value for CS + eliminate pitching, wins and fielding va
SB1 <- lm(data=mb_e1, TEAM_BASERUN_SB~TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTIN
summary(SB1)

#eliminate singles
SB2 <- lm(data=mb_e1, TEAM_BASERUN_SB~TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTIN
summary(SB2)

#fill in NA for SB
mb_e2 <- mb_e1
mb_e2$TEAM_BASERUN_SB[is.na(mb_e2$TEAM_BASERUN_SB)]<-round(79.36805-0.19419*mb_e2$TEAM_BATTING_2B+1.416
```

Building a linear regression model and filling in NA's for CS

```
CS <- lm(data=mb_e2, TEAM_BASERUN_CS~.)
summary(CS)

#eliminate wins, pitching and fielding
CS1 <- lm(data=mb_e2, TEAM_BASERUN_CS~TEAM_BASERUN_SB +TEAM_BATTING_1B +TEAM_BATTING_2B + TEAM_BATTING_
summary(CS1)

#eliminate walks
CS2 <- lm(data=mb_e2, TEAM_BASERUN_CS~TEAM_BASERUN_SB +TEAM_BATTING_2B + TEAM_BATTING_3B +TEAM_BATTING_
summary(CS2)

#fill in NA for CS
mb_e3 <- mb_e2
mb_e3$TEAM_BASERUN_CS[is.na(mb_e3$TEAM_BASERUN_CS)]<-round(49.356793+0.322543*mb_e3$TEAM_BASERUN_SB-0.0
```

Building a regression model and filling in NA's for batting SO's

```
BSO <- lm(data=mb_e3, TEAM_BATTING_SO~.)
summary(BSO)

#eliminate fielding and wins and baserunning and HBP and pitching SO's as it contains similar blanks
BSO1 <- lm(data=mb_e3, TEAM_BATTING_SO~TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTI
summary(BSO1)

#eliminate pitching HR's
BSO2 <- lm(data=mb_e3, TEAM_BATTING_SO~TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTI
summary(BSO2)

#fill in NA for batting SO
mb_e4 <- mb_e3
mb_e4$TEAM_BATTING_SO[is.na(mb_e4$TEAM_BATTING_SO)]<-round(1605-0.8434*mb_e4$TEAM_BATTING_1B+0.2832*mb_
```

Building a regression model and filling in NA's for pitching SO's

```
PSO <- lm(data=mb_e4, TEAM_PITCHING_SO~.)
summary(PSO)

#eliminate wins, fielding, baserunning and HBP
PSO1 <- lm(data=mb_e4, TEAM_PITCHING_SO~TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATT
summary(PSO1)

#eliminate batting 3B's
PSO2 <- lm(data=mb_e4, TEAM_PITCHING_SO~TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_HR + TEAM_BATT
summary(PSO2)

#replace NA with values for pitching SO
mb_e5 <- mb_e4
mb_e5$TEAM_PITCHING_SO[is.na(mb_e5$TEAM_PITCHING_SO)]<-round(4422.87422-0.46455*mb_e5$TEAM_BATTING_1B +
```

Building a linear regression and filling in NA's for DP

```
#build a regression model for DP's

DP <- lm(data=mb_e5, TEAM_FIELDING_DP~.)
summary(DP)

#eliminate wins, hitting, HBP

DP1 <- lm(data=mb_e5, TEAM_FIELDING_DP~ TEAM_BASERUN_SB+TEAM_BASERUN_CS+TEAM_PITCHING_H+TEAM_PITCHING_H
summary(DP1)

#eliminate pitching hits allowed
DP2 <- lm(data=mb_e5, TEAM_FIELDING_DP~ TEAM_BASERUN_SB+TEAM_BASERUN_CS+TEAM_PITCHING_HR+TEAM_PITCHING_
summary(DP2)

#eliminate CS as it makes no sense
DP3 <- lm(data=mb_e5, TEAM_FIELDING_DP~ TEAM_BASERUN_SB+TEAM_PITCHING_HR+TEAM_PITCHING_BB+TEAM_PITCHING_
summary(DP3)

#replace NA with values for DP
mb_e6 <- mb_e5
mb_e6$TEAM_FIELDING_DP[is.na(mb_e6$TEAM_FIELDING_DP)]<- round(158.8-0.1235*mb_e6$TEAM_BASERUN_SB +0.033

summary(mb_e6)
#only NA's left are HBP
```

Data Imputation

```
# read EVALUATION data set
eval_data <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-1/moneyball-eval

# read training data set
mb_e <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-1/moneyball-training
```

```
#eliminate index column
# mb_e1 <- mb_e[,-1]
mb_e1 <- mb_e

#####Creating a new column for batting singles and eliminating hits for batting

#add singles column for hitting
mb_e1$TEAM_BATTING_1B <- as.numeric(mb_e1$TEAM_BATTING_H-mb_e1$TEAM_BATTING_2B-mb_e1$TEAM_BATTING_3B-mb
mb_e1 <- mb_e1[,-3]
mb_e1 <- as.data.frame(mb_e1)

eval_data$TEAM_BATTING_1B <- as.numeric(eval_data$TEAM_BATTING_H - eval_data$TEAM_BATTING_2B - eval_data

# HITS is in second column in eval data
eval_data <- eval_data[,-2]

# ADD A DUMMY COLUMN TO EVAL DATA FOR TARGET WINS
eval_data$TARGET_WINS <- 0
```

Eliminate HBP, CS, and pitching HR's.

```
mb <- mb_e1[,-c(9,10,12)]
# summary(mb)

eval_data <- eval_data[,-c(8,9,11)]
# summary(eval_data)
```

Build model for batting SO using Gelman approach

```
#take out double plays + pitching SO + SB as data set is incomplete + Wins as they are not present in t

BSO.1 <- lm(data=mb, TEAM_BATTING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_SO -TEAM_BASERUN_SB -TA
summary(BSO.1)

#eliminate doubles
BSO.2 <- lm(data=mb, TEAM_BATTING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_SO -TEAM_BASERUN_SB - TA
summary(BSO.2)
vif(BSO.2)

# vif says remove TEAM_PITCHING_BB
BSO.3 <- lm(data=mb, TEAM_BATTING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_SO -TEAM_BASERUN_SB - TA
summary(BSO.3)

# pvals say remove PITCHING_H
BSO.4 <- lm(data=mb, TEAM_BATTING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_SO -TEAM_BASERUN_SB - TA
summary(BSO.4)

vif(BSO.4)

##All p-values are low with a 686.8 F-statistic and adjusted R squared of 0.7236
#take a look
```

```
par(mfrow=c(2,2))
plot(BSO.2)

# ---------------------------------------
# function definition for impute function
impute <- function (a, a.impute){
  ifelse (is.na(a), a.impute,a)
}
# ---------------------------------------

#prediction function
pred.BSO <- round(predict(BSO.4, mb))
BSO.imp <- impute(mb$TEAM_BATTING_SO, pred.BSO)

# impute the evaluation data
pred_eval.BSO <- round(predict(BSO.4, eval_data))
eval.BSO.imp <- impute(eval_data$TEAM_BATTING_SO, pred_eval.BSO)

#####################################################
# Jims added code for diagnostics of imputation

# first, check summaries to ensure similar values
summary(mb$TEAM_BATTING_SO)
summary(BSO.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(mb$TEAM_BATTING_SO, breaks = 200)
hist(BSO.imp, breaks = 200)

# ------------------ eval data checks -------------------------
# first, check summaries to ensure similar values
summary(eval_data$TEAM_BATTING_SO)
summary(eval.BSO.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(eval_data$TEAM_BATTING_SO, breaks = 30)
hist(eval.BSO.imp, breaks = 30)
#####################################################

# update dataframes with imputed values
mb1 <- mb
mb1$TEAM_BATTING_SO <- BSO.imp

eval_data.1 <- eval_data
eval_data.1$TEAM_BATTING_SO <- eval.BSO.imp
```

Build model for Pitching SO

```
#take out double plays + SB as data set is incomplete and wins as they are not present in evaluation da

PSO.1 <- lm(data=mb1, TEAM_PITCHING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_BASERUN_SB - TARGET_WINS)
```

```r
summary(PSO.1)

vif(PSO.1)
# vif says remove TEAM_PITCHING_BB

PSO.2 <- lm(data=mb1, TEAM_PITCHING_SO~. - INDEX -TEAM_FIELDING_DP -TEAM_BASERUN_SB - TARGET_WINS - TEA
summary(PSO.2)

vif(PSO.2)

#all low P value and F statistic of 4719 with adj R squared of 0.9952
#take a look
par(mfrow=c(2,2))
plot(PSO.2)

#place back in the data base with imputed data for SO's
pred.PSO <- round(predict(PSO.2, mb1))
PSO.imp <- impute(mb1$TEAM_PITCHING_SO, pred.PSO)

# impute the evaluation data
pred_eval.PSO <- round(predict(PSO.2, eval_data.1))
eval.PSO.imp <- impute(eval_data.1$TEAM_PITCHING_SO, pred_eval.PSO)

########################################################
# Jims added code for diagnostics of imputation

# first, check summaries to ensure similar values
summary(mb1$TEAM_PITCHING_SO)
summary(PSO.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(mb1$TEAM_PITCHING_SO, breaks = 200)
hist(PSO.imp, breaks = 200)

# ------------------ eval data checks -------------------------
# first, check summaries to ensure similar values
summary(eval_data.1$TEAM_PITCHING_SO)
summary(eval.PSO.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(eval_data.1$TEAM_PITCHING_SO, breaks = 30)
hist(eval.PSO.imp, breaks = 30)

########################################################

# update dataframes with imputed values

mb2 <- mb1
mb2$TEAM_PITCHING_SO <- PSO.imp

eval_data.2 <- eval_data.1
```

```
eval_data.2$TEAM_PITCHING_SO <- eval.PSO.imp
```

Build model for SB

```
#Take out DP as incomplete data and target wins
SB.1 <- lm(data=mb2, TEAM_BASERUN_SB~. -INDEX -TEAM_FIELDING_DP - TARGET_WINS)
summary(SB.1)

#eliminate pitching BB's
SB.2 <- lm(data=mb2, TEAM_BASERUN_SB~. -INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB - TARGET_WINS)
summary(SB.2)

#eliminate singles
SB.3 <- lm(data=mb2, TEAM_BASERUN_SB~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB -TEAM_BATTING_1B - TA
summary(SB.3)

#simplify the model by taking out pitching
SB.4 <- lm(data=mb2, TEAM_BASERUN_SB~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB -TEAM_BATTING_1B - TA
summary(SB.4)

#add singles back in
SB.5 <- lm(data=mb2, TEAM_BASERUN_SB~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB - TARGET_WINS - TEAM_
summary(SB.5)

#eliminate doubles
SB.6 <- lm(data=mb2, TEAM_BASERUN_SB~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB - TARGET_WINS - TEAM_
summary(SB.6)

#eliminate walks
SB.7 <- lm(data=mb2, TEAM_BASERUN_SB~. - INDEX -TEAM_FIELDING_DP -TEAM_PITCHING_BB - TARGET_WINS - TEAM_
summary(SB.7)

#all low P value and F statistic of 202.9 with adj R squared of 0.3427
#take a look
par(mfrow=c(2,2))
plot(SB.7)

#place back in the data base with imputed data for SB's
pred.SB <- round(predict(SB.7, mb2))
SB.imp <- impute(mb2$TEAM_BASERUN_SB, pred.SB)

# impute the evaluation data
pred_eval.SB <- round(predict(SB.7, eval_data.2))
eval.SB.imp <- impute(eval_data.2$TEAM_BASERUN_SB, pred_eval.SB)

#####################################################
# Jims added code for diagnostics of imputation

# first, check summaries to ensure similar values
summary(mb2$TEAM_BASERUN_SB)
summary(SB.imp)

# now plot side-by-side histograms to check similarity of distributions
```

```
par(mfrow = c(2,2))
hist(mb2$TEAM_BASERUN_SB, breaks = 200)
hist(SB.imp, breaks = 200)

# ------------------ eval data checks ------------------------
# first, check summaries to ensure similar values
summary(eval_data.2$TEAM_BASERUN_SB)
summary(eval.SB.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(eval_data.2$TEAM_BASERUN_SB, breaks = 30)
hist(eval.SB.imp, breaks = 30)
#########################################################

# update dataframes with imputed values
mb3 <- mb2
mb3$TEAM_BASERUN_SB <- SB.imp

eval_data.3 <- eval_data.2
eval_data.3$TEAM_BASERUN_SB <- eval.SB.imp
```

Build model to replace DP

```
#remove target wins
DP.1 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS)
summary(DP.1)

#remove batting 2B's
DP.2 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS - TEAM_BATTING_2B)
summary(DP.2)
# results show that EVERYTHING ELSE is statistically signficant, so:

# run vif to check for collinearity
vif(DP.2)
# results show TEAM_BATTING_SO should be removed

# remove TEAM_BATTING_SO
DP.3 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS -TEAM_BATTING_2B - TEAM_BATTING_SO)
summary(DP.3)
# p-value says remove TEAM_PITCHING_SO;


# remove TEAM_PITCHING_SO
DP.4 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TEAM_BATTING_2B -TARGET_WINS -TEAM_BATTING_2B - TEAM_BA
summary(DP.4)
vif(DP.4)
# P values and vif both indicate remove TEAM_PITCHING_BB

# remove TEAM_PITCHING_BB
DP.5 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS -TEAM_BATTING_2B - TEAM_BATTING_SO - TEAM_
summary(DP.5)
vif(DP.5)
```

```
# vif says remove TEAM_FIELDING_E; p-values all < .05 so remove TEAM_FIELDING_E

DP.6 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS -TEAM_BATTING_2B - TEAM_BATTING_SO - TEAM_
summary(DP.6)
vif(DP.6)
# now no collinearity but p-values say remove TEAM_PITCHING_H

DP.7 <- lm(data=mb3, TEAM_FIELDING_DP~. - INDEX -TARGET_WINS -TEAM_BATTING_2B - TEAM_BATTING_SO - TEAM_
summary(DP.7)
vif(DP.7)
# no collinearity, all p-values < .05 so stop


#all low P value and F statistic of 255.8 with adj R squared of 0.3904
#take a look
par(mfrow=c(2,2))
plot(DP.7)


#place back in the data base with imputed data for SB's
# NOTE: Changed DP.4 to DP.7 here
pred.DP <- round(predict(DP.7, mb3))
DP.imp <- impute(mb3$TEAM_FIELDING_DP, pred.DP)

# impute the evaluation data
pred_eval.DP <- round(predict(DP.7, eval_data.3))
eval.DP.imp <- impute(eval_data.3$TEAM_FIELDING_DP, pred_eval.DP)

#########################################################
# Jims added code for diagnostics of imputation

# first, check summaries to ensure similar values
summary(mb3$TEAM_FIELDING_DP)
summary(DP.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(mb3$TEAM_FIELDING_DP, breaks = 200)
hist(DP.imp, breaks = 200)

# ------------------- eval data checks -------------------------
# first, check summaries to ensure similar values
summary(eval_data.3$TEAM_FIELDING_DP)
summary(eval.DP.imp)

# now plot side-by-side histograms to check similarity of distributions
par(mfrow = c(2,2))
hist(eval_data.3$TEAM_FIELDING_DP, breaks = 30)
hist(eval.DP.imp, breaks = 30)
#########################################################

# update data frames with imputed values
mb4 <- mb3
```

```
mb4$TEAM_FIELDING_DP <- DP.imp

eval_data.4 <- eval_data.3
eval_data.4$TEAM_FIELDING_DP <- eval.DP.imp
```

Eliminate unhistorical outliers - DO THIS FOR THE EVAL DATA AS WELL

```
# check rowcount before removal of outliers
nrow(mb4)
nrow(eval_data.4)

############## TEAM PITCHING_SO ##############################
#most pitching SO's is 1450.  So delete all records with more than 1450 pitching SO's
mb5 <- mb4

# fixed error in this line: dataframe in 'which' call was mb1 so changed to mb5
mb5 <- mb5[which(mb5$TEAM_PITCHING_SO < 1450),]

# eval_data.4 <- eval_data.4[which(eval_data.4$TEAM_PITCHING_SO < 1450),]

# check rowcount
nrow(mb5)
nrow(eval_data.4)

############ TEAM_PITCHING_H #################################
#most ever hits by a team is 1730.  So delete all pitching hits >3000 to be conservative with the media
mb6 <- mb5
mb6 <- mb6[which(mb6$TEAM_PITCHING_H < 3001),]

# eval_data.4 <- eval_data.4[which(eval_data.4$TEAM_PITCHING_H < 3001),]

# check rowcount
nrow(mb6)
nrow(eval_data.4)

############ TEAM_FIELDING_E #################################
#most ever errors by a team is 639 by 1883 Philadelphia.  Prorating to 162 games gives a value of 1046.
mb7 <- mb6
# mb7 <- mb7[which(mb7$TEAM_FIELDING_E < 1047),]

eval_data.4 <- eval_data.4[which(eval_data.4$TEAM_FIELDING_E < 1047),]

# ------------------------------------------------------------------
# ------------------------------------------------------------------

# check rowcount: result is 2172 => removed total of 104 rows
nrow(mb7)
nrow(eval_data.4)

dim(mb)-dim(mb7)

#Create box plot and summary data to compare with original data
```

```
describe(mb7)

boxplot(mb7, xlab="Boxplot Predictor Comparitive")

#we removed 104 rows total due to outliers in TRAINING data set.

# now renumber rows of dataframe so that there are no gaps in row numbers
rownames(mb7) <- 1:nrow(mb7)
rownames(eval_data.4) <- 1:nrow(eval_data.4)

# drop INDEX column from training set
# mb7 <- mb7[,-1]

#Create box plot and summary data to compare with original data

describe(mb7)

boxplot(mb7, xlab="Boxplot Predictor Comparitive")
# now drop dummy column from evaluation data
# eval_data.4 <- eval_data.4[,-14]


# create CSV files containing updated data sets
#SMK commented out for now-doesn't work on a mac
#write.csv(mb7, file = "C:/SQLData/621-HW1-Clean-Data.csv", row.names = FALSE, col.names = TRUE)

#write.csv(eval_data.4, file = "C:/SQLData/621-HW1-Clean-EvalData-.csv", row.names = FALSE, col.names =
```

Part 3. Build Models

Model 1: General Model Using Backward Selection

```
library(car)

mb_clean <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clea
```

```
library(alr3)

# code for finding power transforms for skewed variables
hist(mb_clean$TEAM_PITCHING_BB, breaks = 200)

hist(mb_clean$TEAM_PITCHING_H, breaks = 200)

hist(mb_clean$TEAM_FIELDING_E, breaks = 200)

################ TEAM_PITCHING_BB ################################

summary( powerTransform( cbind(TEAM_PITCHING_BB) ~ 1, mb_clean))
```

```
# TEAM_PITCHING_BB = log transform
PBB.T <- log(mb_clean$TEAM_PITCHING_BB)
hist(PBB.T, breaks = 200)

# ---------------------
################ PITCHING_H ###################################
summary( powerTransform( cbind(TEAM_PITCHING_H) ~ 1, mb_clean))
# TEAM_PITCHING_H = 1/y^3

# WORKS!
PH.T <- 1/(mb_clean$TEAM_PITCHING_H ^ 3)
hist(PH.T, breaks = 200)


############### FIELDING_E ######################################
# -------------------------
summary( powerTransform( cbind(TEAM_FIELDING_E) ~ 1, mb_clean))
# TEAM_FIELDING_E = -1 = 1/y or try log

# WORKS!
FE.T <- 1/mb_clean$TEAM_FIELDING_E
hist(FE.T, breaks = 200)

# Now load transformed values into data set

mb.t <- mb_clean

mb.t$TEAM_PITCHING_BB <- PBB.T
mb.t$TEAM_PITCHING_H <- PH.T
mb.t$TEAM_FIELDING_E <- FE.T
```

Build Models

Model using all remaining variables as a starting point

Yields r^2= 0.3347, Adj r^2 = 0.3322, F = 136

```
# keep the clean data set pure
mb <- mb_clean

# use p-value elimination
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.179  -7.684   0.193   7.338  63.241
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      48.966514   5.630361   8.697  < 2e-16 ***
## TEAM_BATTING_2B  -0.042963   0.009058  -4.743 2.24e-06 ***
## TEAM_BATTING_3B   0.119709   0.017111   6.996 3.50e-12 ***
## TEAM_BATTING_HR   0.076628   0.010255   7.473 1.14e-13 ***
## TEAM_BATTING_BB   0.179272   0.016504  10.863  < 2e-16 ***
## TEAM_BATTING_SO  -0.047387   0.011793  -4.018 6.06e-05 ***
## TEAM_BASERUN_SB   0.067351   0.004963  13.570  < 2e-16 ***
## TEAM_PITCHING_H   0.038622   0.004966   7.778 1.13e-14 ***
## TEAM_PITCHING_BB -0.139535   0.014311  -9.750  < 2e-16 ***
## TEAM_PITCHING_SO  0.029867   0.011011   2.712  0.00673 **
## TEAM_FIELDING_E  -0.076068   0.003894 -19.535  < 2e-16 ***
## TEAM_FIELDING_DP -0.118907   0.013074  -9.095  < 2e-16 ***
## TEAM_BATTING_1B  -0.011641   0.006975  -1.669  0.09526 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.62 on 2159 degrees of freedom
## Multiple R-squared:  0.3658, Adjusted R-squared:  0.3623
## F-statistic: 103.8 on 12 and 2159 DF,  p-value: < 2.2e-16
```

```
# p-value indicates remove TEAM_BATTING_1B

# --------------------
# remove TEAM_BATTING_1B
model.2 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.306  -7.736   0.089   7.278  61.942
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      43.726614   4.675707   9.352  < 2e-16 ***
## TEAM_BATTING_2B  -0.037030   0.008334  -4.443 9.32e-06 ***
## TEAM_BATTING_3B   0.123030   0.017002   7.236 6.37e-13 ***
## TEAM_BATTING_HR   0.084581   0.009084   9.311  < 2e-16 ***
## TEAM_BATTING_BB   0.175496   0.016355  10.731  < 2e-16 ***
## TEAM_BATTING_SO  -0.057558   0.010100  -5.699 1.37e-08 ***
## TEAM_BASERUN_SB   0.066640   0.004947  13.471  < 2e-16 ***
## TEAM_PITCHING_H   0.032020   0.003003  10.663  < 2e-16 ***
## TEAM_PITCHING_BB -0.136552   0.014205  -9.613  < 2e-16 ***
## TEAM_PITCHING_SO  0.040806   0.008851   4.610 4.26e-06 ***
## TEAM_FIELDING_E  -0.074297   0.003748 -19.823  < 2e-16 ***
## TEAM_FIELDING_DP -0.120293   0.013053  -9.216  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.62 on 2160 degrees of freedom
## Multiple R-squared:  0.365,  Adjusted R-squared:  0.3618
## F-statistic: 112.9 on 11 and 2160 DF,  p-value: < 2.2e-16
```

```
# p-values are OK so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.319929         3.359689         4.643042        43.103947
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##        83.755735         3.378362         9.471591        37.110476
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        62.742239         4.973211         2.082577
```

```
# vif says remove TEAM_BATTING_SO or PITCHING_SO, so remove PITCHING_SO per other models

# --------------------
#eliminate TEAM_PITCHING_SO
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.949  -7.717   0.285   7.601  63.510
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      38.652404   4.565554   8.466  < 2e-16 ***
## TEAM_BATTING_2B  -0.042295   0.008294  -5.099 3.71e-07 ***
## TEAM_BATTING_3B   0.107468   0.016741   6.419 1.68e-10 ***
## TEAM_BATTING_HR   0.077621   0.009000   8.625  < 2e-16 ***
## TEAM_BATTING_BB   0.123390   0.011876  10.390  < 2e-16 ***
## TEAM_BATTING_SO  -0.012083   0.002182  -5.538 3.43e-08 ***
## TEAM_BASERUN_SB   0.061650   0.004850  12.713  < 2e-16 ***
## TEAM_PITCHING_H   0.036828   0.002829  13.017  < 2e-16 ***
## TEAM_PITCHING_BB -0.090392   0.010123  -8.929  < 2e-16 ***
## TEAM_FIELDING_E  -0.071010   0.003697 -19.208  < 2e-16 ***
## TEAM_FIELDING_DP -0.119234   0.013112  -9.093  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.67 on 2161 degrees of freedom
## Multiple R-squared:  0.3588, Adjusted R-squared:  0.3558
## F-statistic: 120.9 on 10 and 2161 DF,  p-value: < 2.2e-16
```

```
vif(model.3)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.276374         3.227243         4.514804        22.517049
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##         3.871698         3.216671         8.329160        18.672715
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##         4.793258         2.081932
```

```
# vif says remove TEAM_BATTING_BB or PITCHINNG_BB so go with PITCHING_BB

# --------------------
#eliminate TEAM_PITCHING_BB
model.4 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.780  -8.035   0.320   7.776  69.832
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      62.579012   3.763135  16.629  < 2e-16 ***
## TEAM_BATTING_2B  -0.008694   0.007525  -1.155    0.248
## TEAM_BATTING_3B   0.134601   0.016760   8.031 1.57e-15 ***
## TEAM_BATTING_HR   0.103304   0.008681  11.899  < 2e-16 ***
## TEAM_BATTING_BB   0.020938   0.003117   6.716 2.37e-11 ***
## TEAM_BATTING_SO  -0.016911   0.002152  -7.859 6.07e-15 ***
## TEAM_BASERUN_SB   0.063121   0.004934  12.793  < 2e-16 ***
## TEAM_PITCHING_H   0.016891   0.001769   9.549  < 2e-16 ***
## TEAM_FIELDING_E  -0.069910   0.003761 -18.586  < 2e-16 ***
## TEAM_FIELDING_DP -0.108598   0.013294  -8.169 5.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.89 on 2162 degrees of freedom
## Multiple R-squared:  0.3351, Adjusted R-squared:  0.3323
## F-statistic: 121.1 on 9 and 2162 DF,  p-value: < 2.2e-16
```

```
# --------------------
# eliminate TEAM_BATTING_2B
model.5 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO - TEAM_PITCHING_BB -
summary(model.5)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB - TEAM_BATTING_2B, data = mb)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -42.412  -8.089   0.329   7.776  68.696
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      62.392894   3.759976  16.594  < 2e-16 ***
## TEAM_BATTING_3B   0.132616   0.016673   7.954 2.89e-15 ***
## TEAM_BATTING_HR   0.101500   0.008541  11.884  < 2e-16 ***
## TEAM_BATTING_BB   0.020574   0.003102   6.633 4.14e-11 ***
## TEAM_BATTING_SO  -0.017105   0.002145  -7.973 2.49e-15 ***
## TEAM_BASERUN_SB   0.063188   0.004934  12.806  < 2e-16 ***
## TEAM_PITCHING_H   0.015911   0.001553  10.248  < 2e-16 ***
## TEAM_FIELDING_E  -0.068573   0.003579 -19.158  < 2e-16 ***
## TEAM_FIELDING_DP -0.108871   0.013293  -8.190 4.40e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.89 on 2163 degrees of freedom
## Multiple R-squared:  0.3347, Adjusted R-squared:  0.3322
## F-statistic:   136 on 8 and 2163 DF,  p-value: < 2.2e-16
```

```
# p-values < .05 so check for collinearity
vif(model.5)
```

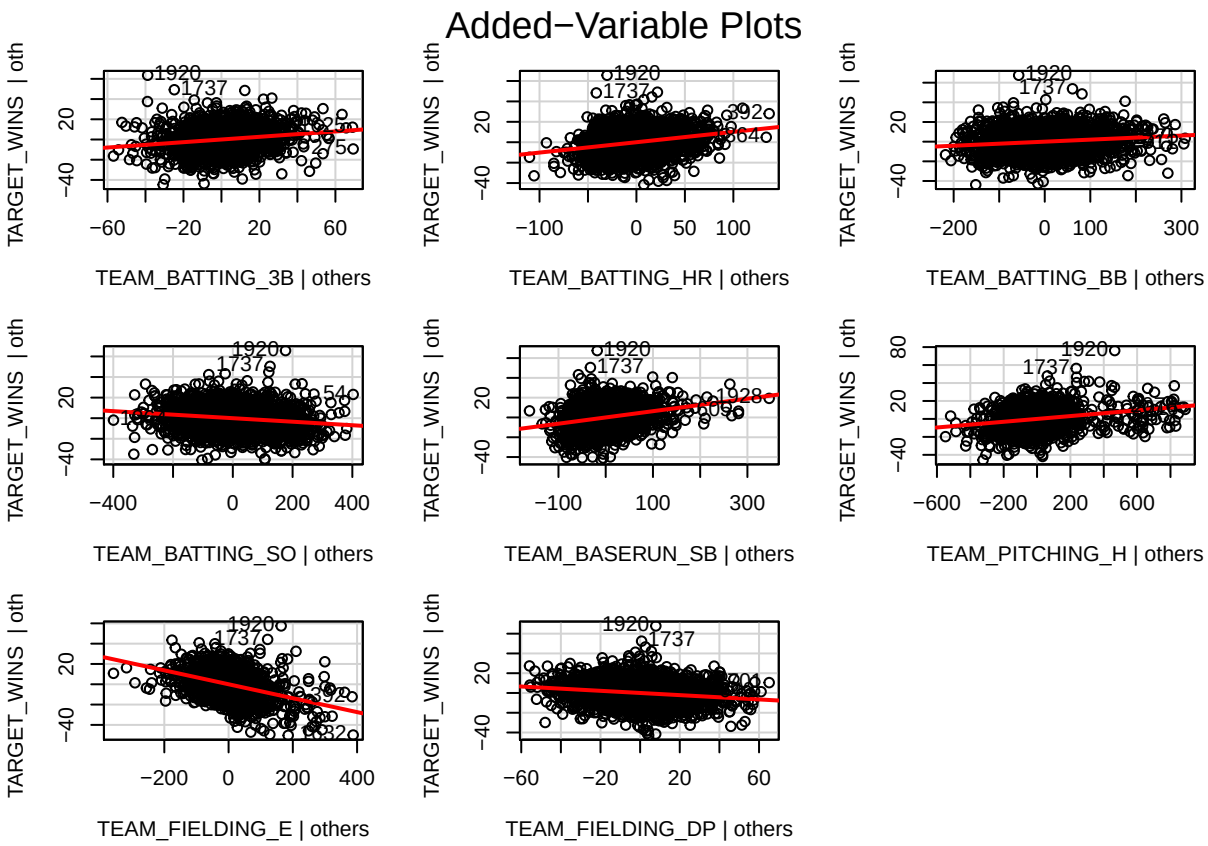
```
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##         3.088083         3.922549         1.481720         3.611668
##  TEAM_BASERUN_SB  TEAM_PITCHING_H  TEAM_FIELDING_E TEAM_FIELDING_DP
##         3.212525         2.419816         4.334892         2.064098
```

```
# no colinearity so STOP HERE
```

DIAGNOSTICS

Plots are linear except for errors

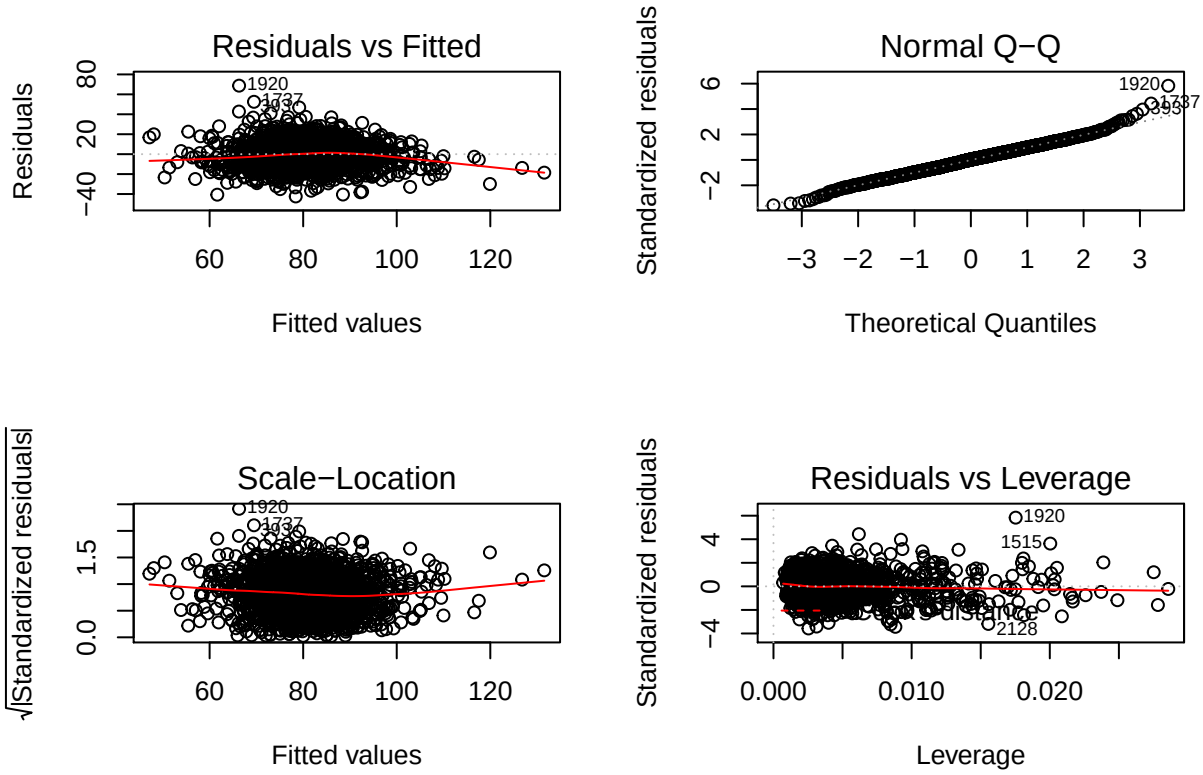
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.5, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Lack of constant variability in Resid vs. Fitted. Normal QQ shows a bit of skew in upper right end but not drastic; Residuals appear to be within 2 std devs. Outliers at 1920, 1737, 393, 1515

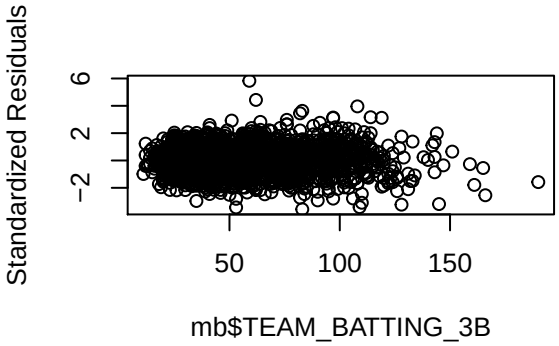
```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.5)
```

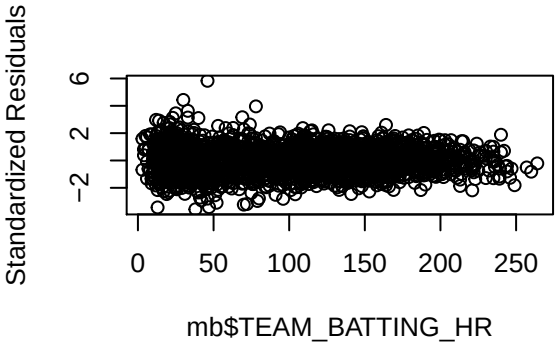


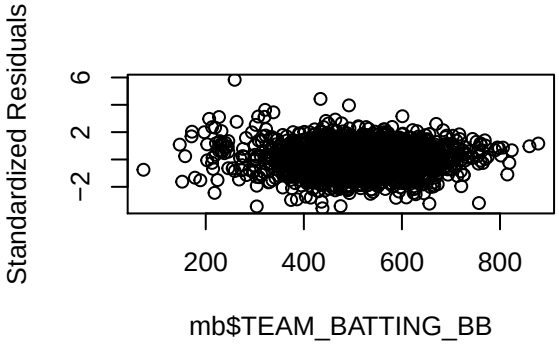
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

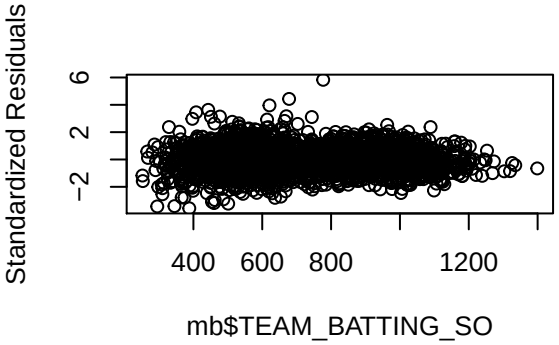
Results show lack of constant variability for several variables: 3B, HR, SB, Pitch_H, Fielding_E, BATT_BB, BATT_SO, FIELDING_DP

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.5)
par(mfrow=c(2,2))

plot(mb$TEAM_BATTING_3B, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_HR, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
```

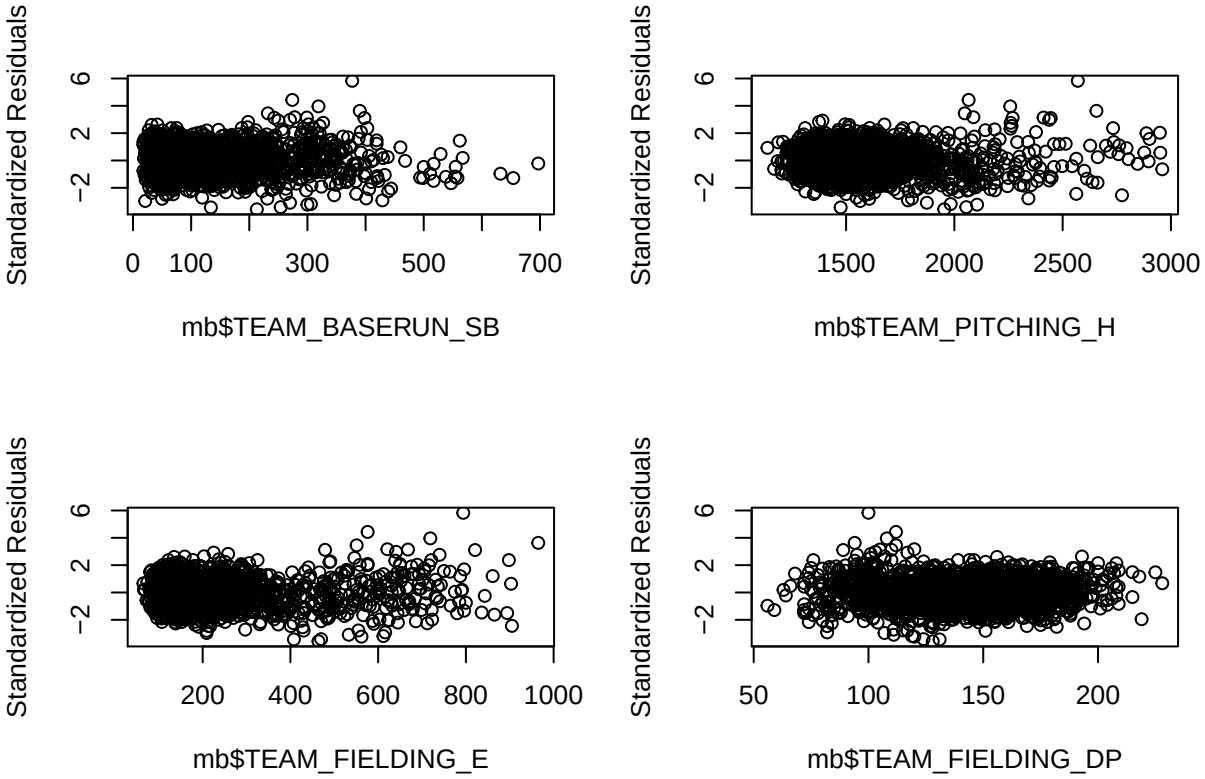








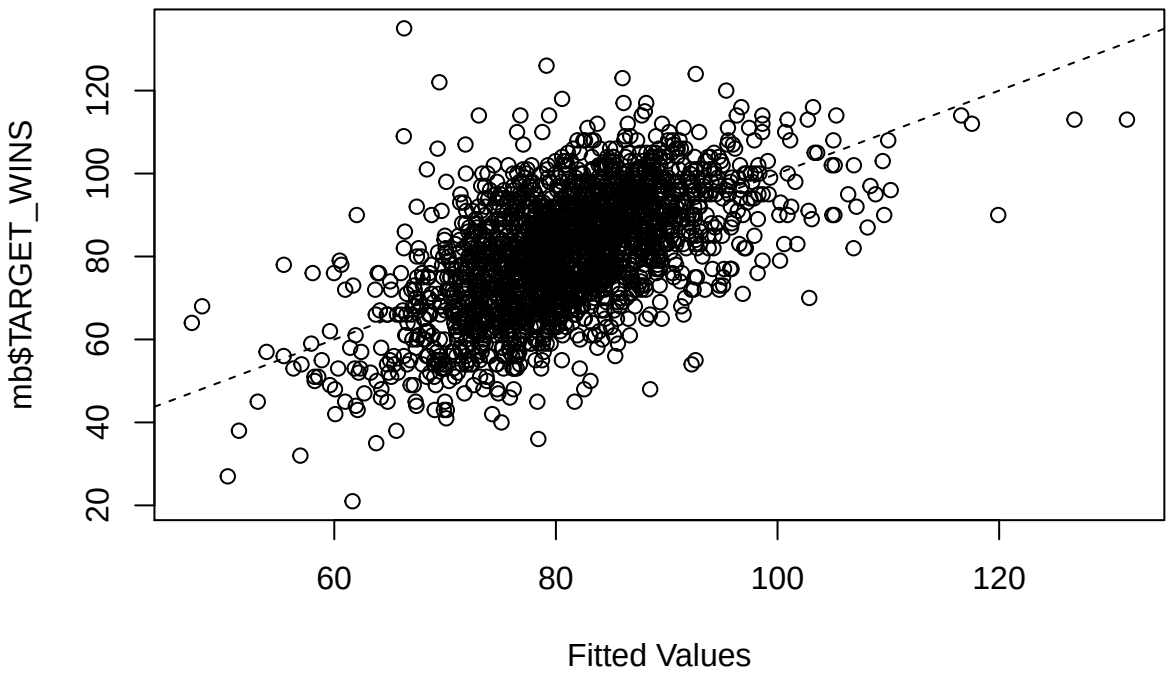

```
plot(mb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship whose slope might be skewed by outliers in upper right of plot

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.5$fitted.values
# nrow(fit1)

par(mfrow = c(1,1))
plot(fit1, mb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb$TARGET_WINS),lty=2)
```



Now cleanup data objects that are no longer required

```
rm(model, model.2, model.3, model.4)
```

REMOVE OUTLIERS AND REFIT

Per Cooks Distance, remove items 1920, 1737, 393, 1515

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem <- mb_clean[-c(1920, 1737, 393, 1515),]

# renumber rows
rownames(mb_rem) <- 1:nrow(mb_rem)
```

Now refit first model from above: all variables

Yields r^2= 0.3504, Adj r^2 = 0.348, F = 145.6

```
# keep the clean data set pure
mb <- mb_rem

# use p-value elimination
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.312  -7.606   0.169   7.370  45.974
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      52.702659   5.533457   9.524  < 2e-16 ***
## TEAM_BATTING_2B  -0.048364   0.008899  -5.435 6.10e-08 ***
## TEAM_BATTING_3B   0.131471   0.016835   7.809 8.91e-15 ***
## TEAM_BATTING_HR   0.077797   0.010045   7.745 1.46e-14 ***
## TEAM_BATTING_BB   0.184502   0.016217  11.377  < 2e-16 ***
## TEAM_BATTING_SO  -0.049773   0.011616  -4.285 1.91e-05 ***
## TEAM_BASERUN_SB   0.069365   0.004863  14.265  < 2e-16 ***
## TEAM_PITCHING_H   0.039055   0.004879   8.005 1.94e-15 ***
## TEAM_PITCHING_BB -0.144207   0.014074 -10.246  < 2e-16 ***
## TEAM_PITCHING_SO  0.031063   0.010855   2.862  0.00425 **
## TEAM_FIELDING_E  -0.081383   0.003851 -21.135  < 2e-16 ***
## TEAM_FIELDING_DP -0.119896   0.012795  -9.370  < 2e-16 ***
## TEAM_BATTING_1B  -0.013777   0.006856  -2.010  0.04460 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.37 on 2155 degrees of freedom
## Multiple R-squared:  0.3839, Adjusted R-squared:  0.3805
## F-statistic: 111.9 on 12 and 2155 DF,  p-value: < 2.2e-16
```

```
vif(model)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.753180         3.434748         5.922306        44.068099
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##       115.662896         3.377840        25.603370        37.973500
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP  TEAM_BATTING_1B
##        98.479804         5.337711         2.083157         8.252408
```

```
# vif indicates remove TEAM_PITCHING_SO

# --------------------
# remove TEAM_PITCHING_SO
model.2 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.494  -7.559   0.328   7.446  45.527
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      55.515612   5.454519  10.178  < 2e-16 ***
## TEAM_BATTING_2B  -0.056967   0.008389  -6.790 1.44e-11 ***
## TEAM_BATTING_3B   0.120478   0.016419   7.338 3.06e-13 ***
## TEAM_BATTING_HR   0.066476   0.009248   7.188 9.02e-13 ***
## TEAM_BATTING_BB   0.162650   0.014330  11.350  < 2e-16 ***
## TEAM_BATTING_SO  -0.017208   0.002335  -7.368 2.45e-13 ***
## TEAM_BASERUN_SB   0.067604   0.004831  13.992  < 2e-16 ***
## TEAM_PITCHING_H   0.048059   0.003735  12.866  < 2e-16 ***
## TEAM_PITCHING_BB -0.124471   0.012289 -10.129  < 2e-16 ***
## TEAM_FIELDING_E  -0.081495   0.003857 -21.130  < 2e-16 ***
## TEAM_FIELDING_DP -0.117961   0.012798  -9.217  < 2e-16 ***
## TEAM_BATTING_1B  -0.025520   0.005501  -4.639 3.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.39 on 2156 degrees of freedom
## Multiple R-squared:  0.3816, Adjusted R-squared:  0.3784
## F-statistic: 120.9 on 11 and 2156 DF,  p-value: < 2.2e-16
```

```
# p-values are OK so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.438954         3.255919         5.003748        34.297363
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##         4.659832         3.323768        14.956756        28.854387
##  TEAM_FIELDING_E TEAM_FIELDING_DP  TEAM_BATTING_1B
##         5.337159         2.077336         5.295547
```

```
# vif says remove TEAM_PITCHING_BB

# -------------------
#eliminate TEAM_PITCHING_BB
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.824  -7.989   0.294   7.673  42.263
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      58.243821   5.574685  10.448  < 2e-16 ***
## TEAM_BATTING_2B  -0.014168   0.007416  -1.910   0.0562 .
## TEAM_BATTING_3B   0.142709   0.016650   8.571  < 2e-16 ***
```

```
## TEAM_BATTING_HR   0.106734   0.008545  12.491  < 2e-16 ***
## TEAM_BATTING_BB   0.020695   0.003061   6.762 1.75e-11 ***
## TEAM_BATTING_SO  -0.015795   0.002386  -6.621 4.49e-11 ***
## TEAM_BASERUN_SB   0.063058   0.004923  12.810  < 2e-16 ***
## TEAM_PITCHING_H   0.015173   0.001890   8.028 1.61e-15 ***
## TEAM_FIELDING_E  -0.072930   0.003851 -18.940  < 2e-16 ***
## TEAM_FIELDING_DP -0.111549   0.013080  -8.528  < 2e-16 ***
## TEAM_BATTING_1B   0.007382   0.004543   1.625   0.1043
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.65 on 2157 degrees of freedom
## Multiple R-squared:  0.3522, Adjusted R-squared:  0.3492
## F-statistic: 117.3 on 10 and 2157 DF,  p-value: < 2.2e-16
```

```
# p-values say remove TEAM_BATTING_1B

# --------------------
#eliminate TEAM_BATTING_1B
model.4 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_1B - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.166  -7.970   0.318   7.689  41.838
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      65.022700   3.699490  17.576  < 2e-16 ***
## TEAM_BATTING_2B  -0.013481   0.007407  -1.820   0.0689 .
## TEAM_BATTING_3B   0.146314   0.016508   8.863  < 2e-16 ***
## TEAM_BATTING_HR   0.105677   0.008524  12.398  < 2e-16 ***
## TEAM_BATTING_BB   0.020919   0.003059   6.839 1.03e-11 ***
## TEAM_BATTING_SO  -0.017600   0.002112  -8.334  < 2e-16 ***
## TEAM_BASERUN_SB   0.064519   0.004842  13.326  < 2e-16 ***
## TEAM_PITCHING_H   0.016390   0.001736   9.443  < 2e-16 ***
## TEAM_FIELDING_E  -0.074524   0.003725 -20.007  < 2e-16 ***
## TEAM_FIELDING_DP -0.109715   0.013037  -8.416  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.66 on 2158 degrees of freedom
## Multiple R-squared:  0.3514, Adjusted R-squared:  0.3487
## F-statistic: 129.9 on 9 and 2158 DF,  p-value: < 2.2e-16
```

```
vif(model.4)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
```

```
##         1.814319         3.140953         4.055967         1.491080
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H  TEAM_FIELDING_E
##         3.636171         3.185297         3.081945         4.750943
## TEAM_FIELDING_DP
##         2.056837
```

```
# pvals say remove TEAM_BATTING_2B
model.5 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_BATTING_2B - TEAM_BATTING_1B - TEAM_PITCHING_SO - T
summary(model.5)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_BATTING_2B - TEAM_BATTING_1B -
##     TEAM_PITCHING_SO - TEAM_PITCHING_BB, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.605  -8.030   0.317   7.757  42.164
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      64.703618   3.697313  17.500  < 2e-16 ***
## TEAM_BATTING_3B   0.143076   0.016420   8.713  < 2e-16 ***
## TEAM_BATTING_HR   0.102858   0.008386  12.265  < 2e-16 ***
## TEAM_BATTING_BB   0.020349   0.003044   6.684 2.94e-11 ***
## TEAM_BATTING_SO  -0.017890   0.002107  -8.491  < 2e-16 ***
## TEAM_BASERUN_SB   0.064609   0.004844  13.338  < 2e-16 ***
## TEAM_PITCHING_H   0.014888   0.001528   9.745  < 2e-16 ***
## TEAM_FIELDING_E  -0.072407   0.003541 -20.450  < 2e-16 ***
## TEAM_FIELDING_DP -0.110117   0.013042  -8.443  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.66 on 2159 degrees of freedom
## Multiple R-squared:  0.3504, Adjusted R-squared:  0.348
## F-statistic: 145.6 on 8 and 2159 DF,  p-value: < 2.2e-16
```

```
vif(model.5)
```

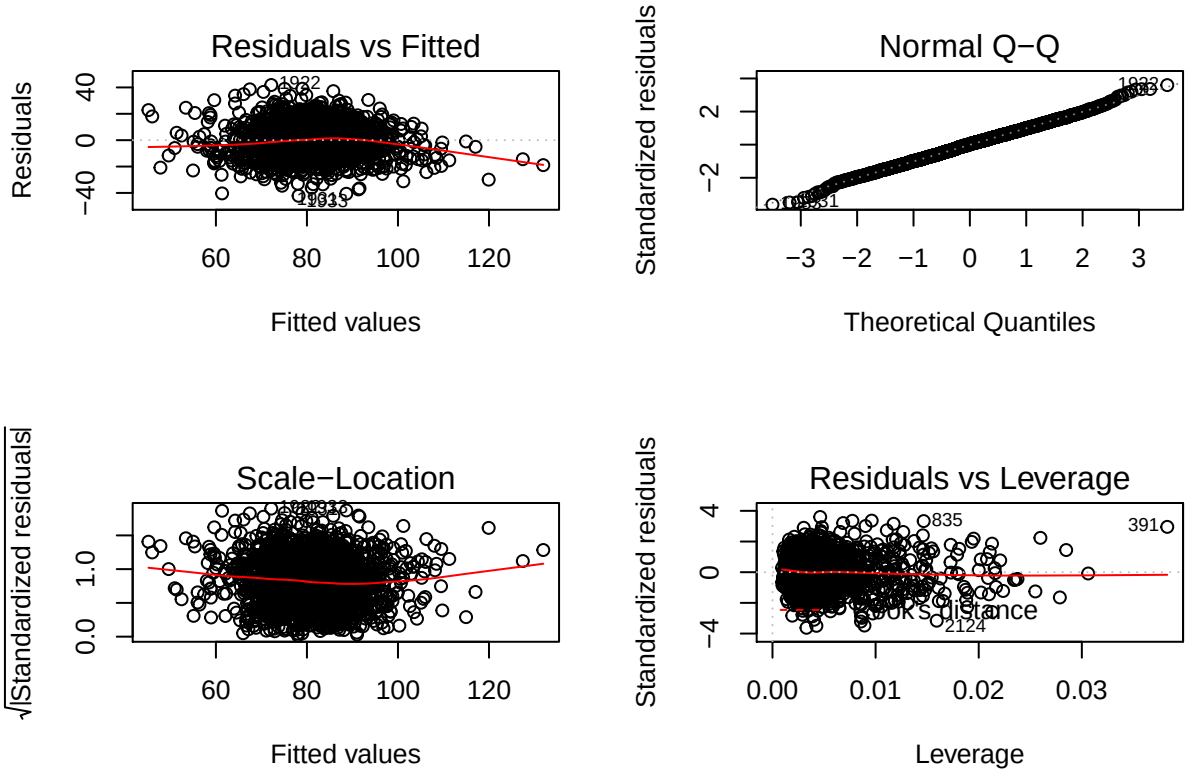
```
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##         3.104460         3.922081         1.475492         3.615480
##  TEAM_BASERUN_SB  TEAM_PITCHING_H  TEAM_FIELDING_E TEAM_FIELDING_DP
##         3.184962         2.384787         4.287809         2.056247
```

```
# vif and pvals OK so stop
```

SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Lack of constant variability in Resid vs. Fitted. Normal QQ shows a bit of skew in upper right end but not drastic; Residuals appear to be within 2 std devs. **Might not be a good model**.

```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.4)
```



Plots show outliers so remove them and re-fit

Per Cooks Distance, remove items 1931, 391, 820, 1933, 835, 2124

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem2 <- mb[-c(1931, 391, 820, 1933, 835, 2124),]

# renumber rows
rownames(mb_rem2) <- 1:nrow(mb_rem2)
```

Model using all remaining variables as a starting point

Yields r^2= 0.3598, Adj r^2 = 0.3572, F = 134.4

```
# keep the clean data set pure
mb <- mb_rem2

# use p-value elimination
```

```r
model <- lm(data=mb, TARGET_WINS ~ . - INDEX )
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.835  -7.607   0.200   7.360  45.489
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      53.581847   5.491812   9.757  < 2e-16 ***
## TEAM_BATTING_2B  -0.051601   0.008852  -5.829 6.40e-09 ***
## TEAM_BATTING_3B   0.131444   0.016694   7.874 5.42e-15 ***
## TEAM_BATTING_HR   0.080041   0.009950   8.044 1.42e-15 ***
## TEAM_BATTING_BB   0.171434   0.016181  10.595  < 2e-16 ***
## TEAM_BATTING_SO  -0.042020   0.011568  -3.633 0.000287 ***
## TEAM_BASERUN_SB   0.068541   0.004809  14.252  < 2e-16 ***
## TEAM_PITCHING_H   0.039065   0.004831   8.086 1.02e-15 ***
## TEAM_PITCHING_BB -0.132140   0.014071  -9.391  < 2e-16 ***
## TEAM_PITCHING_SO  0.023143   0.010823   2.138 0.032608 *
## TEAM_FIELDING_E  -0.080763   0.003829 -21.093  < 2e-16 ***
## TEAM_FIELDING_DP -0.118407   0.012650  -9.360  < 2e-16 ***
## TEAM_BATTING_1B  -0.013730   0.006805  -2.018 0.043761 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.24 on 2149 degrees of freedom
## Multiple R-squared:  0.3885, Adjusted R-squared:  0.3851
## F-statistic: 113.8 on 12 and 2149 DF,  p-value: < 2.2e-16
```

```r
# p-values all < .05 so check collinearity
vif(model)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.747597         3.421239         5.926475        44.308309
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##       116.779721         3.345094        25.197359        38.414216
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP  TEAM_BATTING_1B
##        99.525121         5.265275         2.072666         8.259655
```

```r
# vif says remove TEAM_PITCHING_SO

# --------------------
#eliminate TEAM_PITCHING_SO
model.2 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
```

```
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.771  -7.636   0.258   7.342  45.148
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      55.723220   5.404214  10.311  < 2e-16 ***
## TEAM_BATTING_2B  -0.057976   0.008342  -6.950 4.82e-12 ***
## TEAM_BATTING_3B   0.123500   0.016289   7.582 5.04e-14 ***
## TEAM_BATTING_HR   0.071757   0.009173   7.823 8.03e-15 ***
## TEAM_BATTING_BB   0.154860   0.014216  10.893  < 2e-16 ***
## TEAM_BATTING_SO  -0.017783   0.002312  -7.691 2.20e-14 ***
## TEAM_BASERUN_SB   0.067235   0.004774  14.083  < 2e-16 ***
## TEAM_PITCHING_H   0.045685   0.003712  12.308  < 2e-16 ***
## TEAM_PITCHING_BB -0.117136   0.012207  -9.596  < 2e-16 ***
## TEAM_FIELDING_E  -0.080871   0.003832 -21.106  < 2e-16 ***
## TEAM_FIELDING_DP -0.116948   0.012642  -9.251  < 2e-16 ***
## TEAM_BATTING_1B  -0.022420   0.005463  -4.104 4.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.25 on 2150 degrees of freedom
## Multiple R-squared:  0.3872, Adjusted R-squared:  0.384
## F-statistic: 123.5 on 11 and 2150 DF,  p-value: < 2.2e-16
```

```
# p-values OK so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.435898         3.251820         5.028029        34.141690
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##         4.657608         3.291089        14.849242        28.862800
##  TEAM_FIELDING_E TEAM_FIELDING_DP  TEAM_BATTING_1B
##         5.264363         2.066635         5.314335
```

```
# vif says remove TEAM_PITCHING_BB

# --------------------
#eliminate TEAM_BATTING_BB
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.474  -7.928   0.289   7.545  42.297
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      58.415319   5.509998  10.602  < 2e-16 ***
## TEAM_BATTING_2B  -0.018310   0.007397  -2.475   0.0134 *
## TEAM_BATTING_3B   0.145166   0.016470   8.814  < 2e-16 ***
## TEAM_BATTING_HR   0.109983   0.008436  13.038  < 2e-16 ***
## TEAM_BATTING_BB   0.021452   0.003028   7.084 1.89e-12 ***
## TEAM_BATTING_SO  -0.016549   0.002357  -7.021 2.93e-12 ***
## TEAM_BASERUN_SB   0.063016   0.004854  12.984  < 2e-16 ***
## TEAM_PITCHING_H   0.014679   0.001865   7.872 5.48e-15 ***
## TEAM_FIELDING_E  -0.073003   0.003821 -19.104  < 2e-16 ***
## TEAM_FIELDING_DP -0.110856   0.012890  -8.600  < 2e-16 ***
## TEAM_BATTING_1B   0.008578   0.004498   1.907   0.0567 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.48 on 2151 degrees of freedom
## Multiple R-squared:  0.3609, Adjusted R-squared:  0.358
## F-statistic: 121.5 on 10 and 2151 DF,  p-value: < 2.2e-16
```

```
# p-values says remove TEAM_BATTING_1B

# --------------------
#eliminate TEAM_BATTING_1B
model.4 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB - TEAM_BATTING_1B)
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB -
##     TEAM_BATTING_1B, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.519  -8.001   0.336   7.614  41.812
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.261095   3.667019  18.069  < 2e-16 ***
## TEAM_BATTING_2B  -0.017347   0.007384  -2.349   0.0189 *
## TEAM_BATTING_3B   0.149391   0.016330   9.148  < 2e-16 ***
## TEAM_BATTING_HR   0.108695   0.008414  12.919  < 2e-16 ***
## TEAM_BATTING_BB   0.021727   0.003027   7.178 9.67e-13 ***
## TEAM_BATTING_SO  -0.018634   0.002089  -8.920  < 2e-16 ***
## TEAM_BASERUN_SB   0.064701   0.004775  13.549  < 2e-16 ***
## TEAM_PITCHING_H   0.016082   0.001715   9.379  < 2e-16 ***
## TEAM_FIELDING_E  -0.074833   0.003701 -20.219  < 2e-16 ***
## TEAM_FIELDING_DP -0.108762   0.012851  -8.463  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.49 on 2152 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3572
## F-statistic: 134.4 on 9 and 2152 DF,  p-value: < 2.2e-16
```

```r
# p-values OK so check collinearity
vif(model.4)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         1.829117         3.131630         4.053547         1.482965
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H  TEAM_FIELDING_E
##         3.643091         3.155029         3.036321         4.706195
## TEAM_FIELDING_DP
##         2.046463
```

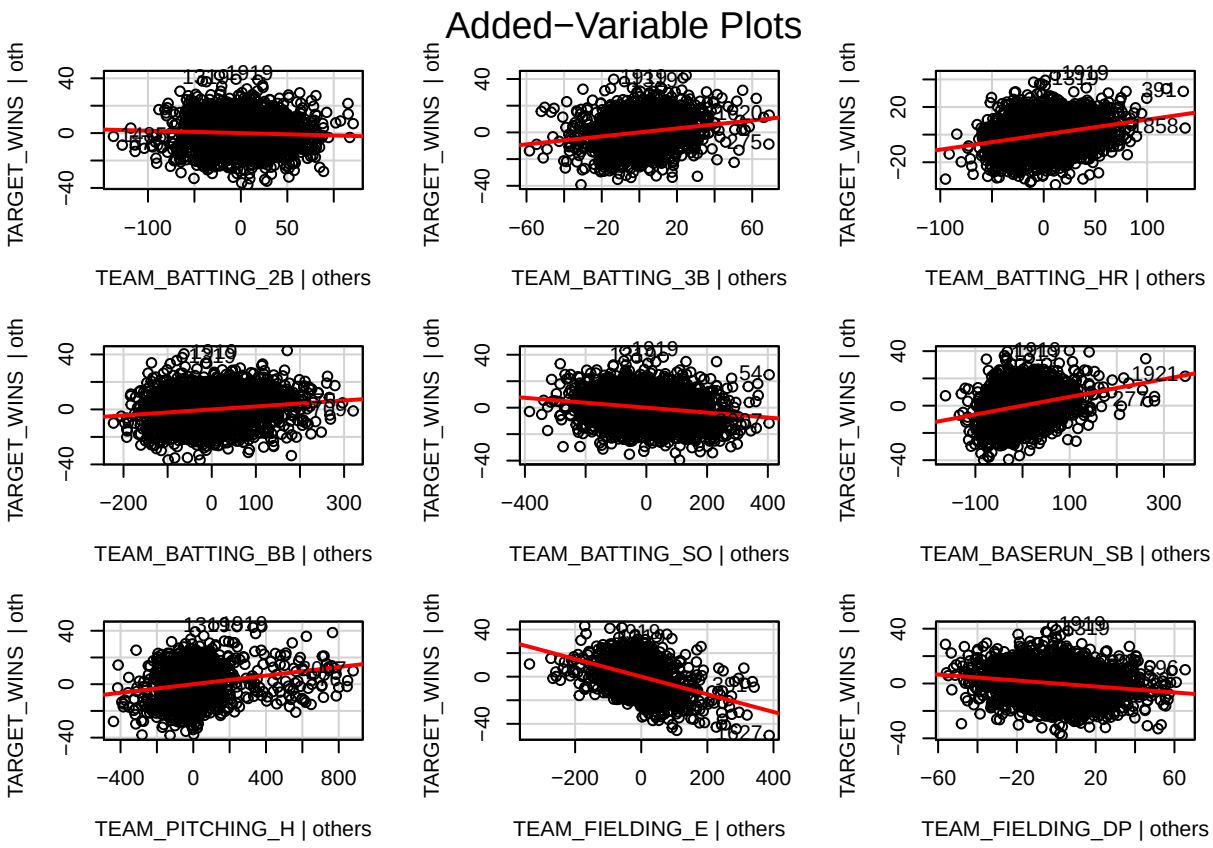
```r
# vif OK so STOP

# get MSE of residuals
anova(model.4)
```

```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq  F value    Pr(>F)
## TEAM_BATTING_2B     1  23852   23852 180.7371 < 2.2e-16 ***
## TEAM_BATTING_3B     1  12881   12881  97.6052 < 2.2e-16 ***
## TEAM_BATTING_HR     1  30779   30779 233.2296 < 2.2e-16 ***
## TEAM_BATTING_BB     1  15162   15162 114.8928 < 2.2e-16 ***
## TEAM_BATTING_SO     1   5965    5965  45.2032 2.268e-11 ***
## TEAM_BASERUN_SB     1  12795   12795  96.9513 < 2.2e-16 ***
## TEAM_PITCHING_H     1     63      63   0.4781    0.4894
## TEAM_FIELDING_E     1  48693   48693 368.9741 < 2.2e-16 ***
## TEAM_FIELDING_DP    1   9452    9452  71.6227 < 2.2e-16 ***
## Residuals        2152 283999     132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diagnostics

Plots are linear except for errors

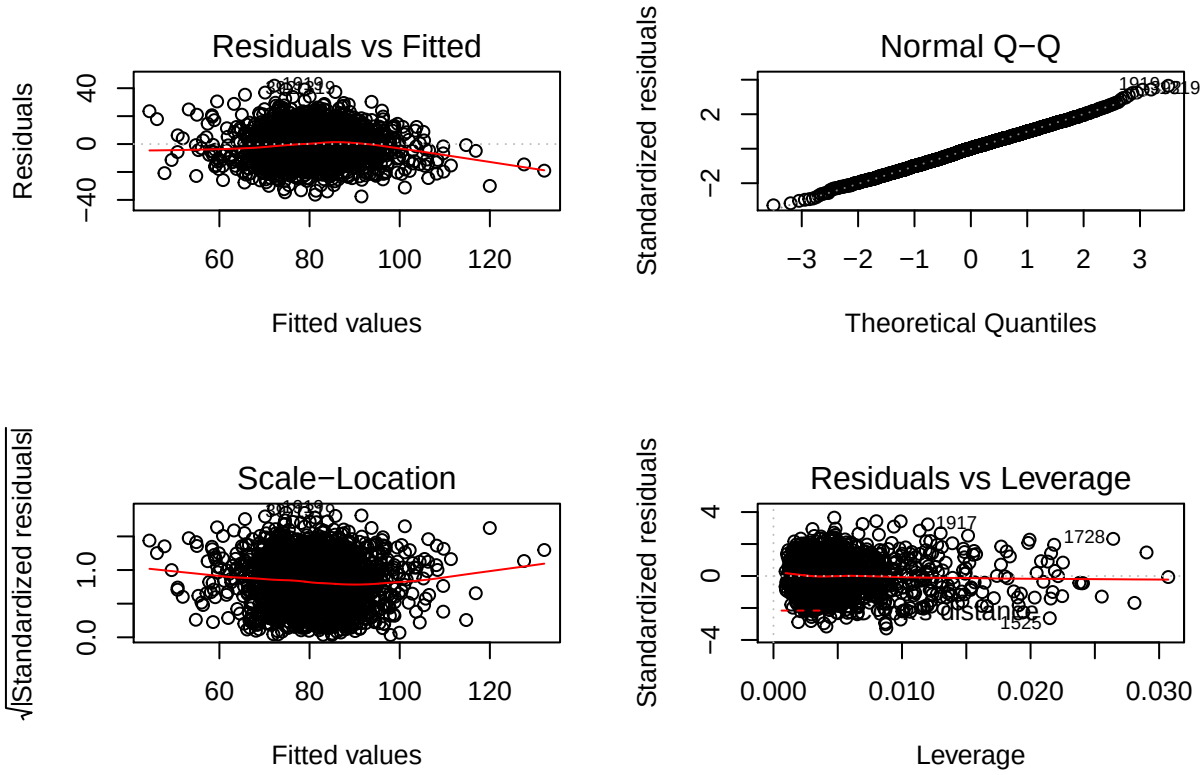
```r
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.4, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Lack of constant variability in Resid vs. Fitted. Normal QQ shows a bit of skew in upper right end but not drastic; Residuals appear to be within 2 std devs.

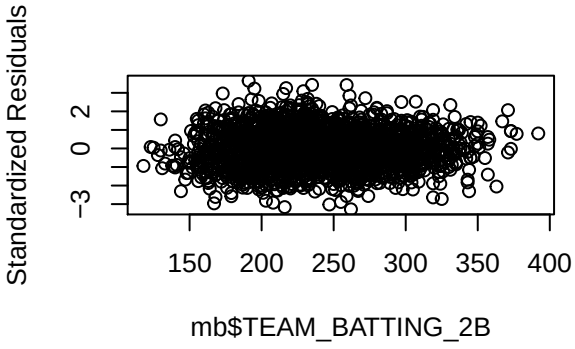
```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.4)
```

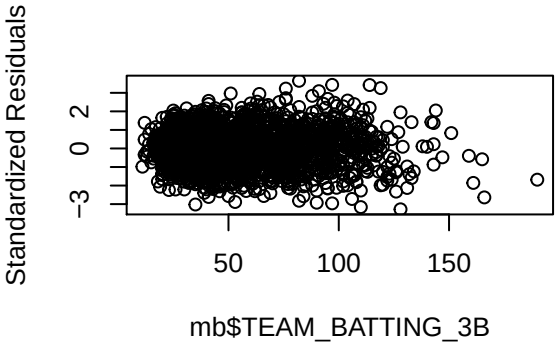


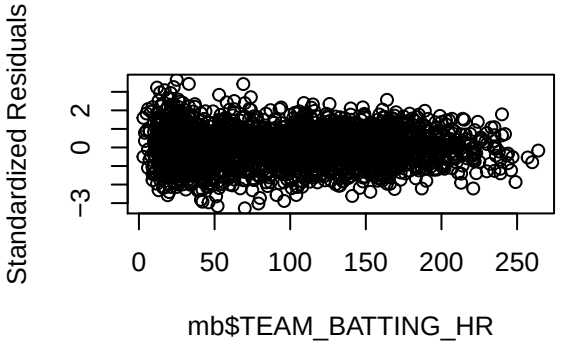
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

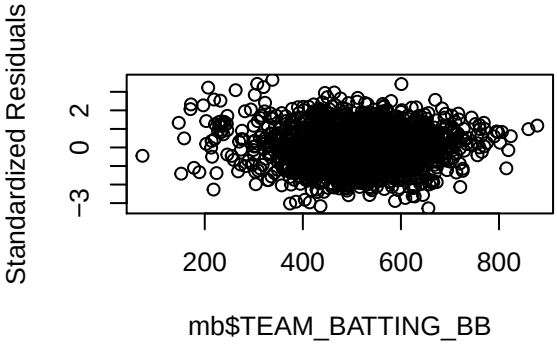
Results show lack of constant variability for several variables: 3B, HR, SB, Pitch_H, Fielding_E, PITCH_BB, PITCH_SO, FIELDING_DP

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.4)
par(mfrow=c(2,2))

plot(mb$TEAM_BATTING_2B, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_3B, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_HR, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
```

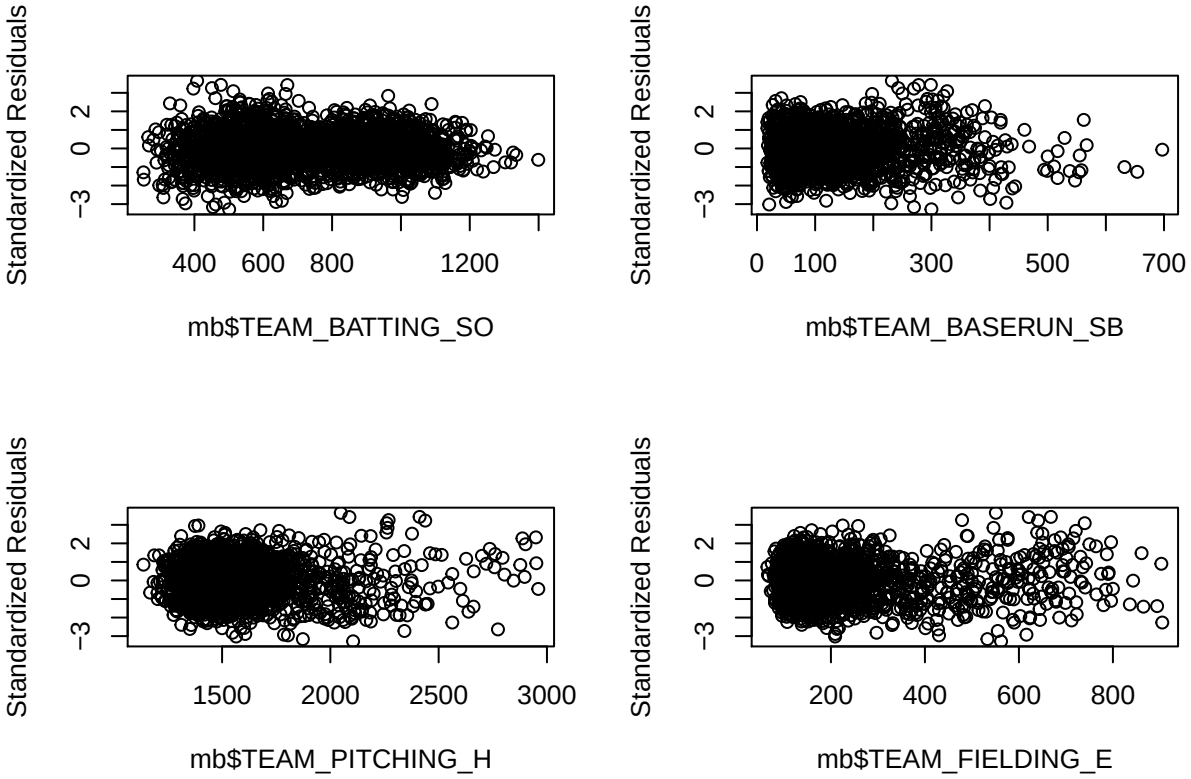




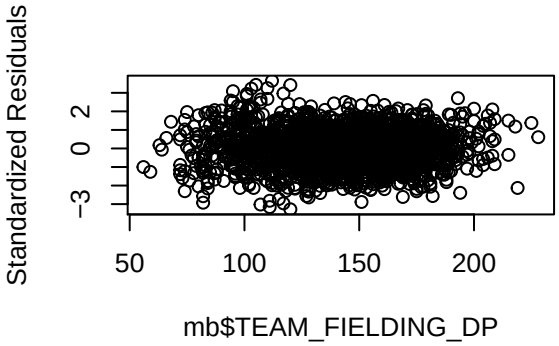





```
plot(mb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
```



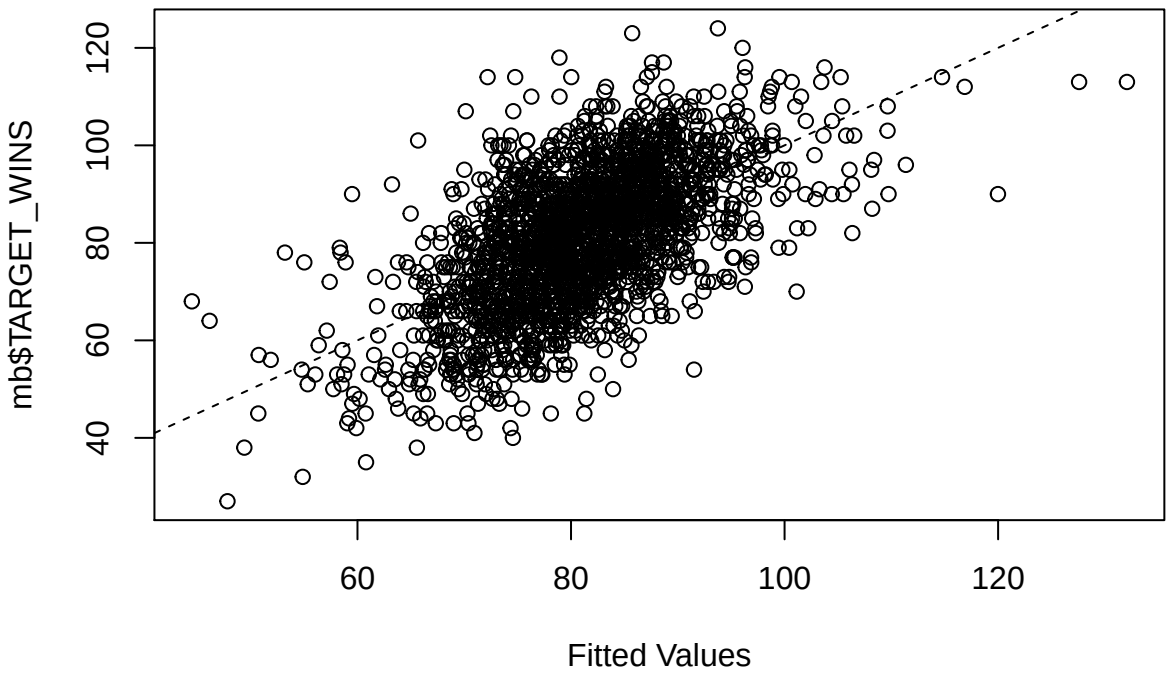

```
plot(mb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship whose slope might be skewed by outliers in upper right of plot

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.4$fitted.values
# nrow(fit1)

par(mfrow = c(1,1))
plot(fit1, mb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb$TARGET_WINS),lty=2)
```



Now try same model but with FIELD_E transformed by Box-Cox recommended power transform

```
# TEAM_FIELDING_E: Box-cox yields -1 => 1/y

mb$TEAM_FIELDING_E <- 1/mb$TEAM_FIELDING_E
```

Yields r^2= 0.3168, Adj r^2 = 0.3143, F = 124.8

```
# use p-value elimination
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.440  -7.481   0.083   7.789  41.052
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)       2.414e+01  5.587e+00   4.321 1.62e-05 ***
## TEAM_BATTING_2B  -2.377e-02  8.948e-03  -2.656  0.00797 **
## TEAM_BATTING_3B   1.520e-01  1.771e-02   8.585  < 2e-16 ***
## TEAM_BATTING_HR   8.730e-02  1.038e-02   8.414  < 2e-16 ***
## TEAM_BATTING_BB   1.226e-01  1.663e-02   7.372 2.38e-13 ***
## TEAM_BATTING_SO  -4.739e-02  1.204e-02  -3.936 8.55e-05 ***
## TEAM_BASERUN_SB   3.330e-02  4.524e-03   7.360 2.60e-13 ***
## TEAM_PITCHING_H   1.082e-02  4.739e-03   2.284  0.02250 *
## TEAM_PITCHING_BB -8.508e-02  1.440e-02  -5.907 4.05e-09 ***
## TEAM_PITCHING_SO  2.482e-02  1.126e-02   2.205  0.02754 *
## TEAM_FIELDING_E   3.247e+03  2.057e+02  15.781  < 2e-16 ***
## TEAM_FIELDING_DP -1.143e-01  1.318e-02  -8.673  < 2e-16 ***
## TEAM_BATTING_1B   1.841e-02  6.831e-03   2.696  0.00708 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.69 on 2149 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.3348
## F-statistic: 91.65 on 12 and 2149 DF,  p-value: < 2.2e-16
```

```
# p-values all < .05 so check collinearity

vif(model)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
##         2.595850         3.558823         5.957395        43.240147
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##       116.965149         2.736438        22.414047        37.210796
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP  TEAM_BATTING_1B
##        99.513424         4.243335         2.081098         7.692617
```

```
# vif says remove TEAM_PITCHING_SO

# --------------------
#eliminate TEAM_PITCHING_SO
model.2 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.855  -7.541   0.243   7.891  39.291
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.640e+01  5.498e+00   4.802 1.68e-06 ***
## TEAM_BATTING_2B  -3.055e-02  8.410e-03  -3.633 0.000287 ***
## TEAM_BATTING_3B   1.435e-01  1.730e-02   8.296  < 2e-16 ***
## TEAM_BATTING_HR   7.844e-02  9.574e-03   8.193 4.34e-16 ***
## TEAM_BATTING_BB   1.047e-01  1.453e-02   7.205 8.01e-13 ***
```

```
## TEAM_BATTING_SO  -2.140e-02  2.459e-03  -8.701  < 2e-16 ***
## TEAM_BASERUN_SB   3.184e-02  4.480e-03   7.108 1.59e-12 ***
## TEAM_PITCHING_H   1.788e-02  3.498e-03   5.112 3.48e-07 ***
## TEAM_PITCHING_BB -6.891e-02  1.241e-02  -5.553 3.16e-08 ***
## TEAM_FIELDING_E   3.250e+03  2.059e+02  15.784  < 2e-16 ***
## TEAM_FIELDING_DP -1.128e-01  1.317e-02  -8.558  < 2e-16 ***
## TEAM_BATTING_1B   9.141e-03  5.388e-03   1.697 0.089933 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.7 on 2150 degrees of freedom
## Multiple R-squared:  0.337,  Adjusted R-squared:  0.3336
## F-statistic: 99.36 on 11 and 2150 DF,  p-value: < 2.2e-16
```

```
# p-values says remove TEAM_BATTING_1B

# --------------------
#eliminate TEAM_BATTING_1B
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.322  -7.571   0.210   7.727  39.339
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.140e+01  4.642e+00   6.765 1.71e-11 ***
## TEAM_BATTING_2B  -3.284e-02  8.305e-03  -3.954 7.94e-05 ***
## TEAM_BATTING_3B   1.435e-01  1.731e-02   8.290  < 2e-16 ***
## TEAM_BATTING_HR   7.398e-02  9.211e-03   8.032 1.56e-15 ***
## TEAM_BATTING_BB   1.187e-01  1.199e-02   9.902  < 2e-16 ***
## TEAM_BATTING_SO  -2.305e-02  2.258e-03 -10.212  < 2e-16 ***
## TEAM_BASERUN_SB   3.240e-02  4.469e-03   7.250 5.79e-13 ***
## TEAM_PITCHING_H   2.159e-02  2.733e-03   7.900 4.42e-15 ***
## TEAM_PITCHING_BB -8.082e-02  1.024e-02  -7.893 4.67e-15 ***
## TEAM_FIELDING_E   3.274e+03  2.055e+02  15.931  < 2e-16 ***
## TEAM_FIELDING_DP -1.111e-01  1.315e-02  -8.453  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.7 on 2151 degrees of freedom
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.3331
## F-statistic: 108.9 on 10 and 2151 DF,  p-value: < 2.2e-16
```

```
vif(model.3)
```

```
##  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
```

```
##         2.230211         3.389546         4.682396        22.413215
##  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_BB
##         4.100904         2.663642         7.432730        18.756439
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##         4.223007         2.063646
```

```
# p-values says remove TEAM_PITCHING_BB

# -------------------
#eliminate TEAM_BATTING_BB
model.4 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B - TEAM_PITCHING_BB)
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B -
##     TEAM_PITCHING_BB, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.522  -7.843   0.386   7.968  39.441
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.286e+01  3.815e+00  13.856  < 2e-16 ***
## TEAM_BATTING_2B  -3.694e-03  7.545e-03  -0.490   0.6245
## TEAM_BATTING_3B   1.696e-01  1.723e-02   9.847  < 2e-16 ***
## TEAM_BATTING_HR   9.684e-02  8.867e-03  10.921  < 2e-16 ***
## TEAM_BATTING_BB   2.722e-02  3.103e-03   8.771  < 2e-16 ***
## TEAM_BATTING_SO  -2.741e-02  2.220e-03 -12.349  < 2e-16 ***
## TEAM_BASERUN_SB   3.418e-02  4.527e-03   7.550 6.38e-14 ***
## TEAM_PITCHING_H   3.888e-03  1.584e-03   2.455   0.0142 *
## TEAM_FIELDING_E   3.282e+03  2.084e+02  15.744  < 2e-16 ***
## TEAM_FIELDING_DP -1.022e-01  1.328e-02  -7.693 2.17e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.87 on 2152 degrees of freedom
## Multiple R-squared:  0.3169, Adjusted R-squared:  0.3141
## F-statistic: 110.9 on 9 and 2152 DF,  p-value: < 2.2e-16
```

```
# p-values say remove doubles
model.5 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B - TEAM_PITCHING_BB -
summary(model.5)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B -
##     TEAM_PITCHING_BB - TEAM_BATTING_2B, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.512  -7.826   0.349   8.005  39.581
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.288e+01  3.815e+00  13.862  < 2e-16 ***
## TEAM_BATTING_3B   1.685e-01  1.707e-02   9.873  < 2e-16 ***
## TEAM_BATTING_HR   9.616e-02  8.757e-03  10.981  < 2e-16 ***
## TEAM_BATTING_BB   2.701e-02  3.075e-03   8.785  < 2e-16 ***
## TEAM_BATTING_SO  -2.741e-02  2.219e-03 -12.350  < 2e-16 ***
## TEAM_BASERUN_SB   3.443e-02  4.496e-03   7.658 2.84e-14 ***
## TEAM_PITCHING_H   3.568e-03  1.442e-03   2.474   0.0134 *
## TEAM_FIELDING_E   3.252e+03  1.997e+02  16.288  < 2e-16 ***
## TEAM_FIELDING_DP -1.023e-01  1.328e-02  -7.708 1.94e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.86 on 2153 degrees of freedom
## Multiple R-squared:  0.3168, Adjusted R-squared:  0.3143
## F-statistic: 124.8 on 8 and 2153 DF,  p-value: < 2.2e-16
```

```
vif(model.5)
```

```
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##         3.206158         4.116662         1.434886         3.855309
##  TEAM_BASERUN_SB  TEAM_PITCHING_H  TEAM_FIELDING_E TEAM_FIELDING_DP
##         2.622091         2.014360         3.877112         2.047278
```

```
# vif OK so STOP

# turn off scientific formatting of results
options(scipen=999)
model.5
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_BATTING_1B -
##     TEAM_PITCHING_BB - TEAM_BATTING_2B, data = mb)
##
## Coefficients:
##      (Intercept)   TEAM_BATTING_3B   TEAM_BATTING_HR   TEAM_BATTING_BB
##        52.876911          0.168490          0.096164          0.027014
##  TEAM_BATTING_SO   TEAM_BASERUN_SB   TEAM_PITCHING_H   TEAM_FIELDING_E
##        -0.027410          0.034431          0.003568       3252.309198
## TEAM_FIELDING_DP
##        -0.102330
```

```
anova(model.5)
```

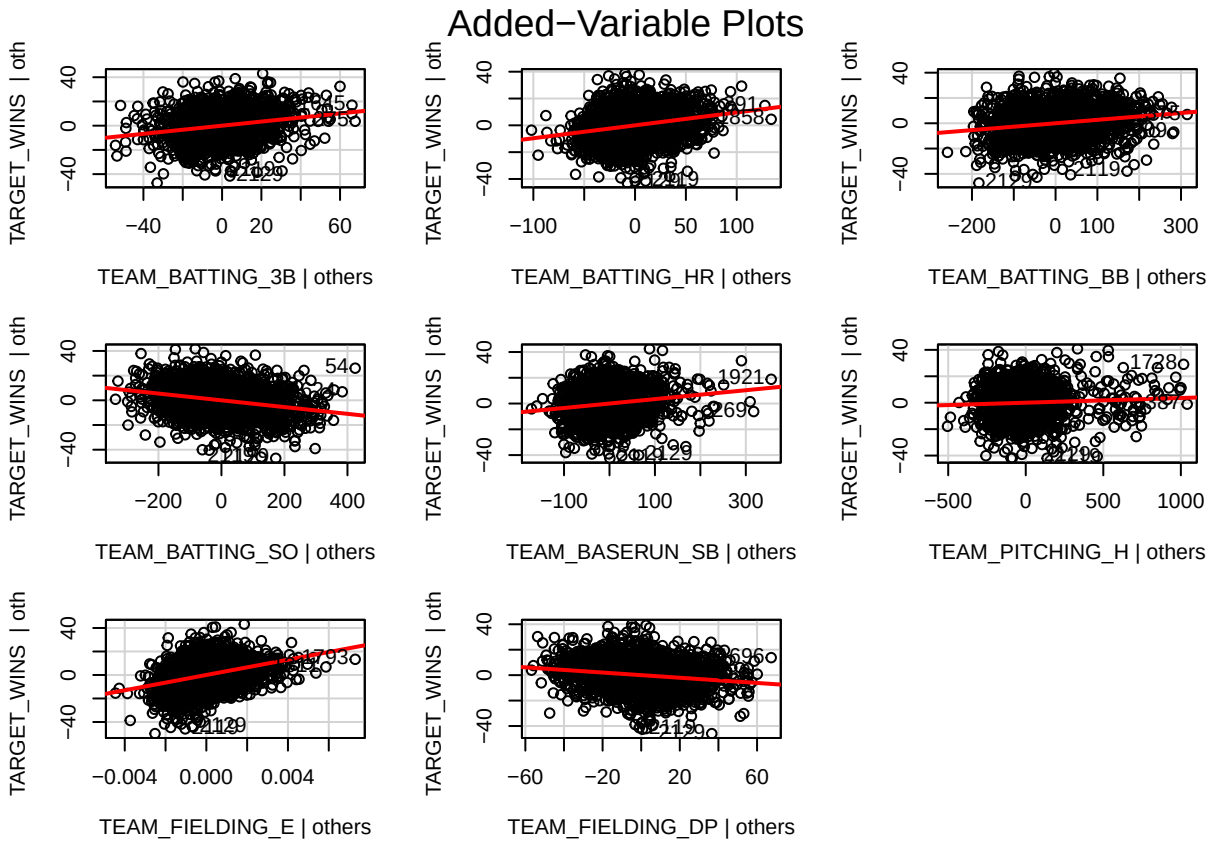
```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq  F value                Pr(>F)
## TEAM_BATTING_3B     1   7851    7851  55.7747   0.00000000000011740 ***
```

```
## TEAM_BATTING_HR     1  55626   55626 395.1504 < 0.00000000000000022 ***
## TEAM_BATTING_BB     1  16652   16652 118.2947 < 0.00000000000000022 ***
## TEAM_BATTING_SO     1   6913    6913  49.1095   0.00000000000322538 ***
## TEAM_BASERUN_SB     1  12274   12274  87.1878 < 0.00000000000000022 ***
## TEAM_PITCHING_H     1    118     118   0.8379                0.3601
## TEAM_FIELDING_E     1  32763   32763 232.7428 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1   8364    8364  59.4139   0.00000000000001937 ***
## Residuals        2153 303080     141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diagnostics

Plots are all linear

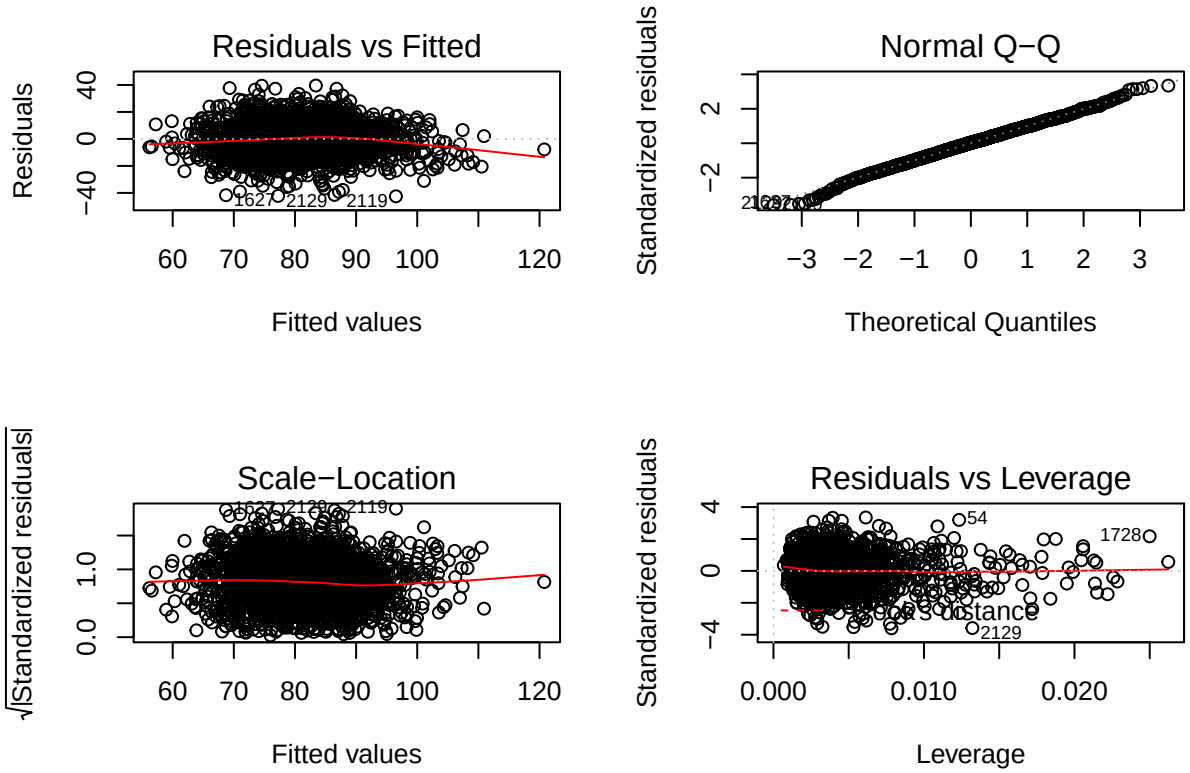
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.5, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Some lack of constant variability in Resid vs. Fitted. Normal QQ shows a bit of skew in lower left end but not drastic; Residuals not all within 2 std devs.

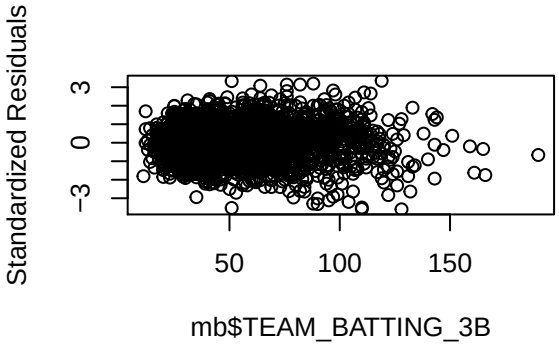
```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.5)
```

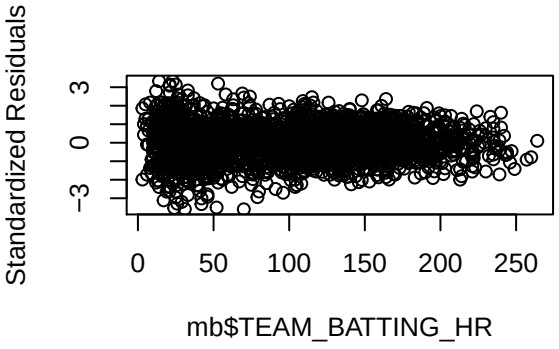


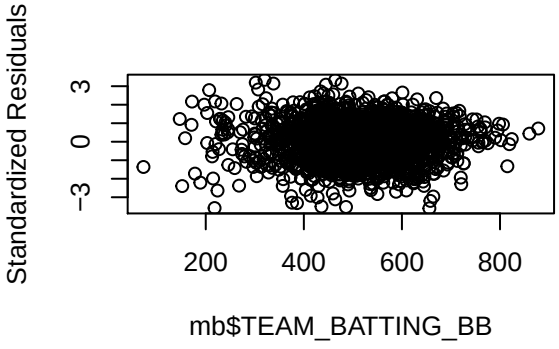
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

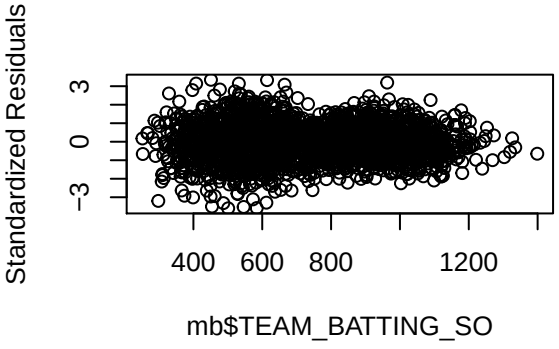
Results show lack of constant variability for several variables: 3B, HR, BB, SO, SB, Pitch_H, Fielding_E

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.5)
par(mfrow=c(2,2))

plot(mb$TEAM_BATTING_3B, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_HR, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
```

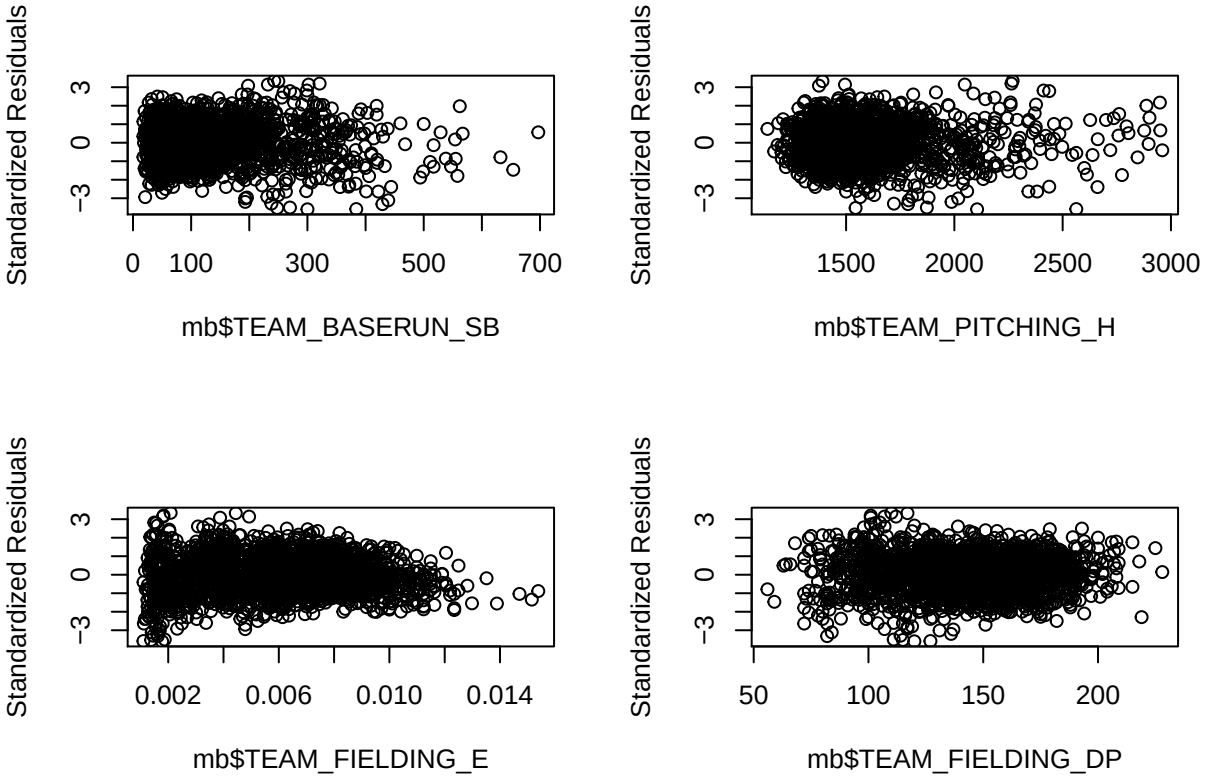








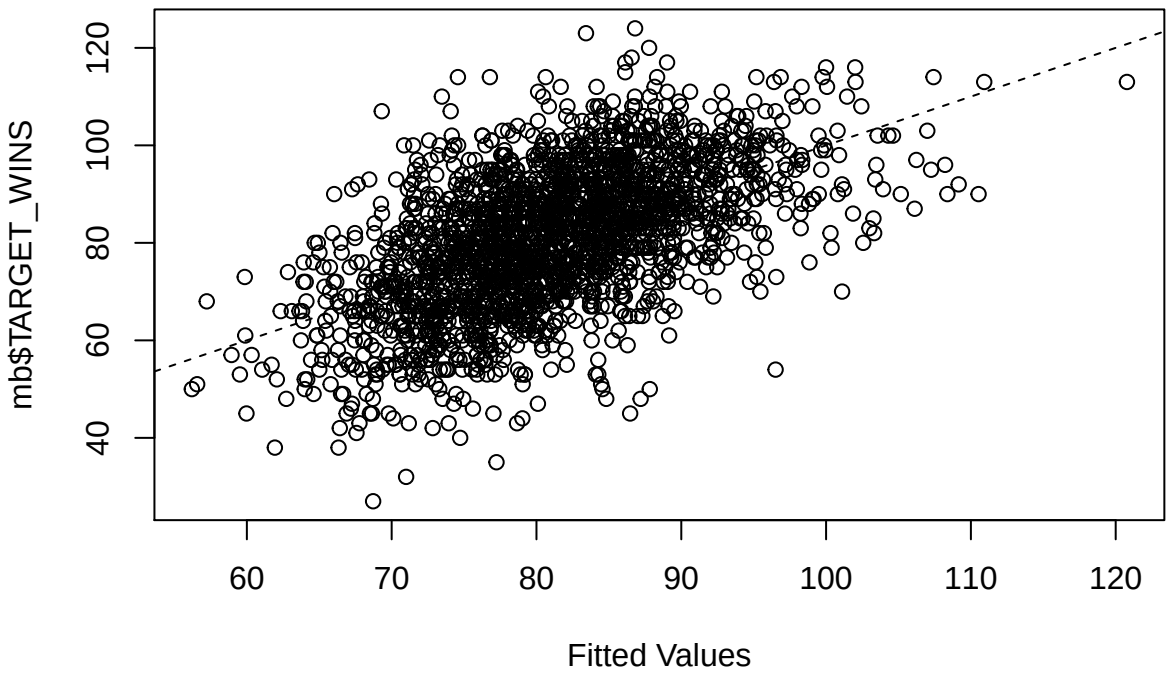

```
plot(mb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship whose slope might be slightly skewed by outliers in upper right of plot

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.5$fitted.values
# nrow(fit1)

par(mfrow = c(1,1))
plot(fit1, mb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb$TARGET_WINS),lty=2)
```




```
# clean up objects in memory
rm(list = ls())
```

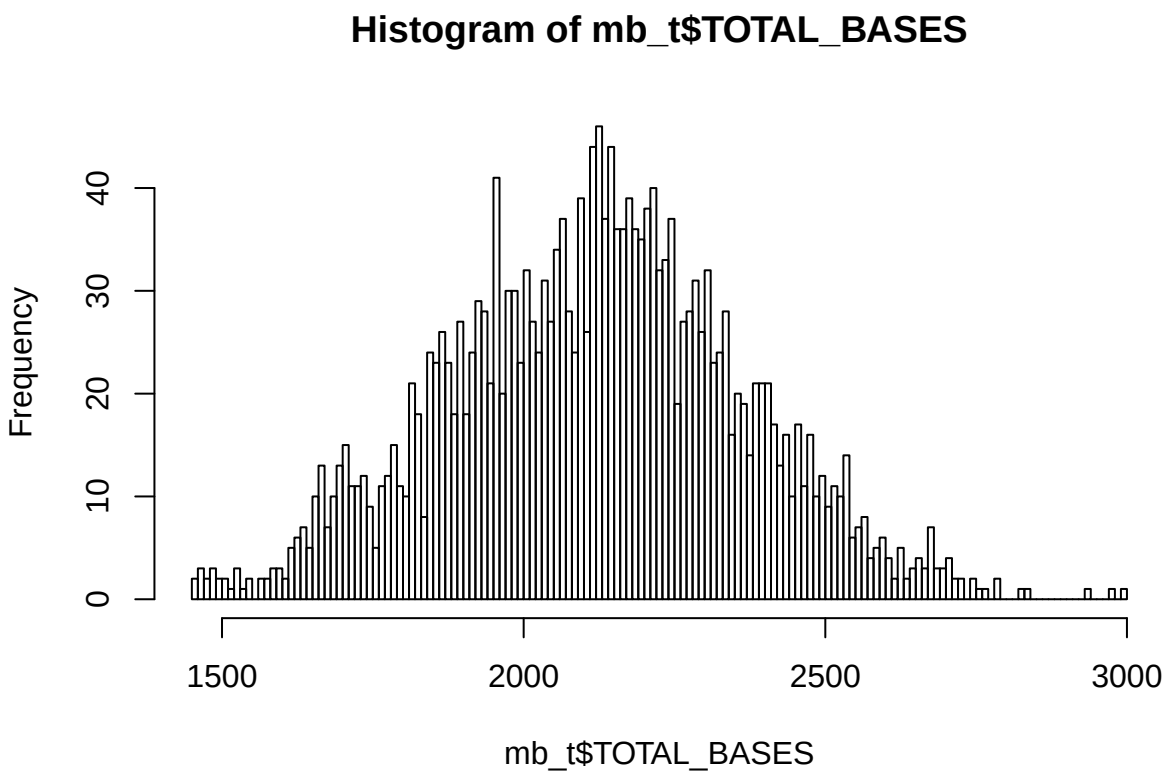
Model 2: Total Bases

```
library(car)

mb_clean <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clean
```

Build a model with Total Bases added and all of the other hitting vars removed

First, create the new variable and discard its components

```
mb_t <- mb_clean

mb_t$TOTAL_BASES <- mb_clean$TEAM_BATTING_1B + (2 * mb_clean$TEAM_BATTING_2B) +
                        (3 * mb_clean$TEAM_BATTING_3B) + (4 * mb_clean$TEAM_BATTING_HR)

# plot histogram to check shape of distribution
par(mfrow = c(1,1))
hist(mb_t$TOTAL_BASES, breaks = 200)
```




```
# now drop 1B, 2B, 3B, HR
mb_tb <- mb_t[,c(1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 15)]

########################################################################
# check correlation with WINS and run simple linear model
cor(mb_tb$TARGET_WINS, mb_tb$TOTAL_BASES)
```

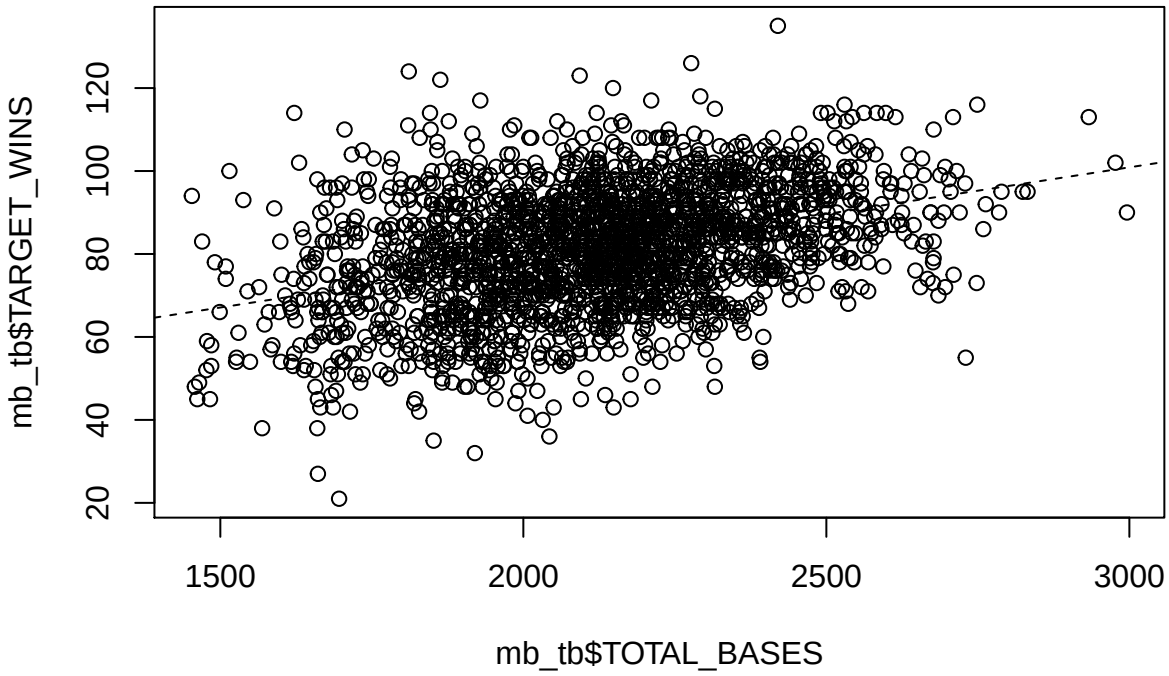
```
## [1] 0.3817283
```

```
mtest <- lm(data=mb_tb, TARGET_WINS ~ TOTAL_BASES)
summary(mtest)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TOTAL_BASES, data = mb_tb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.515  -9.242   0.330   9.134  49.895
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) 33.309929   2.498670   13.33 <0.0000000000000002 ***
## TOTAL_BASES  0.022526   0.001171   19.24 <0.0000000000000002 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.45 on 2170 degrees of freedom
## Multiple R-squared:  0.1457, Adjusted R-squared:  0.1453
## F-statistic: 370.1 on 1 and 2170 DF,  p-value: < 0.00000000000000022
```

```
# shows .381 correlation and Adj R^2 of 0.1453

plot(mb_tb$TARGET_WINS ~ mb_tb$TOTAL_BASES)
abline(lm(mb_tb$TARGET_WINS ~ mb_tb$TOTAL_BASES), lty=2)
```




```
# plot doesn't show unusual relationship
################################################################################
```

Yields r^2= 0.3175, Adj. R^2 = 0.3153, F = 143.8

```
# fit model
model <- lm(data=mb_tb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb_tb)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -41.709  -8.023  -0.060   7.601  53.946
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      39.488500   4.028435   9.802 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.150859   0.016604   9.085 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.068106   0.011085  -6.144  0.00000000095530289 ***
## TEAM_BASERUN_SB   0.066840   0.004931  13.554 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.017824   0.003839   4.643  0.00000363574932803 ***
## TEAM_PITCHING_BB -0.113929   0.014384  -7.921  0.00000000000000375 ***
## TEAM_PITCHING_SO  0.051088   0.009671   5.283  0.00000014005177514 ***
## TEAM_FIELDING_E  -0.063746   0.003731 -17.084 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.122049   0.013247  -9.213 < 0.0000000000000002 ***
## TOTAL_BASES       0.014634   0.002346   6.239  0.00000000052927091 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.87 on 2162 degrees of freedom
## Multiple R-squared:  0.3368, Adjusted R-squared:  0.3341
## F-statistic:   122 on 9 and 2162 DF,  p-value: < 0.00000000000000022
```

```
# All p-values < .05 so check collinearity
vif(model)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        42.583711        96.685910         3.217711        14.833674
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        36.467493        71.783178         4.724229         2.055734
##      TOTAL_BASES
##         5.151420
```

```
# vif indicates remove TEAM_BATTING_SO or TEAM_PITCHING_SO. CHoose PITCHING_SO as in other models

# ----------------------
# remove TEAM_PITCHING_SO
model.2 <- lm(data=mb_tb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb_tb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.524  -7.899   0.111   7.765  53.216
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      32.957133   3.857791   8.543 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.099826   0.013589   7.346    0.000000000000287 ***
## TEAM_BATTING_SO  -0.010106   0.001537  -6.576    0.000000000060537 ***
## TEAM_BASERUN_SB   0.060228   0.004799  12.549 < 0.0000000000000002 ***
```

```
## TEAM_PITCHING_H   0.027950   0.003347   8.352 < 0.0000000000000002 ***
## TEAM_PITCHING_BB -0.067548   0.011464  -5.892    0.000000004411900 ***
## TEAM_FIELDING_E  -0.061895   0.003738 -16.559 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.118816   0.013315  -8.923 < 0.0000000000000002 ***
## TOTAL_BASES       0.009245   0.002125   4.350    0.000014249957148 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.94 on 2163 degrees of freedom
## Multiple R-squared:  0.3283, Adjusted R-squared:  0.3258
## F-statistic: 132.1 on 8 and 2163 DF,  p-value: < 0.00000000000000022
```

```
# All p-values < .05 so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        28.169559         1.835552         3.010433        11.135442
## TEAM_PITCHING_BB  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##        22.880704         4.682562         2.051346         4.177202
```

```
# vif indicates remove TEAM_PITCHING_BB

# ----------------------
# remove TEAM_PITCHING_BB
model.3 <- lm(data=mb_tb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB,
##     data = mb_tb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.761  -8.220   0.160   7.833  58.011
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      45.087126   3.287882  13.713 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.021880   0.003130   6.990     0.00000000000365 ***
## TEAM_BATTING_SO  -0.013037   0.001465  -8.897 < 0.0000000000000002 ***
## TEAM_BASERUN_SB   0.060536   0.004836  12.517 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.011527   0.001867   6.175     0.00000000078674 ***
## TEAM_FIELDING_E  -0.059620   0.003747 -15.912 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.114299   0.013396  -8.532 < 0.0000000000000002 ***
## TOTAL_BASES       0.017639   0.001590  11.097 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.04 on 2164 degrees of freedom
## Multiple R-squared:  0.3175, Adjusted R-squared:  0.3153
## F-statistic: 143.8 on 7 and 2164 DF,  p-value: < 0.00000000000000022
```

```r
# All p-values < .05 so check collinearity
vif(model.3)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##         1.471732         1.643149         3.010077         3.411334
##  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##         4.632607         2.044547         2.300575
```

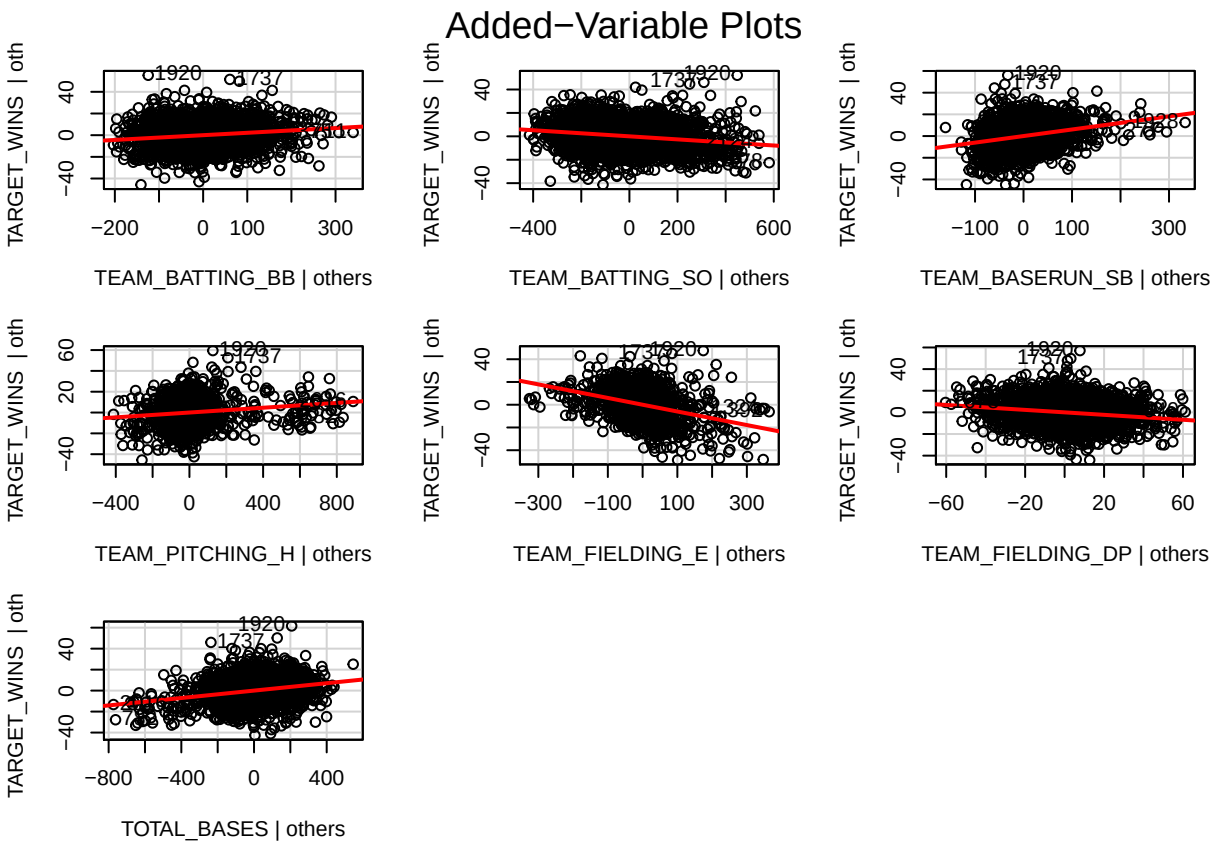
```r
# no further collinearity issues so STOP

# check 95% confidence intervals for coefficients
confint(model.3)
```

```
##                          2.5 %      97.5 %
## (Intercept)      38.639390134 51.53486272
## TEAM_BATTING_BB   0.015741199  0.02801786
## TEAM_BATTING_SO  -0.015910936 -0.01016372
## TEAM_BASERUN_SB   0.051051063  0.07002015
## TEAM_PITCHING_H   0.007866465  0.01518776
## TEAM_FIELDING_E  -0.066967986 -0.05227251
## TEAM_FIELDING_DP -0.140569909 -0.08802864
## TOTAL_BASES       0.014522018  0.02075646
```

Diagnostics

Plots for Fielding_E shows skew. Others are pretty linear

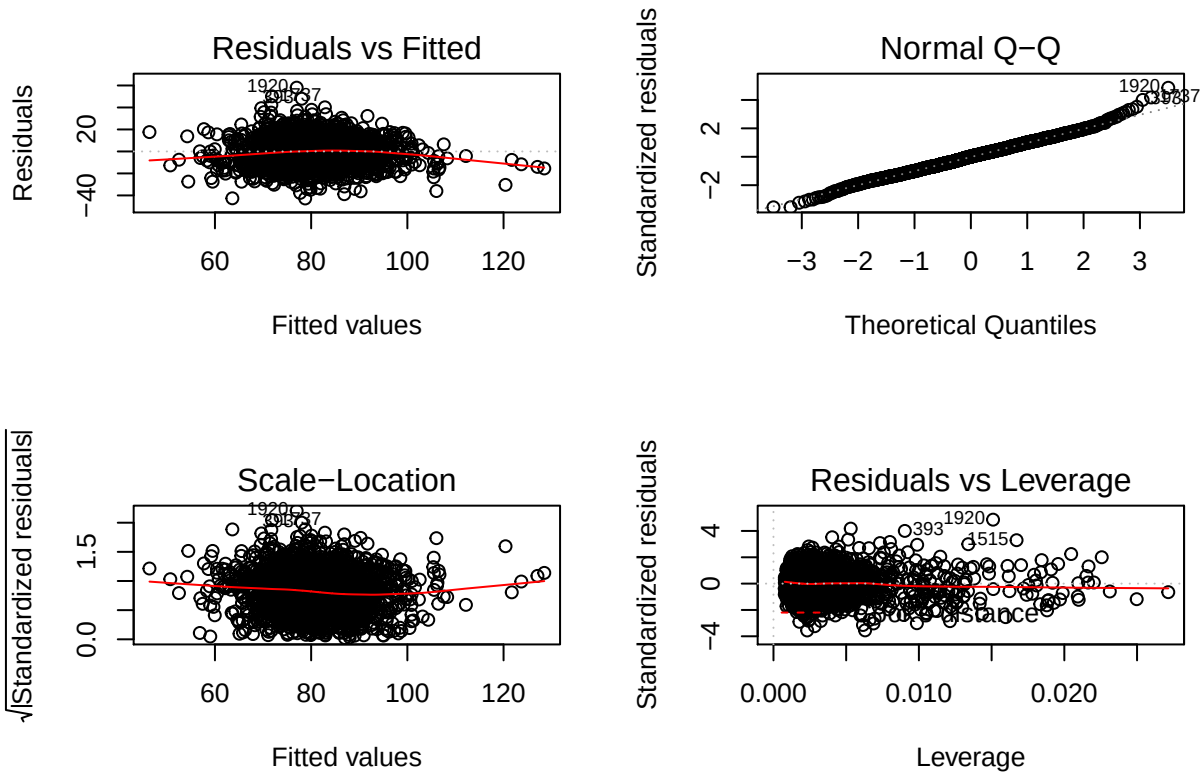
```r
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.3, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots look good except for outliers. Lack of Constant variability in Resid vs. Fitted at very large values of Yhat; normal distribution of residuals except for outliers, most residuals within 2 std dev and well within Cook's distance

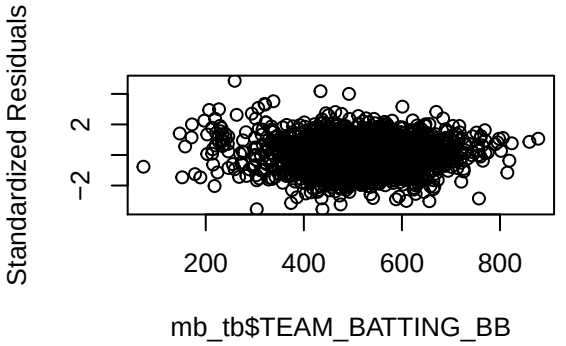
```
#Figure 5.6 on page 129 MARR text
par(mfrow=c(2,2))
plot(model.3)
```

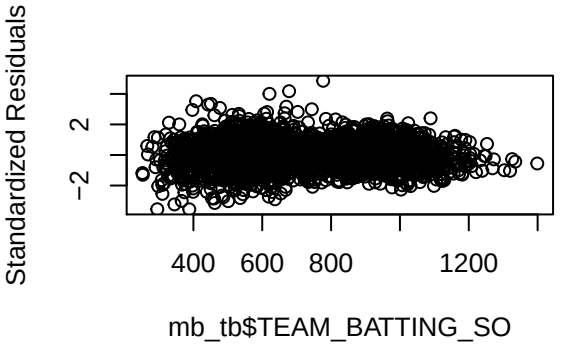


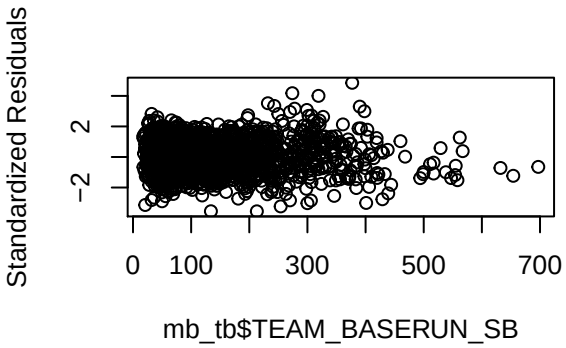
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

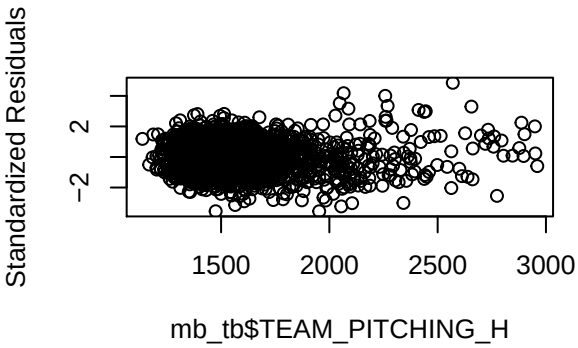
Results show lack of constant variability for BASERUN_SB, PITCHING_H, PITCHING_SO, FIELDING_E, FIELDING_DP

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.3)
par(mfrow=c(2,2))

plot(mb_tb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb_tb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb_tb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb_tb$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
```

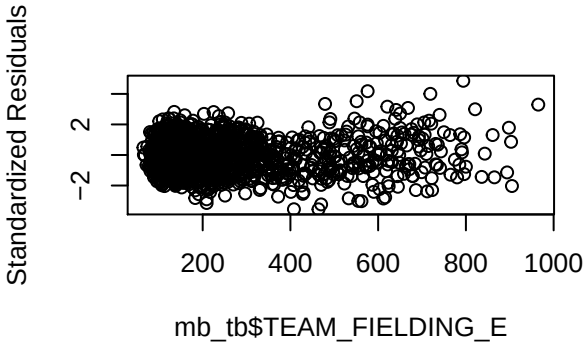


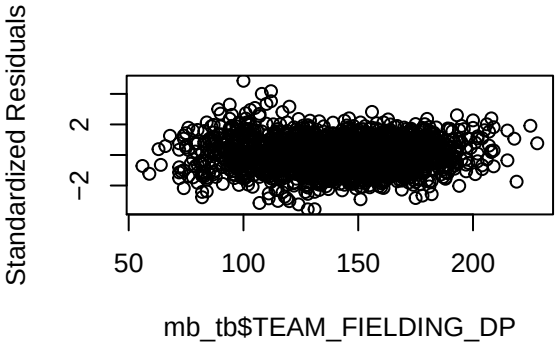


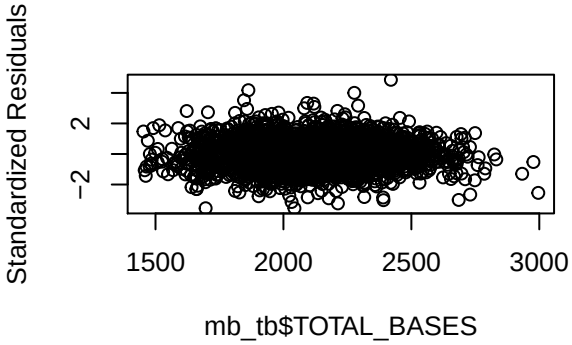





```
plot(mb_tb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb_tb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb_tb$TOTAL_BASES, StanRes1, ylab="Standardized Residuals")
```



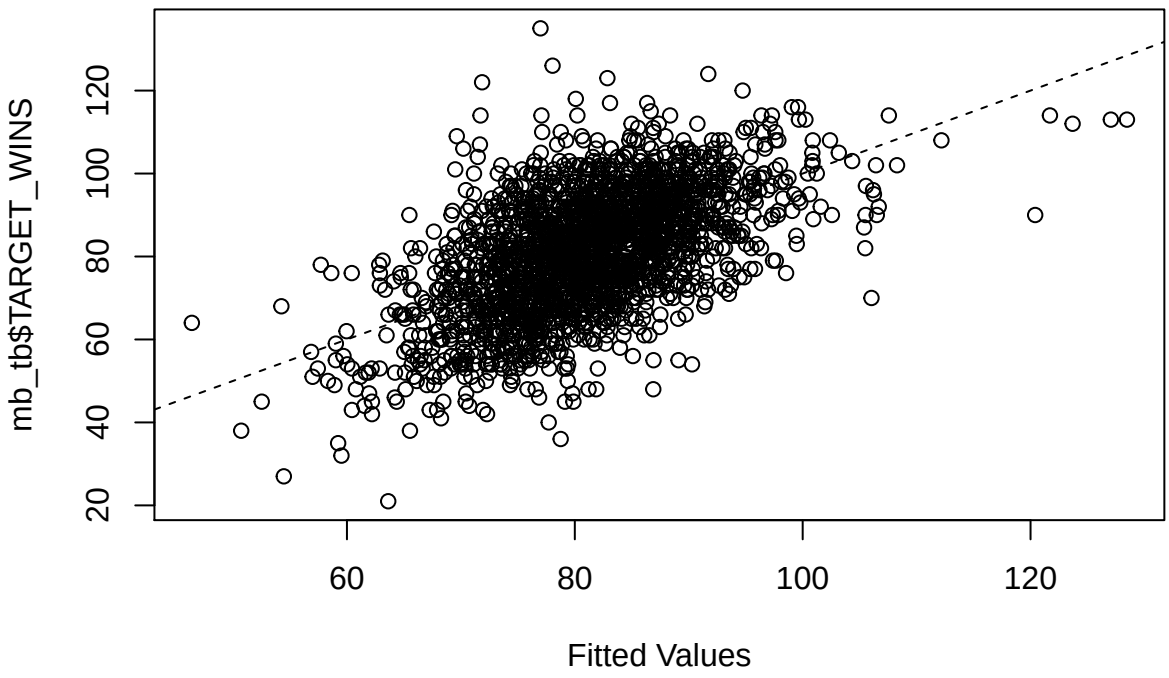




PLOT Y AGAINST FITTED VALUES

Might be some skew due to outliers

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.3$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb_tb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb_tb$TARGET_WINS),lty=2)
```



REMOVE OUTLIERS AND REFIT

Per Cooks Distance, remove items 1920, 1737, 393, 1515

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem <- mb_tb[-c(1920, 1737, 393, 1515),]

# renumber rows
rownames(mb_rem) <- 1:nrow(mb_rem)
```

Now refit first model from above: all variables

Yields r^2= 0.3287, Adj r^2 = 0.3265, F = 151.1

```
# keep the clean data set pure
mb <- mb_rem

# use p-value elimination
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
```

```
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.516  -8.071   0.049   7.522  47.810
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      42.441249   3.987025  10.645 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.154335   0.016410   9.405 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.072357   0.010950  -6.608 0.00000000048869495 ***
## TEAM_BASERUN_SB   0.069009   0.004860  14.199 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.016802   0.003788   4.435 0.000009658905673537 ***
## TEAM_PITCHING_BB -0.117154   0.014229  -8.234 0.00000000000000311 ***
## TEAM_PITCHING_SO  0.053942   0.009551   5.648 0.00000018371250928 ***
## TEAM_FIELDING_E  -0.067499   0.003699 -18.248 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.123715   0.013030  -9.495 < 0.0000000000000002 ***
## TOTAL_BASES       0.014732   0.002311   6.375 0.00000000222616213 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.67 on 2158 degrees of freedom
## Multiple R-squared:  0.3493, Adjusted R-squared:  0.3466
## F-statistic: 128.7 on 9 and 2158 DF,  p-value: < 0.00000000000000022
```

```
# pvals all < .05 so check collinearity

vif(model)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        42.781793        97.442002         3.199402        14.634840
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        36.800848        72.280509         4.670280         2.048269
##      TOTAL_BASES
##         5.161481
```

```
# vif indicates remove TEAM_BATTING_SO and PITCHING_SO. CHoose PITCHING_SO again

# --------------------
# remove TEAM_PITCHING_SO
model.2 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.867  -7.942   0.093   7.845  44.545
##
## Coefficients:
```

```
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      35.499141   3.819886   9.293 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.100306   0.013428   7.470    0.000000000000116 ***
## TEAM_BATTING_SO  -0.011103   0.001519  -7.308    0.000000000000379 ***
## TEAM_BASERUN_SB   0.061945   0.004730  13.097 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.027469   0.003307   8.305 < 0.0000000000000002 ***
## TEAM_PITCHING_BB -0.068029   0.011341  -5.998    0.000000002329868 ***
## TEAM_FIELDING_E  -0.065403   0.003707 -17.645 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.120236   0.013108  -9.173 < 0.0000000000000002 ***
## TOTAL_BASES       0.009060   0.002096   4.323    0.000016126085257 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.76 on 2159 degrees of freedom
## Multiple R-squared:  0.3397, Adjusted R-squared:  0.3373
## F-statistic: 138.8 on 8 and 2159 DF,  p-value: < 0.00000000000000022
```

```
# p-values are OK so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        28.243542         1.849306         2.987512        10.997742
## TEAM_PITCHING_BB  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##        23.050031         4.623291         2.043691         4.186652
```

```
# vif says remove TEAM_BATTING_BB

# -------------------
#eliminate TEAM_BATTING_BB or PITCHING_BB so choose PITCHING_BB
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_SO)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_SO,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.456  -8.196   0.174   7.785  43.234
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      47.746708   3.254440  14.671 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.021878   0.003085   7.092     0.00000000000178 ***
## TEAM_BATTING_SO  -0.014068   0.001448  -9.715 < 0.0000000000000002 ***
## TEAM_BASERUN_SB   0.062245   0.004768  13.055 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.010923   0.001840   5.937     0.00000000336791 ***
## TEAM_FIELDING_E  -0.063123   0.003717 -16.983 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.115737   0.013192  -8.773 < 0.0000000000000002 ***
## TOTAL_BASES       0.017495   0.001567  11.166 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 11.85 on 2160 degrees of freedom
## Multiple R-squared:  0.3287, Adjusted R-squared:  0.3265
## F-statistic: 151.1 on 7 and 2160 DF,  p-value: < 0.00000000000000022
```

```
# p-values OK so check collinearity
vif(model.3)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##         1.466753         1.653485         2.987180         3.348759
##  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##         4.574645         2.036999         2.302144
```

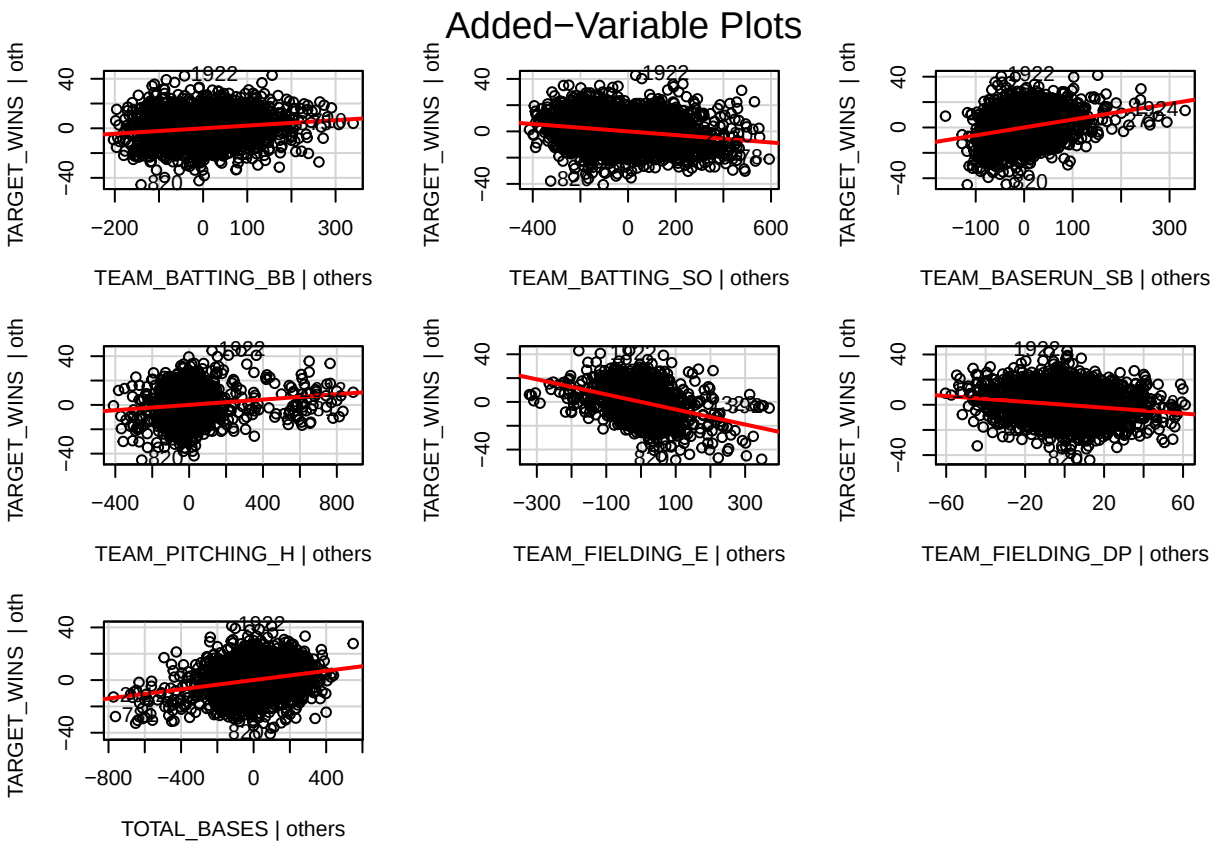
```
# vif and pvals OK so STOP HERE

# check 95% confidence intervals for coefficients
confint(model.3)
```

```
##                        2.5 %      97.5 %
## (Intercept)      41.364546361 54.12886979
## TEAM_BATTING_BB   0.015828537  0.02792739
## TEAM_BATTING_SO  -0.016907860 -0.01122823
## TEAM_BASERUN_SB   0.052894909  0.07159443
## TEAM_PITCHING_H   0.007315473  0.01453144
## TEAM_FIELDING_E  -0.070411557 -0.05583395
## TEAM_FIELDING_DP -0.141607287 -0.08986612
## TOTAL_BASES       0.014422320  0.02056738
```

Diagnostics

Plots for Fielding_E shows skew. Others are pretty linear

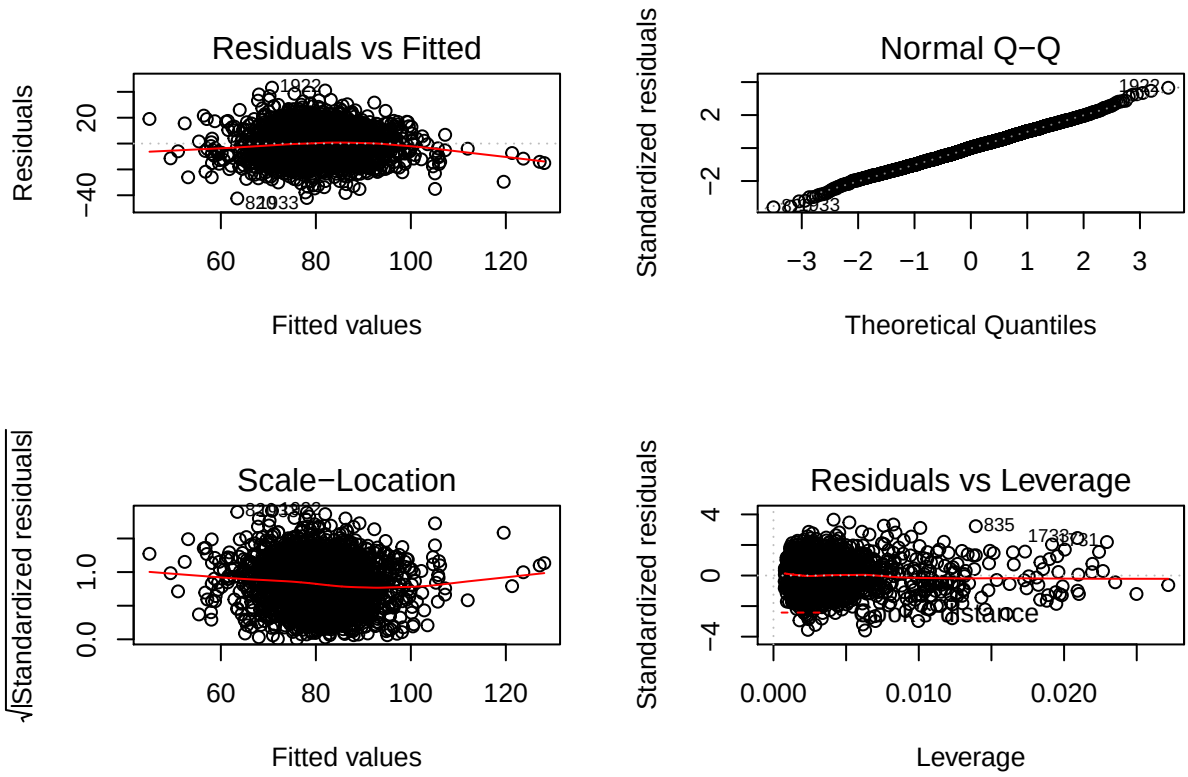
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.3, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Lack of constant variability in Resid vs. Fitted. Normal QQ shows a bit of skew in upper right end but not drastic; Some residuals appear to be outside 2 std devs. **Might not be a good model**. Outliers at 1528, 1922, 820, 1933, 1733, 835

```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.3)
```



Plots show outliers so remove them and re-fit

Per Cooks Distance, remove 1528, 1922, 820, 1933, 1733, 835

```
############ SECOND SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem2 <- mb[-c(1528, 1922, 820, 1933, 1733, 835),]

# renumber rows
rownames(mb_rem2) <- 1:nrow(mb_rem2)
```

Model using all remaining variables as a starting point

Yields r^2= 0.33655, Adj r^2 = 0.3343, F = 156

```
# keep the clean data set pure
mb <- mb_rem2

# use p-value elimination
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
```

```
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.179  -8.045   0.044   7.493  44.739
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      43.011185   3.992399  10.773 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.147363   0.016384   8.994 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.068624   0.010941  -6.272   0.0000000004293871 ***
## TEAM_BASERUN_SB   0.069068   0.004809  14.363 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.016311   0.003801   4.291   0.0000185345938702 ***
## TEAM_PITCHING_BB -0.110736   0.014225  -7.784   0.0000000000000108 ***
## TEAM_PITCHING_SO  0.049930   0.009554   5.226   0.0000001900846778 ***
## TEAM_FIELDING_E  -0.067668   0.003665 -18.461 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.123738   0.012888  -9.601 < 0.0000000000000002 ***
## TOTAL_BASES       0.015058   0.002300   6.546   0.0000000000734869 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.54 on 2152 degrees of freedom
## Multiple R-squared:  0.3547, Adjusted R-squared:  0.352
## F-statistic: 131.4 on 9 and 2152 DF,  p-value: < 0.00000000000000022
```

```
# p-values all < .05 so check collinearity

vif(model)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        43.067052        98.922118         3.175291        14.605057
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        37.353473        73.550826         4.602276         2.044406
##      TOTAL_BASES
##         5.191730
```

```
# vif says remove TEAM_BATTING_SO

# --------------------
#eliminate TEAM_BATTING_SO or PITCHING_SO so choose PITCHING_SO again
model.2 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.995  -7.977   0.065   7.750  42.855
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
```

```
## (Intercept)      36.541180   3.818699   9.569 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.097250   0.013365   7.276  0.00000000000047808 ***
## TEAM_BATTING_SO  -0.011989   0.001512  -7.928  0.00000000000000353 ***
## TEAM_BASERUN_SB   0.062616   0.004676  13.391 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.026191   0.003317   7.895  0.00000000000000459 ***
## TEAM_PITCHING_BB -0.065120   0.011301  -5.762  0.00000000948828590 ***
## TEAM_FIELDING_E  -0.065835   0.003671 -17.935 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.120626   0.012953  -9.313 < 0.0000000000000002 ***
## TOTAL_BASES       0.009848   0.002086   4.722  0.00000248533829969 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.61 on 2153 degrees of freedom
## Multiple R-squared:  0.3465, Adjusted R-squared:  0.3441
## F-statistic: 142.7 on 8 and 2153 DF,  p-value: < 0.00000000000000022
```

```
# p-values OK so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##        28.313487         1.866693         2.965972        10.991773
## TEAM_PITCHING_BB  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##        23.288881         4.560154         2.040042         4.216460
```

```
# vif says remove TEAM_BATTING_BB or PITCHING_BB so choose PITCHING_BB again

# --------------------
#eliminate TEAM_PITCHING_BB
model.3 <- lm(data=mb, TARGET_WINS ~. - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_SO - TEAM_PITCHING_BB,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.210  -8.207   0.091   7.683  41.423
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      48.485759   3.231012  15.006 < 0.0000000000000002 ***
## TEAM_BATTING_BB   0.022237   0.003050   7.291    0.000000000000431 ***
## TEAM_BATTING_SO  -0.014906   0.001436 -10.384 < 0.0000000000000002 ***
## TEAM_BASERUN_SB   0.062907   0.004711  13.355 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.010197   0.001831   5.571    0.000000028565450 ***
## TEAM_FIELDING_E  -0.063675   0.003679 -17.308 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.116577   0.013030  -8.947 < 0.0000000000000002 ***
## TOTAL_BASES       0.017963   0.001550  11.591 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.7 on 2154 degrees of freedom
## Multiple R-squared:  0.3365, Adjusted R-squared:  0.3343
## F-statistic:   156 on 7 and 2154 DF,  p-value: < 0.00000000000000022
```

```
vif(model.3)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##         1.452828         1.657492         2.965626         3.297456
##  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##         4.512577         2.034037         2.293841
```

```
# p-values and VIF OK so STOP

anova(model.3)
```

```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq F value                Pr(>F)
## TEAM_BATTING_BB     1  30649   30649 223.999 < 0.00000000000000022 ***
## TEAM_BATTING_SO     1   7298    7298  53.341  0.00000000000392795 ***
## TEAM_BASERUN_SB     1   9655    9655  70.563 < 0.00000000000000022 ***
## TEAM_PITCHING_H     1  12282   12282  89.763 < 0.00000000000000022 ***
## TEAM_FIELDING_E     1  62709   62709 458.310 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1   8464    8464  61.863  0.00000000000005775 ***
## TOTAL_BASES         1  18384   18384 134.357 < 0.00000000000000022 ***
## Residuals        2154 294726     137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

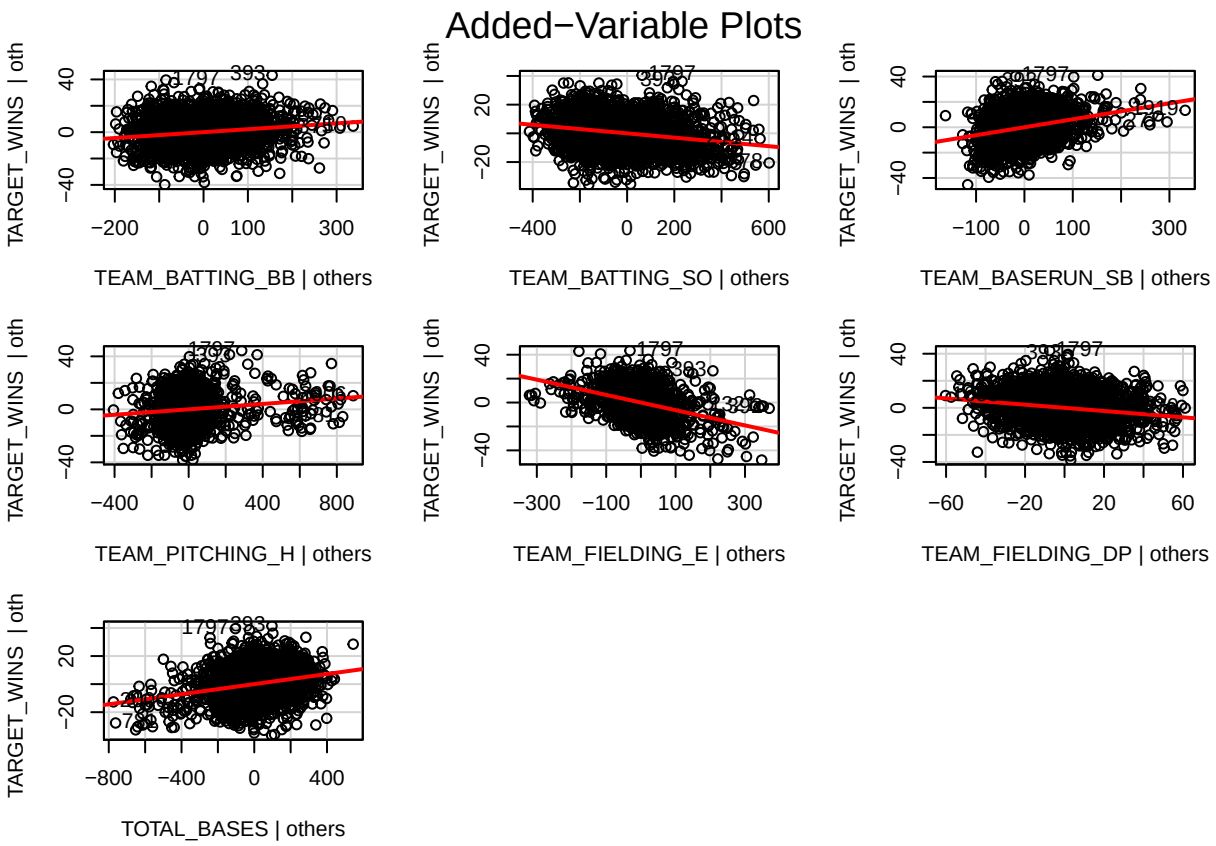
```
# check 95% confidence intervals for coefficients
confint(model.3)
```

```
##                        2.5 %      97.5 %
## (Intercept)      42.149531058 54.82198731
## TEAM_BATTING_BB   0.016256026  0.02821876
## TEAM_BATTING_SO  -0.017720942 -0.01209071
## TEAM_BASERUN_SB   0.053669147  0.07214436
## TEAM_PITCHING_H   0.006607378  0.01378706
## TEAM_FIELDING_E  -0.070888990 -0.05646022
## TEAM_FIELDING_DP -0.142128959 -0.09102440
## TOTAL_BASES       0.014924337  0.02100264
```

Diagnostics

Plots for Fielding_E shows skew. Others are pretty linear

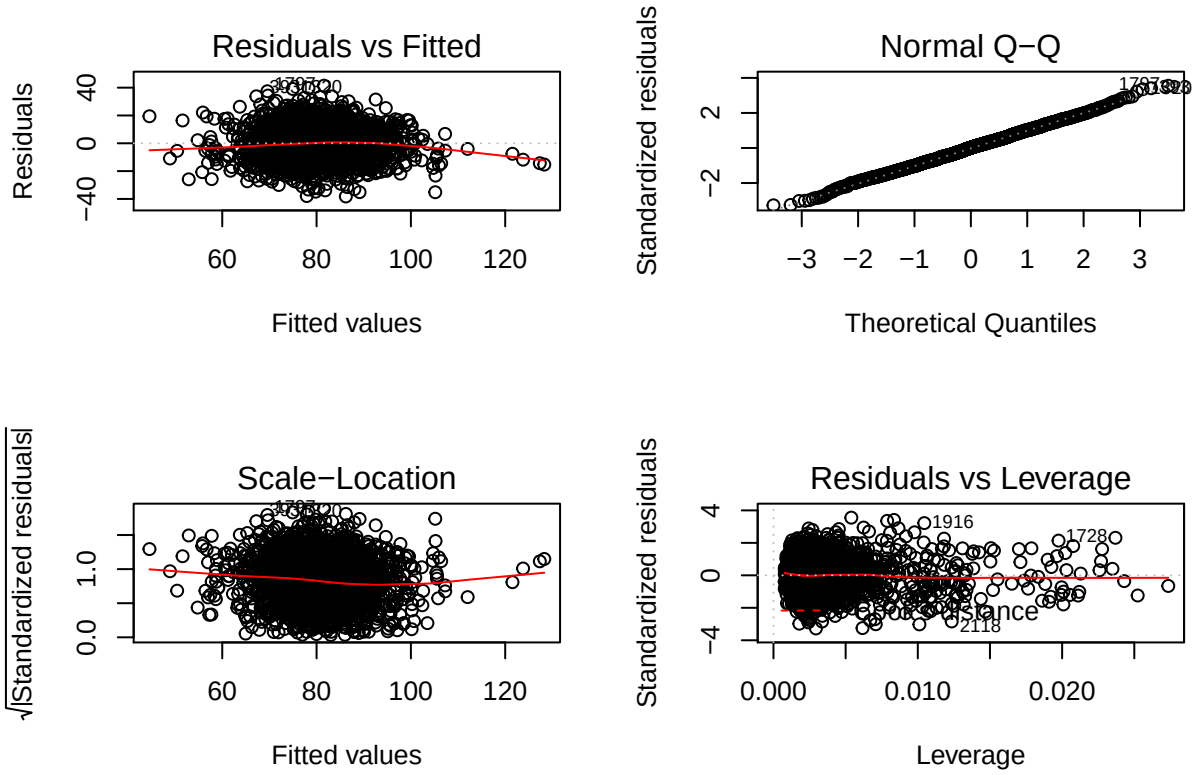
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.3, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Lack of constant variability in Resid vs. Fitted but only for extreme outliers. Normal QQ looks very good; Some Residuals appear to be outside 2 std devs

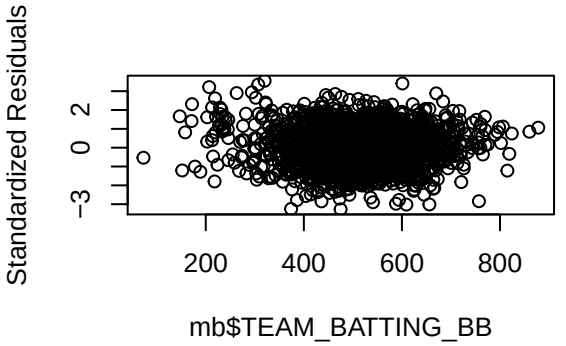
```
# plot summary residual plots
par(mfrow=c(2,2))
plot(model.3)
```

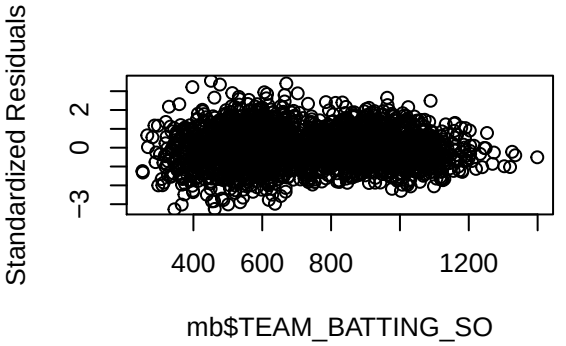


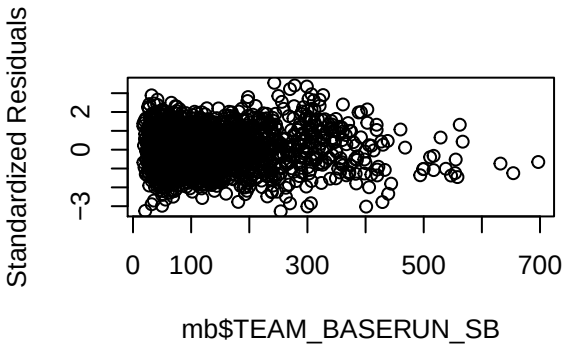
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

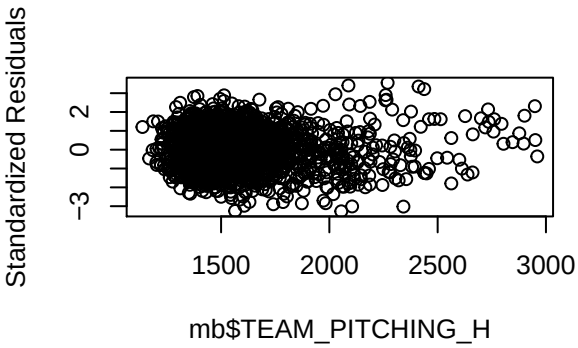
Results show lack of constant variability for BASERUN_SB, PITCHING_H, PITCHING_SO, FIELDING_E, FIELDING_DP

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.3)
par(mfrow=c(2,2))

plot(mb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
```

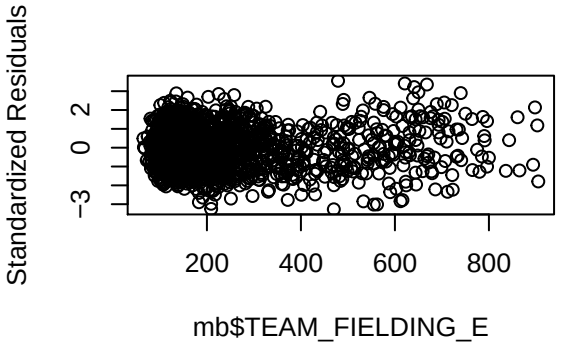


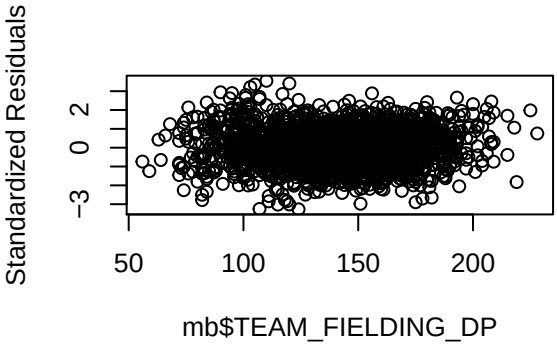


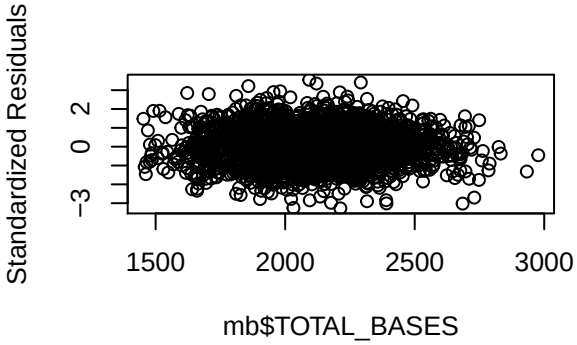





```
plot(mb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb$TOTAL_BASES, StanRes1, ylab="Standardized Residuals")
```



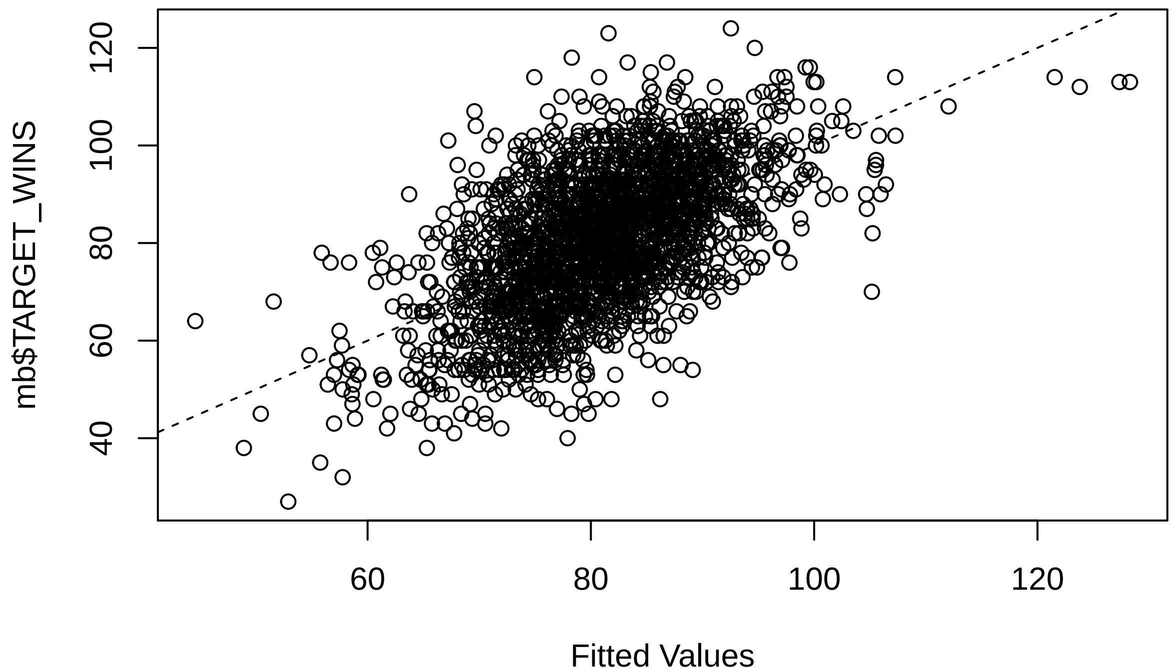




PLOT Y AGAINST FITTED VALUES

Might be some skew due to outliers

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.3$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb$TARGET_WINS),lty=2)
```



——————————————————————————— # ——————————————————————————

Now try same model but with FIELD_E transformed according to Box-Cox

```
# TEAM_FIELDING_E: Box-Cox yields power xform of -1 => 1/y

mb$TEAM_FIELDING_E <- 1/mb$TEAM_FIELDING_E
```

Now refit first model from above: all variables

Model using all remaining variables as a starting point

Yields r^2= 0.3048, Adj r^2 = 0.3029, F = 157.5

```
# fit model
model <- lm(data=mb, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.449  -7.663   0.017   7.875  44.521
##
## Coefficients:
##                     Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)        33.940450    4.062200   8.355 < 0.0000000000000002 ***
## TEAM_BATTING_BB     0.109123    0.016792   6.499    0.000000000100466 ***
## TEAM_BATTING_SO    -0.053628    0.011230  -4.775    0.000001914819121 ***
## TEAM_BASERUN_SB     0.040792    0.004460   9.146 < 0.0000000000000002 ***
## TEAM_PITCHING_H     0.004967    0.003821   1.300              0.19381
## TEAM_PITCHING_BB   -0.074106    0.014533  -5.099    0.000000370699407 ***
## TEAM_PITCHING_SO    0.027384    0.009820   2.789              0.00534 **
## TEAM_FIELDING_E  2695.772954  195.354383  13.799 < 0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.117721    0.013328  -8.833 < 0.0000000000000002 ***
## TOTAL_BASES         0.017392    0.002373   7.328    0.000000000000329 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.91 on 2152 degrees of freedom
## Multiple R-squared:  0.3133, Adjusted R-squared:  0.3104
## F-statistic: 109.1 on 9 and 2152 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_H

# ----------------------
# remove TEAM_PITCHING_H
model.2 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.790  -7.818   0.083   7.925  45.364
##
## Coefficients:
##                     Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)        37.568737    2.951626  12.728 < 0.0000000000000002 ***
## TEAM_BATTING_BB     0.103043    0.016130   6.388    0.000000000020485 ***
## TEAM_BATTING_SO    -0.062243    0.009067  -6.865    0.000000000000868 ***
## TEAM_BASERUN_SB     0.042673    0.004219  10.113 < 0.0000000000000002 ***
## TEAM_PITCHING_BB   -0.069557    0.014107  -4.931    0.000000882250333 ***
## TEAM_PITCHING_SO    0.034309    0.008250   4.159    0.000033285114111 ***
## TEAM_FIELDING_E  2664.401602  193.888720  13.742 < 0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.116848    0.013313  -8.777 < 0.0000000000000002 ***
## TOTAL_BASES         0.020029    0.001231  16.267 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.91 on 2153 degrees of freedom
## Multiple R-squared:  0.3128, Adjusted R-squared:  0.3102
```

```
## F-statistic: 122.5 on 8 and 2153 DF,  p-value: < 0.00000000000000022
```

```
vif(model.2)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_PITCHING_BB
##        39.211674        63.814882         2.296332        34.506646
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##        51.516155         3.631039         2.049298         1.397321
```

```
# vif says remove TEAM_BATTING_SO or PITCHING_SO so discard PITCHING_SO again

# ----------------------
# remove TEAM_PITCHING_SO
model.3 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO,
##     data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.511  -7.762   0.113   8.049  44.351
##
## Coefficients:
##                     Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)        39.139798    2.938399  13.320 < 0.0000000000000002 ***
## TEAM_BATTING_BB     0.042183    0.006807   6.197       0.000000000688 ***
## TEAM_BATTING_SO    -0.025194    0.001690 -14.906 < 0.0000000000000002 ***
## TEAM_BASERUN_SB     0.041624    0.004228   9.845 < 0.0000000000000002 ***
## TEAM_PITCHING_BB   -0.016370    0.005975  -2.740               0.0062 **
## TEAM_FIELDING_E  2640.049236  194.531863  13.571 < 0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.113486    0.013339  -8.508 < 0.0000000000000002 ***
## TOTAL_BASES         0.019976    0.001236  16.164 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.95 on 2154 degrees of freedom
## Multiple R-squared:  0.3072, Adjusted R-squared:  0.305
## F-statistic: 136.5 on 7 and 2154 DF,  p-value: < 0.00000000000000022
```

```
vif(model.3)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_PITCHING_BB
##         6.931405         2.200940         2.288133         6.142926
##  TEAM_FIELDING_E TEAM_FIELDING_DP      TOTAL_BASES
##         3.627727         2.041739         1.397168
```

```
# vif say remove TEAM_BATTING_BB or PITCHING_BB - go with PITCHING_BB again

# ----------------------
```

```r
# remove TEAM_PITCHING_BB
model.4 <- lm(data=mb, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO - TEAM_PITCHING_BB)
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB, data = mb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.763  -7.899   0.209   8.121  43.323
##
## Coefficients:
##                     Estimate  Std. Error t value            Pr(>|t|)
## (Intercept)        39.164119    2.942818  13.308 <0.0000000000000002 ***
## TEAM_BATTING_BB     0.025404    0.002976   8.535 <0.0000000000000002 ***
## TEAM_BATTING_SO    -0.024551    0.001676 -14.645 <0.0000000000000002 ***
## TEAM_BASERUN_SB     0.038058    0.004028   9.447 <0.0000000000000002 ***
## TEAM_FIELDING_E  2714.549336  192.912713  14.071 <0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.114953    0.013348  -8.612 <0.0000000000000002 ***
## TOTAL_BASES         0.019678    0.001233  15.961 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.97 on 2155 degrees of freedom
## Multiple R-squared:  0.3048, Adjusted R-squared:  0.3029
## F-statistic: 157.5 on 6 and 2155 DF,  p-value: < 0.00000000000000022
```

```r
# pvals all < .05 so check collinearity
vif(model.4)
```

```
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_E
##         1.321097         2.158564         2.071297         3.556849
## TEAM_FIELDING_DP      TOTAL_BASES
##         2.038449         1.386329
```

```r
options(scipen=999)
model.4
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB, data = mb)
##
## Coefficients:
##      (Intercept)   TEAM_BATTING_BB   TEAM_BATTING_SO   TEAM_BASERUN_SB
##         39.16412           0.02540          -0.02455           0.03806
##  TEAM_FIELDING_E  TEAM_FIELDING_DP       TOTAL_BASES
##       2714.54934          -0.11495           0.01968
```

```r
anova(model.4)
```

```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq F value                Pr(>F)
## TEAM_BATTING_BB     1  30649   30649 213.905 < 0.00000000000000022 ***
## TEAM_BATTING_SO     1   7298    7298  50.937 0.0000000000012976902 ***
## TEAM_BASERUN_SB     1   9655    9655  67.383 0.0000000000000003804 ***
## TEAM_FIELDING_E     1  46368   46368 323.609 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1   4920    4920  34.334 0.0000000053558270759 ***
## TOTAL_BASES         1  36500   36500 254.739 < 0.00000000000000022 ***
## Residuals        2155 308778     143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

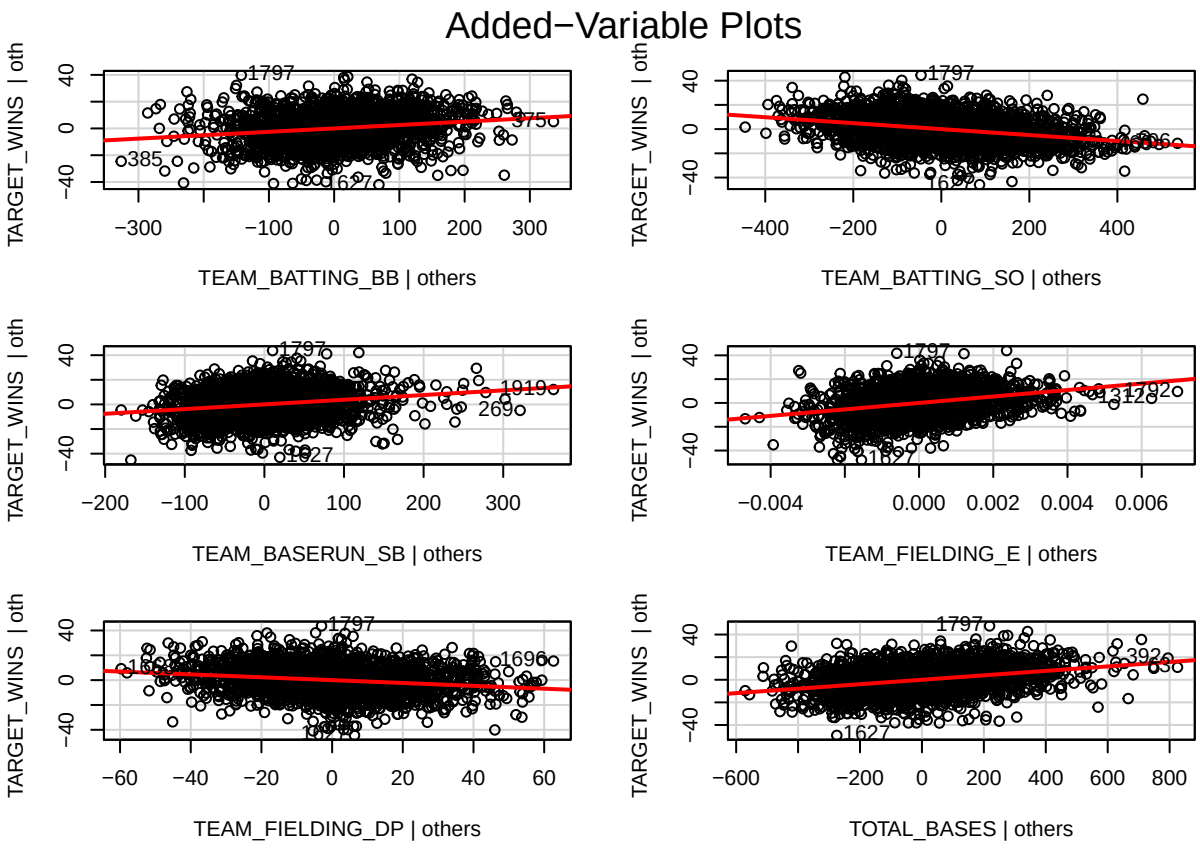
```r
# no collinearity so STOP
# check 95% confidence intervals for coefficients
confint(model.4)
```

```
##                          2.5 %        97.5 %
## (Intercept)        33.39305950   44.93517803
## TEAM_BATTING_BB     0.01956697    0.03124054
## TEAM_BATTING_SO    -0.02783859   -0.02126358
## TEAM_BASERUN_SB     0.03015808    0.04595838
## TEAM_FIELDING_E  2336.23488507 3092.86378594
## TEAM_FIELDING_DP   -0.14112951   -0.08877633
## TOTAL_BASES         0.01725985    0.02209542
```

Diagnostics

Plots show all variables are linear to response

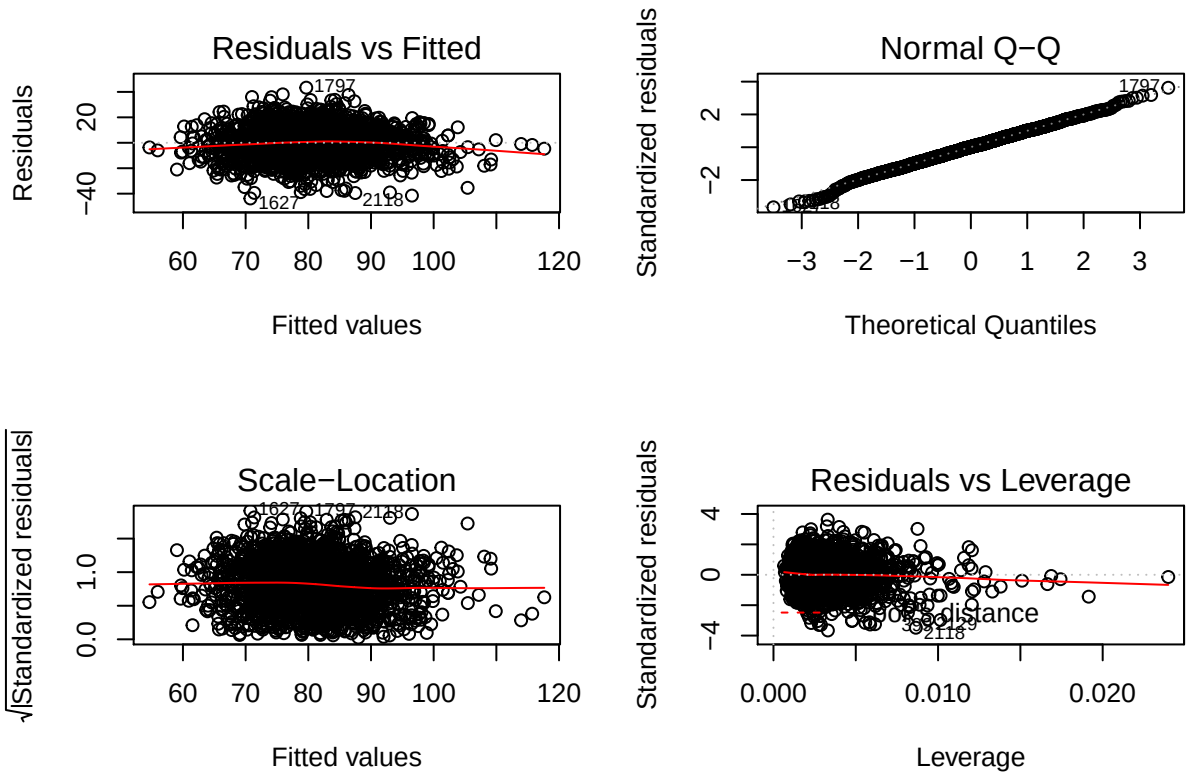
```r
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.4, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Some lack of Constant variability in Resid vs. Fitted at both ends; normal distribution of residuals; many residuals outside of 2 std devs **PROBABLY NOT A GOOD MODEL**

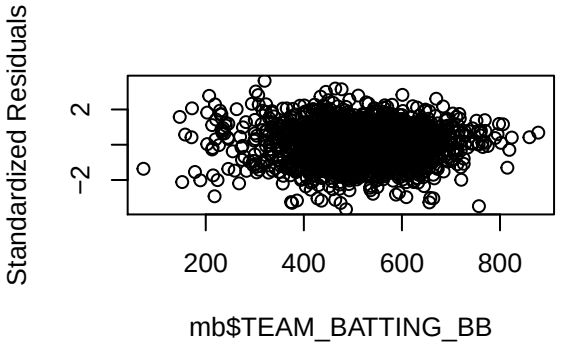
```
#Figure 5.6 on page 129 MARR text
par(mfrow=c(2,2))
plot(model.4)
```

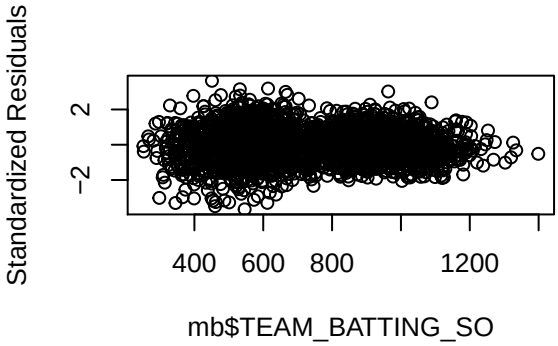


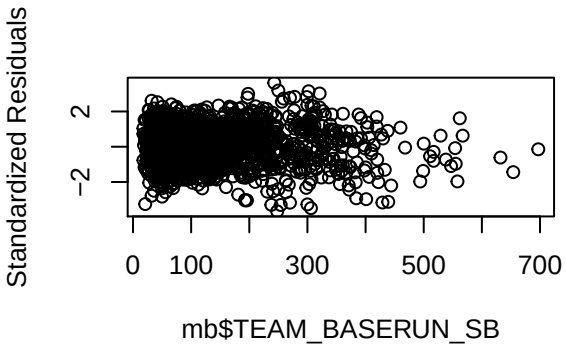
PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

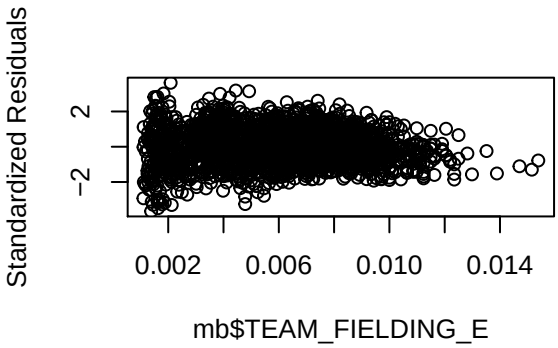
Results show lack of constant variability for PITCHING_SO, FIELDING_E

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.4)
par(mfrow=c(2,2))

plot(mb$TEAM_BATTING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")
plot(mb$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
```

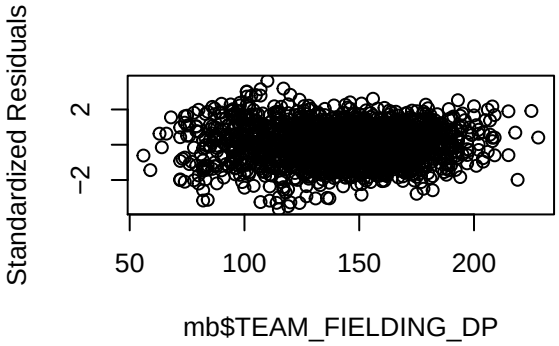


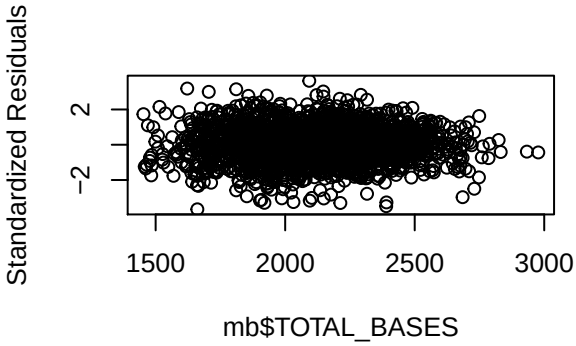







```
plot(mb$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb$TOTAL_BASES, StanRes1, ylab="Standardized Residuals")
```

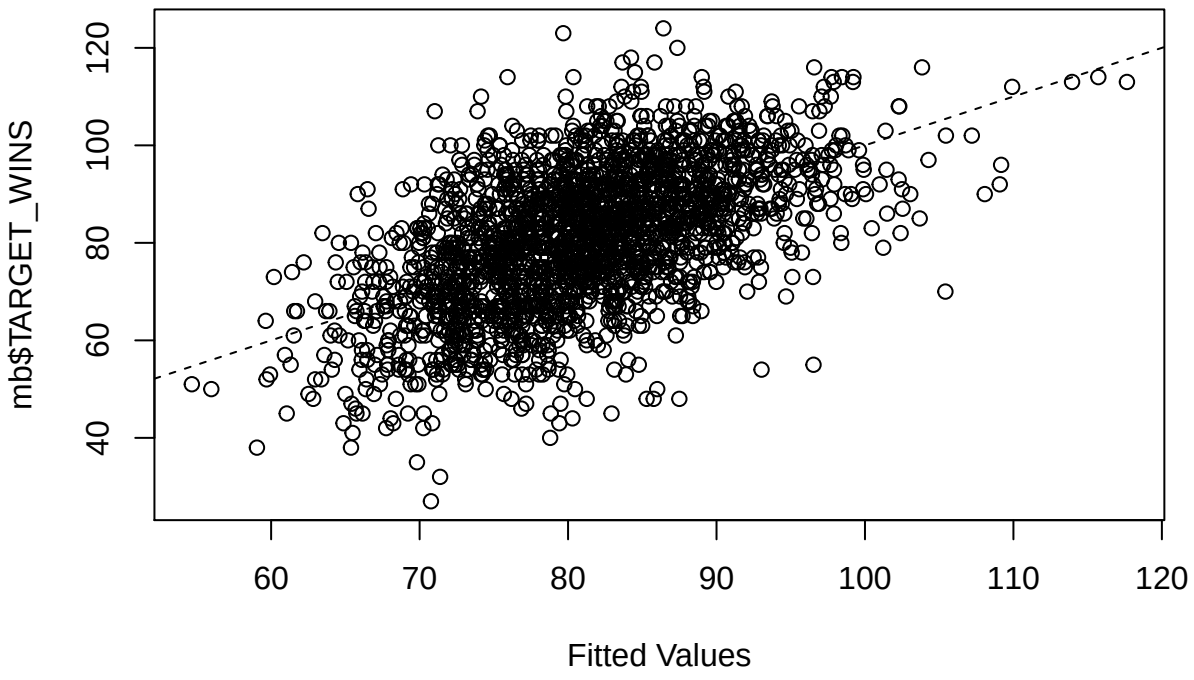




PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship with no pattern or skew

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.4$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb$TARGET_WINS),lty=2)
```




```
# clean up objects in memory
rm(list = ls())
```

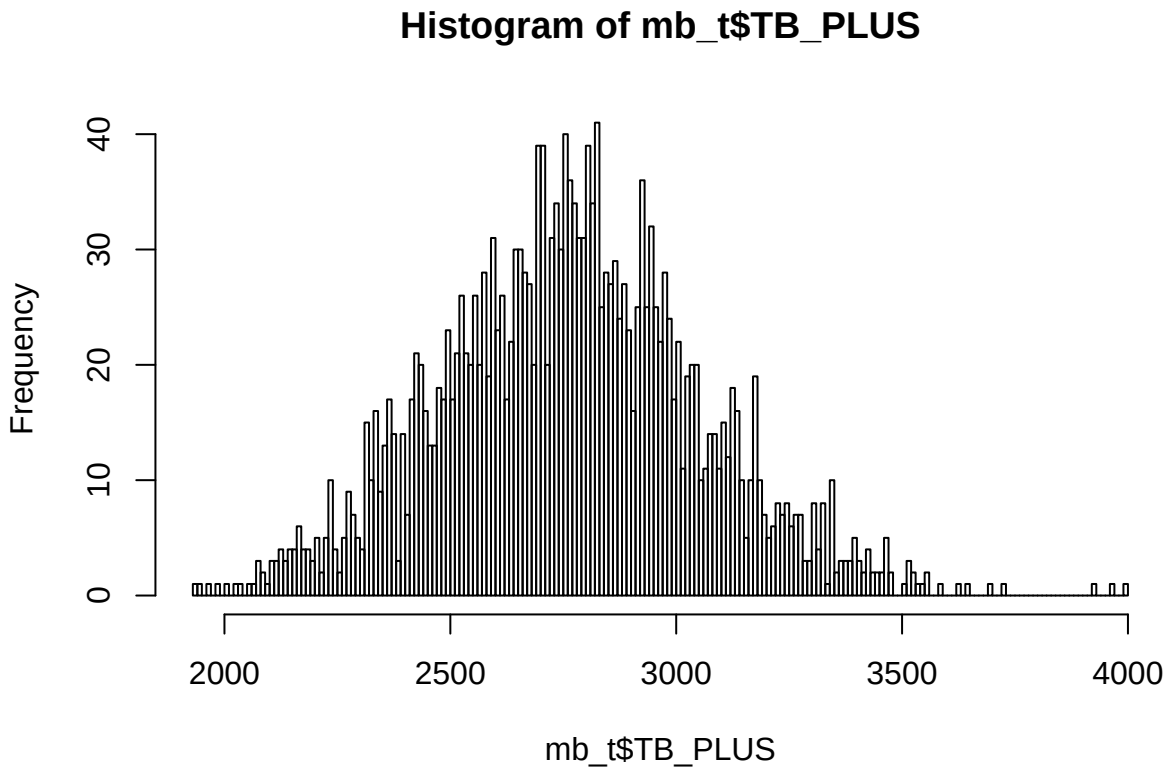
Model 3: Total Bases PLUS

```
library(car)

# read clean data set from Github

mb_clean <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clean
```

Build a model with Total Bases + SB + BB added and all of the other hitting vars removed

```
# create new variable and drop its components
mb_t <- mb_clean

mb_t$TB_PLUS <- mb_clean$TEAM_BATTING_1B + (2 * mb_clean$TEAM_BATTING_2B) +
                        (3 * mb_clean$TEAM_BATTING_3B) + (4 * mb_clean$TEAM_BATTING_HR) +
                        mb_clean$TEAM_BATTING_BB + mb_clean$TEAM_BASERUN_SB
```

```r
par(mfrow = c(1,1))
hist(mb_t$TB_PLUS, breaks = 200)
```




```r
# now drop 1B, 2B, 3B, HR, BB, SB
mb_tbp <- mb_t[,c(1, 2, 7, 9, 10, 11, 12, 13, 15)]

#########################################################################
# check correlation with WINS and run simple linear model
cor(mb_tbp$TARGET_WINS, mb_tbp$TB_PLUS)
```

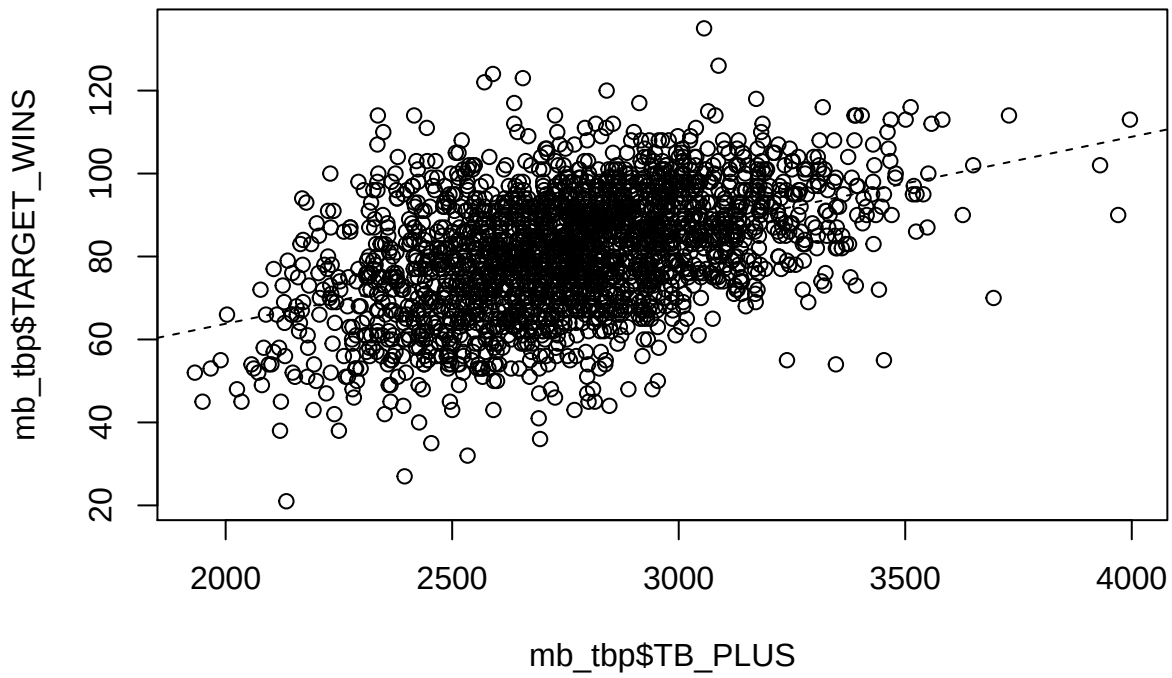
```
## [1] 0.4478658
```

```r
mtest <- lm(data=mb_tbp, TARGET_WINS ~ TB_PLUS)
summary(mtest)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TB_PLUS, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.805  -9.087   0.294   8.845  47.414
##
## Coefficients:
```

```
##               Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 18.7055981  2.6867939   6.962     0.00000000000443 ***
## TB_PLUS      0.0225394  0.0009659  23.334 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.01 on 2170 degrees of freedom
## Multiple R-squared:  0.2006, Adjusted R-squared:  0.2002
## F-statistic: 544.5 on 1 and 2170 DF,  p-value: < 0.00000000000000022
```

```
# shows .448 correlation and Adj R^2 of 0.2002 => better than TOTAL_BASES

plot(mb_tbp$TARGET_WINS ~ mb_tbp$TB_PLUS)
abline(lm(mb_tbp$TARGET_WINS ~ mb_tbp$TB_PLUS), lty=2)
```




```
# plot doesn't show unusual relationship
################################################################################
```

Yields r^2= 0.2845, Adj. R^2 = 0.2832, F = 215.4

```
# fit model
model <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
```

```
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.983  -8.050   0.222   8.210  55.388
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      51.167966   3.948869  12.958 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.036400   0.010160  -3.583             0.000347 ***
## TEAM_PITCHING_H   0.001971   0.003587   0.549             0.582772
## TEAM_PITCHING_BB -0.006469   0.003749  -1.726             0.084516 .
## TEAM_PITCHING_SO  0.020377   0.008595   2.371             0.017836 *
## TEAM_FIELDING_E  -0.042058   0.003130 -13.437 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.157922   0.012357 -12.780 < 0.0000000000000002 ***
## TB_PLUS           0.026357   0.002152  12.247 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.22 on 2164 degrees of freedom
## Multiple R-squared:  0.2965, Adjusted R-squared:  0.2942
## F-statistic: 130.3 on 7 and 2164 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_H

# ----------------------
# remove TEAM_PITCHING_H
model.2 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.105  -8.066   0.209   8.169  55.518
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      52.382736   3.271336  16.013 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.041392   0.004546  -9.106 < 0.0000000000000002 ***
## TEAM_PITCHING_BB -0.007564   0.003175  -2.383               0.0173 *
## TEAM_PITCHING_SO  0.024536   0.004070   6.028        0.00000000195 ***
## TEAM_FIELDING_E  -0.041729   0.003072 -13.584 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.157238   0.012293 -12.791 < 0.0000000000000002 ***
## TB_PLUS           0.027353   0.001161  23.559 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.22 on 2165 degrees of freedom
## Multiple R-squared:  0.2964, Adjusted R-squared:  0.2945
## F-statistic:   152 on 6 and 2165 DF,  p-value: < 0.00000000000000022
```

```r
# All p-values < .05 so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_SO TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##        15.346530         1.676592        12.003206         3.021956
## TEAM_FIELDING_DP          TB_PLUS
##         1.670760         1.637689
```

```r
# vif indicates remove TEAM_BATTING_SO or TEAM_PITCHING_SO, so try removing PITCHING_SO

# remove TEAM_PITCHING_SO
model.3 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO,
##     data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.713  -8.090   0.168   8.157  58.048
##
## Coefficients:
##                   Estimate Std. Error t value            Pr(>|t|)
## (Intercept)      49.697441   3.267190  15.211 <0.0000000000000002 ***
## TEAM_BATTING_SO  -0.015274   0.001386 -11.022 <0.0000000000000002 ***
## TEAM_PITCHING_BB -0.001905   0.003057  -0.623               0.533
## TEAM_FIELDING_E  -0.030940   0.002517 -12.293 <0.0000000000000002 ***
## TEAM_FIELDING_DP -0.150607   0.012343 -12.202 <0.0000000000000002 ***
## TB_PLUS           0.026000   0.001148  22.641 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.32 on 2166 degrees of freedom
## Multiple R-squared:  0.2846, Adjusted R-squared:  0.283
## F-statistic: 172.3 on 5 and 2166 DF,  p-value: < 0.00000000000000022
```

```r
# p-vals say remove TEAM_PITCHING_BB

# ----------------------
# remove TEAM_PITCHING_BB
model.4 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO - TEAM_PITCHING
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO -
##     TEAM_PITCHING_BB, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -46.491  -8.058   0.195   8.127  58.467
##
## Coefficients:
##                    Estimate Std. Error t value            Pr(>|t|)
## (Intercept)      49.7754486  3.2643296   15.25 <0.0000000000000002 ***
## TEAM_BATTING_SO  -0.0151681  0.0013752  -11.03 <0.0000000000000002 ***
## TEAM_FIELDING_E  -0.0309293  0.0025164  -12.29 <0.0000000000000002 ***
## TEAM_FIELDING_DP -0.1512808  0.0122935  -12.31 <0.0000000000000002 ***
## TB_PLUS           0.0255985  0.0009504   26.93 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.32 on 2167 degrees of freedom
## Multiple R-squared:  0.2845, Adjusted R-squared:  0.2832
## F-statistic: 215.4 on 4 and 2167 DF,  p-value: < 0.00000000000000022
```

```
# pvals all < .05 so check collinearity
vif(model.4)
```

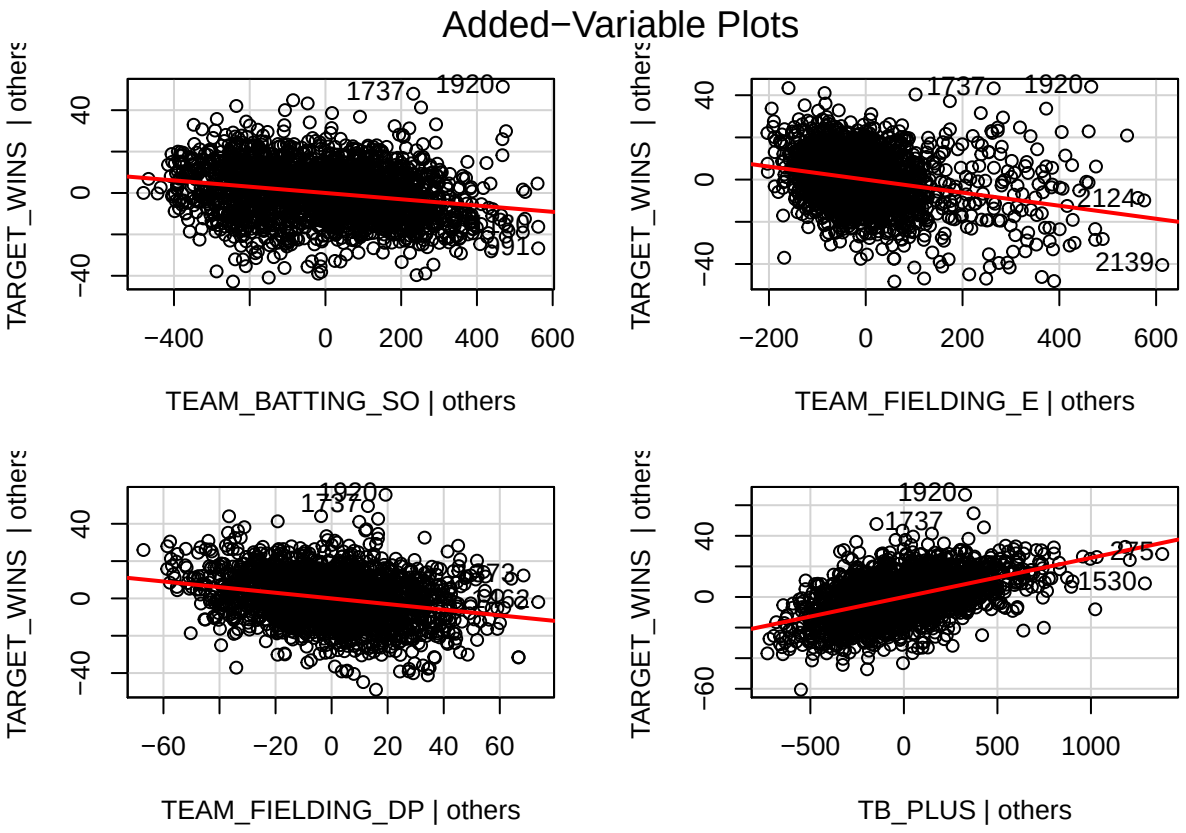
```
##  TEAM_BATTING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP          TB_PLUS
##         1.382409         1.995902         1.644666         1.080068
```

```
# no collinearity so STOP
```

Diagnostics

Plots for Fielding_E shows skew. Others are pretty linear

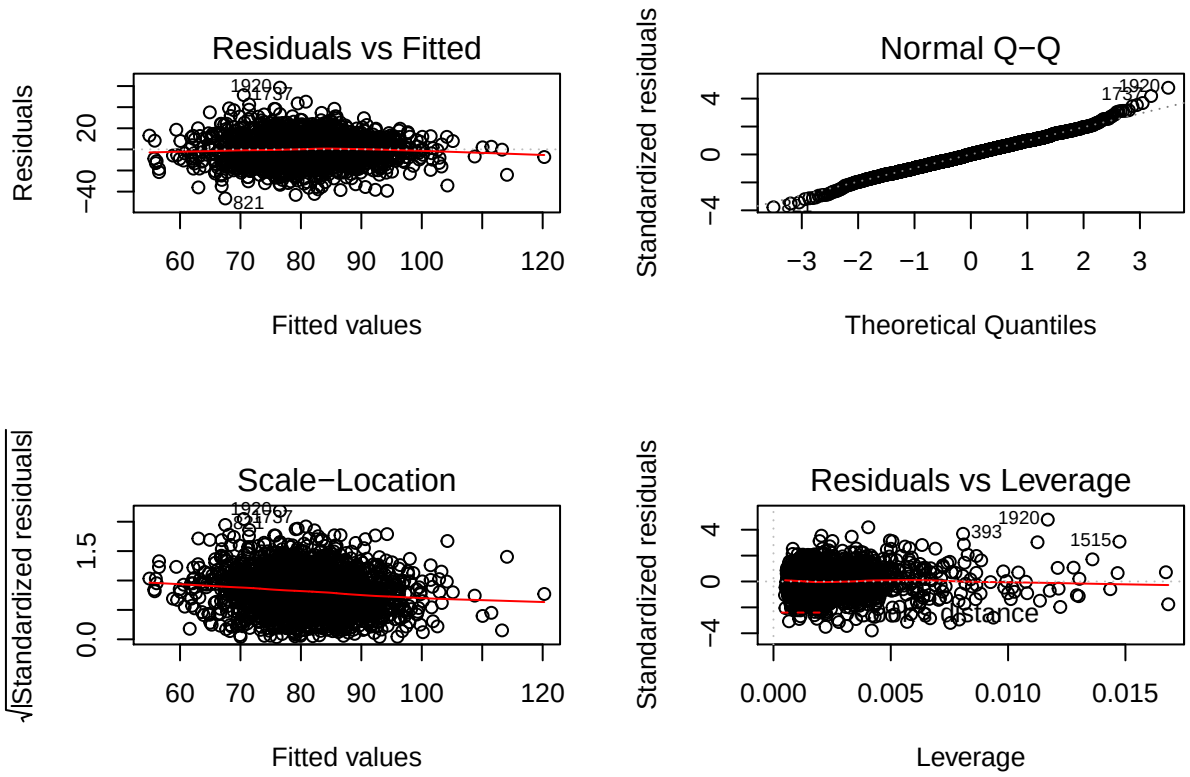
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.4, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: outliers at 2012, 1820, 859. Lack of Constant variability in Resid vs. Fitted at very large values of Yhat; normal distribution of residuals except for outliers, most residuals within 2 std dev and well within Cook's distance

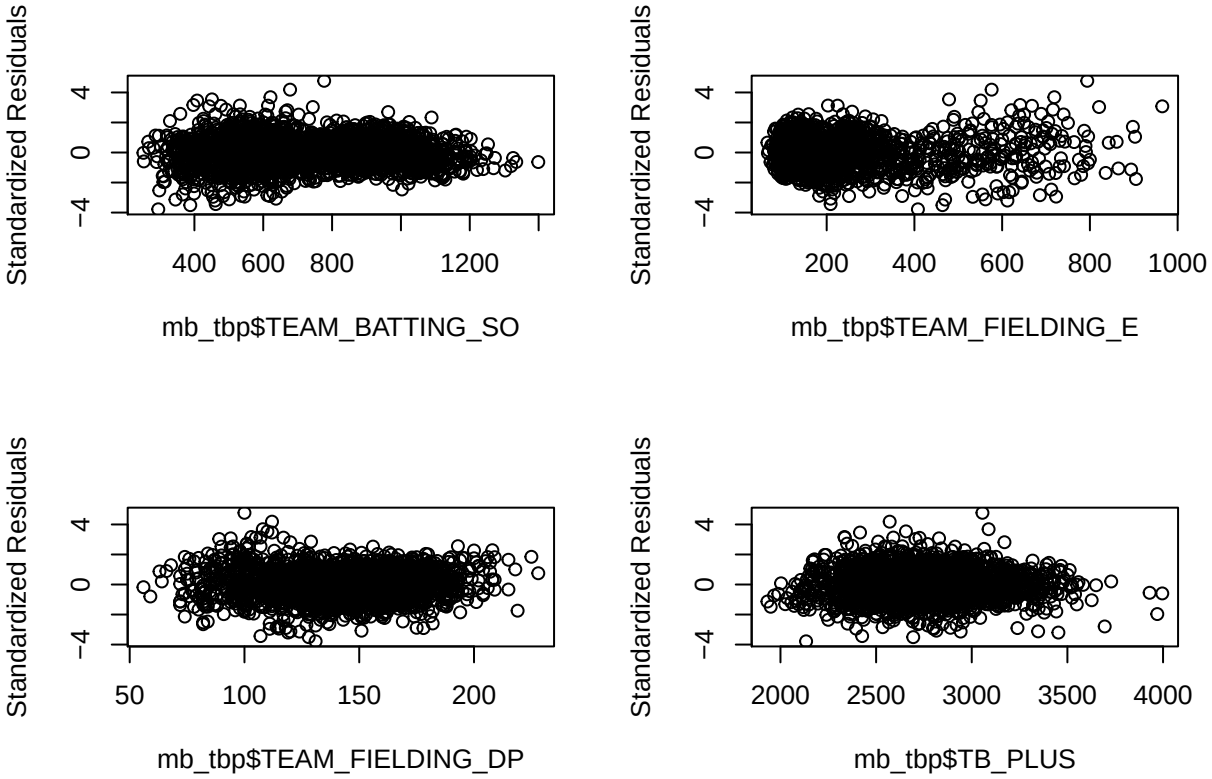
```
#Figure 5.6 on page 129 MARR text
par(mfrow=c(2,2))
plot(model.4)
```



PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

Results show lack of constant variability for PITCHING_SO, FIELDING_E, FIELDING_DP, TB_PLUS

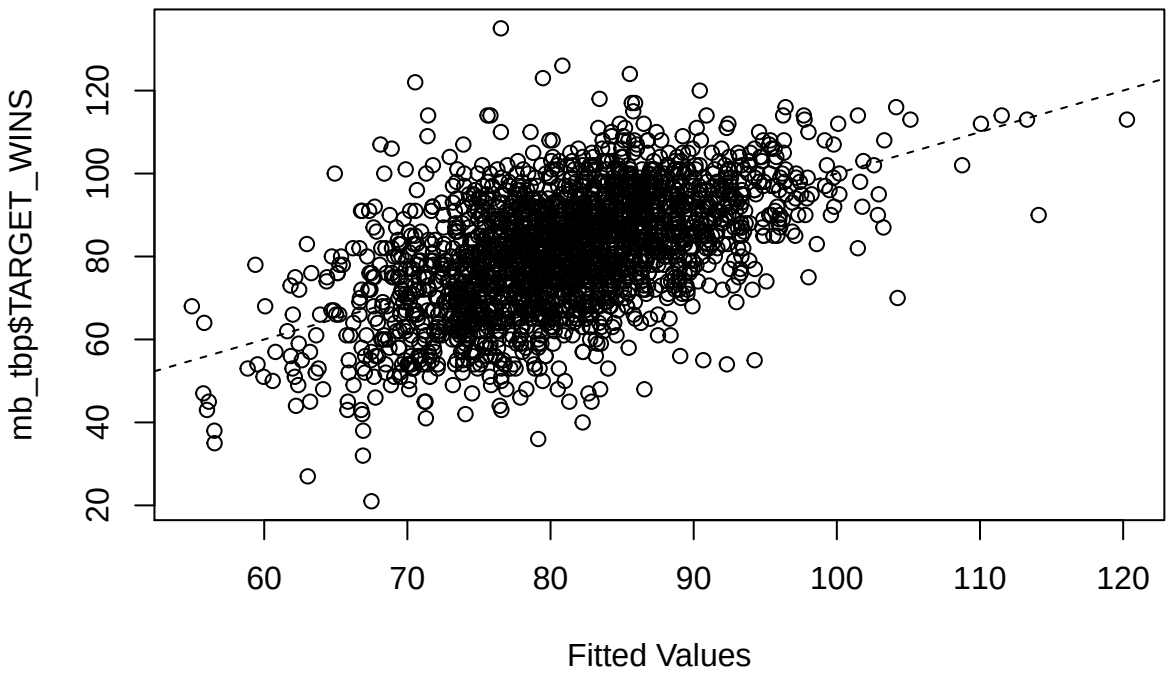
```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.4)
par(mfrow=c(2,2))

plot(mb_tbp$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TB_PLUS, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship with no pattern or skew

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.4$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb_tbp$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb_tbp$TARGET_WINS),lty=2)
```



REMOVE OUTLIERS AND REFIT

Per Cooks Distance, remove items 836, 821, 1920, 1737, 1515

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem <- mb_tbp[-c(836, 821, 1920, 1737, 1515),]

# save first data set
mb_tbp_orig <- mb_tbp

# renumber rows
rownames(mb_rem) <- 1:nrow(mb_rem)
```

Now refit first model from above: all variables

Yields r^2= 0.2944, Adj r^2 = 0.2931, F = 225.5

```
# keep the clean data set pure
mb_tbp <- mb_rem

# fit model
model <- lm(data=mb_tbp, TARGET_WINS ~ .- INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.431  -8.016   0.145   8.107  46.678
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      54.3616387  3.9168318  13.879 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.0391427  0.0100487  -3.895             0.000101 ***
## TEAM_PITCHING_H   0.0004962  0.0035469   0.140             0.888754
## TEAM_PITCHING_BB -0.0063810  0.0037061  -1.722             0.085255 .
## TEAM_PITCHING_SO  0.0216633  0.0084996   2.549             0.010880 *
## TEAM_FIELDING_E  -0.0441685  0.0031036 -14.232 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.1600617  0.0121816 -13.140 < 0.0000000000000002 ***
## TB_PLUS           0.0266454  0.0021256  12.536 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.03 on 2159 degrees of freedom
## Multiple R-squared:  0.3048, Adjusted R-squared:  0.3025
## F-statistic: 135.2 on 7 and 2159 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_H

# ----------------------
# remove TEAM_PITCHING_H
model.2 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.427  -7.980   0.143   8.125  46.678
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      54.670270   3.235709  16.896 < 0.0000000000000002 ***
## TEAM_BATTING_SO  -0.040401   0.004482  -9.014 < 0.0000000000000002 ***
## TEAM_PITCHING_BB -0.006657   0.003139  -2.121                0.034 *
## TEAM_PITCHING_SO  0.022711   0.004019   5.651         0.0000000181 ***
## TEAM_FIELDING_E  -0.044089   0.003051 -14.451 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.159894   0.012120 -13.193 < 0.0000000000000002 ***
## TB_PLUS           0.026896   0.001146  23.459 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.03 on 2160 degrees of freedom
## Multiple R-squared:  0.3048, Adjusted R-squared:  0.3028
```

```
## F-statistic: 157.8 on 6 and 2160 DF,  p-value: < 0.00000000000000022
```

```
# All p-values < .05 so check collinearity
vif(model.2)
```

```
##  TEAM_BATTING_SO TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##        15.343276         1.680924        12.009587         2.981041
## TEAM_FIELDING_DP          TB_PLUS
##         1.666368         1.641975
```

```
# vif indicates remove TEAM_BATTING_SO or TEAM_PITCHING_SO, so remove PITCHING_SO again

# ----------------------
# remove TEAM_PITCHING_SO
model.3 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO,
##     data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.795  -8.018   0.197   8.113  46.888
##
## Coefficients:
##                   Estimate Std. Error t value            Pr(>|t|)
## (Intercept)      52.267886   3.230538  16.179 <0.0000000000000002 ***
## TEAM_BATTING_SO  -0.016264   0.001368 -11.886 <0.0000000000000002 ***
## TEAM_PITCHING_BB -0.001378   0.003018  -0.457               0.648
## TEAM_FIELDING_E  -0.034211   0.002518 -13.585 <0.0000000000000002 ***
## TEAM_FIELDING_DP -0.153843   0.012159 -12.653 <0.0000000000000002 ***
## TB_PLUS           0.025628   0.001132  22.633 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.12 on 2161 degrees of freedom
## Multiple R-squared:  0.2945, Adjusted R-squared:  0.2928
## F-statistic: 180.4 on 5 and 2161 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_BB


# ----------------------
# remove TEAM_PITCHING_BB
model.4 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO - TEAM_PITCHING
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_SO -
```

```
##     TEAM_PITCHING_BB, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.678  -7.990   0.207   8.069  46.752
##
## Coefficients:
##                   Estimate Std. Error t value            Pr(>|t|)
## (Intercept)      52.3303439  3.2270486   16.22 <0.0000000000000002 ***
## TEAM_BATTING_SO  -0.0161904  0.0013585  -11.92 <0.0000000000000002 ***
## TEAM_FIELDING_E  -0.0342138  0.0025179  -13.59 <0.0000000000000002 ***
## TEAM_FIELDING_DP -0.1543417  0.0121072  -12.75 <0.0000000000000002 ***
## TB_PLUS           0.0253371  0.0009362   27.06 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.12 on 2162 degrees of freedom
## Multiple R-squared:  0.2944, Adjusted R-squared:  0.2931
## F-statistic: 225.5 on 4 and 2162 DF,  p-value: < 0.00000000000000022
```

```
# pvals all < .05 so check collinearity
vif(model.4)
```

```
##  TEAM_BATTING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP          TB_PLUS
##         1.390169         2.002462         1.640012         1.079847
```

```
anova(model)
```

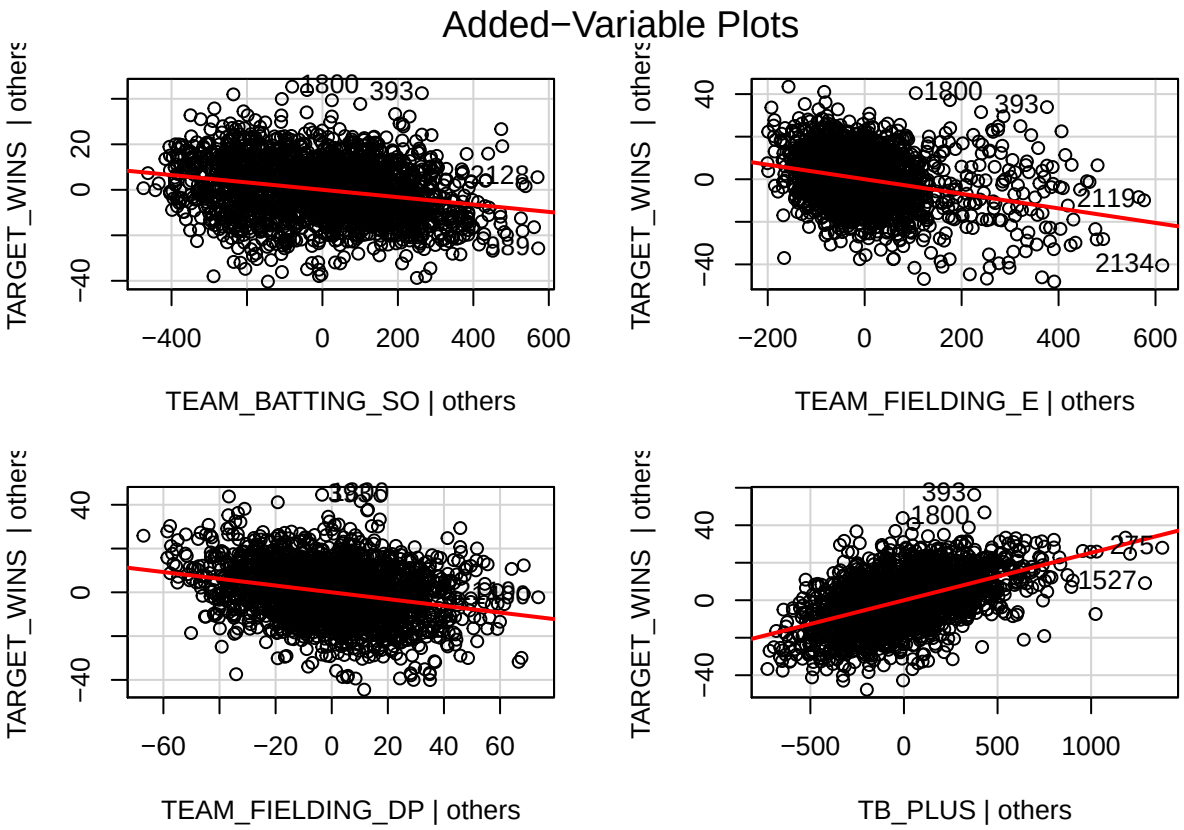
```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq F value               Pr(>F)
## TEAM_BATTING_SO     1   2189    2189  15.114            0.0001043 ***
## TEAM_PITCHING_H     1  13926   13926  96.157 < 0.00000000000000022 ***
## TEAM_PITCHING_BB    1  21937   21937 151.472 < 0.00000000000000022 ***
## TEAM_PITCHING_SO    1  36638   36638 252.979 < 0.00000000000000022 ***
## TEAM_FIELDING_E     1  14063   14063  97.103 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1  25556   25556 176.456 < 0.00000000000000022 ***
## TB_PLUS             1  22758   22758 157.139 < 0.00000000000000022 ***
## Residuals        2159 312681     145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# no collinearity so STOP
```

Diagnostics

Plots for Fielding_E shows skew. Others are pretty linear

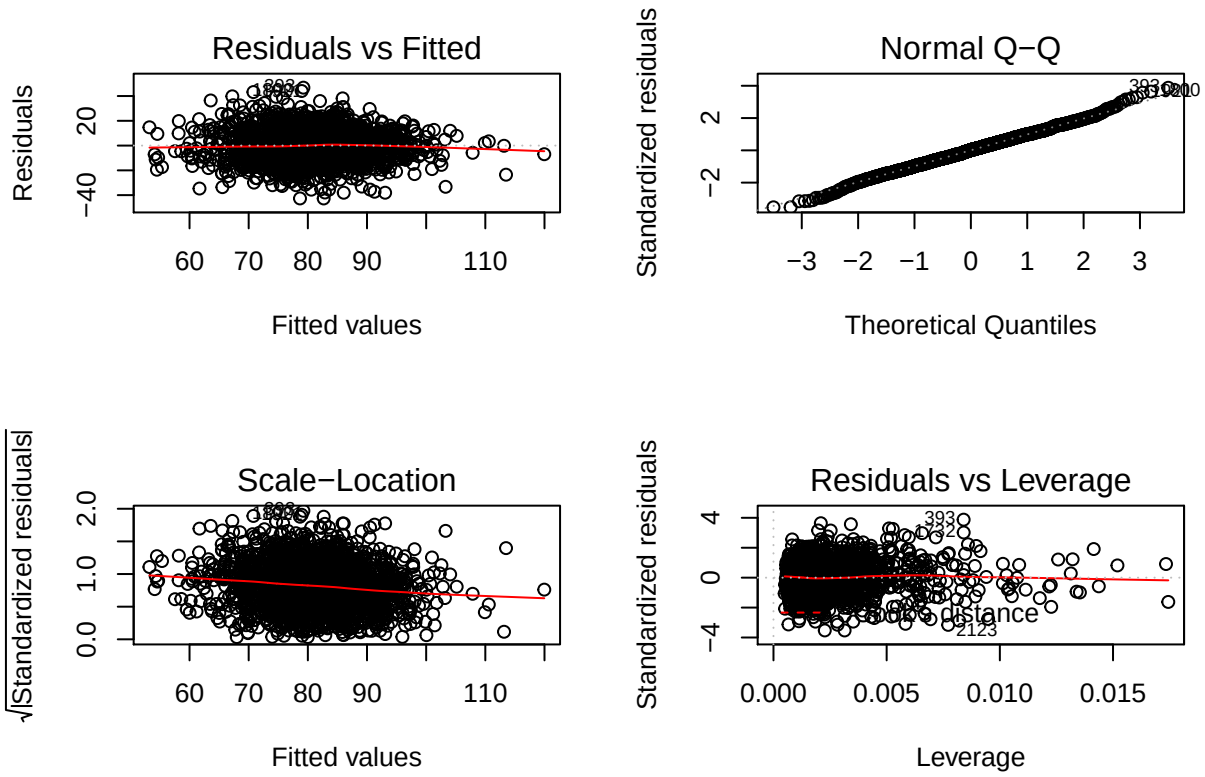
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.4, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Some lack of Constant variability in Resid vs. Fitted at very large values of Yhat; normal distribution of residuals; most residuals within 2 std dev and well within Cook's distance

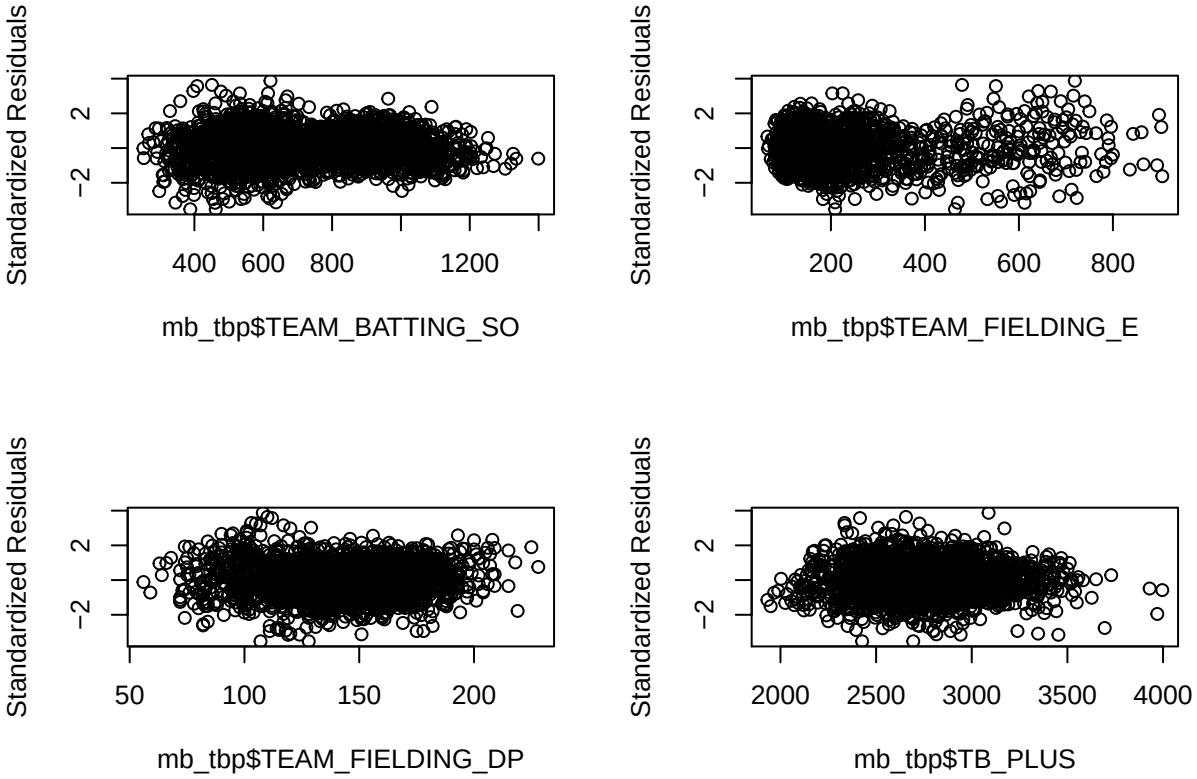
```
#Figure 5.6 on page 129 MARR text
par(mfrow=c(2,2))
plot(model.4)
```



PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

Results show lack of constant variability for PITCHING_SO, FIELDING_E, FIELDING_DP, TB_PLUS

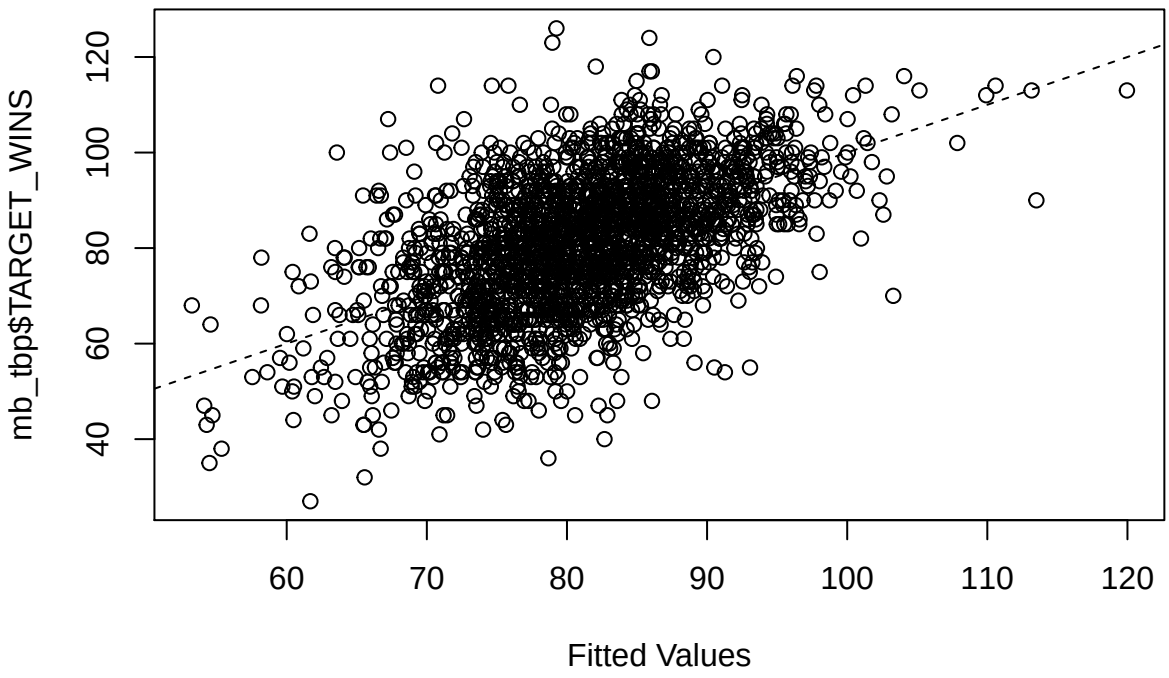
```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.4)
par(mfrow=c(2,2))

plot(mb_tbp$TEAM_BATTING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TB_PLUS, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship with no pattern or skew

```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.4$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb_tbp$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb_tbp$TARGET_WINS),lty=2)
```



Now try same model but with FIELD_E transformed using Box-Cox

```
# TEAM_FIELDING_E: Box-cox says -1 power transform => 1/y

mb_tbp$TEAM_FIELDING_E <- 1/mb_tbp$TEAM_FIELDING_E
```

Now refit first model from above: Start with all variables

Yields r^2= 0.2932, Adj r^2 = 0.2919, F = 224.3

```
# fit model
model <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX)
summary(model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.857  -7.906   0.310   8.238  43.204
##
## Coefficients:
```

```
##                      Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)        44.224515    3.776056  11.712 < 0.0000000000000002 ***
## TEAM_BATTING_SO    -0.033405    0.010094  -3.310              0.00095 ***
## TEAM_PITCHING_H    -0.003282    0.003541  -0.927              0.35409
## TEAM_PITCHING_BB   -0.002060    0.003735  -0.552              0.58128
## TEAM_PITCHING_SO    0.008654    0.008520   1.016              0.30983
## TEAM_FIELDING_E  2367.159014  184.199571  12.851 < 0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.138889    0.011786 -11.784 < 0.0000000000000002 ***
## TB_PLUS             0.023995    0.002172  11.049 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.13 on 2159 degrees of freedom
## Multiple R-squared:  0.2936, Adjusted R-squared:  0.2913
## F-statistic: 128.2 on 7 and 2159 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_BB

# ----------------------
# remove TEAM_PITCHING_BB
model.2 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB)
summary(model.2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.992  -7.888   0.292   8.267  43.111
##
## Coefficients:
##                      Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)        43.622192    3.614144  12.070 < 0.0000000000000002 ***
## TEAM_BATTING_SO    -0.030130    0.008160  -3.692             0.000228 ***
## TEAM_PITCHING_H    -0.002208    0.002957  -0.747             0.455333
## TEAM_PITCHING_SO    0.005899    0.006900   0.855             0.392714
## TEAM_FIELDING_E  2371.413436  184.008382  12.888 < 0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.139854    0.011654 -12.001 < 0.0000000000000002 ***
## TB_PLUS             0.023137    0.001515  15.268 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.13 on 2160 degrees of freedom
## Multiple R-squared:  0.2935, Adjusted R-squared:  0.2915
## F-statistic: 149.5 on 6 and 2160 DF,  p-value: < 0.00000000000000022
```

```
# p-values say remove TEAM_PITCHING_H

# ----------------------
# remove TEAM_PITCHING_H
model.3 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H)
summary(model.3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H,
##     data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.307  -7.918   0.345   8.182  42.508
##
## Coefficients:
##                      Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)        42.0041157    2.8921302  14.524 < 0.0000000000000002
## TEAM_BATTING_SO    -0.0247576    0.0038507  -6.429       0.000000000157
## TEAM_PITCHING_SO    0.0013897    0.0033380   0.416                0.677
## TEAM_FIELDING_E  2387.2244422  182.7672567  13.062 < 0.0000000000000002
## TEAM_FIELDING_DP   -0.1396085    0.0116479 -11.986 < 0.0000000000000002
## TB_PLUS             0.0222592    0.0009561  23.280 < 0.0000000000000002
##
## (Intercept)      ***
## TEAM_BATTING_SO  ***
## TEAM_PITCHING_SO
## TEAM_FIELDING_E  ***
## TEAM_FIELDING_DP ***
## TB_PLUS          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.13 on 2161 degrees of freedom
## Multiple R-squared:  0.2933, Adjusted R-squared:  0.2917
## F-statistic: 179.4 on 5 and 2161 DF,  p-value: < 0.00000000000000022
```

```
# p-vals say remove TEAM_PITCHING_SO

# ----------------------
# remove TEAM_PITCHING_SO
model.4 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H - TEAM_PITCHING
summary(model.4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H -
##     TEAM_PITCHING_SO, data = mb_tbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.310  -7.909   0.320   8.191  42.644
##
## Coefficients:
##                      Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)        42.1597649    2.8673139   14.70 <0.0000000000000002 ***
## TEAM_BATTING_SO    -0.0233150    0.0016789  -13.89 <0.0000000000000002 ***
## TEAM_FIELDING_E  2366.8259932  176.0432309   13.45 <0.0000000000000002 ***
## TEAM_FIELDING_DP   -0.1400776    0.0115910  -12.09 <0.0000000000000002 ***
## TB_PLUS             0.0222811    0.0009545   23.34 <0.0000000000000002 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.13 on 2162 degrees of freedom
## Multiple R-squared:  0.2932, Adjusted R-squared:  0.2919
## F-statistic: 224.3 on 4 and 2162 DF,  p-value: < 0.00000000000000022
```

```
# pvals all < .05 so check collinearity
vif(model.4)
```

```
##  TEAM_BATTING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP          TB_PLUS
##         2.119682         2.906955         1.500660         1.120613
```

```
# no collinearity so STOP
options(scipen=999)
model.4
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H -
##     TEAM_PITCHING_SO, data = mb_tbp)
##
## Coefficients:
##      (Intercept)   TEAM_BATTING_SO   TEAM_FIELDING_E  TEAM_FIELDING_DP
##         42.15976          -0.02331        2366.82599          -0.14008
##          TB_PLUS
##          0.02228
```

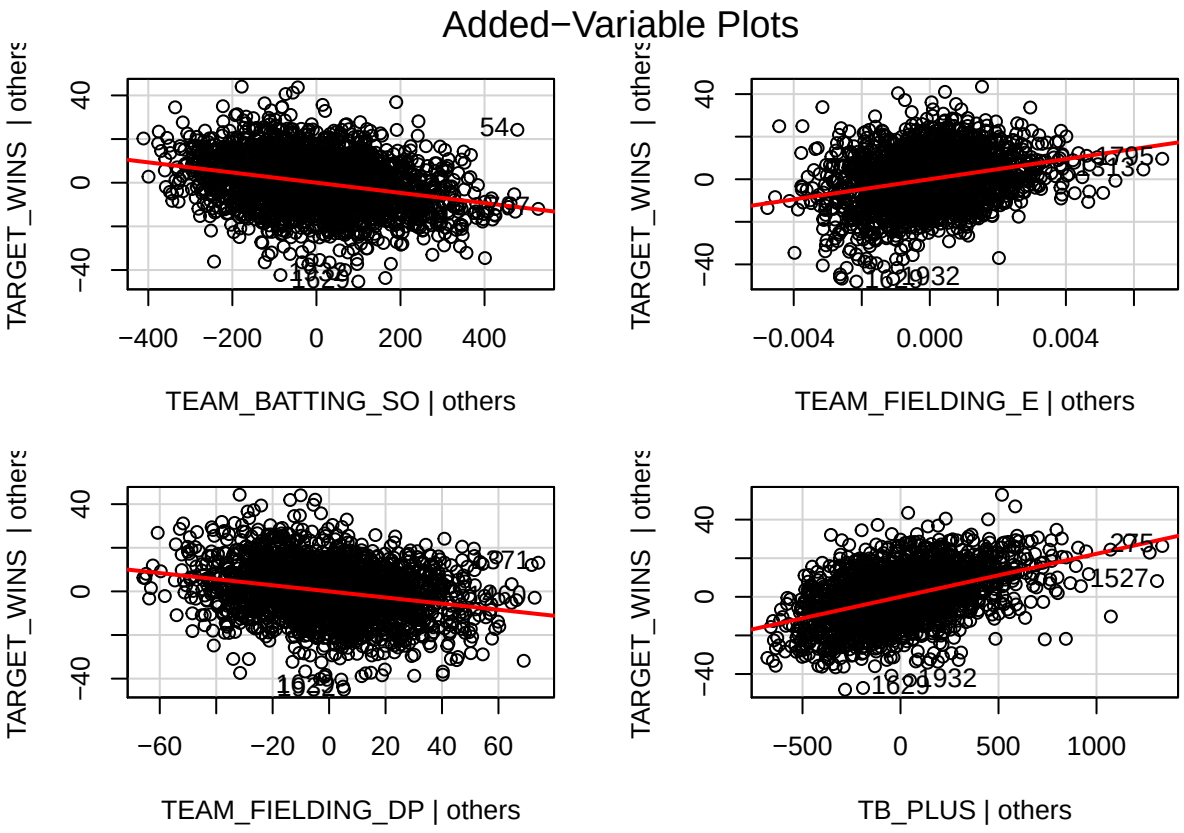
```
anova(model.4)
```

```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq F value                Pr(>F)
## TEAM_BATTING_SO     1   2189    2189  14.888             0.0001174 ***
## TEAM_FIELDING_E     1  34681   34681 235.888 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1  14903   14903 101.365 < 0.00000000000000022 ***
## TB_PLUS             1  80112   80112 544.895 < 0.00000000000000022 ***
## Residuals        2162 317863     147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diagnostics

Plots show all variables are linear to response

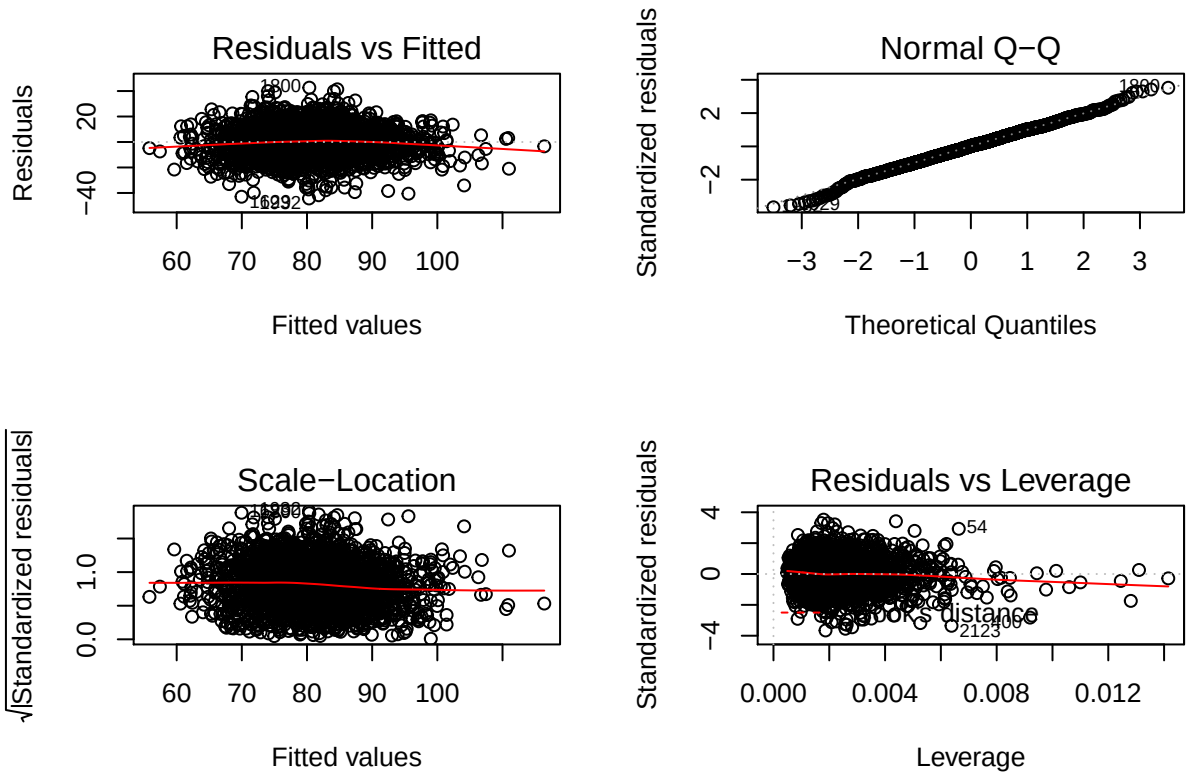
```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(model.4, id.n = 2)
```



SUMMARY MODEL DIAGNOSTIC PLOTS

Plots: Some lack of Constant variability in Resid vs. Fitted at very large values of Yhat; normal distribution of residuals; most residuals within 2 std dev and well within Cook's distance

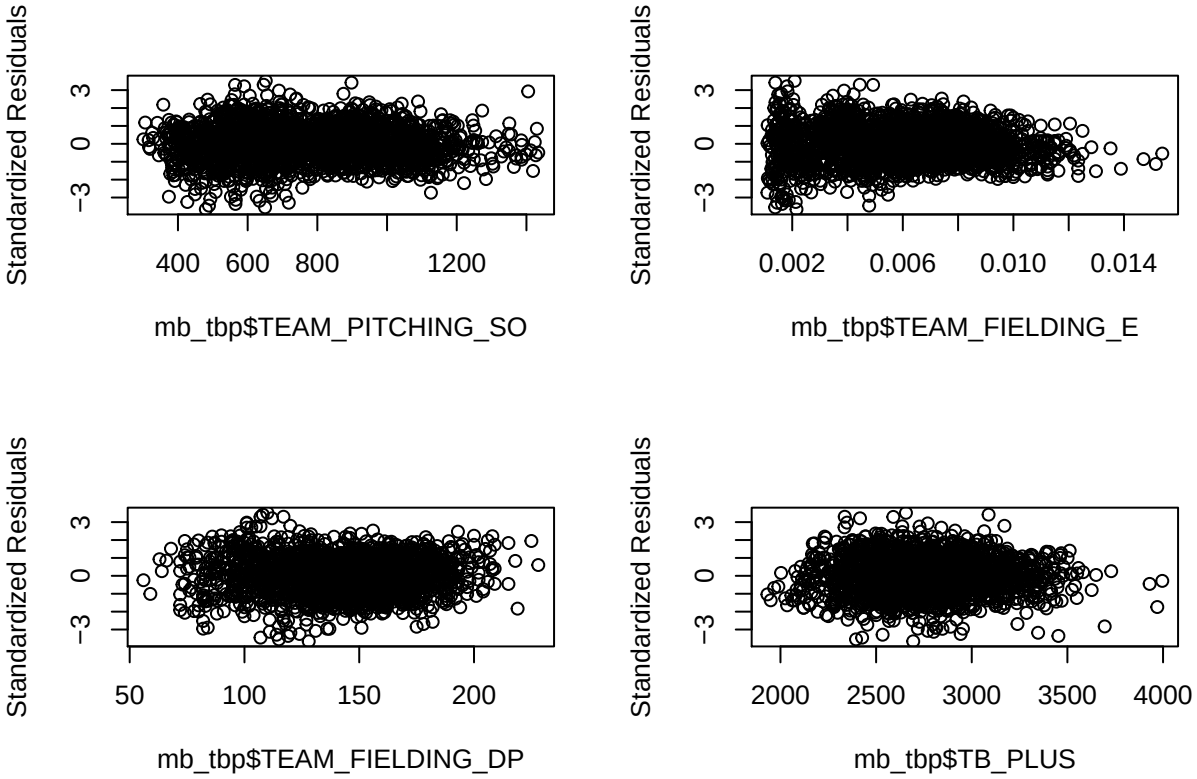
```
#Figure 5.6 on page 129 MARR text
par(mfrow=c(2,2))
plot(model.4)
```



PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

Results show lack of constant variability for PITCHING_SO, FIELDING_E, FIELDING_DP, TB_PLUS

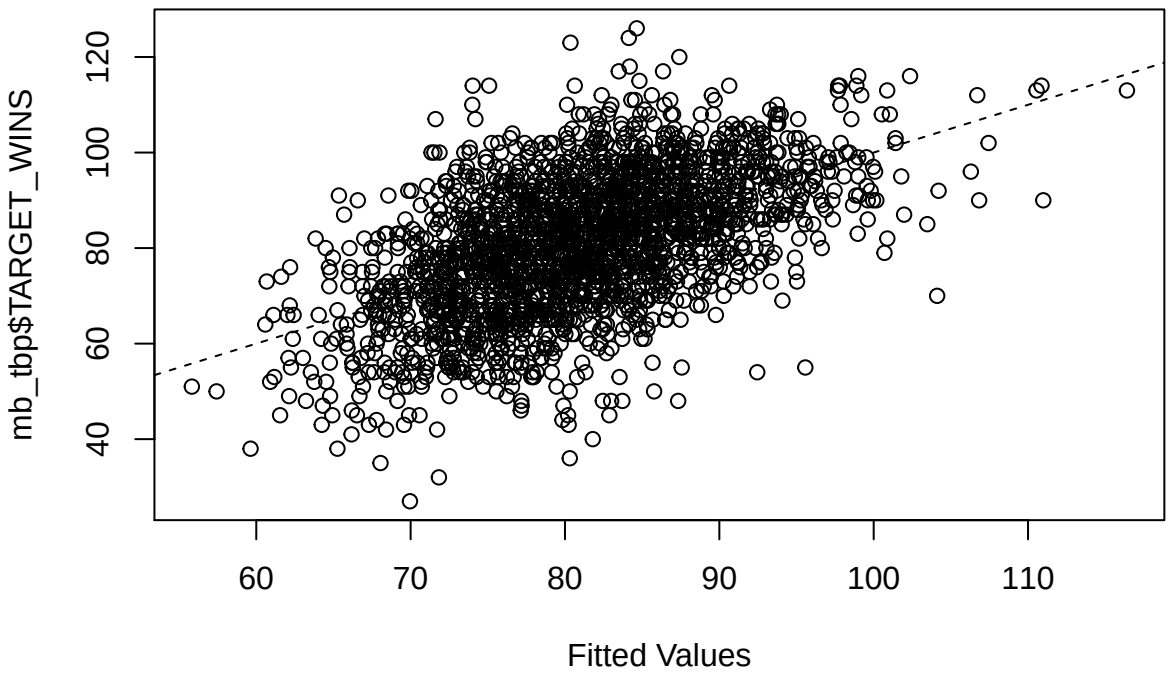
```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

StanRes1 <- rstandard(model.4)
par(mfrow=c(2,2))

plot(mb_tbp$TEAM_PITCHING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb_tbp$TB_PLUS, StanRes1, ylab="Standardized Residuals")
```



PLOT Y AGAINST FITTED VALUES

Plot shows a linear relationship with no pattern or skew

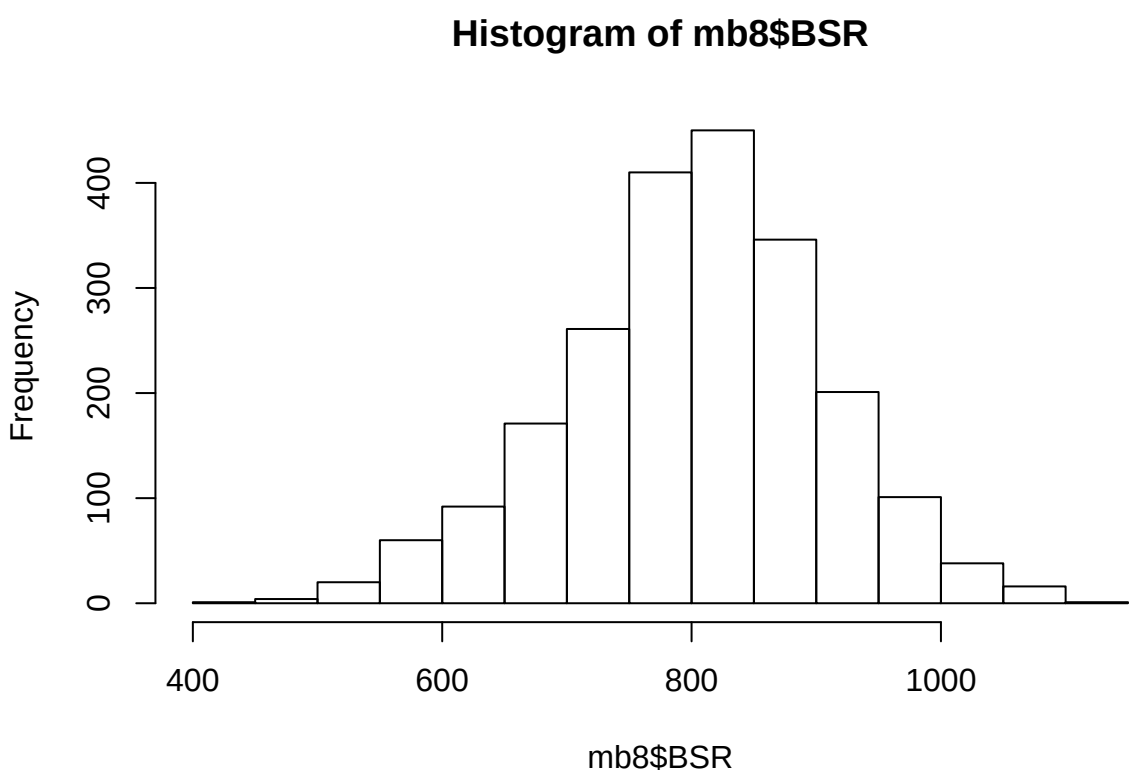
```
# get rows have no NA's from data frame
# NoNA <- mb_mods[!rowSums(is.na(mb_mods[1:13])), ]

fit1 <- model.4$fitted.values

par(mfrow = c(1,1))
plot(fit1, mb_tbp$TARGET_WINS,xlab="Fitted Values")
abline(lsfit(fit1, mb_tbp$TARGET_WINS),lty=2)
```




```
# clean up objects in memory
rm(list = ls())
```

Model 4: Sabermetrics Model

```
#Define BSR
mb_clean <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clean
mb8 <- mb_clean
A <- mb8$TEAM_BATTING_1B + mb8$TEAM_BATTING_2B + mb8$TEAM_BATTING_3B + mb8$TEAM_BATTING_BB
B <- 1.02*(1.4*(mb8$TEAM_BATTING_1B + 2*mb8$TEAM_BATTING_2B + 3*mb8$TEAM_BATTING_3B +  4*mb8$TEAM_BATTI
C <- 3*(mb8$TEAM_BATTING_1B + mb8$TEAM_BATTING_2B + mb8$TEAM_BATTING_3B + mb8$TEAM_BATTING_HR)
D <- mb8$TEAM_BATTING_HR

mb8$BSR <- (A*B)/(B+C) + D
hist(mb8$BSR)
```




```
#the data look normally distributed

#look at model starting with all
test <- lm(data=mb8, TARGET_WINS~.)
summary(test)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = mb8)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.085  -7.609   0.258   7.248  60.755
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      50.0970369  5.6394346   8.883 < 0.0000000000000002 ***
## INDEX            -0.0002511  0.0003432  -0.732             0.464516
## TEAM_BATTING_2B  -0.3177963  0.1063109  -2.989             0.002828 **
## TEAM_BATTING_3B  -0.3081077  0.1660502  -1.856             0.063660 .
## TEAM_BATTING_HR  -0.5624740  0.2465633  -2.281             0.022630 *
## TEAM_BATTING_BB  -0.2281423  0.1577843  -1.446             0.148347
## TEAM_BATTING_SO  -0.0446718  0.0118225  -3.779             0.000162 ***
## TEAM_BASERUN_SB   0.0683845  0.0049742  13.748 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.0348975  0.0051637   6.758      0.0000000000179 ***
## TEAM_PITCHING_BB -0.1259901  0.0152050  -8.286 < 0.0000000000000002 ***
```

```
## TEAM_PITCHING_SO  0.0273675  0.0110377   2.479             0.013234 *
## TEAM_FIELDING_E  -0.0764979  0.0038928 -19.651 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.1229190  0.0131766  -9.329 < 0.0000000000000002 ***
## TEAM_BATTING_1B  -0.1373735  0.0489780  -2.805             0.005080 **
## BSR               0.6199285  0.2389229   2.595             0.009532 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.6 on 2157 degrees of freedom
## Multiple R-squared:  0.3679, Adjusted R-squared:  0.3638
## F-statistic: 89.69 on 14 and 2157 DF,  p-value: < 0.00000000000000022
```

```
vif(test)
```

```
##            INDEX  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR
##         1.026470       378.693783       321.509919      3431.648281
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##      4025.010742       115.120460         3.426874        28.095381
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        42.655116        97.880174         5.382010         2.128980
##  TEAM_BATTING_1B              BSR
##       406.172400     10103.718692
```

```
#take out all hitting values for reasons of collinearity

test.1 <- lm(data=mb8, TARGET_WINS~BSR+TEAM_FIELDING_DP+TEAM_FIELDING_E+TEAM_BASERUN_SB+TEAM_PITCHING_S
summary(test.1)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BSR + TEAM_FIELDING_DP + TEAM_FIELDING_E +
##     TEAM_BASERUN_SB + TEAM_PITCHING_SO + TEAM_PITCHING_H + TEAM_PITCHING_BB,
##     data = mb8)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.109  -8.251   0.117   7.835  55.707
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      40.687320   3.146659  12.930 < 0.0000000000000002 ***
## BSR               0.062189   0.004529  13.730 < 0.0000000000000002 ***
## TEAM_FIELDING_DP -0.116615   0.013348  -8.736 < 0.0000000000000002 ***
## TEAM_FIELDING_E  -0.058885   0.003688 -15.965 < 0.0000000000000002 ***
## TEAM_BASERUN_SB   0.060347   0.004785  12.612 < 0.0000000000000002 ***
## TEAM_PITCHING_SO -0.011457   0.001218  -9.409 < 0.0000000000000002 ***
## TEAM_PITCHING_H   0.019419   0.001450  13.392 < 0.0000000000000002 ***
## TEAM_PITCHING_BB -0.017603   0.003746  -4.700           0.00000277 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.99 on 2164 degrees of freedom
## Multiple R-squared:  0.3229, Adjusted R-squared:  0.3207
## F-statistic: 147.4 on 7 and 2164 DF,  p-value: < 0.00000000000000022
```

```
vif(test.1)
```

```
##              BSR TEAM_FIELDING_DP  TEAM_FIELDING_E  TEAM_BASERUN_SB
##         3.400715         2.046156         4.525367         2.969798
## TEAM_PITCHING_SO  TEAM_PITCHING_H TEAM_PITCHING_BB
##         1.115534         2.074930         2.424214
```

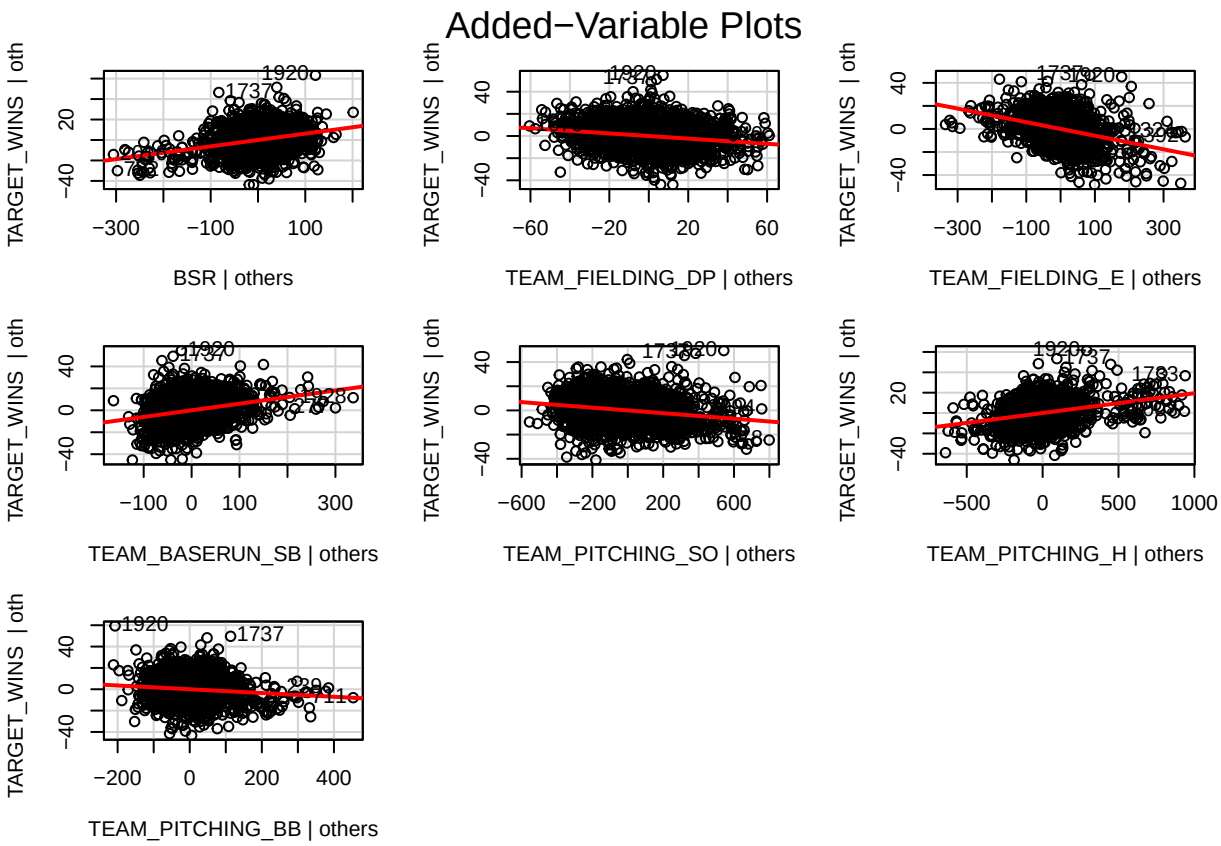
```
#p-values and variability look good
#run anova to get MSE
anova(test.1)
```

```
## Analysis of Variance Table
##
## Response: TARGET_WINS
##                    Df Sum Sq Mean Sq  F value                Pr(>F)
## BSR                 1  60634   60634 421.8899 < 0.00000000000000022 ***
## TEAM_FIELDING_DP    1  24932   24932 173.4790 < 0.00000000000000022 ***
## TEAM_FIELDING_E     1    822     822   5.7184               0.01687 *
## TEAM_BASERUN_SB     1  19522   19522 135.8354 < 0.00000000000000022 ***
## TEAM_PITCHING_SO    1  13730   13730  95.5311 < 0.00000000000000022 ***
## TEAM_PITCHING_H     1  25523   25523 177.5877 < 0.00000000000000022 ***
## TEAM_PITCHING_BB    1   3174    3174  22.0866            0.00000277 ***
## Residuals        2164 311010     144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

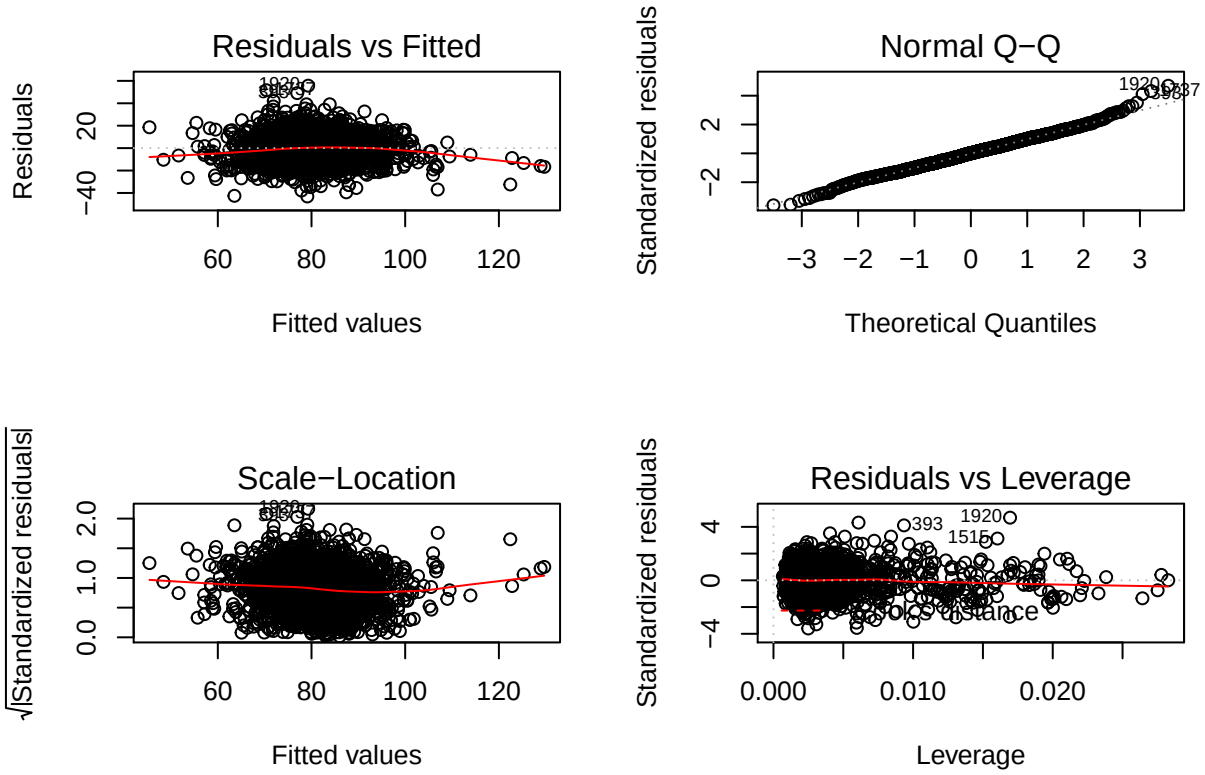
```
###MODEL DIAGNOSTICS
##### Linear Model Diagnostic Plots w/ R
#Test 1
vif(test.1)
```

```
##              BSR TEAM_FIELDING_DP  TEAM_FIELDING_E  TEAM_BASERUN_SB
##         3.400715         2.046156         4.525367         2.969798
## TEAM_PITCHING_SO  TEAM_PITCHING_H TEAM_PITCHING_BB
##         1.115534         2.074930         2.424214
```

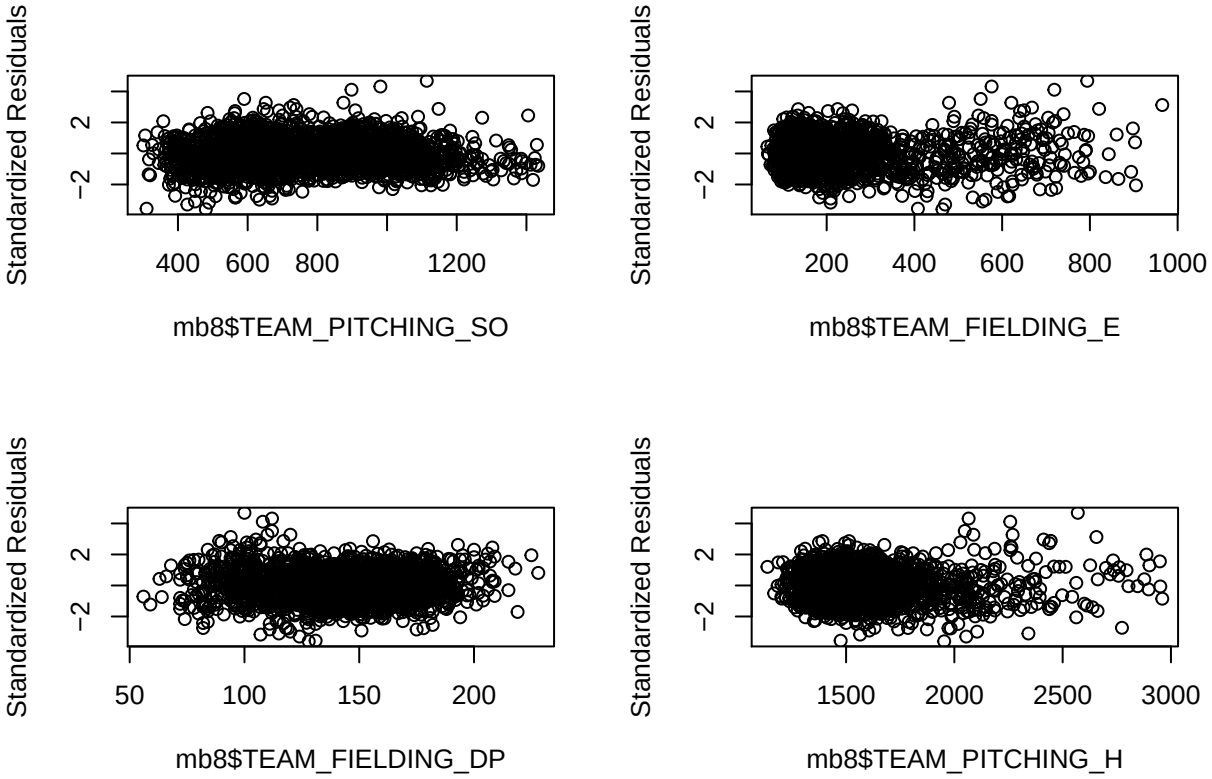
```
#Test 2
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(test.1, id.n = 2)
```



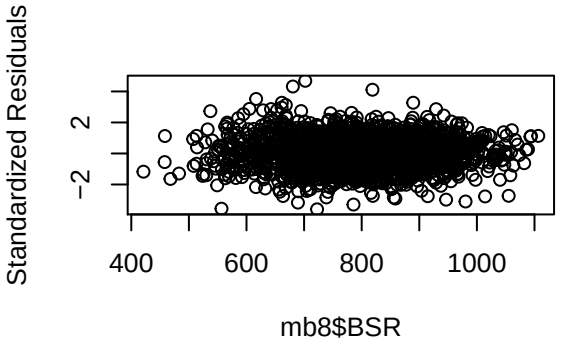

```
#Test 3
par(mfrow=c(2,2))
plot(test.1)
```

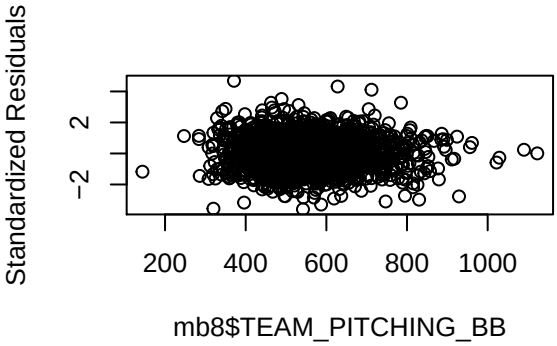


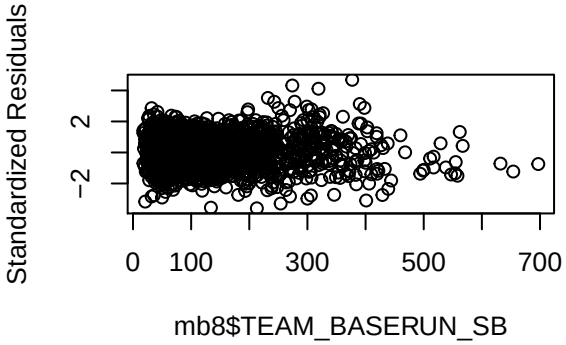

```
#Test 4
StanRes1 <- rstandard(test.1)

# plot these in groups of 4 so that they are legible
par(mfrow=c(2,2))

plot(mb8$TEAM_PITCHING_SO, StanRes1, ylab="Standardized Residuals")
plot(mb8$TEAM_FIELDING_E, StanRes1, ylab="Standardized Residuals")
plot(mb8$TEAM_FIELDING_DP, StanRes1, ylab="Standardized Residuals")
plot(mb8$TEAM_PITCHING_H, StanRes1, ylab="Standardized Residuals")
```



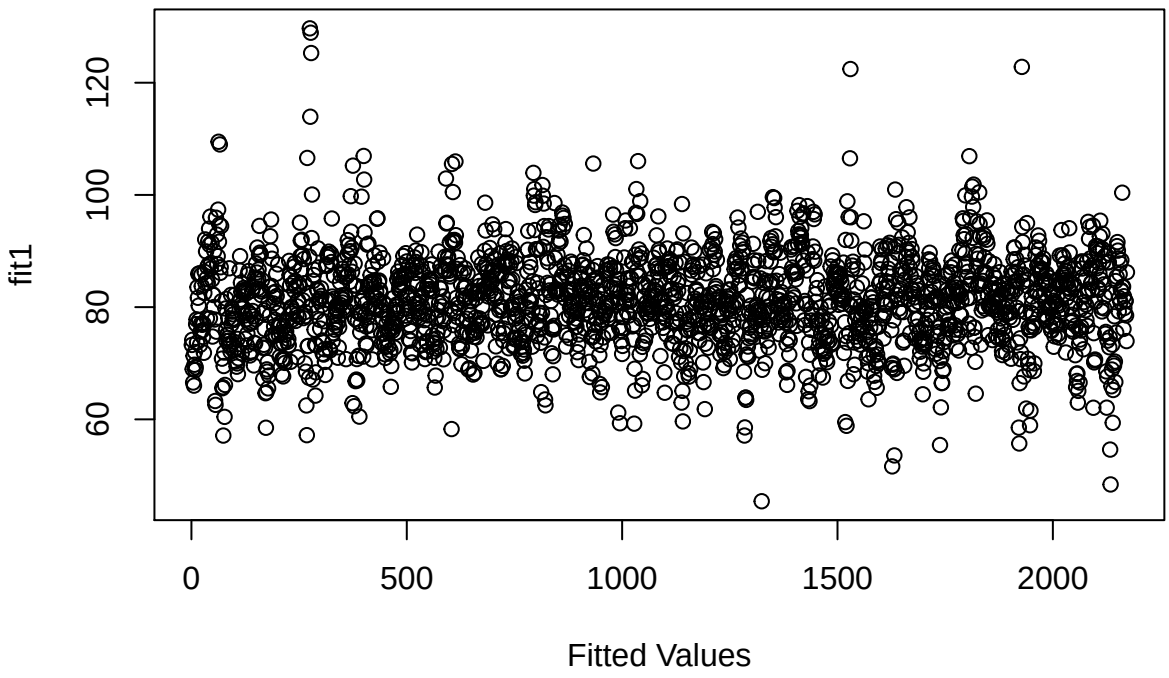

```
par(mfrow=c(2,2))
plot(mb8$BSR, StanRes1, ylab="Standardized Residuals")
plot(mb8$TEAM_PITCHING_BB, StanRes1, ylab="Standardized Residuals")
plot(mb8$TEAM_BASERUN_SB, StanRes1, ylab="Standardized Residuals")

#Test 5
fit1 <- test.1$fitted.values

par(mfrow = c(1,1))
```








```
plot(fit1, test.1$TARGET_WINS,xlab="Fitted Values")
```




```
rm(list = ls())
```

Model 5

SINGLE PREDICTOR ANALYSIS & TRANSFORMATIONS

Model SMK Generlized Equation

Review descriptive statistics to confirm each variable is within acceptable bounds and contains no missing data. Review Density plots of 13 variables for skewness to identify which may require transformation.

```
#assign model to "clean" data set
lm.smk <- mb7
#remove bad leverage points from diagnostic tests
lm.smk <- lm.smk[-c(1737,1920,226,391,385,1702,840,602,1928,1937,2109,2128,269,711,125),]
# now renumber rows of dataframe so that there are no gaps in row numbers
rownames(lm.smk) <- 1:nrow(lm.smk)
nrow(lm.smk)

#SINGLES:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.1B <- lm.smk$TEAM_BATTING_1B
m1 <- lm(mW~mb.1B, data = lm.smk)
```

```
StanRes1 <- rstandard(m1)
plot(density(mb.1B),main="Singles");rug(mb.1B)
plot(mb.1B,StanRes1,xlab="Singles",ylab="Standardized Residuals");abline(lsfit(mb.1B,StanRes1),lty=2,col
qqnorm(mb.1B,ylab="Y");qqline(mb.1B,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.1B, family="bcPower")$lambda# round(-1.983968) => -2 => 1/(y^2)
tmb.1B <- mb.1B^(-2)
plot(density(tmb.1B),main="Singles");rug(tmb.1B)
plot(tmb.1B,StanRes1,xlab="Singles",ylab="Standardized Residuals");abline(lsfit(tmb.1B,StanRes1),lty=2,
qqnorm(tmb.1B,ylab="Y");qqline(tmb.1B,lty=2,col=2)

#DOUBLES:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.2B <- lm.smk$TEAM_BATTING_2B
m1 <- lm(mW~mb.2B, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.2B),main="Doubles");rug(mb.2B)
plot(mb.2B,StanRes1,xlab="Doubles",ylab="Standardized Residuals");abline(lsfit(mb.2B,StanRes1),lty=2,col
qqnorm(mb.2B,ylab="Y");qqline(mb.2B,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.2B, family="bcPower")$lambda# round(0.5362315) => y^.5 => sqrt(y)
tmb.2B <- sqrt(mb.2B)
plot(density(tmb.2B),main="Doubles");rug(tmb.2B)
plot(tmb.2B,StanRes1,xlab="Doubles",ylab="Standardized Residuals");abline(lsfit(tmb.2B,StanRes1),lty=2,
qqnorm(tmb.2B,ylab="Y");qqline(tmb.2B,lty=2,col=2)

#TRIPLES:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.3B <- lm.smk$TEAM_BATTING_3B
m1 <- lm(mW~mb.3B, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.3B),main="Triples");rug(mb.3B)
plot(mb.3B,StanRes1,xlab="Triples",ylab="Standardized Residuals");abline(lsfit(mb.3B,StanRes1),lty=2,col
qqnorm(mb.3B,ylab="Y");qqline(mb.3B,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.3B, family="bcPower")$lambda# round(-0.03308475) => (1/y^30)
tmb.3B <- mb.3B^(-30)
plot(density(tmb.3B),main="Triples");rug(tmb.3B)
plot(tmb.3B,StanRes1,xlab="Triples",ylab="Standardized Residuals");abline(lsfit(tmb.3B,StanRes1),lty=2,
qqnorm(tmb.3B,ylab="Y");qqline(tmb.3B,lty=2,col=2)

#HOMERUNS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.HR <- lm.smk$TEAM_BATTING_HR
m1 <- lm(mW~mb.HR, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.HR),main="Homeruns");rug(mb.HR)
plot(mb.HR,StanRes1,xlab="Homeruns",ylab="Standardized Residuals");abline(lsfit(mb.HR,StanRes1),lty=2,co
qqnorm(mb.HR,ylab="Y");qqline(mb.HR,lty=2,col=2)
##TRANSFORMATION
```

```
powerTransform(mb.HR, family="bcPower")$lambda# round(0.6348318) => y^(2/3)
tmb.HR <- mb.HR^(2/3)
plot(density(tmb.HR),main="Homeruns");rug(tmb.HR)
plot(tmb.HR,StanRes1,xlab="Homeruns",ylab="Standardized Residuals");abline(lsfit(tmb.HR,StanRes1),lty=2
qqnorm(tmb.HR,ylab="Y");qqline(tmb.HR,lty=2,col=2)

#SLUGGING
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.SL <- lm.smk$TEAM_BATTING_HR + 2 * lm.smk$TEAM_BATTING_3B
m1 <- lm(mW~mb.SL, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.SL),main="Slugging");rug(mb.SL)
plot(mb.SL,StanRes1,xlab="Slugging",ylab="Standardized Residuals");abline(lsfit(mb.SL,StanRes1),lty=2,c
qqnorm(mb.SL,ylab="Y");qqline(mb.SL,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.SL, family="bcPower")$lambda# round(0.5562253) => y^(.5)
tmb.SL <- sqrt(mb.SL)
plot(density(tmb.SL),main="Slugging");rug(tmb.SL)
plot(tmb.SL,StanRes1,xlab="Slugging",ylab="Standardized Residuals");abline(lsfit(tmb.SL,StanRes1),lty=2
qqnorm(tmb.SL,ylab="Y");qqline(tmb.SL,lty=2,col=2)

#WALKS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.BB <- lm.smk$TEAM_BATTING_BB
m1 <- lm(mW~mb.BB, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.BB),main="Walks");rug(mb.BB)
plot(mb.BB,StanRes1,xlab="Walks",ylab="Standardized Residuals");abline(lsfit(mb.BB,StanRes1),lty=2,col=
qqnorm(mb.BB,ylab="Y");qqline(mb.BB,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.BB, family="bcPower")$lambda# round(1.434735) => y^(3/2)
tmb.BB <- mb.BB^(3/2)
plot(density(tmb.BB),main="Walks");rug(tmb.BB)
plot(tmb.BB,StanRes1,xlab="Walks",ylab="Standardized Residuals");abline(lsfit(tmb.BB,StanRes1),lty=2,co
qqnorm(tmb.BB,ylab="Y");qqline(tmb.BB,lty=2,col=2)

#STRIKEOUTS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.SO <- lm.smk$TEAM_BATTING_SO
m1 <- lm(mW~mb.SO, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.SO),main="Strikeouts");rug(mb.SO)
plot(mb.SO,StanRes1,xlab="Strikeouts",ylab="Standardized Residuals");abline(lsfit(mb.SO,StanRes1),lty=2
qqnorm(mb.SO,ylab="Y");qqline(mb.SO,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.SO, family="bcPower")$lambda# round(0.7159533) => y^(3/4)
tmb.SO <- mb.SO^(3/4)
plot(density(tmb.SO),main="Strikeouts");rug(tmb.SO)
plot(tmb.SO,StanRes1,xlab="Strikeouts",ylab="Standardized Residuals");abline(lsfit(tmb.SO,StanRes1),lty
qqnorm(tmb.SO,ylab="Y");qqline(tmb.SO,lty=2,col=2)
```

```r
#STOLEN BASES:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mb.SB <- lm.smk$TEAM_BASERUN_SB
m1 <- lm(mW~mb.SB, data = lm.smk)
StanRes1 <- rstandard(m1)
plot(density(mb.SB),main="Stolen Bases");rug(mb.SB)
plot(mb.SB,StanRes1,xlab="Stolen Bases",ylab="Standardized Residuals");abline(lsfit(mb.SB,StanRes1),lty
qqnorm(mb.SB,ylab="Y");qqline(mb.SB,lty=2,col=2)
##TRANSFORMATION
powerTransform(mb.SB, family="bcPower")$lambda# round(-0.03916886) => y^(-1/25)
tmb.SB <- mb.SB^(-1/25)
plot(density(tmb.SB),main="Stolen Bases");rug(tmb.SB)
plot(tmb.SB,StanRes1,xlab="Stolen Bases",ylab="Standardized Residuals");abline(lsfit(tmb.SB,StanRes1),l
qqnorm(tmb.SB,ylab="Y");qqline(tmb.SB,lty=2,col=2)

#PITCHING HITS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mp.H <- lm.smk$TEAM_PITCHING_H
m1 <- lm(mW~mp.H, data = lm.smk)
plot(density(mp.H),main="Pitching Hits");rug(mp.H)
plot(mp.H,StanRes1,xlab="Pitching Hits",ylab="Standardized Residuals");abline(lsfit(mp.H,StanRes1),lty=
qqnorm(mp.H, ylab = "Y");qqline(mp.H, lty = 2, col=2)
##TRANSFORMATION
powerTransform(mp.H,family="bcPower")$lambda# round(-3.097364) => y^(-3)
tmp.H <- mp.H^(-3)
plot(density(tmp.H),main="Pitching Hits");rug(tmp.H)
plot(tmp.H,StanRes1,xlab="Pitching Hits",ylab="Standardized Residuals");abline(lsfit(tmp.H,StanRes1),lty
qqnorm(tmp.H,ylab="Y");qqline(tmp.H,lty=2,col=2)

#PITCHING WALKS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mp.BB <- lm.smk$TEAM_PITCHING_BB
m1 <- lm(mW~mp.BB, data = lm.smk)
plot(density(mp.BB),main="Pitching Walks");rug(mp.BB)
plot(mp.BB,StanRes1,xlab="Pitching Walks",ylab="Standardized Residuals");abline(lsfit(mp.BB,StanRes1),l
qqnorm(mp.BB, ylab = "Y");qqline(mp.BB,lty = 2,col=2)
##TRANSFORMATION
powerTransform(mp.BB, family="bcPower")$lambda# round(0.1609713) => y^(1/6)
tmp.BB <- mp.BB^(1/6)
plot(density(tmp.BB),main="Pitching Walks");rug(tmp.BB)
plot(tmp.BB,StanRes1,xlab="Pitching Walks",ylab="Standardized Residuals");abline(lsfit(tmp.BB,StanRes1)
qqnorm(tmp.BB,ylab="Y");qqline(tmp.BB,lty=2,col=2)

#PITCHING STRIKEOUTS:
par(mfrow=c(2,3))
mW  <- lm.smk$TARGET_WINS
mp.SO <- lm.smk$TEAM_PITCHING_SO
m1 <- lm(mW~mp.SO, data = lm.smk)
plot(density(mp.SO),main="Pitching Strikeouts");rug(mp.SO)
plot(mp.SO,StanRes1,xlab="Pitching Strikeouts",ylab="Standardized Residuals");abline(lsfit(mp.SO,StanRes
```

```
qqnorm(mp.SO, ylab = "Y");qqline(mp.SO,lty = 2,col=2)
##TRANSFORMATION
powerTransform(mp.SO, family="bcPower")$lambda#round(0.6522561) => y^(2/3)
tmp.SO <- mp.SO^(2/3)
plot(density(tmp.SO),main="Pitching Strikeouts");rug(tmp.SO)
plot(tmp.SO,StanRes1,xlab="Pitching Strikeouts",ylab="Standardized Residuals");abline(lsfit(tmp.SO,Stanl
qqnorm(tmp.SO,ylab="Y");qqline(tmp.SO,lty=2,col=2)

#FIELDING ERRORS:
par(mfrow=c(2,3))
mW   <- lm.smk$TARGET_WINS
mf.E <- lm.smk$TEAM_FIELDING_E
m1 <- lm(mW~mf.E, data = lm.smk)
plot(density(mf.E),main="Fielding Errors");rug(mf.E)
plot(mf.E,StanRes1,xlab="Fielding Errors",ylab="Standardized Residuals");abline(lsfit(mf.E,StanRes1),lty
qqnorm(mf.E, ylab = "Y");qqline(mf.E, lty = 2, col=2)
##TRANSFORMATION
powerTransform(mf.E,family="bcPower")$lambda#round(-0.939657) => (-9/10)
tmf.E <- mf.E^(-(9/10))
plot(density(tmf.E),main="Fielding Errors");rug(tmf.E)
plot(tmf.E,StanRes1,xlab="Fielding Errors",ylab="Standardized Residuals");abline(lsfit(tmf.E,StanRes1),l
qqnorm(tmf.E,ylab="Y");qqline(tmf.E,lty=2,col=2)

#DOUBLE PLAYS:
par(mfrow=c(2,3))
mW   <- lm.smk$TARGET_WINS
mf.DP <- lm.smk$TEAM_FIELDING_DP
m1 <- lm(mW~mf.DP, data = lm.smk)
plot(density(mf.DP),main="Fielding Doubleplays");rug(mf.DP)
plot(mf.DP,StanRes1,xlab="Fielding Doubleplays",ylab="Standardized Residuals");abline(lsfit(mf.DP,StanRe
qqnorm(mf.DP, ylab = "Y");qqline(mf.DP,lty = 2, col=2)
##TRANSFORMATION
powerTransform(mf.DP,family="bcPower")$lambda# round(1.49645) => 1.5
tmf.DP <- mf.DP^(3/2)
plot(density(tmf.DP),main="Fielding Doubleplays");rug(tmf.DP)
plot(tmf.DP,StanRes1,xlab="Fielding Doubleplays",ylab="Standardized Residuals");abline(lsfit(tmf.DP,Stan
qqnorm(tmf.DP,ylab="Y");qqline(tmf.DP,lty=2,col=2)

#FIELDING YIELD
par(mfrow=c(2,3))
mW   <- lm.smk$TARGET_WINS
mf.FY <- lm.smk$TEAM_FIELDING_E + lm.smk$TEAM_FIELDING_DP * 2
m1 <- lm(mW~mf.FY, data = lm.smk)
plot(density(mf.FY),main="Fielding Yield");rug(mf.FY)
plot(mf.FY,StanRes1,xlab="Fielding Yield",ylab="Standardized Residuals");abline(lsfit(mf.FY,StanRes1),l
qqnorm(mf.FY, ylab = "Y");qqline(mf.FY,lty = 2, col=2)
##TRANSFORMATION
powerTransform(mf.FY,family="bcPower")$lambda# round(-2.066982) => -2
tmf.FY <- mf.FY^(-2)
plot(density(tmf.FY),main="Fielding Yield");rug(tmf.FY)
plot(tmf.FY,StanRes1,xlab="Fielding Yield",ylab="Standardized Residuals");abline(lsfit(tmf.FY,StanRes1)
qqnorm(tmf.FY,ylab="Y");qqline(tmf.FY,lty=2,col=2)
```

```
par(mfrow=c(1,1))
```

Evaluate Correlations

Evaluate Correlation between predictors so as to not introduce collinearity into the model.

```
# assign model one "lm.1" to data set with all NA's imputed and "bad" leverage points removed
par(cex = 0.65)
corrplot(
    cor(lm.smk[c(1:13)]),
    type='lower',
    tl.srt=45,
    addshade="positive",
    addCoef.col = rgb(0,0,0,alpha=0.3),
    addCoefasPercent = TRUE)
nrow(lm.smk)
```

Model Selection Strategy

Start with p.Hits & p.Walks

```
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Hits <- tmp.H
p.Walks <- tmp.BB
m1 <- lm(Wins ~ p.Hits+p.Walks)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Hits+p.Walks)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 and no VIFs  > 5

#DIAGNOSTIC2.  generate Added Variable Plots: should show linear relationship between response & predic
par(mfrow=c(2,2))
avPlots(m1, ~.,ask=FALSE, id.n = 2)
#relationship is linear

#DIAGNOSTIC3.  generate Summary Diagnostic Plots
par(mfrow=c(2,2))
plot(m1)
#Upper Left plot "Residuals vs Fitted"
# clear predictable pattern
# uniform variability for all fitted values
#Upper Right
#  normality in residuals
```

```r
#Lower Right plot "Residuals vs. Leverage"
#  normal distribution, and uniform distribution of residuals
#  no significant leverage points

##DIAGNOSTIC4.  generate Standardized Residual Plots against each predictor
par(mfrow=c(2,2))
StanRest <- rstandard(m1)
plot(p.Hits,StanRest,ylab="Standardized Residuals")
plot(p.Walks,StanRest,ylab="Standardized Residuals")
plot(m1$fitted.values,StanRest,ylab="Standardized Residuals",xlab="Fitted Values")
#Examine plots for constant variability of residuals across ALL predictor.
# uniform distribution of residuals

#DIAGNOSTIC5.  generate plot of Y "response variable"" against Fitted Values "regression model"
par(mfrow = c(2,2))
plot(m1$fitted.values,Wins,xlab="Fitted Values",ylab=expression(Wins^lambda))
abline(lsfit(m1$fitted.values,Wins))
plot(m1)
# normal distribution, and uniform distribution of residuals
```

Add b.Singles & b.Doubles

```r
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Hits <- tmp.H
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
m1 <- lm(Wins ~ p.Hits+p.Walks+b.Singles+b.Doubles)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Hits+p.Walks+b.Singles+b.Doubles)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values of p.Hits >  0.05 so it gets removed
```

Removed p.Hits

```r
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
m1 <- lm(Wins ~ p.Walks+b.Singles+b.Doubles)
```

```
#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.Doubles)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 and no VIFs  > 5 and adjusted $R^2$ increased

#DIAGNOSTIC2.  generate Added Variable Plots: should show linear relationship between response & predic
par(mfrow=c(2,2))
avPlots(m1, ~.,ask=FALSE, id.n = 2)
#relationship is linear

#DIAGNOSTIC3.  generate Summary Diagnostic Plots
par(mfrow=c(2,2))
plot(m1)
##Upper Left plot "Residuals vs Fitted"
# clear predictable pattern
# uniform variability for all fitted values
#Upper Right
#  normality in residuals
#Lower Right plot "Residuals vs. Leverage"
#  normal distribution, and uniform distribution of residuals
#  no significant leverage points

#DIAGNOSTIC4.  generate Standardized Residual Plots against each predictor
par(mfrow=c(2,2))
StanRest <- rstandard(m1)
plot(p.Walks,StanRest,ylab="Standardized Residuals")
plot(b.Singles,StanRest,ylab="Standardized Residuals")
plot(b.Doubles,StanRest,ylab="Standardized Residuals")
plot(m1$fitted.values,StanRest,ylab="Standardized Residuals",xlab="Fitted Values")
#Examine plots for constant variability of residuals across ALL predictor.
# uniform distribution of residuals

#DIAGNOSTIC5.  generate plot of Y "response variable"" against Fitted Values "regression model"
par(mfrow = c(2,2))
plot(m1$fitted.values,Wins,xlab="Fitted Values",ylab=expression(Wins^lambda))
abline(lsfit(m1$fitted.values,Wins))
plot(m1)
# normal distribution, and uniform distribution of residuals
```

Added Stolen Bases and Double Plays

```
nrow(lm.smk)
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Walks <- tmp.BB
b.Singles <- tmb.1B
```

```r
b.Doubles <- tmb.2B
b.StolenBases <- tmb.SB
f.DoublePlays <- tmf.DP
m1 <- lm(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 and no VIFs  > 5

#DIAGNOSTIC2.  generate Added Variable Plots: should show linear relationship between response & predic
par(mfrow=c(2,2))
avPlots(m1, ~.,ask=FALSE, id.n = 2)
#relationship is linear

#DIAGNOSTIC3.  generate Summary Diagnostic Plots
par(mfrow=c(2,2))
plot(m1)
#Upper Left plot "Residuals vs Fitted"
# clear predictable pattern
# uniform variability for all fitted values
#Upper Right
#  normality in residuals
#Lower Right plot "Residuals vs. Leverage"
#  normal distribution, and uniform distribution of residuals
#  no significant leverage points

#DIAGNOSTIC4.  generate Standardized Residual Plots against each predictor
par(mfrow=c(2,2))
StanRest <- rstandard(m1)
plot(p.Walks,StanRest,ylab="Standardized Residuals")
plot(b.Singles,StanRest,ylab="Standardized Residuals")
plot(b.Doubles,StanRest,ylab="Standardized Residuals")
plot(b.StolenBases,StanRest,ylab="Standardized Residuals")
plot(f.DoublePlays,StanRest,ylab="Standardized Residuals")
plot(m1$fitted.values,StanRest,ylab="Standardized Residuals",xlab="Fitted Values")
#Examine plots for constant variability of residuals across ALL predictor.
# uniform distribution of residuals

#DIAGNOSTIC5.  generate plot of Y "response variable"" against Fitted Values "regression model"
par(mfrow = c(2,2))
plot(m1$fitted.values,Wins,xlab="Fitted Values",ylab=expression(Wins^lambda))
abline(lsfit(m1$fitted.values,Wins))
plot(m1)
#If plot doesn't shows a linear relationship with no pattern or skew the model lacks normality.

# normal distribution, and uniform distribution of residuals
```

Added b.Walks and p.Strikeouts

```
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
b.StolenBases <- tmb.SB
f.DoublePlays <- tmf.DP
b.Walks <- tmb.BB
p.StrikeOuts <- tmp.SO
m1 <- lm(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays+b.Walks+p.StrikeOuts)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays+b.Walks+p.StrikeOuts)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 but VIFs  > 5
#highest vif is b.Walks so it gets removed
```

Remove b.Walks

```
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
b.StolenBases <- tmb.SB
f.DoublePlays <- tmf.DP
p.StrikeOuts <- tmp.SO
m1 <- lm(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays+p.StrikeOuts)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays+p.StrikeOuts)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-value for b.Walks > 0.05 all VIFs < 5
```

Add b.StrikeOuts, & b.Slugging

```r
nrow(lm.smk)
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
b.StolenBases <- tmb.SB
f.DoublePlays <- tmf.DP
b.StrikeOuts <- tmb.SO
b.Slugging <- tmb.SL
m1 <- lm(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+f.DoublePlays+b.StrikeOuts+b.Slugging)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.Doubles+b.StolenBases+b.StrikeOuts+b.Slugging)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 and no VIFs  > 5

#DIAGNOSTIC2.  generate Added Variable Plots: should show linear relationship between response & predic
par(mfrow=c(2,2))
avPlots(m1, ~.,ask=FALSE, id.n = 2)

#relationship is linear

#DIAGNOSTIC3.  generate Summary Diagnostic Plots
par(mfrow=c(2,2))
plot(m1)
#Upper Left plot "Residuals vs Fitted"
# clear predictable pattern
# uniform variability for all fitted values
#Upper Right
#  normality in residuals
#Lower Right plot "Residuals vs. Leverage"
#  normal distribution, and uniform distribution of residuals
#  no significant leverage points

#DIAGNOSTIC4.  generate Standardized Residual Plots against each predictor
par(mfrow=c(2,2))
StanRest <- rstandard(m1)
plot(p.Walks,StanRest,ylab="Standardized Residuals")
plot(b.Singles,StanRest,ylab="Standardized Residuals")
plot(b.Doubles,StanRest,ylab="Standardized Residuals")
plot(b.StolenBases,StanRest,ylab="Standardized Residuals")
plot(f.DoublePlays,StanRest,ylab="Standardized Residuals")
plot(b.StrikeOuts,StanRest,ylab="Standardized Residuals")
plot(b.Slugging,StanRest,ylab="Standardized Residuals")
plot(m1$fitted.values,StanRest,ylab="Standardized Residuals",xlab="Fitted Values")
```

```
#Examine plots for constant variability of residuals across ALL predictor.
# uniform distribution of residuals

#DIAGNOSTIC5.  generate plot of Y "response variable"" against Fitted Values "regression model"
par(mfrow = c(2,2))
plot(m1$fitted.values,Wins,xlab="Fitted Values",ylab=expression(Wins^lambda))
abline(lsfit(m1$fitted.values,Wins))
plot(m1)
# normal distribution, and uniform distribution of residuals
```

Add b.Fielding

$$\widehat{wins} = \hat{\beta}_0 + \hat{\beta}_1 \times p.Walks\hat{\beta}_2 \times b.Singles + \hat{\beta}_3 \times b.Doubles + \hat{\beta}_4 \times b.StolenBases +$$

$$\hat{\beta}_5 \times f.DoublePlays + \hat{\beta}_6 \times b.StrikeOuts + \hat{\beta}_7 \times b.Slugging + \hat{\beta}_8 \times b.FieldingYield +$$

nrow(lm.smk)

```
#VARIABLES
#variables have been transformed first as individual predictors
Wins  <- mW
Index <- lm.smk$INDEX
p.Walks <- tmp.BB
b.Singles <- tmb.1B
b.Doubles <- tmb.2B
b.StolenBases <- tmb.SB
b.StrikeOuts <- tmb.SO
b.Slugging <- tmb.SL
f.Fielding <- tmf.FY

m1 <- lm(Wins ~ -Index+p.Walks+b.Singles+b.Doubles+b.StolenBases+b.StrikeOuts+b.Slugging+f.Fielding)

#PAIRWISE PLOT
par(mfrow=c(1,1))
pairs(Wins ~ p.Walks+b.Singles+b.StolenBases+b.StrikeOuts+b.Slugging+f.Fielding)

#MODEL DIAGNOSTICS
summary(m1)

#DIAGNOSTIC1. show collinearity of variables after checking p-values to < 0.05.
vif(m1)
#p-values are all < 0.05 and no VIFs  > 5

#DIAGNOSTIC2.  generate Added Variable Plots: should show linear relationship between response & predic
par(mfrow=c(2,2))
avPlots(m1, ~.,ask=FALSE, id.n = 2)

#relationship is linear

#DIAGNOSTIC3.  generate Summary Diagnostic Plots
par(mfrow=c(2,2))
plot(m1)
#Upper Left plot "Residuals vs Fitted"
```

```
# clear predictable pattern
# uniform variability for all fitted values
#Upper Right
#  normality in residuals
#Lower Right plot "Residuals vs. Leverage"
#  normal distribution, and uniform distribution of residuals
#  no significant leverage points

#DIAGNOSTIC4.  generate Standardized Residual Plots against each predictor
par(mfrow=c(2,2))
StanRest <- rstandard(m1)
plot(p.Walks,StanRest,ylab="Standardized Residuals")
plot(b.Singles,StanRest,ylab="Standardized Residuals")
plot(b.Doubles,StanRest,ylab="Standardized Residuals")
plot(b.StolenBases,StanRest,ylab="Standardized Residuals")
plot(f.DoublePlays,StanRest,ylab="Standardized Residuals")
plot(b.StrikeOuts,StanRest,ylab="Standardized Residuals")
plot(b.Slugging,StanRest,ylab="Standardized Residuals")
plot(f.Fielding,StanRest,ylab="Standardized Residuals")
plot(m1$fitted.values,StanRest,ylab="Standardized Residuals",xlab="Fitted Values")
#Examine plots for constant variability of residuals across ALL predictor.
# uniform distribution of residuals

#DIAGNOSTIC5.  generate plot of Y "response variable"" against Fitted Values "regression model"
par(mfrow = c(2,2))
plot(m1$fitted.values,Wins,xlab="Fitted Values",ylab=expression(Wins^lambda))
abline(lsfit(m1$fitted.values,Wins))
plot(m1)
# normal distribution, and uniform distribution of residuals
```

```
#pred_eval.m1 <- round(predict(m1, eval_data))
#eval.BSO.imp <- impute(eval_data$TARGET_WINS, pred_eval.m1)
```

Part 4. Select Models

```
##############################################################################################
#
# This file loads our preferred predictive model (TOTAL BASES PLUS)
# and uses that model to predict the TARGET_WINS variable of the MLB Evaluation
# data set.
#
# When finished, two separate files are written to a local hard disk directory:
#
# - one containing the entire EVALUATION data set after the TARGET_WINS variable has
#   been updated with the predicted values for each record;
#
# - one containing ONLY the INDEX and TARGET_WINS variables from the EVALUATION data set
#
# - NO screen output is generated at all by this code
#
```

```
#############################################################################################

# ------------------------------------------------------------
# read clean data set from Github

mb_clean <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clean

# ------------------------------------------------------------

# Build a model with Total Bases + SB + BB added and all of the other hitting vars removed

# create new variable and drop its components
mb_t <- mb_clean

mb_t$TB_PLUS <- mb_clean$TEAM_BATTING_1B + (2 * mb_clean$TEAM_BATTING_2B) +
                        (3 * mb_clean$TEAM_BATTING_3B) + (4 * mb_clean$TEAM_BATTING_HR) +
                        mb_clean$TEAM_BATTING_BB + mb_clean$TEAM_BASERUN_SB

# par(mfrow = c(1,1))
# hist(mb_t$TB_PLUS, breaks = 200)

# now drop 1B, 2B, 3B, HR, BB, SB
mb_tbp <- mb_t[,c(1, 2, 7, 9, 10, 11, 12, 13, 15)]


# ---------------------------------------------------------------------------------------------

# REMOVE OUTLIERS AND REFIT

# Per Cooks Distance, remove items 836, 821, 1920, 1737, 1515


############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
mb_rem <- mb_tbp[-c(836, 821, 1920, 1737, 1515),]

# save first data set
mb_tbp_orig <- mb_tbp

# renumber rows
rownames(mb_rem) <- 1:nrow(mb_rem)

# keep the clean data set pure
mb_tbp <- mb_rem

# ---------------------------------------------------------------------------------------------

## Now try same model but with FIELD_E transformed using Box-Cox

# TEAM_FIELDING_E: Box-cox says -1 power transform => 1/y

mb_tbp$TEAM_FIELDING_E <- 1/mb_tbp$TEAM_FIELDING_E
```

```
# Now refit first model from above: Start with all variables
model.4 <- lm(data=mb_tbp, TARGET_WINS ~ . - INDEX - TEAM_PITCHING_BB - TEAM_PITCHING_H - TEAM_PITCHING

# summary(model.4)
```

```
# Now load evaluation data set and predict TARGET WINS

# load EVAL data set
eval.d <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/621-HW1-Clean-

# save original data
eval.2 <- eval.d

# creaet TB_PLUS and drop component variables
eval.2$TB_PLUS <- eval.2$TEAM_BATTING_1B + (2 * eval.2$TEAM_BATTING_2B) +
                        (3 * eval.2$TEAM_BATTING_3B) + (4 * eval.2$TEAM_BATTING_HR) +
                        eval.2$TEAM_BATTING_BB + eval.2$TEAM_BASERUN_SB

# par(mfrow = c(1,1))
# hist(eval.d$TB_PLUS, breaks = 30)

# now drop 1B, 2B, 3B, HR, BB, SB
eval.2 <- eval.2[,c(1, 6, 8, 9, 10, 11, 12, 14, 15)]

# transform TEAM_FIELDING_E using 1/y
eval.2$TEAM_FIELDING_E <- 1/eval.2$TEAM_FIELDING_E


# now predict TARGET_WINS using model.4
pred.TW <- round(predict(model.4, eval.2))

# add predicted variables to TARGET_WINS variable
eval.2$TARGET_WINS <- pred.TW
eval.d$TARGET_WINS <- pred.TW
```

```
# write entire updated EVAL data set to a CSV
write.csv(eval.d, file = "C:/SQLData/HW1-PRED-EVAL-ALLDATA.csv", row.names = FALSE)

# write full model EVAL data to a CSV file
write.csv(eval.d, file = "C:/SQLData/HW1-PRED-EVAL-ALL_M_DATA.csv", row.names = FALSE)

# now write just INDEX and TARGET_WINS to a separate file
eval.3 <- eval.2[,c(1,8)]

write.csv(eval.3, file = "C:/SQLData/HW1-PRED-EVAL-WINS-ONLY.csv", row.names = FALSE)

# end

#look at summary statistics

eval.w <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W1/master/HW1-PRED-EVAL-
define(eval.w)
```

```r
# clean up objects in memory
rm(list = ls())
```