Critical Thinking Group 4 : DATA621 Homework 1

Table of Contents

TEAM Members:

Rajwant Mishra
Priya Shaji
Debabrata Kabiraj
Isabel Ramesar
Sin Ying Wong
Fan Xu

---

1 Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season.

We have been given a dataset with 2276 records summarizing a major league baseball team's season. The records span 1871 to 2006 inclusive. All statistics have been adjusted to match the performance of a 162 game season.

Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided).

Glossary of data

Code

Below is a short description of the variables of interest in the data set:

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of wins	
TEAM_BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
TEAM_BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
TEAM_BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
TEAM_BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
TEAM_BATTING_BB	Walks by batters	Positive Impact on Wins
TEAM_BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
TEAM_BATTING_SO	Strikeouts by batters	Negative Impact on Wins
TEAM_BASERUN_SB	Stolen bases	Positive Impact on Wins
TEAM_BASERUN_CS	Caught stealing	Negative Impact on Wins
TEAM_FIELDING_E	Errors	Negative Impact on Wins
TEAM_FIELDING_DP	Double Plays	Positive Impact on Wins
TEAM_PITCHING_BB	Walks allowed	Negative Impact on Wins
TEAM_PITCHING_H	Hits allowed	Negative Impact on Wins
TEAM_PITCHING_HR	Homeruns allowed	Negative Impact on Wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

2 Deliverables

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

3 DATA EXPLORATION

The data set describes baseball team statistics for the years 1871 to 2006 inclusive. Each record in the data set represents the performance of the team for the given year adjusted to the current length of the season - 162 games. The data set includes 16 variables and the training set includes 2,276 records.

Load the data and understand the data by using some stats and plot

Code

3.1 View rows and columns, variable types

Glimpse of the data shows that all variables are numeric, no categorical variable is present here. We do lots of NA for few predictors in the data set. In our further analysis we will try to identify:

- Structure of the each predictors
- How Many NA and Zero , is it significant to remove them or replace them with some predicted value.
- Statistical summary of the data

Code

```
## Observations: 2,276
## Variables: 17
## $ INDEX            <int> 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 15, 16, 17, 18...
## $ TARGET_WINS      <int> 39, 70, 86, 70, 82, 75, 80, 85, 86, 76, 78, 68, 72...
## $ TEAM_BATTING_H   <int> 1445, 1339, 1377, 1387, 1297, 1279, 1244, 1273, 13...
## $ TEAM_BATTING_2B  <int> 194, 219, 232, 209, 186, 200, 179, 171, 197, 213, ...
## $ TEAM_BATTING_3B  <int> 39, 22, 35, 38, 27, 36, 54, 37, 40, 18, 27, 31, 41...
## $ TEAM_BATTING_HR  <int> 13, 190, 137, 96, 102, 92, 122, 115, 114, 96, 82, ...
## $ TEAM_BATTING_BB  <int> 143, 685, 602, 451, 472, 443, 525, 456, 447, 441, ...
## $ TEAM_BATTING_SO  <int> 842, 1075, 917, 922, 920, 973, 1062, 1027, 922, 82...
## $ TEAM_BASERUN_SB  <int> NA, 37, 46, 43, 49, 107, 80, 40, 69, 72, 60, 119, ...
## $ TEAM_BASERUN_CS  <int> NA, 28, 27, 30, 39, 59, 54, 36, 27, 34, 39, 79, 10...
## $ TEAM_BATTING_HBP <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ TEAM_PITCHING_H  <int> 9364, 1347, 1377, 1396, 1297, 1279, 1244, 1281, 13...
## $ TEAM_PITCHING_HR <int> 84, 191, 137, 97, 102, 92, 122, 116, 114, 96, 86, ...
## $ TEAM_PITCHING_BB <int> 927, 689, 602, 454, 472, 443, 525, 459, 447, 441, ...
## $ TEAM_PITCHING_SO <int> 5456, 1082, 917, 928, 920, 973, 1062, 1033, 922, 8...
## $ TEAM_FIELDING_E  <int> 1011, 193, 175, 164, 138, 123, 136, 112, 127, 131,...
## $ TEAM_FIELDING_DP <int> NA, 155, 153, 156, 168, 149, 186, 136, 169, 159, 1...
```

Sample 6 rows with sample 7 columns

Code

	INDEX <int>	TARGET_WINS <int>	TEAM_BATTING_H <int>	TEAM_BATTING_2B <int>	TEAM_BATTING_3B <int>	TEAM_BATTING_HR <int>
1	1	39	1445	194	39	13
2	2	70	1339	219	22	190
3	3	86	1377	232	35	137
4	4	70	1387	209	38	96
5	5	82	1297	186	27	102

	INDEX <int>	TARGET_WINS <int>	TEAM_BATTING_H <int>	TEAM_BATTING_2B <int>	TEAM_BATTING_3B <int>	TEAM_BATTING_HR <int>	
6	6	75	1279	200	36	92	

6 rows | 1-7 of 18 columns

Show entire dataset of training data:

	INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTING_BB	TEAM_BATTING_SO	TEAM_BASERUN_SB	TEAM_BASERUN_CS	TEAM_BATTING_HBP	TEAM_PITCHING_H	TEAM_PITCHING_HR	TEAM_PITCHING_BB	TEAM_PITCHING_SO	TEAM_FIELDING_E	TEAM_FIELDING_DP
1	1	39	1445	194	39	13	143	842				9364	84	927	5456	1011	
2	2	70	1339	219	22	190	685	1075	37	28		1347	191	689	1082	193	155
3	3	86	1377	232	35	137	602	917	46	27		1377	137	602	917	175	153
4	4	70	1387	209	38	96	451	922	43	30		1396	97	454	928	164	156
5	5	82	1297	186	27	102	472	920	49	39		1297	102	472	920	138	168
6	6	75	1279	200	36	92	443	973	107	59		1279	92	443	973	123	149
7	7	80	1244	179	54	122	525	1062	80	54		1244	122	525	1062	136	186
8	8	85	1273	171	37	115	456	1027	40	36		1281	116	459	1033	112	136
9	11	86	1391	197	40	114	447	922	69	27		1391	114	447	922	127	169
10	12	76	1271	213	18	96	441	827	72	34		1271	96	441	827	131	159

Showing 1 to 10 of 2,276 entries

Previous12345…228Next

Show entire dataset of evaluation data

	INDEX	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTING_BB	TEAM_BATTING_SO	TEAM_BASERUN_SB	TEAM_BASERUN_CS	TEAM_BATTING_HBP	TEAM_PITCHING_H	TEAM_PITCHING_HR	TEAM_PITCHING_BB	TEAM_PITCHING_SO	TEAM_FIELDING_E	TEAM_FIELDING_DP
1	9	1209	170	33	83	447	1080	62	50		1209	83	447	1080	140	156
2	10	1221	151	29	88	516	929	54	39		1221	88	516	929	135	164
3	14	1395	183	29	93	509	816	59	47		1395	93	509	816	156	153
4	47	1539	309	29	159	486	914	148	57	42	1539	159	486	914	124	154
5	60	1445	203	68	5	95	416				3902	14	257	1123	616	130
6	63	1431	236	53	10	215	377				2793	20	420	736	572	105
7	74	1430	219	55	37	568	527	365			1544	40	613	569	490	
8	83	1385	158	42	33	356	609	185			1626	39	418	715	328	104
9	98	1259	177	78	23	466	689	150			1342	25	497	734	226	132
10	120	1397	212	42	58	452	584	52			1489	62	482	622	184	145

Showing 1 to 10 of 259 entries

Previous12345…26Next

3.2 Structure of data

Dimension of Test dataset is, 2276 X 17 with 2276 number of observation in test data.

Summary of the test data shows very clearly that we have six predictors which has NA and `BATTING_HBP` and `BASERUN_CS` have the max number of NAs in the data set.

```
##       INDEX         TARGET_WINS      TEAM_BATTING_H TEAM_BATTING_2B
```

```
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
##  1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
##  NA's   :131     NA's   :772     NA's   :2085
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##                                    NA's   :102
##  TEAM_FIELDING_DP
##  Min.   : 52.0
##  1st Qu.:131.0
##  Median :149.0
##  Mean   :146.4
##  3rd Qu.:164.0
##  Max.   :228.0
##  NA's   :286
```

Code

```
## mtd
##
##  17  Variables      2276  Observations
## --------------------------------------------------------------------------------
## INDEX
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0     2276        1     1268    850.4    125.8    252.5
##      .25      .50      .75      .90      .95
##    630.8   1270.5   1915.5   2287.5   2407.2
##
## lowest :    1    2    3    4    5, highest: 2531 2532 2533 2534 2535
## --------------------------------------------------------------------------------
## TARGET_WINS
```

```
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      108        1    80.79    17.47     54.0     61.0
##      .25      .50      .75      .90      .95
##     71.0     82.0     92.0     99.5    104.0
##
## lowest :   0  12  14  17  21, highest: 128 129 134 135 146
## ------------------------------------------------------------------------
## TEAM_BATTING_H
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      569        1     1469    149.8     1282     1315
##      .25      .50      .75      .90      .95
##     1383     1454     1537     1636     1695
##
## lowest :  891  992 1009 1116 1122, highest: 2333 2343 2372 2496 2554
## ------------------------------------------------------------------------
## TEAM_BATTING_2B
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      240        1    241.2    52.89      167      182
##      .25      .50      .75      .90      .95
##      208      238      273      303      320
##
## lowest :  69 112 113 118 123, highest: 382 392 393 403 458
## ------------------------------------------------------------------------
## TEAM_BATTING_3B
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      144        1    55.25    30.34       23       27
##      .25      .50      .75      .90      .95
##       34       47       72       96      108
##
## lowest :   0   8   9  11  12, highest: 166 190 197 200 223
## ------------------------------------------------------------------------
## TEAM_BATTING_HR
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      243        1    99.61    69.49     14.0     20.0
##      .25      .50      .75      .90      .95
##     42.0    102.0    147.0    179.5    199.0
##
## lowest :   0   3   4   5   6, highest: 247 249 257 260 264
## ------------------------------------------------------------------------
## TEAM_BATTING_BB
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      533        1    501.6    130.1    248.2    363.5
##      .25      .50      .75      .90      .95
##    451.0    512.0    580.0    635.0    670.2
##
## lowest :   0  12  29  34  45, highest: 815 819 824 860 878
## ------------------------------------------------------------------------
## TEAM_BATTING_SO
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2174      102      822        1    735.6    282.2      359      421
##      .25      .50      .75      .90      .95
##      548      750      930     1049     1103
##
## lowest :    0   66   67   72   74, highest: 1303 1320 1326 1335 1399
## ------------------------------------------------------------------------
```

```
## TEAM_BASERUN_SB
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2145      131      348        1    124.8    87.96     35.0     44.0
##      .25      .50      .75      .90      .95
##     66.0    101.0    156.0    231.0    301.8
##
## lowest :   0  14  18  19  20, highest: 562 567 632 654 697
## ------------------------------------------------------------------------------
## TEAM_BASERUN_CS
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1504      772      128        1     52.8    23.24       24       30
##      .25      .50      .75      .90      .95
##       38       49       62       77       91
##
## lowest :   0   7  11  12  14, highest: 171 186 193 200 201
## ------------------------------------------------------------------------------
## TEAM_BATTING_HBP
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      191     2085       55    0.999    59.36    14.61     40.0     44.0
##      .25      .50      .75      .90      .95
##     50.5     58.0     67.0     76.0     82.5
##
## lowest : 29 30 35 38 39, highest: 87 88 89 90 95
## ------------------------------------------------------------------------------
## TEAM_PITCHING_H
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      843        1     1779    628.1     1316     1356
##      .25      .50      .75      .90      .95
##     1419     1518     1682     2058     2563
##
## lowest :  1137  1168  1184  1187  1202, highest: 16038 16871 20088 24057 30132
## ------------------------------------------------------------------------------
## TEAM_PITCHING_HR
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      256        1    105.7    70.02     18.0     25.0
##      .25      .50      .75      .90      .95
##     50.0    107.0    150.0    187.0    209.2
##
## lowest :   0   3   4   5   6, highest: 291 297 301 320 343
## ------------------------------------------------------------------------------
## TEAM_PITCHING_BB
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      535        1      553    140.7    377.0    417.5
##      .25      .50      .75      .90      .95
##    476.0    536.5    611.0    693.5    757.0
##
## lowest :    0  119  124  131  140, highest: 2169 2396 2840 2876 3645
## ------------------------------------------------------------------------------
## TEAM_PITCHING_SO
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2174      102      823        1    817.7    316.9    421.3    490.0
##      .25      .50      .75      .90      .95
##    615.0    813.5    968.0   1095.0   1173.0
##
## lowest :     0   181   205   208   252, highest:  3450  4224  5456 12758 19278
```

```
##
## Value          0   200   400   600   800  1000  1200  1400  1600  1800  2200
## Frequency     20     7   211   554   593   580   156    35     7     2     1
## Proportion 0.009 0.003 0.097 0.255 0.273 0.267 0.072 0.016 0.003 0.001 0.000
##
## Value       2400  3400  4200  5400 12800 19200
## Frequency      3     1     1     1     1     1
## Proportion 0.001 0.000 0.000 0.000 0.000 0.000
##
## For the frequency table, variable is rounded to the nearest 200
## ----------------------------------------------------------------------------
## TEAM_FIELDING_E
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2276        0      549        1    246.5    190.4    100.0    109.0
##      .25      .50      .75      .90      .95
##    127.0    159.0    249.2    542.0    716.0
##
## lowest :   65   66   68   72   74, highest: 1567 1728 1740 1890 1898
## ----------------------------------------------------------------------------
## TEAM_FIELDING_DP
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1990      286      144        1    146.4    29.29       98      109
##      .25      .50      .75      .90      .95
##      131      149      164      178      186
##
## lowest :  52  64  68  71  72, highest: 215 218 219 225 228
## ----------------------------------------------------------------------------
Code
##  [1] "INDEX"            "TARGET_WINS"      "TEAM_BATTING_H"   "TEAM_BATTING_2B"
##  [5] "TEAM_BATTING_3B"  "TEAM_BATTING_HR"  "TEAM_BATTING_BB"  "TEAM_BATTING_SO"
##  [9] "TEAM_BASERUN_SB"  "TEAM_BASERUN_CS"  "TEAM_BATTING_HBP" "TEAM_PITCHING_H"
## [13] "TEAM_PITCHING_HR" "TEAM_PITCHING_BB" "TEAM_PITCHING_SO" "TEAM_FIELDING_E"
## [17] "TEAM_FIELDING_DP"
Code
## 'data.frame':    2276 obs. of  17 variables:
##  $ INDEX           : int  1 2 3 4 5 6 7 8 11 12 ...
##  $ TARGET_WINS     : int  39 70 86 70 82 75 80 85 86 76 ...
##  $ TEAM_BATTING_H  : int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
##  $ TEAM_BATTING_2B : int  194 219 232 209 186 200 179 171 197 213 ...
##  $ TEAM_BATTING_3B : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ TEAM_BATTING_HR : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ TEAM_BATTING_BB : int  143 685 602 451 472 443 525 456 447 441 ...
##  $ TEAM_BATTING_SO : int  842 1075 917 922 920 973 1062 1027 922 827 ...
##  $ TEAM_BASERUN_SB : int  NA 37 46 43 49 107 80 40 69 72 ...
##  $ TEAM_BASERUN_CS : int  NA 28 27 30 39 59 54 36 27 34 ...
##  $ TEAM_BATTING_HBP: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TEAM_PITCHING_H : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
##  $ TEAM_PITCHING_HR: int  84 191 137 97 102 92 122 116 114 96 ...
##  $ TEAM_PITCHING_BB: int  927 689 602 454 472 443 525 459 447 441 ...
##  $ TEAM_PITCHING_SO: int  5456 1082 917 928 920 973 1062 1033 922 827 ...
##  $ TEAM_FIELDING_E : int  1011 193 175 164 138 123 136 112 127 131 ...
##  $ TEAM_FIELDING_DP: int  NA 155 153 156 168 149 186 136 169 159 ...
```

3.3 Mean and Median of the data

Code

	INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTI BB
	Min. : 1.0	Min. : 0.00	Min. : 891	Min. : 69.0	Min. : 0.00	Min. : 0.00	Min. : 0.0
	1st Qu.: 630.8	1st Qu.: 71.00	1st Qu.:1383	1st Qu.:208.0	1st Qu.: 34.00	1st Qu.: 42.00	1st Qu.:451.0
	Median :1270.5	Median : 82.00	Median :1454	Median :238.0	Median : 47.00	Median :102.00	Median :512.0
	Mean :1268.5	Mean : 80.79	Mean :1469	Mean :241.2	Mean : 55.25	Mean : 99.61	Mean :501.6
	3rd Qu.:1915 .5	3rd Qu.: 92.00	3rd Qu.:1537	3rd Qu.:273.0	3rd Qu.: 72.00	3rd Qu.:147.00	3rd Qu.:580.0
	Max. :2535.0	Max. :146.00	Max. :2554	Max. :458.0	Max. :223.00	Max. :264.00	Max. :878.0
	NA	NA	NA	NA	NA	NA	NA

`BATTING_HBP` is showing very close mean and median value, and we suspect its due less number of datapoints. Remember we noted highest number of NA in this predictor. Apart from `FIELDING_E` we don't see any big difference in the mean and median of the data.

3.4 Rename Columns

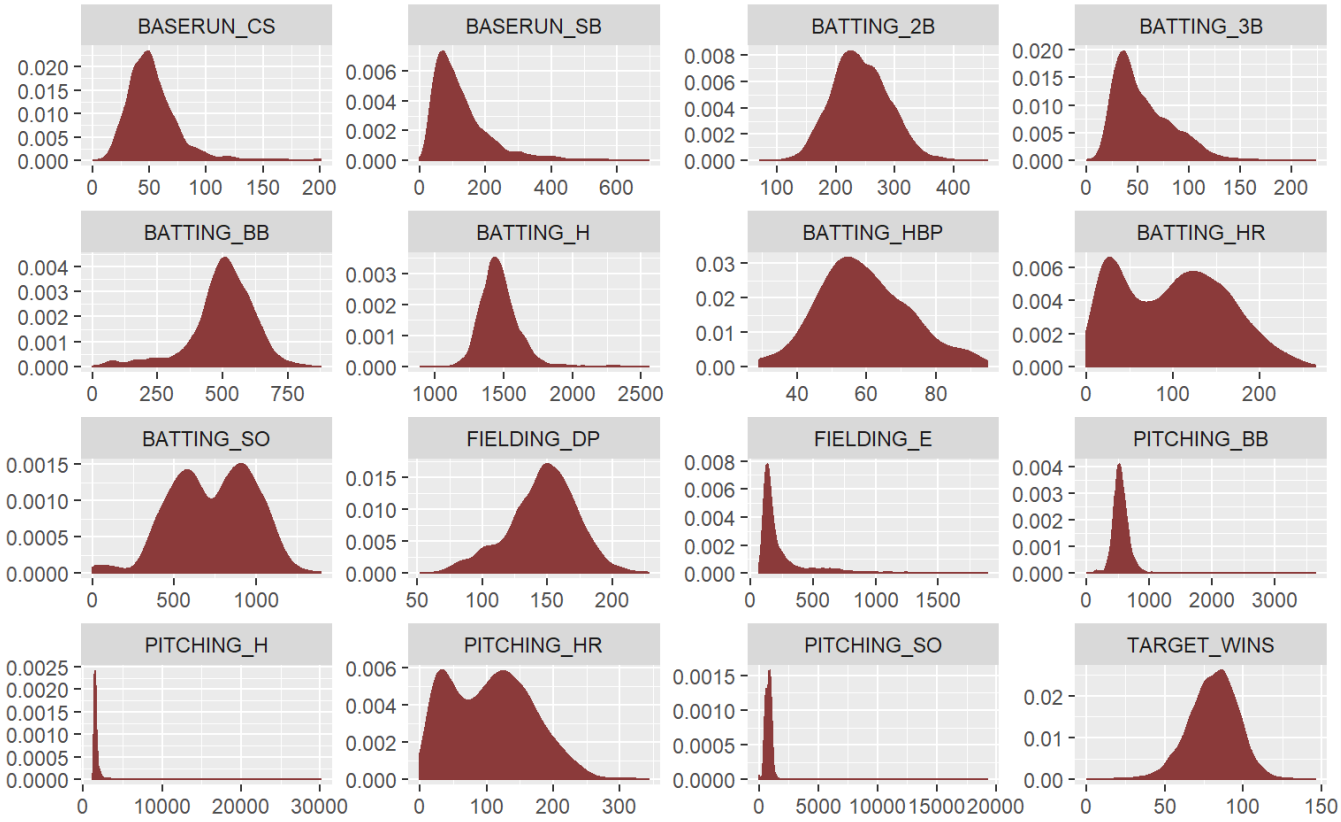
Here we removing the `TEAM_` from the column name so that we can display it in the plots, and make it easy to read.

Names Before: INDEX, TARGET_WINS, TEAM_BATTING_H, TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATTING_HR, TEAM_BATTING_BB, TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_H, TEAM_PITCHING_HR, TEAM_PITCHING_BB, TEAM_PITCHING_SO, TEAM_FIELDING_E, TEAM_FIELDING_DP

Code

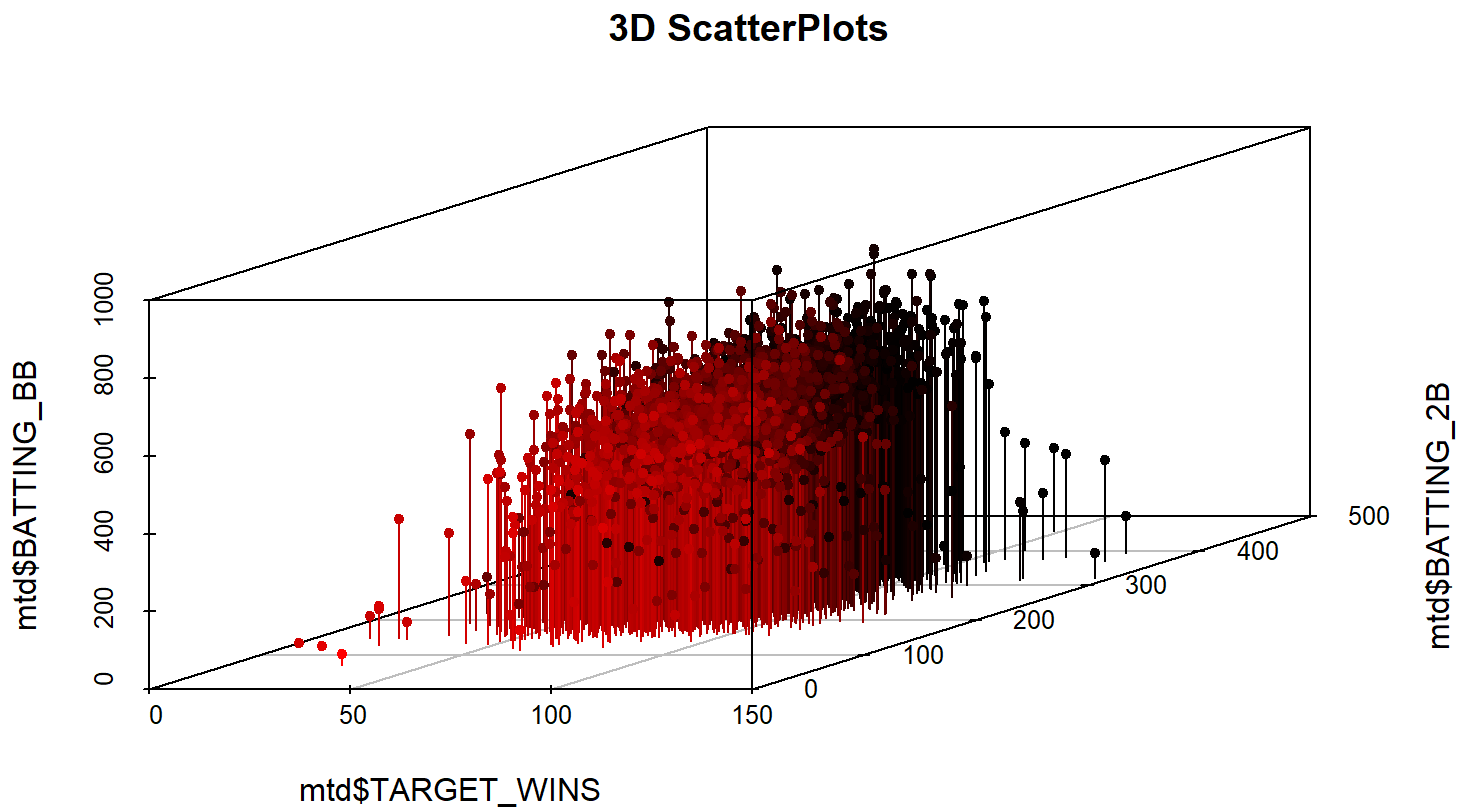
Names After : TARGET_WINS, BATTING_H, BATTING_2B, BATTING_3B, BATTING_HR, BATTING_BB, BATTING_SO, BASERUN_SB, BASERUN_CS, BATTING_HBP, PITCHING_H, PITCHING_HR, PITCHING_BB, PITCHING_SO, FIELDING_E, FIELDING_DP

3.5 Visualize the data



In the histogram plot above, we see that the batting, pitching home-run and batting strike-out variables are bi modal. `TARGET_WINS` and `TEAM_BATTING_2B` has most the normal distribution. `PITCHING_H` and `PITCHING_SO` have the most skewed data distribution. The skewed graphs are all rght-skewed except `BATTING_BB`.

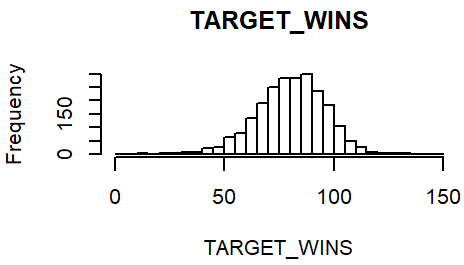
Code

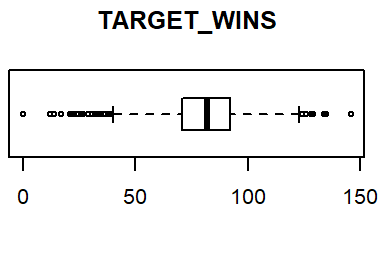


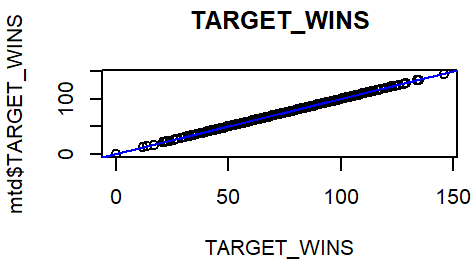
The above 3-D scatter plot, shows the data variance between the `TARGET_WINS`, `TEAM_BATTING_2B` and `TEAM_BATTING_BB` to provide a comparative 3D view.

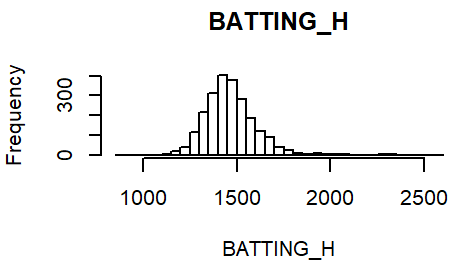
3.6 Multivariate Plot

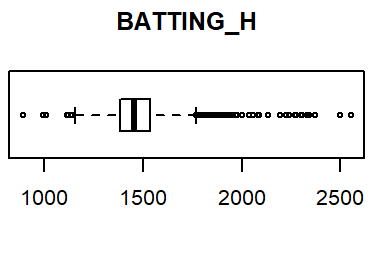
We will evalaute `Frequency (Histogram of Variables)` and `Regression fit` of each predictor with `TARGET_WIN`.

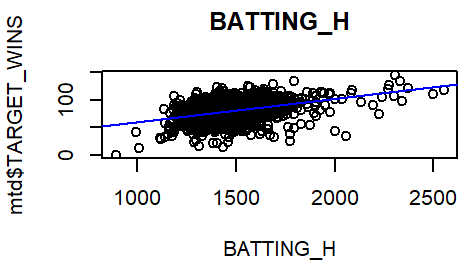

TARGET_WINS
Frequency
0
150
0
50
100
150
TARGET_WINS

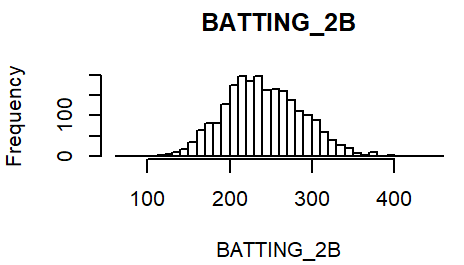

TARGET_WINS
0
50
100
150

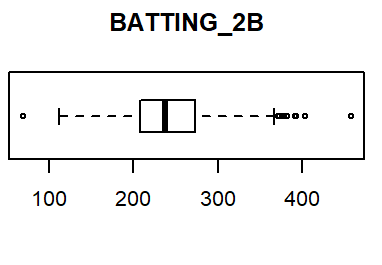

TARGET_WINS
mtd$TARGET_WINS
0
100
0
50
100
150
TARGET_WINS

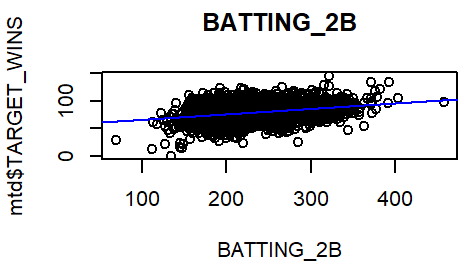

BATTING_H
Frequency
0
300
1000
1500
2000
2500
BATTING_H

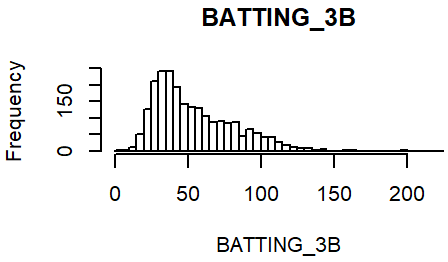

BATTING_H
1000
1500
2000
2500

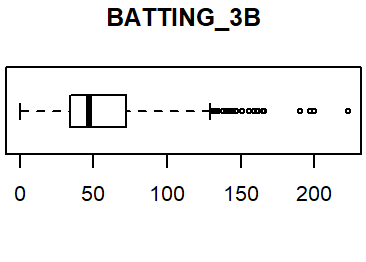

BATTING_H
mtd$TARGET_WINS
0
100
1000
1500
2000
2500
BATTING_H

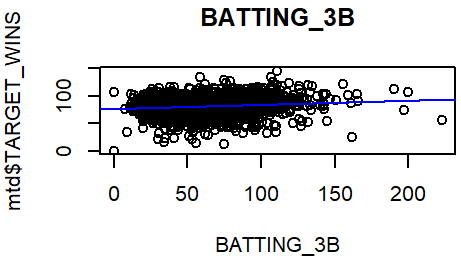

BATTING_2B
Frequency
0
100
100
200
300
400
BATTING_2B

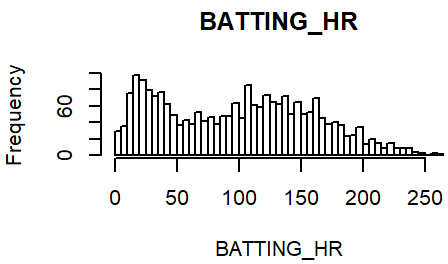

BATTING_2B
100
200
300
400

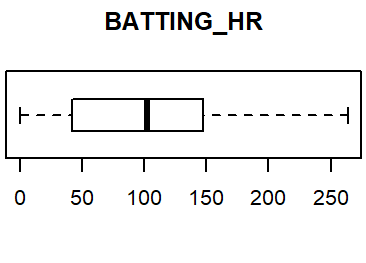

BATTING_2B
mtd$TARGET_WINS
0
100
100
200
300
400
BATTING_2B

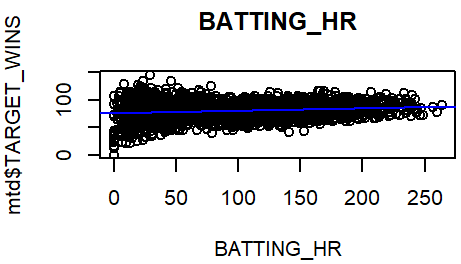

BATTING_3B
Frequency
0
150
0
50
100
150
200
BATTING_3B

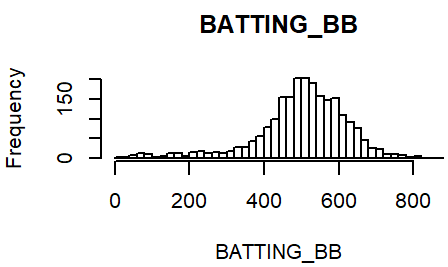

BATTING_3B
0
50
100
150
200

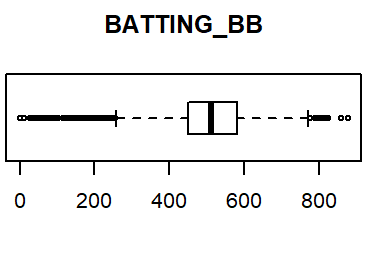

BATTING_3B
mtd$TARGET_WINS
0
100
0
50
100
150
200
BATTING_3B

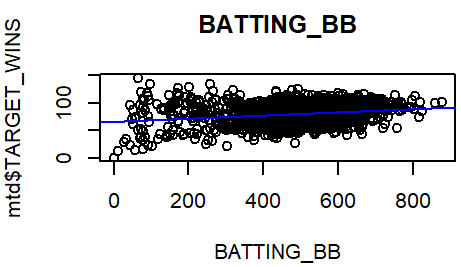

BATTING_HR
Frequency
0
60
0
50
100
150
200
250
BATTING_HR

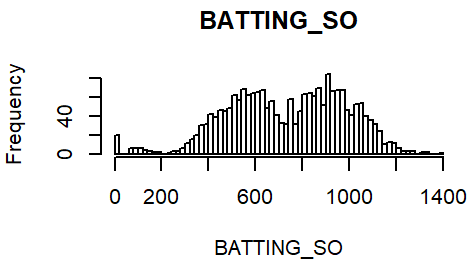

BATTING_HR
0
50
100
150
200
250

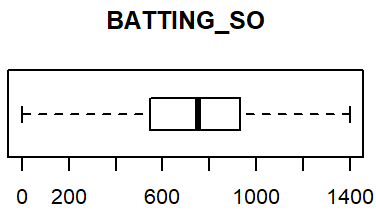

BATTING_HR
mtd$TARGET_WINS
0
100
0
50
100
150
200
250
BATTING_HR

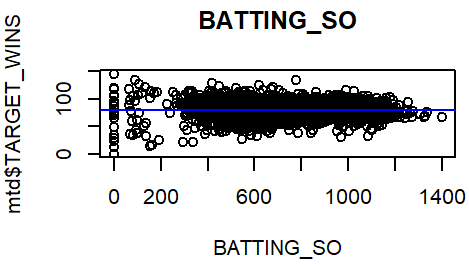

BATTING_BB
Frequency
0
150
0
200
400
600
800
BATTING_BB

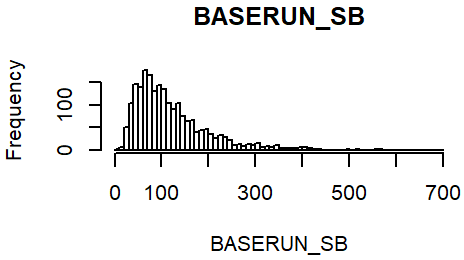

BATTING_BB
0
200
400
600
800

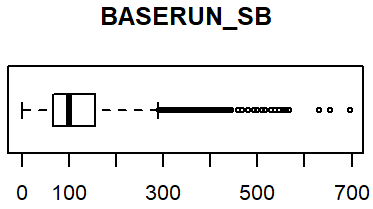

BATTING_BB
mtd$TARGET_WINS
0
100
0
200
400
600
800
BATTING_BB

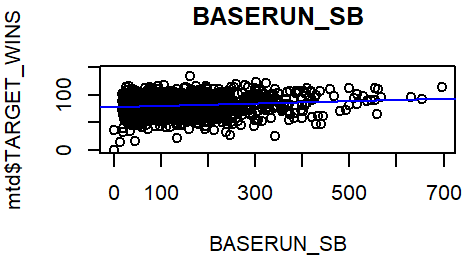

BATTING_SO
Frequency
0
40
0
200
600
1000
1400
BATTING_SO

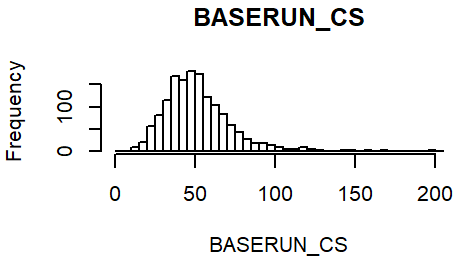

BATTING_SO
0
200
600
1000
1400

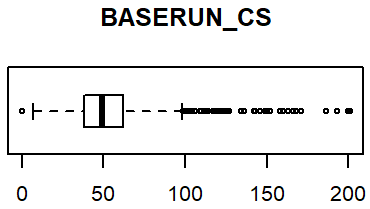

BATTING_SO
mtd$TARGET_WINS
0
100
0
200
600
1000
1400
BATTING_SO

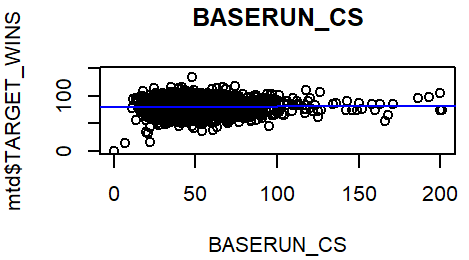

BASERUN_SB
Frequency
0
100
0
100
300
500
700
BASERUN_SB

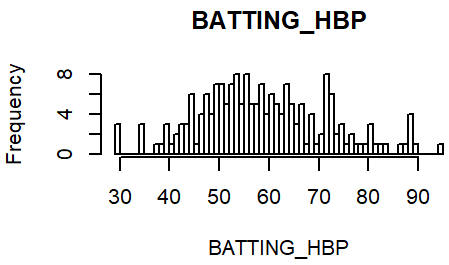

BASERUN_SB
0
100
300
500
700

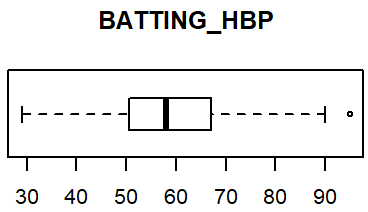

BASERUN_SB
mtd$TARGET_WINS
0
100
0
100
300
500
700
BASERUN_SB

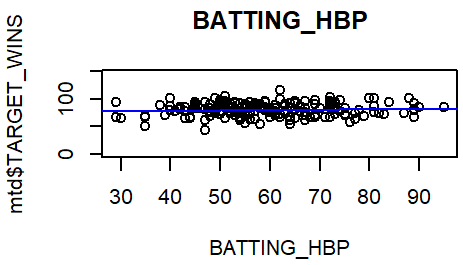

BASERUN_CS
Frequency
0
100
0
50
100
150
200
BASERUN_CS

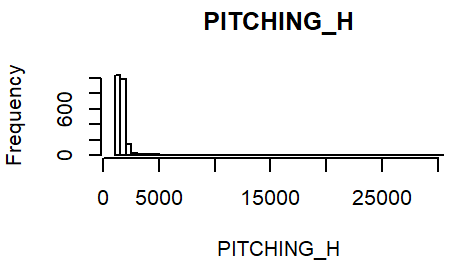

BASERUN_CS
0
50
100
150
200

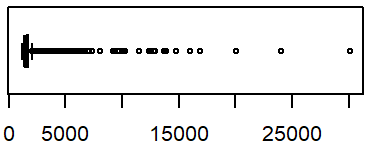

BASERUN_CS
mtd$TARGET_WINS
0
100
0
50
100
150
200
BASERUN_CS

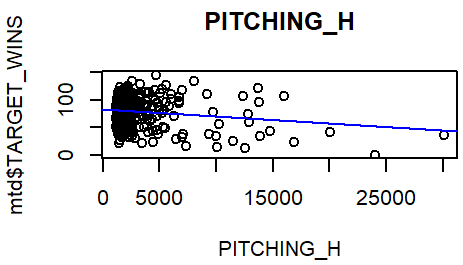

BATTING_HBP
Frequency
0
4
8
30
40
50
60
70
80
90
BATTING_HBP

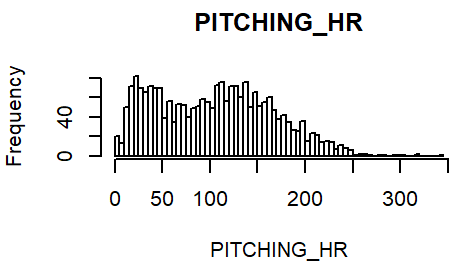

BATTING_HBP
30
40
50
60
70
80
90

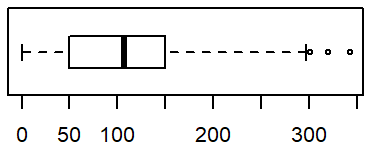

BATTING_HBP
mtd$TARGET_WINS
0
100
30
40
50
60
70
80
90
BATTING_HBP

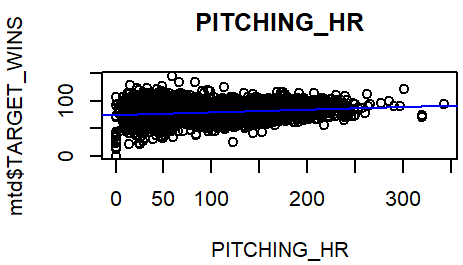

PITCHING_H
Frequency
0
600
0
5000
15000
25000
PITCHING_H

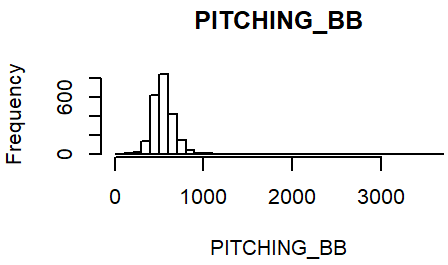

PITCHING_H
0
5000
15000
25000

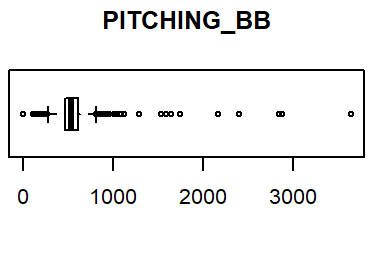

PITCHING_H
mtd$TARGET_WINS
0
100
0
5000
15000
25000
PITCHING_H

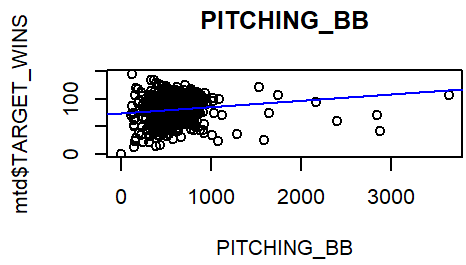

PITCHING_HR
Frequency
0
40
0
50
100
200
300
PITCHING_HR

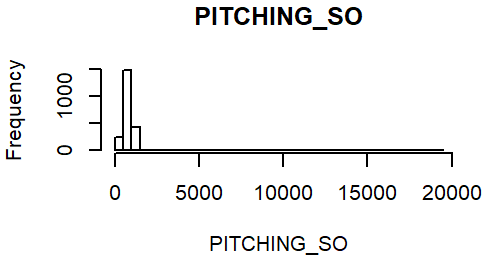

PITCHING_HR
0
50
100
200
300

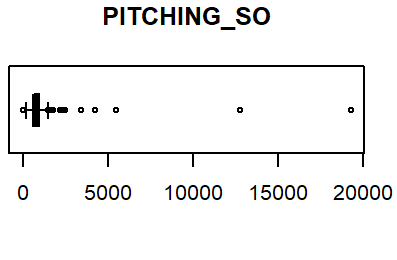

PITCHING_HR
mtd$TARGET_WINS
0
100
0
50
100
200
300
PITCHING_HR

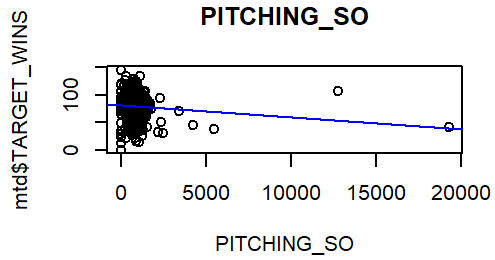

PITCHING_BB
Frequency
0
600
0
1000
2000
3000
PITCHING_BB

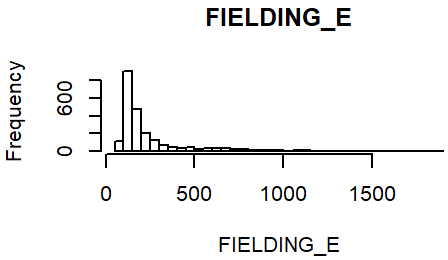

PITCHING_BB
0
1000
2000
3000

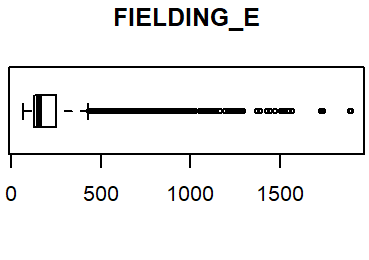

PITCHING_BB
mtd$TARGET_WINS
0
100
0
1000
2000
3000
PITCHING_BB

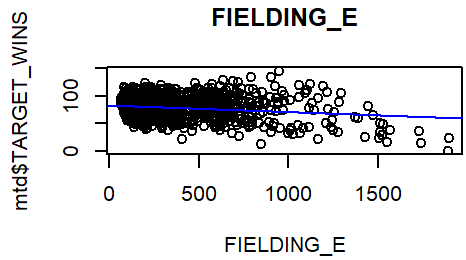

PITCHING_SO
Frequency
0
1000
0
5000
10000
15000
20000
PITCHING_SO

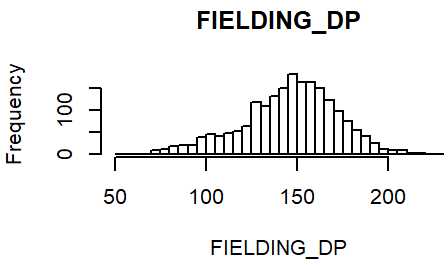

PITCHING_SO
0
5000
10000
15000
20000

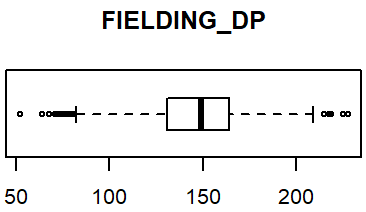

PITCHING_SO
mtd$TARGET_WINS
0
100
0
5000
10000
15000
20000
PITCHING_SO

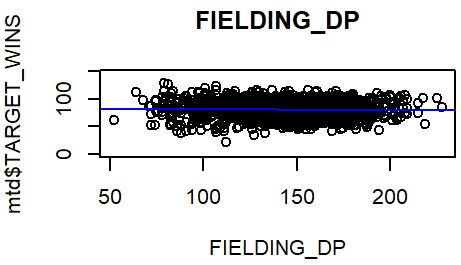

FIELDING_E
Frequency
0
600
0
500
1000
1500
FIELDING_E


FIELDING_E
0
500
1000
1500


FIELDING_E
mtd$TARGET_WINS
0
100
0
500
1000
1500
FIELDING_E







As can be seen from above histogram, boxplot and scatter plot with regression line shows the spread of the data points. More than half of the variables show skewness. A box-cox transformation may help to mitigate the skewness. Plot also shows very few variables are normally distributed.

3.7 Missing or NA Values

We are trying to see how many `NA` is present in the dataset.

variable	n	percent
BATTING_HBP	2085	92%
BASERUN_CS	772	34%
FIELDING_DP	286	13%
BASERUN_SB	131	5.8%
BATTING_SO	102	4.5%
PITCHING_SO	102	4.5%

The variable `BATTING_HBP` (hit by pitcher) is missing over 90% of it's data.

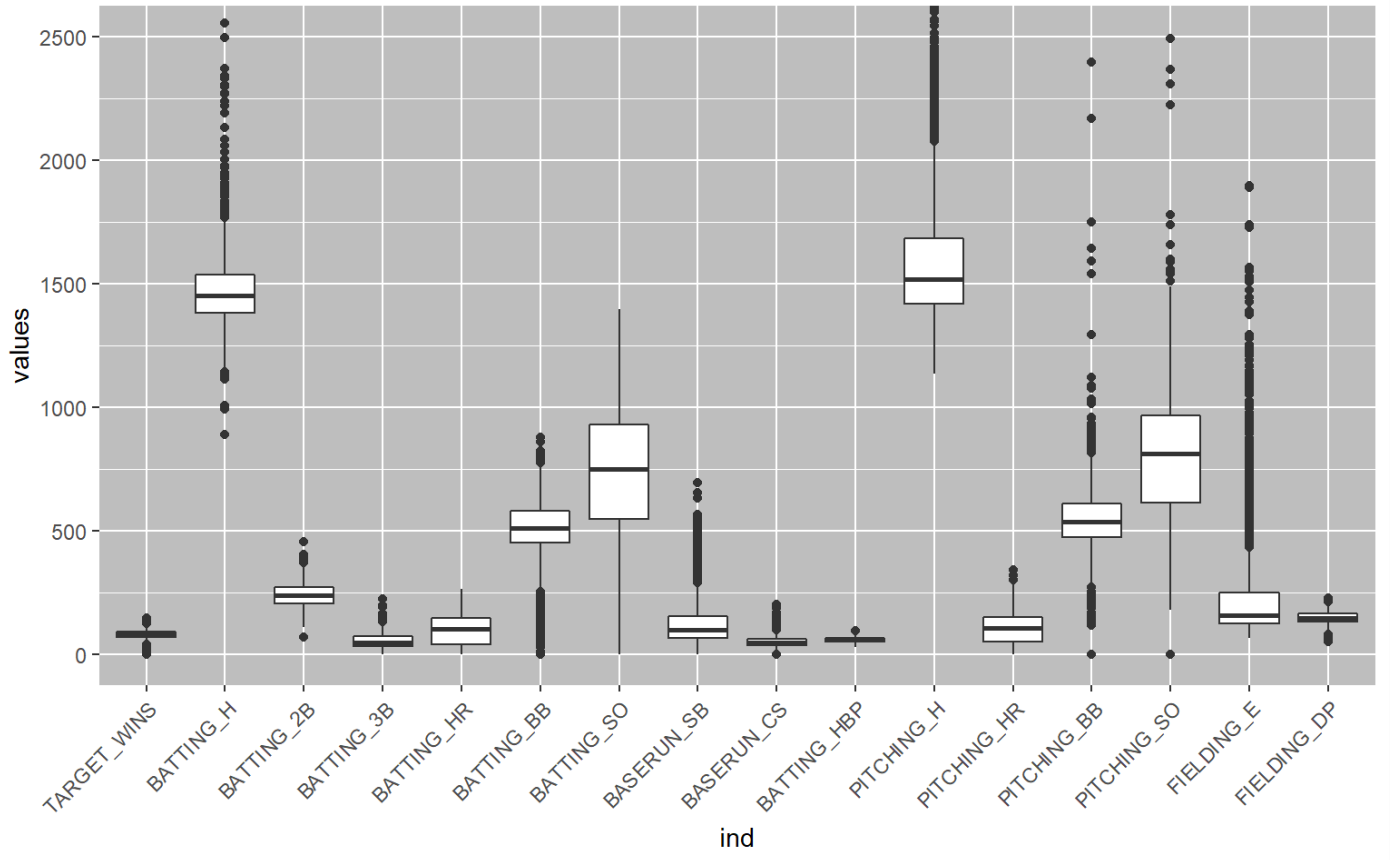
3.8 Zero Values

Code

variable	n	percent
BATTING_SO	20	0.9%
PITCHING_SO	20	0.9%
BATTING_HR	15	0.7%
PITCHING_HR	15	0.7%
BASERUN_SB	2	0.1%
BATTING_3B	2	0.1%
BASERUN_CS	1	0%
BATTING_BB	1	0%
PITCHING_BB	1	0%
TARGET_WINS	1	0%

As can be inferred from above, there are very few zero values exists.

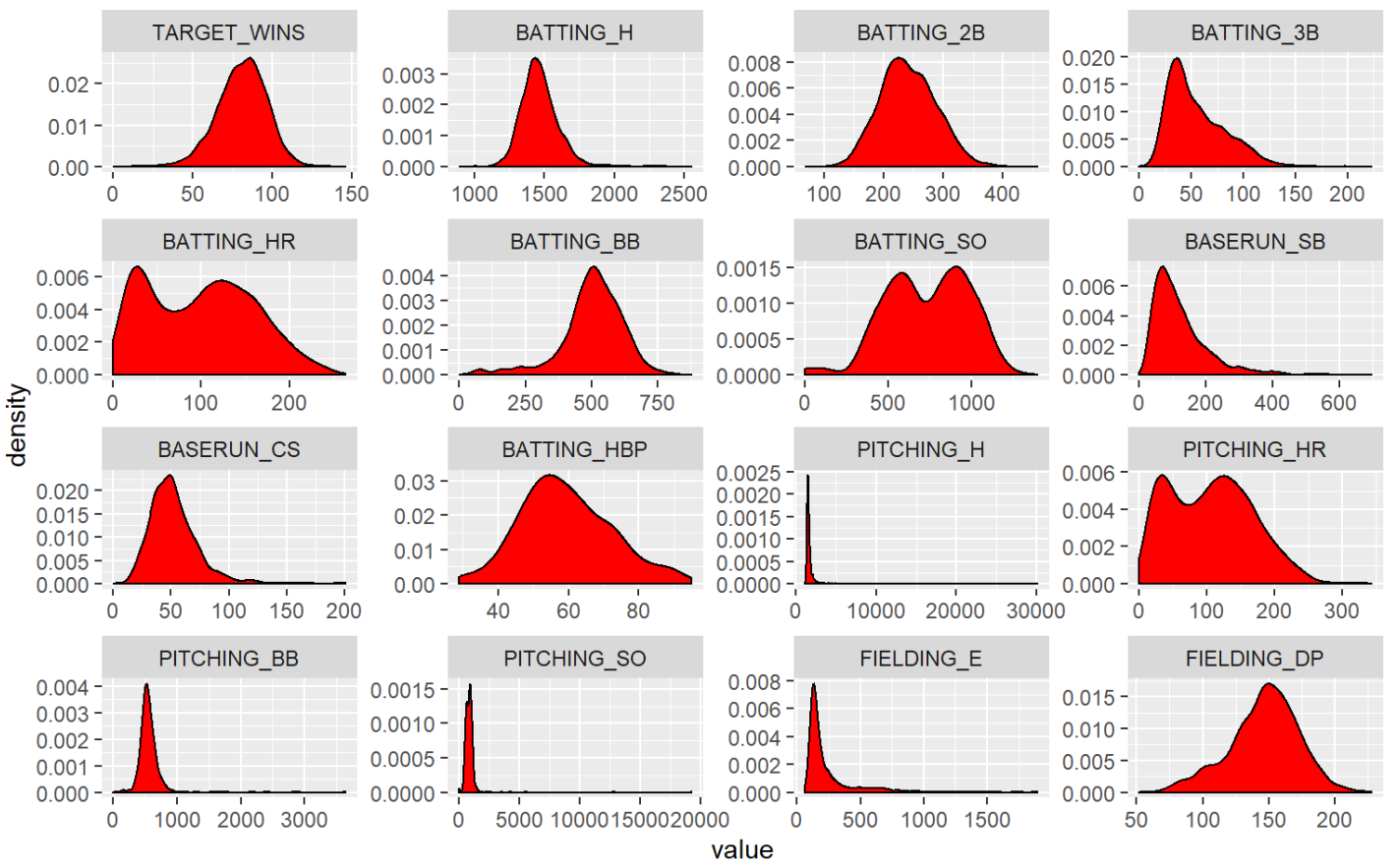
3.9 Checking for outliers



The box plots reveal that a great majority of the explanatory variables have high variances. Many of the medians and means are also not aligned which demonstrates the outliers' effects.

The variance of some of the explanatory variables greatly exceeds the variance of the response "win" variable. The dataset has many outlines with some observations that are more extreme than the 1.5 * IQR of the box plot whiskers.

3.10 Checking for skewness in the data



As per above, there are several variables like `PITCHING_H`, `PITCHING_BB`, `PITCHING_SO` and `FIELDING_E` are extremely skewed as there are many outliers.

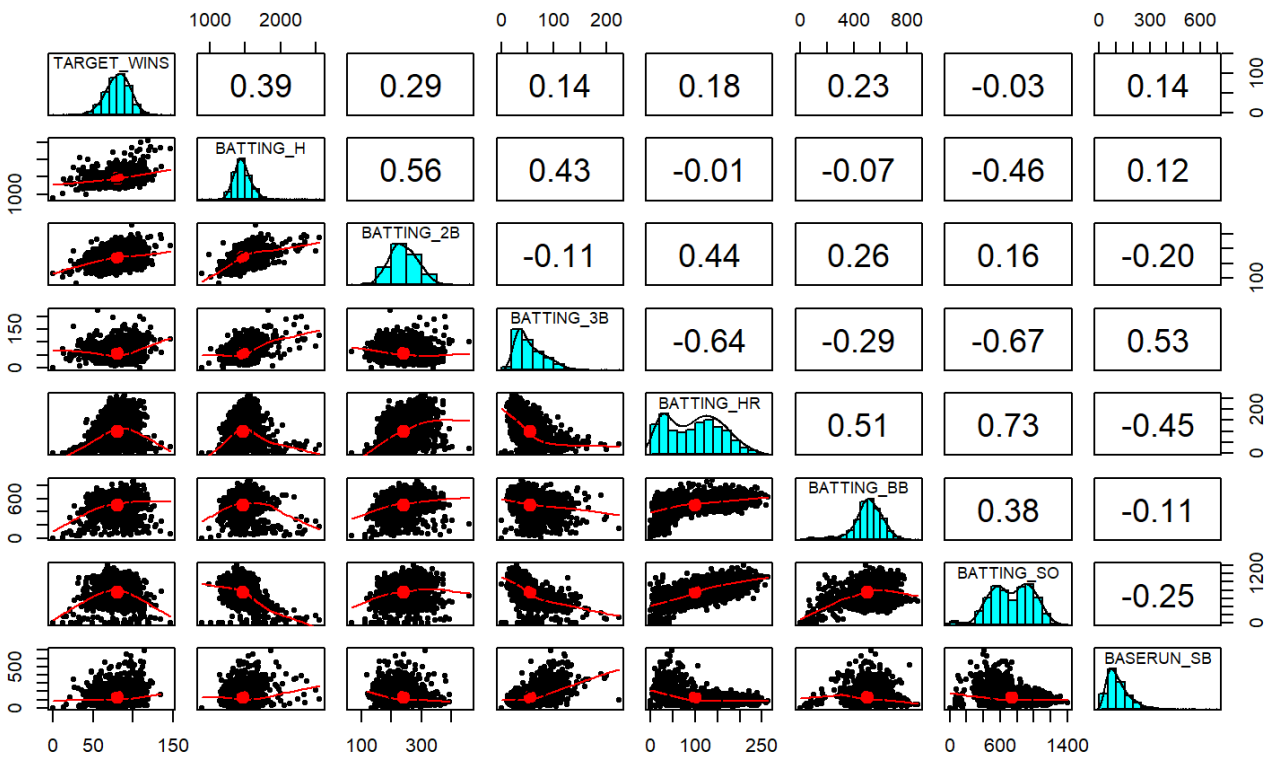
3.11 Finding correlations

Below shows the comparative correlations between the 16 variables as it shows the correlation coefficients and thus find correlated variables. Whichever adhere to a fitted straight red line well, ie. change in synch with each other. If the points lie close to the line but the line is curved, it's good nonlinear association and one can still be defined by other. Each individual plot shows the relationship between the variable in the horizontal vs the vertical of the grid. Each individual plot shows the relationship between the variable in the horizontal vs the vertical of the grid, whereas the diagonal is showing a histogram of each variable.

	TARGET_WINS	BATTING_H	BATTING_2B	BATTING_3B	BATTING_HR	BATTING_BB	BATTING_SO	BASERUN_SB	BASERUN_CS	BATTING_HBP	PITCHING_H	PITCHING_HR	PITCHING_BB	PITCHING_SO	FIELDING_E	FIELDING_DP
TARGET_WINS	1	0.46994665019557 2	0.31298399728000 4	-0.12434586296445 4	0.42241683411725 3	0.46868792650956	-0.22889272717982 3	0.01483639244265 2	-0.17875597924553 3	0.07350424230863 67	0.47123430638660 3	0.42246682099103 62	0.46839881792150 9	-0.22936480744398 5	-0.38668800441918	-0.19586600647086 6
BATTING_H	0.46994665019557 2	1	0.56177285553659 1	0.21391883444482 7	0.39627592732646 16	0.19735234388538 4	-0.34174328360081 8	0.07167495209622 44	0.09377544509123 3	-0.02911217568404 2	0.99919269234311	0.39495629870017 72	0.19529071054447 4	0.34445000520832 4	-0.25381637566043 5	0.01776945640927 98
BATTING_2B	0.31298399728000 4	0.56177285553659 1	1	0.04203440702680 67	0.25099045402781 7	0.19749256203079 4	-0.06415122582505 27	0.18768278795873 8	0.20413883707355 8	0.04608475314331 4	0.56045354847602	0.24999874755379 97	0.19592156551330 6	0.06616615375123 98	-0.19427027309153 1	-0.02488808148561 96
BATTING_3B	-0.12434586296445 4	0.21391883444482 7	0.04203440702680 67	1	-0.21879927259709 9	-0.20584392173080 7	0.19291840997956 7	0.16946086152565 7	0.23213977723850 5	-0.17424715383881 2	0.21250322022830 9	-0.21973263458534 48	0.20675382803574 4	0.19386653934694 5	-0.06513145051854 77	0.13314757845402 6
BATTING_HR	0.42241683411725 3	0.39627592732646 16	0.25099045402781 7	-0.21879927259709 9	1	0.45638161304111	0.21045443915641 7	-0.19021893151843 4	-0.27579837542521 2	0.10618116006506	0.39549389642229 2	0.99993258646418 2	0.45542467590360 5	0.20829573758333 7	0.01567397468584 87	-0.06182221869454 81
BATTING_BB	0.46868792650956	0.19735234388538 4	0.19749256203079 4	-0.20584392173080 7	0.45638161304111	1	0.21833871090989 8	-0.08806123372839 94	-0.20878059828045	0.04746000675647 86	0.19848686711087	0.45659282586473 83	0.99988139512683 3	0.21793252991785 5	-0.07847126148573 61	-0.07929077523897 07
BATTING_SO	-0.22889272717982 3	-0.34174328360081 8	-0.06415122582505 27	-0.19291840997956 7	0.21045443915641 17	0.21833871090989 8	1	-0.07475973606911 17	-0.05613035483306 89	0.22094219426166 5	-0.34145320559324 5	0.21111617381655	0.21895783249330 8	0.99976835262601 6	0.30814540314676	-0.12319071533322 4
BASERUN_SB	0.01483639244265 2	0.07167495209622 44	-0.18768278795873 8	0.16946086152565 7	-0.19021893151843 4	0.08806123372839 94	-0.07475973606911 17	1	0.62473780756125 1	-0.06400498161815 3	0.07395373115091 14	-0.18948057321796 8	-0.08741901908690 81	-0.07351324525453 42	0.04292340952797 43	-0.13023053722114 2
BASERUN_CS	-0.17875597924553 3	-0.09377544509123 3	-0.20413883707355 8	0.23213977723850 5	-0.27579837542521 2	-0.20878059828045	0.05613035483306 89	0.62473780756125 1	1	0.07051389575442 45	0.09297789288385 45	-0.27547149530431 11	-0.20847015356706 2	-0.05530833644396 19	0.20770118890490 9	-0.00676423304233 07
BATTING_HBP	0.07350424230863 67	-0.02911217568404 2	0.04608475314331 4	-0.17424715383881 2	0.10618116006506 6	0.04746000675647 86	0.22094219426166 5	-0.06400498161815 3	-0.07051389575442 45	1	-0.02769699489860 31	0.10675877976688 98	0.04785137104390 57	0.22157375412853 3	0.04178971227113 44	-0.07120824116291 95

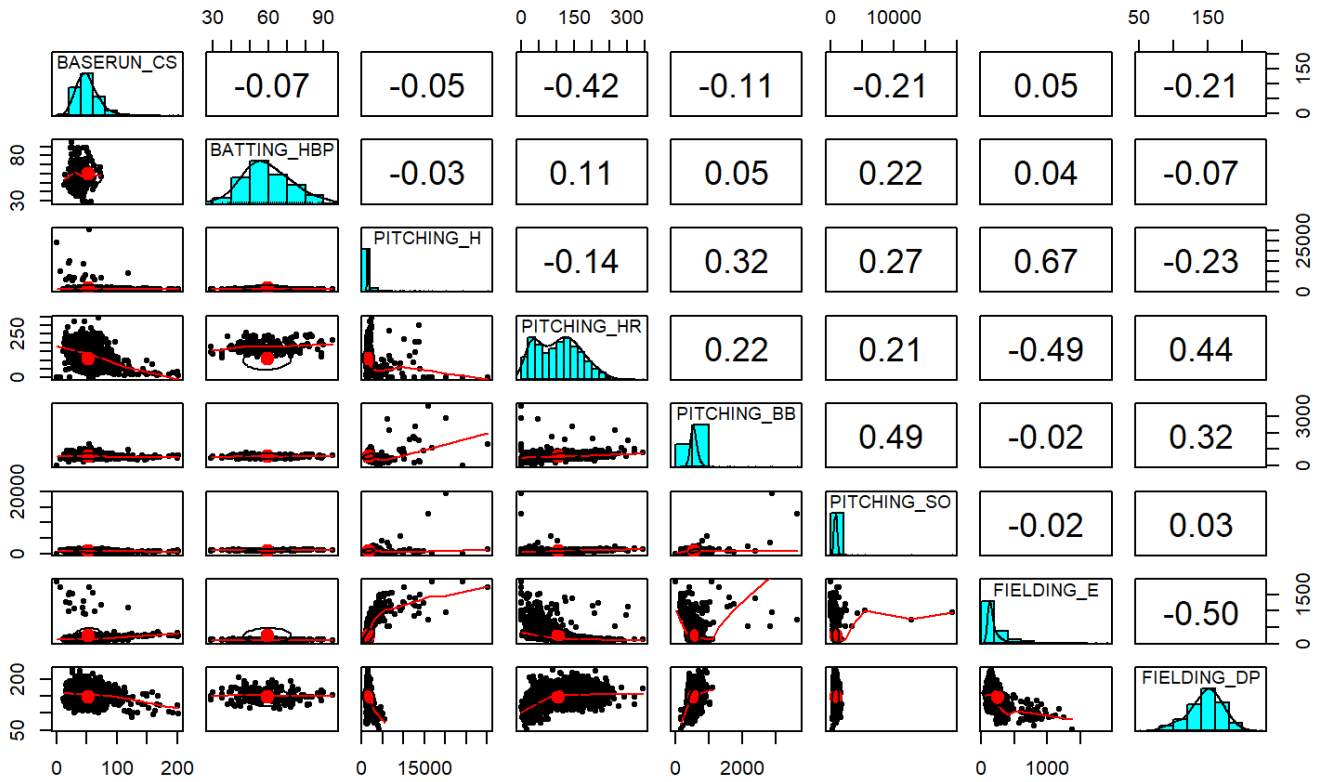
Showing 1 to 10 of 16 entries

Previous12Next



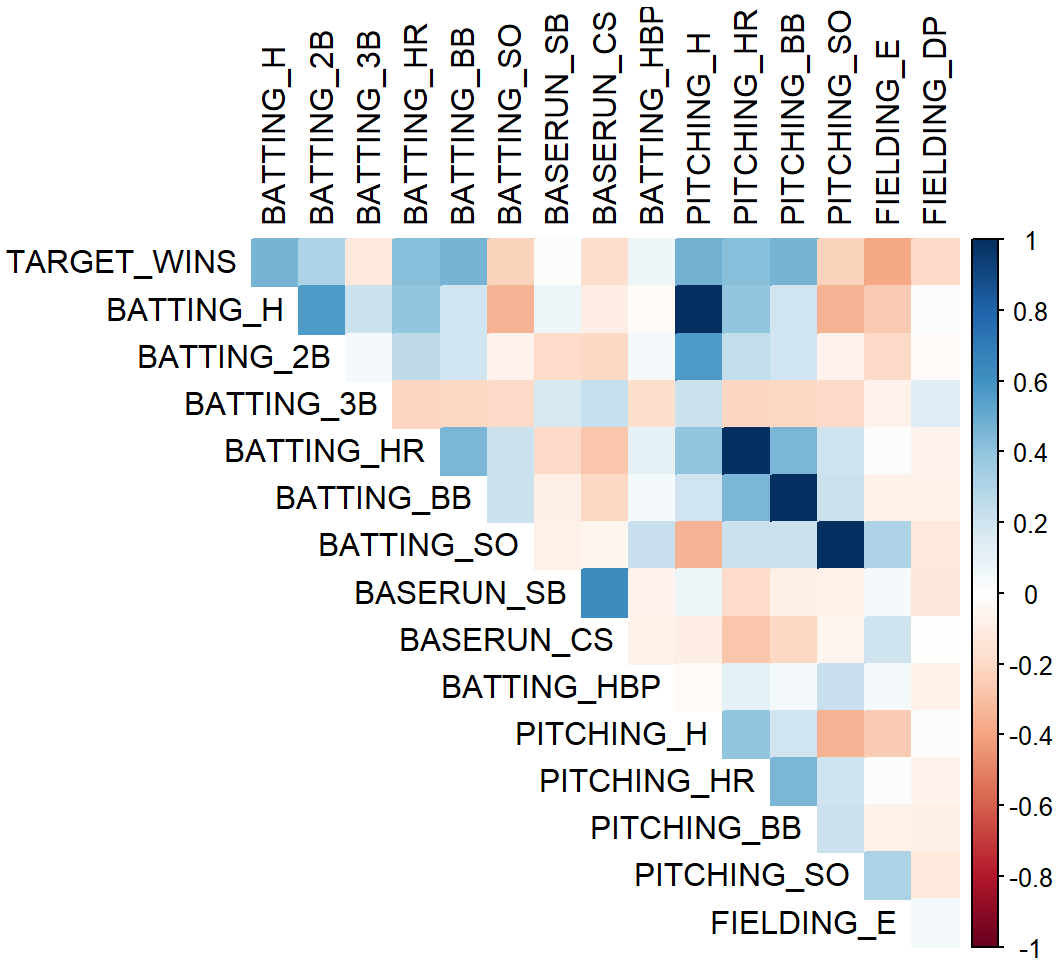
As can be seen from above, `TARGET_WINS` vs `BATTING_2B` is continuous and hence correlated and so is `BATTING_BB` and `BATTING_HR`.

Code



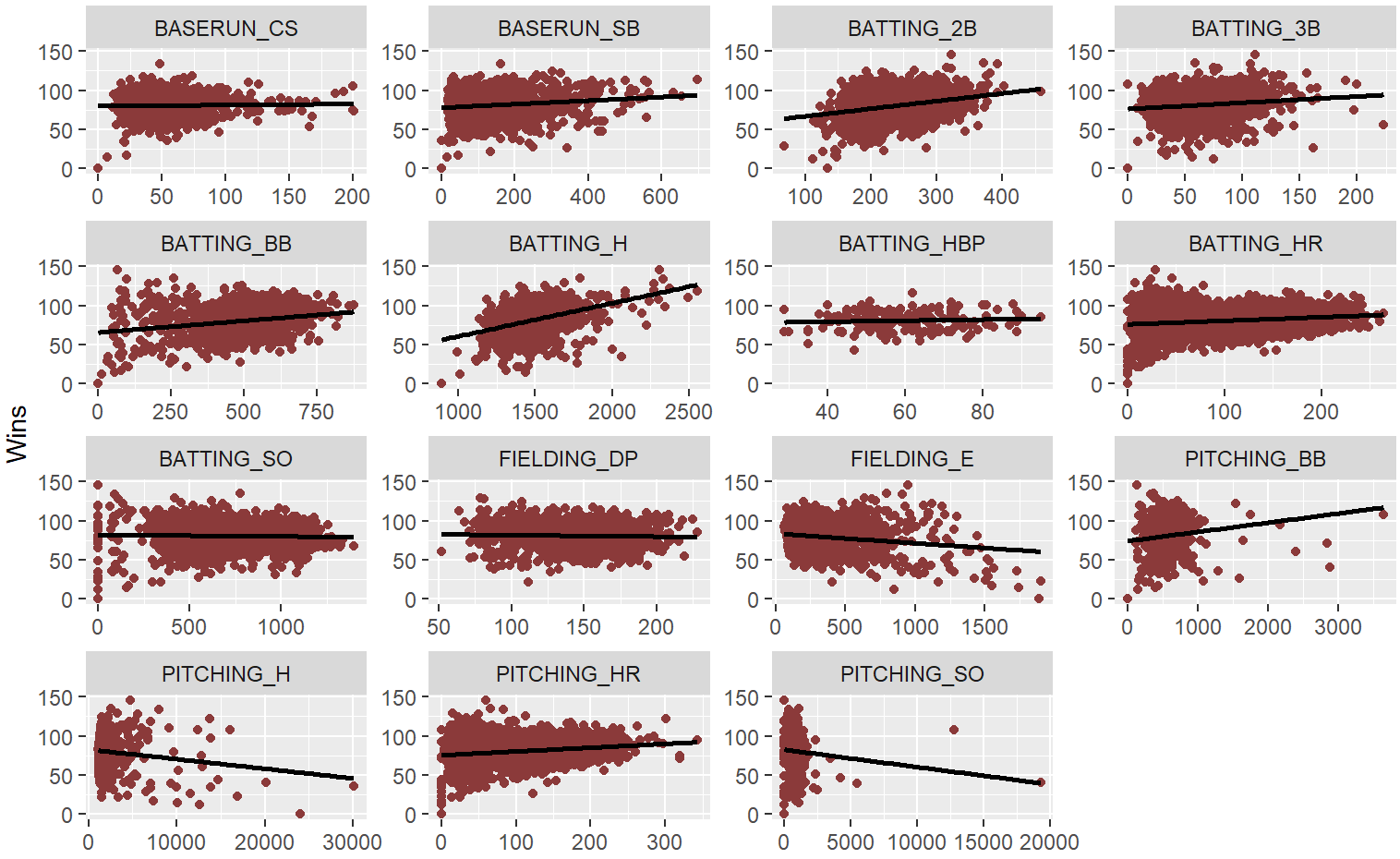
As can be seen from above, `BASERUN_CS` vs `BATTING_HBP` is continuous and hence correlated whereas `PITCHING_SO` and `FIELDING_E` is not correlated at all.

Code



Also, there are some negatively correlated variables. According to the correlation heatmap, the values that correspond most positively are BATTING_H, BATTING_2B, BATTING_HR, BATTING_BB, PITCHING_H, PITCHING_HR, and PITCHING_BB.

Code



Above shows how the data is distributed when compared to the linear regression.
Clearly, `PITCHING_H` and `PITCHING_SO` are highly heteroscedastic. Comparatively, `BATTING_HBP` is most homoscedastic.

Code

```
##              TARGET_WINS   BATTING_H
## TARGET_WINS   1.00000000  0.46994665
## BATTING_H     0.46994665  1.00000000
## BATTING_2B    0.31298400  0.56177286
## BATTING_3B   -0.12434586  0.21391883
## BATTING_HR    0.42241683  0.39627593
## BATTING_BB    0.46868793  0.19735234
## BATTING_SO   -0.22889273 -0.34174328
## BASERUN_SB    0.01483639  0.07167495
## BASERUN_CS   -0.17875598 -0.09377545
## BATTING_HBP   0.07350424 -0.02911218
## PITCHING_H    0.47123431  0.99919269
## PITCHING_HR   0.42246683  0.39495630
## PITCHING_BB   0.46839882  0.19529071
## PITCHING_SO -0.22936481 -0.34445001
```

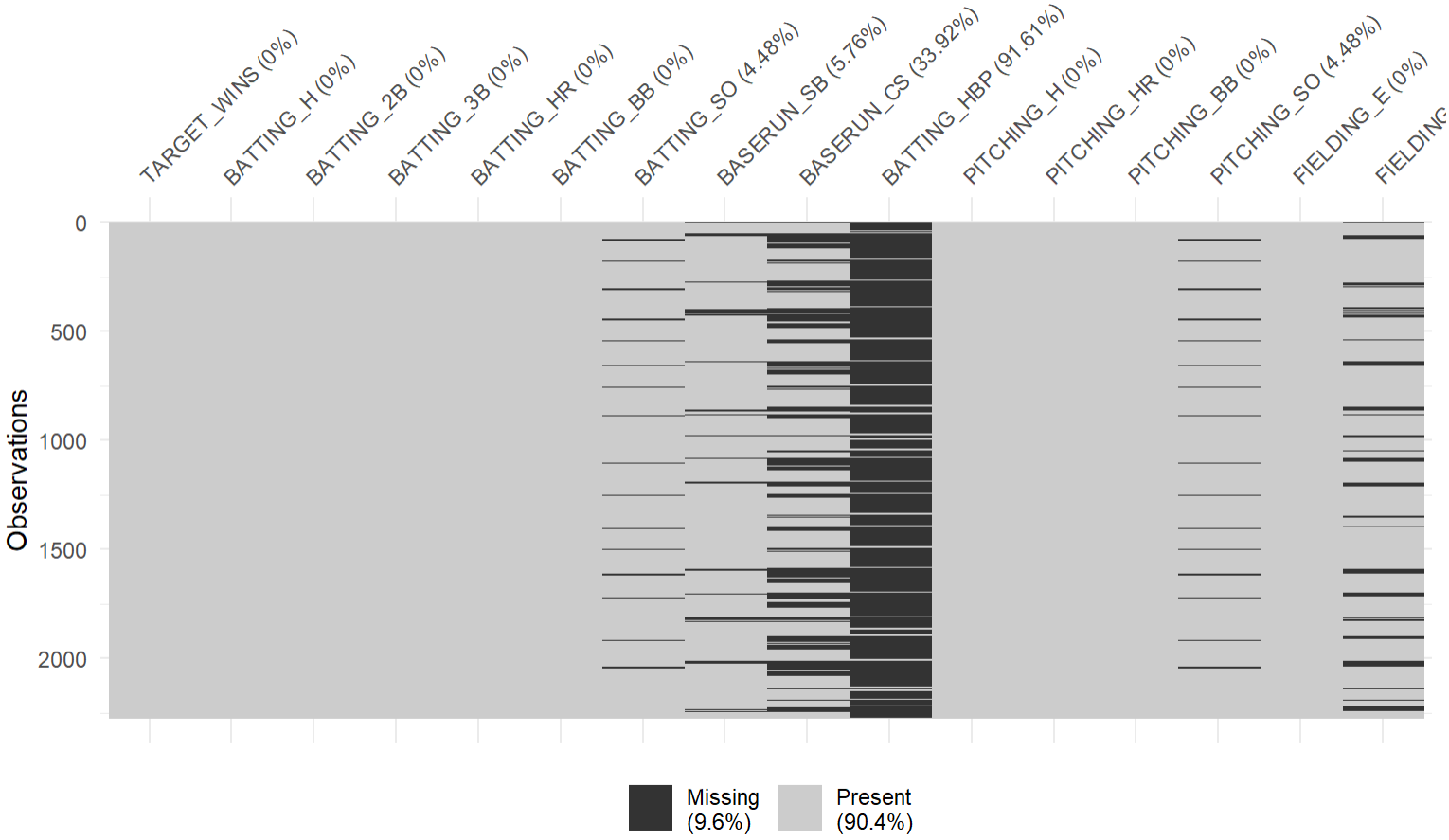
```
## FIELDING_E  -0.38668800 -0.25381638
## FIELDING_DP -0.19586601  0.01776946
```

Above shows the correlation coefficient of each variable compared to `TARGET_WINS` and `BATTING_H`.

3.12 Missing value by Graph

Here will see how much of data is missing in each predictor.

Code



Here from the plots we can see outliers in PITCHING_H,PITCHING_BB and PITCHING_SO

Also, since BATTING_H is a combination of BATTING_2B, BATTING_3B, BATTING_HR (and also includes batted singles), we will create a new variable BATTING_1B equaling BATTING_H - BATTING_2B - BATTING_3B - BATTING_HR and after creating this we will remove BATTING_H

3.13 Initial Observations

- *Response variable (TARGET_WINS) looks to be normally distributed which means there are good teams, bad teams as well as average teams.*
- *There are also quite a few variables with missing values. We may need to deal with these in order to have the largest data set possible for modeling.*
- *A couple variables are bimodal (TEAM_BATTING_HR, TEAM_BATTING_SO, TEAM_PITCHING_HR). This may be a challenge as some of them are missing values and that may be a challenge in filling in missing values.*
- *Some variables are right skewed (TEAM_BASERUN_CS, TEAM_BASERUN_SB, etc.). This might support the good team theory. It may also introduce non-normally distributed residuals in the model. We shall see.*
- *Dataset covers a wide time period spanning across multiple "eras" of baseball.*

4 DATA PREPARATION

4.1 Fixing Missing/Zero Values

- Remove the invalid data and prepare it for imputation.
- We could "discard" the `TEAM_BATTING_HBP`,due to the `high percentage of missing data`; particularly, replacing it by "ZERO" should not be advisable since the minimum value recorded is 29 and replacing it with a median value would not be much helpful due to high percentage of missing values. *We decided not to consider this variable for our study*.
- A typical professional league baseball game has 9 innings (extra innings come to play in the event of a tie) in length, and in each inning one can only pitch 3 strikeouts. There have been a maximum of 27 potential strikeouts upto a maximum of by 162 games for each of the 30 teams in the American League (AL) and National League (NL), played over approximately six months in Major League Baseball (MLB) season. Therefore having more than 4374 strikeouts (9x3x162) is not possible. Incidentally, the maximum strikeouts in any baseball season has been 513 by Matt Kilroy in the year 1886 as part of Baltimore Orioles within American Association League.

Code

4.2 Imputing the values using KNN

K-Nearest Neighbors (KNN) : K Nearest Neighbors is an algorithm that is useful for matching a point with its closest k neighbors in a multi-dimensional space. Therefore, a point value can be approximated by the values of the points that are closest to it, based on other variables.

The KNN imputation algorithm helps in imputing missing data by finding the k closest neighbors to the observation with missing data and then imputing them based on the the non-missing values in the neighbors. Most common method used for KNN is weighted mean

Code

As can be observed from above KNN imputation (result table below), the models did not behave favorably resulting in high RMSE and low R squared which results in poor prediction due to generation of highly correlated data.

Model Name	RMSE	R^2
model1	13.1079	0.271328
model2	13.2033	0.26092
model3	13.1079	0.27133
model4	13.3301	0.24664
model5	13.2601	0.25328
model6	13.0805	0.27403

Since `BATTING_H` is a combination of `BATTING_2B`, `BATTING_3B`, `BATTING_HR` (and also includes batted singles), we will create a new variable `BATTING_1B` equaling `BATTING_H - BATTING_2B - BATTING_3B - BATTING_HR` and after creating this we will remove `BATTING_H`

Code

5 BUILD MODELS

Kitchen Sink Model : With all variables to determine the base model provided. This would allow to see which variables are significant in our dataset, and allows to make other models based on that.

Code

5.1 Model 1 (Kitchen Sink Model/Backward Elimination)

Predictor: All Variables Response : TARGET_WINS

Code

```
##
```

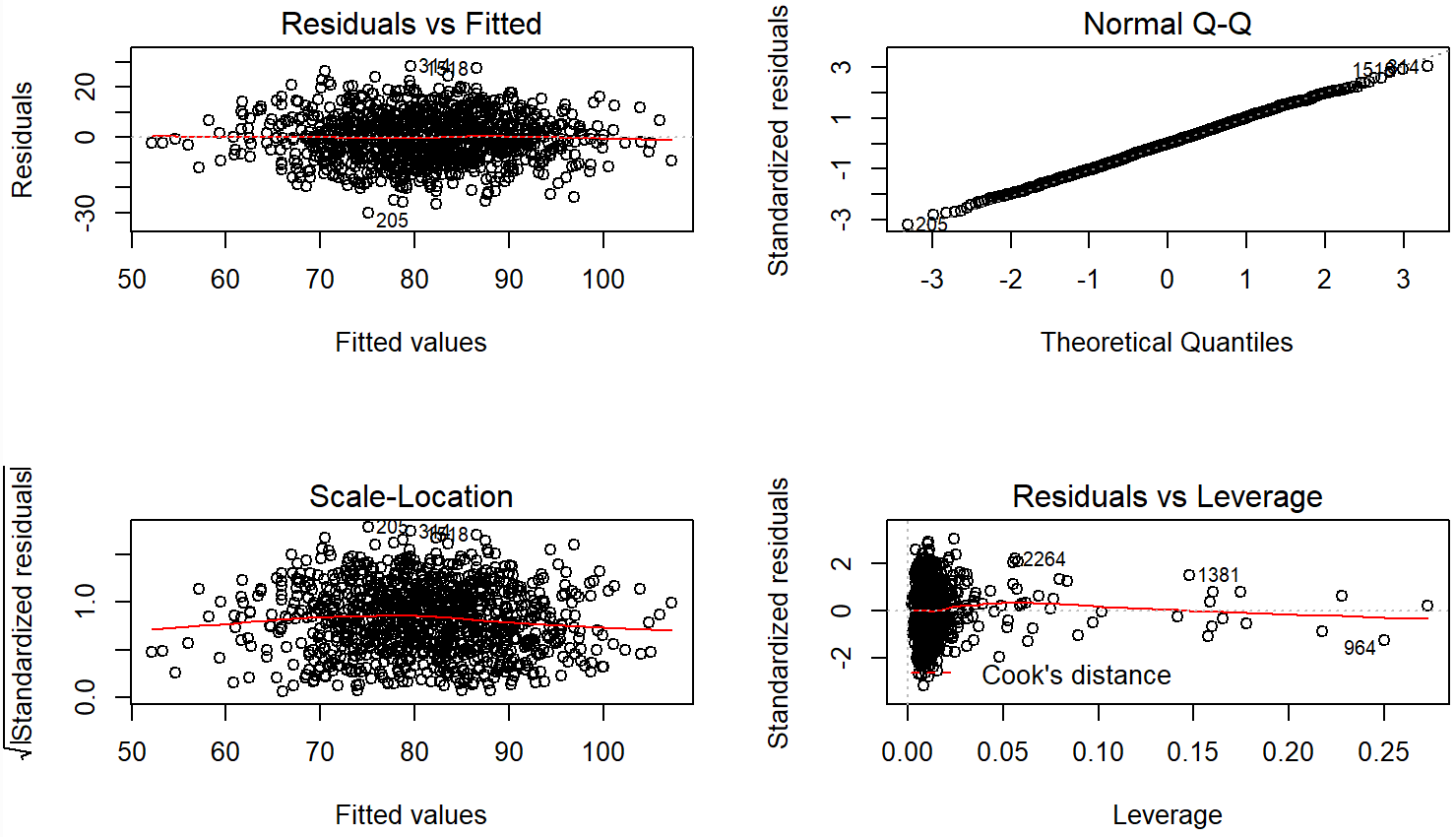
```
## Call:
## lm(formula = TARGET_WINS ~ ., data = moneyball_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.0724  -6.5828  -0.1407   6.4786  28.3847
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.53113    7.79100   7.513 1.25e-13 ***
## BATTING_H    0.01653    0.02346   0.704 0.481330
## BATTING_2B  -0.07540    0.01100  -6.854 1.23e-11 ***
## BATTING_3B   0.17325    0.02552   6.789 1.90e-11 ***
## BATTING_HR   0.13176    0.09460   1.393 0.163944
## BATTING_BB   0.02796    0.05440   0.514 0.607397
## BATTING_SO   0.01254    0.02769   0.453 0.650670
## BASERUN_SB   0.03694    0.01026   3.600 0.000334 ***
## BASERUN_CS   0.05115    0.02196   2.329 0.020032 *
## PITCHING_H   0.01747    0.02210   0.791 0.429325
## PITCHING_HR -0.02926    0.09070  -0.323 0.747075
## PITCHING_BB  0.01110    0.05237   0.212 0.832216
## PITCHING_SO -0.03241    0.02645  -1.225 0.220789
## FIELDING_E  -0.16207    0.01230 -13.176  < 2e-16 ***
## FIELDING_DP -0.10625    0.01545  -6.875 1.07e-11 ***
## BATTING_1B        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.469 on 1037 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.4421, Adjusted R-squared:  0.4346
## F-statistic:  58.7 on 14 and 1037 DF,  p-value: < 2.2e-16
```

Code

It does a fairly good job predicting, but there are a lot of variables that are not statistically significant. We see the that P-value is less than .05 which makes it one of the possible model but not all the coefficients of the `model1` are significant.

5.1.1 Plot Model1

Code



From the above residual plots let's analyze if the assumptions of our model is correct or not:

1. The variability of the points is approximately the same in the mid values of x with the decrese of variations towards the two end points which depicts that the plot is unbaised and homoscedastic except for few outliers.
2. Normal q-q plot fulfills the assumptions of normality.

But since few coefficients of the model are not significant, let's see if assumptions of other models are true.

5.2 Model 2 : Simple Model

With only the significant variables: Pick variables that had high correlations and include the pitching variables

`Predictor`: BATTING_H + BATTING_3B + BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + PITCHING_SO + PITCHING_H + PITCHING_SO + FIELDING_E + FIELDING_DP `Response` : TARGET_WINS

Code

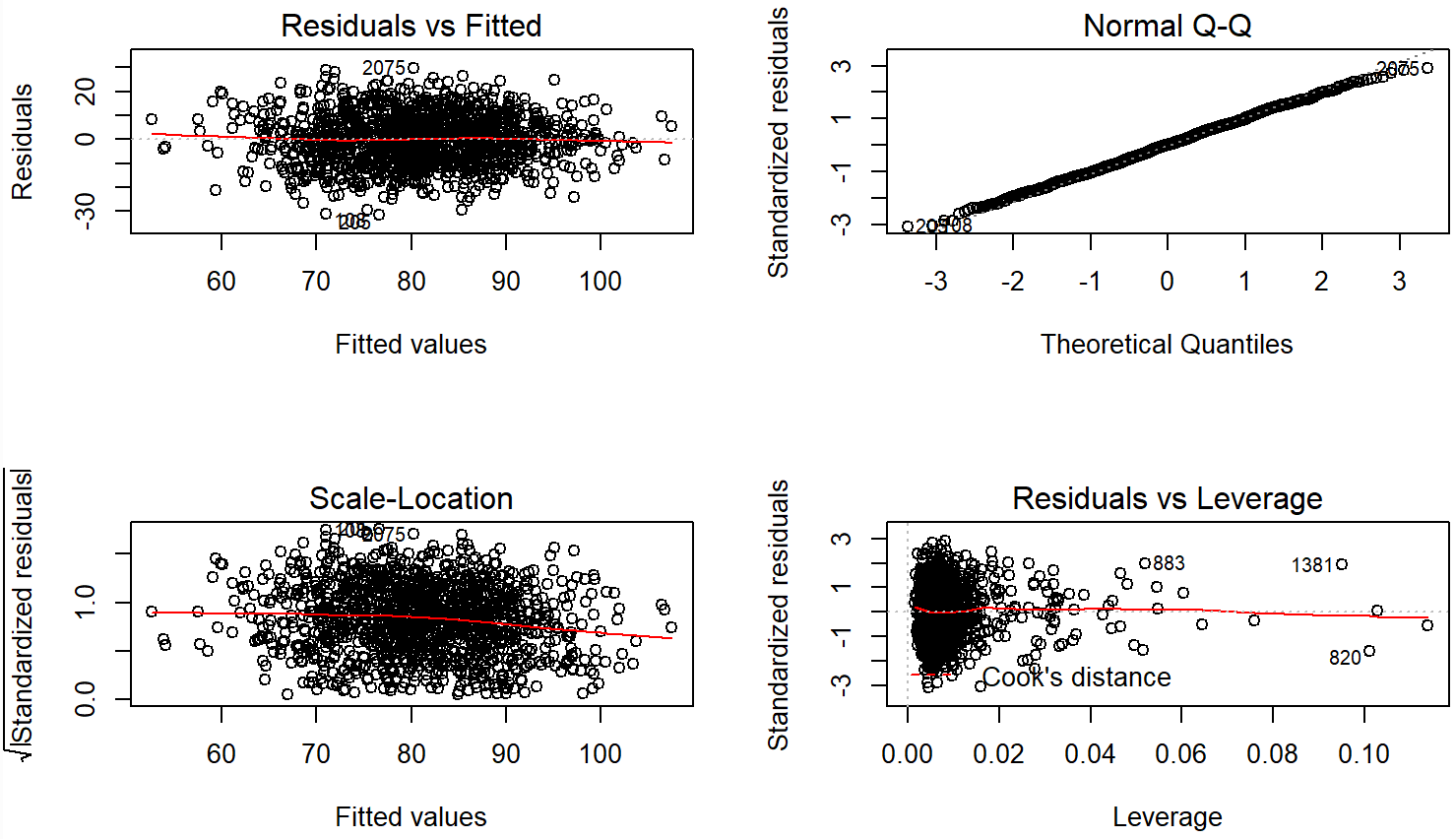
```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_H + BATTING_3B + BATTING_HR +
##     BATTING_BB + BATTING_SO + BASERUN_SB + PITCHING_SO + PITCHING_H +
##     PITCHING_SO + FIELDING_E + FIELDING_DP, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.633  -7.407   0.103   7.218  29.771
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.346701   6.624503  11.072  < 2e-16 ***
## BATTING_H   -0.036127   0.012857  -2.810 0.005032 **
## BATTING_3B   0.201222   0.022342   9.007  < 2e-16 ***
## BATTING_HR   0.114499   0.010869  10.535  < 2e-16 ***
## BATTING_BB   0.032347   0.003796   8.522  < 2e-16 ***
## BATTING_SO   0.048172   0.020693   2.328 0.020072 *
## BASERUN_SB   0.074635   0.006672  11.186  < 2e-16 ***
## PITCHING_SO -0.071270   0.019581  -3.640 0.000284 ***
## PITCHING_H   0.043819   0.011707   3.743 0.000190 ***
## FIELDING_E  -0.111738   0.008436 -13.245  < 2e-16 ***
## FIELDING_DP -0.105429   0.014630  -7.206 9.77e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.29 on 1286 degrees of freedom
##   (298 observations deleted due to missingness)
## Multiple R-squared:  0.3949, Adjusted R-squared:  0.3902
## F-statistic: 83.92 on 10 and 1286 DF,  p-value: < 2.2e-16
```

Code

For model 2, since we have only considered significant values from model 1, Multiple R-squared value is 0.39 which is a good representation that our model fits the data.

This model also does a good job predicting, and all variables are statistically significant.

5.2.1 Plot Model 2



From the above residual plots let's analyze if the assumptions of our model is correct or not:

1. The variability of the points is approximately the same throughout the values of x which depicts that this plot is also unbiased and homoscedastic with very few(minimum) outliers.
2. Normal q-q plot fulfills the assumptions of normality.

From the above points , assumptions of model 2 is true, let's see if assumptions of our next models are true.

5.3 Model 3 : Higher Order Stepwise Regression

Only taking the variable from the Model1 that are significant.

`Predictor`: BATTING_2B+BATTING_3B+BASERUN_SB+BASERUN_CS+FIELDING_E+FIELDING_DP
`Response` : TARGET_WINS

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_2B + BATTING_3B + BASERUN_SB +
##     BASERUN_CS + FIELDING_E + FIELDING_DP, data = moneyball_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.0056  -7.9628  -0.3434   8.0241  30.3356
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 93.226932   4.171175  22.350   <2e-16 ***
## BATTING_2B   0.019018   0.008810   2.159   0.0311 *
## BATTING_3B   0.273238   0.025450  10.736   <2e-16 ***
## BASERUN_SB   0.018523   0.011820   1.567   0.1174
## BASERUN_CS   0.007483   0.025892   0.289   0.7726
## FIELDING_E  -0.169187   0.013894 -12.177   <2e-16 ***
## FIELDING_DP -0.043599   0.018145  -2.403   0.0164 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.44 on 1045 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.1794, Adjusted R-squared:  0.1747
## F-statistic: 38.08 on 6 and 1045 DF,  p-value: < 2.2e-16
```

`Predictor`: BATTING_3B + FIELDING_E + BATTING_2B + FIELDING_DP

`Response` : TARGET_WINS

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_3B + FIELDING_E + BATTING_2B +
##     FIELDING_DP, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.154  -9.095   0.359   8.972  47.276
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.11824    3.17547  23.026  < 2e-16 ***
## BATTING_3B   0.15080    0.01793   8.411  < 2e-16 ***
## FIELDING_E  -0.02936    0.00371  -7.913 5.08e-15 ***
## BATTING_2B   0.06870    0.00816   8.418  < 2e-16 ***
## FIELDING_DP -0.07547    0.01579  -4.780 1.94e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.17 on 1396 degrees of freedom
```

```
##   (194 observations deleted due to missingness)
## Multiple R-squared:  0.1159, Adjusted R-squared:  0.1134
## F-statistic: 45.75 on 4 and 1396 DF,  p-value: < 2.2e-16
```

Code

As we see above, in Model3a in which “BATTING_3B,FIELDING_E ,BATTING_2B,FIELDING_DP” are significant and are considered in the model. We get Multiple R-squared as 0.17.

In Model3b we chose “BATTING_3B + FIELDING_E + BATTING_2B + FIELDING_DP” as they are significant coefficients in model3a. We get Multiple R-squared as 0.11

Further reducing the variables(TEAM_PITCHING_SO and TEAM_BATTING_SO are having high correlation, TEAM_BATTING_H and TEAM_PITCHING_H are also having high correlation, TEAM_BATTING_SO and TEAM_PITCHING_SO are also having high correlation):

`Predictor`: BATTING_1B + BATTING_2B + BATTING_3B + BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS + PITCHING_H + PITCHING_HR + PITCHING_BB + PITCHING_SO + FIELDING_E + FIELDING_DP + Quadratic
`Response` : TARGET_WINS

Code

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_1B + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     PITCHING_H + PITCHING_HR + PITCHING_BB + PITCHING_SO + FIELDING_E +
##     FIELDING_DP + +I(BATTING_1B^2) + I(BATTING_2B^2) + I(BATTING_3B^2) +
##     I(BATTING_HR^2) + I(BATTING_BB^2) + I(BATTING_SO^2) + +I(BASERUN_SB^2) +
##     I(BASERUN_CS^2) + +I(PITCHING_H^2) + I(PITCHING_HR^2) + I(PITCHING_BB^2) +
##     I(PITCHING_SO^2) + +I(FIELDING_E^2) + I(FIELDING_DP^2) +
##     +I(BATTING_2B^3) + I(BATTING_3B^3) + I(BATTING_HR^3) + I(BATTING_BB^3) +
##     I(BATTING_SO^3) + +I(BASERUN_SB^3) + I(BASERUN_CS^3) + +I(PITCHING_H^3) +
##     I(PITCHING_HR^3) + I(PITCHING_BB^3) + I(PITCHING_SO^3) +
##     +I(FIELDING_E^3) + I(FIELDING_DP^3) + +I(BATTING_1B^3) +
##     I(BATTING_2B^4) + I(BATTING_3B^4) + I(BATTING_HR^4) + I(BATTING_BB^4) +
##     I(BATTING_SO^4) + +I(BASERUN_SB^4) + I(BASERUN_CS^4) + +I(PITCHING_H^4) +
##     I(PITCHING_HR^4) + I(PITCHING_BB^4) + I(PITCHING_SO^4) +
##     +I(FIELDING_E^4) + I(FIELDING_DP^4) + I(BATTING_1B^4), data = moneyball_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.7539  -6.1490   0.0937   6.2226  25.7811
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.434e+02  2.086e+03   0.308   0.7578
```

```
## BATTING_1B       -4.275e-02  7.965e+00  -0.005   0.9957
## BATTING_2B        1.593e+00  3.124e+00   0.510   0.6103
## BATTING_3B        4.008e-02  5.314e-01   0.075   0.9399
## BATTING_HR        7.473e-02  3.574e+00   0.021   0.9833
## BATTING_BB       -3.481e+00  4.281e+00  -0.813   0.4163
## BATTING_SO        4.714e+00  2.646e+00   1.782   0.0751 .
## BASERUN_SB       -1.475e-01  1.446e-01  -1.020   0.3079
## BASERUN_CS        3.247e-01  2.786e-01   1.166   0.2441
## PITCHING_H       -1.155e+00  1.608e+00  -0.718   0.4729
## PITCHING_HR      -4.535e-01  3.363e+00  -0.135   0.8928
## PITCHING_BB       1.707e+00  4.053e+00   0.421   0.6737
## PITCHING_SO      -4.704e+00  2.483e+00  -1.895   0.0584 .
## FIELDING_E        9.206e-01  7.065e-01   1.303   0.1928
## FIELDING_DP       4.511e+00  5.613e+00   0.804   0.4218
## I(BATTING_1B^2)  -8.511e-04  1.115e-02  -0.076   0.9392
## I(BATTING_2B^2)  -9.045e-03  1.856e-02  -0.487   0.6261
## I(BATTING_3B^2)  -4.834e-03  1.438e-02  -0.336   0.7368
## I(BATTING_HR^2)  -2.382e-03  1.894e-02  -0.126   0.8999
## I(BATTING_BB^2)   3.664e-03  6.098e-03   0.601   0.5480
## I(BATTING_SO^2)  -3.707e-03  2.208e-03  -1.678   0.0936 .
## I(BASERUN_SB^2)   1.996e-03  1.811e-03   1.102   0.2707
## I(BASERUN_CS^2)  -6.253e-03  5.490e-03  -1.139   0.2550
## I(PITCHING_H^2)   1.182e-03  1.481e-03   0.799   0.4247
## I(PITCHING_HR^2)  5.229e-03  1.657e-02   0.316   0.7523
## I(PITCHING_BB^2)  8.306e-04  5.222e-03   0.159   0.8737
## I(PITCHING_SO^2)  3.715e-03  1.921e-03   1.933   0.0535 .
## I(FIELDING_E^2)  -9.853e-03  6.458e-03  -1.526   0.1274
## I(FIELDING_DP^2) -4.347e-02  5.435e-02  -0.800   0.4240
## I(BATTING_2B^3)   1.744e-05  4.819e-05   0.362   0.7176
## I(BATTING_3B^3)   7.869e-05  1.626e-04   0.484   0.6285
## I(BATTING_HR^3)   9.244e-06  5.800e-05   0.159   0.8734
## I(BATTING_BB^3)  -2.801e-06  5.302e-06  -0.528   0.5975
## I(BATTING_SO^3)   1.650e-06  1.107e-06   1.491   0.1362
## I(BASERUN_SB^3)  -8.207e-06  9.124e-06  -0.899   0.3686
## I(BASERUN_CS^3)   4.770e-05  4.287e-05   1.112   0.2662
## I(PITCHING_H^3)  -4.353e-07  5.801e-07  -0.750   0.4533
## I(PITCHING_HR^3) -1.968e-05  4.662e-05  -0.422   0.6730
## I(PITCHING_BB^3) -2.120e-06  3.931e-06  -0.539   0.5898
## I(PITCHING_SO^3) -1.679e-06  8.729e-07  -1.924   0.0547 .
## I(FIELDING_E^3)   3.783e-05  2.515e-05   1.504   0.1328
## I(FIELDING_DP^3)  1.749e-04  2.310e-04   0.757   0.4493
## I(BATTING_1B^3)   9.316e-07  6.912e-06   0.135   0.8928
## I(BATTING_2B^4)  -1.013e-08  4.626e-08  -0.219   0.8268
## I(BATTING_3B^4)  -3.432e-07  6.361e-07  -0.539   0.5897
## I(BATTING_HR^4)  -1.270e-08  7.269e-08  -0.175   0.8614
## I(BATTING_BB^4)   1.074e-09  1.949e-09   0.551   0.5819
## I(BATTING_SO^4)  -2.990e-10  2.352e-10  -1.271   0.2040
## I(BASERUN_SB^4)   1.164e-08  1.552e-08   0.750   0.4534
## I(BASERUN_CS^4)  -1.111e-07  1.118e-07  -0.994   0.3202
## I(PITCHING_H^4)   5.729e-11  8.340e-11   0.687   0.4923
## I(PITCHING_HR^4)  2.373e-08  5.223e-08   0.454   0.6497
## I(PITCHING_BB^4)  9.361e-10  1.207e-09   0.775   0.4383
## I(PITCHING_SO^4)  3.076e-10  1.630e-10   1.888   0.0594 .
## I(FIELDING_E^4)  -5.152e-08  3.514e-08  -1.466   0.1429
## I(FIELDING_DP^4) -2.518e-07  3.637e-07  -0.692   0.4888
```

```
## I(BATTING_1B^4)  -3.103e-10  1.599e-09  -0.194   0.8462
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.331 on 995 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.4802, Adjusted R-squared:  0.4509
## F-statistic: 16.41 on 56 and 995 DF,  p-value: < 2.2e-16
```

Code

5.4 StepBack Model

For StepBack Model, we have used MASS::stepAIC() function, which will generate the variables to create a system generated model. And as we see all the coefficients generated are significant.

Code

```
##
## Call:
## lm(formula = poly_call[2], data = moneyball_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.3740  -6.3034  -0.1952   6.2077  26.1001
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.188e+02  1.044e+02   3.055  0.00231 **
## BATTING_2B        8.256e-01  5.205e-01   1.586  0.11298
## BATTING_BB       -3.382e+00  1.266e+00  -2.672  0.00767 **
## BATTING_SO        4.697e+00  1.527e+00   3.076  0.00215 **
## PITCHING_HR      -3.189e-01  1.537e-01  -2.075  0.03820 *
## PITCHING_BB       2.486e+00  9.366e-01   2.654  0.00808 **
## PITCHING_SO      -4.661e+00  1.444e+00  -3.227  0.00129 **
## I(BATTING_1B^2)  -9.801e-04  3.806e-04  -2.575  0.01016 *
## I(BATTING_2B^2)  -4.315e-03  2.036e-03  -2.119  0.03431 *
## I(BATTING_BB^2)   2.121e-03  8.614e-04   2.463  0.01396 *
## I(BATTING_SO^2)  -3.678e-03  1.485e-03  -2.477  0.01341 *
## I(BASERUN_SB^2)   1.715e-04  3.723e-05   4.607 4.61e-06 ***
## I(PITCHING_H^2)   9.739e-05  2.992e-05   3.255  0.00117 **
## I(PITCHING_HR^2)  3.322e-03  1.587e-03   2.093  0.03658 *
## I(PITCHING_SO^2)  3.624e-03  1.293e-03   2.802  0.00517 **
## I(FIELDING_E^2)  -9.489e-04  1.638e-04  -5.794 9.14e-09 ***
## I(FIELDING_DP^2) -1.756e-03  5.248e-04  -3.346  0.00085 ***
## I(BATTING_2B^3)   6.004e-06  2.623e-06   2.289  0.02227 *
## I(BATTING_3B^3)   6.187e-06  3.269e-06   1.893  0.05867 .
## I(BATTING_BB^3)  -5.279e-07  2.966e-07  -1.780  0.07540 .
## I(BATTING_SO^3)   1.640e-06  8.379e-07   1.957  0.05061 .
## I(BASERUN_CS^3)   1.830e-06  7.946e-07   2.303  0.02145 *
## I(PITCHING_H^3)  -2.172e-08  7.806e-09  -2.782  0.00550 **
## I(PITCHING_HR^3) -1.344e-05  6.971e-06  -1.928  0.05409 .
## I(PITCHING_BB^3) -1.602e-06  6.497e-07  -2.465  0.01385 *
```

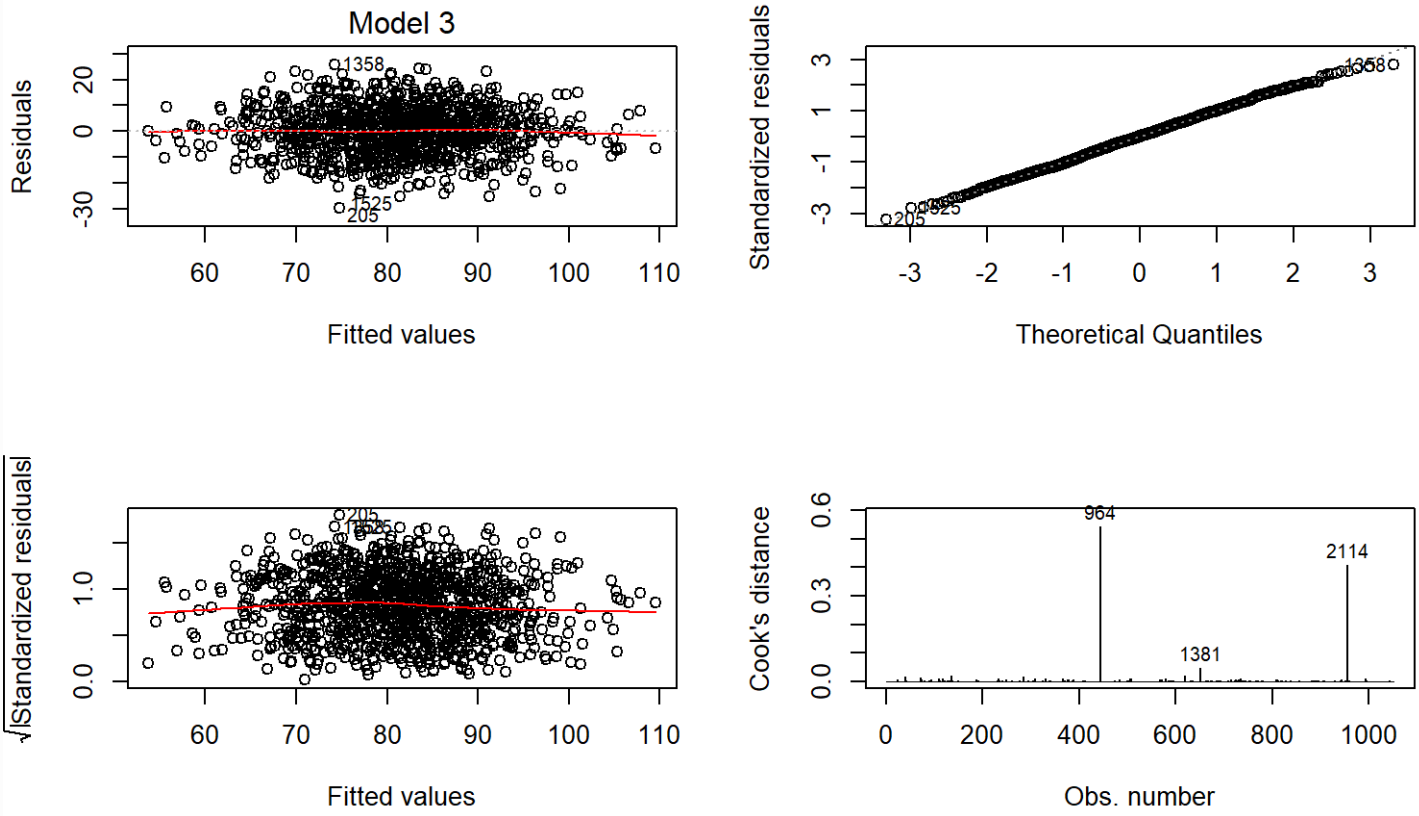
```
## I(PITCHING_SO^3) -1.612e-06  6.526e-07  -2.470  0.01369 *
## I(FIELDING_E^3)   1.803e-06  5.822e-07   3.096  0.00201 **
## I(FIELDING_DP^3)  6.100e-06  2.188e-06   2.788  0.00540 **
## I(BATTING_1B^3)   1.125e-06  4.713e-07   2.387  0.01716 *
## I(BATTING_SO^4)  -3.008e-10  1.936e-10  -1.553  0.12063
## I(PITCHING_HR^4)  1.703e-08  1.056e-08   1.612  0.10721
## I(PITCHING_BB^4)  8.033e-10  3.398e-10   2.364  0.01826 *
## I(PITCHING_SO^4)  2.908e-10  1.304e-10   2.230  0.02596 *
## I(BATTING_1B^4)  -3.810e-10  1.636e-10  -2.329  0.02003 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.267 on 1018 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.4755, Adjusted R-squared:  0.4585
## F-statistic: 27.96 on 33 and 1018 DF,  p-value: < 2.2e-16
```

For Model3, we take quadratic equation of predictors to analyze if multicollinearity exist . As we see, p-value is significant, so multicollinearity exist between the predictors.

5.4.1 Plot Model3, Model3a, Model3b

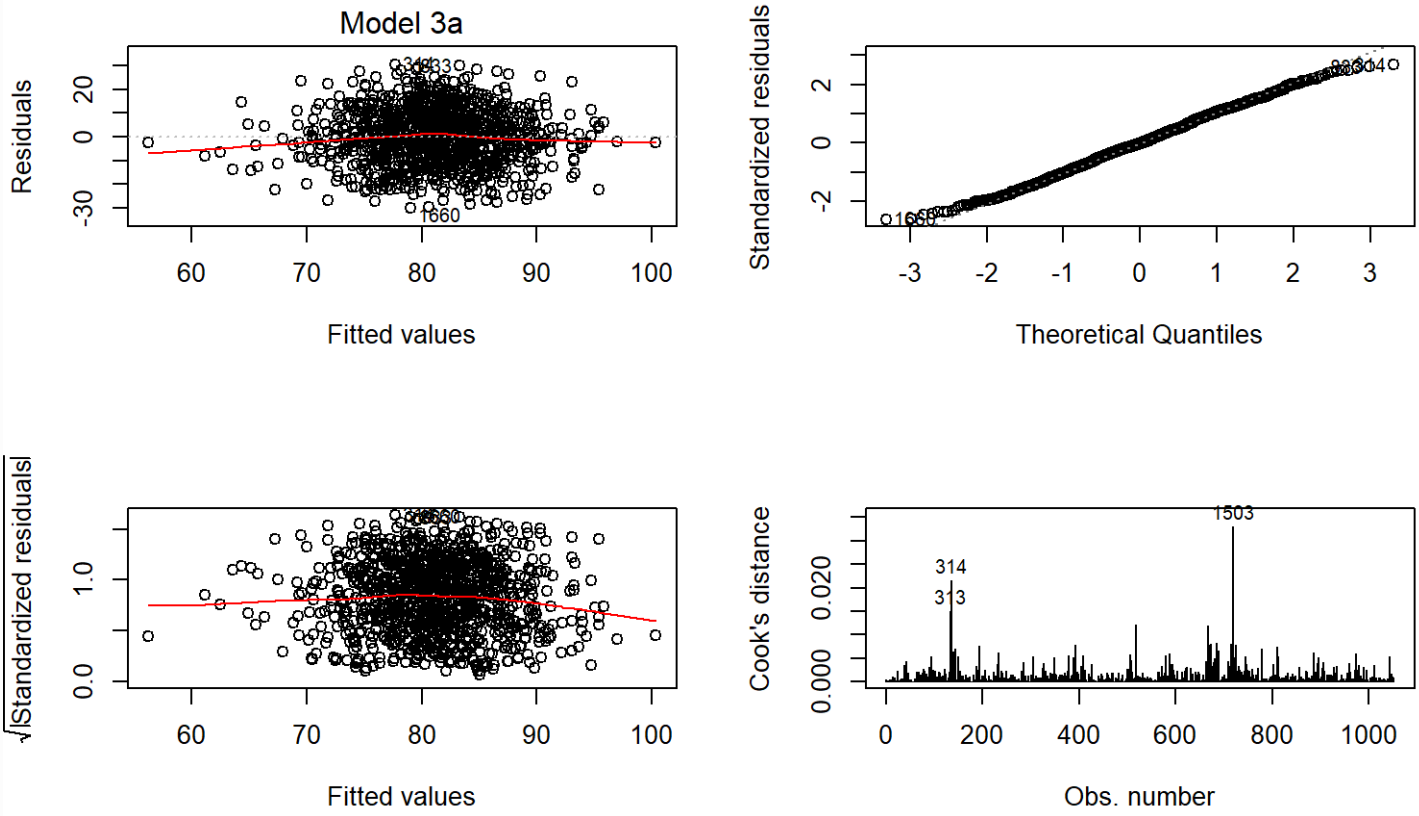
From the above residual plots let's analyze if the assumptions of our model is correct or not:

Model3:



1. The variability of the points is approximately the same throughout the values of x with some outliers towards for the ends and the variability also differs in two ends. We can say that the model is homoscedastic.
2. Normal q-q plot fulfills the assumptions of normality with the exception of outliers.

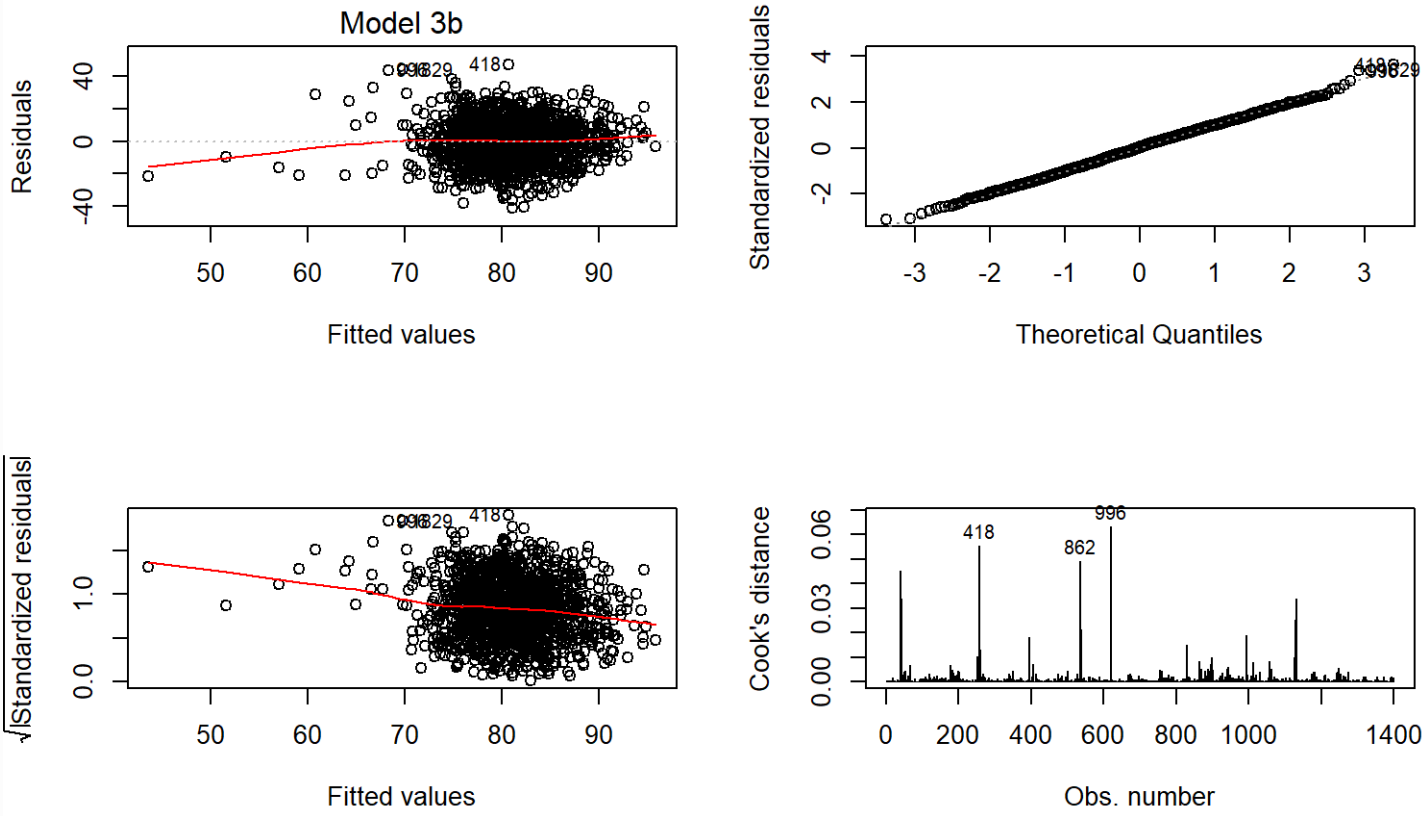
Model3a :



“BATTING_3B,FIELDING_E ,BATTING_2B,FIELDING_DP

1. Among the points scattered, variability of few points are not constant throughout.
2. Normal q-q plot fulfills the assumptions of normality with the exception of outliers.

Model3b :



For Model3b, we choose the significant variables from model3a,

1. The points scattered does not have a constant variability, which shows that the assumptions of this model does not hold true

6 SELECT MODELS

We have created couple of models in the last step, let's review the result for each of our model:

ModelName	Adjusted.R2	P.Value	AIC	Note
model1	0.4346	8.26675339500243e-121	7732.17046271654	BATTING_2B,BATTING_3B,BASERUN_SB,BASERUN_CS,FIELDING_E,FIELDING_DP
model2	0.3902	9.43169458989572e-133	9741.06557425804	All are significant
model3a	0.1747	6.064035000153e-42	8122.0744174421	BATTING_3B,FIELDING_E,BATTING_2B,FIELDING_DP are significant
model3b	0.1134	3.7241282367616e-36	11207.2018569633	All are significant
model3	0.4509	1.43731937269178e-105	7741.77841260617	Nothing is significant
step_back	0.4585	5.27149347920012e-119	7705.29597731889	more vars significant

Showing 1 to 7 of 7 entries

Previous1Next

6.0.1 Multicollinearity

Lets Evaluate if we have any multicollinearity in our model1s.Multicollinearity (also collinearity) is a statistical phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, meaning that one can be linearly predicted from the others with a non-trivial degree of accuracy.

We will user alias function to detect the collinearity of all the predictor in the model1.

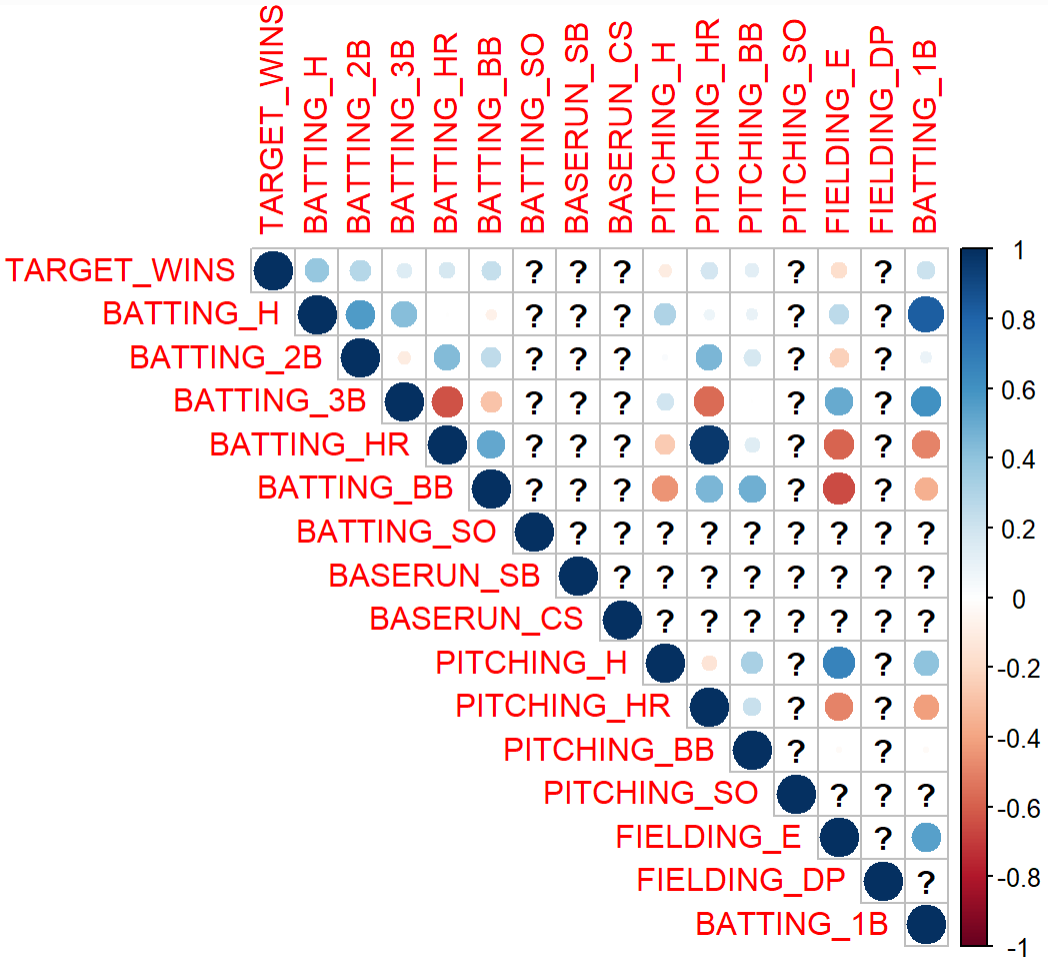
6.0.1.1 Model 1

Code

```
## Model :
## TARGET_WINS ~ BATTING_H + BATTING_2B + BATTING_3B + BATTING_HR +
##     BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS + PITCHING_H +
##     PITCHING_HR + PITCHING_BB + PITCHING_SO + FIELDING_E + FIELDING_DP +
##     BATTING_1B
##
## Complete :
##            (Intercept) BATTING_H BATTING_2B BATTING_3B BATTING_HR BATTING_BB
## BATTING_1B  0           1        -1         -1         -1          0
##            BATTING_SO BASERUN_SB BASERUN_CS PITCHING_H PITCHING_HR PITCHING_BB
## BATTING_1B  0          0          0          0          0           0
##            PITCHING_SO FIELDING_E FIELDING_DP
```

```
## BATTING_1B  0          0          0
```

Code



Code

Result shows that `BATTING_1B` is correlated with `BATTING_H`, `BATTING_2B`,`BATTING_3B`, `BATTING_HR`.
Here `+1` and `-1` are indicative of sign of coefficient of the respective predictor while stating the value for `BATTING_1B`.

Corrplot also suggest the same except, it doesn't show high correlation between `BATTING_H``BATTING_HR`.
In our Model2 , we will just follow the p-value significance test and build the model.

Code

RMSE	
<dbl>	<db
9.804207	0.42556

1 row

6.0.2 Model 2

Here `alias` doesn't suggest any correlated predictor. Now we can run VIF (variance inflation factor), which measures how much the variance of a regression coefficient is inflated due to multicollinearity in the model. The smallest possible value of VIF is one (absence of multicollinearity). Here we will look for VIF value, if that exceeds 5 or 10 indicates a problematic amount of collinearity. "Read More"['http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/']

Code

```
## Model :
## TARGET_WINS ~ BATTING_H + BATTING_3B + BATTING_HR + BATTING_BB +
##     BATTING_SO + BASERUN_SB + PITCHING_SO + PITCHING_H + PITCHING_SO +
##     FIELDING_E + FIELDING_DP
```

Code

```
##   BATTING_H  BATTING_3B  BATTING_HR  BATTING_BB  BATTING_SO  BASERUN_SB
##   23.591594    2.924829    4.274146    1.259010  242.802006    1.539592
## PITCHING_SO  PITCHING_H  FIELDING_E FIELDING_DP
##  225.307718   48.406757    2.835717    1.353810
```

VIF output suggest that BATTING_H, PITCHING_H, BATTING_SO,PITCHING_SO are highly impacting model due their colinear relation.

Code

RMSE <dbl>	R2 <dbl>
10.25912	0.3883479

1 row

6.0.2.1 Model 3

Code

RMSE <dbl>	R2 <dbl>
10.06308	0.4060436

1 row

6.0.2.2 Model 4

Code

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

```
## lm(formula = TARGET_WINS ~ . - BATTING_H - BATTING_2B - BATTING_3B -
##     BATTING_HR, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.334  -6.834  -0.136   6.517  29.480
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 59.857266   8.110353   7.380 3.23e-13 ***
## BATTING_BB   0.006719   0.039339   0.171 0.864410
## BATTING_SO   0.006949   0.022410   0.310 0.756561
## BASERUN_SB   0.035119   0.010675   3.290 0.001036 **
## BASERUN_CS   0.068018   0.022780   2.986 0.002894 **
## PITCHING_H  -0.002634   0.006751  -0.390 0.696514
## PITCHING_HR  0.116181   0.012748   9.113  < 2e-16 ***
## PITCHING_BB  0.030035   0.037698   0.797 0.425796
## PITCHING_SO -0.033549   0.021345  -1.572 0.116309
## FIELDING_E  -0.127737   0.012193 -10.476  < 2e-16 ***
## FIELDING_DP -0.104855   0.016090  -6.517 1.12e-10 ***
## BATTING_1B   0.038734   0.010312   3.756 0.000182 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.86 on 1040 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.3933, Adjusted R-squared:  0.3869
## F-statistic:  61.3 on 11 and 1040 DF,  p-value: < 2.2e-16
Code
##  BATTING_BB  BATTING_SO  BASERUN_SB  BASERUN_CS  PITCHING_H PITCHING_HR
##  107.539027  216.776484    2.415563    2.721623   14.163628    4.448142
## PITCHING_BB PITCHING_SO  FIELDING_E FIELDING_DP  BATTING_1B
##  144.662915  216.288753    2.187153    1.133447    7.973818
```

Code

RMSE	R2
<dbl>	<dbl>
9.922245	0.4109811

1 row

6.0.2.3 Model 5

Code

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - PITCHING_SO - PITCHING_BB - BATTING_H -
##     BATTING_2B - BATTING_3B - BATTING_HR, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.408  -6.629  -0.164   6.503  29.704
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.129049   8.109072   7.415 2.51e-13 ***
## BATTING_BB   0.038506   0.004083   9.430  < 2e-16 ***
## BATTING_SO  -0.027830   0.002911  -9.562  < 2e-16 ***
## BASERUN_SB   0.036013   0.010592   3.400   0.0007 ***
## BASERUN_CS   0.066311   0.022725   2.918   0.0036 **
## PITCHING_H  -0.010813   0.002702  -4.002 6.71e-05 ***
## PITCHING_HR  0.123928   0.010677  11.607  < 2e-16 ***
## FIELDING_E  -0.128182   0.012162 -10.540  < 2e-16 ***
## FIELDING_DP -0.105752   0.016091  -6.572 7.82e-11 ***
## BATTING_1B   0.049404   0.006386   7.737 2.40e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.87 on 1042 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.3909, Adjusted R-squared:  0.3857
## F-statistic: 74.32 on 9 and 1042 DF,  p-value: < 2.2e-16
Code
##  BATTING_BB  BATTING_SO  BASERUN_SB  BASERUN_CS  PITCHING_H PITCHING_HR
##    1.156266    3.649407    2.373748    2.703075    2.263550    3.113814
##  FIELDING_E FIELDING_DP  BATTING_1B
##    2.171454    1.131320    3.051488
```

Code

RMSE <dbl>	R2 <dbl>
9.991091	0.4029489

1 row

6.0.2.4 Model 6 (Step back)

VIF result suggest that all the predictors in the model `step_back` have no multicollinearity exist in them.

Code

```
##
## Call:
## lm(formula = poly_call[2], data = moneyball_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.3740  -6.3034  -0.1952   6.2077  26.1001
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.188e+02  1.044e+02   3.055  0.00231 **
## BATTING_2B        8.256e-01  5.205e-01   1.586  0.11298
## BATTING_BB       -3.382e+00  1.266e+00  -2.672  0.00767 **
## BATTING_SO        4.697e+00  1.527e+00   3.076  0.00215 **
## PITCHING_HR      -3.189e-01  1.537e-01  -2.075  0.03820 *
```

```
## PITCHING_BB       2.486e+00  9.366e-01   2.654  0.00808 **
## PITCHING_SO      -4.661e+00  1.444e+00  -3.227  0.00129 **
## I(BATTING_1B^2)  -9.801e-04  3.806e-04  -2.575  0.01016 *
## I(BATTING_2B^2)  -4.315e-03  2.036e-03  -2.119  0.03431 *
## I(BATTING_BB^2)   2.121e-03  8.614e-04   2.463  0.01396 *
## I(BATTING_SO^2)  -3.678e-03  1.485e-03  -2.477  0.01341 *
## I(BASERUN_SB^2)   1.715e-04  3.723e-05   4.607 4.61e-06 ***
## I(PITCHING_H^2)   9.739e-05  2.992e-05   3.255  0.00117 **
## I(PITCHING_HR^2)  3.322e-03  1.587e-03   2.093  0.03658 *
## I(PITCHING_SO^2)  3.624e-03  1.293e-03   2.802  0.00517 **
## I(FIELDING_E^2)  -9.489e-04  1.638e-04  -5.794 9.14e-09 ***
## I(FIELDING_DP^2) -1.756e-03  5.248e-04  -3.346  0.00085 ***
## I(BATTING_2B^3)   6.004e-06  2.623e-06   2.289  0.02227 *
## I(BATTING_3B^3)   6.187e-06  3.269e-06   1.893  0.05867 .
## I(BATTING_BB^3)  -5.279e-07  2.966e-07  -1.780  0.07540 .
## I(BATTING_SO^3)   1.640e-06  8.379e-07   1.957  0.05061 .
## I(BASERUN_CS^3)   1.830e-06  7.946e-07   2.303  0.02145 *
## I(PITCHING_H^3)  -2.172e-08  7.806e-09  -2.782  0.00550 **
## I(PITCHING_HR^3) -1.344e-05  6.971e-06  -1.928  0.05409 .
## I(PITCHING_BB^3) -1.602e-06  6.497e-07  -2.465  0.01385 *
## I(PITCHING_SO^3) -1.612e-06  6.526e-07  -2.470  0.01369 *
## I(FIELDING_E^3)   1.803e-06  5.822e-07   3.096  0.00201 **
## I(FIELDING_DP^3)  6.100e-06  2.188e-06   2.788  0.00540 **
## I(BATTING_1B^3)   1.125e-06  4.713e-07   2.387  0.01716 *
## I(BATTING_SO^4)  -3.008e-10  1.936e-10  -1.553  0.12063
## I(PITCHING_HR^4)  1.703e-08  1.056e-08   1.612  0.10721
## I(PITCHING_BB^4)  8.033e-10  3.398e-10   2.364  0.01826 *
## I(PITCHING_SO^4)  2.908e-10  1.304e-10   2.230  0.02596 *
## I(BATTING_1B^4)  -3.810e-10  1.636e-10  -2.329  0.02003 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.267 on 1018 degrees of freedom
##   (543 observations deleted due to missingness)
## Multiple R-squared:  0.4755, Adjusted R-squared:  0.4585
## F-statistic: 27.96 on 33 and 1018 DF,  p-value: < 2.2e-16
Code
##       BATTING_2B       BATTING_BB       BATTING_SO      PITCHING_HR
##     5.806061e+03     1.260278e+05     1.139708e+06     7.315647e+02
##      PITCHING_BB      PITCHING_SO  I(BATTING_1B^2)  I(BATTING_2B^2)
##     1.010821e+05     1.121060e+06     5.540838e+04     2.348083e+04
##  I(BATTING_BB^2)  I(BATTING_SO^2)  I(BASERUN_SB^2)  I(PITCHING_H^2)
##     7.326199e+04     2.792078e+06     1.809761e+00     3.504061e+03
## I(PITCHING_HR^2) I(PITCHING_SO^2)  I(FIELDING_E^2) I(FIELDING_DP^2)
##     6.304249e+03     2.781515e+06     4.920758e+01     1.318676e+02
##  I(BATTING_2B^3)  I(BATTING_3B^3)  I(BATTING_BB^3)  I(BATTING_SO^3)
##     6.147220e+03     6.670212e+00     6.549632e+03     1.487152e+06
##  I(BASERUN_CS^3)  I(PITCHING_H^3) I(PITCHING_HR^3) I(PITCHING_BB^3)
##     2.175917e+00     1.586654e+03     8.034833e+03     6.941504e+04
## I(PITCHING_SO^3)  I(FIELDING_E^3) I(FIELDING_DP^3)  I(BATTING_1B^3)
##     1.500893e+06     4.620075e+01     1.312350e+02     2.198039e+05
##  I(BATTING_SO^4) I(PITCHING_HR^4) I(PITCHING_BB^4) I(PITCHING_SO^4)
##     1.174857e+05     1.297475e+03     1.615989e+04     1.210049e+05
##  I(BATTING_1B^4)
```

```
##     5.522947e+04
```

Code

RMSE <dbl>	R2 <dbl>
9.770826	0.4287342

1 row

Lets only consider Model with better RMSE and R2 and check it with AIC test:

Model Name	RMSE	R^2
model1	9.80421	0.42556
model2	10.2591	0.38835
model3	10.0631	0.40604
model4	9.92225	0.41098
model5	9.99109	0.40295
Step Back	9.77083	0.428734

Lets run the AIC weight test to evaluate the best model out of few selected models :

Code

```
##           dAICc df weight
## step_back   0.0 35 1
## model4    106.9 13 <0.001
## model5    106.9 11 <0.001
```

In Both test `Model1` is doing well, but since its not a parsimonious model we decided to check among `model4` and `model5` and `step_back`. Which is a parsimonious model, with no multicollinearity among the predictors. We also note how multicollinearity in models were impacting its effect on overall performance of the model.

Selected Model = `step_back`

6.1 Predict of Eval data

Run the `step_backward` model on Eval data.

Predicted <dbl>	BATTING_H <int>	BATTING_2B <int>	BATTING_3B <int>	BATTING_HR <int>	BATTING_BB <int>	BATTING_SO <int>	BASERUN_SB <int>	BASERUN_CS <int>
61.94855	1209	170	33	83	447	1080	62	50
67.08517	1221	151	29	88	516	929	54	39
73.70651	1395	183	29	93	509	816	59	47
82.63449	1539	309	29	159	486	914	148	57
81.55696	1496	239	55	164	462	670	48	28
86.81114	1420	223	57	186	511	751	31	21
84.45098	1460	232	22	176	503	680	27	8
91.57616	1411	195	22	141	485	665	59	48
80.77111	1434	192	30	153	434	747	57	46
70.96997	1297	204	22	130	491	1008	84	55

1-10 of 170 rows | 1-9 of 16 columns

From the three models, model3 is a more parsimonious model. There is no significant difference in R2, Adjusted R2 and RMSE even when the treatment for multi-collinearity was done.

7 CONCLUSION

This report covers an attempt to build a model to predict number of wins of a baseball team in a season based on several offensive and defensive statistics. Resulting model explained about 36% of variability in the target variable and included most of the provided explanatory variables. Some potentially helpful variables were not included in the data set. For instance, number of At Bats can be used to calculate on-base percentage which may correlate strongly with winning percentage. The model can be revised with additional variables or further analysis.

	kitchen_sink_error	simple_error	step_back_error
	Min. :-28.3735	Min. :-27.2876	Min. :-32.00000
	1st Qu.: -6.9033	1st Qu.: -7.6292	1st Qu.: -7.00000
	Median : -0.1124	Median : 0.2432	Median : 0.00000
	Mean : -0.0408	Mean : -0.1372	Mean : -0.07143
	3rd Qu.: 6.4889	3rd Qu.: 6.5731	3rd Qu.: 7.00000
	Max. : 27.6495	Max. : 29.6379	Max. : 32.00000
	NA’s :247	NA’s :143	NA’s :247

Appendix:

https://github.com/Rajwantmishra/DATA621_CR4/blob/master/HW1/HomeWork1.Rmd

Thank you