Homework #1

Alice Ding

2023-09-20

Overview

In this homework assignment, we 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 and 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.

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

- `INDEX`: Identification Variable (do not use)
- `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

Using `moneyball-training-data.csv`, we will explore the data, prepare the data, build a few multiple regression models, and then choose the one that best fits in order to predict the number of wins.

Data Exploration

To start, we'll begin by getting an idea of what our data looks like.

	vars	n	mean	sd	median	trimmed	mad	min	max	range	ske
INDEX	1	2276	1268.46	736.35	1270.5	1268.57	952.57	1	2535	2534	0.0
TARGET_WINS	2	2276	80.79	15.75	82.0	81.31	14.83	0	146	146	-0.4
TEAM_BATTING_H	3	2276	1469.27	144.59	1454.0	1459.04	114.16	891	2554	1663	1.5
TEAM_BATTING_2B	4	2276	241.25	46.80	238.0	240.40	47.44	69	458	389	0.2
TEAM_BATTING_3B	5	2276	55.25	27.94	47.0	52.18	23.72	0	223	223	1.1
TEAM_BATTING_HR	6	2276	99.61	60.55	102.0	97.39	78.58	0	264	264	0.1
TEAM_BATTING_BB	7	2276	501.56	122.67	512.0	512.18	94.89	0	878	878	-1.0
TEAM_BATTING_SO	8	2174	735.61	248.53	750.0	742.31	284.66	0	1399	1399	-0.3
TEAM_BASERUN_SB	9	2145	124.76	87.79	101.0	110.81	60.79	0	697	697	1.9
TEAM_BASERUN_CS	10	1504	52.80	22.96	49.0	50.36	17.79	0	201	201	1.9
TEAM_BATTING_HBP	11	191	59.36	12.97	58.0	58.86	11.86	29	95	66	0.3
TEAM_PITCHING_H	12	2276	1779.21	1406.84	1518.0	1555.90	174.95	1137	30132	28995	10.3
TEAM_PITCHING_HR	13	2276	105.70	61.30	107.0	103.16	74.13	0	343	343	0.2
TEAM_PITCHING_BB	14	2276	553.01	166.36	536.5	542.62	98.59	0	3645	3645	6.7
TEAM_PITCHING_SO	15	2174	817.73	553.09	813.5	796.93	257.23	0	19278	19278	22.1
TEAM_FIELDING_E	16	2276	246.48	227.77	159.0	193.44	62.27	65	1898	1833	2.9
TEAM_FIELDING_DP	17	1990	146.39	26.23	149.0	147.58	23.72	52	228	176	-0.3

Overall Stats

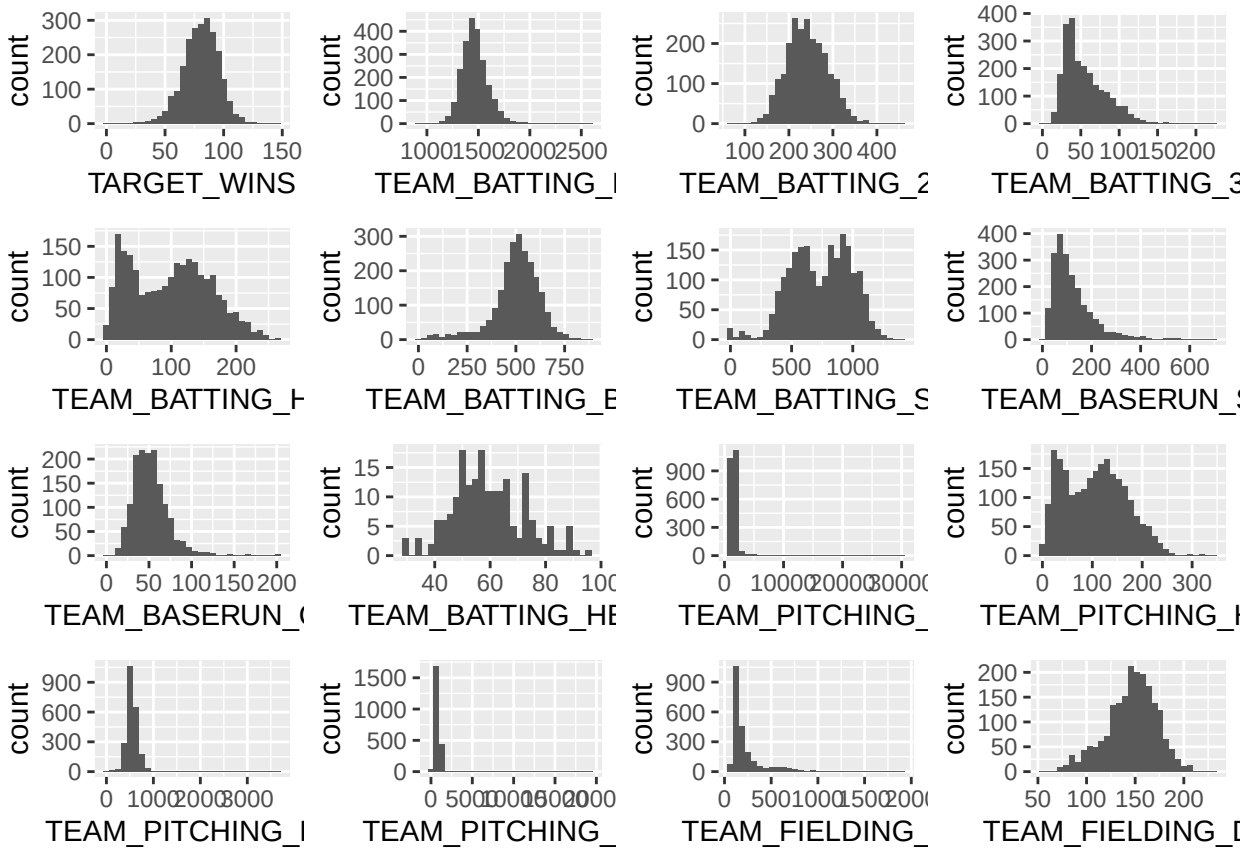
First, we'll view the summary and then we'll check if there are data points missing before cleaning the fields up to make sure they're ready for analysis.

One interesting thing to point out from the start is that the average wins for a team is ~81; there are 162 games in a season as given by the description of the dataset, so that means a team wins about half their games and loses the other. Some other interesting stats to bring to light are an average of ~100 home runs, ~736 strike outs, and ~502 walks by batters over the course of the season which would equal ~0.6 home runs, ~4.5 strike outs, and ~3 walks per game.

Inspecting for missing data, it looks like there's quite a few with NA's; we'll deal with those in the data preparation section.

Distributions

Let's see what all of these fields look like distribution wise.

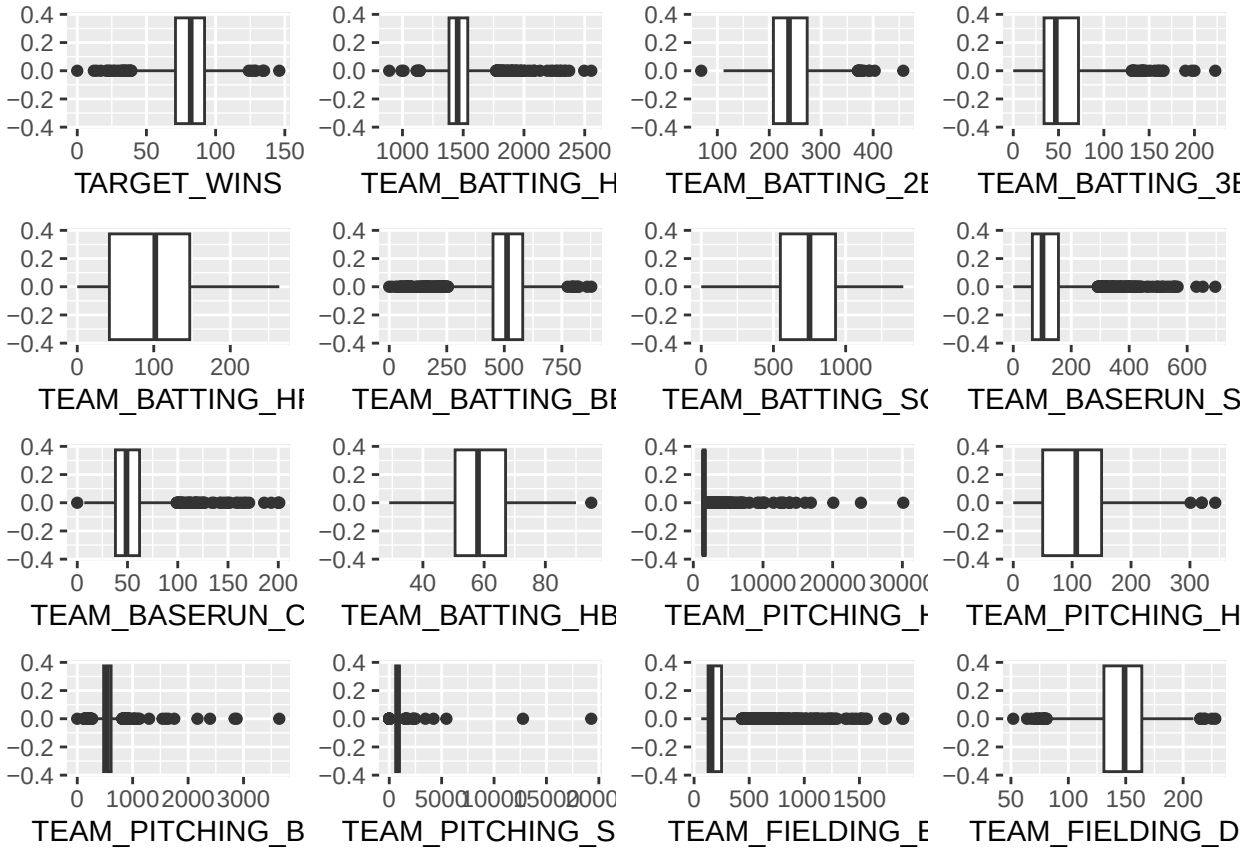


At first glance, it looks like these fields are relatively normal or have a good curve:

- `TARGET_WINS`
- `TEAM_BATTING_H`
- `TEAM_BATTING_2B`
- `TEAM_BATTING_BB`
- `TEAM_PITCHING_BB`
- `TEAM_FIELDING_DP`

The rest either are pretty skewed in either direction or have no pattern really at all.

How do these look as boxplots?

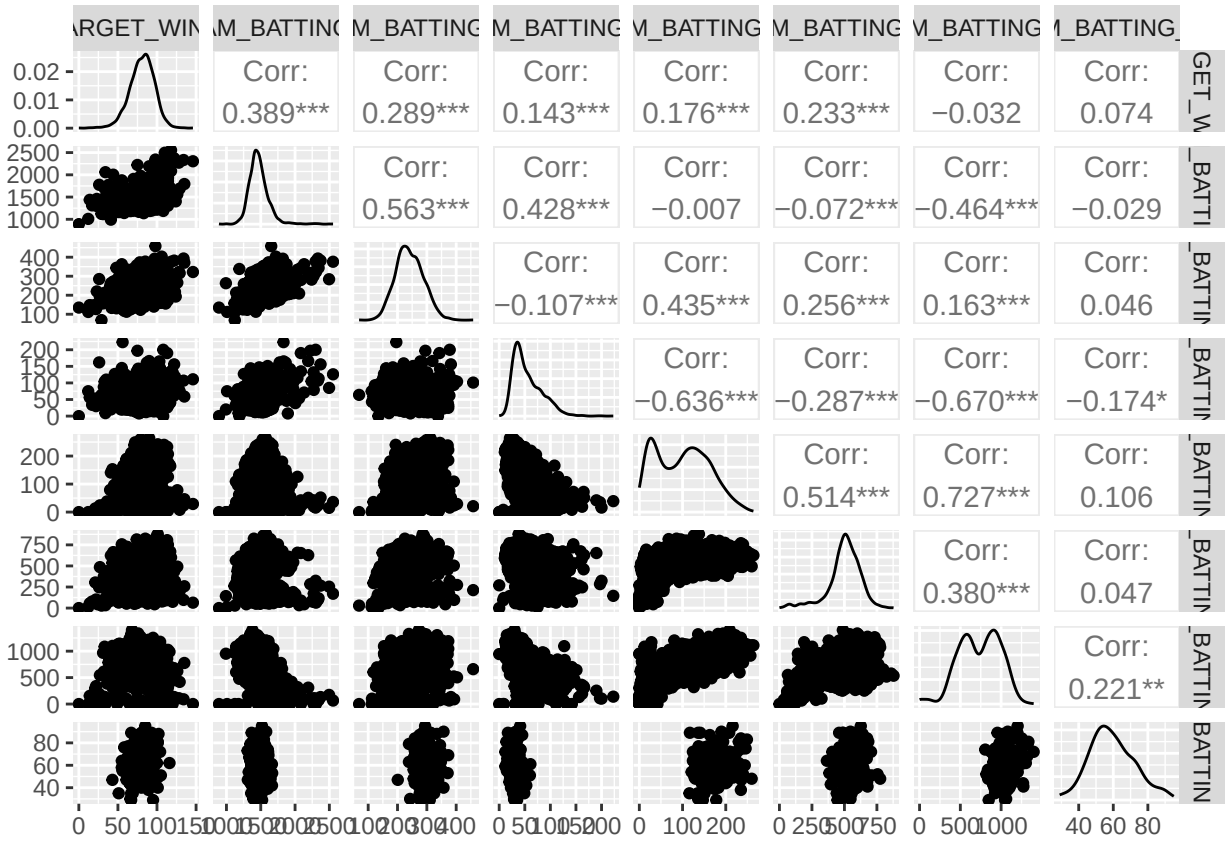


In many of these fields, there seems to be quite a lot of outliers that may need to be imputed.

Now that we have a sense of how the data is distributed, what do the relationships between the variables as well as with our target look like?

Correlations and Relationships

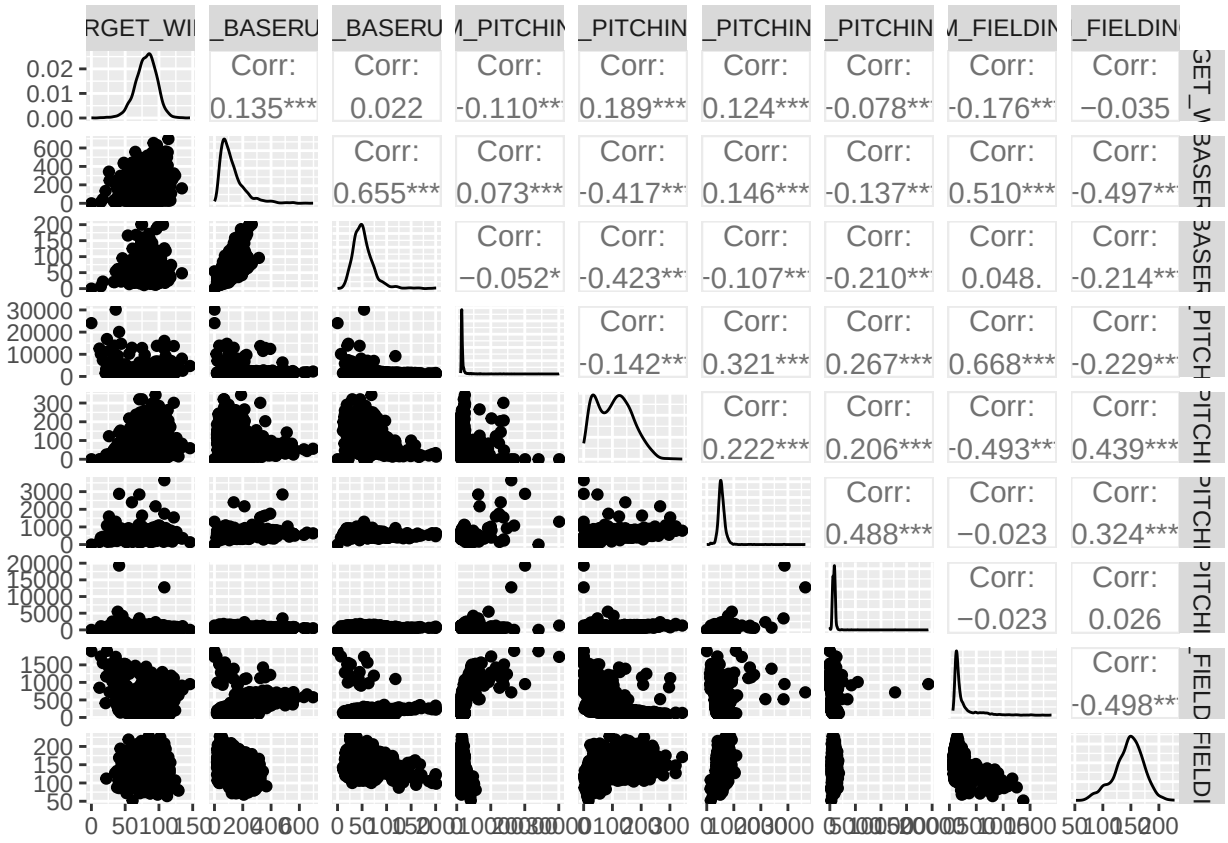
Let's see how each of these fields correlates with `TARGET_WINS` – we'll start with the batting fields.



Interestingly, it seems that every field is positively correlated except for `TEAM_BATTING_SO` (which makes sense as we were told that they have a positive impact except for the last one) and the positively impacted ones are ones that are statistically significant, minus `TEAM_BATTING_HBP`.

These fields are also pretty correlated with one another for the most part which may serve as an issue for our model.

What do the relationships look like for the rest of the fields?



Out of all of these fields, there are four with negative impacts:

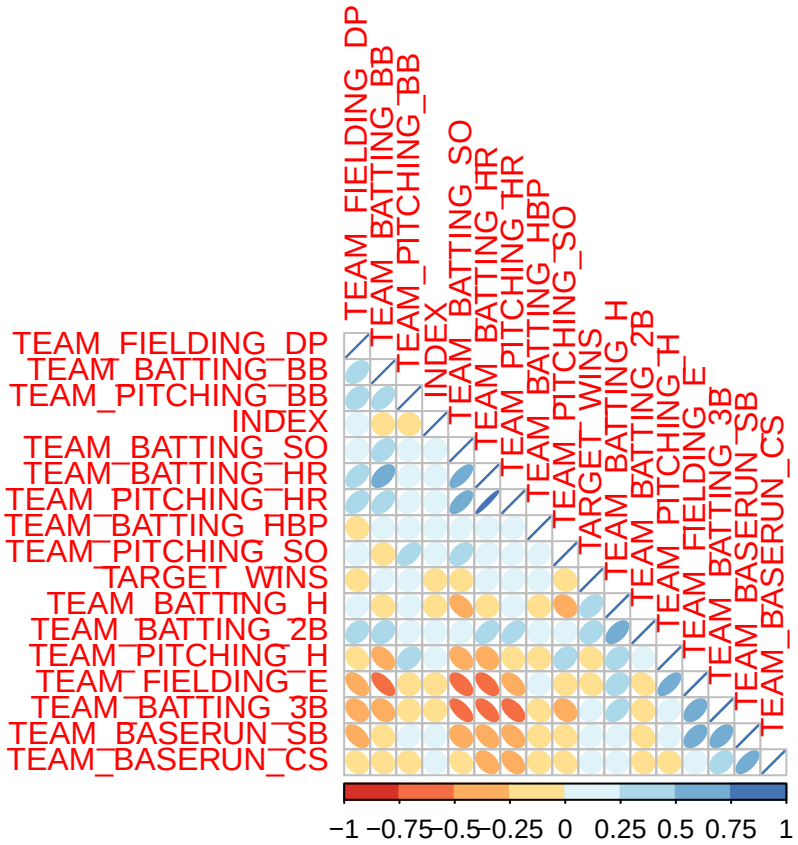
- `TEAM_PITCHING_H`
- `TEAM_PITCHING_SO`
- `TEAM_PITCHING_E`
- `TEAM_FIELDING_DP`

And the only ones that aren't statistically significant are:

- `TEAM_BASERUN_CS`
- `TEAM_FIELDING_DP`

Again, these fields are also pretty correlated with each other which may be an issue.

To view a more concise correlation analysis overall:



Looking at this, we can see an extremely strong correlation between `TEAM_PITCHING_HR` and `TEAM_BATTING_HR`.

Keeping this information in mind as we move closer to creating our model, we'll move to the next step of preparing our data.

Data Preparation

Missing Data

We saw earlier that quite a few fields had missing data; to deal with each of these, the details will be below as we should handle situations on a case-by-case basis. We will use a limit of 20% as the max we will allow for missing data. Note that median was picked a majority of the time here as it is less prone to outliers than average:

- `TEAM_BATTING_SO`: 102 NA's (4.48%) – imputing median
- `TEAM_BASERUN_SB`: 131 NA's (5.76%) – imputing median
- `TEAM_BASERUN_CS`: 772 NA's (33.92%) – too much missing, removing this field
- `TEAM_BATTING_HBP`: 2085 NA's (91.61%) – too much missing, removing this field
- `TEAM_PITCHING_SO`: 102 NA's (4.48%) – imputing median
- `TEAM_FIELDING_DP`: 286 NA's (12.57%) – imputing median

In addition to these changes, we will remove the following fields:

- `INDEX`: told not to use

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
TARGET_WINS	1	2276	80.79	15.75	82.0	81.31	14.83	0	146	146	-0.4
TEAM_BATTING_H	2	2276	1469.27	144.59	1454.0	1459.04	114.16	891	2554	1663	1.5
TEAM_BATTING_2B	3	2276	241.25	46.80	238.0	240.40	47.44	69	458	389	0.2
TEAM_BATTING_3B	4	2276	55.25	27.94	47.0	52.18	23.72	0	223	223	1.1
TEAM_BATTING_HR	5	2276	99.61	60.55	102.0	97.39	78.58	0	264	264	0.1
TEAM_BATTING_BB	6	2276	501.56	122.67	512.0	512.18	94.89	0	878	878	-1.0
TEAM_BATTING_SO	7	2276	736.25	242.91	750.0	742.82	272.80	0	1399	1399	-0.3
TEAM_BASERUN_SB	8	2276	123.39	85.41	101.0	109.73	57.82	0	697	697	2.0
TEAM_PITCHING_H	9	2276	1779.21	1406.84	1518.0	1555.90	174.95	1137	30132	28995	10.3
TEAM_PITCHING_BB	10	2276	553.01	166.36	536.5	542.62	98.59	0	3645	3645	6.7
TEAM_PITCHING_SO	11	2276	817.54	540.54	813.5	797.90	245.37	0	19278	19278	22.6
TEAM_FIELDING_E	12	2276	246.48	227.77	159.0	193.44	62.27	65	1898	1833	2.9
TEAM_FIELDING_DP	13	2276	146.72	24.54	149.0	147.91	19.27	52	228	176	-0.4

- `TEAM_PITCHING_HR`: due to the high correlation with `TEAM_BATTING_HR`, this is being removed for a cleaner dataset

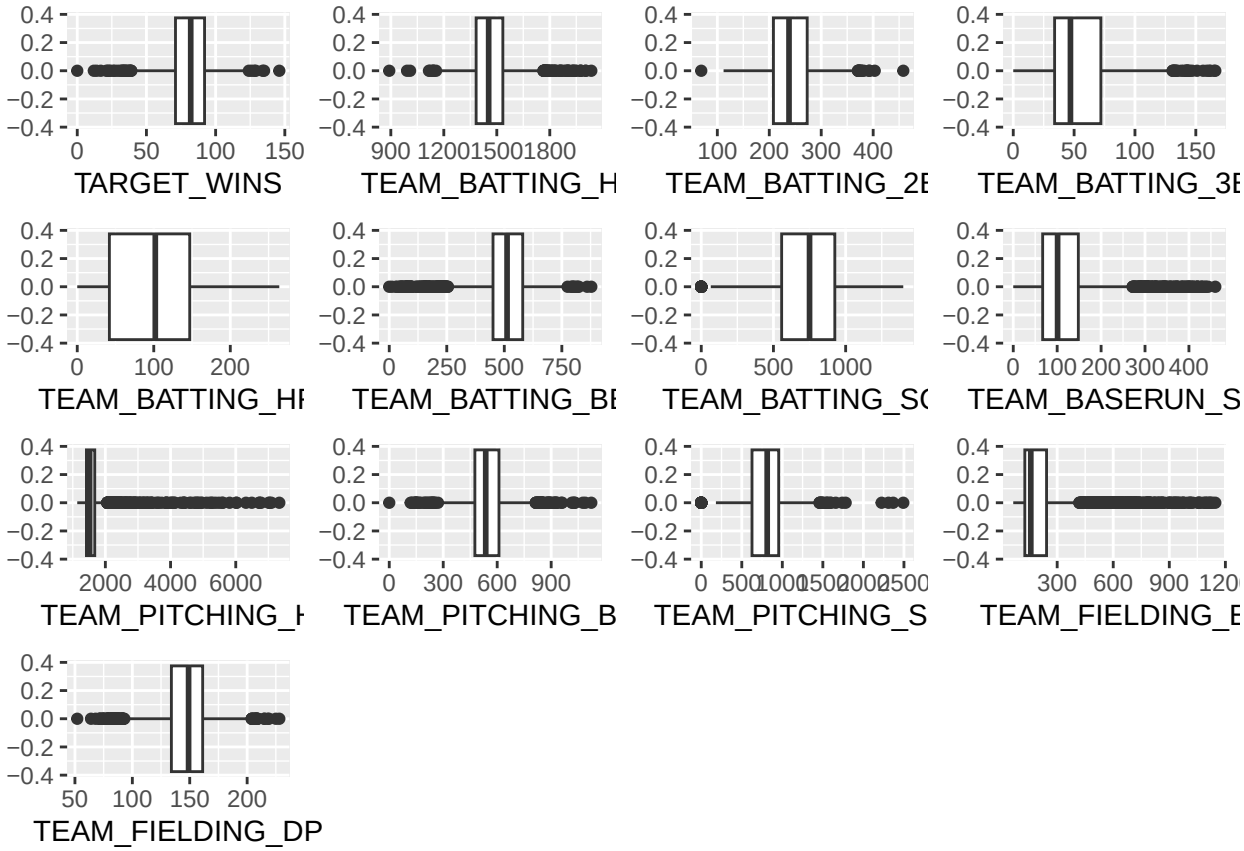
No more nulls!

Outliers

There are some pretty extreme outliers scattered throughout most of the fields (see the boxplot in the previous section). While it is understandable that these may happen occasionally, it is a safe assumption to believe that the really extreme ones won't happen in your average game. To account for these, we will use the median of the data again to replace these outliers if they are more than four standard deviations from the mean for the following fields:

- `TEAM_BATTING_H`: 16 records
- `TEAM_BATTING_3B`: 4 records
- `TEAM_BASERUN_SB`: 19 records
- `TEAM_PITCHING_H`: 21 records
- `TEAM_PITCHING_BB`: 10 records
- `TEAM_PITCHING_SO`: 5 records
- `TEAM_FIELDING_E`: 29 records

This will be a total of 104 changed records.



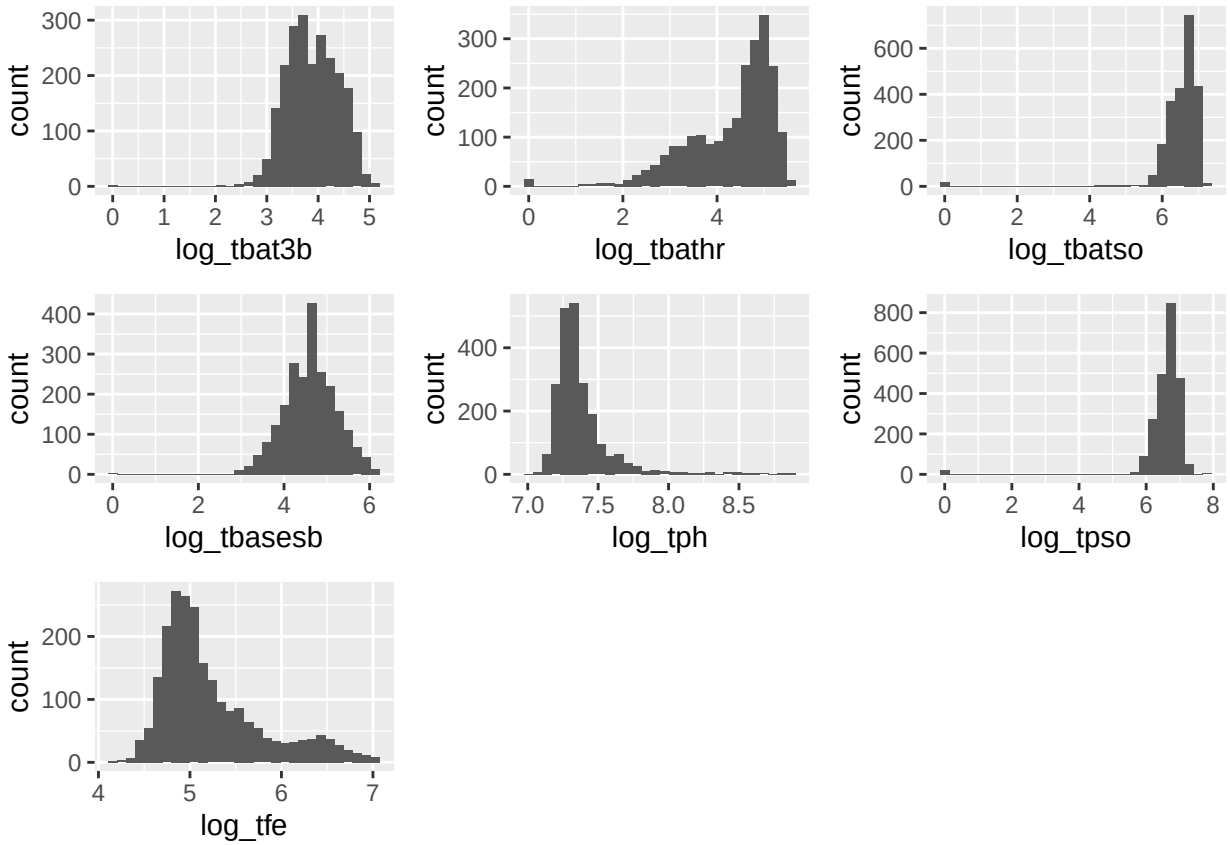
Compared to the box plots before transforming this data, it does look a bit cleaner!

Transform Non-Normal Variables

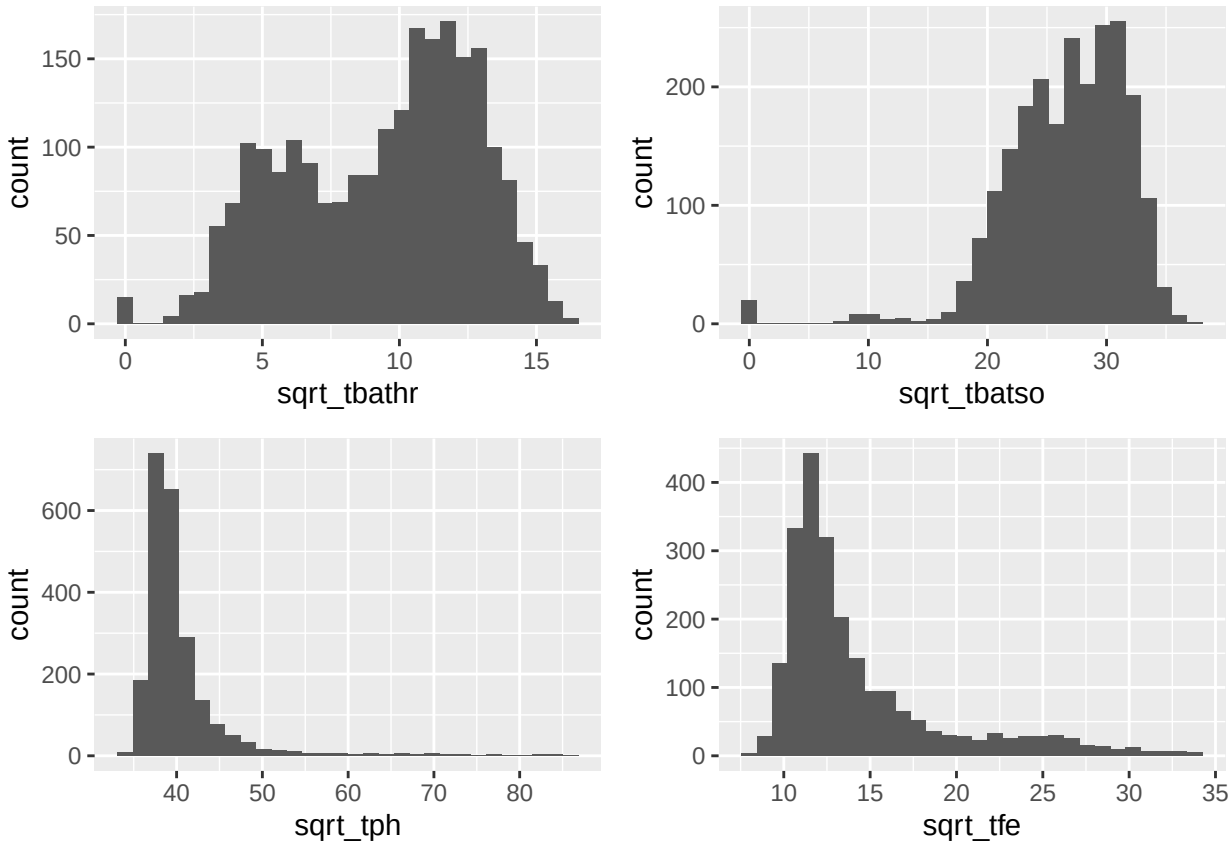
The last alteration before modeling is ensuring that our variables are normal by transforming the ones that don't seem to have much of normal distribution. The fields with distributions that aren't as normal are:

- `TEAM_BATTING_3B`
- `TEAM_BATTING_HR`
- `TEAM_BATTING_SO`
- `TEAM_BASERUN_SB`
- `TEAM_PITCHING_H`
- `TEAM_PITCHING_SO`
- `TEAM_FIELDING_E`

We'll try transforming these with `log` first and if that doesn't work, then we'll `sqrt` it.



It looks like this fixed a few variables, however `TEAM_BATTING_HR`, `TEAM_BATTING_SO`, `TEAM_PITCHING_H`, and `TEAM_FIELDING_E` still look a little off. Let's trying using `sqrt` on them.



While not perfectly normal, these look better than how they started – we can move onto modeling now that we've finished trying to transform all of our variables!

Model Creation

Before doing anything, we will split the data into training and test sets with a 70/30 split.

We'll go through two sets of models:

- Model 1: Start from using all the coefficients as is and only use the transformed ones if they don't seem to have a solid impact on the model
- Model 2: Start with all normalized (to the best of our ability) variables and select from there

Model 1A

This first model will use all the fields pre-transformed ones.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_FIELDING_E + TEAM_FIELDING_DP + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H + TEAM_PITCHING_SO, data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.926  -8.384  -0.026   8.774  60.258
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.4477157  6.4437935   5.191 2.37e-07 ***
## TEAM_BATTING_H    0.0220635  0.0047094   4.685 3.04e-06 ***
## TEAM_BATTING_2B   0.0208637  0.0104379   1.999 0.045797 *
## TEAM_BATTING_3B   0.1020283  0.0210024   4.858 1.30e-06 ***
## TEAM_BATTING_HR   0.0978542  0.0117049   8.360  < 2e-16 ***
## TEAM_BATTING_BB   0.0520581  0.0079146   6.577 6.48e-11 ***
## TEAM_BATTING_SO  -0.0080839  0.0051670  -1.565 0.117894
## TEAM_BASERUN_SB   0.0221353  0.0054761   4.042 5.55e-05 ***
## TEAM_FIELDING_E   0.0033783  0.0031288   1.080 0.280426
## TEAM_FIELDING_DP -0.1228365  0.0160025  -7.676 2.85e-14 ***
## TEAM_PITCHING_BB -0.0252696  0.0068363  -3.696 0.000226 ***
## TEAM_PITCHING_H   0.0025736  0.0008828   2.915 0.003603 **
## TEAM_PITCHING_SO -0.0019535  0.0039245  -0.498 0.618716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.45 on 1582 degrees of freedom
## Multiple R-squared:  0.262,  Adjusted R-squared:  0.2564
## F-statistic:  46.8 on 12 and 1582 DF,  p-value: < 2.2e-16
```

Coefficient Evaluation

Looking at the model's coefficients and whether they had a positive or negative impact, `TEAM_FIELDING_E`, `TEAM_FIELDING_DP`, `TEAM_PITCHING_H`, and `TEAM_PITCHING_SO` do not make sense – double plays should have a positive impact, hits allowed should have a negative impact, and strikeouts by pitchers should have a positive impact; it seems like these coefficients are counter-intuitive as they are all opposite.

There are several possible reasons for this mismatch:

- Collinearity: It's possible that these fields are correlated with other variables that have a stronger negative/positive impact on wins (depending on the direction they're going in) that are opposite what we expect.
- Sample Size: The effect of these factors on wins may be subtle and require a larger sample size to be accurately reflected in the model.
- Interactions: There might be interactions or nonlinear relationships at play that the linear regression model cannot capture.

At the very least, `TEAM_FIELDING_E`, `TEAM_PITCHING_H`, and `TEAM_PITCHING_SO` don't have much of an impact on the numbers as their absolute values are less than 0.01 – `TEAM_FIELDING_DP` however is at a -0.123 which holds a bit more power. We'll opt to drop the first three due to their small impact and keep the last as it seems to be important to the model.

Significance Evaluation

A majority of the fields used are statistically significant at a 0 code level sans `TEAM_BATTING_SO`, `TEAM_FIELDING_E`, and `TEAM_PITCHING_SO`. We have transformed versions of these fields so we will be using that now, specifically `sqrt_tbatso` (`TEAM_FIELDING_E` and `TEAM_PITCHING_SO` were dropped in the previous step).

Model 1B

To review the changes, we will be removing `TEAM_FIELDING_E`, `TEAM_PITCHING_H`, and `TEAM_PITCHING_SO` in this model and using `sqrt_tbatso` instead of its original field.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + sqrt_tbatso +
##     TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_PITCHING_BB, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.815  -8.496  -0.108   8.884  63.588
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      39.122385   6.883252   5.684 1.57e-08 ***
## TEAM_BATTING_H    0.026200   0.004509   5.811 7.51e-09 ***
## TEAM_BATTING_2B   0.018155   0.010399   1.746 0.081038 .
## TEAM_BATTING_3B   0.115909   0.019845   5.841 6.30e-09 ***
## TEAM_BATTING_HR   0.093271   0.010888   8.567  < 2e-16 ***
## TEAM_BATTING_BB   0.043323   0.005494   7.886 5.78e-15 ***
## sqrt_tbatso      -0.456442   0.112906  -4.043 5.54e-05 ***
## TEAM_BASERUN_SB   0.021067   0.005273   3.995 6.76e-05 ***
## TEAM_FIELDING_DP -0.124489   0.016011  -7.775 1.35e-14 ***
## TEAM_PITCHING_BB -0.019644   0.005106  -3.847 0.000124 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.5 on 1585 degrees of freedom
## Multiple R-squared:  0.2552, Adjusted R-squared:  0.251
## F-statistic: 60.34 on 9 and 1585 DF,  p-value: < 2.2e-16
```

Coefficient Evaluation

`TEAM_FIELDING_DP` is still negative in this model and actually has more of an impact in this model than the previous. It is very statistically significant and so we will opt to keep it for the next run.

Significance Evaluation

It's interesting how `TEAM_BATTING_2B` seems to have lost most of its significance. We can test out removing it in our next iteration.

Our transformed `sqrt_tbatso` seems to have performed much better in the meantime and we'll continue using it as is.

Model 1C

To summarize our changes, we will just be removing `TEAM_BATTING_2B` from the model in our third iteration.

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

```
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + sqrt_tbatso + TEAM_BASERUN_SB +
##     TEAM_FIELDING_DP + TEAM_PITCHING_BB, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.090  -8.434  -0.092   8.856  66.551
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      35.649301   6.593754   5.407 7.41e-08 ***
## TEAM_BATTING_H    0.030906   0.003617   8.545  < 2e-16 ***
## TEAM_BATTING_3B   0.111486   0.019696   5.660 1.79e-08 ***
## TEAM_BATTING_HR   0.094831   0.010858   8.734  < 2e-16 ***
## TEAM_BATTING_BB   0.043927   0.005487   8.006 2.27e-15 ***
## sqrt_tbatso      -0.422645   0.111306  -3.797 0.000152 ***
## TEAM_BASERUN_SB   0.020057   0.005245   3.824 0.000136 ***
## TEAM_FIELDING_DP -0.124243   0.016021  -7.755 1.57e-14 ***
## TEAM_PITCHING_BB -0.019752   0.005109  -3.866 0.000115 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.5 on 1586 degrees of freedom
## Multiple R-squared:  0.2538, Adjusted R-squared:   0.25
## F-statistic: 67.42 on 8 and 1586 DF,  p-value: < 2.2e-16
```

Coefficient Evaluation

TEAM_PITCHING_H in this iteration is the only counter-intuitive coefficient value and it is still quite small in impact.

Significance Evaluation

Interestingly enough, all of our variables are at a high level of significance.

Overall, this model performed pretty similarly to the previous iteration.

Model 2A

This model will begin using normalized variables and transformed versions if their original forms aren't normal.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     log_tbat3b + sqrt_tbathr + TEAM_BATTING_BB + sqrt_tbatso +
##     log_tbasesb + sqrt_tfe + TEAM_FIELDING_DP + TEAM_PITCHING_BB +
##     sqrt_tph + log_tpso, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.574  -8.535  -0.139   8.785  54.677
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.987136   7.909271  -0.631 0.528430
## TEAM_BATTING_H    0.020671   0.004724   4.376 1.29e-05 ***
## TEAM_BATTING_2B   0.014970   0.010329   1.449 0.147430
## log_tbat3b        6.396708   1.104477   5.792 8.39e-09 ***
## sqrt_tbathr       1.918556   0.221489   8.662  < 2e-16 ***
## TEAM_BATTING_BB   0.048012   0.006997   6.862 9.73e-12 ***
## sqrt_tbatso      -0.356901   0.183054  -1.950 0.051388 .
## log_tbasesb       3.898806   0.671195   5.809 7.59e-09 ***
## sqrt_tfe          0.012914   0.122345   0.106 0.915952
## TEAM_FIELDING_DP -0.124448   0.016277  -7.646 3.58e-14 ***
## TEAM_PITCHING_BB -0.021824   0.005973  -3.654 0.000267 ***
## sqrt_tph          0.348281   0.098431   3.538 0.000414 ***
## log_tpso         -1.177755   0.910507  -1.294 0.196022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.34 on 1582 degrees of freedom
## Multiple R-squared:  0.274,  Adjusted R-squared:  0.2685
## F-statistic: 49.75 on 12 and 1582 DF,  p-value: < 2.2e-16
```

Coefficient Evaluation

Similar to model 1, `TEAM_FIELDING_DP`, `sqrt_tfe`, `sqrt_tph`, and `log_tpso` are counter intuitive for this model where there expected impact does not match the coefficient presented. Nonetheless, all of these fields have a strong level of significance and a high level of impact; due to these factors, we will opt to keep them in the model even though it doesn't make sense conceptually.

Significance Evaluation

`TEAM_BATTING_2B`, `sqrt_tfe`, `log_tpso`, and our intercept suffer from not being significant in this model; since they are nowhere near close to even being slightly significant, we will opt to remove the features mentioned from the next iteration as there's nothing we can do about the intercept.

Model 2B

This model removes `TEAM_BATTING_2B`, `sqrt_tfe` and `log_tpso` from this iteration.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + log_tbat3b + sqrt_tbathr +
##     TEAM_BATTING_BB + sqrt_tbatso + log_tbasesb + TEAM_FIELDING_DP +
##     TEAM_PITCHING_BB + sqrt_tph, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.130  -8.596  -0.252   8.645  57.058
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.867014   7.644656  -1.160    0.246
## TEAM_BATTING_H    0.023419   0.003769   6.214 6.57e-10 ***
```

```
## log_tbat3b        5.869785   1.004009   5.846 6.09e-09 ***
## sqrt_tbathr       2.001921   0.210898   9.492  < 2e-16 ***
## TEAM_BATTING_BB   0.050490   0.006609   7.640 3.74e-14 ***
## sqrt_tbatso      -0.507026   0.117087  -4.330 1.58e-05 ***
## log_tbasesb       3.988417   0.647055   6.164 8.98e-10 ***
## TEAM_FIELDING_DP -0.123281   0.016233  -7.595 5.25e-14 ***
## TEAM_PITCHING_BB -0.024465   0.005555  -4.404 1.13e-05 ***
## sqrt_tph          0.367516   0.092041   3.993 6.82e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.34 on 1585 degrees of freedom
## Multiple R-squared:  0.2721, Adjusted R-squared:  0.268
## F-statistic: 65.83 on 9 and 1585 DF,  p-value: < 2.2e-16
```

Coefficient Evaluation

Once again, `TEAM_FIELDING_DP` and `sqrt_tph` continue to be counterintuitive yet at a high level of significance.

Significance Evaluation

The intercept has gotten closer to significance, however it still hasn't reached at least 0.1, unfortunately.

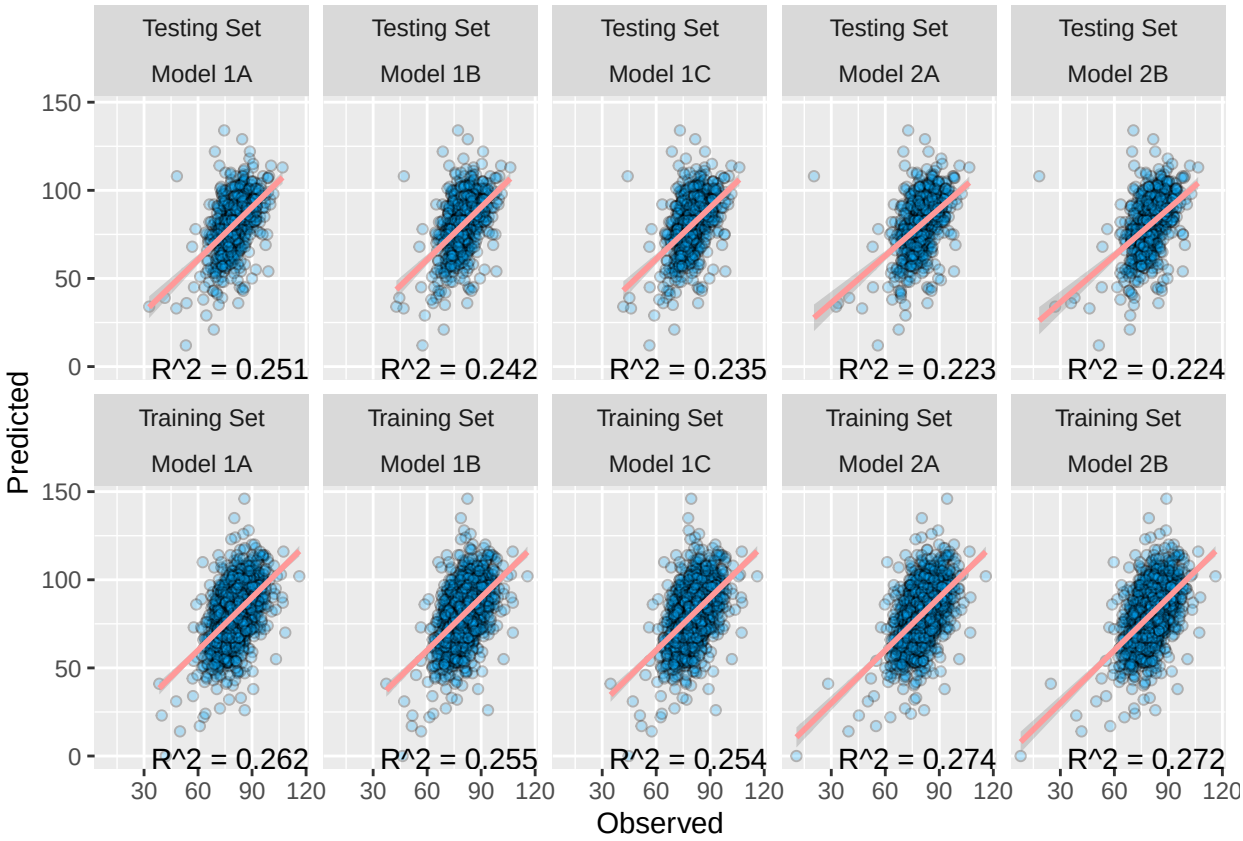
Model Selection

We'll start by looking at mean squared error, adjusted r-squared, and F-statistics before plotting residuals after.

```
##               MSE Adjusted.R.Squared F.Statistic     F.p.value
## Model 1A 179.3393          0.2563886    46.79941  1.438270e-95
## Model 1B 180.9881          0.2509727    60.34371  4.031269e-95
## Model 1C 181.3361          0.2500056    67.41864  2.134609e-95
## Model 2A 176.4246          0.2684745    49.75067 4.220953e-101
## Model 2B 176.8815          0.2679679    65.83335 6.355863e-103
```

In general, it looks like all of these models performed similarly when comparing MSE values; the second iterations (2A and 2B) perform marginally better as they are lower in value. Looking at adjusted r-squared, the second iterations once again pull ahead slightly with 2A performing a bit better than 2B. With the f-statistics, model 1C has the highest at 67.41, however model 2B is not too far behind with 65.83.

Next, we will compare how these models do with the test dataset and compare residuals.



Interestingly, it seems that the first set of models performed better than the second iteration when using the test dataset. The best performing model was the first one which was just using all features in the state they're provided (so untransformed). Visually, the residual plots don't seem to vary too much; they all are around the same r-squared so the change between them isn't too apparent.

Final Selection

Based on all of the factors shown above, model 2B seems to be the most viable. It performs solidly when we compared the MSE, adjusted r-squared, and F-statistic while also was the slightly better performing one out of the second round of models when using the test dataset. It uses transformed/more normalized variables while also filtering out the statistically insignificant features as well, resulting in a well-performing model with relevant features.