Stephen Woods

Moneyball Assignment #2

Spring 2022

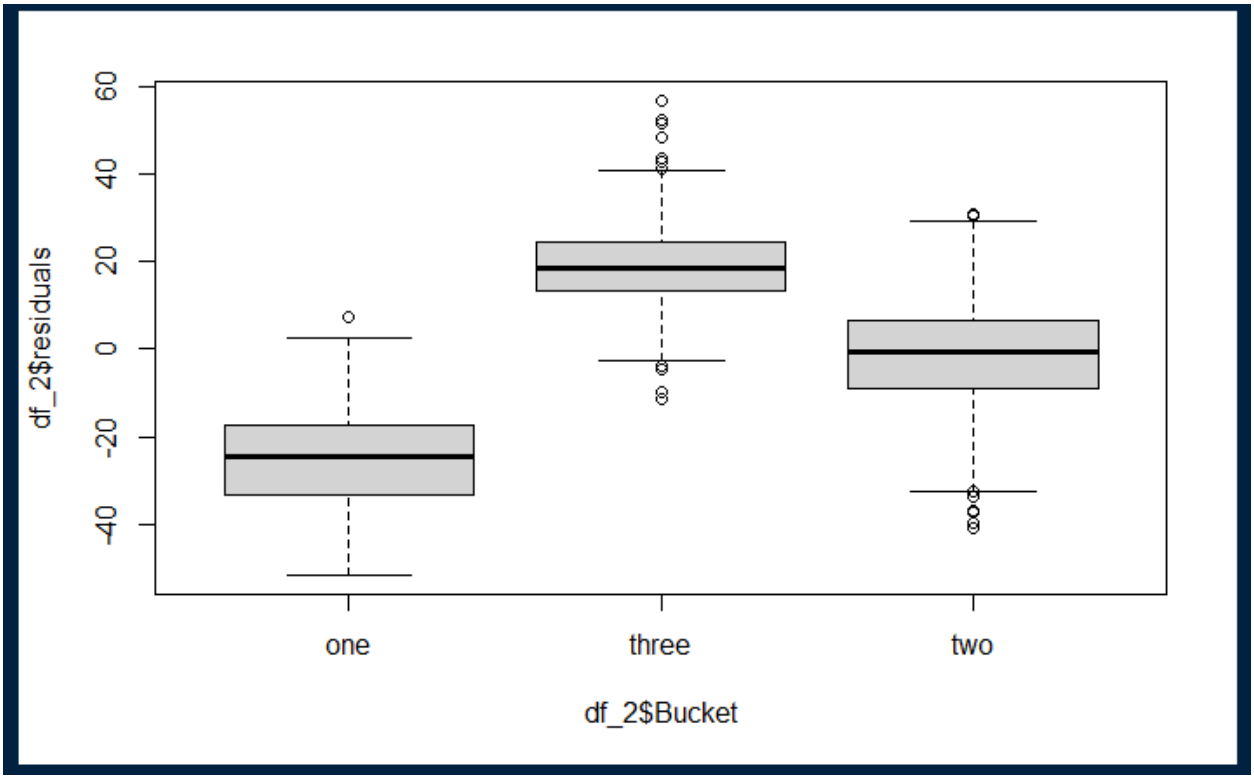
Summary

This notebook features linear modeling on a dataset related to the movie 'Moneyball'. The objective of this notebook is to form a linear model that predicts a given baseball team's total number of games won for a season.

Section 1 – Feature Engineering

Grouping target wins into three buckets shows that my linear model fit best to bucket 2, where the mean difference between predicted wins and target wins was only -1.37. Mean residuals for the other groups were larger. This indicates my linear model does not predict as well to values on either end of the wins spectrum.

Bucket <chr>	TARGET_WINS <dbl>	Pred_wins <dbl>	mean_diff <dbl>
one	39.66667	64.52701	-24.860340
three	106.31579	87.23773	19.078058
two	79.26401	80.63151	-1.367499



I then grouped target wins into four buckets, created dummy variables for each of those four buckets, but dropped the first bucket/dummy variable as that will be considered my base category. Adding the win category dummy variables to my model increased adjusted r squared from .3204 to .7615. This makes sense, as I essentially gave my model the answer broken down into buckets.

Adding truncated team batting hits dummy variables to my model, with one dummy representing hits below 1122 and one dummy for hits above 2333, increased my adjusted r square value from .7615 to .7618. I consider this a negligible difference.

lr_md5, model results before log transforming the response variable:

```
Residuals:
    Min      1Q  Median      3Q     Max
-51.610  -8.874   0.134   8.583  57.706

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.321e+01  5.405e+00   2.443 0.014628 *
TEAM_BATTING_H        4.478e-02  4.048e-03  11.062  < 2e-16 ***
TEAM_BATTING_2B      -2.061e-02  9.399e-03  -2.193 0.028395 *
TEAM_BATTING_3B       7.346e-02  1.712e-02   4.291 1.86e-05 ***
TEAM_BATTING_HR       4.616e-02  2.817e-02   1.639 0.101385
TEAM_BATTING_BB       7.772e-03  6.274e-03   1.239 0.215512
TEAM_BATTING_SO      -8.855e-03  2.699e-03  -3.281 0.001050 **
TEAM_BASERUN_SB       3.611e-02  4.389e-03   8.228 3.19e-16 ***
TEAM_PITCHING_H      -5.326e-04  4.490e-04  -1.186 0.235648
TEAM_PITCHING_HR      9.590e-03  2.505e-02   0.383 0.701910
TEAM_PITCHING_BB     -3.524e-03  4.625e-03  -0.762 0.446161
TEAM_PITCHING_SO      4.318e-03  1.042e-03   4.143 3.55e-05 ***
TEAM_FIELDING_E      -2.194e-02  2.463e-03  -8.909  < 2e-16 ***
trunc_team_batting_1122 -3.221e+01  9.297e+00  -3.464 0.000541 ***
trunc_team_batting_2333  8.271e+00  7.208e+00   1.148 0.251289
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.25 on 2260 degrees of freedom
Multiple R-squared:  0.2887,    Adjusted R-squared:  0.2843
F-statistic: 65.53 on 14 and 2260 DF,  p-value: < 2.2e-16
```

Lr_md6, model results after log transforming the response variable:

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.20211 -0.10638  0.01387  0.11561  0.60719

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              3.421e+00  7.341e-02  46.606  < 2e-16 ***
TEAM_BATTING_H           6.644e-04  5.498e-05  12.084  < 2e-16 ***
TEAM_BATTING_2B         -3.274e-04  1.277e-04  -2.564 0.010399 *
TEAM_BATTING_3B          9.923e-04  2.325e-04   4.267 2.06e-05 ***
TEAM_BATTING_HR          3.219e-04  3.825e-04   0.842 0.400092
TEAM_BATTING_BB          4.203e-05  8.520e-05   0.493 0.621829
TEAM_BATTING_SO         -9.998e-05  3.666e-05  -2.728 0.006430 **
TEAM_BASERUN_SB          4.887e-04  5.961e-05   8.198 4.04e-16 ***
TEAM_PITCHING_H         -2.040e-05  6.098e-06  -3.346 0.000832 ***
TEAM_PITCHING_HR         2.982e-04  3.402e-04   0.876 0.380932
TEAM_PITCHING_BB        -8.230e-06  6.281e-05  -0.131 0.895769
TEAM_PITCHING_SO         7.494e-05  1.415e-05   5.295 1.31e-07 ***
TEAM_FIELDING_E         -3.865e-04  3.345e-05 -11.554  < 2e-16 ***
trunc_team_batting_1122 -8.311e-01  1.263e-01  -6.582 5.75e-11 ***
trunc_team_batting_2333  1.221e-01  9.789e-02   1.247 0.212576
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.18 on 2260 degrees of freedom
Multiple R-squared:  0.337,     Adjusted R-squared:  0.3329
F-statistic: 82.06 on 14 and 2260 DF,  p-value: < 2.2e-16
```

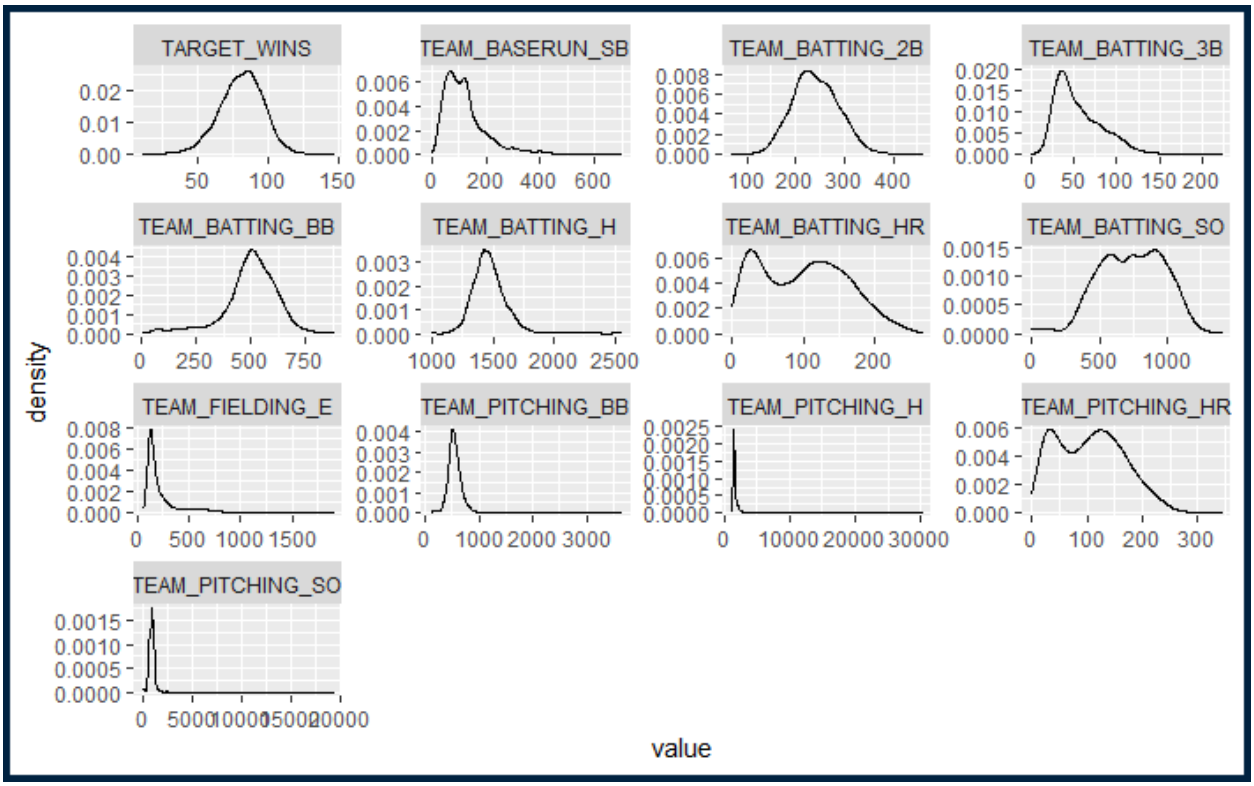
Interpretation for each model should ultimately be the same, but log transforming the response causes the coefficients to be in log form, so they are smaller in magnitude than the coefficients of the model that was not log transformed. So, for the log transformed model, a one unit increase in xi will cause a coefficient_xi * 100% increase in y. Log transforming the response variable can potentially improve the fit when the assumptions of normality are violated. Specifically, log transforming a response can be beneficial when the residual distribution is not normal. We observed this non normality when looking at the residuals differences between group means for target wins. The middle group had lower residual values, while the outer groups had higher residual values.

Section 2 – Transformations

A base linear model using all fields as predictors in my dataframe produced an adjusted r square value of .2843. Log transforming the response variable increased adjusted r square from .2843 to .3329, an increase of .0486.

Computing VIFS for each predictor in this model showed VIFS above 30 for TEAM_BATTING_HR and TEAM_PITCHING_HR. I chose to drop the TEAM_BATTING_HR variable from my model to avoid multicollinearity between TEAM_BATTING_HR and TEAM_PITCHING_HR. This increased the adjusted r square of my model by .01.

I then generated a histogram of all predictor variables in my model to determine if any predictor variable transformations made sense:



We see many variables feature strong positive right skew. Taking the sqrt of some of these right skew variables may help them resemble a more normal distribution. Doing so increased the adjusted r square of my model by .0058. This model (lr_md8) produced an adjusted r square of .3388 making it my highest performing model.

Section 3 – Model Selection

lr_md8 was selected as my highest performing model, producing an adjusted r square of .3388.

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.19934 -0.10522  0.01143  0.11380  0.63294

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              3.573e+00  8.503e-02  42.021  < 2e-16 ***
TEAM_BATTING_H           6.515e-04  5.477e-05  11.895  < 2e-16 ***
TEAM_BATTING_2B         -3.680e-04  1.273e-04  -2.890 0.003885 **
TEAM_BATTING_3B          1.218e-03  2.293e-04   5.312 1.19e-07 ***
TEAM_BATTING_BB          1.845e-04  9.117e-05   2.024 0.043094 *
TEAM_BATTING_SO         -1.472e-04  3.596e-05  -4.092 4.43e-05 ***
sqrt(TEAM_BASERUN_SB)    1.424e-02  1.484e-03   9.596  < 2e-16 ***
TEAM_PITCHING_H         -2.234e-05  5.865e-06  -3.808 0.000144 ***
TEAM_PITCHING_HR         5.773e-04  1.184e-04   4.877 1.15e-06 ***
sqrt(TEAM_PITCHING_BB)  -5.391e-03  3.389e-03  -1.591 0.111762
TEAM_PITCHING_SO         9.315e-05  1.253e-05   7.432 1.51e-13 ***
sqrt(TEAM_FIELDING_E)   -1.670e-02  1.424e-03 -11.721  < 2e-16 ***
trunc_team_batting_1122 -8.699e-01  1.250e-01  -6.960 4.43e-12 ***
trunc_team_batting_2333  1.202e-01  9.604e-02   1.252 0.210802
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1792 on 2261 degrees of freedom
Multiple R-squared:  0.3426,    Adjusted R-squared:  0.3388
F-statistic: 90.63 on 13 and 2261 DF,  p-value: < 2.2e-16
```

Note log transforming the response variable caused an increase of .0486, but square root transformations of predictors did not change model performance much. Interestingly, lr_md8 was the highest performing model on the training set, but the 2nd lowest performing model on the Kaggle test set:

Submission and Description	Public Score
submission6.csv an hour ago by Stephen Woods lr_md10	13.29084
submission5.csv an hour ago by Stephen Woods lr_md8	13.29084
submission4.csv an hour ago by Stephen Woods lr_md6	13.29915
submission3.csv 2 hours ago by Stephen Woods lr_md7	13.23259
submission2.csv 2 hours ago by Stephen Woods lr_md5	13.06406
submission1.csv 21 days ago by Stephen Woods linear model 1	12.44092

It appears that as I added more engineered features and variable or response transformations to my model, training accuracy increased but testing accuracy decreased. In other words, my models progressively generalized to test data worse. Further, my linear model from assignment 1 (submission 1/linear model 1), which featured a log10 transformation of the response and no engineered features, performed significantly better than all models from assignment 2.

Section 4 – lr_md8 Model Equation and comparison to lr_md2

Below is the equation for lr_md8, my highest performing model (on the training data) from assignment two:

Y = 3.57292 + 0.00065 * TEAM_BATTING_H + -0.00037 * TEAM_BATTING_2B + 0.00122 * TEAM_BATTING_3B + 0.00018 * TEAM_BATTING_BB + -0.00015 * TEAM_BATTING_SO + 0.01424 * sqrt(TEAM_BASERUN_SB) + -2e-05 * TEAM_PITCHING_H + 0.00058 * TEAM_PITCHING_HR + -0.00539 * sqrt(TEAM_PITCHING_BB) + 9e-05 * TEAM_PITCHING_SO + -0.0167 * sqrt(TEAM_FIELDING_E) + -0.86988 * trunc_team_batting_1122 + 0.12022 * trunc_team_batting_2333 + e

As these coefficients are log transformed, their magnitude is more difficult to interpret.

Below is the formula for my model from assignment one, which performed significantly better on the Kaggle test set:

Y = 1.44874 + 0.00032 * TEAM_BATTING_H + -0.00017 * TEAM_BATTING_2B + 0.00037 * TEAM_BATTING_3B + 9e-05 * TEAM_BATTING_HR + -3e-05 * TEAM_BATTING_BB + -1e-05 * TEAM_BATTING_SO + 2e-04 * TEAM_BASERUN_SB + -1e-05 * TEAM_PITCHING_H + 0.00013 * TEAM_PITCHING_HR + 4e-05 * TEAM_PITCHING_BB + 1e-05 * TEAM_PITCHING_SO + -0.00016 * TEAM_FIELDING_E + e

The only differences are the removal of TEAM_BATTING_HR, sqrt transforming some predictors, and the added trunc_team_batting fields to lr_md8. I will be curious to see how model performance compares on the other test set.