Introduction

This project is a deep dive into creating and selecting an adequate model to predict the wins of a baseball team. Linear Regression Models were created and compared against each other on different criteria, including AIC,VIFs and R-Squared values. There was also a multitude of variables provided to help develop the winning model. Amongst those variables, there was also missing and incorrect data. Most of this had to be imputed in order to maintain the integrity of the data set and delve actionable insights.

Section 1. Data Exploration

A thorough investigation of the Moneyball data was completed in order to get an idea of the relationship and structure of the data in the csv file. To get an idea of this, the str(), head() and summary() functions were run in R to get some outputs. There was a total of 2,276 observations across 17 variables. The summary function provided the basic statistics including 1st and 3rd quartiles along with means and medians. Most of the variables appeared to have similar means and medians, which helps the analyst understand that the data should be evenly distributed. The most glaring issue that was brought to light by the summary function was the variables that needed to be imputed. The NA's signified missing or lost data. TEAM_BASERUNNING_CS and TEAM_BATTING_HBP were the variables with the most missing data. These variables, along with others were addressed in the data prep stage. It was at this point that I also did a correlation matrix in R to see the relationships from a high level that TARGET_WINS had with the predictors. An image of this is below. There was also adequate enough data to be able to see that most of the batting statistics along with a few pitching statistics were closely correlated with the TARGET_WINS. A scatterplot matrix was put together as well and this also helped to back up and support the finding in the correlation matrix. The higher scoring correlations had a much better-defined trend/regression line amongst the data points. The next step in the data exploration process involved looking at histograms and boxplots. They helped to identify outliers as well as the density of where values were falling.

	INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTING_BB	TEAM_BATTING_SO
INDEX	1.00	-0.02	-0.02	0.01	-0.01	0.05	-0.03	NA
TARGET_WINS	-0.02	1.00	0.39	0.29	0.14	0.18	0.23	NA
TEAM_BATTING_H	-0.02	0.39	1.00	0.56	0.43	-0.01	-0.07	NA
TEAM_BATTING_2B	0.01	0.29	0.56	1.00	-0.11	0.44	0.26	NA
TEAM_BATTING_3B	-0.01	0.14	0.43	-0.11	1.00	-0.64	-0.29	NA
TEAM_BATTING_HR	0.05	0.18	-0.01	0.44	-0.64	1.00	0.51	NA
TEAM_BATTING_BB	-0.03	0.23	-0.07	0.26	-0.29	0.51	1.00	NA
TEAM_BATTING_SO	NA	NA	NA	NA	NA	NA	NA	1
TEAM_BASERUN_SB	NA	NA	NA	NA	NA	NA	NA	NA
TEAM_BASERUN_CS	NA	NA	NA	NA	NA	NA	NA	NA
TEAM_BATTING_HBP	NA	NA	NA	NA	NA	NA	NA	NA
TEAM_PITCHING_H	0.02	-0.11	0.30	0.02	0.19	-0.25	-0.45	NA
TEAM_PITCHING_HR	0.05	0.19	0.07	0.45	-0.57	0.97	0.46	NA
TEAM_PITCHING_BB	-0.02	0.12	0.09	0.18	0.00	0.14	0.49	NA
TEAM_PITCHING_SO	NA	NA	NA	NA	NA	NA	NA	NA
TEAM_FIELDING_E	-0.01	-0.18	0.26	-0.24	0.51	-0.59	-0.66	NA
TEAM_FIELDING_DP	NA	NA	NA	NA	NA	NA	NA	NA

variables had negative coeffients. In baseball, those are variables that typlically are considered positve statistics and help a team score runs or prevent them from happening, as is the case with the fielding variable. Overall, it didn't have much difference to the subset 12, which is the model that is discussed in the next section and ended up being the winning model.

Section 4. Select Model

Subset 12 ended up being the champion model. It was a model built using regsubsets and it had parameters that created 12 different sets of variables, with each set being the best selection for that number of variables. I used nvmax = 12 and nbest =1. This would be the same as subset 1 having 1 variable, subset 2 having 2 variable and subset 3 having 3 variables, up until it got to 12. I used a summary output that used asterics to identify how to pair the variables. Below is a picture of it.

	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTING_H	TEAM_BATTING_BB	TEAM_BATTING_SO	TEAM_BASERUN_SB	TEAM_BASERUN_CS
1 (1)	" "	" "	" "	"*"	" "	" "	" "	" "
2 (1)	" "	" "	" "	"*"	" "	" "	" "	" "
3 (1)	" "	" "	" "	"*"	" "	" "	"*"	" "
4 (1)	" "	" "	" "	"*"	" "	" "	"*"	" "
5 (1)	" "	" "	" "	"*"	" "	" "	"*"	" "
6 (1)	" "	" "	" "	"*"	" "	"*"	"*"	" "
7 (1)	" "	" "	"*"	"*"	" "	"*"	"*"	" "
8 (1)	" "	"*"	"*"	"*"	" "	"*"	"*"	" "
9 (1)	"*"	" "	"*"	"*"	"*"	"*"	"*"	" "
10 (1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	" "
11 (1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	" "
12 (1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

	TEAM_PITCHING_HR	TEAM_PITCHING_BB	TEAM_PITCHING_SO	TEAM_FIELDING_E	TEAM_FIELDING_DP
1 (1)	" "	" "	" "	" "	" "
2 (1)	" "	" "	" "	"*"	" "
3 (1)	" "	" "	" "	"*"	" "
4 (1)	" "	" "	" "	"*"	"*"
5 (1)	"*"	" "	" "	"*"	"*"
6 (1)	"*"	" "	" "	"*"	"*"
7 (1)	" "	" "	"*"	"*"	"*"
8 (1)	" "	" "	"*"	"*"	"*"
9 (1)	" "	" "	"*"	"*"	"*"
10 (1)	" "	" "	"*"	"*"	"*"
11 (1)	"*"	" "	"*"	"*"	"*"
12 (1)	"*"	" "	"*"	"*"	"*"

Subset 12 ended up with an R-Squared of .361. The adjusted R-Squared was .3576. The MSE was 158.4964 and the AIC was 18016.61. The R-Squared score and MSE were better for Subset 12 than Model 2, while the adjusted R-Squared and AIC were better for Model 2. My only concerns with this model was that I ended up with negative coefficients for TEAM_FIELDING_DP, TEAM_BATTING_2B and a positive coefficient for TEAM_BASERUNNING_CS. These are all metrics where the opposite is considered to be a success in baseball. Ultimately, I decided that having a lower mse would allow the model to more accurately reflect the data, even though the adjusted R-Squared and AIC both didn't affect or punished the model for additional variables.

Section 5. Model Formula

TARGET WINS =
30.9909146
- 0.0980972 * TEAM_FIELDING_DP
- 0.0351613 * TEAM_FIELDING_E
- 0.0163923 * TEAM_BATTING_SO
+ 0.0457459 * TEAM_BASERUN_SB
+ 0.0861166 * TEAM_BATTING_HR
+ 0.0447497 * TEAM_BATTING_H
+ 0.0017051 * TEAM_PITCHING_BB
+ 0.0266074 * TEAM_BATTING_3B
- 0.0176317 * TEAM_BATTING_2B
- 0.0107410 * TEAM_BASERUN_CS
+ 0.0029857 * TEAM_PITCHING_SO
+ 0.0049060 * TEAM_BATTING_BB

The above model is winning model, subset 12, which was discussed in section 4. The model makes sense with most coefficients. My main concerns are with the FIELDING_DP, PITCHING_BB, and BATTING_2B variables. They intuitively seem to have the inverse sign coefficients that one would expect, but that can be explained in other ways. PITCHING_BB could possibly help a team if they are intentionally walking the team's best hitters to avoid them from hitting in runners, which would cause less wins. BATTING_2B and FIELDING_DP should have a positive effect on wins, but instead have a negative in this model. This should be further investigated and maybe look further into the data to see if it a specific subset of teams causing this. I also believe that if the data included the flag to signify National League and American League, we could have further looked into the pitching and hitting statistics. One league has a DH, while the other has the pitcher hit. This might cause a significant difference amongst the variables and data.

Model Development Code

This code is provided in the separate file that was submitted.

Conclusion

This assignment was a good challenge in creating a multitude of different linear models to predict the wins of baseball team. The subset model that was the champion model and it included an overall better insight of the data by having the highest percent of variability explained and also the lowest mean square error. Overall, I think more data would help build better models potentially. By having which league the observations represent, we would be able to subset the data and possibly gain another insight to apply to our predictions and models. Over time, I would like to see how this data drifts. In the baseball world today, shifts on defense are being done based on models and statistics to improve wins. It would be interesting to see that as some type of metric as well.

Bingo Bonus

In order to receive a few bonus points (5-10), I decided to try and use a predictive mean matching algorithm within the mice package of R. This enabled me to avoid using the standard imputations of means and medians by individual columns in R by hand. I was able to save and export this file after each step, in-order to analyze what the program did. After looking at a csv file and comparing to the original, I believed the package did as I expected and did not accidentally iterate over the good existing data or cause any other issues. As another part of the bingo bonus, I also created and analyzed 12 individual subset models based upon the matrix output and selection algorithm.