|  |  |
| --- | --- |
| Gabriella Martinez | Maliat Islam |
| Ken Popkin | George Cruz Deschamps |
| Matthew Lucich | Karim Hammoud |



Table of Contents

[DATA EXPLORATION: 4](#_Toc83496886)

[Load the Data 4](#_Toc83496887)

[A picture is worth a thousand words 6](#_Toc83496888)

[DATA PREPARATION 9](#_Toc83496889)

[BUILD MODELS 11](#_Toc83496890)

[Model #1 11](#_Toc83496891)

[Two predictors: Base hits by batters and Hits allowed 11](#_Toc83496892)

[Model #2 12](#_Toc83496893)

[Four predictors: Base hits by batters, Hits allowed, Errors, and Walks allowed 12](#_Toc83496894)

[Model #3 13](#_Toc83496895)

[BSR Model (SaberMetrics) (data imputation) 13](#_Toc83496896)

[(Modified) Backward Elimination Model (omitting NAs) 14](#_Toc83496897)

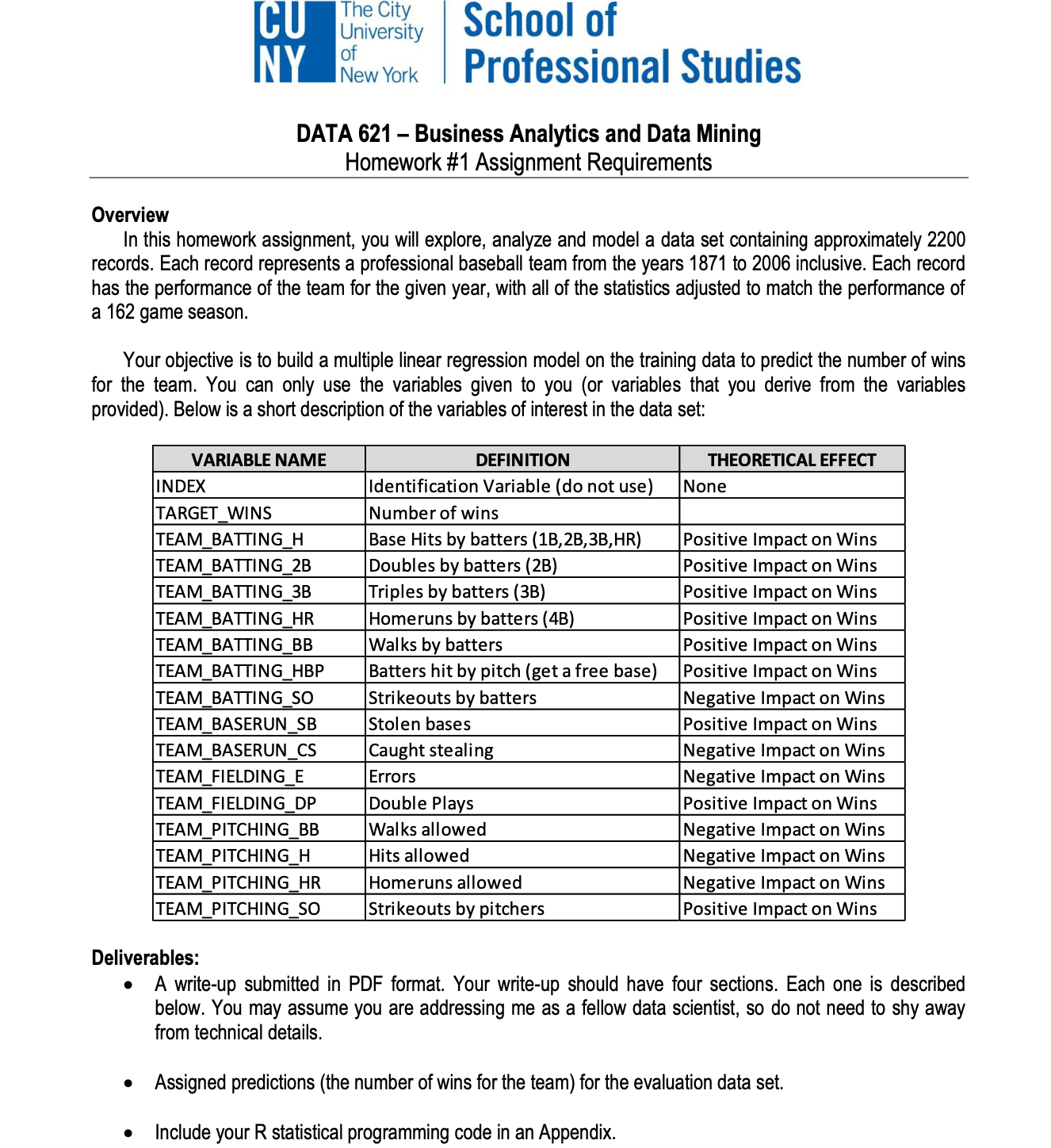
[SELECT MODELS 15](#_Toc83496898)

[Verifying OLS Regression Assumptions 15](#_Toc83496899)

[Model Selection 17](#_Toc83496900)

[References 19](#_Toc83496901)

[Appendix A: Code 20](#_Toc83496902)



In this data set we are trying to identify good and bad teams in major league baseball team’s season. We are assuming some of the predictors will be higher for good teams. We will try to predict how many times a team will win in this season.

## DATA EXPLORATION:

We can observe the response variable (TARGET\_WINS) looks to be normally distributed. This supports the working theory that there are good teams and bad teams. There are also a lot of average teams.

There are also quite a few variables with missing values. and,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.

### Load the Data

#### Summary of the Train data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable Name | Min Value | 1st Quantile | Median | Mean | 3rd Quantile | Max Value | NA’s |
| INDEX | 1.0 | 630.8 | 1270.5 | 1268.5 | 1915.5 | 2535.0 |  |
| TARGET\_WINS | 0.0 | 71.0 | 82.0 | 80.79 | 92.0 | 146.0 |  |
| TEAM\_BATTING\_H | 891 | 1383 | 1454 | 1469 | 1537 | 2554 |  |
| TEAM\_BATTING\_2B | 69.0 | 208.0 | 238.0 | 241.2 | 273.0 | 458.0 |  |
| TEAM\_BATTING\_3B | 0.00 | 34.00 | 47.00 | 55.25 | 72.00 | 223.00 |  |
| TEAM\_BATTING\_HR | 0.00 | 42.00 | 102.00 | 99.61 | 147.00 | 264.00 |  |
| TEAM\_BATTING\_BB | 0.00 | 451.0 | 512.0 | 501.6 | 580.0 | 878.0 |  |
| TEAM\_BATTING\_SO | 0.00 | 548.0 | 750.0 | 735.6 | 930.0 | 1399.0 | 102 |
| TEAM\_BASERUN\_SB | 0 | 66 | 101 | 124.8 | 156 | 697 | 131 |
| TEAM\_BASERUN\_CS | 0 | 38 | 49 | 52.8 | 62 | 201 | 772 |
| TEAM\_BATTING\_HBP | 29 | 50.5 | 58 | 59.36 | 67 | 95 | 2085 |
| TEAM\_PITCHING\_H | 1137 | 1419 | 1518 | 1779 | 1682 | 30132 |  |
| TEAM\_PITCHING\_HR B | 0.0 | 50.0 | 107.0 | 105.7 | 150.0 | 343.0 |  |
| TEAM\_PITCHING\_B | 0.0 | 476.0 | 536.5 | 553.0 | 611.0 | 3645.0 |  |
| TEAM\_PITCHING\_SO | 0.0 | 615.0 | 813.5 | 817.7 | 968.0 | 19278.0 | 102.0 |
| TEAM\_FIELDING\_E | 65.0 | 127.0 | 159.0 | 246.5 | 249.2 | 1898.0 |  |

Table 4.1: Summary of the Train Data

#### Summary of the Test data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable Name | Min Value | 1st Quantile | Median | Mean | 3rd Quantile | Max Value | NA’s |
| INDEX | 9 | 708 | 1249 | 1264 | 1832 | 2525 |  |
| TEAM\_BATTING\_H | 819 | 1387 | 1455 | 1469 | 1548 | 2170 |  |
| TEAM\_BATTING\_2B | 44 | 210 | 239 | 241.3 | 278.5 | 376 |  |
| TEAM\_BATTING\_3B | 14 | 35 | 52 | 55.91 | 72 | 155 |  |
| TEAM\_BATTING\_HR | 0 | 44.5 | 101 | 95.63 | 135.5 | 242 |  |
| TEAM\_BATTING\_BB | 15 | 436.5 | 509 | 499 | 565.5 | 792 |  |
| TEAM\_BASERUN\_SB | 0 | 59 | 92 | 123.7 | 151.8 | 580 | 13 |
| TEAM\_BASERUN\_CS | 0 | 38 | 49.5 | 52.32 | 63 | 154 | 87 |
| TEAM\_BATTING\_HBP | 42 | 53.5 | 62 | 62.37 | 67.5 | 96 | 240 |
| TEAM\_PITCHING\_H | 1155 | 1426 | 1515 | 1813 | 1681 | 22768 |  |
| TEAM\_PITCHING\_HR | 0 | 52 | 104 | 102.1 | 142.5 | 336 |  |
| TEAM\_PITCHING\_BB | 136 | 471 | 526 | 552.4 | 606.5 | 2008 |  |
| TEAM\_PITCHING\_SO | 0 | 613 | 745 | 799.7 | 938 | 9963 | 18 |
| TEAM\_FIELDING\_E | 73 | 131 | 163 | 249.7 | 252 | 1568 |  |
| TEAM\_FIELDING\_DP | 69 | 131 | 148 | 146.1 | 164 | 204 | 31 |

Table 5.1: Summary of the Test Data

#### Standard Deviation for the Train Data Variables

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | 693.28867 | **TEAM\_BATTING\_HR** | 56.33221 |
| **TEAM\_BATTING\_H** | 150.65523 | **TEAM\_BATTING\_BB** | 120.59215 |
| **TEAM\_BATTING\_2B** | 49.51612 | **TEAM\_BATTING\_SO** | 243.11114 |
| **TEAM\_BATTING\_3B** | 27.1441 | **TEAM\_BASERUN\_SB** | 93.38796 |
| **TEAM\_BASERUN\_CS** | 23.10457 | **TEAM\_PITCHING\_BB** | 172.9501 |
| **TEAM\_BATTING\_HBP** | 12.707 | **TEAM\_PITCHING\_SO** | 634.3059 |
| **TEAM\_PITCHING\_H** | 1662.913 | **TEAM\_FIELDING\_E** | 230.9026 |
| **TEAM\_PITCHING\_HR** | 57.6549 | **TEAM\_FIELDING\_DP** | 25.88387 |

Table 5.2: Summary of the Train Data

#### Standard Deviation for all of the test data

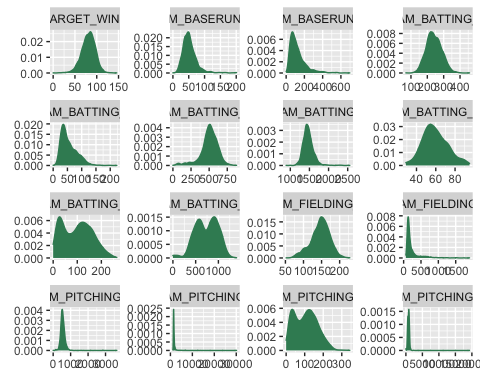
|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | 736.34904 | **TEAM\_BATTING\_HR** | 60.54687 |
| **TEAM\_BATTING\_H** | 144.59120 | **TEAM\_BATTING\_BB** | 122.67086 |
| **TEAM\_BATTING\_2B** | 46.80141 | **TEAM\_BATTING\_SO** | 248.52642 |
| **TEAM\_BATTING\_3B** | 27.93856 | **TEAM\_BASERUN\_SB** | 87.79117 |
| **TEAM\_BASERUN\_CS** | 22.95634 | **TEAM\_PITCHING\_BB** | 166.35736 |
| **TEAM\_BATTING\_HBP** | 12.96712 | **TEAM\_PITCHING\_SO** | 553.08503 |
| **TEAM\_PITCHING\_H** | 1406.84293 | **TEAM\_FIELDING\_E** | 227.77097 |
| **TEAM\_PITCHING\_HR** | 61.29875 | **TEAM\_FIELDING\_DP** | 26.22639 |
| **TARGET\_WINS** | 15.75215 |  |  |

Table 5.2: Summary of the Test Data

### A picture is worth a thousand words

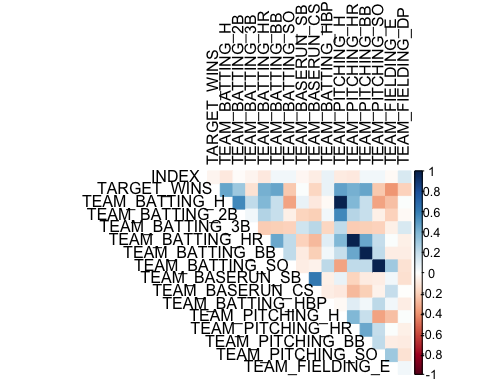
#### Box plot of the Train data

#### Variable Distributions



#### Log Variable Distributions

#### Correlations with Response Variable



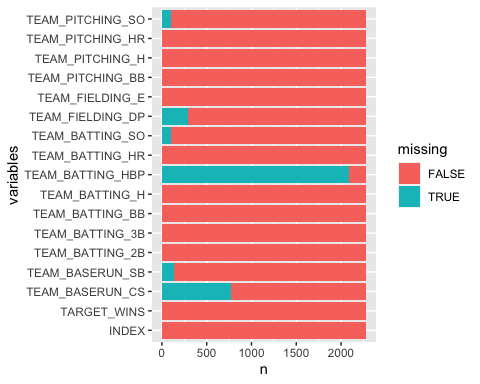
### DATA PREPARATION

As part of the Data Preparation, we gathered counts of the missing values (NA) for the train data set [[1]](#footnote-1)

|  |  |
| --- | --- |
| Variable Name | NA’s |
| TEAM\_BATTING\_SO | 102 |
| TEAM\_BASERUN\_SB | 131 |
| TEAM\_BASERUN\_CS | 772 |
| TEAM\_BATTING\_HBP | 2085 |
| TEAM\_PITCHING\_SO | 102.0 |

Visualization and percentage of NA values [[2]](#footnote-2)

#### 

Alternative NA values visualization [[3]](#footnote-3) 

Since 92% of the data for the TEAM\_BATTING\_HBP is missing, the variable has been removed from both test and train data. TEAM\_BASERUN\_CS is a runner up with the next highest amount of NA at 34%.

We can see that some of the variables, in special TEAM\_BATTING\_HBP, have an inordinate amount of NA’s and will probably not be useful in our projections.

## BUILD MODELS

## Model #1

### Two predictors: Base hits by batters and Hits allowed

Using a manual review, below are the features selected for the first model and the supporting reason/s.

TEAM\_BATTING\_H = Base hits by batters: it’s impossible to win in baseball without getting to the bases and hitting the ball is the primary means to accomplish this.

TEAM\_PITCHING\_H = Hits allowed: winning without a good defense is difficult and in baseball preventing the other team from getting hits is a good defense strategy.

Only two features are selected for the first model - start small and build up seems like a good approach.

When we create the Regression Model and print a summary we get:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TARGET WINS** | | |
| *Predictors* | *Estimates* | *CI* | *p* |
| (Intercept) | 15.64 | 8.65 – 22.64 | **1.22e-05** |
| TEAM\_BATTING\_H | 0.05 | 0.04 – 0.05 | **2.20e-74** |
| TEAM\_PITCHING\_H | -0.00 | -0.00 – -0.00 | **1.77e-25** |
| Observations | 1707 | | |
| R2 / R2 adjusted | 0.189 / 0.188 | | |

The p values are 0, which per the criteria of “keep a feature if the p-value is <0.05” recommends that we keep both these features. But, the adjusted R-squared is TERRIBLE at around 21%. Even though the R-squared is poor it’s simple to run this model with the test data, so we’ll do that next.

Evaluate the first model results using RMSE

## 13.6336

## Model #2

### Four predictors: Base hits by batters, Hits allowed, Errors, and Walks allowed

Using a manual review, below are the features selected for the second model and the supporting reason/s.

We’ll keep the features from the first model (due to low p-values) and add two more features… TEAM\_FIELDING\_E = Errors: errors are costly in terms of immediate impact, but could also impact the team in other ways (i.e. a high occurrence could impact team comraderie and confidence in each other)

TEAM\_PITCHING\_BB = Walks allowed: putting players on base for “free” is more opportunity for points

Create the Regression Model

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TARGET WINS** | | |
| *Predictors* | *Estimates* | *CI* | *p* |
| (Intercept) | 7.26 | 0.16 – 14.37 | **4.52e-02** |
| TEAM\_BATTING\_H | 0.05 | 0.04 – 0.05 | **3.08e-83** |
| TEAM\_PITCHING\_H | -0.00 | -0.00 – -0.00 | **1.89e-07** |
| TEAM\_FIELDING\_E | -0.01 | -0.02 – -0.01 | **9.07e-10** |
| TEAM\_PITCHING\_BB | 0.01 | 0.01 – 0.02 | **2.46e-08** |
| Observations | 1707 | | |
| R2 / R2 adjusted | 0.238 / 0.236 | | |

Evaluate the second model results using RMSE

## 13.30535

The increase from two features in the first model to four features in the second model did not yield a noticeable improvement. The Adjusted R2 on the training data improved slightly, but the RMSE for all practical purposes stayed the same at around 13; which is a poor RMSE implying that both models have poor predictive capability.

## Model #3

### BSR Model (SaberMetrics) (data imputation)

*Base runs (BsR) is a baseball statistic invented by sabermetrician David Smyth to estimate the number of runs a team “should have”* *scored given their component offensive statistics, as well as the number of runs a hitter or pitcher creates or allows.* *It measures essentially the same thing as Bill James runs created, but as sabermetrician Tom M. Tango points out, base* *runs models the reality of the run-scoring process “significantly better than any other run estimator”.*

*Cleaning Data*

For the creation of Model 3, besides the removal of the HBP variable, we also imputed missing data by linear regression to: TEAM\_BATTING\_SO, TEAM\_BASERUN\_SB, TEAM\_BASERUN\_CS, TEAM\_PITCHING\_SO and TEAM\_FIELDING\_DP

The simplest formula for BsR, uses only the most common batting statistics:

Create the Regression Model *BSR*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TARGET WINS** | | |
| *Predictors* | *Estimates* | *CI* | *p* |
| (Intercept) | 45.24 | 39.04 – 51.44 | **6.20e-44** |
| BSR | 0.05 | 0.05 – 0.05 | **1.49e-113** |
| TEAM\_PITCHING\_SO | 0.01 | 0.01 – 0.01 | **2.67e-13** |
| TEAM\_FIELDING\_E | -0.04 | -0.04 – -0.04 | **3.50e-78** |
| TEAM\_FIELDING\_DP | -0.17 | -0.19 – -0.14 | **1.79e-39** |
| Observations | 1707 | | |
| R2 / R2 adjusted | 0.316 / 0.314 | | |

Evaluate the model results using RMSE: **14.25345**

Model #4

### (Modified) Backward Elimination Model (omitting NAs)

Due to previously learning how to perform Backward Elimination and it being possible to perform manually, we decided to include a model that resulted from the procedure. The process was performed with imputed data (via MICE) as well as data with NAs removed. The latter showed stronger results, therefore the final model was fitted with the NA omitted data.

According to Faraway, Backward Elimination is when you start with all predictors in the model, then remove the predictor with the highest p-value as long as it is above your p-value threshold (e.g. 0.05). Then refit the model and continue the process until only predictors with p-values below your threshold remain.

Additionally, we took steps to remove variables with non-intuitive coefficients. For instance, TEAM\_FIELDING\_DP and TEAM\_PITCHING\_SO were unexpectedly showing negative effects on wins. While there could be potential intervening variables giving these variables true predictive power, we opted to remove the variables from the model due to the possibility they were significant by chance and due to our bias towards parsimony. Further, RMSE did not drastically worsen when removed.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TARGET WINS** | | |
| *Predictors* | *Estimates* | *CI* | *p* |
| (Intercept) | 56.25 | 44.40 – 68.10 | **1.11e-18** |
| TEAM\_BASERUN\_SB | 0.05 | 0.02 – 0.07 | **5.20e-04** |
| TEAM\_BATTING\_HR | 0.06 | 0.02 – 0.09 | **6.22e-04** |
| TEAM\_BATTING\_BB | 0.04 | 0.02 – 0.05 | **4.87e-06** |
| TEAM\_FIELDING\_E | -0.05 | -0.08 – -0.01 | **1.49e-02** |
| Observations | 364 | | |
| R2 / R2 adjusted | 0.210 / 0.201 | | |

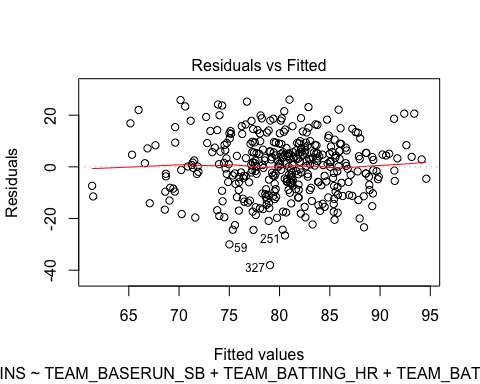
RMSE: 11.081

## SELECT MODELS

### Verifying OLS Regression Assumptions

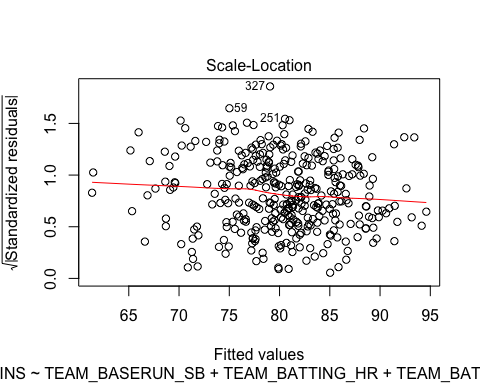
Assumption: Mean of residuals is zero

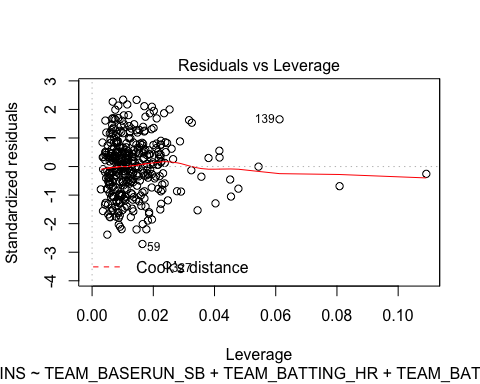
## [1] 2.396206e-17

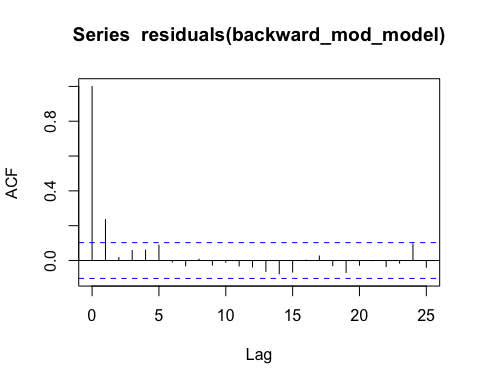


Chart, line chart

Description automatically generated







### Model Selection

First, before fully evaluating models we validated that all individual predictors had p-values below 0.05, the cutoff for a 95% confidence level. Additionally, we validated that the models F-statistics were also significant at a 95% confidence level.

Then, the two primary statistics used to choose our final model were adjusted R-squared and root mean square error (RMSE). Adjusted R-squared helped guide model selection since, like R-squared, adjusted R-squared measures the amount of variation in the dependent variable explained by the independent variables, except with a correction to ensure only independent variables with predictive power raise the statistic. RMSE was perhaps even more crucial to model selection as it is the measure of the standard deviation of the residuals, essentially a measure of accuracy in the same units as the response variable. To ensure the model can generalize to unobserved data, we calculated the RMSE on our test set.

Backward elimination saw a RMSE of approximately 10, noticeably outperforming other models. Therefore, we chose the backward elimination model even with a slightly worse adjusted R-squared. Additionally, since all top performing models included four predictors, parsimony was not a consideration.

Lastly, we verified the forward selection model meets OLS regression assumptions. These included: no significant multicollinearity, the mean of residuals is zero, homoscedasticity of residuals, and no significant auto-correlation. We deemed all assumptions had been met, but note, there is a slight trend in the residuals vs fitted plot (Assumption: Homoscedasticity of residuals) which may indicate a small nonlinear trend.

### References

Bhandari, Aniruddha, “Key Difference between R-squared and Adjusted R-squared for Regression Analysis”, Analytics Vidhya, 2020 <https://www.analyticsvidhya.com/blog/2020/07/difference-between-r-squared-and-adjusted-r-squared/>

Glen., Stephanie “RMSE: Root Mean Square Error”, StatisticsHowTo.com <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/>

Gupta, Aryansh, “Linear Regression Assumptions and Diagnostics in R”, RPubs, <https://rpubs.com/aryn999/LinearRegressionAssumptionsAndDiagnosticsInR>

Kim, Bommae, “Understanding Diagnostic Plots for Linear Regression Analysis”, University of Virginia Library, <https://data.library.virginia.edu/diagnostic-plots/>

Base Runs. 18 Nov. 2010, <http://tangotiger.net/wiki_archive/Base_Runs.html>.

Lüdecke, Daniel. Summary of Regression Models as HTML Table. 10 July 2021, <https://cran.r-project.org/web/packages/sjPlot/vignettes/tab_model_estimates.html>.

## 

## Appendix A: Code

## DATA EXPLORATION:  
#We can observe the response variable (TARGET\_WINS) looks to be normally distributed. This supports the working theory that there are good teams and bad teams. There are also a lot of average teams.  
  
#There are also quite a few variables with missing values. and,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.  
  
### Load the Data  
  
  
# Set seed for reproducibility  
set.seed(621)  
  
train <-read.csv("https://raw.githubusercontent.com/akarimhammoud/Data\_621/main/Assignment\_1/data/moneyball-training-data.csv")  
  
evaluation <-read.csv("https://raw.githubusercontent.com/akarimhammoud/Data\_621/main/Assignment\_1/data/moneyball-evaluation-data.csv")  
  
  
# Summary of the data  
  
summary(train)  
summary(evaluation)  
  
  
# Glimpse of the data  
  
glimpse(train)  
  
glimpse(evaluation)  
  
  
# Find SD for all of the train and test data  
  
apply(train,2,sd, na.rm=TRUE)  
  
apply(evaluation,2,sd, na.rm=TRUE)  
  
  
# Box plot the data   
  
ggplot(stack(train), aes(x = ind, y = values)) +   
 geom\_boxplot() +  
 theme(legend.position="none") +  
 theme(axis.text.x=element\_text(angle=45, hjust=1))   
  
# Variable Distributions  
  
train %>%  
 gather(variable, value, TARGET\_WINS:TEAM\_FIELDING\_DP) %>%  
 ggplot(., aes(value)) +   
 geom\_density(fill = "#3A8B63", color="#3A8B63") +   
 facet\_wrap(~variable, scales ="free", ncol = 4) +  
 labs(x = element\_blank(), y = element\_blank())  
  
  
#Log Variable Distributions  
  
  
train\_log <- log(train)  
  
train\_log %>%  
 gather(variable, value, TARGET\_WINS:TEAM\_FIELDING\_DP) %>%  
 ggplot(., aes(value)) +   
 geom\_density(fill = "#3A8B63", color="#3A8B63") +   
 facet\_wrap(~variable, scales ="free", ncol = 4) +  
 labs(x = element\_blank(), y = element\_blank())  
  
  
# Correlations with Response Variable  
  
train %>%  
 gather(variable, value, -TARGET\_WINS) %>%  
 ggplot(., aes(value, TARGET\_WINS)) +   
 geom\_point(fill = "#628B3A", color="#628B3A") +   
 geom\_smooth(method = "lm", se = FALSE, color = "black") +   
 facet\_wrap(~variable, scales ="free", ncol = 4) +  
 labs(x = element\_blank(), y = "Wins")  
  
  
  
train %>%   
 cor(., use = "complete.obs") %>%  
 corrplot(., method = "color", type = "upper", tl.col = "black", diag = FALSE)  
  
  
  
### DATA PREPARATION  
  
# ^[https://statisticsglobe.com/count-number-of-na-values-in-vector-and-column-in-r]  
  
#NA counts for the train data set  
colSums(is.na(train))  
  
# ^[https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html]  
  
#visulaization and percentage of NA values  
vis\_miss(train)  
  
  
  
# ^[https://datavizpyr.com/visualizing-missing-data-with-barplot-in-r/]  
  
#alternative NA values visualization  
train %>%  
 summarise\_all(list(~is.na(.)))%>%  
 pivot\_longer(everything(),  
 names\_to = "variables", values\_to="missing") %>%  
 count(variables, missing) %>%  
 ggplot(aes(y=variables,x=n,fill=missing))+  
 geom\_col()  
  
  
#Since 92% of the data for the TEAM\_BATTING\_HBP is missing, the variable has been removed from both test #and train data. TEAM\_BASERUN\_CS is a runner up with the next highest amount of NA at 34%.  
  
  
#removes the TEAM\_BATTING\_HBP due to high # of NAs  
train\_full <- train %>% dplyr::select(-c(TEAM\_BATTING\_HBP))  
evaluation <- evaluation %>% dplyr::select(-c(TEAM\_BATTING\_HBP))  
  
  
# ^[https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R-Manual/R-Manual5.html]  
  
#creates CSV in your current working directory of R  
write.csv(train\_full,'hw1\_train\_data.csv')  
write.csv(evaluation, 'hw1\_evaluation\_data.csv')  
  
  
  
# Create train, test split  
train <- train\_full %>% dplyr::sample\_frac(.75)  
test <- dplyr::anti\_join(train\_full, train, by = 'INDEX')  
  
  
  
## BUILD MODELS  
  
  
## Model #1  
### Two predictors: Base hits by batters and Hits allowed  
#Using a manual review, below are the features selected for the first model and the supporting reason/s.  
  
#TEAM\_BATTING\_H = Base hits by batters: it's impossible to win in baseball without getting to the bases # and hitting the ball is the primary means to accomplish this.  
  
#TEAM\_PITCHING\_H = Hits allowed: winning without a good defense is difficult and in baseball preventing #the other team from getting hits is a good defense strategy.  
  
#Only two features are selected for the first model - start small and build up seems like a good approach.  
  
#<B> Create the Regression Model </B>   
  
# Build the first model and produce a summary  
first\_model <- lm(TARGET\_WINS ~ TEAM\_BATTING\_H + TEAM\_PITCHING\_H, data = train)  
summary(first\_model)  
  
  
  
#The p values are 0, which per the criteria of "keep a feature if the p-value is <0.05" recommends that #we keep both these features. But, the adjusted R-squared is TERRIBLE at around 21%. Even though the #R-squared is poor it's simple to run this model with the test data, so we'll do that next.   
  
  
#Predict with the first model training data  
first\_model\_predictions = predict(first\_model, test)  
  
#Evaluate the first model results using RMSE  
rmse(test$TARGET\_WINS, first\_model\_predictions)  
  
  
  
## Model #2  
### Four predictors: Base hits by batters, Hits allowed, Errors, and Walks allowed  
#Using a manual review, below are the features selected for the second model and the supporting reason/s.  
  
#We'll keep the features from the first model (due to low p-values) and add two more features...  
#TEAM\_FIELDING\_E = Errors: errors are costly in terms of immediate impact, but could also impact the team in other ways (i.e. a high occurrence could impact team comraderie and confidence in each other)  
  
#TEAM\_PITCHING\_BB = Walks allowed: putting players on base for "free" is more opportunity for points   
  
#<B> Create the Regression Model </B>   
  
  
# Build the second model and produce a summary  
second\_model <- lm(TARGET\_WINS ~ TEAM\_BATTING\_H + TEAM\_PITCHING\_H + TEAM\_FIELDING\_E + TEAM\_PITCHING\_BB, data = train)  
summary(second\_model)  
  
  
#Predict with the second model training data  
second\_model\_predictions = predict(second\_model,test)  
  
#Evaluate the second model results using RMSE  
rmse(test$TARGET\_WINS, second\_model\_predictions)  
  
#The increase from two features in the first model to four features in the second model did not yield a noticeable improvement. The Adjusted R2 on the training data improved slightly, but the RMSE for all practical purposes stayed the same at around 13; which is a poor RMSE implying that both models have poor predictive capability.  
  
## Model #3  
### BSR Model (SaberMetrics) (data imputation)  
# \*Base runs (BsR) is a baseball statistic invented by sabermetrician David Smyth to estimate the number of runs a team "should have"\*  
#\*scored given their component offensive statistics, as well as the number of runs a hitter or pitcher #creates or allows.\*   
#\*It measures essentially the same thing as Bill James runs created, but as sabermetrician Tom M. Tango points out, base\*   
#\*runs models the reality of the run-scoring process "significantly better than any other run estimator".\*  
  
#\*Cleaning Data\*  
  
# load data  
data <- read.csv('hw1\_train\_data.csv')  
  
#imput data by regression:   
data\_imp <- mice(data, method = "norm.predict", m = 1)  
  
#complete data   
data\_complete <- complete(data\_imp)  
  
  
# The simplest, uses only the most common batting statistics[2]  
  
#$A = H + BB - HR$  
#$B = (1.4 \* TB - .6 \* H - 3 \* HR + .1 \* BB) \* 1.02$  
#$C = AB - H$  
#$D = HR$  
  
#$BsR = \frac{(A \* B)}{(B + C)} + D$  
  
  
data3 <- data\_complete %>%   
 rowwise() %>%  
 mutate(TEAM\_BATTING\_AB = sum( TEAM\_BATTING\_H,TEAM\_BATTING\_BB,TEAM\_BATTING\_SO, na.rm=TRUE),  
 TEAM\_BATTING\_1B = TEAM\_BATTING\_H - (TEAM\_BATTING\_2B + TEAM\_BATTING\_3B + TEAM\_BATTING\_HR),  
 TEAM\_BATTING\_TB = TEAM\_BATTING\_1B + (2 \* TEAM\_BATTING\_2B) + (3 \* TEAM\_BATTING\_3B) + (4 \* TEAM\_BATTING\_HR),  
 BSR\_A = TEAM\_BATTING\_H + TEAM\_BATTING\_BB - TEAM\_BATTING\_HR,  
 BSR\_B = (( 1.4 \* TEAM\_BATTING\_TB) - ( 0.6 \* TEAM\_BATTING\_H) - (3 \* TEAM\_BATTING\_HR) + (0.1 \* TEAM\_BATTING\_BB)) \* 1.02,  
 BSR\_C = TEAM\_BATTING\_AB - TEAM\_BATTING\_H,  
 BSR = ((BSR\_A\*BSR\_B)/(BSR\_B + BSR\_C)) + TEAM\_BATTING\_HR  
 )  
  
data3 <- as.data.frame(data3)  
train3 <- data3 %>% dplyr::sample\_frac(.75)  
test3 <- dplyr::anti\_join(data3, train3, by = 'X')  
  
  
#<B> Create the Regression Model </B>   
#\*BSR\*  
  
rmdata3 <- train3 %>%  
 dplyr::select(BSR, TEAM\_PITCHING\_SO, TEAM\_FIELDING\_E, TEAM\_FIELDING\_DP, TARGET\_WINS)  
  
#Build the second model and produce a summary  
GModel3 <- lm(TARGET\_WINS ~ BSR + TEAM\_PITCHING\_SO + TEAM\_FIELDING\_E + TEAM\_FIELDING\_DP, data = rmdata3)  
summary(GModel3)  
  
  
  
#Predict with the second model training data  
GModel3\_predictions = predict(GModel3,test3)  
  
#Evaluate the second model results using RMSE  
rmse(test3$TARGET\_WINS, GModel3\_predictions)  
  
  
  
  
## Model #4  
### (Modified) Backward Elimination Model (omitting NAs)  
  
#Due to previously learning how to perform Backward Elimination and it being possible to perform manually, we decided to include a model that resulted from the procedure. The process was performed with imputed data (via MICE) as well as data with NAs removed. The latter showed stronger results, therefore the final model was fitted with the NA omitted data.   
  
#According to Faraway, Backward Elimination is when you start with all predictors in the model, then remove the predictor with the highest p-value as long as it is above your p-value threshold (e.g. 0.05). Then refit the model and continue the process until only predictors with p-values below your threshold remain.  
  
#Additionally, we took steps to remove variables with non-intuitive coefficients. For instance, TEAM\_FIELDING\_DP and TEAM\_PITCHING\_SO were unexpectedly showing negative effects on wins. While there could be potential intervening variables giving these variables true predictive power, we opted to remove the variables from the model due to the possibility they were significant by chance and due to our bias towards parsimony. Further, RMSE did not drastically worsen when removed.  
  
  
# Remove NAs  
train\_no\_na <- na.omit(train)   
test\_no\_na <- na.omit(test)   
  
# Fit model  
backward\_model <- lm(TARGET\_WINS ~ TEAM\_BASERUN\_SB + TEAM\_BATTING\_HR + TEAM\_BATTING\_BB + TEAM\_BASERUN\_SB   
 + TEAM\_PITCHING\_SO + TEAM\_FIELDING\_E + TEAM\_FIELDING\_DP, data = test\_no\_na)  
  
# Fit modified model  
backward\_mod\_model <- lm(TARGET\_WINS ~ TEAM\_BASERUN\_SB + TEAM\_BATTING\_HR + TEAM\_BATTING\_BB + TEAM\_FIELDING\_E,   
 data = test\_no\_na)  
  
  
# View summary  
summary(backward\_mod\_model)  
  
  
  
# Make predictions on test set  
backward\_model\_predictions = predict(backward\_mod\_model, test\_no\_na)  
  
# Obtain RMSE between actuals and predicted  
rmse(test\_no\_na$TARGET\_WINS, backward\_model\_predictions)  
  
  
  
# Make predictions on evaluation data  
backward\_model\_predictions\_evaluation = predict(backward\_mod\_model, evaluation)  
  
# Final predictions on evaluation set  
write.csv(backward\_model\_predictions\_evaluation, 'evaluation\_predictions.csv')  
  
  
  
  
## SELECT MODELS  
  
  
### Verifying OLS Regression Assumptions  
  
# Assumption: No Multicollinearity (VIF under 5)  
vif(backward\_mod\_model)  
  
# Assumption: Mean of residuals is zero  
mean(residuals(backward\_mod\_model))  
  
# Assumption: Homoscedasticity of residuals  
plot(backward\_mod\_model)  
  
# Assumption: No auto-correlation  
acf(residuals(backward\_mod\_model), lags=20)

1. <https://statisticsglobe.com/count-number-of-na-values-in-vector-and-column-in-r> [↑](#footnote-ref-1)
2. <https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html> [↑](#footnote-ref-2)
3. <https://datavizpyr.com/visualizing-missing-data-with-barplot-in-r/> [↑](#footnote-ref-3)