Data 621 Assignment 1

Critical Thinking Group2

Gabriella Martinez
Ken Popkin
Matthew Lucich

Maliat Islam
George Cruz Deschamps
Karim Hammoud

Data 621 Assignment 1

Critical Thinking Group2

Table of Contents

Data 621 Assignment 1

Critical Thinking Group2

CUNY The City University of New York | School of Professional Studies

DATA 621 – Business Analytics and Data Mining
Homework #1 Assignment Requirements

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.

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). 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

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.

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 621 Assignment 1

Critical Thinking Group2

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		

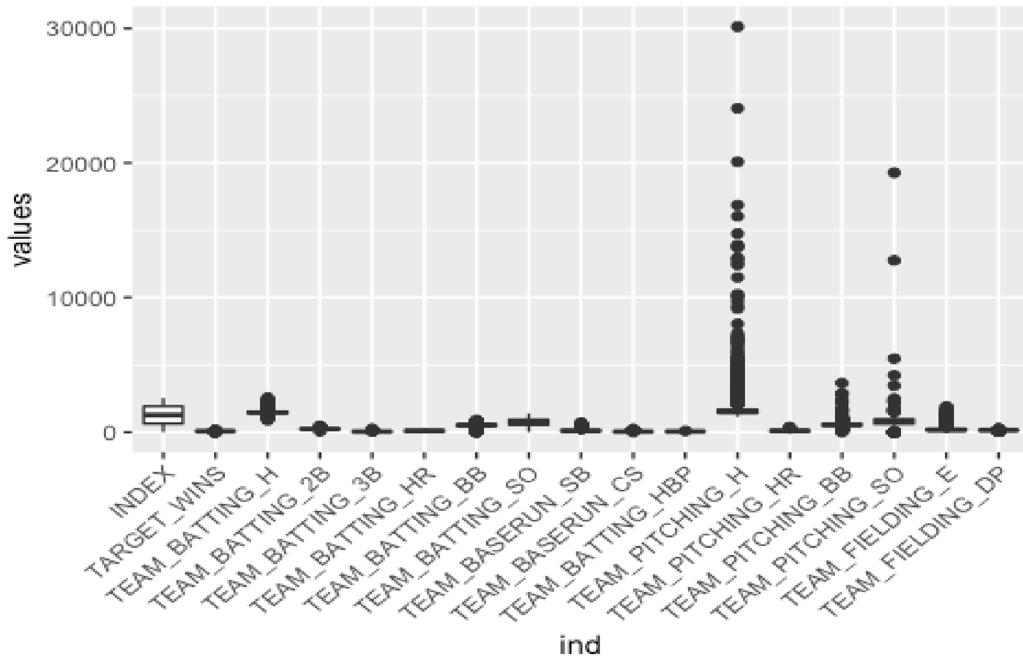
Table 5.2: Summary of the Test Data

Data 621 Assignment 1

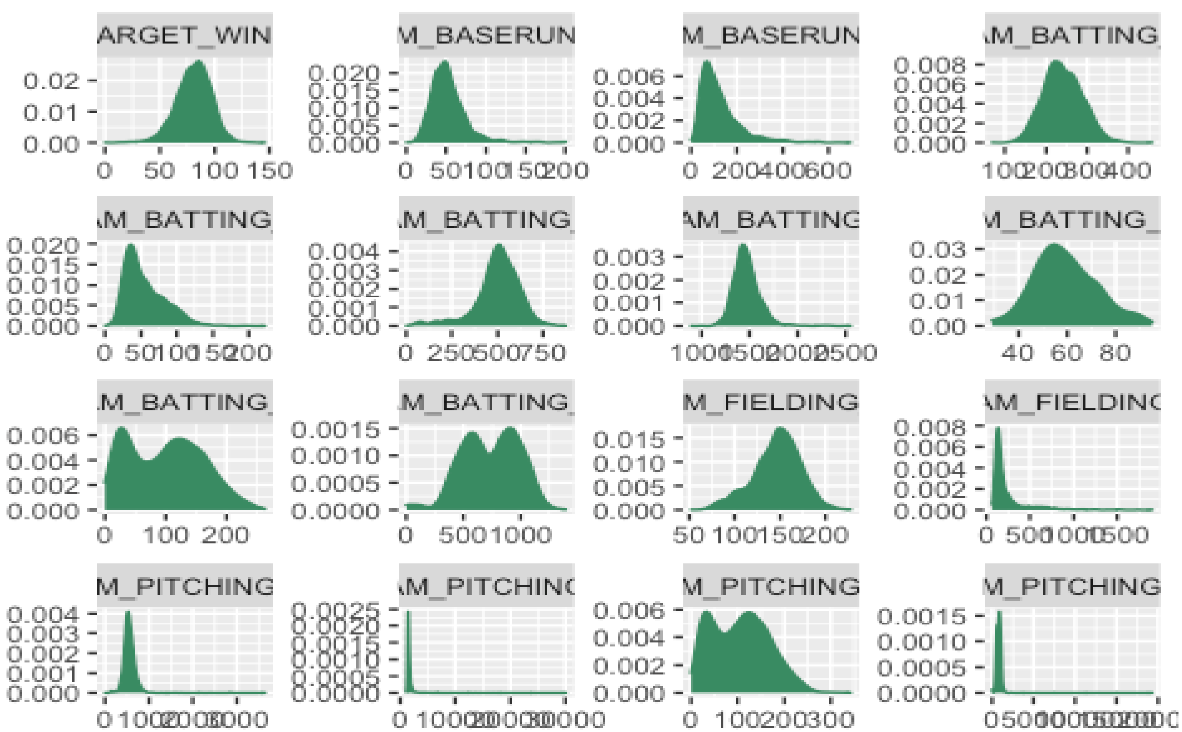
Critical Thinking Group2

A picture is worth a thousand words

Box plot of the Train data



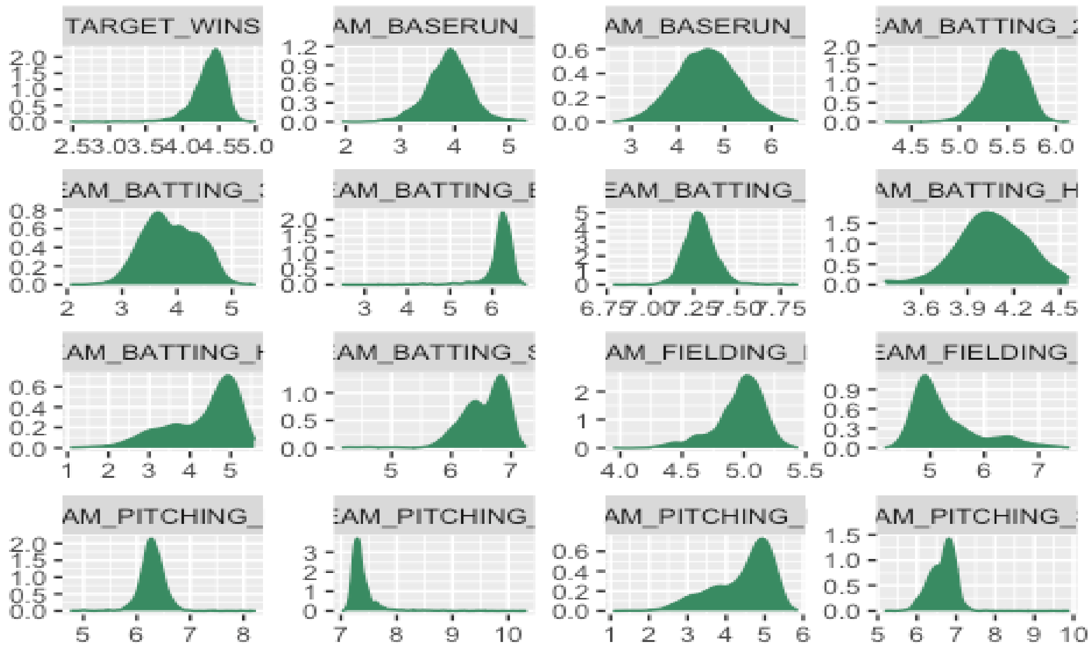
Variable Distributions



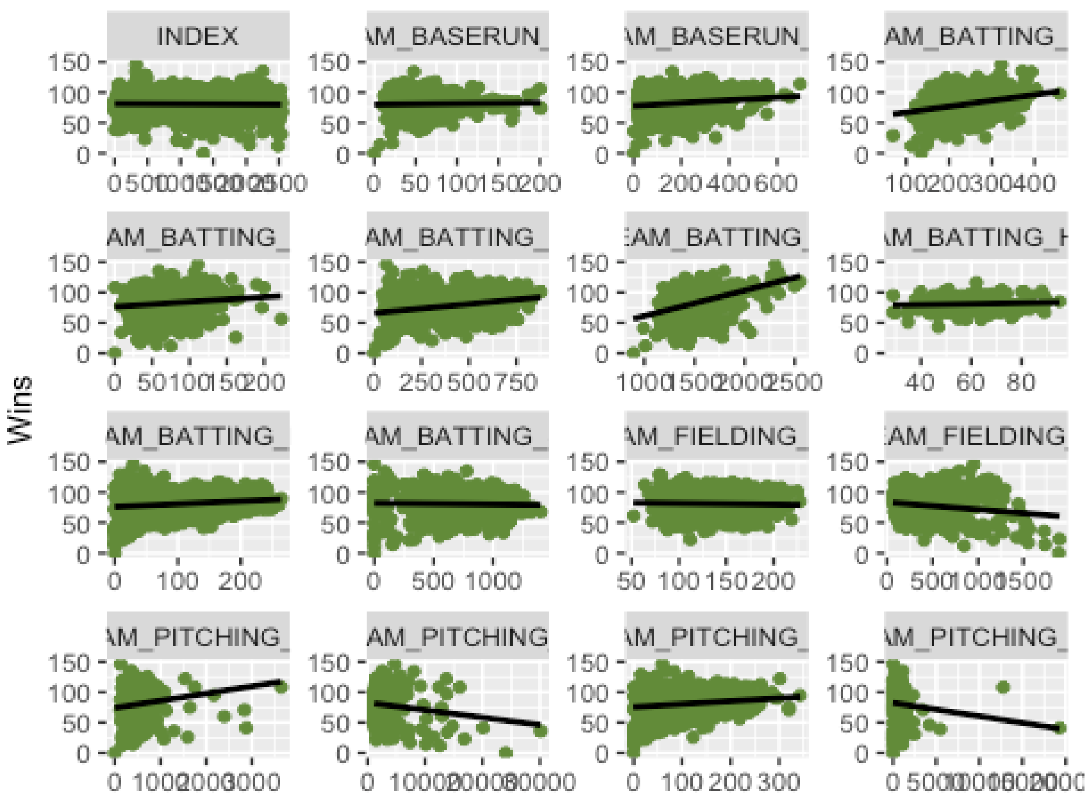
Data 621 Assignment 1

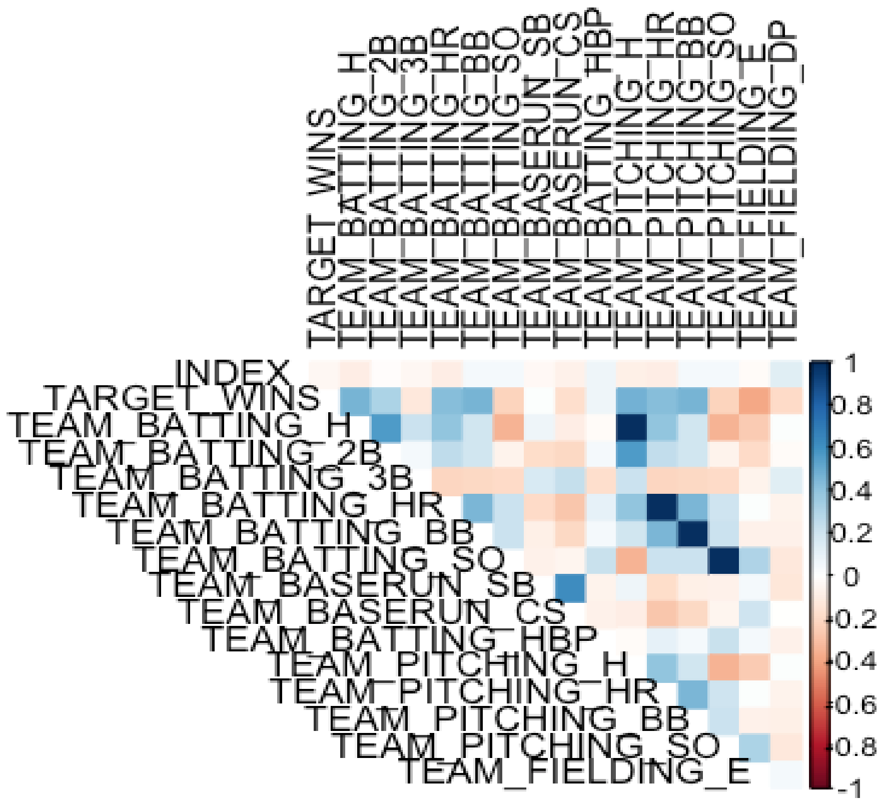
Critical Thinking Group2

Log Variable Distributions



Correlations with Response Variable





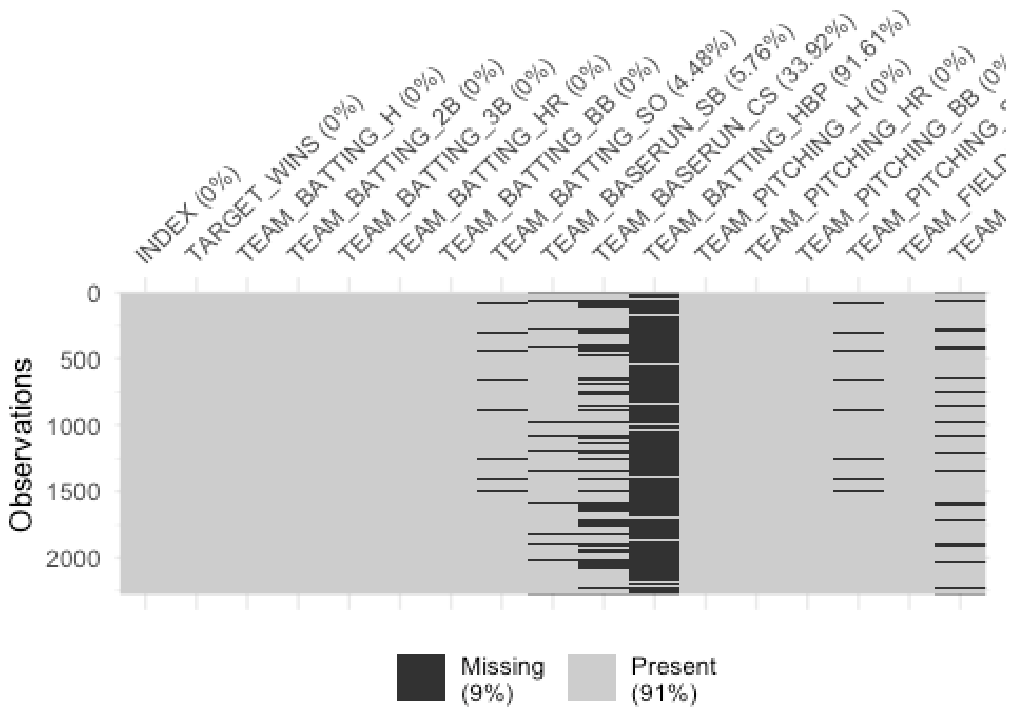
TARGET_WINS
TEAM_BATTING_H
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
TEAM_BATTING_SO
TEAM_BASERUN_SB
TEAM_BASERUN_CS
TEAM_BATTING_HBP
TEAM_PITCHING_H
TEAM_PITCHING_HR
TEAM_PITCHING_BB
TEAM_PITCHING_SO
TEAM_FIELDING_E
TEAM_FIELDING_DP
INDEX
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

DATA PREPARATION

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

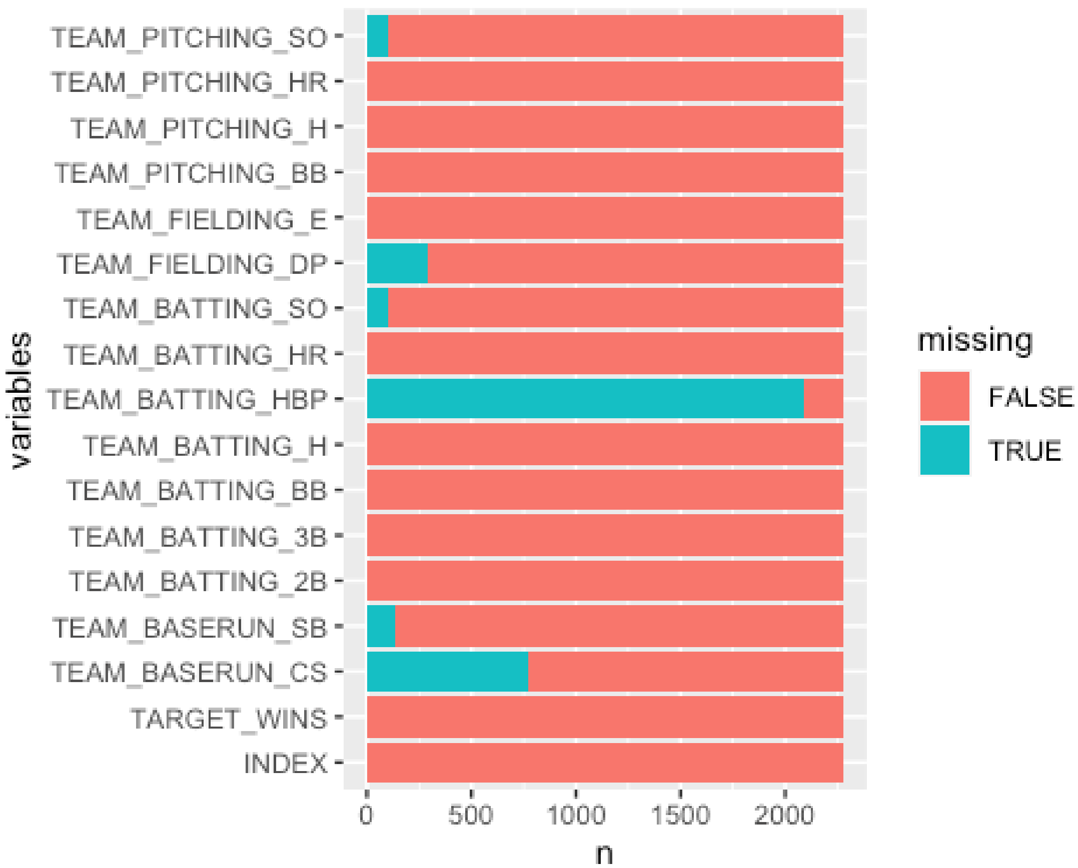
Variable Name	NA's
TEAM_BATTING_SO	102
TEAM_BASERUN_SB	131
TEAM_BASERUN_CS	772
TEAM_BATTING_HBP	2085
TEAM_PITCHING_SO	102.0

Visualization and percentage of NA values [2]



[1] https://statisticsglobe.com/count-number-of-na-values-in-vector-and-column-in-r

[2] https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html



Alternative NA values visualization [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.

[3] https://datavizpyr.com/visualizing-missing-data-with-barplot-in-r/

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		
R^2 / R^2 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		
R^2 / R^2 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:

$A = H + BB - HR$

$B = (1.4 * TB - .6 * H - 3 * HR + .1 * BB) * 1.02$

$C = AB - H\ D = HR$

$$BsR = \frac{(A * B)}{(B + C)} + D$$

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		
R^2 / R^2 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		
R^2 / R^2 adjusted	0.210 / 0.201		

```
RMSE: 11.081
```

Data 621 Assignment 1

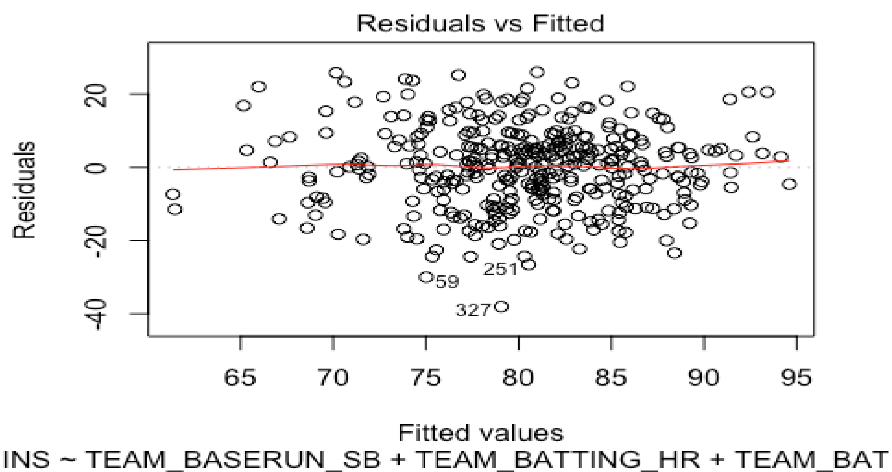
Critical Thinking Group2

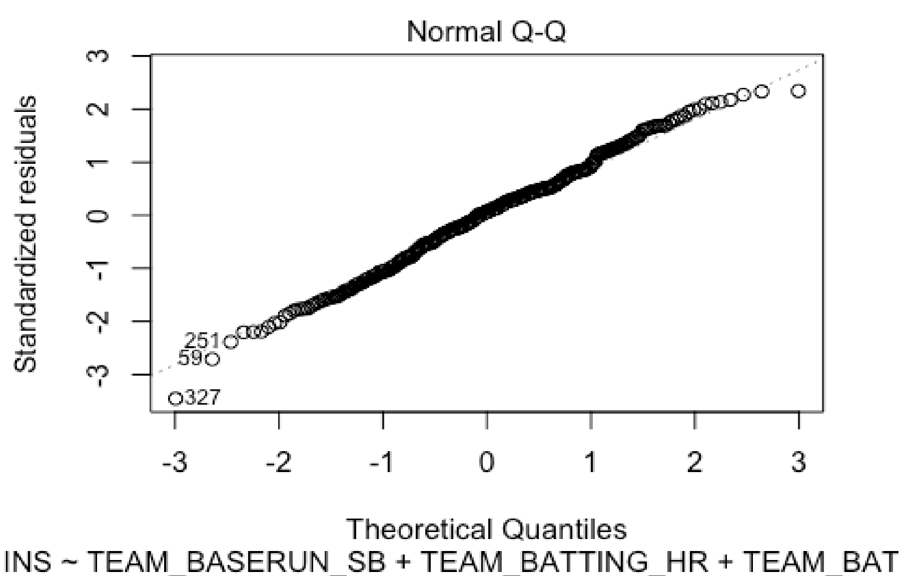
SELECT MODELS

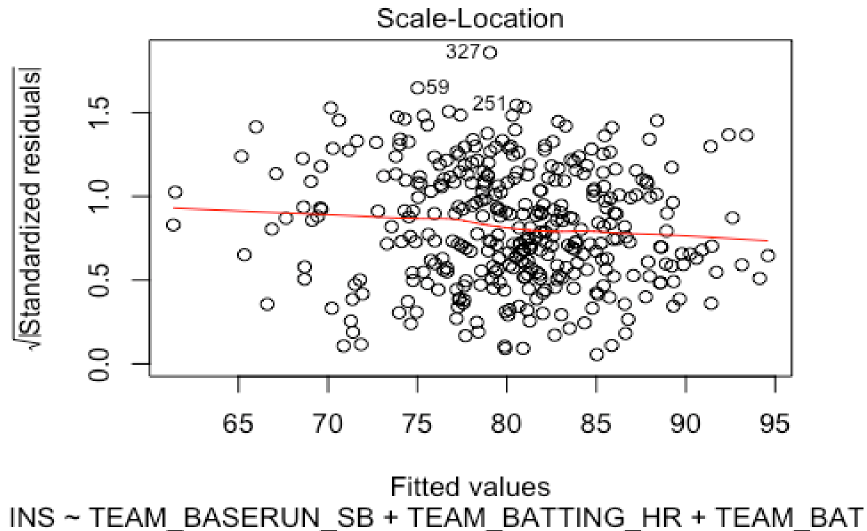
Verifying OLS Regression Assumptions

Assumption: Mean of residuals is zero

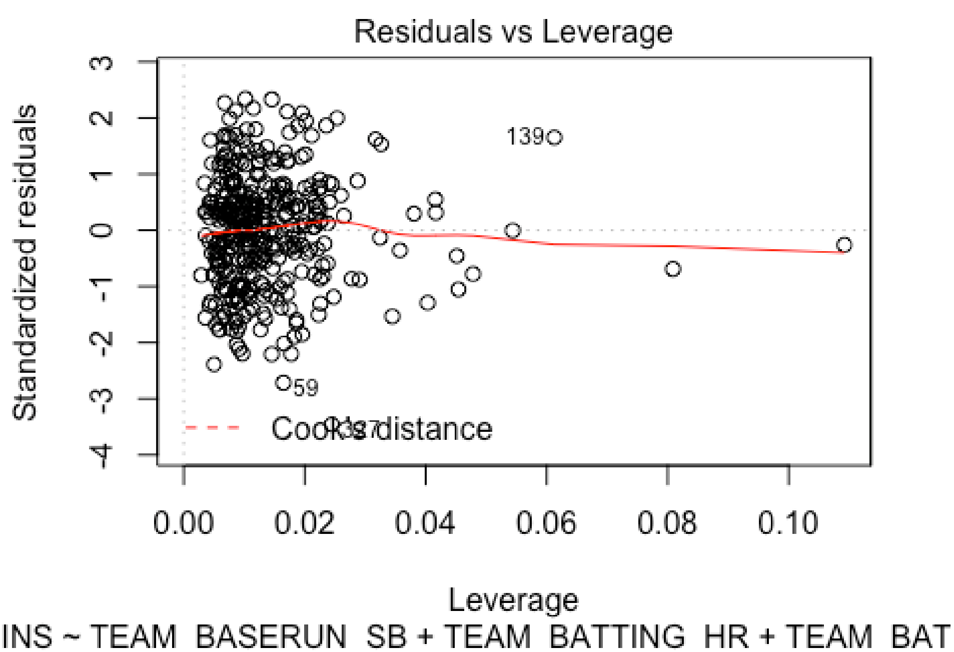
```
## [1] 2.396206e-17
```



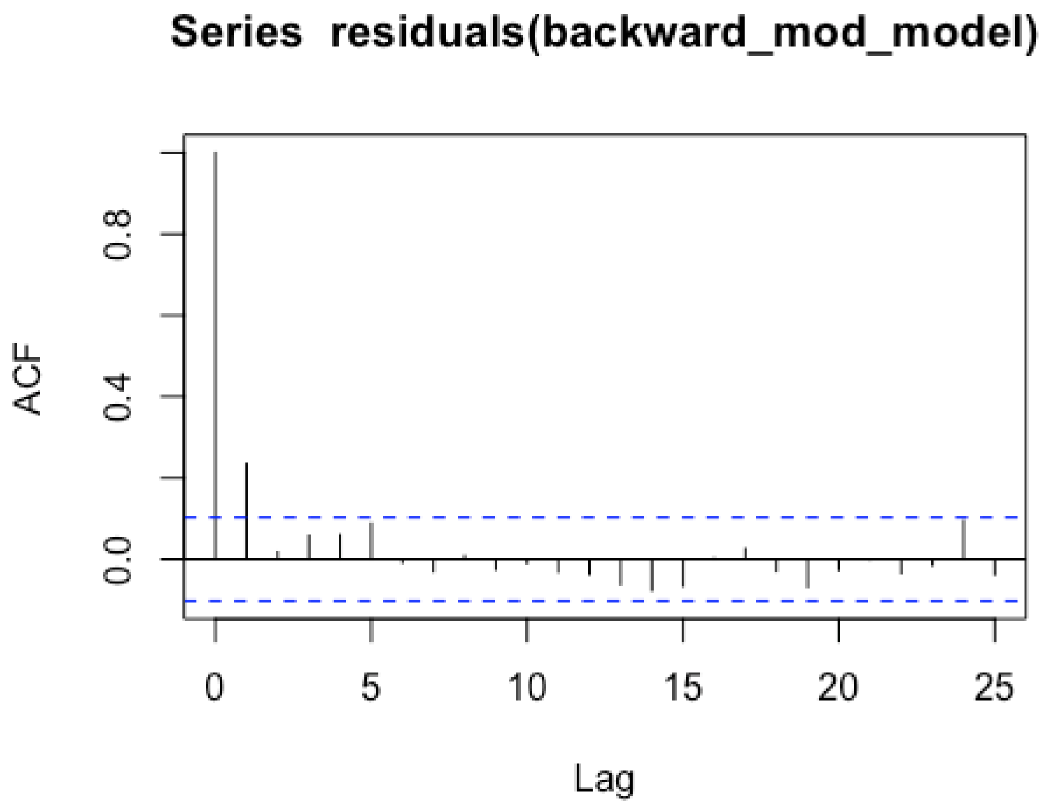




Scale-Location
√|Standardized residuals|
Fitted values
INS ~ TEAM_BASERUN_SB + TEAM_BATTING_HR + TEAM_BAT



Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
INS ~ TEAM_BASERUN_SB + TEAM_BATTING_HR + TEAM_BAT



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.

Data 621 Assignment 1

Critical Thinking Group2

Appendix A: Code

```
## DATA EXPLORATION:
#We can observe the response variable (TARGET_WINS) looks to be normally dist
ributed. This supports the working theory that there are good teams and bad t
eams. 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 residual
s 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/ma
in/Assignment_1/data/moneyball-training-data.csv")

evaluation <-read.csv("https://raw.githubusercontent.com/akarimhammoud/Data_6
21/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 = FALS
E)
```

```r
### DATA PREPARATION

# ^[https://statisticsglobe.com/count-number-of-na-values-in-vector-and-colum
n-in-r]

#NA counts for the train data set
colSums(is.na(train))

# ^[https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisa
tion.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 w
ith 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 a
nd the supporting reason/s.

#TEAM_BATTING_H = Base hits by batters:  it's impossible to win in baseball w
ithout getting to the bases # and hitting the ball is the primary means to ac
complish 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 defens
e strategy.

#Only two features are selected for the first model - start small and build u
p 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 = trai
n)
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 s
imple 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 al
lowed
#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 t
wo more features...
#TEAM_FIELDING_E = Errors: errors are costly in terms of immediate impact, bu
t could also impact the team in other ways (i.e. a high occurrence could impa
ct 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_FIEL
DING_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 se
cond model did not yield a noticeable improvement.  The Adjusted R2 on the tr
aining 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 po
or predictive capability.

## Model #3
### BSR Model (SaberMetrics) (data imputation)
# *Base runs (BsR) is a baseball statistic invented by sabermetrician David S
myth 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 s
abermetrician Tom M. Tango points out, base*
#*runs models the reality of the run-scoring process "significantly better th
an 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_S
O, na.rm=TRUE),
         TEAM_BATTING_1B = TEAM_BATTING_H - (TEAM_BATTING_2B + TEAM_BATTING_3
B + TEAM_BATTING_HR),
         TEAM_BATTING_TB = TEAM_BATTING_1B + (2 * TEAM_BATTING_2B) + (3 * TEA
M_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 * T
EAM_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, TAR
GET_WINS)

#Build the second model and produce a summary
GModel3 <- lm(TARGET_WINS ~ BSR + TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_F
IELDING_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 fro
m the procedure. The process was performed with imputed data (via MICE) as we
ll as data with NAs removed. The latter showed stronger results, therefore th
e final model was fitted with the NA omitted data.

#According to Faraway, Backward Elimination is when you start with all predic
tors 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 c
ontinue the process until only predictors with p-values below your threshold 
remain.

#Additionally, we took steps to remove variables with non-intuitive coefficie
nts. For instance, TEAM_FIELDING_DP and TEAM_PITCHING_SO were unexpectedly sh
owing negative effects on wins. While there could be potential intervening va
riables giving these variables true predictive power, we opted to remove the 
variables from the model due to the possibility they were significant by chan
ce and due to our bias towards parsimony. Further, RMSE did not drastically w
orsen 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_B
ATTING_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 + TE
AM_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, evaluatio
n)

# 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)
```

```r
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)
```