Code

- Show All Code
- Hide All Code
-
- Download Rmd

Data 621 - HW1

MoneyBall Predictor - Predicting The Number Of Wins

Data 621 - HW1

JV-KN-MAR-TS

2023-10-08

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

## Warning: package 'skimr' was built under R version 4.3.1

## corrplot 0.92 loaded

## Warning: package 'kableExtra' was built under R version 4.3.1
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
##
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

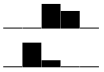
Data Exploration

```
## The MoneyBall training dataset contain 2276 rows and 17 columns as shown below :

## [1] 2276   17

## The following is a brief summary of the first 6 rows are :

## The datatypes of the rows are NUMERIC as is shown below

##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##        "numeric"        "numeric"        "numeric"        "numeric"
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##        "numeric"        "numeric"        "numeric"        "numeric"
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##        "numeric"        "numeric"        "numeric"        "numeric"
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##        "numeric"        "numeric"        "numeric"        "numeric"
## TEAM_FIELDING_DP
##        "numeric"
```

```
## Exploring the Summary Statistics of the Trainig Dataset
```

Data summary	
Name	train_dt
Number of rows	2276
Number of columns	17
Column type frequency:	
numeric	17
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INDEX	0	1.00	1268.46	736.35	1	630.75	1270.5	1915.50	2535	
TARGET_WINS	0	1.00	80.79	15.75	0	71.00	82.0	92.00	146	
TEAM_BATTING_H	0	1.00	1469.27	144.59	891	1383.00	1454.0	1537.25	2554	
TEAM_BATTING_2B	0	1.00	241.25	46.80	69	208.00	238.0	273.00	458	
TEAM_BATTING_3B	0	1.00	55.25	27.94	0	34.00	47.0	72.00	223	
TEAM_BATTING_HR	0	1.00	99.61	60.55	0	42.00	102.0	147.00	264	
TEAM_BATTING_BB	0	1.00	501.56	122.67	0	451.00	512.0	580.00	878	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
TEAM_BATTING_SO	102	0.96	735.61	248.53	0	548.00	750.0	930.00	1399	▁▆▇▆▁
TEAM_BASERUN_SB	131	0.94	124.76	87.79	0	66.00	101.0	156.00	697	▇▃▁▁▁
TEAM_BASERUN_CS	772	0.66	52.80	22.96	0	38.00	49.0	62.00	201	▂▇▁▁▁
TEAM_BATTING_HBP	2085	0.08	59.36	12.97	29	50.50	58.0	67.00	95	▂▇▇▇▁
TEAM_PITCHING_H	0	1.00	1779.21	1406.84	1137	1419.00	1518.0	1682.50	30132	▇▁▁▁▁
TEAM_PITCHING_HR	0	1.00	105.70	61.30	0	50.00	107.0	150.00	343	▇▇▇▂▁
TEAM_PITCHING_BB	0	1.00	553.01	166.36	0	476.00	536.5	611.00	3645	▇▁▁▁▁
TEAM_PITCHING_SO	102	0.96	817.73	553.09	0	615.00	813.5	968.00	19278	▇▁▁▁▁
TEAM_FIELDING_E	0	1.00	246.48	227.77	65	127.00	159.0	249.25	1898	▇▁▁▁▁
TEAM_FIELDING_DP	286	0.87	146.39	26.23	52	131.00	149.0	164.00	228	▁▃▇▆▁

Density plots of variables

Box Plots of variables

Scatter Plots

Correlation Plot and Matrix

Variable	Variable	Correlation
TARGET_WINS	TEAM_PITCHING_H	0.4712343
TARGET_WINS	TEAM_BATTING_H	0.4699467
TARGET_WINS	TEAM_BATTING_BB	0.4686879
TARGET_WINS	TEAM_PITCHING_BB	0.4683988
TARGET_WINS	TEAM_PITCHING_HR	0.4224668
TARGET_WINS	TEAM_BATTING_HR	0.4224168
TARGET_WINS	TEAM_FIELDING_E	-0.3866880
TARGET_WINS	TEAM_BATTING_2B	0.3129840
TARGET_WINS	TEAM_PITCHING_SO	-0.2293648
TARGET_WINS	TEAM_BATTING_SO	-0.2288927
TARGET_WINS	TEAM_FIELDING_DP	-0.1958660
TARGET_WINS	TEAM_BASERUN_CS	-0.1787560
TARGET_WINS	TEAM_BATTING_3B	-0.1243459
TARGET_WINS	TEAM_BATTING_HBP	0.0735042
TARGET_WINS	TEAM_BASERUN_SB	0.0148364

Analysis of Data Exploration based on above Computations :

TEAM_BATTING_HBP variable has 2,085 missing values (91.7%)

TEAM_BASERUN_CS has 772 missing values (33.9%)

TEAM_BASERUN_SB variable has 131 missing variable (5.7%)

TEAM_BATTING_SO variable has 102 missing values (4.5%)

TEAM_PITCHING_SO variable has 102 missing values (4.5%)

TEAM_FIELDING_DP variable has 286 missing values(12.5%)

TEAM_PITCHING_H has the highest mean value of 1779.21 among the 17 variables

TEAM_BASERUN_CS has the lowest mean

TEAM_PITCHING_H also has the highest median value of 1518 among the 17 variables

TEAM_BATTING_3B has the lowest median

TARGET_WINS has half of its values between 71 (25th percentile) and 92 (75th percentile)

TEAM_PITCHING_BB, TEAM_PITCHING_H and TEAM_PITCHING_SO has the largest number of outliers

TARGET_WINS has the highest positive correlation of 0.47
with TEAM_BATTING_H, TEAM_BATTING_BB and TEAM_PITCHING_H

TARGET_WINS has the highest negative correlation of 0.39 with TEAM_FIELDING_E

Data Preparation

Fixing the dataset deficiencies to account for missing data and outliers

---

Managing Missing Values

TEAM_BATTING_HBP has 91.7% of its' values missing, we will replace those with the
MLB 2018 and 2019 averages of 65

TEAM_BASERUN_CS has 33.96% of its' values missing, we will replace those with the
MLB 2018 and 2019 averages of 30

TEAM_BASERUN_SB has 5.7% of its' values missing, we will replace those with the
MEAN of the existing values

TEAM_BATTING_SO has 4.5% of its' values missing, we will replace those with the
MEAN of the existing values

TEAM_PITCHING_SO has 4.5% of its' values missing, we will replace those with the
MEAN of the existing values

TEAM_FIELDING_DP has 12.5% of its' values missing, we will replace those with the
MEAN of the existing values

Managing Outliers

From the Data Exploration we see that the following varaibles contain Outlier Values :
TEAM_PITCHING_SO
TEAM_PITCHING_BB
TEAM_PITCHING_H

We will account for these Outliers by applying defining outliers as
values that are more than 1.5 times the interquartile range (IQR)
below the first quartile (Q1) or above the third quartile (Q3)
By performing the following computations on the dataset

```
## Calculating first quartile (Q1) and third quartile (Q3) for each variable

## Calculating IQR for each variable

## Setting lower and upper bounds for outliers

## Removing outliers from the dataframe
```

Adding New Variables

```
## We will add to computed varaibles to enhance the model selection
##  BATTING AVERAGE = TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS)

## AND
##  RUN DIFFERENTIAL = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR
```

Resultant Enhanced Dataset Overview

```
## Exploring the Summary Statistics of the Enhanced Trainig Dataset
```

Data summary	
Name	train_dt_prep1
Number of rows	2027
Number of columns	19
Column type frequency:	
numeric	19
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INDEX	0	1	1272.09	730.61	2.00	648.50	1277.00	1904.50	2534.0	▇▇▇▇▇
TARGET_WINS	0	1	80.58	14.15	21.00	71.00	81.00	91.00	124.0	▁▃▇▃▁
TEAM_BATTING_H	0	1	1451.81	112.36	1137.00	1377.00	1445.00	1523.00	1876.0	▁▇▇▂▁
TEAM_BATTING_2B	0	1	240.80	45.34	118.00	208.00	238.00	272.00	392.0	▁▆▇▃▁
TEAM_BATTING_3B	0	1	52.23	24.62	11.00	33.00	45.00	67.00	147.0	▇▅▃▂▁
TEAM_BATTING_HR	0	1	105.32	58.86	3.00	52.00	109.00	149.00	264.0	▇▅▇▇▂
TEAM_BATTING_BB	0	1	518.79	89.12	189.00	461.00	517.00	581.00	775.0	▁▃▇▇▁
TEAM_BATTING_SO	0	1	765.46	216.90	268.00	589.00	760.00	933.00	1399.0	▂▅▇▇▃
TEAM_BASERUN_SB	0	1	119.27	79.53	18.00	65.00	100.00	149.00	654.0	▇▃▁▁▁
TEAM_BASERUN_CS	0	1	46.19	21.99	11.00	30.00	40.00	56.00	201.0	▇▅▁▁▁
TEAM_BATTING_HBP	0	1	64.47	4.30	29.00	65.00	65.00	65.00	95.0	▁▁▇▁▁
TEAM_PITCHING_H	0	1	1525.48	163.70	1137.00	1408.50	1495.00	1611.00	2073.0	▁▇▆▂▁
TEAM_PITCHING_HR	0	1	108.37	58.78	3.00	57.00	111.00	152.00	264.0	▇▅▇▇▂
TEAM_PITCHING_BB	0	1	543.00	91.44	284.00	482.00	536.00	601.00	810.0	▁▅▇▃▁
TEAM_PITCHING_SO	0	1	797.87	207.57	301.00	637.00	817.73	948.00	1434.0	▂▅▇▅▂
TEAM_FIELDING_E	0	1	194.79	118.64	65.00	125.00	153.00	215.00	791.0	▇▃▁▁▁
TEAM_FIELDING_DP	0	1	147.30	24.25	68.00	134.00	147.00	163.00	225.0	▁▃▇▃▁
BATT_AVG	0	1	0.25	0.01	0.21	0.24	0.25	0.26	0.3	▁▅▇▃▁
RUN_DIFF	0	1	-76.72	113.19	-767.00	-92.00	-70.00	0.00	29.0	▁▁▁▁▇

Box Plots of Enhanced Dataset :

Evaluation Dataset

Data Preparation of Eval Dataset

```
## The MoneyBall Evaluation dataset contain 256 rows and 16 columns as shown below :

## [1] 259  16

## The following is a brief summary of the first 6 rows are :

## The datatypes of the rows are NUMERIC as is shown below

##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##        "numeric"        "numeric"        "numeric"        "numeric"
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##        "numeric"        "numeric"        "numeric"        "numeric"
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##        "numeric"        "numeric"        "numeric"        "numeric"
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        "numeric"        "numeric"        "numeric"        "numeric"

## Exploring the Summary Statistics of the Eval Dataset
```

Data summary

Name	eval_dt
Number of rows	259
Number of columns	16
Column type frequency:	
numeric	16
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INDEX	0	1.00	1263.93	693.29	9	708.0	1249.0	1832.50	2525	▇▇▇▇▇
TEAM_BATTING_H	0	1.00	1469.39	150.66	819	1387.0	1455.0	1548.00	2170	▁▂▇▁▁
TEAM_BATTING_2B	0	1.00	241.32	49.52	44	210.0	239.0	278.50	376	▁▃▇▇▂
TEAM_BATTING_3B	0	1.00	55.91	27.14	14	35.0	52.0	72.00	155	▇▆▂▁▁
TEAM_BATTING_HR	0	1.00	95.63	56.33	0	44.5	101.0	135.50	242	▇▅▇▃▂
TEAM_BATTING_BB	0	1.00	498.96	120.59	15	436.5	509.0	565.50	792	▁▂▅▇▂
TEAM_BATTING_SO	18	0.93	709.34	243.11	0	545.0	686.0	912.00	1268	▁▃▇▇▂
TEAM_BASERUN_SB	13	0.95	123.70	93.39	0	59.0	92.0	151.75	580	▇▃▂▁▁
TEAM_BASERUN_CS	87	0.66	52.32	23.10	0	38.0	49.5	63.00	154	▂▅▇▃▁
TEAM_BATTING_HBP	240	0.07	62.37	12.71	42	53.5	62.0	67.50	96	▂▅▇▂▂
TEAM_PITCHING_H	0	1.00	1813.46	1662.91	1155	1426.5	1515.0	1681.00	22768	▇▁▁▁▁
TEAM_PITCHING_HR	0	1.00	102.15	57.65	0	52.0	104.0	142.50	336	▇▇▇▂▁
TEAM_PITCHING_BB	0	1.00	552.42	172.95	136	471.0	526.0	606.50	2008	▂▇▂▁▁
TEAM_PITCHING_SO	18	0.93	799.67	634.31	0	613.0	745.0	938.00	9963	▇▁▁▁▁

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
TEAM_FIELDING_E	0	1.00	249.75	230.90	73	131.0	163.0	252.00	1568	▇▁▁▁▁
TEAM_FIELDING_DP	31	0.88	146.06	25.88	69	131.0	148.0	164.00	204	▁▃▇▇▂

```
## We will replace the missing values of TEAM_BATTING_HBP the
##  MLB 2018 and 2019 averages of 65

## We will replace the missing values of TEAM_BASERUN_CS
##  MLB 2018 and 2019 averages of 30

## We will replace the missing values of TEAM_BASERUN_SB with the
##  MEAN of the existing values

## We will replace the missing values of TEAM_BATTING_SO with the
##  MEAN of the existing values

## We will replace the missing values of TEAM_PITCHING_SO with the
##  MEAN of the existing values

## We will replace the missing values of TEAM_FIELDING_DP with the
##  MEAN of the existing values

## We will add to computed varaibles to enhance the model selection
##  BATTING AVERAGE = TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS)

## AND
##  RUN DIFFERENTIAL = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR

## Exploring the Summary Statistics of the Revised Eval Dataset
```

Data summary

Name	eval_dt
Number of rows	259
Number of columns	18
Column type frequency:	
numeric	18
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INDEX	0	1	1263.93	693.29	9.00	708.00	1249.00	1832.50	2525.00	▇▇▇▇▇
TEAM_BATTING_H	0	1	1469.39	150.66	819.00	1387.00	1455.00	1548.00	2170.00	▁▂▇▁▁
TEAM_BATTING_2B	0	1	241.32	49.52	44.00	210.00	239.00	278.50	376.00	▁▃▇▅▁
TEAM_BATTING_3B	0	1	55.91	27.14	14.00	35.00	52.00	72.00	155.00	▇▇▅▂▁
TEAM_BATTING_HR	0	1	95.63	56.33	0.00	44.50	101.00	135.50	242.00	▇▃▇▅▂
TEAM_BATTING_BB	0	1	498.96	120.59	15.00	436.50	509.00	565.50	792.00	▁▁▃▇▁
TEAM_BATTING_SO	0	1	709.34	234.48	0.00	565.00	709.34	904.50	1268.00	▁▃▇▅▂
TEAM_BASERUN_SB	0	1	123.70	91.00	0.00	60.50	96.00	149.00	580.00	▇▇▂▁▁
TEAM_BASERUN_CS	0	1	44.82	21.57	0.00	30.00	38.00	56.00	154.00	▇▇▇▃▁
TEAM_BATTING_HBP	0	1	64.81	3.43	42.00	65.00	65.00	65.00	96.00	▁▁▇▁▁
TEAM_PITCHING_H	0	1	1813.46	1662.91	1155.00	1426.50	1515.00	1681.00	22768.00	▇▁▁▁▁
TEAM_PITCHING_HR	0	1	102.15	57.65	0.00	52.00	104.00	142.50	336.00	▇▇▇▃▁
TEAM_PITCHING_BB	0	1	552.42	172.95	136.00	471.00	526.00	606.50	2008.00	▇▇▃▁▁
TEAM_PITCHING_SO	0	1	799.67	611.78	0.00	622.50	782.00	927.50	9963.00	▇▁▁▁▁

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
TEAM_FIELDING_E	0	1	249.75	230.90	73.00	131.00	163.00	252.00	1568.00	▇▁▁▁▁
TEAM_FIELDING_DP	0	1	146.06	24.28	69.00	134.50	146.06	160.50	204.00	▁▃▇▇▁
BATT_AVG	0	1	0.25	0.02	0.16	0.24	0.25	0.26	0.34	▁▁▇▃▁
RUN_DIFF	0	1	-350.59	1645.93	-21222.00	-104.50	-78.00	0.00	32.00	▁▁▁▁▁▇

```
## The Eval dataset is now ready to accept predictions from the Training dataset
```

Models

All Models Utilize the Enhanced Training Dataset

Model 1

```
## Exploring a model using ONLY Predictor Variables which have a THEORETICAL NEGATIVE Impact on Wins

##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_SO + TEAM_BASERUN_CS +
##     TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR,
##     data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.840  -8.707   0.397   8.861  46.275
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      36.218467   4.674908   7.747 1.47e-14 ***
## TEAM_BATTING_SO  -0.003553   0.002270  -1.565    0.118
## TEAM_BASERUN_CS  -0.001633   0.013531  -0.121    0.904
## TEAM_FIELDING_E  -0.037445   0.003702 -10.114  < 2e-16 ***
## TEAM_PITCHING_BB  0.028007   0.003388   8.268 2.45e-16 ***
## TEAM_PITCHING_H   0.024952   0.002592   9.625  < 2e-16 ***
## TEAM_PITCHING_HR  0.010911   0.008892   1.227    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.86 on 2020 degrees of freedom
## Multiple R-squared:  0.1757, Adjusted R-squared:  0.1733
## F-statistic: 71.77 on 6 and 2020 DF,  p-value: < 2.2e-16

## We see that the p-values of the 6 predictor variables have significant impact on the Target Wins
##  Also, the R-Squared indicates that this model explains 17.6% of the variability in Target Wins
```

Model 2

```
## Exploring a model using ONLY Predictor Variables which have a THEORETICAL POSITIVE Impact on Wins

##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_HBP +
##     TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_PITCHING_SO, data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.469  -7.676   0.274   8.202  53.162
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.534150   7.466009   2.750 0.006006 **
## TEAM_BATTING_H    0.029106   0.004652   6.256 4.80e-10 ***
## TEAM_BATTING_2B  -0.014292   0.009722  -1.470 0.141692
```

```
## TEAM_BATTING_3B   0.093901   0.019032   4.934 8.72e-07 ***
## TEAM_BATTING_HR   0.102029   0.009611  10.616  < 2e-16 ***
## TEAM_BATTING_BB   0.038940   0.003506  11.107  < 2e-16 ***
## TEAM_BATTING_HBP  0.116268   0.063667   1.826 0.067969 .
## TEAM_BASERUN_SB   0.027295   0.004367   6.250 5.00e-10 ***
## TEAM_FIELDING_DP -0.125141   0.012888  -9.710  < 2e-16 ***
## TEAM_PITCHING_SO -0.008692   0.002317  -3.751 0.000181 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.1 on 2017 degrees of freedom
## Multiple R-squared:  0.2723, Adjusted R-squared:  0.269
## F-statistic: 83.85 on 9 and 2017 DF,  p-value: < 2.2e-16

## We see that the p-values of the 9 predictor variables have significant impact on the Target Wins
##  Also, the R-Squared indicates that this model explains 27.2% of the variability in Target Wins
```

Model 3

```
## Exploring a model using 11 Impactful Predictor Variables on Wins

##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB +
##     TEAM_BATTING_HBP + TEAM_PITCHING_SO + TEAM_BATTING_SO + TEAM_FIELDING_DP +
##     TEAM_FIELDING_E, data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.210  -7.712   0.310   7.432  64.161
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.860816   7.122401   4.754 2.13e-06 ***
## TEAM_BATTING_H    0.029109   0.004384   6.640 4.02e-11 ***
## TEAM_BATTING_2B  -0.033821   0.009233  -3.663 0.000256 ***
## TEAM_BATTING_3B   0.163483   0.018703   8.741  < 2e-16 ***
## TEAM_BATTING_HR   0.093971   0.009346  10.054  < 2e-16 ***
## TEAM_BATTING_BB   0.031098   0.003321   9.364  < 2e-16 ***
## TEAM_BASERUN_SB   0.067572   0.004736  14.267  < 2e-16 ***
## TEAM_BATTING_HBP  0.116939   0.059665   1.960 0.050144 .
## TEAM_PITCHING_SO  0.038614   0.006470   5.968 2.83e-09 ***
## TEAM_BATTING_SO  -0.051553   0.006939  -7.429 1.61e-13 ***
## TEAM_FIELDING_DP -0.103591   0.012178  -8.506  < 2e-16 ***
## TEAM_FIELDING_E  -0.070070   0.004165 -16.822  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.33 on 2015 degrees of freedom
## Multiple R-squared:  0.3623, Adjusted R-squared:  0.3588
## F-statistic: 104.1 on 11 and 2015 DF,  p-value: < 2.2e-16

## We see that the p-values of the 11 predictor variables have significant impact on the Target Wins
##  Also, the R-Squared indicates that this model explains 36.2% of the variability in Target Wins
```

Model 4

```
## Exploring a model using 13 Impactful Predictor Variables on Wins

##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB +
##     TEAM_BATTING_HBP + TEAM_PITCHING_SO + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H + TEAM_BATTING_SO + TEAM_FIELDING_DP + TEAM_FIELDING_E,
##     data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.100  -7.429   0.044   7.400  65.873
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.113940   7.044918   4.700 2.77e-06 ***
## TEAM_BATTING_H   -0.038499   0.013545  -2.842 0.004523 **
## TEAM_BATTING_2B  -0.034809   0.009109  -3.821 0.000137 ***
## TEAM_BATTING_3B   0.165058   0.018445   8.949  < 2e-16 ***
## TEAM_BATTING_HR   0.091887   0.009275   9.906  < 2e-16 ***
## TEAM_BATTING_BB   0.306389   0.036027   8.504  < 2e-16 ***
## TEAM_BASERUN_SB   0.076790   0.004828  15.905  < 2e-16 ***
## TEAM_BATTING_HBP  0.111535   0.058843   1.895 0.058176 .
## TEAM_PITCHING_SO  0.072014   0.014512   4.962 7.55e-07 ***
## TEAM_PITCHING_BB -0.258084   0.033631  -7.674 2.58e-14 ***
## TEAM_PITCHING_H   0.064690   0.012138   5.330 1.09e-07 ***
## TEAM_BATTING_SO  -0.089180   0.015501  -5.753 1.01e-08 ***
## TEAM_FIELDING_DP -0.099023   0.012055  -8.214 3.77e-16 ***
## TEAM_FIELDING_E  -0.077030   0.004447 -17.323  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.17 on 2013 degrees of freedom
## Multiple R-squared:  0.3806, Adjusted R-squared:  0.3766
## F-statistic: 95.17 on 13 and 2013 DF,  p-value: < 2.2e-16

## We see that the p-values of the 13 predictor variables have significant impact on the Target Wins
##  Also the R-Squared indicates that this model explains 38.2% of the variability in Target Wins
```

Model 5

This Model Examines the entire Training Dataset

Examining Model Fit and Effects of MultiCollinearity

```
## Exploring a model using ALL Predictor Variables on Wins

## Checking the structure of the dataset

## Dropping the 'INDEX' variable

## tibble [2,027 × 17] (S3: tbl_df/tbl/data.frame)
##  $ TEAM_BATTING_H  : num [1:2027] 1339 1377 1387 1297 1279 ...
##  $ TEAM_BATTING_2B : num [1:2027] 219 232 209 186 200 179 171 197 213 179 ...
##  $ TEAM_BATTING_3B : num [1:2027] 22 35 38 27 36 54 37 40 18 27 ...
##  $ TEAM_BATTING_HR : num [1:2027] 190 137 96 102 92 122 115 114 96 82 ...
##  $ TEAM_BATTING_BB : num [1:2027] 685 602 451 472 443 525 456 447 441 374 ...
##  $ TEAM_BATTING_SO : num [1:2027] 1075 917 922 920 973 ...
##  $ TEAM_BASERUN_SB : num [1:2027] 37 46 43 49 107 80 40 69 72 60 ...
##  $ TEAM_BASERUN_CS : num [1:2027] 28 27 30 39 59 54 36 27 34 39 ...
##  $ TEAM_BATTING_HBP: num [1:2027] 65 65 65 65 65 65 65 65 65 65 ...
##  $ TEAM_PITCHING_H : num [1:2027] 1347 1377 1396 1297 1279 ...
##  $ TEAM_PITCHING_HR: num [1:2027] 191 137 97 102 92 122 116 114 96 86 ...
##  $ TEAM_PITCHING_BB: num [1:2027] 689 602 454 472 443 525 459 447 441 391 ...
##  $ TEAM_PITCHING_SO: num [1:2027] 1082 917 928 920 973 ...
##  $ TEAM_FIELDING_E : num [1:2027] 193 175 164 138 123 136 112 127 131 119 ...
##  $ TEAM_FIELDING_DP: num [1:2027] 155 153 156 168 149 186 136 169 159 141 ...
##  $ BATT_AVG        : num [1:2027] 0.236 0.241 0.242 0.23 0.229 ...
##  $ RUN_DIFF        : num [1:2027] -9 0 -10 0 0 0 -9 0 0 -63 ...

## [1] 2027   17

##  TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B  TEAM_BATTING_HR
##  Min.   :1137   Min.   :118.0   Min.   : 11.00   Min.   :  3.0
##  1st Qu.:1377   1st Qu.:208.0   1st Qu.: 33.00   1st Qu.: 52.0
##  Median :1445   Median :238.0   Median : 45.00   Median :109.0
##  Mean   :1452   Mean   :240.8   Mean   : 52.23   Mean   :105.3
##  3rd Qu.:1523   3rd Qu.:272.0   3rd Qu.: 67.00   3rd Qu.:149.0
##  Max.   :1876   Max.   :392.0   Max.   :147.00   Max.   :264.0
##  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_BASERUN_CS
##  Min.   :189.0   Min.   : 268.0   Min.   : 18.0   Min.   : 11.00
##  1st Qu.:461.0   1st Qu.: 589.0   1st Qu.: 65.0   1st Qu.: 30.00
##  Median :517.0   Median : 760.0   Median :100.0   Median : 40.00
##  Mean   :518.8   Mean   : 765.5   Mean   :119.3   Mean   : 46.19
```

```
##  3rd Qu.:581.0   3rd Qu.: 933.0   3rd Qu.:149.0   3rd Qu.: 56.00
##  Max.   :775.0   Max.   :1399.0   Max.   :654.0   Max.   :201.00
##  TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :29.00    Min.   :1137    Min.   :  3.0    Min.   :284
##  1st Qu.:65.00    1st Qu.:1408    1st Qu.: 57.0    1st Qu.:482
##  Median :65.00    Median :1495    Median :111.0    Median :536
##  Mean   :64.47    Mean   :1525    Mean   :108.4    Mean   :543
##  3rd Qu.:65.00    3rd Qu.:1611    3rd Qu.:152.0    3rd Qu.:601
##  Max.   :95.00    Max.   :2073    Max.   :264.0    Max.   :810
##  TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP    BATT_AVG
##  Min.   : 301.0   Min.   : 65.0   Min.   : 68.0    Min.   :0.2080
##  1st Qu.: 637.0   1st Qu.:125.0   1st Qu.:134.0    1st Qu.:0.2415
##  Median : 817.7   Median :153.0   Median :147.0    Median :0.2501
##  Mean   : 797.9   Mean   :194.8   Mean   :147.3    Mean   :0.2509
##  3rd Qu.: 948.0   3rd Qu.:215.0   3rd Qu.:163.0    3rd Qu.:0.2603
##  Max.   :1434.0   Max.   :791.0   Max.   :225.0    Max.   :0.3016
##     RUN_DIFF
##  Min.   :-767.00
##  1st Qu.: -92.00
##  Median : -70.00
##  Mean   : -76.72
##  3rd Qu.:   0.00
##  Max.   :  29.00

## The Predictor Variables included in the Model 5 V1 Regression are :

##  [1] "BATT_AVG"         "RUN_DIFF"         "TEAM_BASERUN_CS"  "TEAM_BASERUN_SB"
##  [5] "TEAM_BATTING_2B"  "TEAM_BATTING_3B"  "TEAM_BATTING_BB"  "TEAM_BATTING_H"
##  [9] "TEAM_BATTING_HBP" "TEAM_BATTING_HR"  "TEAM_BATTING_SO"  "TEAM_FIELDING_DP"
## [13] "TEAM_FIELDING_E"  "TEAM_PITCHING_BB" "TEAM_PITCHING_H"  "TEAM_PITCHING_HR"
## [17] "TEAM_PITCHING_SO"
```

```
## The formula for regression m5_v1 is :

## TARGET_WINS ~ BATT_AVG + RUN_DIFF + TEAM_BASERUN_CS + TEAM_BASERUN_SB +
##     TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BATTING_H +
##     TEAM_BATTING_HBP + TEAM_BATTING_HR + TEAM_BATTING_SO + TEAM_FIELDING_DP +
##     TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR +
##     TEAM_PITCHING_SO

##
## Call:
## lm(formula = (model5_f1), data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.148  -7.291  -0.084   7.213  60.418
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       34.400288  39.907287   0.862   0.3888
## BATT_AVG         -25.061934 638.223672  -0.039   0.9687
## RUN_DIFF           0.658058   0.099836   6.591 5.55e-11 ***
## TEAM_BASERUN_CS   -0.052637   0.029698  -1.772   0.0765 .
## TEAM_BASERUN_SB    0.082575   0.004958  16.656  < 2e-16 ***
## TEAM_BATTING_2B   -0.036156   0.008981  -4.026 5.88e-05 ***
## TEAM_BATTING_3B    0.177188   0.018233   9.718  < 2e-16 ***
## TEAM_BATTING_BB    0.231924   0.038209   6.070 1.53e-09 ***
## TEAM_BATTING_H    -0.671809   0.123588  -5.436 6.12e-08 ***
## TEAM_BATTING_HBP   0.148729   0.058081   2.561   0.0105 *
## TEAM_BATTING_HR    0.111888   0.010030  11.155  < 2e-16 ***
## TEAM_BATTING_SO   -0.155243   0.018007  -8.621  < 2e-16 ***
## TEAM_FIELDING_DP  -0.083703   0.012048  -6.947 5.01e-12 ***
## TEAM_FIELDING_E   -0.092275   0.004777 -19.318  < 2e-16 ***
## TEAM_PITCHING_BB  -0.186556   0.035828  -5.207 2.12e-07 ***
## TEAM_PITCHING_H    0.702975   0.097266   7.227 6.96e-13 ***
## TEAM_PITCHING_HR         NA         NA      NA       NA
## TEAM_PITCHING_SO   0.133734   0.016839   7.942 3.28e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.99 on 2010 degrees of freedom
## Multiple R-squared:  0.4011, Adjusted R-squared:  0.3964
## F-statistic: 84.15 on 16 and 2010 DF,  p-value: < 2.2e-16
```

```
## Dropping the undefined variable **TEAM_PITCHING_HR**
##  We are dropping this predictor variable since the model residuals is indicating
##  that it is identical to another predictor or it is
##  perfectly predicted by the combination of the other two predictors

## The resulting dataset set is

##  [1] "BATT_AVG"         "RUN_DIFF"         "TEAM_BASERUN_CS"  "TEAM_BASERUN_SB"
##  [5] "TEAM_BATTING_2B"  "TEAM_BATTING_3B"  "TEAM_BATTING_BB"  "TEAM_BATTING_H"
##  [9] "TEAM_BATTING_HBP" "TEAM_BATTING_HR"  "TEAM_BATTING_SO"  "TEAM_FIELDING_DP"
## [13] "TEAM_FIELDING_E"  "TEAM_PITCHING_BB" "TEAM_PITCHING_H"  "TEAM_PITCHING_SO"

## The formula for regression v2 after removing the undefined variable is

## TARGET_WINS ~ BATT_AVG + RUN_DIFF + TEAM_BASERUN_CS + TEAM_BASERUN_SB +
##     TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BATTING_H +
##     TEAM_BATTING_HBP + TEAM_BATTING_HR + TEAM_BATTING_SO + TEAM_FIELDING_DP +
##     TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_SO

## The regression model is now shown as

##
## Call:
## lm(formula = (model5_f2), data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.148  -7.291  -0.084   7.213  60.418
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       34.400288  39.907287   0.862   0.3888
## BATT_AVG         -25.061934 638.223672  -0.039   0.9687
## RUN_DIFF           0.658058   0.099836   6.591 5.55e-11 ***
## TEAM_BASERUN_CS   -0.052637   0.029698  -1.772   0.0765 .
## TEAM_BASERUN_SB    0.082575   0.004958  16.656  < 2e-16 ***
## TEAM_BATTING_2B   -0.036156   0.008981  -4.026 5.88e-05 ***
## TEAM_BATTING_3B    0.177188   0.018233   9.718  < 2e-16 ***
## TEAM_BATTING_BB    0.231924   0.038209   6.070 1.53e-09 ***
## TEAM_BATTING_H    -0.671809   0.123588  -5.436 6.12e-08 ***
## TEAM_BATTING_HBP   0.148729   0.058081   2.561   0.0105 *
## TEAM_BATTING_HR    0.111888   0.010030  11.155  < 2e-16 ***
## TEAM_BATTING_SO   -0.155243   0.018007  -8.621  < 2e-16 ***
## TEAM_FIELDING_DP  -0.083703   0.012048  -6.947 5.01e-12 ***
## TEAM_FIELDING_E   -0.092275   0.004777 -19.318  < 2e-16 ***
## TEAM_PITCHING_BB  -0.186556   0.035828  -5.207 2.12e-07 ***
## TEAM_PITCHING_H    0.702975   0.097266   7.227 6.96e-13 ***
## TEAM_PITCHING_SO   0.133734   0.016839   7.942 3.28e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.99 on 2010 degrees of freedom
## Multiple R-squared:  0.4011, Adjusted R-squared:  0.3964
## F-statistic: 84.15 on 16 and 2010 DF,  p-value: < 2.2e-16

## We see that the p-values of the model using all valid predictor variables have significant
##  impact on the Target Wins
##  Also, the R-Squared indicates that this model explains 40.1% of the variability in Target Wins
##  This R-Squared values is not significantly improved from Models
##  M2 (27.2%), M3 (36.2%), M4 (38.2%) and M5v1 (40.1%)
```

Testing for Multicollinearity

```
## -----VIF Scoring-----

## Multicollinearity occurs when two or more predictor variables
##  are highly correlated to each other, such that they do not provide unique
##  or independent information in the regression model.
##  If the degree of correlation is high enough between variables,
##  it can cause problems when fitting and interpreting the regression model.

## To test this model for Multicollinearity we will employ the
##  imcdiag function from the 'mctest' library and examine the
```

```
##  Variance Inflation Factor (VIF) score
##  Note : Scores over 5 are moderately multicollinear. Scores over 10 are very problematic

##
## Call:
## imcdiag(mod = lm_m5_v2)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##                        VIF    TOL          Wi          Fi Leamer      CVIF
## BATT_AVG          1443.9506 0.0007 193451.5722 207372.6095 0.0263 2166.9590
## RUN_DIFF          2141.0492 0.0005 286909.2689 307555.6488 0.0216 3213.1058
## TEAM_BASERUN_CS      7.1481 0.1399    824.2546    883.5691 0.3740   10.7273
## TEAM_BASERUN_SB      2.6065 0.3837    215.3763    230.8751 0.6194    3.9116
## TEAM_BATTING_2B      2.7791 0.3598    238.5225    255.6869 0.5999    4.1707
## TEAM_BATTING_3B      3.3788 0.2960    318.9224    341.8724 0.5440    5.0707
## TEAM_BATTING_BB    194.4082 0.0051  25929.5886  27795.5170 0.0717  291.7514
## TEAM_BATTING_H    3233.1755 0.0003 433326.9925 464509.7902 0.0176 4852.0766
## TEAM_BATTING_HBP     1.0457 0.9563      6.1254      6.5662 0.9779    1.5693
## TEAM_BATTING_HR      5.8426 0.1712    649.2306    695.9501 0.4137    8.7681
## TEAM_BATTING_SO    255.7535 0.0039  34153.9560  36611.7210 0.0625  383.8133
## TEAM_FIELDING_DP     1.4308 0.6989     57.7579     61.9142 0.8360    2.1472
## TEAM_FIELDING_E      5.3847 0.1857    587.8379    630.1395 0.4309    8.0809
## TEAM_PITCHING_BB   179.9510 0.0056  23991.3594  25717.8101 0.0745  270.0552
## TEAM_PITCHING_H   4250.6957 0.0002 569742.5372 610741.9825 0.0153 6379.0850
## TEAM_PITCHING_SO   204.8213 0.0049  27325.6445  29292.0349 0.0699  307.3785
##                  Klein   IND1   IND2
## BATT_AVG             1 0.0000 1.2504
## RUN_DIFF             1 0.0000 1.2506
## TEAM_BASERUN_CS      1 0.0010 1.0762
## TEAM_BASERUN_SB      1 0.0029 0.7712
## TEAM_BATTING_2B      1 0.0027 0.8010
## TEAM_BATTING_3B      1 0.0022 0.8809
## TEAM_BATTING_BB      1 0.0000 1.2448
## TEAM_BATTING_H       1 0.0000 1.2508
## TEAM_BATTING_HBP     0 0.0071 0.0547
## TEAM_BATTING_HR      1 0.0013 1.0371
## TEAM_BATTING_SO      1 0.0000 1.2463
## TEAM_FIELDING_DP     0 0.0052 0.3767
## TEAM_FIELDING_E      1 0.0014 1.0189
## TEAM_PITCHING_BB     1 0.0000 1.2443
## TEAM_PITCHING_H      1 0.0000 1.2509
## TEAM_PITCHING_SO     1 0.0000 1.2451
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## BATT_AVG , TEAM_BASERUN_CS , coefficient(s) are non-significant may be due to multicollinearity
##
## R-square of y on all x: 0.4011
##
## * use method argument to check which regressors may be the reason of collinearity
## ===================================
```

```
## FIXING MULTICOLLINEARITY

## We will remove the fields from the Regression Model that caused MultiCollinearity

## The resulting dataset set is

## [1] "TEAM_BASERUN_CS"  "TEAM_BASERUN_SB"  "TEAM_BATTING_2B"  "TEAM_BATTING_3B"
## [5] "TEAM_BATTING_HBP" "TEAM_BATTING_HR"  "TEAM_FIELDING_DP" "TEAM_FIELDING_E"

## The formula for regression v3 after removing the MultiCollinear variables is

## TARGET_WINS ~ TEAM_BASERUN_CS + TEAM_BASERUN_SB + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HBP + TEAM_BATTING_HR + TEAM_FIELDING_DP +
##     TEAM_FIELDING_E

## The regression model is now shown as

##
## Call:
```

```
## lm(formula = (model5_f3), data = train_dt_prep1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.492  -8.438   0.384   8.068  69.539
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      50.920432   4.912914  10.365  < 2e-16 ***
## TEAM_BASERUN_CS  -0.064042   0.013126  -4.879 1.15e-06 ***
## TEAM_BASERUN_SB   0.067400   0.004800  14.042  < 2e-16 ***
## TEAM_BATTING_2B   0.023652   0.007117   3.323 0.000905 ***
## TEAM_BATTING_3B   0.293346   0.016615  17.655  < 2e-16 ***
## TEAM_BATTING_HBP  0.220345   0.063084   3.493 0.000488 ***
## TEAM_BATTING_HR   0.087448   0.007478  11.694  < 2e-16 ***
## TEAM_FIELDING_DP -0.046297   0.012485  -3.708 0.000214 ***
## TEAM_FIELDING_E  -0.066893   0.003995 -16.745  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.02 on 2018 degrees of freedom
## Multiple R-squared:  0.2815, Adjusted R-squared:  0.2787
## F-statistic: 98.85 on 8 and 2018 DF,  p-value: < 2.2e-16

## VIF Scoring for Model V3

##
## Call:
## imcdiag(mod = lm_m5_v3)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##                      VIF    TOL       Wi       Fi Leamer   CVIF Klein   IND1
## TEAM_BASERUN_CS  1.1685 0.8558  48.6025  56.7310 0.9251 0.9840     0 0.0030
## TEAM_BASERUN_SB  2.0445 0.4891 301.2763 351.6631 0.6994 1.7217     1 0.0017
## TEAM_BATTING_2B  1.4606 0.6847 132.8487 155.0669 0.8274 1.2300     1 0.0024
## TEAM_BATTING_3B  2.3480 0.4259 388.7950 453.8188 0.6526 1.9772     1 0.0015
## TEAM_BATTING_HBP 1.0323 0.9687   9.3264  10.8862 0.9842 0.8693     0 0.0034
## TEAM_BATTING_HR  2.7179 0.3679 495.4986 578.3680 0.6066 2.2887     1 0.0013
## TEAM_FIELDING_DP 1.2857 0.7778  82.4113  96.1941 0.8819 1.0827     0 0.0027
## TEAM_FIELDING_E  3.1518 0.3173 620.6323 724.4297 0.5633 2.6541     1 0.0011
##                    IND2
## TEAM_BASERUN_CS  0.3706
## TEAM_BASERUN_SB  1.3130
## TEAM_BATTING_2B  0.8104
## TEAM_BATTING_3B  1.4754
## TEAM_BATTING_HBP 0.0805
## TEAM_BATTING_HR  1.6244
## TEAM_FIELDING_DP 0.5711
## TEAM_FIELDING_E  1.7546
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## * all coefficients have significant t-ratios
##
## R-square of y on all x: 0.2815
##
## * use method argument to check which regressors may be the reason of collinearity
## ===================================

## NOTE : THE VIF SCORES FOR MODEL5V3 ARE WELL WITHIN THE RANGES FOR NO
##  MULTICOLLINEARITY EFFECTS
##  THIS MODEL PERFORMS WITH A R-SQUARED OF ONLY 28.2%
##  THIS IS NOT THE OPTIMAL MODEL AMONG THE MODELS IN THIS PROJECT
```

Model Selection and Predictions

Model Selection

```
## Our model selection discussion include the following coefficients for each of the 7 models created

## The coefficients are : R-Squared, Mean Squared Error, F-Statistic, Degrees of Freedom

## R-Squared is a statistical measure that indicates how much of the variation of a dependent
##  variable is explained by an independent variable in a regression mode
##  Typically the higher the R-Squared (50%-90%)  the better the correlation and fit of the model
##   This is a general rule of thumb, the acceptable value is subject to the dataset being examined

## The Mean Squared Error measures how close a regression line is to a set of data points
##  There is no correct value for MSE. Simply put, the lower the value the better and 0 means
##  the model is perfect

## F-statistic, also known as F-value is used in regression analysis to identify the means
##  between two populations are significantly different or not
##  The higher the F value, the better the model

## Degrees of freedom are the number of independent variables that can be estimated
##  in a statistical analysis and tell you how many items can be randomly selected before
##  constraints must be put in place
##  A higher degree of freedom means more power to reject a false null hypothesis
##  and find a significant result
```

Tabulating the Coefficients of the Regression Models

```
## Tabulating coefficients from each regression model

data= matrix(c(1:28), ncol=4, byrow=TRUE)

colnames(data) = c('R-Squared','Mean-Sq-Error','F-Statistic','Degrees-Freedom')
rownames(data) <- c('Model-1','Model-2','Model-3','Model-4','Model-5-1','Model-5-2','Model-5-3')
mmatrix=as.data.frame(data)

## R-Squared

mmatrix[1,1] = rsq_m1
mmatrix[2,1] = rsq_m2
mmatrix[3,1] = rsq_m3
mmatrix[4,1] = rsq_m4
mmatrix[5,1] = rsq_lm_m5_v1
mmatrix[6,1] = rsq_lm_m5_v2
mmatrix[7,1] = rsq_lm_m5_v3

## Mean-Sq-Error

mmatrix[1,2] = mse_m1
mmatrix[2,2] = mse_m2
mmatrix[3,2] = mse_m3
mmatrix[4,2] = mse_m4
mmatrix[5,2] = mse_lm_m5_v1
mmatrix[6,2] = mse_lm_m5_v2
mmatrix[7,2] = mse_lm_m5_v3

## F-Statistic

mmatrix[1,3] = fstat_m1
mmatrix[2,3] = fstat_m2
mmatrix[3,3] = fstat_m3
mmatrix[4,3] = fstat_m4
mmatrix[5,3] = fstat_lm_m5_v1
mmatrix[6,3] = fstat_lm_m5_v2
mmatrix[7,3] = fstat_lm_m5_v3

## Degrees-Freedom

mmatrix[1,3] = df_m1
mmatrix[2,3] = df_m2
mmatrix[3,3] = df_m3
mmatrix[4,3] = df_m4
mmatrix[5,3] = df_lm_m5_v1
```

```
mmatrix[6,3] = df_lm_m5_v2
mmatrix[7,3] = df_lm_m5_v3

mmatrix
```

Our Selected Model

```
## We Examined a total of 7 models

## Our focus on models 5-1, 5-2, 5-3 was primarily to discuss the possible
##  effects of MultiCollinearity between the predictor variables

## As is shown in the table, models 5-1 and 5-2 have similar proeperities
##  since the difference is that model5-2 removes the one variable not defined

## This omission can be seen in the increase of the Degrees of Freem in
##  model 5-2 over 5-1, their R-Squared remain the sample

## Model 5-3 is a result of ommitting the recommended variables based
##  on the VIF scres, this significantly decreased the R-Squared value of
##  the model but increased the Degrees of Freedom to the highest calues of all the models

## We selected to Reommend Model 4, we think that this model best fits the training
##  dataset and will be the most effective predictor of the Evaluation dataset
```

Applying Model 4 to the Evaluation Dataset

We demonstrated the Prediction using multiple models

```
## Model m1 Predictions

##         1         2         3         4         5         6
##  70.64886  73.67706  77.47932  81.98209 116.33858  95.08353

## Model m2 Predictions

##        1        2        3        4        5        6
## 62.60739 66.14266 73.48252 80.48939 55.19009 64.57778

## Model m3 Predictions

##        1        2        3        4        5        6
## 62.24544 66.13393 71.83673 81.44868 64.84525 57.81100

## Model lm_m5_v2 Predictions

##         1         2         3         4         5         6
##  58.81992  65.78855  71.09923  81.05945 193.15707  97.50683

## ----------     OUR PREFERRED MODEL     -----------

## Model m4 Predictions

##         1         2         3         4         5         6
##  60.76596  66.44468  71.80472  81.09637 197.49724  99.30263
```

Data summary	
Name	prediction_m4
Number of rows	259
Number of columns	1
Column type frequency:	
numeric	1
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
data	0	1	91.13	87.92	43.84	74.49	81.58	88.35	1319.33	▇▁▁▁▁

Appendix

Program Code

```
library(tidyverse)
library(skimr)
library(corrplot)
library(knitr)
library(kableExtra)
library(ggthemes)
library(mctest)
library(forecast)
library(MASS)

# URL and CSV file names
url <- "https://raw.githubusercontent.com/Naik-Khyati/data_621/main/hw1/input/"
csv_name1 <- "moneyball-evaluation-data"
csv_name2 <- "moneyball-training-data"

# Read the CSV files into data frames
eval_dt <- read_csv(paste0(url, csv_name1, ".csv"))
train_dt <- read_csv(paste0(url, csv_name2, ".csv"))


cat("The MoneyBall training dataset contain 2276 rows and 17 columns as shown below :")

dim(train_dt)

cat("The following is a brief summary of the first 6 rows are :")

head(train_dt)

cat("The datatypes of the rows are NUMERIC as is shown below", "\n")

column_data_types <- sapply(train_dt, class)
print(column_data_types)


cat("Exploring the Summary Statistics of the Trainig Dataset", "\n")

exploration_summary <- skim(train_dt)

# Display the summary table
exploration_summary

##
train_dt %>%
  gather(variable, value, TARGET_WINS:TEAM_FIELDING_DP) %>%
  ggplot(., aes(value)) +
  geom_density(fill = "Blue", color="Blue") +
  facet_wrap(~variable, scales ="free", ncol = 4)

## Outliers

#Gather the data to create box plots with variable names on the y-axis

gathered_train_dt <- train_dt %>%
  gather(variable, value, -INDEX)

# Create the box plot with variable names on the y-axis
```

```r
ggplot(gathered_train_dt, aes(x = variable, y = value)) +
  geom_boxplot() +
  labs(title = "Box Plots of Numeric Variables", y = "Variable Name") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for readability

 train_dt %>%
  gather(variable, value, -TARGET_WINS) %>%
  ggplot(., aes(value, TARGET_WINS)) +
  geom_point(fill = "blue", color="blue") +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  facet_wrap(~variable, scales ="free", ncol = 4) +
  labs(x = element_blank(), y = "Wins")

temp <- train_dt  %>%
  cor(., use = "complete.obs") #%>%

temp[lower.tri(temp, diag=TRUE)] <- ""
temp <- temp %>%
  as.data.frame() %>%
  rownames_to_column() %>%
  gather(Variable, Correlation, -rowname) %>%
  filter(Variable != rowname) %>%
  filter(Correlation != "") %>%
  mutate(Correlation = as.numeric(Correlation)) %>%
  rename(` Variable` = rowname) %>%
  arrange(desc(abs(Correlation)))

  temp %>%
  filter(` Variable` == "TARGET_WINS") %>%
  kable() %>%
  kable_styling()


## Correlation plot
# Select numeric variables for the correlation matrix
numeric_vars <- select_if(train_dt, is.numeric)

# Calculate the correlation matrix with missing values replaced by 0
correlation_matrix <- cor(numeric_vars, use = "complete.obs")  # Replace NAs with 0

# Create a heat map of the correlation matrix with a specified font
corrplot(correlation_matrix, method = "color", type = "upper", tl.cex = 0.7, tl.srt = 45)

cat("TEAM_BATTING_HBP variable has 2,085 missing values (91.7%)")

cat("TEAM_BASERUN_CS has 772 missing values (33.9%)")

cat("TEAM_BASERUN_SB variable has 131 missing variable (5.7%)")

cat("TEAM_BATTING_SO variable has 102 missing values (4.5%)")

cat("TEAM_PITCHING_SO variable has 102 missing values (4.5%)")

cat("TEAM_FIELDING_DP variable has 286 missing values(12.5%)")

cat("TEAM_PITCHING_H has the highest mean value of 1779.21 among the 17 variables")

cat("TEAM_BASERUN_CS has the lowest mean")

cat("TEAM_PITCHING_H also has the highest median value of 1518 among the 17 variables")

cat("TEAM_BATTING_3B has the lowest median")

cat("TARGET_WINS has half of its values between 71 (25th percentile) and 92 (75th percentile)")

cat("TEAM_PITCHING_BB, TEAM_PITCHING_H and TEAM_PITCHING_SO has the largest number of outliers")

cat("TARGET_WINS has the highest positive correlation of 0.47", "\n", "with TEAM_BATTING_H, TEAM_BATTING_BB and
         TEAM_PITCHING_H")
```

```r
cat("TARGET_WINS has the highest negative correlation of 0.39 with TEAM_FIELDING_E")


cat("Fixing the dataset deficiencies to account for missing data and outliers", "\n")


cat("TEAM_BATTING_HBP has 91.7% of its' values missing, we will replace those with the", "\n", "MLB 2018 and 2019
        averages of 65" )
train_dt <- train_dt %>%
  mutate(TEAM_BATTING_HBP = replace_na(TEAM_BATTING_HBP,65))


cat("TEAM_BASERUN_CS has 33.96% of its' values missing, we will replace those with the", "\n", "MLB 2018 and 2019
        averages of 30" )
train_dt <- train_dt %>%
  mutate(TEAM_BASERUN_CS  = replace_na(TEAM_BASERUN_CS,30))


cat("TEAM_BASERUN_SB has 5.7% of its' values missing, we will replace those with the", "\n", "MEAN of the existing
        values" )

mean_value_sb <- mean(train_dt$TEAM_BASERUN_SB, na.rm = TRUE)

train_dt$TEAM_BASERUN_SB[is.na(train_dt$TEAM_BASERUN_SB)] <- mean_value_sb


cat("TEAM_BATTING_SO has 4.5% of its' values missing, we will replace those with the", "\n", "MEAN of the existing
        values" )

mean_value_sbt <- mean(train_dt$TEAM_BATTING_SO, na.rm = TRUE)

train_dt$TEAM_BATTING_SO[is.na(train_dt$TEAM_BATTING_SO)] <- mean_value_sbt


cat("TEAM_PITCHING_SO has 4.5% of its' values missing, we will replace those with the", "\n", "MEAN of the existing
        values" )

mean_value_p <- mean(train_dt$TEAM_PITCHING_SO, na.rm = TRUE)

train_dt$TEAM_PITCHING_SO[is.na(train_dt$TEAM_PITCHING_SO)] <- mean_value_p


cat("TEAM_FIELDING_DP has 12.5% of its' values missing, we will replace those with the", "\n", "MEAN of the existing
        values" )

mean_value_f_dp <- mean(train_dt$TEAM_FIELDING_DP, na.rm = TRUE)

train_dt$TEAM_FIELDING_DP[is.na(train_dt$TEAM_FIELDING_DP)] <- mean_value_f_dp

cat("From the Data Exploration we see that the following varaibles contain Outlier Values :", "\n",
        "TEAM_PITCHING_SO", "\n", "TEAM_PITCHING_BB", "\n", "TEAM_PITCHING_H")

cat("We will account for these Outliers by applying defining outliers as ", "\n", "values that are more than 1.5
        times the interquartile range (IQR)", "\n","below the first quartile (Q1) or above the third quartile
        (Q3)","\n","By performing the following computations on the dataset")

cat("Calculating first quartile (Q1) and third quartile (Q3) for each variable","\n")

Q1_SO <- quantile(train_dt$TEAM_PITCHING_SO, 0.25)
Q3_SO <- quantile(train_dt$TEAM_PITCHING_SO, 0.75)

Q1_BB <- quantile(train_dt$TEAM_PITCHING_BB, 0.25)
Q3_BB <- quantile(train_dt$TEAM_PITCHING_BB, 0.75)

Q1_H <- quantile(train_dt$TEAM_PITCHING_H, 0.25)
Q3_H <- quantile(train_dt$TEAM_PITCHING_H, 0.75)

cat("Calculating IQR for each variable", "\n")

IQR_SO <- Q3_SO - Q1_SO
```

```r
IQR_BB <- Q3_BB - Q1_BB
IQR_H <- Q3_H - Q1_H

cat("Setting lower and upper bounds for outliers","\n")

lower_bound_SO <- Q1_SO - 1.5 * IQR_SO
upper_bound_SO <- Q3_SO + 1.5 * IQR_SO

lower_bound_BB <- Q1_BB - 1.5 * IQR_BB
upper_bound_BB <- Q3_BB + 1.5 * IQR_BB

lower_bound_H <- Q1_H - 1.5 * IQR_H
upper_bound_H <- Q3_H + 1.5 * IQR_H

cat("Removing outliers from the dataframe","\n")

train_dt_prep1 <- train_dt %>%
  filter(
    TEAM_PITCHING_SO >= lower_bound_SO & TEAM_PITCHING_SO <= upper_bound_SO,
    TEAM_PITCHING_BB >= lower_bound_BB & TEAM_PITCHING_BB <= upper_bound_BB,
    TEAM_PITCHING_H >= lower_bound_H & TEAM_PITCHING_H <= upper_bound_H
  )


cat("We will add to computed varaibles to enhance the model selection", "\n", "BATTING AVERAGE =
         TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS)")

train_dt_prep1 <- train_dt_prep1 %>%
  mutate(BATT_AVG = TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS))

cat("AND", "\n", "RUN DIFFERENTIAL = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR")

train_dt_prep1 <- train_dt_prep1 %>%
  mutate(RUN_DIFF = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR)


cat("Exploring the Summary Statistics of the Enhanced Trainig Dataset", "\n")

exploration_summary <- skim(train_dt_prep1)

# Display the summary table
exploration_summary

## Outliers

#Gather the data to create box plots with variable names on the y-axis

gathered_train_dt_prep1 <- train_dt_prep1 %>%
  gather(variable, value, -INDEX)

# Create the box plot with variable names on the y-axis

ggplot(gathered_train_dt_prep1, aes(x = variable, y = value)) +
  geom_boxplot() +
  labs(title = "Box Plots of Numeric Variables", y = "Variable Name") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for readability


cat("The MoneyBall Evaluation dataset contain 256 rows and 16 columns as shown below :")

dim(eval_dt)

cat("The following is a brief summary of the first 6 rows are :")

head(eval_dt)

cat("The datatypes of the rows are NUMERIC as is shown below", "\n")

column_data_types_eval <- sapply(eval_dt, class)
print(column_data_types_eval)
```

```r
cat("Exploring the Summary Statistics of the Eval Dataset", "\n")

exploration_summary_eval <- skim(eval_dt)

# Display the summary table
exploration_summary_eval

##

cat("We will replace the missing values of TEAM_BATTING_HBP the", "\n", "MLB 2018 and 2019 averages of 65" )

eval_dt <- eval_dt %>%
  mutate(TEAM_BATTING_HBP = replace_na(TEAM_BATTING_HBP,65))

##

cat("We will replace the missing values of TEAM_BASERUN_CS", "\n", "MLB 2018 and 2019 averages of 30" )

eval_dt <- eval_dt %>%
  mutate(TEAM_BASERUN_CS  = replace_na(TEAM_BASERUN_CS,30))

##

cat("We will replace the missing values of TEAM_BASERUN_SB with the", "\n", "MEAN of the existing values")

mean_value_sb_eval <- mean(eval_dt$TEAM_BASERUN_SB, na.rm = TRUE)

eval_dt$TEAM_BASERUN_SB[is.na(eval_dt$TEAM_BASERUN_SB)] <- mean_value_sb_eval

##

cat("We will replace the missing values of TEAM_BATTING_SO with the", "\n", "MEAN of the existing values")

mean_value_sbt_eval <- mean(eval_dt$TEAM_BATTING_SO, na.rm = TRUE)

eval_dt$TEAM_BATTING_SO[is.na(eval_dt$TEAM_BATTING_SO)] <- mean_value_sbt_eval

##

cat("We will replace the missing values of TEAM_PITCHING_SO with the", "\n", "MEAN of the existing values" )

mean_value_p_eval <- mean(eval_dt$TEAM_PITCHING_SO, na.rm = TRUE)

eval_dt$TEAM_PITCHING_SO[is.na(eval_dt$TEAM_PITCHING_SO)] <- mean_value_p_eval

##

cat("We will replace the missing values of TEAM_FIELDING_DP with the", "\n", "MEAN of the existing values")

mean_value_f_dp_eval <- mean(eval_dt$TEAM_FIELDING_DP, na.rm = TRUE)

eval_dt$TEAM_FIELDING_DP[is.na(eval_dt$TEAM_FIELDING_DP)] <- mean_value_f_dp_eval

##

cat("We will add to computed varaibles to enhance the model selection", "\n", "BATTING AVERAGE =
         TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS)")

eval_dt <- eval_dt %>%
  mutate(BATT_AVG = TEAM_BATTING_H/(TEAM_BATTING_H + 4374 - TEAM_BASERUN_CS))

##

cat("AND", "\n", "RUN DIFFERENTIAL = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR")

eval_dt <- eval_dt %>%
  mutate(RUN_DIFF = TEAM_BATTING_H + TEAM_BATTING_HR - TEAM_PITCHING_H - TEAM_PITCHING_HR)

##

cat("Exploring the Summary Statistics of the Revised Eval Dataset", "\n")
```

```r
exploration_summary_eval2 <- skim(eval_dt)

# Display the summary table
exploration_summary_eval2

##

cat("The Eval dataset is now ready to accept predictions from the Training dataset")


cat("Exploring a model using ONLY Predictor Variables which have a THEORETICAL NEGATIVE Impact on Wins")

m1 <-   lm(TARGET_WINS ~ TEAM_BATTING_SO + TEAM_BASERUN_CS + TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H +
         TEAM_PITCHING_HR, data = train_dt_prep1)

summary(m1)
cat("We see that the p-values of the 6 predictor variables have significant impact on the Target Wins", "\n", "Also,
         the R-Squared indicates that this model explains 17.6% of the variability in Target Wins ")

## The R-Squared value of this model is given as

rsq_m1 <- (summary(m1)$r.squared)

## The Mean Squared Error of this model is given as

mse_m1 <- mean(summary(m1)$residuals^2)

## The F-Statistic of this model is given as

fstat_m1 <- (summary(m1)$fstatistic[1])

## The P-Value of this model is given as

pval_m1 <- (summary(m1)$coefficients[,4])

## The degrees of freedom for this model is given as

df_m1 <- (summary(m1)$fstatistic[3])


cat("Exploring a model using ONLY Predictor Variables which have a THEORETICAL POSITIVE Impact on Wins")

m2 <-   lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
         TEAM_BATTING_HBP + TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_PITCHING_SO, data = train_dt_prep1)

summary(m2)

cat("We see that the p-values of the 9 predictor variables have significant impact on the Target Wins", "\n", "Also,
         the R-Squared indicates that this model explains 27.2% of the variability in Target Wins ")

## The R-Squared value of this model is given as

rsq_m2 <- (summary(m2)$r.squared)

## The Mean Squared Error of this model is given as

mse_m2 <- mean(summary(m2)$residuals^2)

## The F-Statistic of this model is given as

fstat_m2 <- (summary(m2)$fstatistic[1])

## The P-Value of this model is given as

pval_m2 <- (summary(m2)$coefficients[,4])

## The degrees of freedom for this model is given as

df_m2 <- (summary(m2)$fstatistic[3])
```

```r
cat("Exploring a model using 11 Impactful Predictor Variables on Wins")

m3 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
         TEAM_BASERUN_SB + TEAM_BATTING_HBP + TEAM_PITCHING_SO + TEAM_BATTING_SO +TEAM_FIELDING_DP+TEAM_FIELDING_E,
         data = train_dt_prep1)

summary(m3)

cat("We see that the p-values of the 11 predictor variables have significant impact on the Target Wins", "\n",
         "Also, the R-Squared indicates that this model explains 36.2% of the variability in Target Wins ")

## The R-Squared value of this model is given as

rsq_m3 <- (summary(m3)$r.squared)

## The Mean Squared Error of this model is given as

mse_m3 <- mean(summary(m3)$residuals^2)

## The F-Statistic of this model is given as

fstat_m3 <- (summary(m3)$fstatistic[1])

## The P-Value of this model is given as

pval_m3 <- (summary(m3)$coefficients[,4])

## The degrees of freedom for this model is given as

df_m3 <- (summary(m3)$fstatistic[3])


cat("Exploring a model using 13 Impactful Predictor Variables on Wins")

m4 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
         TEAM_BASERUN_SB + TEAM_BATTING_HBP + TEAM_PITCHING_SO +TEAM_PITCHING_BB+TEAM_PITCHING_H+ TEAM_BATTING_SO
         +TEAM_FIELDING_DP+TEAM_FIELDING_E, data = train_dt_prep1)

summary(m4)

cat("We see that the p-values of the 13 predictor variables have significant impact on the Target Wins", "\n", "Also
         the R-Squared indicates that this model explains 38.2% of the variability in Target Wins ")

## The R-Squared value of this model is given as

rsq_m4 <- (summary(m4)$r.squared)

## The Mean Squared Error of this model is given as

mse_m4 <- mean(summary(m4)$residuals^2)

## The F-Statistic of this model is given as

fstat_m4 <- (summary(m4)$fstatistic[1])

## The P-Value of this model is given as

pval_m4 <- (summary(m4)$coefficients[,4])

## The degrees of freedom for this model is given as

df_m4 <- (summary(m4)$fstatistic[3])


cat("Exploring a model using ALL Predictor Variables on Wins")

cat("Checking the structure of the dataset","\n")

cat("Dropping the 'INDEX' variable")
```

```r
train_dt_prep2 <- subset(train_dt_prep1, select = -c(INDEX,TARGET_WINS))

str(train_dt_prep2)

dim(train_dt_prep2)

summary(train_dt_prep2)

train_dt_prep2[!complete.cases(train_dt_prep2),]

cat("The Predictor Variables included in the Model 5 V1 Regression are : ", "\n")

sort(colnames(train_dt_prep2))


model5_f1 <- as.formula(paste("TARGET_WINS", "~",
        paste(sort(colnames(train_dt_prep2)), collapse = "+"),
        sep = ""
    ))

cat("The formula for regression m5_v1 is :","\n")

model5_f1

lm_m5_v1 <- lm((model5_f1),data = train_dt_prep1 )

summary(lm_m5_v1)

##########################

## The R-Squared value of this model is given as

rsq_lm_m5_v1 <- (summary(lm_m5_v1)$r.squared)

## The Mean Squared Error of this model is given as

mse_lm_m5_v1 <- mean(summary(lm_m5_v1)$residuals^2)

## The F-Statistic of this model is given as

fstat_lm_m5_v1 <- (summary(lm_m5_v1)$fstatistic[1])

## The P-Value of this model is given as

pval_lm_m5_v1 <- (summary(lm_m5_v1)$coefficients[,4])

## The degrees of freedom for this model is given as

df_lm_m5_v1 <- (summary(lm_m5_v1)$fstatistic[3])

##########################

cat("Dropping the undefined variable **TEAM_PITCHING_HR**","\n",
"We are dropping this predictor variable since the model residuals is indicating", "\n",
"that it is identical to another predictor or it is","\n",
"perfectly predicted by the combination of the other two predictors")

train_dt_prep3 <- subset(train_dt_prep2, select = -c(TEAM_PITCHING_HR))

cat("The resulting dataset set is","\n")

sort(colnames(train_dt_prep3))

cat("The formula for regression v2 after removing the undefined variable is","\n")

model5_f2 <- as.formula(paste("TARGET_WINS", "~",
        paste(sort(colnames(train_dt_prep3)), collapse = "+"),
        sep = ""
    ))
```

```r
model5_f2

cat("The regression model is now shown as", "\n")

lm_m5_v2 <- lm((model5_f2),data = train_dt_prep1 )

summary(lm_m5_v2)

################

## The R-Squared value of this model is given as

rsq_lm_m5_v2 <- (summary(lm_m5_v2)$r.squared)

## The Mean Squared Error of this model is given as

mse_lm_m5_v2 <- mean(summary(lm_m5_v2)$residuals^2)

## The F-Statistic of this model (M1) is given as

fstat_lm_m5_v2 <- (summary(lm_m5_v2)$fstatistic[1])

## The P-Value of this model is given as

pval_lm_m5_v2 <- (summary(lm_m5_v2)$coefficients[,4])

## The degrees of freedom for this model is given as

df_lm_m5_v2 <- (summary(lm_m5_v2)$fstatistic[3])

################

cat("We see that the p-values of the model using all valid predictor variables have significant", "\n",
"impact on the Target Wins", "\n",
"Also, the R-Squared indicates that this model explains 40.1% of the variability in Target Wins","\n",
"This R-Squared values is not significantly improved from Models", "\n",
"M2 (27.2%), M3 (36.2%), M4 (38.2%) and M5v1 (40.1%)")



cat("-----VIF Scoring-----")

cat("Multicollinearity occurs when two or more predictor variables", "\n",
"are highly correlated to each other, such that they do not provide unique", "\n",
"or independent information in the regression model.", "\n",
"If the degree of correlation is high enough between variables,", "\n",
"it can cause problems when fitting and interpreting the regression model.")

cat("To test this model for Multicollinearity we will employ the", "\n",
"imcdiag function from the 'mctest' library and examine the", "\n",
"Variance Inflation Factor (VIF) score", "\n",
"Note : Scores over 5 are moderately multicollinear. Scores over 10 are very problematic")

imcdiag(lm_m5_v2)


cat("FIXING MULTICOLLINEARITY")

cat("We will remove the fields from the Regression Model that caused MultiCollinearity")

train_dt_prep4 <- subset(train_dt_prep3, select = -c(BATT_AVG, RUN_DIFF, TEAM_BATTING_BB, TEAM_BATTING_H,
         TEAM_BATTING_SO, TEAM_PITCHING_BB, TEAM_PITCHING_H, TEAM_PITCHING_SO ))

cat("The resulting dataset set is","\n")

sort(colnames(train_dt_prep4))

cat("The formula for regression v3 after removing the MultiCollinear variables is","\n")

model5_f3 <- as.formula(paste("TARGET_WINS", "~",
        paste(sort(colnames(train_dt_prep4)), collapse = "+"),
```

```
        sep = ""
    ))

model5_f3

cat("The regression model is now shown as", "\n")

lm_m5_v3 <- lm((model5_f3),data = train_dt_prep1 )

summary(lm_m5_v3)

##############################

## The R-Squared value of this model is given as

rsq_lm_m5_v3 <- (summary(lm_m5_v3)$r.squared)

## The Mean Squared Error of this model is given as

mse_lm_m5_v3 <- mean(summary(lm_m5_v3)$residuals^2)

## The F-Statistic of this model (M1) is given as

fstat_lm_m5_v3 <- (summary(lm_m5_v3)$fstatistic[1])

## The P-Value of this model is given as

pval_lm_m5_v3 <- (summary(lm_m5_v3)$coefficients[,4])

## The degrees of freedom for this model is given as

df_lm_m5_v3 <- (summary(lm_m5_v3)$fstatistic[3])

################################


cat("VIF Scoring for Model V3")

imcdiag(lm_m5_v3)

cat("NOTE : THE VIF SCORES FOR MODEL5V3 ARE WELL WITHIN THE RANGES FOR NO", "\n",
    "MULTICOLLINEARITY EFFECTS", "\n",
    "THIS MODEL PERFORMS WITH A R-SQUARED OF ONLY 28.2%", "\n",
    "THIS IS NOT THE OPTIMAL MODEL AMONG THE MODELS IN THIS PROJECT")


cat("Our model selection discussion include the following coefficients for each of the 7 models created")

cat("The coefficients are : R-Squared, Mean Squared Error, F-Statistic, Degrees of Freedom", "\n")

cat("R-Squared is a statistical measure that indicates how much of the variation of a dependent", "\n",
    "variable is explained by an independent variable in a regression mode", "\n",
    "Typically the higher the R-Squared (50%-90%)  the better the correlation and fit of the model", "\n",
    " This is a general rule of thumb, the acceptable value is subject to the dataset being examined")

cat("The Mean Squared Error measures how close a regression line is to a set of data points", "\n",
    "There is no correct value for MSE. Simply put, the lower the value the better and 0 means", "\n",
    "the model is perfect")

cat("F-statistic, also known as F-value is used in regression analysis to identify the means", "\n",
"between two populations are significantly different or not","\n",
"The higher the F value, the better the model")

cat("Degrees of freedom are the number of independent variables that can be estimated", "\n",
"in a statistical analysis and tell you how many items can be randomly selected before", "\n",
"constraints must be put in place", "\n",
"A higher degree of freedom means more power to reject a false null hypothesis", "\n",
"and find a significant result")


## Tabulating coefficients from each regression model
```

```r
data= matrix(c(1:28), ncol=4, byrow=TRUE)

colnames(data) = c('R-Squared','Mean-Sq-Error','F-Statistic','Degrees-Freedom')
rownames(data) <- c('Model-1','Model-2','Model-3','Model-4','Model-5-1','Model-5-2','Model-5-3')
mmatrix=as.data.frame(data)

## R-Squared

mmatrix[1,1] = rsq_m1
mmatrix[2,1] = rsq_m2
mmatrix[3,1] = rsq_m3
mmatrix[4,1] = rsq_m4
mmatrix[5,1] = rsq_lm_m5_v1
mmatrix[6,1] = rsq_lm_m5_v2
mmatrix[7,1] = rsq_lm_m5_v3

## Mean-Sq-Error

mmatrix[1,2] = mse_m1
mmatrix[2,2] = mse_m2
mmatrix[3,2] = mse_m3
mmatrix[4,2] = mse_m4
mmatrix[5,2] = mse_lm_m5_v1
mmatrix[6,2] = mse_lm_m5_v2
mmatrix[7,2] = mse_lm_m5_v3

## F-Statistic

mmatrix[1,3] = fstat_m1
mmatrix[2,3] = fstat_m2
mmatrix[3,3] = fstat_m3
mmatrix[4,3] = fstat_m4
mmatrix[5,3] = fstat_lm_m5_v1
mmatrix[6,3] = fstat_lm_m5_v2
mmatrix[7,3] = fstat_lm_m5_v3

## Degrees-Freedom

mmatrix[1,3] = df_m1
mmatrix[2,3] = df_m2
mmatrix[3,3] = df_m3
mmatrix[4,3] = df_m4
mmatrix[5,3] = df_lm_m5_v1
mmatrix[6,3] = df_lm_m5_v2
mmatrix[7,3] = df_lm_m5_v3

mmatrix


cat("We Examined a total of 7 models")

cat("Our focus on models 5-1, 5-2, 5-3 was primarily to discuss the possible", "\n",
    "effects of MultiCollinearity between the predictor variables")

cat("As is shown in the table, models 5-1 and 5-2 have similar proeperities", "\n",
    "since the difference is that model5-2 removes the one variable not defined")

cat("This omission can be seen in the increase of the Degrees of Freem in", "\n",
    "model 5-2 over 5-1, their R-Squared remain the sample")

cat("Model 5-3 is a result of ommitting the recommended variables based", "\n",
    "on the VIF scres, this significantly decreased the R-Squared value of", "\n",
    "the model but increased the Degrees of Freedom to the highest calues of all the models")

cat("We selected to Reommend Model 4, we think that this model best fits the training", "\n",
    "dataset and will be the most effective predictor of the Evaluation dataset")


cat("Model m1 Predictions")
```

```r
prediction_m1 <- predict(m1,eval_dt, type = "response")

head(prediction_m1)

##

cat("Model m2 Predictions")

prediction_m2 <- predict(m2,eval_dt, type = "response")

head(prediction_m2)

##

cat("Model m3 Predictions")

prediction_m3 <- predict(m3,eval_dt, type = "response")

head(prediction_m3)

##

cat("Model lm_m5_v2 Predictions")

prediction_lm_m5_v2 <- predict(lm_m5_v2,eval_dt, type = "response")

head(prediction_lm_m5_v2)

cat("----------     OUR PREFERRED MODEL     -----------")

cat("Model m4 Predictions")

prediction_m4 <- predict(m4,eval_dt, type = "response")

head(prediction_m4)

exploration_summary_prediction_M4 <- skim(prediction_m4)

# Display the summary table
exploration_summary_prediction_M4
```

References

https://mathworld.wolfram.com/ExponentialSumFormulas.html

https://pubs.wsb.wisc.edu/academics/analytics-using-r-2019/gamma-variables-optional.html

https://www.programmingr.com/examples/neat-tricks/sample-r-function/rexp/

https://bookdown.org/rdpeng/rprogdatascience/simulation.html

https://math.stackexchange.com/questions/2189317/mean-of-gamma-distribution

https://www.youtube.com/watch?v=cI-WFRqXbKM