Astrid Barreras

Load libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Lahman)
library(ggplot2)
library(reshape2)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(knitr)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1
## ✔ readr     2.1.5
```

```
## ── 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 confli
cts to become errors
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

Read in Data

I am using Teams data from the Lahman R package. Documentation for this dataset can be found in Lahmans website (https://www.dropbox.com/scl/fi/9i2nhlskvfkqy7mbuqem7/readme2023.txt?rlkey=0dnwx7ujztm0z40b8dmggfcr0&dl=0).

```r
# Load datasets from package
data(package = "Lahman")
```

```r
# Load Teams dataset
data("Teams")
```

```
# Create a data frame with the column names and descriptions
data_description <- data.frame(
  teams_col = c("yearID", "lgID", "teamID", "franchID", "divID", "Rank", "G", "GHom
e", "W", "L",
             "DivWin", "WCWin", "LgWin", "WSWin", "R", "AB", "H", "2B", "3B", "HR",
"BB", "SO",
             "SB", "CS", "HBP", "SF", "RA", "ER", "ERA", "CG", "SHO", "SV", "IPOut
s", "HA", "HRA",
             "BBA", "SOA", "E", "DP", "FP", "name", "park", "attendance", "BPF", "PP
F", "teamIDBR",
             "teamIDlahman45", "teamIDretro"),
  col_des = c("Year", "League", "Team", "Franchise (links to TeamsFranchise table)",
                  "Team's division", "Position in final standings", "Games played",
                  "Games played at home", "Wins", "Losses", "Division Winner (Y or
N)",
                  "Wild Card Winner (Y or N)", "League Champion (Y or N)",
                  "World Series Winner (Y or N)", "Runs scored", "At bats", "Hits by
batters",
                  "Doubles", "Triples", "Home runs by batters", "Walks by batters",
                  "Strikeouts by batters", "Stolen bases", "Caught stealing", "Batte
rs hit by pitch",
                  "Sacrifice flies", "Opponents runs scored", "Earned runs allowed",
                  "Earned run average", "Complete games", "Shutouts", "Saves",
                  "Outs Pitched (innings pitched x 3)", "Hits allowed", "Home runs a
llowed",
                  "Walks allowed", "Strikeouts by pitchers", "Errors", "Double Play
s",
                  "Fielding percentage", "Team's full name", "Name of team's home ba
llpark",
                  "Home attendance total", "Three-year park factor for batters",
                  "Three-year park factor for pitchers", "Team ID used by Baseball R
eference website",
                  "Team ID used in Lahman database version 4.5", "Team ID used by Re
trosheet")
)

# Display the table
kable(data_description, col.names = c("Column", "Description"), caption = "TEAMS Dat
a Description")
```

TEAMS Data Description

Column	Description
yearID	Year
lgID	League
teamID	Team

Column	Description
franchID	Franchise (links to TeamsFranchise table)
divID	Team’s division
Rank	Position in final standings
G	Games played
GHome	Games played at home
W	Wins
L	Losses
DivWin	Division Winner (Y or N)
WCWin	Wild Card Winner (Y or N)
LgWin	League Champion (Y or N)
WSWin	World Series Winner (Y or N)
R	Runs scored
AB	At bats
H	Hits by batters
2B	Doubles
3B	Triples
HR	Home runs by batters
BB	Walks by batters
SO	Strikeouts by batters
SB	Stolen bases
CS	Caught stealing
HBP	Batters hit by pitch
SF	Sacrifice flies
RA	Opponents runs scored
ER	Earned runs allowed
ERA	Earned run average
CG	Complete games
SHO	Shutouts
SV	Saves

Column	Description
IPOuts	Outs Pitched (innings pitched x 3)
HA	Hits allowed
HRA	Home runs allowed
BBA	Walks allowed
SOA	Strikeouts by pitchers
E	Errors
DP	Double Plays
FP	Fielding percentage
name	Team's full name
park	Name of team's home ballpark
attendance	Home attendance total
BPF	Three-year park factor for batters
PPF	Three-year park factor for pitchers
teamIDBR	Team ID used by Baseball Reference website
teamIDlahman45	Team ID used in Lahman database version 4.5
teamIDretro	Team ID used by Retrosheet

View of the Data

```
# Show the first 6 rows of the dataset
head(Teams)
```

```
##   yearID lgID teamID franchID divID Rank  G Ghome  W  L DivWin WCWin LgWin
## 1   1871   NA    BS1      BNA  <NA>    3 31    NA 20 10   <NA>  <NA>     N
## 2   1871   NA    CH1      CNA  <NA>    2 28    NA 19  9   <NA>  <NA>     N
## 3   1871   NA    CL1      CFC  <NA>    8 29    NA 10 19   <NA>  <NA>     N
## 4   1871   NA    FW1      KEK  <NA>    7 19    NA  7 12   <NA>  <NA>     N
## 5   1871   NA    NY2      NNA  <NA>    5 33    NA 16 17   <NA>  <NA>     N
## 6   1871   NA    PH1      PNA  <NA>    1 28    NA 21  7   <NA>  <NA>     Y
##   WSWin   R   AB   H X2B X3B HR BB SO SB CS HBP SF  RA  ER  ERA CG SHO SV
## 1  <NA> 401 1372 426  70  37  3 60 19 73 16  NA NA 303 109 3.55 22   1  3
## 2  <NA> 302 1196 323  52  21 10 60 22 69 21  NA NA 241  77 2.76 25   0  1
## 3  <NA> 249 1186 328  35  40  7 26 25 18  8  NA NA 341 116 4.11 23   0  0
## 4  <NA> 137  746 178  19   8  2 33  9 16  4  NA NA 243  97 5.17 19   1  0
## 5  <NA> 302 1404 403  43  21  1 33 15 46 15  NA NA 313 121 3.72 32   1  0
## 6  <NA> 376 1281 410  66  27  9 46 23 56 12  NA NA 266 137 4.95 27   0  0
##   IPouts  HA HRA BBA SOA   E DP    FP                    name
## 1    828 367   2  42  23 243 24 0.834    Boston Red Stockings
## 2    753 308   6  28  22 229 16 0.829 Chicago White Stockings
## 3    762 346  13  53  34 234 15 0.818  Cleveland Forest Citys
## 4    507 261   5  21  17 163  8 0.803    Fort Wayne Kekiongas
## 5    879 373   7  42  22 235 14 0.840        New York Mutuals
## 6    747 329   3  53  16 194 13 0.845  Philadelphia Athletics
##                           park attendance BPF PPF teamIDBR teamIDlahman45
## 1          South End Grounds I         NA 103  98      BOS            BS1
## 2      Union Base-Ball Grounds         NA 104 102      CHI            CH1
## 3 National Association Grounds         NA  96 100      CLE            CL1
## 4               Hamilton Field         NA 101 107      KEK            FW1
## 5     Union Grounds (Brooklyn)         NA  90  88      NYU            NY2
## 6     Jefferson Street Grounds         NA 102  98      ATH            PH1
##   teamIDretro
## 1         BS1
## 2         CH1
## 3         CL1
## 4         FW1
## 5         NY2
## 6         PH1
```

```
# Show structure of the dataset
str(Teams)
```

```
## 'data.frame':    3045 obs. of  48 variables:
##  $ yearID        : int  1871 1871 1871 1871 1871 1871 1871 1871 1871 1872 ...
##  $ lgID          : Factor w/ 7 levels "AA","AL","FL",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ teamID        : Factor w/ 149 levels "ALT","ANA","ARI",..: 24 31 39 56 90 97 1
11 136 142 8 ...
##  $ franchID      : Factor w/ 120 levels "ALT","ANA","ARI",..: 13 36 25 56 70 85 9
1 109 77 9 ...
##  $ divID         : chr  NA NA NA NA ...
##  $ Rank          : int  3 2 8 7 5 1 9 6 4 2 ...
##  $ G             : int  31 28 29 19 33 28 25 29 32 58 ...
##  $ Ghome         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ W             : int  20 19 10 7 16 21 4 13 15 35 ...
##  $ L             : int  10 9 19 12 17 7 21 15 15 19 ...
##  $ DivWin        : chr  NA NA NA NA ...
##  $ WCWin         : chr  NA NA NA NA ...
##  $ LgWin         : chr  "N" "N" "N" "N" ...
##  $ WSWin         : chr  NA NA NA NA ...
##  $ R             : int  401 302 249 137 302 376 231 351 310 617 ...
##  $ AB            : int  1372 1196 1186 746 1404 1281 1036 1248 1353 2571 ...
##  $ H             : int  426 323 328 178 403 410 274 384 375 753 ...
##  $ X2B           : int  70 52 35 19 43 66 44 51 54 106 ...
##  $ X3B           : int  37 21 40 8 21 27 25 34 26 31 ...
##  $ HR            : int  3 10 7 2 1 9 3 6 6 14 ...
##  $ BB            : int  60 60 26 33 33 46 38 49 48 29 ...
##  $ SO            : int  19 22 25 9 15 23 30 19 13 28 ...
##  $ SB            : int  73 69 18 16 46 56 53 62 48 53 ...
##  $ CS            : int  16 21 8 4 15 12 10 24 13 18 ...
##  $ HBP           : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SF            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ RA            : int  303 241 341 243 313 266 287 362 303 434 ...
##  $ ER            : int  109 77 116 97 121 137 108 153 137 166 ...
##  $ ERA           : num  3.55 2.76 4.11 5.17 3.72 4.95 4.3 5.51 4.37 2.9 ...
##  $ CG            : int  22 25 23 19 32 27 23 28 32 48 ...
##  $ SHO           : int  1 0 0 1 1 0 1 0 0 1 ...
##  $ SV            : int  3 1 0 0 0 0 0 0 0 1 ...
##  $ IPouts        : int  828 753 762 507 879 747 678 750 846 1548 ...
##  $ HA            : int  367 308 346 261 373 329 315 431 371 573 ...
##  $ HRA           : int  2 6 13 5 7 3 3 4 4 3 ...
##  $ BBA           : int  42 28 53 21 42 53 34 75 45 63 ...
##  $ SOA           : int  23 22 34 17 22 16 16 12 13 77 ...
##  $ E             : int  243 229 234 163 235 194 220 198 218 432 ...
##  $ DP            : int  24 16 15 8 14 13 14 22 20 22 ...
##  $ FP            : num  0.834 0.829 0.818 0.803 0.84 0.845 0.821 0.845 0.85 0.83
...
##  $ name          : chr  "Boston Red Stockings" "Chicago White Stockings" "Clevela
nd Forest Citys" "Fort Wayne Kekiongas" ...
##  $ park          : chr  "South End Grounds I" "Union Base-Ball Grounds" "National
Association Grounds" "Hamilton Field" ...
```

```
##  $ attendance    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BPF           : int  103 104 96 101 90 102 97 101 94 106 ...
##  $ PPF           : int  98 102 100 107 88 98 99 100 98 102 ...
##  $ teamIDBR      : chr  "BOS" "CHI" "CLE" "KEK" ...
##  $ teamIDlahman45: chr  "BS1" "CH1" "CL1" "FW1" ...
##  $ teamIDretro   : chr  "BS1" "CH1" "CL1" "FW1" ...
```

All variables in the dataset are set to their appropriate data types.

Missing Values

```
# Get columns with missing values and their percentage of missing data
missing_data <- colMeans(is.na(Teams)) * 100
missing_data<- missing_data[missing_data > 0]

# Display the results
print(missing_data)
```

```
##      divID      Ghome     DivWin      WCWin      LgWin      WSWin         SO
## 49.8193760 13.1034483 50.7389163 71.6256158  0.9195402 11.7241379  0.5254516
##         SB         CS        HBP         SF       park attendance
##  4.1050903 27.2906404 38.0295567 50.6075534  1.1165846  9.1625616
```

```
# Deleting columns with > 50% missing data
Teams <- Teams %>% select(-divID, -SF, - DivWin, -WCWin)
```

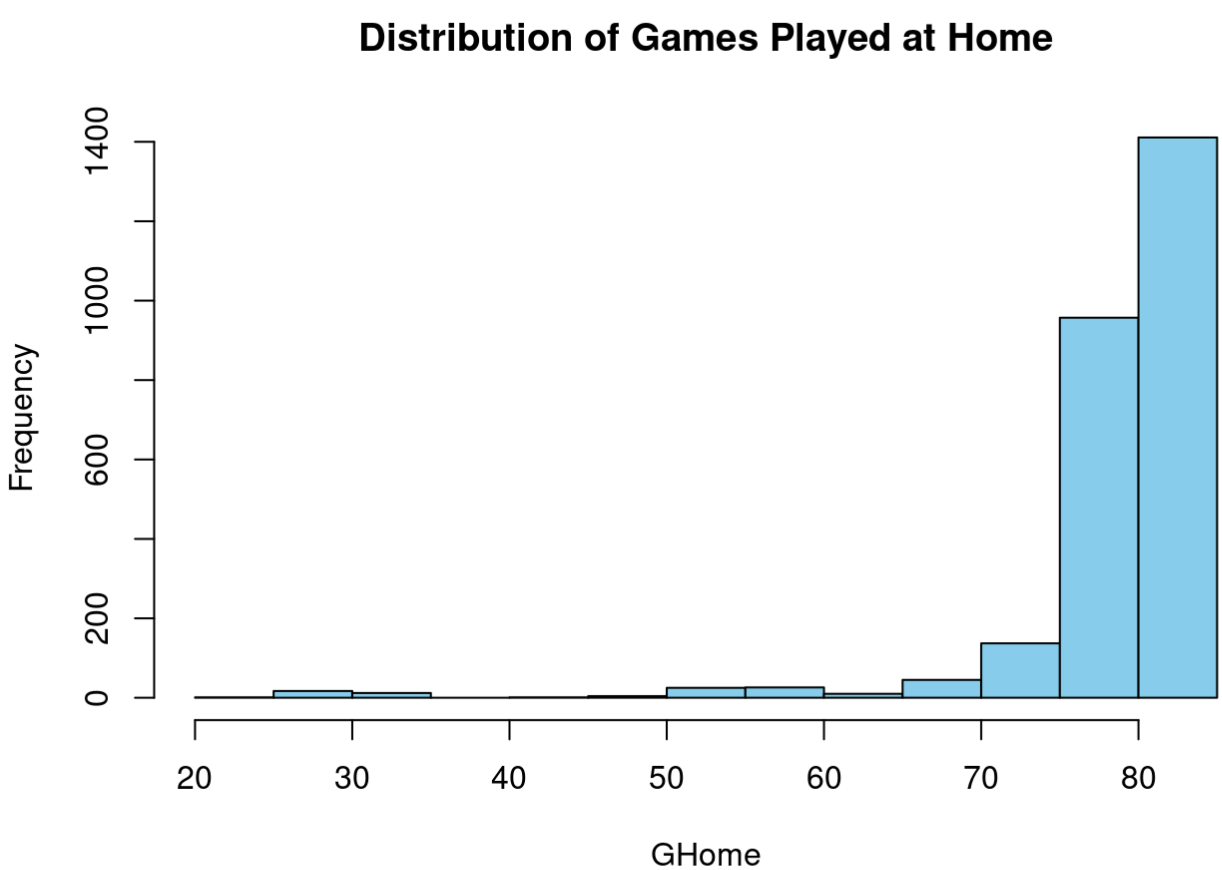
Columns divId, SF, DivWin, and WCWin were deleted as they contained more than 50% missing data. SF (sacrifice flies) began being tracked in 1954 and has undergone some statistical changes over the years. Given, imputation would not be a viable method to handle the missing values in this column. divID can be ignored as it was implemented in 1969 as teams were divided into East and West. DivWin was implemented in 1969 when divisions were first introduced. WCWin was implemented in 1995 when the playoff format was restructured, and has undergone further restructing in 2012 and 2022. Missing values for divID, DivWin, and WCWin should not be imputed as it would cause inaccuracy in data, given the leagues changes.

```
# Checking to see if the above columns were deleted
names(Teams)
```

```
##  [1] "yearID"         "lgID"           "teamID"         "franchID"
##  [5] "Rank"           "G"              "Ghome"          "W"
##  [9] "L"              "LgWin"          "WSWin"          "R"
## [13] "AB"             "H"              "X2B"            "X3B"
## [17] "HR"             "BB"             "SO"             "SB"
## [21] "CS"             "HBP"            "RA"             "ER"
## [25] "ERA"            "CG"             "SHO"            "SV"
## [29] "IPouts"         "HA"             "HRA"            "BBA"
## [33] "SOA"            "E"              "DP"             "FP"
## [37] "name"           "park"           "attendance"     "BPF"
## [41] "PPF"            "teamIDBR"       "teamIDlahman45" "teamIDretro"
```

Columns were successfully deleted.

```
# Visualize distribution of Games played at home (GHome)
hist(Teams$Ghome, main = "Distribution of Games Played at Home", xlab = "GHome", col
= "skyblue")
```



The bar graph above for the distribution of GHome illustrates a distribution skewed to the right.

```
# Imputation of Ghome missing data with median
Teams$Ghome[is.na(Teams$Ghome)] <- median(Teams$Ghome, na.rm = TRUE)
```

Imputation of Ghome is done with the median since the data distribution is skewed to the right and since the number of home games is typically consistent from season to season, team to team.

```
# Check if column has any missing values
any(is.na(Teams$Ghome))
```

```
## [1] FALSE
```

This shows that the imputation method worked.

```
# Subset rows with missing LgWin values
missing_lgwin <- Teams[is.na(Teams$LgWin), ]

# View the subset
print(missing_lgwin)
```

```
##      yearID lgID teamID franchID Rank   G Ghome  W  L LgWin WSWin   R   AB    H
## 2154   1994   AL    BAL      BAL    2 112    55 63 49  <NA>  <NA> 589 3856 1047
## 2155   1994   AL    BOS      BOS    4 115    64 54 61  <NA>  <NA> 552 3940 1038
## 2156   1994   AL    CAL      ANA    4 115    63 47 68  <NA>  <NA> 543 3943 1042
## 2157   1994   AL    CHA      CHW    1 113    53 67 46  <NA>  <NA> 633 3942 1133
## 2158   1994   AL    CLE      CLE    2 113    51 66 47  <NA>  <NA> 679 4022 1165
## 2159   1994   AL    DET      DET    5 115    58 53 62  <NA>  <NA> 652 3955 1048
## 2160   1994   AL    KCA      KCR    3 115    59 64 51  <NA>  <NA> 574 3911 1051
## 2161   1994   AL    MIN      MIN    4 113    59 53 60  <NA>  <NA> 594 3952 1092
## 2162   1994   AL    ML4      MIL    5 115    56 53 62  <NA>  <NA> 547 3978 1045
## 2163   1994   AL    NYA      NYY    1 113    57 70 43  <NA>  <NA> 670 3986 1155
## 2164   1994   AL    OAK      OAK    2 114    56 51 63  <NA>  <NA> 549 3885 1009
## 2165   1994   AL    SEA      SEA    3 112    44 49 63  <NA>  <NA> 569 3883 1045
## 2166   1994   AL    TEX      TEX    1 114    63 52 62  <NA>  <NA> 613 3983 1114
## 2167   1994   AL    TOR      TOR    3 115    59 55 60  <NA>  <NA> 566 3962 1064
## 2168   1994   NL    ATL      ATL    2 114    55 68 46  <NA>  <NA> 542 3861 1031
## 2169   1994   NL    CHN      CHC    5 113    59 49 64  <NA>  <NA> 500 3918 1015
## 2170   1994   NL    CIN      CIN    1 115    60 66 48  <NA>  <NA> 609 3999 1142
## 2171   1994   NL    COL      COL    3 117    57 53 64  <NA>  <NA> 573 4006 1098
## 2172   1994   NL    FLO      FLA    5 115    59 51 64  <NA>  <NA> 468 3926 1043
## 2173   1994   NL    HOU      HOU    2 115    59 66 49  <NA>  <NA> 602 3955 1099
## 2174   1994   NL    LAN      LAD    1 114    55 58 56  <NA>  <NA> 532 3904 1055
## 2175   1994   NL    MON      WSN    1 114    52 74 40  <NA>  <NA> 585 4000 1111
## 2176   1994   NL    NYN      NYM    3 113    53 55 58  <NA>  <NA> 506 3869  966
## 2177   1994   NL    PHI      PHI    4 115    60 54 61  <NA>  <NA> 521 3927 1028
## 2178   1994   NL    PIT      PIT    3 114    61 53 61  <NA>  <NA> 466 3864 1001
## 2179   1994   NL    SDN      SDP    4 117    57 47 70  <NA>  <NA> 479 4068 1117
## 2180   1994   NL    SFN      SFG    2 115    60 55 60  <NA>  <NA> 504 3869  963
## 2181   1994   NL    SLN      STL    3 115    56 53 61  <NA>  <NA> 535 3902 1026
##      X2B X3B  HR  BB  SO  SB CS HBP  RA  ER  ERA CG SHO SV IPouts   HA HRA BBA
## 2154 185  20 139 438 655  69 13  39 497 478 4.31 13   4 37   2993 1005 131 351
## 2155 222  19 120 404 723  81 38  31 621 564 4.93  6   3 30   3088 1104 120 450
## 2156 178  16 120 402 715  65 54  27 660 618 5.42 11   4 21   3081 1149 150 436
## 2157 175  39 121 497 568  77 27  20 498 445 3.96 13   9 20   3034  964 115 377
## 2158 240  20 167 382 629 131 48  18 562 494 4.36 17   5 21   3056 1097  94 404
## 2159 216  25 161 520 897  46 33  34 671 609 5.38 15   1 20   3054 1139 148 449
## 2160 211  38 100 376 698 140 62  33 532 485 4.23  5   6 38   3095 1018  95 392
## 2161 239  23 103 359 635  94 30  41 688 634 5.68  6   4 29   3015 1197 153 388
## 2162 238  21  99 417 680  59 37  33 586 532 4.62 11   3 23   3108 1071 127 421
## 2163 238  16 139 530 660  55 40  31 534 492 4.34  8   2 31   3059 1045 120 398
## 2164 178  13 113 417 686  91 39  18 589 535 4.80 12   9 23   3010  979 128 510
## 2165 211  18 153 372 652  48 21  26 616 546 4.99 13   7 21   2952 1051 109 486
## 2166 198  27 124 437 730  82 35  36 697 620 5.45 10   4 26   3069 1176 157 394
## 2167 210  30 115 387 691  79 26  38 579 535 4.70 13   4 26   3075 1053 127 482
## 2168 198  18 137 377 668  48 31  22 448 407 3.57 16   8 26   3079  929  76 378
## 2169 189  26 109 364 750  69 53  27 549 508 4.47  5   5 27   3071 1054 120 392
## 2170 211  36 124 388 738 119 51  29 490 436 3.78  6   6 27   3115 1037 117 339
## 2171 206  39 125 378 761  91 53  23 638 590 5.15  4   5 28   3093 1185 120 448
```

```
## 2172 180  24  94 349 746  65 26  40 576 507 4.50  5   7 30   3045 1069 120 428
## 2173 252  25 120 394 718 124 44  43 503 454 3.97  9   6 29   3089 1043 102 367
## 2174 160  29 115 366 687  74 37  19 509 470 4.17 14   5 20   3042 1041  90 354
## 2175 246  30 108 379 669 137 36  40 454 410 3.56  4   8 46   3110  970 100 288
## 2176 164  21 117 336 807  25 26  52 526 470 4.13  7   3 35   3069 1069 117 332
## 2177 208  28  80 396 711  67 24  31 497 438 3.85  7   6 30   3073 1028  98 377
## 2178 198  23  80 349 725  53 25  22 580 518 4.64  8   2 24   3017 1094 117 370
## 2179 200  19  92 319 762  79 37  31 531 474 4.08  8   6 27   3137 1008  99 393
## 2180 159  32 123 364 719 114 40  39 500 454 3.99  2   4 33   3076 1014 122 372
## 2181 213  27 108 434 686  76 46  33 621 581 5.14  7   7 29   3054 1154 134 355
##      SOA   E  DP    FP                  name                          park
## 2154 666  57 103 0.986     Baltimore Orioles   Oriole Park at Camden Yards
## 2155 729  81 124 0.981        Boston Red Sox                Fenway Park II
## 2156 682  76 110 0.983     California Angels               Anaheim Stadium
## 2157 754  79  91 0.981     Chicago White Sox              Comiskey Park II
## 2158 666  90 119 0.980     Cleveland Indians                  Jacobs Field
## 2159 560  82  90 0.981        Detroit Tigers                 Tiger Stadium
## 2160 717  80 102 0.982    Kansas City Royals              Kauffman Stadium
## 2161 602  75  99 0.982       Minnesota Twins   Hubert H Humphrey Metrodome
## 2162 577  85 130 0.981     Milwaukee Brewers                County Stadium
## 2163 656  80 122 0.982      New York Yankees             Yankee Stadium II
## 2164 732  88 105 0.979     Oakland Athletics              Oakland Coliseum
## 2165 763  95 102 0.977      Seattle Mariners                      Kingdome
## 2166 683 106 106 0.976         Texas Rangers     The Ballpark at Arlington
## 2167 832  81 105 0.981     Toronto Blue Jays                       Skydome
## 2168 865  81  85 0.982        Atlanta Braves Atlanta-Fulton County Stadium
## 2169 717  81 110 0.982          Chicago Cubs                 Wrigley Field
## 2170 799  73  91 0.983       Cincinnati Reds            Riverfront Stadium
## 2171 703  84 117 0.981      Colorado Rockies             Mile High Stadium
## 2172 649  95 111 0.978       Florida Marlins            Joe Robbie Stadium
## 2173 739  76 110 0.983        Houston Astros                     Astrodome
## 2174 732  88 104 0.980   Los Angeles Dodgers                Dodger Stadium
## 2175 805  94  90 0.979        Montreal Expos               Stade Olympique
## 2176 640  89 112 0.980         New York Mets                  Shea Stadium
## 2177 699  94  96 0.978 Philadelphia Phillies              Veterans Stadium
## 2178 650  91 131 0.980    Pittsburgh Pirates          Three Rivers Stadium
## 2179 862 111  82 0.975      San Diego Padres           Jack Murphy Stadium
## 2180 655  68 113 0.985  San Francisco Giants              Candlestick Park
## 2181 632  80 119 0.982   St. Louis Cardinals               Busch Stadium II
##      attendance BPF PPF teamIDBR teamIDlahman45 teamIDretro
## 2154    2535359 105 104      BAL            BAL         BAL
## 2155    1775818 105 105      BOS            BOS         BOS
## 2156    1512622 101 101      CAL            CAL         CAL
## 2157    1697398  99  98      CHW            CHA         CHA
## 2158    1995174  99  97      CLE            CLE         CLE
## 2159    1184783 101 101      DET            DET         DET
## 2160    1400494 104 104      KCR            KCA         KCA
## 2161    1398565 100 102      MIN            MIN         MIN
## 2162    1268399 104 105      MIL            MIL         MIL
```

```
## 2163    1675556  97  96      NYY            NYA         NYA
## 2164    1242692  91  92      OAK            OAK         OAK
## 2165    1104206 102 102      SEA            SEA         SEA
## 2166    2503198 100 101      TEX            TEX         TEX
## 2167    2907933 100 100      TOR            TOR         TOR
## 2168    2539240 102 100      ATL            ATL         ATL
## 2169    1845208  99  99      CHC            CHN         CHN
## 2170    1897681  99  99      CIN            CIN         CIN
## 2171    3281511 117 118      COL            COL         COL
## 2172    1937467 102 103      FLA            FLO         FLO
## 2173    1561136  95  94      HOU            HOU         HOU
## 2174    2279355  94  94      LAD            LAN         LAN
## 2175    1276250 101 101      MON            MON         MON
## 2176    1151471  99  99      NYM            NYN         NYN
## 2177    2290971 102 102      PHI            PHI         PHI
## 2178    1222520 101 102      PIT            PIT         PIT
## 2179     953857  97  98      SDP            SDN         SDN
## 2180    1704608  94  94      SFG            SFN         SFN
## 2181    1866544  98  99      STL            SLN         SLN
```

Missing data for LgWin is only associated with 1994. In 1994, MLB players went on strike and it cut the season short.

```
# Imputation of LgWin missing data with "N"
Teams$LgWin[is.na(Teams$LgWin)] <- "N"
```

Imputation of LgWin's missing values is done with “N” given the unique historical context, as there were no league winners that year.

```
# Check if column has any missing values
any(is.na(Teams$LgWin))
```

```
## [1] FALSE
```

This shows that the imputation method worked.

```
# Subset rows with missing WSWin values
missing_wsw <- Teams[is.na(Teams$WSWin), ]

# Count missing WSWin by yearID in table form
table(missing_wsw$yearID)
```

```
##
## 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1890 1891
##    9   11    9    8   13    8    6    6    8    8    8   14   16   12    8   17
## 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1904 1914 1915 1994
##   12   12   12   12   12   12   12   12    8   16   16   16    8    8   28
```

Missing data for WSWin is associated with some historical context. First, WSWin was not tracked before 1903 since World Series was not yet established. Second, no World Series was held in 1904 due to disputes between the leagues. Third, in 1994, MLB players went on strike and no World Series was held. Lastly, data missing for 1914 and 1915 could be due to inconsistencies in data tracking.

```
# Imputation of WSWin's missing data with "N"
Teams$WSWin[is.na(Teams$WSWin) & Teams$yearID < 1903] <- "N"
Teams$WSWin[is.na(Teams$WSWin) & Teams$yearID == 1904] <- "N"
Teams$WSWin[is.na(Teams$WSWin) & Teams$yearID == 1994] <- "N"
Teams$WSWin[is.na(Teams$WSWin) & Teams$yearID %in% c(1914, 1915)] <- "N"
```

Imputation of missing values is done with “N” given the historical context described above.

```
# Check if column has any missing values
any(is.na(Teams$WSWin))
```

```
## [1] FALSE
```

This shows that the imputation method worked.

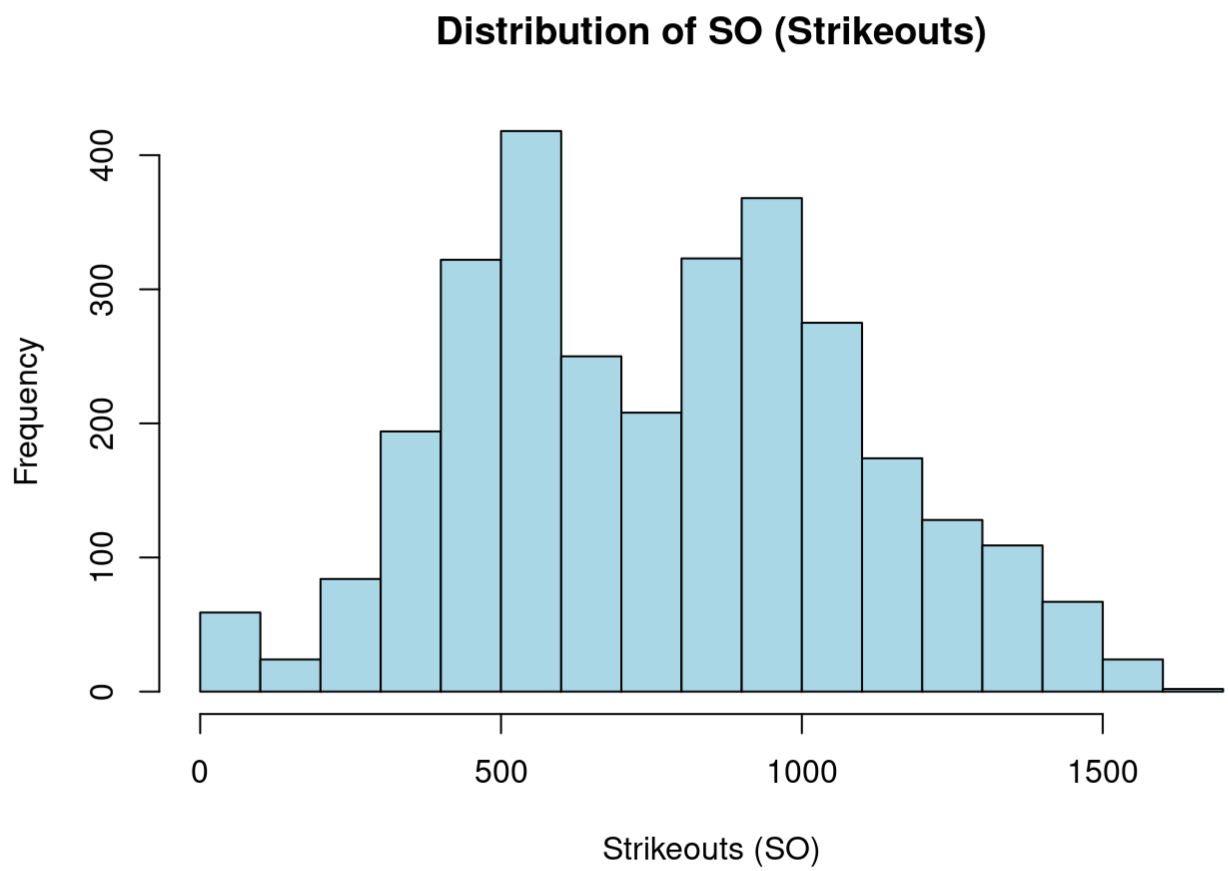
```
# Subset rows with missing SO values
missing_so <- Teams[is.na(Teams$SO), ]

# Count missing SO by yearID in table form
table(missing_so$yearID)
```

```
##
## 1911 1912
##    8    8
```

There are only two years with missing values for SO, this could be due to inconsistencies in tracking the data.

```
# Histogram to check the distribution of SO
hist(Teams$SO, main = "Distribution of SO (Strikeouts)", xlab = "Strikeouts (SO)", c
ol = "lightblue", breaks = 20)
```



The distribution of SO is roughly normally distributed so the best imputation method is mean.

```
# Imputation of missing SO values with the mean SO for 1911 and 1912
Teams$SO[is.na(Teams$SO) & Teams$yearID == 1911] <- mean(Teams$SO[Teams$yearID == 19
11], na.rm = TRUE)
Teams$SO[is.na(Teams$SO) & Teams$yearID == 1912] <- mean(Teams$SO[Teams$yearID == 19
12], na.rm = TRUE)
```

Imputation of missing data is done with the mean given that the data is roughly normally distributed.

```
# Check if column has any missing values
any(is.na(Teams$SO))
```

```
## [1] FALSE
```

This shows that the imputation method worked.

```
# Subset rows with missing park values
missing_park <- Teams[is.na(Teams$park), ]

# Count missing park by yearID
table(missing_park$yearID)
```

```
##
## 1884 1890 1914 1915
##   12    8    7    7
```

Missing data for park could be due to inconsistencies in tracking the data point.

```
# Imputation of missing park values with "Unknown"
Teams$park[is.na(Teams$park)] <- "Unknown"
```

Since the parks are unknown, imputation is done with “unknown”.

```
# Check if column has any missing values
any(is.na(Teams$park))
```

```
## [1] FALSE
```

This shows that the imputation method worked.

```
# Subset rows with missing SB values
missing_sb <- Teams[is.na(Teams$SB), ]

# Count missing SB by yearID
table(missing_sb$yearID)
```

```
##
## 1872 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885
##    2    8    6    6    8    8    8   14   16   33   16
```

SB missing data is associated with years prior to 1886, which is consistent with when MLB began tracking SB.

```
# Filter dataset to include only years < 1886
Teams_SB <- Teams[!(Teams$yearID < 1886 & is.na(Teams$SB)), ]
```

The best method is to filter out the NA values in SB, since imputation with 0, mean, median, or mode would produce inaccurate data. This was saved to a new df so that the Teams df could continue to be analyzed. A complete update to Teams df will be made at the end.

```
# Subset rows with missing attendance values
missing_att <- Teams[is.na(Teams$attendance), ]

# Count missing attendance by yearID
table(missing_att$yearID)
```

```
##
## 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886
##    9   11    9    8   13    8    6    6    8    8    8   14   16   33   16   16
## 1887 1888 1889 1890 1891 1914 1915
##   16   16   16   17    9    8    8
```

Missing values for attendance is consistent with when MLB began tracking attendance more accurately, where before 1891, it was not. Interesting enough 1914 and 1915 are years with missing values and could be due to inconsistencies in data tracking.

```
# Filter dataset to exclude rows with missing attendance in years < 1892, 1914, and
1915
Teams_attendance <- Teams[!(Teams$yearID < 1892 & is.na(Teams$attendance)) &
                                !(Teams$yearID %in% c(1914, 1915) & is.na(Teams$atte
ndance)), ]
```

Excluding the years with inconsistent data instead of imputation with 0, mean, median, or mode since the latter would produce inaccurate data. This was saved to a new df so that the Teams df could continue to be analyzed. A complete update to Teams df will be made at the end.

```
# Subset rows with missing CS values
missing_cs <- Teams[is.na(Teams$CS), ]

# Count missing CS by yearID
table(missing_cs$yearID)
```

```
##
## 1872 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889
##    1    1    8    6    6    8    8    8   14   16   33   16   16   16   16   16
## 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905
##   25   17   12   12   12   12   12   12   12   12    8   16   16   16   16   16
## 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1926 1927
##   16   16   16   16   16   16   16   16   16    8   16   16   16   16    8    8
## 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943
##    8    8    8    8    8    8    8    8    8    8    8    8    8    8    8    8
## 1944 1945 1946 1947 1948 1949 1950
##    8    8    8    8    8    8    8
```

CS missing data is associated with years prior to 1951, which is consistent with when MLB began tracking CS.

```
# Filter out rows where CS is NA
Teams_CS <- Teams %>% filter(!is.na(CS))
```

The best method is to filter out the NA values in CS, since imputation with 0, mean, median, or mode would produce inaccurate data. This was saved to a new df so that the Teams df could continue to be analyzed. A complete update to Teams df will be made at the end.

```
# Subset rows with missing HBP values
missing_hbp <- Teams[is.na(Teams$HBP), ]

# Count missing HBP by yearID
table(missing_hbp$yearID)
```

```
##
## 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886
##    9   11    9    8   13    8    6    6    8    8    8   14   16   20    8    8
## 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926
##   16   16   16   24   24   16   16   16   16   16   16   16   16   16   16   16
## 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942
##   16   16   16   16   16   16   16   16   16   16   16   16   16   16   16   16
## 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958
##   16   16   16   16   16   16   16   16   16   16   16   16   16   16   16   16
## 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969
##   16   16   18   20   20   20   20   20   20   20   24
```

HBP missing data is associated with years prior to 1970, which is consistent with when MLB began tracking HBP.

```
# Filter out rows where HBP is NA
Teams_HBP <- Teams %>% filter(!is.na(HBP))
```

The best method is to filter out the NA values in HBP, since imputation with 0, mean, median, or mode would produce accurate data. This was saved to a new df so that the Teams df could continue to be analyzed. A complete update to Teams df will be made at the end.

```
# Create a dataset with all filters above
Teams_filtered <- Teams %>%
  filter(!(yearID < 1892 & is.na(attendance)) &
         !(yearID %in% c(1914, 1915) & is.na(attendance)) &
         !(yearID < 1886 & is.na(SB)) &
         !is.na(CS) &
         !is.na(HBP))
```

```
# Check if column has any missing values
any(is.na(Teams_filtered))
```

```
## [1] FALSE
```

All the removing of columns, excluding of missing values and imputations worked.

```
summary(Teams_filtered)
```

```
##      yearID      lgID          teamID         franchID          Rank      
##  Min.   :1970   AA:  0   ATL    :  54   ANA    :  54   Min.   :1.000  
##  1st Qu.:1985   AL:753   BAL    :  54   ATL    :  54   1st Qu.:2.000  
##  Median :1998   FL:  0   BOS    :  54   BAL    :  54   Median :3.000  
##  Mean   :1998   NA:  0   CHA    :  54   BOS    :  54   Mean   :3.263  
##  3rd Qu.:2011   NL:751   CHN    :  54   CHC    :  54   3rd Qu.:5.000  
##  Max.   :2023   PL:  0   CIN    :  54   CHW    :  54   Max.   :7.000  
##                 UA:  0   (Other):1180   (Other):1180                  
##        G             Ghome             W                L         
##  Min.   : 58.0   Min.   :24.00   Min.   : 19.00   Min.   : 17.00  
##  1st Qu.:162.0   1st Qu.:81.00   1st Qu.: 71.00   1st Qu.: 71.00  
##  Median :162.0   Median :81.00   Median : 80.00   Median : 79.00  
##  Mean   :157.6   Mean   :78.79   Mean   : 78.77   Mean   : 78.77  
##  3rd Qu.:162.0   3rd Qu.:81.00   3rd Qu.: 89.00   3rd Qu.: 88.00  
##  Max.   :164.0   Max.   :84.00   Max.   :116.00   Max.   :119.00  
##                                                                   
##     LgWin              WSWin                 R                AB      
##  Length:1504        Length:1504        Min.   : 219.0   Min.   :1752  
##  Class :character   Class :character   1st Qu.: 649.0   1st Qu.:5444  
##  Mode  :character   Mode  :character   Median : 710.0   Median :5508  
##                                        Mean   : 704.4   Mean   :5373  
##                                        3rd Qu.: 772.0   3rd Qu.:5572  
##                                        Max.   :1009.0   Max.   :5781  
##                                                                       
##        H             X2B             X3B             HR              BB       
##  Min.   : 390   Min.   : 73.0   Min.   : 3.0   Min.   : 32.0   Min.   :147.0  
##  1st Qu.:1351   1st Qu.:234.0   1st Qu.:24.0   1st Qu.:120.0   1st Qu.:471.0  
##  Median :1415   Median :265.0   Median :30.0   Median :150.0   Median :519.0  
##  Mean   :1391   Mean   :258.7   Mean   :31.3   Mean   :151.1   Mean   :515.8  
##  3rd Qu.:1479   3rd Qu.:289.0   3rd Qu.:38.0   3rd Qu.:181.0   3rd Qu.:569.0  
##  Max.   :1684   Max.   :376.0   Max.   :79.0   Max.   :307.0   Max.   :775.0  
##                                                                               
##        SO               SB               CS              HBP        
##  Min.   : 379.0   Min.   : 14.0   Min.   :  3.00   Min.   :  7.00  
##  1st Qu.: 858.0   1st Qu.: 72.0   1st Qu.: 33.00   1st Qu.: 31.00  
##  Median : 991.5   Median : 97.0   Median : 44.00   Median : 43.00  
##  Mean   :1015.0   Mean   :100.7   Mean   : 45.09   Mean   : 45.44  
##  3rd Qu.:1164.0   3rd Qu.:125.0   3rd Qu.: 55.00   3rd Qu.: 58.00  
##  Max.   :1654.0   Max.   :341.0   Max.   :123.00   Max.   :112.00  
##                                                                    
##        RA               ER              ERA              CG       
##  Min.   : 209.0   Min.   : 181.0   Min.   :2.530   Min.   : 0.00  
##  1st Qu.: 647.0   1st Qu.: 583.0   1st Qu.:3.690   1st Qu.: 3.00  
##  Median : 708.0   Median : 642.0   Median :4.040   Median : 8.00  
##  Mean   : 704.4   Mean   : 639.6   Mean   :4.096   Mean   :14.98  
##  3rd Qu.: 775.2   3rd Qu.: 709.2   3rd Qu.:4.490   3rd Qu.:23.00  
##  Max.   :1103.0   Max.   :1015.0   Max.   :6.380   Max.   :94.00  
## 
```

```
##       SHO              SV           IPouts          HA            HRA       
##  Min.   : 0.000   Min.   : 6.00   Min.   :1419   Min.   : 376   Min.   : 40.0  
##  1st Qu.: 6.000   1st Qu.:32.00   1st Qu.:4299   1st Qu.:1348   1st Qu.:124.0  
##  Median : 9.000   Median :38.00   Median :4333   Median :1416   Median :152.0  
##  Mean   : 9.307   Mean   :37.63   Mean   :4224   Mean   :1391   Mean   :151.1  
##  3rd Qu.:12.000   3rd Qu.:44.00   3rd Qu.:4367   3rd Qu.:1484   3rd Qu.:178.0  
##  Max.   :24.000   Max.   :68.00   Max.   :4485   Max.   :1734   Max.   :305.0  
##                                                                                
##       BBA             SOA                E                DP       
##  Min.   :145.0   Min.   : 388.0   Min.   : 20.0   Min.   : 33.0  
##  1st Qu.:474.0   1st Qu.: 859.0   1st Qu.: 94.0   1st Qu.:134.0  
##  Median :519.0   Median : 997.5   Median :110.0   Median :147.0  
##  Mean   :515.8   Mean   :1015.0   Mean   :112.1   Mean   :149.1  
##  3rd Qu.:569.0   3rd Qu.:1173.2   3rd Qu.:131.0   3rd Qu.:161.0  
##  Max.   :784.0   Max.   :1687.0   Max.   :199.0   Max.   :460.0  
##                                                                  
##        FP             name               park             attendance     
##  Min.   :0.9680   Length:1504        Length:1504        Min.   :      0  
##  1st Qu.:0.9790   Class :character   Class :character   1st Qu.:1422570  
##  Median :0.9820   Mode  :character   Mode  :character   Median :1979127  
##  Mean   :0.9813                                         Mean   :2016429  
##  3rd Qu.:0.9840                                         3rd Qu.:2590766  
##  Max.   :0.9910                                         Max.   :4483350  
##                                                                          
##       BPF             PPF          teamIDBR         teamIDlahman45    
##  Min.   : 88.0   Min.   : 88.0   Length:1504        Length:1504       
##  1st Qu.: 97.0   1st Qu.: 97.0   Class :character   Class :character  
##  Median :100.0   Median :100.0   Mode  :character   Mode  :character  
##  Mean   :100.2   Mean   :100.2                                        
##  3rd Qu.:103.0   3rd Qu.:103.0                                        
##  Max.   :129.0   Max.   :129.0                                        
##                                                                       
##  teamIDretro       
##  Length:1504       
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
## 
```

Feature Engineering

```
# Creating a column to hold the values for run differential
Teams_filtered$run_differential <- Teams_filtered$R - Teams_filtered$RA
```

Run differential is calculated by subtracting runs allowed from runs scored. Run differential is evaluated as positive if a team scores more runs than it allows and negative if a team allows more runs than it scores. This calculation can be used to predict the expected win total for a team.

```
# Creating a column to hold the values for winning percentage
Teams_filtered$winning_percentage <- Teams_filtered$W / Teams_filtered$G
```

In order to compare teams expected winning percentage to the actual winning percentage, the winning percentage was calculated by dividing the number of games played by number of games won.

```
# Creating a column to hold the values for Pythagorean expectation
Teams_filtered$pythagorean_expectation <- Teams_filtered$R^1.83 / (Teams_filtered$R^
1.83 + Teams_filtered$RA^1.83)
```

Created by Bill James in order to evaluate a teams performance by comparing the expected winning percentage to the actual winning percentage. This can be calculated using the formula below: (Runs Scored^{1}.83)/((Runs Scored^{1}.83)+ (Runs Allowed^{1}.83))

```
# Creating a column to hold the values for under/over performance
Teams_filtered$performance <- ifelse(Teams_filtered$winning_percentage > Teams_filte
red$pythagorean_expectation,
                                     "Overperformed", "Underperformed")
```

Comparison of expected winning percentage to the actual winning percentage can be done by classifying teams as over or under performing. Teams who overperformed are determined if their winning percentage is higher than the Pythagorean expectation (expected winning percentage). Teams who underperformed are determined if their winning percentage is lower than the Pythagorean expectation (expected winning percentage).

```
# Creating a column to hold the values for historical success based on World Series
or league wins
Teams_filtered$historical_success <- ifelse(Teams_filtered$WSWin == "Y", "Champion",
                                            ifelse(Teams_filtered$LgWin == "Y", "League Winn
er", "Non-Winner"))
```

Historical success feature was created to delineate Champions, for winning the World Series, League Winner, for winning the league by not World Series, and Non-Winner, for teams who did not win either. The purpose of the feature is to analyze success levels across teams by categorizing them, for comparison of other variables such as RS or ERA to see what differences lie between levels, and this could support the predictive model to be developed in the next part of the assignment.

Print the data types of each column (e.g., use the ’str() function in RStudio).

```
str(Teams_filtered)
```

```
## 'data.frame':    1504 obs. of  49 variables:
##  $ yearID              : int  1970 1970 1970 1970 1970 1970 1970 1970 1970 197
0 ...
##  $ lgID                : Factor w/ 7 levels "AA","AL","FL",..: 2 2 2 2 2 2 2 2
2 2 ...
##  $ teamID              : Factor w/ 149 levels "ALT","ANA","ARI",..: 5 16 30 33
45 52 66 79 83 93 ...
##  $ franchID            : Factor w/ 120 levels "ALT","ANA","ARI",..: 6 14 2 29
32 41 54 63 62 75 ...
##  $ Rank                : int  1 3 3 6 5 4 4 1 4 2 ...
##  $ G                   : int  162 162 162 162 162 162 162 162 163 163 ...
##  $ Ghome               : num  81 81 81 84 81 81 79 81 81 81 ...
##  $ W                   : int  108 87 86 56 76 79 65 98 65 93 ...
##  $ L                   : int  54 75 76 106 86 83 97 64 97 69 ...
##  $ LgWin               : chr  "Y" "N" "N" "N" ...
##  $ WSWin               : chr  "Y" "N" "N" "N" ...
##  $ R                   : int  792 786 631 633 649 666 611 744 613 680 ...
##  $ AB                  : int  5545 5535 5532 5514 5463 5377 5503 5483 5395 549
2 ...
##  $ H                   : int  1424 1450 1391 1394 1358 1282 1341 1438 1305 138
1 ...
##  $ X2B                 : int  213 252 197 192 197 207 202 230 202 208 ...
##  $ X3B                 : int  25 28 40 20 23 38 41 41 24 41 ...
##  $ HR                  : int  179 203 114 123 183 148 97 153 126 111 ...
##  $ BB                  : int  717 594 447 477 503 656 514 501 592 588 ...
##  $ SO                  : num  952 855 922 872 909 825 958 905 985 808 ...
##  $ SB                  : int  84 50 69 53 25 29 97 57 91 105 ...
##  $ CS                  : int  39 48 27 33 36 30 53 52 73 61 ...
##  $ HBP                 : int  44 40 29 42 37 34 21 42 36 25 ...
##  $ RA                  : int  574 722 630 822 675 731 705 605 751 612 ...
##  $ ER                  : int  517 622 566 722 630 658 615 520 676 530 ...
##  $ ERA                 : num  3.15 3.87 3.48 4.54 3.91 4.09 3.78 3.23 4.21 3.2
4 ...
##  $ CG                  : int  60 38 21 20 34 33 30 26 31 36 ...
##  $ SHO                 : int  12 8 10 6 8 9 11 12 2 6 ...
##  $ SV                  : int  31 44 49 30 35 39 25 58 27 49 ...
##  $ IPouts              : int  4436 4339 4387 4291 4354 4342 4391 4345 4340 441
5 ...
##  $ HA                  : int  1317 1391 1280 1554 1333 1443 1346 1329 1397 138
6 ...
##  $ HRA                 : int  139 156 154 164 163 153 138 130 146 130 ...
##  $ BBA                 : int  469 594 559 556 689 623 641 486 587 451 ...
##  $ SOA                 : int  941 1003 922 762 1076 1045 915 940 895 777 ...
##  $ E                   : int  117 156 127 165 133 133 152 123 136 130 ...
##  $ DP                  : int  148 131 169 187 168 142 162 130 142 146 ...
##  $ FP                  : num  0.981 0.974 0.98 0.975 0.979 0.978 0.976 0.98 0.
978 0.98 ...
##  $ name                : chr  "Baltimore Orioles" "Boston Red Sox" "California
```

```
Angels" "Chicago White Sox" ...
##  $ park                   : chr  "Memorial Stadium" "Fenway Park II" "Anaheim Sta
dium" "Comiskey Park" ...
##  $ attendance             : int  1057069 1595278 1077741 495355 729752 1501293 69
3047 1261887 933690 1136879 ...
##  $ BPF                    : int  101 108 96 101 104 101 99 103 100 95 ...
##  $ PPF                    : int  98 107 97 102 105 101 100 102 101 95 ...
##  $ teamIDBR               : chr  "BAL" "BOS" "CAL" "CHW" ...
##  $ teamIDlahman45         : chr  "BAL" "BOS" "CAL" "CHA" ...
##  $ teamIDretro            : chr  "BAL" "BOS" "CAL" "CHA" ...
##  $ run_differential       : int  218 64 1 -189 -26 -65 -94 139 -138 68 ...
##  $ winning_percentage     : num  0.667 0.537 0.531 0.346 0.469 ...
##  $ pythagorean_expectation: num  0.643 0.539 0.501 0.383 0.482 ...
##  $ performance            : chr  "Overperformed" "Underperformed" "Overperformed"
"Underperformed" ...
##  $ historical_success     : chr  "Champion" "Non-Winner" "Non-Winner" "Non-Winne
r" ...
```

Show summary of the columns (e.g., use the 'summary()' function in RStudio).

```
summary(Teams_filtered)
```

```
##      yearID     lgID         teamID        franchID         Rank
##  Min.   :1970   AA:  0   ATL     :  54   ANA     :  54   Min.   :1.000
##  1st Qu.:1985   AL:753   BAL     :  54   ATL     :  54   1st Qu.:2.000
##  Median :1998   FL:  0   BOS     :  54   BAL     :  54   Median :3.000
##  Mean   :1998   NA:  0   CHA     :  54   BOS     :  54   Mean   :3.263
##  3rd Qu.:2011   NL:751   CHN     :  54   CHC     :  54   3rd Qu.:5.000
##  Max.   :2023   PL:  0   CIN     :  54   CHW     :  54   Max.   :7.000
##                 UA:  0   (Other):1180   (Other):1180
##        G              Ghome              W                L
##  Min.   : 58.0   Min.   :24.00   Min.   : 19.00   Min.   : 17.00
##  1st Qu.:162.0   1st Qu.:81.00   1st Qu.: 71.00   1st Qu.: 71.00
##  Median :162.0   Median :81.00   Median : 80.00   Median : 79.00
##  Mean   :157.6   Mean   :78.79   Mean   : 78.77   Mean   : 78.77
##  3rd Qu.:162.0   3rd Qu.:81.00   3rd Qu.: 89.00   3rd Qu.: 88.00
##  Max.   :164.0   Max.   :84.00   Max.   :116.00   Max.   :119.00
##
##     LgWin              WSWin                 R                AB
##  Length:1504        Length:1504        Min.   : 219.0   Min.   :1752
##  Class :character   Class :character   1st Qu.: 649.0   1st Qu.:5444
##  Mode  :character   Mode  :character   Median : 710.0   Median :5508
##                                        Mean   : 704.4   Mean   :5373
##                                        3rd Qu.: 772.0   3rd Qu.:5572
##                                        Max.   :1009.0   Max.   :5781
##
##        H             X2B             X3B             HR              BB
##  Min.   : 390   Min.   : 73.0   Min.   : 3.0   Min.   : 32.0   Min.   :147.0
##  1st Qu.:1351   1st Qu.:234.0   1st Qu.:24.0   1st Qu.:120.0   1st Qu.:471.0
##  Median :1415   Median :265.0   Median :30.0   Median :150.0   Median :519.0
##  Mean   :1391   Mean   :258.7   Mean   :31.3   Mean   :151.1   Mean   :515.8
##  3rd Qu.:1479   3rd Qu.:289.0   3rd Qu.:38.0   3rd Qu.:181.0   3rd Qu.:569.0
##  Max.   :1684   Max.   :376.0   Max.   :79.0   Max.   :307.0   Max.   :775.0
##
##        SO               SB               CS               HBP
##  Min.   : 379.0   Min.   : 14.0   Min.   :  3.00   Min.   :  7.00
##  1st Qu.: 858.0   1st Qu.: 72.0   1st Qu.: 33.00   1st Qu.: 31.00
##  Median : 991.5   Median : 97.0   Median : 44.00   Median : 43.00
##  Mean   :1015.0   Mean   :100.7   Mean   : 45.09   Mean   : 45.44
##  3rd Qu.:1164.0   3rd Qu.:125.0   3rd Qu.: 55.00   3rd Qu.: 58.00
##  Max.   :1654.0   Max.   :341.0   Max.   :123.00   Max.   :112.00
##
##        RA               ER              ERA              CG
##  Min.   : 209.0   Min.   : 181.0   Min.   :2.530   Min.   : 0.00
##  1st Qu.: 647.0   1st Qu.: 583.0   1st Qu.:3.690   1st Qu.: 3.00
##  Median : 708.0   Median : 642.0   Median :4.040   Median : 8.00
##  Mean   : 704.4   Mean   : 639.6   Mean   :4.096   Mean   :14.98
##  3rd Qu.: 775.2   3rd Qu.: 709.2   3rd Qu.:4.490   3rd Qu.:23.00
##  Max.   :1103.0   Max.   :1015.0   Max.   :6.380   Max.   :94.00
##
```

```
##       SHO               SV            IPouts          HA            HRA       
##  Min.   : 0.000   Min.   : 6.00   Min.   :1419   Min.   : 376   Min.   : 40.0  
##  1st Qu.: 6.000   1st Qu.:32.00   1st Qu.:4299   1st Qu.:1348   1st Qu.:124.0  
##  Median : 9.000   Median :38.00   Median :4333   Median :1416   Median :152.0  
##  Mean   : 9.307   Mean   :37.63   Mean   :4224   Mean   :1391   Mean   :151.1  
##  3rd Qu.:12.000   3rd Qu.:44.00   3rd Qu.:4367   3rd Qu.:1484   3rd Qu.:178.0  
##  Max.   :24.000   Max.   :68.00   Max.   :4485   Max.   :1734   Max.   :305.0  
##                                                                                
##       BBA              SOA               E                DP       
##  Min.   :145.0   Min.   : 388.0   Min.   : 20.0   Min.   : 33.0  
##  1st Qu.:474.0   1st Qu.: 859.0   1st Qu.: 94.0   1st Qu.:134.0  
##  Median :519.0   Median : 997.5   Median :110.0   Median :147.0  
##  Mean   :515.8   Mean   :1015.0   Mean   :112.1   Mean   :149.1  
##  3rd Qu.:569.0   3rd Qu.:1173.2   3rd Qu.:131.0   3rd Qu.:161.0  
##  Max.   :784.0   Max.   :1687.0   Max.   :199.0   Max.   :460.0  
##                                                                  
##        FP              name               park             attendance     
##  Min.   :0.9680   Length:1504        Length:1504        Min.   :      0  
##  1st Qu.:0.9790   Class :character   Class :character   1st Qu.:1422570  
##  Median :0.9820   Mode  :character   Mode  :character   Median :1979127  
##  Mean   :0.9813                                         Mean   :2016429  
##  3rd Qu.:0.9840                                         3rd Qu.:2590766  
##  Max.   :0.9910                                         Max.   :4483350  
##                                                                           
##       BPF             PPF           teamIDBR         teamIDlahman45    
##  Min.   : 88.0   Min.   : 88.0   Length:1504        Length:1504       
##  1st Qu.: 97.0   1st Qu.: 97.0   Class :character   Class :character  
##  Median :100.0   Median :100.0   Mode  :character   Mode  :character  
##  Mean   :100.2   Mean   :100.2                                        
##  3rd Qu.:103.0   3rd Qu.:103.0                                        
##  Max.   :129.0   Max.   :129.0                                        
##                                                                       
##  teamIDretro        run_differential winning_percentage pythagorean_expectation
##  Length:1504        Min.   :-339.0   Min.   :0.2654     Min.   :0.3023         
##  Class :character   1st Qu.: -73.0   1st Qu.:0.4506     1st Qu.:0.4515         
##  Mode  :character   Median :   0.5   Median :0.5000     Median :0.5003         
##                     Mean   :   0.0   Mean   :0.4998     Mean   :0.5003         
##                     3rd Qu.:  73.0   3rd Qu.:0.5521     3rd Qu.:0.5476         
##                     Max.   : 334.0   Max.   :0.7167     Max.   :0.7146         
##                                                                                 
##  performance        historical_success
##  Length:1504        Length:1504       
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
##
```

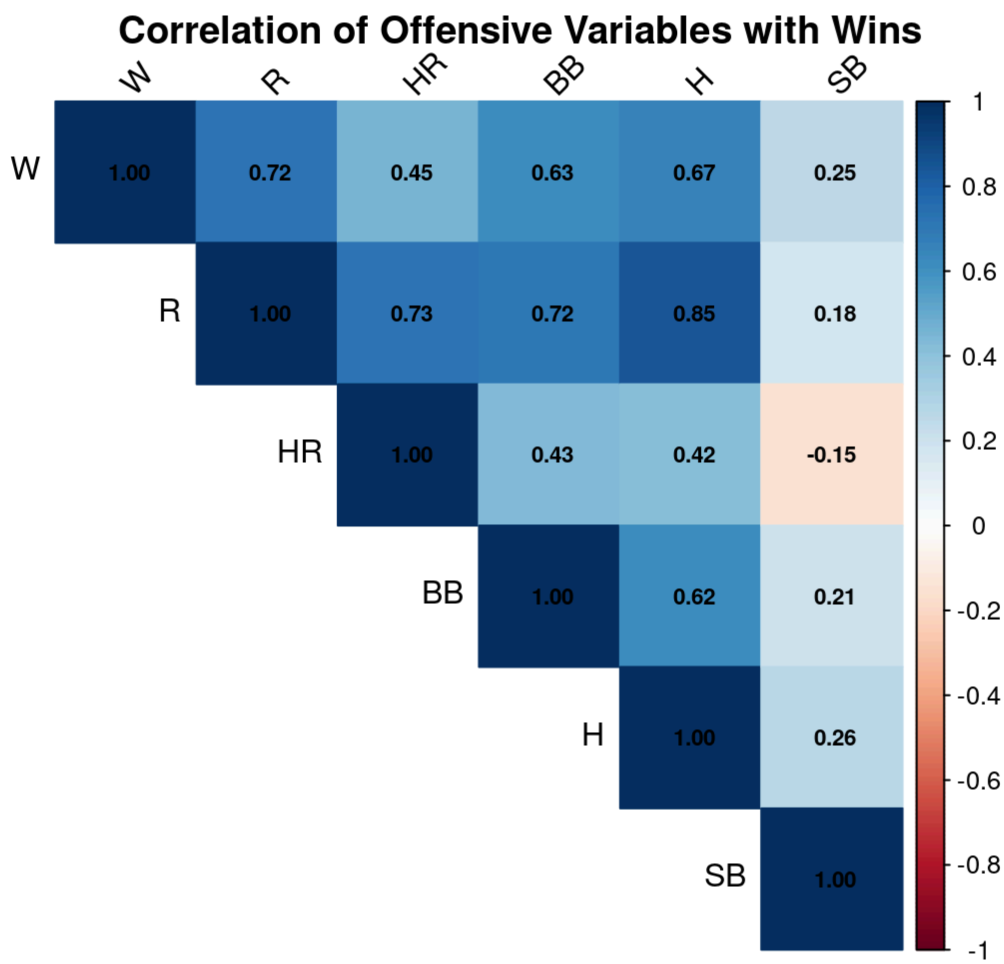
Exploratory Analysis

Identifying which model to use by visualizing linear relationships.

```
# Create an object to hold offensive performance variables with Wins
offensive_vars <- Teams_filtered[, c("W", "R", "HR", "BB", "H", "SB")]
```

```
# Calculate correlation matrices
cor_offensive <- cor(offensive_vars, use = "complete.obs")
```

```
# Create correlation plot for Offensive Performance Variables
corrplot(cor_offensive, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
         title = "Correlation of Offensive Variables with Wins",
         mar = c(0, 0, 1, 0))
```

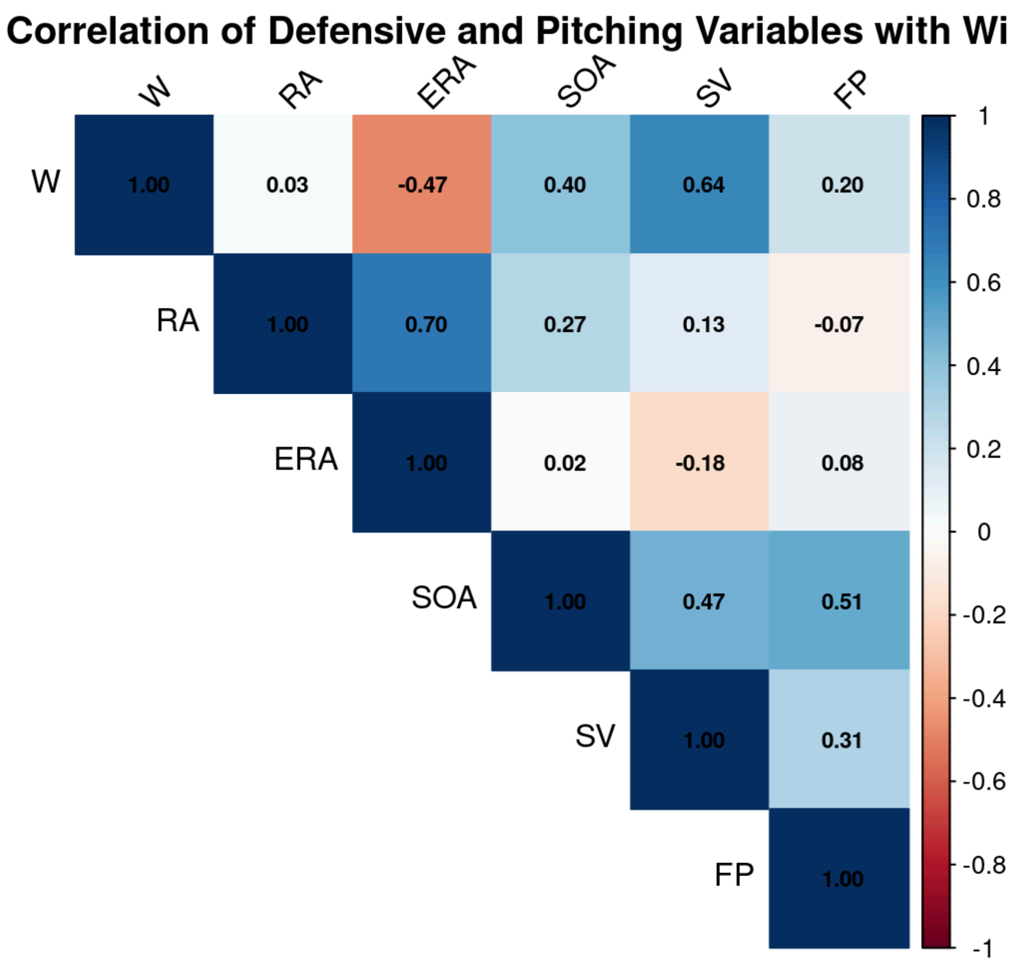


The correlation matrix above illustrates the offensive variables correlated with Wins. The variables most associated with wins are: R - runs scored (0.72), H - hits (0.67), and BB - walks (0.63).

```
# Create object to hold defensive and pitching variables with Wins
defensive_vars <- Teams_filtered[, c("W", "RA", "ERA", "SOA", "SV", "FP")]
```

```
# Calculate correlation matrices
cor_defensive <- cor(defensive_vars, use = "complete.obs")
```

```
# Create correlation plot for Defensive and Pitching Variables
corrplot(cor_defensive, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
         title = "Correlation of Defensive and Pitching Variables with Wins",
         mar = c(0, 0, 1, 0))
```



The correlation matrix above illustrates the defensive variables correlated with Wins. The variables most associated with wins are: SOA - strikeouts by pitchers (0.40) and SV - saves (0.64). These are all variables that will be taken into account for developing the predictive model for MLB wins for Part 2.Surprisingly, RA - runs allowed, has a very low correlation (0.03) with runs and negatively correlated with ERA (-0.47).

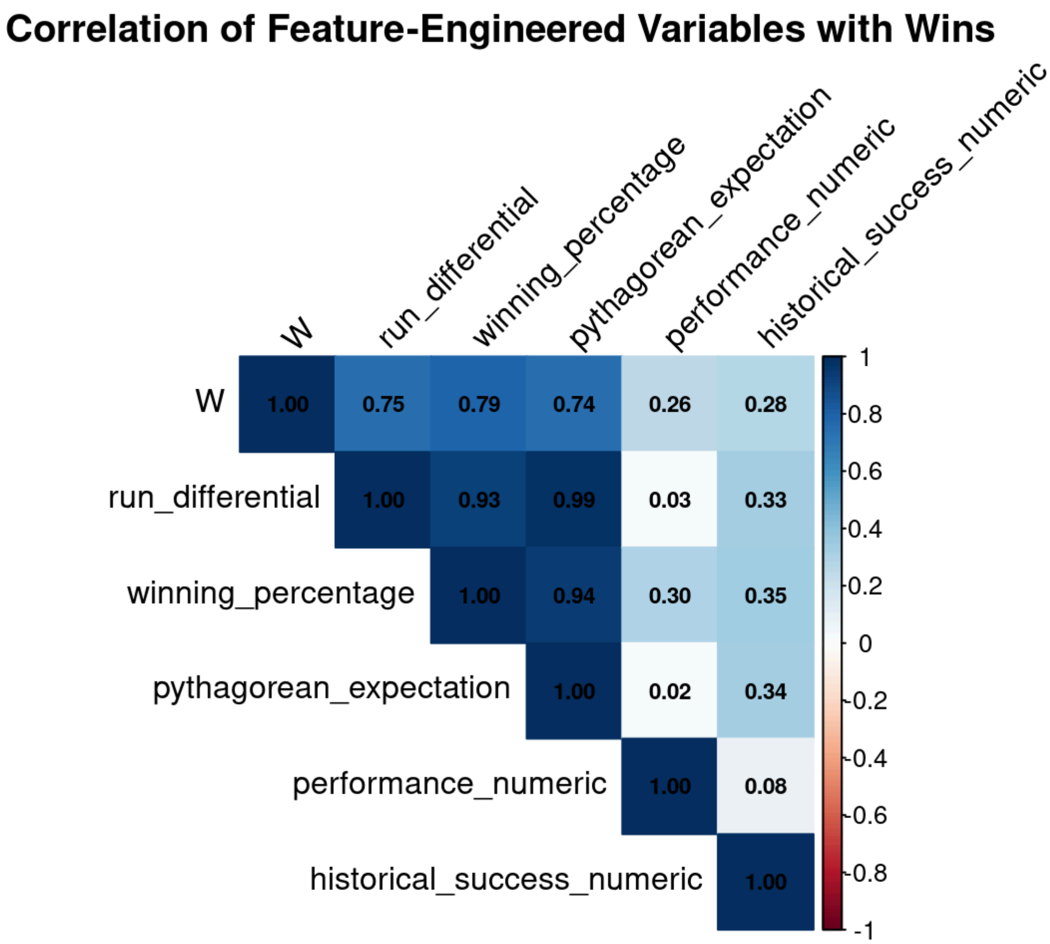
```
# Convert 'performance' and 'historical_success' to numeric
Teams_filtered$performance_numeric <- ifelse(Teams_filtered$performance == "Overperf
ormed", 1, 0)
Teams_filtered$historical_success <- ifelse(Teams_filtered$WSWin == "Y", "Champion",
                                    ifelse(Teams_filtered$LgWin == "Y", "League Winn
er", "Non-Winner"))

# Convert historical_success to numeric encoding
Teams_filtered$historical_success_numeric <- ifelse(Teams_filtered$historical_succes
s == "Champion", 2,
                                           ifelse(Teams_filtered$historical_success
== "League Winner", 1, 0))
```

```
# Subset data with Wins and feature-engineered variables
feature_vars <- Teams_filtered[, c("W", "run_differential", "winning_percentage",
                                    "pythagorean_expectation", "performance_numeri
c",
                                    "historical_success_numeric")]
```

```
# Calculate correlation matrix with Wins included
cor_matrix <- cor(feature_vars, use = "complete.obs")
```

```
# Create a plot the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
         title = "Correlation of Feature-Engineered Variables with Wins",
         mar = c(0, 0, 1, 0))
```

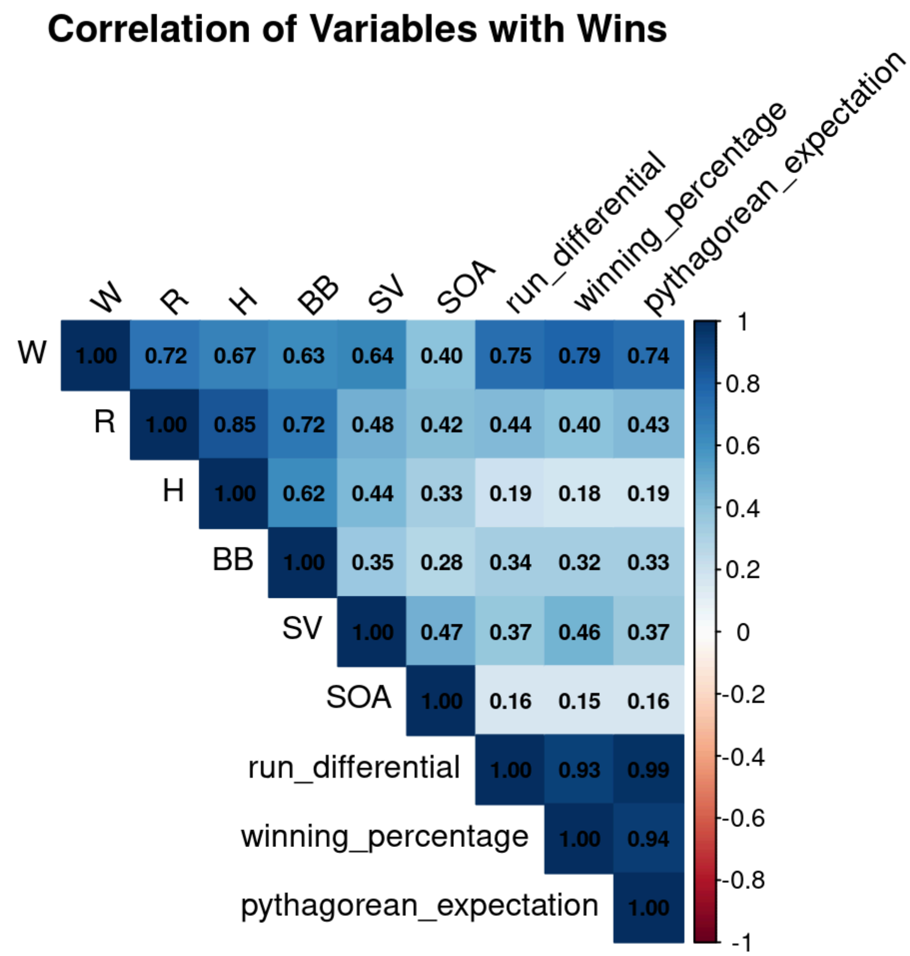


The correlation matrix above illustrates the feature engineered variables correlated with Wins. The variables most associated with wins are: run differential (0.75), winning percentage (0.79), and pythagorean expectation (0.74). These are all variables that will be taken into account for developing the predictive model for MLB wins.

Correlation Matrix of Selected Variables

```
correlated_vars_w <- cor(Teams_filtered %>%
                           select(W, R, H, BB, SV, SOA,
                                  run_differential,
                                  winning_percentage,
                                  pythagorean_expectation,
                                  ))

corrplot(correlated_vars_w, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
         title = "Correlation of Variables with Wins",
         mar = c(0, 0, 1, 0))
```



Train/Test

Spliting the data into train and test datasets.

```
# Set seed for reproducibility
set.seed(123)

# Split data into training (80%) and testing (20%) sets
split_index <- createDataPartition(Teams_filtered$W, p = 0.8, list = FALSE)
train_data <- Teams_filtered[split_index, ]
test_data <- Teams_filtered[-split_index, ]
```

Linear Regression

```r
# Fit the model
linear_model <- lm(W ~ R+ H+ BB+ SV+
                     SOA+ run_differential+
                     winning_percentage+
                     pythagorean_expectation,
                   data = train_data)

# Summarize the model
summary(linear_model)
```

```
##
## Call:
## lm(formula = W ~ R + H + BB + SV + SOA + run_differential + winning_percentage +
##     pythagorean_expectation, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2314  -1.7804   0.1702   1.9038   8.8944
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -2.469e+01  4.272e+00  -5.779 9.56e-09 ***
## R                       -3.702e-02  1.793e-03 -20.645  < 2e-16 ***
## H                        4.851e-02  9.815e-04  49.426  < 2e-16 ***
## BB                       2.929e-02  1.374e-03  21.317  < 2e-16 ***
## SV                       7.623e-02  1.211e-02   6.293 4.35e-10 ***
## SOA                      8.412e-03  4.218e-04  19.944  < 2e-16 ***
## run_differential         4.382e-02  5.424e-03   8.079 1.58e-15 ***
## winning_percentage       1.423e+02  3.581e+00  39.733  < 2e-16 ***
## pythagorean_expectation -7.111e+01  8.957e+00  -7.939 4.66e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.825 on 1196 degrees of freedom
## Multiple R-squared:  0.9593, Adjusted R-squared:  0.959
## F-statistic:  3526 on 8 and 1196 DF,  p-value: < 2.2e-16
```

```r
# Predict Wins on the test data
predictions <- predict(linear_model, newdata = test_data)
```

```
# Calculate performance metrics
actual <- test_data$W
mae <- mean(abs(predictions - actual))  # Mean Absolute Error
rmse <- sqrt(mean((predictions - actual)^2))  # Root Mean Squared Error
r_squared <- 1 - (sum((predictions - actual)^2) / sum((actual - mean(actual))^2))  #
R²

# Print metrics
cat("MAE:", mae, "\n")
```

```
## MAE: 2.331443
```

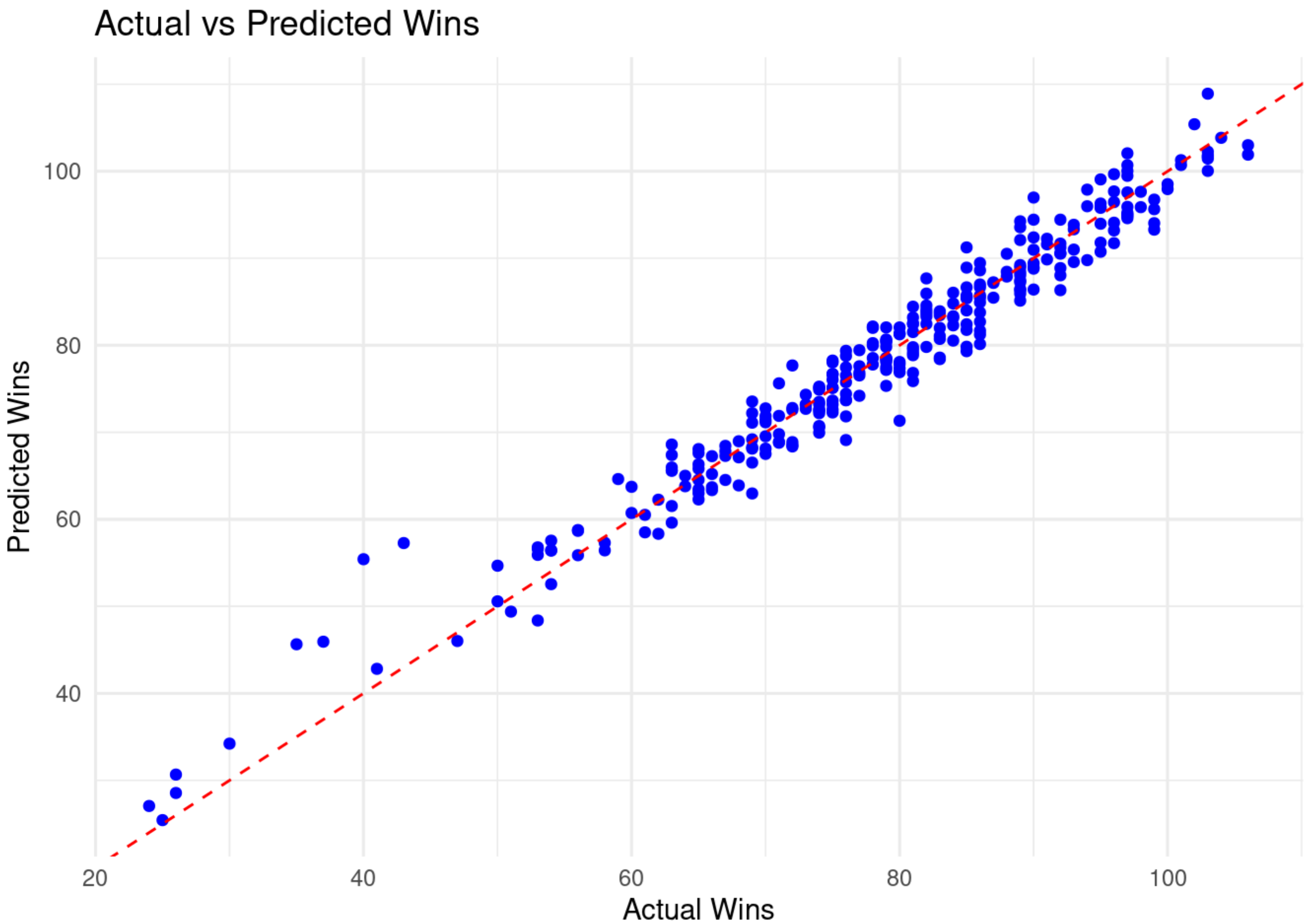
```
cat("RMSE:", rmse, "\n")
```

```
## RMSE: 3.053262
```

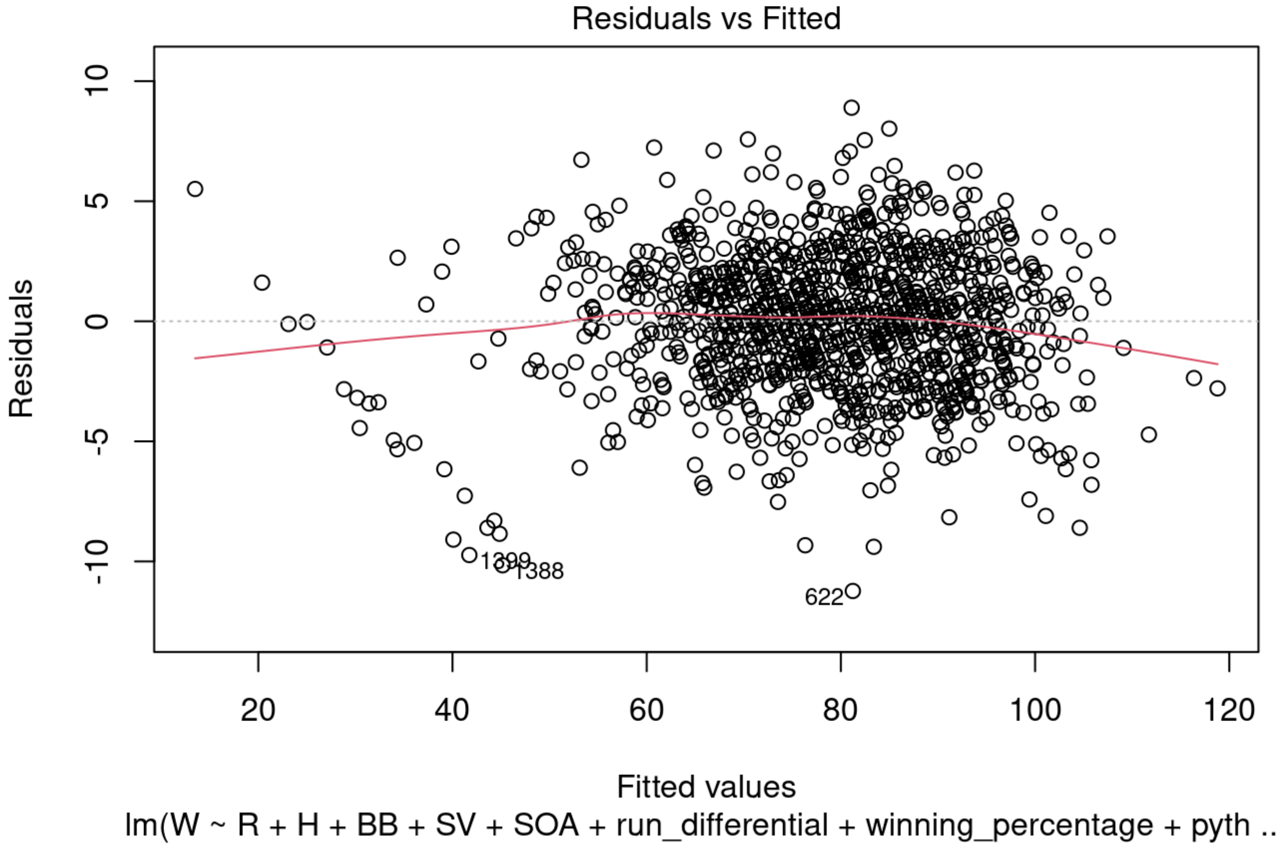
```
cat("R-squared:", r_squared, "\n")
```

```
## R-squared: 0.9588712
```

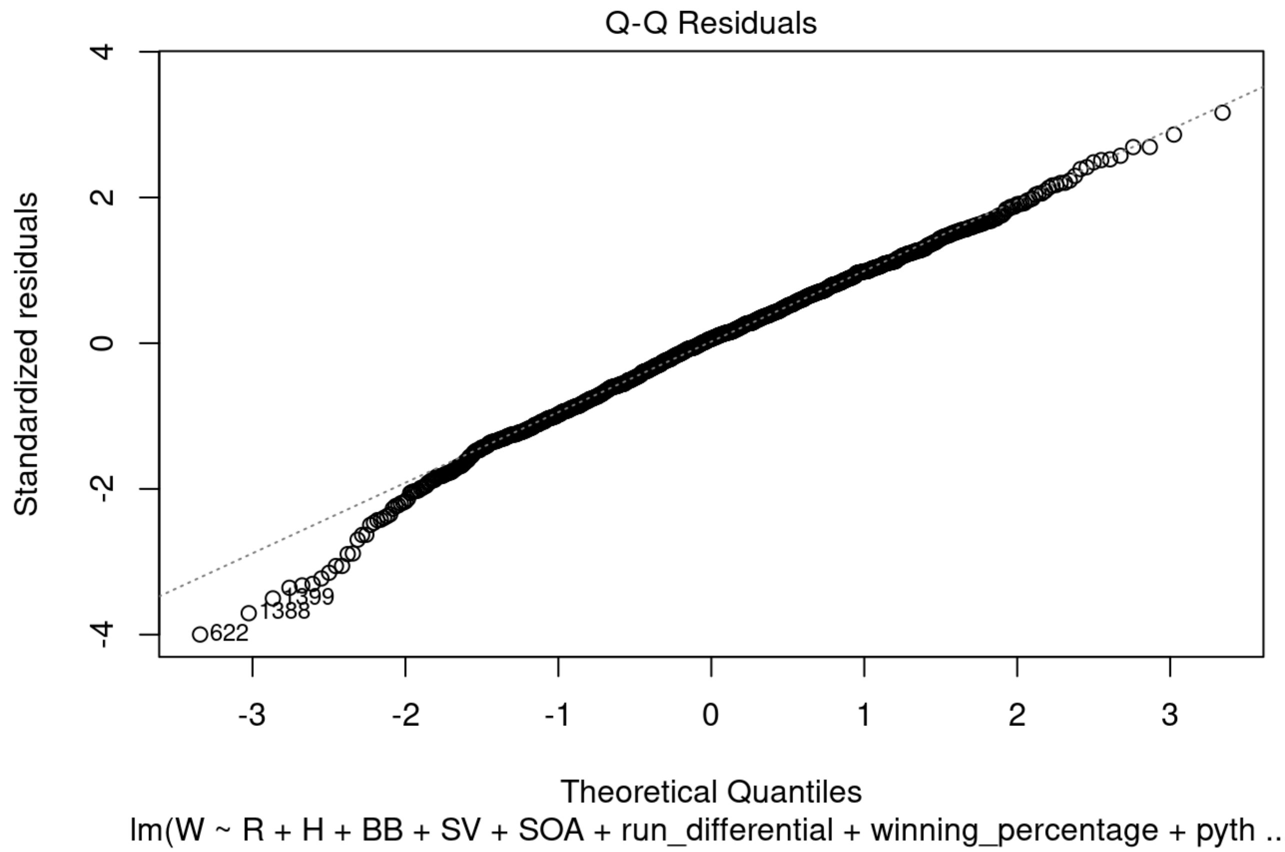
```
# Create a scatter plot
ggplot(data = NULL, aes(x = actual, y = predictions)) +
  geom_point(color = "blue") +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "Actual vs Predicted Wins",
       x = "Actual Wins",
       y = "Predicted Wins") +
  theme_minimal()
```




```
# Residuals vs Fitted plot
plot(linear_model, which = 1)
```




```
# Q-Q plot to check normality
plot(linear_model, which = 2)
```




```
# Cross-validation to validate robustness
train_control <- trainControl(method = "cv", number = 10)
cv_model <- train(W ~ R+ H+ BB+ SV+
                     SOA+ run_differential+
                     winning_percentage+
                     pythagorean_expectation,
                  data = train_data, method = "lm", trControl = train_control)
print(cv_model)
```

```
## Linear Regression
##
## 1205 samples
##    8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1083, 1085, 1085, 1084, 1084, 1086, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.846938  0.9576176  2.233404
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Model Performance Summary Report

The goal of this assignment is to solve a problem by using one of the supervised or unsupervised machine learning algorithms taught in this course: Linear Regression, Logistic Regression, Decision Tree, K-Nearest Neighbor, K-Means Clustering, or DBSCAN.

The problem identified for this dataset early on was to predict MLB wins, which is a continuous numeric target variable. Therefore, the problem attempting to be solved is that of regression. Based on the machine learning algorithms taught in this course, several algorithms can be considered such as: Linear Regression, Decision Tree, or K-Nearest Neighbor. Which algorithm will be used can be derived from insights from exploratory data analysis (performed below) to identify linear relationships with the target variable, `wins` . The goal from identifying linear relationships is to highlight the variables that correlate with `wins` to be used in the predictive model.

The linear regression model's R-squared is 0.9576, very close to 1, indicating that almost 96% of the variance in wins is explained by the model, that is the predictive variables R+ H+ BB+ SV+ SOA+ run_differential+ winning_percentage+ pythagorean_expectation).

The linear regression model's MAE (mean absolute error) is 2.33, indicating that on average the predictions only deviate 2.33 games from the actual wins, highlighting the accuracy of the model due to its low error rate.

The linear regression model's RMSE (root mean squared error) is 2.85, indicating that on average the predictions only deviate 2.33 games from the actual wins, highlighting the accuracy of the model due to its low error rate. RMSE penalized larger errors than MAE, which accounts for the larger RMSE compared to MAE.

Overall, the linear model is an excellent fit for predicting MLB wins.

```
# Save the model
saveRDS(linear_model, file = "mlb_wins_model.rds")
```