Stat139_final_project

2024-11-13

Description of Data:

The data includes a variety of hitting and pitching metrics that come in cumulative and average forms. We may later end up normalizing all the data for easy interpretability on what impacts a team's success the most. We have panel data from a ten year period where the entities are the teams. This will allow us to control for fixed effects across time.

Description of Source:

Baseball reference is a open database that actively updates metrics and has easy to export data sorted by year. Here we load in csv's from different categories (Standard Pitching, Advanced Pitching, Standard Batting, Advanced Batting, and Fielding)

```
#bind all years together
mlb <- rbind(data_2015,data_2016,data_2017,data_2018,data_2019,data_2020,
             data_2021,data_2022,data_2023,data_2024)
```

EDA and Visuals:

Some of the variables we're looking at are:

- SB: Stolen Bases Percentage
- PAge: Pitcher Avg Age
- BatAge: Batter Avg Age
- EV: Average Exit Velocity
- ERA: Earned Run Average
- BB: Bases on Balls
- OPS: On-base percentage plus slugging
- HR: Home Run Percentage
- RS: Runner Support Percentage (Percentage of Baserunners that Eventually Score)
- E: Errors Committed
- RDRS: Defensive Runs Saved

All of these are numerical and have 300 observations in our dataset (with no missing data). The means all look accurate!

We can use the SD / IQR spread measurements to see how varied the skill is within the league.

```
mlb$RS. <- as.numeric(gsub("%", "", mlb$RS.))
mlb$BB. <- as.numeric(gsub("%", "", mlb$BB.))
colnames(mlb)[104] <- "HR_bat"
mlb$HR_percentage <- mlb$HR_bat/mlb$AB
mlb$SB. <- as.numeric(gsub("%", "", mlb$SB.))
```

```r
columns <- c("SB", "PAge", "BatAge", "EV", "ERA", "BB.", "OPS", "HR_percentage", "RS.",
             "E", "Rdrs")

for (col in columns) {
  cat("Statistics for column:", col, "\n")
  column_data <- mlb[[col]]

  cat("- Number of non-missing observations:", sum(!is.na(column_data)), "\n")
  cat("- Number of missing observations:", sum(is.na(column_data)), "\n")
  cat("- Mean:", mean(column_data, na.rm = TRUE), "\n")
  cat("- Median:", median(column_data, na.rm = TRUE), "\n")
  cat("- Standard Deviation:", sd(column_data, na.rm = TRUE), "\n")
  cat("- IQR:", IQR(column_data, na.rm = TRUE), "\n\n")
}
```

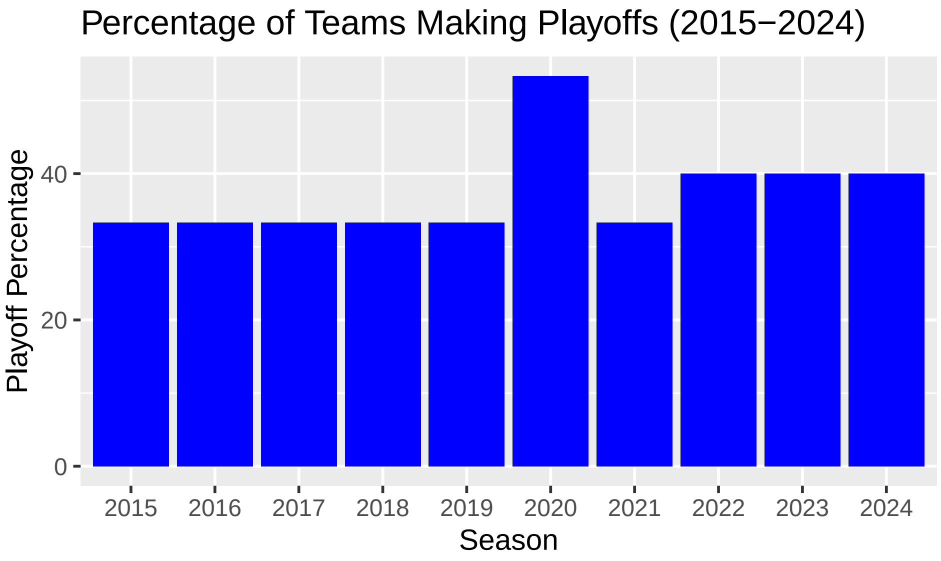
```
## Statistics for column: SB
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 83.42333
## - Median: 80.5
## - Standard Deviation: 36.41544
## - IQR: 46.25
##
## Statistics for column: PAge
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 28.581
## - Median: 28.6
## - Standard Deviation: 1.117124
## - IQR: 1.6
##
## Statistics for column: BatAge
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 28.15733
## - Median: 28.2
## - Standard Deviation: 0.9958675
## - IQR: 1.3
##
## Statistics for column: EV
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 88.34533
## - Median: 88.4
## - Standard Deviation: 0.711735
## - IQR: 1
##
## Statistics for column: ERA
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 4.221667
## - Median: 4.145
## - Standard Deviation: 0.5613581
## - IQR: 0.82
```

```
##
## Statistics for column: BB.
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 8.410667
## - Median: 8.4
## - Standard Deviation: 0.9420072
## - IQR: 1.3
##
## Statistics for column: OPS
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 0.73098
## - Median: 0.729
## - Standard Deviation: 0.04492029
## - IQR: 0.06225
##
## Statistics for column: HR_percentage
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 0.03513405
## - Median: 0.03475591
## - Standard Deviation: 0.006784089
## - IQR: 0.009821661
##
## Statistics for column: RS.
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 30.33
## - Median: 30
## - Standard Deviation: 2.036728
## - IQR: 3
##
## Statistics for column: E
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 85.1
## - Median: 88
## - Standard Deviation: 21.87155
## - IQR: 21
##
## Statistics for column: Rdrs
## - Number of non-missing observations: 300
## - Number of missing observations: 0
## - Mean: 9.783333
## - Median: 7
## - Standard Deviation: 40.5919
## - IQR: 48
```

```
#Coding binary variable for whether or not team reached the playoffs
# potential to add teams that just missed playoffs in original years in order to train model for curren

mlb$playoffs <- ifelse(
 ( (mlb$Tm %in% c("Kansas City Royals", "Toronto Blue Jays", "New York Yankees", "Texas Rangers", "Hous
  (mlb$Tm %in% c("Texas Rangers", "Cleveland Indians","Boston Red Sox", "Toronto Blue Jays", "Baltimore
    (mlb$Tm %in% c("Cleveland Indians","Boston Red Sox","Houston Astros","New York Yankees","Minnesota 
    (mlb$Tm %in% c("Cleveland Indians","Boston Red Sox","Houston Astros","New York Yankees","Oakland Atl
    (mlb$Tm %in% c("Houston Astros","New York Yankees","Minnesota Twins","Oakland Athletics","Tampa Bay
    (mlb$Tm %in% c("Oakland Athletics","Tampa Bay Rays","Minnesota Twins","Cleveland Indians","Houston 
    (mlb$Tm %in% c("Tampa Bay Rays","New York Yankees","Chicago White Sox","Houston Astros","Boston Red
    (mlb$Tm %in% c("New York Yankees","Houston Astros", "Cleveland Guardians","Toronto Blue Jays","Seat
    (mlb$Tm %in% c("Baltimore Orioles","Toronto Blue Jays","Texas Rangers","Tampa Bay Rays","Houston As
    (mlb$Tm %in% c("New York Yankees","Houston Astros", "Cleveland Guardians","Baltimore Orioles","Kansa
    ,1,0)
mlb$playoffs <- as.factor(mlb$playoffs)
```

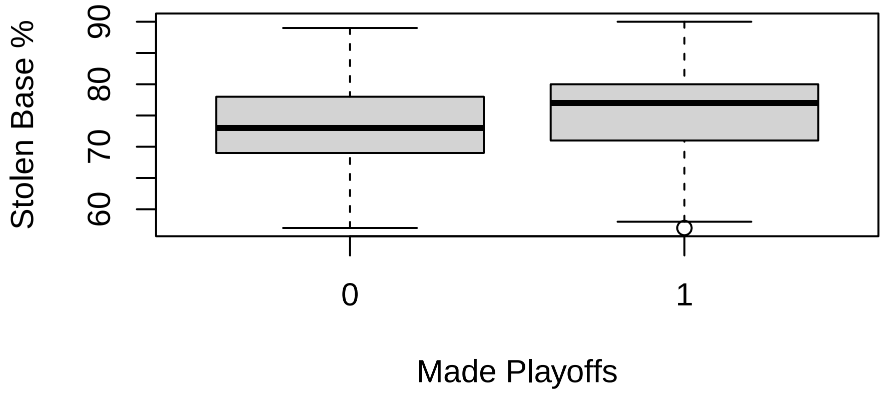
Here is a bar chart depicting how the playoff format has changed with special seasons like covid followed by an expansion two years later.

```
library(ggplot2)
playoff_teams <- c(10, 10, 10, 10, 10, 16, 10, 12, 12, 12)
teams <- c(30,30,30,30,30,30,30,30,30,30)

playoff_percent <- (playoff_teams / teams) * 100

seasons <- 2015:2024

playoff_percent <- data.frame(
  season = seasons,
  playoff_percent = playoff_percent
)

# Plot the bar graph
ggplot(playoff_percent, aes(x = factor(season), y = playoff_percent)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Percentage of Teams Making Playoffs (2015-2024)",
    x = "Season",
    y = "Playoff Percentage"
  )
```

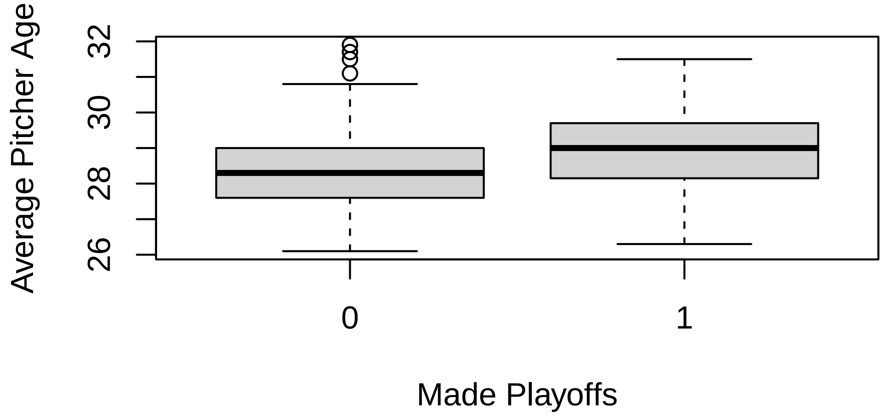


Here are visualizations comparing playoff and non-playoff teams across all predictors.

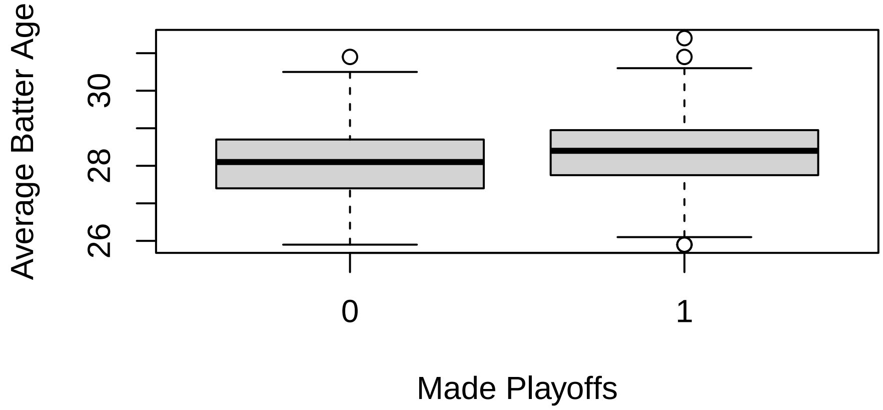
```
plot(mlb$playoffs, mlb$SB., xlab= "Made Playoffs", ylab= "Stolen Base %")
```



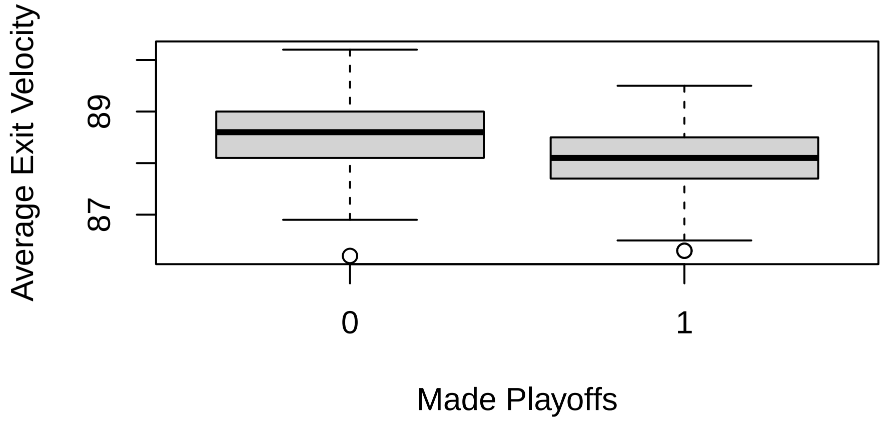
```
plot(mlb$playoffs, mlb$PAge, xlab= "Made Playoffs", ylab= "Average Pitcher Age")
```



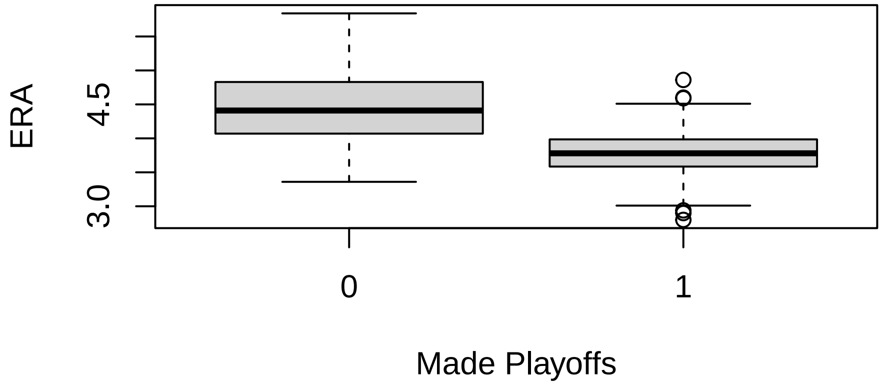
```
plot(mlb$playoffs, mlb$BatAge, xlab= "Made Playoffs", ylab= "Average Batter Age")
```



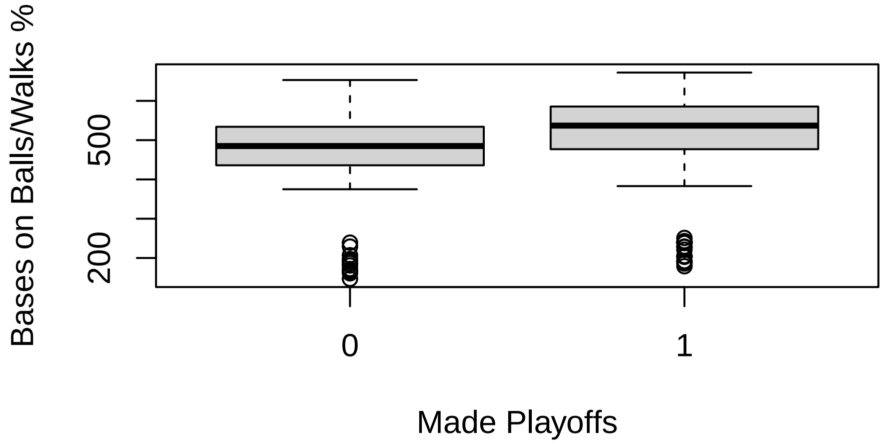
```
plot(mlb$playoffs, mlb$EV, xlab= "Made Playoffs", ylab= "Average Exit Velocity") #FLAG TO CHECK
```



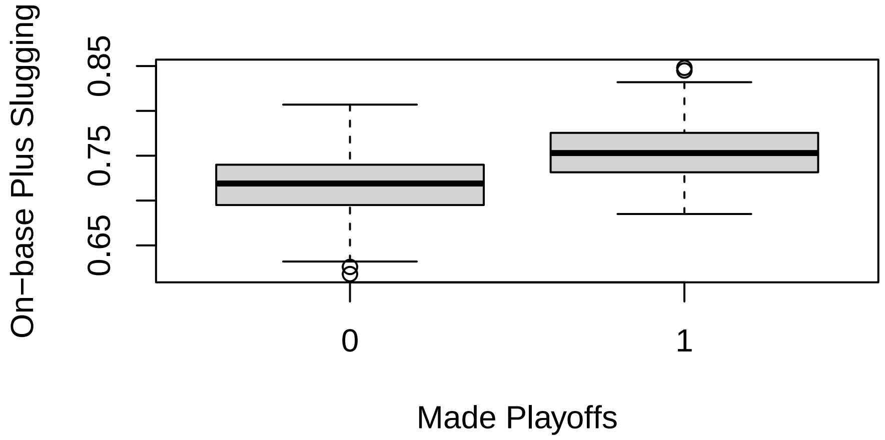
```
plot(mlb$playoffs, mlb$ERA, xlab= "Made Playoffs", ylab= "ERA")
```



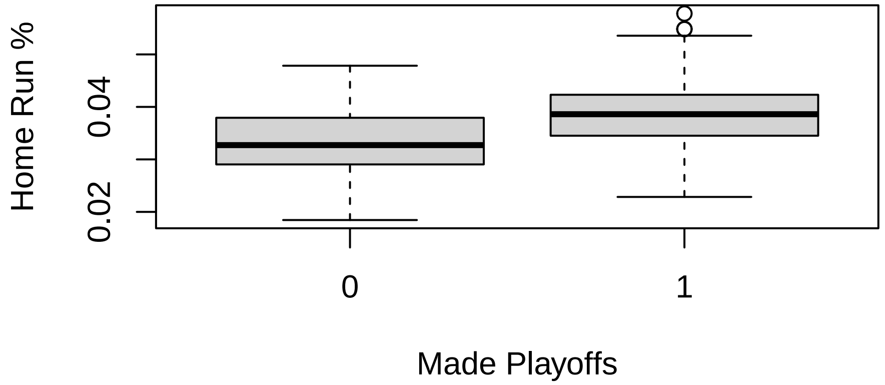
```
colnames(mlb)[108] <- "BB_bat"
plot(mlb$playoffs, mlb$BB_bat, xlab= "Made Playoffs", ylab= "Bases on Balls/Walks %")
```



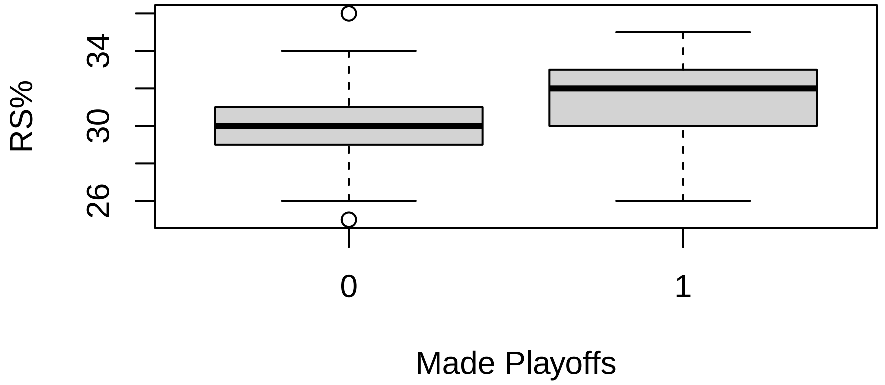
```
colnames(mlb)[113] <- "OPS_bat"
plot(mlb$playoffs, mlb$OPS_bat, xlab= "Made Playoffs", ylab= "On-base Plus Slugging")
```



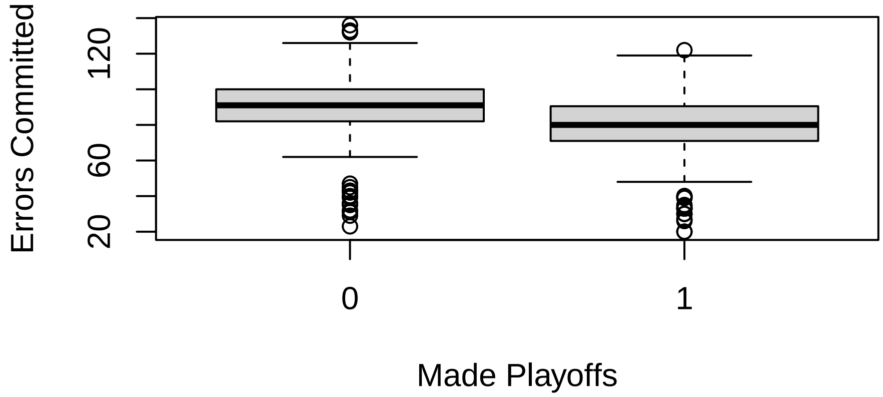
```
plot(mlb$playoffs, mlb$HR_percentage, xlab= "Made Playoffs", ylab= "Home Run %")
```



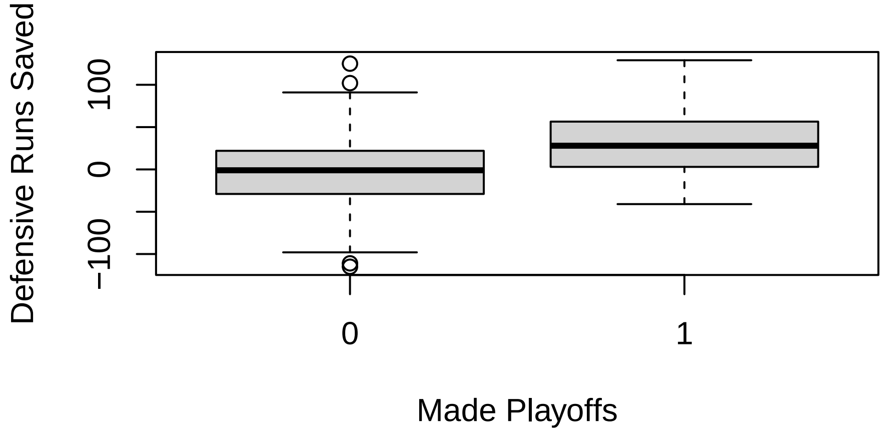
```
plot(mlb$playoffs, mlb$RS., xlab= "Made Playoffs", ylab= "RS%")
```



```
plot(mlb$playoffs, mlb$E, xlab= "Made Playoffs", ylab= "Errors Committed")
```



```
plot(mlb$playoffs, mlb$Rdrs, xlab= "Made Playoffs", ylab= "Defensive Runs Saved")
```



```
#Flagged to check wheter the stat measures team batting or team pitching
```

```
write.csv(mlb, "mlb_edits.csv", row.names = FALSE)
```

Part 1: Hypothesis Testing

In this section, we examine the impact of Intentional Bases on Balls (IBBs) on a team's Win-Loss percentage. This inquiry stems from the strategic decision by pitchers to intentionally walk strong power hitters to reduce the risk of conceding home runs. Our objective is to determine whether this strategy negatively affects a team's Win-Loss percentage by potentially disrupting offensive output. Conversely, it is also plausible that teams with high IBB rates are able to effectively capitalize on these intentional walks, turning them into an advantage. Moreover, having high IBB is likely positively correlated with Home Runs.

We opted to measure IBB as a proportion of IBB to At Bats ((IBB/AB)*100) to normalize across teams and control for differences between teams in plate appearances as teams with more at bats are more likely to have high IBB. IBB as a proportion to at-bats thus displays intentional walks as a function of offensive opportunities.

Additionally, we incorporated team and season fixed effects to control for factors that vary across teams and seasons. These include season-specific dynamics such as rule changes or shortened seasons (e.g. 2020 with COVID), as well as team-level factors like budget constraints or offensive strength, ensuring that our analysis isolates the relationship between IBB and team performance.

In our second model, we have introduced On-Base Percentage (OBP) (measured as a percentage out of 100) and Runs Allowed per Game (RA/G) as control variables. Including OBP allows us to account for a team's offensive strength, as teams with higher OBP are generally better at creating scoring opportunities. Additionally, OBP captures the impact of strong power hitters, who typically have higher OBP values and are more likely to be intentionally walked by the opposition. Omitting OBP could overstate the effect of IBB on team performance by conflating the benefits of a strong offense with the strategic use of intentional walks.

We also include RA/G to control for a team's defensive performance, which is a crucial determinant of overall success. RA/G reflects the ability of a team's pitching staff and defense to limit opposing runs, providing a more balanced assessment of the factors influencing Win-Loss percentage. Together, these controls ensure our model isolates the effect of IBB on team performance while accounting for key offensive and defensive dynamics.

Hypothesis Models: Model 1:

$$W/L_{ijt} = \beta_0 + \beta_1 IBB_{ijt} + \gamma_{ijt} + \delta_{ijt}$$

Model 2:

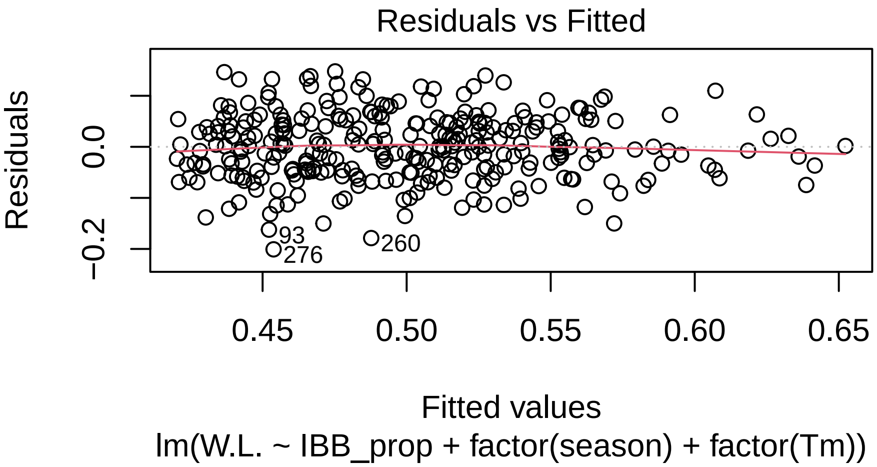
$$W/L_{ijt} = \beta_0 + \beta_1 IBB_{ijt} + \beta_2 OBP_{ijt} + \beta_3 RA/G_{ijt} + \gamma_{ijt} + \delta_{ijt}$$

$$H_0 : \beta_1 = 0 \qquad H_1 : \beta_1 \neq 0$$

```r
colnames(mlb)[120] <- "IBB_bat"
colnames(mlb)[99] <- "AB_bat"
colnames(mlb)[100] <- "Runs_bat"

mlb$IBB_prop <- (mlb$IBB_bat/mlb$AB_bat)*100
mlb$OBP_prop <- mlb$OBP*100

IBB <- lm(W.L. ~ IBB_prop +factor(season) + factor(Tm), data =mlb)
summary(IBB)
```

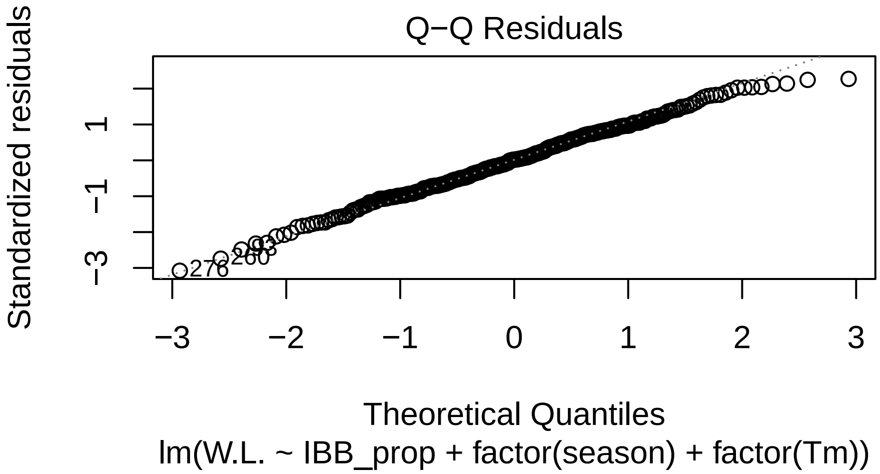
```
##
## Call:
## lm(formula = W.L. ~ IBB_prop + factor(season) + factor(Tm), data = mlb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20081 -0.04585  0.00117  0.04725  0.14782
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           0.403863   0.031533  12.808  < 2e-16
## IBB_prop                              0.112176   0.029289   3.830 0.000161
## factor(season)2016                    0.003791   0.018240   0.208 0.835535
## factor(season)2017                    0.001168   0.018239   0.064 0.948978
## factor(season)2018                    0.003922   0.018241   0.215 0.829918
## factor(season)2019                    0.016182   0.018591   0.870 0.384877
## factor(season)2020                    0.028572   0.019482   1.467 0.143713
## factor(season)2021                    0.018203   0.018697   0.974 0.331154
## factor(season)2022                    0.036747   0.020291   1.811 0.071295
## factor(season)2023                    0.037029   0.020319   1.822 0.069551
## factor(season)2024                    0.034474   0.020042   1.720 0.086618
## factor(Tm)Atlanta Braves              0.042312   0.031616   1.338 0.181969
## factor(Tm)Baltimore Orioles           0.005712   0.032067   0.178 0.858756
## factor(Tm)Boston Red Sox              0.043738   0.031295   1.398 0.163435
## factor(Tm)Chicago Cubs                0.049364   0.031443   1.570 0.117649
## factor(Tm)Chicago White Sox          -0.001201   0.032185  -0.037 0.970252
## factor(Tm)Cincinnati Reds            -0.018561   0.031379  -0.592 0.554686
## factor(Tm)Cleveland Guardians         0.023579   0.047290   0.499 0.618482
## factor(Tm)Cleveland Indians           0.097177   0.034764   2.795 0.005574
## factor(Tm)Colorado Rockies              -0.032625   0.031290  -1.043 0.298090
## factor(Tm)Detroit Tigers                -0.018070   0.031881  -0.567 0.571345
## factor(Tm)Houston Astros                 0.113464   0.031449   3.608 0.000371
## factor(Tm)Kansas City Royals            -0.006945   0.031923  -0.218 0.827955
## factor(Tm)Los Angeles Angels            -0.023010   0.032176  -0.715 0.475183
## factor(Tm)Los Angeles Angels of Anaheim  0.050729   0.074408   0.682 0.495993
## factor(Tm)Los Angeles Dodgers            0.136283   0.031453   4.333 2.11e-05
## factor(Tm)Miami Marlins                 -0.031617   0.031327  -1.009 0.313798
## factor(Tm)Milwaukee Brewers              0.047744   0.031294   1.526 0.128317
## factor(Tm)Minnesota Twins                0.043844   0.031502   1.392 0.165192
## factor(Tm)New York Mets                  0.028222   0.031289   0.902 0.367920
## factor(Tm)New York Yankees               0.103421   0.031472   3.286 0.001157
## factor(Tm)Oakland Athletics              0.021460   0.031981   0.671 0.502802
## factor(Tm)Philadelphia Phillies          0.005182   0.031311   0.166 0.868676
## factor(Tm)Pittsburgh Pirates            -0.028911   0.031289  -0.924 0.356351
## factor(Tm)San Diego Padres               0.018263   0.031352   0.583 0.560718
## factor(Tm)San Francisco Giants           0.019191   0.031297   0.613 0.540280
## factor(Tm)Seattle Mariners               0.048944   0.031685   1.545 0.123638
## factor(Tm)St. Louis Cardinals            0.064049   0.031328   2.044 0.041923
## factor(Tm)Tampa Bay Rays                 0.085155   0.031554   2.699 0.007420
## factor(Tm)Texas Rangers                  0.012411   0.031752   0.391 0.696218
## factor(Tm)Toronto Blue Jays              0.063180   0.032178   1.963 0.050667
## factor(Tm)Washington Nationals          -0.010521   0.031533  -0.334 0.738903
##
## (Intercept)                             ***
## IBB_prop                                ***
## factor(season)2016
## factor(season)2017
## factor(season)2018
## factor(season)2019
## factor(season)2020
## factor(season)2021
## factor(season)2022                      .
## factor(season)2023                      .
## factor(season)2024                      .
## factor(Tm)Atlanta Braves
## factor(Tm)Baltimore Orioles
## factor(Tm)Boston Red Sox
## factor(Tm)Chicago Cubs
## factor(Tm)Chicago White Sox
## factor(Tm)Cincinnati Reds
## factor(Tm)Cleveland Guardians
## factor(Tm)Cleveland Indians             **
## factor(Tm)Colorado Rockies
## factor(Tm)Detroit Tigers
## factor(Tm)Houston Astros                ***
## factor(Tm)Kansas City Royals
## factor(Tm)Los Angeles Angels
## factor(Tm)Los Angeles Angels of Anaheim
## factor(Tm)Los Angeles Dodgers           ***
## factor(Tm)Miami Marlins
## factor(Tm)Milwaukee Brewers
## factor(Tm)Minnesota Twins
## factor(Tm)New York Mets
## factor(Tm)New York Yankees                 **
## factor(Tm)Oakland Athletics
## factor(Tm)Philadelphia Phillies
## factor(Tm)Pittsburgh Pirates
## factor(Tm)San Diego Padres
## factor(Tm)San Francisco Giants
## factor(Tm)Seattle Mariners
## factor(Tm)St. Louis Cardinals              *
## factor(Tm)Tampa Bay Rays                   **
## factor(Tm)Texas Rangers
## factor(Tm)Toronto Blue Jays                .
## factor(Tm)Washington Nationals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06996 on 258 degrees of freedom
## Multiple R-squared:  0.3658, Adjusted R-squared:  0.265
## F-statistic:  3.63 on 41 and 258 DF,  p-value: 1.451e-10
```

```
#test assumptions
plot(IBB, 1)
```



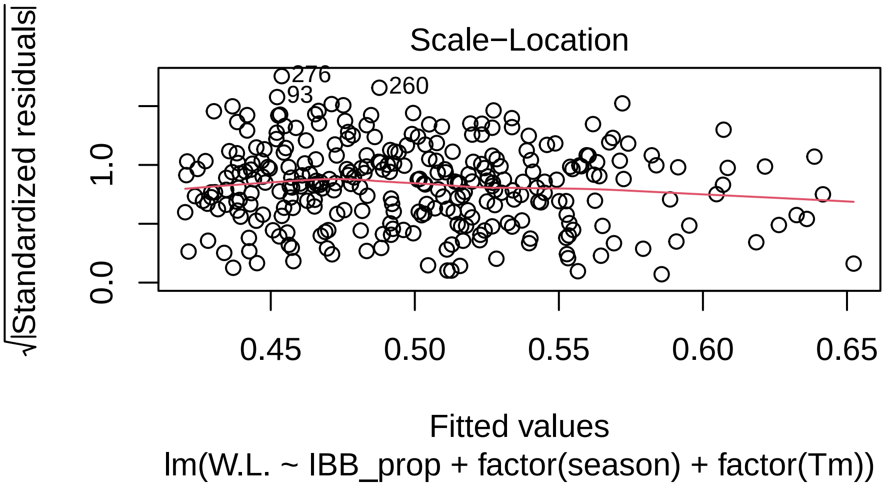
```
plot(IBB,2)
```

```
## Warning: not plotting observations with leverage one:
##   13
```



```
plot(IBB,3)
```

```
## Warning: not plotting observations with leverage one:
##   13
```



```
shapiro.test(residuals(IBB))
```

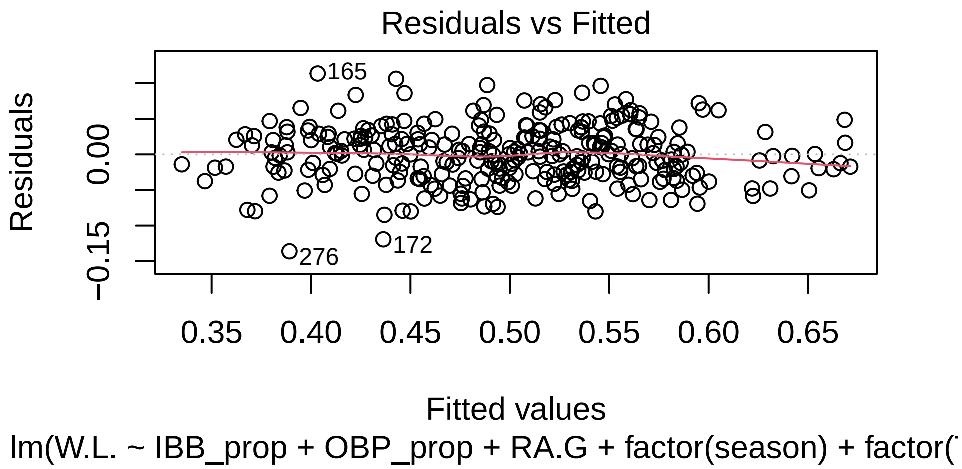
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(IBB)
## W = 0.99546, p-value = 0.5311
```

```
#controlling for OBP and defense
IBB_control <- lm(W.L. ~ IBB_prop + OBP_prop + RA.G + factor(season) + factor(Tm), data = mlb)
summary(IBB_control)
```

```
##
## Call:
## lm(formula = W.L. ~ IBB_prop + OBP_prop + RA.G + factor(season) +
##     factor(Tm), data = mlb)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.136149 -0.027357  0.000209  0.026420  0.113659
```

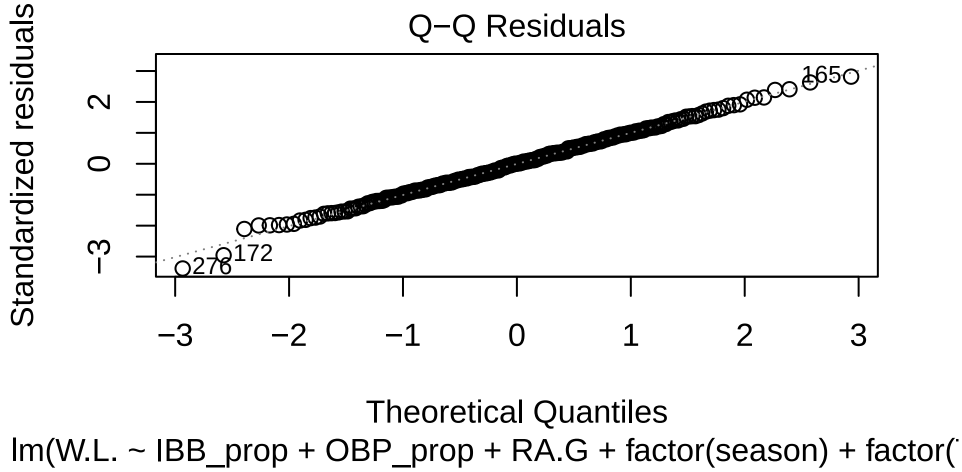
```
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              1.123856   0.097727  11.500  < 2e-16
## IBB_prop                                 0.030832   0.018751   1.644  0.10134
## OBP_prop                                -0.004953   0.004539  -1.091  0.27614
## RA.G                                    -0.113895   0.013691  -8.319 5.33e-15
## factor(season)2016                       0.029611   0.011393   2.599  0.00989
## factor(season)2017                       0.049489   0.011648   4.249 3.01e-05
## factor(season)2018                       0.024555   0.011502   2.135  0.03373
## factor(season)2019                       0.073650   0.012645   5.825 1.71e-08
## factor(season)2020                       0.055498   0.012345   4.495 1.05e-05
## factor(season)2021                       0.037301   0.012020   3.103  0.00213
## factor(season)2022                       0.011267   0.012815   0.879  0.38014
## factor(season)2023                       0.053496   0.012839   4.167 4.23e-05
## factor(season)2024                       0.023463   0.012852   1.826  0.06907
## factor(Tm)Atlanta Braves                 0.018266   0.019688   0.928  0.35439
## factor(Tm)Baltimore Orioles              0.009388   0.019877   0.472  0.63713
## factor(Tm)Boston Red Sox                 0.043254   0.019394   2.230  0.02660
## factor(Tm)Chicago Cubs                  -0.004798   0.019664  -0.244  0.80742
## factor(Tm)Chicago White Sox             -0.024068   0.020029  -1.202  0.23059
## factor(Tm)Cincinnati Reds               -0.014997   0.019456  -0.771  0.44154
## factor(Tm)Cleveland Guardians           -0.028859   0.029433  -0.980  0.32777
## factor(Tm)Cleveland Indians             -0.015652   0.022301  -0.702  0.48341
## factor(Tm)Colorado Rockies               0.051253   0.019834   2.584  0.01032
## factor(Tm)Detroit Tigers                -0.017079   0.019801  -0.863  0.38919
## factor(Tm)Houston Astros                 0.011741   0.020128   0.583  0.56017
## factor(Tm)Kansas City Royals            -0.017273   0.020086  -0.860  0.39063
## factor(Tm)Los Angeles Angels            -0.017323   0.019957  -0.868  0.38617
## factor(Tm)Los Angeles Angels of Anaheim  0.011776   0.046154   0.255  0.79881
## factor(Tm)Los Angeles Dodgers            0.017558   0.020797   0.844  0.39933
## factor(Tm)Miami Marlins                 -0.038223   0.019546  -1.956  0.05160
## factor(Tm)Milwaukee Brewers             -0.007569   0.019583  -0.387  0.69943
## factor(Tm)Minnesota Twins                0.013707   0.019616   0.699  0.48532
## factor(Tm)New York Mets                 -0.014256   0.019524  -0.730  0.46594
## factor(Tm)New York Yankees               0.025892   0.019983   1.296  0.19625
## factor(Tm)Oakland Athletics             -0.012653   0.019892  -0.636  0.52529
## factor(Tm)Philadelphia Phillies          0.003217   0.019403   0.166  0.86844
## factor(Tm)Pittsburgh Pirates            -0.022074   0.019521  -1.131  0.25919
## factor(Tm)San Diego Padres              -0.021265   0.019532  -1.089  0.27729
## factor(Tm)San Francisco Giants          -0.027372   0.019531  -1.401  0.16229
## factor(Tm)Seattle Mariners               0.002763   0.019987   0.138  0.89015
## factor(Tm)St. Louis Cardinals           -0.002114   0.020030  -0.106  0.91602
## factor(Tm)Tampa Bay Rays                -0.012196   0.020251  -0.602  0.54754
## factor(Tm)Texas Rangers                  0.017186   0.019677   0.873  0.38325
## factor(Tm)Toronto Blue Jays              0.021749   0.020043   1.085  0.27890
## factor(Tm)Washington Nationals          -0.015466   0.019541  -0.791  0.42943
##
## (Intercept)                             ***
## IBB_prop
## OBP_prop
## RA.G                                    ***
## factor(season)2016                      **
## factor(season)2017                      ***
## factor(season)2018                      *  
## factor(season)2019                      ***
## factor(season)2020                      ***
## factor(season)2021                      ** 
## factor(season)2022                         
## factor(season)2023                      ***
## factor(season)2024                      .  
## factor(Tm)Atlanta Braves                   
## factor(Tm)Baltimore Orioles                
## factor(Tm)Boston Red Sox                *  
## factor(Tm)Chicago Cubs                     
## factor(Tm)Chicago White Sox                
## factor(Tm)Cincinnati Reds                  
## factor(Tm)Cleveland Guardians              
## factor(Tm)Cleveland Indians                
## factor(Tm)Colorado Rockies              *  
## factor(Tm)Detroit Tigers                   
## factor(Tm)Houston Astros                   
## factor(Tm)Kansas City Royals               
## factor(Tm)Los Angeles Angels               
## factor(Tm)Los Angeles Angels of Anaheim    
## factor(Tm)Los Angeles Dodgers              
## factor(Tm)Miami Marlins                 .  
## factor(Tm)Milwaukee Brewers                
## factor(Tm)Minnesota Twins                  
## factor(Tm)New York Mets                    
## factor(Tm)New York Yankees                 
## factor(Tm)Oakland Athletics                
## factor(Tm)Philadelphia Phillies            
## factor(Tm)Pittsburgh Pirates               
## factor(Tm)San Diego Padres                 
## factor(Tm)San Francisco Giants             
## factor(Tm)Seattle Mariners                 
## factor(Tm)St. Louis Cardinals              
## factor(Tm)Tampa Bay Rays                   
## factor(Tm)Texas Rangers                    
## factor(Tm)Toronto Blue Jays                
## factor(Tm)Washington Nationals             
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.04335 on 256 degrees of freedom
## Multiple R-squared:  0.7584, Adjusted R-squared:  0.7178 
## F-statistic: 18.69 on 43 and 256 DF,  p-value: < 2.2e-16
```

```r
#test assumptions
plot(IBB_control, 1)
```



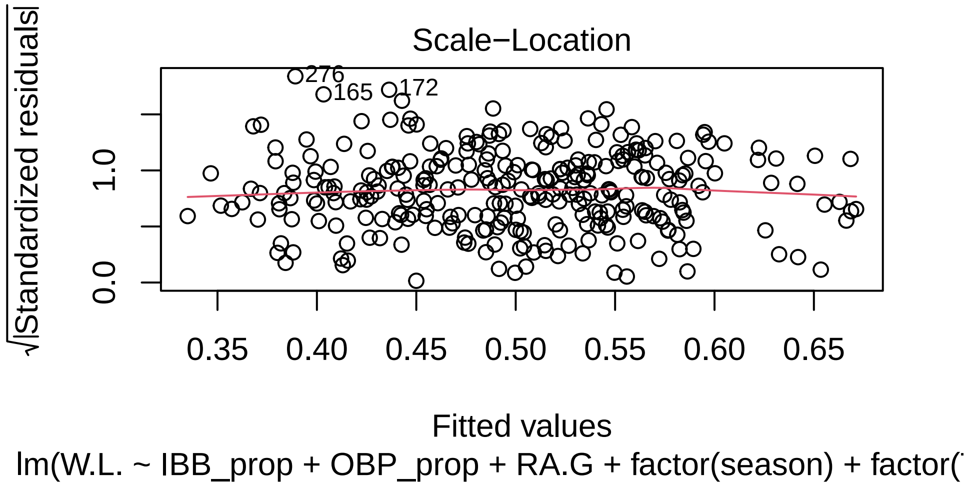
```
plot(IBB_control,2)
```

```
## Warning: not plotting observations with leverage one:
##   13
```



```
plot(IBB_control,3)
```

```
## Warning: not plotting observations with leverage one:
##   13
```



```
shapiro.test(residuals(IBB_control))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(IBB_control)
## W = 0.99785, p-value = 0.9664
```

```
#interaction
IBB_int <- lm(W.L. ~ IBB_prop + OBP_prop + RA.G + IBB_prop*OBP_prop + factor(season) + factor(Tm), data
summary(IBB_int)
```

```
##
## Call:
## lm(formula = W.L. ~ IBB_prop + OBP_prop + RA.G + IBB_prop * OBP_prop +
##     factor(season) + factor(Tm), data = mlb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13135 -0.02762  0.00115  0.02647  0.11789
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.248147   0.141669   8.810  < 2e-16
## IBB_prop                               -0.257766   0.239083  -1.078  0.28199
## OBP_prop                               -0.008855   0.005563  -1.592  0.11267
## RA.G                                   -0.113547   0.013682  -8.299 6.16e-15
## factor(season)2016                      0.029077   0.011391   2.553  0.01127
## factor(season)2017                      0.049138   0.011641   4.221 3.38e-05
## factor(season)2018                      0.024989   0.011497   2.173  0.03067
## factor(season)2019                      0.073465   0.012634   5.815 1.81e-08
## factor(season)2020                      0.053877   0.012407   4.343 2.03e-05
## factor(season)2021                      0.037374   0.012009   3.112  0.00207
## factor(season)2022                      0.010624   0.012815   0.829  0.40785
## factor(season)2023                      0.052922   0.012836   4.123 5.06e-05
## factor(season)2024                      0.022013   0.012896   1.707  0.08904
## factor(Tm)Atlanta Braves                0.016042   0.019755   0.812  0.41754
## factor(Tm)Baltimore Orioles             0.010275   0.019872   0.517  0.60556
## factor(Tm)Boston Red Sox                0.044424   0.019400   2.290  0.02285
## factor(Tm)Chicago Cubs                 -0.002096   0.019772  -0.106  0.91565
## factor(Tm)Chicago White Sox            -0.024048   0.020010  -1.202  0.23057
## factor(Tm)Cincinnati Reds              -0.016318   0.019469  -0.838  0.40271
## factor(Tm)Cleveland Guardians          -0.025070   0.029572  -0.848  0.39737
## factor(Tm)Cleveland Indians            -0.015363   0.022282  -0.689  0.49115
## factor(Tm)Colorado Rockies              0.050340   0.019830   2.539  0.01173
## factor(Tm)Detroit Tigers               -0.017057   0.019783  -0.862  0.38937
## factor(Tm)Houston Astros                0.010958   0.020120   0.545  0.58649
## factor(Tm)Kansas City Royals           -0.015047   0.020152  -0.747  0.45595
## factor(Tm)Los Angeles Angels           -0.017188   0.019939  -0.862  0.38947
## factor(Tm)Los Angeles Angels of Anaheim  0.012939   0.046122   0.281  0.77930
## factor(Tm)Los Angeles Dodgers           0.022722   0.021212   1.071  0.28509
## factor(Tm)Miami Marlins                -0.039052   0.019540  -1.999  0.04672
## factor(Tm)Milwaukee Brewers            -0.008009   0.019569  -0.409  0.68266
## factor(Tm)Minnesota Twins               0.012753   0.019614   0.650  0.51615
## factor(Tm)New York Mets                 -0.012912   0.019537  -0.661  0.50928
## factor(Tm)New York Yankees               0.025787   0.019965   1.292  0.19766
## factor(Tm)Oakland Athletics             -0.012589   0.019874  -0.633  0.52702
## factor(Tm)Philadelphia Phillies          0.001872   0.019417   0.096  0.92326
## factor(Tm)Pittsburgh Pirates            -0.022252   0.019504  -1.141  0.25497
## factor(Tm)San Diego Padres              -0.021484   0.019515  -1.101  0.27198
## factor(Tm)San Francisco Giants          -0.025816   0.019556  -1.320  0.18798
## factor(Tm)Seattle Mariners               0.001530   0.019995   0.077  0.93906
## factor(Tm)St. Louis Cardinals           -0.002189   0.020012  -0.109  0.91299
## factor(Tm)Tampa Bay Rays                -0.013778   0.020275  -0.680  0.49739
## factor(Tm)Texas Rangers                  0.017743   0.019664   0.902  0.36776
## factor(Tm)Toronto Blue Jays              0.021461   0.020026   1.072  0.28491
## factor(Tm)Washington Nationals          -0.012786   0.019649  -0.651  0.51582
## IBB_prop:OBP_prop                        0.008977   0.007414   1.211  0.22708
##
## (Intercept)                             ***
## IBB_prop
## OBP_prop
## RA.G                                    ***
## factor(season)2016                      *
## factor(season)2017                      ***
## factor(season)2018                      *
## factor(season)2019                      ***
## factor(season)2020                      ***
## factor(season)2021                      **
## factor(season)2022
## factor(season)2023                      ***
## factor(season)2024                      .
## factor(Tm)Atlanta Braves
## factor(Tm)Baltimore Orioles
## factor(Tm)Boston Red Sox                *
## factor(Tm)Chicago Cubs
## factor(Tm)Chicago White Sox
## factor(Tm)Cincinnati Reds
## factor(Tm)Cleveland Guardians
## factor(Tm)Cleveland Indians
## factor(Tm)Colorado Rockies              *
## factor(Tm)Detroit Tigers
## factor(Tm)Houston Astros
## factor(Tm)Kansas City Royals
## factor(Tm)Los Angeles Angels
## factor(Tm)Los Angeles Angels of Anaheim
## factor(Tm)Los Angeles Dodgers
## factor(Tm)Miami Marlins                 *
## factor(Tm)Milwaukee Brewers
## factor(Tm)Minnesota Twins
## factor(Tm)New York Mets
## factor(Tm)New York Yankees
## factor(Tm)Oakland Athletics
## factor(Tm)Philadelphia Phillies
## factor(Tm)Pittsburgh Pirates
## factor(Tm)San Diego Padres
## factor(Tm)San Francisco Giants
## factor(Tm)Seattle Mariners
## factor(Tm)St. Louis Cardinals
## factor(Tm)Tampa Bay Rays
## factor(Tm)Texas Rangers
## factor(Tm)Toronto Blue Jays
## factor(Tm)Washington Nationals
## IBB_prop:OBP_prop
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04331 on 255 degrees of freedom
## Multiple R-squared:  0.7598, Adjusted R-squared:  0.7183
## F-statistic: 18.33 on 44 and 255 DF,  p-value: < 2.2e-16
```

Interpretations: Model 1:

$$\hat{\beta_0} = 0.404 \qquad \hat{\beta_1} = 0.112$$

The coefficient for our variable of interest, IBB, is 0.112 with a p-value of 0.00016 which is significant at our set alpha of 0.05. This tells us that on average, an increase in IBB/AB by one percentage point is associated with a 0.112 increase in Win-Loss percentage We can reject the null that intentional walks has no effect on win-loss percentage. This proves that a team being intentionally walked does not have a negative impact on win-loss percantage and likely does not disrupt offensive flow. It instead appears to have a positive impact on win-loss percentage as it is an indicator of having strong hitters who contribute more offensive opportunities.

Model 2:

$$\hat{\beta_0} = 0.062 \qquad \hat{\beta_1} = -0.015 \qquad \hat{\beta_2} = 0.029 \qquad \hat{\beta_3} = -0.108$$

After controlling for OBP and RA/G, our coefficient of interest becomes negative and insignificant at the 95% level with a p-value of 0.29. The coefficients for OBP and RA/G are both significant. This suggests that the effects of intentional walks on Win-Loss proportion are mediated or controlled by OBP and RA/G. That is, having a higher proportion of IBB is associated with having higher offensive strength, explained by OBP as pitchers fear giving up runs to power hitters with high OBP. Therefore, we cannot say that IBB has a direct effect on team performance and is more of a refection of a team's offensive strength rather than a direct driver of performance through interrupted offensive flow.

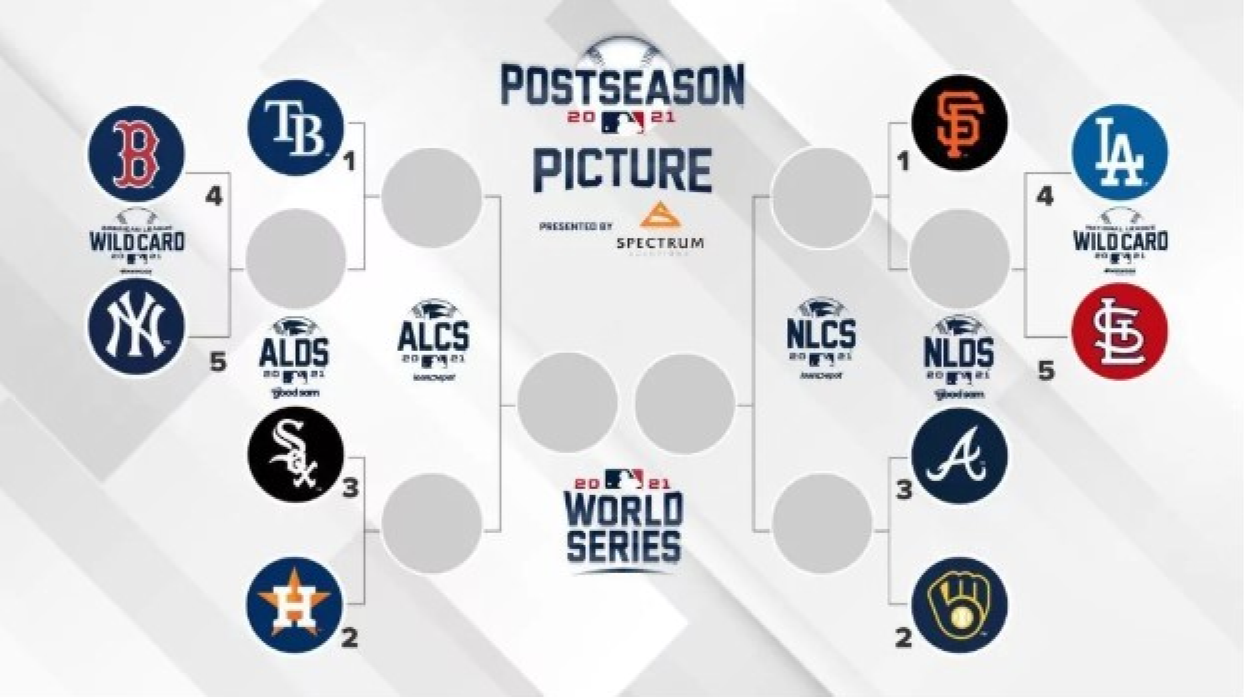
These results can be reconciled with the fact that the effect of IBB is likely highly context dependent. How it benefits or hurts a team relies heavily on game state, lineup, and situations with high leverage where the team can capitalize. For example, an intentional walk has a very different effect when the batting team has 2 outs and empty bases in the 1st inning compared to having no outs, bases loaded in the 9th inning with a strong hitter in the hole. Therefore, because the effects of IBB are highly context dependent, it's possible that it plays a much more situational role and has less impact on overall season results and has more in-game impact. Additionally, IBB can be more reflective of opponent strategy and defensive quality rather than team performance.

Part 2: Discussion on Playoff format change

The MLB expanded their postseason from 10 to 12 teams in 2022. This rule change is not only exciting for fans as their teams should be in the hunt for a playoff spot each season, but the change should be welcomed by statisticians. The exogenous rule change provides a setting for a quasi-random study where the performance of the two additional teams that now make the playoffs is studied. This will allow inference on whether there are systematic differences between what it takes to win to get into the playoffs and the World Series. Over the past three seasons, fans have had the luxury to watch more frequent upsets. While there are different theories on why lower-seeded teams outperform their opponents in the playoffs, we will set up a statistical framework for testing the unexpected high quality play of the 11th and 12th seeds.

Before we dive into the statistical framework, we will clarify how the playoffs work now versus how they did in the Wild Card era which was the previous format. From 2012-2019 and 2021, the playoffs consisted of ten teams. There are six division winners split across the American and National Leagues. Then, each league has two Wild Card teams that did not win their division but held the best records in their league out of non-division winning teams. Each league's two Wild Card teams played each other in a win-or-go-home match. The winner would go on to play in the division series against the division winner with the best record and the other two division winners would play each other. The two teams to win the Division Series would go on to play in the League Championship Series. The winner of each league's championship would meet in the World Series. The Division Series was a best-of-five with the exception of 2012 which was a best-of-three. The League Championship Series and World Series were both best-of-sevens.

The most recent playoffs of the 10-team format



Now the stakes are different. An additional team makes the playoffs in each league where the division winner with the lowest record now participates against the lowest Wild Card team in a best-of-three and the top-two Wild Card teams play each other in a 3-game series as well. The top two division winners maintain their guaranteed spot in the Division Series. The other rounds remain the same. For a visual of the new format refer to the image below.

In the previous format which consists of nine seasons, and three teams made the World Series, two ended up winning the World Series as well. In 2014, the Kansas City Royals and San Francisco Giants were both Wild Card teams that met in the World Series with the Giants winning. In 2019, the Washington Nationals won the World Series as a Wild Card. In the new format we have already seen Wild Card teams make or even win the World Series in just three seasons. In the first year of this format, 2022, the Phillies were not only a Wild Card team, but the lowest ranked team in the National League. Therefore, without this change, they would have not made the playoffs let alone make it to the World Series. The following season saw two Wild Card teams clash in the World Series as 2023 one of which was the lowest seed in the National League as the Philadelphia Phillies were the year prior. This team is the Arizona Diamondbacks which were eliminated by the 5th seed in the American League, the Texas Rangers. So in three seasons, we have seen three out of the six teams to make a World Series be a Wild Card team, two of which were the 6th seeds that would not make it in the prior format. With the miraculous runs witnessed in two of the past three seasons, it is reasonable to expect more which leads to the question of why are teams that performed better all season losing to a team that would have not even made the playoffs? For the 6th seed to advance to the World Series, they must beat the division winners with the best and worst record with the former being heavy favorites usually. The Phillies and Diamondbacks beat division rivals in the National League Division Series after taking down the NL Central winner in the Wild Card round.

This happened two times in the first three seasons under this format and creates a setting to study how

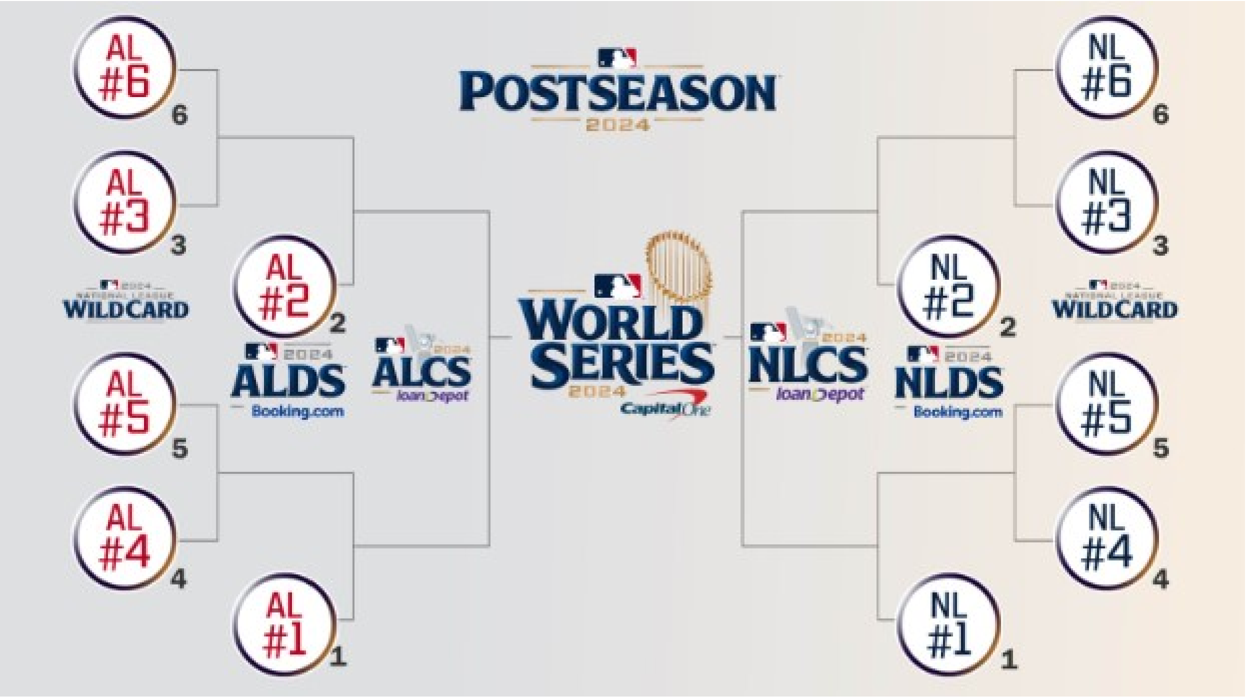


Figure 1: The most recent playoffs of the 10-team format via MLB.com

to build a team that can make the playoffs and rise above expectations and consistently beat the best of the best. Since the sample size is quite small, we will not perform the analyses. However, in a couple of seasons, the following framework should apply. We can use a linear model with two predictors. The first will be an indicator variable for whether or not the season in which the ith team made the playoffs was in the prior or current format. The data will exclude 2020 which took on a different format because of Covid-19. The indicator $Post_i$ takes on a value of 1 for the playoff teams in the 2022 playoffs and onward. The data collection can continue until a different format is taken on or we have sufficient data. The indicator takes on a value of 0 for playoff teams in 2012-2019 and 2021.

The second predictor will be an indicator called $SixSeed_i$ where the team in each league to make the playoffs with the lowest record takes on a value of 1 and any other playoff team has a value of 0. The predictor will consider the hypothetical teams to make it that played under the prior format as if they played under the current format. Further an interaction term will be included between the two indicator variables. Therefore our model is the following:

$Y_i = \beta_0 + \beta_1 Post_i + \beta_2 SixSeed_i + \beta_3(Post_i X SixSeed_i)$

$E(Y_i|Post_i = 0, SixSeed_i = 0) = \beta_0$ $E(Y_i|Post_i = 1, SixSeed_i = 0) = \beta_0 + \beta_1$ $E(Y_i|Post_i = 0, SixSeed_i = 1) = \beta_0 + \beta_2$ $E(Y_i|Post_i = 1, SixSeed_i = 1) = \beta_0 + \beta_1 + \beta_2 + \beta_3$

For instance a team with a 1st to 5th seed in the prior format will have a predicted outcome of solely the intercept since both predictors are 0. A 1st to 5th seed in the current format will have a predicted outcome of the inercept + β_1. The other conditional expectations are included above for all scenarios.

The outcome of interest for us is the number of playoff games won given their seed and format. Other outcomes could be World Series won, series wins, or World Series appearances. Since their seed will relate to factors such as the team performance within the regular season we do not need to consider predictors other than what format the teams played under. We figured that it would be ideal to start with an outcome like wins because it will give us information on precisely how far we expect a team to make it. If a team wins less than two games as a Wild Card, they will not advance to the divison series.

Part 3: Prediction for 2025 Using common baseball stats or regularization?

Baseline Model:

We will use a probit model to predict accurately which teams will make the playoffs based on select predictors. The way in which we will test our model for accuracy are the cross validation techniques learned in class such as k-fold cross validation and lasso. We will start using a baseline model off of substantive knowledge, but we will evaluate its success to generalized cross validation techniques.

Our first model will include a mixture of predictors that capture pitcher and hitter success from standard and advanced metrics. Since we have thirty observations each year for ten years, our model will have to be quite small to avoid overfitting.

Let $Playoff_i$ be an indicator variable that takes one if the ith team makes the playoffs and 0 otherwise. Whether or not a team makes the playoffs is an amazing proxy for team success since it is a goal that all mlb teams strive to achieve in order to have a chance to win the World Series. One challenge to consider is that the playoffs expanded within our study period which will make constraining the number of teams to make the playoffs a challenge. An alternative would be a model that predicts how many wins each team gets, but this faces similar constraint problems since we are not individually simulating games where we can fit the model to have one team win and the other lose.

Our initial model is

$$Pr(Playoff_i = 1) = \beta_0 + \beta_1 SB + \beta_2 PAge + \beta_3 BatAge + \beta_4 EV + \beta_5 ERA + \beta_6 BB + \beta_7 OPS + \beta_8 HR + \beta_9 RS + \beta_{10} E + \beta_{11} RDRS$$

Hypothesis of Interest:

Our hypothesis will be that a team's ability to makes the playoffs can be predicted accurately with a few select metrics that perhaps are not the conventionally cited ideas agreed upon by baseball fans and analysts. We can conduct formal hypothesis tests on the significance of different coefficients, but the main goal is predictive accuracy, so our model will change often based on machine learning techniques due to the volume of predictors available. Our first model will be based on well-known metrics that are known to be associated with winning, but may not be as important as they are made out to be. This model will rely on well-known stats often attributed to team success such as Homeruns, On-Base Percentage, and Strikeouts as well as some advanced metrics. We will compare our preliminary model based on substantive knowledge to our final model to evaluate whether the metrics often considered key to winning are as important as they appear or if there are underlying predictors that are better indicators of winning. We expect that there are more underlying metrics that explain winning more accurately than the conventional ones used.

Analysis: First let's fit a linear regression model solely based on our substantive knowledge. Since our EDA comparing playoff and non-playoff teams on each predictor in our substantive model shows systemic differences in how these types of teams play. We can fit the model.

```r
normalized_mlb <- mlb
predictors <- c("SB", "PAge", "BatAge", "EV", "ERA", "BB", "OPS", "HR", "RS.", "E", "Rdrs")
normalized_mlb[predictors] <- scale(mlb[predictors])

summary(logit<-glm(playoffs~SB + PAge + BatAge + EV + ERA + BB + OPS + HR + RS. + E + Rdrs,
                   data=normalized_mlb,family="binomial"))$coefficients
```

```
##                 Estimate Std. Error     z value     Pr(>|z|)
## (Intercept) -1.783337342  0.2984645 -5.97504043 2.300331e-09
## SB          -0.566566228  0.2591336 -2.18638699 2.878730e-02
## PAge        -0.003846076  0.2180744 -0.01763653 9.859288e-01
## BatAge       0.335134678  0.2141375  1.56504447 1.175725e-01
## EV          -0.526405568  0.2502106 -2.10385031 3.539151e-02
## ERA         -3.964467885  0.9116666 -4.34859412 1.370130e-05
## BB            0.462845465  0.4317155  1.07210768 2.836717e-01
## OPS           0.896345915  0.7085298  1.26507872 2.058431e-01
## HR           -0.183074165  0.3988022 -0.45906011 6.461910e-01
## RS.           1.759350542  0.2829622  6.21761703 5.047617e-10
## E            -1.114913167  0.3827114 -2.91319581 3.577502e-03
## Rdrs          0.268624221  0.2710993  0.99087010 3.217490e-01
```

Our ultimate goal is to evaluate how well we can predict using only a few predictors from substantive knowledge and from a regularization method like ridge. To compare the two, we will need to calculate the root mean square prediction error. Before fitting the ridge model, we will calculate the MSPE of the substantive model. We will also consider another technique: principal component analysis. This method helps to maximize how well we capture the variability in the data with a limited number of predictors. It offers an alternative model with only a few predictors that will likely outperform a model based on substantive knowledge. Once we finish evaluating the models based on RMSPE, we can choose which model we expect to be most helpful for the upcoming season. To see how well our model performs, we can use projections for the upcoming year for each team as our predictors to see how aligned our model is with other predictions.

```r
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```r
k <- 10
set.seed(139)

mlb_logit <- normalized_mlb %>%
  select("SB", "PAge", "BatAge", "EV", "ERA", "BB", "OPS", "HR", "RS.", "E", "Rdrs", "playoffs")
mlb_logit$playoffs <- as.numeric(as.character(mlb_logit$playoffs))
mlb_logit$fold <- sample(rep(1:k, length.out = nrow(mlb_logit)))

mspe_vals_logit <- numeric(k)

for (i in 1:k) {
  train_data_logit <- mlb_logit %>% filter(fold != 1)
  test_data_logit <- mlb_logit %>% filter(fold == 1)
  logit_base <-glm(playoffs~SB + PAge + BatAge + EV + ERA + BB + OPS + HR + RS. + E + Rdrs,
                   data=train_data_logit,family="binomial")
  test_data_logit$predicted_prob <- predict(logit_base, newdata = test_data_logit, type = "response")
  test_data_logit$mspe <- (test_data_logit$predicted_prob - test_data_logit$playoffs)^2
  mspe_vals_logit[i] <- mean(test_data_logit$mspe)
}
logit_mspe <- sqrt(mean(mspe_vals_logit))
cat("Logit Regression MSPE:", logit_mspe, "\n")
```

```
## Logit Regression MSPE: 0.3011769
```

On average this model is 30% off, so we expect that our predictions can be more accurate with machine learning techniques like regularization.

RIDGE/LASSO

We fit four models in the regularization section. The first two were Ridge and LASSO models that included all of the numeric variables in the data. The last two included 45 variables in the data, after filtering out repetitive variables and non-numeric variables.

Of our regularization efforts, the best model (with LASSO on all of the numeric variables) decrease MPSE from 0.301 to 0.290. This increase in prediction accuracy was a lot lower than we expected, especially since we added many new predictors into this model. The result, for us, was that the simplicity of the original model outweighed the importance of a 0.011 decrease in MPSE.

We first prepared the dataset and selected which variables to include (into our full model with all numeric data).

```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```r
library(dplyr)

set.seed(139)
k <- 10

# Convert percent variables into numerics
percent_vars <- c("FB.", "GB.", "LD.", "HardH.", "SO.", "HR.")
for (var in percent_vars) {
  mlb[[var]] <- as.numeric(gsub("%", "", mlb[[var]])) / 100  # Remove '%' and scale
}

# Exclude Opponent and playoffs from the predictors, only use numeric values
predictor_vars <- setdiff(names(mlb)[sapply(mlb, is.numeric)], c("Opponent", "playoffs"))

# Prepare dataset for cross-validation
mlb_cv <- mlb %>%
  select(all_of(c(predictor_vars, "playoffs"))) %>%
  mutate(playoffs = as.numeric(as.character(playoffs)),
         fold = sample(rep(1:k, length.out = nrow(mlb))))
```

We then trained the Ridge and LASSO regression models and compared their performance. We saw an increased performance from the base model in both models, with LASSO outperforming Ridge regularization.

```r
# Ridge Regression
mspe_vals_ridge <- numeric(k)

for (i in 1:k) {
  train_data <- mlb_cv %>% filter(fold != i)
  test_data <- mlb_cv %>% filter(fold == i)

  # Prepare X and y for train and test
  X_train <- as.matrix(scale(train_data %>% select(all_of(predictor_vars))))
  y_train <- train_data$playoffs
  X_test <- as.matrix(scale(test_data %>% select(all_of(predictor_vars))))
  y_test <- test_data$playoffs

  # Fit Ridge model
  ridge_model <- glmnet(X_train, y_train, alpha = 0, family = "binomial")

  # Predict for the test set
  y_pred <- predict(ridge_model, X_test, s = cv.glmnet(X_train, y_train, alpha = 0, family = "binomial"

  # Compute MSPE for this fold
  test_data$mspe <- (y_pred - y_test)^2
  mspe_vals_ridge[i] <- mean(test_data$mspe)
}

ridge_mspe <- sqrt(mean(mspe_vals_ridge))
cat("Ridge Regression MSPE:", ridge_mspe, "\n")
```

```
## Ridge Regression MSPE: 0.298073
```

```
# LASSO Regression
mspe_vals_lasso <- numeric(k)

for (i in 1:k) {
  train_data <- mlb_cv %>% filter(fold != i)
  test_data <- mlb_cv %>% filter(fold == i)

  # Prepare X and y for train and test
  X_train <- as.matrix(scale(train_data %>% select(all_of(predictor_vars))))
  y_train <- train_data$playoffs
  X_test <- as.matrix(scale(test_data %>% select(all_of(predictor_vars))))
  y_test <- test_data$playoffs

  # Fit LASSO model
  lasso_model <- glmnet(X_train, y_train, alpha = 1, family = "binomial")

  # Predict for the test set
  y_pred <- predict(lasso_model, X_test, s = cv.glmnet(X_train, y_train, alpha = 1, family = "binomial"

  # Compute MSPE for this fold
  test_data$mspe <- (y_pred - y_test)^2
  mspe_vals_lasso[i] <- mean(test_data$mspe)
}

lasso_mspe <- sqrt(mean(mspe_vals_lasso))
cat("LASSO Regression MSPE:", lasso_mspe, "\n")
```

```
## LASSO Regression MSPE: 0.2902795
```

We then tried to use more limited models, eliminating some redundancies in the data. We limited the data down to 45 variables, and then trained the same models. The Ridge and LASSO models had an MPSE of 0.304 and 0.291, respectively. These results lead us to a very similar model to the ones obtained with the same regularization techniques. The LASSO model again outperformed the Ridge, and was only 0.001 worse than the model with the full data in prediction!

```
# Limited models, exclude repetitive / redundant variables, categorical variables
predictor_vars <- c(
  # Pitching
  "PAge", "RA.G", "ERA", "GF", "IP", "H", "R", "ER", "HR", "BB", "IBB", "SO",
  "HBP", "BK", "WP", "BF", "WHIP", "FIP", "LOB",

  # Batting
  "BatAge", "R.G", "PA", "AB_bat", "Runs_bat", "H", "X2B", "X3B", "HR_bat",
  "RBI", "SB", "BB_bat", "OPS_bat", "HR_percentage",

  # Fielding
  "Rdrs", "DefEff", "E",

  # Advanced metrics
  "EV", "HardH.", "LD.", "GB.", "FB.", "WPA", "RE24"
)

# Prepare dataset for cross-validation
mlb_cv <- mlb %>%
  select(all_of(c(predictor_vars, "playoffs"))) %>%
  mutate(playoffs = as.numeric(as.character(playoffs)),
         fold = sample(rep(1:k, length.out = nrow(mlb))))
```

```
# Ridge Regression
mspe_vals_ridge <- numeric(k)

for (i in 1:k) {
  train_data <- mlb_cv %>% filter(fold != i)
  test_data <- mlb_cv %>% filter(fold == i)

  # Prepare X and y for train and test
  X_train <- as.matrix(scale(train_data %>% select(all_of(predictor_vars))))
  y_train <- train_data$playoffs
  X_test <- as.matrix(scale(test_data %>% select(all_of(predictor_vars))))
  y_test <- test_data$playoffs

  # Fit Ridge model
  ridge_model <- glmnet(X_train, y_train, alpha = 0, family = "binomial")

  # Predict for the test set
  y_pred <- predict(ridge_model, X_test, s = cv.glmnet(X_train, y_train, alpha = 0, family = "binomial"

  # Compute MSPE for this fold
  test_data$mspe <- (y_pred - y_test)^2
  mspe_vals_ridge[i] <- mean(test_data$mspe)
}

ridge_mspe <- sqrt(mean(mspe_vals_ridge))
cat("Ridge Regression MSPE:", ridge_mspe, "\n")
```

```
## Ridge Regression MSPE: 0.3037014
```

```
# LASSO Regression
mspe_vals_lasso <- numeric(k)

for (i in 1:k) {
  train_data <- mlb_cv %>% filter(fold != i)
  test_data <- mlb_cv %>% filter(fold == i)

  # Prepare X and y for train and test
  X_train <- as.matrix(scale(train_data %>% select(all_of(predictor_vars))))
  y_train <- train_data$playoffs
  X_test <- as.matrix(scale(test_data %>% select(all_of(predictor_vars))))
  y_test <- test_data$playoffs

  # Fit LASSO model
  lasso_model <- glmnet(X_train, y_train, alpha = 1, family = "binomial")

  # Predict for the test set
  y_pred <- predict(lasso_model, X_test, s = cv.glmnet(X_train, y_train, alpha = 1, family = "binomial"

  # Compute MSPE for this fold
  test_data$mspe <- (y_pred - y_test)^2
  mspe_vals_lasso[i] <- mean(test_data$mspe)
}
```

```
## Warning: from glmnet C++ code (error code -100); Convergence for 100th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
lasso_mspe <- sqrt(mean(mspe_vals_lasso))
cat("LASSO Regression MSPE:", lasso_mspe, "\n")
```

```
## LASSO Regression MSPE: 0.2914324
```

A couple of things to consider are team budget and time-fixed effects. Given a team's budget, we can develop an expectation of the caliber of players a team can field. For time fixed effects, we can control for the differences in playoff appearances as a factor of time which would capture the exogenous change in teams earning a spot.

Lastly, we will use principal component analysis to see if we can arrive at more accurate predictions.

```
k <- 10
set.seed(139)

# Convert percent variables into numerics
percent_vars <- c("FB.", "GB.", "LD.", "HardH.", "SO.", "HR.")
for (var in percent_vars) {
  mlb[[var]] <- as.numeric(gsub("%", "", mlb[[var]])) / 100  # Remove '%' and scale
}

# Exclude Opponent and playoffs from the predictors, only use numeric values
predictor_vars <- setdiff(names(mlb)[sapply(mlb, is.numeric)], c("Opponent", "playoffs"))

# Prepare dataset for cross-validation
mlb_cv <- mlb %>%
  select(all_of(c(predictor_vars, "playoffs"))) %>%
  mutate(playoffs = as.numeric(as.character(playoffs)),
         fold = sample(rep(1:k, length.out = nrow(mlb))))

# Ridge Regression
mspe_vals_pca <- numeric(k)
```

```
for (i in 1:k) {
  train_data <- mlb_cv %>% filter(fold != i)
  test_data <- mlb_cv %>% filter(fold == i)

  # Prepare X and y for train and test
  X_train <- as.matrix(scale(train_data %>% select(all_of(predictor_vars))))
  y_train <- train_data$playoffs
  X_test <- as.matrix(scale(test_data %>% select(all_of(predictor_vars))))
  y_test <- test_data$playoffs

  # Fit PCA model
  pca_model <- prcomp(X_train, center = TRUE, scale. = TRUE)

  explained_variance <- cumsum(pca_model$sdev^2 / sum(pca_model$sdev^2))
  num_components <- which(explained_variance >= 0.95)[1]

  #Transform data based on num_componenets
  X_train_pca <- pca_model$x[, 1:num_components]
  X_test <- scale(test_data %>% select(all_of(predictor_vars)))
  X_test_pca <- predict(pca_model, newdata = X_test)[, 1:num_components]

  # Fit a logistic regression model using the principal components
  pca_glm <- glm(y_train ~ ., data = as.data.frame(cbind(y_train, X_train_pca)), family = "binomial")

  # Predict for the test set
  y_pred <- predict(pca_glm, newdata = as.data.frame(X_test_pca), type = "response")

  # Compute MSPE for this fold
  test_data$mspe <- (y_pred - test_data$playoffs)^2
  mspe_vals_pca[i] <- mean(test_data$mspe)
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
pca_mspe <- sqrt(mean(mspe_vals_pca))
cat("PCA-based Logistic Regression MSPE:", pca_mspe, "\n")
```

```
## PCA-based Logistic Regression MSPE: 0.3277009
```

The reason PCA is not working well here is that the current data we have does not explain enough of the variance with just a select few predictors such that it is hard to remove anyone. The predective power is low to begin with, so we need to add other predictors of interest like budget and time fixed effects.