Red Sox Personal Project

Jimmy Dysart

2023-04-13

Purpose

I am making this project to express my passion about working for the Boston Red Sox. All my life, I've wanted to live in Boston as a grownup. I am extremely passionate about sports and some of my favorite experiences in life are at ballparks with my friends and siblings. I admire Boston's sports culture. I want to use this project to show why I am an excellent fit for the Boston Red Sox Baseball Analytics Internship.

Introduction

Regular Season Baseball is all about finding a way to get into the playoffs. In the playoffs, anything can happen. In baseball, the ONLY sure-fire way to get into the playoffs is to win your division. That means in an ultra-competitive division like the AL East, all a team needs to do to get into the playoffs is be the `#1 seed`. Since the 1998 season, there has been 5 teams in the AL East:

- Boston Red Sox
- New York Yankees
- Toronto Blue Jays
- Baltimore Orioles
- Tampa Bay Rays (was "Devil Rays" until 2008)

My Question

I am curious what it takes to be a `#1 seed` in baseball's hardest division to play in. I want to compare a bunch of different `Pitching` statistical metrics with the regular season final rankings in the AL East. I will be using `year-by-year team pitching data` from the 1998 season to the 2022 season from each team in the AL East. I want to find the best predicting variables for winning the division.

Gathering Data

To gather the data I am using year-by-year team pitching data from the (team names are hyperlinked with data source):

- Boston Red Sox
- New York Yankees
- Toronto Blue Jays

- Baltimore Orioles
- Tampa Bay Rays

I now need to import the data so I perform data analysis on it.

Baseball Reference has a pretty neat feature where I could make the data into excel format then copy/paste into excel and export from excel as a csv file.

Boston Data:

```
library(readr)
# Red Sox Raw Data
RedSox_raw <- read_csv("RedSox_raw.csv")
```

```
## Rows: 123 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Lg
## dbl (24): Year, W, L, Finish, RA/G, ERA, G, CG, tSho, SV, IP, H, R, ER, HR, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
RedSox_raw<- RedSox_raw %>% filter(Year >= 1998)

RedSox_raw <- RedSox_raw %>% mutate(Team = "BOS") # Adding team abbreviation for when I merge all the d
```

New York Data:

```
# Yankees Raw Data
Yankees_raw <- read_csv("Yankees_raw.csv")
```

```
## Rows: 121 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Lg
## dbl (24): Year, W, L, Finish, RA/G, ERA, G, CG, tSho, SV, IP, H, R, ER, HR, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Yankees_raw<- Yankees_raw %>% filter(Year >= 1998)

Yankees_raw <- Yankees_raw %>% mutate(Team = "NYY")
```

Toronto Data:

```
# Blue Jays Raw Data
BlueJays_raw <- read_csv("BlueJays_raw.csv")
```

```
## Rows: 47 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Lg
## dbl (24): Year, W, L, Finish, RA/G, ERA, G, CG, tSho, SV, IP, H, R, ER, HR, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
BlueJays_raw<- BlueJays_raw %>% filter(Year >= 1998)

BlueJays_raw <- BlueJays_raw %>% mutate(Team = "TOR")
```

Baltimore Data:

```
# Orioles Raw Data
Orioles_raw <- read_csv("Orioles_raw.csv")
```

```
## Rows: 123 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Lg
## dbl (24): Year, W, L, Finish, RA/G, ERA, G, CG, tSho, SV, IP, H, R, ER, HR, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Orioles_raw<- Orioles_raw %>% filter(Year >= 1998)

Orioles_raw <- Orioles_raw %>% mutate(Team = "BAL")
```

Tampa Bay Data:

```
# Rays Raw Data
Rays_raw <- read_csv("Rays_raw.csv")
```

```
## Rows: 26 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Lg
## dbl (24): Year, W, L, Finish, RA/G, ERA, G, CG, tSho, SV, IP, H, R, ER, HR, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Rays_raw <- Rays_raw %>% mutate(Team = "TBD")
```

```
dim(Rays_raw) # only 26 observations
```

```
## [1] 26 26
```

Notice that the Rays raw data does not need to be filtered because it was already filtered in baseball reference for after 1998.

Appending Rows From 5 data frames into 1 data frame

To easily manipulate and analyze the data, I want to `row bind` all of the smaller data frames into a single larger data frame.

```
AL_EAST <- rbind(RedSox_raw,Yankees_raw,BlueJays_raw,Orioles_raw,Rays_raw)
```

I want to create the outcome variable to make this a binary classification problem.

```
AL_EAST <- AL_EAST %>% mutate(Finish = if_else(Finish == 1, 1,0)) # Making outcome variable binary
```

Since the year 2020 only had 60 games, I am taking it out of the data set because it will inaccurately depict the statistics it takes to win the AL East. Also, I will be removing the year 2023 because only ~9 games have been played this season.

```
AL_EAST <- AL_EAST %>% filter(Year!= 2023)
AL_EAST <- AL_EAST %>% filter(Year!= 2020)
```

Now that I have all of my seperate data frames in one larger data frame, I want to perform some quick data analysis to look at trends within the data.

```
head(AL_EAST)
```

```
## # A tibble: 6 x 26
##    Year Lg          W     L Finish `RA/G`   ERA     G    CG  tSho    SV    IP
##   <dbl> <chr>   <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  2022 AL East    78    84      0   4.86  4.53   162     5    10    39 1431
## 2  2021 AL East    92    70      0   4.62  4.26   162     0     7    49 1419
## 3  2019 AL East    84    78      0   5.11  4.7    162     1     8    33 1471
## 4  2018 AL East   108    54      1   3.99  3.75   162     2    14    46 1458.
## 5  2017 AL East    93    69      1   4.12  3.7    162     5    11    39 1482.
## 6  2016 AL East    93    69      1   4.28  4      162     9     5    43 1439.
## # ... with 14 more variables: H <dbl>, R <dbl>, ER <dbl>, HR <dbl>, BB <dbl>,
## #   SO <dbl>, WHIP <dbl>, SO9 <dbl>, HR9 <dbl>, E <dbl>, DP <dbl>,
## #   `Fld%` <dbl>, PAge <dbl>, Team <chr>
```

I want to rename some of the variables that have weird characters.

```
AL_EAST <- rename(AL_EAST, RA_G = `RA/G`)

AL_EAST <-rename(AL_EAST, Fld = `Fld%`)
```

I have 26 different columns. Here is a codebook for the different columns.

The outcome variable:

- `Finish`: Place finished in league(1 if first; 0 otherwise)

The predictor variables:

- `Year`: Year of baseball season

- `Lg`: League(Always AL East)
- `W`: Amount of Wins in that season
- `L`: Amount of Loses in that season
- `RA_G`: Runs Allowed Per Game
- `ERA`: $9 * (ER/IP)$
- `G`: Games Played
- `CG`: Complete Games
- `tSho`: Team shutouts
- `SV`: Saves
- `IP`: Innings Pitched
- `H`: Hits Against
- `R`: Runs Against
- `ER`: Earned Runs Against
- `HR`: Home Runs Against
- `BB`: Bases on Balls Against
- `SO`: Strikeouts
- `WHIP`: $(BB + H)/IP$
- `SO9`: $9 * (SO/IP)$
- `HR9`: $9 * (HR/IP)$
- `E`: Errors Committed
- `DP`: Double Plays
- `Fld`: Fielding Percentage
- `PAge`: Pitchers Average Age

Data Analysis

In this section I will look at how the different predictor variables interact with eachother. Since I scraped and cleaned this data set myself, there is no NA values.

```
sum(is.na(AL_EAST) == TRUE) # 0 NA Values; just to make sure
```

```
## [1] 0
```

I am intrigued by the variable `PAge`. So, I want to see how it compares within and across the AL East.

```
AL_EAST %>% summarise(avg_PAge = mean(PAge)) #29.1
```

```
## # A tibble: 1 x 1
##   avg_PAge
##      <dbl>
## 1     29.1
```

```
AL_EAST %>% group_by(Team) %>% summarise(avg_PAge = mean(PAge)) %>%
  arrange(avg_PAge)
```

```
## # A tibble: 5 x 2
##   Team  avg_PAge
##   <chr>    <dbl>
## 1 TBD       27.8
## 2 BAL       28.2
## 3 TOR       28.7
## 4 BOS       30.1
## 5 NYY       30.6
```

The average pitchers age over all years is 29.1. Interestingly, only BOS and NYY have average pitching ages over that threshold.

I wonder if there is a positive correlation between average starting pitchers age and winning the AL East?

This is a type of question that will be answered thoroughly with regression modeling later.

I want to look at innings pitched for each team. This will tell me what teams tend to go into overtime more often than others.

```
AL_EAST %>% summarise(avg_IP = mean(IP)) #1443
```

```
## # A tibble: 1 x 1
##   avg_IP
##    <dbl>
## 1  1443.
```

```
AL_EAST %>% group_by(Team) %>% summarise(avg_IP = mean(IP)) %>%
  arrange(avg_IP)
```

```
## # A tibble: 5 x 2
##   Team  avg_IP
##   <chr>  <dbl>
## 1 BAL    1438.
## 2 TBD    1442.
## 3 TOR    1443.
## 4 NYY    1446.
## 5 BOS    1448.
```

It looks like Boston and NYY go into overtime more than the other 3 teams in the AL East.

I also want to look at how the different teams compare based on their WHIP. WHIP is a common baseball stastistics to measure the performance of pitchers.

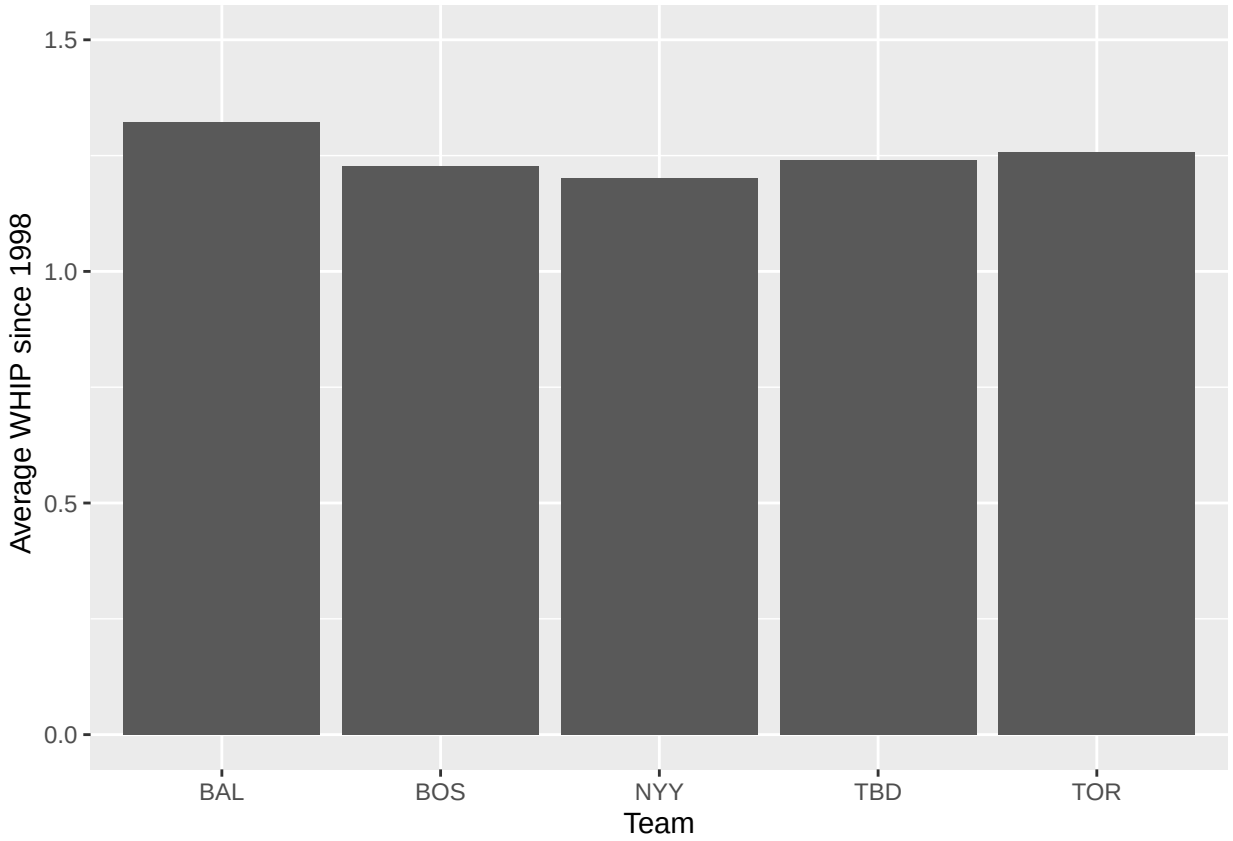
```
AL_EAST %>% summarise(avg_WHIP = mean(WHIP)) #1.35
```

```
## # A tibble: 1 x 1
##   avg_WHIP
##      <dbl>
## 1     1.35
```

```
AL_EAST %>% group_by(Team) %>% summarise(avg_WHIP = mean(WHIP)) %>%
  arrange(avg_WHIP)
```

```
## # A tibble: 5 x 2
##   Team  avg_WHIP
##   <chr>    <dbl>
## 1 NYY       1.30
## 2 BOS       1.33
## 3 TBD       1.34
## 4 TOR       1.36
## 5 BAL       1.43
```

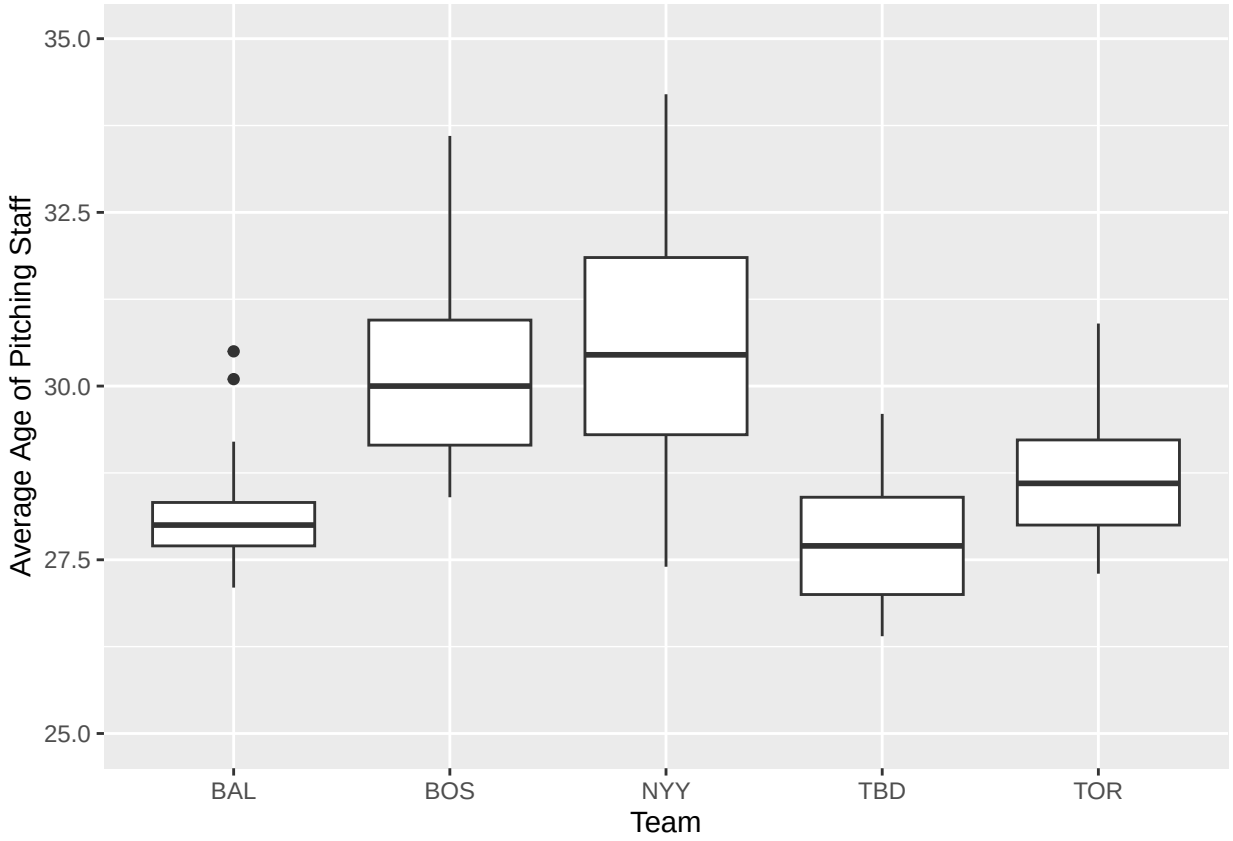
```
ggplot(aes(x = Team, y = WHIP/26),data = AL_EAST)+
       geom_col() + ylab(" Average WHIP since 1998") + scale_y_continuous(limits = c(0,1.5))
```



The average WHIP in the AL East is 1.35, and not suprisingly the Yankees and Red Sox have the lowest average WHIP among pitching staffs.

I also want to look at a boxplot grouped by team that shows the pitching staff average age over the years.

```
ggplot(aes(x = Team, y = PAge),data = AL_EAST)+
       geom_boxplot() + ylab("Average Age of Pitching Staff") + scale_y_continuous(limits = c(25,35))
```

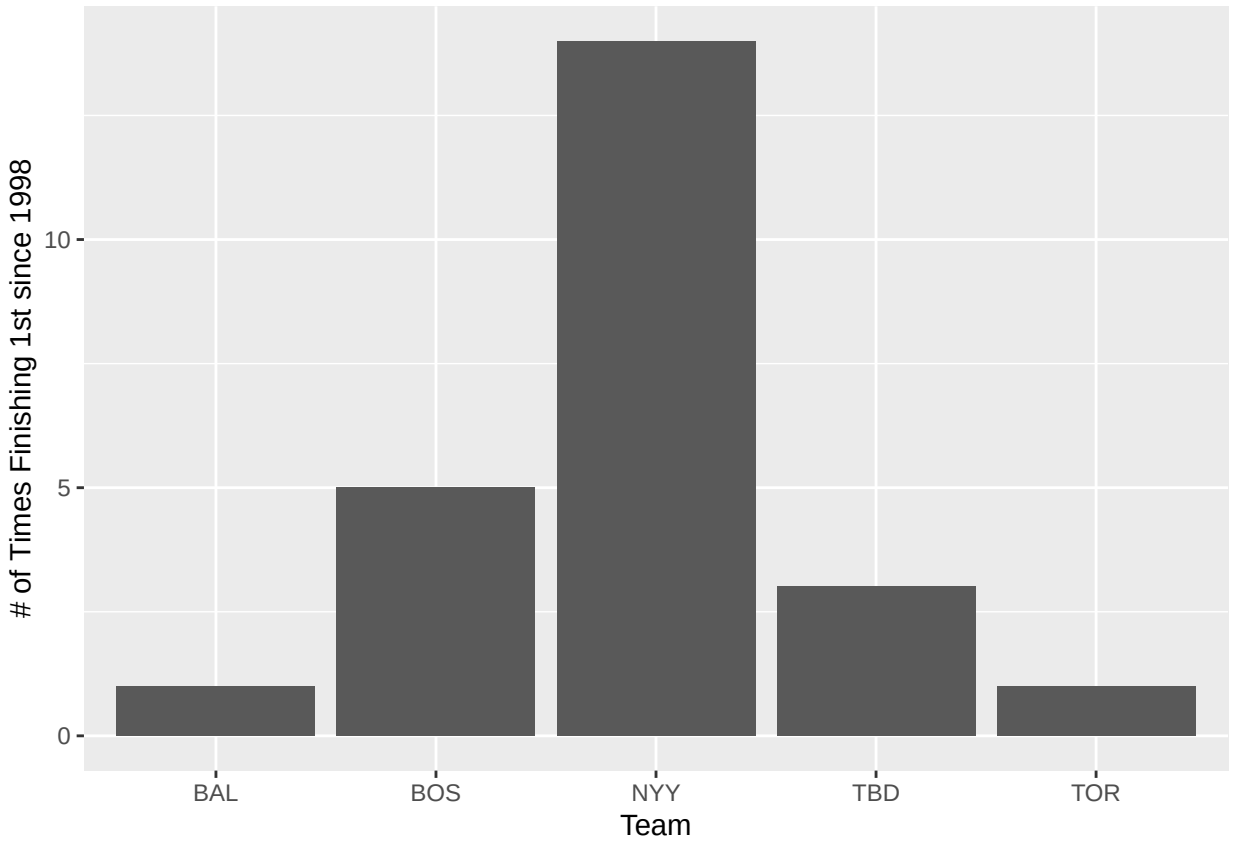


From this boxplot, I can tell that the NYY has the largest range of ages over the years and also has the largest median age.

The lowest year of pitching staff age belongs to the Rays and the smallest range of ages over the years belongs to Baltimore.

Continuing with the data analysis, I want to look at the outcome variable `Finish`.

```
ggplot(aes(x = Team, y = Finish),data = AL_EAST)+
       geom_col() + ylab("# of Times Finishing 1st since 1998")
```



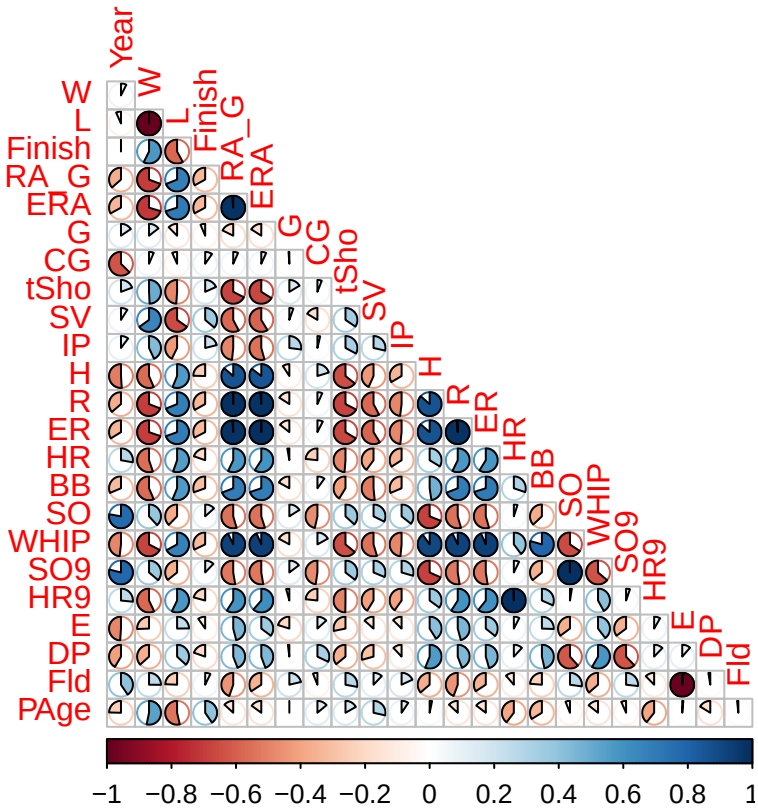

```
AL_EAST %>% count(Finish,Team) %>% filter(Finish == 1)
```

```
## # A tibble: 5 x 3
##   Finish Team      n
##    <dbl> <chr> <int>
## 1      1 BAL       1
## 2      1 BOS       5
## 3      1 NYY      14
## 4      1 TBD       3
## 5      1 TOR       1
```

It looks like the Yankees have dominated the AL East in the past 25 years with over half of the 1st place finishes in the regular season.

I want to look at the correlation between different predictor variables.

```
AL_EAST %>%
  dplyr::select(is.numeric) %>%
  cor() %>%
  corrplot(type = "lower", diag = FALSE, method = "pie")
```



Interesting findings from correlation plot between of the numeric predictor variables:

- Looking at how all of the different numeric variables compare with the variable `Year`, I noticed that there is a general negative trend between time and most of the statistics
- Over time, there is a decline in `complete games` but an increase in `strikeouts per 9 innings`
- `Average pitching age` has no strong correlation with any other variable
- Higher `Runs Allowed per game` is extremely correlated with a higher ERA.
- `Fielding Percentage` has an extremely strong negative correlation with `Errors`. This means they are opposites.(Makes sense)

Model Building

Now that I have done a dive into the data between predictor variables, I want to do a more extensive comparison the data to the outcome variable, `Finish`. I want to find the best predictive variables for finishing first in the AL East. To do this, I will use a logistic regression model. This is used when I am working with a discrete outcome variable.

For the model, I can remove:

- LG -> all teams are in the same divison
- G -> practically every season was 162 games
- Year -> only one team can win the league every year

- W and L -> Higher win count will lead to higher chances at finishing first and vice versa for higher L count and not finishing first

```
AL_EAST$Team <- factor(AL_EAST$Team)

log_reg <- glm(Finish~ RA_G+ERA+CG+tSho+SV+ IP+H+R+ER+HR+BB+SO+WHIP+SO9+HR9+E+DP+Fld+PAge+Team, data = A
```

```
summary(log_reg)
```

```
##
## Call:
## glm(formula = Finish ~ RA_G + ERA + CG + tSho + SV + IP + H +
##     R + ER + HR + BB + SO + WHIP + SO9 + HR9 + E + DP + Fld +
##     PAge + Team, data = AL_EAST)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.73849  -0.19290  -0.04953   0.13476   0.79869
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.798e+01  1.367e+02   0.497  0.62013
## RA_G         2.287e+00  3.335e+00   0.686  0.49446
## ERA         -3.591e+00  8.721e+00  -0.412  0.68147
## CG           1.251e-02  9.356e-03   1.337  0.18451
## tSho        -6.081e-03  1.376e-02  -0.442  0.65946
## SV           6.504e-03  5.449e-03   1.194  0.23551
## IP           4.295e-02  5.286e-02   0.813  0.41851
## H           -4.385e-02  4.268e-02  -1.027  0.30683
## R           -1.862e-02  2.069e-02  -0.900  0.37041
## ER           2.474e-02  5.485e-02   0.451  0.65291
## HR          -2.074e-03  6.297e-02  -0.033  0.97379
## BB          -4.304e-02  4.261e-02  -1.010  0.31491
## SO           1.166e-02  2.151e-02   0.542  0.58891
## WHIP         6.285e+01  6.142e+01   1.023  0.30877
## SO9         -1.951e+00  3.443e+00  -0.567  0.57222
## HR9          1.167e+00  1.010e+01   0.116  0.90825
## E           -1.947e-02  1.682e-02  -1.158  0.24987
## DP           1.081e-03  2.733e-03   0.395  0.69335
## Fld         -1.315e+02  1.022e+02  -1.287  0.20128
## PAge         8.713e-02  3.294e-02   2.645  0.00955 **
## TeamBOS      7.774e-02  1.363e-01   0.570  0.56969
## TeamNYY      2.658e-01  1.448e-01   1.835  0.06956 .
## TeamTBD      6.908e-02  1.172e-01   0.589  0.55709
## TeamTOR      7.181e-05  1.120e-01   0.001  0.99949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1151578)
##
##     Null deviance: 19.200  on 119  degrees of freedom
## Residual deviance: 11.055  on  96  degrees of freedom
## AIC: 104.39
```

```
##
## Number of Fisher Scoring iterations: 2
```

It looks like there is collinearity between multiple predictor variables. This means that I want to reduce the number of predictor variables and find the best fitting model for the data set. To do that I will use stepAIC and lasso regression. I am using lasso regression instead of ridge regression because I want to completely remove the variables that aren't influential and are the cause of multicollinearity.

Step_AIC

The best logistic regression model from the step AIC function was:

```
best_log_AIC <- glm(Finish ~CG+SV+H+R+BB+SO+WHIP+SO9+HR9+E+Fld+PAge, data = AL_EAST)
```

```
summary(best_log_AIC)
```

```
##
## Call:
## glm(formula = Finish ~ CG + SV + H + R + BB + SO + WHIP + SO9 +
##     HR9 + E + Fld + PAge, data = AL_EAST)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7063  -0.2191  -0.0686   0.1763   0.7455
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.955e+02  9.067e+01   2.156  0.03331 *
## CG           1.387e-02  8.827e-03   1.572  0.11894
## SV           9.699e-03  5.066e-03   1.915  0.05821 .
## H           -1.321e-02  8.000e-03  -1.652  0.10149
## R           -2.996e-03  1.432e-03  -2.092  0.03882 *
## BB          -1.232e-02  8.060e-03  -1.528  0.12947
## SO           2.679e-02  1.193e-02   2.246  0.02677 *
## WHIP         1.932e+01  1.159e+01   1.666  0.09862 .
## SO9         -4.389e+00  1.916e+00  -2.291  0.02393 *
## HR9          9.979e-01  3.367e-01   2.963  0.00375 **
## E           -3.100e-02  1.503e-02  -2.063  0.04157 *
## Fld         -1.983e+02  9.054e+01  -2.190  0.03068 *
## PAge         1.148e-01  2.483e-02   4.624 1.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1118911)
##
##     Null deviance: 19.200  on 119  degrees of freedom
## Residual deviance: 11.972  on 107  degrees of freedom
## AIC: 91.958
##
## Number of Fisher Scoring iterations: 2
```

The significant variables are all of those that have a p value less than 0.05. If the estimate for the coefficient is negative, then there is a negative correlation between that predictor variable and finishing first in the AL East.

For example, an increase in BB or Balls by the pitching staff means a decrease in finish placing since the estimate is negative.

Lasso Regression

```
pred_var <- data.matrix(AL_EAST[, c('RA_G', 'ERA','CG','tSho','SV','IP','H','R','ER', 'HR','BB','SO','W
```

This vector makes it easy to use the cv.glmnet function. Important to note about the cv.glmnet function is it uses a baseline of k=10 k-folds cross validation.

```
lasso_mod <- cv.glmnet(pred_var, AL_EAST$Finish,alpha = 1)

best_lambda <- lasso_mod$lambda.min

best_lambda
```

```
## [1] 0.01435226
```

```
best_lasso <- glmnet(pred_var, AL_EAST$Finish,alpha = 1, lambda = best_lambda)

coef(best_lasso)
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -3.4327460922
## RA_G         .
## ERA          .
## CG           0.0081425852
## tSho        -0.0002231102
## SV           0.0079640667
## IP           0.0010146372
## H           -0.0002030437
## R           -0.0007548238
## ER           .
## HR           0.0016287950
## BB           .
## SO           .
## WHIP         .
## SO9          .
## HR9          .
## E            .
## DP           .
## Fld          .
## PAge         0.0813033863
## Team         .
```

According to lasso regression, any of the variables without a value next to their name got dropped because they were not significant to the model.

```
lasso_final_log <- glm(Finish~ CG+tSho+SV+IP+H+R+HR+PAge,data = AL_EAST)

summary(lasso_final_log)
```

```
##
## Call:
## glm(formula = Finish ~ CG + tSho + SV + IP + H + R + HR + PAge,
##     data = AL_EAST)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6178  -0.2451  -0.1055   0.1957   0.7559
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.8418868  3.9217061  -1.235   0.2196
## CG           0.0162763  0.0082873   1.964   0.0520 .
## tSho        -0.0138540  0.0127283  -1.088   0.2788
## SV           0.0093425  0.0051285   1.822   0.0712 .
## IP           0.0019325  0.0025738   0.751   0.4543
## H           -0.0005379  0.0007748  -0.694   0.4889
## R           -0.0011295  0.0009991  -1.130   0.2607
## HR           0.0035187  0.0017689   1.989   0.0491 *
## PAge         0.0995985  0.0225020   4.426 2.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1195172)
##
##     Null deviance: 19.200  on 119  degrees of freedom
## Residual deviance: 13.266  on 111  degrees of freedom
## AIC: 96.274
##
## Number of Fisher Scoring iterations: 2
```

Conclusion

According to variable selection used on the logistic regression model, it seems like there is one singular strong predictor for `Finish` outcome.

`Pitcher age` is a very strong predictor when it comes to finishing first in the AL East. A reason for this could be because of the high pressure and high stake games in the AL East(playing at Fenway and Yankees Stadium). More mature and experienced pitchers are better at handling the pressure and gaining crucial wins to win the division.

According to Step_AIC, some other solid predictor variables for predicting a first place finish in the AL East are HR9 and SO9. Interestingly, an increase in home runs given up per 9 innings lead to a decrease in finishing placement. On the other hand, an increase in Strike outs per 9 innings lead to an increase in finishing placement.

Next Steps

A next step for this project is to add in data from the 5 other divisions in baseball. Once I have all of the pitching data from all the teams, I can do division comparisons and find trends within the AL East compared to other divisons in the league.

In addition, the introduction of 5 times as much data would allow me to do training splits and possible build a predictive model about winning a division across the entirety of the MLB.

Another route to go would be to add the batting data on the baseball reference website. I think that this route would be a hassle and not the route to go because there would be too many predictor variables compared to observations.