Project EDA

George Daher

3/28/2023

Background

All the data used is pulled from either baseball reference using the baseballr R package, or from Retrosheet game logs. All the data in both databases is complete for the years (2012 to 2019) that we are considering in our analysis. The objective of our analysis is to determine whether there exists a predictive relationship between previous hitting statistics and order in the lineup for hitters/lineups in the MLB. XXXXX

Cleaning the Data

The first step of cleaning the data requires defining a set of helper functions to make the data. The first function cleans the game log files by removing the unnecessary columns and labeling the remaining ones so it is easier to handle for a merge. The second function uses the baseballr packages in order to pull hitting splits from baseball reference from the specified year. The two merge functions merge the hitting splits and game logs into one dataset that has one hitter per row including information about their lineup position, basic information about the game played and previous year hitting splits.

```
library(baseballr); library(janitor); library(RcppParallel); library(lubridate);library(dplyr);library(s
```

```
## Warning: package 'baseballr' was built under R version 4.1.3
```

```
## Warning: package 'janitor' was built under R version 4.1.3
```

```
## Warning: package 'RcppParallel' was built under R version 4.1.3
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
cleanLogs <- function(logs) {
  outlogs <- logs[-c(2:3,12:89,94:105,160:161)]
  colnames(outlogs) <- c("Date","VisitingTeam","VisitingTeamLeague","VisitingGame#","HomeTeam",
                         "HomeTeamLeague","HomeGame#","VisitingScore","HomeScore","visitingManagerID",
                         "visitingManagerName","homeManagerID","homeManagerName",
                         "visitor1ID","visitor1Name","visitor1Position",
                         "visitor2ID","visitor2Name","visitor2Position",
                         "visitor3ID","visitor3Name","visitor3Position",
```

```
                          "visitor4ID","visitor4Name","visitor4Position",
                          "visitor5ID","visitor5Name","visitor5Position",
                          "visitor6ID","visitor6Name","visitor6Position",
                          "visitor7ID","visitor7Name","visitor7Position",
                          "visitor8ID","visitor8Name","visitor8Position",
                          "visitor9ID","visitor9Name","visitor9Position",
                          "home1ID","home1Name","home1Position",
                          "home2ID","home2Name","home2Position",
                          "home3ID","home3Name","home3Position",
                          "home4ID","home4Name","home4Position",
                          "home5ID","home5Name","home5Position",
                          "home6ID","home6Name","home6Position",
                          "home7ID","home7Name","home7Position",
                          "home8ID","home8Name","home8Position",
                          "home9ID","home9Name","home9Position")
  return(outlogs)
}
yearSplits <- function(year) {
  splits <- data.frame(bref_daily_batter(paste(year,"01","01",sep="-"),paste(year,"12","31",sep="-")))
  splits <- splits[-c(1:2,4,5)]
  splits$Name <- iconv(splits$Name,from="UTF-8",to="ASCII//TRANSLIT")
  splits$Name <- str_replace_all(splits$Name," Jr\\.","")
  return(splits)
}
mergeAll <- function(logs,splits) {
  out <- mergePosition(logs,splits,"visitor",1)
  out <- rbind(out,mergePosition(logs,splits,"visitor",2))
  out <- rbind(out,mergePosition(logs,splits,"visitor",3))
  out <- rbind(out,mergePosition(logs,splits,"visitor",4))
  out <- rbind(out,mergePosition(logs,splits,"visitor",5))
  out <- rbind(out,mergePosition(logs,splits,"visitor",6))
  out <- rbind(out,mergePosition(logs,splits,"visitor",7))
  out <- rbind(out,mergePosition(logs,splits,"visitor",8))
  out <- rbind(out,mergePosition(logs,splits,"visitor",9))
  out <- rbind(out,mergePosition(logs,splits,"home",1))
  out <- rbind(out,mergePosition(logs,splits,"home",2))
  out <- rbind(out,mergePosition(logs,splits,"home",3))
  out <- rbind(out,mergePosition(logs,splits,"home",4))
  out <- rbind(out,mergePosition(logs,splits,"home",5))
  out <- rbind(out,mergePosition(logs,splits,"home",6))
  out <- rbind(out,mergePosition(logs,splits,"home",7))
  out <- rbind(out,mergePosition(logs,splits,"home",8))
  out <- rbind(out,mergePosition(logs,splits,"home",9))
}
mergePosition <- function(logs,splits,team,num){
  logs[,paste0(team,num,"Name")] <- gsub("\\.","",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("i-M","i M",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("n-J","n J",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("Dee Gordon","Dee Strange-Gordon",logs[,paste0(team,num,"Name"]
  logs[,paste0(team,num,"Name")] <- gsub("Giovanny Urshela","Gio Urshela",logs[,paste0(team,num,"Name")]
  logs[,paste0(team,num,"Name")] <- gsub("Michael Taylor","Michael A. Taylor",logs[,paste0(team,num,"Na
  logs[,paste0(team,num,"Name")] <- gsub("Vincent Velasquez","Vince Velasquez",logs[,paste0(team,num,"Na
  logs[,paste0(team,num,"Name")] <- gsub("Michael Brosseau","Mike Brosseau",logs[,paste0(team,num,"Name
```

```
  logs[,paste0(team,num,"Name")] <- gsub("Nate Lowe","Nathanial Lowe",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("Phillip Ervin","Phil Ervin",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("Josh Fuentes","Joshua Fuentes",logs[,paste0(team,num,"Name")])
  logs[,paste0(team,num,"Name")] <- gsub("Yulieski Gurriel","Yuli Gurriel",logs[,paste0(team,num,"Name")
  logs[,paste0(team,num,"Name")] <- gsub("Steve Wilkerson","Stevie Wilkerson",logs[,paste0(team,num,"Na
  logs[,paste0(team,num,"Name")] <- gsub("Mike Soroka","Michael Soroka",logs[,paste0(team,num,"Name")])
  out <- merge(splits,logs,by.x="Name",by.y=paste0(team,num,"Name"))
  out <- mutate(out[,c(1:35,37,39)],homeAway=team,lineupPosition=as.numeric(num))
}
```

With all these helper functions defined, actually importing and merging the data for each year is trivial.

```
gl2012 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2012.txt", header=FALSE))
gl2013 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2013.txt", header=FALSE))
gl2014 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2014.txt", header=FALSE))
gl2015 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2015.txt", header=FALSE))
gl2016 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2016.txt", header=FALSE))
gl2017 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2017.txt", header=FALSE))
gl2018 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2018.txt", header=FALSE))
gl2019 <- cleanLogs(read.csv("~/College/MATH 203/Project Data/gameLogs/gl2019.txt", header=FALSE))
split2012 <- yearSplits(2012)
split2013 <- yearSplits(2013)
split2014 <- yearSplits(2014)
split2015 <- yearSplits(2015)
split2016 <- yearSplits(2016)
split2017 <- yearSplits(2017)
split2018 <- yearSplits(2018)
split2019 <- yearSplits(2019)
master <- mergeAll(gl2012,split2012)
master <- rbind(master,mergeAll(gl2013,split2013))
master <- rbind(master,mergeAll(gl2014,split2014))
master <- rbind(master,mergeAll(gl2015,split2015))
master <- rbind(master,mergeAll(gl2016,split2016))
master <- rbind(master,mergeAll(gl2017,split2017))
master <- rbind(master,mergeAll(gl2018,split2018))
master <- rbind(master,mergeAll(gl2019,split2019))
```

Data Structure

After cleaning and refactoring the data to be in a very usable state, there end up being 336494 observations of 39 different measurements. Note that many statistical tests will be performed using samples from the dataset as considering the entire dataset would be unnecessary. WIth all this set up, we can explore the data.

```
##  [1] "Name"              "Team"              "G"
##  [4] "PA"                "AB"                "R"
##  [7] "H"                 "X1B"               "X2B"
## [10] "X3B"               "HR"                "RBI"
## [13] "BB"                "IBB"               "uBB"
## [16] "SO"                "HBP"               "SH"
## [19] "SF"                "GDP"               "SB"
## [22] "CS"                "BA"                "OBP"
```

```
## [25] "SLG"                "OPS"                "Date"
## [28] "VisitingTeam"       "VisitingTeamLeague" "VisitingGame#"
## [31] "HomeTeam"           "HomeTeamLeague"     "HomeGame#"
## [34] "VisitingScore"      "HomeScore"          "visitingManagerName"
## [37] "homeManagerName"    "homeAway"           "lineupPosition"
```

The response variable for the dataset is the last column: "lineupPosition." All other variables can be split into two broad categories. The hitting statistics are the main set of explanatory variables that we will analyze with respect to the response and other variables such as team, manager and game number exist to address potentially confounding variables that are worth considering but not the main variables of interest.

```
str(master)
```

```
## 'data.frame':    334441 obs. of  39 variables:
##  $ Name               : chr  "Aaron Hill" "Aaron Hill" "Adam Eaton" "Adam Eaton" ...
##  $ Team               : chr  "Arizona" "Arizona" "Arizona" "Arizona" ...
##  $ G                  : num  155 155 22 22 22 22 22 22 22 22 ...
##  $ PA                 : num  668 668 103 103 103 103 103 103 103 103 ...
##  $ AB                 : num  609 609 85 85 85 85 85 85 85 85 ...
##  $ R                  : num  93 93 19 19 19 19 19 19 19 19 ...
##  $ H                  : num  184 184 22 22 22 22 22 22 22 22 ...
##  $ X1B                : num  108 108 15 15 15 15 15 15 15 15 ...
##  $ X2B                : num  44 44 3 3 3 3 3 3 3 3 ...
##  $ X3B                : num  6 6 2 2 2 2 2 2 2 2 ...
##  $ HR                 : num  26 26 2 2 2 2 2 2 2 2 ...
##  $ RBI                : num  85 85 5 5 5 5 5 5 5 5 ...
##  $ BB                 : num  52 52 14 14 14 14 14 14 14 14 ...
##  $ IBB                : num  7 7 0 0 0 0 0 0 0 0 ...
##  $ uBB                : num  45 45 14 14 14 14 14 14 14 14 ...
##  $ SO                 : num  86 86 15 15 15 15 15 15 15 15 ...
##  $ HBP                : num  4 4 3 3 3 3 3 3 3 3 ...
##  $ SH                 : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ SF                 : num  2 2 0 0 0 0 0 0 0 0 ...
##  $ GDP                : num  15 15 0 0 0 0 0 0 0 0 ...
##  $ SB                 : num  14 14 2 2 2 2 2 2 2 2 ...
##  $ CS                 : num  3 3 2 2 2 2 2 2 2 2 ...
##  $ BA                 : num  0.302 0.302 0.259 0.259 0.259 0.259 0.259 0.259 0.259 0.259 ...
##  $ OBP                : num  0.36 0.36 0.382 0.382 0.382 0.382 0.382 0.382 0.382 0.382 ...
##  $ SLG                : num  0.522 0.522 0.412 0.412 0.412 0.412 0.412 0.412 0.412 0.412 ...
##  $ OPS                : num  0.882 0.882 0.794 0.794 0.794 0.794 0.794 0.794 0.794 0.794 ...
##  $ Date               : int  20120529 20120830 20120926 20120904 20120908 20120907 20120922 20120923
##  $ VisitingTeam       : chr  "ARI" "ARI" "ARI" "ARI" ...
##  $ VisitingTeamLeague : chr  "NL" "NL" "NL" "NL" ...
##  $ VisitingGame#      : int  50 132 155 137 140 139 151 152 138 153 ...
##  $ HomeTeam           : chr  "SFN" "LAN" "SFN" "SFN" ...
##  $ HomeTeamLeague     : chr  "NL" "NL" "NL" "NL" ...
##  $ HomeGame#          : int  50 132 155 136 140 139 151 152 137 153 ...
##  $ VisitingScore      : int  1 2 0 8 8 5 8 10 6 2 ...
##  $ HomeScore          : int  3 0 6 6 5 6 7 7 2 4 ...
##  $ visitingManagerName: chr  "Kirk Gibson" "Kirk Gibson" "Kirk Gibson" "Kirk Gibson" ...
##  $ homeManagerName    : chr  "Bruce Bochy" "Don Mattingly" "Bruce Bochy" "Bruce Bochy" ...
##  $ homeAway           : chr  "visitor" "visitor" "visitor" "visitor" ...
##  $ lineupPosition     : num  1 1 1 1 1 1 1 1 1 1 ...
```

Variables 1 and 2 serve as identifiers for the rows when combined with Variable 27 (Date). Variables 3 through 26 are explanatory variables. Variables 27 to 33 and 36 to 38 are alternative explanatory variables that may have some impact on the response but are not the focus of the analysis. Variable 39 is the response variable.

By creating a vector with the indexes for the numerical variables of interest, analysis of the distribution becomes easier.

```
inum <- c(3:26,30,33,39)
icat <- c(28,31,38)

summary(master[,inum])
```

```
##        G                PA              AB              R         
##  Min.   :  1.0   Min.   :  1.0   Min.   :  0.0   Min.   :  0.00  
##  1st Qu.: 85.0   1st Qu.:282.0   1st Qu.:253.0   1st Qu.: 30.00  
##  Median :125.0   Median :472.0   Median :423.0   Median : 53.00  
##  Mean   :111.5   Mean   :432.5   Mean   :387.4   Mean   : 52.46  
##  3rd Qu.:147.0   3rd Qu.:606.0   3rd Qu.:542.0   3rd Qu.: 75.00  
##  Max.   :162.0   Max.   :747.0   Max.   :684.0   Max.   :137.00  
##                                                                  
##        H              X1B             X2B             X3B        
##  Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   : 0.000  
##  1st Qu.: 61.0   1st Qu.: 39.00   1st Qu.:12.00   1st Qu.: 0.000  
##  Median :107.0   Median : 68.00   Median :20.00   Median : 1.000  
##  Mean   :101.8   Mean   : 65.95   Mean   :20.26   Mean   : 2.038  
##  3rd Qu.:145.0   3rd Qu.: 92.00   3rd Qu.:29.00   3rd Qu.: 3.000  
##  Max.   :225.0   Max.   :170.00   Max.   :58.00   Max.   :15.000  
##                                                                   
##        HR             RBI              BB             IBB        
##  Min.   : 0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.0  
##  1st Qu.: 5.00   1st Qu.: 27.00   1st Qu.: 19.00   1st Qu.: 0.0  
##  Median :12.00   Median : 50.00   Median : 34.00   Median : 1.0  
##  Mean   :13.56   Mean   : 50.32   Mean   : 36.62   Mean   : 2.5  
##  3rd Qu.:21.00   3rd Qu.: 73.00   3rd Qu.: 51.00   3rd Qu.: 3.0  
##  Max.   :59.00   Max.   :139.00   Max.   :143.00   Max.   :29.0  
##                                                                  
##       uBB              SO              HBP             SH        
##  Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 18.00   1st Qu.: 55.00   1st Qu.: 1.000   1st Qu.: 0.000  
##  Median : 31.00   Median : 85.00   Median : 3.000   Median : 0.000  
##  Mean   : 34.12   Mean   : 85.61   Mean   : 4.022   Mean   : 1.457  
##  3rd Qu.: 48.00   3rd Qu.:115.00   3rd Qu.: 6.000   3rd Qu.: 2.000  
##  Max.   :128.00   Max.   :222.00   Max.   :31.000   Max.   :17.000  
##                                                                     
##        SF              GDP              SB              CS        
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 1.000   1st Qu.: 4.000   1st Qu.: 1.000   1st Qu.: 0.000  
##  Median : 3.000   Median : 8.000   Median : 3.000   Median : 1.000  
##  Mean   : 2.969   Mean   : 8.644   Mean   : 6.536   Mean   : 2.068  
##  3rd Qu.: 4.000   3rd Qu.:12.000   3rd Qu.: 9.000   3rd Qu.: 3.000  
##  Max.   :15.000   Max.   :31.000   Max.   :64.000   Max.   :23.000  
##                                                                     
##        BA              OBP              SLG              OPS        
```

```
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.2310   1st Qu.:0.2940   1st Qu.:0.3570   1st Qu.:0.660
##  Median :0.2560   Median :0.3220   Median :0.4120   Median :0.735
##  Mean   :0.2492   Mean   :0.3141   Mean   :0.4029   Mean   :0.717
##  3rd Qu.:0.2790   3rd Qu.:0.3470   3rd Qu.:0.4630   3rd Qu.:0.805
##  Max.   :1.0000   Max.   :1.0000   Max.   :2.0000   Max.   :2.667
##  NA's   :6        NA's   :4        NA's   :6        NA's   :6
##  VisitingGame#      HomeGame#      lineupPosition
##  Min.   :  1.00   Min.   :  1.00   Min.   :1.000
##  1st Qu.: 41.00   1st Qu.: 41.00   1st Qu.:3.000
##  Median : 82.00   Median : 81.00   Median :5.000
##  Mean   : 81.46   Mean   : 81.46   Mean   :4.994
##  3rd Qu.:122.00   3rd Qu.:122.00   3rd Qu.:7.000
##  Max.   :163.00   Max.   :163.00   Max.   :9.000
##
```

```
table(master$visitingManagerName)
```

```
##
##        A.J. Hinch       Aaron Boone     Alan Trammell         Alex Cora
##              7114              2718                18              2576
##        Andy Green         Bo Porter        Bob Melvin   Bobby Valentine
##              5549              2437             11298              1427
##       Brad Ausmus        Brad Mills      Brandon Hyde     Brian Snitker
##              6886              1019              1440              5270
##       Bruce Bochy       Bryan Price    Buck Showalter       Buddy Black
##             11196              5826              9564              8986
##    Charlie Manuel   Charlie Montoyo        Chip Hale     Chris Speier
##              2584              1443              2872                89
##    Chris Woodward      Clint Hurdle    Craig Counsell       Dale Sveum
##              1448             11262              6762              2887
##      Dan Jennings     Dave Martinez      Dave Roberts     Davey Johnson
##              1040              2775              5670              2839
##        David Bell      DeMarlo Hale       Dick Scott        Dino Ebel
##              1381                35                17                46
##     Don Mattingly     Don Wakamatsu       Dusty Baker       Eric Wedge
##             10888               108              5683              2582
## Freddie Benavides    Fredi Gonzalez       Gabe Kapler     Jeff Banister
##                72              5839              2556              5473
##      Jerry Narron       Jim Leyland     Jim Riggleman         Jim Tracy
##                18              2871              1084              1449
##       Joe Girardi        Joe Maddon       Joe McEwing      John Farrell
##              8529             11167                27              8003
##      John Gibbons        Kevin Cash       Kirk Gibson   Lloyd McClendon
##              8040              6951              4238              2719
##        Manny Acta     Matt Williams   Mickey Callaway      Mike Matheny
##              1426              2776              2610              9314
##      Mike Redmond     Mike Scioscia       Mike Shildt          Ned Yost
##              3151              9662              2012             11305
##     Ozzie Guillen       Pat Murphy      Paul Molitor     Pete Mackanin
##              1446               851              5605              3618
##     Rich Renteria    Robby Thompson     Robin Ventura    Rocco Baldelli
##              5303               255              6641              1389
##       Rod Barajas    Ron Gardenhire      Ron Roenicke    Ron Washington
```

```
##               52             6825             4391             3936
##         Ron Wotus    Ryne Sandberg    Scott Servais    Terry Collins
##               53             2339             5608             8474
##    Terry Francona       Tim Bogar      Tom Lawless     Tom Runnells
##             9629               90              250               70
##  Tony DeFrancesco     Tony Lovullo     Trey Hillman       Walt Weiss
##              372             4617               51             5549
```

```
table(master$homeManagerName)
```

```
##
##        A.J. Hinch       Aaron Boone     Alan Trammell         Alex Cora
##              7109              2706                54              2574
##        Andy Green         Bo Porter        Bob Melvin   Bobby Valentine
##              5504              2550             11312              1426
##       Brad Ausmus        Brad Mills     Brandon Hyde     Brian Snitker
##              6857              1175              1440              5263
##       Bruce Bochy       Bryan Price    Buck Showalter       Buddy Black
##             11258              5783              9447              9037
##    Charlie Manuel   Charlie Montoyo        Chip Hale      Chris Speier
##              2447              1444              2871               157
##    Chris Woodward      Clint Hurdle    Craig Counsell        Dale Sveum
##              1444             11156              6820              2892
##      Dan Jennings     Dave Martinez     Dave Roberts     Davey Johnson
##              1072              2793              5703              2828
##        David Bell      DeMarlo Hale       Don Cooper     Don Mattingly
##              1400                36                17             10844
##     Don Wakamatsu       Dusty Baker       Eric Wedge Freddie Benavides
##                45              5575              2612                53
##    Fredi Gonzalez       Gabe Kapler       Gene Lamont     Jeff Banister
##              5888              2562                16              5552
##       Jim Leyland     Jim Riggleman         Jim Tracy       Joe Girardi
##              2878              1070              1439              8516
##        Joe Maddon       Joe McEwing      John Farrell      John Gibbons
##             11171                28              7993              8065
##         Josh Bard        Kevin Cash       Kirk Gibson   Lloyd McClendon
##                16              7071              4203              2686
##        Manny Acta       Mark Parent     Matt Williams   Mickey Callaway
##              1330                17              2783              2606
##      Mike Matheny      Mike Redmond     Mike Scioscia       Mike Shildt
##              9327              3097              9694              2004
##          Ned Yost     Ozzie Guillen        Pat Murphy      Paul Molitor
##             11272              1438               821              5618
##     Pete Mackanin     Rich Renteria    Robby Thompson     Robin Ventura
##              3657              5269               229              6608
##    Rocco Baldelli       Rod Barajas    Ron Gardenhire      Ron Roenicke
##              1397                92              6830              4404
##    Ron Washington     Ryne Sandberg      Sandy Alomar     Scott Servais
##              3793              2434               107              5562
##     Terry Collins    Terry Francona   Terry Steinbach         Tim Bogar
##              8514              9485                34               236
##       Tom Lawless        Tom Prince  Tony DeFrancesco      Tony Lovullo
##               154                52               355              4694
##      Trent Jewett        Walt Weiss
```

```
##                51              5619
```

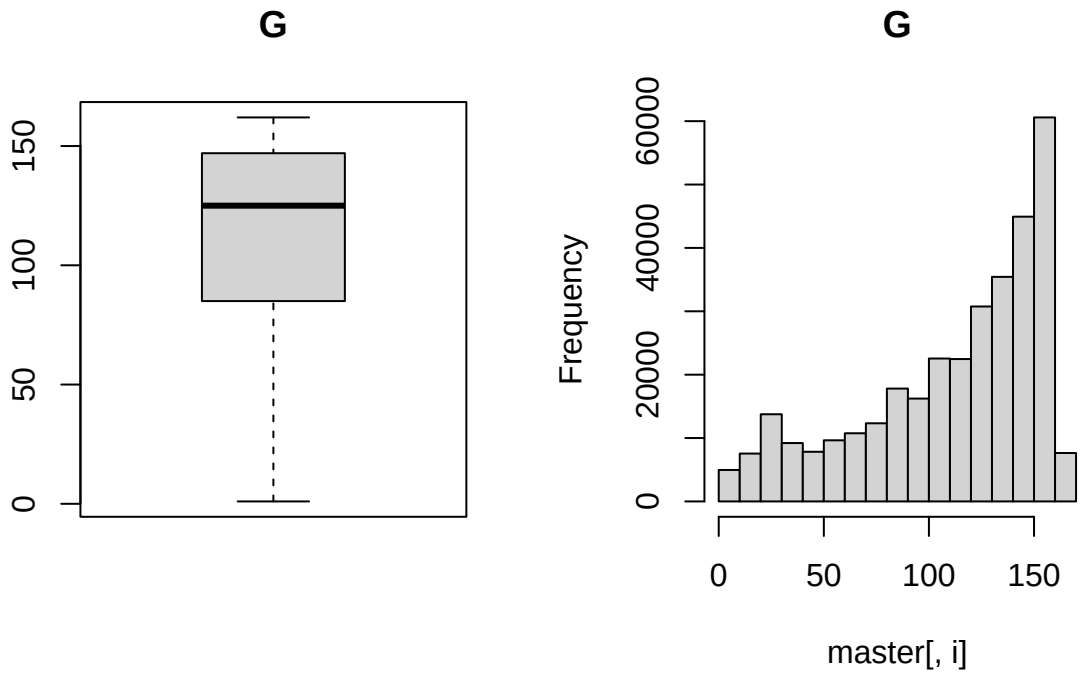
Before examining the distributions and bar graphs for some of the variables, it is important to consider that the distribution of some of the variables is trivial. Most notably, the distribution of the response variable: lineup position, will be compleely uniform by the way the data has been collected and by simple fact from the rules of baseball.

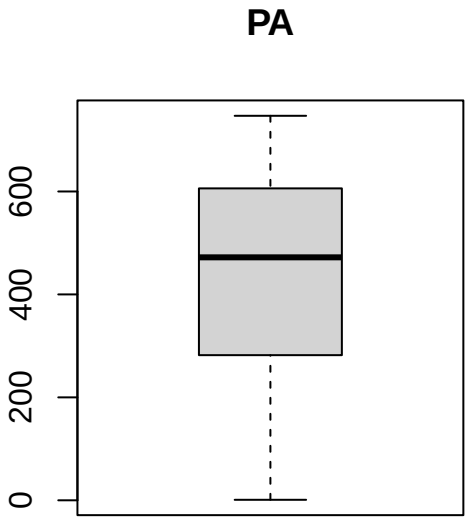
Similarly, many of the alternative explantory variables follow uniform categorical distributions for this same reason. These variables include the categorical variables of team and whether the player was on the home or away team (variable homeAway). This also includes the numeric variables of home and visitor game number and date of the game.

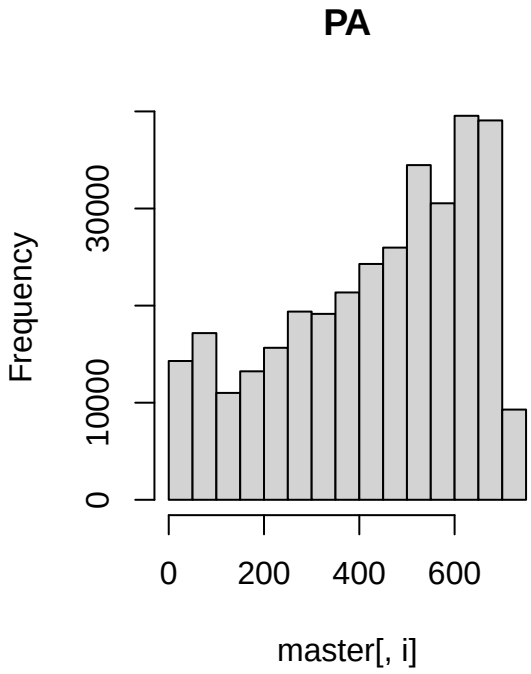
Of all these uniformly distirbuted some will not be exactly uniformly distributed because there was some data loss when cleaning and merging the data, but the fact that the actual distributions for these variables is still important.

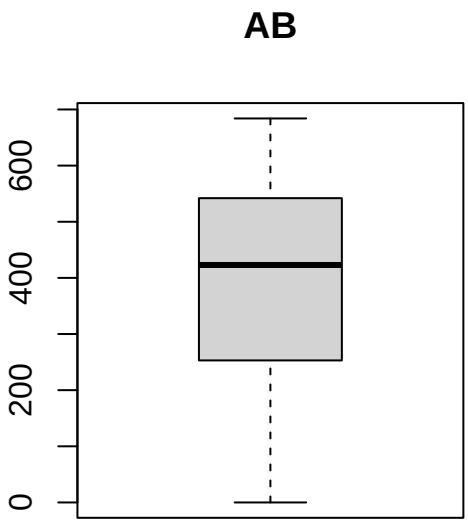
Having acknowledged all this, we can make boxplots and histograms for the numeric variables.

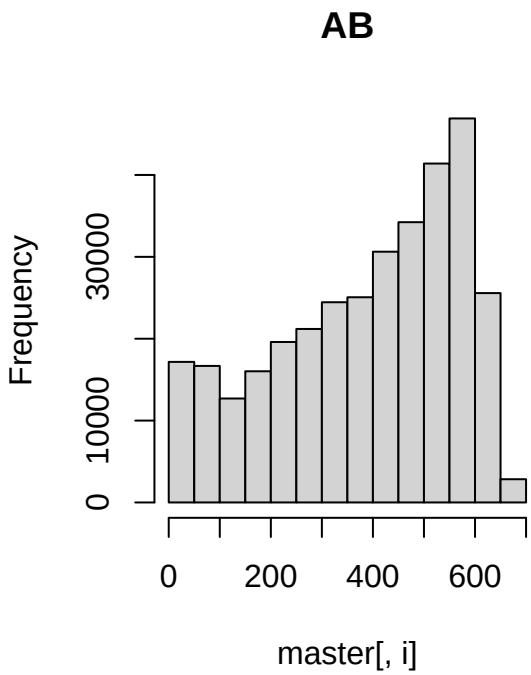
```
par(mfrow=c(1,2))
for(i in inum) {
  boxplot(master[,i],main=names(master[i]),type="l")
  hist(master[,i],main=names(master[i]))
}
```

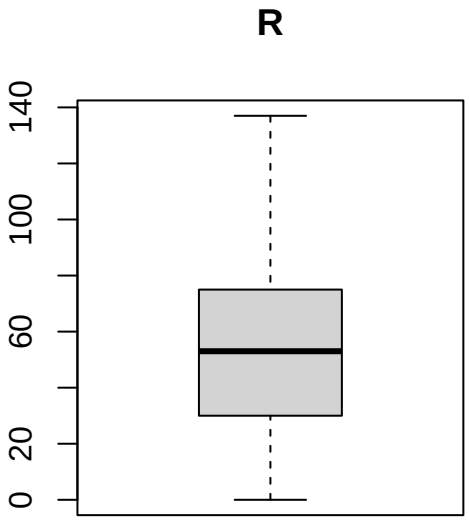


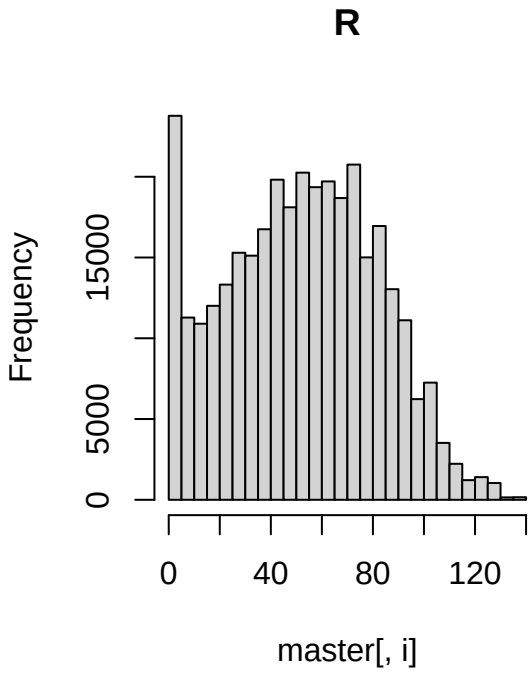

PA
0
200
400
600

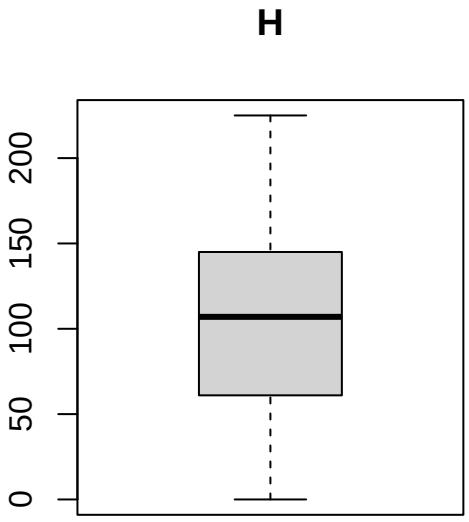

PA
Frequency
0
10000
30000
0
200
400
600
master[, i]

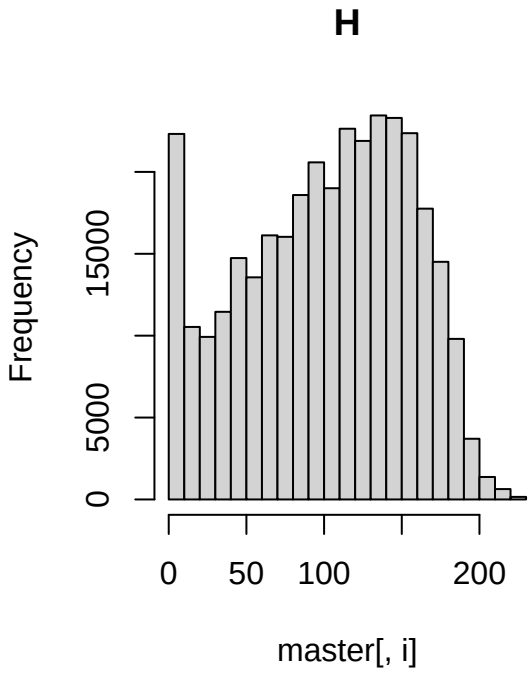

AB
0
200
400
600

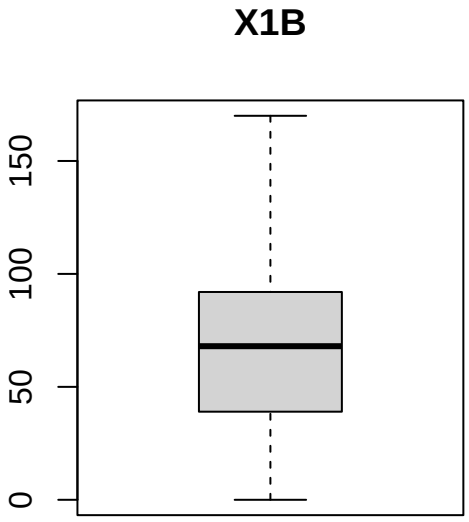

AB
Frequency
0
10000
30000
0
200
400
600
master[, i]

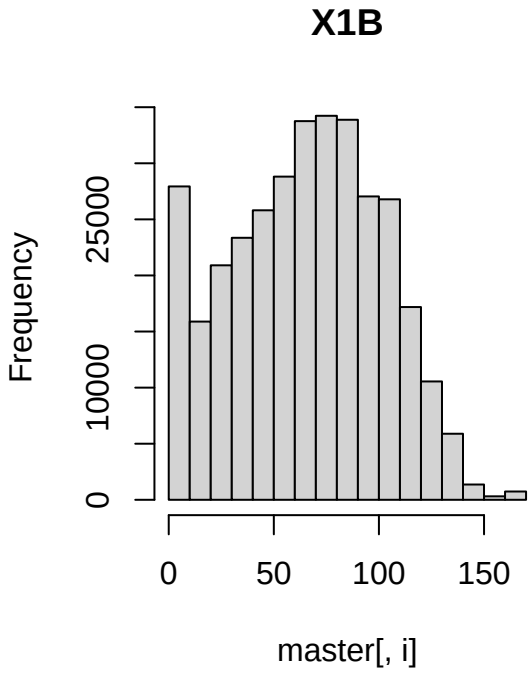

R
0
20
60
100
140

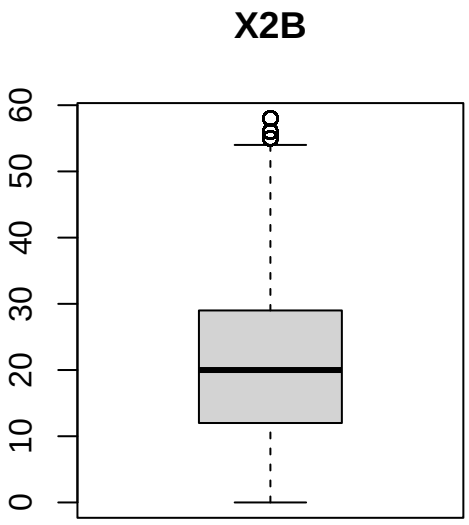

R
Frequency
0
5000
15000
0
40
80
120
master[, i]

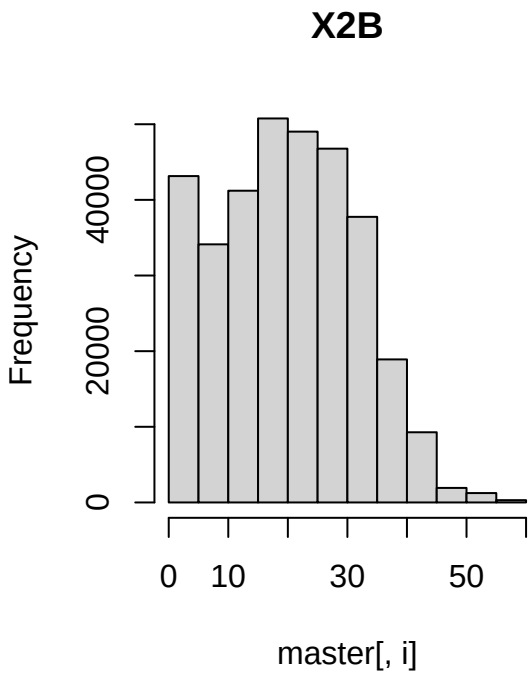

H
0
50
100
150
200

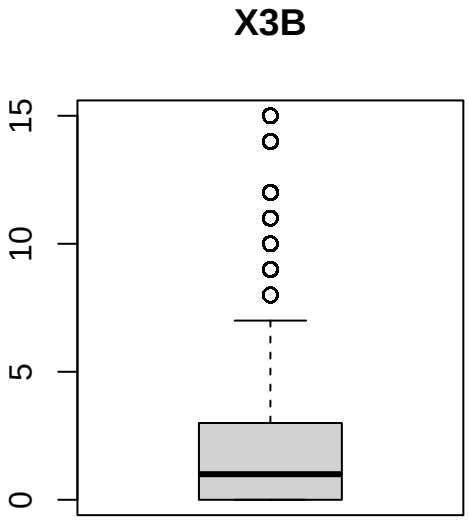

H
Frequency
0
5000
15000
0
50
100
200
master[, i]

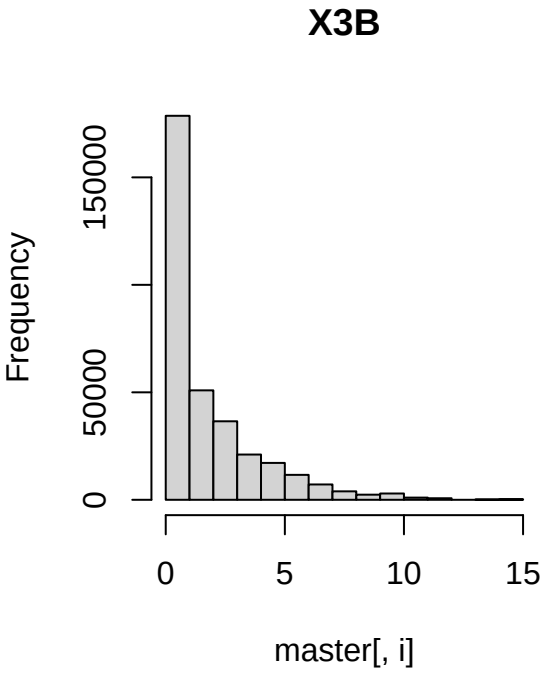

X1B
0
50
100
150

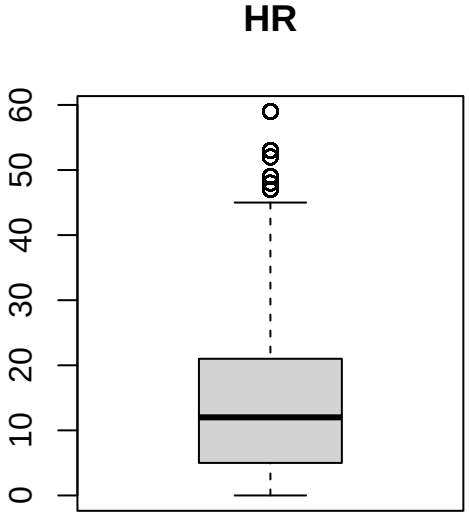

X1B
Frequency
0
10000
25000
0
50
100
150
master[, i]

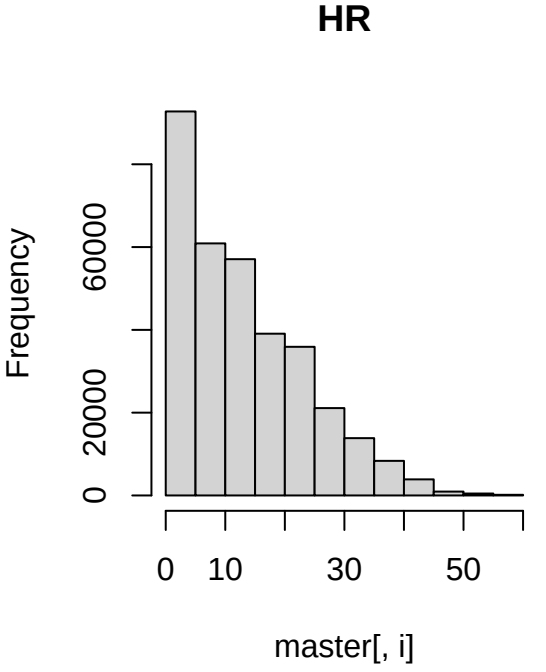

X2B
0
10
20
30
40
50
60

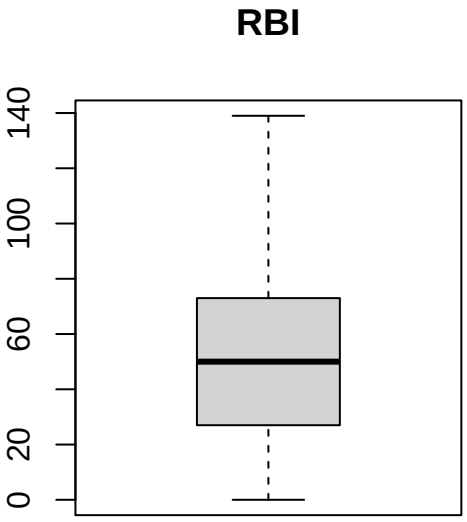

X2B
Frequency
0
20000
40000
0
10
30
50
master[, i]

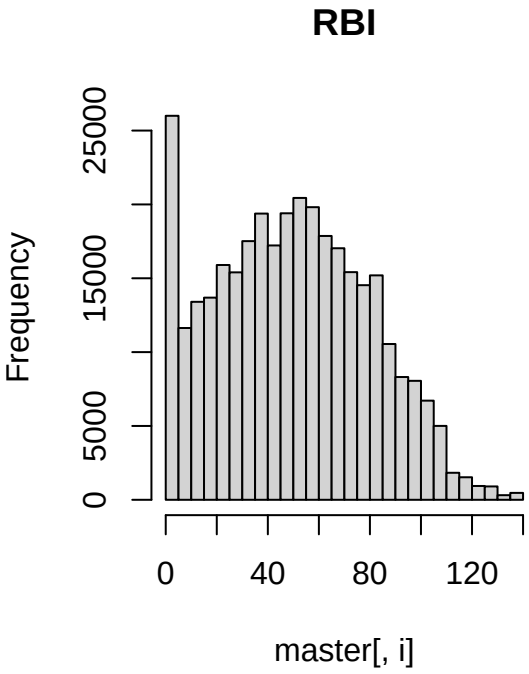

X3B
0
5
10
15

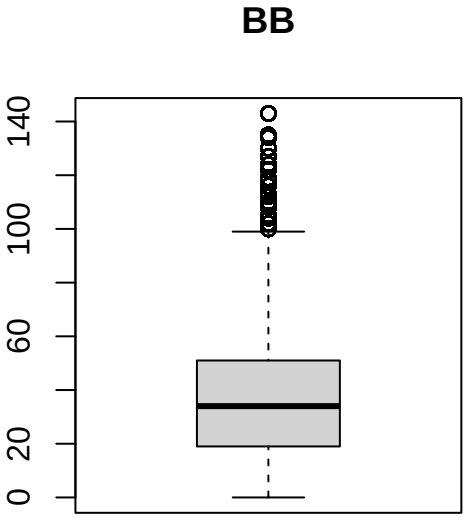

X3B
Frequency
0
50000
150000
0
5
10
15
master[, i]

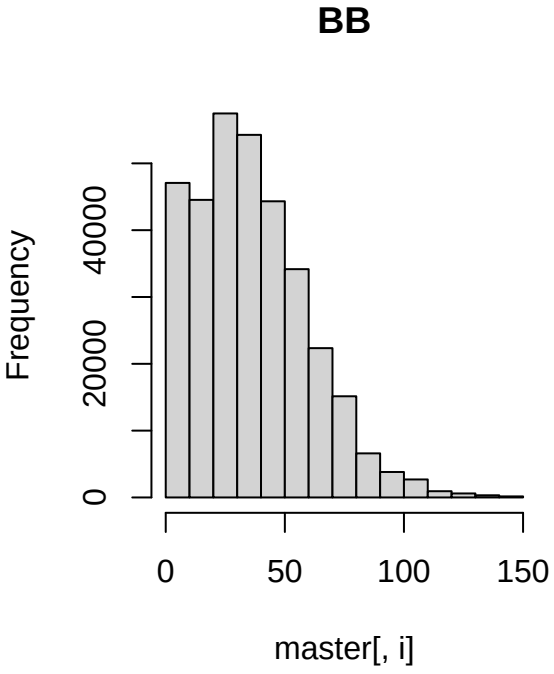

HR
0
10
20
30
40
50
60

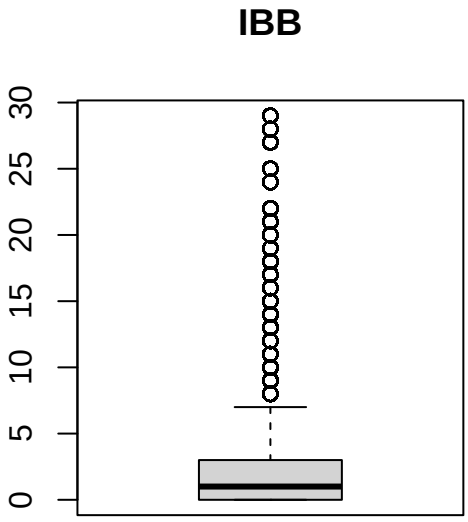

HR
Frequency
0
20000
60000
0
10
30
50
master[, i]

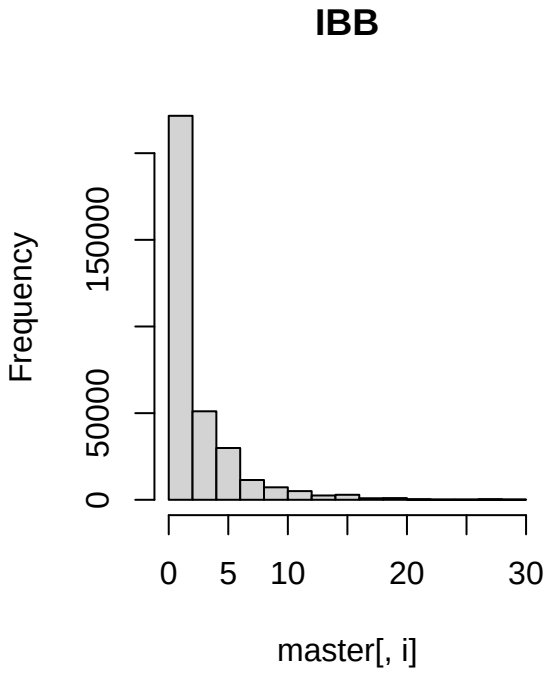

RBI
0
20
60
100
140

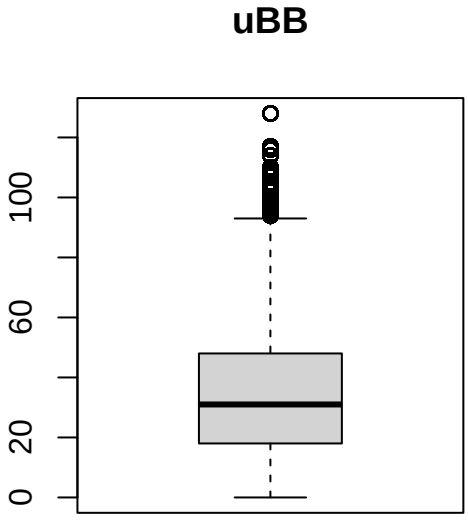

RBI
Frequency
0
5000
15000
25000
0
40
80
120
master[, i]

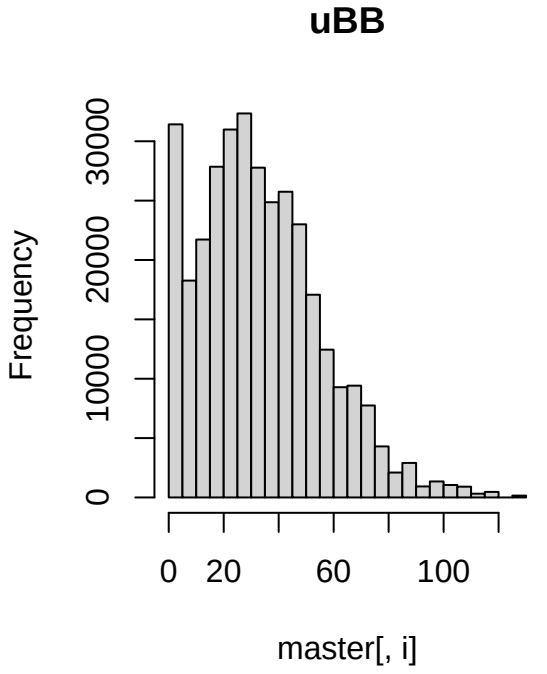

BB
0
20
60
100
140

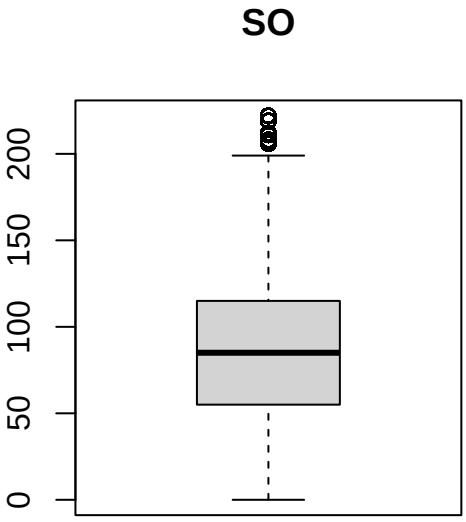

BB
Frequency
0
20000
40000
0
50
100
150
master[, i]

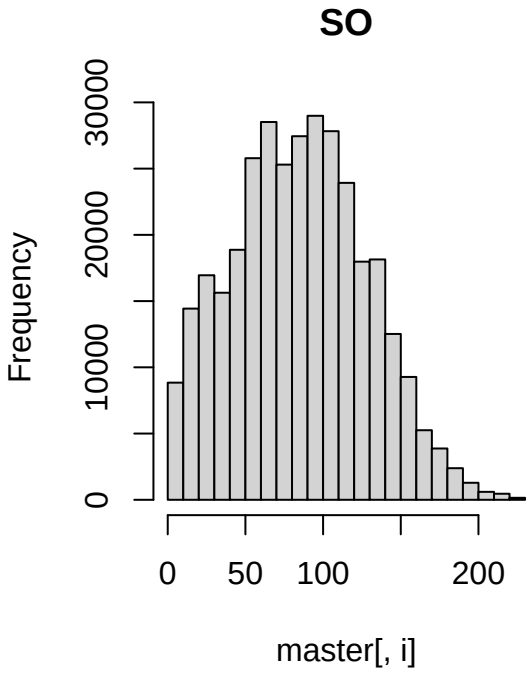

lBB
0
5
10
15
20
25
30

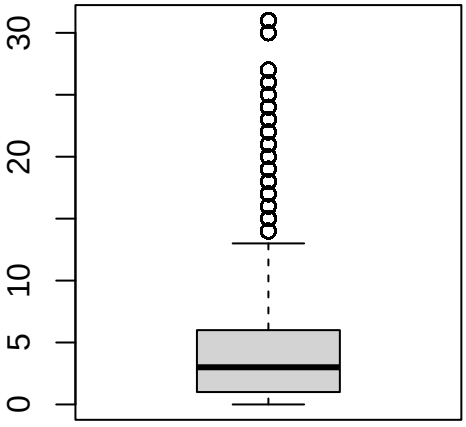

lBB
Frequency
0
50000
150000
0
5
10
20
30
master[, i]

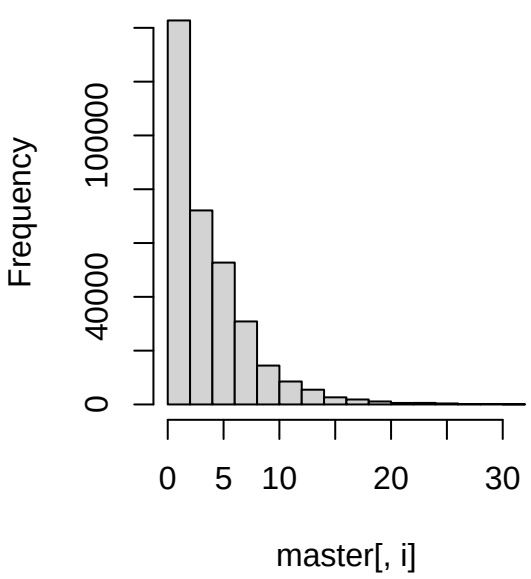

uBB
0
20
60
100

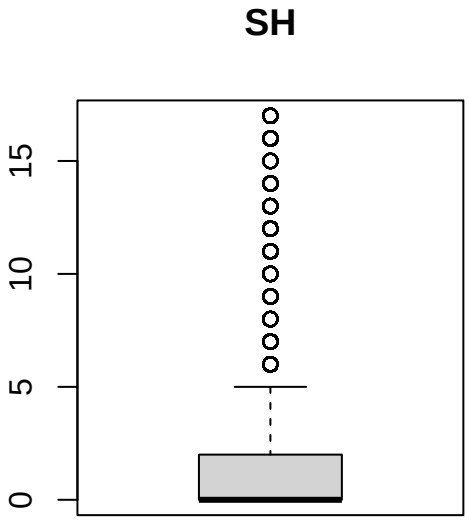

uBB
Frequency
0
10000
20000
30000
0
20
60
100
master[, i]

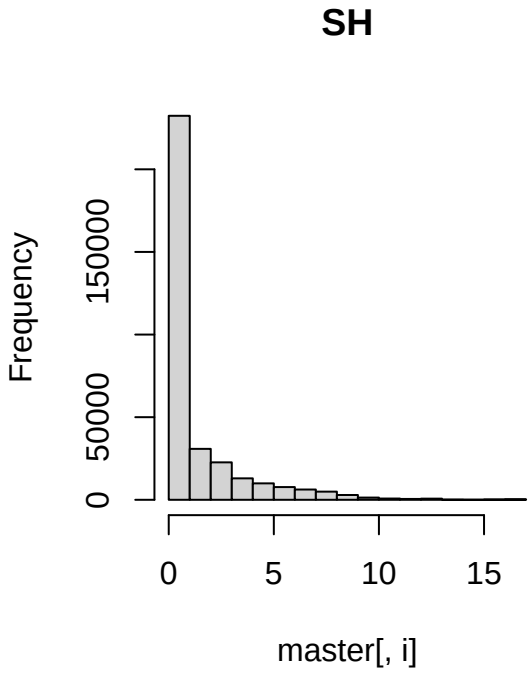

SO
0
50
100
150
200

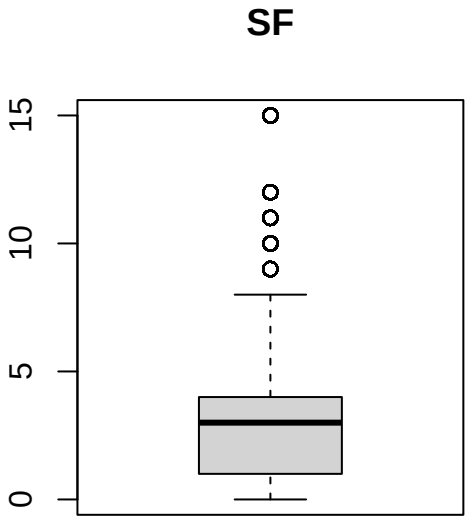

SO
Frequency
0
10000
20000
30000
0
50
100
200
master[, i]

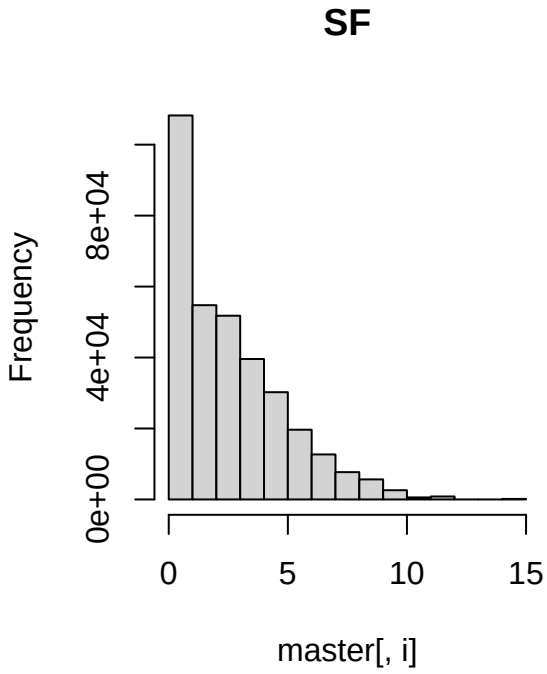

HBP
0
5
10
20
30

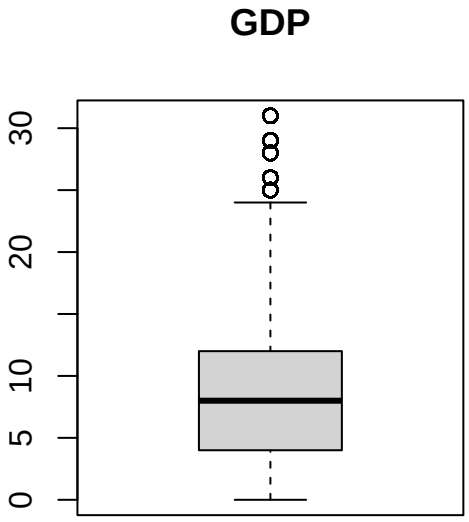

HBP
Frequency
0
40000
100000
0
5
10
20
30
master[, i]

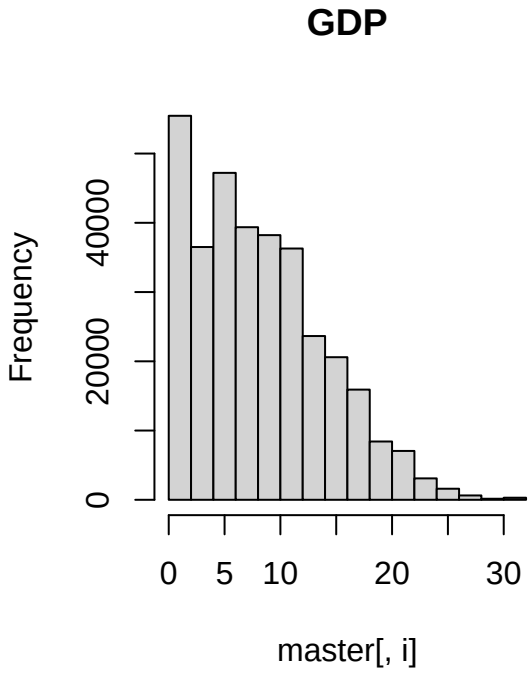

SH
0
5
10
15

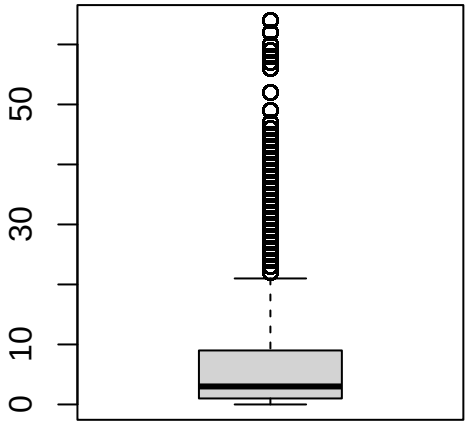

SH
Frequency
0
50000
150000
0
5
10
15
master[, i]

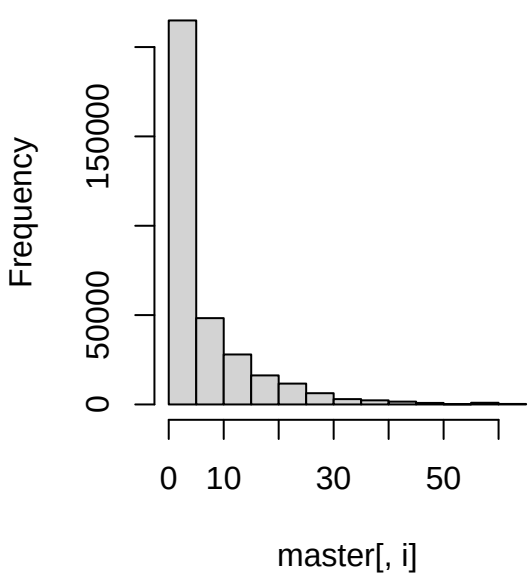

SF
0
5
10
15

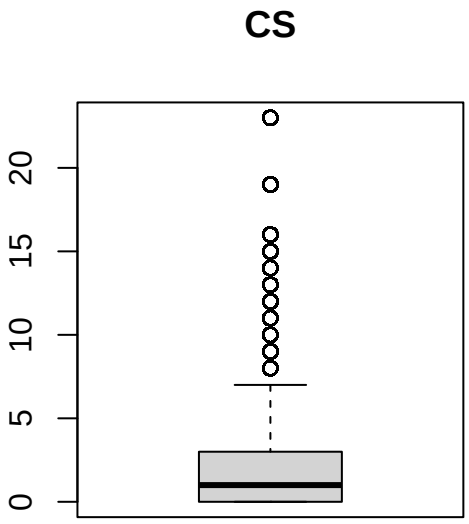

SF
Frequency
0e+00
4e+04
8e+04
0
5
10
15
master[, i]

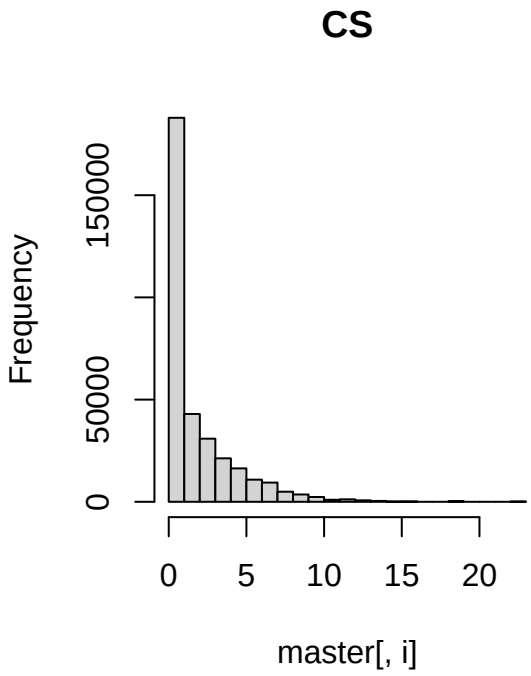

GDP
0
5
10
20
30

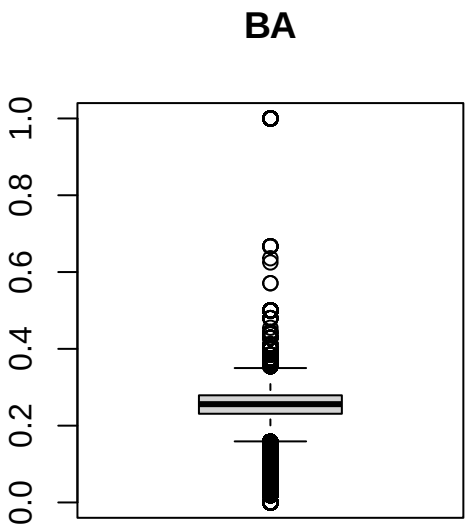

GDP
Frequency
0
20000
40000
0
5
10
20
30
master[, i]

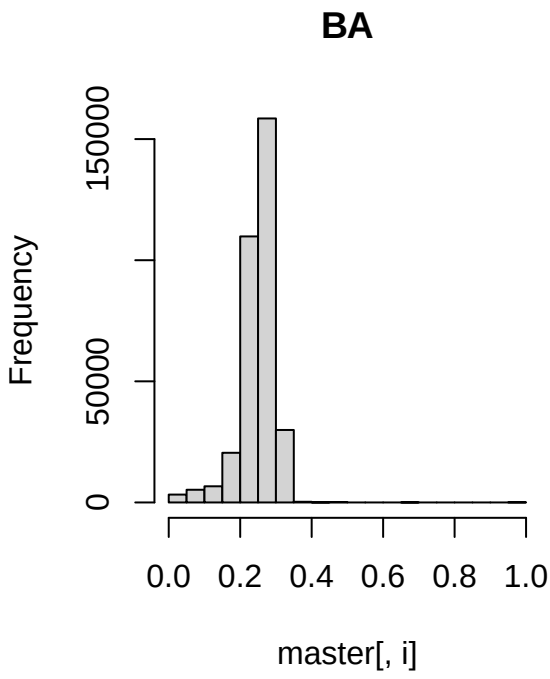

SB
0
10
30
50

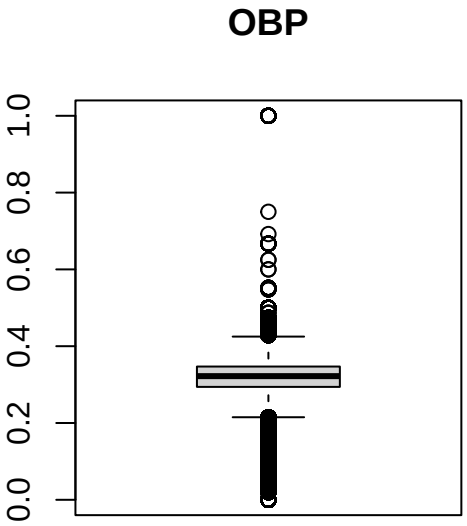

SB
Frequency
0
50000
150000
0
10
30
50
master[, i]

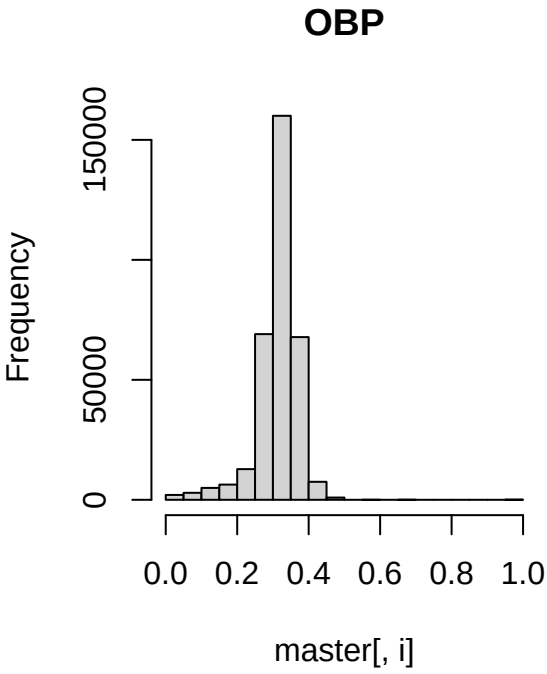

CS
0
5
10
15
20

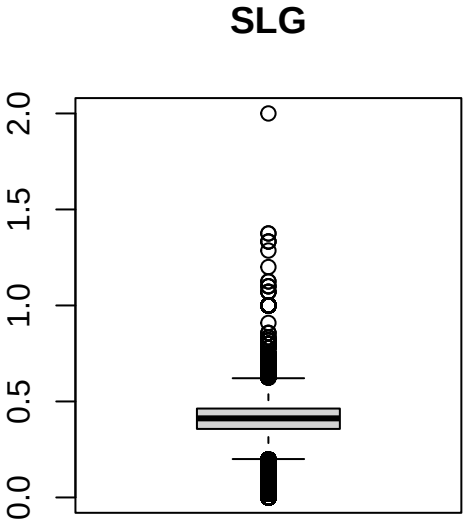

CS
Frequency
0
50000
150000
0
5
10
15
20
master[, i]

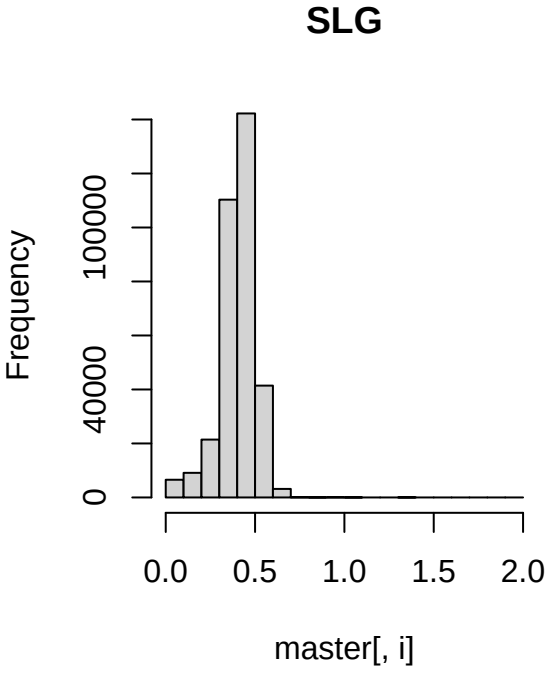

BA
0.0
0.2
0.4
0.6
0.8
1.0

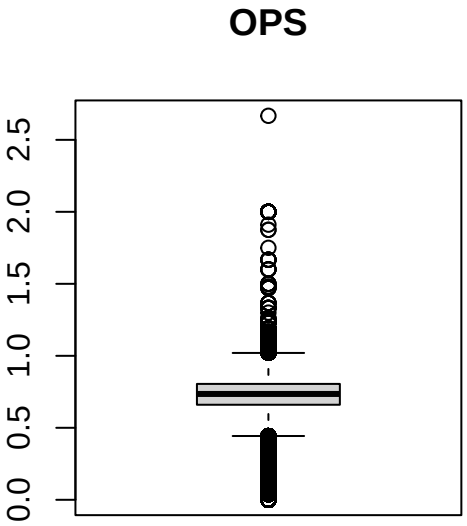

BA
Frequency
0
50000
150000
0.0
0.2
0.4
0.6
0.8
1.0
master[, i]

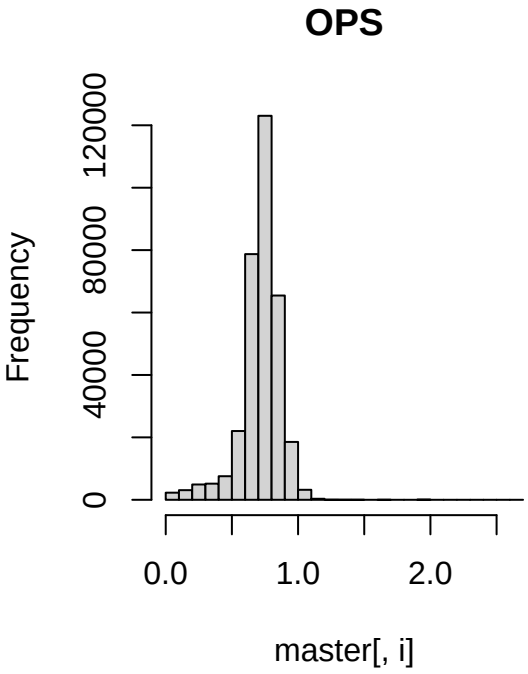

OBP
0.0
0.2
0.4
0.6
0.8
1.0

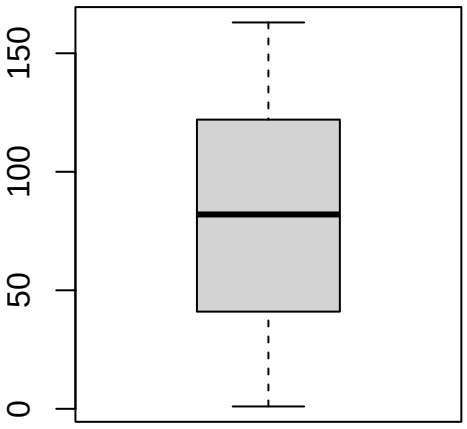

OBP
Frequency
0
50000
150000
0.0
0.2
0.4
0.6
0.8
1.0
master[, i]

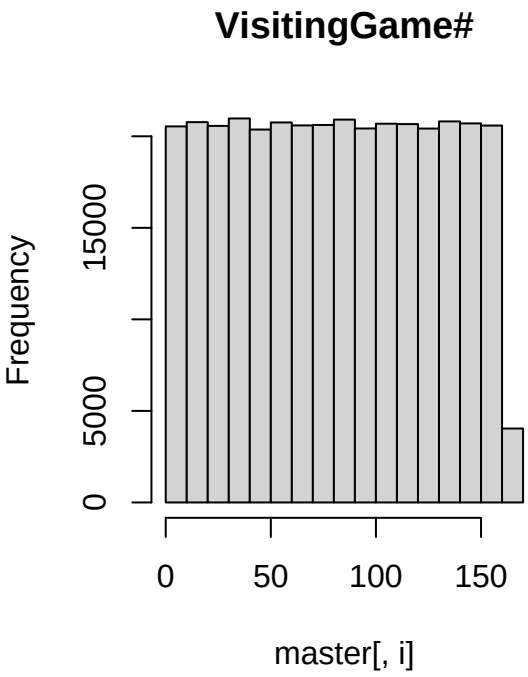

SLG
0.0
0.5
1.0
1.5
2.0

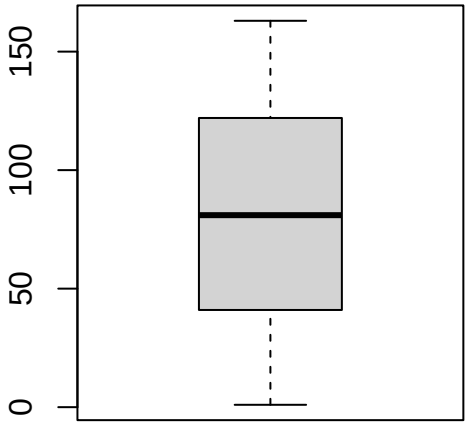

SLG
Frequency
0
40000
100000
0.0
0.5
1.0
1.5
2.0
master[, i]

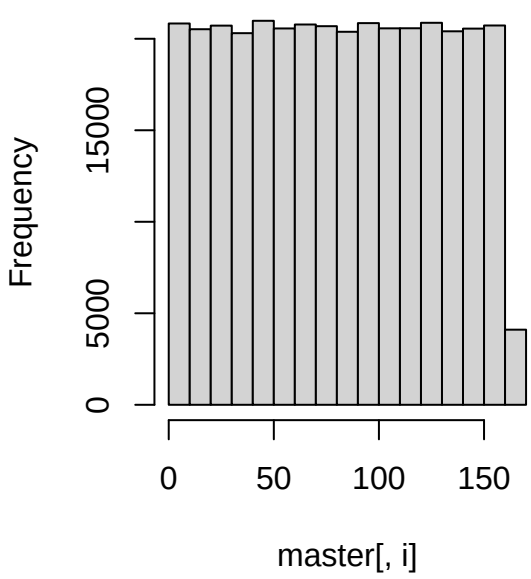

OPS
0.0
0.5
1.0
1.5
2.0
2.5

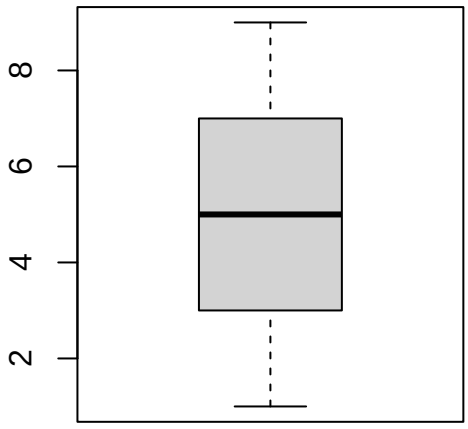

OPS
Frequency
0
40000
80000
120000
0.0
1.0
2.0
master[, i]

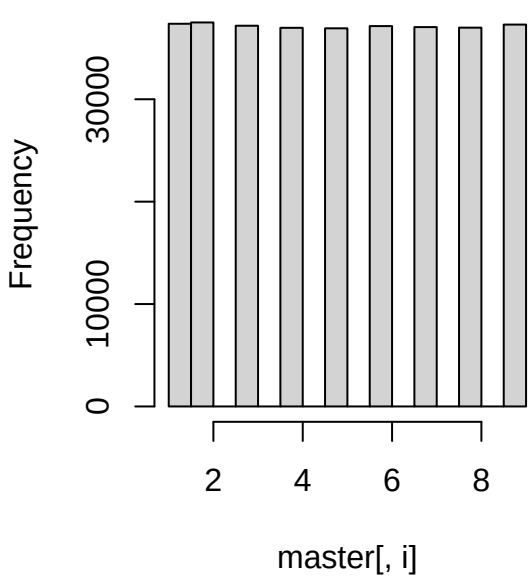

VisitingGame#
0
50
100
150


VisitingGame#
Frequency
0
5000
15000
0
50
100
150
master[, i]





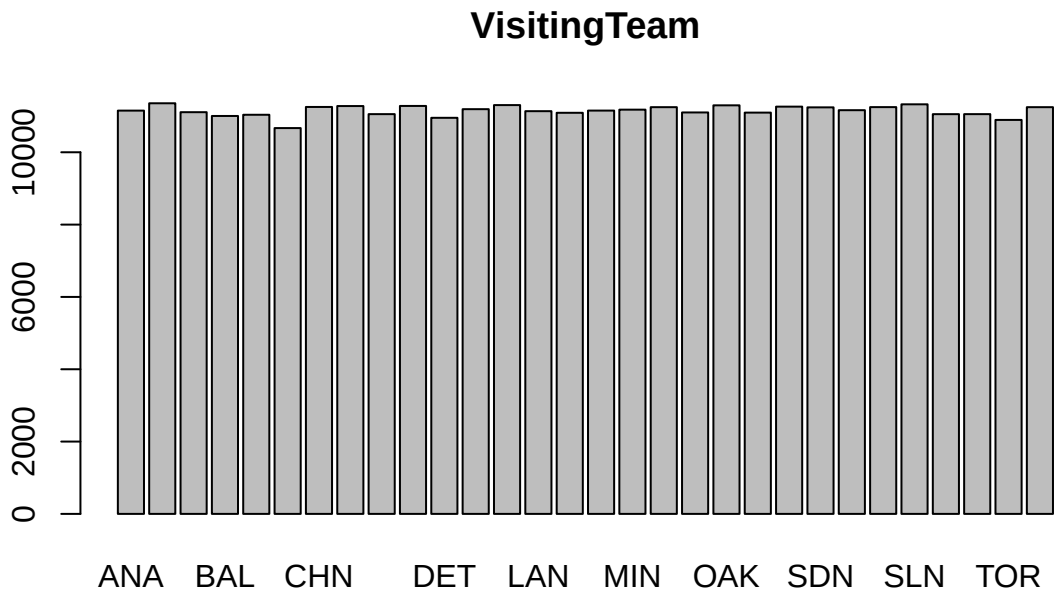


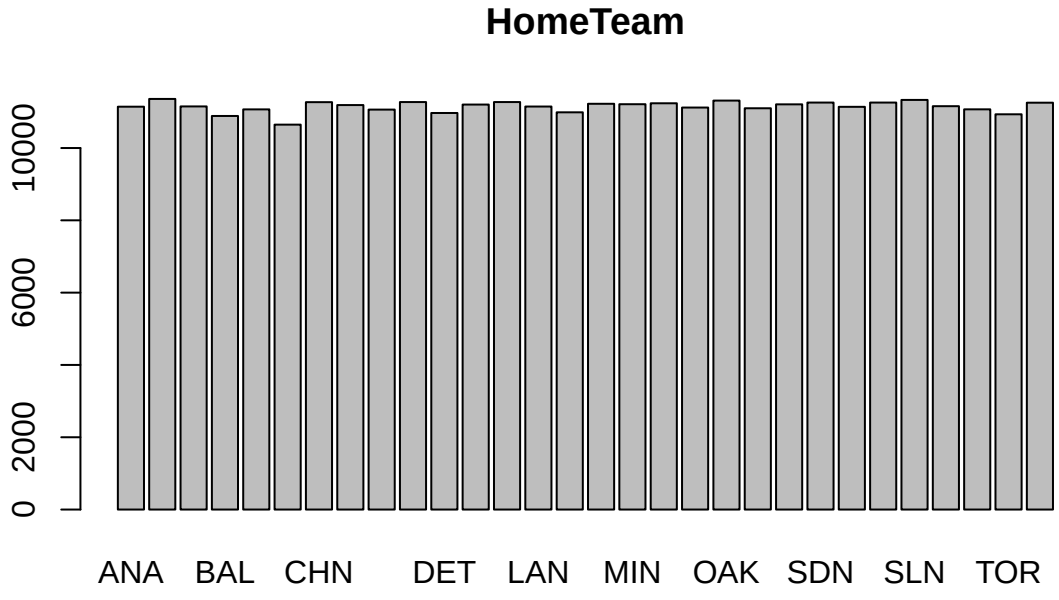


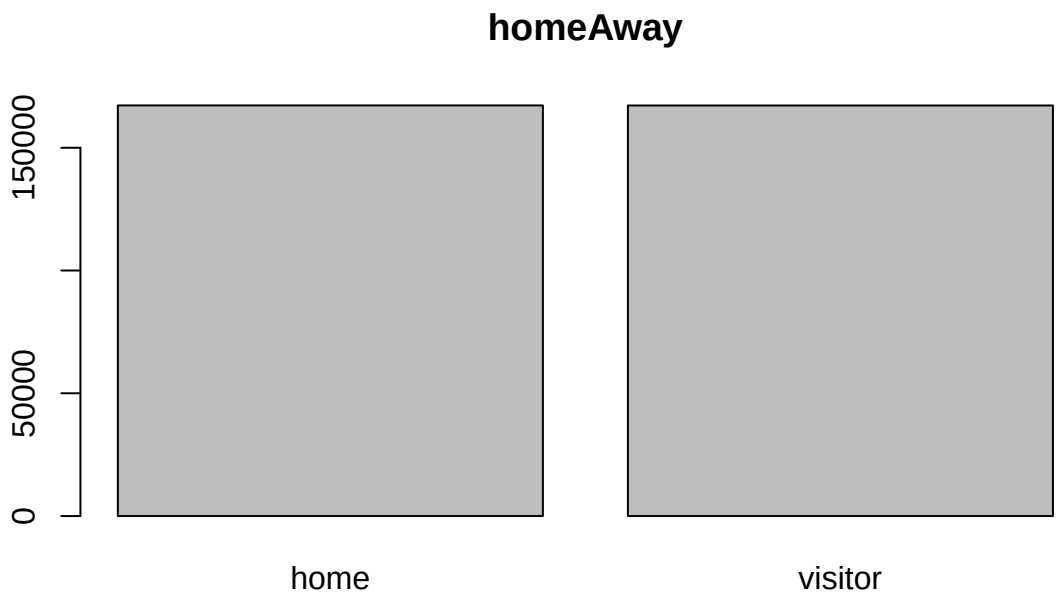
From these boxplots and histograms, it is clear that most of the data needs to be normalized in some way. However we will check the categorical data before doing so.

As explained above, the distributions for these categorical variables is uniform

```r
for(i in icat){
  barplot(table(master[,i]),main=names(master[i]))
}
```







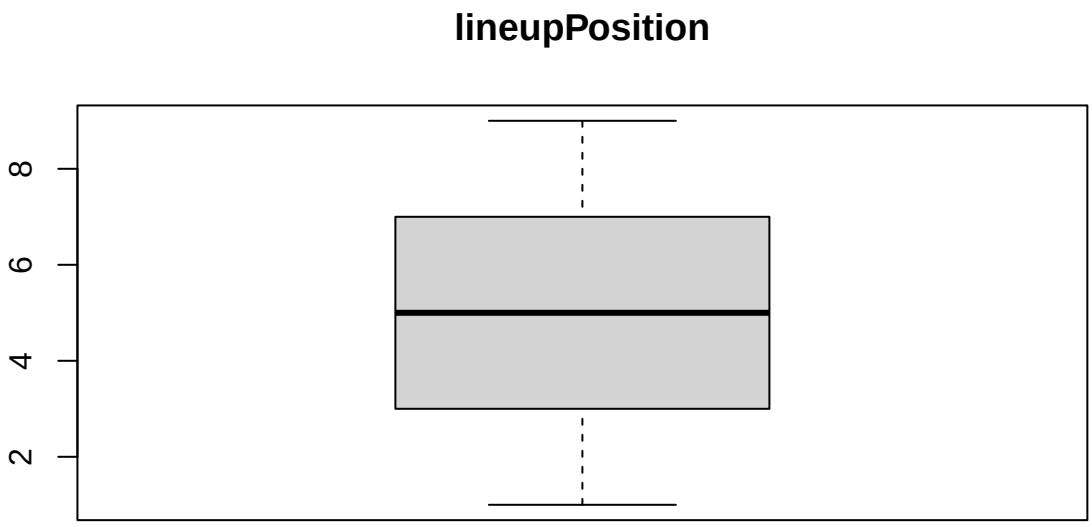
We can normalize many of the hitting statistics by converting them to rate over a number of at-bats and then dividing by the league average and multiplying by 100 to get a normalized statistic. We will only do this for certain statistics because of the overlap in many of the variables. The plus in the name means the statistic is normalized, we also included a calculation for runs created (RC) because it is another composite hitting statistic that is used in baseball even if it is not normalized.

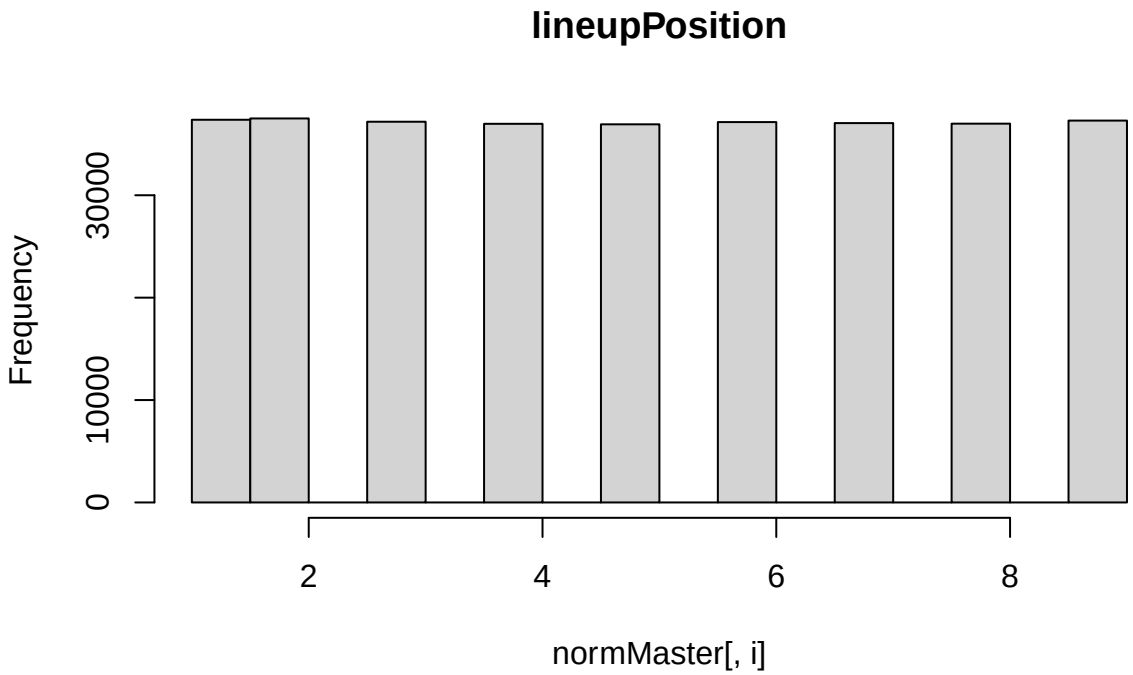
We create a vector for the indexes of these new statistics.

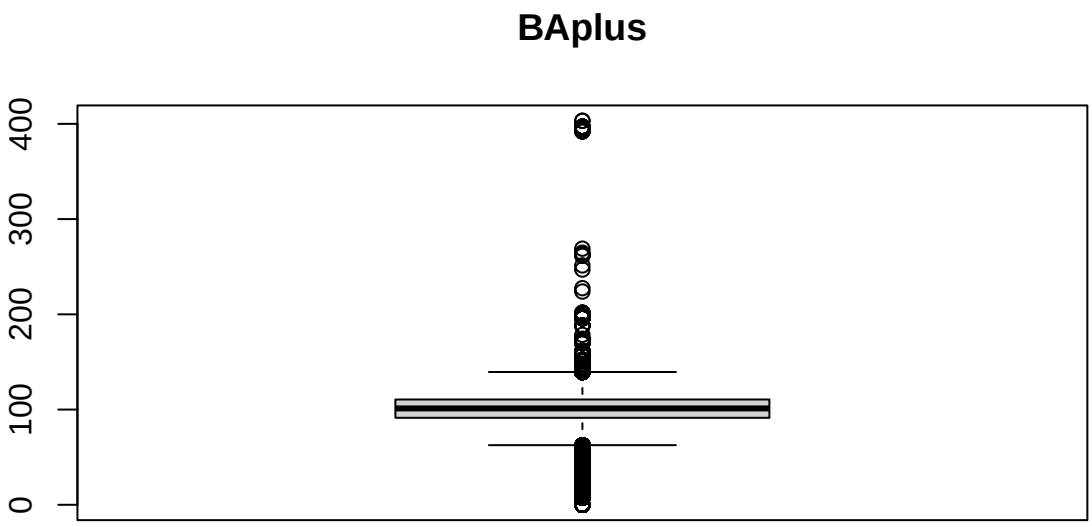
```
yearByYearAverages <- read.csv("~/College/MATH 203/Project Data/yearByYearAverages.csv")
yearByYearTotals <- read.csv("~/College/MATH 203/Project Data/yearByYearTotals.csv")
normMaster <- mutate(master,BAplus=(master$BA/(yearByYearAverages[2023-year(ymd(master$Date)),20]))*100
                     OBPplus=(master$OBP/(yearByYearAverages[2023-year(ymd(master$Date)),21]))*100,
                     SLGplus=(master$SLG/(yearByYearAverages[2023-year(ymd(master$Date)),22]))*100,
                     OPSplus_alt=(master$OPS/(yearByYearAverages[2023-year(ymd(master$Date)),23]))*100,
                     RC=(master$X1B*1+master$X2B*2+master$X3B*3+master$HR*4)*(master$H+master$BB)/(mast
icomposite = c(39:44)
```

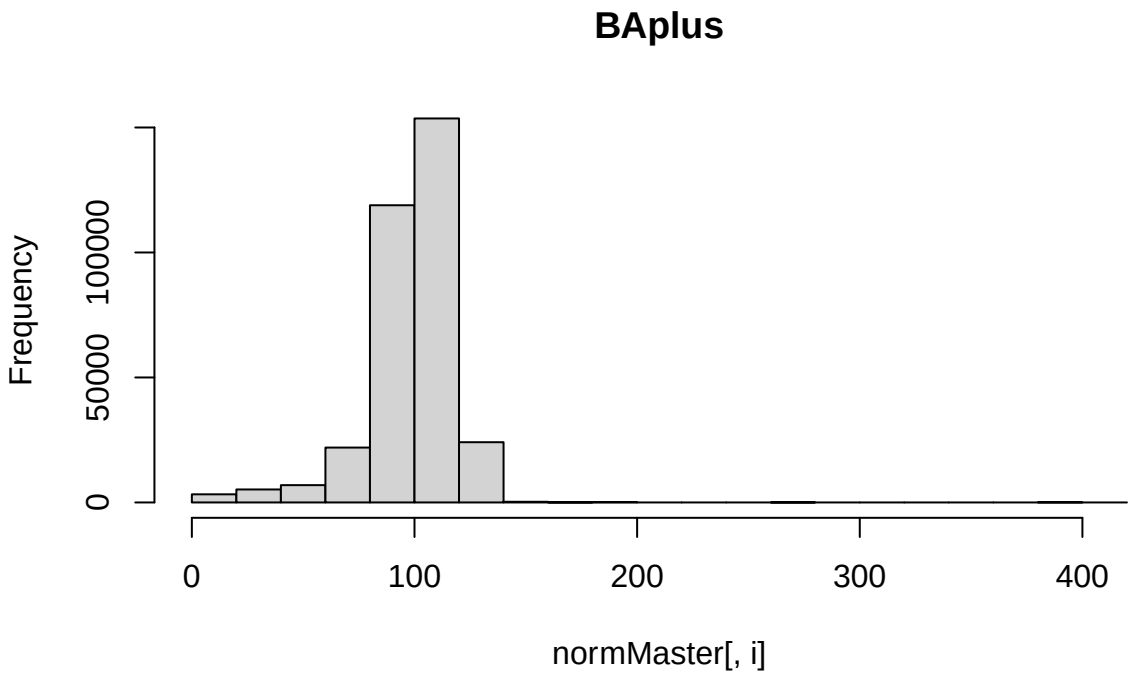
We can make boxplots and histograms for these new variables.

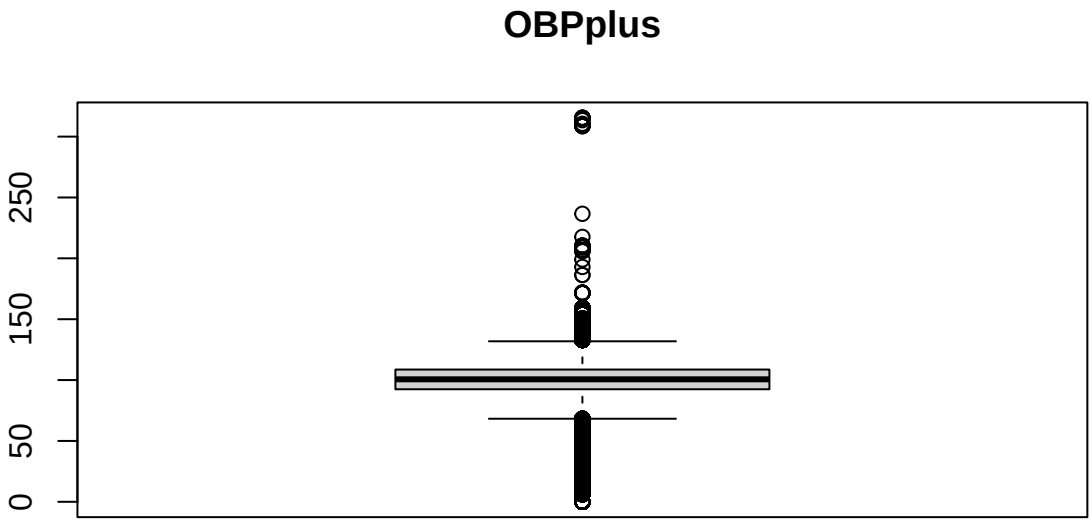
```
for(i in icomposite) {
  boxplot(normMaster[,i],main=names(normMaster[i]),type="l")
  hist(normMaster[,i],main=names(normMaster[i]))
}
```

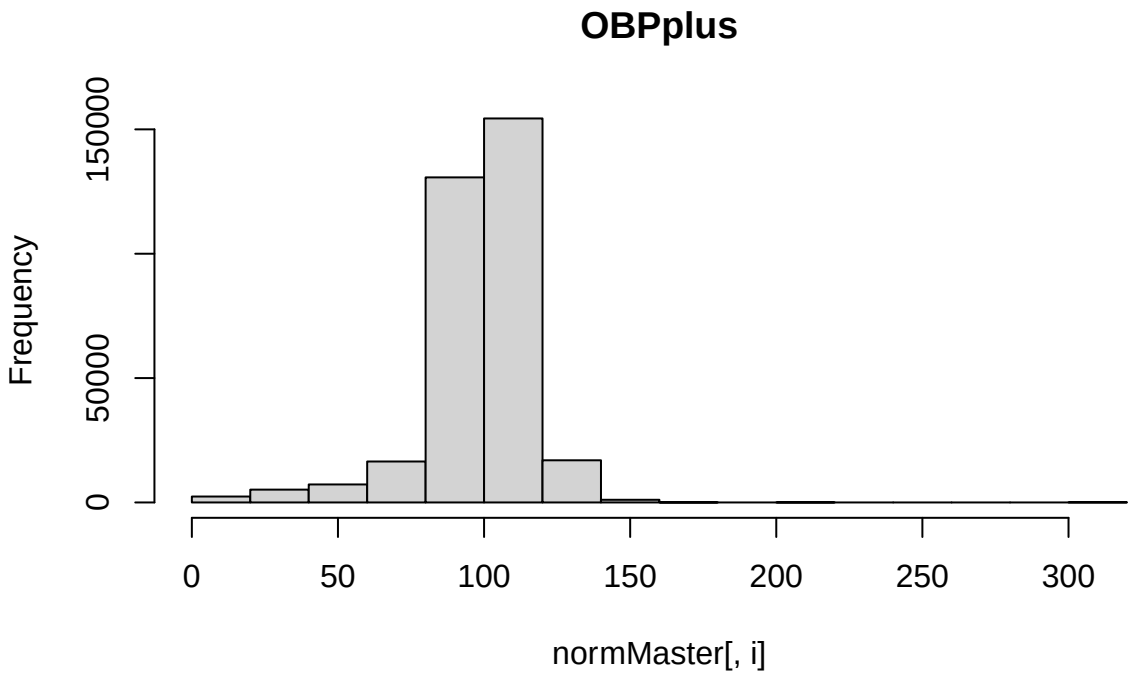

lineupPosition
2
4
6
8

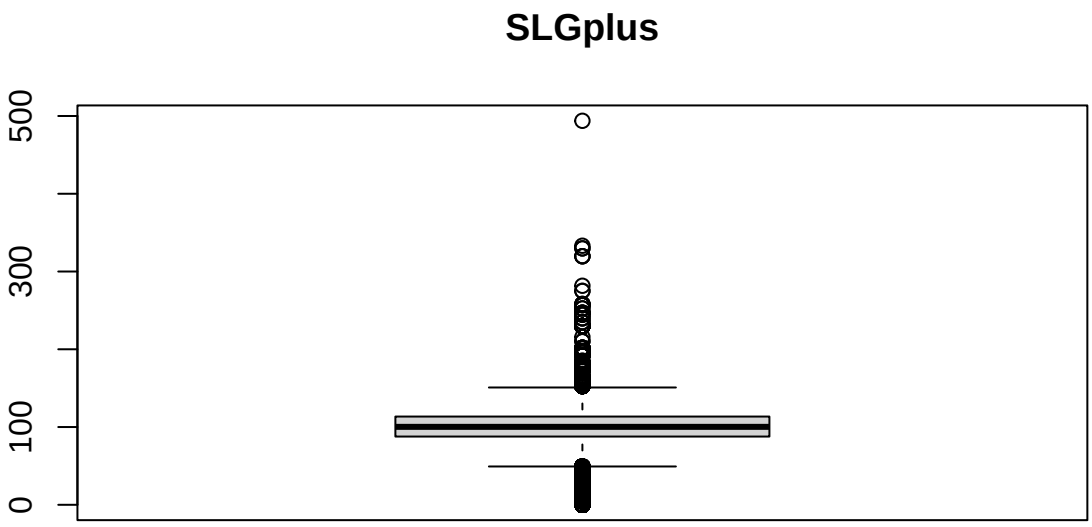

lineupPosition
Frequency
0
10000
30000
2
4
6
8
normMaster[, i]

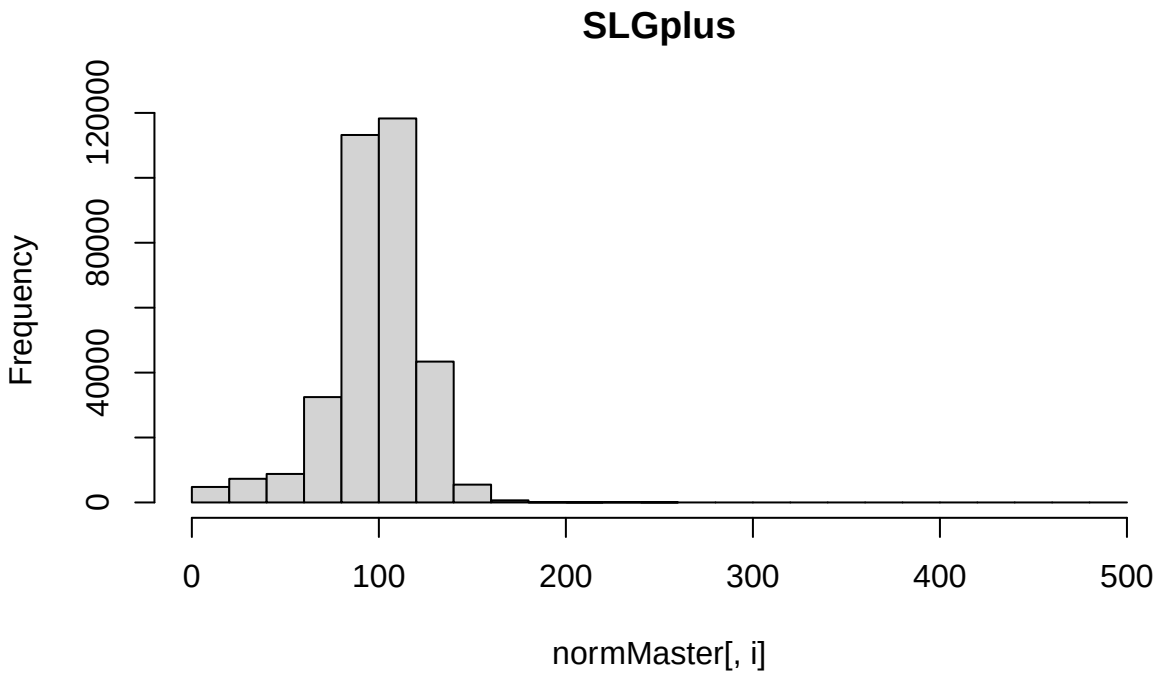

BAplus
0
100
200
300
400

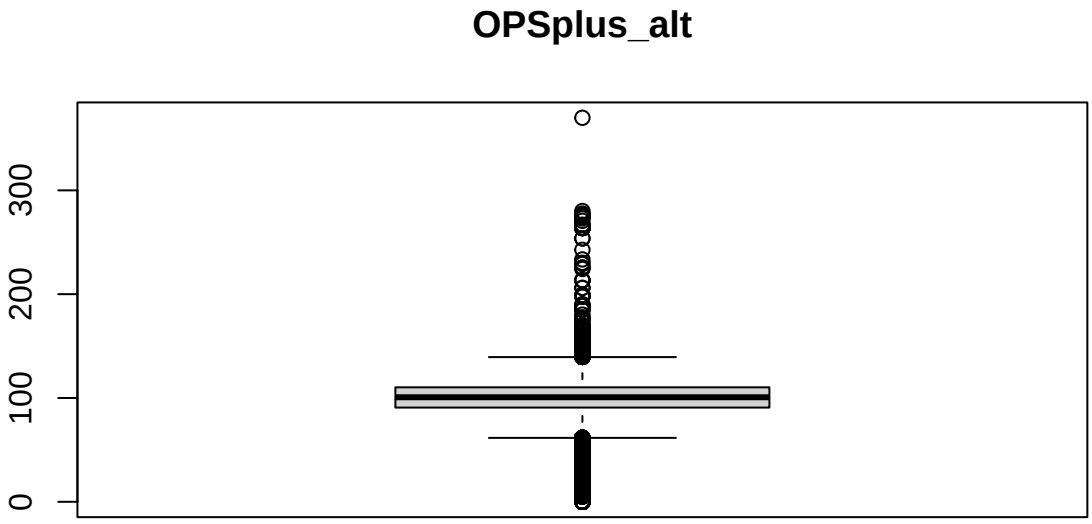

BAplus
Frequency
0
50000
100000
0
100
200
300
400
normMaster[, i]

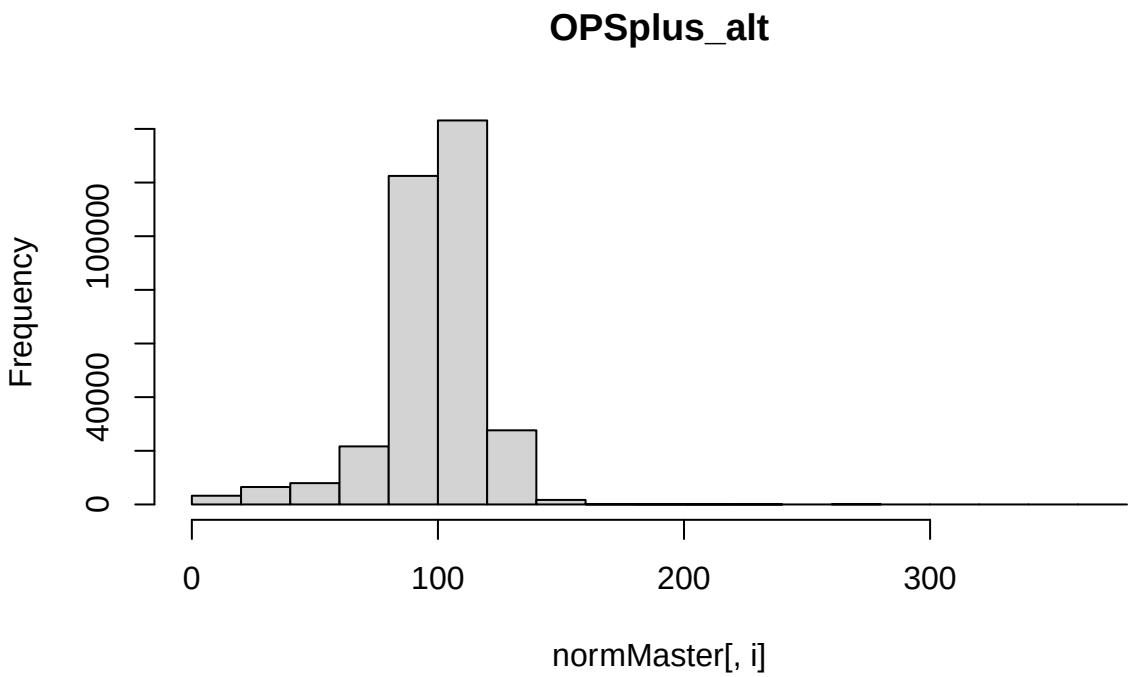

OBPplus
0
50
150
250

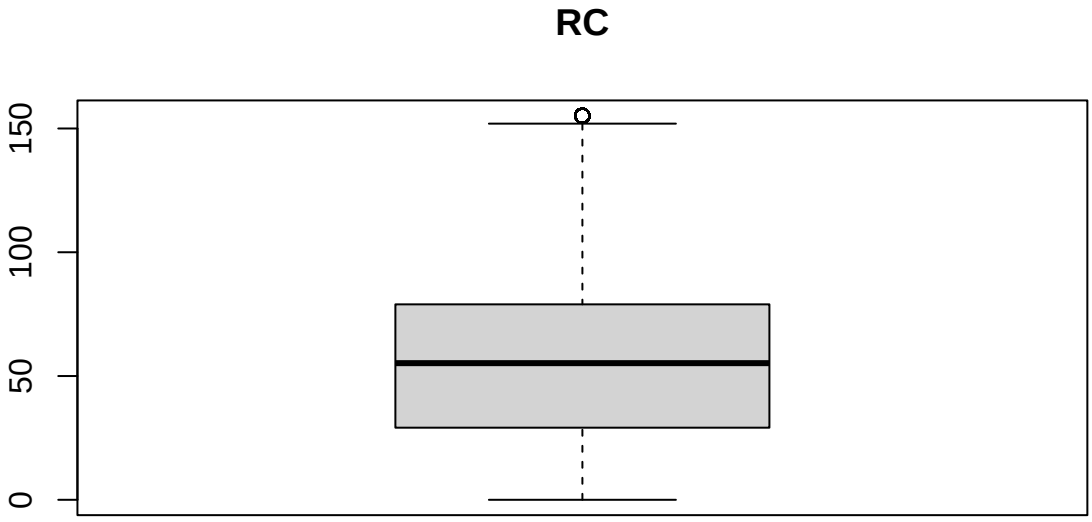

OBPplus
Frequency
0
50000
150000
0
50
100
150
200
250
300
normMaster[, i]

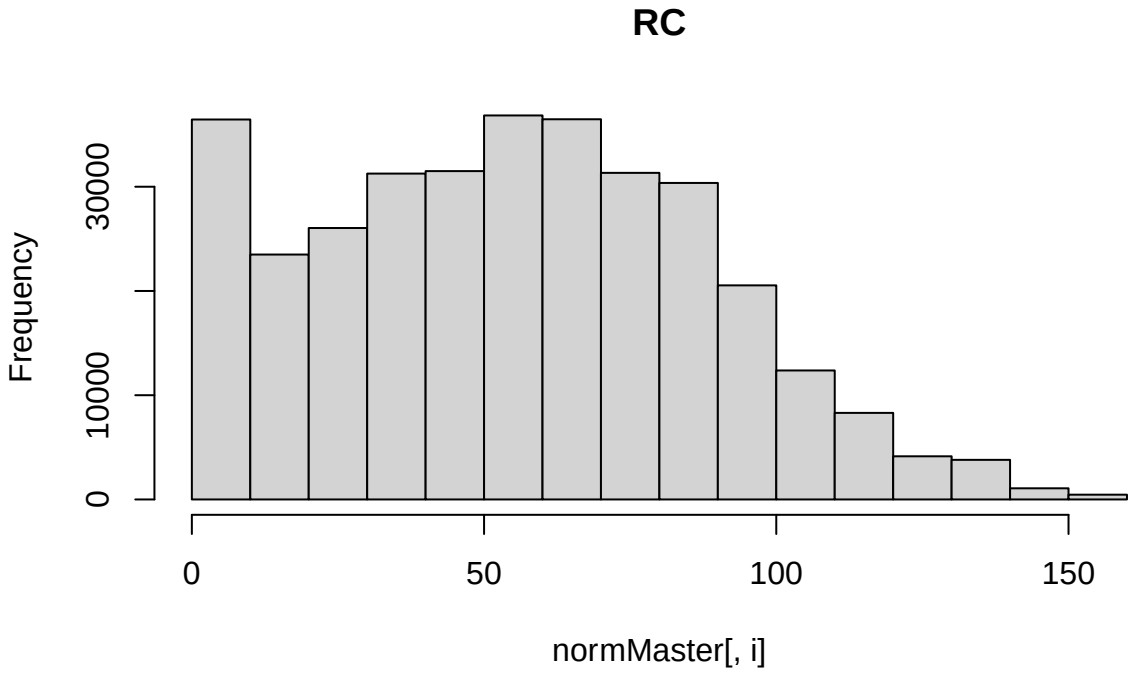

SLGplus
500
300
100
0


SLGplus
Frequency
120000
80000
40000
0
0
100
200
300
400
500
normMaster[, i]


OPSplus_alt
0
100
200
300


OPSplus_alt
Frequency
0
40000
100000
0
100
200
300
normMaster[, i]





Except for RC, which is not meant to be normalized, the normalization of the variables seems to be successful.

We do not need to fit a distribution for the response because lineup position is uniformally distributed by definition.

The next major step is to run correlations on the data.Due to the nature of the statistic we know that

we will see a lot of correlation between the variables because many of the statistic are derived from the same counting stats and better hitters will have higher statistic across the board even if the statistic being considered measures different things.

```
cor <- cor(normMaster[,c(icomposite,inum)],use="complete.obs")
cor
```

```
##                  lineupPosition      BAplus     OBPplus     SLGplus OPSplus_alt
## lineupPosition     1.0000000000 -0.46478307 -0.48909096 -0.46190869 -0.49263819
## BAplus            -0.4647830710  1.00000000  0.89224770  0.81891659  0.88294776
## OBPplus           -0.4890909636  0.89224770  1.00000000  0.82697993  0.93027900
## SLGplus           -0.4619086888  0.81891659  0.82697993  1.00000000  0.97554788
## OPSplus_alt       -0.4926381930  0.88294776  0.93027900  0.97554788  1.00000000
## RC                -0.5591423781  0.64403669  0.66539170  0.74661789  0.74727873
## G                 -0.4846919155  0.55779232  0.57462514  0.59572228  0.61326592
## PA                -0.5477810932  0.57315937  0.58753479  0.61418467  0.63042570
## AB                -0.5399094306  0.57716336  0.56190960  0.60301077  0.61314170
## R                 -0.5837772086  0.58929812  0.62206677  0.67645833  0.68453934
## H                 -0.5653377024  0.66677562  0.61925943  0.64850867  0.66528780
## X1B               -0.5401554151  0.66209291  0.57717530  0.52090222  0.56563721
## X2B               -0.5088365316  0.60174634  0.57965508  0.65012811  0.65075450
## X3B               -0.3228497742  0.32399121  0.26325620  0.26070607  0.27300915
## HR                -0.3855674785  0.38240590  0.46581494  0.72114780  0.65258288
## RBI               -0.4563090756  0.53704236  0.56541342  0.72458113  0.69373389
## BB                -0.4796312486  0.41186847  0.63936097  0.56477177  0.61825800
## IBB               -0.2172932114  0.30754458  0.39961800  0.41791285  0.42870983
## uBB               -0.4846018894  0.39730753  0.62886857  0.54539761  0.60152960
## SO                -0.3726048284  0.32576090  0.42410063  0.54368872  0.52033792
## HBP               -0.2761674273  0.27670672  0.37864115  0.34580893  0.37334488
## SH                 0.1239487103 -0.16857051 -0.25441393 -0.35527426 -0.33068369
## SF                -0.3127217666  0.37242673  0.35954287  0.41366971  0.41048005
## GDP               -0.3165156920  0.43146405  0.38988364  0.41907163  0.42589900
## SB                -0.3451563317  0.28796345  0.23774998  0.16143568  0.19836007
## CS                -0.3221046289  0.27176751  0.22315150  0.15587232  0.18926003
## BA                -0.4642954625  0.99902411  0.89125620  0.81832380  0.88226614
## OBP               -0.4883302926  0.89106825  0.99861777  0.82583959  0.92897815
## SLG               -0.4573396481  0.81040277  0.81810056  0.98896797  0.96493035
## OPS               -0.4894499776  0.87730853  0.92412228  0.96891312  0.99326209
## VisitingGame#     -0.0007760569  0.03769985  0.02156861  0.03175885  0.02912984
## HomeGame#         -0.0007556406  0.03756007  0.02140676  0.03160174  0.02896337
## lineupPosition.1   1.0000000000 -0.46478307 -0.48909096 -0.46190869 -0.49263819
##                           RC           G          PA          AB           R
## lineupPosition   -0.55914238 -0.48469192 -0.54778109 -0.53990943 -0.58377721
## BAplus            0.64403669  0.55779232  0.57315937  0.57716336  0.58929812
## OBPplus           0.66539170  0.57462514  0.58753479  0.56190960  0.62206677
## SLGplus           0.74661789  0.59572228  0.61418467  0.60301077  0.67645833
## OPSplus_alt       0.74727873  0.61326592  0.63042570  0.61314170  0.68453934
## RC                1.00000000  0.85900638  0.91754614  0.90279312  0.95267089
## G                 0.85900638  1.00000000  0.96880219  0.96762670  0.87955658
## PA                0.91754614  0.96880219  1.00000000  0.99641283  0.93441586
## AB                0.90279312  0.96762670  0.99641283  1.00000000  0.92105432
## R                 0.95267089  0.87955658  0.93441586  0.92105432  1.00000000
## H                 0.94391348  0.92582390  0.97031284  0.97457721  0.93055737
## X1B               0.84151604  0.87643994  0.91587697  0.92830723  0.84087921
```

```
## X2B                0.89309492  0.84716982  0.88926245  0.88831022  0.86699281
## X3B                0.40081649  0.42108553  0.43612154  0.44550248  0.46401510
## HR                 0.82652250  0.66669378  0.70723455  0.68614278  0.77715461
## RBI                0.92770866  0.83702772  0.87702247  0.86559365  0.87415547
## BB                 0.81227320  0.74182594  0.78425069  0.73112924  0.80598278
## IBB                0.56612817  0.43262696  0.45910669  0.43056928  0.45903771
## uBB                0.78984612  0.73448339  0.77621766  0.72324175  0.79971844
## SO                 0.71142963  0.77765306  0.77973308  0.76565626  0.74875895
## HBP                0.44747406  0.45231486  0.46687880  0.44443177  0.47692841
## SH                -0.21190613 -0.08674551 -0.10163303 -0.09306603 -0.13066498
## SF                 0.59085533  0.60497915  0.62894273  0.62052645  0.56971176
## GDP                0.63657085  0.67778199  0.69568735  0.70501385  0.58142823
## SB                 0.34869198  0.37886340  0.40905753  0.41601610  0.45636165
## CS                 0.32637672  0.38794609  0.40625743  0.41537092  0.42357213
## BA                 0.64493839  0.55720845  0.57303792  0.57717541  0.58964129
## OBP                0.66841957  0.57388682  0.58700248  0.56111862  0.62588340
## SLG                0.74951699  0.58942701  0.60781119  0.59585261  0.68313819
## OPS                0.75090434  0.60924964  0.62641684  0.60852744  0.69087223
## VisitingGame#     -0.01019950 -0.03226758 -0.02610362 -0.02531657 -0.01528989
## HomeGame#         -0.01033983 -0.03235272 -0.02618231 -0.02538877 -0.01542814
## lineupPosition.1  -0.55914238 -0.48469192 -0.54778109 -0.53990943 -0.58377721
##                            H          X1B         X2B          X3B          HR
## lineupPosition    -0.56533770 -0.540155415 -0.50883653 -0.322849774 -0.38556748
## BAplus             0.66677562  0.662092911  0.60174634  0.323991213  0.38240590
## OBPplus            0.61925943  0.577175295  0.57965508  0.263256197  0.46581494
## SLGplus            0.64850867  0.520902216  0.65012811  0.260706074  0.72114780
## OPSplus_alt        0.66528780  0.565637211  0.65075450  0.273009145  0.65258288
## RC                 0.94391348  0.841516042  0.89309492  0.400816493  0.82652250
## G                  0.92582390  0.876439941  0.84716982  0.421085527  0.66669378
## PA                 0.97031284  0.915876970  0.88926245  0.436121545  0.70723455
## AB                 0.97457721  0.928307230  0.88831022  0.445502482  0.68614278
## R                  0.93055737  0.840879207  0.86699281  0.464015099  0.77715461
## H                  1.00000000  0.964074844  0.90815800  0.464886548  0.66768336
## X1B                0.96407484  1.000000000  0.81008756  0.467047633  0.47676393
## X2B                0.90815800  0.810087564  1.00000000  0.378989247  0.64252401
## X3B                0.46488655  0.467047633  0.37898925  1.000000000  0.12482936
## HR                 0.66768336  0.476763930  0.64252401  0.124829364  1.00000000
## RBI                0.86467382  0.734881958  0.82747110  0.257744601  0.88998436
## BB                 0.70601559  0.599821602  0.68595590  0.251326631  0.70735267
## IBB                0.46095862  0.384620307  0.44504133  0.085638218  0.50510724
## uBB                0.69136566  0.588465598  0.67216127  0.258324991  0.68593795
## SO                 0.68302208  0.562320605  0.64585961  0.298471641  0.75111800
## HBP                0.43184598  0.377358784  0.40975783  0.166322122  0.40589296
## SH                -0.09650362  0.008607071 -0.16574014  0.173800796 -0.36395998
## SF                 0.61273171  0.567673849  0.58998594  0.206059922  0.46619364
## GDP                0.70166171  0.688747975  0.62744629  0.112853323  0.48459327
## SB                 0.43724134  0.484852646  0.33009588  0.546492553  0.08036388
## CS                 0.42381084  0.465057494  0.33709735  0.535672548  0.07425615
## BA                 0.66748034  0.662935050  0.60160544  0.324087506  0.38323427
## OBP                0.61874660  0.573962136  0.57954293  0.261513813  0.47442470
## SLG                0.64130686  0.506125506  0.64458568  0.254463711  0.74184399
## OPS                0.66071725  0.554754491  0.64761908  0.268351661  0.67044087
## VisitingGame#     -0.01471796 -0.013897171 -0.01591623  0.009982551 -0.01167597
## HomeGame#         -0.01481780 -0.013967272 -0.01603575  0.009945154 -0.01179996
```

```
## lineupPosition.1 -0.56533770 -0.540155415 -0.50883653 -0.322849774 -0.38556748
##                         RBI          BB         IBB         uBB          SO
## lineupPosition   -0.45630908 -0.47963125 -0.21729321 -0.48460189 -0.37260483
## BAplus            0.53704236  0.41186847  0.30754458  0.39730753  0.32576090
## OBPplus           0.56541342  0.63936097  0.39961800  0.62886857  0.42410063
## SLGplus           0.72458113  0.56477177  0.41791285  0.54539761  0.54368872
## OPSplus_alt       0.69373389  0.61825800  0.42870983  0.60152960  0.52033792
## RC                0.92770866  0.81227320  0.56612817  0.78984612  0.71142963
## G                 0.83702772  0.74182594  0.43262696  0.73448339  0.77765306
## PA                0.87702247  0.78425069  0.45910669  0.77621766  0.77973308
## AB                0.86559365  0.73112924  0.43056928  0.72324175  0.76565626
## R                 0.87415547  0.80598278  0.45903771  0.79971844  0.74875895
## H                 0.86467382  0.70601559  0.46095862  0.69136566  0.68302208
## X1B               0.73488196  0.59982160  0.38462031  0.58846560  0.56232060
## X2B               0.82747110  0.68595590  0.44504133  0.67216127  0.64585961
## X3B               0.25774460  0.25132663  0.08563822  0.25832499  0.29847164
## HR                0.88998436  0.70735267  0.50510724  0.68593795  0.75111800
## RBI               1.00000000  0.75621731  0.54501113  0.73254320  0.74895102
## BB                0.75621731  1.00000000  0.57218233  0.99181602  0.70537604
## IBB               0.54501113  0.57218233  1.00000000  0.46278969  0.33774160
## uBB               0.73254320  0.99181602  0.46278969  1.00000000  0.70985652
## SO                0.74895102  0.70537604  0.33774160  0.70985652  1.00000000
## HBP               0.42820602  0.40158113  0.20821948  0.40165072  0.42702047
## SH               -0.28968863 -0.22642989 -0.20438708 -0.21292776 -0.20156708
## SF                0.65082191  0.49011718  0.34524003  0.47601760  0.41345147
## GDP               0.66780368  0.47423077  0.36520187  0.45573849  0.43546993
## SB                0.18992944  0.23965632  0.06684101  0.24863695  0.25248183
## CS                0.18015370  0.21897874  0.05229979  0.22855046  0.26684382
## BA                0.53778398  0.41101250  0.30848023  0.39623666  0.32420282
## OBP               0.56920998  0.64133275  0.39761513  0.63131166  0.42711832
## SLG               0.72996300  0.56661741  0.40584976  0.54927050  0.55004613
## OPS               0.69949934  0.62049917  0.42042508  0.60524182  0.52639210
## VisitingGame#    -0.01489225 -0.02487260 -0.01878527 -0.02396015 -0.02456981
## HomeGame#        -0.01503978 -0.02498215 -0.01879457 -0.02407711 -0.02461793
## lineupPosition.1 -0.45630908 -0.47963125 -0.21729321 -0.48460189 -0.37260483
##                         HBP            SH          SF         GDP           SB
## lineupPosition   -0.27616743  0.1239487103 -0.31272177 -0.31651569 -0.345156332
## BAplus            0.27670672 -0.1685705130  0.37242673  0.43146405  0.287963450
## OBPplus           0.37864115 -0.2544139331  0.35954287  0.38988364  0.237749976
## SLGplus           0.34580893 -0.3552742563  0.41366971  0.41907163  0.161435677
## OPSplus_alt       0.37334488 -0.3306836901  0.41048005  0.42589900  0.198360067
## RC                0.44747406 -0.2119061288  0.59085533  0.63657085  0.348691981
## G                 0.45231486 -0.0867455102  0.60497915  0.67778199  0.378863400
## PA                0.46687880 -0.1016330301  0.62894273  0.69568735  0.409057533
## AB                0.44443177 -0.0930660306  0.62052645  0.70501385  0.416016098
## R                 0.47692841 -0.1306649831  0.56971176  0.58142823  0.456361648
## H                 0.43184598 -0.0965036162  0.61273171  0.70166171  0.437241339
## X1B               0.37735878  0.0086070714  0.56767385  0.68874798  0.484852646
## X2B               0.40975783 -0.1657401427  0.58998594  0.62744629  0.330095880
## X3B               0.16632212  0.1738007957  0.20605992  0.11285332  0.546492553
## HR                0.40589296 -0.3639599810  0.46619364  0.48459327  0.080363883
## RBI               0.42820602 -0.2896886297  0.65082191  0.66780368  0.189929441
## BB                0.40158113 -0.2264298906  0.49011718  0.47423077  0.239656324
## IBB               0.20821948 -0.2043870802  0.34524003  0.36520187  0.066841010
```

```
## uBB               0.40165072 -0.2129277636  0.47601760  0.45573849  0.248636954
## SO                0.42702047 -0.2015670774  0.41345147  0.43546993  0.252481831
## HBP               1.00000000 -0.0789891885  0.25706679  0.24727193  0.168437164
## SH               -0.07898919  1.0000000000 -0.11931111 -0.17293097  0.268474565
## SF                0.25706679 -0.1193111088  1.00000000  0.49596338  0.162395241
## GDP               0.24727193 -0.1729309653  0.49596338  1.00000000  0.098181446
## SB                0.16843716  0.2684745650  0.16239524  0.09818145  1.000000000
## CS                0.19375894  0.2617294040  0.15315798  0.10169416  0.774348286
## BA                0.27418065 -0.1661030688  0.37173162  0.43346996  0.288699917
## OBP               0.38023987 -0.2596472890  0.35654417  0.38954627  0.235687264
## SLG               0.35114862 -0.3701120792  0.40172379  0.41200831  0.153219312
## OPS               0.37776630 -0.3433658432  0.40186212  0.42131556  0.192024793
## VisitingGame#    -0.01337678 -0.0006827126 -0.01840872 -0.02548258 -0.002339050
## HomeGame#        -0.01342713 -0.0006522385 -0.01845564 -0.02553015 -0.002409817
## lineupPosition.1 -0.27616743  0.1239487103 -0.31272177 -0.31651569 -0.345156332
##                           CS          BA         OBP        SLG         OPS
## lineupPosition   -0.322104629 -0.46429546 -0.48833029 -0.4573396 -0.48944998
## BAplus            0.271767512  0.99902411  0.89106825  0.8104028  0.87730853
## OBPplus           0.223151499  0.89125620  0.99861777  0.8181006  0.92412228
## SLGplus           0.155872318  0.81832380  0.82583959  0.9889680  0.96891312
## OPSplus_alt       0.189260026  0.88226614  0.92897815  0.9649304  0.99326209
## RC                0.326376720  0.64493839  0.66841957  0.7495170  0.75090434
## G                 0.387946090  0.55720845  0.57388682  0.5894270  0.60924964
## PA                0.406257430  0.57303792  0.58700248  0.6078112  0.62641684
## AB                0.415370915  0.57717541  0.56111862  0.5958526  0.60852744
## R                 0.423572130  0.58964129  0.62588340  0.6831382  0.69087223
## H                 0.423810844  0.66748034  0.61874660  0.6413069  0.66071725
## X1B               0.465057494  0.66293505  0.57396214  0.5061255  0.55475449
## X2B               0.337097348  0.60160544  0.57954293  0.6445857  0.64761908
## X3B               0.535672548  0.32408751  0.26151381  0.2544637  0.26835166
## HR                0.074256152  0.38323427  0.47442470  0.7418440  0.67044087
## RBI               0.180153697  0.53778398  0.56920998  0.7299630  0.69949934
## BB                0.218978740  0.41101250  0.64133275  0.5666174  0.62049917
## IBB               0.052299787  0.30848023  0.39761513  0.4058498  0.42042508
## uBB               0.228550462  0.39623666  0.63131166  0.5492705  0.60524182
## SO                0.266843824  0.32420282  0.42711832  0.5500461  0.52639210
## HBP               0.193758945  0.27418065  0.38023987  0.3511486  0.37776630
## SH                0.261729404 -0.16610307 -0.25964729 -0.3701121 -0.34336584
## SF                0.153157984  0.37173162  0.35654417  0.4017238  0.40186212
## GDP               0.101694159  0.43346996  0.38954627  0.4120083  0.42131556
## SB                0.774348286  0.28869992  0.23568726  0.1532193  0.19202479
## CS                1.000000000  0.27261200  0.21994156  0.1451816  0.18070247
## BA                0.272612001  1.00000000  0.89106176  0.8111513  0.87779849
## OBP               0.219941564  0.89106176  1.00000000  0.8240989  0.92858769
## SLG               0.145181649  0.81115134  0.82409886  1.0000000  0.97545589
## OPS               0.180702467  0.87779849  0.92858769  0.9754559  1.00000000
## VisitingGame#    -0.006611609  0.03760868  0.02159894  0.0315283  0.02904960
## HomeGame#        -0.006640165  0.03747004  0.02143229  0.0313512  0.02886851
## lineupPosition.1 -0.322104629 -0.46429546 -0.48833029 -0.4573396 -0.48944998
##                  VisitingGame#     HomeGame# lineupPosition.1
## lineupPosition   -0.0007760569 -0.0007556406     1.0000000000
## BAplus            0.0376998538  0.0375600747    -0.4647830710
## OBPplus           0.0215686141  0.0214067591    -0.4890909636
## SLGplus           0.0317588497  0.0316017406    -0.4619086888
```

```
## OPSplus_alt       0.0291298365  0.0289633730    -0.4926381930
## RC               -0.0101995021 -0.0103398303    -0.5591423781
## G                -0.0322675761 -0.0323527213    -0.4846919155
## PA               -0.0261036197 -0.0261823137    -0.5477810932
## AB               -0.0253165680 -0.0253887660    -0.5399094306
## R                -0.0152898893 -0.0154281402    -0.5837772086
## H                -0.0147179637 -0.0148178025    -0.5653377024
## X1B              -0.0138971712 -0.0139672723    -0.5401554151
## X2B              -0.0159162314 -0.0160357543    -0.5088365316
## X3B               0.0099825511  0.0099451542    -0.3228497742
## HR               -0.0116759745 -0.0117999557    -0.3855674785
## RBI              -0.0148922542 -0.0150397788    -0.4563090756
## BB               -0.0248726037 -0.0249821463    -0.4796312486
## IBB              -0.0187852748 -0.0187945665    -0.2172932114
## uBB              -0.0239601516 -0.0240771087    -0.4846018894
## SO               -0.0245698117 -0.0246179305    -0.3726048284
## HBP              -0.0133767759 -0.0134271303    -0.2761674273
## SH               -0.0006827126 -0.0006522385     0.1239487103
## SF               -0.0184087186 -0.0184556437    -0.3127217666
## GDP              -0.0254825847 -0.0255301535    -0.3165156920
## SB               -0.0023390498 -0.0024098169    -0.3451563317
## CS               -0.0066116089 -0.0066401653    -0.3221046289
## BA                0.0376086788  0.0374700412    -0.4642954625
## OBP               0.0215989393  0.0214322876    -0.4883302926
## SLG               0.0315282996  0.0313512038    -0.4573396481
## OPS               0.0290496012  0.0288685068    -0.4894499776
## VisitingGame#     1.0000000000  0.9994131669    -0.0007760569
## HomeGame#         0.9994131669  1.0000000000    -0.0007556406
## lineupPosition.1 -0.0007760569 -0.0007556406     1.0000000000
```

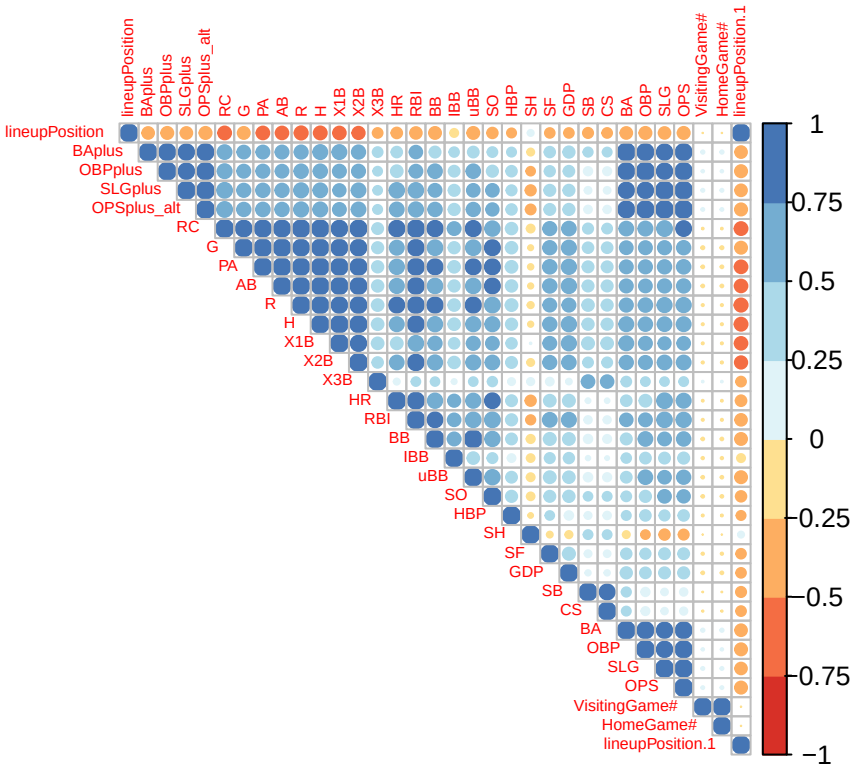
This does not mean much in its current form, so we need to display it graphically.

```
library(corrplot);library(RColorBrewer);library(PerformanceAnalytics)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3

## corrplot 0.92 loaded

## Warning: package 'PerformanceAnalytics' was built under R version 4.1.3

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## ###################################### WARNING ######################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                              #
## # Calls to lag(my_xts) that you enter or source() into this session won't     #
## # work correctly.                                                              #
## #                                                                              #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                   #
## #                                                                              #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop   #
## # dplyr from breaking base R's lag() function.                                 #
## ###################################### WARNING ######################################

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```
corrplot(cor,type="upper",col=brewer.pal(n=8,name="RdYlBu"),tl.cex=0.5)
```



As we predicted, there are a lot of strong correlations between the variables. When we consider a multiple regression later in our analysis, we will need to consider interaction effects.

The final thing we can look as is the dsitributions for all of the statistics but stratified over lineup position.

```
for(i in icomposite){
  print(names(normMaster[i]))
  print(summary(normMaster[,i]))
  print(tapply(normMaster[,i],normMaster$lineupPosition,summary))
}
```

```
## [1] "lineupPosition"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   5.000   4.994   7.000   9.000
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2       2       2       2       2       2
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3       3       3       3       3       3
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       4       4       4       4       4       4
##
```

```
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       5       5       5       5       5       5
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       6       6       6       6       6       6
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       7       7       7       7       7       7
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       8       8       8       8       8       8
##
## $`9`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       9       9       9       9       9       9
##
## [1] "BAplus"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   91.34  101.20   98.50  110.59  403.23       6
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   99.21  106.32  106.13  113.33  196.85
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    99.2   106.7   106.4   114.7   196.9
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   103.1   110.3   110.2   117.5   396.8
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   96.86  105.10  104.86  112.94  396.83
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   94.39  101.97  101.49  109.68  396.83
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   91.34   99.22   98.59  107.06  396.83
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   88.24   96.47   95.68  104.38  396.83
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   83.33   93.15   91.59  101.57  396.83
```

```
##
## $`9`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   47.45   79.53   71.47   95.16  403.23       6
##
## [1] "OBPplus"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   92.45  100.62   98.33  108.64  315.46       4
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   98.75  104.95  104.83  111.36  157.73
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   98.11  104.46  105.11  111.95  209.09
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   103.2   109.7   110.9   118.2   309.6
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   98.43  105.36  105.62  112.85  309.60
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   95.25  101.55  101.33  107.96  309.60
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   92.55   98.75   98.49  105.02  309.60
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   89.31   96.54   95.68  102.82  309.60
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   85.85   93.79   92.07  100.93  315.46
##
## $`9`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   49.04   78.86   70.96   94.12  315.46       4
##
## [1] "SLGplus"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   87.77  100.25   98.22  113.47  493.83       6
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   91.37  100.00  100.79  108.97  156.48
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   93.43  103.21  104.51  115.28  182.72
```

```
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   105.2   116.3   116.0   126.5   229.9
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   102.8   112.4   112.8   123.2   281.7
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   96.55  105.75  105.86  116.09  256.83
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   91.08  100.00  100.10  110.80  229.89
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   84.35   95.07   94.29  104.32  333.16
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   76.53   88.05   86.67   98.32  329.74
##
## $`9`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   36.69   68.94   63.04   88.05  493.83       6
##
## [1] "OPSplus_alt"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   90.77  100.69   98.27  110.28  369.90       6
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    95.2   102.6   102.6   109.8   149.5
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   96.27  104.16  104.77  112.62  161.19
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   105.5   112.9   113.8   122.0   263.9
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   101.7   109.3   109.6   117.2   263.9
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    97.1   103.7   103.9   111.5   263.9
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    0.00   92.72   99.72   99.38  107.32  263.85
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   87.34   96.00   94.89  102.91  273.00
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   82.01   90.80   89.03   98.49  277.39
##
## $`9`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   42.22   74.17   66.50   90.26  369.90       6
##
## [1] "RC"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   29.12   55.19   55.09   78.95  155.12       4
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   47.74   66.93   66.64   85.48  151.96
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   45.79   68.58   67.90   89.29  142.81
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   69.81   87.96   87.31  105.29  155.12
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   56.62   77.55   75.37   92.81  155.11
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   42.29   62.02   60.48   78.83  142.81
##
## $`6`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   31.93   51.01   49.84   66.67  142.81
##
## $`7`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   21.80   40.69   40.97   58.11  142.81
##
## $`8`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   14.54   28.59   31.65   45.80  142.81
##
## $`9`
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##   0.0000   0.8485   4.7500  15.5856  26.0261 117.0000        4
```

While the statistical significance will have to be examined further, there appear to be differences across

lineup positions for many of the statistics. Based on this, linear regression seems reasonable.