Comparing Baseball Statistics Using Non-Parametric Analysis

Robert Tedesco

1/12/2022

1. Finding the "Best" Baseball Statistics

This work was inspired by Willis' 2001 paper which tested various offensive and defensive baseball statistics using sign tests to determine which statistics are "best" (https://visionlab.uncc.edu/downloads_new/arwillis/publications/reports/am168_final.pdf). Willis compared AVG and OBP, as well as ERA and BBA. Now, I will compare OBP to BABIP, OPS to OBP, and WHIP to ERA.

The first step here is to load the data regarding the variables mentioned above for all MLB teams for the years 1970-2015. This is easy to do using the Lahman package. Our final dataset should include the year, team, number of wins in the season, as well as the associated team statistics (ERA, WHIP, etc).

```
#First, let's load all the data using the Lahman package.
library(Lahman)
```

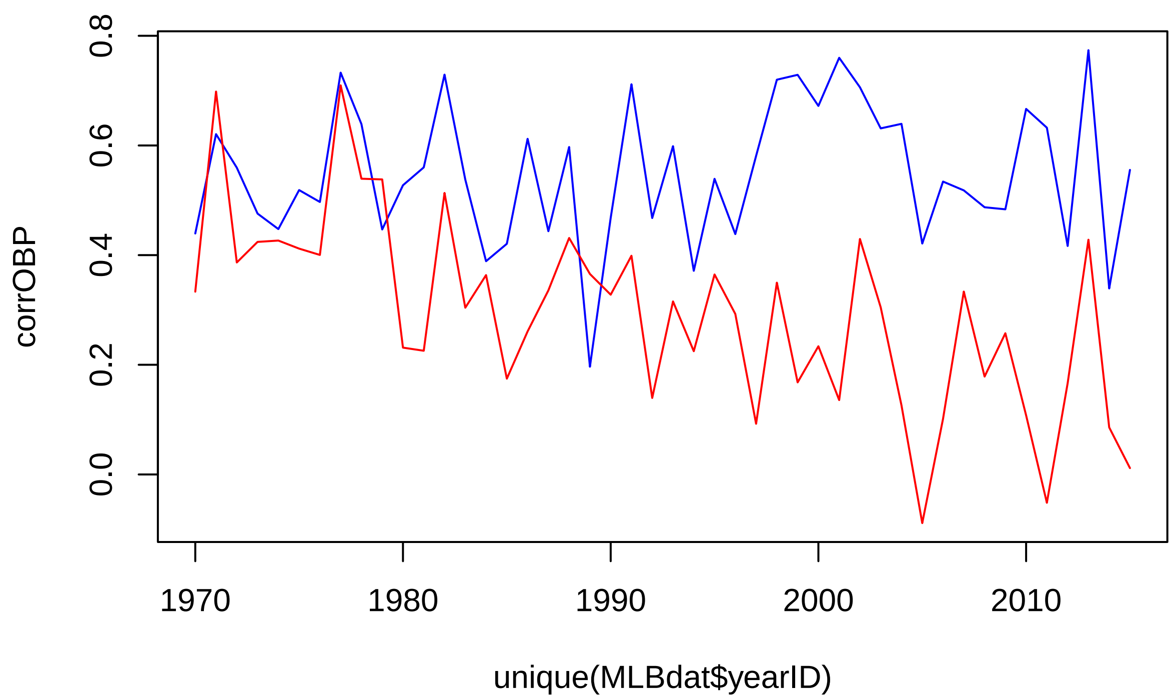
```
## Warning: package 'Lahman' was built under R version 4.1.2
```

```
MLBdat <- Teams %>%
  filter(yearID >= 1970 & yearID <=2015) %>%
  select(yearID, ERA, name, H, AB, IPouts, BB, HBP, SF,
         SO, W, L, X2B, X3B, HR) %>%
  mutate(
    BAOpp = round(H / (H + IPouts), 3),
    WHIP = round((H + BB) * 3 / IPouts, 2),
    PA = AB + BB + BB + HBP + SF,
    OB = H + BB + BB + HBP,
    OBP = 0 + (AB > 0) * round(OB / PA, 3),
    X1B = H - HR - X2B - X3B,
    SLG = (X1B + 2 * X2B + 3 * X3B + 4 * HR) / AB,
    OPS = OBP + SLG,
    BABIP = (H - HR) / (AB - HR - SO + SF)
  )
MLBdat <- MLBdat %>%
  select(yearID, name, W, ERA, WHIP, OBP, OPS, BABIP)
```

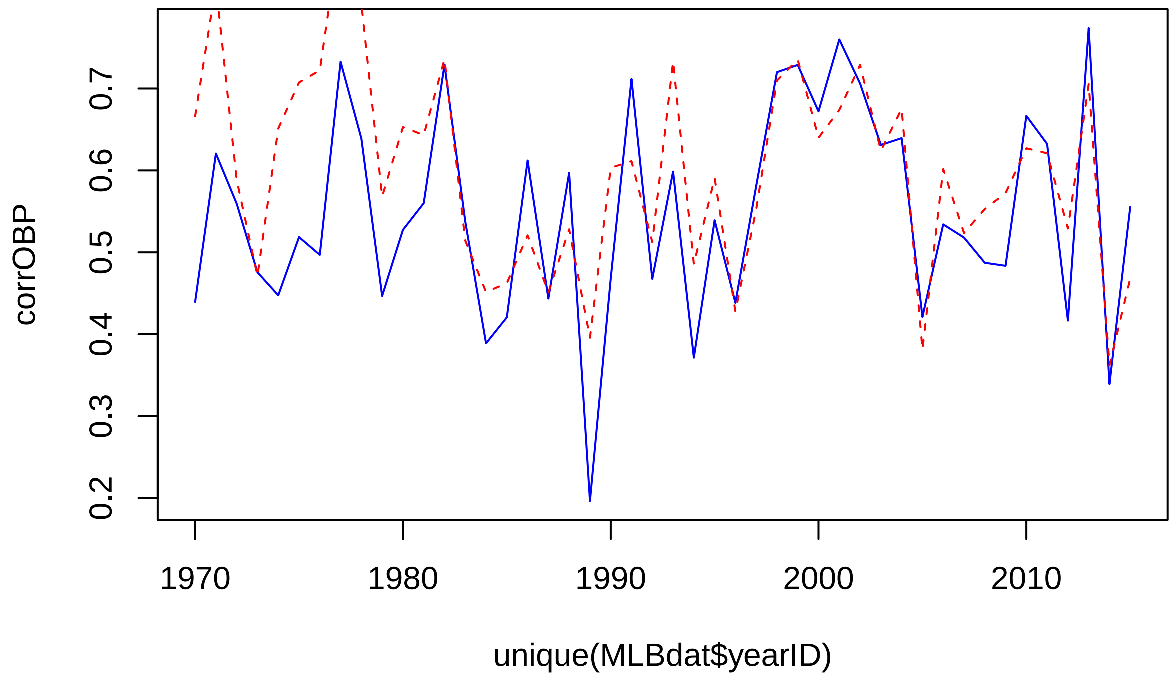
Now that the data is loaded, lets visualize correlations of various statistics with team wins for seasons between 1970-2015. One problem I have with Willis' original work is that pearson correlation is used without considering whether or not these pairs can be assumed to be bivariate normally distrbibuted.

```
###Correlations
corrOBP <- rep(0, length(unique(MLBdat$yearID)))
corrBABIP <- corrOBP
corrOPS <- corrOBP
corrWHIP <- corrOBP
corrERA <- corrOBP
#Deriving correlations for each season.
```

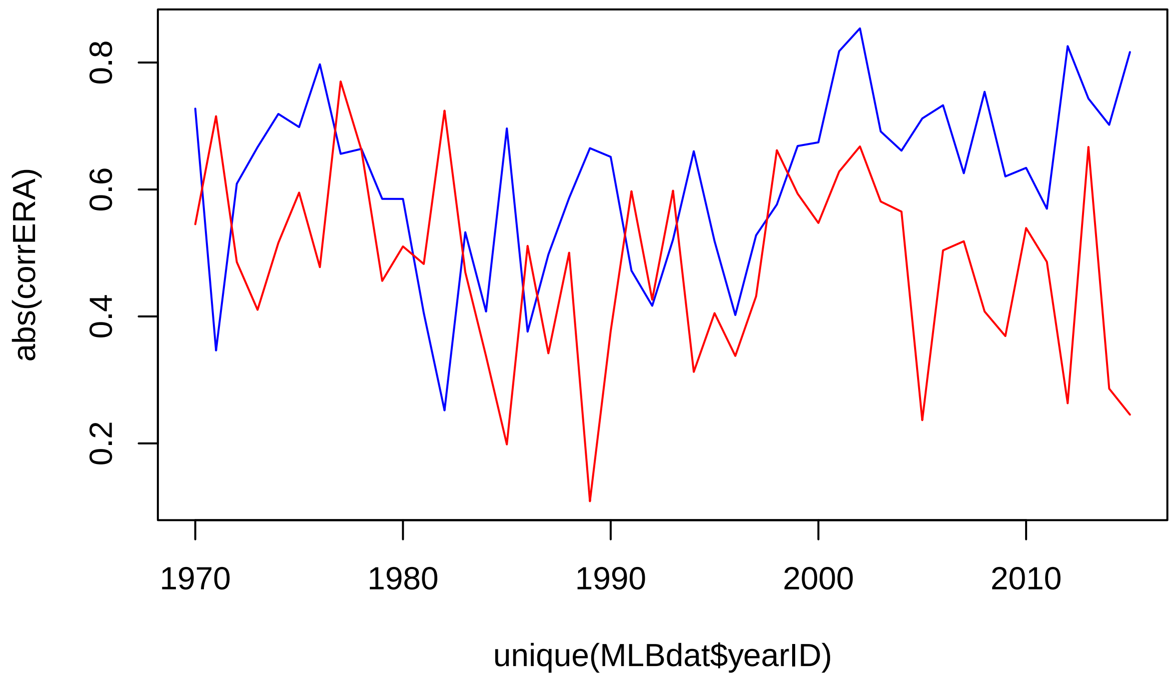
```
for(i in 1:length(unique(MLBdat$yearID))){
  tempDat <- MLBdat %>%
    filter(yearID==unique(MLBdat$yearID)[i])
  corrOBP[i] <- cor(tempDat$W, tempDat$OBP, method = c("spearman"))
  corrBABIP[i] <- cor(tempDat$W, tempDat$BABIP, method = c("spearman"))
  corrOPS[i] <- cor(tempDat$W, tempDat$OPS, method = c("spearman"))
  corrWHIP[i] <- cor(tempDat$W, tempDat$WHIP, method = c("spearman"))
  corrERA[i] <- cor(tempDat$W, tempDat$ERA, method = c("spearman"))
}

###Plot correlations over time

plot(unique(MLBdat$yearID), corrOBP, type="l", col="blue",
     ylim=range(c(corrOBP, corrBABIP)))
lines(unique(MLBdat$yearID), corrBABIP, type="l", col="red")
```




```
#We can observe visually that OBP seems to be the better statistic.

plot(unique(MLBdat$yearID), corrOBP, type="l", col="blue")
lines(unique(MLBdat$yearID), corrOPS, type="l", col="red", lty=2)
```




```
#We expect these two lines to be similar since OPS includes OBP

plot(unique(MLBdat$yearID), abs(corrERA), type="l", col="blue",
     ylim=range(c(abs(corrERA),corrWHIP)))
lines(unique(MLBdat$yearID), corrWHIP, col="red")
```




```r
#We have to use absolute value of ERA because of its inverse relationship.
#For some reason WHIP has a positive relationship with wins?
```

Finally, we can determine which statistic is the "best," if a best even exists. To do this we conduct multiple sign tests regarding the correlation of wins and the correlation of respective statistics using the following alternative hypotheses:

1. $$\mathbb{E}(corr(BABIP)) < \mathbb{E}(corr(OBP))$$
2. $$\mathbb{E}(corr(OBP)) < \mathbb{E}(corr(OPS))$$
3. $$\mathbb{E}(corr(WHIP)) < \mathbb{E}(corr(ERA))$$

The sign test was applied to see if there are significant differences in the mean correlation values for these statistics, this is equivalent to testing $\rho_x = \rho_y$.

```r
library(EnvStats)
```

```
## Warning: package 'EnvStats' was built under R version 4.1.2

##
## Attaching package: 'EnvStats'

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```r
set.seed(15)
signTest(corrBABIP, corrOBP, paired=T, alternative="less")
```

```
##
##  Paired Sign test
##
## data:  corrBABIPcorrOBP
## # Diffs > median of differences = 3, p-value = 2.311e-10
## alternative hypothesis: true median of differences is less than 0
## sample estimates:
## median of differences
##            -0.2480727
```

```r
signTest(corrOBP, corrOPS, paired=T, alternative="less")
```

```
##
##  Paired Sign test
##
## data:  corrOBPcorrOPS
## # Diffs > median of differences = 16, p-value = 0.02704
## alternative hypothesis: true median of differences is less than 0
## sample estimates:
## median of differences
##           -0.03244954
```

```r
signTest(corrWHIP, abs(corrERA), paired=T, alternative="less")
```

```
##
##  Paired Sign test
##
## data:  corrWHIPabs(corrERA)
## # Diffs > median of differences = 9, p-value = 2.028e-05
## alternative hypothesis: true median of differences is less than 0
## sample estimates:
## median of differences
##            -0.1087788
```

We reject all of the null hypotheses and come to the following conclusions:

1. BABIP is less correlated with team season wins than OPS.
2. OBP is less correlated with team wins than OPS!!!
3. WHIP is less correlated with team wins than ERA.

The biggest takeaway is point two, OPS takes OBP and just adds additional information (SLG). Therefore we can expect OPS to be more correlated with team wins and this analysis confirms that.

2. Bootstrapping 2021 Season

```r
#Load game by game data.
library(baseballr)
dates <-
  as.character(seq(as.Date("2021-04-01"), as.Date("2021-10-03") , by = "days"))
games <- data.frame(mlb_game_pks(dates[1]))[,c("game_pk",
                                               "teams.away.team.name",
                                               "teams.home.team.name",
                                               "status.detailedState",
                                               "isTie",
                                               "teams.away.isWinner")]
for (i in 2:length(dates)) {
  gameTemp <- tryCatch(
    mlb_game_pks(dates[i])[, c("game_pk",
                               "teams.away.team.name",
                               "teams.home.team.name",
                               "status.detailedState",
                               "isTie",
                               "teams.away.isWinner")]
    , error = function(e) {skip_to_next <<- TRUE}
  )
  games <- rbind(games, gameTemp)
}
#baseballR has one entry per game, we want one entry per team per game.
head(games)
```

```
##   game_pk teams.away.team.name  teams.home.team.name status.detailedState isTie
## 1  634642     Toronto Blue Jays      New York Yankees                Final FALSE
## 2  634645     Cleveland Indians        Detroit Tigers                Final FALSE
## 3  634638       Minnesota Twins     Milwaukee Brewers                Final FALSE
## 4  634634    Pittsburgh Pirates          Chicago Cubs                Final FALSE
## 5  634622        Atlanta Braves Philadelphia Phillies                Final FALSE
## 6  634615   Los Angeles Dodgers      Colorado Rockies                Final FALSE
##   teams.away.isWinner
## 1                TRUE
## 2               FALSE
## 3               FALSE
## 4                TRUE
## 5               FALSE
## 6               FALSE
```

```r
#Since baseballr does not allow us to have observations for each team...
#Replicate the dataframe and then bind them together!
games2 <- games[,c(1,2,4,5,6)]
colnames(games2)[2] <- "Team"
games3 <- games[,-2]
colnames(games3)[2] <- "Team"
games3$teams.away.isWinner <- as.logical(abs(
                              as.integer(games2$teams.away.isWinner)-1))
gamesFinal <- rbind(games2,games3)
#Now do more misc cleaning.
colnames(gamesFinal)[c(4,5)] <- c("Tie","Win")
gamesFinal <- gamesFinal[gamesFinal$game_pk!=1 &
                           gamesFinal$Team!="American League All-Stars" &
```

```r
                            gamesFinal$Team!="National League All-Stars",]
gamesFinal <- gamesFinal %>%
  filter(status.detailedState=="Final")
#Finally, we now have our desired data.frame
head(gamesFinal)
```

```
##   game_pk                Team status.detailedState   Tie   Win
## 1  634642   Toronto Blue Jays                Final FALSE  TRUE
## 2  634645   Cleveland Indians                Final FALSE FALSE
## 3  634638     Minnesota Twins                Final FALSE FALSE
## 4  634634  Pittsburgh Pirates                Final FALSE  TRUE
## 5  634622      Atlanta Braves                Final FALSE FALSE
## 6  634615 Los Angeles Dodgers                Final FALSE FALSE
```

```r
#Bootstrap n=1000 seasons of 162 games using 2021 game by game data above.
reps <- 1000
sim.results <- data.frame(Team=F, wins=F)[-1,]

for (i in 1:reps){
  mlb.sample <- gamesFinal %>% group_by(Team) %>% sample_n(size = 162, replace=TRUE)

  sample.standings <- mlb.sample %>% group_by(Team) %>%
    summarize(wins=sum(Win==T))

  sim.results <- rbind(sim.results, sample.standings)
}
sim.summary <- sim.results %>% group_by(Team) %>% summarize(
  min.wins = min(wins), max.wins=max(wins), sd(wins), mean(wins)
)
(sim.summary)
```

```
## # A tibble: 30 x 5
##    Team                 min.wins max.wins `sd(wins)` `mean(wins)`
##    <chr>                   <int>    <int>      <dbl>        <dbl>
##  1 Arizona Diamondbacks       32       72       6.04         52.6
##  2 Atlanta Braves             71      110       6.32         88.1
##  3 Baltimore Orioles          35       72       6.00         51.9
##  4 Boston Red Sox             71      111       6.16         92.1
##  5 Chicago Cubs               53       90       6.25         70.0
##  6 Chicago White Sox          69      117       6.55         92.4
##  7 Cincinnati Reds            63      101       6.20         82.6
##  8 Cleveland Indians          59      100       6.41         80.4
##  9 Colorado Rockies           56       90       6.22         74.4
## 10 Detroit Tigers             55      100       6.41         77.1
## # ... with 20 more rows
```

From our simulation summary we can see that the Mets have an average of 78 games won per simulated season while the Braves have 88. Since the standard deviation for all teams wins is roughly 5 or 6, we can say that the Mets are about two standard deviations away from their division leaders- they need to improve their team statistics!

3. Bootstrapping to compare two players

Let's say we want to find out whether or Pete Alonzo's OBP ABILITY is better than Francisco Lindor's OBP ABILITY . This is equivalent to testing:

$$H_0 : \pi_A = \pi_F$$

$$H_A : \pi_A > \pi_F$$

```
#I should have multiple years worth of data here...
#I used 2021 data from baseball-reference.com
Pete.PA <-  637
Lindor.PA       <-  524
Total.PA    <-  637 +   524
Total.OB    <-  169 +   219
total.OBP   <-  Total.OB/Total.PA
obs.diff.OBP    <-  .344 -  .322

sim.ARod.OBP    <-  rbinom(10000,   Pete.PA, total.OBP)/Pete.PA
sim.Mendoza.OBP <-  rbinom(10000,   Lindor.PA, total.OBP)/Lindor.PA
null.dist.vec       <-  sim.ARod.OBP - sim.Mendoza.OBP
p.value     <-  sum(null.dist.vec   >= obs.diff.OBP)/10000

p.value
```

```
## [1] 0.2139
```

There is insufficient evidence to conclude that either player has a significantly higher OBP ABILITY than the other. Again, more than one season worth of data should be considered in the future. Perhaps this techniques is very useful for evaluating new players against players already established in the league.