cleaning the batting projections

William Foote

3/15/2021

Exploratory functions

```
# nrow(bzip)
# nrow(bsteamer)
# nrow(bdc)
# nrow(bbat)
nrow(batc)
```

```
## [1] 629
```

Reorder largest to smallest

```
ordered <- c("bsteamer", "bzip", "batc", "bbat", "bdc")
```

The formula for Runs Created is as follows (roughly):

=((H+BB+HBP-CS)*(TB)+.26*(BB+HBP))+.52(SB))/(AB+BB+HBP))

Note

TB = SLG * AB

Make Runs Created

First, make TB

```
# bsteamer$TB <- with(bsteamer, SLG * AB)
# bzip$TB <- with(bzip, SLG * AB)
batc$TB <- with(batc, SLG * AB)
# bbat$TB <- with(bbat, SLG * AB)
# bdc$TB <- with(bdc, SLG * AB)
```

Next try making RC

```
# bsteamer$RC <- with(bsteamer, ((H+BB+HBP-CS)*(TB+.26*(BB+HBP))+.52*(SB))/(AB+BB+HBP))
# bzip$RC <- with(bzip, ((H+BB+HBP-CS)*(TB+.26*(BB+HBP))+.52*(SB))/(AB+BB+HBP))
batc$RC <- with(batc, ((H+BB+HBP-CS)*(TB+.26*(BB))+.52*(SB))/(AB+BB))
# bbat$RC <- with(bbat, ((H+BB-CS)*(TB+.26*(BB))+.52*(SB))/(AB+BB))
# bdc$RC <- with(bdc, ((H+BB+HBP-CS)*(TB+.26*(BB+HBP))+.52*(SB))/(AB+BB+HBP))
```

Note, bbat and batc don't have HBP projected

Split into a separate data frame

Helper function

```r
cats <- c("R", "HR", "RBI", "RC", "AVG", "OPS")

me1 <- function(x, y, cat = "R", by = "playerid") { # If i were 1
  temp <- merge(x[, c("Name", by, cat)], y[, c(by, cat)], by = by)
  temp
}

me2 <- function(x, y, cat = "R", by = "playerid") { # If i were greater than 1
  temp <- merge(x, y[, c(by, cat)], by = by)
  temp
}
```

Cleaning the data didnt use in 2022

```r
# # temp_r <- me1(bsteamer, bzip, cat = "R")
# tr2 <- me2(temp_r, batc, cat = "R")
# # tr3 <- me2(tr2, bbat, cat = "R")
# # tr4 <- me2(tr3, bdc, cat = "R")
# # tr4
#
# temp_hr <- me1(bsteamer, bzip, cat = "HR")
# thr2 <- me2(temp_hr, batc, cat = "HR")
# thr3 <- me2(thr2, bbat, cat = "HR")
# thr4 <- me2(thr3, bdc, cat = "HR")
# thr4
#
#
# temp_rbi <- me1(bsteamer, bzip, cat = "RBI")
# trbi2 <- me2(temp_rbi, batc, cat = "RBI")
# trbi3 <- me2(trbi2, bbat, cat = "RBI")
# trbi4 <- me2(trbi3, bdc, cat = "RBI")
# trbi4
#
# temp_rc <- me1(bsteamer, bzip, cat = "RC")
# trc2 <- me2(temp_rc, batc, cat = "RC")
# trc3 <- me2(trc2, bbat, cat = "RC")
# trc4 <- me2(trc3, bdc, cat = "RC")
# trc4
#
# temp_avg <- me1(bsteamer, bzip, cat = "AVG")
# tavg2 <- me2(temp_avg, batc, cat = "AVG")
# tavg3 <- me2(tavg2, bbat, cat = "AVG")
# tavg4 <- me2(tavg3, bdc, cat = "AVG")
# tavg4
#
# temp_ops <- me1(bsteamer, bzip, cat = "OPS")
# tops2 <- me2(temp_ops, batc, cat = "OPS")
# tops3 <- me2(tops2, bbat, cat = "OPS")
# tops4 <- me2(tops3, bdc, cat = "OPS")
# tops4
#
# batting <- cbind(tr4[, c(1, 2)], R = apply(tr4[, c(-1, -2)], 1, mean))
# batting <- cbind(batting, HR = cbind(thr4[, c(1, 2)], HR = apply(thr4[, c(-1, -2)], 1, mean))[, -c(1,
# batting <- cbind(batting, RBI = cbind(trbi4[, c(1, 2)], RBI = apply(trbi4[, c(-1, -2)], 1, mean))[, -
# batting <- cbind(batting, RC = cbind(trc4[, c(1, 2)], RC = apply(trc4[, c(-1, -2)], 1, mean))[, -c(1,
# batting <- cbind(batting, AVG = cbind(tavg4[, c(1, 2)], AVG = apply(tavg4[, c(-1, -2)], 1, mean))[, -
# batting <- cbind(batting, OPS = cbind(tops4[, c(1, 2)], OPS = apply(tops4[, c(-1, -2)], 1, mean))[, -
# batting
```

2022

```
batting <- batc[, c(1, 2, 32, 10, 9, 11, 34, 18, 21)]
```

Write in the z-score function

```
z_score <- function(x) {
  x <- as.numeric(x)
  m <- mean(x, na.rm = TRUE)
  s <- sd(x, na.rm = TRUE)
  z <- (x - m) / s
  z
}
```

Use the z-score function

```
batting_2 <- batting
temp <- apply(batting_2[, -c(1:3)], 2, z_score)
batting_2 <- cbind(batting_2[, c(1, 2, 3)], temp)
sums <- apply(batting_2[, -c(1, 2, 3)], 1, sum)
batting_3 <- batting_2
batting_3$pr <- sums
batting_ordered <- batting_3[order(batting_3$pr, decreasing = TRUE), ]
batting_ordered$rank <- 1:629
```

See if it works

```
batting_in_order <- batting_3[with(batting_3, order(batting_3$pr, decreasing = TRUE)), ]
head(batting_in_order)
```

```
##                     Name Team playerid        R       HR      RBI        RC
## 1              Juan Soto  WSN    20123 2.555867 2.521952 2.401650 3.490595
## 6  Vladimir Guerrero Jr.  TOR    19611 2.381748 3.029246 2.507809 2.999555
## 2     Fernando Tatis Jr.  SDP    19709 2.416572 3.232163 2.366264 2.450470
## 10          Bryce Harper  PHI    11579 2.207629 2.623411 2.224719 2.692109
## 3             Mike Trout  LAA    10155 1.963863 2.521952 1.977016 2.542047
## 16       Freddie Freeman          5361 2.242453 2.014658 2.012402 2.627802
##          AVG      OPS       pr
## 1  3.150420 4.423115 18.54360
## 6  2.876609 3.517792 17.31276
## 2  1.963907 3.312037 15.74141
## 10 1.553190 3.270886 14.57194
## 3  1.735731 3.723548 14.46416
## 16 2.374623 2.708489 13.98043
```

```
# write.csv(batting_in_order, "batc_pr.csv")
```