Homerun Derby Model

Paul Hatini

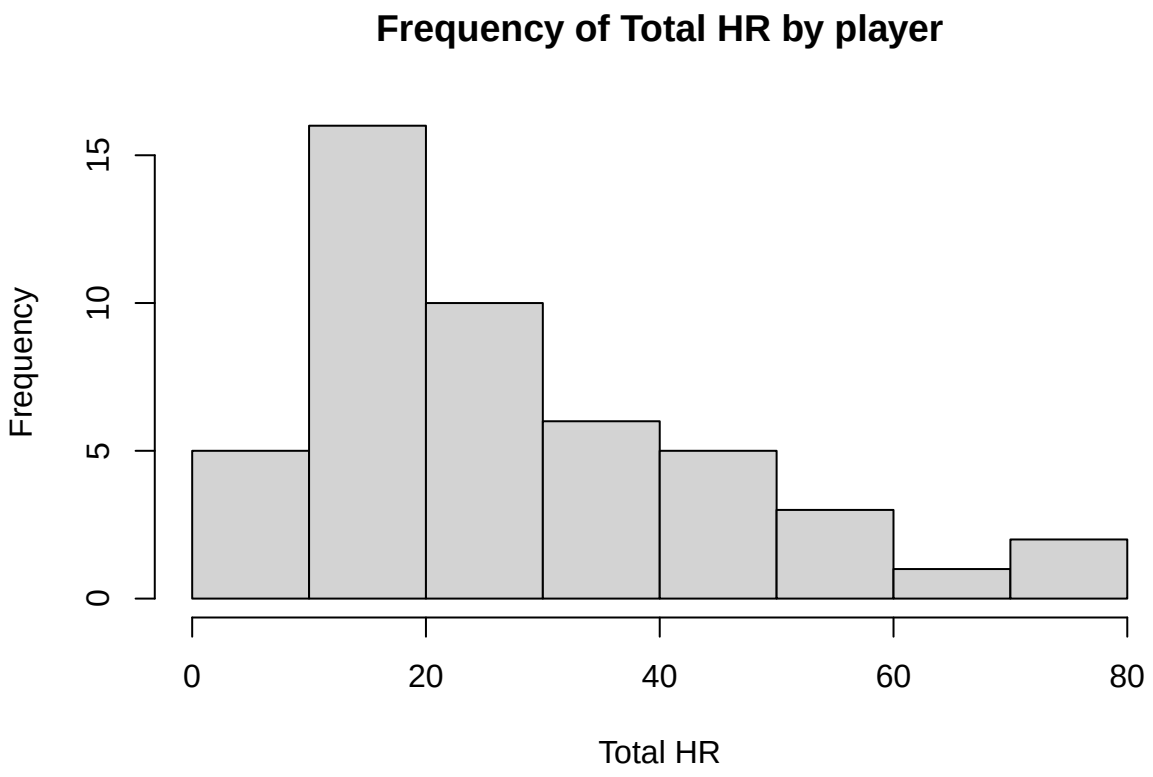
7/26/2021

```
library(stats)
```

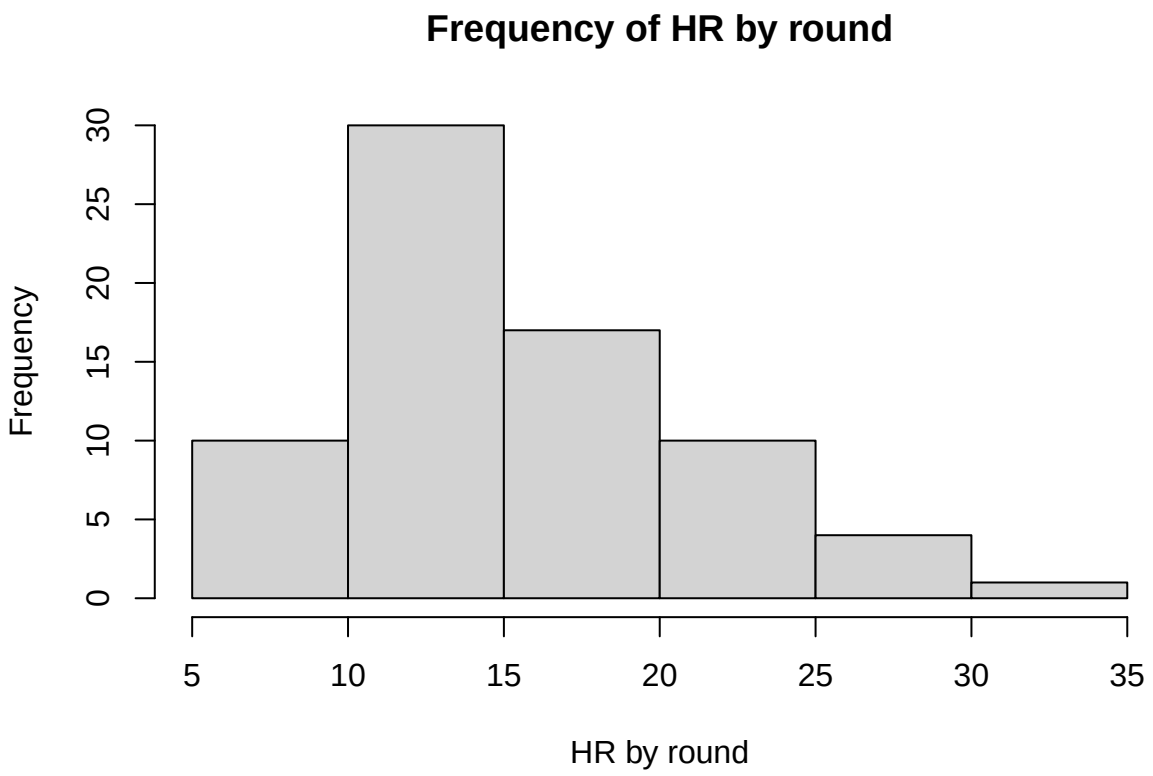
```
hrd21 <- read.csv("hrd21.csv", header = TRUE)
hrd19 <- read.csv("hrd19.csv", header = TRUE)
hrd18 <- read.csv("hrd18.csv", header = TRUE)
hrd17 <- read.csv("hrd17.csv", header = TRUE)
hrd16 <- read.csv("hrd16.csv", header = TRUE)
hrd15 <- read.csv("hrd15.csv", header = TRUE)

hrd21$Year <- 2021
hrd19$Year <- 2019
hrd18$Year <- 2018
hrd17$Year <- 2017
hrd16$Year <- 2016
hrd15$Year <- 2015

hrd_stats <- rbind(hrd21, hrd19, hrd18, hrd17, hrd16, hrd15)
hrd_results <- read.csv("hrd_results.csv", header = TRUE)
hrd <- merge(hrd_results, hrd_stats, by=c("Year", "Name"))

hrd$Semifinals[is.na(hrd$Semifinals)] <- 0
hrd$Final[is.na(hrd$Final)] <- 0
hrd$average <- hrd$Total/(rowSums(hrd[,c(3,4,5)] !=0))

hrd <- hrd[,colSums(is.na(hrd)) == 0] # remove NA
hrd$Age.Rng <- NULL # remove age range

clean <- function(x) {
  gsub("\\%", "", x)
} # function to remove %

hrd <- data.frame(apply(hrd, 2, clean)) # apply clean function by column

hrd$Dol <- as.numeric(gsub("\\$", "", hrd$Dol))
hrd$Dol[40] <- 0.5

hrd[,9:259] <- data.frame(apply(hrd[,9:259], 2, as.numeric))
hrd$Total <- as.numeric(hrd$Total)

write.csv(hrd, file="hrd.csv")

hrd$playerid <- NULL
```

```
hrd$Events <- NULL
hrd$HardHit <- NULL
hrd$Barrels <- NULL

hrd$Quarterfinals <- as.numeric(hrd$Quarterfinals)
hrd$Semifinals <- as.numeric(hrd$Semifinals)
hrd$Final <- as.numeric(hrd$Final)
```

```
hist(hrd$Total, main="Frequency of Total HR by player", xlab="Total HR")
```




```
hist(c(hrd$Quarterfinals[which(hrd$Quarterfinals != 0)],
       hrd$Semifinals[which(hrd$Semifinals != 0)], hrd$Finals[which(hrd$Finals != 0)]), main="Frequency
```



Above, I've plotted a histogram of home run totals per round and per derby. These appear approximately normally distributed.

Regression

In order to predict the winner of the 2021 home run derby I've built a multiple regression model shown below. As features I've included relevant batting statistics including wOBA, swing percentage, contact percentage, hard hit percentage, average exit velocity, max exit velocity, launch angle, barrel percentage, and various others displayed in the model below. The outcome variable here is HR by round. I hitter data from fangraphs.com and compiled the HR totals and by round by hand from records. I employed a backward stepwise feature selection technique to identify features of high importance.

```
hrd_small <- data.frame(cbind(hrd$average, hrd$Age, hrd$IBB, hrd$HBP, hrd$BB.K, hrd$ISO,
                              hrd$GB.FB, hrd$wOBA, hrd$WAR, hrd$Dol, hrd$Clutch, hrd$FB..1,
                              hrd$wFB, hrd$Swing., hrd$Contact., hrd$Zone., hrd$Pull.,
                              hrd$Hard., hrd$TTO., hrd$EV, hrd$LA, hrd$Barrel., hrd$maxEV))
colnames(hrd_small) <- c("average", "Age", "IBB","HBP", "BB.K", "ISO", "GB.FB", "wOBA",
                         "WAR", "Dol", "Clutch", "FB..1", "wFB", "Swing.", "Contact.",
                         "Zone.", "Pull.", "Hard.", "TTO.", "EV", "LA", "Barrel.", "maxEV")

LR_fit_full_small <- lm(average ~ ., data = hrd_small[0:40,], )

LR_fit_full_small_select <- step(LR_fit_full_small,direction="backward",test="F")
```

The model resulting from backward stepwise selection is displayed below Hard hit percentage and max exit velocity are the strongest predictors of HR derby round performance. Adjusted R-squared 0.4714 is decent. Model p-value 0.0007331 suggests that with high confidence there is a linear association between predictors and outcome.

```
summary(LR_fit_full_small_select)
```

```
##
## Call:
## lm(formula = average ~ Age + IBB + ISO + Dol + Clutch + Pull. +
##     Hard. + TTO. + EV + maxEV, data = hrd_small[0:40, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6250 -1.8107 -0.1232  1.7917  5.0217
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.75411   32.84064  -0.602 0.552171
## Age          -0.33069    0.18936  -1.746 0.091323 .
## IBB          -0.23434    0.18133  -1.292 0.206449
## ISO         -28.68964   16.21632  -1.769 0.087380 .
## Dol          -0.15854    0.07879  -2.012 0.053583 .
## Clutch       -1.58128    0.79393  -1.992 0.055895 .
## Pull.         0.27398    0.15102   1.814 0.080002 .
## Hard.         0.65501    0.17878   3.664 0.000989 ***
## TTO.         -0.17220    0.09238  -1.864 0.072474 .
## EV           -0.65758    0.43791  -1.502 0.144000
## maxEV         0.72941    0.23020   3.169 0.003595 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.184 on 29 degrees of freedom
## Multiple R-squared:  0.6069, Adjusted R-squared:  0.4714
## F-statistic: 4.478 on 10 and 29 DF,  p-value: 0.0007331
```

Predictions for HR by round for the 2021 homerun derby participants are displayed below.

```
predictions <- predict(LR_fit_full_small_select, newdata = hrd_small[41:48,], type = "response")
cbind(hrd$Name[41:48], predictions)
```

```
##                     predictions
## 41 "Joey Gallo"     "12.1800694398639"
## 42 "Juan Soto"      "12.931270389646"
## 43 "Matt Olson"     "11.4718726387867"
## 44 "Pete Alonso"    "18.2468174053453"
## 45 "Salvador Perez" "16.7231398591619"
## 46 "Shohei Ohtani"  "10.7620649935548"
## 47 "Trevor Story"   "13.1046784906206"
## 48 "Trey Mancini"   "14.5018129686311"
```

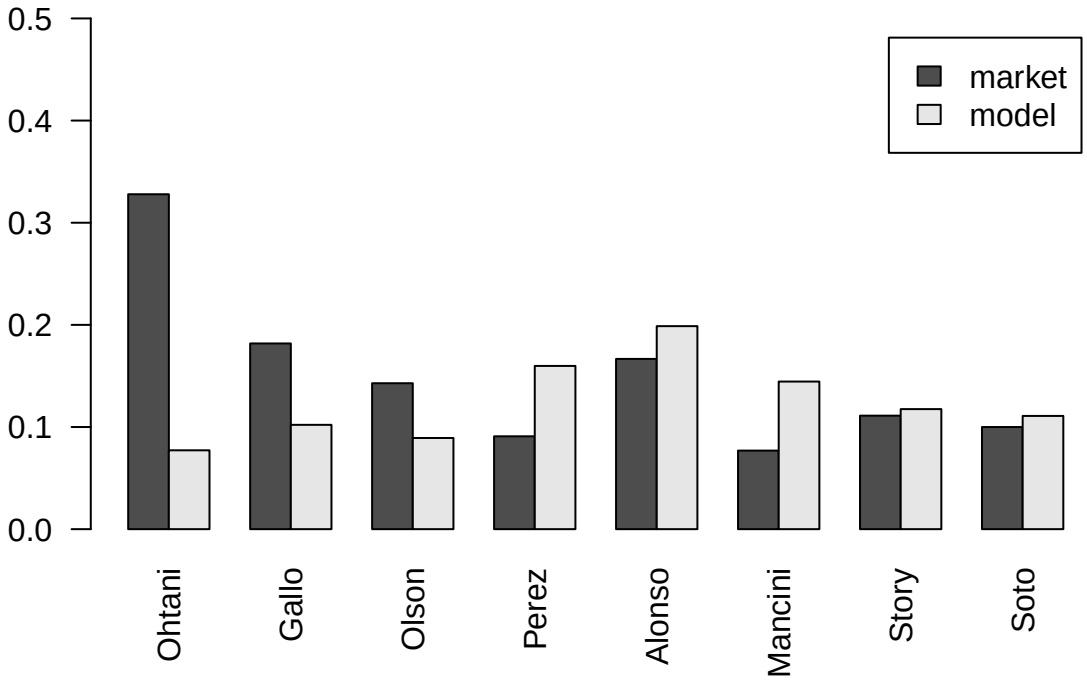
Assuming that HR samples from each player are normally distributed around their mean, the probability of a sample drawn from one distribution being greater than a sample drawn from another can be calculated analytically. That is, the probability that one player will beat another in HR derby matchup can be computed. As a result, the HR derby can be simulated and winning probability can be determined.

```
pred_2021 <- as.data.frame(cbind(hrd$Name[41:48], hrd$Odds[41:48], c(2,8,3,5,4,1,7,6), predictions))
colnames(pred_2021) <- c("names", "odds", "seeds", "pred")
pred_2021$pred <- as.numeric(pred_2021$pred)
pred_2021$odds <- as.numeric(pred_2021$odds)
pred_2021 <- pred_2021[order(pred_2021$seeds),]

var <- (summary(LR_fit_full_small_select)$sigma)**2
```

```r
hrd$Quarterfinals <- as.numeric(hrd$Quarterfinals)
hrd$Semifinals <- as.numeric(hrd$Semifinals)
hrd$Final <- as.numeric(hrd$Final)

vec <- c(hrd$Quarterfinals[which(hrd$Quarterfinals != 0)], hrd$Semifinals[which(hrd$Semifinals != 0)], h


sim_matchup <- function(pred_1, pred_2, var_1, var_2) {
  prob <- pnorm(0, mean=(pred_1-pred_2), sd=var_1+var_2, lower.tail=FALSE)
  sim <- rbinom(1,1,prob)
  if (sim==1) {return(pred_1)} else {return(pred_2)}
}
```

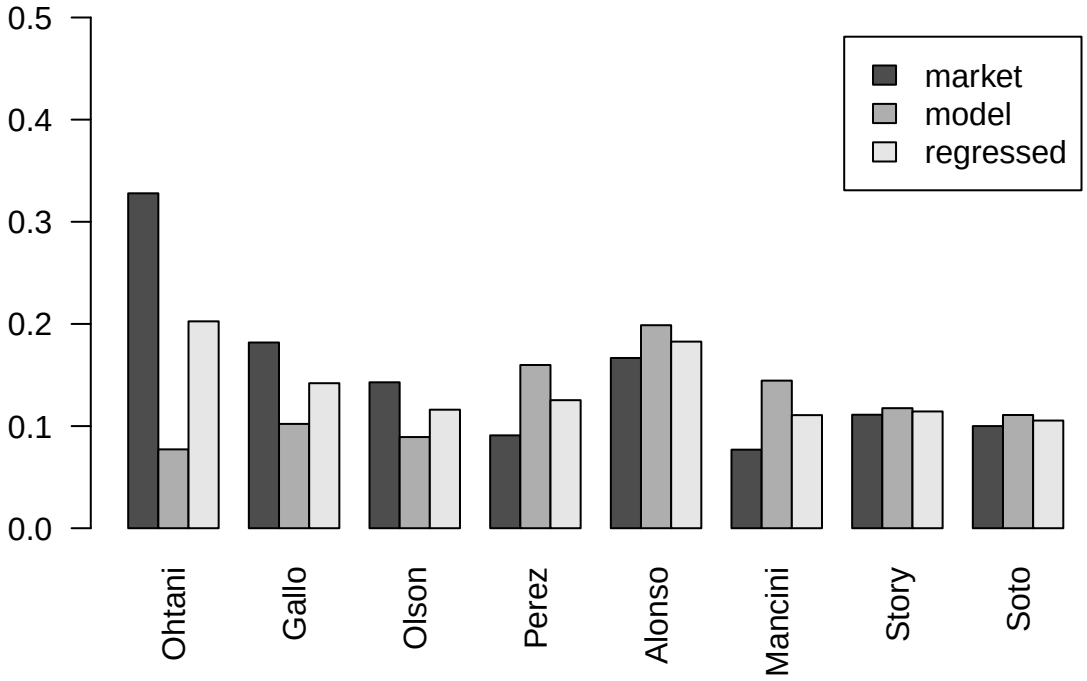
```r
#pred_2021[order(pred_2021$seeds),]

sim_derby <- function(pred){
  return(sim_matchup(sim_matchup(sim_matchup(pred[1], pred[8], var, var), sim_matchup(pred[4], pred[5],
              sim_matchup(sim_matchup(pred[2], pred[7], var, var), sim_matchup(pred[3], pred[6], var, va
}

n <- 1000000
pred <- as.numeric(pred_2021$pred[order(pred_2021$seeds)])
sims <- replicate(n, sim_derby(pred), simplify=FALSE)

probs <- c(length(sims[which(sims==pred[1])])/n,
length(sims[which(sims==pred[2])])/n,
length(sims[which(sims==pred[3])])/n,
length(sims[which(sims==pred[4])])/n,
length(sims[which(sims==pred[5])])/n,
length(sims[which(sims==pred[6])])/n,
length(sims[which(sims==pred[7])])/n,
length(sims[which(sims==pred[8])])/n)

pred_2021 <- cbind(pred_2021, probs)

odds_to_probs <- function(odds) {
  return(100/(odds+100))
}

pred_2021 <- cbind(pred_2021, sapply(pred_2021$odds, odds_to_probs))
```

```r
barplot(rbind(sapply(pred_2021$odds, odds_to_probs), probs), beside = TRUE, ylim=c(0,0.5), names.arg=c(
```



Obviously, my model is in fair disagreement with the market. In order to respect the consensus I'll regress my predictions back to the market odds 50/50.

```
colnames(pred_2021)[which(colnames(pred_2021) == "sapply(pred_2021$odds, odds_to_probs)")] <- "market"

pred_2021$regress <- 0.5*pred_2021$probs + 0.5*pred_2021$market

barplot(rbind(pred_2021$market, probs, pred_2021$regress), beside = TRUE, ylim=c(0,0.5), names.arg=c("O
```



```
pred_2021[which(pred_2021$regress/pred_2021$market > 1.05),]
```

```
##             names odds seeds     pred    probs     market   regress
## 45 Salvador Perez 1000     4 16.72314 0.159798 0.09090909 0.1253535
## 44    Pete Alonso  500     5 18.24682 0.198744 0.16666667 0.1827053
## 48   Trey Mancini 1200     6 14.50181 0.144503 0.07692308 0.1107130
## 42      Juan Soto  900     8 12.93127 0.110846 0.10000000 0.1054230
```

My betting strategy:

```
allocation <- (pred_2021$regress[which(pred_2021$regress/pred_2021$market > 1.05)] - pred_2021$market[wh

#allocation*pred_2021$odds[which(pred_2021$regress/pred_2021$market > 1.05)]

result <- c("LOSS", "WIN", "LOSS", "LOSS")
settle <- c(allocation[1]*-100,
            allocation[2]*pred_2021$odds[which(pred_2021$regress/pred_2021$market > 1.05)][2],
            allocation[c(3,4)]*-100)

#sum(allocation[c(1,3,4)])*-100 + allocation[2]*pred_2021$odds[which(pred_2021$regress/pred_2021$market

settle
```

```
## [1] -38.401292  89.405614 -37.671613  -6.045972
```

```
name <- pred_2021$names[which(pred_2021$regress/pred_2021$market > 1.05)]
market <- pred_2021$market[which(pred_2021$regress/pred_2021$market > 1.05)]
regress <- pred_2021$regress[which(pred_2021$regress/pred_2021$market > 1.05)]
edge <- (regress-market)#/market ## redo this
allocation <- edge/sum(edge)


df <- data.frame(name = name,
                 market = market,
                 regress = regress,
                 edge = edge,
                 allocation = allocation,
                 result = result,
                 settle = settle)
knitr::kable(df)
```

name	market	regress	edge	allocation	result	settle
Salvador Perez	0.0909091	0.1253535	0.0344445	0.3840129	LOSS	-38.401292
Pete Alonso	0.1666667	0.1827053	0.0160387	0.1788112	WIN	89.405614
Trey Mancini	0.0769231	0.1107130	0.0337900	0.3767161	LOSS	-37.671613
Juan Soto	0.1000000	0.1054230	0.0054230	0.0604597	LOSS	-6.045972

With this unit allocation, strategy I'd profit 7.46%.

```
allocation <- (pred_2021$probs[which(pred_2021$probs/pred_2021$market > 1.05)] - pred_2021$market[which

result <- c("LOSS", "WIN", "LOSS", "LOSS", "LOSS")
settle <- c(allocation[1]*-100,
            allocation[2]*pred_2021$odds[which(pred_2021$probs/pred_2021$market > 1.05)][2],
            allocation[c(3,4,5)]*-100)
```

```
#sum(allocation[c(1,3,4)])*-100 + allocation[2]*pred_2021$odds[which(pred_2021$probs/pred_2021$market >


name <- pred_2021$names[which(pred_2021$probs/pred_2021$market > 1.05)]
market <- pred_2021$market[which(pred_2021$probs/pred_2021$market > 1.05)]
model <- pred_2021$probs[which(pred_2021$probs/pred_2021$market > 1.05)]
edge <- (model-market)/market ## redo this
allocation <- edge/sum(edge)
settle <- c(allocation[1]*-100,
            allocation[2]*pred_2021$odds[which(pred_2021$probs/pred_2021$market > 1.05)][2],
            allocation[c(3,4,5)]*-100)


df <- data.frame(name = name,
                 market = market,
                 model = model,
                 edge = edge,
                 allocation = allocation,
                 result = result,
                 settle = settle)
knitr::kable(df)
```

name	market	model	edge	allocation	result	settle
Salvador Perez	0.0909091	0.159798	0.757778	0.3798451	LOSS	-37.984507
Pete Alonso	0.1666667	0.198744	0.192464	0.0964748	WIN	48.237414
Trey Mancini	0.0769231	0.144503	0.878539	0.4403779	LOSS	-44.037793
Trevor Story	0.1111111	0.117525	0.057725	0.0289353	LOSS	-2.893533
Juan Soto	0.1000000	0.110846	0.108460	0.0543668	LOSS	-5.436684