Trying to Understand Pitcher Ability in Limiting Extra Base Hits

Owen McGrattan

1/4/2019

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(readr)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```r
# read in batted ball data
pitch_17 <- read_csv("statcast_pitch_2017.csv")
pitch_18 <- read_csv("statcast_pitch_2018.csv")
batted <- rbind(pitch_17, pitch_18)
batted <- filter(batted, !(is.na(barrel)), description != "foul")


all_pitchers_17 <- pitch_17 %>% filter(pitch_number == 1) %>% group_by(player_name) %>% summarise(tbf =
all_pitchers_18 <- pitch_18 %>% filter(pitch_number == 1) %>% group_by(player_name) %>% summarise(tbf =

rm(pitch_17, pitch_18)
```

```r
# calculate spray angle
batted$hc_x <- batted$hc_x - 125.42
```

```r
batted$hc_y <- 198.27 - batted$hc_y
batted$spray_angle <- round(
  atan((batted$hc_x) / (batted$hc_y)) * 180 / pi * 0.75)
# make sure we dont have weird spray angles or launch angles
batted <- batted %>% filter(spray_angle <= 45, spray_angle >= -45, launch_angle >= -50, launch_angle <=
# filter out NAs from events
batted <- batted[!(is.na(batted$spray_angle)) & !(is.na(batted$events)) & !(is.na(batted$if_fielding_al
```

```r
# predicting hit probabilities is nice, but let's get more specific,
# get probabilities for 2bs 3bs and hrs
# create a binary variable denoting extra base hits
xtras <- rep(0, length(batted$events))
xtra <- c("double", "triple", "home_run")
# add a binary variable denoting hit or no hit
for (i in 1:length(batted$events)) {
  if (batted$events[i] %in% xtra) {
    xtras[i] <- 1
  } else {
    xtras[i] <- 0
  }
}

batted$xtras <- xtras
```

```r
batted$xtras <- as.factor(batted$xtras)
batted$if_fielding_alignment <- as.factor(batted$if_fielding_alignment)
batted$of_fielding_alignment <- as.factor(batted$of_fielding_alignment)
batted$stand <- as.factor(batted$stand)
batted$events <- as.factor(batted$events)
```

```r
set.seed(27)
# oversample our extra base hits before fitting our model
over_sample <- ovun.sample(xtras ~ (launch_angle) + (launch_speed) + (spray_angle) + p_throws + stand,
```

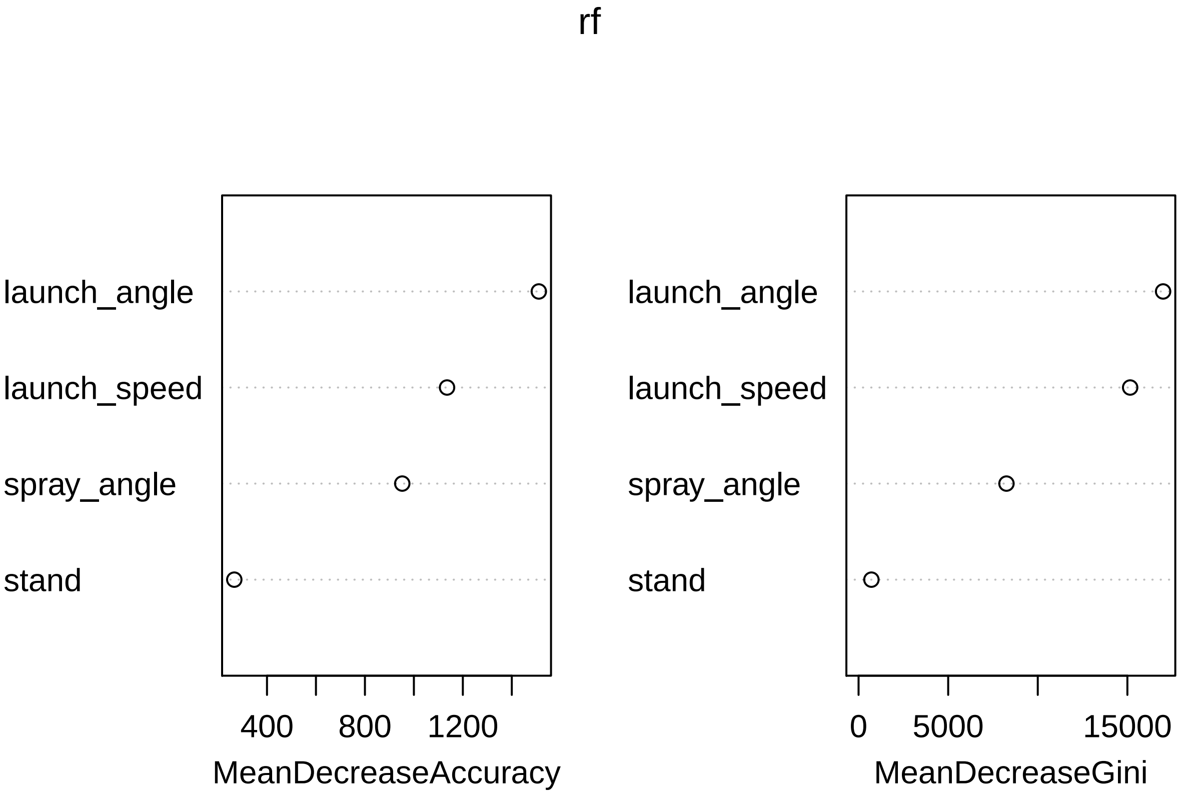
```
## Warning in (function (formula, data, method, subset, na.action, N, p = 0.5, : Transformations of var
##  New data have been generated by using non-transformed variables.
##
```

```r
set.seed(27)
rf <- randomForest(xtras ~ (launch_angle) + (launch_speed) + (spray_angle) + stand , data = over_sample
```

```r
rf
```

```
##
## Call:
##  randomForest(formula = xtras ~ (launch_angle) + (launch_speed) +      (spray_angle) + stand, data =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 2.75%
## Confusion matrix:
##       0     1 class.error
## 0 50083  2276 0.043469127
## 1   117 34494 0.003380428
```

```r
varImpPlot(rf)
```



```r
# gather predictions and probabilities
preds <- predict(rf, batted, type = "prob")
batted$prob_extra <- preds[,2]
```

```r
# let's determine most ideal probability cutoff
p_s <- seq(0.3, 0.9, by = 0.025)
acc <- rep(0, length(p_s))
for (i in 1:length(p_s)) {
  pred <- as.vector(preds[,2])
  pred[pred > p_s[i]] = 1
  pred[pred <= p_s[i]] = 0
  acc[i] <- mean(pred == batted$xtras)
}

p_s[which.max(acc)]
```

```
## [1] 0.65
```

```r
# add xbh prob pcts for 2018 players
eight <- filter(batted, game_year == 2018)
uniq <- unique(sort(eight$player_name))
xXBH <- rep(0, length(uniq))
XBH <- rep(0, length(uniq))
gb <- rep(0, length(uniq))
ld <- rep(0, length(uniq))
fb <- rep(0, length(uniq))
pu <- rep(0, length(uniq))
```

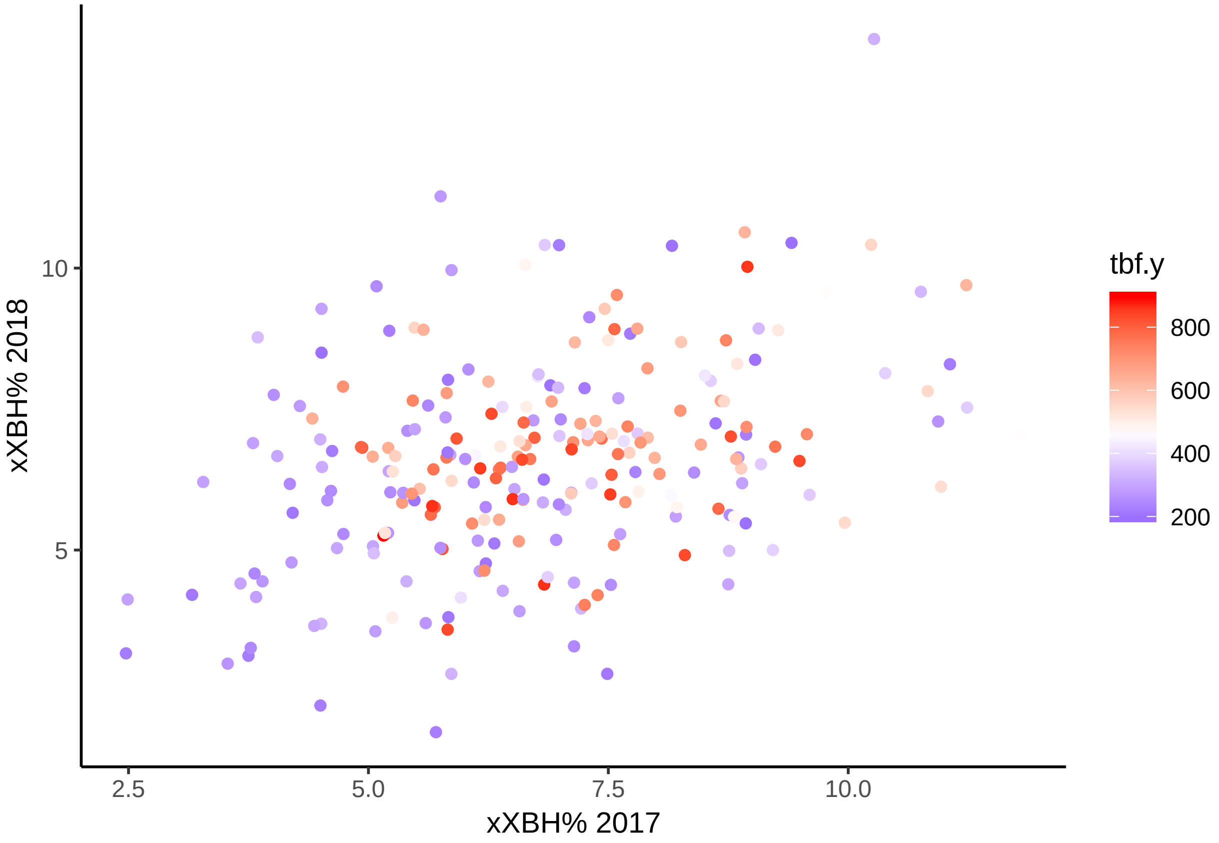
```r
for (i in 1:length(uniq)) {
  plyr <- filter(batted, player_name == uniq[i], game_year == 2018)
  xXBH[i] <- nrow(filter(plyr, prob_extra >= 0.65))
  XBH[i] <- sum(as.numeric(as.character(plyr$xtras)))
  gb[i] <- nrow(filter(plyr, bb_type == "ground_ball")) / nrow(plyr)
  fb[i] <- nrow(filter(plyr, bb_type == "fly_ball")) / nrow(plyr)
  ld[i] <- nrow(filter(plyr, bb_type == "line_drive")) / nrow(plyr)
  pu[i] <- nrow(filter(plyr, bb_type == "popup")) / nrow(plyr)
}

# group by for each of our pitchers
gru <- batted %>% filter(game_year == 2018) %>%
  group_by(player_name) %>%
  summarise(
    exit_velo = mean(launch_speed, na.rm = TRUE),
    launch_angle = mean(launch_angle, na.rm = TRUE),
    n = n()
)


gru$xXBH <- xXBH
gru$XBH <- XBH
gru$ld <- ld * 100
gru$fb <- fb * 100
gru$pu <- pu * 100
gru$gb <- gb * 100
gru <- merge(gru, all_pitchers_18, by = "player_name")
```

```r
# insert 2017 guys
# lowest extra base hit probabilities
seven <- filter(batted, game_year == 2017)
uniq <- unique(sort(seven$player_name))

xXBH <- rep(0, length(uniq))
XBH <- rep(0, length(uniq))
gb <- rep(0, length(uniq))
ld <- rep(0, length(uniq))
fb <- rep(0, length(uniq))
pu <- rep(0, length(uniq))
for (i in 1:length(uniq)) {
  plyr <- filter(batted, player_name == uniq[i], game_year == 2017)
  xXBH[i] <- nrow(filter(plyr, prob_extra >= 0.65))
  XBH[i] <- sum(as.numeric(as.character(plyr$xtras)))
  gb[i] <- nrow(filter(plyr, bb_type == "ground_ball")) / nrow(plyr)
  fb[i] <- nrow(filter(plyr, bb_type == "fly_ball")) / nrow(plyr)
  ld[i] <- nrow(filter(plyr, bb_type == "line_drive")) / nrow(plyr)
  pu[i] <- nrow(filter(plyr, bb_type == "popup")) / nrow(plyr)
}


# group by for each of our pitchers
gru_7 <- batted %>% filter(game_year == 2017) %>%
  group_by(player_name) %>%
  summarise(
    exit_velo = mean(launch_speed, na.rm = TRUE),
    launch_angle = mean(launch_angle, na.rm = TRUE),
    n = n()
)

gru_7$xXBH <- xXBH
gru_7$XBH <- XBH
gru_7$ld <- ld * 100
gru_7$fb <- fb * 100
gru_7$pu <- pu * 100
gru_7$gb <- gb * 100
gru_7 <- merge(gru_7, all_pitchers_17, by = "player_name")
```

```
mer <- merge(gru_7, gru, by = "player_name")
mer <- filter(mer, tbf.x >= 200, tbf.y >= 200)
```

```
# xXBH / batters faced
mer$xbh_pct.x <- ((mer$xXBH.x / mer$tbf.x) * 100)
mer$xbh_pct.y <- ((mer$xXBH.y / mer$tbf.y) * 100)
```

```
# pct_xbh 17 & 18
ggplot(mer) +
  geom_point(aes(x = xbh_pct.x, y = xbh_pct.y, color = tbf.y)) +
  labs(x = "xXBH% 2017", y = "xXBH% 2018") +
  scale_color_gradient2(midpoint=mean(mer$tbf.y), low="blue", mid="white",
                        high="red", space ="Lab" ) +
  theme_classic()
```



```r
mer$x_minus_a_18 <- ((mer$xXBH.y - mer$XBH.y) / mer$n.y) * 100
mer$x_minus_a_17 <- ((mer$xXBH.x - mer$XBH.x) / mer$n.x) * 100
```