Lesson_39_Boardsheet

Kevin Cummiskey

5/6/2020

```
library(tidyverse)
library(baseballr)
library(viridis)
library(lubridate)
```

Today, we are going to look at how the probability of a hit changes with launch angle and speed.

https://www.mlb.com/news/new-statcast-metric-barrels-has-best-hit-balls-c201699298

First, let's download StatCast data for 2017 using the scrape_statcast_savant function. Darn, it only lets us download 40,000 pitches (about 15 days) at a time. Ok, maybe I should start doing this:

```
#Note R stores dates as the number of days since Jan 1, 1970



#March 31, 2017 to April 13, 2017
scrape_statcast_savant(start_date = as_date(17256), end_date = as_date(17270))
#April 14, 2017 to April 28, 2017
scrape_statcast_savant(start_date = as_date(17271), end_date = as_date(17285))

# This is going to get tedious going all the way to September
```

What does splitting, applying, combining mean?

How can we use this concept to make it easier to read in this data?

```
dates <- tibble(start_date = as_date(seq(17226, 17454, 15)),
                end_date = as_date(start_date + 14))

dates
```

```
## # A tibble: 16 x 2
##    start_date end_date
##    <date>     <date>
##  1 2017-03-01 2017-03-15
##  2 2017-03-16 2017-03-30
##  3 2017-03-31 2017-04-14
##  4 2017-04-15 2017-04-29
##  5 2017-04-30 2017-05-14
##  6 2017-05-15 2017-05-29
##  7 2017-05-30 2017-06-13
##  8 2017-06-14 2017-06-28
##  9 2017-06-29 2017-07-13
## 10 2017-07-14 2017-07-28
## 11 2017-07-29 2017-08-12
## 12 2017-08-13 2017-08-27
## 13 2017-08-28 2017-09-11
## 14 2017-09-12 2017-09-26
## 15 2017-09-27 2017-10-11
## 16 2017-10-12 2017-10-26
```

Now, let's apply the scrape_statcaset_savant using the rows of `dates` as the arguments on the function.

```
dates %>%
  pmap(scrape_statcast_savant) %>%
  do.call(rbind, .) %>%
  filter(type == "X") -> batted2017
```

Now, we have a data frame called `batted2017` containing all batted balls in the 2017 season.

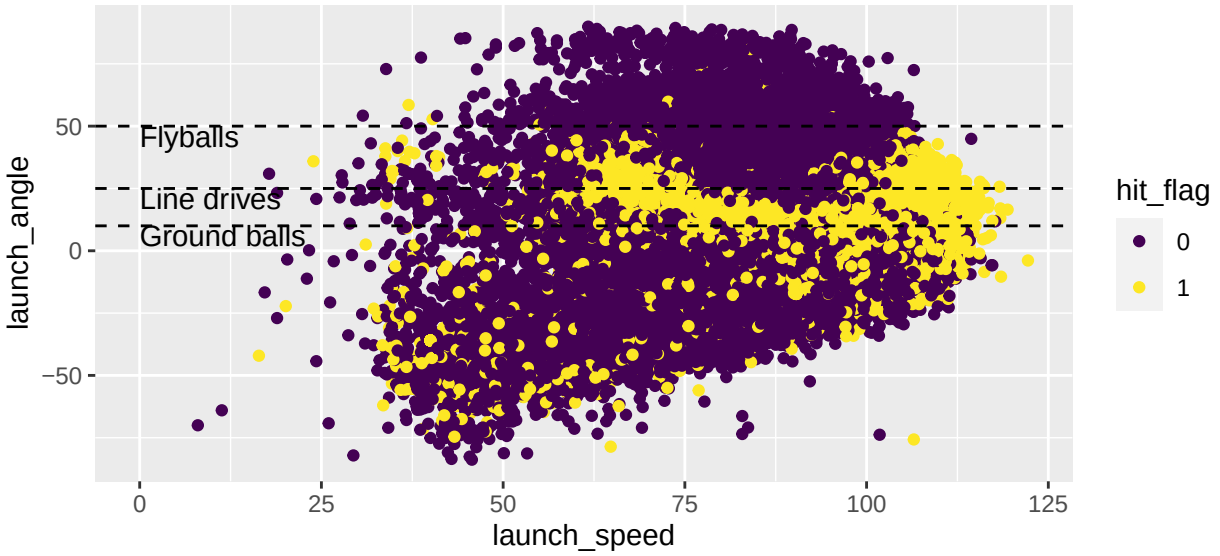
```
# determine if batted ball was a hit
batted2017 %>%
  mutate(hit_flag = factor(ifelse(events %in% c("single", "double",
                                                "triple", "home_run"),1,0))) -> batted2017

#Plot each batted ball by whether it's a hit or not
guidelines <- tibble(
  launch_angle = c(10,25,50),
  launch_speed = 40,
  label = c("Ground balls", "Line drives", "Flyballs")
)

ev_plot <- batted2017 %>%
  sample_n(nrow(.)/2) %>%
  ggplot(aes(x = launch_speed,
             y = launch_angle,
             color = hit_flag)) +
  geom_point() +
  scale_color_viridis(discrete = TRUE) +
  geom_hline(data = guidelines,
             aes(yintercept = launch_angle),
             color = "black",
             linetype = 2) +
  geom_text(data = guidelines,
```

```
            aes(label = label, x = 0, y = launch_angle - 4),
            color = "black", hjust = "left")

ev_plot
```



Instead of the plot above, why might we want to view hit probability as a smoothed function of launch speed and angle?

Is a linear function of launch speed and velocity appropriate for this data?

Let's fit a generalized additive model for hit probability as a smoothed function of launch speed and angle. Let $Y_i \sim \text{Bernoulli}(\pi_i)$ equal 1 when batted ball i is a hit, and 0 otherwise. Our model is:

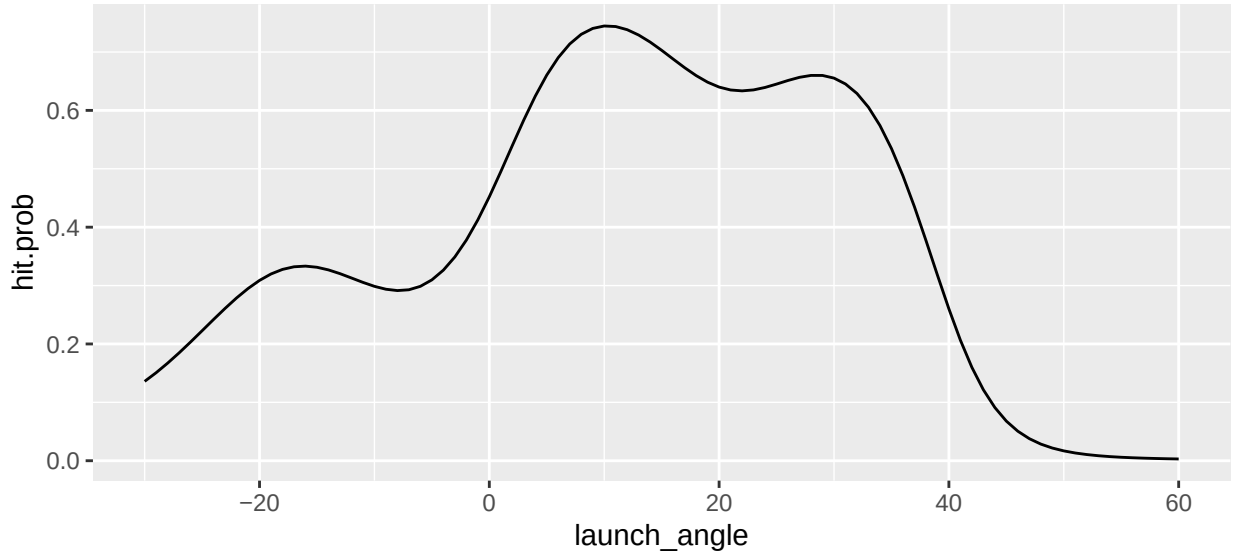
$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = s(Angle_i, Speed_i)$$

```
library(mgcv)

fit <- batted2017 %>%
  gam(hit_flag ~ s(launch_speed, launch_angle),
      family = "binomial",
      data = .)
```

Let's look at some predictions from the model. First, let's look at batted balls with an speed for 100mph for various launch angles.

```
batted100mph <- tibble(launch_angle = seq(-30,60),
                       launch_speed = 100) %>%
  mutate(hit.prob = predict(fit, type = "response",
                  newdata = .))
batted100mph %>%
  ggplot(aes(x = launch_angle,
             y = hit.prob)) +
  geom_line()
```

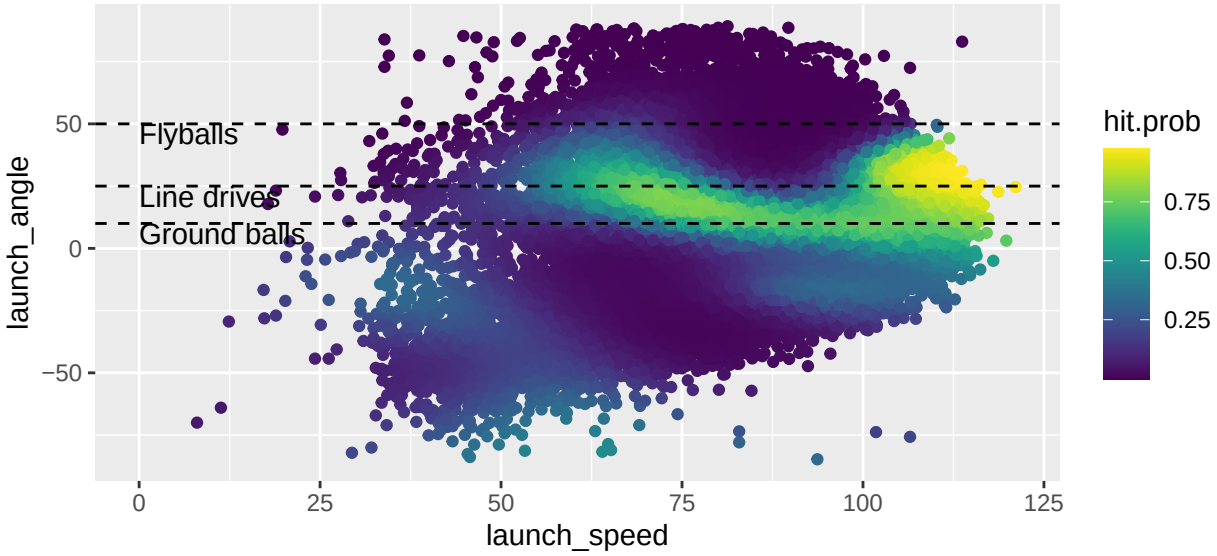


Here are hit probabilities for each batted ball in the 2017.

```
batted2017 %>%
  mutate(hit.prob = predict(fit,
                            type = "response",
                            newdata = .)) -> batted2017

ev_plot2 <- batted2017 %>%
  sample_n(nrow(.)/2) %>%
  ggplot(aes(x = launch_speed,
             y = launch_angle,
             color = hit.prob)) +
  geom_point() +
  scale_color_viridis() +
  geom_hline(data = guidelines,
             aes(yintercept = launch_angle),
             color = "black",
             linetype = 2) +
  geom_text(data = guidelines,
            aes(label = label, x = 0, y = launch_angle - 4),
            color = "black", hjust = "left")

ev_plot2
```



Let's find the batted ball with the highest hit probability that was not a hit.

```
batted2017 %>%
  filter(hit_flag == 0) %>%
  arrange(-hit.prob) %>%
  head(1) %>%
  select(player_name, launch_angle, launch_speed, events)
```

```
## # A tibble: 1 x 4
##   player_name    launch_angle launch_speed events
##   <chr>                 <dbl>        <dbl> <chr>
## 1 Miguel Cabrera         30.4         108. field_out
```