MA388 Sabermetrics: Lesson 15

Catcher Framing Ability Modeling

LTC Jim Pleuss

Last class, we began comparing called strikes for Buster Posey and Yadier Molina. Today let's look at their results from June 2019.

Let's grab the data first. This should look familiar (see lesson 13).

```
library(tidyverse)
library(knitr)
library(broom)
library(baseballr)
```

```
# Retrieve pitch level data from June 2019. (See previous lesson.)
# If you've already collected the pitch data below, just load it.  Otherwise,
# collect it from scratch.

if("statcast_june_2019.rds" %in% dir(getwd())){
  pitches <- readRDS("statcast_june_2019.rds")
}else{

  # Retrieve pitch-level data from June 2019.

get_statcast_pitches <- function(start_day, end_day, chunk_size = 5) {

  # Coerce to Date
```

```
  start_day <- as.Date(start_day)
  end_day   <- as.Date(end_day)

  # Create sequence of chunk start dates
  chunk_starts <- seq(start_day, end_day, by = paste(chunk_size, "days"))

  # Build data
  pitch_data <- map_dfr(chunk_starts, function(chunk_start) {

    chunk_end <- min(chunk_start + days(chunk_size - 1), end_day)
       statcast_search(
        start_date = chunk_start,
        end_date   = chunk_end
      )
  })
  return(pitch_data)
}

  # If we want to limit to a certain number of pitches, we might take the head()
  # or sample_n() to get the number we want.
  pitches <- get_statcast_pitches('2019-06-01','2019-06-30')

  saveRDS(pitches, "statcast_june_2019.rds")
}

pitches <- readRDS("statcast_june_2019.rds")
```

Now we can pull in the catcher name and display the strike and ball counts for both catchers.

```
# Add catcher's name to the pitch data from the MLB master list.

mlbIDs <-
  baseballr::chadwick_player_lu() |>
  mutate(mlb_name = paste(name_first, name_last),
    mlb_id = key_mlbam)

pitches <- pitches |>
  left_join(select(mlbIDs, mlb_name, mlb_id),
            by = c("fielder_2" = "mlb_id")) |>
  rename(catcher_name = mlb_name)

# Look only at pitches taken when Molina or Posey are catching.
pitches_taken_subset <- pitches |>
  filter(catcher_name %in% c("Buster Posey", "Yadier Molina"),
```

```
        description %in% c("ball", "called_strike")) |>
  mutate(called_strike=description =='called_strike')

# Form a 2x2 table.
pitches_taken_subset |>
  count(catcher_name, description) |>
  pivot_wider(id_cols = description,
              names_from = catcher_name,
              values_from = n) |>
  kable(caption = "Results of taken pitches (June 2019)")
```

Table 1: Results of taken pitches (June 2019)

description	Buster Posey	Yadier Molina
ball	608	764
called_strike	295	391

Logistic Regression

We can use logistic regression to adjust for confounding variables. Consider the results in Table 1 and the following logistic regression model for the data. Let Y_i be a random variable for whether pitch i was a called strike such that $Y_i \sim \text{Bernoulli}(\pi_i)$ and

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 \text{Molina}_i$$

where $\text{Molina}_i = 1$ if Yadier Molina was catcher and $\text{Molina}_i = 0$ if Buster Posey was catcher.

Fit the model above and report the final model equation.

Based on this model, is there evidence Molina has more called strikes in the long term?

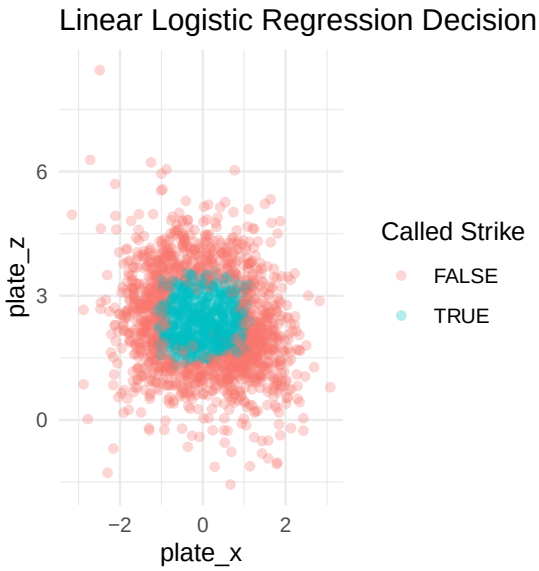
A better analysis would also adjust for the location of the pitch. We need a

model for called strikes based on location. You might be tempted to consider the following model:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 \text{plate\_x}_i + \beta_2 \text{plate\_z}_i$$

Is this an effective model for adjusting for location? Explain.

```
# Fit simple linear logistic model
linear_fit <- pitches_taken_subset |>
  glm(called_strike ~ plate_x + plate_z,
      data = _,
      family = "binomial")

# Create grid over strike zone region
grid <- expand.grid(
  plate_x = seq(-2, 2, length.out = 200),
  plate_z = seq(0, 5, length.out = 200)
)

# Predicted probabilities
grid$prob <- predict(linear_fit, newdata = grid, type = "response")



# Plot
ggplot() +
  geom_point(
    data = pitches_taken_subset,
    aes(x = plate_x, y = plate_z, color = factor(called_strike)),
    alpha = 0.3
  ) +
  coord_equal() +
  labs(
    title = "Linear Logistic Regression Decision Boundary",
    color = "Called Strike"
  ) +
  theme_minimal()
```



Let's consider an alternative, specifically a smooth function of the location. We can write the model like this:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + f(\text{plate\_x}_i, \text{plate\_z}_i)$$

Fit the model in R using the `mgcv` package (Section 7.4).

```
library(mgcv)

strike_mod <- gam(description == "called_strike" ~ s(plate_x, plate_z), family = "binomial",
strike_mod |> summary()
```

```
Family: binomial
Link function: logit

Formula:
description == "called_strike" ~ s(plate_x, plate_z)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -13.394      5.124  -2.614  0.00895 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Approximate significance of smooth terms:
                     edf Ref.df Chi.sq p-value
s(plate_x,plate_z) 25.01  26.11  276.9  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.756   Deviance explained = 73.5%
UBRE = -0.63749  Scale est. = 1         n = 2057
```

We want this model to produce strike predictions based on location. Let's look at the predicted strike probability for a pitch at `plate_x = -1` and `plate_z = 2.5` (Section 7.4.1). This would place the pitch just outside the left edge of home plate (from the pitcher's perspective) in the vertical middle of the strike zone.

```
strike_mod |>
  augment(type.predict = "response",
          newdata = data.frame(plate_x = -1,
                               plate_z = 2.5))
```

```
# A tibble: 1 x 4
  plate_x plate_z .fitted .se.fit
    <dbl>   <dbl>   <dbl>   <dbl>
1      -1     2.5   0.405  0.0875
```

Now, let's look at the prediction surface. First, we create a grid of points and then calculate the predictions at those points (Section 7.4.2).

```
library(modelr) #data_grid function

# Create a grid.
grid <- pitches_taken_subset |>
  data_grid(plate_x = seq_range(plate_x, n = 100),
            plate_z = seq_range(plate_z, n = 100))

grid |> head(5)
```

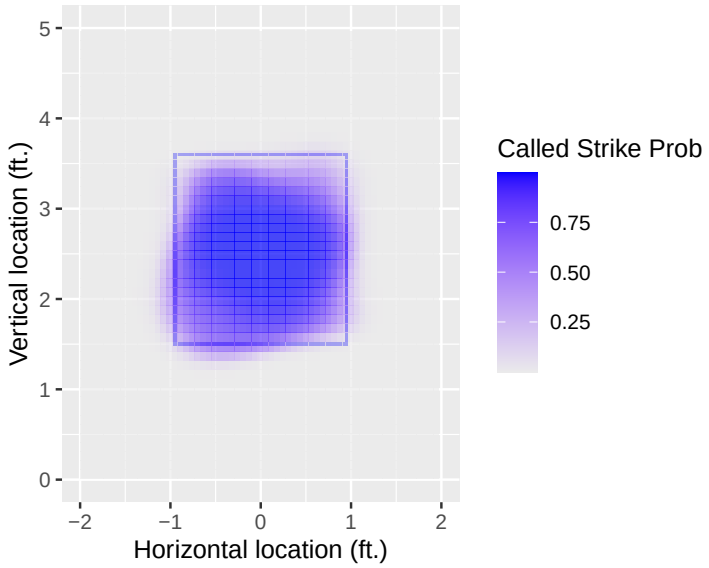
```
# A tibble: 5 x 2
  plate_x plate_z
    <dbl>   <dbl>
1   -3.16   -1.56
2   -3.16   -1.46
3   -3.16   -1.36
4   -3.16   -1.26
5   -3.16   -1.16
```

```
# Calculate predicted probabilities for strikes on the grid.
grid <- strike_mod |>
  augment(type.predict = "response",
          newdata = grid)

grid |> head(5)
```

```
# A tibble: 5 x 4
  plate_x plate_z .fitted .se.fit
    <dbl>   <dbl>   <dbl>   <dbl>
1   -3.16   -1.56 0.00275   0.761
2   -3.16   -1.46 0.00260   0.703
3   -3.16   -1.36 0.00240   0.632
4   -3.16   -1.26 0.00215   0.552
5   -3.16   -1.16 0.00187   0.468
```

Plot the results on the strike zone.

```
plate_width <- 17 + 2 * (9/pi)
k_zone_plot <- ggplot(NULL, aes(x = plate_x, y = plate_z)) +
  geom_rect(xmin = -(plate_width/2)/12,
            xmax = (plate_width/2)/12,
            ymin = 1.5,
            ymax = 3.6,
            color = "blue",
            alpha = 0) +
  coord_equal() +
  scale_x_continuous("Horizontal location (ft.)", limits = c(-2,2)) +
  scale_y_continuous("Vertical location (ft.)", limits = c(0,5))

k_zone_plot %+%
  grid +
  geom_tile(aes(fill = .fitted), alpha = 0.7) +
  scale_fill_gradient(low = "gray92", high = "blue") +
  labs(fill = "Called Strike Prob")
```



Now, we have a model for called strikes and pitch location. We can use this model adjust for pitch location in our catcher model:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 \text{Molina}_i + f(\text{plate\_x}_i, \text{plate\_z}_i)$$

Interpret β_1 in this model.

```
strike_mod_molina <- gam(description == "called_strike" ~ s(plate_x, plate_z) + catcher_name
                  family = "binomial", data = pitches_taken_subset)
strike_mod_molina |>
  summary()
```

```
Family: binomial
Link function: logit

Formula:
description == "called_strike" ~ s(plate_x, plate_z) + catcher_name

Parametric coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -13.5836     5.1579  -2.634  0.00845 **
```

```
catcher_nameYadier Molina   0.2329     0.1981   1.175  0.23986
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                     edf Ref.df Chi.sq p-value
s(plate_x,plate_z) 25.04  26.13  276.2  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.757   Deviance explained = 73.6%
UBRE = -0.63719  Scale est. = 1         n = 2057
```

Based on this analysis, is there evidence that Molina has a higher called strike probability after adjusting for pitch location? Explain.

Right now we're looking at just two catchers in a binary categorical sense. How might we extend this to many catchers at the same time?