UNITED STATES MILITARY ACADEMY

SCORE FINAL SUBMISSION

MA388: SABERMETRICS

SECTION C1

LTC MIKE POWELL

BY

CDT EVAN SMIDA ‘26, CO A1
CDT JOHN GOERTEMILLER ‘25, CO A1

WEST POINT, NEW YORK

9 MAY 2024

JG
ES I CERTIFY THAT I HAVE COMPLETELY DOCUMENTED ALL SOURCES THAT I USED TO COMPLETE THIS ASSIGNMENT AND THAT I ACKNOWLEDGED ALL ASSISTANCE I RECEIVED IN THE COMPLETION OF THIS ASSIGNMENT.

_____ I CERTIFY THAT I DID NOT USE ANY SOURCES OR RECEIVE ANY ASSISTANCE REQUIRING DOCUMENTATION WHILE COMPLETING THIS ASSIGNMENT.

SIGNATURES: Evan Smida John Goertemiller

MA388 Sabermetrics: Final SCORE Module Submission

Statcast: Questionable Game-Ending Strike 3 Calls

CDT Evan Smida and CDT John Goertemiller

1) Learning Objectives

By completing this module we hope for participants:

- *to gain an understanding of Generalized Additive Models (GAMs)*
- *to be able to create visualizations to better understand multiple types of models*
- *to determine which scenarios require linear regression vs GAMs*

2) Introduction

In baseball, some of the most controversial plays are those which occur towards the end of the game, where nearly everyone is tired and perhaps ready to go home. When this applies to the umpires, we see some crazy calls in the last few innings.

Video:

*https://www.youtube.com/watch?v=6zvwd-DtzdQ

While the video demonstrates an egregiously bad call, there are pitches more moderate than that which are still a potentially questionable called strike. Mistakes likes these make us wonder, would we be better off with automated officiating? This lesson aims to introduce students to the tools necessary answer such questions through the use of Generalized Additive Models (GAMs) in an effort to understand how game-ending strike 3 calls may be classified as questionable, and in the process try to quantify how much better off the MLB might be.

3) Data

In our module we began our analysis with Statcast data from 2023. Statcast is the technology that tracks pitch and player positioning data for every pitch since 2015. This Statcast data was put into a csv file that we loaded, but individual pitch data can be searched for and found at https://baseballsavant.mlb.com/statcast_search. Our data frame, "df_23", contains pitch-level data on every ball thrown during the 2023 MLB season. It contains 716,373 observations or pitches, and carries data about the pitch speed, location, type, and more. For the purposes of this exploration, plate location (x and z) and description (called strike or ball) are the most important variables.

```
library(tidyverse)
library(knitr)
library(broom)
library(ggrepel)
```

```r
df_23 <- read_csv("statcast2023.csv")
df_23 %>% select(pitch_type, release_speed, plate_x, plate_z, description) %>%
  head(10) %>% kable()
```

pitch_type	release_speed	plate_x	plate_z	description
CH	80.4	-0.56	0.98	swinging_strike_blocked
CH	80.2	0.22	2.25	foul
CU	81.6	-0.12	2.66	foul
CH	79.0	-0.36	1.30	ball
CH	80.0	-0.52	1.80	foul_tip
FF	95.1	0.20	2.00	hit_into_play
SI	90.8	-1.35	2.12	ball
CU	80.9	0.83	-0.48	blocked_ball
SI	94.4	-0.81	1.90	called_strike
CU	80.3	0.43	1.67	foul

4) Methods/Resources

In making this module, we used the textbooks "Analyzing Baseball Data with R (3rd Edition)" by Jim Albert et al. and "An Introduction to Statistical Learning (2nd Edition)" by Gareth James et al. The first textbook gave us experience with coding GAMs, getting us acquainted with the syntax for the GAM function, as well as numerous others throughout the module. The second textbook was used to get a more fundamental understanding of generalized additive models, what they accomplish, their strengths and weaknesses, and how they differ from other statistical techniques.

A GAM is much like a linear regression model, but a GAM can handle nonlinear relationships as well, doing a better job at representing the interactions between our independent and dependent variables.

The general notation for Generalized Additive Models is

$$g(E(y_i)) = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \ldots + f_p(x_{ip})$$

An important assumption we make in using this model is that each explanatory variable is independent of the others. We will explore this in our example below.

5) Exercise

Let's start by find game-ending pitches that were called strikes. What criteria must be met for a pitch to end the game?

- Greater than or equal to the 9th inning, 2 outs, 2 strikes, and the batting team must have fewer runs scored than the fielding team.

How many pitches fit these criteria?

```r
df_23 %>%
  filter(description == "called_strike",
         outs_when_up == 2,
         strikes == 2,
```

```
        inning >= 9,
        bat_score < fld_score) %>%
  nrow()
```

```
## [1] 173
```

Now we know that there were 173 instances in which the batting team's half inning was finished because of a called strike, and since the batting team's score was less than the fielding team's, the game finished with that called strike. How can we narrow this down further, to game-ending pitches of games that were still winnable for the batting team?

- For the currently losing batting team, this means that the number of base runners plus the batter must be greater than the difference in score between the winning and losing teams, to give the losing team a chance to win. To extend the game, the difference between the fielding team's score and the batting teams score with base runners and the batter could be equal to 0.

How many games have met this criteria?

```
df_23 %>%
  filter(description == "called_strike",
         outs_when_up == 2,
         strikes == 2,
         inning >= 9,
         bat_score < fld_score) %>%
  mutate(could_have_won = case_when(
    inning_topbot == "Top" & home_score > away_score ~ home_score -
      (away_score + 1 + !is.na(on_1b)+ !is.na(on_2b) + !is.na(on_3b)) <= 0,
    inning_topbot == "Bot" & home_score < away_score ~ away_score -
      (home_score + 1 + !is.na(on_1b)+ !is.na(on_2b) + !is.na(on_3b)) <= 0,
    TRUE ~ 0)) %>%
  count(could_have_won) %>% kable()
```

could_have_won	n
0	133
1	40

Out of the 173 instances, 40 were still winnable for the currently losing batting team. It is with the data from these 40 pitches that you must later determine which of these strikes might have been questionable. To accomplish this, sample from the Statcast pitch data.

```
library(broom)
library(mgcv)

#Save the data frame of the 40 game-ending pitches
game_enders <- df_23 %>%
  filter(description == "called_strike",
         outs_when_up == 2,
         strikes == 2,
         inning >= 9) %>%
  mutate(could_have_won = case_when(
```

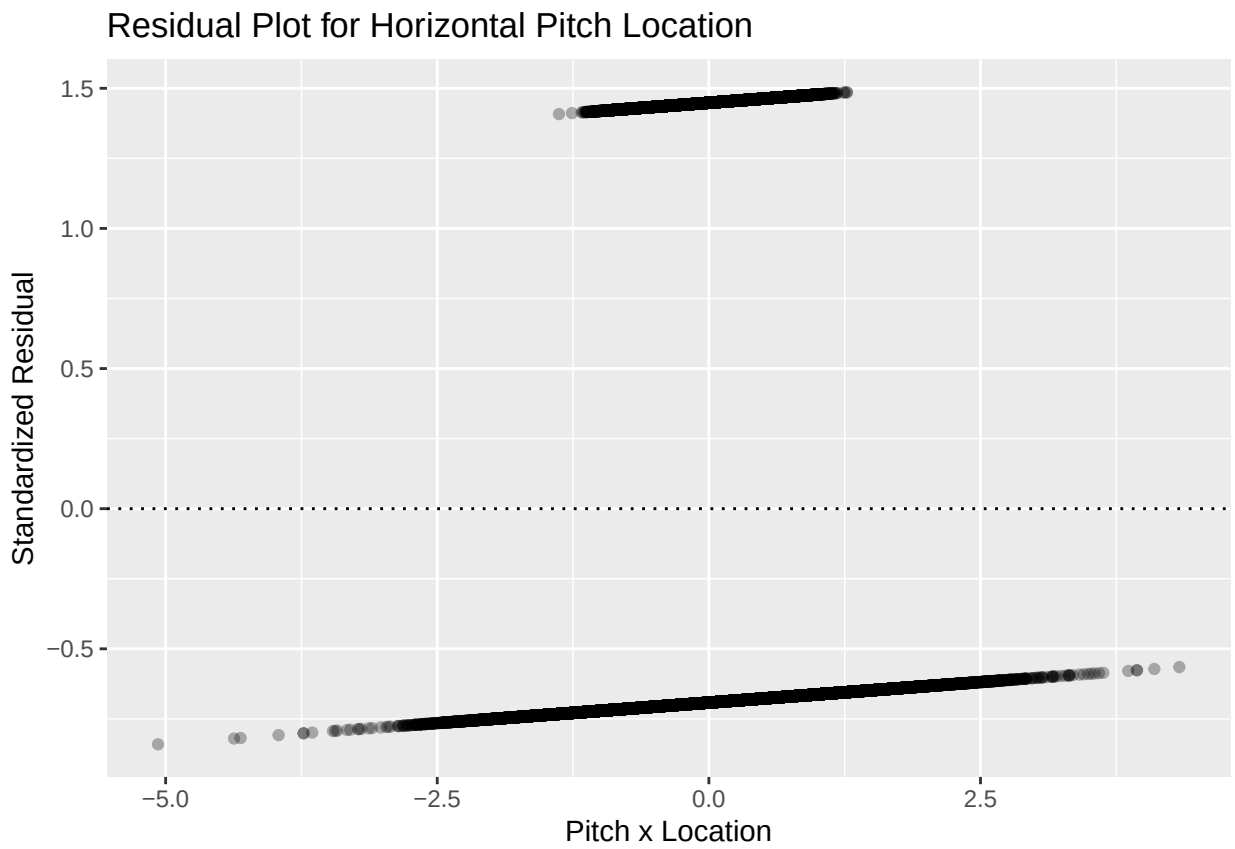
```
    inning_topbot == "Top" & home_score > away_score ~ home_score -
      (away_score + 1 + !is.na(on_1b)+ !is.na(on_2b) + !is.na(on_3b)) <= 0,
    inning_topbot == "Bot" & home_score < away_score ~ away_score -
      (home_score + 1 + !is.na(on_1b)+ !is.na(on_2b) + !is.na(on_3b)) <= 0,
    TRUE ~ 0)) %>%
  filter(could_have_won == TRUE)

#Sample pitches for model
pitches_taken <- df_23 %>%
  filter(plate_x != "NA" ,
         description %in% c("ball", "called_strike")) %>%
  sample_n(40000)
```

We want to associate pitch location with the likelihood of it being a strike. let's start with a linear model. Use a linear model to approximate the likelihood of a pitch being called a strike based on a pitch's horizontal location (pitch_x). Plot the residuals to see if this linear model fits the validity condition of "Linearity of Residuals" as required.

```
strike_lm <-
  lm(formula = description == "called_strike" ~ plate_x, data = pitches_taken)

strike_lm %>% tidy() %>% kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.323	0.002	138.145	0
plate_x	-0.014	0.002	-5.789	0

```
strike_lm_aug <- augment(strike_lm, data = pitches_taken)

ggplot(strike_lm_aug, aes(x= plate_x, y= .std.resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, linetype = 3) +
  labs(x = "Pitch x Location",
       y = "Standardized Residual",
       title = "Residual Plot for Horizontal Pitch Location")
```



Why is this model poorly represented by a linear model?

- In reality, the probability of a pitch being called a strike increases as the pitch approaches the center of the plate from either side. This can't be accurately represented by a linear model because the change in strike probability is not constant.

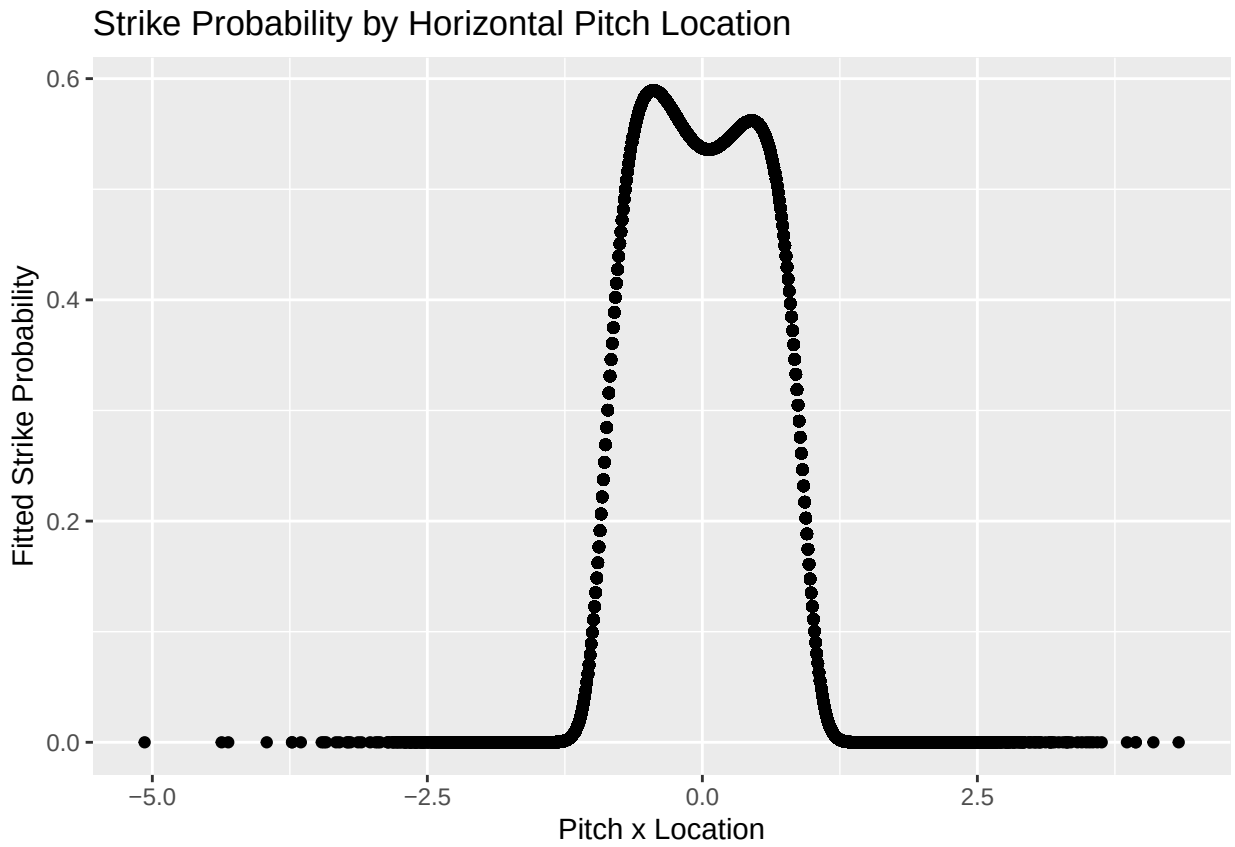
What types of models can better represent this relationship?

- GAMs!

Let's try a 1-dimensional GAM instead. The notation for our 1-d GAM is

$$g(E[Y]) = \beta_0 + f(plate\_x)$$

```
strike_mod_1d <- gam(description == "called_strike" ~ s(plate_x),
                     family = "binomial", data = pitches_taken)
strike_mod_1d_fitted <- strike_mod_1d %>%
  augment(type.predict = "response")

strike_mod_1d_fitted %>%
  ggplot(aes(x = plate_x, y = .fitted)) +
  geom_point() +
  labs(
    x = "Pitch x Location",
    y = "Fitted Strike Probability",
    title = "Strike Probability by Horizontal Pitch Location")
```



How can we improve on this model?

- Strike probability also depends on vertical location. We should add an additional explanatory variable to account for this.

Now let's try a 2-d GAM to further our analysis. The GAM can be represented as :

$$g(E[Y]) = \beta_0 + f(plate\_x, plate\_z)$$

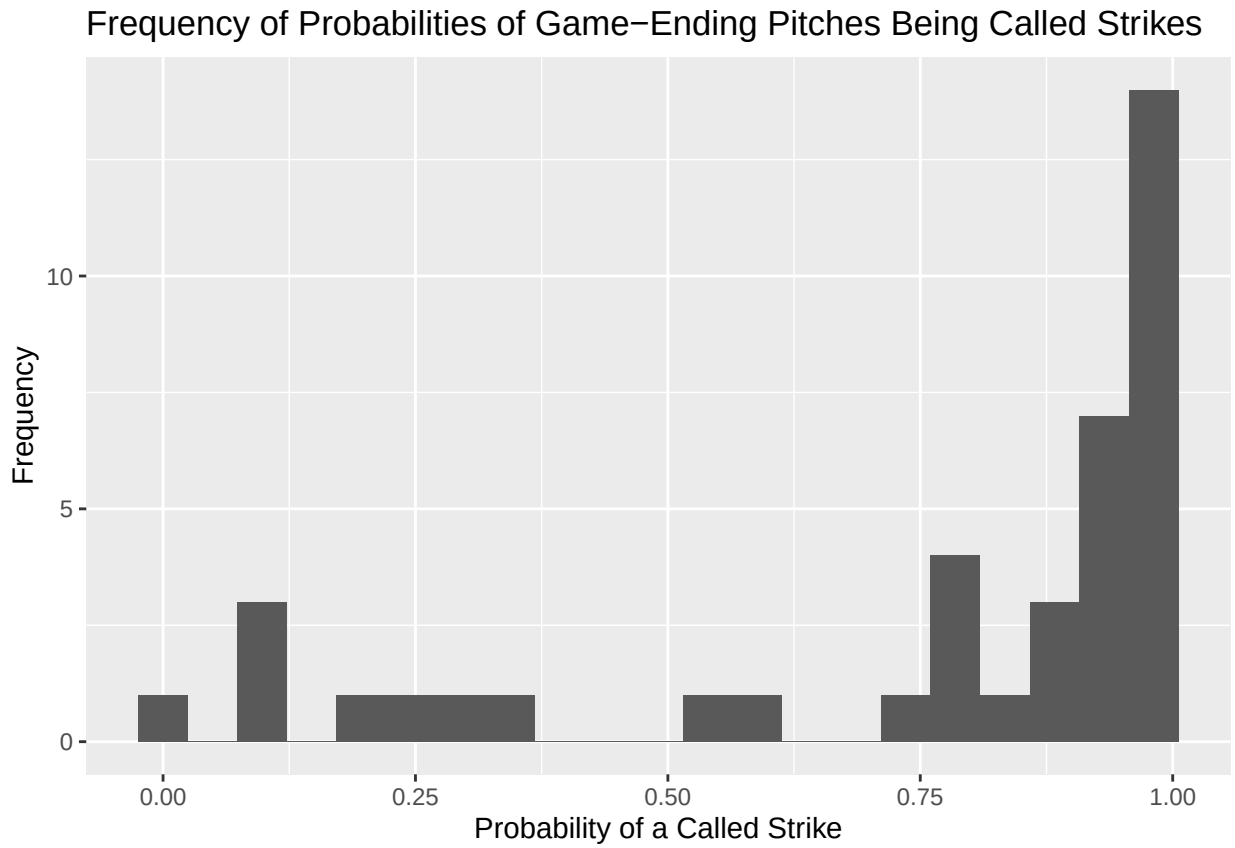
Define a GAM to explain the binomial response of whether the pitch was a called strike based on the x and z coordinates of the pitch as it passes over the plate.

```
#Create the model
strike_mod <- gam(description == "called_strike" ~ s(plate_x, plate_z),
                  family = "binomial", data = pitches_taken)
```

Once you have obtained this model for strike probability, fit this model to the data set of the 40 game-ending pitches, then visualize the probability frequencies with a histogram.

```
game_enders_fitted <- strike_mod %>%
  augment(type.predict = "response",
          newdata = game_enders)

game_enders_fitted %>%
  ggplot(aes(x = .fitted)) +
```

```
  geom_histogram(bins =21) +
  labs(x= "Probability of a Called Strike",
       y= "Frequency",
       title = "Frequency of Probabilities of Game-Ending Pitches Being Called Strikes")
```



What does the histogram mean?

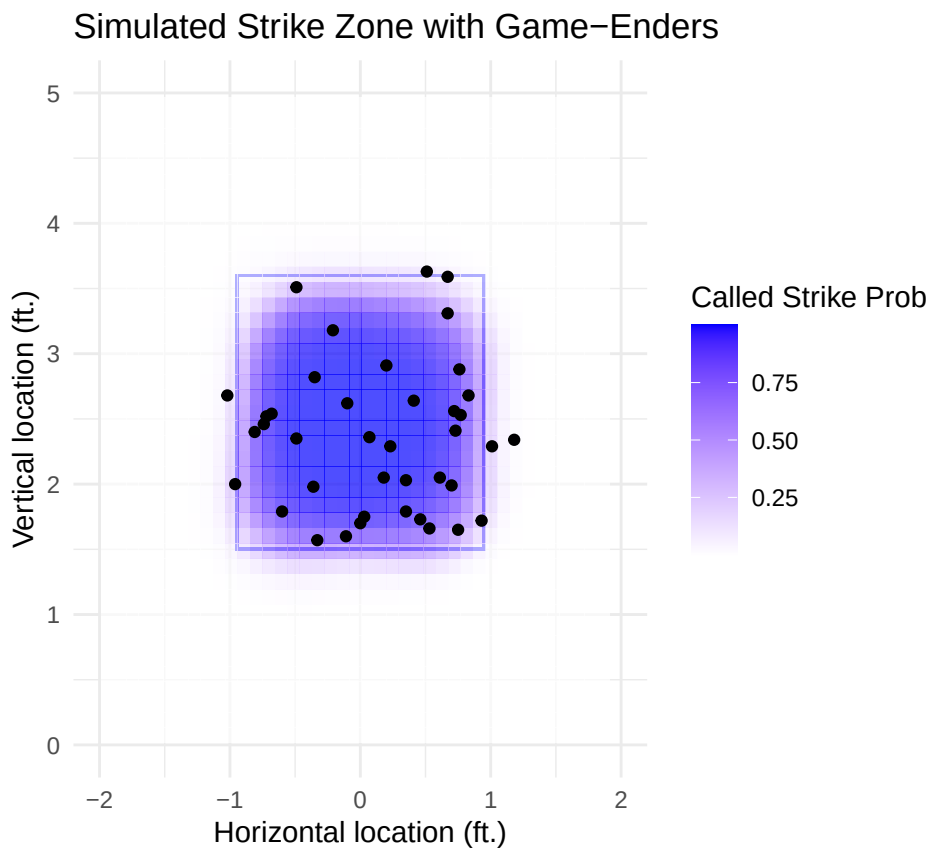
- This histogram shows that most of the game-ending pitches were almost certainly strikes, but there are a good few that are still questionable.

Visualize this on a simulated strike zone.

```
library(modelr)

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

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

Plot the results on the strike zone, demonstrating the change in probability density as you pass through it.

```r
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))

plot <- k_zone_plot %+%
  grid +
  geom_tile(aes(fill = .fitted), alpha = 0.7) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(fill = "Called Strike Prob",
       title = "Simulated Strike Zone with Game-Enders") +
  geom_point(data = game_enders) +
  theme_minimal()

# plotly::ggplotly(plot)
plot
```



Now that you have this visual, filter based on called strike probability of 25% or less. This means that according to your GAM, similarly placed pitches were called a strike at most a quarter of the time.

```
game_enders_fitted %>%
  filter(.fitted <= 0.25) %>%
  count() %>% kable()
```

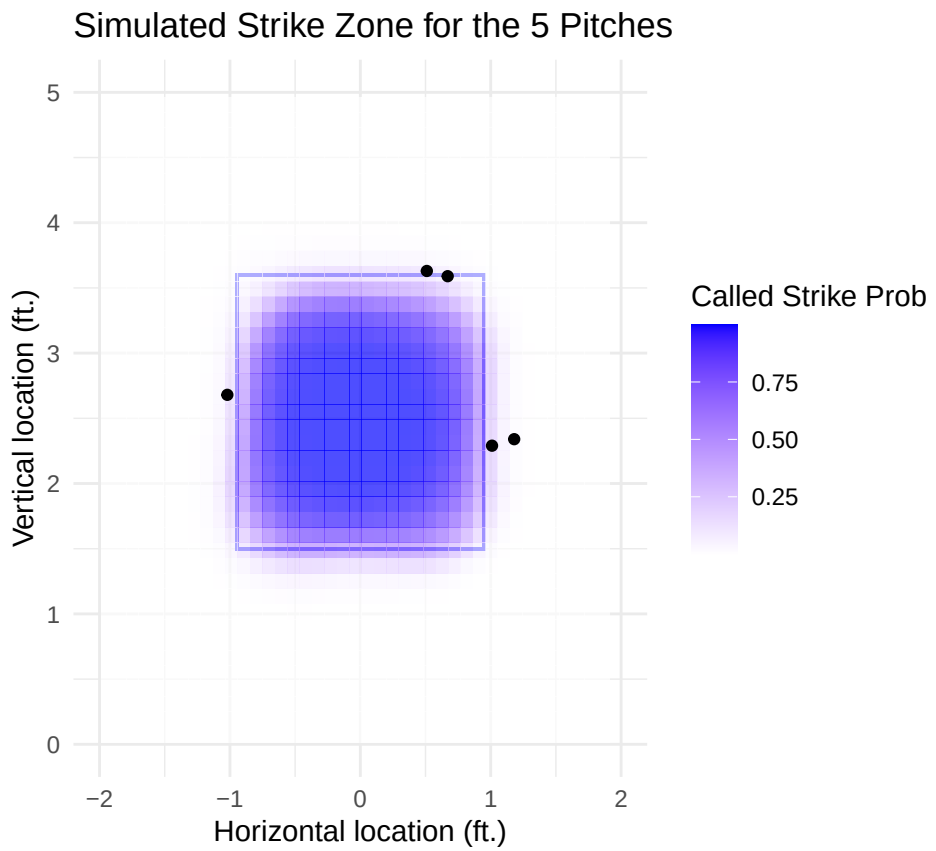
n
5

A total of 5 games have been decided on pitches that historically were at best 1 in 4. let's see what games these were, and what these pitches looked like.

```
Questionables <- game_enders_fitted %>%
  filter(.fitted <= 0.25)

Questionables%>%
  select(game_date, home_team, away_team, bat_score, fld_score) %>%
  kable()
```

game_date	home_team	away_team	bat_score	fld_score
5/13/2023	BAL	PIT	0	2
5/20/2023	STL	LAD	5	6
7/4/2023	LAD	PIT	7	9
8/4/2023	MIN	AZ	2	3
9/6/2023	PIT	MIL	4	5

```
q_plot <- k_zone_plot %+%
  grid +
  geom_tile(aes(fill = .fitted), alpha = 0.7) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(fill = "Called Strike Prob",
       title = "Simulated Strike Zone for the 5 Pitches") +
  geom_point(data = Questionables) +
  theme_minimal()
q_plot
```



It is safe to say, these are some tough calls.

6) Conclusions

GAMs These are a very useful tool in representing nonlinear relationships, especially as it pertains to binomial probability, however GAMs have far more expansive applications with other nonlinear density functions.

Should we switch to automated officiating? Out of just over 173 instances of a game ending on a called strike, 5 of those games can be classified as questionable and had a significant chance of the batter winning the game for their team. If umpires have a 97% accuracy in the game-ending called strikes, it is hard to make a case that it is absolutely necessary to switch to automated officiating. For much of America, it is the human error in baseball that makes baseball interesting. And for the MLB, any press might be good press, so people sharing videos of egregiously bad calls might be the publicity they are looking for.

Further Research In future research we would like to observe the antithesis of this module's subject, obvious strikes that were called balls. Compared to the game-ending consequence of a called strike, a questionable called ball gives the batter an additional chance to score, one that could change the game's outcome. Once this analysis is complete, we could try to use ANOVA to compare the variance in rates of incorrect calling within strikes and balls and then compare the variance between the rate of incorrect strike calling and the rate of incorrect ball calls.

Works Cited

Albert, Jim, et al. Analyzing Baseball Data with R. 3rd ed., CRC Press, 2021.

ChatGPT. Assistance given to the author, AI. Used to create some LaTeX code for the formulas for GAMs and some debugging. OpenAI, (https://chat.openai.com/share/7f8055b1-54ed-45c6-9c95-d70ac360747a). West Point, NY, 04MAY2024.

James, Gareth, et al. An Introduction to Statistical Learning: with Applications in R. 2nd ed., Springer, 2021.