Strike Zones

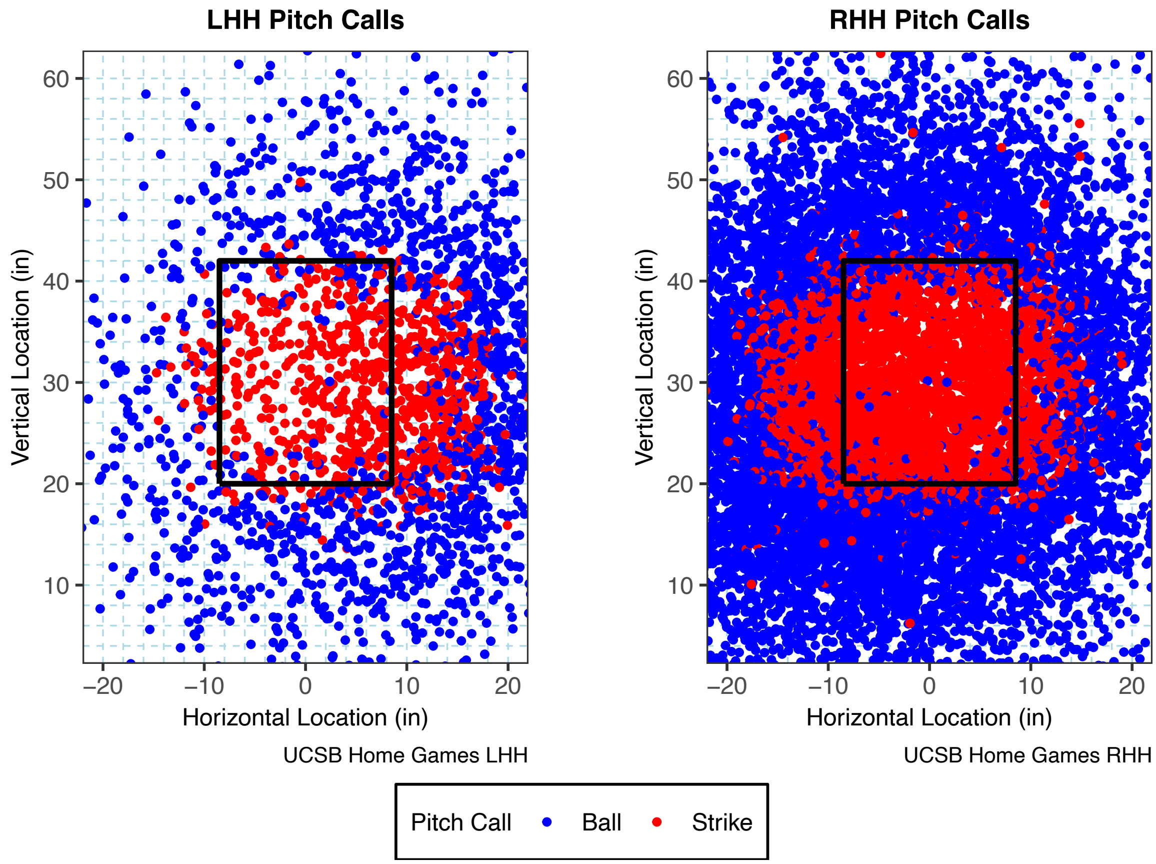
Taylor Villahermosa

9/29/2019

Objective: To create a strike zone of pitch locations of greater than 50% called strike probability.

We will use the data set Ultimate_Excel (6/17/19), which contains data from UCSB home games and scrimmages from 1/26/18 to 5/25/19.

Scatter Plots of Called Pitches



Based on this data, we can predict the called strike probability based on location. We will fit a generalized additive model to the data, predict called strike probability based on location, and plot the predicted called strike probability as a heat map.

A generalized additive model (GAM) on a logit transformation scale gives the equation:

$$E(Strike\ Called_i) = f_1(PLS_i, PLH_i)$$

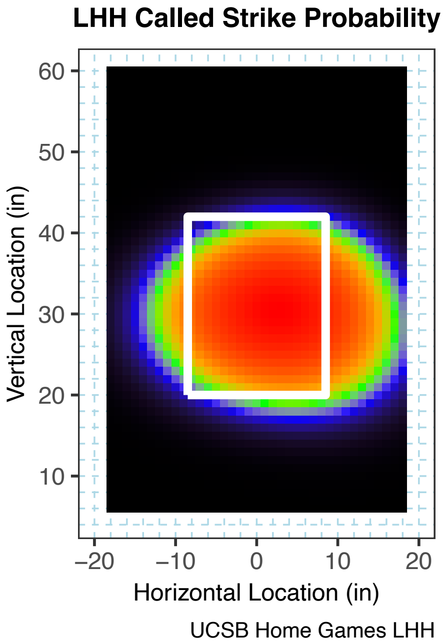
f_1 is a smooth function and $E(Strike\ Called_i)$ is the expected strike called value.

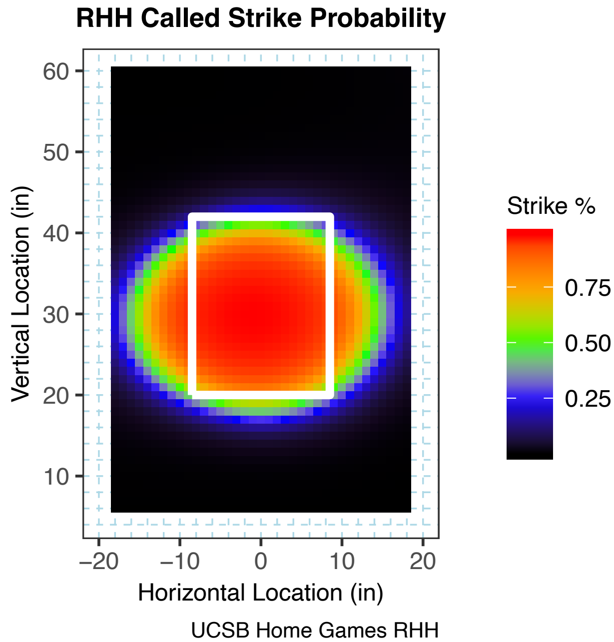
Using this model, we predict the strike called value for every square inch over the plate.

To get value on scale of probability (percentage), transform with the inverse logit function:

$$Strike\ Probability = \frac{e^x}{1+e^x} \text{ where } x = Strike\ Called$$

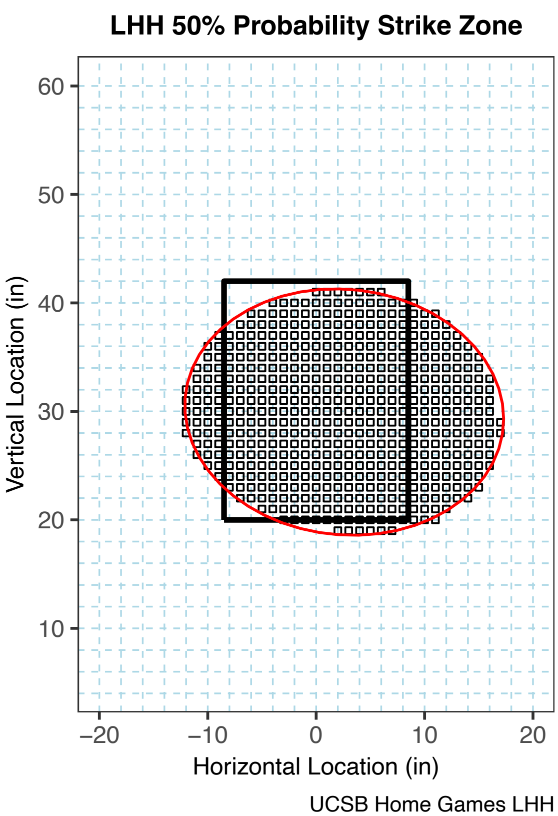
Heat Maps of Called Strike Probability by Location

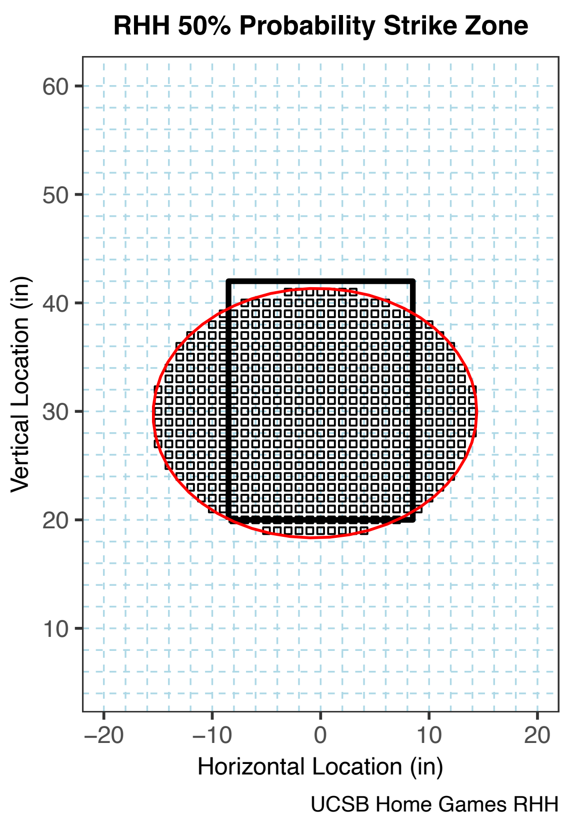




Locations Where Called Strike Probability is Greater than 50%.

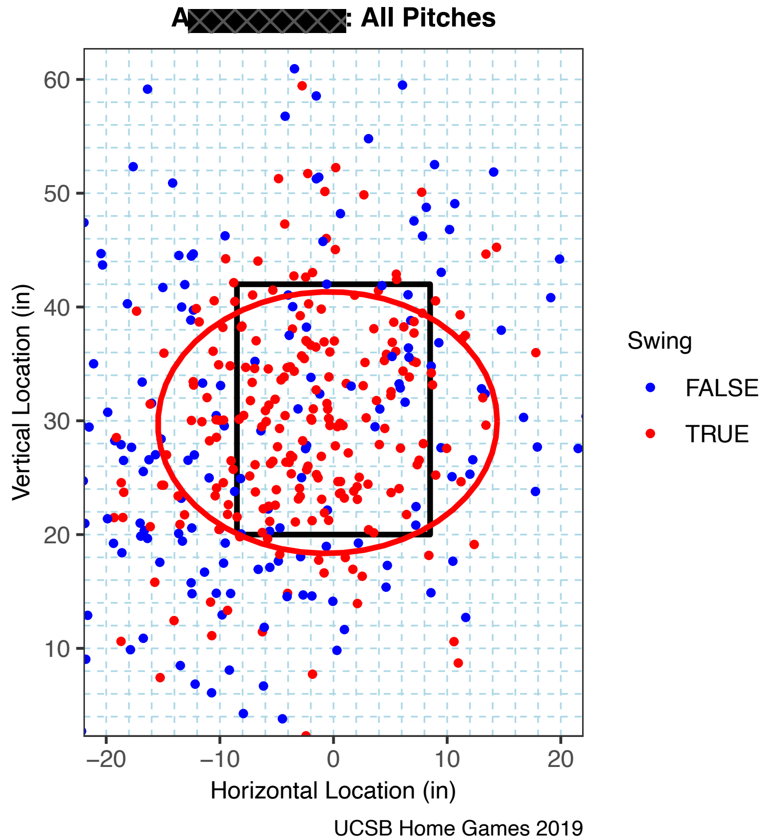
Plotting locations where called strike probability is greater than 50% and fitting an ellipse around those points.





Now we can use these ellipses as the 50% called strike probability strike zones.

For each player, we can use the new strike zone ellipses to calculate plate discipline statistics such as chase rate. Here is an example using A[redacted] from the 2019 season:



Using the 50% called strike probability ellipses, we can calculate the following plate discipline statistics for each player.

Table 1: Plate Discipline Statistics

Stat	T[redacted]	A[redacted]	[redacted]	E[redacted]
O Swing	19%	34%	23%	16%
Z Swing	63%	76%	69%	60%
Swing	42%	54%	46%	39%
O Contact	13%	20%	13%	10%
Z Contact	53%	61%	54%	55%
Contact	81%	73%	72%	85%
Zone	51%	48%	51%	51%
Whiff	8%	15%	13%	6%

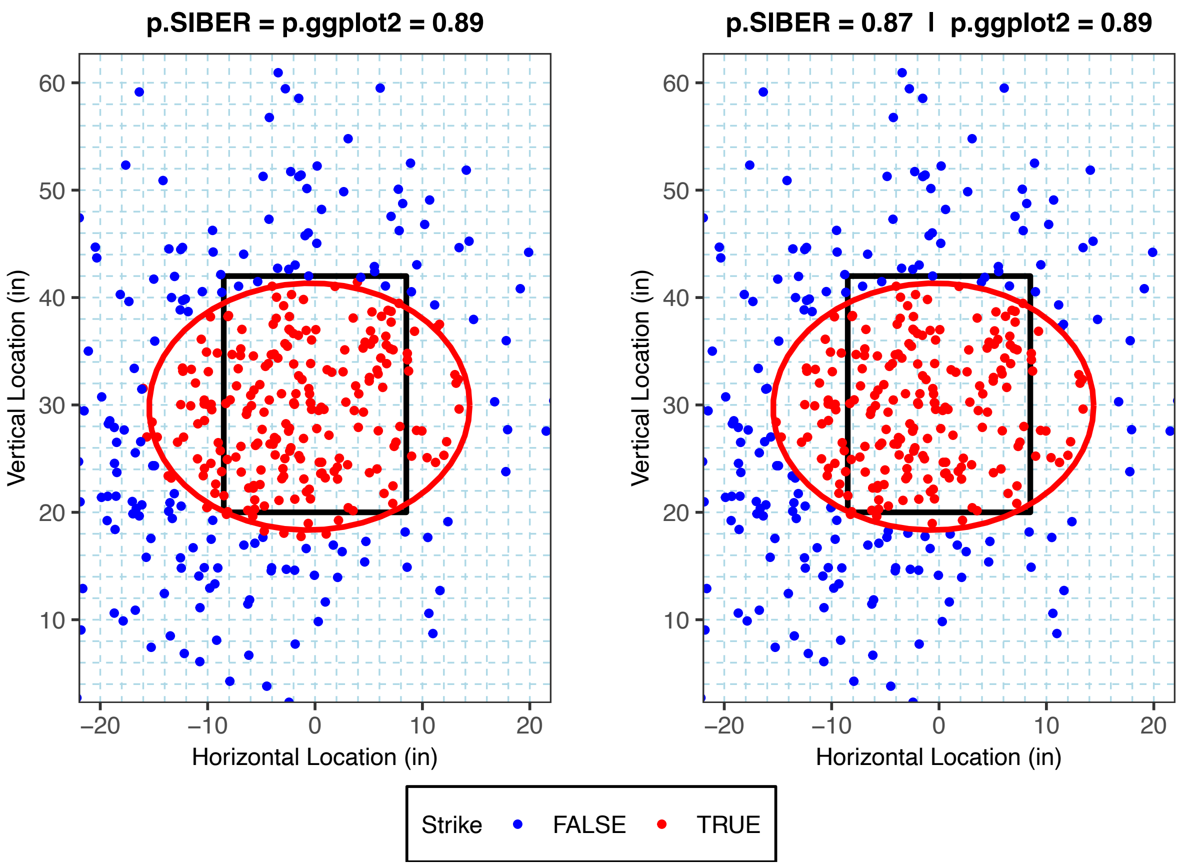
Notes on Calculating Strike Zone Ellipses

When creating ellipses around the data, you have to specify level = p $\in (0, 1)$. We are using both packages ggplot2 and SIBER to create the ellipses. ggplot2 draws the ellipse on the graph, and SIBER calculates if a pitch is inside or outside of the strikezone ellipse.

However, ggplot2 and SIBER ellipse calculations are slightly different. For the strikezone ellipses calculated from Ultimate_Excel (6/17/19) with greater than 50% called strike probability, the optimal ellipses are created with ggplot2 level = 0.89 and SIBER level = 0.87.

To make strike zone ellipses using different ultimates and/or different called strike probabilities, you have to cross check the ggplot2 ellipse with the SIBER calculations and adjust the SIBER level accordingly.

The plots below show the slight differences between ggplot2 and SIBER levels.



The plot on the left has strikes (red) that are outside of the strikezone. After adjusting the SIBER level, the right plot has only strikes within the strikezone and balls outside of it.

Appendix

```
library(dplyr)
library(mgcv)
library(SIBER)
library(ggplot2)
library(ggpubr)
library(knitr)
```

```
# ggplot formatting: adds gridlines and x and y axis labels
theme.formatting <- theme_bw() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 10),
        plot.subtitle = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.text = element_text(size = 9),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 9),
        panel.grid.major = element_line(colour="lightblue", size=0.3, linetype = "dashed"),
        panel.grid.minor = element_line(colour="lightblue", size=0.3, linetype = "dashed"))

scales <- list(scale_x_continuous(breaks = seq(-20,20,10),minor_breaks = seq(-20 , 20, 2)),
  scale_y_continuous(breaks = seq(10,60,10),minor_breaks = seq(0, 60, 2)),
  coord_fixed(xlim=c(-20,20),ylim=c(5,60)))

labels <- labs(x = "Horizontal Location (in)", y = "Vertical Location (in)")
```

```
# subsetting ultimate by called strikes and balls
called.pitches <- read.csv("/Users/taylorvillahermosa/Desktop/Ultimate_Excel.csv") %>%
  mutate(PLS = 12*PlateLocSide, PLH = 12*PlateLocHeight,
         strike.called = ifelse(PitchCall == "StrikeCalled", TRUE, FALSE)) %>%
  filter(PLH > 0, PLS < 35, PLS > -35, !is.na(PLS),
         PitchCall %in% c("StrikeCalled", "BallCalled"))

# building theoretical strike zone (the rectangle)
strike.zone.df <- data.frame(x = c(-8.5, -8.5, 8.5, 8.5, -8.5),
                             y = c(20, 42, 42, 20, 20))
```

```
# scatter plot code
LHH.scatter <- ggplot(called.pitches %>% filter(BatterSide == "Left"))+
  theme.formatting + scales + labels +
  labs(title = "LHH Pitch Calls", caption = "UCSB Home Games LHH", color="Pitch Call") +
  geom_point(aes(PLS,PLH,color = PitchCall), size = 1)+
  scale_color_manual(values=c("blue","red"), labels = c("Ball", "Strike"))+
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black") +
  theme(legend.background = element_rect(linetype="solid", colour ="black"))

RHH.scatter <- ggplot(called.pitches %>% filter(BatterSide == "Right"))+
  theme.formatting + scales + labels +
  labs(title = "RHH Pitch Calls", caption = "UCSB Home Games RHH", color="Pitch Call") +
  geom_point(aes(PLS,PLH,color = PitchCall), size = 1)+
  scale_color_manual(values=c("blue","red"), labels = c("Ball", "Strike"))+
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black")
```

```r
ggarrange(LHH.scatter, RHH.scatter, ncol = 2, common.legend = T, legend = "bottom")
```

```r
#  50% strike probability
k.prob <- .5

# percentile of the ellipse to fit around the strike zone
p.ggplot2 <- .89

#  (horz, vert) location coordinates for every square inch
x <- seq(-18, 18, length.out=37)
z <- seq(6, 60, length.out=55)
location.coordinates <- data.frame(PLS = c(outer(x, z * 0 + 1)),
                                   PLH = c(outer(x * 0 + 1, z)))
```

```r
# Right Handed Hitters: Predicting Called Strike Probability
RHH.strike.zone.data <- called.pitches %>% filter(BatterSide == "Right")
RHH.strike.model.fit <- gam(strike.called ~ s(PLS, PLH,k=10),
                            family = binomial,
                            data = RHH.strike.zone.data)

RHH.strike.predict.data <- location.coordinates
RHH.strike.model.predict <- predict(RHH.strike.model.fit, RHH.strike.predict.data)
RHH.strike.predict.data <- RHH.strike.predict.data %>%
  mutate(strike.prob = exp(RHH.strike.model.predict) / (1 + exp(RHH.strike.model.predict)))
```

```r
# Left Handed Hitters: Predicting Called Strike Probability
LHH.strike.zone.data <- called.pitches %>% filter(BatterSide == "Left")
LHH.strike.model.fit <- gam(strike.called ~ s(PLS, PLH, k=10),
                            family = binomial,
                            data = LHH.strike.zone.data)

LHH.strike.predict.data <- location.coordinates
LHH.strike.model.predict <- predict(LHH.strike.model.fit, LHH.strike.predict.data)
LHH.strike.predict.data <- LHH.strike.predict.data %>%
  mutate(strike.prob = exp(LHH.strike.model.predict) / (1 + exp(LHH.strike.model.predict)))
```

```r
# heatmap code
LHH.heatmap <- ggplot(LHH.strike.predict.data) +
  geom_tile(aes(x = PLS, y = PLH, fill = strike.prob)) +
  scale_fill_gradientn(colours=c("black","blue","green","orange","red"))+
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1.5, color = "white") +
  theme.formatting + scales + labels +
  labs(title = "LHH Called Strike Probability",
       caption = "UCSB Home Games LHH", fill = "Strike %") +
  theme(plot.margin=unit(c(0,1,0,0),"cm"), legend.box.margin=margin(20,20,20,20))

RHH.heatmap <- ggplot(RHH.strike.predict.data) +
  geom_tile(aes(x = PLS, y = PLH, fill = strike.prob)) +
  scale_fill_gradientn(colours=c("black","blue","green","orange","red")) +
  theme.formatting + scales + labels +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1.5, color = "white") +
  labs(title = "RHH Called Strike Probability",
       caption = "UCSB Home Games RHH", fill = "Strike %") +
  theme(plot.margin=unit(c(0,0,0,1),"cm"))
```

```r
ggarrange(LHH.heatmap, RHH.heatmap, ncol = 2, common.legend = T, legend = "right")
```

```r
# plotting > 50% called strike probability locations and fitting ellipses around the data
LHH.50 <- ggplot(LHH.strike.predict.data %>% filter(strike.prob >= k.prob)) +
  theme.formatting + scales + labels +
  theme(plot.margin=unit(c(0,0,0,1),"cm")) +
  labs(title = "LHH 50% Probability Strike Zone", caption = "UCSB Home Games LHH") +
  geom_point(aes(PLS,PLH), size = 1, pch = 22) +
  scale_color_manual(values=c("blue","red")) +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black") +
  stat_ellipse(aes(PLS,PLH), level = p.ggplot2, color = "red")

RHH.50 <- ggplot(RHH.strike.predict.data %>% filter(strike.prob >= k.prob))+
  theme.formatting + scales + labels +
  theme(plot.margin=unit(c(0,0,0,1),"cm"))+
  labs(title = "RHH 50% Probability Strike Zone", caption = "UCSB Home Games RHH") +
  geom_point(aes(PLS,PLH), size = 1, pch = 22) +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black") +
  stat_ellipse(aes(PLS,PLH),level = p.ggplot2, color = "red")

ggarrange(LHH.50, RHH.50, ncol=2)
```

```r
#  data frame of RHH 50% strike probability locations
RHH.X <- RHH.strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLS)
RHH.Y <- RHH.strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLH)
RHH.XY <- na.omit(data.frame(RHH.X,RHH.Y))

#  RHH ellipse calculations
mu.RHH <- colMeans(RHH.XY) # center of the ellipse
Sigma.RHH <- cov(RHH.XY) # covariance matrix of the ellipse

#  data frame of LHH 50% strike probability locations
LHH.X <- LHH.strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLS)
LHH.Y <- LHH.strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLH)
LHH.XY <- na.omit(data.frame(LHH.X,LHH.Y))

#  LHH ellipse calculations
mu.LHH <- colMeans(LHH.XY) # center of the ellipse
Sigma.LHH <- cov(LHH.XY) # covariance matrix of the ellipse
```

```r
# example using [redacted]
master <- read.csv("/Users/taylorvillahermosa/Desktop/Ultimate_Excel.csv") %>%
  mutate(Swing = ifelse(PitchCall %in% c("StrikeCalled", "BallCalled"), FALSE, TRUE),
         Date = as.Date(Date,"%m/%d/%y")) %>%
  filter(Date > "2019-02-15")

player.data <- master %>% filter(Batter == "[redacted]")

if(player.data$BatterSide[1] == "Right"){
  strike.predict.data <- RHH.strike.predict.data
} else {
  strike.predict.data <- LHH.strike.predict.data}

k.prob <- .5
```

```
X <- strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLS)
Y <- strike.predict.data %>% filter(strike.prob >= k.prob) %>% select(PLH)
XY <- na.omit(data.frame(X,Y))

all.pitches <- data.frame(player.data$PLS,player.data$PLH)

# calculate ellipse
mu <- colMeans(XY) # center of the ellipse
Sigma <- cov(XY) # covariance matrix of the ellipse

# percentile of the ellipse
p.SIBER <- .87

# StrikeZone = TRUE/FALSE if pitch is inside/outside of strikezone ellipse
player.data <- player.data %>%
  mutate(StrikeZone = ellipseInOut(pointsToEllipsoid(all.pitches, Sigma, mu), p = p.SIBER))

# Armani all pitches plot
ggplot(RHH.strike.predict.data %>% filter(strike.prob >= .5))+
  theme.formatting + scales + labels +
  labs(title = "[redacted]: All Pitches",
       caption = "UCSB Home Games 2019", color="Swing") +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black")+
  geom_point(data=player.data %>% filter(!is.na(StrikeZone)),
             aes(x = PLS,y = PLH, col = Swing), size = 1) +
  scale_color_manual(values = c("blue","red")) +
  stat_ellipse(aes(PLS,PLH), level=p.ggplot2, color = "red", size = 1)
```

```
# plate discipline statistics
# uses functions from "Plate Discipline.R"
source("/Users/taylorvillahermosa/Desktop/Swing Efficiency/Plate Discipline.R")
plate.stats <- data.frame("Plate Discipline Stats" = c("O Swing", "Z Swing",
                                                       "Swing", "O Contact",
                                                       "Z Contact", "Contact",
                                                       "Zone", "Whiff"),
                          "[redacted]" = swing.summary(master %>%
                                                     filter(Batter == "[redacted]")),
                          "[redacted]" = swing.summary(master %>%
                                                     filter(Batter == "[redacted]")),
                          "[redacted]" = swing.summary(master %>%
                                                  filter(Batter == "[redacted]")),
                          "[redacted]" = swing.summary(master %>%
                                                  filter(Batter == "[redacted]")))
names(plate.stats) <- c("Stat", "[redacted]", "[redacted]", "[redacted]", "[redacted]")
kable(plate.stats, caption = "Plate Discipline Statistics", format = "pandoc")
```

```
# calculating TRUE/FALSE using p = 0.87 and = 0.89
player.data <- player.data %>%
  mutate(StrikeZone.87 = ellipseInOut(pointsToEllipsoid(all.pitches, Sigma, mu), p = p.SIBER),
         StrikeZone.89 = ellipseInOut(pointsToEllipsoid(all.pitches, Sigma, mu), p = p.ggplot2))

# plot with p = 0.89
same.p <- ggplot(RHH.strike.predict.data %>% filter(strike.prob >= .5))+
  theme.formatting + scales + labels +
```

```r
  labs(title = "p.SIBER = p.ggplot2 = 0.89",color="Strike") +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black")+
  geom_point(data=player.data %>% filter(!is.na(StrikeZone.89)),
             aes(x = PLS,y = PLH, col = StrikeZone.89), size = 1) +
  scale_color_manual(values = c("blue","red")) +
  stat_ellipse(aes(PLS,PLH), level=p.ggplot2, color = "red", size = 1) +
  theme(legend.background = element_rect(linetype="solid", colour ="black"))

# plot with 0.87
adjusted.p <- ggplot(RHH.strike.predict.data %>% filter(strike.prob >= .5))+
  theme.formatting + scales + labels +
  labs(title = "p.SIBER = 0.87  |  p.ggplot2 = 0.89",color="Strike") +
  geom_path(data = strike.zone.df, aes(x, y), lwd = 1, color = "black")+
  geom_point(data=player.data %>% filter(!is.na(StrikeZone.87)),
             aes(x = PLS,y = PLH, col = StrikeZone.87), size = 1) +
  scale_color_manual(values = c("blue","red")) +
  stat_ellipse(aes(PLS,PLH), level = p.ggplot2, color = "red", size = 1)

ggarrange(same.p, adjusted.p, ncol=2, common.legend = T, legend = "bottom")
```