Strikezone

Michael Cambaliza

December 12, 2020

Load the data

```
master <- read_xlsx("~/Documents/R/BASEBALL/Ultimate.xlsx")
```

Setting up relevant databases

Filter the data to called pitches only with relevant variables

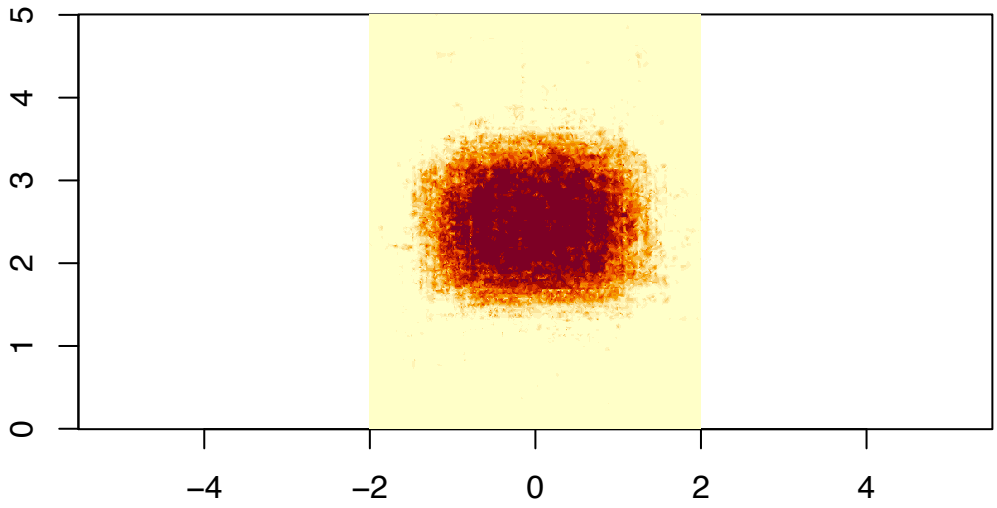
Train the model

Use randomForest to estimate prediction

```
#Create RandomForest. This code takes a while to run.
zonetrees <- randomForest(Call~V+H, data=called)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
called <- called %>%
  mutate(Probability = round(predict(zonetrees), 3))

zoneimg <- interp(x=called$H, y=called$V, z=called$Probability,
            xo=seq(-2, 2, by=0.01),
            yo=seq(0, 5, by=0.01), duplicate="mean")
image(zoneimg, xaxs="i", asp=1)
```

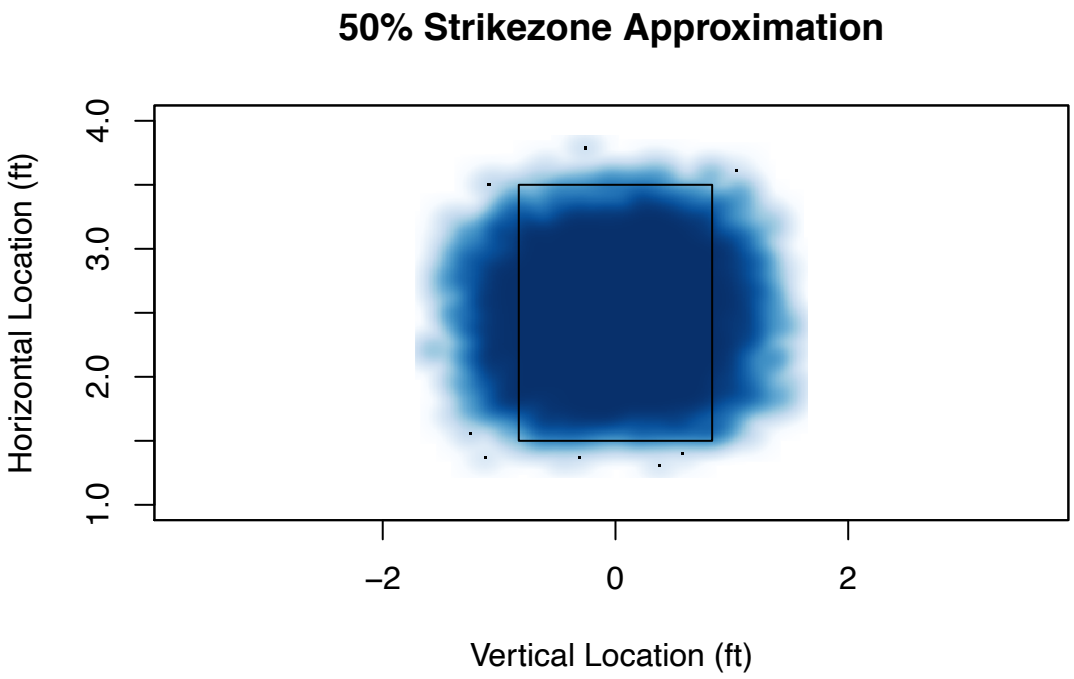



```
Strikes <- called %>%
  filter(Probability > 0.75)

pitches <- master %>%
  filter(BatterSide != 'Undefined', is.na(PlateLocHeight) == FALSE) %>%
  select(BatterSide, PitchCall, PlateLocHeight, PlateLocSide) %>%
  mutate(PlateLocHeight = round(PlateLocHeight, 2), PlateLocSide = round(PlateLocSide, 2), BatterSide
           ifelse(BatterSide == 'Right', 'R', 'L'))
names(pitches) <- c("Side","Call","V","H")

pitches <- pitches %>%
  filter(V < 3.5, V > 1.5, H > -0.83, H < 0.83)

pitches <- pitches %>%
  mutate(Probability = predict(zonetrees, newdata = pitches)) %>%
  filter(Probability > 0.75)
```

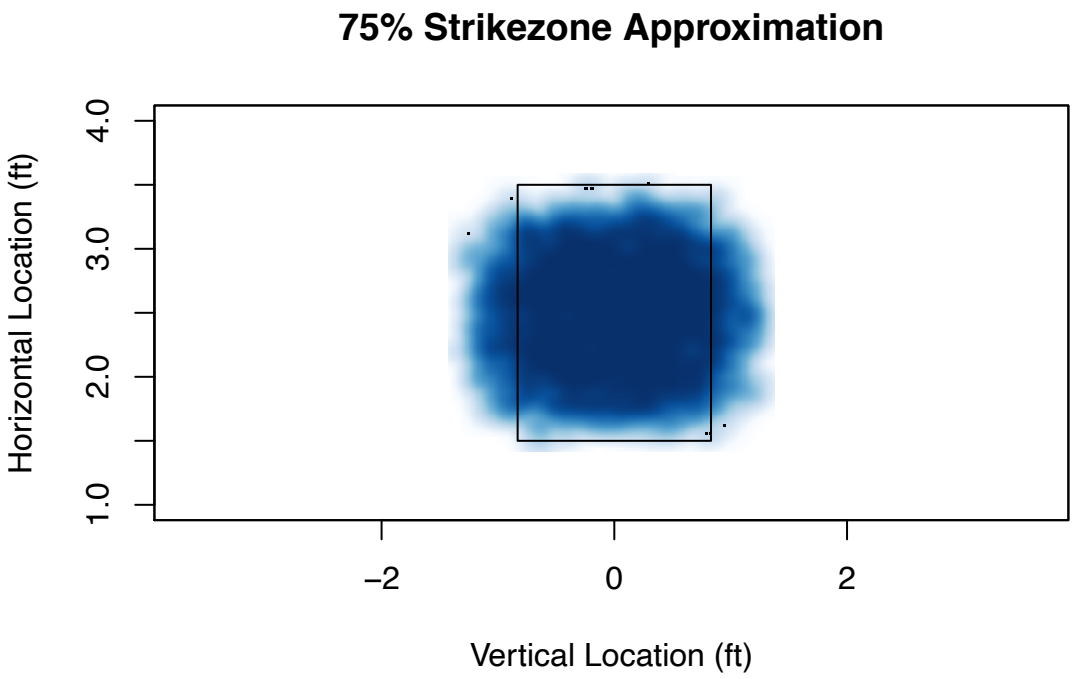
Use the model to predict 75% strikezone

```
xcoor <- seq(-2, 2, by=0.01)
ycoor <- seq(0, 5, by=0.01)

xn <- length(xcoor)
yn <- length(ycoor)

k <- 0
```

```
n = xn * yn
coors <- data.frame(V = numeric(n), H = numeric(n))

for(i in ycoor){
  coors$V[(k*xn+1):((k+1)*xn)] <- i
  k = k+1
}

k <- 1

for(i in xcoor){
  tmp <- seq(k, n, xn)
  coors$H[tmp] <- i
  k = k+1
}

temp <- coors

rm(k, tmp, yn, xn, xcoor, ycoor, n, i)


xr <- 9.97/12
xl <- -xr
yt <- 3.5
yb <- 1.5

coors <- coors %>%
  mutate(Probability = round(predict(zonetrees, newdata = coors), 3)) %>%
  filter(Probability >= 0.5)

smoothScatter(coors$H, coors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "50% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```

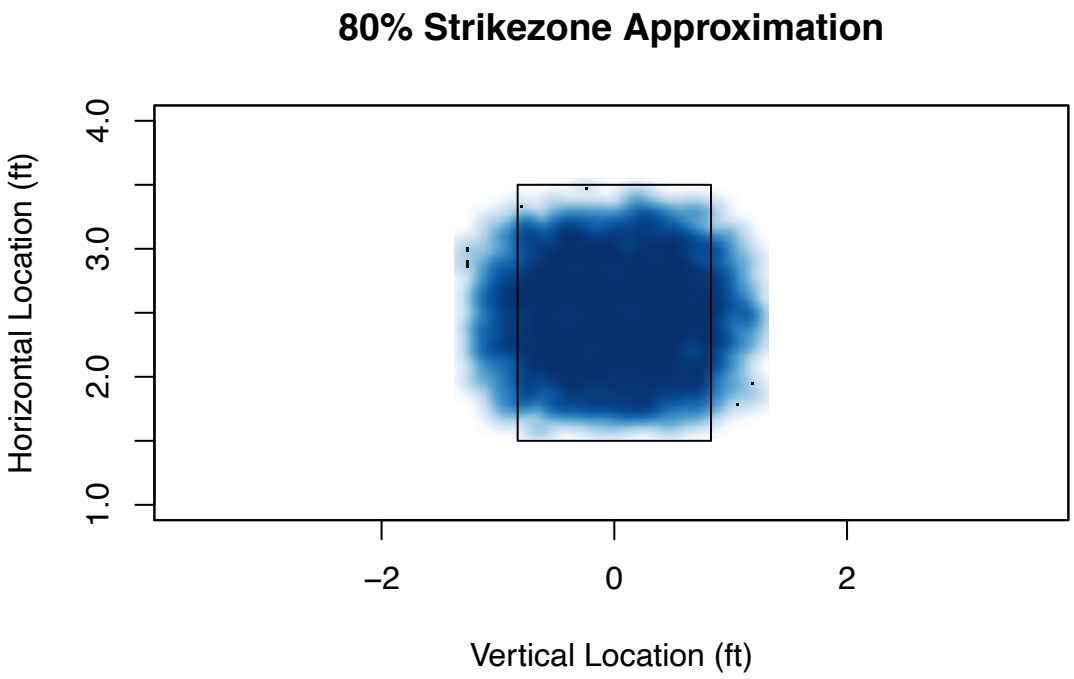



```
coors <- coors %>%
  filter(Probability >= 0.75)

smoothScatter(coors$H, coors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "75% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```

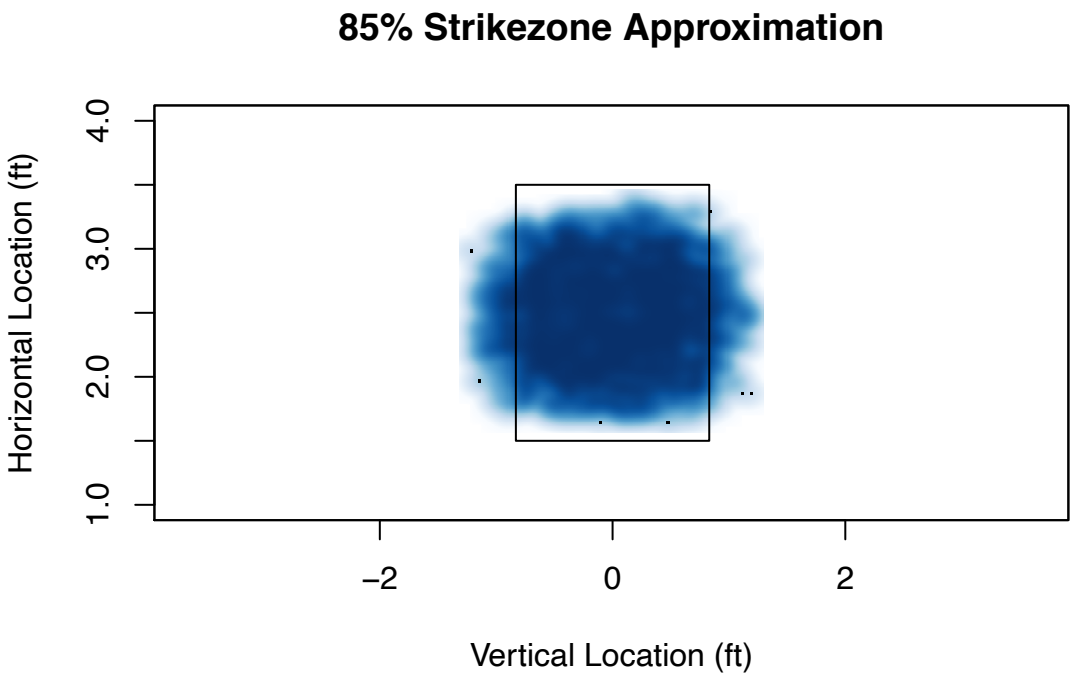


Predict different strike zones with different percentages

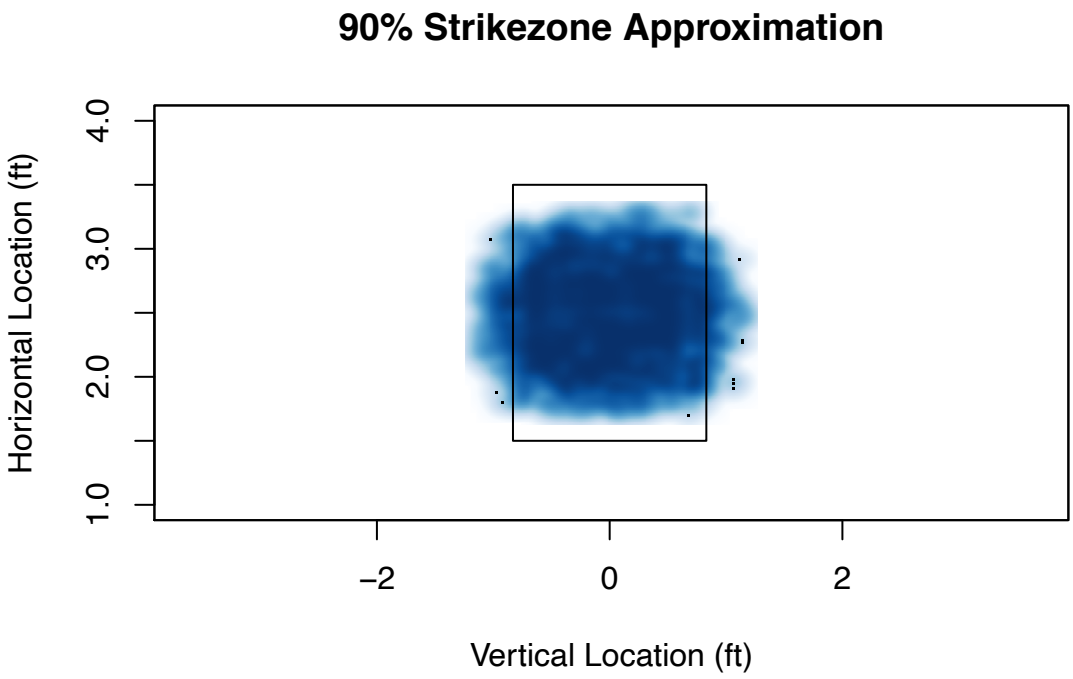
```
coors <- coors %>%
  filter(Probability >= 0.8)

smoothScatter(coors$H, coors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "80% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```




```
coors <- coors %>%
  filter(Probability >= 0.85)

smoothScatter(coors$H, coors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "85% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```




```
coors <- coors %>%
  filter(Probability >= 0.9)

smoothScatter(coors$H, coors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "90% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```



Separate the strikezone by sides

```
Rcall <- called %>%
  filter(Side == "R")
rtrees <- randomForest(Call~V+H, data=Rcall)
```

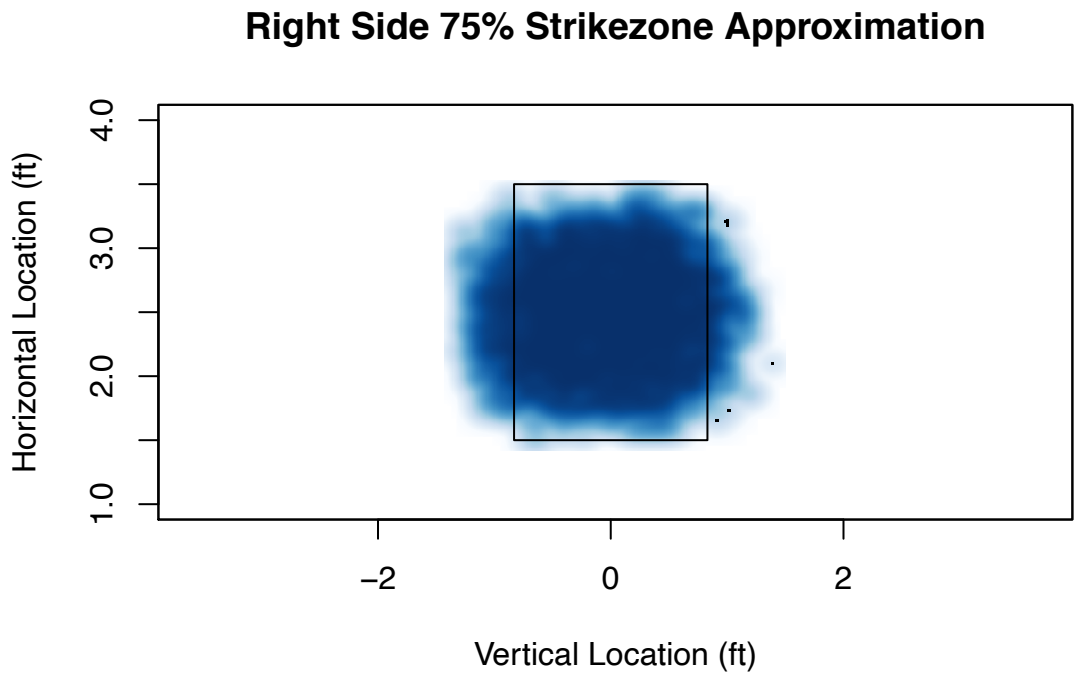
```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
rcoors <- temp

rcoors <- rcoors %>%
  mutate(Probability = round(predict(rtrees, newdata = rcoors), 3)) %>%
  filter(Probability >= 0.75)

Lcall <- called %>%
  filter(Side == "L")
ltrees <- randomForest(Call~V+H, data=Lcall)
```

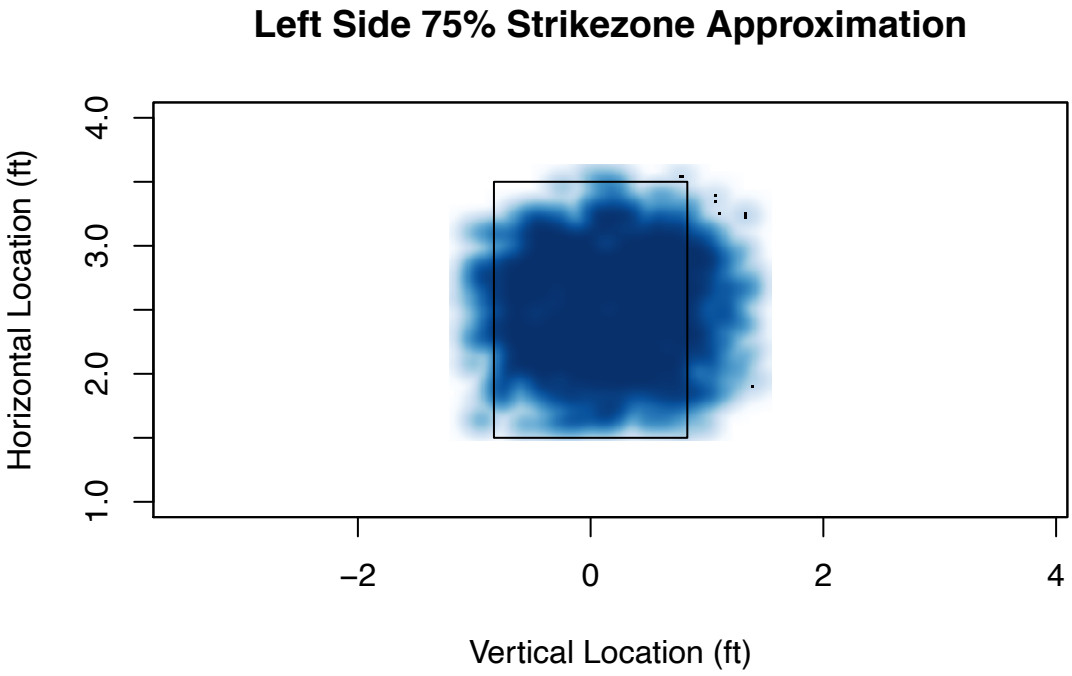
```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
lcoors <- temp

lcoors <- lcoors %>%
```

```
  mutate(Probability = round(predict(ltrees, newdata = lcoors), 3)) %>%
  filter(Probability >= 0.75)

smoothScatter(rcoors$H, rcoors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "Right Side 75% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```




```
smoothScatter(lcoors$H, lcoors$V, asp = 1.1, ylim = c(1 ,4), nrpoints = 10, nbin = 300,
              main = "Left Side 75% Strikezone Approximation", xlab = "Vertical Location (ft)",
              ylab = "Horizontal Location (ft)")
rect(xl, yb, xr, yt)
```


Left Side 75% Strikezone Approximation
Horizontal Location (ft)
4.0
3.0
2.0
1.0
−2
0
2
4
Vertical Location (ft)