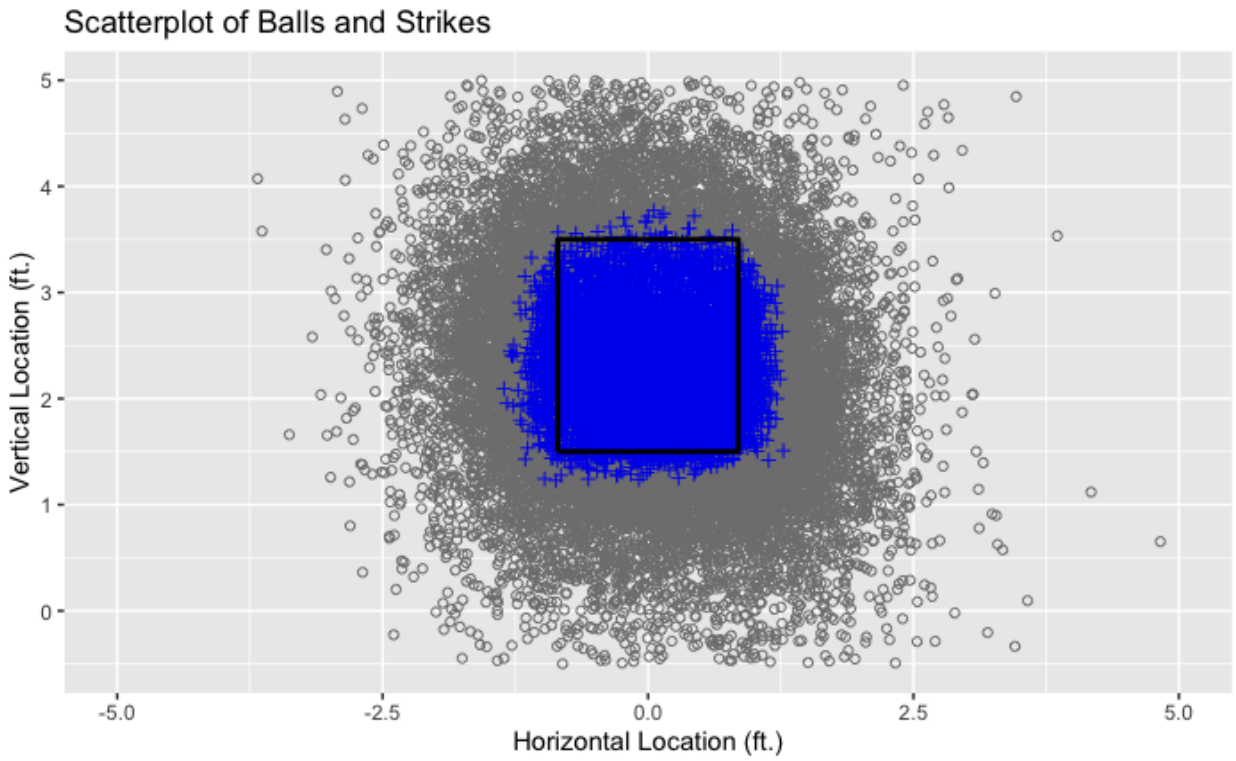
Shamir Naran

Baseball Report

There is a common belief among baseball fans that an umpire is more likely to call a “fringe pitch” as a strike if there are already 3 balls on the batter. In this report, we assess whether or not the count (number of strikes and number of balls) impacts the probability of a strike being called if all pitch conditions are comparable.

First let’s view a plot of our raw data.

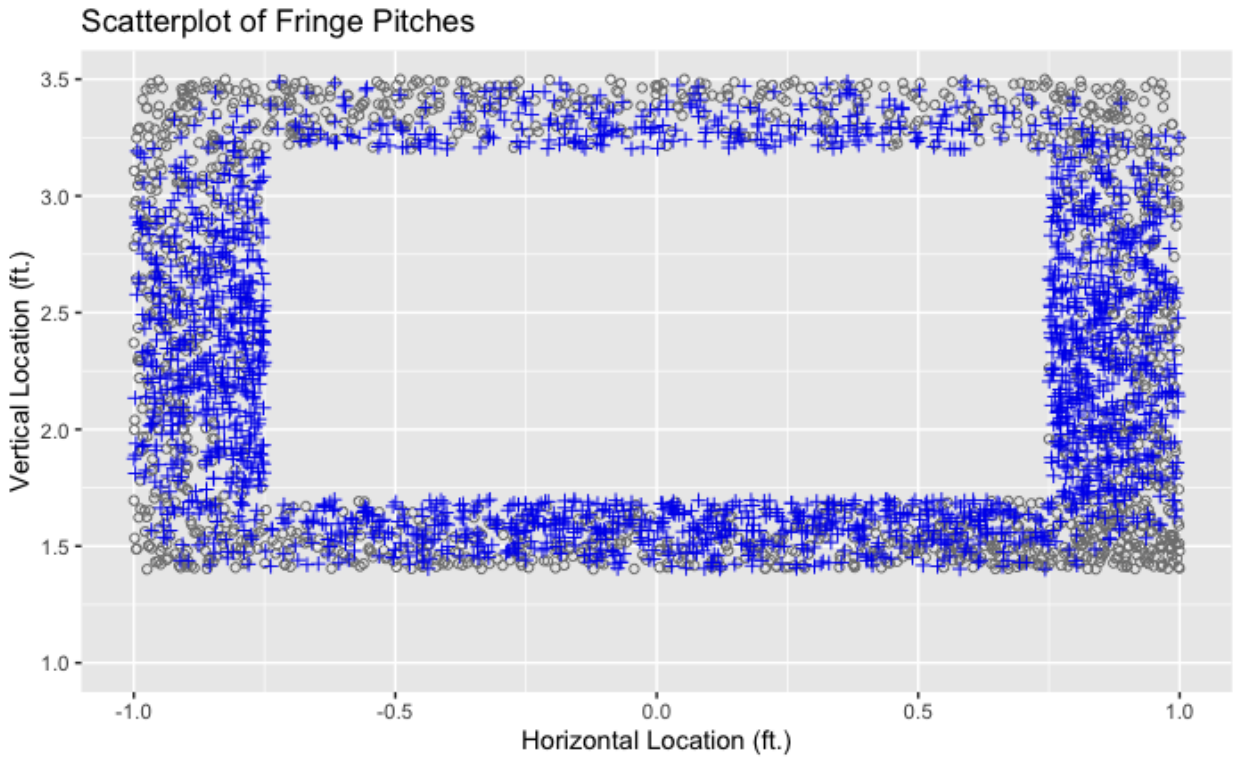


The blue “+” represent strikes and the grey “o” represent balls. In order to do the analysis, we only want to look at “fringe pitches”. These are pictures that are on the borderline where the umpire actually has to make a decision.

For our purposes, fringe pitches were identified as pitches:

(-1 < Horizontal Location < -0.75) & (1.4 < Vertical Location < 3.5)
(0.75 < Horizontal Location < 1) & (1.4 < Vertical Location < 3.5)
(-1 < Horizontal Location < 1) & (1.4 < Vertical Location < 1.7)
(-1 < Horizontal Location < 1) & (3.2 < Vertical Location < 3.5)

This decision is subjective. The determination was used to get a good mix of balls and strikes within our fringe pitch data.



Above is a scatterplot of the fringe pitches. 43.3% of the fringe pitches are balls and 56.7% of the fringe pitches are called strikes.

We added a new variable called “count”. This variable was split up into 3 categories:

1) Less than 3 balls
2) 3 balls and less than 2 strikes
3) 3 balls and 2 strikes (full count)

Note: We split the data into 3 balls and less than 3 balls while also acknowledging that there is an important distinction whether the count is 3-0 or 3-1, and 3-2. For this reason, there are 3 categories.

Two new variables are created. Distance (ft). from the center of the strike zone and run differential which calculates how close the game is.

Logistic Regression was used to model our data. From this model we found the following significant predictors.

	95% Confidence Intervals for our Significant Predictors		
	Odds Ratio	**Lower Bound**	**Upper Bound**
Release Speed	Decrease	5.56%	0.66%
Balls	Increase	52.38%	52.38%
Strikes	Decrease	5.62%	32.08%
Distance	Decrease	764.87%	650.57%

According to this model, we're 95% confident that on average, holding all else constant ...

a one unit increase in release speed will decrease the odds of a called strike between [5.56% and 0.663%].

a one unit increase in strikes will decrease the odds of a called strike between [52.38% and 52.38%].

a one unit increase in balls will increase the odds of a called strike between [5.62% ,32.08%].

a one unit increase in distance will decrease the odds of a called strike between [764.87% and 650.57%].

Making Predictions

Let's see how our model is predicting: (See R code for detail)

For this exercise, we look at a 94.3 mph four-seam fastball (FF) in a 1 run game with distance from the center of the zone at 1 ft. These are held constant.

Case 1: In a 3-2 count, the probability that this pitch is a strike is 55.80%.

Case2 : In a 3-1 count, the probability that this pitch is a strike is 56.37%.

Case 3: In a 0-0 count, the probability that this pitch is a strike is 62.28%.

Conclusions

Distance was the most significant predictor when determining a called strike or ball. Our intuition would lead us to believe that this correct since a ball and strike should solely be dependent on location.

What we're really concerned about is the cases. None of the cases were significant predictors in our model. There was indication that there will be an increase in probability of a called strike in a full count or a less than 3 ball count as compared to a 3 ball and less than 2 strike count. This probability is captured in our example above where the highest probability of a strike came in a 0-0 count.

While the cases weren't significant, the number of balls and strikes did end up being significant predictors. This leads us to believe that the number of balls and strikes does impact the probability of a strike being called but maybe not for these specific cases. It would be worthy to explore this theory further with a larger set of data.

Math 536 Final Exam: Appendix R Code

Shamir Naran

Due May 15, 2020

Problem 1

```
library(ggplot2)
library(esquisse)
```

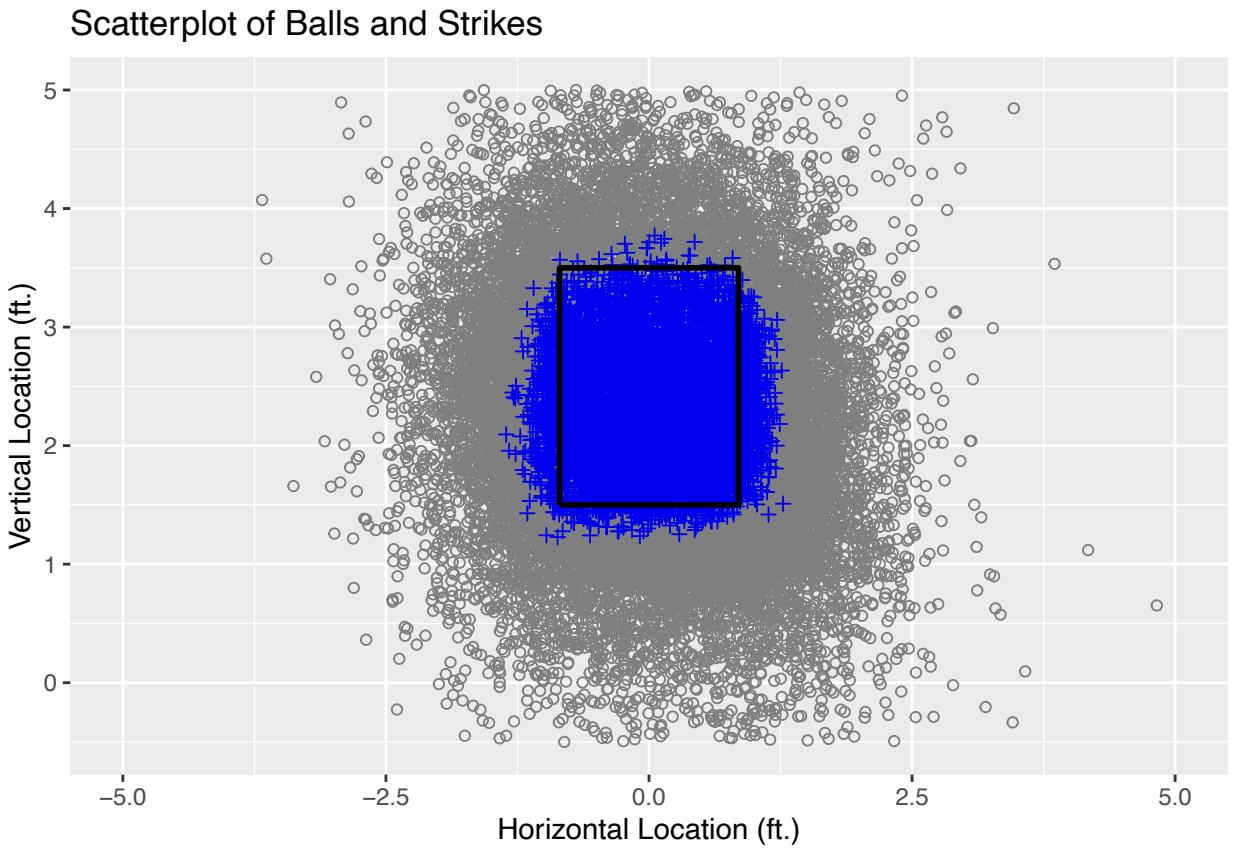
```
# Read in the data

baseball <- read.csv("baseball.csv", header = TRUE)
```

```
# Data Cleaning

baseball <- na.omit(baseball) # remove all rows containing an NA
baseball <- baseball[-c(1,4)] # we don't need "X" or the "pitcher" column
```

```
# change called ball to 0 and called strike to 1

baseball$description <- ifelse(baseball$description == "ball", 0, 1)
```

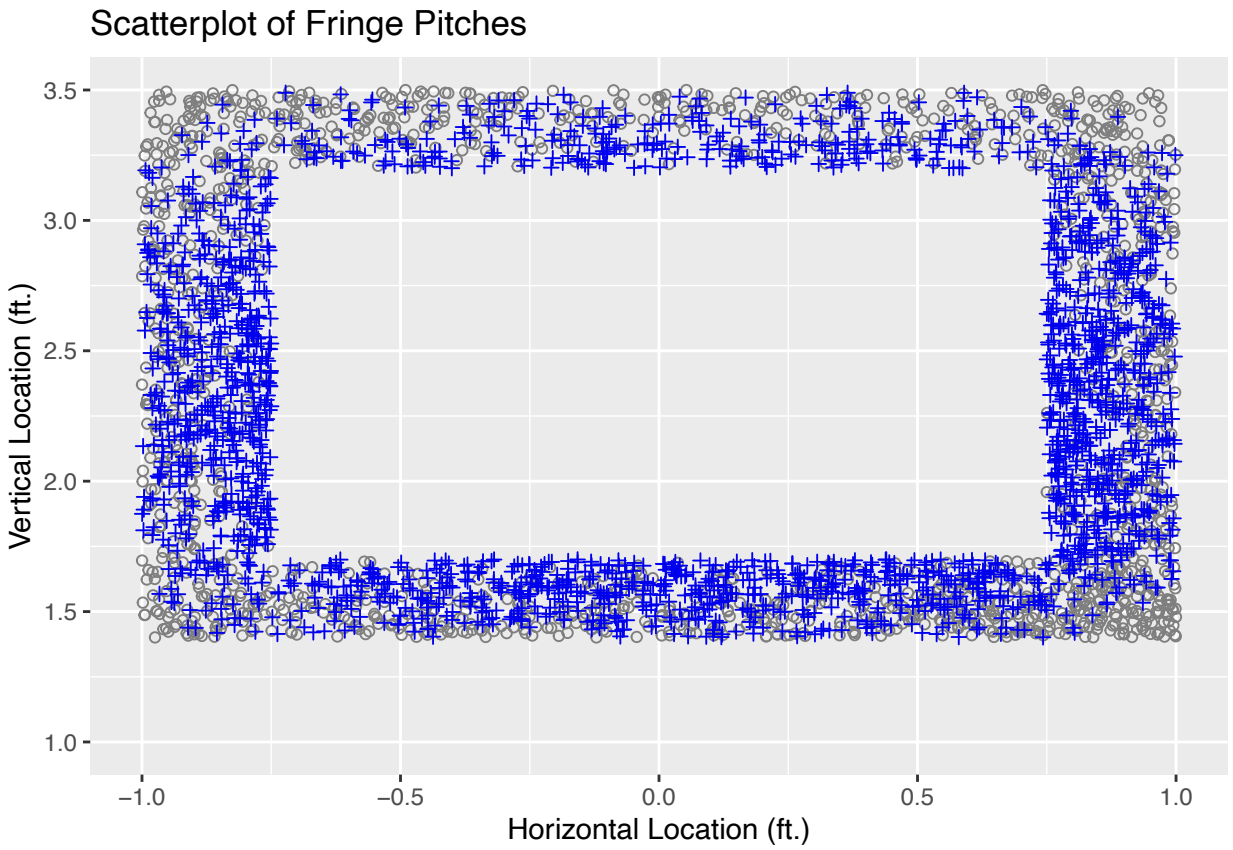
```
# Define Strike Zone (from the internet)

TopStrikeZone <- 3.5
BotStrikeZone <- 1.5
LeftStrikeZone <- -0.85
RightStrikeZone <- 0.85
StrikeZone <- data.frame(
  x=c(LeftStrikeZone, LeftStrikeZone, RightStrikeZone, RightStrikeZone, LeftStrikeZone),
  y=c(BotStrikeZone, TopStrikeZone, TopStrikeZone, BotStrikeZone, BotStrikeZone))
```

```
ggplot() +
  geom_point(data = data.frame(baseball$plate_x[baseball$description == 0],baseball$plate_z[baseball$des
             mapping = aes(baseball$plate_x[baseball$description == 0],baseball$plate_z[baseball$descri
             col = "gray50", shape = 1) +
  geom_point(data = data.frame(baseball$plate_x[baseball$description == 1],baseball$plate_z[baseball$des
             mapping = aes(baseball$plate_x[baseball$description == 1],baseball$plate_z[baseball$descri
             col = "blue2", shape = 3) +
  geom_path(aes(x,y), data = StrikeZone, lwd = 1,col = "black") +
  xlim(-5, 5) + ylim(-0.5, 5) +
  labs(title = "Scatterplot of Balls and Strikes", x = "Horizontal Location (ft.)", y = "Vertical Locati
```




```
# Identify "Fringe Pitches" (add and subtract 0.05 ft. from the strike zone)

baseball = baseball[!(baseball$plate_x > -0.75 & baseball$plate_x < 0.75 & baseball$plate_z > 1.7 & base
baseball = baseball[(baseball$plate_x > -1 & baseball$plate_x < 1 & baseball$plate_z > 1.4 & baseball$pl
```

```
ggplot() +
  geom_point(data = data.frame(baseball$plate_x[baseball$description == 0],baseball$plate_z[baseball$des
             mapping = aes(baseball$plate_x[baseball$description == 0],baseball$plate_z[baseball$descrip
             col = "gray50", shape = 1) +
  geom_point(data = data.frame(baseball$plate_x[baseball$description == 1],baseball$plate_z[baseball$des
             mapping = aes(baseball$plate_x[baseball$description == 1],baseball$plate_z[baseball$descrip
             col = "blue2", shape = 3) +
  xlim(-1, 1) + ylim(1, 3.5) +
  labs(title = "Scatterplot of Fringe Pitches", x = "Horizontal Location (ft.)", y = "Vertical Location
```




```
# Good mix of balls and strikes

length(which(baseball$description == 0)) # number of balls in data
```

```
## [1] 1663
```

```
length(which(baseball$description == 1)) # number of strikes in data
```

```
## [1] 2157
```

```
length(which(baseball$description == 0)) / length(baseball$description) # % of balls
```

```
## [1] 0.4353403
```

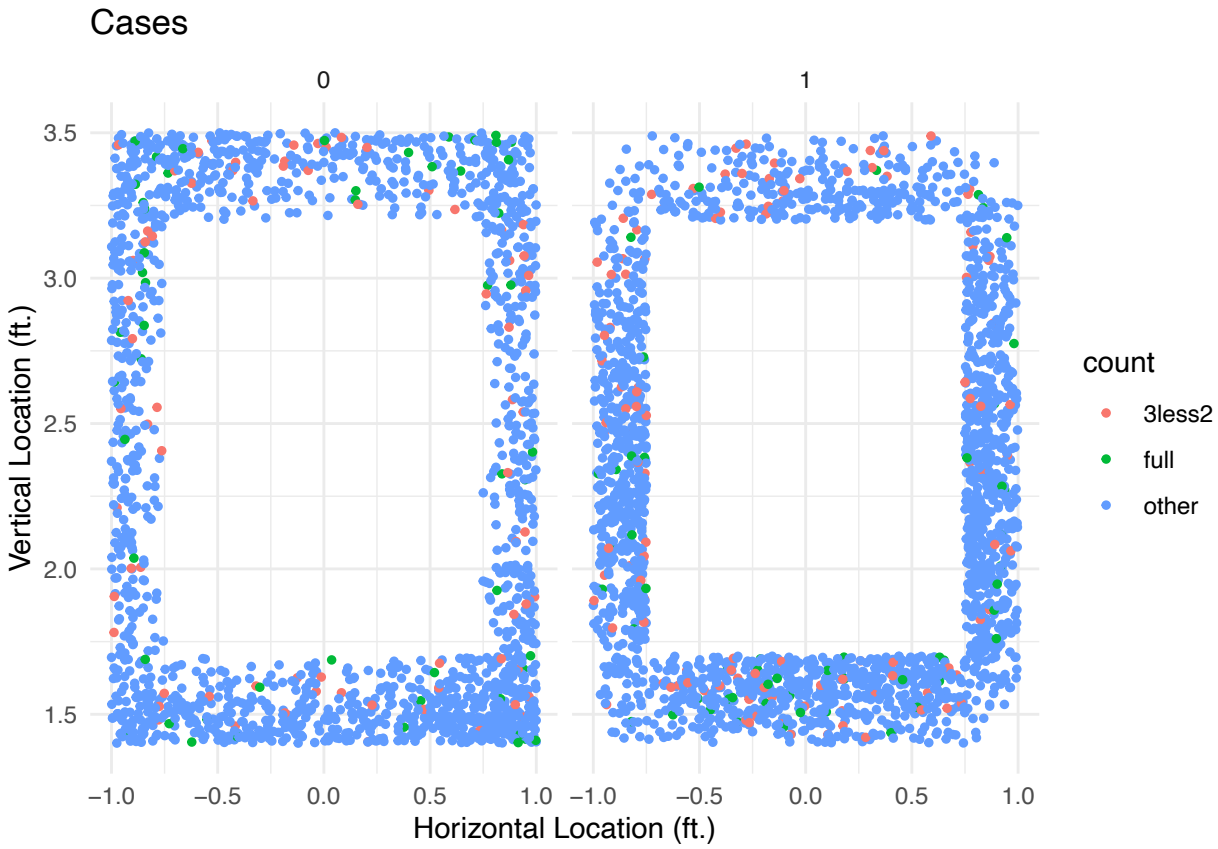
```
length(which(baseball$description == 1)) / length(baseball$description) # % of strikes
```

```
## [1] 0.5646597
```

```
# Create new Variables (count, run differential, distance)

baseball[ , "count"] = ""
baseball[ , "run_diff"] = ""
baseball[ , "distance"] = ""
```

```
baseball[baseball$ball == 3 & baseball$strikes < 2,]$count = "3less2"
baseball[baseball$ball == 3 & baseball$strikes == 2,]$count = "full"
baseball[baseball$ball != 3,]$count = "other"

baseball$run_diff <- abs(baseball$home_score - baseball$away_score)

baseball$distance <- sqrt((baseball$plate_x)^2+(baseball$plate_z-2.4)^2)
```

```
ggplot(baseball) +
  aes(x = plate_x, y = plate_z, colour = count) +
  geom_point(size = 1L) +
  scale_color_hue() +
  theme_minimal() +
  facet_wrap(vars(description)) +
  labs(title = "Cases", x = "Horizontal Location (ft.)", y = "Vertical Location (ft.)")
```

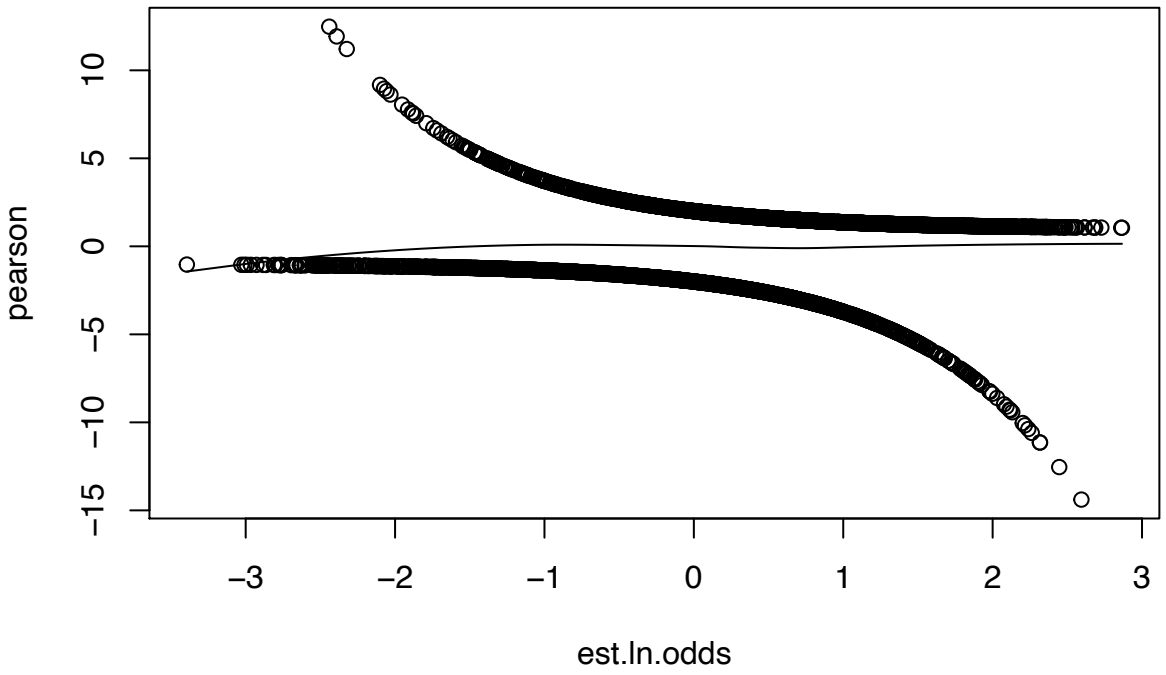



```
# Fit a Logistic Regression Model

model <- glm(description ~ pitch_type + release_speed + balls + strikes + count + run_diff + distance,
             data = baseball, family = "binomial")
sum.model <- summary(model)
sum.model
```

```
##
## Call:
```

```
## glm(formula = description ~ pitch_type + release_speed + balls +
##     strikes + count + run_diff + distance, family = "binomial",
##     data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3093  -0.9982   0.5491   0.9102   2.2468
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    9.833456   1.103900   8.908  < 2e-16 ***
## pitch_typeCU  -0.365294   0.208499  -1.752  0.07977 .
## pitch_typeEP  -1.538646   1.264262  -1.217  0.22359
## pitch_typeFC  -0.125130   0.233945  -0.535  0.59274
## pitch_typeFF   0.351420   0.199234   1.764  0.07776 .
## pitch_typeFS  -0.360922   0.368433  -0.980  0.32728
## pitch_typeFT   0.313989   0.208046   1.509  0.13124
## pitch_typeKC  -0.339682   0.272739  -1.245  0.21297
## pitch_typeSI   0.191732   0.222977   0.860  0.38986
## pitch_typeSL   0.148307   0.175955   0.843  0.39930
## release_speed -0.031125   0.012498  -2.490  0.01276 *
## balls          0.166491   0.057038   2.919  0.00351 **
## strikes       -0.411401   0.057383  -7.169 7.53e-13 ***
## countfull      0.388255   0.285324   1.361  0.17359
## countother     0.333469   0.218401   1.527  0.12680
## run_diff      -0.004542   0.017419  -0.261  0.79428
## distance      -7.077194   0.291581 -24.272  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5231.6  on 3819  degrees of freedom
## Residual deviance: 4371.8  on 3803  degrees of freedom
## AIC: 4405.8
##
## Number of Fisher Scoring iterations: 3
```

```
# Our diagnostic plot looks good

pearson <- (baseball$description - model$fit) / (model$fit * (1-model$fit))
est.ln.odds <- log(model$fit/(1-model$fit))
loess1 = loess(pearson~est.ln.odds)
plot(est.ln.odds,pearson)
lines(est.ln.odds[order(est.ln.odds)],loess1$fit[order(est.ln.odds)])
```




```
# 95% Confidence Interval for all predictors

lb <- exp(sum.model$coefficients[,1] - qnorm(0.975) * sum.model$coefficients[,2])
ub <- exp(sum.model$coefficients[,1] + qnorm(0.975) * sum.model$coefficients[,2])

cbind(lb,ub)
```

```
##                          lb           ub
## (Intercept)   2.142764e+03 1.622771e+05
## pitch_typeCU  4.611902e-01 1.044310e+00
## pitch_typeEP  1.801495e-02 2.558092e+00
## pitch_typeFC  5.578562e-01 1.395698e+00
## pitch_typeFF  9.616819e-01 2.099945e+00
## pitch_typeFS  3.385656e-01 1.435043e+00
## pitch_typeFT  9.104883e-01 2.058036e+00
## pitch_typeKC  4.171789e-01 1.215160e+00
## pitch_typeSI  7.824735e-01 1.875283e+00
## pitch_typeSL  8.215531e-01 1.637504e+00
## release_speed 9.458974e-01 9.933937e-01
## balls         1.056221e+00 1.320862e+00
## strikes       5.922246e-01 7.416091e-01
## countfull     8.428464e-01 2.579204e+00
## countother    9.097458e-01 2.141545e+00
## run_diff      9.620552e-01 1.030041e+00
## distance      4.766717e-04 1.494886e-03
```

Significant predictors are release speed, balls, strikes, and distance.

```
# Some calculations

log(.99708)
```

```
## [1] -0.002924272
```

```
log(9.458974e-01)
```

```
## [1] -0.05562117
```

```
log(9.933937e-01)
```

```
## [1] -0.006628218
```

```
log(5.922246e-01)
```

```
## [1] -0.5238693
```

```
log(7.416091e-01)
```

```
## [1] -0.298933
```

```
log(4.766717e-04)
```

```
## [1] -7.648683
```

```
log(1.494886e-03)
```

```
## [1] -6.505705
```

According to this model, we're 95% confident that on average, holding all else constant ...

a one unit increase in release speed will decrease the odds of a called strike between [5.56% and 0.663%].

a one unit increase in strikes will decrease the odds of a called strike between [52.38% and 52.38%].

a one unit increase in balls will increase the odds of a called strike between [5.62% ,32.08%].

a one unit increase in distance will decrease the odds of a called strike between [764.87% and 650.57%].

```
# Lets predict holding all else constant

full <- data.frame(pitch_type = "FF", release_speed = 94.3, balls = 3, strikes = 2,
                   distance = 1, count = "full", run_diff = 1)

three.less.two <- data.frame(pitch_type = "FF", release_speed = 94.3, balls = 3, strikes = 1,
                   distance = 1, count = "3less2", run_diff = 1)

other <- data.frame(pitch_type = "FF", release_speed = 94.3, balls = 0, strikes = 0,
                   distance = 1, count = "other", run_diff = 1)


predict.full <- predict.glm(model, newdata = full, type = "response")
predict.three.less.two <- predict.glm(model, newdata = three.less.two, type = "response")
predict.other <- predict.glm(model, newdata = other, type = "response")

list(predict.full = predict.full, predict.three.less.two = predict.three.less.two, predict.other = predi
```

```
## $predict.full
##         1
## 0.5579889
##
## $predict.three.less.two
##         1
## 0.5636896
##
## $predict.other
##         1
## 0.6228228
```