Cincinnati Reds

Assessment 2.5

Sam Rizzuto

7 February 2022

Load Necessary Libraries

```
library(dplyr)
library(mgcv)
library(parallel)
library(visreg)
library(ggplot2)
library(randomForest)
library(e1071)
library(caret)
library(flexclust)
library(factoextra)
library(knitr)

#set working directory
setwd("~/Desktop/22-ds")

#load in datasets
trainDF <- read.csv("train.csv")
testDF <- read.csv("test.csv")
```

Filtering Data and Removing Outliers Through IQR

```
Q1_Angle <- quantile(trainDF$ANGLE, .25)
Q3_Angle <- quantile(trainDF$ANGLE, .75)
IQR_Angle <- IQR(trainDF$ANGLE)

Q1_EXIT_SPEED <- quantile(trainDF$EXIT_SPEED, .25)
Q3_EXIT_SPEED <- quantile(trainDF$EXIT_SPEED, .75)
IQR_EXIT_SPEED <- IQR(trainDF$EXIT_SPEED)

Q1_DIRECTION <- quantile(trainDF$DIRECTION, .25)
Q3_DIRECTION <- quantile(trainDF$DIRECTION, .75)
IQR_DIRECTION <- IQR(trainDF$DIRECTION)

Q1_ReleaseSpeed <- quantile(trainDF$RELEASE_SPEED, .25)
Q3_ReleaseSpeed <- quantile(trainDF$RELEASE_SPEED, .75)
IQR_ReleaseSpeed <- IQR(trainDF$RELEASE_SPEED)

Q1_PlateX <- quantile(trainDF$PLATE_X, .25)
Q3_PlateX <- quantile(trainDF$PLATE_X, .75)
IQR_PlateX <- IQR(trainDF$PLATE_X)

Q1_PlateZ <- quantile(trainDF$PLATE_Z, .25)
Q3_PlateZ <- quantile(trainDF$PLATE_Z, .75)
IQR_PlateZ <- IQR(trainDF$PLATE_Z)

Q1_InducedVertBreak <- quantile(trainDF$INDUCED_VERTICAL_BREAK, .25)
Q3_InducedVertBreak <- quantile(trainDF$INDUCED_VERTICAL_BREAK, .75)
IQR_InducedVertBreak <- IQR(trainDF$INDUCED_VERTICAL_BREAK)

Q1_HorizontalBreak <- quantile(trainDF$HORIZONTAL_BREAK, .25)
Q3_HorizontalBreak <- quantile(trainDF$HORIZONTAL_BREAK, .75)
IQR_HorizontalBreak <- IQR(trainDF$HORIZONTAL_BREAK)

Q1_VertApproachAngle <- quantile(trainDF$VERTICAL_APPROACH_ANGLE, .25)
Q3_VertApproachAngle <- quantile(trainDF$VERTICAL_APPROACH_ANGLE, .75)
IQR_VertApproachAngle <- IQR(trainDF$VERTICAL_APPROACH_ANGLE)
```

```
Q1_HorizApproachAngle <- quantile(trainDF$HORIZONTAL_APPROACH_ANGLE, .25)
Q3_HorizApproachAngle <- quantile(trainDF$HORIZONTAL_APPROACH_ANGLE, .75)
IQR_HorizApproachAngle <- IQR(trainDF$HORIZONTAL_APPROACH_ANGLE)


trainDF <- subset(trainDF, trainDF$ANGLE > (Q1_Angle - 1.5*IQR_Angle) &
                    trainDF$ANGLE < (Q3_Angle + 1.5*IQR_Angle))
trainDF <- subset(trainDF, trainDF$EXIT_SPEED > (Q1_EXIT_SPEED - 1.5*IQR_EXIT_SPEED) &
                    trainDF$EXIT_SPEED < (Q3_EXIT_SPEED + 1.5*IQR_EXIT_SPEED))
trainDF <- subset(trainDF, trainDF$DIRECTION > (Q1_DIRECTION - 1.5*IQR_DIRECTION) &
                    trainDF$DIRECTION < (Q3_DIRECTION + 1.5*IQR_DIRECTION))

trainDF <- subset(trainDF, trainDF$RELEASE_SPEED > (Q1_ReleaseSpeed - 1.5*IQR_ReleaseSpeed) &
                    trainDF$RELEASE_SPEED < (Q3_ReleaseSpeed + 1.5*IQR_ReleaseSpeed))
trainDF <- subset(trainDF, trainDF$PLATE_X > (Q1_PlateX - 1.5*IQR_PlateX) &
                    trainDF$PLATE_X < (Q3_PlateX + 1.5*IQR_PlateX))
trainDF <- subset(trainDF, trainDF$PLATE_Z > (Q1_PlateZ - 1.5*IQR_PlateZ) &
                    trainDF$PLATE_Z < (Q3_PlateZ + 1.5*IQR_PlateZ))
trainDF <- subset(trainDF, trainDF$INDUCED_VERTICAL_BREAK > (Q1_InducedVertBreak - 1.5*IQR_InducedVertBreak) &
                    trainDF$INDUCED_VERTICAL_BREAK < (Q3_InducedVertBreak + 1.5*IQR_InducedVertBreak))
trainDF <- subset(trainDF, trainDF$HORIZONTAL_BREAK > (Q1_HorizontalBreak - 1.5*IQR_HorizontalBreak) &
                    trainDF$HORIZONTAL_BREAK < (Q3_HorizontalBreak + 1.5*IQR_HorizontalBreak))
trainDF <- subset(trainDF, trainDF$VERTICAL_APPROACH_ANGLE > (Q1_VertApproachAngle - 1.5*IQR_VertApproachAngle
                    trainDF$VERTICAL_APPROACH_ANGLE < (Q3_VertApproachAngle + 1.5*IQR_VertApproachAngle))
trainDF <- subset(trainDF, trainDF$HORIZONTAL_APPROACH_ANGLE >
                    (Q1_HorizApproachAngle - 1.5*IQR_HorizApproachAngle) &
                    trainDF$HORIZONTAL_APPROACH_ANGLE < (Q3_HorizApproachAngle + 1.5*IQR_HorizApproachAngle))

#disregard strikeouts, hbp, walks
trainDF <- trainDF %>% filter(PITCH_RESULT_KEY == "InPlay")
#goes from original 26417 rows to 24510 after cleaning

summary(trainDF)
```

```
   BATTER_UID          AVG              OBP              SLG
 Min.   :  2.00   Min.   :0.1850   Min.   :0.2570   Min.   :0.2880
 1st Qu.: 36.00   1st Qu.:0.2523   1st Qu.:0.3210   1st Qu.:0.3770
 Median : 77.00   Median :0.2670   Median :0.3380   Median :0.4125
 Mean   : 76.96   Mean   :0.2665   Mean   :0.3387   Mean   :0.4216
 3rd Qu.:118.00   3rd Qu.:0.2860   3rd Qu.:0.3580   3rd Qu.:0.4545
 Max.   :157.00   Max.   :0.3330   Max.   :0.4440   Max.   :0.6130
   VENUE_KEY         OUTS            BALLS          STRIKES
 Min.   :2528   Min.   :0.0000   Min.   :0.000   Min.   :0.000
 1st Qu.:2745   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.000
 Median :2843   Median :1.0000   Median :1.000   Median :1.000
 Mean   :3510   Mean   :0.9632   Mean   :1.106   Mean   :1.076
 3rd Qu.:4669   3rd Qu.:2.0000   3rd Qu.:2.000   3rd Qu.:2.000
 Max.   :5472   Max.   :2.0000   Max.   :4.000   Max.   :2.000
   BATS_LEFT       THROWS_LEFT      PITCH_NUMBER    RELEASE_SPEED
 Min.   :0.0000   Min.   :0.0000   Min.   : 1.000   Min.   : 70.69
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.: 84.41
 Median :0.0000   Median :0.0000   Median : 3.000   Median : 89.41
 Mean   :0.4092   Mean   :0.2502   Mean   : 3.369   Mean   : 88.48
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 5.000   3rd Qu.: 92.82
 Max.   :1.0000   Max.   :1.0000   Max.   :14.000   Max.   :102.25
    PLATE_X             PLATE_Z       INDUCED_VERTICAL_BREAK HORIZONTAL_BREAK
 Min.   :-1.526430   Min.   :0.8597   Min.   :-14.609        Min.   :-26.017
 1st Qu.:-0.379685   1st Qu.:2.0195   1st Qu.:  3.729        1st Qu.:-10.609
 Median : 0.004996   Median :2.3900   Median :  9.928        Median : -3.428
```

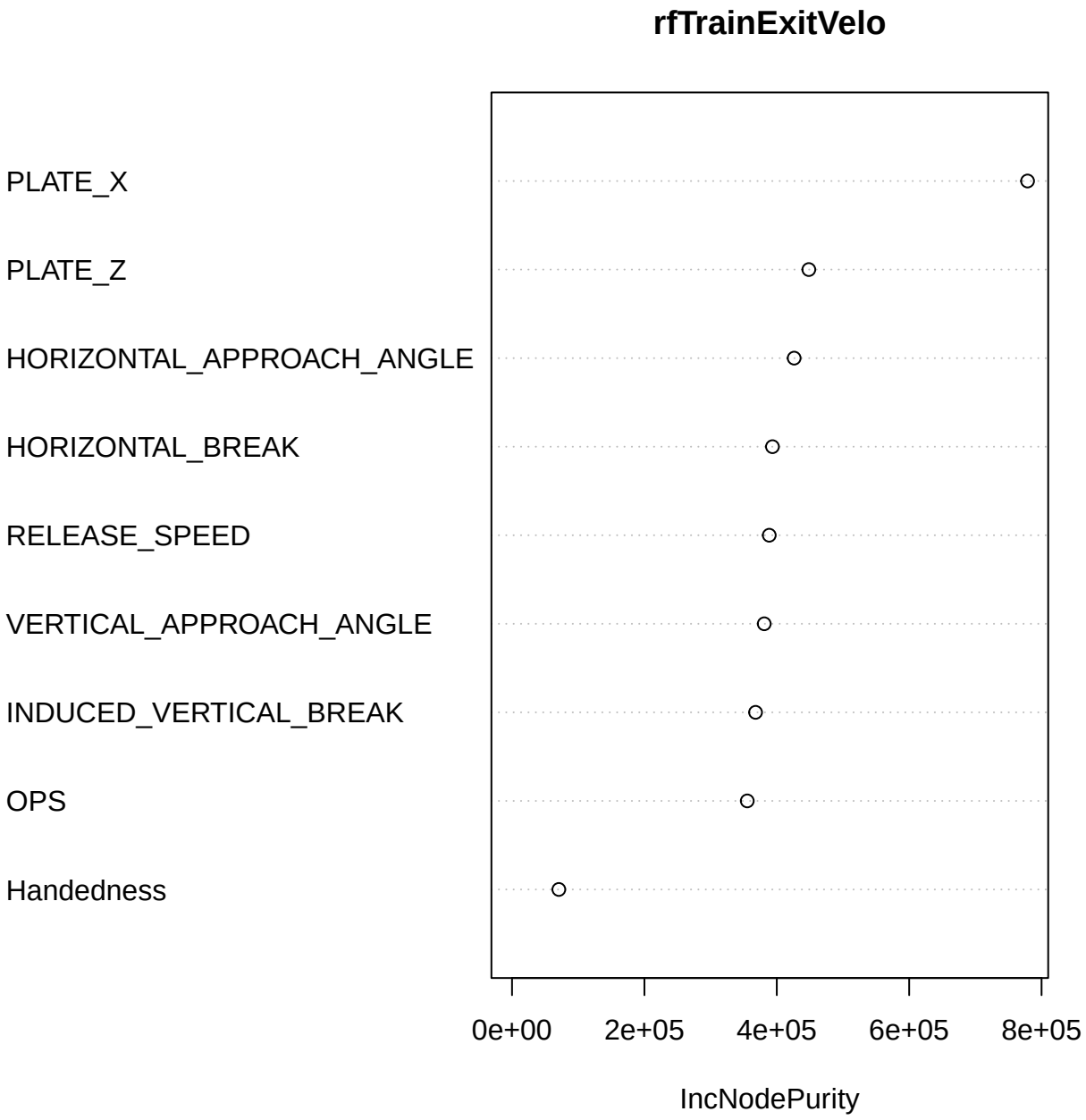
```
 Mean   :-0.001254   Mean   :2.3965   Mean   :  8.933        Mean   : -2.280
 3rd Qu.: 0.372303   3rd Qu.:2.7676   3rd Qu.: 15.245        3rd Qu.:  5.561
 Max.   : 1.518160   Max.   :3.9181   Max.   : 32.694        Max.   : 25.943
 VERTICAL_APPROACH_ANGLE HORIZONTAL_APPROACH_ANGLE   EXIT_SPEED
 Min.   :-10.290         Min.   :-4.1542           Min.   : 51.29
 1st Qu.: -7.252         1st Qu.:-0.3791           1st Qu.: 81.23
 Median : -6.208         Median : 0.8640           Median : 91.49
 Mean   : -6.371         Mean   : 0.7543           Mean   : 89.32
 3rd Qu.: -5.376         3rd Qu.: 2.0378           3rd Qu.: 98.98
 Max.   : -2.787         Max.   : 5.7566           Max.   :118.64
     ANGLE           DIRECTION       EVENT_RESULT_KEY   PITCH_RESULT_KEY
 Min.   :-58.917   Min.   :-68.576   Length:24510       Length:24510
 1st Qu.: -4.377   1st Qu.:-17.733   Class :character   Class :character
 Median : 13.619   Median : -1.480   Mode  :character   Mode  :character
 Mean   : 13.177   Mean   : -1.374
 3rd Qu.: 29.796   3rd Qu.: 14.795
 Max.   : 82.393   Max.   : 65.317
       PA          X1B              X2B               X3B
 Min.   :1   Min.   :0.0000   Min.   :0.00000   Min.   :0.000000
 1st Qu.:1   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.000000
 Median :1   Median :0.0000   Median :0.00000   Median :0.000000
 Mean   :1   Mean   :0.2225   Mean   :0.08156   Mean   :0.008772
 3rd Qu.:1   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.000000
 Max.   :1   Max.   :1.0000   Max.   :1.00000   Max.   :1.000000
       HR
 Min.   :0.00000
 1st Qu.:0.00000
 Median :0.00000
 Mean   :0.05859
 3rd Qu.:0.00000
 Max.   :1.00000
```

```
######################
```

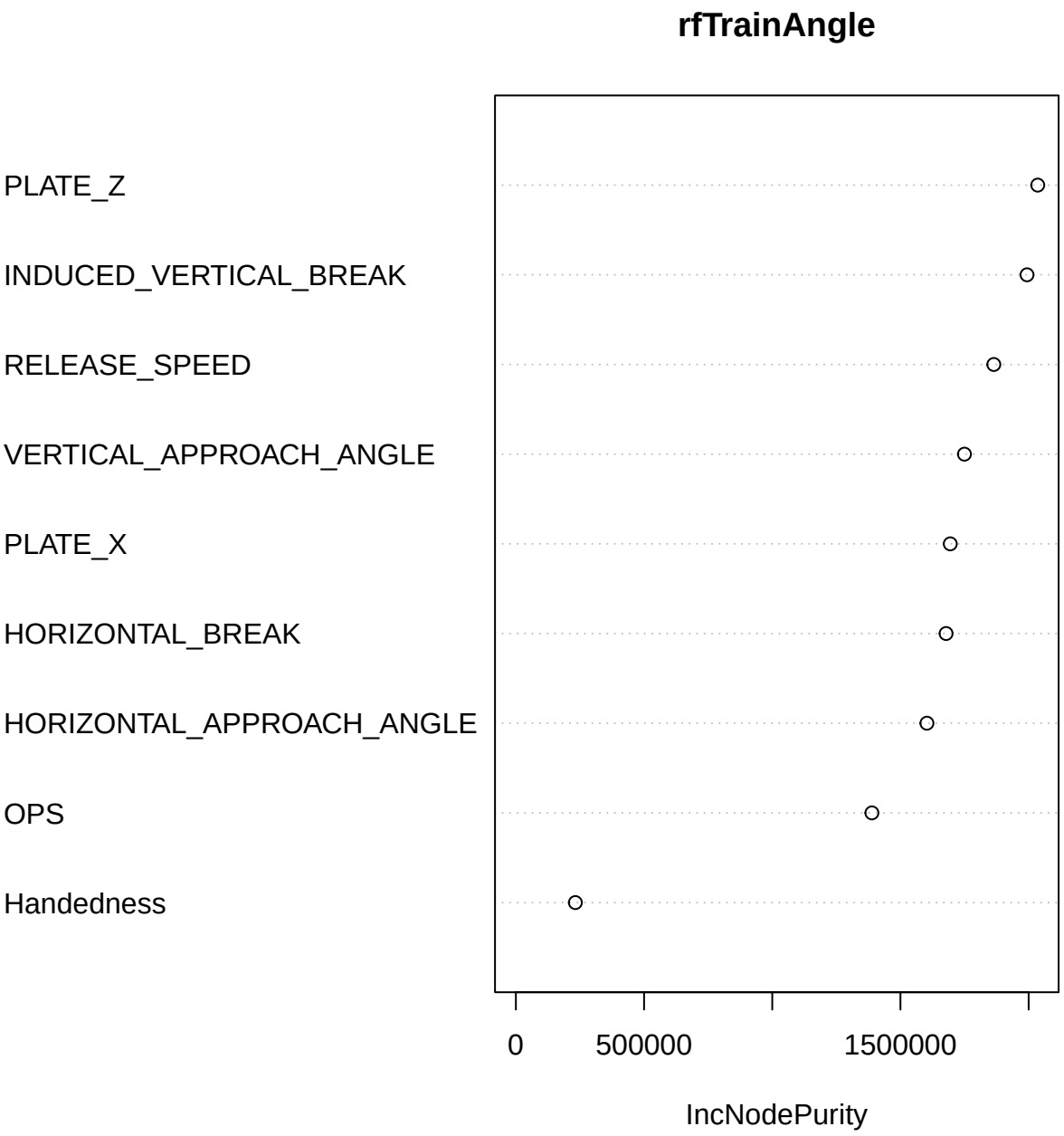
```
#create ops and handedness variables
trainDF <- trainDF %>% mutate(OPS = OBP + SLG)
testDF <- testDF %>% mutate(OPS = OBP + SLG)
trainDF <- trainDF %>% mutate(Handedness = if_else(THROWS_LEFT == BATS_LEFT, 0, 1))
testDF <- testDF %>% mutate(Handedness = if_else(THROWS_LEFT == BATS_LEFT, 0, 1))
#0 if same hands facing eachother, 1 if opposite
```

Random Forest Model

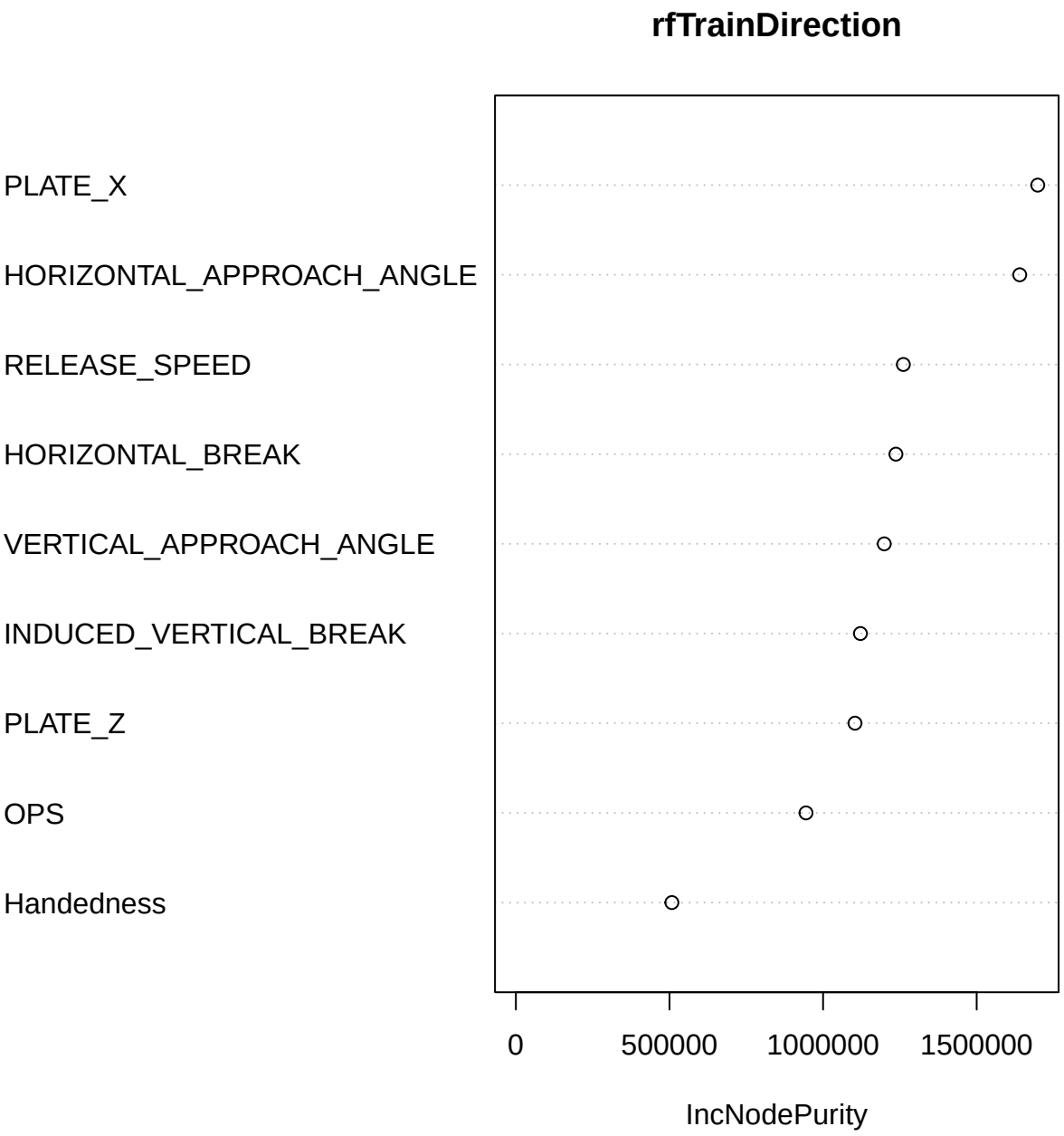
```
#running random forest on 3 prediction variables
rfTrainExitVelo <- randomForest(EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                                  INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                                  VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                                  OPS + Handedness, data = trainDF)
rfTrainAngle <- randomForest(ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                               INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                               VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                               OPS + Handedness, data = trainDF)
rfTrainDirection <- randomForest(DIRECTION ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                                   INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                                   VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                                   OPS + Handedness, data = trainDF)
#viewing importance plots of each model type to determine most significant vars in model
varImpPlot(rfTrainExitVelo) #drop handedness
```



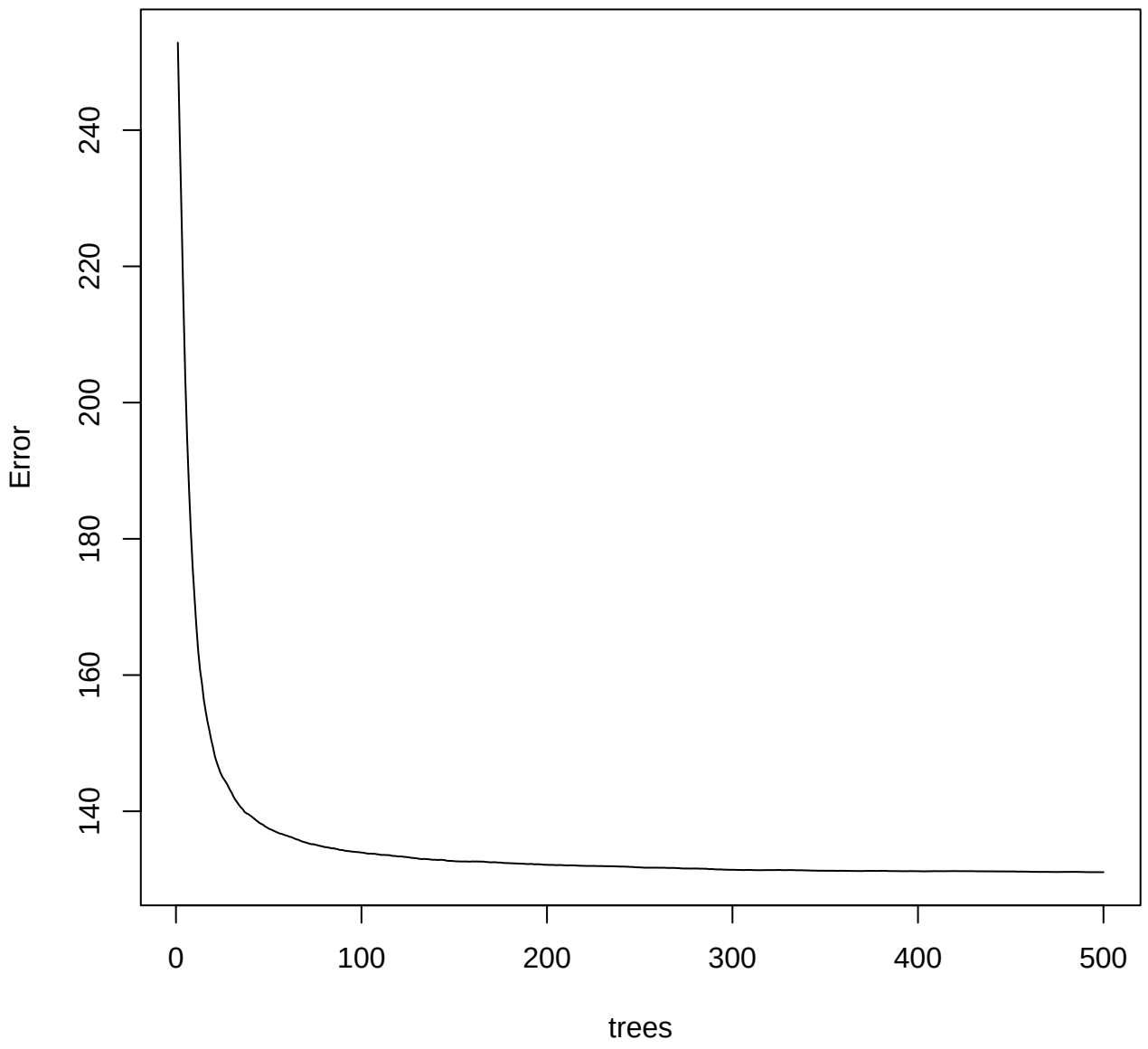

```
varImpPlot(rfTrainAngle) #drop ops, handedness
```




```
varImpPlot(rfTrainDirection) #drop handedness
```




```
#choose first grouping vars
rfUpdatedExitVelo <- randomForest(EXIT_SPEED ~ PLATE_X + PLATE_Z + HORIZONTAL_APPROACH_ANGLE +
                                  HORIZONTAL_BREAK + RELEASE_SPEED +
                                  VERTICAL_APPROACH_ANGLE + INDUCED_VERTICAL_BREAK + OPS, data = trainDF)
importance(rfUpdatedExitVelo) #importance vars of rf exit velo model

                          IncNodePurity
PLATE_X                        757661.5
PLATE_Z                        444438.7
HORIZONTAL_APPROACH_ANGLE      433829.5
HORIZONTAL_BREAK               396574.2
RELEASE_SPEED                  394198.0
VERTICAL_APPROACH_ANGLE        392711.0
INDUCED_VERTICAL_BREAK         382078.9
OPS                            364389.8

plot(rfUpdatedExitVelo) #error of rf exit velo model
```




```
rfUpdatedAngle <- randomForest(ANGLE ~ PLATE_Z + INDUCED_VERTICAL_BREAK + RELEASE_SPEED +
                                 VERTICAL_APPROACH_ANGLE +
                               PLATE_X + HORIZONTAL_BREAK + HORIZONTAL_APPROACH_ANGLE , data = trainDF)

rfUpdatedDirection <- randomForest(DIRECTION ~  PLATE_X + HORIZONTAL_APPROACH_ANGLE + RELEASE_SPEED +
                                     HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE +
                                     INDUCED_VERTICAL_BREAK + PLATE_Z + OPS, data = trainDF)



#plot randomForest results back in trainDF
trainDF <- trainDF %>%
  mutate(exitVelo_RF = round(predict(rfUpdatedExitVelo, newdata = .),3))
trainDF <- trainDF %>%
  mutate(angle_RF = round(predict(rfUpdatedAngle, newdata = .),3))
trainDF <- trainDF %>%
  mutate(direction_RF = round(predict(rfUpdatedDirection, newdata = .),3))
```

SVM Model

```
#running svm on exit velo
svmExitVelo <- svm(EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                     INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                     VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                     OPS + Handedness, data = trainDF, cost = 100, gamma = 1)
#removing predictor variable of exit velo
svmExitVelo_Pred <- round(predict(svmExitVelo, trainDF[,-19]), 3)
#add svm exit velo into training df
trainDF <- trainDF %>% mutate(exitVelo_SVM = svmExitVelo_Pred)



#running svm on angle
svmAngle <- svm(ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                   INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                   VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                   OPS + Handedness, data = trainDF, cost = 100, gamma = 1)
#removing predictor variable of angle
svmAngle_Pred <- round(predict(svmAngle, trainDF[,-20]), 3)
#adding prob of angle into training df
trainDF <- trainDF %>% mutate(angle_SVM = svmAngle_Pred)



#running svm model on direction
svmDirection <- svm(DIRECTION ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
                      INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                      VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
                      OPS + Handedness, data = trainDF, cost = 100, gamma = 1)
#removing predictor variable of direction
svmDirection_Pred <- round(predict(svmDirection, trainDF[,-21]), 3)
#adding direction into training df
trainDF <- trainDF %>% mutate(direction_SVM = svmDirection_Pred)
```

Running GAM Model

```
####generalized additive model to include most important variables in training dataset
options(mc.cores = parallel::detectCores())#run model in parallel

gam_EV <- bam(EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
              HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
            data = trainDF, family = gaussian, method = "GCV.Cp")
summary(gam_EV) #drop platex, platez, vert approach angle


Family: gaussian
Link function: identity

Formula:
EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
    HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
    OPS + Handedness

Parametric coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)              62.49285    3.30876  18.887  < 2e-16 ***
RELEASE_SPEED             0.18434    0.02492   7.398 1.42e-13 ***
```

```
PLATE_X                    0.07433    0.17082   0.435   0.6635
PLATE_Z                    0.06586    0.21768   0.303   0.7622
INDUCED_VERTICAL_BREAK     0.03296    0.01815   1.816   0.0694 .
HORIZONTAL_BREAK          -0.02235    0.00887  -2.520   0.0118 *
VERTICAL_APPROACH_ANGLE   -0.06991    0.15320  -0.456   0.6482
HORIZONTAL_APPROACH_ANGLE -0.08299    0.04914  -1.689   0.0912 .
OPS                       12.22985    0.99085  12.343  < 2e-16 ***
Handedness                 0.66970    0.16287   4.112 3.94e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R-sq.(adj) =  0.0163   Deviance explained = 1.67%
GCV = 154.34  Scale est. = 154.28    n = 24510
```

```
gam_EV_Upd <- bam(EXIT_SPEED ~ RELEASE_SPEED + INDUCED_VERTICAL_BREAK +
                HORIZONTAL_BREAK + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
              data = trainDF, family = gaussian, method = "GCV.Cp")
```

```
gam_Ang <- bam(ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
              HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
            data = trainDF, family = gaussian, method = "GCV.Cp")
summary(gam_Ang) #no drop
```

```
Family: gaussian
Link function: identity

Formula:
ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
    HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
    OPS + Handedness

Parametric coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               80.24128    6.36217  12.612  < 2e-16 ***
RELEASE_SPEED             -1.09556    0.04791 -22.867  < 2e-16 ***
PLATE_X                   -1.21579    0.32846  -3.702 0.000215 ***
PLATE_Z                    8.14747    0.41856  19.466  < 2e-16 ***
INDUCED_VERTICAL_BREAK     0.78607    0.03490  22.522  < 2e-16 ***
HORIZONTAL_BREAK           0.06256    0.01706   3.668 0.000245 ***
VERTICAL_APPROACH_ANGLE    0.55608    0.29459   1.888 0.059084 .
HORIZONTAL_APPROACH_ANGLE  0.75715    0.09448   8.014 1.16e-15 ***
OPS                        7.73271    1.90523   4.059 4.95e-05 ***
Handedness                 1.10150    0.31318   3.517 0.000437 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R-sq.(adj) =  0.0814   Deviance explained = 8.17%
GCV = 570.63  Scale est. = 570.4     n = 24510
```

```
gam_Ang_Upd <- gam_Ang
```

```
gam_Dir <- bam(DIRECTION ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
              HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
            data = trainDF, family = gaussian, method = "GCV.Cp")
summary(gam_Dir) #drop platez
```

```
Family: gaussian
```

```
Link function: identity

Formula:
DIRECTION ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
    HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
    OPS + Handedness

Parametric coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               -8.41722    5.44751  -1.545 0.122323
RELEASE_SPEED              0.14499    0.04102   3.534 0.000409 ***
PLATE_X                    9.80451    0.28124  34.862  < 2e-16 ***
PLATE_Z                    0.06672    0.35839   0.186 0.852306
INDUCED_VERTICAL_BREAK    -0.07327    0.02989  -2.452 0.014230 *
HORIZONTAL_BREAK          -0.24674    0.01460 -16.896  < 2e-16 ***
VERTICAL_APPROACH_ANGLE    0.48057    0.25223   1.905 0.056760 .
HORIZONTAL_APPROACH_ANGLE  0.68024    0.08090   8.408  < 2e-16 ***
OPS                       -9.65788    1.63132  -5.920 3.26e-09 ***
Handedness                 8.09336    0.26816  30.181  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R-sq.(adj) =  0.098   Deviance explained = 9.83%
GCV = 418.35  Scale est. = 418.18    n = 24510
```

```
gam_Dir_Upd <- bam(DIRECTION ~ RELEASE_SPEED + PLATE_X + INDUCED_VERTICAL_BREAK +
                  HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
                data = trainDF, family = gaussian, method = "GCV.Cp")

#coefficients of each model to use in prediciting values
gamEV <- gam_EV_Upd$coefficients
gamAng <- gam_Ang_Upd$coefficients
gamDir <- gam_Dir_Upd$coefficients

trainDF <- trainDF %>%
  mutate(exitVelo_GAM = round(gamEV[1] + gamEV[2]*RELEASE_SPEED + gamEV[3]*INDUCED_VERTICAL_BREAK +
                                gamEV[4]*HORIZONTAL_BREAK + gamEV[5]*HORIZONTAL_APPROACH_ANGLE +
                                gamEV[6]*OPS + gamEV[7]*Handedness, 3))

trainDF <- trainDF %>%
  mutate(angle_GAM = round(gamAng[1] + gamAng[2]*RELEASE_SPEED + gamAng[3]*PLATE_X +
                             gamAng[4]*PLATE_Z + gamAng[5]*INDUCED_VERTICAL_BREAK +
                             gamAng[6]*HORIZONTAL_BREAK + gamAng[7]*VERTICAL_APPROACH_ANGLE +
                             gamAng[8]*HORIZONTAL_APPROACH_ANGLE +
                             gamAng[9]*OPS + gamAng[10]*Handedness, 3))

trainDF <- trainDF %>%
  mutate(direction_GAM = round(gamDir[1] + gamDir[2]*RELEASE_SPEED + gamDir[3]*PLATE_X +
                                 gamDir[4]*INDUCED_VERTICAL_BREAK +gamDir[5]*HORIZONTAL_BREAK +
                                 gamDir[6]*VERTICAL_APPROACH_ANGLE +
                                 gamDir[7]*HORIZONTAL_APPROACH_ANGLE + gamDir[8]*OPS +
                                 gamDir[9]*Handedness, 3))
```

Smaller GAM Model to Create Visuals

```
###smaller gam
#exit speed just on x and y location
```

```
gam_Small <- bam(EXIT_SPEED ~ s(PLATE_X, PLATE_Z),
            data = trainDF, family = gaussian, method = "GCV.Cp")

xs <- matrix(data=seq(from=-2, to=2, length=50), nrow=50, ncol=50)
ys <- t(matrix(data=seq(from=0,to=5, length=50), nrow=50, ncol=50))

gamSmallFit <- data.frame(PLATE_X = as.vector(xs), PLATE_Z = as.vector(ys))
exitVeloPred <- predict(gam_Small, gamSmallFit, types = "response")
exitVeloPred <- matrix(exitVeloPred, nrow = 50, ncol = 50)
density(exitVeloPred)
```

```
Call:
    density.default(x = exitVeloPred)

Data: exitVeloPred (2500 obs.); Bandwidth 'bw' = 1.451

       x               y
 Min.   :56.32   Min.   :2.710e-06
 1st Qu.:66.96   1st Qu.:5.796e-03
 Median :77.59   Median :2.556e-02
 Mean   :77.59   Mean   :2.349e-02
 3rd Qu.:88.22   3rd Qu.:3.826e-02
 Max.   :98.85   Max.   :4.460e-02
```

```
summary(trainDF$EXIT_SPEED)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  51.29   81.23   91.49   89.32   98.98  118.64
```

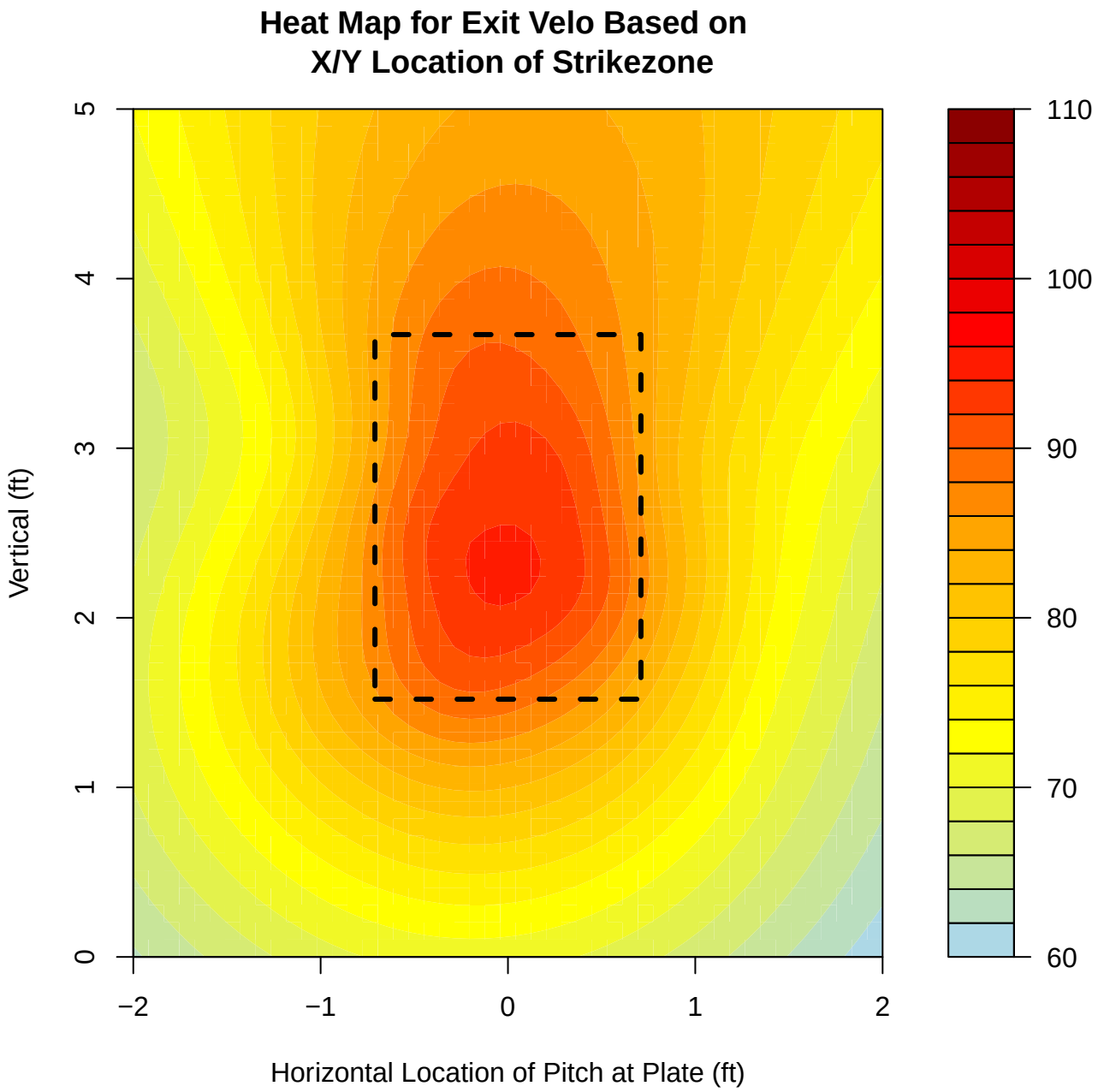
```
summary(trainDF$PLATE_X) #-1.5min to 1.5max, so round to -2 and 2 for x
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.526430 -0.379685  0.004996 -0.001254  0.372303  1.518160
```

```
summary(trainDF$PLATE_Z) #0.8 to 4.9, so round to 0 and 5
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.8597  2.0195  2.3900  2.3965  2.7676  3.9181
```

```
#range of 60-110 to get full exit velo based on iqr
#width of HP is 23in
#height of strikezone is 25.79in based on baseball prospectus,
#with the bottom starting at 18.29in above the ground, which is 1.52ft
#to get the top of the zone, add 1.52ft plus the height (25.79in) to get 44.08in, or 3.67ft
#To create width of strikezone (17in), divide by 2 = 8.5in on each side from the middle
#and convert to ft = 0.71ft from the center
#so now when drawing strikezone, it has width -0.71 to 0.71 and height 1.52 to 3.67
filled.contour(x=seq(from=-2, to=2, length=50), y=seq(from=0, to=5, length=50), z = exitVeloPred,
               zlim=c(60,110),
               color.palette = colorRampPalette(c("lightblue","yellow","orange", "red", "darkred")),
               plot.axes = { rect(-0.71, 1.52, 0.71, 3.67, border="black", lty="dashed", lwd=3)
                 axis(1, at=c(-2,-1,0,1,2), pos=0, labels=c(-2,-1,0,1,2), las=0, col="black")
                 axis(2, at=c(0,1,2,3,4,5), pos=-2, labels=c(0,1,2,3,4,5), las=0, col="black")
                 },
               main = "Heat Map for Exit Velo Based on \n X/Y Location of Strikezone",
               ylab = "Vertical (ft)",
               xlab = "Horizontal Location of Pitch at Plate (ft)")
```

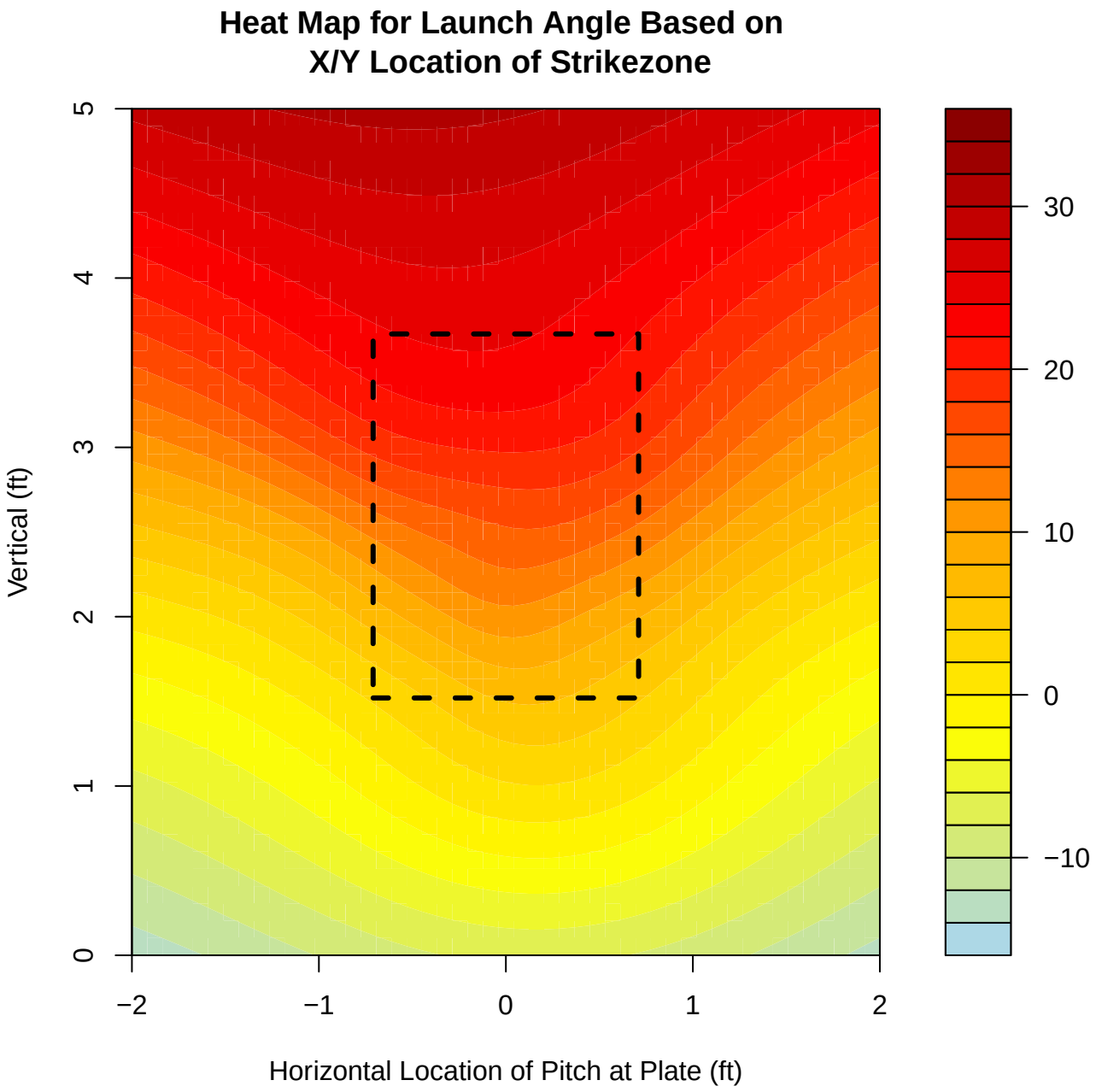



```
#launch angle
gam_Small2 <- bam(ANGLE ~ s(PLATE_X, PLATE_Z),
                  data = trainDF, family = gaussian, method = "GCV.Cp")
gamSmallFit2 <- data.frame(PLATE_X = as.vector(xs), PLATE_Z = as.vector(ys))
anglePred <- predict(gam_Small2, gamSmallFit2, types = "response")
anglePred <- matrix(anglePred, nrow = 50, ncol = 50)
summary(trainDF$ANGLE)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-58.917  -4.377  13.619  13.177  29.796  82.393
```

```
filled.contour(x=seq(from=-2, to=2, length=50), y=seq(from=0, to=5, length=50), z = anglePred,
               zlim=c(-15,35),
               color.palette = colorRampPalette(c("lightblue","yellow","orange", "red", "darkred")),
               plot.axes = { rect(-0.71, 1.52, 0.71, 3.67, border="black", lty="dashed", lwd=3)
                 axis(1, at=c(-2,-1,0,1,2), pos=0, labels=c(-2,-1,0,1,2), las=0, col="black")
                 axis(2, at=c(0,1,2,3,4,5), pos=-2, labels=c(0,1,2,3,4,5), las=0, col="black")
               },
               main = "Heat Map for Launch Angle Based on \n X/Y Location of Strikezone",
```

```
       ylab = "Vertical (ft)",
       xlab = "Horizontal Location of Pitch at Plate (ft)")
```

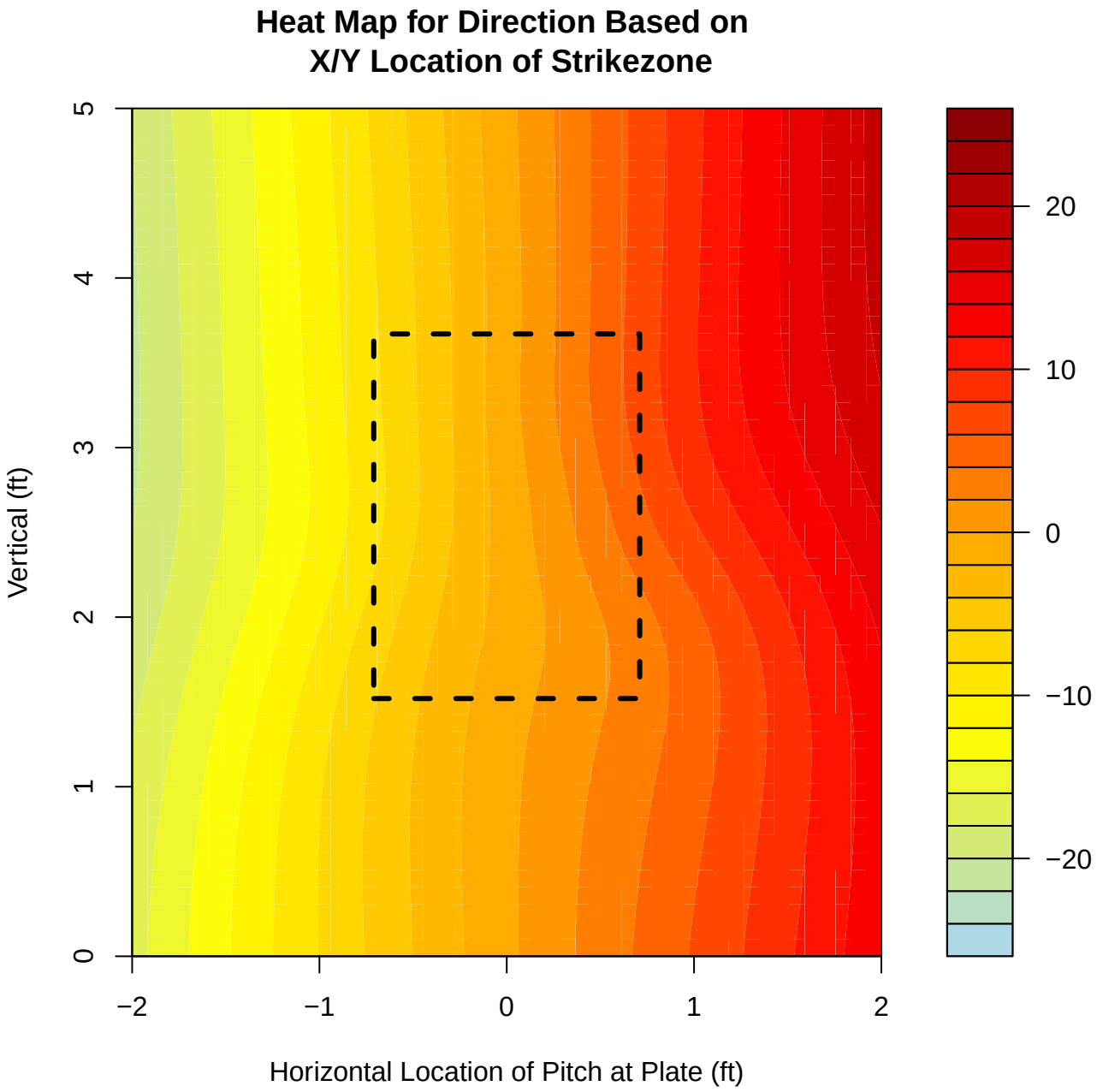



```
#direction
gam_Small3 <- bam(DIRECTION ~ s(PLATE_X, PLATE_Z),
                  data = trainDF, family = gaussian, method = "GCV.Cp")
gamSmallFit3 <- data.frame(PLATE_X = as.vector(xs), PLATE_Z = as.vector(ys))
directionPred <- predict(gam_Small3, gamSmallFit3, types = "response")
directionPred <- matrix(directionPred, nrow = 50, ncol = 50)
summary(trainDF$DIRECTION)
```

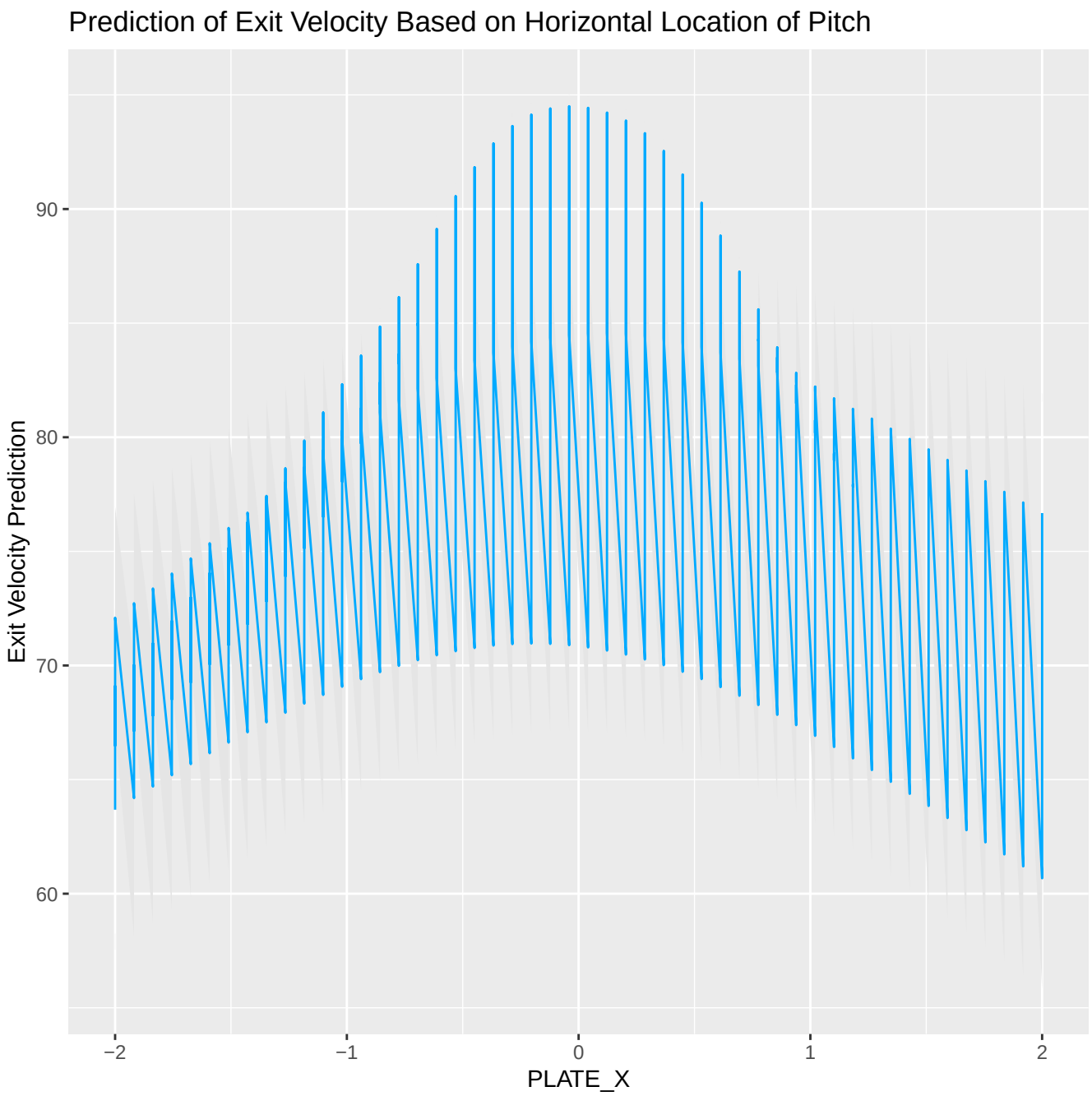
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-68.576 -17.733  -1.480  -1.374  14.795  65.317
```

```
#not taking into account handedness (l/r are combined here)
filled.contour(x=seq(from=-2, to=2, length=50), y=seq(from=0, to=5, length=50), z = directionPred,
               zlim=c(-25,25),
               color.palette = colorRampPalette(c("lightblue","yellow","orange", "red", "darkred")),
```

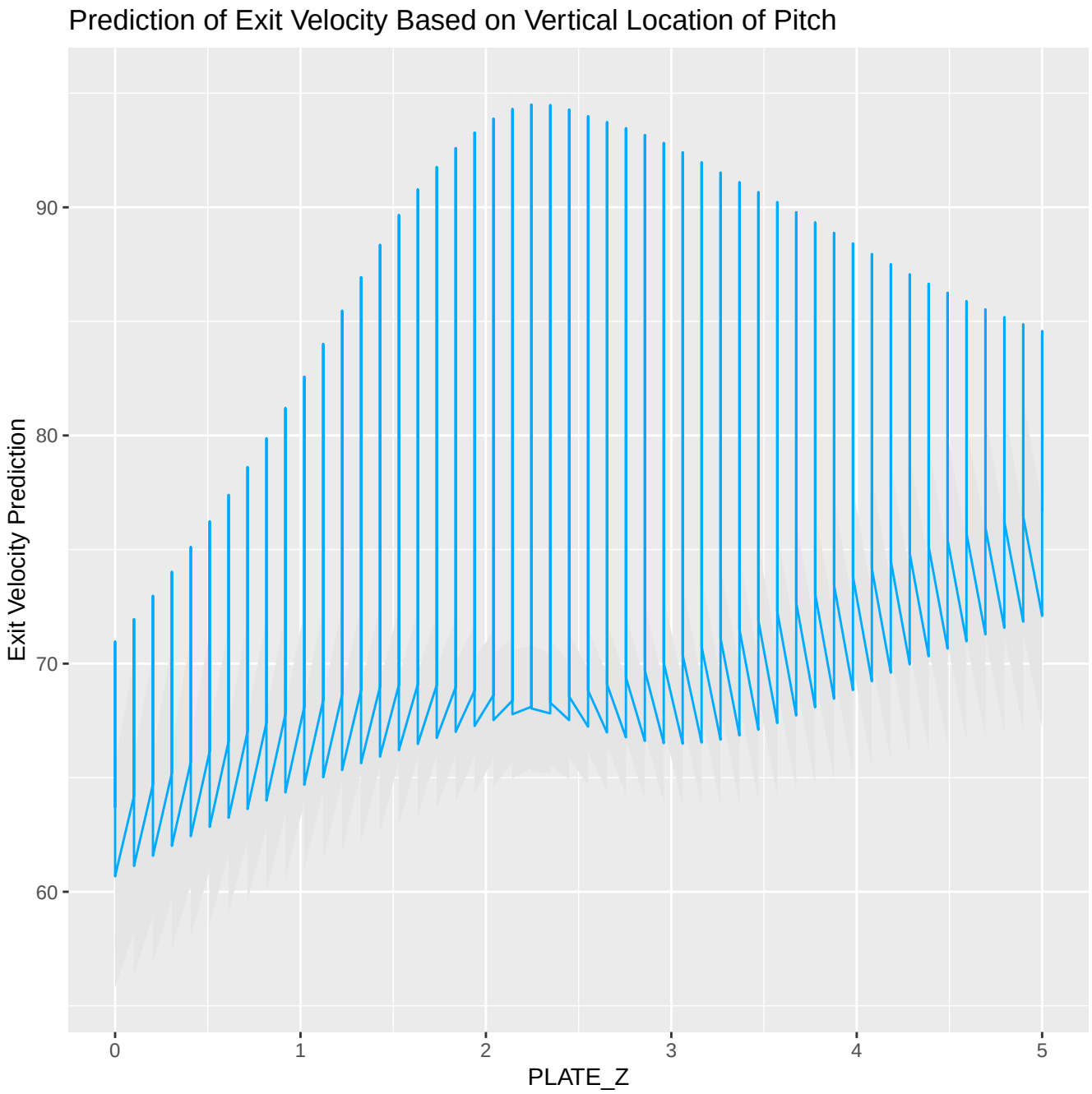
```
plot.axes = { rect(-0.71, 1.52, 0.71, 3.67, border="black", lty="dashed", lwd=3)
  axis(1, at=c(-2,-1,0,1,2), pos=0, labels=c(-2,-1,0,1,2), las=0, col="black")
  axis(2, at=c(0,1,2,3,4,5), pos=-2, labels=c(0,1,2,3,4,5), las=0, col="black")
},
main = "Heat Map for Direction Based on \n X/Y Location of Strikezone",
ylab = "Vertical (ft)",
xlab = "Horizontal Location of Pitch at Plate (ft)")
```



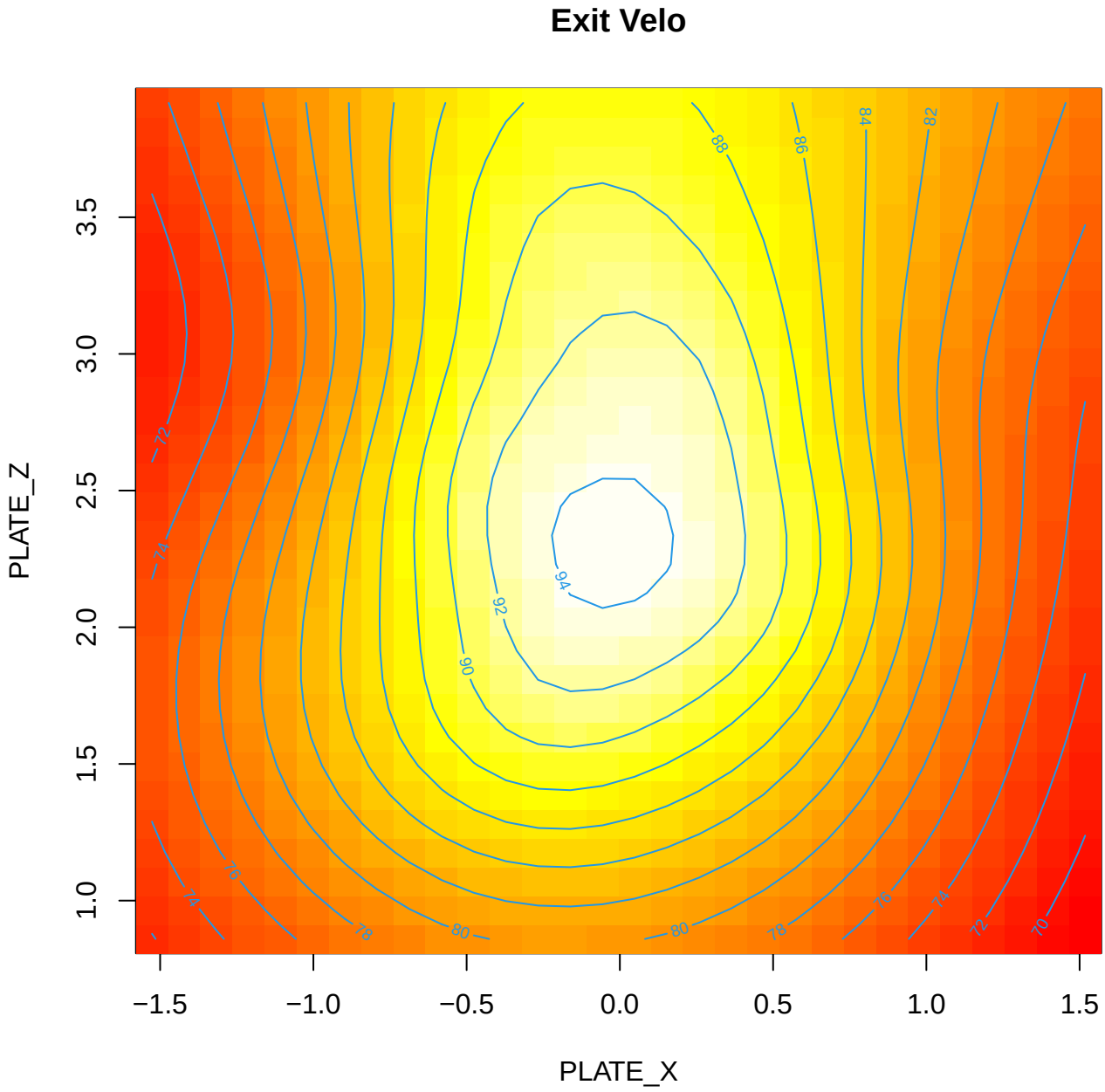

```
#building predicted values with upper/lower bounds
exitVeloPredConf <- predict(gam_Small, gamSmallFit, types = "response", se = TRUE)
predVals <- data.frame(gamSmallFit, exitVeloPredConf) %>%
  mutate(lower = exitVeloPredConf$fit - 1.96*exitVeloPredConf$se.fit,
         upper = exitVeloPredConf$fit + 1.96*exitVeloPredConf$se.fit)

ggplot(aes(x=PLATE_X,y=exitVeloPredConf$fit), data=predVals) +
  geom_ribbon(aes(ymin = lower, ymax=upper), fill='gray90') +
  geom_line(color='#00aaff') + ylab("Exit Velocity Prediction") +
  ggtitle("Prediction of Exit Velocity Based on Horizontal Location of Pitch")
```



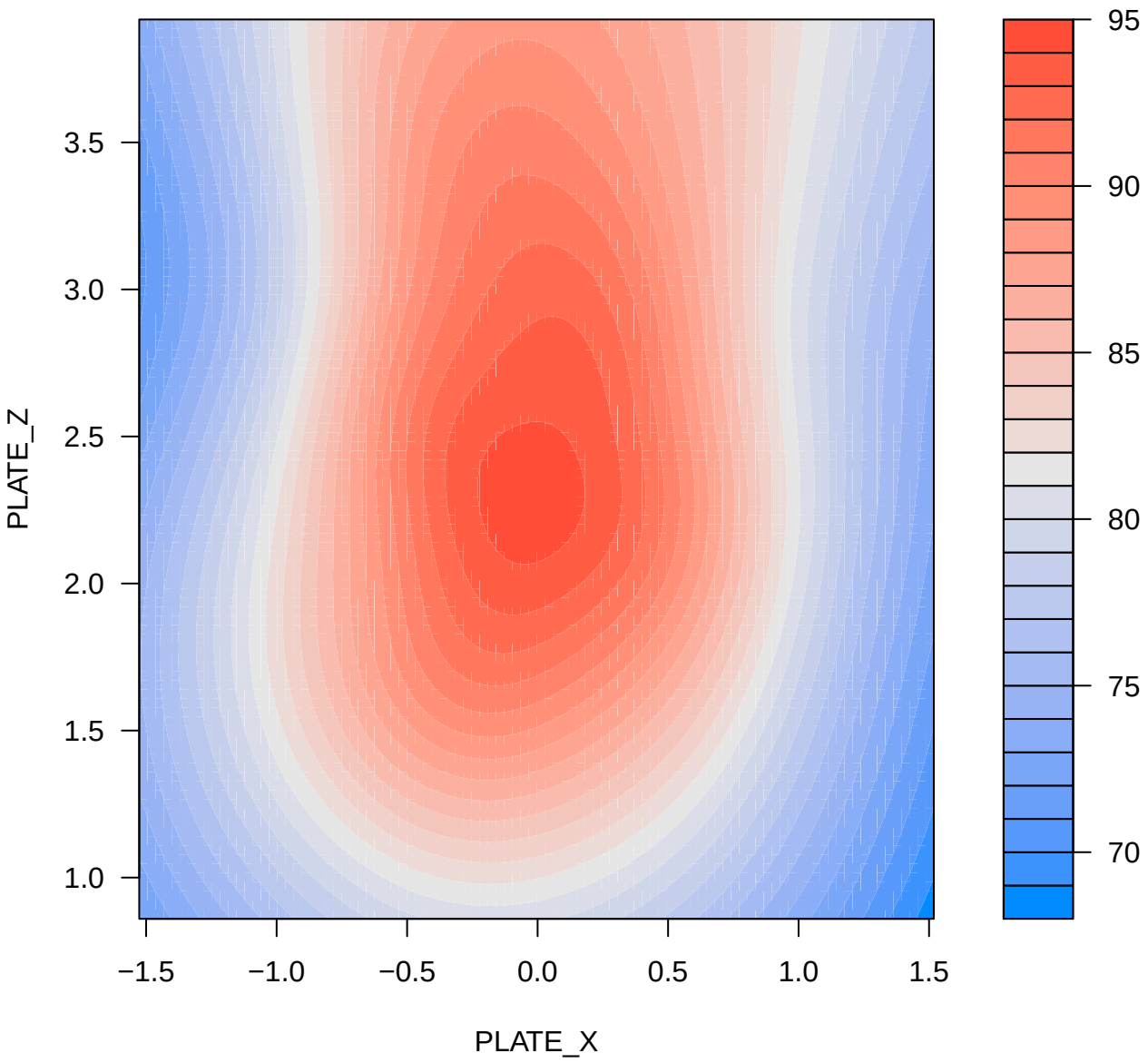

```
ggplot(aes(x=PLATE_Z,y=exitVeloPredConf$fit), data=predVals) +
  geom_ribbon(aes(ymin = lower, ymax=upper), fill='gray90') +
  geom_line(color='#00aaff') + ylab("Exit Velocity Prediction") +
  ggtitle("Prediction of Exit Velocity Based on Vertical Location of Pitch")
```



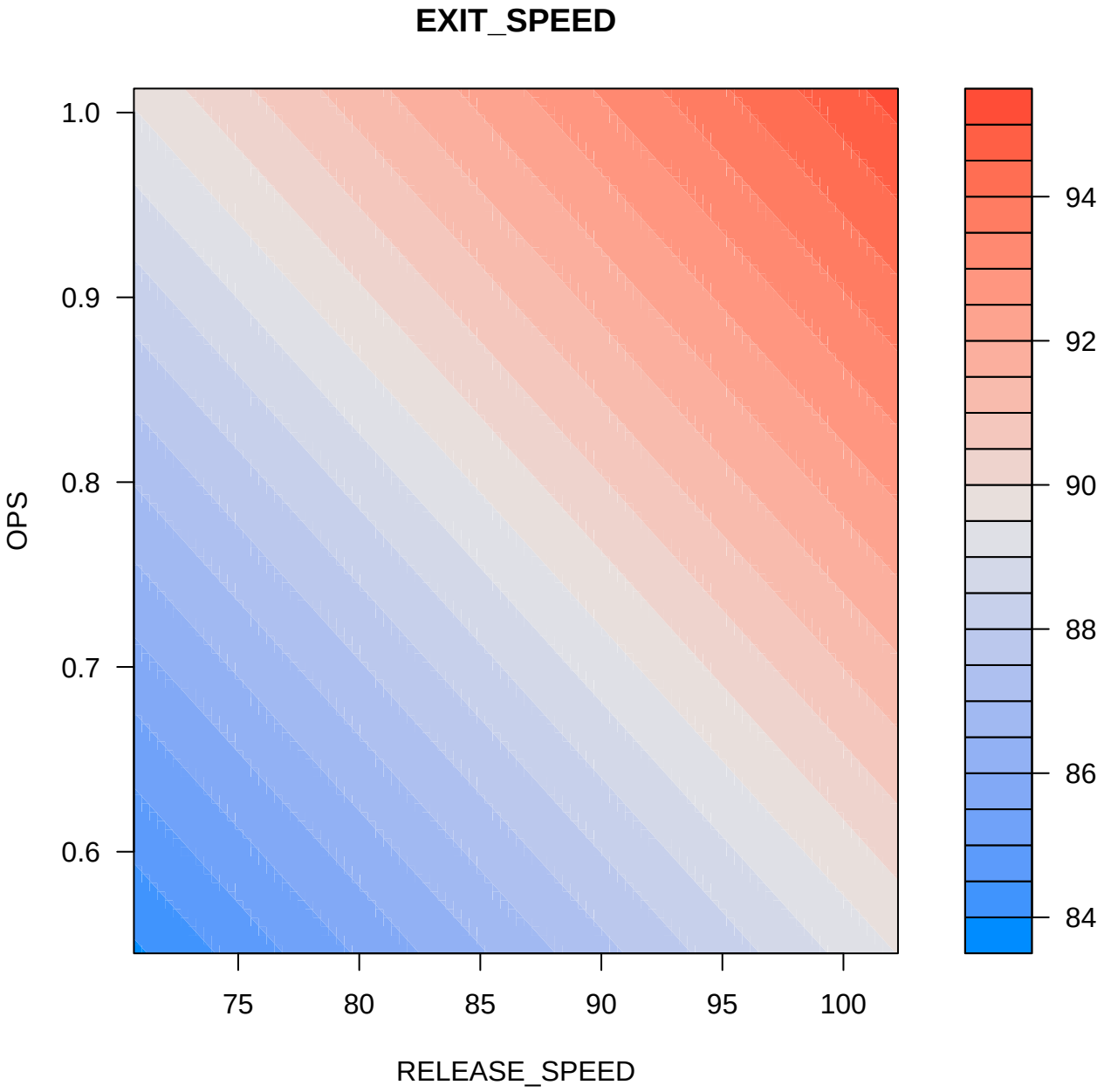

```
#both plots show where the highest exit velo is based on x and y location of pitch

vis.gam(gam_Small, type='response', plot.type='contour', main = "Exit Velo")
```




```
visreg2d(gam_Small, xvar='PLATE_X', yvar='PLATE_Z', scale='response')
```




```
visreg2d(gam_EV_Upd, xvar='RELEASE_SPEED', yvar='OPS', scale='response')
```




```
anova(gam_EV_Upd, gam_Small, test="Chisq")

Analysis of Deviance Table

Model 1: EXIT_SPEED ~ RELEASE_SPEED + INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
    HORIZONTAL_APPROACH_ANGLE + OPS + Handedness
Model 2: EXIT_SPEED ~ s(PLATE_X, PLATE_Z)
  Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
1     24503    3779820
2     24482    3334731 20.664   445089 < 2.2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

anova(gam_EV_Upd)

Family: gaussian
Link function: identity

Formula:
```

```
EXIT_SPEED ~ RELEASE_SPEED + INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
    HORIZONTAL_APPROACH_ANGLE + OPS + Handedness

Parametric Terms:
                          df       F  p-value
RELEASE_SPEED              1  74.264  < 2e-16
INDUCED_VERTICAL_BREAK     1   3.892   0.0485
HORIZONTAL_BREAK           1   6.122   0.0134
HORIZONTAL_APPROACH_ANGLE  1   2.810   0.0937
OPS                        1 152.669  < 2e-16
Handedness                 1  17.063 3.63e-05

anova(gam_Ang_Upd)

Family: gaussian
Link function: identity

Formula:
ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
    HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
    OPS + Handedness

Parametric Terms:
                          df       F  p-value
RELEASE_SPEED              1 522.915  < 2e-16
PLATE_X                    1  13.701 0.000215
PLATE_Z                    1 378.906  < 2e-16
INDUCED_VERTICAL_BREAK     1 507.219  < 2e-16
HORIZONTAL_BREAK           1  13.454 0.000245
VERTICAL_APPROACH_ANGLE    1   3.563 0.059084
HORIZONTAL_APPROACH_ANGLE  1  64.218 1.16e-15
OPS                        1  16.473 4.95e-05
Handedness                 1  12.370 0.000437

anova(gam_Dir_Upd)

Family: gaussian
Link function: identity

Formula:
DIRECTION ~ RELEASE_SPEED + PLATE_X + INDUCED_VERTICAL_BREAK +
    HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE +
    OPS + Handedness

Parametric Terms:
                          df        F  p-value
RELEASE_SPEED              1   13.982 0.000185
PLATE_X                    1 1216.430  < 2e-16
INDUCED_VERTICAL_BREAK     1    7.122 0.007618
HORIZONTAL_BREAK           1  286.490  < 2e-16
VERTICAL_APPROACH_ANGLE    1    8.440 0.003674
HORIZONTAL_APPROACH_ANGLE  1   70.704  < 2e-16
OPS                        1   35.056 3.25e-09
Handedness                 1  915.549  < 2e-16
```

GLM model

```
############exit velo
exitVeloCalc <- glm(EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
```

```
                    HORIZONTAL_BREAK +
                    VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
                data = trainDF,
                family = gaussian)
summary(exitVeloCalc)

Call:
glm(formula = EXIT_SPEED ~ RELEASE_SPEED + PLATE_X + PLATE_Z +
    INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK + VERTICAL_APPROACH_ANGLE +
    HORIZONTAL_APPROACH_ANGLE + OPS + Handedness, family = gaussian,
    data = trainDF)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-40.874   -7.901    2.099    9.573   31.016

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               62.49285    3.30876  18.887  < 2e-16 ***
RELEASE_SPEED              0.18434    0.02492   7.398 1.42e-13 ***
PLATE_X                    0.07433    0.17082   0.435   0.6635
PLATE_Z                    0.06586    0.21768   0.303   0.7622
INDUCED_VERTICAL_BREAK     0.03296    0.01815   1.816   0.0694 .
HORIZONTAL_BREAK          -0.02235    0.00887  -2.520   0.0118 *
VERTICAL_APPROACH_ANGLE   -0.06991    0.15320  -0.456   0.6482
HORIZONTAL_APPROACH_ANGLE -0.08299    0.04914  -1.689   0.0912 .
OPS                       12.22985    0.99085  12.343  < 2e-16 ***
Handedness                 0.66970    0.16287   4.112 3.94e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 154.2758)

    Null deviance: 3843946  on 24509  degrees of freedom
Residual deviance: 3779758  on 24500  degrees of freedom
AIC: 193068

Number of Fisher Scoring iterations: 2

exitVeloCalc$coefficients #coefficients used to determine exit velo

              (Intercept)             RELEASE_SPEED                   PLATE_X
              62.49285293                0.18433823                0.07433175
                  PLATE_Z    INDUCED_VERTICAL_BREAK          HORIZONTAL_BREAK
                0.06586319                0.03296162               -0.02234830
  VERTICAL_APPROACH_ANGLE HORIZONTAL_APPROACH_ANGLE                       OPS
              -0.06991142               -0.08298921               12.22984853
               Handedness
                0.66969731

anova(exitVeloCalc, test = "Chisq")

Analysis of Deviance Table

Model: gaussian, link: identity

Response: EXIT_SPEED

Terms added sequentially (first to last)
```

```
                         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                     24509    3843946
RELEASE_SPEED             1    35498     24508    3808449 < 2.2e-16 ***
PLATE_X                   1      106     24507    3808343  0.407391
PLATE_Z                   1        0     24506    3808342  0.964155
INDUCED_VERTICAL_BREAK    1     1266     24505    3807077  0.004179 **
HORIZONTAL_BREAK          1      792     24504    3806285  0.023460 *
VERTICAL_APPROACH_ANGLE   1       29     24503    3806256  0.665747
HORIZONTAL_APPROACH_ANGLE 1     1088     24502    3805167  0.007905 **
OPS                       1    22801     24501    3782366 < 2.2e-16 ***
Handedness                1     2608     24500    3779758 3.927e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
############angle
angleCalc <- glm(ANGLE ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK + HORIZONTAL_BREAK +
                     VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
                   data = trainDF,
                   family = gaussian)
anova(angleCalc, test = "Chisq")
```

```
Analysis of Deviance Table

Model: gaussian, link: identity

Response: ANGLE

Terms added sequentially (first to last)

                         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                     24509   15217969
RELEASE_SPEED             1    13523     24508   15204446 1.121e-06 ***
PLATE_X                   1      368     24507   15204078 0.4220621
PLATE_Z                   1   686790     24506   14517288 < 2.2e-16 ***
INDUCED_VERTICAL_BREAK    1   484625     24505   14032663 < 2.2e-16 ***
HORIZONTAL_BREAK          1     7192     24504   14025471 0.0003839 ***
VERTICAL_APPROACH_ANGLE   1     3371     24503   14022100 0.0150540 *
HORIZONTAL_APPROACH_ANGLE 1    31636     24502   13990464 9.525e-14 ***
OPS                       1     8640     24501   13981824 9.941e-05 ***
Handedness                1     7056     24500   13974768 0.0004363 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
############direction
directionCalc <- glm(DIRECTION ~ RELEASE_SPEED + PLATE_X + PLATE_Z + INDUCED_VERTICAL_BREAK +
                       HORIZONTAL_BREAK +
                     VERTICAL_APPROACH_ANGLE + HORIZONTAL_APPROACH_ANGLE + OPS + Handedness,
                   data = trainDF,
                   family = gaussian)
anova(directionCalc, test = "Chisq")
```

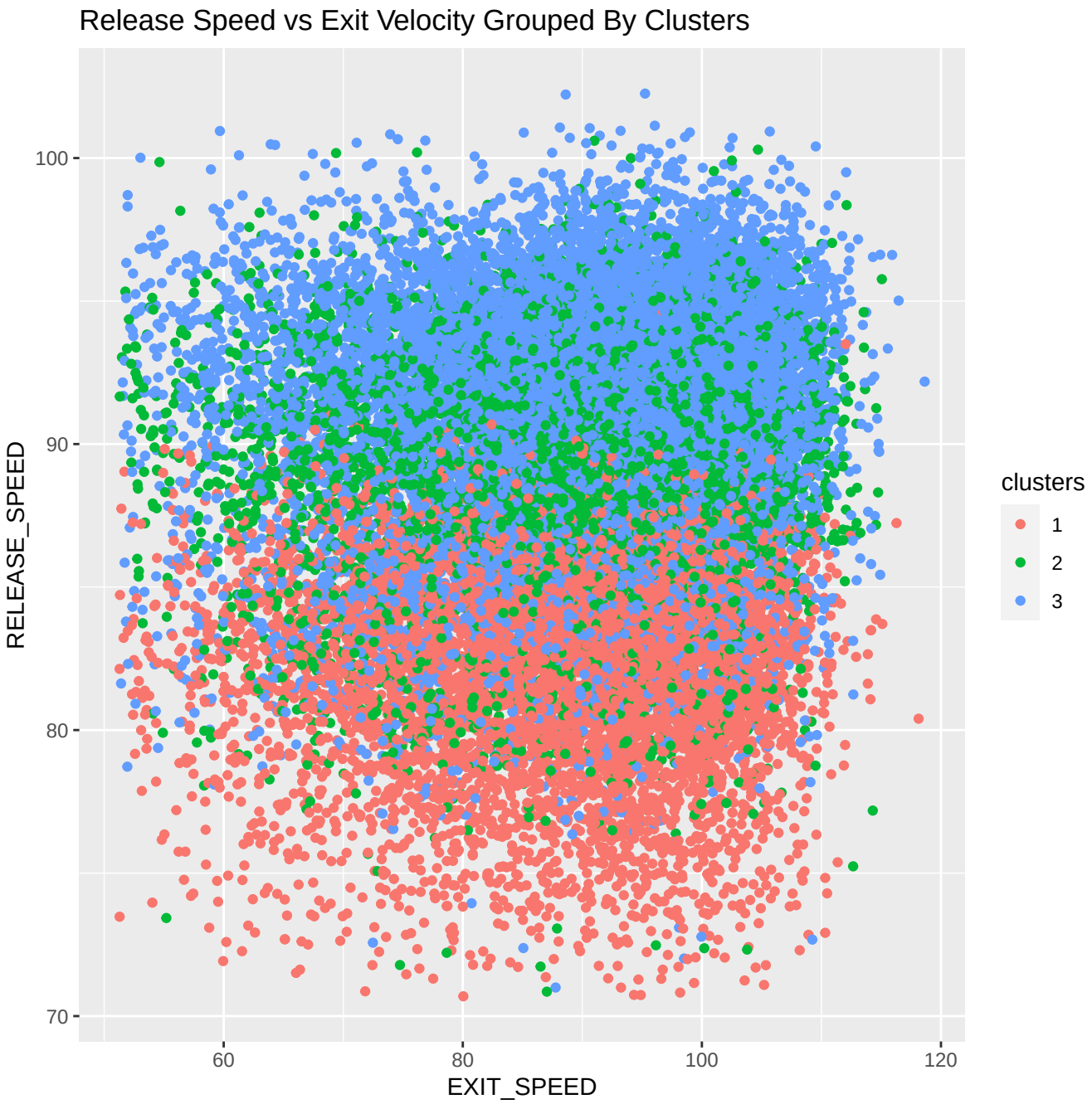
```
Analysis of Deviance Table

Model: gaussian, link: identity

Response: DIRECTION

Terms added sequentially (first to last)
```

```
                         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                     24509   11362518
RELEASE_SPEED             1    37027     24508   11325490 < 2.2e-16 ***
PLATE_X                   1   559080     24507   10766410 < 2.2e-16 ***
PLATE_Z                   1      407     24506   10766003  0.323712
INDUCED_VERTICAL_BREAK    1      378     24505   10765625  0.341485
HORIZONTAL_BREAK          1   108229     24504   10657396 < 2.2e-16 ***
VERTICAL_APPROACH_ANGLE   1     3662     24503   10653734  0.003085 **
HORIZONTAL_APPROACH_ANGLE 1     4473     24502   10649261  0.001073 **
OPS                       1    22892     24501   10626369 1.375e-13 ***
Handedness                1   380929     24500   10245440 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

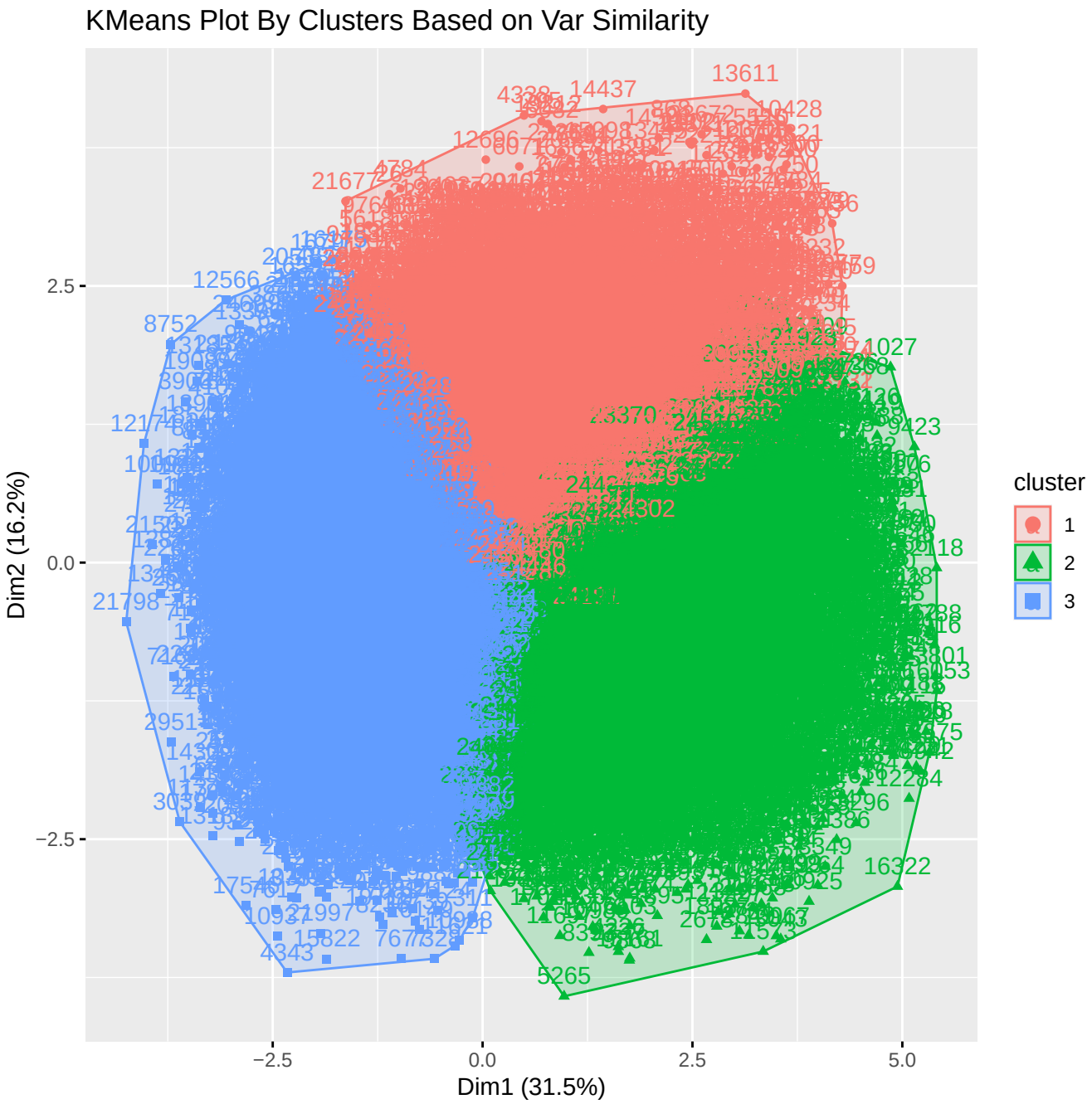
```
evc <- exitVeloCalc$coefficients
ac <- angleCalc$coefficients
dc <- directionCalc$coefficients

#adding new glm predicted columns to training df
trainDF <- trainDF %>%
  mutate(exitVelo_GLM = round(evc[1] + evc[2]*RELEASE_SPEED + evc[3]*PLATE_X + evc[4]*PLATE_Z +
                                evc[5]*INDUCED_VERTICAL_BREAK + evc[6]*HORIZONTAL_BREAK +
                                evc[7]*VERTICAL_APPROACH_ANGLE +
                                evc[8]*HORIZONTAL_APPROACH_ANGLE + evc[9]*OPS + evc[10]*Handedness, 3))


trainDF <- trainDF %>%
  mutate(angle_GLM = round(ac[1] + ac[2]*RELEASE_SPEED + ac[3]*PLATE_X + ac[4]*PLATE_Z +
                             ac[5]*INDUCED_VERTICAL_BREAK + ac[6]*HORIZONTAL_BREAK +
                             ac[7]*VERTICAL_APPROACH_ANGLE +
                             ac[8]*HORIZONTAL_APPROACH_ANGLE + ac[9]*OPS + ac[10]*Handedness, 3))

trainDF <- trainDF %>%
  mutate(direction_GLM = round(dc[1] + dc[2]*RELEASE_SPEED + dc[3]*PLATE_X + dc[4]*PLATE_Z +
                                 dc[5]*INDUCED_VERTICAL_BREAK + dc[6]*HORIZONTAL_BREAK +
                                 dc[7]*VERTICAL_APPROACH_ANGLE +
                                 dc[8]*HORIZONTAL_APPROACH_ANGLE + dc[9]*OPS + dc[10]*Handedness, 3))
```

KMeans

```
subsetTrain <- trainDF[,c(19, 12:18, 29:30)]
km1 <- kmeans(subsetTrain[,2:10], 3, iter.max = 100)

clusters <- factor(km1$cluster)
ggplot(trainDF, aes(x = EXIT_SPEED, y = RELEASE_SPEED, color = clusters)) + geom_point() +
  ggtitle("Release Speed vs Exit Velocity Grouped By Clusters")
```




```
km2 <- kmeans(scale(trainDF[, c(12:18, 29:30)]), 3, nstart = 25, iter.max = 100)
fviz_cluster(km2, data = trainDF[, c(12:18, 29:30)],
             main = "KMeans Plot By Clusters Based on Var Similarity")
```



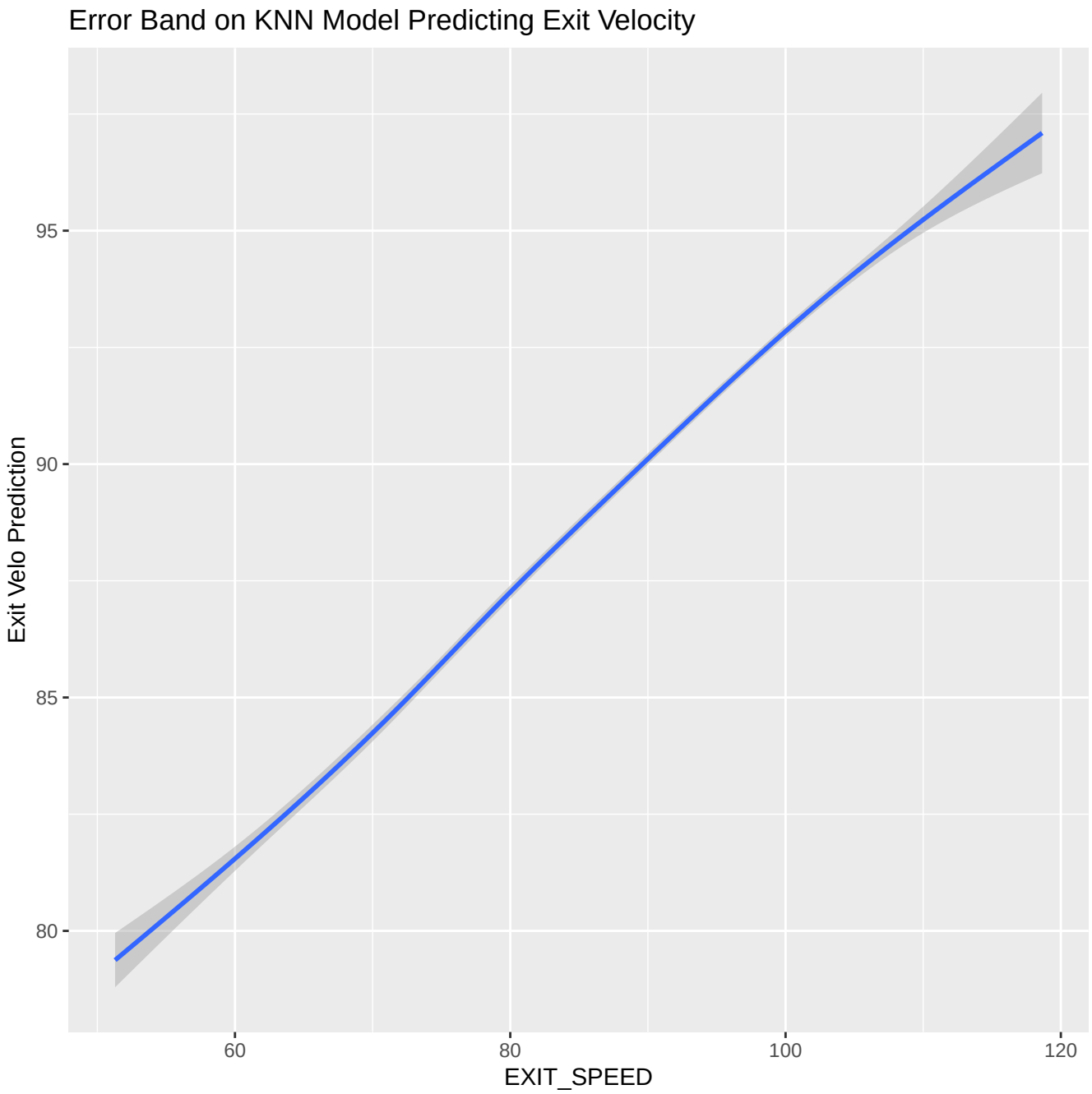

```
# Dimension reduction PCA
pca <- prcomp(trainDF[, c(12:18, 29:30)],  scale = TRUE)
cords <- as.data.frame(get_pca_ind(pca)$coord)
cords$cluster1 <- factor(km2$cluster)
cords$EXIT_SPEED <- trainDF$EXIT_SPEED
head(cords) #see first few rows dimensions and clusters
```

```
       Dim.1        Dim.2      Dim.3      Dim.4      Dim.5      Dim.6
1 -0.2670835 -0.612154122 -1.5278253 -3.0041345 -1.5721867 -0.6846484
2 -1.6181788 -0.685284226 -1.1083727 -1.5976944 -0.0713787  0.5994411
3 -0.7583726  0.615782416  1.7885790 -0.4184670  1.1029973  0.9452969
4  2.1475458  2.563730676 -0.3782569  0.8190211 -0.5348873  0.1488507
5 -1.8329566  0.005665152 -1.9021775  0.1834935  1.6097560 -0.2961044
6  0.4233652  0.720917324  1.3316927 -0.4232247 -0.5920003 -0.2493779
       Dim.7       Dim.8        Dim.9 cluster1 EXIT_SPEED
1  0.3249521  0.01195926  0.181060498        3   83.65304
2  0.3777947 -0.26351196 -0.054952233        3   95.66794
```

```
3  0.9258171  0.37129333 -0.166779325        3   86.94758
4  1.2165254  0.42395082 -0.032647612        1   76.26321
5 -0.2560795  0.26898614  0.009755342        3   87.25558
6  1.1292003 -1.62054318  0.077016925        1   82.83160
```

KNN

```
#######knn on exit velo
trainKNN_x_EV <- trainDF[,c(12:18,29:30)]
trainKNN_y_EV <- trainDF[,19]
knnModel_EV <- knnreg(trainKNN_x_EV, trainKNN_y_EV)
testKNN_x_EV <- testDF[,c(12:20)]

#testdf pred vals
knnpred_y_EV <- predict(knnModel_EV, data.frame(testKNN_x_EV))

knnpred_xTconf_EV <- predict(knnModel_EV, data.frame(trainKNN_x_EV), interval = "confidence", level = 0.9)
ggplot(trainDF, aes(EXIT_SPEED, knnpred_xTconf_EV)) + geom_smooth() +
  ggtitle("Error Band on KNN Model Predicting Exit Velocity") + ylab("Exit Velo Prediction")
```




```
mse <- mean((trainKNN_y_EV - knnpred_xTconf_EV)^2)
mae <- MAE(trainKNN_y_EV, knnpred_xTconf_EV)
rmse <- RMSE(trainKNN_y_EV, knnpred_xTconf_EV)
mse
```

```
[1] 109.4011
```

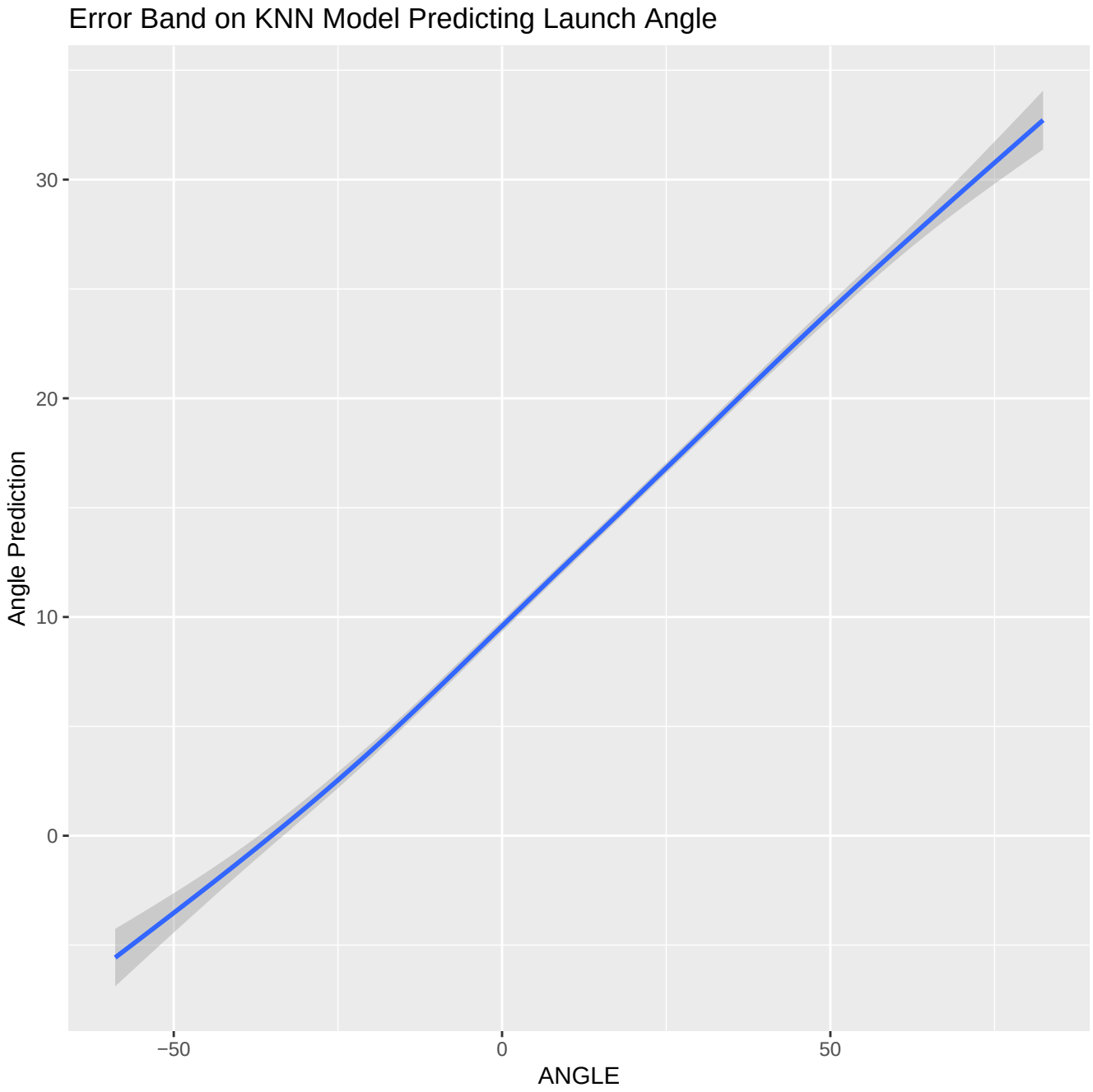
```
mae
```

```
[1] 8.328806
```

```
rmse
```

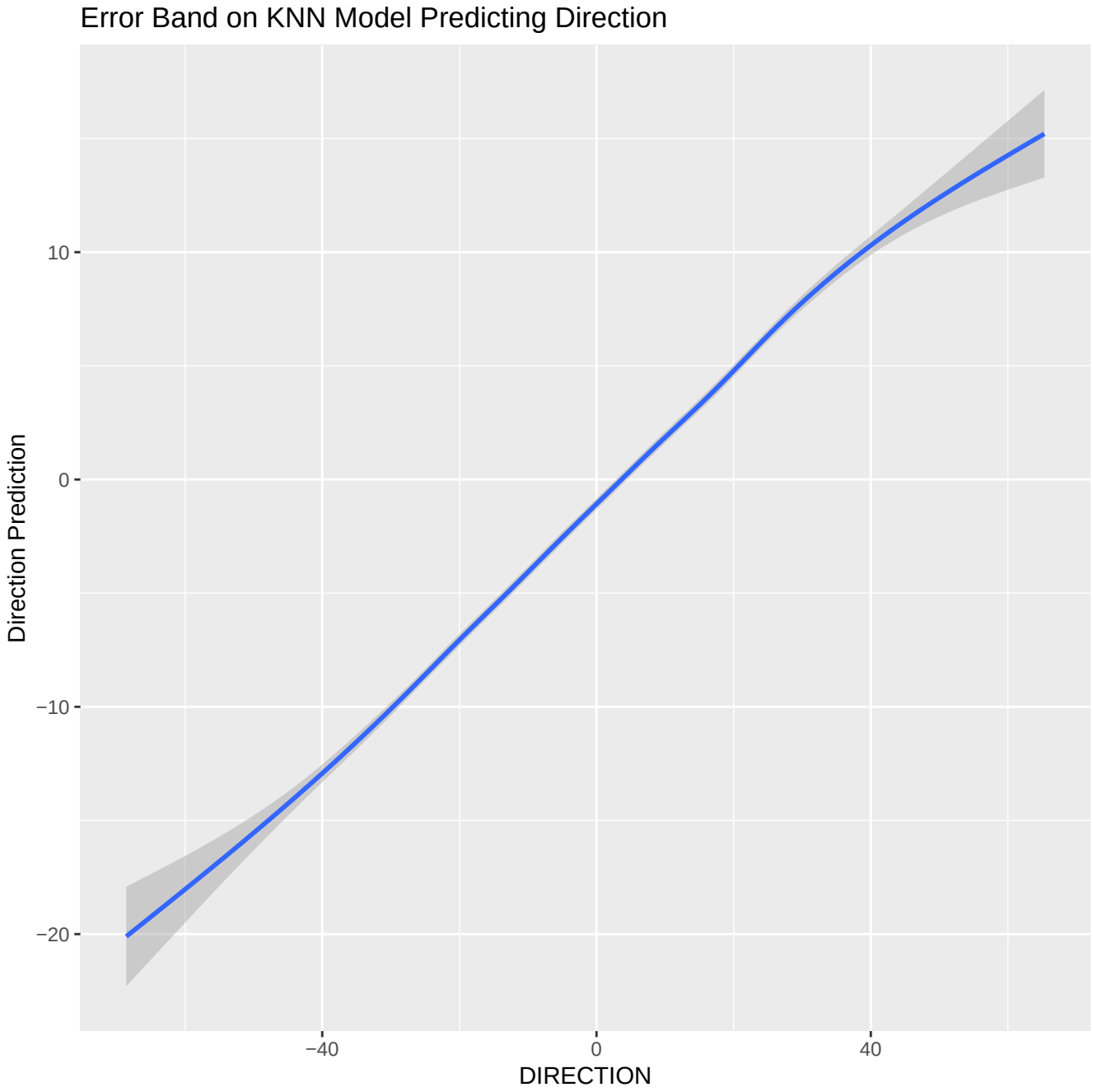
```
[1] 10.4595
```

```
#####knn on angle
trainKNN_x_ang <- trainDF[,c(12:18,29:30)]
trainKNN_y_ang <- trainDF[,20]
knnModel_ang <- knnreg(trainKNN_x_ang, trainKNN_y_ang)
testKNN_x_ang <- testDF[,c(12:20)]

knnpred_y_ang <- predict(knnModel_ang, data.frame(testKNN_x_ang))

knnpred_xTconf_ang <- predict(knnModel_ang, data.frame(trainKNN_x_ang), interval = "confidence", level = 0.9)
ggplot(trainDF, aes(ANGLE, knnpred_xTconf_ang)) + geom_smooth() +
  ggtitle("Error Band on KNN Model Predicting Launch Angle") + ylab("Angle Prediction")
```




```
#####knn on direction
trainKNN_x_dir <- trainDF[,c(12:18,29:30)]
trainKNN_y_dir <- trainDF[,21]
knnModel_dir <- knnreg(trainKNN_x_dir, trainKNN_y_dir)
testKNN_x_dir <- testDF[,c(12:20)]
```

```
knnpred_y_dir <- predict(knnModel_dir, data.frame(testKNN_x_dir))

knnpred_xTconf_dir <- predict(knnModel_dir, data.frame(trainKNN_x_dir), interval = "confidence", level = 0.9)
ggplot(trainDF, aes(DIRECTION, knnpred_xTconf_dir)) + geom_smooth() +
  ggtitle("Error Band on KNN Model Predicting Direction") + ylab("Direction Prediction")
```




```
#add pred vals to training df
trainDF <- trainDF %>% mutate(exitVelo_KNN = round(knnpred_xTconf_EV,3))
trainDF <- trainDF %>% mutate(angle_KNN = round(knnpred_xTconf_ang,3))
trainDF <- trainDF %>% mutate(direction_KNN = round(knnpred_xTconf_dir,3))
```

Finding Absolute Errors on Training DF Models

```
######################
mean(abs(trainDF$EXIT_SPEED - trainDF$exitVelo_RF))
```

```
[1] 4.016803

mean(abs(trainDF$EXIT_SPEED - trainDF$exitVelo_SVM))

[1] 1.266979

mean(abs(trainDF$EXIT_SPEED - trainDF$exitVelo_GAM))

[1] 10.11333

mean(abs(trainDF$EXIT_SPEED - trainDF$exitVelo_GLM))

[1] 10.11355

mean(abs(trainDF$EXIT_SPEED - trainDF$exitVelo_KNN))

[1] 8.328807

mean(abs(trainDF$ANGLE - trainDF$angle_RF))

[1] 8.196231

mean(abs(trainDF$ANGLE - trainDF$angle_SVM))

[1] 2.589948

mean(abs(trainDF$ANGLE - trainDF$angle_GAM))

[1] 19.04803

mean(abs(trainDF$ANGLE - trainDF$angle_GLM))

[1] 19.04803

mean(abs(trainDF$ANGLE - trainDF$angle_KNN))

[1] 16.74011

mean(abs(trainDF$DIRECTION - trainDF$direction_RF))

[1] 7.18387

mean(abs(trainDF$DIRECTION - trainDF$direction_SVM))

[1] 2.210109

mean(abs(trainDF$DIRECTION - trainDF$direction_GAM))

[1] 16.75261

mean(abs(trainDF$DIRECTION - trainDF$direction_GLM))

[1] 16.75257

mean(abs(trainDF$DIRECTION - trainDF$direction_KNN))

[1] 14.50776

######################
```

It can be seen that SVM has the smallest mean error rate

Testing Data

```
##################Testing Data
####gam
#adding gam predicted values to test df
testDF <- testDF %>%
  mutate(exitVelo_GAM = round(gamEV[1] + gamEV[2]*RELEASE_SPEED + gamEV[3]*INDUCED_VERTICAL_BREAK +
                                gamEV[4]*HORIZONTAL_BREAK + gamEV[5]*HORIZONTAL_APPROACH_ANGLE +
                                gamEV[6]*OPS + gamEV[7]*Handedness, 3))

testDF <- testDF %>%
  mutate(angle_GAM = round(gamAng[1] + gamAng[2]*RELEASE_SPEED + gamAng[3]*PLATE_X +
                             gamAng[4]*PLATE_Z + gamAng[5]*INDUCED_VERTICAL_BREAK +
                             gamAng[6]*HORIZONTAL_BREAK + gamAng[7]*VERTICAL_APPROACH_ANGLE +
                             gamAng[8]*HORIZONTAL_APPROACH_ANGLE +
                             gamAng[9]*OPS + gamAng[10]*Handedness, 3))

testDF <- testDF %>%
  mutate(direction_GAM = round(gamDir[1] + gamDir[2]*RELEASE_SPEED + gamDir[3]*PLATE_X +
                                 gamDir[4]*INDUCED_VERTICAL_BREAK + gamDir[5]*HORIZONTAL_BREAK +
                                 gamDir[6]*VERTICAL_APPROACH_ANGLE +
                                 gamDir[7]*HORIZONTAL_APPROACH_ANGLE +
                                 gamDir[8]*OPS + gamDir[9]*Handedness, 3))

####glm
#adding glm predicted values to test df
testDF <- testDF %>%
  mutate(exitVelo_GLM = round(evc[1] + evc[2]*RELEASE_SPEED + evc[3]*PLATE_X + evc[4]*PLATE_Z +
                                evc[5]*INDUCED_VERTICAL_BREAK + evc[6]*HORIZONTAL_BREAK +
                                evc[7]*VERTICAL_APPROACH_ANGLE +
                                evc[8]*HORIZONTAL_APPROACH_ANGLE + evc[9]*OPS +
                                evc[10]*Handedness, 3))


testDF <- testDF %>%
  mutate(angle_GLM = round(ac[1] + ac[2]*RELEASE_SPEED + ac[3]*PLATE_X + ac[4]*PLATE_Z +
                             ac[5]*INDUCED_VERTICAL_BREAK + ac[6]*HORIZONTAL_BREAK +
                             ac[7]*VERTICAL_APPROACH_ANGLE +
                             ac[8]*HORIZONTAL_APPROACH_ANGLE + ac[9]*OPS +
                             ac[10]*Handedness, 3))

testDF <- testDF %>%
  mutate(direction_GLM = round(dc[1] + dc[2]*RELEASE_SPEED + dc[3]*PLATE_X + dc[4]*PLATE_Z +
                                 dc[5]*INDUCED_VERTICAL_BREAK + dc[6]*HORIZONTAL_BREAK +
                                 dc[7]*VERTICAL_APPROACH_ANGLE +
                                 dc[8]*HORIZONTAL_APPROACH_ANGLE + dc[9]*OPS +
                                 dc[10]*Handedness, 3))

###knn
#adding knn predicted values to test df
testDF <- testDF %>% mutate(exitVelo_KNN = round(knnpred_y_EV,3))
testDF <- testDF %>% mutate(angle_KNN = round(knnpred_y_ang,3))
testDF <- testDF %>% mutate(direction_KNN = round(knnpred_y_dir,3))

#View first few rows of my altered df's to see my predicted values for each model
head(trainDF)
```

```
  BATTER_UID    AVG    OBP    SLG VENUE_KEY OUTS BALLS STRIKES BATS_LEFT
1         29 0.2320 0.2860 0.2950      2852    1     2       1         1
2         87 0.2160 0.2820 0.3610      4271    1     2       0         1
```

```
3         20 0.2695 0.3390 0.3835      2528    0     3       2         0
4        147 0.2895 0.3655 0.4510      4670    1     1       2         0
5         99 0.2660 0.3285 0.3560      2852    2     2       2         0
6        110 0.2570 0.3365 0.4250      2852    2     1       2         0
  THROWS_LEFT PITCH_NUMBER RELEASE_SPEED    PLATE_X PLATE_Z
1           0            4      95.11150  0.7963360 1.35507
2           0            3      92.76369  0.4548490 2.59344
3           0            6      88.76340 -0.3001910 3.06310
4           1            4      81.78240 -0.1708880 1.84865
5           1            5      92.06150 -0.1243550 3.24001
6           0            4      81.60239  0.0411996 1.97743
  INDUCED_VERTICAL_BREAK HORIZONTAL_BREAK VERTICAL_APPROACH_ANGLE
1               17.01530        -4.657780                -6.54525
2               17.30840        -8.322809                -4.98211
3                2.46511       -20.656200                -5.81156
4               -3.59941        -7.826159                -8.56734
5               15.57420        10.426299                -4.67467
6               14.52030       -18.717300                -7.20663
  HORIZONTAL_APPROACH_ANGLE EXIT_SPEED      ANGLE  DIRECTION
1                  1.646740   83.65304 -14.851092  13.405146
2                  1.529110   95.66794   3.929680  21.932704
3                  0.266562   86.94758  22.556687  15.621360
4                 -2.952540   76.26321 -13.785541 -46.406194
5                 -1.051080   87.25558   1.316303  14.561501
6                  0.059810   82.83160  59.353564   1.017197
           EVENT_RESULT_KEY PITCH_RESULT_KEY PA X1B X2B X3B HR    OPS
1                 field_out           InPlay  1   0   0   0  0 0.5810
2                    single           InPlay  1   1   0   0  0 0.6430
3                 field_out           InPlay  1   0   0   0  0 0.7225
4 grounded_into_double_play           InPlay  1   0   0   0  0 0.8165
5                    single           InPlay  1   1   0   0  0 0.6845
6                 field_out           InPlay  1   0   0   0  0 0.7615
  Handedness exitVelo_RF angle_RF direction_RF exitVelo_SVM angle_SVM
1          1      83.272   -6.309       14.217       84.905   -12.358
2          1      93.311    8.092       15.756       94.418     6.427
3          0      88.044   18.882       10.823       88.198    20.066
4          1      80.806   -6.580      -26.147       77.517   -16.287
5          1      89.061   14.891        8.066       88.505     3.813
6          0      85.159   44.449        3.064       84.085    56.851
  direction_SVM exitVelo_GAM angle_GAM direction_GAM exitVelo_GLM angle_GLM
1        15.552       88.854     2.399        13.656       88.935     2.399
2        20.501       89.289    16.736        10.981       89.309    16.736
3        17.778       88.816    11.520        -2.975       88.798    11.520
4       -44.246       89.198     3.010        -1.830       89.233     3.010
5        12.408       89.425    21.826        -1.285       89.403    21.826
6        -1.138       88.336    19.072        -3.264       88.377    19.072
  direction_GLM exitVelo_KNN angle_KNN direction_KNN
1        13.630       90.318     7.715        12.695
2        10.980       91.891    17.399         4.056
3        -2.960       93.822    17.837        -7.282
4        -1.835       72.780   -16.943       -22.993
5        -1.265       99.492    21.210        -5.426
6        -3.273       83.394    37.113         9.861
```

```
head(testDF)
```

```
  BATTER_UID    AVG    OBP    SLG VENUE_KEY OUTS BALLS STRIKES BATS_LEFT
1         81 0.2610 0.3285 0.3655      2528    2     1       2         1
2        125 0.2295 0.3255 0.4005      2683    1     1       1         1
3         21 0.2745 0.3365 0.4425      2724    2     2       0         0
```

```
4         73 0.2890 0.3810 0.3740      2772    1     0       1         1
5        142 0.3195 0.3605 0.4435      5472    0     0       1         1
6         81 0.2610 0.3285 0.3655      2843    1     0       2         0
  THROWS_LEFT PITCH_NUMBER RELEASE_SPEED    PLATE_X  PLATE_Z
1           0            6      89.33580 -0.3844000 2.530500
2           0            3      87.97424  0.5956410 2.485661
3           0            3      90.37970 -0.0741844 2.732740
4           1            2      80.51560  0.3506650 1.272440
5           0            2      84.06917  1.0804490 1.934018
6           1            3      90.85476  0.7055900 3.354112
  INDUCED_VERTICAL_BREAK HORIZONTAL_BREAK VERTICAL_APPROACH_ANGLE
1             16.4084988        -4.477510               -4.957750
2              0.6219336        -1.303324               -7.055572
3             15.1757994        -1.611590               -5.580090
4             11.2826996        13.452300               -7.813180
5             -3.2285538        11.446440               -8.635422
6             17.5839863         8.335738               -4.328052
  HORIZONTAL_APPROACH_ANGLE   OPS Handedness exitVelo_GAM angle_GAM
1                  2.421590 0.694          1       89.129    21.616
2                  2.367589 0.726          1       88.768     8.380
3                  2.571330 0.779          0       89.578    20.276
4                 -0.765851 0.755          0       87.357    12.597
5                  3.918451 0.804          1       88.532     6.244
6                 -1.838810 0.694          1       89.488    24.187
  direction_GAM exitVelo_GLM angle_GLM direction_GLM exitVelo_KNN angle_KNN
1         1.513       89.043    21.616         1.494       92.040    17.602
2         9.906       88.813     8.380         9.921       85.722    10.084
3        -5.041       89.568    20.276        -5.028      102.093     8.851
4        -8.883       87.359    12.597        -8.934       88.236    17.053
5        10.737       88.617     6.244        10.750       89.092    16.577
6         6.594       89.520    24.187         6.612       95.074    26.324
  direction_KNN
1        16.051
2        -2.762
3         3.839
4        -5.117
5        -0.403
6        -0.504
```

```
#writing to working directory to display final result tables
# write.csv(trainDF, "myTrainDF.csv")
# write.csv(testDF, "myTestDF.csv")
```

Another approach I though of to run a simulation (but did not get to because of time limit) is this:

Find the distribution of each of the 3 vars based on pitch type (say there are 4 pitch types) from k means and using the batting/pitch info, repeatedly generate outcome result (from model in assessment 2) for each var based on each pitch type and then randomly grab a exit velo, LA, and direction and pull the assigned outcome (out, 1b, 2b, 3b, hr) then randomly runs 1000 times, take the avg, and assign that average to be the exit velo, LA, and direction, respectively, on test data

```
smallSmall <- trainDF[1:5,12:44]

store2 <- list()
for (i in 1:nrow(smallSmall)) {
  storeTest <- sample(smallSmall[i,30:33], 1, prob = abs(c(smallSmall[i,30],
                                                           smallSmall[i,31],
                                                           smallSmall[i,32],
```

```
                                                    smallSmall[i,33])))
  store2[[i]] <- storeTest
}
store2



trainDF <- trainDF %>% mutate(swingSpeed = EXIT_SPEED - (0.2 * trainDF$RELEASE_SPEED) / (1 + 0.2))
newDF <- trainDF %>% group_by(BATTER_UID) %>% summarise(avgSS = mean(swingSpeed),
                                                        n = n())
newDF %>% arrange(desc(avgSS))

hist(newDF$avgSS)

#0.2 is value of wooden
#avg 70mph bat speed according to blast motion
#runif(nrow(trainDF), min = 40, max = 80) represents randomly assigning bat speed
exitVeloFormula <- (0.2 * trainDF$RELEASE_SPEED) + (1 + 0.2)*runif(nrow(trainDF), min = 40, max = 80)
summary(exitVeloFormula)
round(asin(trainDF$PLATE_X / exitVeloFormula*(21.922)),3)
```