Final Project OPR/STA 9750

Yasmine Ben-Brahim & Tetsu Higuchi

5/23/2022

Section 1: Introduction and Background

We chose to do our project on whether or not there were certain variables of a baseball game that could tell us whether or not a ball-in-play would result in a home run. Home run's are one of the most exciting moments across all of sports. Our research hypothesis is that a home run doesn't just happen by chance, and that there must be certain factors of the pitch that is thrown that could make a home-run more likely.

We found our dataset on Kaggle.com, titled "Baseball - MLB Predict Home Runs." We will be using this data set to find the best predictors indicative of a home run by analyzing the key baseball statistics provided. All of this data is compiled from all US Major League Baseball (MLB) games played between the months of July 2021 - December 2021.

Although the data set included 4 CSV files, we used two of them for our research. One is called "park_dimensions", which included data on the actual ball parks where the baseball games were played, such as distances and heights the park walls. The other file we used is called "train", which included the bulk of the variables we wanted to analyze, as well as the binary variable "is_home_run", which indicated whether or not the ball hit resulted in a home run or not.

We loaded the appropriate libraries, and created a merged data set called "baseball", where we merged the two CSV files we chose.

```
library(tidyverse)
library(caTools)
library(rpart.plot)
library(stargazer)

park_dims = read.csv('park_dimensions.csv')
train = read.csv('train.csv')
baseball = merge(train,
                 park_dims,
                 by.x = 'park',
                 by.y = 'park')
```

Section 2: Data Description and Variable Selection

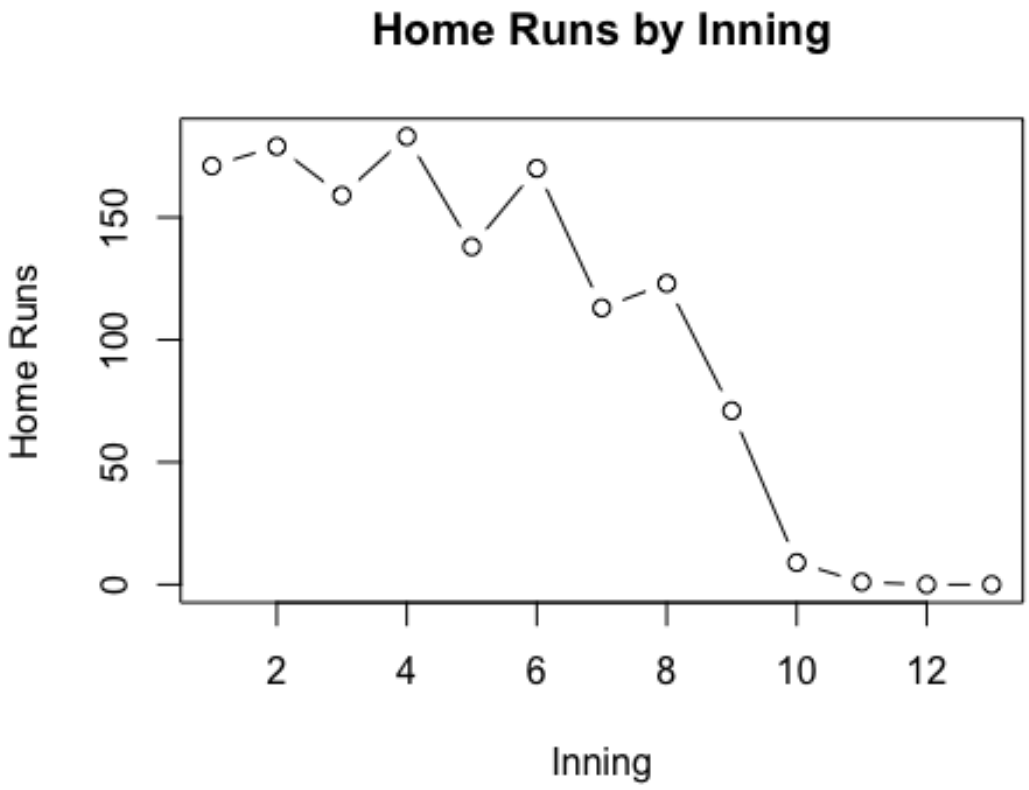
Originally, the data set contained 46,244 rows (or observations) and 27 columns which included 11,805 null values in “launch_speed” and 11,785 null values in “launch_angle”. After removing the null values and the variables we weren’t interested in, we lost a little over 20,300 observations. However, we were still left with 25,944 observations, which we felt was still plenty enough to make substantiated analysis.

```
baseball = baseball[,c(1,2,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
                       26,27,28,29,30,31,32,33,25)]
baseball = na.omit(baseball)
```

Section 3: Identifying Relationships and Trends

##Graph 1.1: Home Runs vs Inning

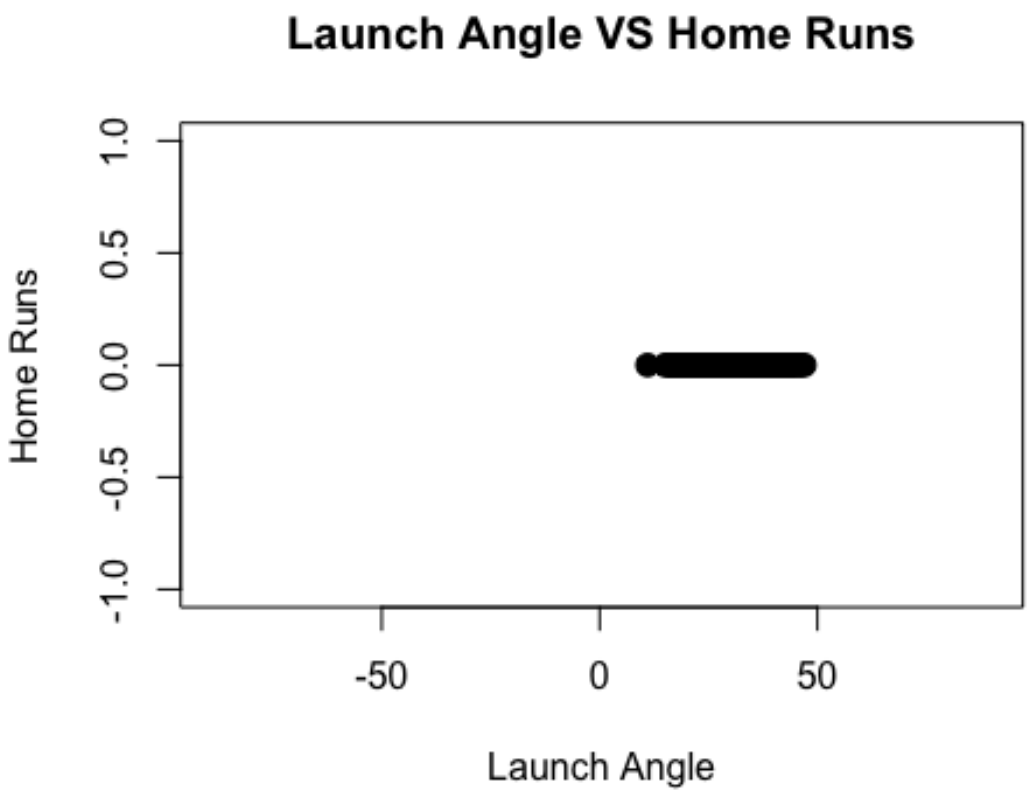
```
aggr.by.inning = aggregate(baseball$is_home_run,
                           by=list(baseball$inning),
                           FUN=sum)
plot(aggr.by.inning, type='b', main='Home Runs by Inning',
     xlab= 'Inning', ylab='Home Runs')
```



We first wanted to run a couple of analysis and plots to help us gain a better understanding of the variables we chose to keep, and explore some of the relationships between those variables and home runs. The first relationship we plotted was home runs by inning. To our surprise, we found out that most of the home runs hit in our data set within the first 4 innings of a game (out of the usual 9, although games can go on longer than 9 innings if the game is at a tie by the end of the 9th inning.) We concluded that the majority of home runs were hit within the first 4 innings because players were probably the strongest and least tired during them, as opposed to later innings.

##Graph 1.2: Home Runs vs Launch Angle

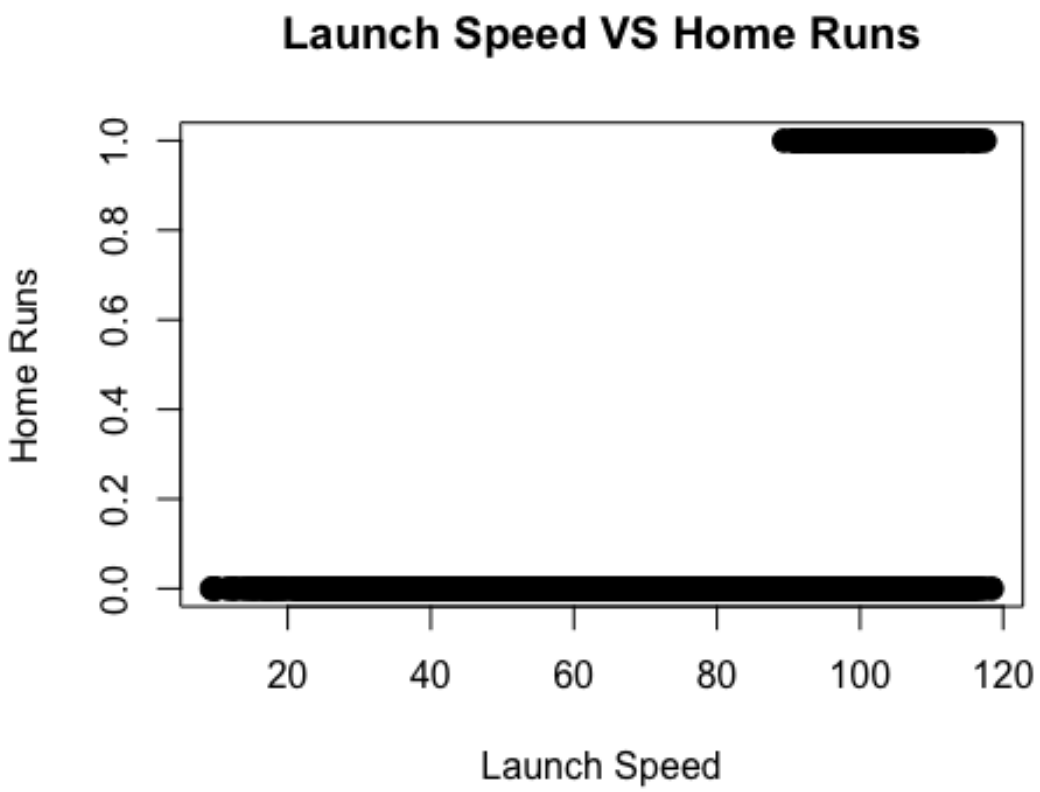
```
plot(log(baseball$is_home_run)~ baseball$launch_angle, data = baseball,
     pch = 20, cex = 2, col = 'black', main = 'Launch Angle VS Home Runs',
     xlab= 'Launch Angle', ylab='Home Runs')
```



The second relationship we plotted was launch angle (at what angle did the ball hit leave the bat) and home runs. Our plot shows that all home runs hit in our data set had a launch angle between around 10 degrees to 50 degrees.

##Graph 1.3: Home Runs vs Launch Speed

```
plot(baseball$is_home_run~ baseball$launch_speed, data = baseball,
     pch = 20, cex = 2, col = 'black', main = 'Launch Speed VS Home Runs',
     xlab= 'Launch Speed', ylab='Home Runs')
```



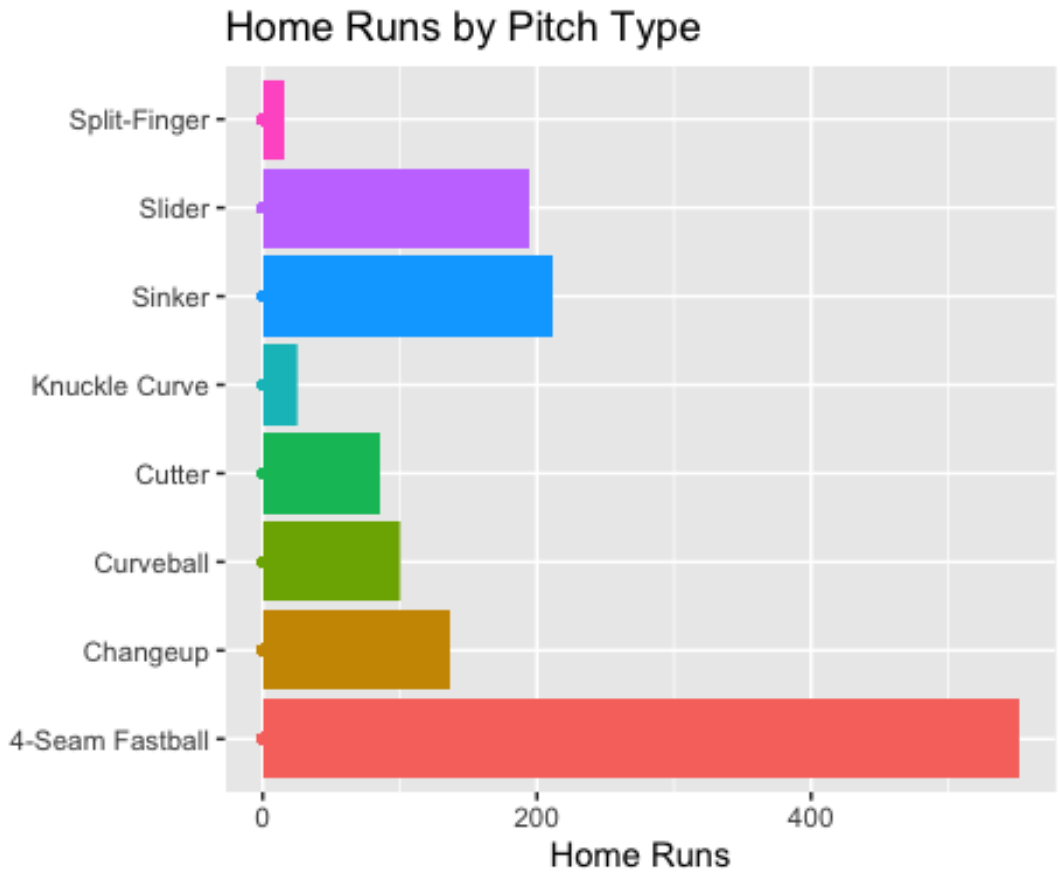
The third relationship we plotted was launch speed (at what speed (in mph) did the ball hit leave the bat) and home runs. Our plot shows that all home runs hit in our data set had a launch speed that was greater than 85mph.

After running the plots for launch angle and launch speed, we determined that these two variables were the most important indicators of a home run, and decided to use them in our decision tree, which was one of the main models for our project.

We can infer from the two plots above that the ball hit has a better chance to be a home run when the ball is launched at an angle between 10 and 50 degrees and the launch speed is greater than 85 mph.

##Graph 1.4: Home Runs vs Pitch

```
ggplot(baseball, aes(baseball$is_home_run, baseball$pitch_name,
                         fill=baseball$pitch_name)) +
  geom_point(aes(col=pitch_name), size=1) +
  geom_col() +
  theme(legend.position = 'none') +
  labs(title = "Home Runs by Pitch Type",
        y= "",
       x= "Home Runs")
```



Since our merged data set "baseball" contains 23 variables, we wanted to analyze a few more relationships before diving into our main models. The next relationship we plotted was home runs by pitch type. A pitch type refers to what type of pitch was thrown, whether it was a slider, a curve ball, a fast ball, etc. Our plot shows that nearly 600 home runs in our data set was hit off of a fast ball, and was clearly the majority by far. The runner ups for pitch type were sinkers (a little over 200 home runs) and sliders (a little below 200 home runs). This intuitively made sense, because a fast ball is a fairly straightforward pitch, without any curves or surprises. It also made intuitive sense that sinkers and sliders were runner ups, since these pitches are usually thrown slower, so if the batter swings correctly, they have a good chance of hitting the ball hard and at the right angle.

##Graph 1.5: Pitch Speed vs Launch Speed of Home Runs

```
home_runs = subset(baseball, baseball$is_home_run==1)

ggplot(home_runs, aes(x=home_runs$pitch_mph,
                      y=home_runs$launch_speed,
                      shape = Cover,
                      color = Cover,
                      fill= Cover)) +
  geom_point(aes(col=Cover), size=1) +
  geom_smooth(method="lm", color="black") +
  labs(title = "Scatterplot: Linear Relationship",
```

```
       subtitle = "Pitch Speed vs Launch Speed of Home Runs",
       y= "Launch Speed",
       x= "Pitch MPH")
```

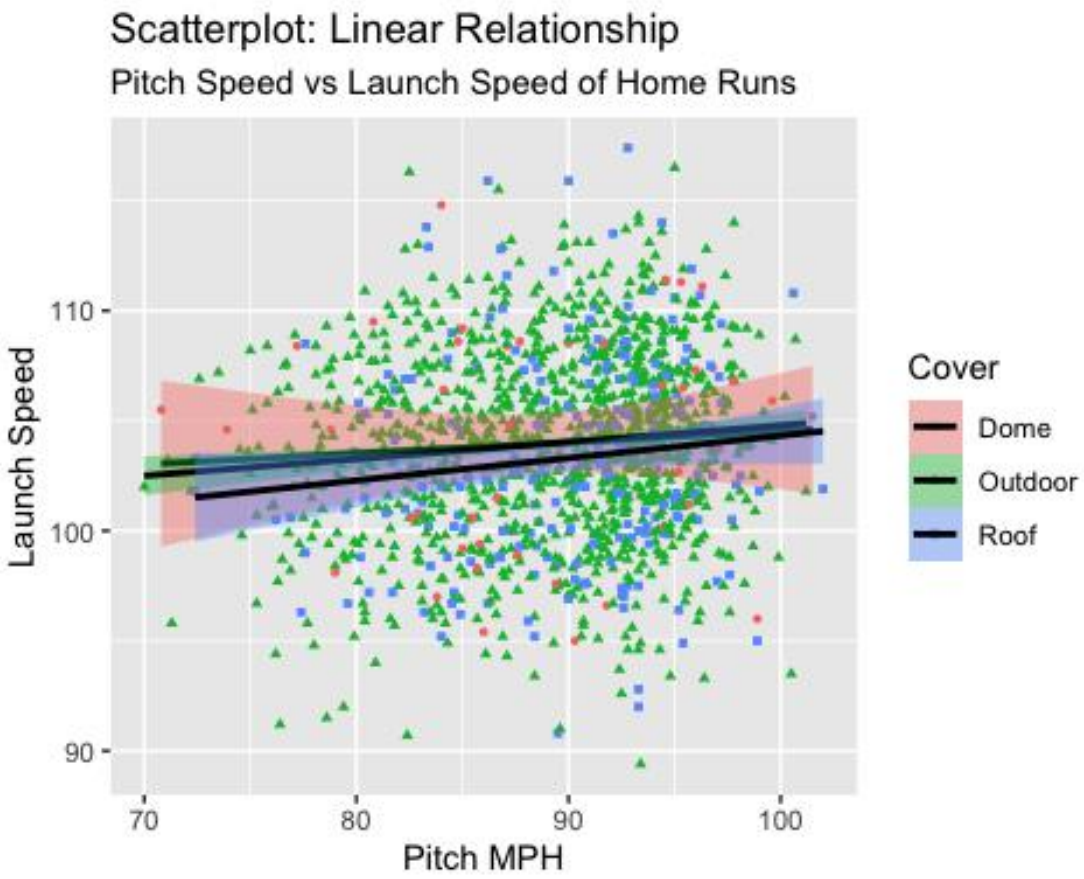
```
## Warning: Use of `home_runs$pitch_mph` is discouraged. Use `pitch_mph`
instead.

## Warning: Use of `home_runs$launch_speed` is discouraged. Use
`launch_speed`
## instead.

## Warning: Use of `home_runs$pitch_mph` is discouraged. Use `pitch_mph`
instead.

## Warning: Use of `home_runs$launch_speed` is discouraged. Use
`launch_speed`
## instead.

## `geom_smooth()` using formula 'y ~ x'
```



We also wanted to plot the relationship between pitch speed (the speed at which the ball is thrown) and launch speed of home run balls. We added an additional layer of analysis by adding a fill color for each type of ball park cover, which were either domed, outdoor (open), or roofed.

We found a lot of information on this plot. We saw that there were a lot more home runs hit in outdoor parks. We also found a slightly positive slope (relationship) between pitch speed and launch speed, meaning that the faster the pitch was thrown at, the faster the ball traveled off the bat. Launch speeds were all over 85mph, which further strengthens our assumption in section 1.3.

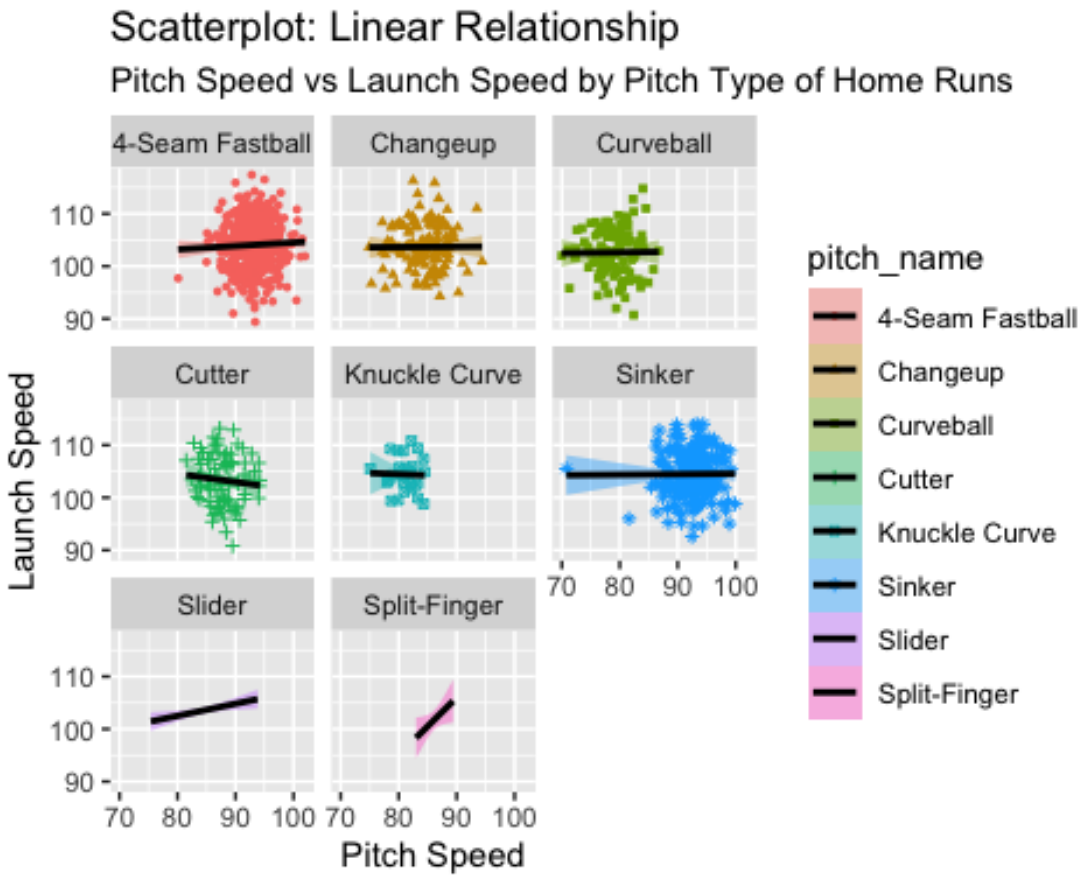
##Graph 1.6: Pitch Speed vs Launch Speed by Pitch Type of Home Runs

```
ggplot(home_runs, aes(x=pitch_mph,
                      y=launch_speed,
                      shape = pitch_name,
                      color = pitch_name,
                      fill= pitch_name)) +
  geom_point(aes(col=pitch_name), size=1) +
  geom_smooth(method="lm", color="black") +
  labs(title = "Scatterplot: Linear Relationship",
       subtitle = "Pitch Speed vs Launch Speed by Pitch Type of Home Runs",
       y= "Launch Speed",
       x= "Pitch Speed") +
  facet_wrap(~pitch_name)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
because
## more than 6 becomes difficult to discriminate; you have 8. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 209 rows containing missing values (geom_point).
```



Lastly, we wanted to analyze the relationship between pitch speed and launch speed, while considering the type of pitch thrown. This analysis gave us some great information on how fast certain types of pitches are usually thrown. For example, a “sinker” pitch could range anywhere between 70 to 100 mph. The fast ball was unsurprisingly the fastest type of pitch thrown, and the knuckle curve is the slowest type of pitch thrown. It was also interesting to see the slope of launch speeds for home runs depending on the type and speed of pitch thrown. Most of the slopes were positive. However, the cutter and knuckle curve pitches had negative slopes. This means that the faster these types of pitches were thrown, the slower the launch speeds for balls that resulted in home runs.

Section 4: Modeling

```
set.seed(23)

ind = sample(2, nrow(baseball),
             replace=TRUE,
             prob=c(0.67, 0.33))
baseball.training = baseball[ind==1, 1:26]
baseball.test = baseball[ind==2, 1:26]
baseball.trainLabels = baseball[ind==1, 27]
baseball.testLabels = baseball[ind==2, 27]
```

For our analysis, we randomly split 33% of the total 25,866 observations into a test dataset and the remaining 67% into a training set using a designated seed number. Our test set contains 8641 observations while the training set each contains 17,225 observations. Placing data into train or test sets relies on random numbers, so we set the seed to a fixed value to ensure reproducibility of our results. As expected, our two sets were very similar. In the training set, 848 out of the 17,225 observations recorded a home run being hit, 4.92% of all events. In the test set 469 home runs were recorded out of the 8,641 observations, 5.43% of all events.

Logistic Regression

```
trainingWithLabel = baseball.training
trainingWithLabel$is_home_run = baseball.trainLabels

logisticModel = glm(is_home_run~ .,
                    data=trainingWithLabel,
                    family='binomial')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

prediction = predict(logisticModel,
                     baseball.test,
                     type='response')

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

baseball.test$predicted = ifelse(prediction>0.7, 1, 0)

(confusionMatrix =
    table(Actual_Value = baseball.testLabels,
          Predicted_Value = prediction>0.7))

##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  8155   17
##            1   411   58
```

```
sensitivity = function(cm) {
  return(cm[1,1]/(cm[1,1]+cm[1,2]))
}
specificity = function(cm) {
  return(cm[2,2]/(cm[2,1]+cm[2,2]))
}
accuracy = function(cm) {
  return((cm[1,1]+cm[2,2])/(cm[1,1]+cm[1,2]+cm[2,1]+cm[2,2]))
}

sensitivity(confusionMatrix)

## [1] 0.9979197

specificity(confusionMatrix)

## [1] 0.1236674

accuracy(confusionMatrix)

## [1] 0.9504687
```

In our first logistic regression model, we include all predictors out of the list we chose above and train the model on the train set with labels. The sensitivity of the model, or its ability to determine the home runs correctly, is 99.79%. The specificity of the model, which measures how the test is effective when used on negative cases, is 12.37%. And the accuracy of the model is 95.05%. At first glance, we can see that the model performed strongly with the exception of the low specificity rate. However, since we are more interested in predicting home runs rather than negative cases, we aren't too worried about this. Additionally, since we included all of our chosen variables there are many predictors that are not statistically significant for this logistic regression. In fact, the results of the initial regression model show that eleven of the sixty-five variables are statistically significant where the p-values are less than the significant level, $\boldsymbol{\alpha} = 0.1$. Note that the sixty-five variables contain the dummy variables for all baseline categorical variables

Logistic Regression with Backwards Selection

```
logisticModelBS = glm(is_home_run~ pitcher_id + bearing + inning +
                        balls + strikes + pitch_mph + launch_angle +
                        launch_speed,
                      data=trainingWithLabel,
                      family='binomial')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

prediction = predict(logisticModelBS,
                     baseball.test,
                     type='response')
baseball.test$predicted = ifelse(prediction>0.7, 1, 0)

(confusionMatrix =
```

```
    table(Actual_Value = baseball.testLabels,
          Predicted_Value = prediction>0.7))
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  8159   13
##            1   425   44
```

```
# Summary Metrics
sensitivity(confusionMatrix)
```

```
## [1] 0.9984092
```

```
specificity(confusionMatrix)
```

```
## [1] 0.09381663
```

```
accuracy(confusionMatrix)
```

```
## [1] 0.9493114
```

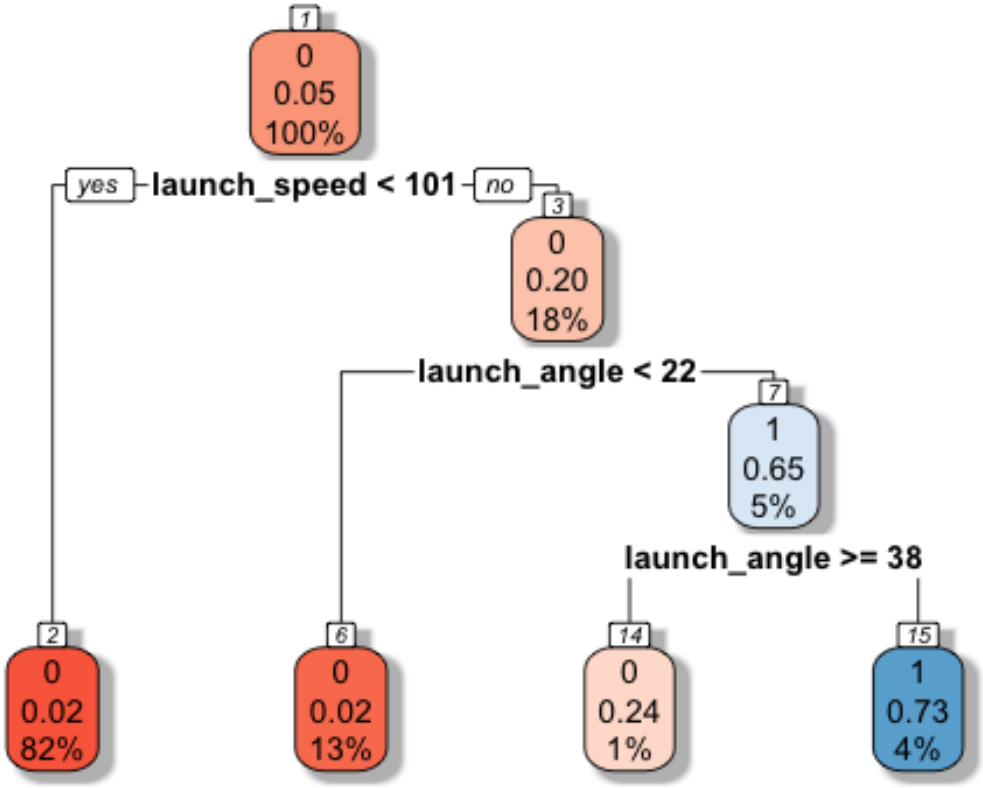
Our second logistic regression model is adapted from our first model after performing backward selection and using only the significant features from the first model. The sensitivity of the second model is 99.84%, the specificity is 9.38% and the accuracy of the model is 94.93%. This model performs a bit worse than our first one however the sensitivity slightly improved and all features are statistically significant except balls and pitcherid.

Decision Tree

```
model = rpart(is_home_run~ pitcher_id + bearing + inning +
                balls + strikes + pitch_mph + launch_angle +
                launch_speed,
              data=trainingWithLabel,
              control=rpart.control(maxdepth=3),
              method='class')

prediction = predict(model,
                     baseball.test,
                     type='class')

library(rpart.plot)
rpart.plot(model, box.palette="RdBu", shadow.col="gray", nn=TRUE)
```




```
(confusionMatrix =
    table(Actual_Value = baseball.testLabels,
          Predicted_Value = prediction))

##              Predicted_Value
## Actual_Value    0    1
##            0 8077   95
##            1  166  303

sensitivity(confusionMatrix)

## [1] 0.9883749

specificity(confusionMatrix)

## [1] 0.6460554

accuracy(confusionMatrix)

## [1] 0.9697952
```

Lastly, we run a decision tree model with the same variables from our second logistic regression model. The decision tree performed very well across the board with 98.84% sensitivity, 64.61% specificity, and 96.98% accuracy. The tree has 7 nodes where the most

important variables, launch angle and launch speed, are used as the main determinants of the tree flow. When the launch speed is less than 101 mph, there is an 82% probability that the ball will not be a home run and an 18% chance that it will be. Given the launch speed is faster than 101 mph, there is a 13% chance that the ball will not be a home run if the launch angle is less than 22 degrees and a 5% chance that it will be if the ball is hit at a launch angle greater than 22 degrees. Given the launch speed is greater than 101 mph and the launch angle is between 22 and 38 degrees, there's a 4% chance it will be a home run and a 1% chance that it won't be if the launch angle is greater than 38 degrees. Therefore, we can assume that the best likeness for a home run is if the ball is hit faster than 101 mph and at a launch angle between 22 and 38 degrees.

Section 5: Conclusion

Our team learned a lot about the specific variables that surround any ball-in-play of a baseball play. By using various plots, logistic regression, and the decision tree, we were able to slice this massive data set of 46,244 original observations and 25 variables to analyze relationships that helped us pinpoint which variables were the best indicators of a home run. Although we concluded that launch speed and launch angles were the two most important variables to look at, there were also other variables such as pitch type and pitch speed that produced sub-layers of analysis.

Now we can ask the question of how might this research be used in real-life? Coaches and teams all over the world can analyze this data in order to train their players on how to hit more home runs and train their pitchers on how to avoid more home runs. Baseball is a sport of numbers, from speed, angles, to distance, and the more data we have on these various variables, the better prepared teams, coaches, and players can be for when it comes to game time.

Thank you for your time in reading our research!