Rays

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Loading Libraries and Data

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'purrr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggpubr)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(readr)
tinytex::install_tinytex(force = TRUE)
```

```
## The directory /usr/local/bin is not writable. I recommend that you make it writable. See https://gitl

## tlmgr install tlgpg

## tlmgr update --self

## tlmgr install tlgpg

## tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg

## tlmgr option repository 'https://mirror.mwt.me/ctan/systems/texlive/tlnet'

## tlmgr update --list
```

```
battedBallData <- read_csv("~/Desktop/battedBallData.csv")
```

```
## Rows: 73375 Columns: 7

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): hittype
## dbl (6): batter, pitcher, speed_A, vangle_A, speed_B, vangle_B
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Rays <- battedBallData
```

Grouping by Batter

```
Rays_grouped_batter <- Rays %>%
  group_by(batter) %>%
  summarize(
    Avg_speed_a = mean(speed_A, na.rm = TRUE),
    Avg_speed_b = mean(speed_B, na.rm = TRUE),
    Avg_angle_a = mean(vangle_A, na.rm = TRUE),
    Avg_angle_b = mean(vangle_B, na.rm = TRUE),
    'fly_ball%' = round(100*sum(ifelse(hittype == 'fly_ball', 1, 0), na.rm = TRUE)/n(), 1),
    'ground_ball%' = round(100*sum(ifelse(hittype == 'ground_ball', 1, 0), na.rm = TRUE)/n(), 1),
    'line_drive%' = round(100*sum(ifelse(hittype == 'line_drive', 1, 0), na.rm = TRUE)/n(), 1),
    'pop_up%' = round(100*sum(ifelse(hittype == 'popup', 1, 0), na.rm = TRUE)/n(), 1),
    BBEs = n()
  )
```

Grouping by Hittype

```
Rays_grouped_hittype <- Rays %>%
  group_by(hittype) %>%
  summarize(
    Avg_speed_a = mean(speed_A, na.rm = TRUE),
    Avg_speed_b = mean(speed_B, na.rm = TRUE),
    Avg_angle_a = mean(vangle_A, na.rm = TRUE),
    Avg_angle_b = mean(vangle_B, na.rm = TRUE),
    count = n()
  ) %>%
  filter(hittype != "U")
```

Making a New Dataset with Fangraphs 2021 and 2022 data

```
FanGraphs_Leaderboard_15_ <- read_csv("~/Desktop/FanGraphs Leaderboard (15).csv")
```

```
## Rows: 932 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Name, Team, IFFB%, LD%, GB%, FB%
## dbl (5): Season, G, EV, LA, playerid
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
FG_22 <- FanGraphs_Leaderboard_15_ %>%
  filter(Season == 2022) %>%
  mutate(`LD%` = as.numeric(sub("%","", `LD%`)),
         `IFFB%` = as.numeric(sub("%","", `IFFB%`)),
         `FB%` = as.numeric(sub("%","", `FB%`)),
         `GB%` = as.numeric(sub("%","", `GB%`)),
         )



FG_21_EV <- FanGraphs_Leaderboard_15_ %>%
  filter(Season == 2021) %>%
  summarize(Name = Name,
    EV_21 = EV)

FG_joined <- merge(FG_22, FG_21_EV, by = "Name")

head(FG_joined)
```

```
##           Name Season Team   G   EV   LA IFFB%  LD%  GB%  FB% playerid EV_21
## 1 A.J. Pollock   2022  CHW 138 88.8 11.8  17.7 19.0 45.2 35.8     9256  90.3
## 2  Aaron Hicks   2022  NYY 130 87.9 14.8  14.8 16.4 44.4 39.3     5297  89.3
## 3  Aaron Judge   2022  NYY 157 95.8 14.9   4.6 19.3 37.3 43.5    15640  95.8
## 4 Abraham Toro   2022  SEA 109 87.0 16.7   9.9 14.4 39.5 46.0    19844  86.8
## 5  Adam Duvall   2022  ATL  86 88.3 23.4  12.2 18.6 29.3 52.1    10950  89.8
## 6   Adam Engel   2022  CHW 119 87.9 15.0  13.6 21.4 39.3 39.3    15082  86.8
```

Creating a Prediction Algorithm for Next Year EV

```
goodColumns <- c('LA', 'IFFB%', 'LD%', 'GB%', 'FB%', 'EV_21', 'EV')

inTrain <- createDataPartition(FG_joined$EV,p=0.7,list = FALSE)
training <- FG_joined[inTrain,goodColumns]
testing <- FG_joined[-inTrain,goodColumns]

method = 'lm'

ctrl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)
modelFit <- train(EV ~ ., method = method, data = training, trControl = ctrl)

summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.135 -1.019  0.066  1.024  3.879
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -54.16120  174.34962  -0.311  0.75634
## LA             -0.21450    0.07884  -2.721  0.00699 **
## `\\`IFFB%\\``  -0.05965    0.02426  -2.459  0.01463 *
## `\\`LD%\\``     0.87653    1.74234   0.503  0.61537
## `\\`GB%\\``     0.76940    1.74606   0.441  0.65986
## `\\`FB%\\``     0.94345    1.74262   0.541  0.58873
## EV_21           0.67976    0.04721  14.399  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 243 degrees of freedom
## Multiple R-squared:  0.5499, Adjusted R-squared:  0.5388
## F-statistic: 49.48 on 6 and 243 DF,  p-value: < 2.2e-16
```

```
model2 <- train(EV ~ EV_21 + LA, method=method, data=training, trControl=ctrl)

summary(model2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2551 -0.8996 -0.0424  1.0315  3.5116
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.45711    4.11542   5.457 1.18e-07 ***
## EV_21        0.74243    0.04626  16.049  < 2e-16 ***
## LA           0.00299    0.02077   0.144    0.886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.565 on 247 degrees of freedom
## Multiple R-squared:  0.5111, Adjusted R-squared:  0.5071
## F-statistic: 129.1 on 2 and 247 DF,  p-value: < 2.2e-16
```

```
model3 <- train(EV ~ EV_21 + `IFFB%`, method=method, data=training, trControl=ctrl)

summary(model3)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5216 -1.0047 -0.0132  0.9835  3.7596
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   25.42480    4.16080   6.111 3.84e-09 ***
## EV_21          0.71672    0.04624  15.501  < 2e-16 ***
## `\\`IFFB%\\`` -0.06406    0.02144  -2.987   0.0031 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.538 on 247 degrees of freedom
## Multiple R-squared:  0.5281, Adjusted R-squared:  0.5243
## F-statistic: 138.2 on 2 and 247 DF,  p-value: < 2.2e-16
```

Preparing for Replacing NA's

```
Rays <- Rays %>%
  filter(!is.na(speed_A) | !is.na(speed_B))

table(is.na(Rays$speed_B))
```

```
##
## FALSE  TRUE
## 71973   860
```

```
table(is.na(Rays$speed_A))
```

```
##
## FALSE  TRUE
## 65803  7030
```

Replacing NA's in speed A

```
Rays_A <- Rays %>%
  filter(is.na(speed_A) != TRUE,
         is.na(speed_B) != TRUE,
         hittype != 'U')


goodColumns <- c('speed_B', 'vangle_B', 'speed_A')

inTrain <- createDataPartition(Rays_A$speed_A,p=0.7,list = FALSE)
training <- Rays_A[inTrain,goodColumns]
testing <- Rays_A[-inTrain,goodColumns]

method = 'lm'

ctrl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)
modelFit <- train(speed_A ~ ., method = method, data = training, trControl = ctrl)

summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -63.049  -3.771  -0.034   3.187  40.748
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 32.777326   0.151241   216.7   <2e-16 ***
## speed_B      0.728813   0.001936   376.4   <2e-16 ***
## vangle_B    -0.197501   0.001531  -129.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.462 on 45459 degrees of freedom
## Multiple R-squared:  0.7576, Adjusted R-squared:  0.7576
## F-statistic: 7.105e+04 on 2 and 45459 DF,  p-value: < 2.2e-16
```

```
Rays_test <- Rays %>%
  mutate(
    speed_A = ifelse(is.na(speed_A), 32.73 + 0.72*speed_B - 0.196*vangle_B, speed_A)
  )

table(is.na(Rays_test$speed_A))
```

```
##
## FALSE
## 72833
```

Replacing NA's in speed B

```
Rays_A <- Rays %>%
  filter(is.na(speed_A) != TRUE,
         is.na(speed_B) != TRUE,
         hittype != 'U')


goodColumns <- c('speed_A', 'vangle_A', 'speed_B')

inTrain <- createDataPartition(Rays_A$speed_B,p=0.7,list = FALSE)
training <- Rays_A[inTrain,goodColumns]
testing <- Rays_A[-inTrain,goodColumns]

method = 'lm'

ctrl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)
modelFit <- train(speed_B ~ ., method = method, data = training, trControl = ctrl)

summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.543  -4.023   0.890   4.832  75.793
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.56001    0.22600  -51.15   <2e-16 ***
## speed_A       0.99800    0.00254  392.93   <2e-16 ***
## vangle_A      0.28537    0.00140  203.90   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.075 on 45459 degrees of freedom
## Multiple R-squared:  0.8266, Adjusted R-squared:  0.8266
## F-statistic: 1.084e+05 on 2 and 45459 DF,  p-value: < 2.2e-16
```

```
Rays_test <- Rays_test %>%
  mutate(
    speed_B = ifelse(is.na(speed_B), -11.638 + 0.998*speed_A - 0.286*vangle_A, speed_B)
  )

table(is.na(Rays_test$speed_B))
```

```
##
## FALSE
## 72833
```

Replacing NA's in vangle A

```
Rays_A <- Rays %>%
  filter(is.na(speed_A) != TRUE,
         is.na(speed_B) != TRUE,
         hittype != 'U')


goodColumns <- c('speed_B', 'vangle_B', 'vangle_A')

inTrain <- createDataPartition(Rays_A$vangle_A,p=0.7,list = FALSE)
training <- Rays_A[inTrain,goodColumns]
testing <- Rays_A[-inTrain,goodColumns]

method = 'lm'

ctrl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)
modelFit <- train(vangle_A ~ ., method = method, data = training, trControl = ctrl)

summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -114.860   -1.433    0.075    2.053   81.622
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.283710   0.126215  -129.0   <2e-16 ***
## speed_B       0.181259   0.001616   112.2   <2e-16 ***
## vangle_B      1.014678   0.001279   793.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.397 on 45459 degrees of freedom
## Multiple R-squared:  0.9487, Adjusted R-squared:  0.9487
## F-statistic: 4.201e+05 on 2 and 45459 DF,  p-value: < 2.2e-16
```

```
Rays_test <- Rays_test %>%
  mutate(
    vangle_A = ifelse(is.na(vangle_A), -16.408 + 0.182*speed_B + 1.105*vangle_B, vangle_A)
  )

table(is.na(Rays_test$vangle_A))
```

```
##
## FALSE
## 72833
```

Replacing NA's in vangle B

```
Rays_A <- Rays %>%
  filter(is.na(speed_A) != TRUE,
         is.na(speed_B) != TRUE,
         hittype != 'U')


goodColumns <- c('speed_B', 'vangle_A', 'vangle_B')

inTrain <- createDataPartition(Rays_A$vangle_B,p=0.7,list = FALSE)
training <- Rays_A[inTrain,goodColumns]
testing <- Rays_A[-inTrain,goodColumns]

method = 'lm'

ctrl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)
modelFit <- train(vangle_B ~ ., method = method, data = training, trControl = ctrl)

summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.962  -2.056  -0.182   1.677 109.566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.003209   0.124960  104.06   <2e-16 ***
## speed_B     -0.131806   0.001606  -82.08   <2e-16 ***
## vangle_A     0.920288   0.001146  802.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.08 on 45459 degrees of freedom
## Multiple R-squared:  0.9443, Adjusted R-squared:  0.9443
## F-statistic: 3.851e+05 on 2 and 45459 DF,  p-value: < 2.2e-16
```

```
Rays_test <- Rays_test %>%
  mutate(
    vangle_B = ifelse(is.na(vangle_B), 13.017 + -0.132*speed_B + 0.921*vangle_A, vangle_B)
  )

table(is.na(Rays_test$vangle_B))
```

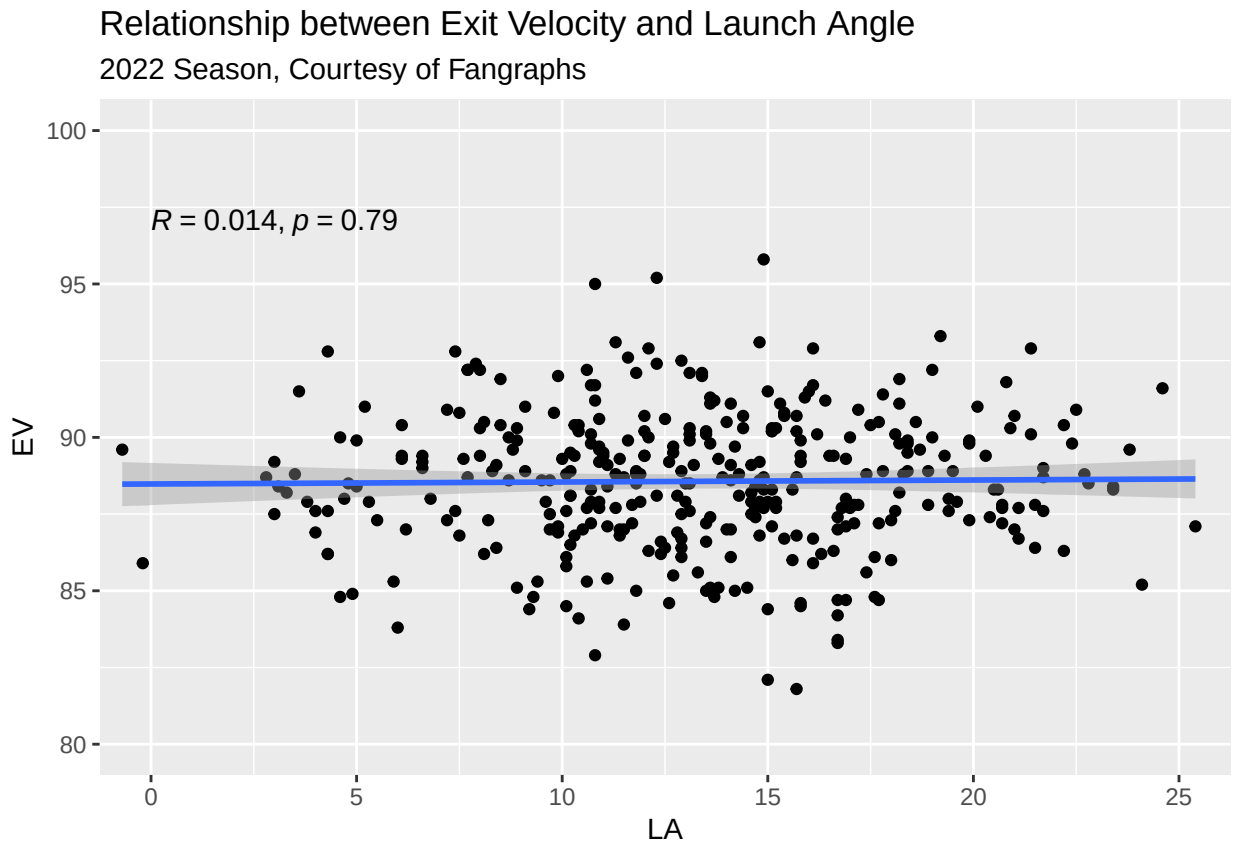
```
##
## FALSE
## 72833
```

Creating a viable velocity to predict

Looking at relationship between launch angle and exit velocity in MLB

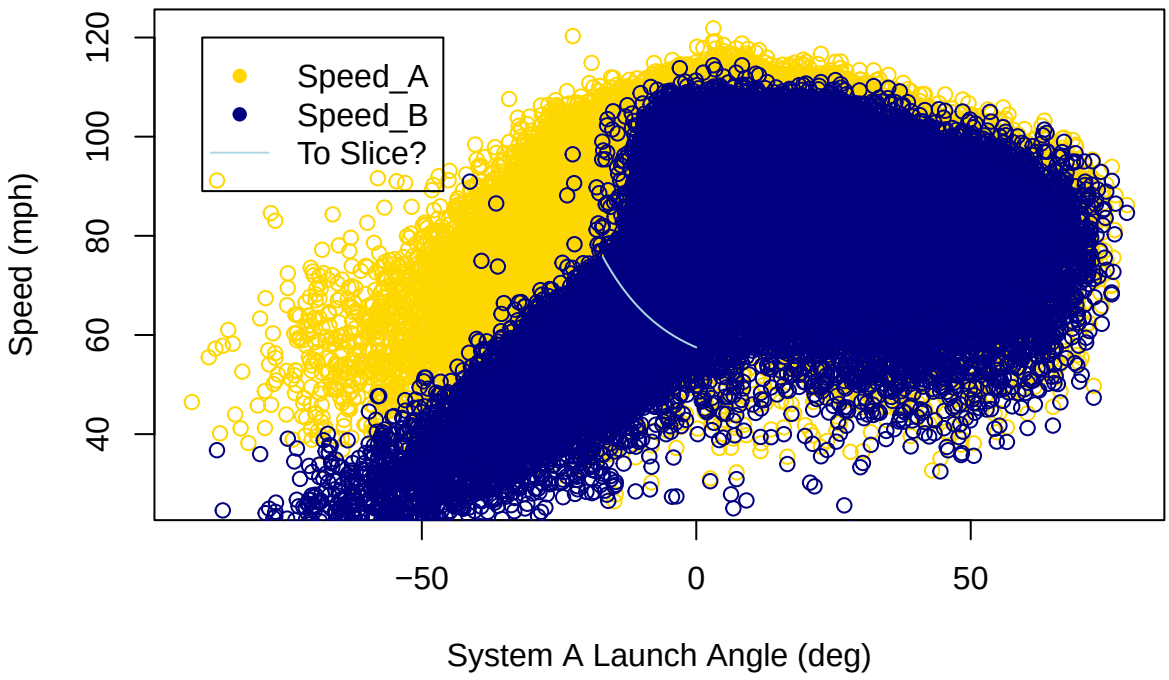
```
ggplot(data = FG_joined,
  aes(LA, EV)) +
  geom_point() +
  stat_cor(method = "pearson", label.x = 0, label.y = 97) +
  geom_smooth(method = 'lm') +
  coord_cartesian(ylim = c(80,100), xlim = c(0,25)) +
  labs(title = 'Relationship between Exit Velocity and Launch Angle', subtitle = '2022 Season, Courtesy
```

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



Investigating Differences Between Systems

```
plot(Rays$vangle_A, Rays$speed_A, col = 'Gold',
     main = 'Relationship Between Launch Angle and Speed',
     xlab = 'System A Launch Angle (deg)',
     ylab = 'Speed (mph)')
points(Rays$vangle_A, Rays$speed_B, col = 'Navy Blue')
legend(x = -90, y = 120, legend = c('Speed_A', 'Speed_B', 'To Slice?'),
       col = c('Gold', 'Navy Blue', 'Light Blue'),
       pch = c(16, 16, NA),
       lty = c(NA, NA, 1))
curve(2.17^-((x/9)-2.2) + 52, add = TRUE, from = -17, to = 0, col = 'Light Blue')
```




```
cor.test(Rays$vangle_A, Rays$speed_A)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Rays$vangle_A and Rays$speed_A
## t = 31.072, df = 65801, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1127156 0.1277758
## sample estimates:
##       cor
## 0.1202526
```

```
cor.test(Rays$vangle_B, Rays$speed_B)
```

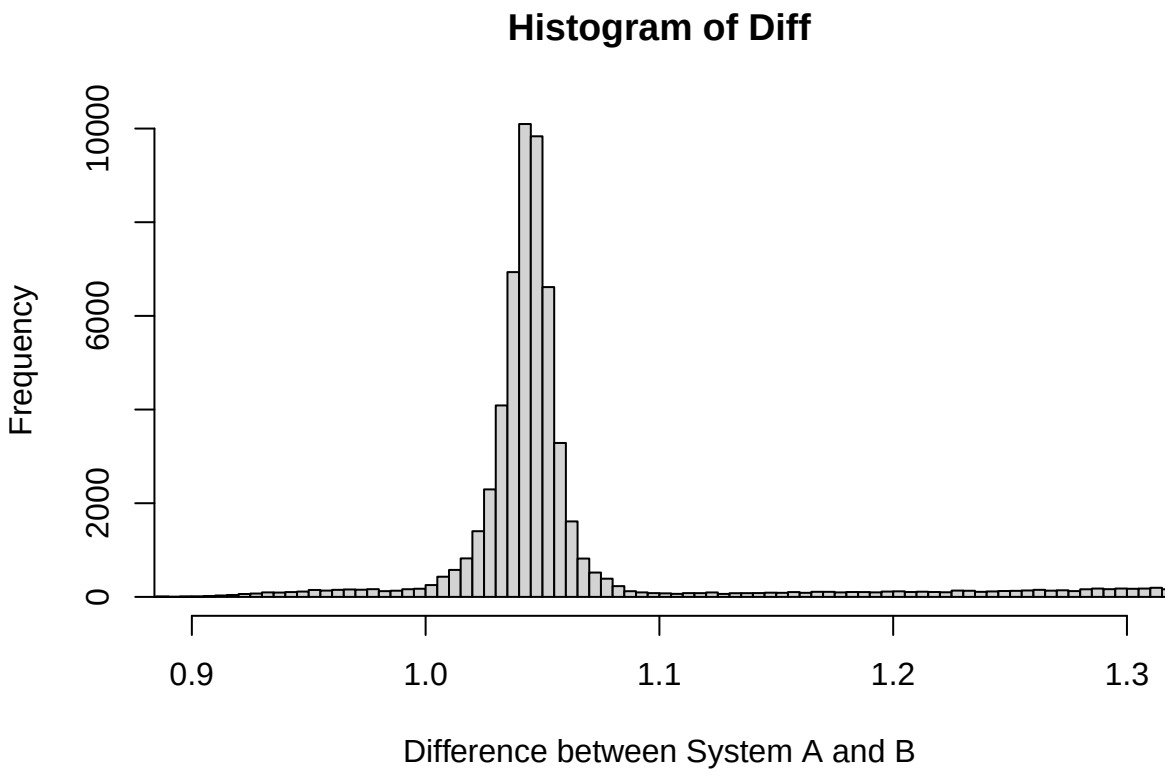
```
##
##  Pearson's product-moment correlation
##
## data:  Rays$vangle_B and Rays$speed_B
## t = 99.623, df = 71971, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3416836 0.3545244
```

```
## sample estimates:
##       cor
## 0.3481204
```

```
test <- Rays_test

test$diff <- test$speed_A/test$speed_B

summary(test$diff)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3787  1.0395  1.0486  1.1655  1.2025 10.0266
```

```
hist(test$diff, xlim = c(0.9,1.3), breaks = 1500, xlab = 'Difference between System A and B', main = 'Hi
```



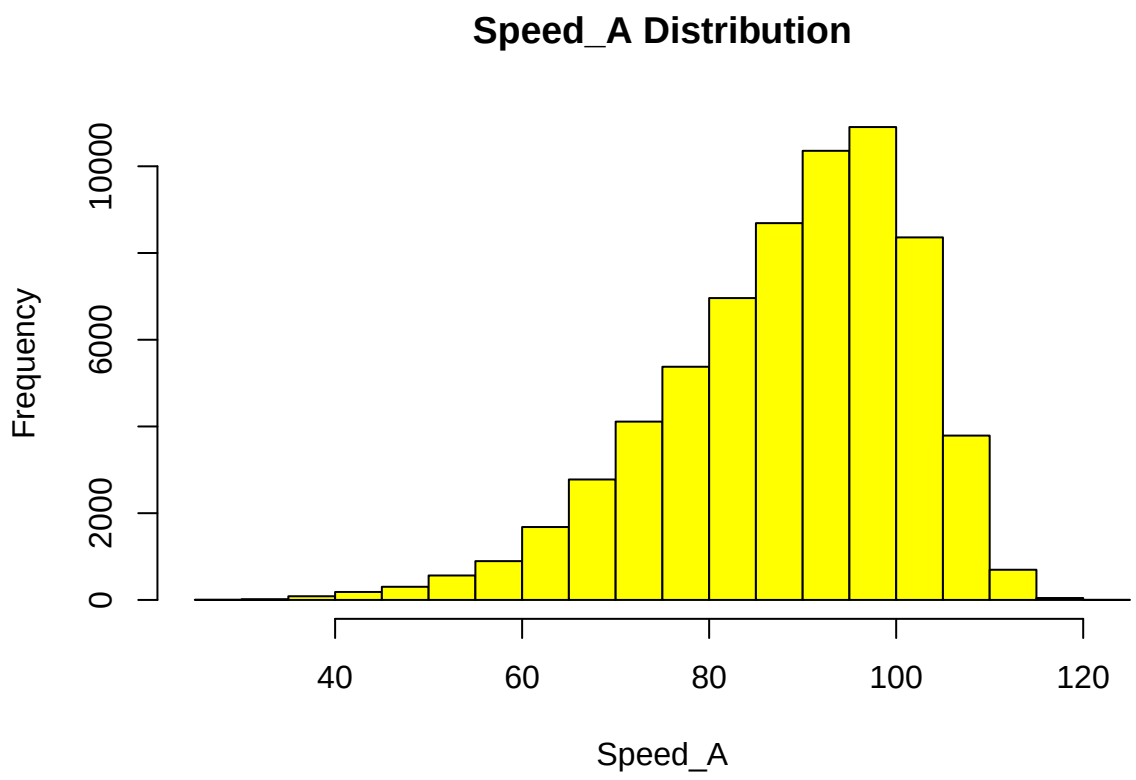
Creating New Speed

```
PitchByPitchData <- read_csv("~/Desktop/PitchByPitchData.csv")
```

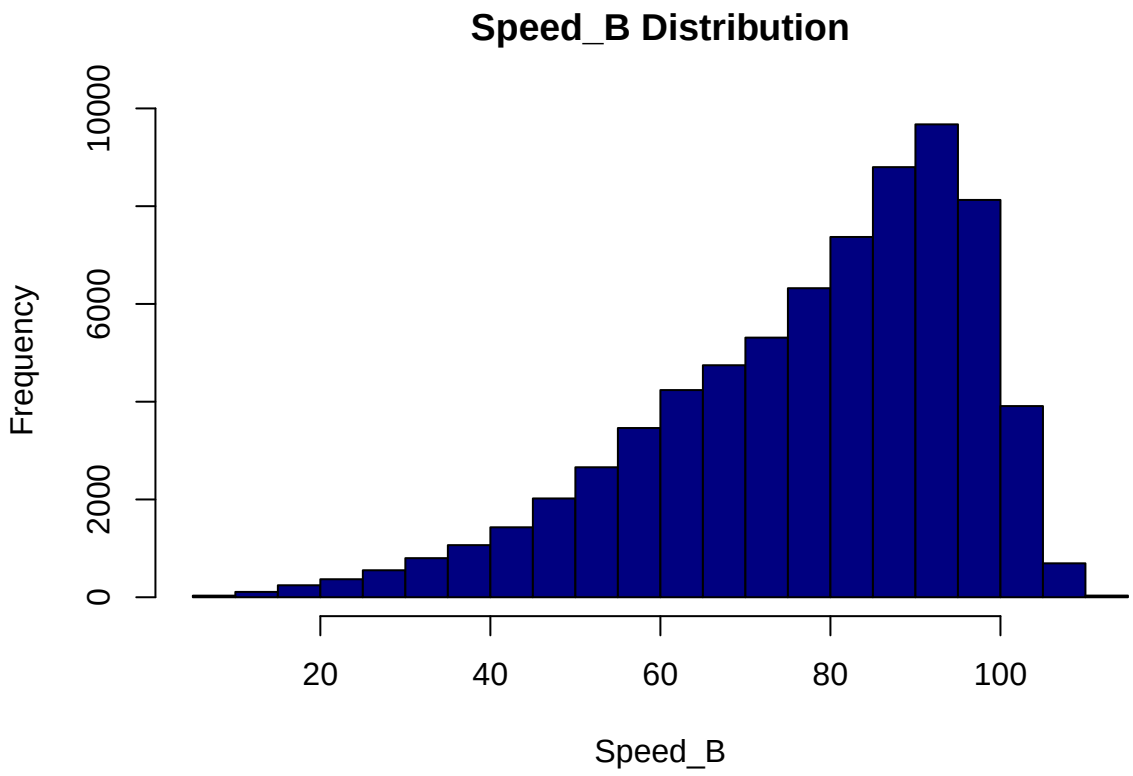
```
## Rows: 4314901 Columns: 93
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr  (17): pitch_type, player_name, events, description, des, game_type, sta...
## dbl  (68): release_speed, release_pos_x, release_pos_z, batter, pitcher, zon...
## lgl   (7): spin_dir, spin_rate_deprecated, break_angle_deprecated, break_len...
## date  (1): game_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

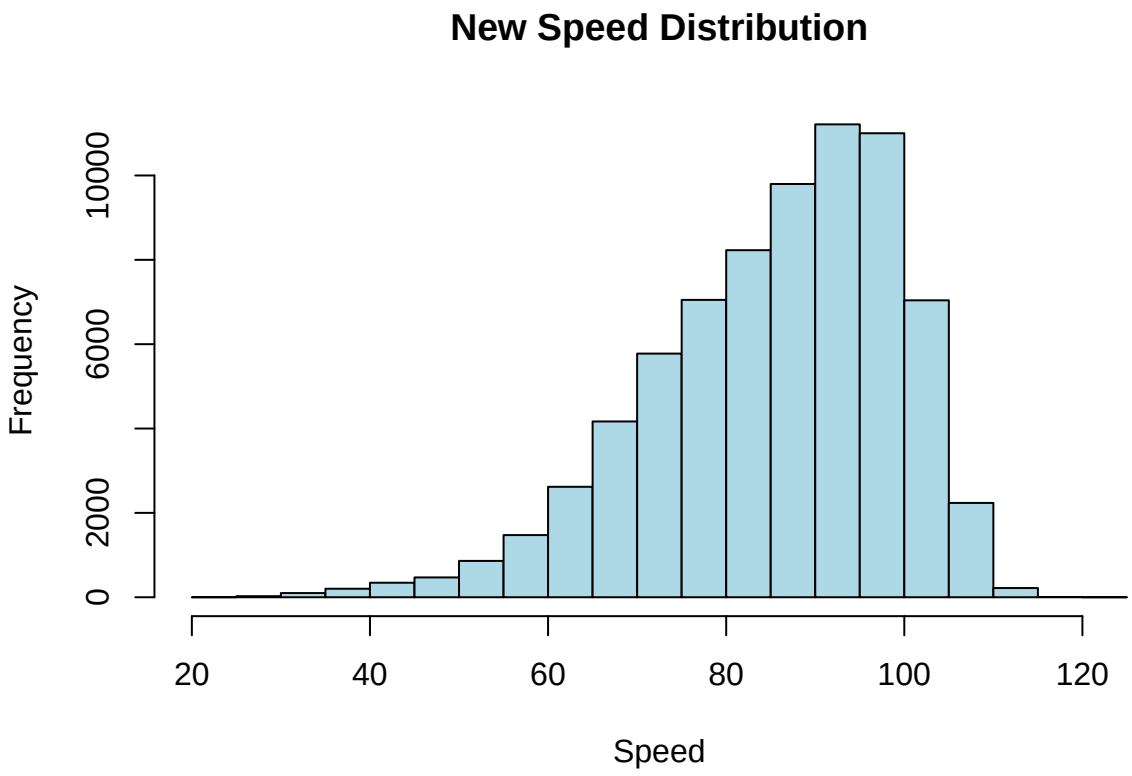
```r
test <- test %>%
  mutate(
    speed = case_when(
      diff < 1.1 & diff > 1 ~ (speed_A + speed_B)/2,
      diff > 1.1 | diff < 1 ~ speed_A,
      is.na(speed_B) ~ speed_A,
      is.na(speed_A) ~ test$speed_A[which.min(abs(vangle_A - vangle_B))],
      TRUE ~ 0,
  ))

test <- test %>%
  filter(speed != 0)

hist(Rays$speed_A, col = 'Yellow', xlab = 'Speed_A', main = 'Speed_A Distribution')
```



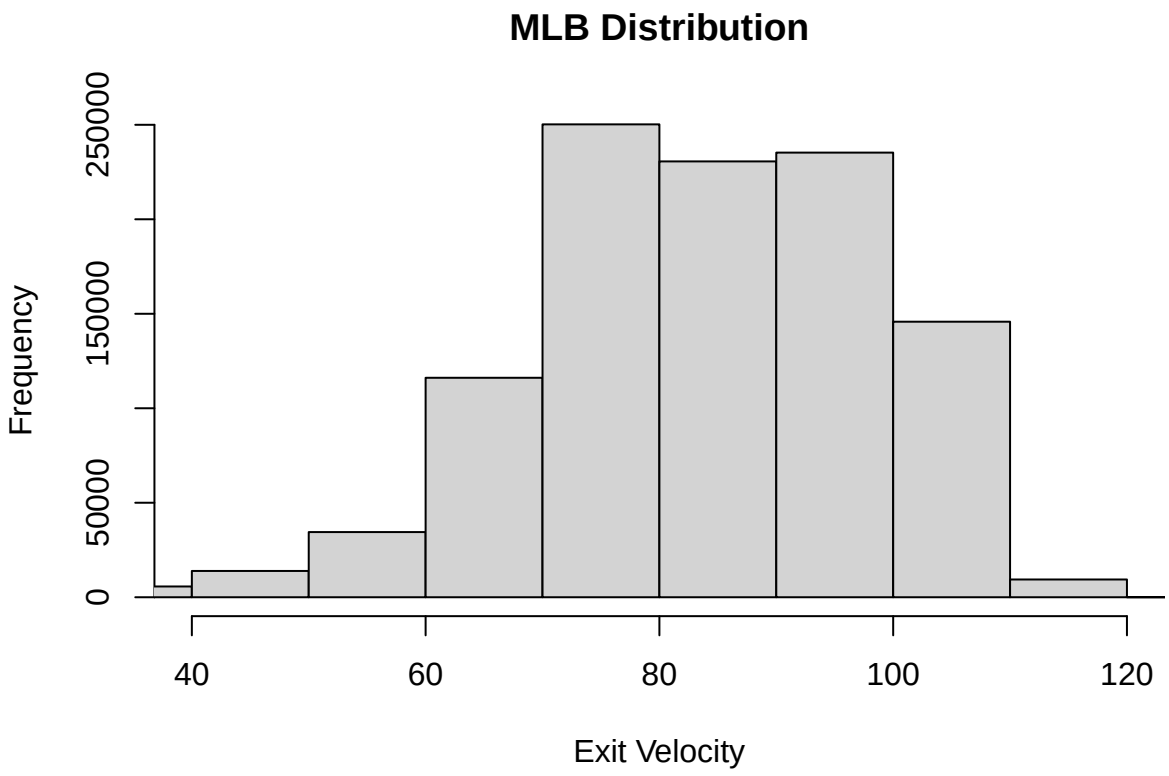

```r
hist(Rays$speed_B, col = 'Navy Blue', xlab = 'Speed_B', main = 'Speed_B Distribution')
```




```r
hist(test$speed, col = 'Light Blue', xlab = 'Speed', main = 'New Speed Distribution')
```




```
hist(PitchByPitchData$launch_speed, xlim = c(40, 120), breaks = 30, xlab = 'Exit Velocity', main = 'MLB
```



Adding New Speed to Dataset

```
Rays_final <- test %>%
  summarise(
    batter = batter,
    pitcher = pitcher,
    hittype = hittype,
    speed = speed,
    angle = vangle_A,
  )

table(is.na(Rays_final))
```

```
##
##  FALSE
## 364165
```

Create Grouped Dataset With New Speed

```
Rays_grouped_batter_final <- Rays_final %>%
  group_by(batter) %>%
  summarize(
    Avg_True_EV = round(mean(speed), 2),
    Avg_LA = round(mean(angle), 2),
```

```
    'fly_ball%' = round(100*sum(ifelse(hittype == 'fly_ball', 1, 0))/n(), 1),
    'ground_ball%' = round(100*sum(ifelse(hittype == 'ground_ball', 1, 0))/n(), 1),
    'line_drive%' = round(100*sum(ifelse(hittype == 'line_drive', 1, 0))/n(), 1),
    'pop_up%' = round(100*sum(ifelse(hittype == 'popup', 1, 0))/n(), 1),
    BBE = n()
  ) %>%
  filter(BBE >= 50) %>%
  select(-BBE)
```

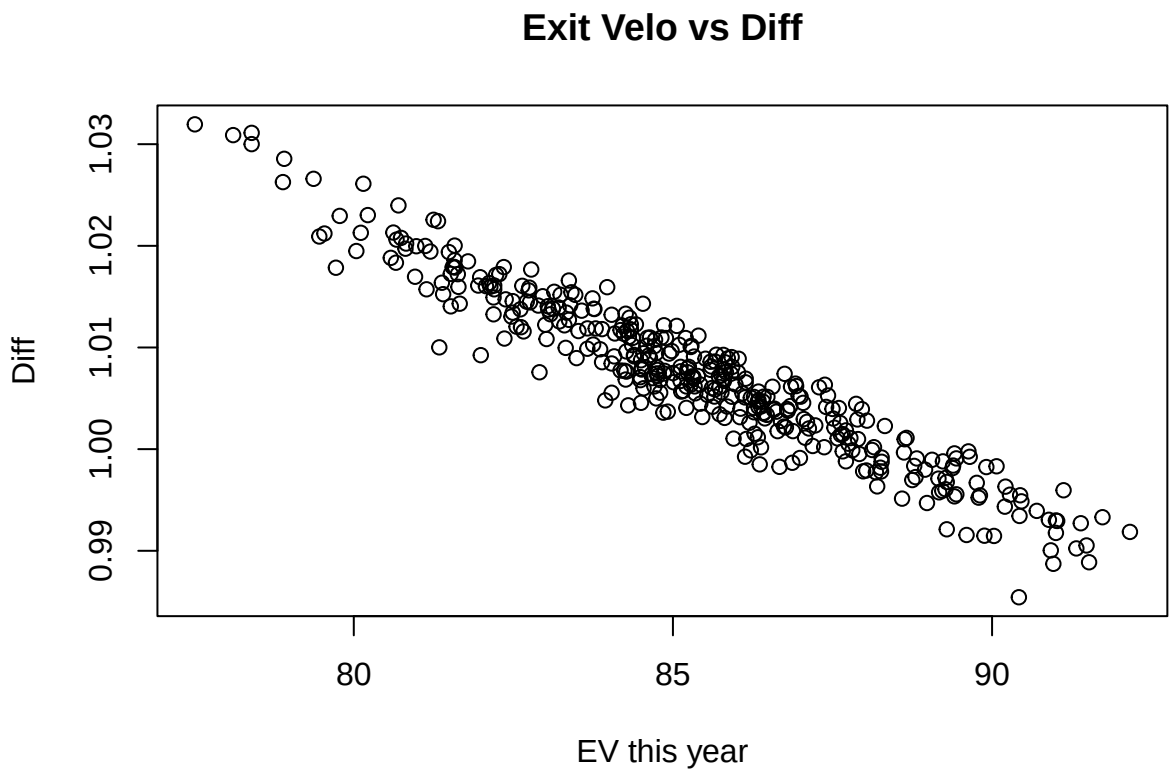
The Final Dataset

```
Rays_grouped_batter_final <- Rays_grouped_batter_final %>%
  mutate(EV_next = 20.651 + Avg_True_EV*0.770 - `pop_up%` * 0.062)

Rays_prediction <- Rays_grouped_batter_final %>%
  summarise(
    batter = batter,
    EV_next = EV_next,
  )

Rays_prediction
```

```
## # A tibble: 384 x 2
##    batter EV_next
##     <dbl>   <dbl>
##  1      1    87.4
##  2      4    87.3
##  3      5    86.2
##  4      6    83.1
##  5      7    86.2
##  6      8    88.2
##  7     10    85.2
##  8     11    87.8
##  9     14    84.6
## 10     15    88.0
## # ... with 374 more rows
```

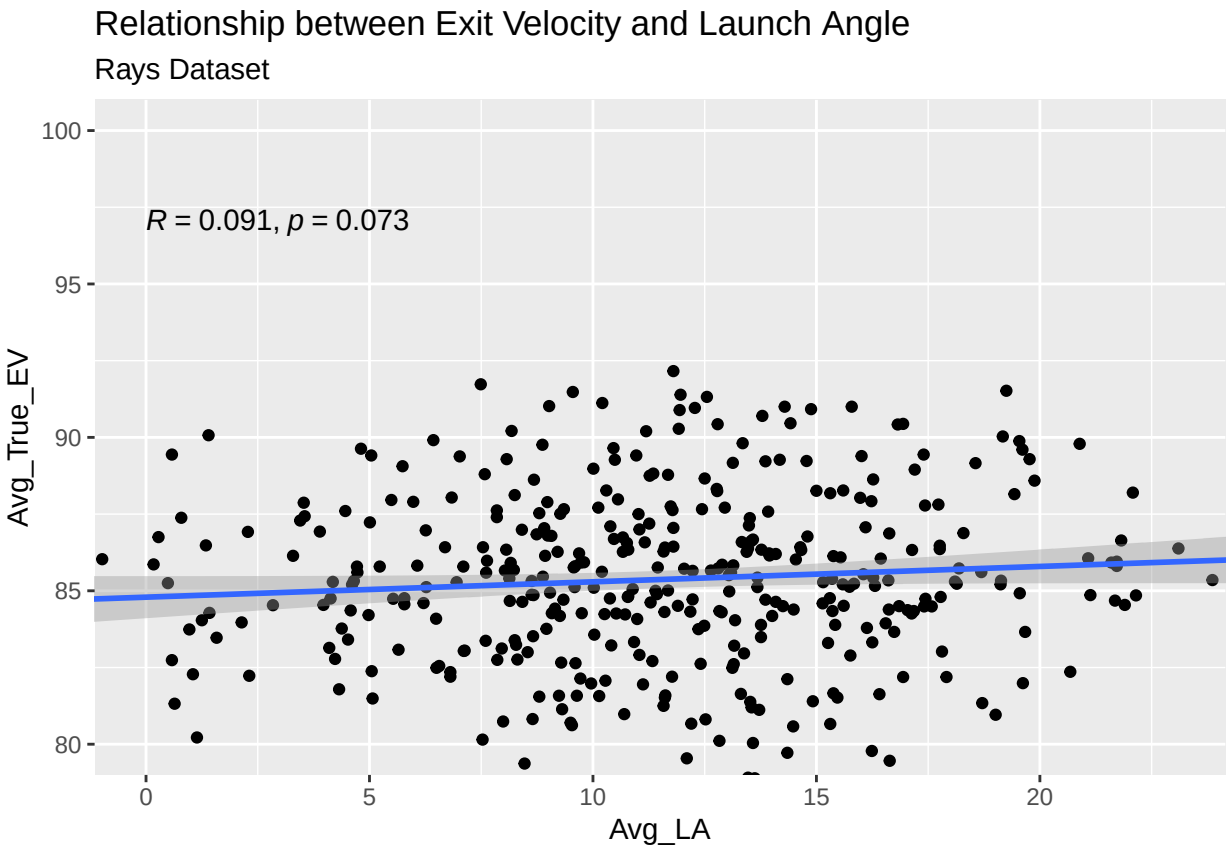
Testing Prediction

```
Rays_grouped_batter_final$diff <- Rays_grouped_batter_final$EV_next/Rays_grouped_batter_final$Avg_True_

plot(Rays_grouped_batter_final$Avg_True_EV, Rays_grouped_batter_final$diff, xlab = 'EV this year', ylab
```




```
ggplot(data = Rays_grouped_batter_final,
       aes(Avg_LA, Avg_True_EV)) +
  geom_point() +
  stat_cor(label.x = 0, label.y = 97) +
  geom_smooth(method = 'lm') +
  coord_cartesian(ylim = c(80,100), xlim = c(0,23)) +
  labs(title = 'Relationship between Exit Velocity and Launch Angle', subtitle = 'Rays Dataset')
```

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


Relationship between Exit Velocity and Launch Angle
Rays Dataset
R = 0.091, p = 0.073
Avg_True_EV
Avg_LA
100
95
90
85
80
0
5
10
15
20