Final Project 2021, Jakob Pachucinski and Alon Kremerman

```
library(tidyverse)
library(mosaic)
library(Lahman)
library(mdsr)
library(tidymodels)

#For the project, we want to explore whether certain factors such as height,
weight, and birthplace
#effect a certain stat on hitters or pitchers
#For pitchers we will explore ERA and SO
#For the hitters, we evaluated OBP, AVG, SLG, SO, HR, SB
#We hypothesized that heavier players would have a bigger slugging percentage
#While lighter players would have more stolen bases
#Also, shorter players might have a higher OBP because they have a smaller
strike zone
#While taller players may have more SOs with a bigger strike zone
#For the pitching, we suggested that taller pitchers may have more strikeouts
#Since technically they release the ball closer to the plate
help("Lahman")

## starting httpd help server ... done

#Will use this data frame for the personal stats such as height, weight, etc.
head(People)

##    playerID birthYear birthMonth birthDay birthCountry birthState
birthCity
## 1 aardsda01      1981         12       27          USA         CO
Denver
## 2 aaronha01      1934          2        5          USA         AL
Mobile
## 3 aaronto01      1939          8        5          USA         AL
Mobile
## 4  aasedo01      1954          9        8          USA         CA
Orange
## 5  abadan01      1972          8       25          USA         FL Palm
Beach
## 6  abadfe01      1985         12       17         D.R.  La Romana  La
Romana
##   deathYear deathMonth deathDay deathCountry deathState deathCity
nameFirst
## 1        NA         NA       NA         <NA>       <NA>      <NA>
David
## 2      2021          1       22          USA         GA   Atlanta
Hank
## 3      1984          8       16          USA         GA   Atlanta
```

```
Tommie
## 4        NA          NA        NA          <NA>        <NA>       <NA>
Don
## 5        NA          NA        NA          <NA>        <NA>       <NA>
Andy
## 6        NA          NA        NA          <NA>        <NA>       <NA>
Fernando
##   nameLast        nameGiven weight height bats throws      debut
finalGame
## 1  Aardsma      David Allan    215     75    R      R 2004-04-06 2015-08-
23
## 2    Aaron      Henry Louis    180     72    R      R 1954-04-13 1976-10-
03
## 3    Aaron       Tommie Lee    190     75    R      R 1962-04-10 1971-09-
26
## 4     Aase   Donald William    190     75    R      R 1977-07-26 1990-10-
03
## 5     Abad    Fausto Andres    184     73    L      L 2001-09-10 2006-04-
13
## 6     Abad Fernando Antonio    235     74    L      L 2010-07-28 2019-09-
28
##    retroID   bbrefID  deathDate  birthDate
## 1 aardd001 aardsda01       <NA> 1981-12-27
## 2 aaroh101 aaronha01 2021-01-22 1934-02-05
## 3 aarot101 aaronto01 1984-08-16 1939-08-05
## 4 aased001  aasedo01       <NA> 1954-09-08
## 5 abada001  abadan01       <NA> 1972-08-25
## 6 abadf001  abadfe01       <NA> 1985-12-17

#Selected only the columns that we "might" use and dropped all the NA values
players1 <- People%>%
  select(playerID, birthYear, birthCountry, birthState, nameGiven, weight,
height, debut, bats)%>%
  drop_na()
head(players1)

##    playerID birthYear birthCountry birthState        nameGiven weight
height
## 1 aardsda01      1981          USA         CO      David Allan    215
75
## 2 aaronha01      1934          USA         AL      Henry Louis    180
72
## 3 aaronto01      1939          USA         AL       Tommie Lee    190
75
## 4  aasedo01      1954          USA         CA   Donald William    190
75
## 5  abadan01      1972          USA         FL    Fausto Andres    184
73
## 6  abadfe01      1985         D.R.  La Romana Fernando Antonio    235
74
```

```
##        debut bats
## 1 2004-04-06    R
## 2 1954-04-13    R
## 3 1962-04-10    R
## 4 1977-07-26    R
## 5 2001-09-10    L
## 6 2010-07-28    L
```

```
#Headed the batting to see which stats we should explore
head(Batting)
```

```
##    playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB
SO
## 1 abercda01   1871     1    TRO   NA  1   4  0  0   0   0  0   0  0  0  0
0
## 2  addybo01   1871     1    RC1   NA 25 118 30 32   6   0  0  13  8  1  4
0
## 3 allisar01   1871     1    CL1   NA 29 137 28 40   4   5  0  19  3  1  2
5
## 4 allisdo01   1871     1    WS3   NA 27 133 28 44  10   2  2  27  1  1  0
2
## 5 ansonca01   1871     1    RC1   NA 25 120 29 39  11   3  0  16  6  2  2
1
## 6 armstbo01   1871     1    FW1   NA 12  49  9 11   2   1  0   5  0  1  0
1
##   IBB HBP SH SF GIDP
## 1  NA  NA NA NA    0
## 2  NA  NA NA NA    0
## 3  NA  NA NA NA    1
## 4  NA  NA NA NA    0
## 5  NA  NA NA NA    0
## 6  NA  NA NA NA    0
```

```
#Grouped by playerID so I could combine all the players years into one total
career
#Then summed all the AB, H, HR, RBI, SB, and SO
#DBL and TRP were meant to be amount of doubles and triples
#Then after filtering dropped all the NAs
batters1 <- Batting%>%
  group_by(playerID)%>%
  summarise(AB = sum(AB), H = sum(H), DBL = sum(X2B), TRP = sum(X3B), HR =
sum(HR), RBI = sum(RBI), SB = sum(SB), SO = sum(SO), BB = sum(BB))%>%
  drop_na()
head(batters1)
```

```
## # A tibble: 6 x 10
##   playerID     AB     H   DBL   TRP    HR   RBI    SB    SO    BB
##   <chr>     <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1 aardsda01     4     0     0     0     0     0     0     2     0
## 2 aaronha01 12364  3771   624    98   755  2297   240  1383  1402
## 3 aaronto01   944   216    42     6    13    94     9   145    86
```

```
## 4 aasedo01      5     0     0     0     0     0     0     3     0
## 5 abadan01     21     2     0     0     0     0     0     5     4
## 6 abadfe01      9     1     0     0     0     0     0     5     0

#Explored the pitching metrics that the Lahman package gave us
head(Pitching)

##    playerID yearID stint teamID lgID  W  L  G GS CG SHO SV IPouts   H  ER
HR BB
## 1 bechtge01   1871     1    PH1   NA  1  2  3  3  2   0  0     78  43  23
0 11
## 2 brainas01   1871     1    WS3   NA 12 15 30 30 30   0  0    792 361 132
4 37
## 3 fergubo01   1871     1    NY2   NA  0  0  1  0  0   0  0      3   8   3
0  0
## 4 fishech01   1871     1    RC1   NA  4 16 24 24 22   1  0    639 295 103
3 31
## 5 fleetfr01   1871     1    NY2   NA  0  1  1  1  1   0  0     27  20  10
0  3
## 6 flowedi01   1871     1    TRO   NA  0  0  1  0  0   0  0      3   1   0
0  0
##   SO BAOpp   ERA IBB WP HBP BK  BFP GF   R SH SF GIDP
## 1  1    NA  7.96  NA  7  NA  0  146  0  42 NA NA   NA
## 2 13    NA  4.50  NA  7  NA  0 1291  0 292 NA NA   NA
## 3  0    NA 27.00  NA  2  NA  0   14  0   9 NA NA   NA
## 4 15    NA  4.35  NA 20  NA  0 1080  1 257 NA NA   NA
## 5  0    NA 10.00  NA  0  NA  0   57  0  21 NA NA   NA
## 6  0    NA  0.00  NA  0  NA  0    3  1   0 NA NA   NA

#Grouped by playerID like we did for the batters to get the career stats
#Summed up the total stats that we wanted
#Dropped the NA values
pitchers1 <- Pitching%>%
  group_by(playerID)%>%
  summarise(seasons = n(), ER = sum(ER), SO = sum(SO), OUTS = sum(IPouts))%>%
  drop_na()
head(pitchers1)

## # A tibble: 6 x 5
##   playerID  seasons    ER    SO  OUTS
##   <chr>       <int> <int> <int> <int>
## 1 aardsda01       9   160   340  1011
## 2 aasedo01       13   468   641  3328
## 3 abadfe01       10   135   280   992
## 4 abbeybe01       6   285   161  1704
## 5 abbeych01       1     1     0     6
## 6 abbotda01       1     9     1    39

#Joined the height/weight data with the batting statistics
batters2 <- batters1%>%
```

```
  inner_join(players1, by = "playerID")
head(batters2)

## # A tibble: 6 x 18
##   playerID     AB     H   DBL   TRP    HR   RBI    SB    SO    BB
birthYear
##   <chr>     <int> <int> <int> <int> <int> <int> <int> <int> <int>
<int>
## 1 aardsda01     4     0     0     0     0     0     0     2     0
1981
## 2 aaronha01 12364  3771   624    98   755  2297   240  1383  1402
1934
## 3 aaronto01   944   216    42     6    13    94     9   145    86
1939
## 4 aasedo01      5     0     0     0     0     0     0     3     0
1954
## 5 abadan01     21     2     0     0     0     0     0     5     4
1972
## 6 abadfe01      9     1     0     0     0     0     0     5     0
1985
## # ... with 7 more variables: birthCountry <chr>, birthState <chr>,
## #   nameGiven <chr>, weight <int>, height <int>, debut <chr>, bats <fct>

#Did the same with the pitching statistics
pitchers2 <- pitchers1%>%
  inner_join(players1, by = "playerID")
head(pitchers2)

## # A tibble: 6 x 13
##   playerID seasons    ER    SO  OUTS birthYear birthCountry birthState
nameGiven
##   <chr>      <int> <int> <int> <int>     <int> <chr>        <chr>
<chr>
## 1 aardsda~       9   160   340  1011      1981 USA          CO
David Al~
## 2 aasedo01      13   468   641  3328      1954 USA          CA
Donald W~
## 3 abadfe01      10   135   280   992      1985 D.R.         La Romana
Fernando~
## 4 abbeybe~       6   285   161  1704      1869 USA          VT
Bert Wood
## 5 abbeych~       1     1     0     6      1866 USA          NE
Charles ~
## 6 abbotda~       1     9     1    39      1862 USA          OH
Leander ~
## # ... with 4 more variables: weight <int>, height <int>, debut <chr>,
## #   bats <fct>

#Filter out to get the players with a certain amount of at-bats
#Decided to use 2000 at-bats as the cutoff
#That way there wouldn't be players with small sample sizes
```

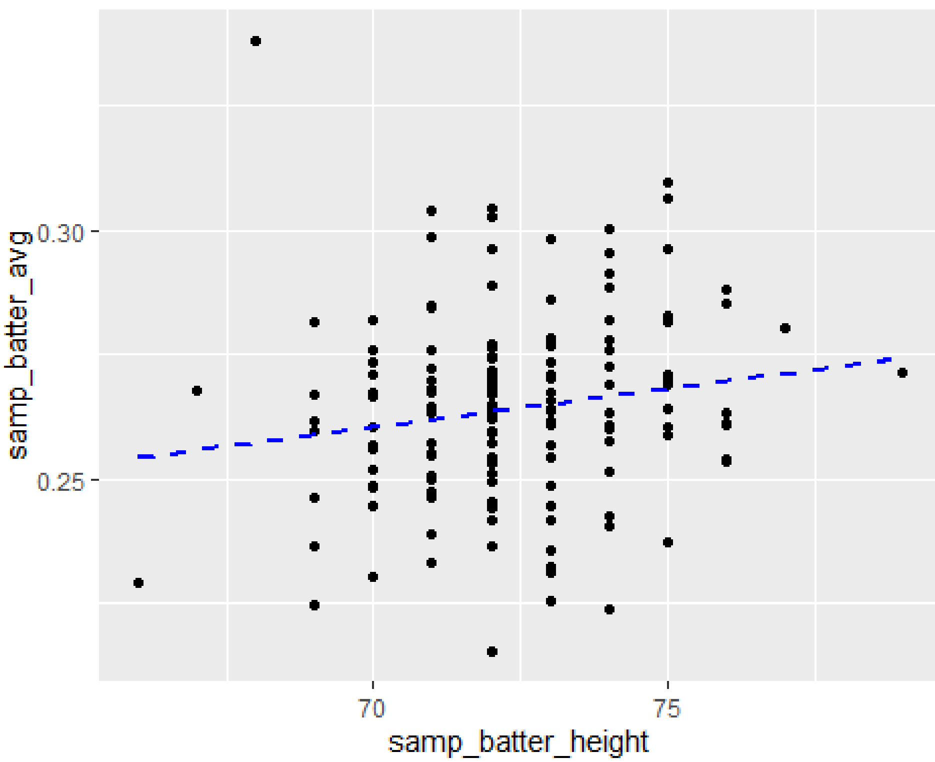
```
#Also filtered out players that were born before 1940
#Then, calculated XBH in order to get slugging percentage
#Also calculated average and on base percentage
#Did not include RBI because we did not think it would be affected by
anything
batters3 <- batters2%>%
  filter(AB > 2000, birthYear > 1940)%>%
  mutate(XBH = DBL + TRP + HR, AVG = H/AB, OBP = (H+BB)/AB)%>%
  mutate(SLG = ((H-XBH)+(DBL*2)+(TRP*3)+(HR*4))/AB, SOpAB=SO/AB, HRpH = HR/H,
SBpH = SB/H)%>%
  select(AB, AVG, OBP, SLG, HRpH, SBpH, SOpAB, birthYear, birthCountry,
birthState, weight, height, bats)
head(batters3)
```

```
## # A tibble: 6 x 13
##      AB   AVG   OBP   SLG   HRpH    SBpH SOpAB birthYear birthCountry
birthState
##   <int> <dbl> <dbl> <dbl>  <dbl>   <dbl> <dbl>     <int> <chr>
<chr>
## 1  2044 0.256 0.321 0.423 0.119  0.0421  0.279      1969 USA            OH
## 2  8480 0.291 0.465 0.475 0.117  0.162   0.217      1974 Venezuela
Aragua
## 3  3787 0.294 0.364 0.520 0.178  0.00898 0.220      1987 Cuba
Cienfuegos
## 4  2125 0.241 0.332 0.367 0.0898 0.0605  0.197      1988 USA            NC
## 5  2385 0.259 0.327 0.467 0.191  0.00647 0.266      1988 USA            PA
## 6  3912 0.255 0.343 0.412 0.130  0.167   0.235      1942 USA            AL
## # ... with 3 more variables: weight <int>, height <int>, bats <fct>
```

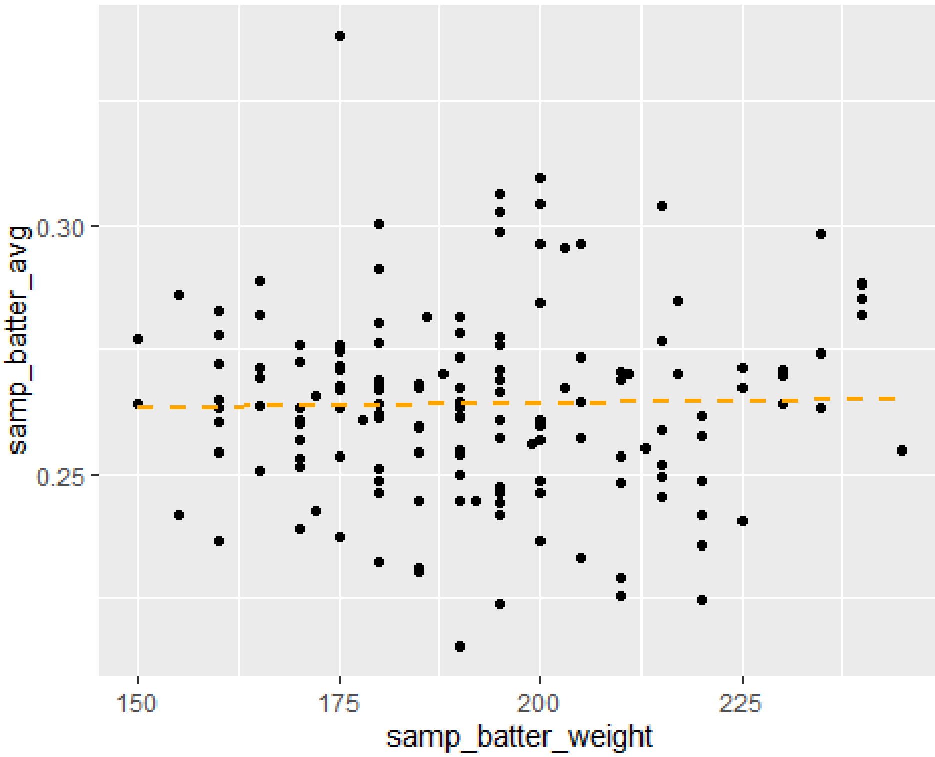
```
#Gave the pitchers a lower threshold for OUTS because they don't play the
whole game
#While most of the time, starting fielders do
#Calculated ERA by multiplying the number of outs in a game by the total
earned runs and dividing by total outs
#Used the same method to calculate a pitchers strikeouts per game
#Then selected the columns we wanted to use
pitchers3 <- pitchers2%>%
  filter(OUTS > 1800, birthYear > 1940)%>%
  mutate(ERA = (27*ER)/OUTS, SOpG = (27*SO)/OUTS)%>%
  select(OUTS, ERA, SOpG, birthYear, birthCountry, birthState, weight,
height)
head(pitchers3)
```

```
## # A tibble: 6 x 8
##    OUTS   ERA  SOpG birthYear birthCountry birthState weight height
##   <int> <dbl> <dbl>     <int> <chr>        <chr>       <int>  <int>
## 1  3328  3.80  5.20      1954 USA          CA            190     75
## 2  3858  4.39  3.39      1951 USA          AR            200     78
## 3  5022  4.25  4.77      1967 USA          MI            200     75
## 4  2162  4.92  6.19      1967 USA          CA            185     75
```

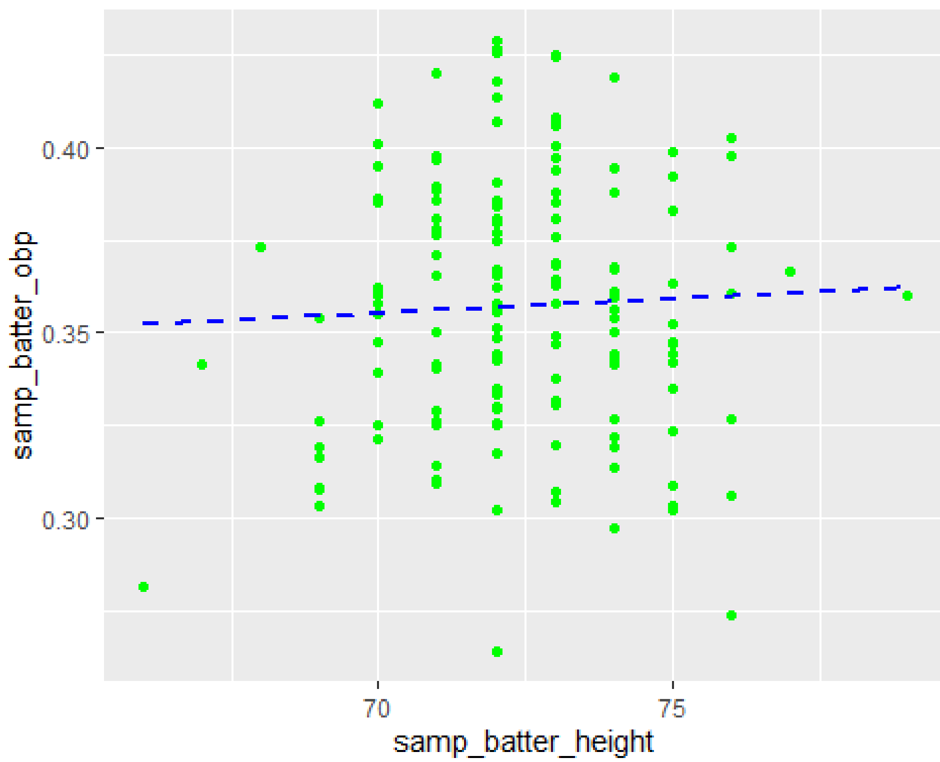
```
## 5  2713  3.97  4.80      1958 USA          TX            210     74
## 6  2608  4.17  7.15      1973 USA          AL            180     75

#Set the seed for the sample and got samples of all the batters height/weight
and hitting stats
#Then went through and compared to see if there was any good fitting line in
the linear regression models
#Color coordinated by making weight with the orange dashed line and height
with the blue
#Also color coordinated the hitting stats
set.seed(51321)
n <- 150
samp_batter_height <- sample(batters3$height,n)
samp_batter_weight <- sample(batters3$weight,n)

samp_batter_avg <- sample(batters3$AVG,n)
samp_batter_obp <- sample(batters3$OBP,n)
samp_batter_slg <- sample(batters3$SLG,n)
samp_batter_hr <- sample(batters3$HRpH,n)
samp_batter_sb <- sample(batters3$SBpH,n)
samp_batter_so <- sample(batters3$SOpAB,n)

fav_stats(batters3$AVG)

##        min        Q1    median        Q3       max      mean          sd
n
##  0.1941676 0.2516878 0.2643683 0.2772197 0.3381783 0.2648844 0.01906285
1248
##  missing
##        0

avg_height <- lm(samp_batter_avg ~ samp_batter_height)
msummary(avg_height)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.1519203  0.0556311   2.731  0.00708 **
## samp_batter_height 0.0015481  0.0007684   2.015  0.04573 *
##
## Residual standard error: 0.01903 on 148 degrees of freedom
## Multiple R-squared:  0.0267, Adjusted R-squared:  0.02012
## F-statistic:  4.06 on 1 and 148 DF,  p-value: 0.04573

gf_point(samp_batter_avg ~ samp_batter_height)%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```



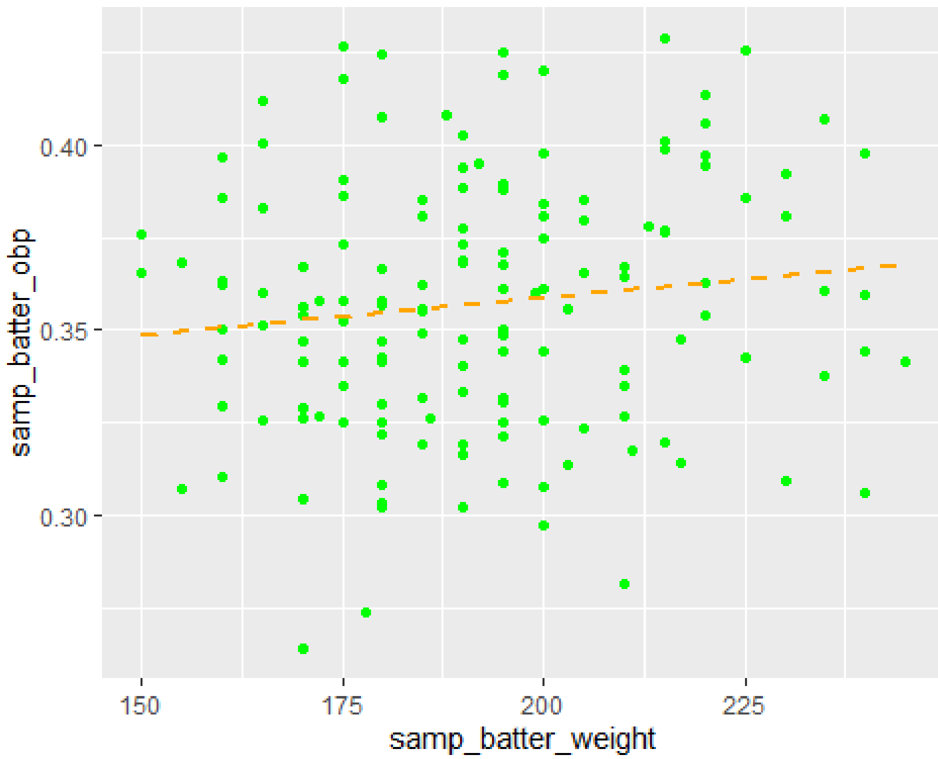

```
avg_weight <- lm(samp_batter_avg ~ samp_batter_weight)
msummary(avg_weight)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.606e-01  1.423e-02  18.310   <2e-16 ***
## samp_batter_weight 1.755e-05  7.354e-05   0.239    0.812
##
## Residual standard error: 0.01928 on 148 degrees of freedom
## Multiple R-squared:  0.0003849,  Adjusted R-squared:  -0.006369
## F-statistic: 0.05698 on 1 and 148 DF,  p-value: 0.8117

gf_point(samp_batter_avg ~ samp_batter_weight)%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```



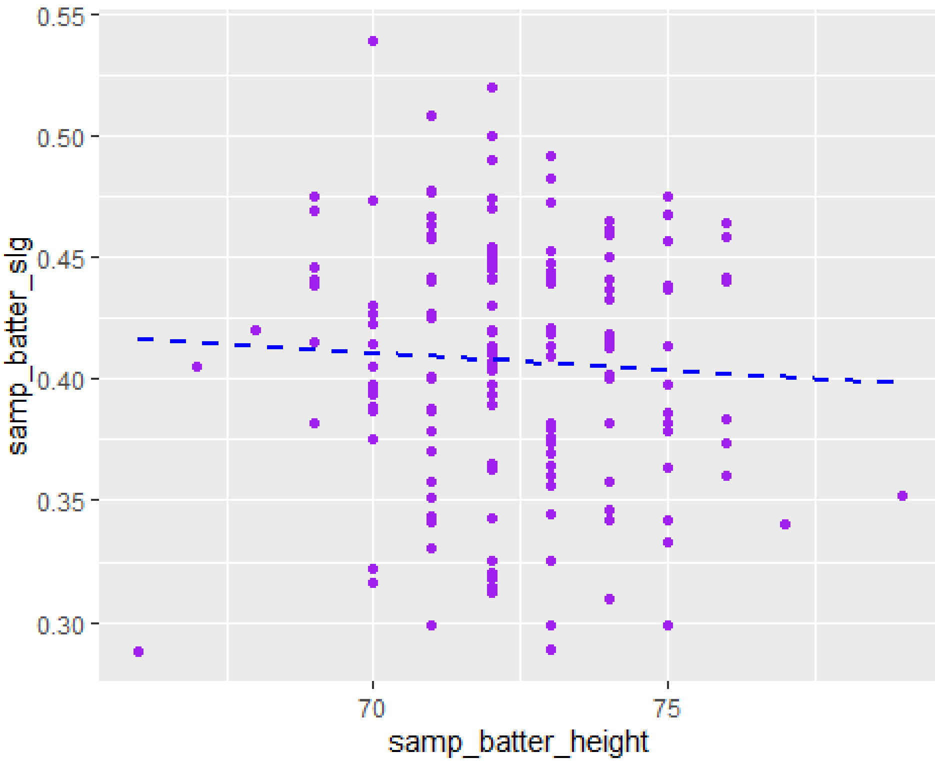

```
obp_height <- lm(samp_batter_obp ~ samp_batter_height)
msummary(obp_height)

##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.302008   0.100898   2.993  0.00324 **
## samp_batter_height 0.000761   0.001394   0.546  0.58585
##
## Residual standard error: 0.03451 on 148 degrees of freedom
## Multiple R-squared:  0.002011,   Adjusted R-squared:  -0.004732
## F-statistic: 0.2982 on 1 and 148 DF,  p-value: 0.5858

gf_point(samp_batter_obp ~ samp_batter_height, color = "green")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```



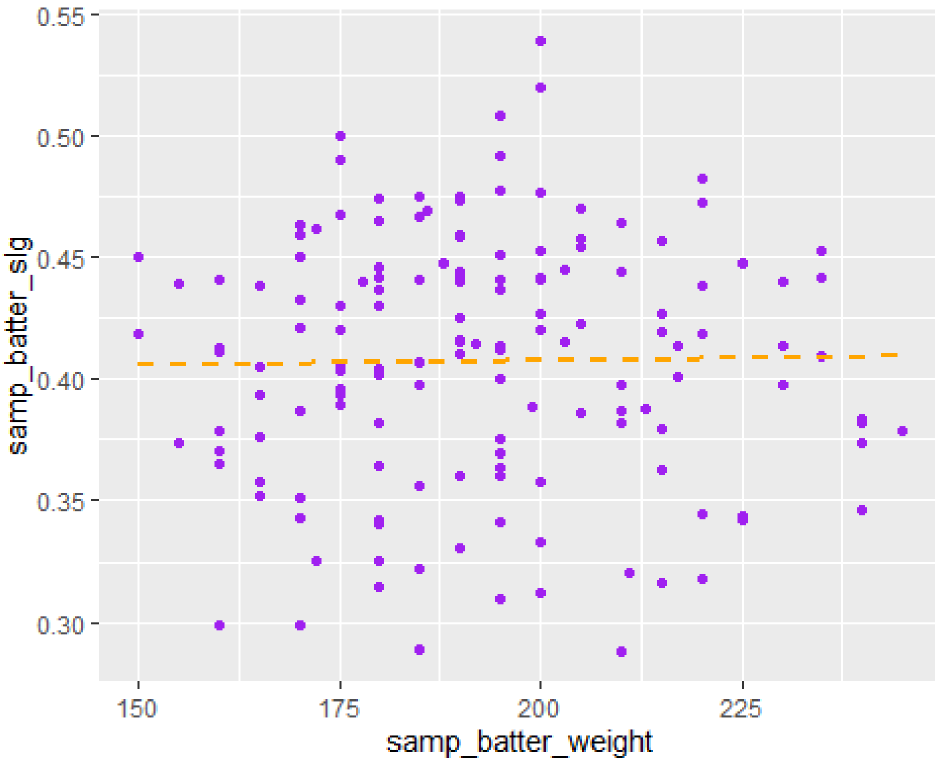

```
obp_weight <- lm(samp_batter_obp ~ samp_batter_weight)
msummary(obp_weight)

##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.3186746  0.0252970  12.597   <2e-16 ***
## samp_batter_weight 0.0001997  0.0001307   1.528    0.129
##
## Residual standard error: 0.03427 on 148 degrees of freedom
## Multiple R-squared:  0.01552,    Adjusted R-squared:  0.008872
## F-statistic: 2.334 on 1 and 148 DF,  p-value: 0.1287

gf_point(samp_batter_obp ~ samp_batter_weight, color = "green")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```



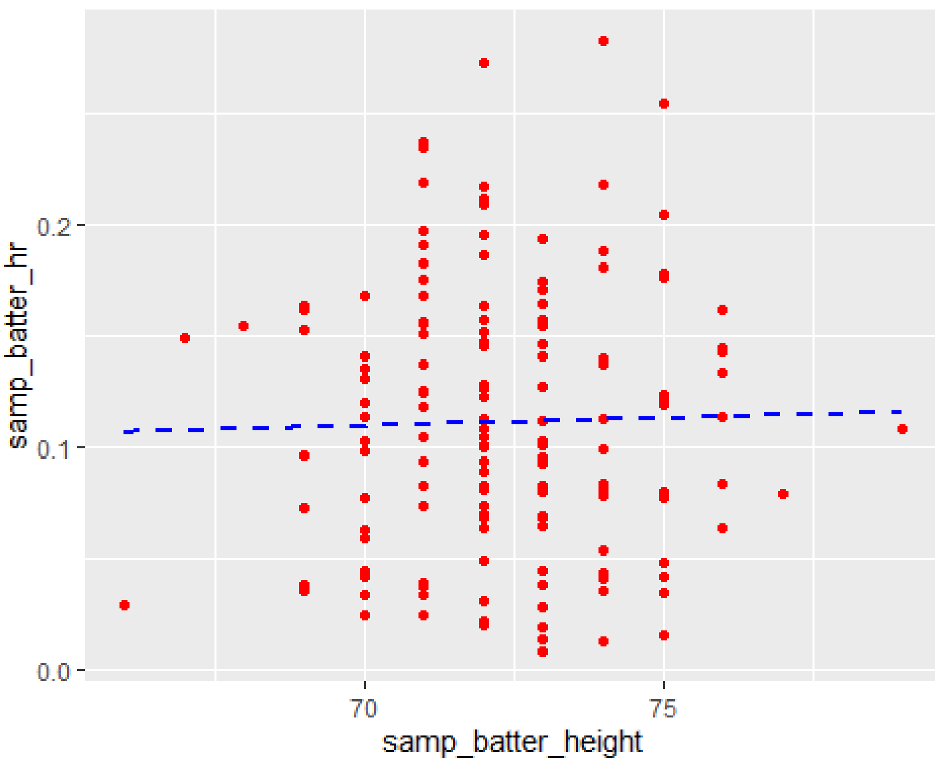

```
slg_height <- lm(samp_batter_slg ~ samp_batter_height)
msummary(slg_height)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.508403   0.152629   3.331  0.00109 **
## samp_batter_height -0.001398   0.002108  -0.663  0.50821
##
## Residual standard error: 0.0522 on 148 degrees of freedom
## Multiple R-squared:  0.002963,   Adjusted R-squared:  -0.003773
## F-statistic: 0.4399 on 1 and 148 DF,  p-value: 0.5082

gf_point(samp_batter_slg ~ samp_batter_height, color = "purple")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```



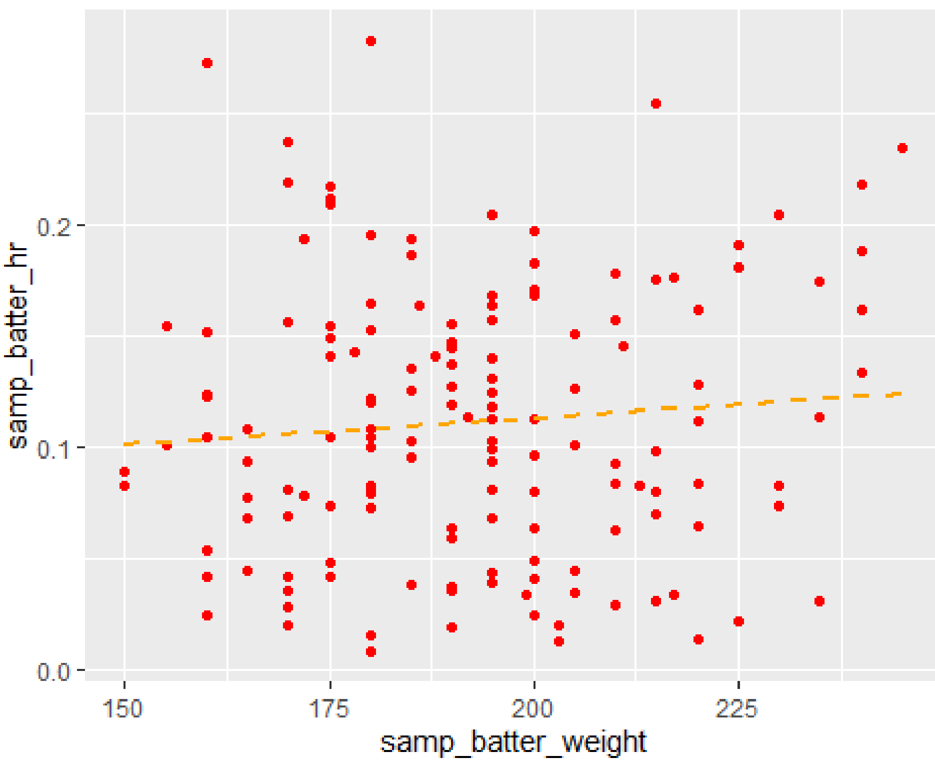

```
slg_weight <- lm(samp_batter_slg ~ samp_batter_weight)
msummary(slg_weight)

##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          4.004e-01  3.858e-02  10.379   <2e-16 ***
## samp_batter_weight 3.517e-05  1.994e-04   0.176     0.86    
## 
## Residual standard error: 0.05227 on 148 degrees of freedom
## Multiple R-squared:  0.0002102,  Adjusted R-squared:  -0.006545 
## F-statistic: 0.03112 on 1 and 148 DF,  p-value: 0.8602

gf_point(samp_batter_slg ~ samp_batter_weight, color = "purple")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```



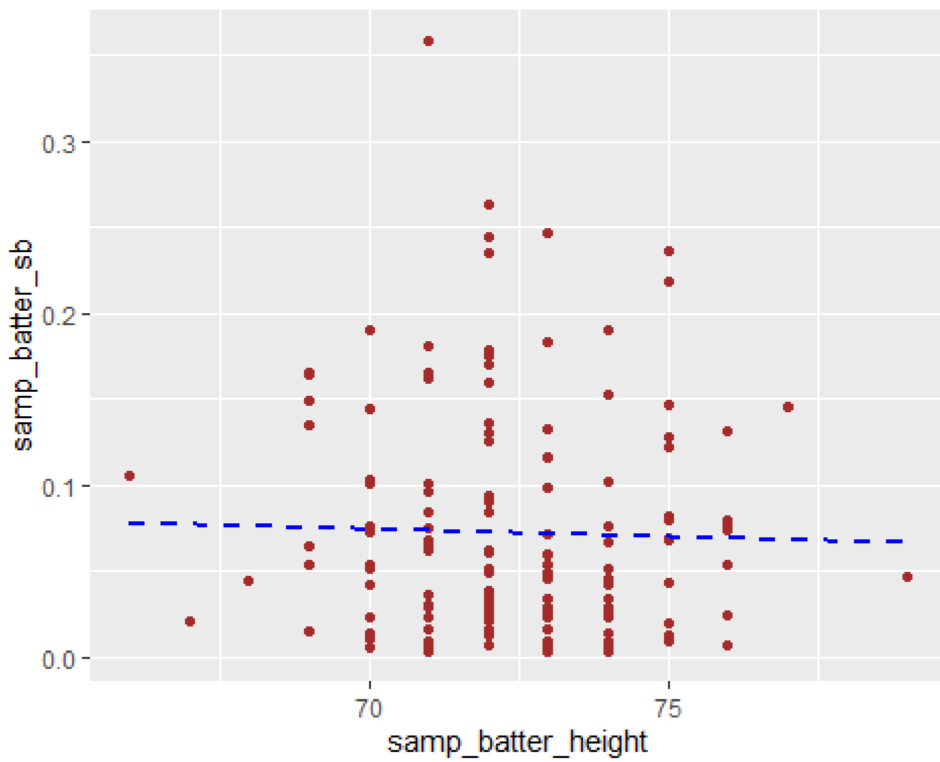

```
hr_height <- lm(samp_batter_hr ~ samp_batter_height)
msummary(hr_height)

##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0607984  0.1793294   0.339    0.735
## samp_batter_height 0.0006967  0.0024769   0.281    0.779
##
## Residual standard error: 0.06133 on 148 degrees of freedom
## Multiple R-squared:  0.0005344,  Adjusted R-squared:  -0.006219
## F-statistic: 0.07913 on 1 and 148 DF,  p-value: 0.7789

gf_point(samp_batter_hr ~ samp_batter_height, color = "red")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```




```
hr_weight <- lm(samp_batter_hr ~ samp_batter_weight)
msummary(hr_height)

##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.0607984  0.1793294   0.339    0.735
## samp_batter_height 0.0006967  0.0024769   0.281    0.779
##
## Residual standard error: 0.06133 on 148 degrees of freedom
## Multiple R-squared:  0.0005344,  Adjusted R-squared:  -0.006219
## F-statistic: 0.07913 on 1 and 148 DF,  p-value: 0.7789

gf_point(samp_batter_hr ~ samp_batter_weight, color = "red")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```

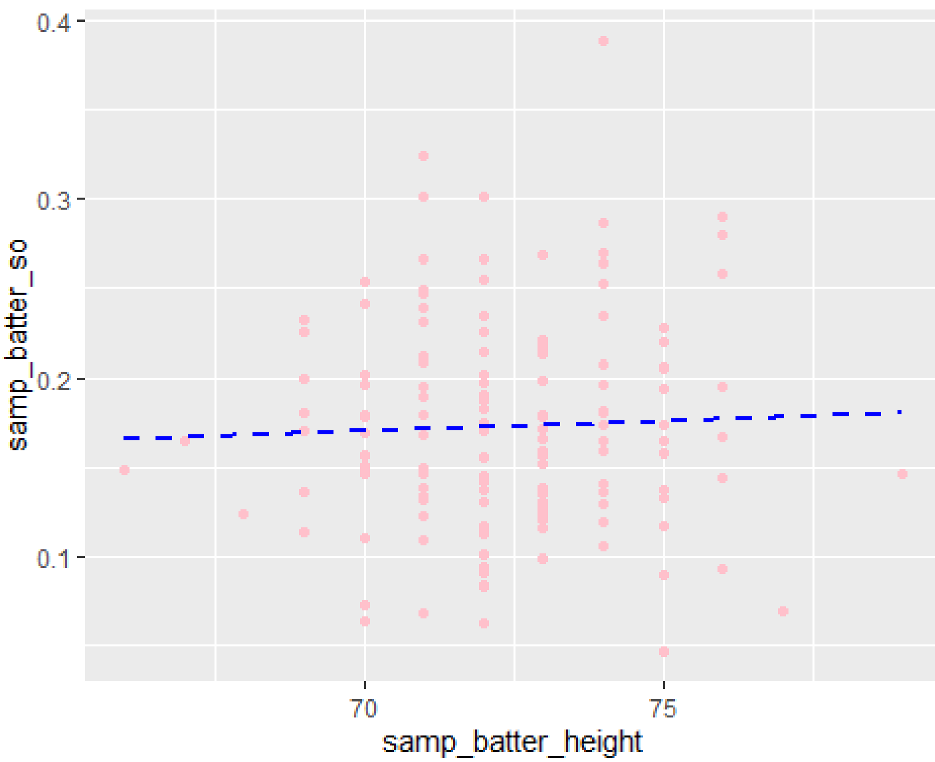



```
sb_height <- lm(samp_batter_sb ~ samp_batter_height)
msummary(sb_height)

##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.1371409  0.1937783   0.708    0.480
## samp_batter_height -0.0009011  0.0026764  -0.337    0.737
##
## Residual standard error: 0.06627 on 148 degrees of freedom
## Multiple R-squared:  0.0007653,  Adjusted R-squared:  -0.005986
## F-statistic: 0.1134 on 1 and 148 DF,  p-value: 0.7368

gf_point(samp_batter_sb ~ samp_batter_height, color = "brown")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```

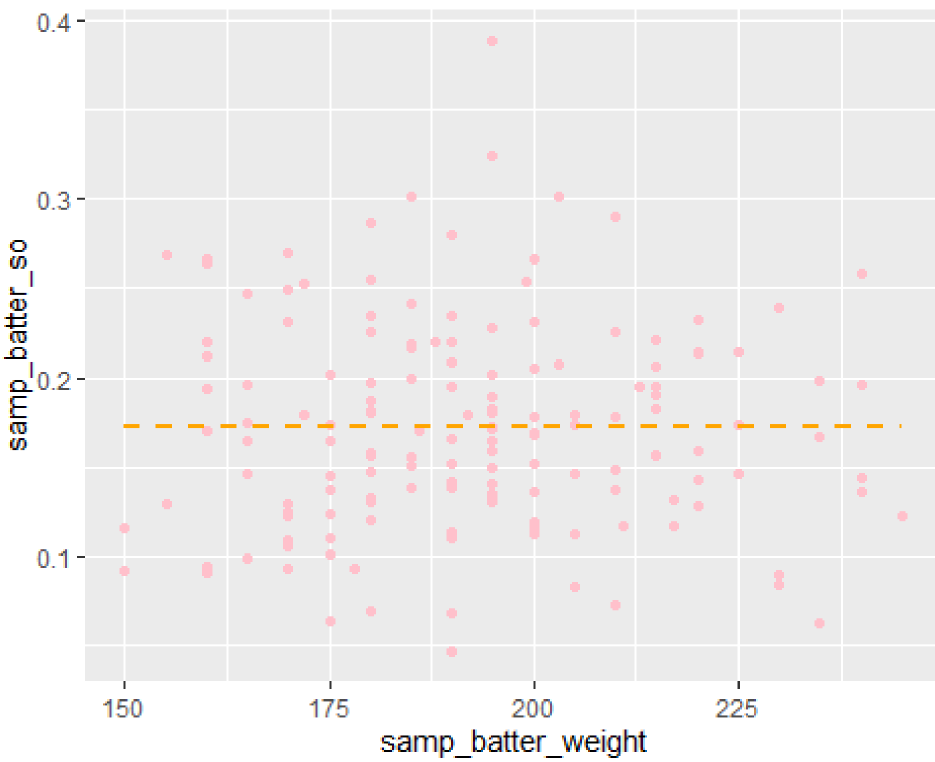



```
sb_weight <- lm(samp_batter_sb ~ samp_batter_weight)
msummary(sb_weight)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.0408981  0.0488677   0.837    0.404
## samp_batter_weight 0.0001613  0.0002525   0.639    0.524
##
## Residual standard error: 0.0662 on 148 degrees of freedom
## Multiple R-squared:  0.00275,    Adjusted R-squared:  -0.003988
## F-statistic: 0.4081 on 1 and 148 DF,  p-value: 0.5239

gf_point(samp_batter_sb ~ samp_batter_weight, color = "brown")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```

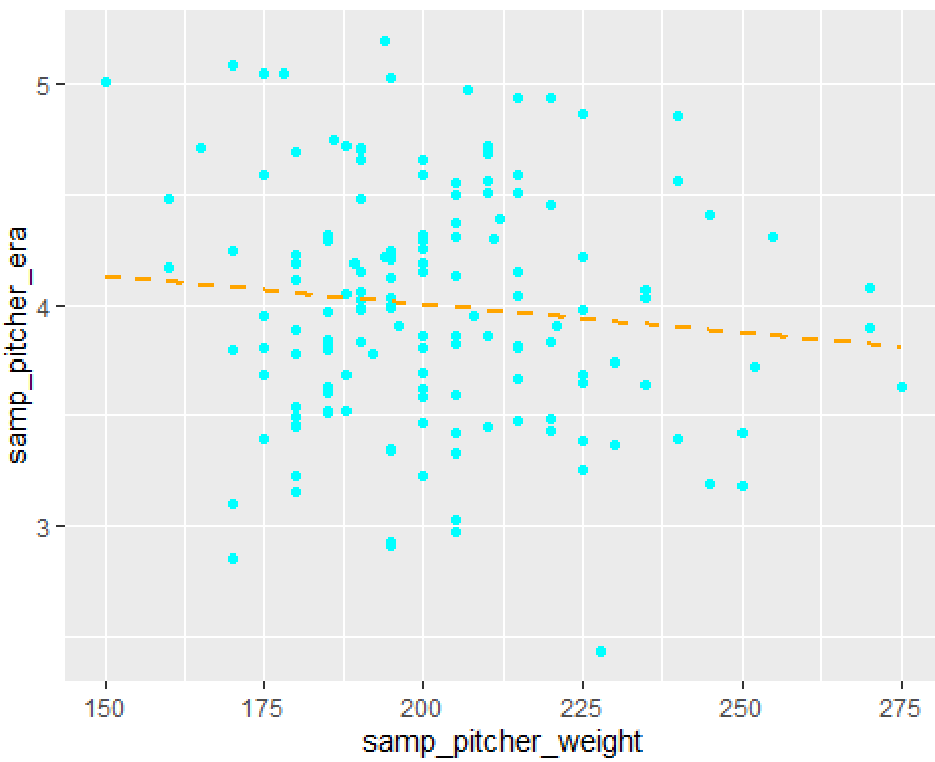

```
soH_height <- lm(samp_batter_so ~ samp_batter_height)
msummary(soH_height)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.091069   0.173172   0.526    0.600
## samp_batter_height 0.001123   0.002392   0.470    0.639
##
## Residual standard error: 0.05922 on 148 degrees of freedom
## Multiple R-squared:  0.001488,   Adjusted R-squared:  -0.005259
## F-statistic: 0.2206 on 1 and 148 DF,  p-value: 0.6393

gf_point(samp_batter_so ~ samp_batter_height, color = "pink")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```



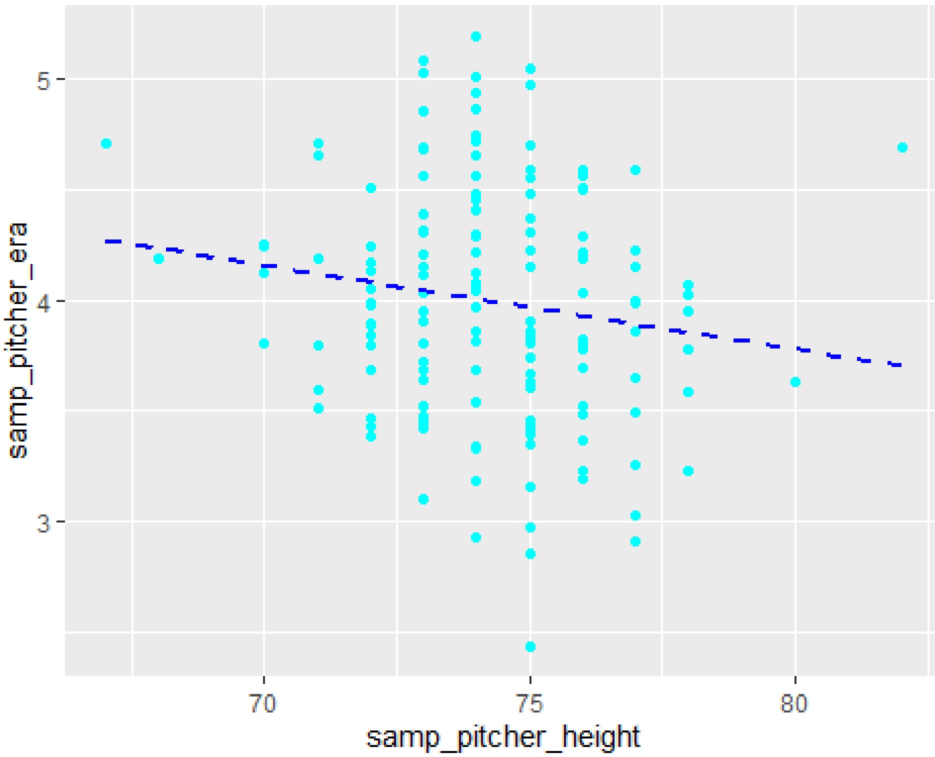

```
soH_weight <- lm(samp_batter_so ~ samp_batter_weight)
msummary(soH_weight)

##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.739e-01  4.375e-02   3.975  0.00011 ***
## samp_batter_weight -7.845e-06  2.260e-04  -0.035  0.97236
##
## Residual standard error: 0.05927 on 148 degrees of freedom
## Multiple R-squared:  8.138e-06,  Adjusted R-squared:  -0.006749
## F-statistic: 0.001204 on 1 and 148 DF,  p-value: 0.9724

gf_point(samp_batter_so ~ samp_batter_weight, color = "pink")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```



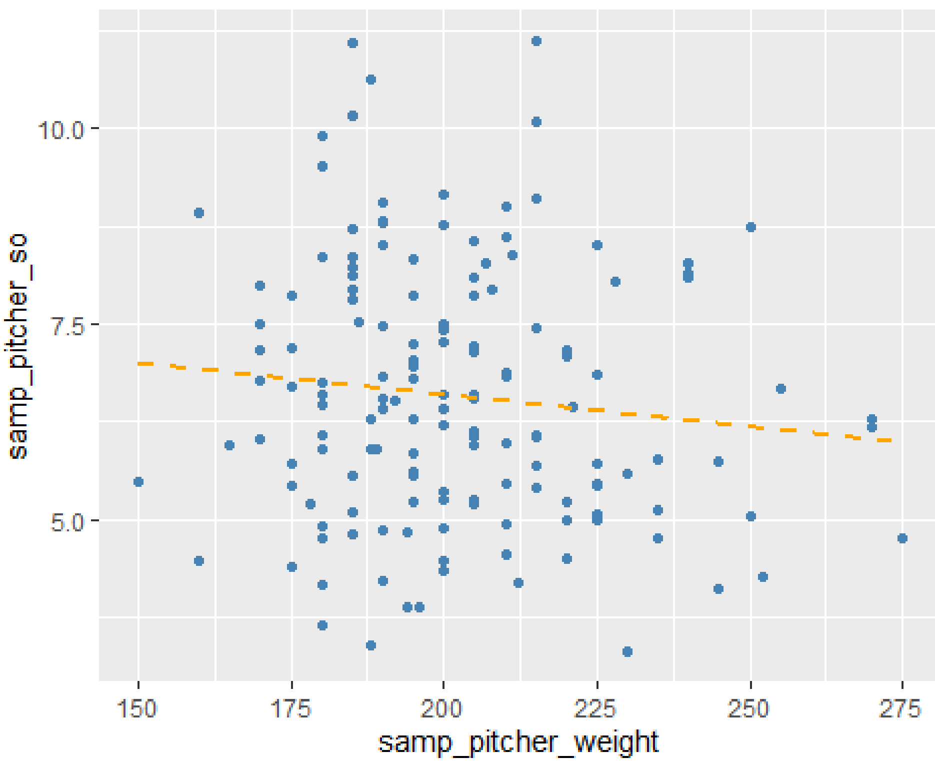

```
#Now we do the same as above except with the pitchers data
#We dont expect to get any significant results after seeing the hitting
graphs
set.seed(51321)
n <- 150
samp_pitcher_height <- sample(pitchers3$height,n)
samp_pitcher_weight <- sample(pitchers3$weight,n)

samp_pitcher_era <- sample(pitchers3$ERA, n)
samp_pitcher_so <- sample(pitchers3$SOpG, n)

era_weight <- lm(samp_pitcher_era ~ samp_pitcher_weight)
msummary(era_weight)

##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.519340   0.397582  11.367   <2e-16 ***
## samp_pitcher_weight -0.002588   0.001957  -1.322    0.188
##
## Residual standard error: 0.5409 on 148 degrees of freedom
## Multiple R-squared:  0.01167,    Adjusted R-squared:  0.004994
## F-statistic: 1.748 on 1 and 148 DF,  p-value: 0.1882

gf_point(samp_pitcher_era ~ samp_pitcher_weight, color = "cyan")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```



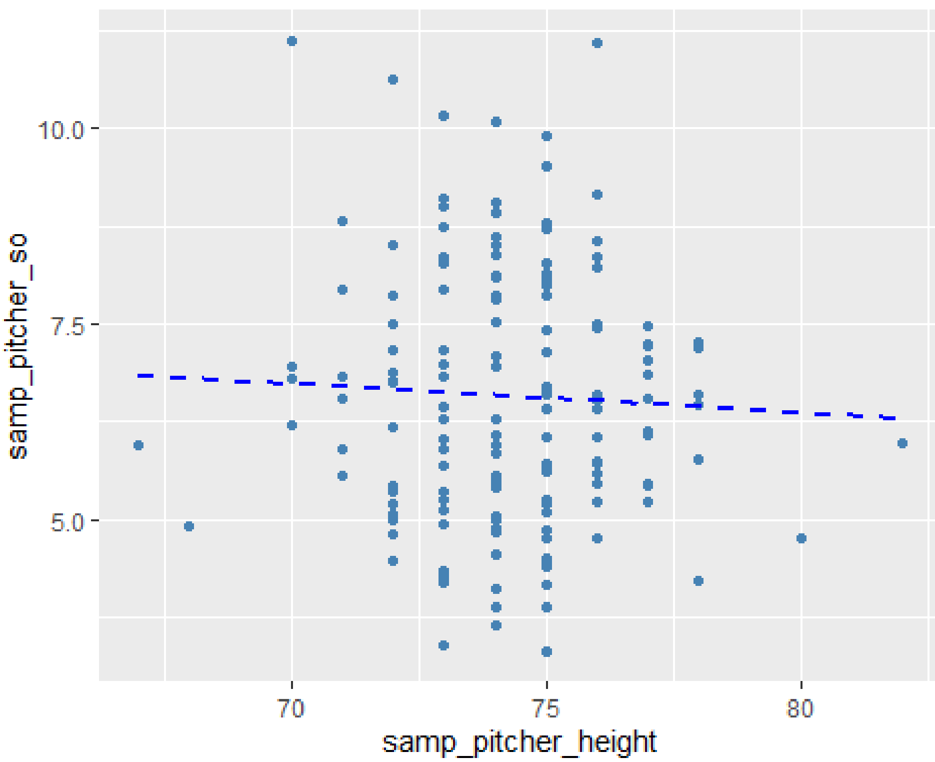

```
era_height <- lm(samp_pitcher_era ~ samp_pitcher_height)
msummary(era_height)

##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.81856    1.53201   4.451 1.67e-05 ***
## samp_pitcher_height -0.03802    0.02063  -1.843   0.0674 .
##
## Residual standard error: 0.5379 on 148 degrees of freedom
## Multiple R-squared:  0.02242,    Adjusted R-squared:  0.01582
## F-statistic: 3.395 on 1 and 148 DF,  p-value: 0.0674

gf_point(samp_pitcher_era ~ samp_pitcher_height, color = "cyan")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```




```
soP_weight <- lm(samp_pitcher_so ~ samp_pitcher_weight)
msummary(soP_weight)

##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.224363   1.202268   6.841 1.94e-10 ***
## samp_pitcher_weight -0.008174   0.005919  -1.381    0.169
##
## Residual standard error: 1.636 on 148 degrees of freedom
## Multiple R-squared:  0.01272,    Adjusted R-squared:  0.006053
## F-statistic: 1.907 on 1 and 148 DF,  p-value: 0.1693

gf_point(samp_pitcher_so ~ samp_pitcher_weight, color = "steelblue")%>%
  gf_lm(size = 1, color = "orange", linetype = "dashed")
```




```
soP_height <- lm(samp_pitcher_so ~ samp_pitcher_height)
msummary(soP_height)

##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9.28859    4.68273   1.984   0.0492 *
## samp_pitcher_height -0.03657    0.06307  -0.580   0.5629
##
## Residual standard error: 1.644 on 148 degrees of freedom
## Multiple R-squared:  0.002267,   Adjusted R-squared:  -0.004474
## F-statistic: 0.3363 on 1 and 148 DF,  p-value: 0.5629

gf_point(samp_pitcher_so ~ samp_pitcher_height, color = "steelblue")%>%
  gf_lm(size = 1, color = "blue", linetype = "dashed")
```




```
#None of the linear regressions really showed a strong correlation
#Decided to stick with the ones that showed the most:
#Batter Avg vs Height
#Pitcher ERA vs Height

#Wanted to see if running an accuracy test and decision tree on Avg and Era
would provide any interesting results
#Started with avg but had to make it a factor
#TRUE will be a good hitter (>= .250)
#FALSE will be a bad hitter (< .250)

batters4 <- batters3%>%
  mutate(good_hitter = ifelse(AVG >= 0.264,TRUE,FALSE))%>%
  select(good_hitter, birthCountry, birthState, weight, height, bats)%>%
  mutate(good_hitter = as.factor(good_hitter))
head(batters4)

## # A tibble: 6 x 6
##   good_hitter birthCountry birthState weight height bats
##   <fct>       <chr>        <chr>       <int>  <int> <fct>
## 1 FALSE       USA          OH            180     71 R
## 2 TRUE        Venezuela    Aragua        220     72 L
## 3 TRUE        Cuba         Cienfuegos    250     75 R
## 4 FALSE       USA          NC            205     73 L
## 5 FALSE       USA          PA            245     75 L
## 6 FALSE       USA          AL            195     71 R
```

```r
#Split the hitters into training and test data sets with the proportion at
75%
nrow(batters4)

## [1] 1248

set.seed(51321)
split_batters <- batters4%>%
  initial_split(prop = 0.75)
train_bat <- split_batters%>%
  training()
test_bat <- split_batters%>%
  testing()
list(train_bat, test_bat)%>%
  map_int(nrow)

## [1] 936 312

#Built the null model as "hitter_null"
hitter_null <- logistic_reg(mode = "classification") %>%
  set_engine("glm") %>%
  fit(good_hitter ~ 1, data = train_bat)
#Created null_hit_pred to test the accuracy of the null prediction
null_hit_pred <- train_bat%>%
  bind_cols(predict(hitter_null, new_data = train_bat, type = "class"))%>%
  rename(good_null = .pred_class)
accuracy(null_hit_pred, good_hitter, good_null)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.505

#The null model seems to be a little over 50% accuracy
#Now we want to see if we can improve it

#Built the first model with only height as a variable
bat_model1 <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_hitter ~ height, data = train_bat)

bat_pred1 <- train_bat%>%
  bind_cols(predict(bat_model1, new_data = train_bat, type = "class"))%>%
  rename(hit_model1 = .pred_class)
accuracy(bat_pred1, good_hitter, hit_model1)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.505

#The estimate is the same as the null
```

```
#Decided to add weight onto the first model
bat_model2 <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_hitter ~ height + weight, data = train_bat)

bat_pred2 <- train_bat%>%
  bind_cols(predict(bat_model2, new_data = train_bat, type = "class"))%>%
  rename(hit_model2 = .pred_class)
accuracy(bat_pred2, good_hitter, hit_model2)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.516

#The estimate increased a little bit
#Now we're going to try to add country and state of birth because in warmer
places they can play all year

#Since height and weight didn't really help too much, we added birthCountry
and state
bat_model3 <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_hitter ~ height + weight + birthCountry + birthState, data =
train_bat)

## Warning: glm.fit: algorithm did not converge

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

bat_pred3 <- train_bat%>%
  bind_cols(predict(bat_model3, new_data = train_bat, type = "class"))%>%
  rename(hit_model3 = .pred_class)

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

accuracy(bat_pred3, good_hitter, hit_model3)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.573

#The prediction increased by about 7% now

bat_test <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_hitter ~ height + weight + birthCountry + birthState, data =
test_bat)
```

```
## Warning: glm.fit: algorithm did not converge

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

```
bat_test
```

```
## parsnip model object
##
## Fit time:  61ms
##
## Call:  stats::glm(formula = good_hitter ~ height + weight + birthCountry +
##     birthState, family = stats::binomial, data = data)
##
## Coefficients:
##                   (Intercept)                         height
##                     2.021e+14                     -3.266e-02
##                        weight                birthCountryCAN
##                    -2.896e-03                      4.301e+15
##              birthCountryCuba                birthCountryD.R.
##                    -4.706e+15                      1.688e+13
##           birthCountryJamaica             birthCountryMexico
##                    -2.021e+14                     -2.021e+14
##         birthCountryNicaragua             birthCountryPanama
##                     4.301e+15                      4.301e+15
##               birthCountryUSA          birthCountryVenezuela
##                    -2.021e+14                     -2.380e+15
##          birthStateAnzoategui                   birthStateAR
##                     2.178e+15                      4.504e+15
##              birthStateAragua         birthStateAtlantico Sur
##                     6.682e+15                             NA
##                  birthStateAZ                   birthStateBC
##                    -1.547e+00                      1.429e+06
##             birthStateBolivar                   birthStateCA
##                     6.682e+15                     -1.043e+00
##            birthStateCamaguey             birthStateCarabobo
##                     9.007e+15                      2.178e+15
##            birthStateChiriqui                birthStateColon
##                    -9.007e+15                             NA
##                  birthStateCT                   birthStateDE
##                    -4.504e+15                     -8.273e-02
##     birthStateDistrito Federal     birthStateDistrito Nacional
##                     2.178e+15                     -2.190e+14
##              birthStateDuarte              birthStateEl Seibo
##                    -2.190e+14                     -2.190e+14
##           birthStateEspaillat                   birthStateFL
##                    -2.190e+14                     -5.398e-01
##                  birthStateGA                birthStateGranma
##                    -3.313e-01                      4.504e+15
##                  birthStateHI                   birthStateIA
##                    -2.618e+01                     -1.445e+00
```

```
##                  birthStateIL                        birthStateIN
##                    -1.336e+00                          -7.614e-01
##            birthStateKingston                        birthStateKS
##                            NA                          -1.873e+00
##                  birthStateKY                        birthStateLA
##                    -1.429e+00                           2.685e+01
##           birthStateLa Habana                  birthStateLa Vega
##                     4.504e+15                          -2.190e+14
##                birthStateLara                        birthStateMA
##                     2.178e+15                          -2.501e+07
##             birthStateMaracay                        birthStateMI
##                     2.178e+15                           3.111e-01
##             birthStateMiranda                        birthStateMN
##                     2.178e+15                           2.678e+01
##                  birthStateMO             birthStateMonte Cristi
##                    -7.454e-01                          -2.190e+14
##                  birthStateMS                        birthStateNC
##                     3.255e-01                          -7.581e-01
##                  birthStateND                        birthStateNE
##                     2.686e+01                          -7.544e-01
##                  birthStateNJ                        birthStateNM
##                    -1.401e+00                          -2.637e+01
##                  birthStateNY                        birthStateOH
##                    -8.263e-01                          -5.701e-01
##                  birthStateOK                        birthStateON
##                    -1.483e+00                           1.103e+06
##                  birthStateOR                        birthStatePA
##                    -4.658e-01                          -4.225e-01
##             birthStatePeravia                        birthStateRI
##                    -2.190e+14                          -4.504e+15
##              birthStateSamana            birthStateSan Cristobal
##                    -2.190e+14                          -2.190e+14
## birthStateSan Pedro de Macoris                birthStateSantiago
##                    -2.190e+14                          -2.190e+14
##           birthStateSao Paulo                        birthStateSC
##                    -4.706e+15                           2.567e-01
##                  birthStateSD                        birthStateSK
##                    -8.689e-01                                  NA
##              birthStateSonora                        birthStateTN
##                            NA                          -1.504e+00
##                  birthStateTX                        birthStateVA
##                    -9.026e-01                           3.060e-01
##         birthStateVilla Clara                        birthStateWA
##                            NA                          -7.151e-02
##                  birthStateWI                        birthStateWV
##                    -8.238e-02                          -2.620e+01
##                  birthStateWY                    birthStateZulia
##                    -8.138e-01                           2.178e+15
##
## Degrees of Freedom: 311 Total (i.e. Null);  234 Residual
```

```
## Null Deviance:       432.4
## Residual Deviance: 347.3     AIC: 503.3
```

```
bat_test_pred <- test_bat%>%
  bind_cols(predict(bat_test, new_data = test_bat, type = "class"))%>%
  rename(hit_test = .pred_class)
```

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

```
accuracy(bat_test_pred, good_hitter, hit_test)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.676
```

```
#The testing estimate is even better
#Therefore this is a pretty decent model to use for predicting good_hitters
#However, 67.6% still isn't that good of a prediction, but for only using
non-game factors, its pretty good
#But since it increased so much, we may have overfit the data
```

```
fav_stats(pitchers3$ERA)
```

```
##       min       Q1   median       Q3      max     mean        sd    n
missing
##  2.208517 3.628254 3.990284 4.357752 6.029462 3.991612 0.5588547 1033
0
```

```
#Now want to build a model for pitchers era
#Will classify a good pitcher with an era <= 4

pitchers4 <- pitchers3%>%
  mutate(good_pitcher = ifelse(ERA <= 4.00, TRUE, FALSE))%>%
  select(good_pitcher, ERA, birthCountry, weight, height)%>%
  mutate(good_pitcher = as.factor(good_pitcher))
head(pitchers4)
```

```
## # A tibble: 6 x 5
##   good_pitcher   ERA birthCountry weight height
##   <fct>        <dbl> <chr>         <int>  <int>
## 1 TRUE          3.80 USA             190     75
## 2 FALSE         4.39 USA             200     78
## 3 FALSE         4.25 USA             200     75
## 4 FALSE         4.92 USA             185     75
## 5 TRUE          3.97 USA             210     74
## 6 FALSE         4.17 USA             180     75
```

```
#Set the same seed for the pitchers data and split the training and test by
75%
nrow(pitchers4)

## [1] 1033

set.seed(51321)
split_batters <- pitchers4%>%
  initial_split(prop = 0.75)
train_pitch <- split_batters%>%
  training()
test_pitch <- split_batters%>%
  testing()
list(train_pitch, test_pitch)%>%
  map_int(nrow)

## [1] 775 258

#Built the null model as "pitcher_null"
pitcher_null <- logistic_reg(mode = "classification") %>%
  set_engine("glm") %>%
  fit(good_pitcher ~ 1, data = train_pitch)
#Created null_pitch_pred to test the accuracy of the null prediction
null_pitch_pred <- train_pitch%>%
  bind_cols(predict(pitcher_null, new_data = train_pitch, type = "class"))%>%
  rename(good_pitch_null = .pred_class)
accuracy(null_pitch_pred, good_pitcher, good_pitch_null)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.510

#The null model seems to be a little over 50% accuracy
#Now we want to see if we can improve it

pitch_model1 <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_pitcher ~ height + weight + birthCountry, data = train_pitch)

pitch_pred1 <- train_pitch%>%
  bind_cols(predict(pitch_model1, new_data = train_pitch, type = "class"))%>%
  rename(pitch1 = .pred_class)
accuracy(pitch_pred1, good_pitcher, pitch1)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.581

#The estimate increased by around 7%, but the estimate still isn't very
strong
```

```
#It was definitely easier to predict the hitters average rather than the
pitchers ERA

#Lets see if the test set performs better
pitch_test <- logistic_reg(mode = "classification")%>%
  set_engine("glm")%>%
  fit(good_pitcher ~ height + weight + birthCountry, data = test_pitch)

pitch_test_pred <- test_pitch%>%
  bind_cols(predict(pitch_test, new_data = test_pitch, type = "class"))%>%
  rename(testing_pitch = .pred_class)
accuracy(pitch_test_pred, good_pitcher, testing_pitch)

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.585

#The testing set was a tiny bit more accurate but still not as good as the
hitters data

#Created a decision tree to see what the different variables affected
form <- as.formula("good_pitcher ~ height + weight + birthCountry")
decision_tree <- decision_tree(mode = "classification")%>%
  set_engine("rpart")%>%
  fit(form, data = train_pitch)
decision_tree

## parsnip model object
##
## Fit time:  11ms
## n= 775
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 775 380 TRUE (0.4903226 0.5096774)
##    2) weight>=201.5 335 142 FALSE (0.5761194 0.4238806) *
##    3) weight< 201.5 440 187 TRUE (0.4250000 0.5750000)
##      6) birthCountry=D.R.,Mexico,Nicaragua,Panama,South Korea 36  14 FALSE
(0.6111111 0.3888889) *
##      7) birthCountry=CAN,Germany,Japan,Netherlands,USA,Venezuela 404 165
TRUE (0.4084158 0.5915842)
##       14) weight>=186 205  96 TRUE (0.4682927 0.5317073)
##         28) weight< 194.5 69  30 FALSE (0.5652174 0.4347826) *
##         29) weight>=194.5 136  57 TRUE (0.4191176 0.5808824) *
##       15) weight< 186 199  69 TRUE (0.3467337 0.6532663) *

#Conclusion:
#Looking at the graphs and models that we made, you can't predict if a hitter
or pitcher will be good based off
```

```
#physical attributes
#What surprised us the most was that there weren't even significant data for heavier hitters having
#a higher slugging percentage
#Also, the fact that there was even a little correlation between a hitters height, and their average surprised me
#Only because the taller hitters had a higher average
#Especially those that were 75 inches tall
#This could be because they walk less, with having a larger strike zone.
#Looking at the graph for OBP, the higher values tend to go to shorter players
#Which would back up the last statement
#As for the pitchers, it seemed that physical attributes didn't seem to affect their stats as much
#That could be because pitching is such a game of mechanics rather than physical status
#Both short pitchers such as Marcus Stroman, and formarly Tim Lincecum, throw hard
#While hitters who are big, tend to always be power hitters (Aaron Judge, Giancarlo Stanton, David Ortiz...)
```