# Project 1

Ashton Passmore

2023-03-26

```
library(dplyr)
```

```
##
## 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 1.3.2
## --

## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v stringr 1.5.0
## v tidyr   1.3.0     v forcats 1.0.0
## v readr   2.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(Lahman)
```

```
## Warning: package 'Lahman' was built under R version 4.2.3
```

## Data set information

The data set that I chose for this project is the teams data set from the Lahman package.

This data set has 2985 observations and 48 variables and it gives yearly statistics for Major League Baseball teams from 1871 - 2021.

The problem I want to solve is I want to know how many home runs each team will give up in 2022 based on data from 2005-2021.I'm predicting for 2022 because it's not in the data set yet and since the season is over I can manually feed the independent variable information into the model I'm going to make to hopefully accurately predict how many home runs a team will give up in 2022. Afterwards I will check my predictions against the real number of home runs team gave up in 2022 to see how accurate my model is. (I'm starting in 2005 because pre 2005 there was a problem in baseball with steroid use and this inflated home runs allowed by a significant margin and all the teams from 2005 are the same teams as today.)

# Cleaning

filter data set to only give data from 2005-2021

```
revisedTeams <- Teams %>%
filter(yearID >= 2005)
head(revisedTeams)
```

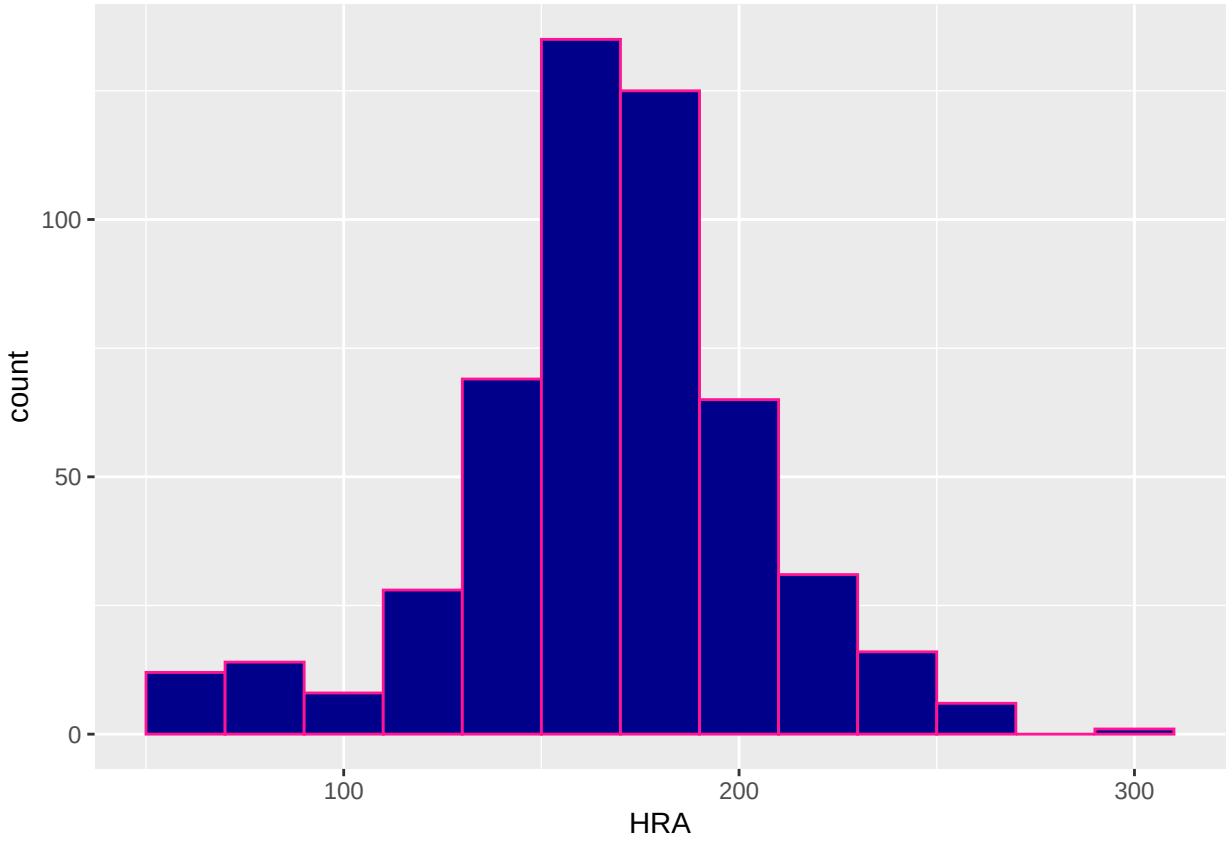
```
##   yearID lgID teamID franchID divID Rank   G Ghome  W  L DivWin WCWin LgWin
## 1   2005   NL    ARI      ARI     W    2 162    81 77 85      N     N     N
## 2   2005   NL    ATL      ATL     E    1 162    81 90 72      Y     N     N
## 3   2005   AL    BAL      BAL     E    4 162    81 74 88      N     N     N
## 4   2005   AL    BOS      BOS     E    2 162    81 95 67      N     Y     N
## 5   2005   AL    CHA      CHW     C    1 162    81 99 63      Y     N     Y
## 6   2005   NL    CHN      CHC     C    4 162    81 79 83      N     N     N
##   WSWin   R   AB    H X2B X3B  HR  BB   SO  SB CS HBP SF  RA  ER  ERA CG SHO SV
## 1     N 696 5550 1419 291  27 191 606 1094  67 26  55 45 856 783 4.84  6  10 45
## 2     N 769 5486 1453 308  37 184 534 1084  92 32  45 46 674 639 3.98  8  12 38
## 3     N 729 5551 1492 296  27 189 447  902  83 37  54 42 800 724 4.56  2   9 38
## 4     N 910 5626 1579 339  21 199 653 1044  45 12  47 63 805 752 4.74  6   8 38
## 5     Y 741 5529 1450 253  23 200 435 1002 137 67  79 49 645 592 3.61  9  10 54
## 6     N 703 5584 1506 323  23 194 419  920  65 39  50 37 714 671 4.19  8  10 39
##   IPouts   HA HRA BBA  SOA   E  DP    FP                 name
## 1   4369 1580 193 537 1038  94 159 0.985 Arizona Diamondbacks
## 2   4331 1487 145 520  929  86 170 0.986       Atlanta Braves
## 3   4283 1458 180 580 1052 107 154 0.982    Baltimore Orioles
## 4   4287 1550 164 440  959 109 135 0.982       Boston Red Sox
## 5   4427 1392 167 459 1040  94 166 0.985    Chicago White Sox
## 6   4320 1357 186 576 1256 101 136 0.983         Chicago Cubs
##                         park attendance BPF PPF teamIDBR teamIDlahman45
## 1           Bank One Ballpark    2059424 103 105      ARI            ARI
## 2                Turner Field    2521167 101 100      ATL            ATL
## 3 Oriole Park at Camden Yards    2624740  99  99      BAL            BAL
## 4              Fenway Park II    2847888 104 104      BOS            BOS
## 5         U.S. Cellular Field    2342833 103 103      CHW            CHA
## 6               Wrigley Field    3099992 104 104      CHC            CHN
##   teamIDretro
## 1         ARI
## 2         ATL
## 3         BAL
## 4         BOS
## 5         CHA
## 6         CHN
```

## Exploratory data analysis

plot of revised data frame histogram of Home runs allowed to see distribution of the data

```
g <- ggplot(data = revisedTeams, aes(x = yearID, y = HRA, color = franchID)) +
  geom_point()
gg <- ggplot(data = revisedTeams, aes(x = HRA)) +
  geom_histogram(binwidth = 20, color = "deeppink", fill = "darkblue")
g
```


```
gg
```




After plotting the data I forgot about the 2020 season which was shortened to 60 games instead of 162 which, gave an unrealistic amount of 0 - 80 home runs allowed so we're going to remove that year.

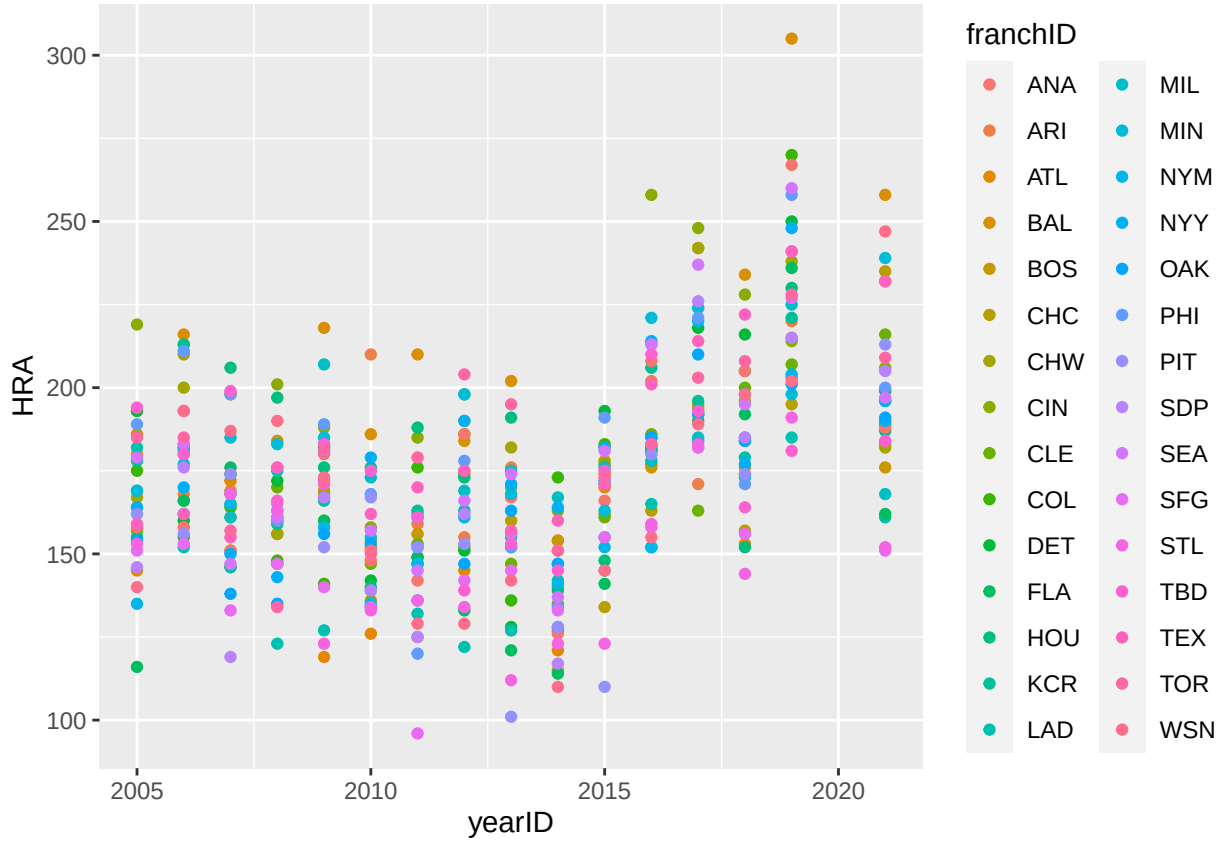
```
RevisedTeams2 <- filter(revisedTeams, yearID != 2020)
head(RevisedTeams2)
```

```
##   yearID lgID teamID franchID divID Rank   G Ghome  W  L DivWin WCWin LgWin
## 1   2005   NL    ARI      ARI     W    2 162    81 77 85      N     N     N
## 2   2005   NL    ATL      ATL     E    1 162    81 90 72      Y     N     N
## 3   2005   AL    BAL      BAL     E    4 162    81 74 88      N     N     N
## 4   2005   AL    BOS      BOS     E    2 162    81 95 67      N     Y     N
## 5   2005   AL    CHA      CHW     C    1 162    81 99 63      Y     N     Y
## 6   2005   NL    CHN      CHC     C    4 162    81 79 83      N     N     N
##   WSWin   R   AB    H X2B X3B  HR  BB   SO  SB CS HBP SF  RA  ER  ERA CG SHO SV
## 1     N 696 5550 1419 291  27 191 606 1094  67 26  55 45 856 783 4.84  6  10 45
## 2     N 769 5486 1453 308  37 184 534 1084  92 32  45 46 674 639 3.98  8  12 38
## 3     N 729 5551 1492 296  27 189 447  902  83 37  54 42 800 724 4.56  2   9 38
## 4     N 910 5626 1579 339  21 199 653 1044  45 12  47 63 805 752 4.74  6   8 38
## 5     Y 741 5529 1450 253  23 200 435 1002 137 67  79 49 645 592 3.61  9  10 54
## 6     N 703 5584 1506 323  23 194 419  920  65 39  50 37 714 671 4.19  8  10 39
##   IPouts   HA HRA BBA  SOA   E  DP    FP                name
## 1   4369 1580 193 537 1038  94 159 0.985 Arizona Diamondbacks
## 2   4331 1487 145 520  929  86 170 0.986       Atlanta Braves
## 3   4283 1458 180 580 1052 107 154 0.982    Baltimore Orioles
## 4   4287 1550 164 440  959 109 135 0.982       Boston Red Sox
## 5   4427 1392 167 459 1040  94 166 0.985    Chicago White Sox
## 6   4320 1357 186 576 1256 101 136 0.983         Chicago Cubs
```

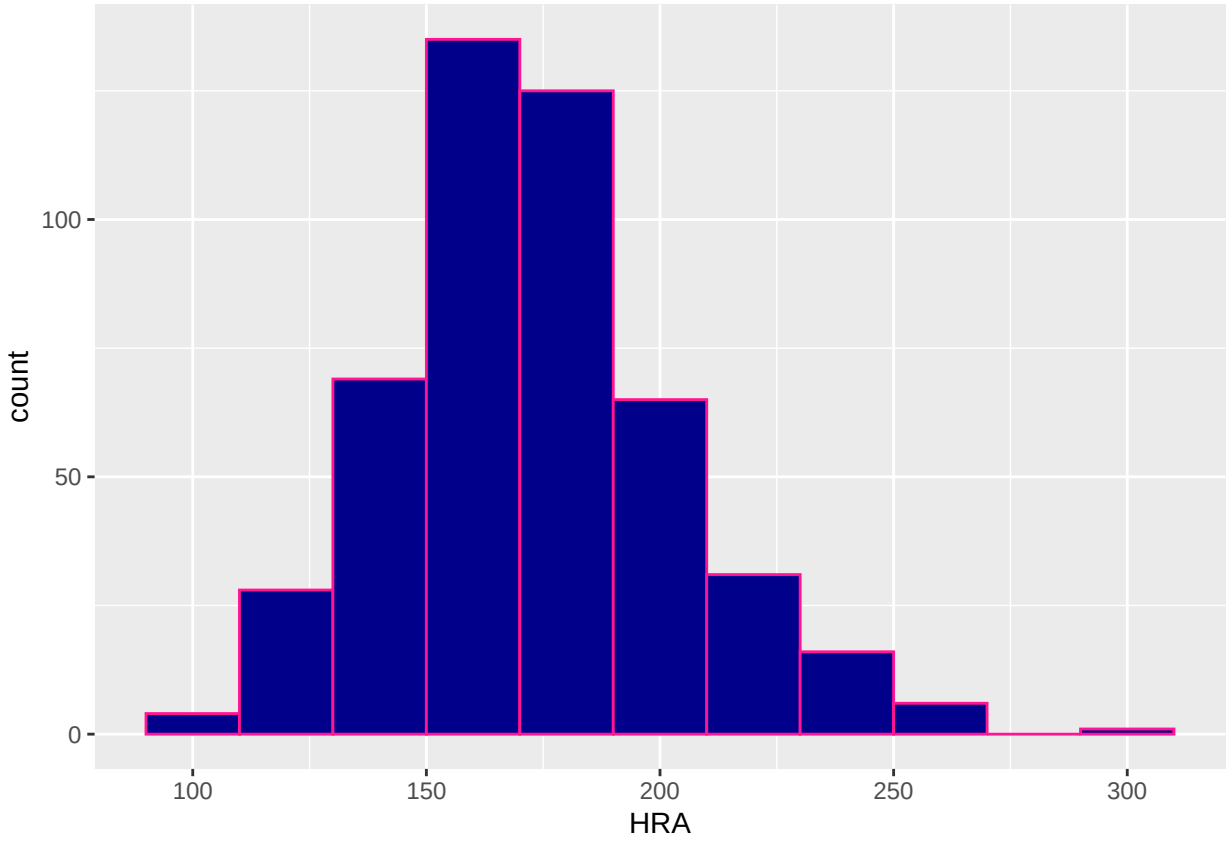
```
##                          park attendance BPF PPF teamIDBR teamIDlahman45
## 1           Bank One Ballpark    2059424 103 105      ARI            ARI
## 2                Turner Field    2521167 101 100      ATL            ATL
## 3 Oriole Park at Camden Yards    2624740  99  99      BAL            BAL
## 4               Fenway Park II    2847888 104 104      BOS            BOS
## 5         U.S. Cellular Field    2342833 103 103      CHW            CHA
## 6                Wrigley Field    3099992 104 104      CHC            CHN
##   teamIDretro
## 1         ARI
## 2         ATL
## 3         BAL
## 4         BOS
## 5         CHA
## 6         CHN
```

revised plots with 2020 removed because of the shortened season

```
g <- ggplot(data = RevisedTeams2, aes(x = yearID, y = HRA, color = franchID)) +
  geom_point()
gg <- ggplot(data = RevisedTeams2, aes(x = HRA)) +
  geom_histogram(binwidth = 20, color = "deeppink", fill = "darkblue")
g
```




```
gg
```




the distribution of Home Runs Allowed looks like a normal distribution maybe slightly left skewed.

## Building the model

I'm using the information out of the book and separating the data into two splits. One of the splits is 80% of the data which is our training set which will be used to train the model and the other 20% is only used for testing our model. The method I'll be using to build the model is step wise linear regression.

**setting up the test and training sets**

```
set.seed(1234)
# training set w/ 80% of total data
train <- RevisedTeams2 %>%
  dplyr::sample_frac(.8)
# test set with remaining 20% of the data
test <- dplyr::anti_join(RevisedTeams2, train)
```

```
## Joining with `by = join_by(yearID, lgID, teamID, franchID, divID, Rank, G,
## Ghome, W, L, DivWin, WCWin, LgWin, WSWin, R, AB, H, X2B, X3B, HR, BB, SO, SB,
## CS, HBP, SF, RA, ER, ERA, CG, SHO, SV, IPouts, HA, HRA, BBA, SOA, E, DP, FP,
## name, park, attendance, BPF, PPF, teamIDBR, teamIDlahman45, teamIDretro)`
```

```
# checking to make sure everything is good
nrow(RevisedTeams2)
```

```
## [1] 480
```

```
nrow(train)
```

```
## [1] 384
```

```
nrow(test)
```

```
## [1] 96
```

```
# the number of rows of test and train add up to the number of rows in
# RevisedTeams2 so we are good.
```

**Choosing Independent Variables** Our dependent variable is home runs against (HRA) but, what independent variables affect our dependent variable?

When looking at the data set any stats that deal with pitching have some sort of relevance to home runs against since you can only give up home runs when your team is on defense. For my independent variables I'm choosing pretty much all of the pitching variables because they could all have an effect on home runs against. I'm choosing Wins(W), Losses(L), Runs Against(RA), Earned Runs (ER), Earned Run Average (ERA), Complete Games(CG), Shut Outs (SHO), Saves(SV), Outs Pitched (IPouts), Hits against (HA), Walks Against (BBA), and finally Strike Outs Against (SOA).

```
fit <- lm(HRA ~ W + L + RA + ER + ERA + CG + SHO + SV + IPouts + HA + BBA + SOA,
          data = train)
summary(fit)
```

```
##
## Call:
## lm(formula = HRA ~ W + L + RA + ER + ERA + CG + SHO + SV + IPouts +
##     HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.070 -11.823  -0.325  11.942  53.568
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107.919753 740.691727   0.146   0.8842
## W             1.957218   3.128842   0.626   0.5320
## L             1.830428   3.134300   0.584   0.5596
## RA            0.078679   0.073532   1.070   0.2853
## ER            0.887815   0.854107   1.039   0.2993
## ERA         -69.310871 136.132108  -0.509   0.6110
## CG            0.322205   0.355674   0.906   0.3656
## SHO          -0.770897   0.300017  -2.570   0.0106 *
## SV            0.251014   0.167728   1.497   0.1354
## IPouts       -0.071149   0.133254  -0.534   0.5937
## HA           -0.194038   0.023821  -8.146 5.82e-15 ***
## BBA          -0.170521   0.019913  -8.563 2.96e-16 ***
## SOA           0.052261   0.008656   6.038 3.79e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.86 on 371 degrees of freedom
## Multiple R-squared:  0.7124, Adjusted R-squared:  0.7031
## F-statistic: 76.59 on 12 and 371 DF,  p-value: < 2.2e-16
```

now I'll perform step wise regression to improve the model (get all variables 0.05 p values and lower) I got rid of ERA because it was the least significant independent variable

```
fit1 <- lm(HRA ~ W + L + RA + ER + CG + SHO + SV + IPouts + HA + BBA + SOA,
          data = train)
summary(fit1)
```

```
##
## Call:
## lm(formula = HRA ~ W + L + RA + ER + CG + SHO + SV + IPouts +
##     HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.937 -12.016  -0.325  11.812  53.583
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.731e+02  4.935e+02  -0.351  0.72594
## W            1.913e+00  3.125e+00   0.612  0.54081
## L            1.779e+00  3.130e+00   0.568  0.57006
## RA           7.987e-02  7.342e-02   1.088  0.27738
## ER           4.546e-01  7.475e-02   6.082 2.95e-09 ***
## CG           3.332e-01  3.547e-01   0.939  0.34810
## SHO         -7.817e-01  2.990e-01  -2.615  0.00929 **
## SV           2.500e-01  1.675e-01   1.492  0.13651
## IPouts      -4.549e-03  2.539e-02  -0.179  0.85792
## HA          -1.940e-01  2.380e-02  -8.151 5.56e-15 ***
## BBA         -1.707e-01  1.989e-02  -8.580 2.60e-16 ***
## SOA          5.236e-02  8.645e-03   6.057 3.39e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.85 on 372 degrees of freedom
## Multiple R-squared:  0.7122, Adjusted R-squared:  0.7037
## F-statistic: 83.69 on 11 and 372 DF,  p-value: < 2.2e-16
```

now I'm just going to continue getting rid of independent variables until all of the independent variables have a p-value of 0.05 or less. Next up to get rid of is IPouts.

```
fit2 <- lm(HRA ~ W + L + RA + ER + CG + SHO + SV + HA + BBA + SOA,
          data = train)
summary(fit2)
```

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

```
## lm(formula = HRA ~ W + L + RA + ER + CG + SHO + SV + HA + BBA +
##     SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.173 -12.005  -0.353  11.859  53.460
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.728e+02  4.928e+02  -0.351  0.72599
## W            1.797e+00  3.053e+00   0.589  0.55649
## L            1.667e+00  3.062e+00   0.544  0.58657
## RA           8.166e-02  7.265e-02   1.124  0.26171
## ER           4.549e-01  7.464e-02   6.095 2.73e-09 ***
## CG           3.343e-01  3.541e-01   0.944  0.34576
## SHO         -7.808e-01  2.985e-01  -2.615  0.00927 **
## SV           2.488e-01  1.672e-01   1.488  0.13758
## HA          -1.955e-01  2.215e-02  -8.828  < 2e-16 ***
## BBA         -1.713e-01  1.960e-02  -8.737  < 2e-16 ***
## SOA          5.199e-02  8.376e-03   6.207 1.44e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.83 on 373 degrees of freedom
## Multiple R-squared:  0.7122, Adjusted R-squared:  0.7045
## F-statistic:  92.3 on 10 and 373 DF,  p-value: < 2.2e-16
```

Getting rid of L

```
fit3 <- lm(HRA ~ W + RA + ER + CG + SHO + SV + HA + BBA + SOA,
          data = train)
summary(fit3)
```

```
##
## Call:
## lm(formula = HRA ~ W + RA + ER + CG + SHO + SV + HA + BBA + SOA,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.144 -12.043  -0.321  11.954  53.473
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 94.698900  35.263235   2.685  0.00757 **
## W            0.136487   0.117057   1.166  0.24436
## RA           0.082006   0.072574   1.130  0.25921
## ER           0.453903   0.074544   6.089 2.82e-09 ***
## CG           0.319821   0.352810   0.906  0.36526
## SHO         -0.774307   0.298018  -2.598  0.00974 **
## SV           0.248842   0.167038   1.490  0.13714
## HA          -0.193962   0.021942  -8.840  < 2e-16 ***
## BBA         -0.171622   0.019570  -8.770  < 2e-16 ***
```

```
## SOA          0.052225   0.008357   6.249 1.12e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.81 on 374 degrees of freedom
## Multiple R-squared:  0.712,  Adjusted R-squared:  0.705
## F-statistic: 102.7 on 9 and 374 DF,  p-value: < 2.2e-16
```

getting rid of CG

```
fit4 <- lm(HRA ~ W + RA + ER + SHO + SV + HA + BBA + SOA,
          data = train)
summary(fit4)
```

```
##
## Call:
## lm(formula = HRA ~ W + RA + ER + SHO + SV + HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.926 -12.290   0.014  12.050  53.924
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.124232  34.290521   2.978  0.00309 **
## W             0.152779   0.115641   1.321  0.18726
## RA            0.078825   0.072472   1.088  0.27744
## ER            0.454527   0.074523   6.099 2.66e-09 ***
## SHO          -0.720077   0.291882  -2.467  0.01407 *
## SV            0.208238   0.160882   1.294  0.19634
## HA           -0.193715   0.021935  -8.831  < 2e-16 ***
## BBA          -0.174278   0.019344  -9.009  < 2e-16 ***
## SOA           0.049357   0.007733   6.383 5.14e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.81 on 375 degrees of freedom
## Multiple R-squared:  0.7113, Adjusted R-squared:  0.7052
## F-statistic: 115.5 on 8 and 375 DF,  p-value: < 2.2e-16
```

getting rid of RA

```
fit4 <- lm(HRA ~ W + ER + SHO + SV + HA + BBA + SOA,
          data = train)
summary(fit4)
```

```
##
## Call:
## lm(formula = HRA ~ W + ER + SHO + SV + HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -48.023 -12.003   0.194  11.853  54.077
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.839118  34.292564   2.999  0.00289 **
## W             0.138640   0.114936   1.206  0.22849
## ER            0.530162   0.026801  19.781  < 2e-16 ***
## SHO          -0.754687   0.290213  -2.600  0.00968 **
## SV            0.193174   0.160324   1.205  0.22900
## HA           -0.189525   0.021599  -8.775  < 2e-16 ***
## BBA          -0.171944   0.019230  -8.942  < 2e-16 ***
## SOA           0.050077   0.007707   6.498 2.59e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.81 on 376 degrees of freedom
## Multiple R-squared:  0.7104, Adjusted R-squared:  0.705
## F-statistic: 131.8 on 7 and 376 DF,  p-value: < 2.2e-16
```

getting rid of SV

```
fit5 <- lm(HRA ~ W + ER + SHO + HA + BBA + SOA,
          data = train)
summary(fit5)
```

```
##
## Call:
## lm(formula = HRA ~ W + ER + SHO + HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.438 -12.461   0.127  11.362  53.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 105.096349  34.261866   3.067  0.00231 **
## W             0.201025   0.102675   1.958  0.05098 .
## ER            0.522314   0.026013  20.079  < 2e-16 ***
## SHO          -0.771140   0.290065  -2.659  0.00818 **
## HA           -0.186075   0.021421  -8.686  < 2e-16 ***
## BBA          -0.170383   0.019197  -8.875  < 2e-16 ***
## SOA           0.050316   0.007709   6.527 2.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.82 on 377 degrees of freedom
## Multiple R-squared:  0.7093, Adjusted R-squared:  0.7047
## F-statistic: 153.3 on 6 and 377 DF,  p-value: < 2.2e-16
```

getting rid of W

```
fit6 <- lm(HRA ~ ER + SHO + HA + BBA + SOA,
          data = train)
summary(fit6)
```

```
##
## Call:
## lm(formula = HRA ~ ER + SHO + HA + BBA + SOA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.893 -12.171   0.116  11.664  54.090
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 129.016503  32.129297   4.016 7.16e-05 ***
## ER            0.508302   0.025103  20.249  < 2e-16 ***
## SHO          -0.705283   0.289186  -2.439   0.0152 *
## HA           -0.185783   0.021501  -8.641  < 2e-16 ***
## BBA          -0.174209   0.019169  -9.088  < 2e-16 ***
## SOA           0.052541   0.007653   6.866 2.73e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.88 on 378 degrees of freedom
## Multiple R-squared:  0.7063, Adjusted R-squared:  0.7025
## F-statistic: 181.8 on 5 and 378 DF,  p-value: < 2.2e-16
```

Since fit6 has only independent variables with a p-value of 0.05 or lower it is the final fit.

Interpretation The point estimate tells us that when all of the independent = 0 the expected result would be 126 +- 32 (Std. Error).

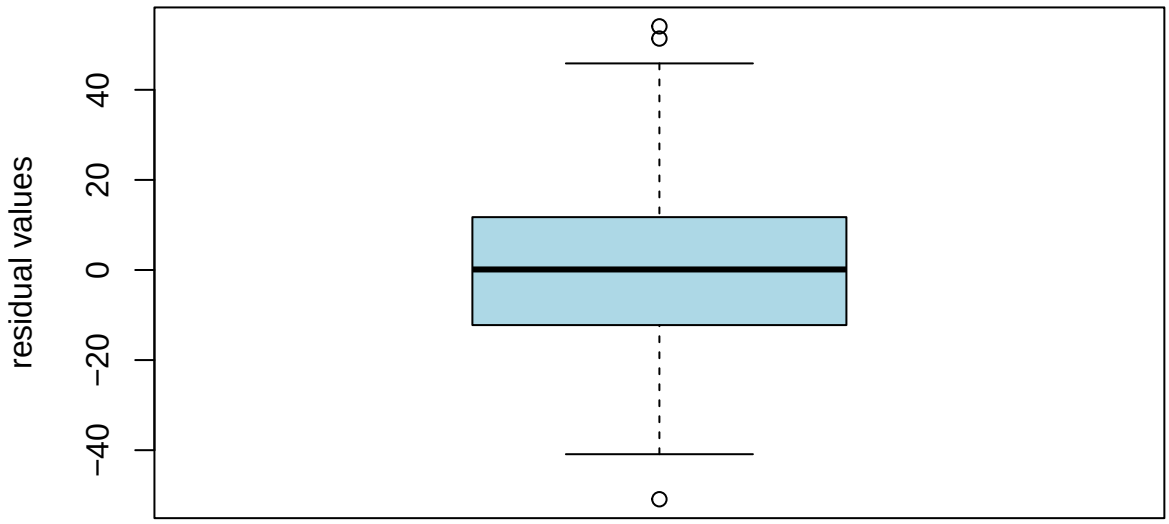
The t-value is 4 and this is how many standard errors our coefficient is from 0. The t-value of 4 tells us that the independent variables are strong predictors of HRA.

The Residual Standard error is 16.88 and this is the standard deviation of the residuals and tells us that our data points are going to be more spread out around the regression line.

The R-squared value tells us how well independent variables fit our dependent variables the closer to 1 the better and the closer to 0 is bad. The value of 0.7025 is good enough and tells us that about 70% of our outputs can be explained and 30% can't be.

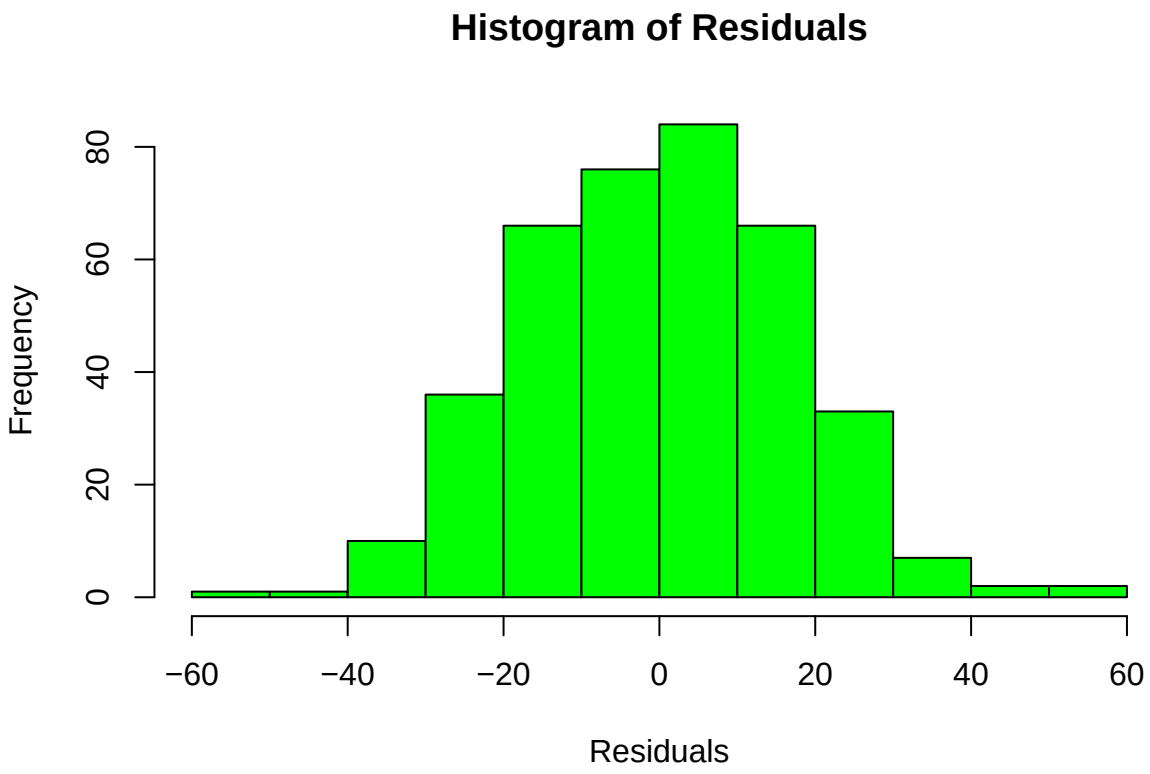
**residuals graph** residuals tell us how far away our predictions are from the actual value. The more normally distributed the residuals are the better our model predicts because the closer to the 0 the more accurate our predictions will be.

```
res <- resid(fit6)
boxplot(res, col = "lightblue", ylab = "residual values")
```




```
plot(fitted(fit6), res, col = "deeppink", xlab = "Home Runs Agianst (HRA)", ylab = "residuals")
abline(0,0)
```


```
hist(res, col = "green", xlab = "Residuals", main = "Histogram of Residuals")
```




The box plot shows us that our model is a good fit for our data because the median seems to be about 0 and Q1 and Q3 look to be the same length our residuals seem to be normally distributed. This is further backed up when looking at the histogram because our data looks normally distributed.

## Data Visualization

Applying Predictive model with the test set

```
predictions <- predict(fit6, test)
head(predictions)
```

```
##        1        2        3        4        5        6
## 187.4136 138.9486 201.7415 147.7924 185.9703 146.9373
```

```
predictDF <- data.frame(test, predictions)
RevisedTeams2DF <- select(predictDF, "yearID", "franchID", "HRA", "predictions")
RevisedTeams2DF$roundedPredictions <- round(RevisedTeams2DF$predictions, 0)
RevisedTeams2DF$TF <- RevisedTeams2DF$HRA == RevisedTeams2DF$roundedPredictions
head(RevisedTeams2DF)
```

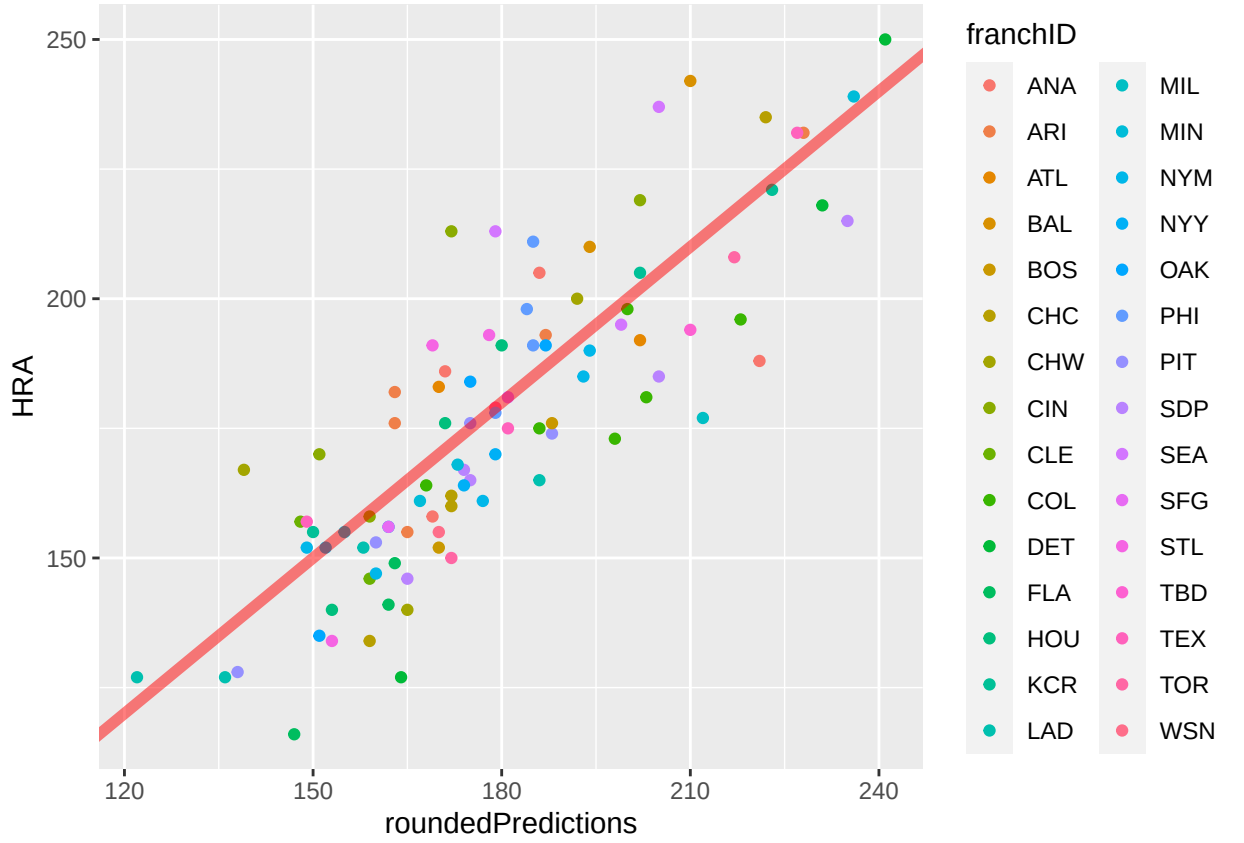
```
##   yearID franchID HRA predictions roundedPredictions    TF
## 1   2005      ARI 193    187.4136                187 FALSE
## 2   2005      CHW 167    138.9486                139 FALSE
## 3   2005      CIN 219    201.7415                202 FALSE
```

```
## 4    2005      CLE 157    147.7924                148 FALSE
## 5    2005      COL 175    185.9703                186 FALSE
## 6    2005      FLA 116    146.9373                147 FALSE
```

I rounded the predictions to the nearest whole number because you can't have fractions of home runs given up.

## Graph of Predictions vs Actual Values Using Test Set

```
g <- ggplot(data = RevisedTeams2DF, aes(x = roundedPredictions, y = HRA, col = franchID)) +
  geom_point() + geom_abline(intercept = 0, slope = 1, color = "red",alpha = 0.5, linewidth = 2)
g
```




our Predictions and the real HRA values appear to have a linear relationship.

## Results

```
results <- RevisedTeams2DF %>%
  group_by(TF) %>%
  summarize(count = n())
results
```

```
## # A tibble: 2 x 2
##   TF    count
##   <lgl> <int>
## 1 FALSE    92
## 2 TRUE      4
```

The model accurately predicted 4/96 of the number of home runs against a team had which is only a little over 4% accurate which isn't really that great but good enough to use on 2022 stats to have an idea of how many home runs a team will give up in 2022.

## Predicting 2022 HRA using 2022 stats from Baseball Reference

importing the csv file and cleaning the data set

```
# making 2022 dataset manually from Baseball Reference website
# importing csv file
Teams2022 <- read.csv("2022 Team Pitching stats MLB.csv", header = T)
# cleaning the data set
Teams_2022 <- Teams2022[-(31:32),]
franchID <- c("ARI", "ATL", "BAL", "BOS", "CHC", "CHW", "CIN", "CLE", "COL", "DET", "HOU", "KCR","ANA",
Teams__2022 <- data.frame(franchID, Teams_2022)
Teams___2022 <- select(Teams__2022, franchID,HR, ER, tSho, H, BB, SO)
Teams2022Rename <- Teams___2022 %>%
  rename(HRA = HR, ER = ER, SHO = tSho, HA = H, BBA = BB, SOA = SO)
head(Teams2022Rename)
```

```
##   franchID HRA  ER SHO   HA BBA  SOA
## 1      ARI 191 676  10 1345 504 1216
## 2      ATL 148 556   9 1224 500 1554
## 3      BAL 171 632  15 1406 443 1214
## 4      BOS 185 721  10 1411 526 1346
## 5      CHC 207 642  11 1342 540 1383
## 6      CHW 166 631  14 1330 533 1450
```

**Predicting** predicting 2022 home run against for each team

```
predictions2 <- predict(fit6, Teams2022Rename)
head(predictions2)
```

```
##        1        2        3        4        5        6
## 191.7856 172.4302 165.0828 205.3953 176.8583 176.1202
```

```
predictDF2022 <- data.frame(Teams2022Rename, predictions2)
RevisedTeams2DF2022 <- select(predictDF2022, "franchID", "HRA", "predictions2")
RevisedTeams2DF2022$roundedPredictions2 <- round(RevisedTeams2DF2022$predictions2, 0)
RevisedTeams2DF2022$TF <- RevisedTeams2DF2022$HRA == RevisedTeams2DF2022$roundedPredictions2
print(RevisedTeams2DF2022)
```

```
##    franchID HRA predictions2 roundedPredictions2    TF
## 1       ARI 191     191.7856                 192 FALSE
```

```
## 2       ATL 148     172.4302                172 FALSE
## 3       BAL 171     165.0828                165 FALSE
## 4       BOS 185     205.3953                205 FALSE
## 5       CHC 207     176.8583                177 FALSE
## 6       CHW 166     176.1202                176 FALSE
## 7       CIN 213     229.0576                229 FALSE
## 8       CLE 172     172.6736                173 FALSE
## 9       COL 184     219.2628                219 FALSE
## 10      DET 167     172.7216                173 FALSE
## 11      HOU 134     144.7034                145 FALSE
## 12      KCR 173     181.4050                181 FALSE
## 13      ANA 168     170.5503                171 FALSE
## 14      LAD 152     146.7883                147 FALSE
## 15      FLA 173     178.5046                179 FALSE
## 16      MIL 190     193.4891                193 FALSE
## 17      MIN 184     183.7376                184  TRUE
## 18      NYM 169     176.3255                176 FALSE
## 19      NYY 157     168.7901                169 FALSE
## 20      OAK 195     205.1297                205 FALSE
## 21      PHI 150     185.6827                186 FALSE
## 22      PIT 164     195.9347                196 FALSE
## 23      SDP 173     189.0725                189 FALSE
## 24      SEA 186     173.2216                173 FALSE
## 25      SFG 132     170.5790                171 FALSE
## 26      STL 146     153.1811                153 FALSE
## 27      TBD 172     170.2134                170 FALSE
## 28      TEX 169     181.9956                182 FALSE
## 29      TOR 180     184.3560                184 FALSE
## 30      WSN 244     219.1880                219 FALSE
```

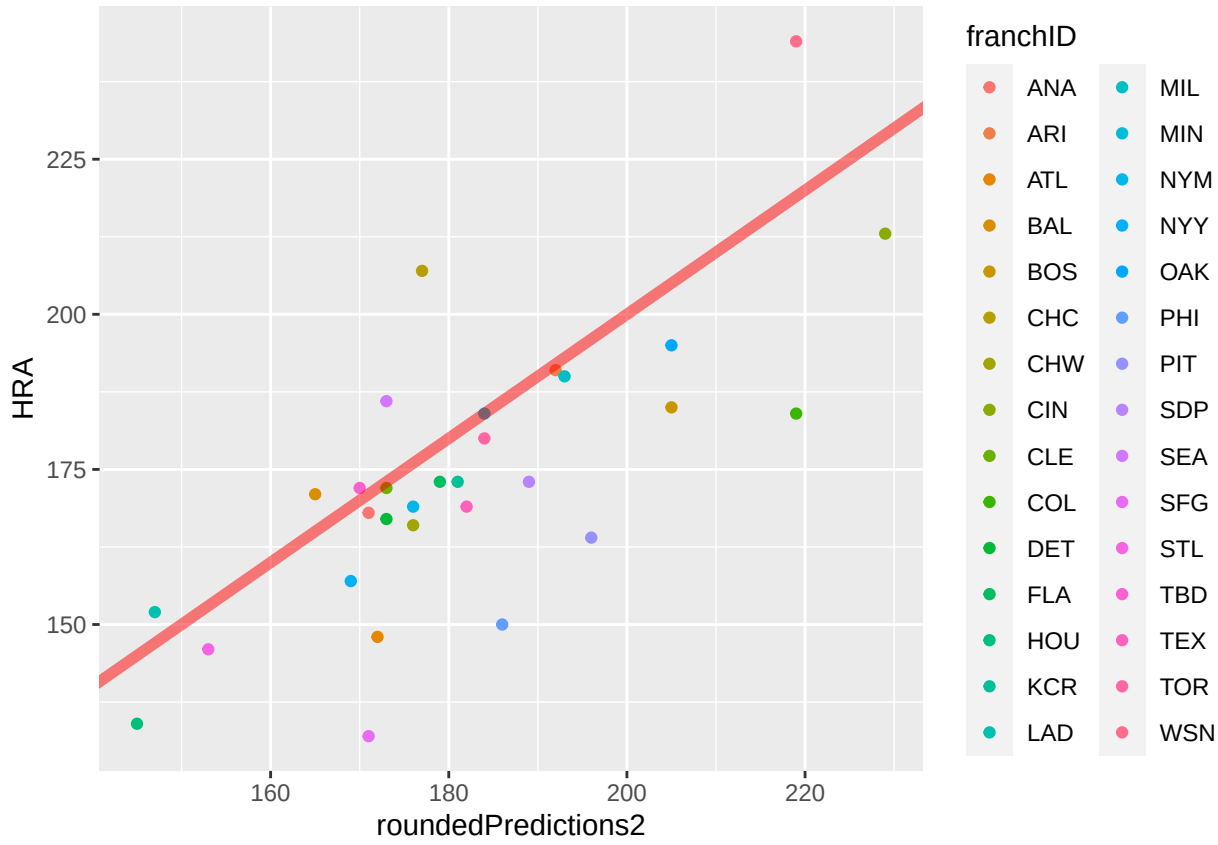
**Results for 2022**

```
results2022 <- RevisedTeams2DF2022 %>%
  group_by(TF) %>%
  summarize(count = n())
results2022
```

```
## # A tibble: 2 x 2
##   TF    count
##   <lgl> <int>
## 1 FALSE    29
## 2 TRUE      1
```

**predictions vs actual values for 2022**

```
g <- ggplot(data = RevisedTeams2DF2022, aes(x = roundedPredictions2, y = HRA, col = franchID)) +
  geom_point() + geom_abline(intercept = 0, slope = 1, color = "red",alpha = 0.5, linewidth = 2)
g
```




```
df <- RevisedTeams2DF2022[c(1,3,6,7,8,10,11,12,13,14,15,16,17,18,19,20,23,24,26,27,28,29),]
df
```

```
##    franchID HRA predictions2 roundedPredictions2    TF
## 1       ARI 191     191.7856                 192 FALSE
## 3       BAL 171     165.0828                 165 FALSE
## 6       CHW 166     176.1202                 176 FALSE
## 7       CIN 213     229.0576                 229 FALSE
## 8       CLE 172     172.6736                 173 FALSE
## 10      DET 167     172.7216                 173 FALSE
## 11      HOU 134     144.7034                 145 FALSE
## 12      KCR 173     181.4050                 181 FALSE
## 13      ANA 168     170.5503                 171 FALSE
## 14      LAD 152     146.7883                 147 FALSE
## 15      FLA 173     178.5046                 179 FALSE
## 16      MIL 190     193.4891                 193 FALSE
## 17      MIN 184     183.7376                 184  TRUE
## 18      NYM 169     176.3255                 176 FALSE
## 19      NYY 157     168.7901                 169 FALSE
## 20      OAK 195     205.1297                 205 FALSE
## 23      SDP 173     189.0725                 189 FALSE
## 24      SEA 186     173.2216                 173 FALSE
## 26      STL 146     153.1811                 153 FALSE
## 27      TBD 172     170.2134                 170 FALSE
## 28      TEX 169     181.9956                 182 FALSE
## 29      TOR 180     184.3560                 184 FALSE
```

The table shows all of the predictions that were within 20 home runs against as the real value which was 22 out of the 30 teams. I think that this is pretty good and shows that the regression model does a decent job of giving us an idea of how many home runs a team will give up in a given year.

## Conclusion

In conclusion I was able to answer my question from the beginning of seeing if I could accurately predict how many home runs each team will give up in 2022. My predictions weren't as accurate as I would've liked but, I know what to do next time in order to get a more accurate predictive model.I think that to make this model better I would've fed it more information from the beginning instead of limiting the years to only 2005-2021. I think that if I would've given a year range from 1985 - 2021 my model would've been better because it would've had more information to go off of. Overall, I think that my predictions for home runs against in 2022 for each team give a solid idea of the real value is.

## References

https://www.baseball-reference.com/leagues/majors/2022.shtml#all_teams_standard_pitching