Hall of Fame Expectancy

Daniel Antantis

Read in raw player data

```
hit_data = read.csv("~/Documents/MLB Project/allhitters.csv")
head(hit_data)
```

```
##   X        name                        url Year Age     Tm Lg   G  PA  AB   R
## 1 0 Henry Aaron /players/a/aaronha01.shtml 1952  18 BSN-min  C  87 345 345  NA
## 2 1 Henry Aaron /players/a/aaronha01.shtml 1953  19 MLN-min  A 137 574 574  NA
## 3 2 Henry Aaron /players/a/aaronha01.shtml 1954  20     MLN NL 122 509 468  58
## 4 3 Henry Aaron /players/a/aaronha01.shtml 1955  21     MLN NL 153 665 602 105
## 5 4 Henry Aaron /players/a/aaronha01.shtml 1956  22     MLN NL 153 660 609 106
## 6 5 Henry Aaron /players/a/aaronha01.shtml 1957  23     MLN NL 151 675 615 118
##     H X2B X3B HR RBI SB CS BB SO    BA   OBP   SLG   OPS OPS.  TB GDP HBP SH SF
## 1 116  19   4  9  NA NA NA NA NA 0.336 0.336 0.493 0.829   NA 170  NA  NA NA NA
## 2 208  36  14 22  NA NA NA NA NA 0.362 0.362 0.589 0.951   NA 338  NA  NA NA NA
## 3 131  27   6 13  69  2  2 28 39 0.280 0.322 0.447 0.769  104 209  13   3  6  4
## 4 189  37   9 27 106  3  1 49 61 0.314 0.366 0.540 0.906  141 325  20   3  7  4
## 5 200  34  14 26  92  2  4 37 54 0.328 0.365 0.558 0.923  151 340  21   2  5  7
## 6 198  27   6 44 132  1  1 57 58 0.322 0.378 0.600 0.978  166 369  13   0  0  3
##   IBB    Pos     Awards
## 1  NA        EAU · NORL
## 2  NA        JCK · SALL
## 3   0  *79/H      RoY-4
## 4   5 *974/H   AS,MVP-9
## 5   6   *9/H   AS,MVP-3
## 6  15  *98/H   AS,MVP-1
```

Read in Hall of Fame data and filter for only hitters by at-bats (AB)

```
url = "https://www.baseball-reference.com/awards/hof_batting.shtml"
tbl = url %>%
  read_html() %>%
  html_nodes('table') %>%
  html_table()
hof = data.frame(tbl)
hof = hof %>% filter(AB > 3000)
```

Filter for players eligible for HOF

```
hof_eligible = hit_data %>%
  na.omit() %>%
  group_by(name) %>%
  filter(AB > 200) %>%
```

```
  summarise(
    Retirement_Year = max(Year),
    AB = sum(AB),
    HR = sum(HR),
    RBI = sum(RBI),
    H = sum(H),
    BA = mean(BA),
    OBP = mean(OBP),
    SB = sum(SB),
    BB = sum(BB),
    IBB = sum(IBB),
    SLG = mean(SLG),
    OPS = mean(OPS),
    TB = sum(TB),
    MVP = sum(ifelse(grepl("MVP-1,", Awards, fixed = TRUE), 1, 0)),
    All_star = sum(ifelse(grepl("AS", Awards, fixed = TRUE), 1, 0)),
    Gold_glove = sum(ifelse(grepl("GG", Awards, fixed = TRUE), 1, 0)),
    Silver_slugger = sum(ifelse(grepl("SS", Awards, fixed = TRUE), 1, 0))
  ) %>%
  filter(Retirement_Year < 2016)
```

Merge dataframes

```
hof_both = data.frame(hof_eligible$name[hof_eligible$name %in% hof$Name])
hof_both = hof_both %>%
  summarise(name = hof_eligible.name.hof_eligible.name..in..hof.Name.)
hof_both$hof = 1
```

Merge players with their stats (excluding players banned from HOF)

```
raw_data = hof_eligible %>%
  left_join(hof_both, by = "name")
raw_data[is.na(raw_data)] = 0
raw_data$ISO = raw_data$SLG - raw_data$BA
model_data = hof_eligible %>%
  left_join(hof_both, by = "name") %>%
  filter(!(name == "Barry Bonds" | name == "Pete Rose" | name == "Mark McGwire"))
model_data[is.na(model_data)] = 0
model_data$RC = ((model_data$H + model_data$BB) * model_data$TB)/(model_data$AB + model_data$BB)
model_data$ISO = model_data$SLG - model_data$BA
```

Split into training and testing sets

```
attach(model_data)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     hof
```

```
sample_size <- floor(0.8 * nrow(model_data))
set.seed(111)
```

```
train <- sample(seq_len(nrow(model_data)), size = sample_size)
model_train = model_data[train,]
model_test = model_data[-train,]
```

Linear Model

```
LDA.fit = lda(hof ~ HR + AB + BA + OPS + BB + MVP + All_star + Gold_glove + Silver_slugger, data = model
LDA.pred = predict(LDA.fit, model_test)
mean(LDA.pred$class != model_test$hof)
```

```
## [1] 0.01976285
```

```
GLM.fit = glm(hof ~ HR + AB + BA + OPS + BB + ISO + MVP + All_star + Gold_glove + Silver_slugger, data =
GLM.probs = predict(GLM.fit, model_test, type = "response")
GLM.pred = rep(0, length(GLM.probs))
GLM.pred[GLM.probs > 0.5] = 1
mean(GLM.pred != model_test$hof)
```

```
## [1] 0.02371542
```

```
QDA.fit = qda(hof ~ HR + AB + BA + OPS + BB + MVP + All_star + Gold_glove + Silver_slugger, data = model
QDA.pred = predict(QDA.fit, model_test)
mean(QDA.pred$class != model_test$hof)
```

```
## [1] 0.06126482
```

```
train.x = model.matrix(hof ~ HR + AB + BA + OPS + BB + MVP + All_star + Gold_glove + Silver_slugger, da
train.hof = model_train$hof
test.x = model.matrix(hof ~ HR + AB + BA + OPS + BB + MVP + All_star + Gold_glove + Silver_slugger, data
test.hof = model_test$hof

ridge.fit = cv.glmnet(train.x, train.hof, alpha = 0)
ridge.lambda = ridge.fit$lambda.min

ridge.probs = predict(ridge.fit, s = ridge.lambda, newx = test.x)
ridge.pred = rep(0, length(ridge.probs))
ridge.pred[ridge.probs > 0.4] = 1
mean(ridge.pred != model_test$hof)
```

```
## [1] 0.03162055
```

```
ridge.coef = predict(ridge.fit, type = "coefficients", s = ridge.lambda)
ridge.coef
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)    -1.481518e-02
## (Intercept)      .
## HR             -5.883310e-05
```

```
## AB              -9.889358e-06
## BA               2.791406e-02
## OPS              1.357075e-02
## BB               4.994942e-05
## MVP              1.127846e-01
## All_star         4.500345e-02
## Gold_glove       3.096033e-03
## Silver_slugger   1.795516e-02
```

Which players were classified incorrectly

```
incorrect_name = ifelse(GLM.pred != model_test$hof, model_test$name, NA)
incorrect_pred = ifelse(GLM.pred != model_test$hof, ridge.pred, NA)
incorrect = data.frame(incorrect_name, incorrect_pred)
incorrect = incorrect %>% na.omit()
```

Test model generalization

```
total.probs = predict(GLM.fit, model_data, type = "response")
total.pred = rep(0, length(total.probs))
total.pred[total.probs > 0.5] = 1
name = ifelse(total.pred != model_data$hof, model_data$name, NA)
pred = ifelse(total.pred != model_data$hof, total.pred, NA)
prob = ifelse(total.pred != model_data$hof, round(total.probs,3), NA)
end1 = data.frame(name, pred, prob)
end1 = end1 %>% na.omit()
end1_data = end1 %>% left_join(model_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0))
head(end1_data)
```

```
##               name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1  Alan Trammell    0 0.107            1995  7950 183  976 2284 0.2848235
## 2   Bill Freehan    1 0.836            1976  6063 200  754 1587 0.2603571
## 3 Bill Mazeroski    0 0.134            1970  7498 137  834 1955 0.2578000
## 4 Billy Williams    0 0.238            1976  9270 424 1466 2693 0.2876875
## 5   Craig Biggio    0 0.275            2007 10753 288 1170 3034 0.2814211
## 6    Dale Murphy    1 0.627            1991  8278 418 1312 2192 0.2632500
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3508824 228  821  48 0.4141765 0.7651176 3344   0        6          4
## 2 0.3364286  24  625  67 0.4075714 0.7438571 2498   0       11          5
## 3 0.2992000  27  429 108 0.3638000 0.6630000 2775   0        7          8
## 4 0.3612500  90 1039 182 0.4870625 0.8481875 4569   0        6          0
## 5 0.3630000 408 1153  66 0.4322632 0.7950526 4668   0        7          4
## 6 0.3424375 170 1034 172 0.4625625 0.8051250 3881   2        7          5
##   Silver_slugger hof        RC       ISO
## 1              3   1 1183.8012 0.1293529
## 2              0   0  826.1926 0.1472143
## 3              0   1  834.5654 0.1060000
## 4              0   1 1654.0409 0.1993750
## 5              5   1 1641.6022 0.1508421
## 6              4   0 1344.5131 0.1993125
```

```
nrow(end1)
```

```
## [1] 44
```

```
name = ifelse(total.pred == 1 & model_data$hof == 1, model_data$name, NA)
pred = ifelse(total.pred == 1 & model_data$hof == 1, total.pred, NA)
prob = ifelse(total.pred == 1 & model_data$hof == 1, round(total.probs,3), NA)
end2 = data.frame(name, pred, prob)
end2 = end2 %>% na.omit()
end2_data = end2 %>% left_join(model_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0))
head(end2_data)
```

```
##                 name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1          Al Kaline    1 0.989            1974 10088 398 1580 3000 0.2959048
## 2       Andre Dawson    1 0.665            1995  9784 436 1570 2738 0.2777895
## 3       Barry Larkin    1 0.922            2004  7622 193  924 2255 0.2969412
## 4    Brooks Robinson    1 0.993            1976 10424 264 1337 2801 0.2658421
## 5     Cal Ripken Jr.    1 1.000            2001 11512 431 1695 3179 0.2767500
## 6 Carl Yastrzemski    1 0.999            1983 11988 452 1844 3419 0.2838696
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3747619 136 1276 133 0.4775714 0.8523333 4842   0       15         10
## 2 0.3218947 313  582 142 0.4815263 0.8033684 4737   1        8          8
## 3 0.3721176 368  903  63 0.4457059 0.8177647 3405   1       12          3
## 4 0.3196842  27  848 120 0.3954737 0.7152632 4197   1       15         16
## 5 0.3399000  36 1128 107 0.4501500 0.7900000 5163   2       19          2
## 6 0.3772609 168 1845 190 0.4580870 0.8355652 5539   1       18          7
##   Silver_slugger hof       RC       ISO
## 1              0   1 1821.928 0.1816667
## 2              4   1 1517.156 0.2037368
## 3              9   1 1261.348 0.1487647
## 4              0   1 1358.663 0.1296316
## 5              8   1 1759.260 0.1734000
## 6              0   1 2107.807 0.1742174
```

```
nrow(end2)
```

```
## [1] 44
```

Lower Threshold

```
total.probs = predict(GLM.fit, model_data, type = "response")
total.pred = rep(0, length(total.probs))
total.pred[total.probs > 0.4] = 1
name = ifelse(total.pred != model_data$hof, model_data$name, NA)
pred = ifelse(total.pred != model_data$hof, total.pred, NA)
prob = ifelse(total.pred != model_data$hof, round(total.probs,3), NA)
end1 = data.frame(name, pred, prob)
end1 = end1 %>% na.omit()
end1_data = end1 %>% left_join(model_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0))
head(end1_data)
```

```
##              name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1  Alan Trammell    0 0.107            1995  7950 183  976 2284 0.2848235
## 2   Bill Freehan    1 0.836            1976  6063 200  754 1587 0.2603571
## 3 Bill Mazeroski    0 0.134            1970  7498 137  834 1955 0.2578000
## 4 Billy Williams    0 0.238            1976  9270 424 1466 2693 0.2876875
## 5   Craig Biggio    0 0.275            2007 10753 288 1170 3034 0.2814211
## 6    Dale Murphy    1 0.627            1991  8278 418 1312 2192 0.2632500
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3508824 228  821  48 0.4141765 0.7651176 3344   0        6          4
## 2 0.3364286  24  625  67 0.4075714 0.7438571 2498   0       11          5
## 3 0.2992000  27  429 108 0.3638000 0.6630000 2775   0        7          8
## 4 0.3612500  90 1039 182 0.4870625 0.8481875 4569   0        6          0
## 5 0.3630000 408 1153  66 0.4322632 0.7950526 4668   0        7          4
## 6 0.3424375 170 1034 172 0.4625625 0.8051250 3881   2        7          5
##   Silver_slugger hof        RC       ISO
## 1              3   1 1183.8012 0.1293529
## 2              0   0  826.1926 0.1472143
## 3              0   1  834.5654 0.1060000
## 4              0   1 1654.0409 0.1993750
## 5              5   1 1641.6022 0.1508421
## 6              4   0 1344.5131 0.1993125
```

```r
nrow(end1)
```

```
## [1] 41
```

```r
name = ifelse(total.pred == 1 & model_data$hof == 1, model_data$name, NA)
pred = ifelse(total.pred == 1 & model_data$hof == 1, total.pred, NA)
prob = ifelse(total.pred == 1 & model_data$hof == 1, round(total.probs,3), NA)
end2 = data.frame(name, pred, prob)
end2 = end2 %>% na.omit()
end2_data = end2 %>% left_join(model_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0))
head(end2_data)
```

```
##                name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1         Al Kaline    1 0.989            1974 10088 398 1580 3000 0.2959048
## 2      Andre Dawson    1 0.665            1995  9784 436 1570 2738 0.2777895
## 3      Barry Larkin    1 0.922            2004  7622 193  924 2255 0.2969412
## 4  Brooks Robinson    1 0.993            1976 10424 264 1337 2801 0.2658421
## 5    Cal Ripken Jr.    1 1.000            2001 11512 431 1695 3179 0.2767500
## 6 Carl Yastrzemski    1 0.999            1983 11988 452 1844 3419 0.2838696
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3747619 136 1276 133 0.4775714 0.8523333 4842   0       15         10
## 2 0.3218947 313  582 142 0.4815263 0.8033684 4737   1        8          8
## 3 0.3721176 368  903  63 0.4457059 0.8177647 3405   1       12          3
## 4 0.3196842  27  848 120 0.3954737 0.7152632 4197   1       15         16
## 5 0.3399000  36 1128 107 0.4501500 0.7900000 5163   2       19          2
## 6 0.3772609 168 1845 190 0.4580870 0.8355652 5539   1       18          7
##   Silver_slugger hof       RC       ISO
## 1              0   1 1821.928 0.1816667
## 2              4   1 1517.156 0.2037368
## 3              9   1 1261.348 0.1487647
## 4              0   1 1358.663 0.1296316
```

```
## 5              8   1 1759.260 0.1734000
## 6              0   1 2107.807 0.1742174
```

```
nrow(end2)
```

```
## [1] 49
```

GLM Deep Dive

```
GLM.fit$coefficients
```

```
##    (Intercept)             HR             AB             BA            OPS
##  -1.614310e+01  -3.296638e-04   4.107136e-04  -9.011997e+01   5.560707e+01
##             BB            ISO            MVP       All_star     Gold_glove
##  -3.677484e-03  -5.026194e+01   1.106218e+00   6.525447e-01  -7.060989e-03
## Silver_slugger
##  -1.430774e-01
```

```
summary(GLM.fit)
```

```
##
## Call:
## glm(formula = hof ~ HR + AB + BA + OPS + BB + ISO + MVP + All_star +
##     Gold_glove + Silver_slugger, family = binomial, data = model_train,
##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1390  -0.1122  -0.0652  -0.0405   3.5985
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.614e+01  4.128e+00  -3.911 9.20e-05 ***
## HR             -3.297e-04  4.182e-03  -0.079  0.93717
## AB              4.107e-04  2.513e-04   1.635  0.10213
## BA             -9.012e+01  3.799e+01  -2.372  0.01767 *
## OPS             5.561e+01  1.973e+01   2.819  0.00482 **
## BB             -3.678e-03  1.638e-03  -2.244  0.02480 *
## ISO            -5.026e+01  2.382e+01  -2.110  0.03486 *
## MVP             1.106e+00  5.849e-01   1.891  0.05858 .
## All_star        6.525e-01  1.009e-01   6.465 1.01e-10 ***
## Gold_glove     -7.061e-03  9.888e-02  -0.071  0.94307
## Silver_slugger -1.431e-01  1.365e-01  -1.048  0.29452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 466.22  on 1611  degrees of freedom
## Residual deviance: 181.48  on 1601  degrees of freedom
##   (408 observations deleted due to missingness)
## AIC: 203.48
##
## Number of Fisher Scoring iterations: 9
```

Would the model incorrectly predict players who would be in HOF if not banned

```
banned  = raw_data %>%
  filter(name == "Barry Bonds" | name == "Mark McGwire" | name == "Pete Rose")
head(banned)
```

```
## # A tibble: 3 x 20
##   name        Retir~1    AB    HR   RBI     H    BA   OBP    SB    BB   IBB   SLG
##   <chr>         <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Barry Bon~    2007  9805   757  1986  2923 0.299 0.443   514  2549   685 0.613
## 2 Mark McGw~    2001  6281   596  1437  1658 0.264 0.394    13  1313   150 0.600
## 3 Pete Rose     1986 14331   160  1337  4328 0.297 0.371   199  1597   170 0.397
## # ... with 8 more variables: OPS <dbl>, TB <dbl>, MVP <dbl>, All_star <dbl>,
## #   Gold_glove <dbl>, Silver_slugger <dbl>, hof <dbl>, ISO <dbl>, and
## #   abbreviated variable name 1: Retirement_Year
```

```
GLM.probs = predict(GLM.fit, banned, type = "response")
GLM.pred = rep(0, length(GLM.probs))
GLM.pred[GLM.probs > 0.5] = 1
incorrect_ban_name = ifelse(GLM.pred != banned$hof, banned$name, NA)
incorrect_ban_pred = ifelse(GLM.pred != banned$hof, GLM.pred, NA)
incorrect_ban_prob = ifelse(GLM.pred != banned$hof, GLM.probs, NA)
incorrect_ban = data.frame(incorrect_ban_name, incorrect_ban_pred, incorrect_ban_prob)
incorrect_ban = incorrect_ban %>% na.omit()
head(incorrect_ban)
```

```
##   incorrect_ban_name incorrect_ban_pred incorrect_ban_prob
## 1        Barry Bonds                  1          0.9999156
## 2       Mark McGwire                  1          0.9684155
## 3          Pete Rose                  1          0.9964391
```

Would the model be better if only more modern players were considered?

```
modern = model_data %>%
  filter(Retirement_Year > 1960)
```

```
sample_size <- floor(0.8 * nrow(modern))
set.seed(111)
mod_train <- sample(seq_len(nrow(modern)), size = sample_size)
modern_train = modern[mod_train,]
modern_test = modern[-mod_train,]
```

```
GLM.modern.fit = glm(hof ~ HR + AB + BA + OPS + BB + ISO + MVP + All_star + Gold_glove + Silver_slugger
GLM.modern.probs = predict(GLM.modern.fit, model_test, type = "response")
GLM.modern.pred = rep(0, length(GLM.modern.probs))
GLM.modern.pred[GLM.modern.probs > 0.5] = 1
mean(GLM.modern.pred != modern_test$hof)
```

```
## Warning in GLM.modern.pred != modern_test$hof: longer object length is not a
## multiple of shorter object length
```

```
## [1] 0.0513834
```

```
modern.probs = predict(GLM.modern.fit, modern, type = "response")
modern.pred = rep(0, length(modern.probs))
modern.pred[modern.probs > 0.4] = 1
name = ifelse(modern.pred != modern$hof, modern$name, NA)
pred = ifelse(modern.pred != modern$hof, modern.pred, NA)
prob = ifelse(modern.pred != modern$hof, round(modern.probs,3), NA)
modern.end1 = data.frame(name, pred, prob)
modern.end1 = modern.end1 %>% na.omit()
modern.end1_data = modern.end1 %>% left_join(modern, by = "name") %>% mutate_all(~replace(., is.na(.), 
head(modern.end1_data)
```

```
##                name pred  prob Retirement_Year   AB  HR  RBI    H        BA
## 1         Al Oliver    1 0.491            1985 9380 219 1360 2843 0.3016667
## 2      Alan Trammell    0 0.096            1995 7950 183  976 2284 0.2848235
## 3       Bill Freehan    1 0.598            1976 6063 200  754 1587 0.2603571
## 4     Bill Mazeroski    0 0.045            1970 7498 137  834 1955 0.2578000
## 5     Billy Williams    0 0.321            1976 9270 424 1466 2693 0.2876875
## 6 Darryl Strawberry    1 0.573            1998 4905 315  925 1285 0.2611818
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3424444  86  555 123 0.4421667 0.7846111 4206   0        7          0
## 2 0.3508824 228  821  48 0.4141765 0.7651176 3344   0        6          4
## 3 0.3364286  24  625  67 0.4075714 0.7438571 2498   0       11          5
## 4 0.2992000  27  429 108 0.3638000 0.6630000 2775   0        7          8
## 5 0.3612500  90 1039 182 0.4870625 0.8481875 4569   0        6          0
## 6 0.3579091 215  732 121 0.5161818 0.8742727 2535   0        8          0
##   Silver_slugger hof        RC       ISO
## 1              3   0 1438.5494 0.1405000
## 2              3   1 1183.8012 0.1293529
## 3              0   0  826.1926 0.1472143
## 4              0   1  834.5654 0.1060000
## 5              0   1 1654.0409 0.1993750
## 6              2   0  907.0596 0.2550000
```

```
nrow(modern.end1)
```

```
## [1] 31
```

```
name = ifelse(modern.pred == 1 & modern$hof == 1, modern$name, NA)
pred = ifelse(modern.pred == 1 & modern$hof == 1, modern.pred, NA)
prob = ifelse(modern.pred == 1 & modern$hof == 1, round(modern.probs,3), NA)
modern.end2 = data.frame(name, pred, prob)
modern.end2 = modern.end2 %>% na.omit()
modern.end2_data = modern.end2 %>% left_join(modern, by = "name") %>% mutate_all(~replace(., is.na(.), 
head(modern.end2_data)
```

```
##               name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1       Al Kaline    1 0.998            1974 10088 398 1580 3000 0.2959048
## 2     Andre Dawson    1 0.676            1995  9784 436 1570 2738 0.2777895
## 3     Barry Larkin    1 0.896            2004  7622 193  924 2255 0.2969412
## 4  Brooks Robinson    1 0.997            1976 10424 264 1337 2801 0.2658421
## 5    Cal Ripken Jr.    1 1.000            2001 11512 431 1695 3179 0.2767500
## 6 Carl Yastrzemski    1 1.000            1983 11988 452 1844 3419 0.2838696
```

```
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3747619 136 1276 133 0.4775714 0.8523333 4842   0       15         10
## 2 0.3218947 313  582 142 0.4815263 0.8033684 4737   1        8          8
## 3 0.3721176 368  903  63 0.4457059 0.8177647 3405   1       12          3
## 4 0.3196842  27  848 120 0.3954737 0.7152632 4197   1       15         16
## 5 0.3399000  36 1128 107 0.4501500 0.7900000 5163   2       19          2
## 6 0.3772609 168 1845 190 0.4580870 0.8355652 5539   1       18          7
##   Silver_slugger hof       RC       ISO
## 1              0   1 1821.928 0.1816667
## 2              4   1 1517.156 0.2037368
## 3              9   1 1261.348 0.1487647
## 4              0   1 1358.663 0.1296316
## 5              8   1 1759.260 0.1734000
## 6              0   1 2107.807 0.1742174
```

```
nrow(modern.end2)
```

```
## [1] 51
```

```
GLM.modern.fit$coefficients
```

```
##    (Intercept)             HR             AB             BA            OPS
##  -3.221320e+01  -1.404267e-02   1.463007e-03  -1.441075e+02   9.002187e+01
##             BB            ISO            MVP       All_star     Gold_glove
##  -4.979748e-03  -4.939107e+01   5.814346e-01   6.636787e-01   3.743445e-02
## Silver_slugger
##  -1.905133e-01
```

```
summary(GLM.modern.fit)
```

```
##
## Call:
## glm(formula = hof ~ HR + AB + BA + OPS + BB + ISO + MVP + All_star +
##     Gold_glove + Silver_slugger, family = binomial, data = modern_train,
##     subset = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.33889  -0.03234  -0.00824  -0.00215   2.95131
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.221e+01  7.400e+00  -4.353 1.34e-05 ***
## HR             -1.404e-02  6.790e-03  -2.068 0.038614 *
## AB              1.463e-03  4.314e-04   3.391 0.000695 ***
## BA             -1.441e+02  7.512e+01  -1.918 0.055061 .
## OPS             9.002e+01  4.091e+01   2.200 0.027773 *
## BB             -4.980e-03  2.905e-03  -1.714 0.086519 .
## ISO            -4.939e+01  4.871e+01  -1.014 0.310582
## MVP             5.814e-01  5.837e-01   0.996 0.319188
## All_star        6.637e-01  1.213e-01   5.471 4.47e-08 ***
## Gold_glove      3.743e-02  1.063e-01   0.352 0.724658
```

```
## Silver_slugger -1.905e-01  1.426e-01  -1.336 0.181510
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.17  on 1490  degrees of freedom
## Residual deviance: 102.70  on 1480  degrees of freedom
##   (529 observations deleted due to missingness)
## AIC: 124.7
##
## Number of Fisher Scoring iterations: 10
```

Filter out players who have retired so that all that remains are players currently playing or not eligible for the HOF

```
current = hit_data %>%
  na.omit() %>%
  group_by(name) %>%
  filter(AB > 200) %>%
  summarise(
    Retirement_Year = max(Year),
    AB = sum(AB),
    HR = sum(HR),
    RBI = sum(RBI),
    H = sum(H),
    BA = mean(BA),
    OBP = mean(OBP),
    SB = sum(SB),
    BB = sum(BB),
    IBB = sum(IBB),
    SLG = mean(SLG),
    OPS = mean(OPS),
    TB = sum(TB),
    MVP = sum(ifelse(grepl("MVP-1,", Awards, fixed = TRUE), 1, 0)),
    All_star = sum(ifelse(grepl("AS", Awards, fixed = TRUE), 1, 0)),
    Gold_glove = sum(ifelse(grepl("GG", Awards, fixed = TRUE), 1, 0)),
    Silver_slugger = sum(ifelse(grepl("SS", Awards, fixed = TRUE), 1, 0))
  ) %>%
  filter(Retirement_Year > 2016)
hof_both = data.frame(current$name[current$name %in% hof$Name])
hof_both = hof_both %>%
  summarise(name = current.name.current.name..in..hof.Name.)
hof_both$hof = 1
raw_data = current %>%
  left_join(hof_both, by = "name")
raw_data[is.na(raw_data)] = 0
raw_data$ISO = raw_data$SLG - raw_data$BA
curr_data = current %>%
  left_join(hof_both, by = "name") %>%
  filter(!(name == "Barry Bonds" | name == "Pete Rose" | name == "Mark McGwire"))
curr_data[is.na(curr_data)] = 0
curr_data$RC = ((curr_data$H + curr_data$BB) * curr_data$TB)/(curr_data$AB + curr_data$BB)
curr_data$ISO = curr_data$SLG - curr_data$BA
```

Test on current players and players not yet inducted into HOF

```
curr.probs = predict(GLM.fit, curr_data, type = "response")
curr.pred = rep(0, length(curr.probs))
curr.pred[curr.probs > 0.4] = 1
name = ifelse(curr.pred == 1, curr_data$name, NA)
pred = ifelse(curr.pred == 1, curr.pred, NA)
prob = ifelse(curr.pred == 1, round(curr.probs,3), NA)
curr1 = data.frame(name, pred, prob)
curr1 = curr1 %>% na.omit()
curr1_data = curr1 %>% left_join(curr_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0))
head(curr1_data)
```

```
##              name pred  prob Retirement_Year    AB  HR  RBI    H        BA
## 1   Albert Pujols    1 0.949            2019 10687 656 2075 3202 0.2985789
## 2    Bryce Harper    1 0.653            2021  4127 234  669 1140 0.2773333
## 3    Buster Posey    1 0.532            2021  4597 148  680 1398 0.3041000
## 4  Carlos Beltran    1 0.576            2017 11021 510 1814 3080 0.2783913
## 5 Freddie Freeman    1 0.467            2021  5472 258  907 1610 0.2956364
## 6      Joey Votto    1 0.400            2021  6222 291  967 1895 0.3008462
##         OBP  SB   BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3754211 114 1322 311 0.5460000 0.9213158 5863   2       10          2
## 2 0.3820000  97  719  85 0.5103333 0.8924444 2113   1        6          0
## 3 0.3741000  20  497  63 0.4657000 0.8397000 2133   1        7          1
## 4 0.3526522 354 1255 123 0.4911739 0.8438261 5429   0        9          3
## 5 0.3873636  49  746  96 0.5159091 0.9032727 2785   1        5          1
## 6 0.4145385  79 1203 140 0.5102308 0.9249231 3217   0        6          1
##   Silver_slugger hof        RC       ISO
## 1              6   0 2208.6945 0.2474211
## 2              1   0  810.5792 0.2330000
## 3              4   0  793.4894 0.1616000
## 4              2   0 1917.1322 0.2127826
## 5              2   0 1055.2364 0.2202727
## 6              0   0 1342.2580 0.2093846
```

```
nrow(curr1)
```

```
## [1] 11
```

Players who are on the edge

```
curr2.pred = rep(0, length(curr.probs))
curr2.pred[curr.probs > 0.25] = 1
name = ifelse(curr2.pred == 1, curr_data$name, NA)
pred = ifelse(curr2.pred == 1, curr.pred, NA)
prob = ifelse(curr2.pred == 1, round(curr.probs,3), NA)
curr2 = data.frame(name, pred, prob)
curr2 = curr2 %>% na.omit()
curr2_data = curr2 %>% left_join(curr_data, by = "name") %>% mutate_all(~replace(., is.na(.), 0)) %>% fi
head(curr2_data)
```

```
##                name pred  prob Retirement_Year   AB  HR  RBI    H        BA
```

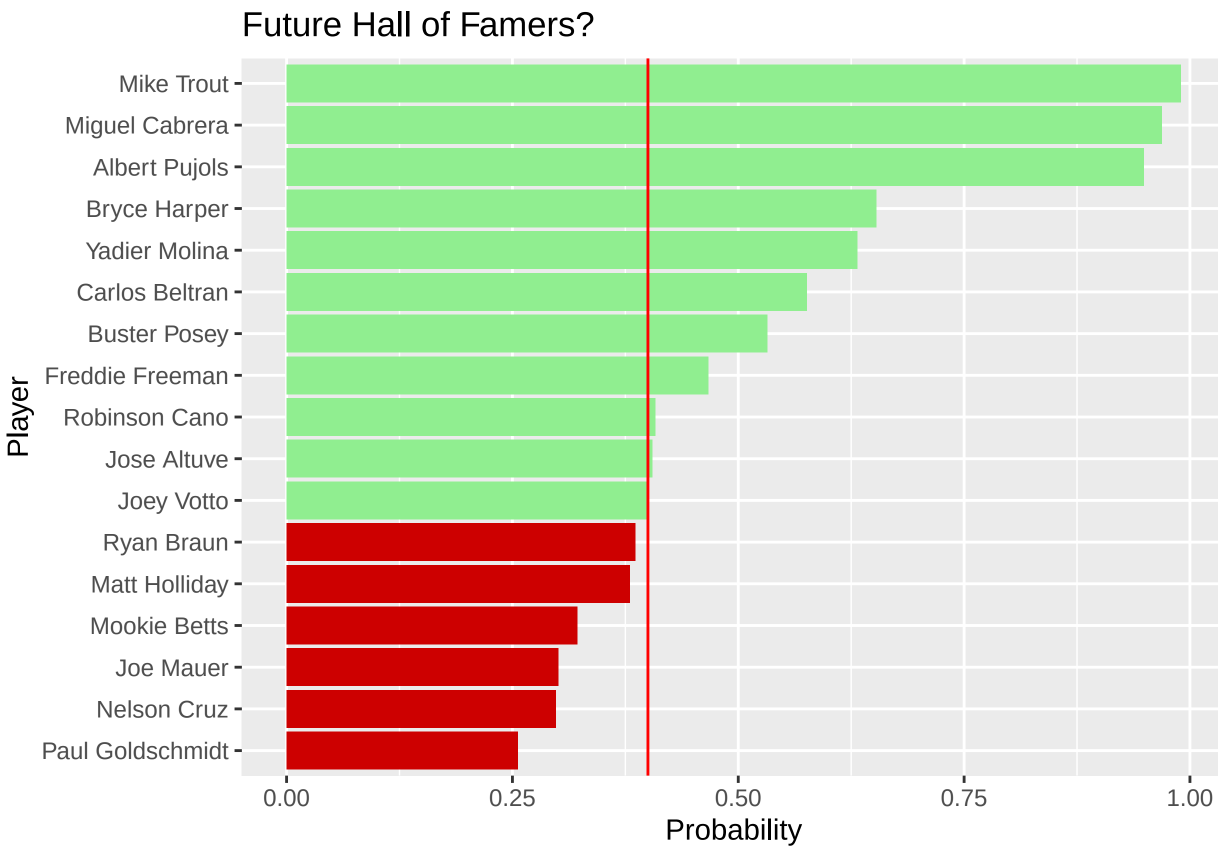
```
## 1         Joe Mauer    0 0.301            2018 6823 137  906 2090 0.3052857
## 2     Matt Holliday    0 0.380            2017 7537 338 1326 2263 0.2979375
## 3      Mookie Betts    0 0.322            2021 3538 163  529 1052 0.2945714
## 4       Nelson Cruz    0 0.298            2021 6235 406 1117 1736 0.2785385
## 5 Paul Goldschmidt    0 0.256            2021 4752 248  830 1387 0.2908889
## 6        Ryan Braun    0 0.386            2019 6493 344 1128 1933 0.2957692
##         OBP  SB  BB IBB       SLG       OPS   TB MVP All_star Gold_glove
## 1 0.3870714  51 928 145 0.4340714 0.8210714 2979   1        6          3
## 2 0.3781250 122 862  65 0.5053125 0.8833125 3852   0        7          0
## 3 0.3727143 136 421  28 0.5242857 0.8970000 1846   1        5          5
## 4 0.3463846  74 614  65 0.5270000 0.8734615 3302   0        7          0
## 5 0.3862222 128 747 105 0.5162222 0.9022222 2469   0        6          3
## 6 0.3579231 215 579  56 0.5291538 0.8871538 3462   1        6          0
##   Silver_slugger hof        RC       ISO
## 1              5   0 1159.9306 0.1287857
## 2              4   0 1433.2063 0.2073750
## 3              4   0  686.8295 0.2297143
## 4              3   0 1132.9683 0.2484615
## 5              4   0  958.1462 0.2253333
## 6              5   0 1229.7149 0.2333846
```

```
nrow(curr2_data)
```

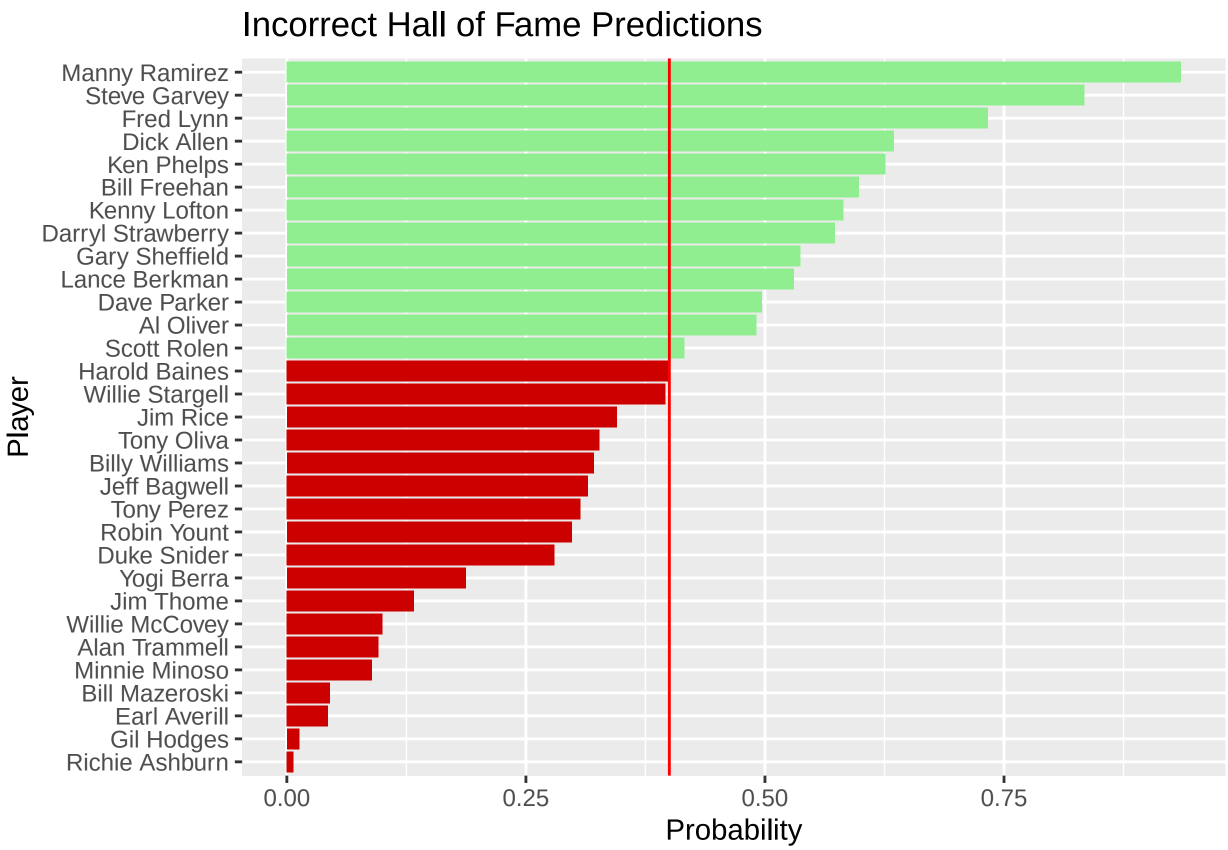
```
## [1] 6
```

```
curr = curr1_data %>% rbind(curr2_data)
```

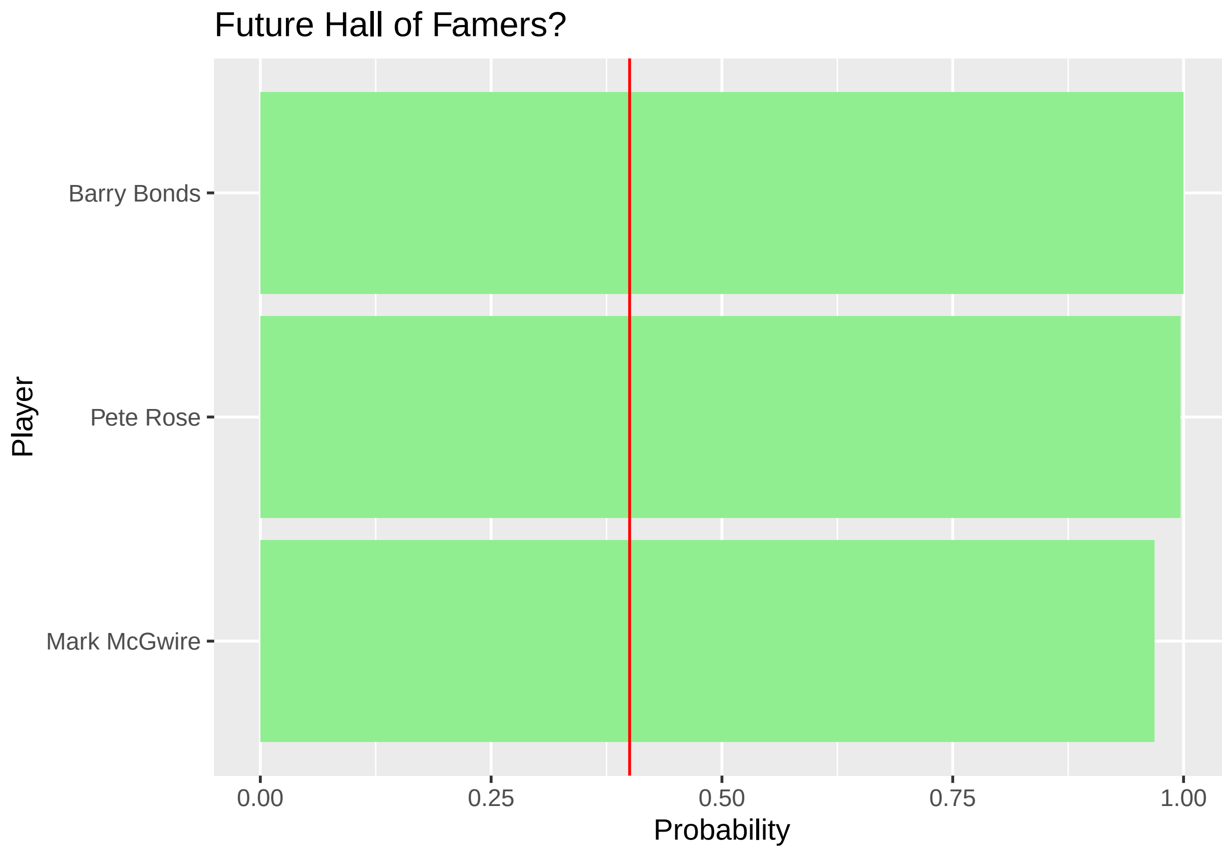
```
ggplot(data = curr, mapping = aes(x = reorder(name, prob), y = prob, fill = as.factor(pred))) + geom_bar
```




```
ggplot(data = modern.end1_data, mapping = aes(x = reorder(name, prob), y = prob, fill = as.factor(pred)
```




```
ggplot(data = incorrect_ban, mapping = aes(x = reorder(incorrect_ban_name, incorrect_ban_prob), y = inc
```


Future Hall of Famers?
Barry Bonds
Pete Rose
Mark McGwire
Player
0.00
0.25
0.50
0.75
1.00
Probability