# Project 2 – SDS 348

*Casey David*

*5/1/2020*

0. The dataset that I have chosen to use for this project is the 'baseball' dataset that is in the 'vcd' package. I chose this dataset because I have spent most of my life hearing my Dad and 2 older brothers talk about baseball; all three of them are able to rattle off random statistics from numerous years and numerous players. I thought this would be an interesting dataset to work with since I understand the game (and I thought it might help me earn some points with my family). In this dataset, there are 25 variables and 322 observations (which all represent a different baseball player). All of the data in this set is statistics and information about these players from the years 1986 and 1987. The variables 'name1' and 'name2' just provide the first and last name of the specific player. Any of the variables that have 'runs' in their name are measuring the number of runs scored by that player in that year. 'Hits' measures the number of hits made by each player, 'atbats' measures the number of times a player has an official appearance at the plate (and doesn't get hit by a pitch or walked by the catcher), 'homer' measures the number of homeruns each player gets, and 'walks' measures the number of times a player gets walked while they are at bat. There is also a variable for salary (sal87) that describes the players' salary for that year in the thousands. There is also a variable to list the players' team ('team'), division ('div86'), and league ('league86', 'league87').

```r
class_diag<-function(probs,truth){

tab<-table(factor(probs>.5,levels=c("FALSE","TRUE")),truth)
acc=sum(diag(tab))/sum(tab)
sens=tab[2,2]/colSums(tab)[2]
spec=tab[1,1]/colSums(tab)[1]
ppv=tab[2,2]/rowSums(tab)[2]

if(is.numeric(truth)==FALSE & is.logical(truth)==FALSE) truth<-as.numeric(truth)-1

#CALCULATE EXACT AUC
ord<-order(probs, decreasing=TRUE)
probs <- probs[ord]; truth <- truth[ord]

TPR=cumsum(truth)/max(1,sum(truth))
FPR=cumsum(!truth)/max(1,sum(!truth))

dup<-c(probs[-1]>=probs[-length(probs)], FALSE)
TPR<-c(0,TPR[!dup],1); FPR<-c(0,FPR[!dup],1)

n <- length(TPR)
auc<- sum( ((TPR[-1]+TPR[-n])/2) * (FPR[-1]-FPR[-n]) )

data.frame(acc,sens,spec,ppv,auc)
}
```

```r
#data loading and prep
library(tidyr)
library(dplyr)
library(ggplot2)
library(tidyr)
```

```r
install.packages("vcd")
library(vcd)

bballdata <- as.data.frame(Baseball)
bballdata <-bballdata %>% na.omit()
```

```r
#MANOVA/ANOVA
baseball <- bballdata %>% filter(posit86 != "UT")
baseball <-baseball %>% filter(posit86 != "S3")
baseball <-baseball %>% filter(posit86 != "O1")
baseball <-baseball %>% filter(posit86 != "OD")
baseball <-baseball %>% filter(posit86 != "3S")
baseball <-baseball %>% filter(posit86 != "OS")
baseball <-baseball %>% filter(posit86 != "1O")
baseball <-baseball %>% filter(posit86 != "DO")
baseball <-baseball %>% filter(posit86 != "2S")
baseball <-baseball %>% filter(posit86 != "32")
baseball <-baseball %>% filter(posit86 != "3O")
baseball <-baseball %>% filter(posit86 != "CD")
baseball <-baseball %>% filter(posit86 != "23")
#here i removed a lot of the positions that aren't common (not the usual 9), and therefore do not have
man1 <- manova(cbind(hits86, homer86, walks86, rbi86)~posit86, data= baseball)

summary(man1)
```

```
## Df Pillai approx F num Df den Df Pr(>F)
## posit86 9 0.70197 5.2975 36 896 < 2.2e-16 ***
## Residuals 224
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```r
summary.aov(man1)
```

```
## Response hits86 :
## Df Sum Sq Mean Sq F value Pr(>F)
## posit86 9 114621 12735.7 8.0996 2.189e-10 ***
## Residuals 224 352214 1572.4
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Response homer86 :
## Df Sum Sq Mean Sq F value Pr(>F)
## posit86 9 5528 614.23 10.522 1.494e-13 ***
## Residuals 224 13077 58.38
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Response walks86 :
## Df Sum Sq Mean Sq F value Pr(>F)
## posit86 9 20702 2300.21 5.7542 3.449e-07 ***
## Residuals 224 89542 399.74
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Response rbi86 :
## Df Sum Sq Mean Sq F value Pr(>F)
## posit86 9 43070 4785.5 9.2881 5.855e-12 ***
## Residuals 224 115412 515.2
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
baseball %>% group_by(posit86)%>%summarize(mean(hits86), mean(homer86), mean(walks86), mean(rbi86))
```

```
## # A tibble: 10 x 5
## posit86 `mean(hits86)` `mean(homer86)` `mean(walks86)`
`mean(rbi86)`
## <fct> <dbl> <dbl> <dbl> <dbl>
## 1 1B 125.  16.4 52.1 68.0
## 2 2B 127.  7 47.5 47.2
## 3 3B 119.  13.9 43.7 60.1
## 4 C 76.8 9.5 32.6 39.8
## 5 CF 129.  12.3 49.4 51.0
## 6 DH 102.  19.2 52.5 65.4
## 7 LF 115.  13.0 48.8 56.6
## 8 OF 74.1 8.18 24.5 35.2
## 9 RF 147.  22.4 54.6 82.9
## 10 SS 114.  6.31 34.7 44.9
```

```
## pairwise
pairwise.t.test(baseball$hits86, baseball$posit86, p.adj= 'none')
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: baseball$hits86 and baseball$posit86
##
## 1B 2B 3B C CF DH LF OF RF
## 2B 0.90329 - - - - - - - -
## 3B 0.56344 0.47235 - - - - - - -
## C 1.3e-05 4.7e-06 5.3e-05 - - - - - -
## CF 0.77187 0.86076 0.38123 4.2e-06 - - - - -
## DH 0.10601 0.08347 0.22115 0.07443 0.06665 - - - -
## LF 0.37823 0.31142 0.70645 0.00109 0.25081 0.38957 - - -
## OF 1.9e-05 8.0e-06 7.6e-05 0.81024 6.8e-06 0.05957
0.00108 - -
## RF 0.06499 0.07798 0.01262 1.5e-09 0.12221 0.00231
0.00895 4.7e-09 -
## SS 0.33924 0.27181 0.67460 0.00047 0.21518 0.37487
0.98986 0.00053 0.00515
##
## P value adjustment method: none
```

```
pairwise.t.test(baseball$walks86, baseball$posit86, p.adj = 'none')
```

```
##
## Pairwise comparisons using t tests with pooled SD
```

```
##
## data: baseball$walks86 and baseball$posit86
##
## 1B 2B 3B C CF DH LF OF RF
## 2B 0.41468 - - - - - - - - -
## 3B 0.12381 0.47502 - - - - - - - -
## C 0.00044 0.00588 0.03313 - - - - - - -
## CF 0.64516 0.73563 0.29902 0.00266 - - - - - -
## DH 0.95399 0.48365 0.20902 0.00507 0.67167 - - - - -
## LF 0.58336 0.82715 0.37474 0.00544 0.91739 0.61823 - - - -
## OF 4.8e-06 9.3e-05 0.00074 0.14806 4.0e-05 0.00018
0.00011 - -
## RF 0.67645 0.22212 0.05284 0.00012 0.38809 0.78200
0.34953 1.2e-06 -
## SS 0.00238 0.02221 0.09671 0.69120 0.01084 0.01398
0.01884 0.07738 0.00072
##
## P value adjustment method: none
```

```
pairwise.t.test(baseball$homer86, baseball$posit86, p.adj = 'none')
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: baseball$homer86 and baseball$posit86
##
## 1B 2B 3B C CF DH LF OF RF
## 2B 2.2e-05 - - - - - - - - -
## 3B 0.24449 0.00084 - - - - - - - -
## C 0.00118 0.22331 0.02560 - - - - - - -
## CF 0.06921 0.01609 0.44254 0.18674 - - - - - -
## DH 0.31410 1.5e-05 0.05256 0.00040 0.01483 - - - - -
## LF 0.14012 0.00944 0.65615 0.11919 0.78250 0.03084 - - - -
## OF 0.00035 0.59390 0.00787 0.53941 0.07175 0.00013
0.04458 - -
## RF 0.00801 3.7e-11 0.00010 7.1e-09 1.4e-05 0.25391
8.4e-05 3.1e-09 -
## SS 5.6e-06 0.74420 0.00025 0.12033 0.00660 4.9e-06
0.00382 0.39804 5.8e-12
##
## P value adjustment method: none
```

```
pairwise.t.test(baseball$rbi86, baseball$posit86, p.adj = 'none')
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: baseball$rbi86 and baseball$posit86
##
## 1B 2B 3B C CF DH LF OF RF
## 2B 0.00138 - - - - - - - - -
## 3B 0.20085 0.03593 - - - - - - - -
## C 9.1e-06 0.22308 0.00065 - - - - - - -
## CF 0.01093 0.55793 0.15288 0.07525 - - - - - -
## DH 0.74622 0.02735 0.50862 0.00160 0.08664 - - - - -
## LF 0.09589 0.16884 0.59202 0.01124 0.42835 0.30207 - - -
## OF 1.9e-06 0.06926 0.00013 0.47369 0.02035 0.00040
0.00264 - -
## RF 0.02796 1.5e-07 0.00042 1.2e-10 4.5e-06 0.03795
0.00022 3.7e-11 -
## SS 0.00039 0.70974 0.01327 0.40404 0.34421 0.01284
0.08538 0.14332 2.5e-08
##
## P value adjustment method: none
```

1. The relationship that I was interested in exploring in this dataset was whether or not there was a relationship between number of hits a player gets vs the position they play in 1986. So, to begin, I ran a MANOVA on this idea but also included walks, homeruns, and rbis from the year 1986. The results from the MANOVA indicated that there was a mean difference across the levels of my 'posit86' categorical variable (p-value very small). From here, since the result was significant, I ran univariate ANOVAS on all four of the numeric variables that I included in the MANOVA. All four of the ANOVAs came back with significant p-values, which indicates that these all show a mean difference across groups. With this result, I moved forward with running pairwise t-tests on all four numeric variables with the categorical 'posit' variable. In total, I ran 1 MANOVA, 4 ANOVAs, and 40 pairwise tests (10 categories in the categorical variable * the 4 numeric variables) which comes out to 45 tests. The probability that I made at least one type I error is 1-.95^45 = 0.900. The Bonferroni correction is 0.5/45 = 0.001111. After this correction, the MANOVA is still significant, and so are all four ANOVAs. From the pairwise tests, I have found that there is a significant difference between Catchers and the general 'OF' (outfield) positions when compared with the other 8 positions. Some of the assumptions for these tests are that this is a random sample and has independent observations. I am not entirely sure if this was a random sample to begin with, but I also made it less random with removing some of the positions in the beginning. There are a lot assumptions on the MANOVAs and they are usually hard to meet/test.

```r
#Randomization test and plot
baseball%>%group_by(div86)%>%summarize(m=mean(hits86))%>%summarize(diff(m))
```

```
## # A tibble: 1 x 1
##   `diff(m)`
##       <dbl>
## 1     -9.03
```

```r
basesamp <- sample_n(baseball, 50)
#rand_dist <-vector()
#for(i in 1:500){
  #new<- data.frame(hits= sample(baseball$hits86), division = baseball$div86)
  #rand_dist[i]<- mean(new[new$div86== "A",]$hits86)- mean(new[new$div86 == #"N",]$hits86)
 # }
#this just kept giving me NAs, no matter what I tested, and I was too late on working on this to ask fo
t.test(data= baseball, hits86~ div86)
```

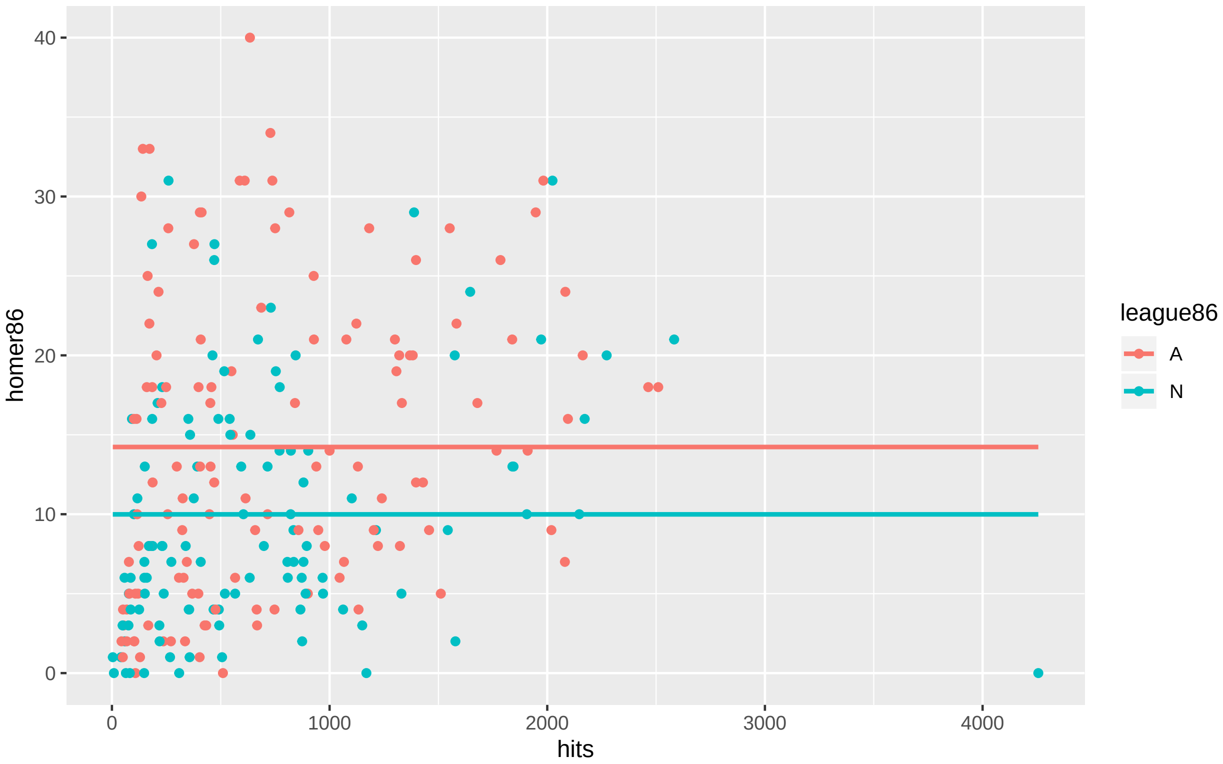
```
##
## Welch Two Sample t-test
##
## data: hits86 by div86
## t = 1.5423, df = 226.45, p-value = 0.1244
## alternative hypothesis: true difference in means is not
equal to 0
## 95 percent confidence interval:
## -2.505898 20.559406
## sample estimates:
## mean in group E mean in group W
## 117.3684 108.3417
```

2. The p-value that resulted from the t-test was not significant, so therefore I was not able to reject the null hypothesis, which states that the true difference in means is equal to 0. The mean hits for each division are very close in range, and therefore do not indicate a significant difference. The alternate hypothesis for this test would have been that the true difference in means is not equal to 0, which would have meant that there was a significant difference in the mean number of hits between the two divisions.

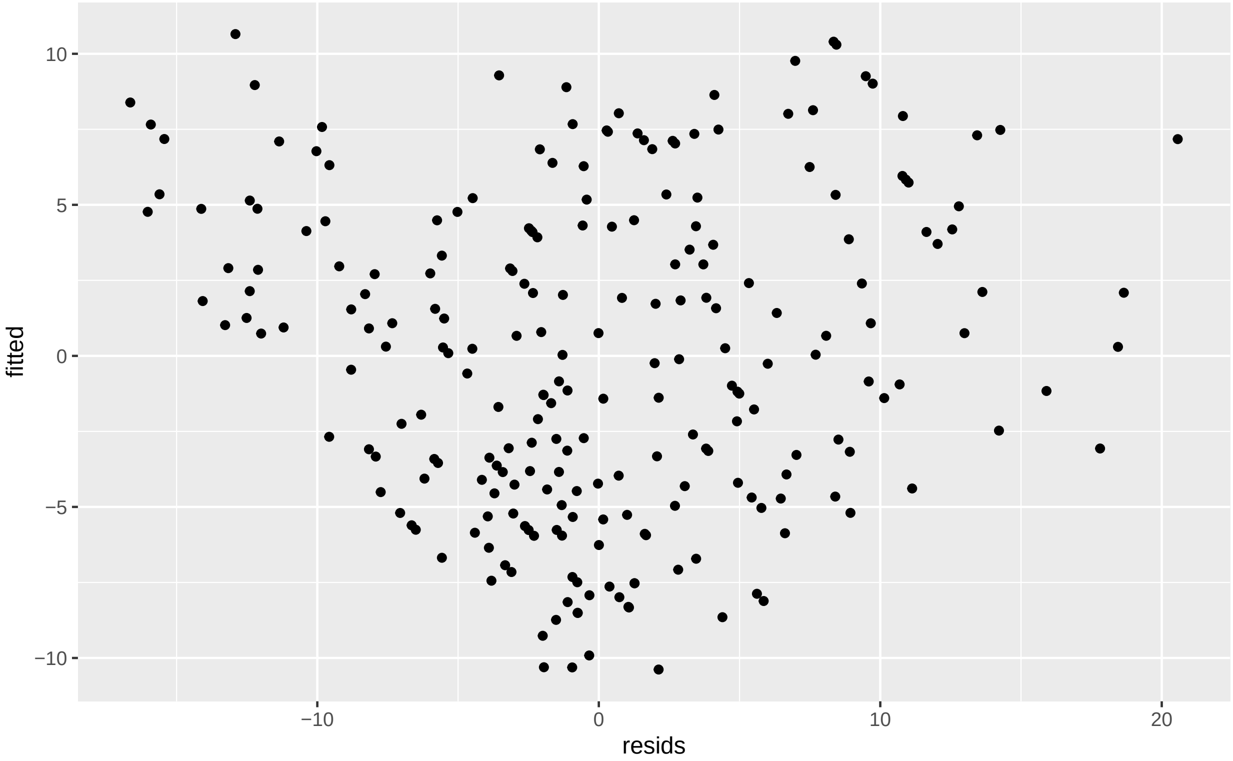
```
#linear regression model
#don't forget to mean-center numeric variables
baseball$homer86_c <- baseball$homer86 - mean(baseball$homer86)
baseball$hits_c <- baseball$hits - mean(baseball$hits)
baseball$atbat86_c <- baseball$atbat86 - mean(baseball$atbat86)
fit1 <- lm(homer86_c~ hits_c * league86 * atbat86_c, data = baseball)
summary(fit1)
```

```
##
## Call:
## lm(formula = homer86_c ~ hits_c * league86 * atbat86_c,
data = baseball)
##
## Residuals:
## Min 1Q Median 3Q Max
## -16.6442 -3.9335 -0.6594 4.3578 20.5685
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.396e+00 7.065e-01 1.976 0.04937 *
## hits_c 1.766e-03 1.148e-03 1.538 0.12549
## league86N -3.259e+00 1.013e+00 -3.216 0.00149 **
## atbat86_c 3.485e-02 5.055e-03 6.894 5.35e-11 ***
## hits_c:league86N 2.218e-04 1.618e-03 0.137 0.89109
## hits_c:atbat86_c -6.689e-06 8.550e-06 -0.782 0.43483
## league86N:atbat86_c -9.498e-03 7.148e-03 -1.329 0.18528
## hits_c:league86N:atbat86_c 1.768e-05 1.118e-05 1.582
0.11509
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Residual standard error: 7.343 on 226 degrees of freedom
## Multiple R-squared: 0.345, Adjusted R-squared: 0.3247
## F-statistic: 17.01 on 7 and 226 DF, p-value: < 2.2e-16
```

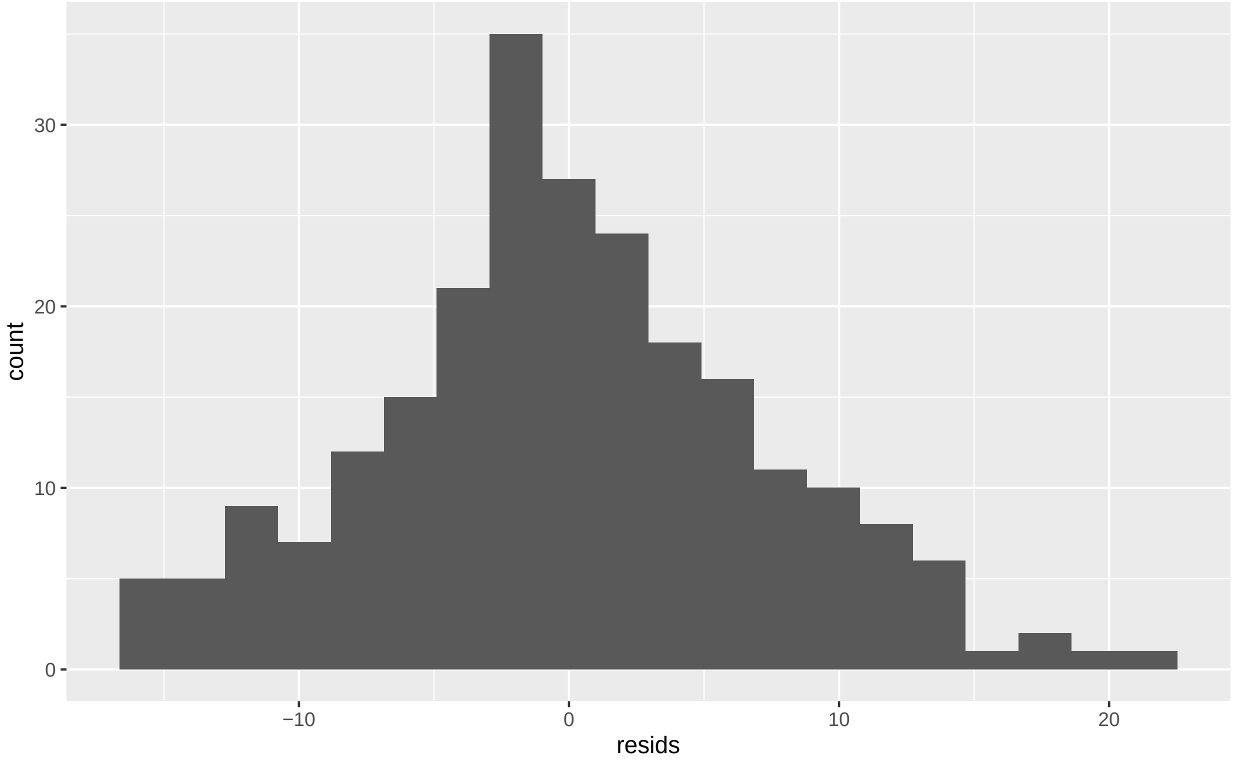
```
ggplot(baseball, aes(x= hits, y=homer86, group=league86)) + geom_point(aes(color=league86)) + geom_smoo
```



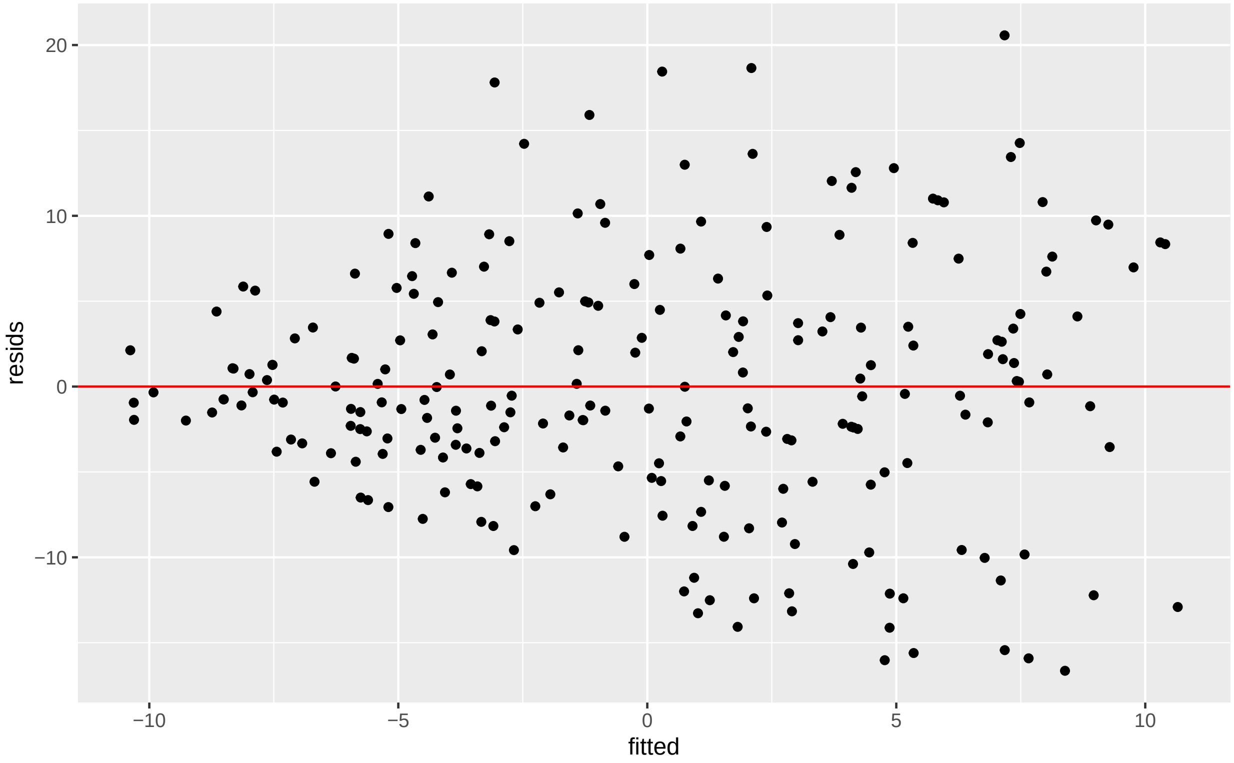
```
resids <-lm(homer86_c~ hits_c * league86 * atbat86_c, data = baseball)$residuals
fitted <-lm(homer86_c~ hits_c * league86 * atbat86_c, data = baseball)$fitted.values
ggplot()+geom_point(aes(resids, fitted))
```



```r
ggplot()+geom_histogram(aes(resids), bins = 20)
```



```r
ggplot()+geom_point(aes(fitted,resids))+geom_hline(yintercept=0, color='red')
```



```
ks.test(resids, "pnorm", mean= 0, sd(resids))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  resids
## D = 0.044919, p-value = 0.7325
## alternative hypothesis: two-sided
```

```
shapiro.test(resids)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resids
## W = 0.99218, p-value = 0.2499
```

```
(sum((baseball$homer86 - mean(baseball$homer86))^2) -sum(fit1$residuals^2))/sum((baseball$homer86-mean(b
```

```
## [1] 0.3450098
```

```
library(sandwich)
library(lmtest)
bptest(fit1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit1
## BP = 46.306, df = 7, p-value = 7.621e-08
```

```
summary(fit1)$coef
```

```
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.396072e+00 7.065069e-01 1.9760204
4.936887e-02
## hits_c 1.766041e-03 1.148396e-03 1.5378328 1.254884e-01
## league86N -3.259399e+00 1.013376e+00 -3.2163768
1.488317e-03
## atbat86_c 3.485355e-02 5.055344e-03 6.8943973
5.345504e-11
## hits_c:league86N 2.217562e-04 1.617725e-03 0.1370790
8.910904e-01
## hits_c:atbat86_c -6.688876e-06 8.549776e-06 -0.7823452
4.348307e-01
## league86N:atbat86_c -9.497675e-03 7.147966e-03
-1.3287242 1.852790e-01
## hits_c:league86N:atbat86_c 1.767929e-05 1.117656e-05
1.5818187 1.150894e-01
```

```
coeftest(fit1, vcov = vcovHC(fit1))
```

```
##
## t test of coefficients:
##
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.3961e+00 7.3478e-01 1.9000 0.058708 .
## hits_c 1.7660e-03 1.0392e-03 1.6994 0.090618 .
## league86N -3.2594e+00 1.0468e+00 -3.1136 0.002087 **
## atbat86_c 3.4854e-02 5.1527e-03 6.7641 1.135e-10 ***
## hits_c:league86N 2.2176e-04 1.6410e-03 0.1351 0.892623
## hits_c:atbat86_c -6.6889e-06 6.8161e-06 -0.9813 0.327475
## league86N:atbat86_c -9.4977e-03 7.1442e-03 -1.3294
0.185048
## hits_c:league86N:atbat86_c 1.7679e-05 1.0900e-05 1.6219
0.106221
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

3. The coefficient for the intercept (going to refer to it as y) is 1.4, which is the result of holding everything else at 0. (I believe the intercept is homeruns examined under league86A (American League). This is not necessarily a meaningful interpretation, as the other variables are not likely to be near 0. There is a very small increase in y (0.00189) for every one unit increase in hits, while also holding everything else constant. There is a decrease by 3.089 in y for the National league vs the American league. There is a 0.0349 increase in y for every 1 unit increase in atbats, holding everything else constant. The coefficients for all of the interactions are very small 0 numbers. The assumptions of linearity, normality, and homoskedasticity were checked graphically and also with a hypothesis test, and passed. I failed to reject the null that true distribution is normal, so the normality of my test is fine! There was a very slight decrease in the standard error when computed with the robust standard error method. The proportion of variation in the outcome that my model explains is 0.345.

```
#same model with bootstrapped standard errors

samp_distn <- replicate(5000, {
  boot_dat <-sample_frac(baseball, replace = T)
  fit2 <-lm(homer86_c~ hits_c * league86 * atbat86_c, data = boot_dat)
  coef(fit2)
})
samp_distn %>% t %>% as.data.frame %>% summarize_all(sd)
```

```
## (Intercept) hits_c league86N atbat86_c hits_c:league86N
hits_c:atbat86_c
## 1 0.7329559 0.001025071 1.018751 0.005123569 0.001565102
6.885098e-06
## league86N:atbat86_c hits_c:league86N:atbat86_c
## 1 0.006857552 1.016518e-05
```

When calculated using the bootstrap method, the standard errors for this model are even slightly smaller than the robust standard errors calculated up above. Since the standard errors are getting slightly smaller, the p-values will be getting slightly bigger.

```
basedata <- baseball %>% mutate(division = ifelse(div86 == "W", 1,0))
fit2 <- glm(division~years+ hits + rbi+ sal87, data= basedata, family = binomial(link = "logit"))
coeftest(fit2)
```

```
##
## z test of coefficients:
##
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.63760021 0.30308899 2.1037 0.0354070 *
## years -0.03041958 0.07045375 -0.4318 0.6659111
## hits 0.00022136 0.00076849 0.2880 0.7733111
## rbi 0.00085713 0.00138849 0.6173 0.5370277
## sal87 -0.00146078 0.00042748 -3.4172 0.0006327 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
exp(coef(fit2))
```

```
## (Intercept) years hits rbi sal87
## 1.8919352 0.9700384 1.0002214 1.0008575 0.9985403
```

```
prob <-predict(fit2, type = "response")
pred<- ifelse(prob>.5,1,0)
table(prediction= pred, truth =basedata$division)%>% addmargins
```

```
##           truth
## prediction   0   1 Sum
##        0    53  35  88
##        1    61  85 146
##        Sum 114 120 234
```

```
#accuracy
(53+85)/234
```

```
## [1] 0.5897436
```

```
#tpr
85/120
```

```
## [1] 0.7083333
```

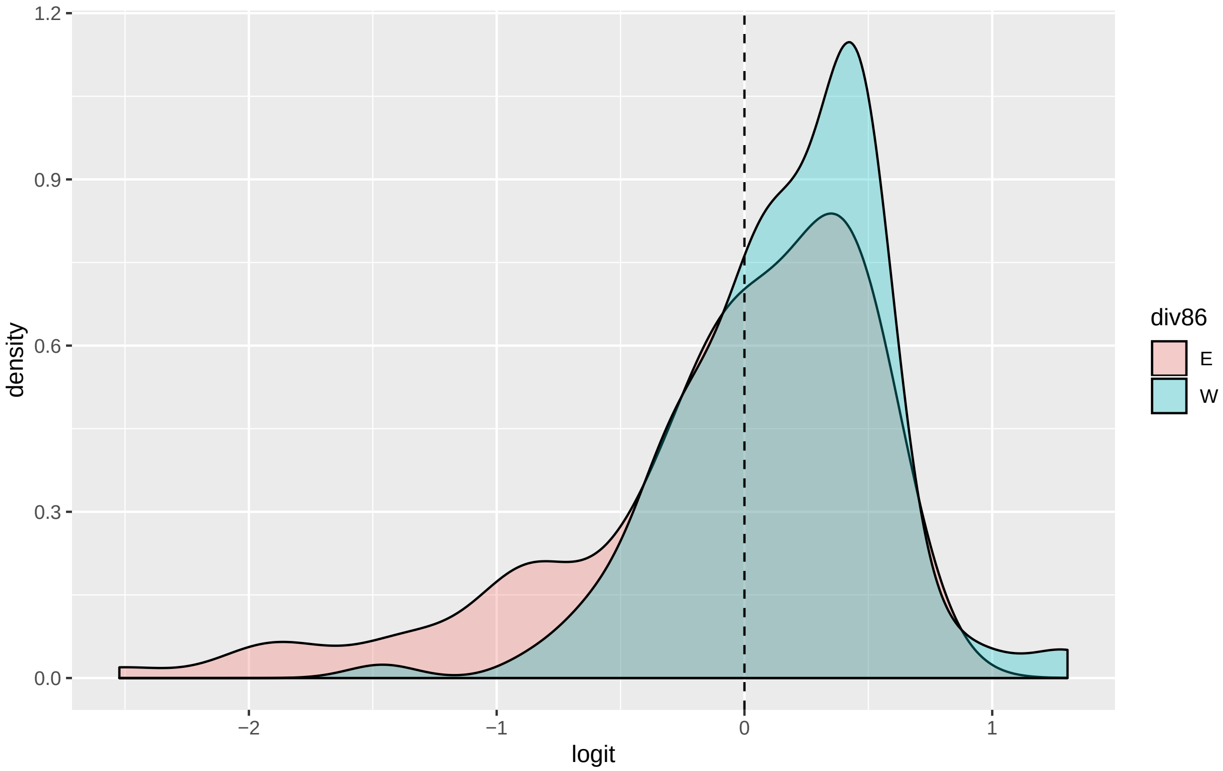
```
#PPV
53/88
```

```
## [1] 0.6022727
```

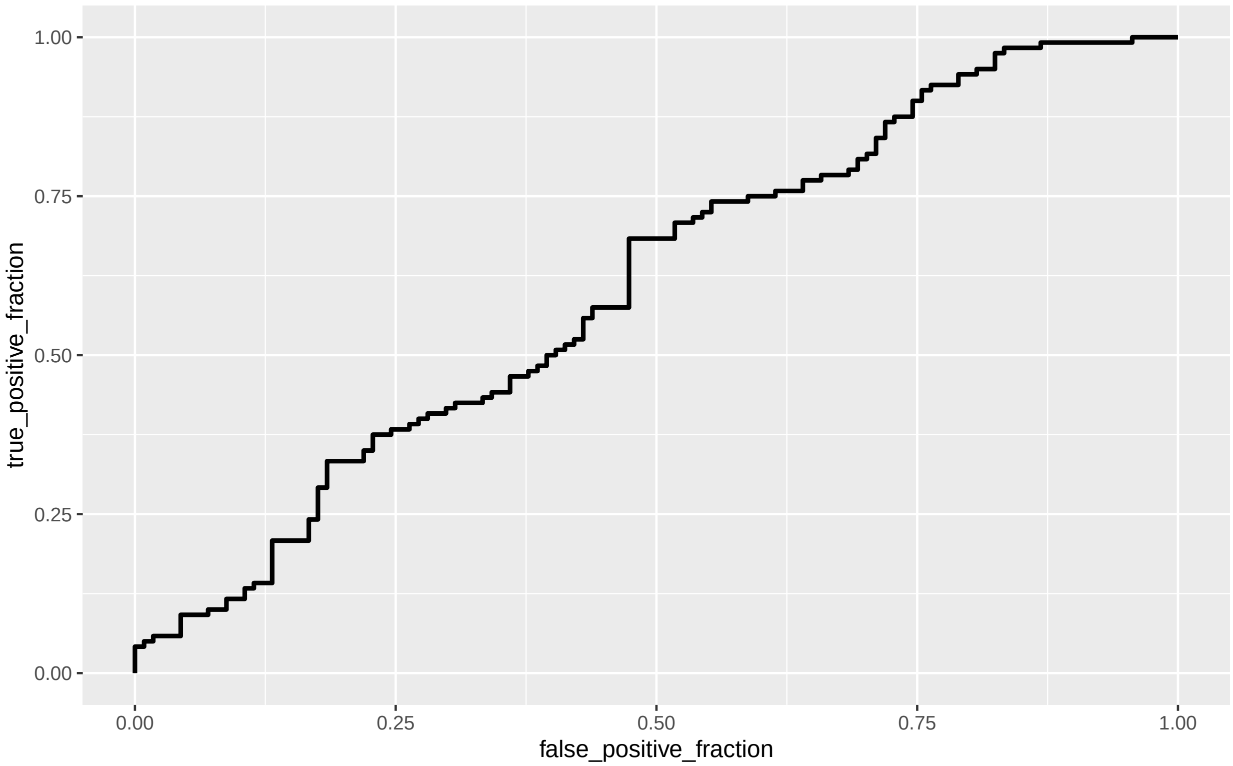
```
#tnr
53/114
```

```
## [1] 0.4649123
```

```
basedata$logit<-predict(fit2)
ggplot(basedata,aes(logit, fill=div86))+geom_density(alpha=.3)+
  geom_vline(xintercept=0,lty=2)
```



```
library(plotROC)
ROCplot <- ggplot(basedata)+ geom_roc(aes(d= division, m=prob), n.cuts = 0)
ROCplot
```



```
calc_auc(ROCplot)
```

```
##   PANEL group       AUC
## 1     1    -1 0.6062865
```

```
set.seed(1234)
k=10

data1 <-basedata[sample(nrow(basedata)),]
folds<-cut(seq(1:nrow(basedata)),breaks=k,labels=F)

diags <-NULL
for(i in 1:k){
  train<-data1[folds!=i,]
  test <-data1[folds==i,]
  truth <- test$division
  fit3 <-glm(division~ years+ hits + rbi+ sal87, data = train, family = "binomial")
  probs <- predict(fit3, newdata = test, type = "response")
  diags <-rbind(diags, class_diag(probs, truth))
}
summarize_all(diags, mean)
```

```
##         acc      sens      spec       ppv       auc
## 1 0.5556159 0.6531258 0.4747541 0.5715327 0.5701255
```

5. When all the other variables are held at 0, the odds of being in the western division are 1.89 (intercept coefficient). When looking at a player for the number of years that they've played, the odds of them being in the Western division are 0.97.

The accuracy was calculated as 0.59, the sensitivity (tpr) is 0.71, the PPV is 0.60, and the specificity (tnr) is 0.46. This model only has a 59% accuracy, which is not the best. It has a higher true positive rate than its true negative rate, which is interesting. The AUC for this model was 0.57, which is classified as Bad! This means that the ROC plot is also not great and that there is not a good trade-off between sensitivity and specificity. The accuracy, sensitivity and recall are reported in the last line resulting from the summarize_all(diags, mean) code.

```
#perform LASSO
newdat <- basedata %>% select(-name1, -name2, -team86, -team87)
newdat <- newdat%>% mutate(bb = ifelse(league86 == "N", 1,0))
head(newdat)
```

```
## atbat86 hits86 homer86 runs86 rbi86 walks86 years atbat
hits homeruns runs rbi walks league86
## 1 185 37 1 23 8 21 2 214 42 1 30 9 24 N
## 2 315 81 7 24 38 39 14 3449 835 69 321 414 375 N
## 3 574 159 21 107 75 59 10 4631 1300 90 702 504 488 A
## 4 239 60 0 30 11 22 6 1941 510 4 309 103 207 A
## 5 202 53 4 31 26 27 9 1876 467 15 192 186 161 N
## 6 594 169 4 74 51 35 11 4408 1133 19 501 336 194 A
## div86 posit86 outs86 assist86 error86 sal87 league87
homer86_c hits_c atbat86_c division
## 1 E 2B 76 127 7 70.000 A -11.25641 -683.3205 -236.5 0
## 2 W C 632 43 10 475.000 N -5.25641 109.6795 -106.5 1
## 3 E SS 238 445 22 517.143 A 8.74359 574.6795 152.5 0
## 4 E 2B 121 151 6 700.000 A -12.25641 -215.3205 -182.5 0
## 5 W C 304 45 11 512.500 N -8.25641 -258.3205 -219.5 1
## 6 W SS 282 421 25 750.000 A -8.25641 407.6795 172.5 1
## logit bb
## 1 0.49151784 1
## 2 0.05754495 1
## 3 0.29773594 0
## 4 -0.36628429 0
## 5 -0.12202330 1
## 6 -0.25380182 0
```

```
library(glmnet)
set.seed(1234)
y <- as.matrix(newdat$hits86)
x <- model.matrix(hits86~., data= newdat)[,-1]
cv.lasso1<-cv.glmnet(x=x[,-1],y=y[,1],family="poisson")
lasso1<-glmnet(x,y,family="poisson",alpha=1,lambda=cv.lasso1$lambda.1se)
coef(lasso1)
```

```
## 50 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 3.7228784669
## atbat86     0.0020517201
## homer86     .
## runs86      0.0013698984
## rbi86       .
## walks86     .
## years       .
## atbat       .
## hits        .
## homeruns    .
## runs        .
## rbi         .
## walks       .
## league86N   .
## div86W      .
## posit861B   .
## posit861O   .
## posit8623   .
## posit862B   .
## posit862S   .
## posit8632   .
## posit863B   .
## posit863O   .
## posit863S   .
## posit86C    .
## posit86CD   .
## posit86CF   .
## posit86CS   .
## posit86DH   .
## posit86DO   .
## posit86LF   .
## posit86O1   .
## posit86OD   .
## posit86OF   .
## posit86OS   .
## posit86RF   .
## posit86S3    .
## posit86SS    .
## posit86UT    .
## outs86       .
## assist86     .
## error86      .
## sal87        .
## league87N    .
## homer86_c    .
## hits_c       .
## atbat86_c    0.0001735721
## division     .
## logit        .
## bb           .
```

```
set.seed(1234)
k=10

data2 <-newdat[sample(nrow(newdat)),]
folds<-cut(seq(1:nrow(newdat)),breaks=k,labels=F)

diags <-NULL
for(i in 1:k){
  train<-data2[folds!=i,]
  test <-data2[folds==i,]
  truth <- test$hits86
  fit4 <-glm(hits86~ atbat86+league86, data = train, family = "poisson")
  probs <- predict(fit4, newdata = test, type = "response")
  diags <-rbind(diags, class_diag(probs, truth))
}
summarize_all(diags, mean)
```

```
##          acc sens spec        ppv auc
## 1 0.04275362    1    0 0.04275362   1
```

6. The only variable that are retained from this lasso on the statistics on hits from the year 1986 is the atbat86 variable and the intercept, which I believe is the American league. The AUC comes out to being 1 and the PPV is 0.0427, which are confusing results to me. But as the game works, the more at bats a player has, the more chances he has to get more hits, so the connection between the two makes sense, but I don't think it should be giving that high of an AUC.