HockeyStats

First, we import the data. Below is the description of each of the variables. Then, we compile all of the years, make sure that Cup is categorical and create a note for the lockout year

# Rank is final rank for the year  
# Team is the team name  
# GP is games played (82 for all full seasons)  
# TOI is time on ice  
# GF is goals for  
# GA is goals against  
# GF60 is goals for per 60 minutes of play  
# GA60 is goals against per 60 minutes of play  
# GF. is goals for percentage, out of all the goals they're involved with  
# SF is shots for (while on ice)  
# SA is shots against (while on ice)  
# Sh. is shot percentage by team while on ice  
# Sv. is save percentage while on the ice  
# PDO is shooting percentage + save percentage while on ice  
# CF is corsi for while on ice  
# CA is corsi against while on ice  
# CF60 is corsi for per 60 minutes of play  
# CA60 is corsi against per 60 minutes of play  
# CF. is corsi for percentage, out of all the corsi events they're involved with  
# CPDO is corsi pdo = fenwick shooting percentage + fenwick save percentage  
# OZFO. is offensive zone faceoff percentage - percentage of faceoffs in offensive zone  
# DZFO. is defensive zone faceoff percentage - percentage of faceoffs in defensive zone  
# NFZO. is neutral zone faceoff percentage - percentage of faceoffs in neutral zone  
# X is   
  
team07<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\200708Team.csv", header=T)  
team08<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\200809Team.csv", header=T)  
team09<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\200910Team.csv", header=T)  
team10<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201011Team.csv", header=T)  
team11<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201112Team.csv", header=T)  
team12<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201213Team.csv", header=T)  
#note that this was the lockout year. don't think it will make a difference in percentages, but might in counts.  
team13<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201314Team.csv", header=T)  
team14<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201415Team.csv", header=T)  
team15<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201516Team.csv", header=T)  
team16<-read.csv("C:\\Users\\Bridget\\Documents\\Math642\\Hockey\\201617Team.csv", header=T)  
#allyears is all bt the current year, as that has not finished yet. that will be the "new data"  
allyears<-rbind(team07, team08, team09, team10, team11, team12, team13, team14, team15)  
allyears$Cup<-as.factor(allyears$Cup)  
str(which(allyears$Year==200809))

## int [1:30] 31 32 33 34 35 36 37 38 39 40 ...

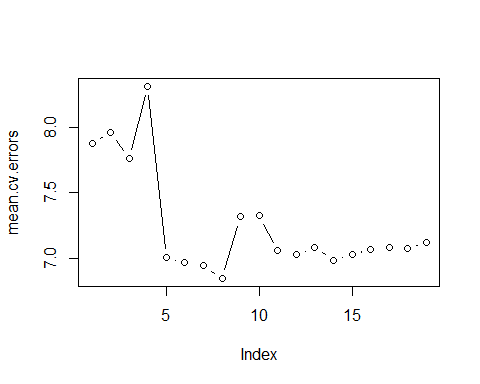
lockout<-which(allyears$Year==201213)  
yearswol<-allyears[-lockout,]  
team16$Cup<-as.factor(c(1,-1, rep(0, 28)))

Now, we start our analysis. First, we want to know what the right number, and which, features to use. So, we use cross-validation with each of the years as test data, in turn. We find that a low number of features leads to the lowest error, 1-3 features does best before a big jump in error.

library(leaps)  
########## create a way of cross-validation where each year individually is the testing set ###  
predict.regsubsets = function (object ,newdata ,id ,...){  
 form=as.formula (object$call [[2]])  
 mat=model.matrix (form ,newdata )  
 coefi =coef(object ,id=id)  
 xvars =names (coefi )  
 mat[,xvars ]%\*% coefi  
}  
#Get the best subset through cross validation. taking out games played and time on ice because games played is constant and time on ice is silly  
folds=c(200708, 200809, 200910, 201011, 201112, 201314, 201415)  
cv.errors<-matrix(NA, 7, 19, dimnames=list(NULL, paste(1:19)))  
for(j in 1:7){  
 best.fit=regsubsets(Rank~GF+GA+GF60+GA60+GF.+SF+SA+Sh.+Sv.+PDO+CF+CA+CF60+CA60+CF.+CPDO+OZFO.+DZFO.+NZFO., data = yearswol[-which(yearswol$Year==folds[j]),], nvmax=19)  
 for(i in 1:19){  
 pred=predict.regsubsets(best.fit,yearswol[which(yearswol$Year==folds[j]),], id=i)  
 cv.errors [j,i]=mean((yearswol[which(yearswol$Year==folds[j]), 1] -pred)^2)  
 }  
}  
mean.cv.errors<-apply(cv.errors, 2, mean)  
mean.cv.errors

## 1 2 3 4 5 6 7 8   
## 7.872856 7.957298 7.762677 8.308998 7.009097 6.969600 6.946382 6.851005   
## 9 10 11 12 13 14 15 16   
## 7.317727 7.326716 7.060109 7.031083 7.085102 6.988771 7.031167 7.071635   
## 17 18 19   
## 7.088489 7.074913 7.124819

#It seems that a 13 or 14 feature selection is best here, which is completely disfferent than how it was before... Although i'll probably choose a 7-8 feature selection becse of simplicity and understanding  
plot(mean.cv.errors, type="b")



#These are the coefficients for these  
coef(best.fit, 8)

## (Intercept) GF GA GF60 GA60   
## 85.767439418 0.857512664 -0.911439996 -59.371774144 58.171772038   
## GF. SA Sh. CF.   
## -2.115766417 0.005332044 1.913272152 0.425164683

lm.8<-lm(Rank~GF+GA+GF60+GA60+GF.+SA+Sh.+CF., data=yearswol)  
summary(lm.8)

##   
## Call:  
## lm(formula = Rank ~ GF + GA + GF60 + GA60 + GF. + SA + Sh. +   
## CF., data = yearswol)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9512 -1.4184 -0.0383 1.6304 7.8485   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 80.966332 25.708519 3.149 0.001852 \*\*   
## GF 1.152352 0.210826 5.466 1.19e-07 \*\*\*  
## GA -1.205087 0.210895 -5.714 3.38e-08 \*\*\*  
## GF60 -79.287371 13.763621 -5.761 2.66e-08 \*\*\*  
## GA60 77.182580 13.557587 5.693 3.77e-08 \*\*\*  
## GF. -2.063256 0.475653 -4.338 2.15e-05 \*\*\*  
## SA 0.005159 0.002294 2.249 0.025480 \*   
## Sh. 2.256616 0.666699 3.385 0.000837 \*\*\*  
## CF. 0.459467 0.155389 2.957 0.003430 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.446 on 231 degrees of freedom  
## Multiple R-squared: 0.9231, Adjusted R-squared: 0.9205   
## F-statistic: 346.7 on 8 and 231 DF, p-value: < 2.2e-16

################################################  
  
#so, now we use this model to predict the outcome of the season! Not sure how good this is as a continous predicotr, because what we are actually looking for are positive integers 1-30, but i wasn't sure how to do that (not within my pay grade you know)  
reg.pred<-predict(lm.8, newdata=team16)  
preds.reg<-data.frame("team"=team16[,2], "predictions"=reg.pred)  
preds.reg[order(preds.reg$predictions),]

## team predictions  
## 1 Washington -33.63329644  
## 2 Minnesota -11.06823013  
## 4 Pittsburgh -8.25815234  
## 3 Columbus -6.63460856  
## 5 Chicago -1.88199868  
## 7 San Jose -1.03284094  
## 6 Montreal 0.08345064  
## 8 Edmonton 6.00188450  
## 9 NY Rangers 8.26453186  
## 10 Nashville 8.56715790  
## 11 NY Islanders 9.40295800  
## 12 Toronto 10.83075992  
## 13 Dallas 11.03826354  
## 14 Anaheim 14.53452025  
## 15 St. Louis 18.29656092  
## 16 Tampa Bay 18.56359422  
## 18 Los Angeles 19.30256655  
## 19 Calgary 20.21988086  
## 17 Winnipeg 20.27588283  
## 21 Boston 21.26973089  
## 20 Detroit 22.17208710  
## 22 Ottawa 22.60543127  
## 23 Buffalo 22.68046533  
## 24 Vancouver 28.82715424  
## 27 New Jersey 33.28089135  
## 25 Florida 33.56976634  
## 26 Arizona 34.32569299  
## 28 Carolina 39.65065407  
## 29 Philadelphia 42.63077787  
## 30 Colorado 64.27472366

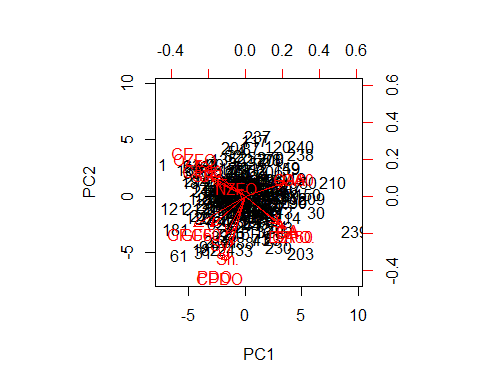
actual2017<-c("Washington", "Columbus", "Minnesota", "Pittsburgh", "Chicago", "Edmonton", "Montreal", "Anaheim", "Nashville", "San Jose", "St. Louis", "NY Ranger", "Toronto", "NY Islanders", "Tampa Bay", "Calgary", "Boston", "Dallas", "Ottawa", "Winnipeg", "Los Angeles", "Detroit", "Carolina", "Philadelphia", "Buffalo", "Florida", "Vancouver", "Arizona", "New Jersey", "Colorado")  
  
##################### DON'T KNOW HOW TO DO IT IN HERE BUT WHEN RPINTING OUT MAKE ALITTLE TABLE OF PREDICTED AND ACTUAL VALUES###########

One reason that the feature seelctor picked so few, and that everything had high errors, is becasue of the high colinearity among the features. We can also perform PCA for a regression and see whta that error is.

#First, we take a look at the kinds of correlations we are dealing with here  
#Top are: CF-SF, TOI-GP, CA-SA, CPDO-DO, GF.-Rank (all above 90%)  
#Next: SF-TOI, SA-TOI, CF-TOI, CA-TOI, SF-GP, SA-GP, SF-GF, CF-GP, SA-GA, CA-GA, CA-GP, Sh.-GF60 (all above 80%)  
#this is a lot of high corelations! It makes sense, looking at the variables that we have (shots against will be very highyl correlated with shots against percentage, after all)  
  
library(reshape)  
cors<-cor(yearswol[,-c(2,24, 25)], use="complete")  
x <- subset(melt(cors), value != 1 | value!=NA)  
x <- x[with(x, order(-abs(x$value))),]  
x

## X1 X2 value  
## 95 GA60 GA 0.954446292  
## 137 GA GA60 0.954446292  
## 72 GF60 GF 0.953200255  
## 114 GF GF60 0.953200255  
## 8 GF. Rank -0.953020889  
## 155 Rank GF. -0.953020889  
## 283 CPDO PDO 0.950863488  
## 409 PDO CPDO 0.950863488  
## 325 CA60 CA 0.932707813  
## 367 CA CA60 0.932707813  
## 302 CF60 CF 0.925831955  
## 344 CF CF60 0.925831955  
## 190 CF SF 0.916171538  
## 295 SF CF 0.916171538  
## 213 CA SA 0.890823958  
## 318 SA CA 0.890823958  
## 192 CF60 SF 0.853165662  
## 339 SF CF60 0.853165662  
## 215 CA60 SA 0.836127465  
## 362 SA CA60 0.836127465  
## 121 Sh. GF60 0.801997454  
## 226 GF60 Sh. 0.801997454  
## 144 Sv. GA60 -0.790212176  
## 249 GA60 Sv. -0.790212176  
## 370 CF. CA60 -0.772595396  
## 391 CA60 CF. -0.772595396  
## 394 OZFO. CF. 0.748618282  
## 436 CF. OZFO. 0.748618282  
## 100 Sv. GA -0.742531205  
## 247 GA Sv. -0.742531205  
## 167 PDO GF. 0.742015812  
## 272 GF. PDO 0.742015812  
## 77 Sh. GF 0.733639399  
## 224 GF Sh. 0.733639399  
## 348 CF. CF60 0.726109861  
## 390 CF60 CF. 0.726109861  
## 96 GF. GA -0.724201611  
## 159 GA GF. -0.724201611  
## 140 GF. GA60 -0.723815062  
## 161 GA60 GF. -0.723815062  
## 118 GF. GF60 0.715628400  
## 160 GF60 GF. 0.715628400  
## 7 GA60 Rank 0.709403281  
## 133 Rank GA60 0.709403281  
## 395 DZFO. CF. -0.709037713  
## 458 CF. DZFO. -0.709037713  
## 5 GA Rank 0.705123838  
## 89 Rank GA 0.705123838  
## 173 CPDO GF. 0.704678129  
## 404 GF. CPDO 0.704678129  
## 74 GF. GF 0.700680112  
## 158 GF GF. 0.700680112  
## 255 PDO Sv. 0.697094220  
## 276 Sv. PDO 0.697094220  
## 239 CPDO Sh. 0.693078662  
## 407 Sh. CPDO 0.693078662  
## 13 PDO Rank -0.686599369  
## 265 Rank PDO -0.686599369  
## 196 OZFO. SF 0.683291073  
## 427 SF OZFO. 0.683291073  
## 329 DZFO. CA 0.683154734  
## 455 CA DZFO. 0.683154734  
## 66 NZFO. TOI -0.681336134  
## 465 TOI NZFO. -0.681336134  
## 219 DZFO. SA 0.678502659  
## 450 SA DZFO. 0.678502659  
## 373 DZFO. CA60 0.673000667  
## 457 CA60 DZFO. 0.673000667  
## 233 PDO Sh. 0.672887210  
## 275 Sh. PDO 0.672887210  
## 326 CF. CA -0.672070761  
## 389 CA CF. -0.672070761  
## 306 OZFO. CF 0.667365069  
## 432 CF OZFO. 0.667365069  
## 19 CPDO Rank -0.664895197  
## 397 Rank CPDO -0.664895197  
## 6 GF60 Rank -0.664688758  
## 111 Rank GF60 -0.664688758  
## 194 CF. SF 0.658345145  
## 383 SF CF. 0.658345145  
## 4 GF Rank -0.650107582  
## 67 Rank GF -0.650107582  
## 216 CF. SA -0.646624429  
## 384 SA CF. -0.646624429  
## 350 OZFO. CF60 0.629215046  
## 434 CF60 OZFO. 0.629215046  
## 304 CF. CF 0.612713290  
## 388 CF CF. 0.612713290  
## 261 CPDO Sv. 0.611225002  
## 408 Sv. CPDO 0.611225002  
## 439 DZFO. OZFO. -0.583718450  
## 460 OZFO. DZFO. -0.583718450  
## 462 NZFO. DZFO. -0.578712315  
## 483 DZFO. NZFO. -0.578712315  
## 172 CF. GF. 0.568639057  
## 382 GF. CF. 0.568639057  
## 103 CA GA 0.568311336  
## 313 GA CA 0.568311336  
## 105 CA60 GA 0.564387062  
## 357 GA CA60 0.564387062  
## 75 SF GF 0.556938916  
## 180 GF SF 0.556938916  
## 18 CF. Rank -0.552521526  
## 375 Rank CF. -0.552521526  
## 58 CF TOI 0.552307383  
## 289 TOI CF 0.552307383  
## 129 CPDO GF60 0.549943759  
## 402 GF60 CPDO 0.549943759  
## 123 PDO GF60 0.547123951  
## 270 GF60 PDO 0.547123951  
## 98 SA GA 0.543689161  
## 203 GA SA 0.543689161  
## 59 CA TOI 0.543217994  
## 311 TOI CA 0.543217994  
## 101 PDO GA -0.542105198  
## 269 GA PDO -0.542105198  
## 85 CPDO GF 0.535626955  
## 400 GF CPDO 0.535626955  
## 145 PDO GA60 -0.528635970  
## 271 GA60 PDO -0.528635970  
## 79 PDO GF 0.525469886  
## 268 GF PDO 0.525469886  
## 149 CA60 GA60 0.520715527  
## 359 GA60 CA60 0.520715527  
## 166 Sv. GF. 0.520500590  
## 250 GF. Sv. 0.520500590  
## 12 Sv. Rank -0.501720989  
## 243 Rank Sv. -0.501720989  
## 165 Sh. GF. 0.495970256  
## 228 GF. Sh. 0.495970256  
## 372 OZFO. CA60 -0.495689903  
## 435 CA60 OZFO. -0.495689903  
## 330 NZFO. CA -0.495439702  
## 477 CA NZFO. -0.495439702  
## 53 SF TOI 0.495021457  
## 179 TOI SF 0.495021457  
## 82 CF60 GF 0.489029281  
## 334 GF CF60 0.489029281  
## 80 CF GF 0.484336479  
## 290 GF CF 0.484336479  
## 107 CPDO GA -0.478223412  
## 401 GA CPDO -0.478223412  
## 220 NZFO. SA -0.473892139  
## 472 SA NZFO. -0.473892139  
## 54 SA TOI 0.473363451  
## 201 TOI SA 0.473363451  
## 151 CPDO GA60 -0.470856803  
## 403 GA60 CPDO -0.470856803  
## 150 CF. GA60 -0.463968710  
## 381 GA60 CF. -0.463968710  
## 106 CF. GA -0.456879333  
## 379 GA CF. -0.456879333  
## 171 CA60 GF. -0.441868553  
## 360 GF. CA60 -0.441868553  
## 11 Sh. Rank -0.438202837  
## 221 Rank Sh. -0.438202837  
## 17 CA60 Rank 0.437831933  
## 353 Rank CA60 0.437831933  
## 126 CF60 GF60 0.431577638  
## 336 GF60 CF60 0.431577638  
## 308 NZFO. CF -0.428529408  
## 476 CF NZFO. -0.428529408  
## 147 CA GA60 0.420568605  
## 315 GA60 CA 0.420568605  
## 163 SF GF. 0.418250422  
## 184 GF. SF 0.418250422  
## 142 SA GA60 0.417429937  
## 205 GA60 SA 0.417429937  
## 9 SF Rank -0.414417994  
## 177 Rank SF -0.414417994  
## 170 CF60 GF. 0.412658867  
## 338 GF. CF60 0.412658867  
## 20 OZFO. Rank -0.412116546  
## 419 Rank OZFO. -0.412116546  
## 119 SF GF60 0.408744913  
## 182 GF60 SF 0.408744913  
## 174 OZFO. GF. 0.407655562  
## 426 GF. OZFO. 0.407655562  
## 152 OZFO. GA60 -0.404946829  
## 425 GA60 OZFO. -0.404946829  
## 164 SA GF. -0.403641346  
## 206 GF. SA -0.403641346  
## 10 SA Rank 0.402334761  
## 199 Rank SA 0.402334761  
## 21 DZFO. Rank 0.398955541  
## 441 Rank DZFO. 0.398955541  
## 169 CA GF. -0.398034614  
## 316 GF. CA -0.398034614  
## 175 DZFO. GF. -0.397407577  
## 448 GF. DZFO. -0.397407577  
## 16 CF60 Rank -0.392081146  
## 331 Rank CF60 -0.392081146  
## 15 CA Rank 0.389871187  
## 309 Rank CA 0.389871187  
## 351 DZFO. CF60 -0.374153230  
## 456 CF60 DZFO. -0.374153230  
## 128 CF. GF60 0.352065697  
## 380 GF60 CF. 0.352065697  
## 84 CF. GF 0.345992957  
## 378 GF CF. 0.345992957  
## 64 OZFO. TOI 0.344216561  
## 421 TOI OZFO. 0.344216561  
## 131 DZFO. GF60 -0.338806961  
## 446 GF60 DZFO. -0.338806961  
## 168 CF GF. 0.336254673  
## 294 GF. CF 0.336254673  
## 198 NZFO. SF -0.335194673  
## 471 SF NZFO. -0.335194673  
## 440 NZFO. OZFO. -0.324370965  
## 482 OZFO. NZFO. -0.324370965  
## 14 CF Rank -0.321445163  
## 287 Rank CF -0.321445163  
## 124 CF GF60 0.318272635  
## 292 GF60 CF 0.318272635  
## 218 OZFO. SA -0.315175406  
## 428 SA OZFO. -0.315175406  
## 264 NZFO. Sv. -0.312974234  
## 474 Sv. NZFO. -0.312974234  
## 109 DZFO. GA 0.303977801  
## 445 GA DZFO. 0.303977801  
## 197 DZFO. SF -0.301375656  
## 449 SF DZFO. -0.301375656  
## 328 OZFO. CA -0.299117112  
## 433 CA OZFO. -0.299117112  
## 108 OZFO. GA -0.290604897  
## 423 GA OZFO. -0.290604897  
## 65 DZFO. TOI 0.288071949  
## 443 TOI DZFO. 0.288071949  
## 242 NZFO. Sh. 0.286385224  
## 473 Sh. NZFO. 0.286385224  
## 374 NZFO. CA60 -0.286171560  
## 479 CA60 NZFO. -0.286171560  
## 86 OZFO. GF 0.279606997  
## 422 GF OZFO. 0.279606997  
## 87 DZFO. GF -0.249122926  
## 444 GF DZFO. -0.249122926  
## 49 GA TOI 0.232806790  
## 91 TOI GA 0.232806790  
## 153 DZFO. GA60 0.227252214  
## 447 GA60 DZFO. 0.227252214  
## 240 OZFO. Sh. -0.222827342  
## 429 Sh. OZFO. -0.222827342  
## 132 NZFO. GF60 0.218048051  
## 468 GF60 NZFO. 0.218048051  
## 307 DZFO. CF -0.207529501  
## 454 CF DZFO. -0.207529501  
## 61 CA60 TOI 0.205599609  
## 355 TOI CA60 0.205599609  
## 55 Sh. TOI -0.203015215  
## 223 TOI Sh. -0.203015215  
## 141 SF GA60 -0.198073949  
## 183 GA60 SF -0.198073949  
## 60 CF60 TOI 0.197739309  
## 333 TOI CF60 0.197739309  
## 352 NZFO. CF60 -0.196094359  
## 478 CF60 NZFO. -0.196094359  
## 263 DZFO. Sv. 0.187890213  
## 452 Sv. DZFO. 0.187890213  
## 393 CPDO CF. -0.179888098  
## 414 CF. CPDO -0.179888098  
## 25 TOI GP 0.177752461  
## 46 GP TOI 0.177752461  
## 130 OZFO. GF60 0.175889903  
## 424 GF60 OZFO. 0.175889903  
## 48 GF TOI 0.175180252  
## 69 TOI GF 0.175180252  
## 234 CF Sh. -0.169454257  
## 297 Sh. CF -0.169454257  
## 301 CA CF 0.169447169  
## 322 CF CA 0.169447169  
## 146 CF GA60 -0.167160197  
## 293 GA60 CF -0.167160197  
## 193 CA60 SF -0.166193292  
## 361 SF CA60 -0.166193292  
## 148 CF60 GA60 -0.161774712  
## 337 GA60 CF60 -0.161774712  
## 416 OZFO. CPDO -0.155729039  
## 437 CPDO OZFO. -0.155729039  
## 120 SA GF60 -0.155717464  
## 204 GF60 SA -0.155717464  
## 210 Sv. SA 0.154512462  
## 252 SA Sv. 0.154512462  
## 187 Sh. SF -0.151922084  
## 229 SF Sh. -0.151922084  
## 125 CA GF60 -0.148608716  
## 314 GF60 CA -0.148608716  
## 154 NZFO. GA60 0.141982843  
## 469 GA60 NZFO. 0.141982843  
## 371 CPDO CA60 0.136813798  
## 413 CA60 CPDO 0.136813798  
## 417 DZFO. CPDO 0.136524991  
## 459 CPDO DZFO. 0.136524991  
## 349 CPDO CF60 -0.129427760  
## 412 CF60 CPDO -0.129427760  
## 50 GF60 TOI -0.128712799  
## 113 TOI GF60 -0.128712799  
## 347 CA60 CF60 -0.127247192  
## 368 CF60 CA60 -0.127247192  
## 305 CPDO CF -0.124874884  
## 410 CF CPDO -0.124874884  
## 44 NZFO. GP -0.123891652  
## 464 GP NZFO. -0.123891652  
## 238 CF. Sh. -0.121548047  
## 385 Sh. CF. -0.121548047  
## 127 CA60 GF60 -0.113637207  
## 358 GF60 CA60 -0.113637207  
## 236 CF60 Sh. -0.109414743  
## 341 Sh. CF60 -0.109414743  
## 214 CF60 SA -0.103750823  
## 340 SA CF60 -0.103750823  
## 104 CF60 GA -0.102881195  
## 335 GA CF60 -0.102881195  
## 56 Sv. TOI 0.102258933  
## 245 TOI Sv. 0.102258933  
## 327 CPDO CA 0.101072888  
## 411 CA CPDO 0.101072888  
## 285 DZFO. PDO 0.100688499  
## 453 PDO DZFO. 0.100688499  
## 262 OZFO. Sv. 0.093668029  
## 430 Sv. OZFO. 0.093668029  
## 212 CF SA 0.090843687  
## 296 SA CF 0.090843687  
## 284 OZFO. PDO -0.090665292  
## 431 PDO OZFO. -0.090665292  
## 43 DZFO. GP 0.088807754  
## 442 GP DZFO. 0.088807754  
## 94 GF60 GA -0.087205827  
## 115 GA GF60 -0.087205827  
## 211 PDO SA 0.085160841  
## 274 SA PDO 0.085160841  
## 143 Sh. GA60 0.079362396  
## 227 GA60 Sh. 0.079362396  
## 37 CA GP 0.075355486  
## 310 GP CA 0.075355486  
## 396 NZFO. CF. 0.074114790  
## 480 CF. NZFO. 0.074114790  
## 256 CF Sv. 0.073462954  
## 298 Sv. CF 0.073462954  
## 28 GF60 GP -0.073176744  
## 112 GP GF60 -0.073176744  
## 217 CPDO SA 0.072974914  
## 406 SA CPDO 0.072974914  
## 282 CF. PDO -0.071984385  
## 387 PDO CF. -0.071984385  
## 188 Sv. SF 0.071960290  
## 251 SF Sv. 0.071960290  
## 237 CA60 Sh. 0.070896458  
## 363 Sh. CA60 0.070896458  
## 57 PDO TOI -0.070046594  
## 267 TOI PDO -0.070046594  
## 73 GA60 GF -0.069448259  
## 136 GF GA60 -0.069448259  
## 278 CF PDO -0.067300623  
## 299 PDO CF -0.067300623  
## 195 CPDO SF -0.066521240  
## 405 SF CPDO -0.066521240  
## 51 GA60 TOI -0.065859565  
## 135 TOI GA60 -0.065859565  
## 110 NZFO. GA -0.062248022  
## 467 GA NZFO. -0.062248022  
## 232 Sv. Sh. -0.061316270  
## 253 Sh. Sv. -0.061316270  
## 189 PDO SF -0.055819311  
## 273 SF PDO -0.055819311  
## 32 SA GP 0.055250521  
## 200 GP SA 0.055250521  
## 241 DZFO. Sh. -0.053713163  
## 451 Sh. DZFO. -0.053713163  
## 176 NZFO. GF. 0.053529788  
## 470 GF. NZFO. 0.053529788  
## 281 CA60 PDO 0.052950173  
## 365 PDO CA60 0.052950173  
## 83 CA60 GF -0.052905416  
## 356 GF CA60 -0.052905416  
## 36 CF GP 0.051530964  
## 288 GP CF 0.051530964  
## 280 CF60 PDO -0.051404643  
## 343 PDO CF60 -0.051404643  
## 22 NZFO. Rank -0.050852279  
## 463 Rank NZFO. -0.050852279  
## 97 SF GA -0.049431770  
## 181 GA SF -0.049431770  
## 52 GF. TOI -0.048184824  
## 157 TOI GF. -0.048184824  
## 29 GA60 GP -0.046528208  
## 134 GP GA60 -0.046528208  
## 63 CPDO TOI -0.045907892  
## 399 TOI CPDO -0.045907892  
## 33 Sh. GP -0.045709229  
## 222 GP Sh. -0.045709229  
## 117 GA60 GF60 -0.043083715  
## 138 GF60 GA60 -0.043083715  
## 324 CF60 CA -0.041813518  
## 345 CA CF60 -0.041813518  
## 257 CA Sv. 0.041271128  
## 320 Sv. CA 0.041271128  
## 209 Sh. SA -0.040890745  
## 230 SA Sh. -0.040890745  
## 122 Sv. GF60 -0.039078323  
## 248 GF60 Sv. -0.039078323  
## 3 TOI Rank 0.038839440  
## 45 Rank TOI 0.038839440  
## 258 CF60 Sv. 0.036722291  
## 342 Sv. CF60 0.036722291  
## 31 SF GP 0.035063033  
## 178 GP SF 0.035063033  
## 191 CA SF 0.034853854  
## 317 SF CA 0.034853854  
## 303 CA60 CF -0.033496259  
## 366 CF CA60 -0.033496259  
## 34 Sv. GP 0.028005042  
## 244 GP Sv. 0.028005042  
## 286 NZFO. PDO -0.026238187  
## 475 PDO NZFO. -0.026238187  
## 40 CF. GP -0.026128294  
## 376 GP CF. -0.026128294  
## 186 SA SF 0.025177636  
## 207 SF SA 0.025177636  
## 38 CF60 GP -0.024798338  
## 332 GP CF60 -0.024798338  
## 2 GP Rank 0.024090558  
## 23 Rank GP 0.024090558  
## 30 GF. GP -0.022767569  
## 156 GP GF. -0.022767569  
## 71 GA GF -0.022474041  
## 92 GF GA -0.022474041  
## 260 CF. Sv. 0.020717243  
## 386 Sv. CF. 0.020717243  
## 42 OZFO. GP 0.020342271  
## 420 GP OZFO. 0.020342271  
## 279 CA PDO 0.019765478  
## 321 PDO CA 0.019765478  
## 62 CF. TOI -0.018474673  
## 377 TOI CF. -0.018474673  
## 235 CA Sh. -0.015083669  
## 319 Sh. CA -0.015083669  
## 81 CA GF 0.015023013  
## 312 GF CA 0.015023013  
## 26 GF GP -0.014364553  
## 68 GP GF -0.014364553  
## 76 SA GF -0.013769666  
## 202 GF SA -0.013769666  
## 35 PDO GP -0.012071661  
## 266 GP PDO -0.012071661  
## 99 Sh. GA 0.011418392  
## 225 GA Sh. 0.011418392  
## 39 CA60 GP 0.010976178  
## 354 GP CA60 0.010976178  
## 88 NZFO. GF 0.009392345  
## 466 GF NZFO. 0.009392345  
## 27 GA GP 0.009151140  
## 90 GP GA 0.009151140  
## 259 CA60 Sv. 0.002708007  
## 364 Sv. CA60 0.002708007  
## 418 NZFO. CPDO -0.002636720  
## 481 CPDO NZFO. -0.002636720  
## 41 CPDO GP 0.002473178  
## 398 GP CPDO 0.002473178  
## 102 CF GA -0.002163300  
## 291 GA CF -0.002163300  
## 78 Sv. GF -0.002039866  
## 246 GF Sv. -0.002039866

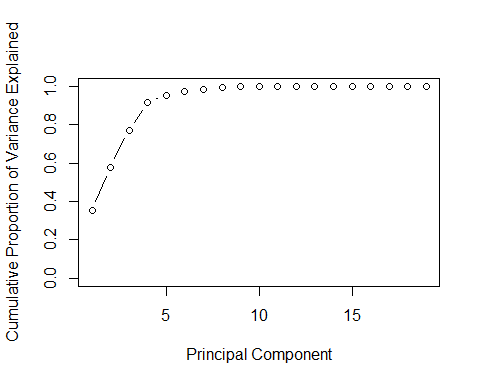
#Now, we preform PCA  
pr.out<-prcomp(yearswol[,-c(1,2,3,4,24,25)], scale=TRUE)  
biplot(pr.out, scale=0)



#This is an interesting plot actually, because almost all of the variables seem to have strong effects. Especially many of them for the second principal component  
pr.var<-pr.out$sdev^2  
pve<-pr.var/sum(pr.var)  
pve

## [1] 3.531021e-01 2.244755e-01 1.942000e-01 1.468556e-01 3.248590e-02  
## [6] 2.180368e-02 1.244252e-02 9.173372e-03 4.687742e-03 2.959912e-04  
## [11] 1.958903e-04 1.033893e-04 6.854941e-05 6.113382e-05 2.505059e-05  
## [16] 1.287275e-05 1.076913e-05 9.178587e-10 7.868607e-12

###so it seems like 4 principal components explain all that variance!  
plot(cumsum (pve ), xlab="Principal Component", ylab ="Cumulative Proportion of Variance Explained", ylim=c(0,1) ,  
type="b")



#But it turns out that the fourth principal component is not significant...  
pcs<-pr.out$x[,1:4]  
str(pcs)

## num [1:240, 1:4] -7.273 -4.015 0.394 -1.848 -0.771 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : chr [1:240] "1" "2" "3" "4" ...  
## ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"

pc.fit<-lm(yearswol$Rank~ pcs[,1]+pcs[,2]+pcs[,3]+pcs[,4])  
#This has a lower adjusted r2 rate than above actually. maybe not as good here... but let's give ourselves a little prediction anyways, why the hell not  
summary(pc.fit)

##   
## Call:  
## lm(formula = yearswol$Rank ~ pcs[, 1] + pcs[, 2] + pcs[, 3] +   
## pcs[, 4])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9488 -1.6801 0.0145 1.9831 9.8331   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 15.50000 0.17769 87.232 <2e-16 \*\*\*  
## pcs[, 1] 2.92058 0.06874 42.485 <2e-16 \*\*\*  
## pcs[, 2] 1.56340 0.08622 18.133 <2e-16 \*\*\*  
## pcs[, 3] -0.13302 0.09270 -1.435 0.153   
## pcs[, 4] -0.15094 0.10660 -1.416 0.158   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.753 on 235 degrees of freedom  
## Multiple R-squared: 0.901, Adjusted R-squared: 0.8993   
## F-statistic: 534.5 on 4 and 235 DF, p-value: < 2.2e-16

################ LASSO ############ We choose lasso for interpretable results, also because i think feature selection might be nice to do  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:reshape':  
##   
## expand

## Loading required package: foreach

## Loaded glmnet 2.0-5

x=model.matrix(Rank~GF+GA+GF60+GA60+GF.+SF+SA+Sh.+Sv.+PDO+CF+CA+CF60+CA60+CF.+CPDO+OZFO.+DZFO.+NZFO., data = yearswol)  
y=yearswol$Rank  
grid =10^ seq (10,-2, length =100)  
train=sample (1: nrow(x), nrow(x)/2)  
test=(- train )  
y.test=y[test]  
  
ridge.mod =glmnet (x[train ,],y[train],alpha =0, lambda =grid)  
cv.out =cv.glmnet (x[train ,],y[train],alpha =0)  
bestlam =cv.out$lambda.min  
ridge.pred=predict (ridge.mod ,s=bestlam ,newx=x[test ,])  
mean(( ridge.pred -y.test)^2)

## [1] 7.183056

out=glmnet (x,y,alpha =0, lambda =grid)  
ridge.coef=predict (out ,type ="coefficients",s=bestlam )[1:20 ,]  
#Wow, this zeros out almost everything!  
ridge.coef

## (Intercept) (Intercept) GF GA GF60   
## 4.437487e+02 0.000000e+00 -5.320379e-02 5.548916e-02 -4.163351e+00   
## GA60 GF. SF SA Sh.   
## 4.388261e+00 -3.973639e-01 -6.639510e-03 5.951977e-03 -3.454219e-01   
## Sv. PDO CF CA CF60   
## -7.029651e-01 -5.780092e-01 -3.450296e-04 4.935012e-04 -4.150597e-02   
## CA60 CF. CPDO OZFO. DZFO.   
## 7.498043e-02 -1.226169e-01 -2.775021e+00 -1.254378e-01 1.091298e-01

ridge.pred<-ridge.coef[1] + ridge.coef[3]\*team16[,5]+ridge.coef[4]\*team16[,6]+ridge.coef[5]\*team16[,7]+ridge.coef[6]\*team16[,8]+ridge.coef[7]\*team16[,9]+ridge.coef[8]\*team16[,10]+ridge.coef[9]\*team16[,11]+ridge.coef[10]\*team16[,12]+ridge.coef[11]\*team16[,13]+ridge.coef[12]\*team16[,14]+ridge.coef[13]\*team16[,15]+ridge.coef[14]\*team16[,16]+ridge.coef[15]\*team16[,17]+ridge.coef[16]\*team16[,18]+ridge.coef[17]\*team16[,19]+ridge.coef[18]\*team16[,20]+ridge.coef[19]\*team16[,21]+ridge.coef[20]\*team16[,22]  
preds.ridge<-data.frame("team"=team16[,2], "predictions"=ridge.pred)  
preds.ridge[order(preds.ridge$predictions),]

## team predictions  
## 1 Washington -10.336561  
## 2 Minnesota 1.965566  
## 4 Pittsburgh 4.230567  
## 3 Columbus 4.882112  
## 5 Chicago 5.658494  
## 7 San Jose 7.842942  
## 6 Montreal 7.925977  
## 8 Edmonton 11.089316  
## 9 NY Rangers 12.247040  
## 10 Nashville 12.983813  
## 11 NY Islanders 13.538550  
## 12 Toronto 14.312404  
## 13 Dallas 15.110055  
## 14 Anaheim 16.356625  
## 18 Los Angeles 17.896226  
## 15 St. Louis 18.052687  
## 16 Tampa Bay 18.168904  
## 17 Winnipeg 19.029346  
## 19 Calgary 19.541865  
## 20 Detroit 19.691666  
## 21 Boston 20.069474  
## 22 Ottawa 20.329771  
## 23 Buffalo 21.142141  
## 24 Vancouver 23.360026  
## 25 Florida 25.603224  
## 27 New Jersey 26.444446  
## 26 Arizona 27.499600  
## 28 Carolina 28.353677  
## 29 Philadelphia 30.229571  
## 30 Colorado 41.588059

lasso.mod =glmnet (x[train ,],y[train],alpha =1, lambda =grid)  
cv.out =cv.glmnet (x[train ,],y[train],alpha =1)  
bestlam =cv.out$lambda.min  
lasso.pred=predict (lasso.mod ,s=bestlam ,newx=x[test ,])  
mean(( lasso.pred -y.test)^2)

## [1] 7.037498

out=glmnet (x,y,alpha =1, lambda =grid)  
lasso.coef=predict (out ,type ="coefficients",s=bestlam )[1:20 ,]  
#Wow, this zeros out almost everything!  
lasso.coef

## (Intercept) (Intercept) GF GA GF60   
## 3.929268e+02 0.000000e+00 0.000000e+00 -1.419298e-02 -5.483244e+00   
## GA60 GF. SF SA Sh.   
## 2.017743e+00 -1.420627e+00 0.000000e+00 2.761490e-03 1.838487e+00   
## Sv. PDO CF CA CF60   
## 0.000000e+00 1.814006e-01 -1.782313e-04 -3.884220e-04 0.000000e+00   
## CA60 CF. CPDO OZFO. DZFO.   
## 1.325230e-01 -2.719755e-03 -3.380408e+00 -1.038148e-01 6.906520e-02

names(team16)

## [1] "Rank" "Team" "GP" "TOI" "GF" "GA" "GF60" "GA60"   
## [9] "GF." "SF" "SA" "Sh." "Sv." "PDO" "CF" "CA"   
## [17] "CF60" "CA60" "CF." "CPDO" "OZFO." "DZFO." "NZFO." "Cup"   
## [25] "Year"

lasso.pred<-lasso.coef[1]+ lasso.coef[6]\*team16[,8]+lasso.coef[7]\*team16[,9]  
preds.lass<-data.frame("team"=team16[,2], "predictions"=lasso.pred)  
preds.lass

## team predictions  
## 1 Washington 304.5597  
## 2 Minnesota 315.5615  
## 3 Columbus 317.1636  
## 4 Pittsburgh 317.7584  
## 5 Chicago 317.6478  
## 6 Montreal 318.6530  
## 7 San Jose 318.9817  
## 8 Edmonton 322.2113  
## 9 NY Rangers 323.5558  
## 10 Nashville 323.7763  
## 11 NY Islanders 324.6878  
## 12 Toronto 325.0852  
## 13 Dallas 324.8301  
## 14 Anaheim 326.1575  
## 15 St. Louis 327.4090  
## 16 Tampa Bay 327.6908  
## 17 Winnipeg 328.6654  
## 18 Los Angeles 328.0780  
## 19 Calgary 328.6275  
## 20 Detroit 329.5151  
## 21 Boston 329.4184  
## 22 Ottawa 330.1166  
## 23 Buffalo 330.1811  
## 24 Vancouver 332.3956  
## 25 Florida 334.6081  
## 26 Arizona 335.1363  
## 27 New Jersey 334.7739  
## 28 Carolina 336.5060  
## 29 Philadelphia 338.7501  
## 30 Colorado 347.1500

preds.lass[order(preds.lass$predictions),]

## team predictions  
## 1 Washington 304.5597  
## 2 Minnesota 315.5615  
## 3 Columbus 317.1636  
## 5 Chicago 317.6478  
## 4 Pittsburgh 317.7584  
## 6 Montreal 318.6530  
## 7 San Jose 318.9817  
## 8 Edmonton 322.2113  
## 9 NY Rangers 323.5558  
## 10 Nashville 323.7763  
## 11 NY Islanders 324.6878  
## 13 Dallas 324.8301  
## 12 Toronto 325.0852  
## 14 Anaheim 326.1575  
## 15 St. Louis 327.4090  
## 16 Tampa Bay 327.6908  
## 18 Los Angeles 328.0780  
## 19 Calgary 328.6275  
## 17 Winnipeg 328.6654  
## 21 Boston 329.4184  
## 20 Detroit 329.5151  
## 22 Ottawa 330.1166  
## 23 Buffalo 330.1811  
## 24 Vancouver 332.3956  
## 25 Florida 334.6081  
## 27 New Jersey 334.7739  
## 26 Arizona 335.1363  
## 28 Carolina 336.5060  
## 29 Philadelphia 338.7501  
## 30 Colorado 347.1500

#now for them all together  
data.frame("Actual"=actual2017, "Reg"=preds.reg[order(preds.reg$predictions),1], "Lasso"=preds.lass[order(preds.lass$predictions),1], "Ridge"=preds.ridge[order(preds.ridge$predictions),1])

## Actual Reg Lasso Ridge  
## 1 Washington Washington Washington Washington  
## 2 Columbus Minnesota Minnesota Minnesota  
## 3 Minnesota Pittsburgh Columbus Pittsburgh  
## 4 Pittsburgh Columbus Chicago Columbus  
## 5 Chicago Chicago Pittsburgh Chicago  
## 6 Edmonton San Jose Montreal San Jose  
## 7 Montreal Montreal San Jose Montreal  
## 8 Anaheim Edmonton Edmonton Edmonton  
## 9 Nashville NY Rangers NY Rangers NY Rangers  
## 10 San Jose Nashville Nashville Nashville  
## 11 St. Louis NY Islanders NY Islanders NY Islanders  
## 12 NY Ranger Toronto Dallas Toronto  
## 13 Toronto Dallas Toronto Dallas  
## 14 NY Islanders Anaheim Anaheim Anaheim  
## 15 Tampa Bay St. Louis St. Louis Los Angeles  
## 16 Calgary Tampa Bay Tampa Bay St. Louis  
## 17 Boston Los Angeles Los Angeles Tampa Bay  
## 18 Dallas Calgary Calgary Winnipeg  
## 19 Ottawa Winnipeg Winnipeg Calgary  
## 20 Winnipeg Boston Boston Detroit  
## 21 Los Angeles Detroit Detroit Boston  
## 22 Detroit Ottawa Ottawa Ottawa  
## 23 Carolina Buffalo Buffalo Buffalo  
## 24 Philadelphia Vancouver Vancouver Vancouver  
## 25 Buffalo New Jersey Florida Florida  
## 26 Florida Florida New Jersey New Jersey  
## 27 Vancouver Arizona Arizona Arizona  
## 28 Arizona Carolina Carolina Carolina  
## 29 New Jersey Philadelphia Philadelphia Philadelphia  
## 30 Colorado Colorado Colorado Colorado

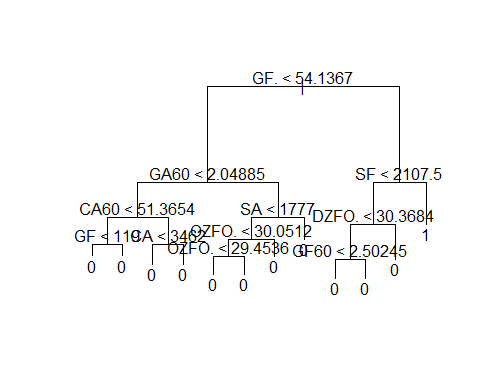
LETS DO SOME CATEGORICAL BUSINESS SHALL WE???

Now, we get into trees to try to predict the cup winner

####### TREES #######  
library(tree)  
tree.cup<-tree(Cup~GF+GA+GF60+GA60+GF.+SF+SA+Sh.+Sv.+PDO+CF+CA+CF60+CA60+CF.+CPDO+OZFO.+DZFO.+NZFO., data=yearswol)  
#miscallsification rate of 5.4%.... not incredibly shabby  
summary(tree.cup)

##   
## Classification tree:  
## tree(formula = Cup ~ GF + GA + GF60 + GA60 + GF. + SF + SA +   
## Sh. + Sv. + PDO + CF + CA + CF60 + CA60 + CF. + CPDO + OZFO. +   
## DZFO. + NZFO., data = yearswol)  
## Variables actually used in tree construction:  
## [1] "GF." "GA60" "CA60" "GF" "CA" "SA" "OZFO." "SF"   
## [9] "DZFO." "GF60"   
## Number of terminal nodes: 12   
## Residual mean deviance: 0.2177 = 49.64 / 228   
## Misclassification error rate: 0.05417 = 13 / 240

plot(tree.cup)  
plot(tree.cup)  
#doesn't work in this new r.....  
text(tree.cup, pretty=0)



#This only classes as 1 a few times...  
tree.cup

## node), split, n, deviance, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 240 139.700 0 ( 0.033333 0.933333 0.033333 )   
## 2) GF. < 54.1367 204 59.540 0 ( 0.024510 0.970588 0.004902 )   
## 4) GA60 < 2.04885 32 28.610 0 ( 0.093750 0.875000 0.031250 )   
## 8) CA60 < 51.3654 20 7.941 0 ( 0.000000 0.950000 0.050000 )   
## 16) GF < 119 5 5.004 0 ( 0.000000 0.800000 0.200000 ) \*  
## 17) GF > 119 15 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 9) CA60 > 51.3654 12 13.500 0 ( 0.250000 0.750000 0.000000 )   
## 18) CA < 3462 6 8.318 0 ( 0.500000 0.500000 0.000000 ) \*  
## 19) CA > 3462 6 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 5) GA60 > 2.04885 172 21.790 0 ( 0.011628 0.988372 0.000000 )   
## 10) SA < 1777 42 16.080 0 ( 0.047619 0.952381 0.000000 )   
## 20) OZFO. < 30.0512 14 11.480 0 ( 0.142857 0.857143 0.000000 )   
## 40) OZFO. < 29.4536 9 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 41) OZFO. > 29.4536 5 6.730 0 ( 0.400000 0.600000 0.000000 ) \*  
## 21) OZFO. > 30.0512 28 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 11) SA > 1777 130 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 3) GF. > 54.1367 36 54.760 0 ( 0.083333 0.722222 0.194444 )   
## 6) SF < 2107.5 31 38.780 0 ( 0.096774 0.806452 0.096774 )   
## 12) DZFO. < 30.3684 16 29.490 0 ( 0.187500 0.625000 0.187500 )   
## 24) GF60 < 2.50245 8 8.997 0 ( 0.000000 0.750000 0.250000 ) \*  
## 25) GF60 > 2.50245 8 15.590 0 ( 0.375000 0.500000 0.125000 ) \*  
## 13) DZFO. > 30.3684 15 0.000 0 ( 0.000000 1.000000 0.000000 ) \*  
## 7) SF > 2107.5 5 5.004 1 ( 0.000000 0.200000 0.800000 ) \*

yearswol$Cup

## [1] 1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## [24] 0 0 0 0 0 0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 0 0   
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0   
## [70] 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1  
## [93] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## [116] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -1  
## [139] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0   
## [162] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0   
## [185] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## [208] 0 0 0 0 0 0 1 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0   
## [231] 0 0 0 0 0 0 0 0 0 0   
## Levels: -1 0 1

library(ISLR)  
train=sample (1:nrow(yearswol), nrow(yearswol)/2)  
yearswol.test<-yearswol[-train,]  
Cup.test<-yearswol$Cup[-train]  
tree.pred<-predict(tree.cup, yearswol.test, type="class")  
table(tree.pred, Cup.test)

## Cup.test  
## tree.pred -1 0 1  
## -1 0 1 0  
## 0 3 111 3  
## 1 0 0 2

mean(tree.pred == Cup.test)

## [1] 0.9416667

#So this only predicts 0s......  
predict(tree.cup, team16, type="class")

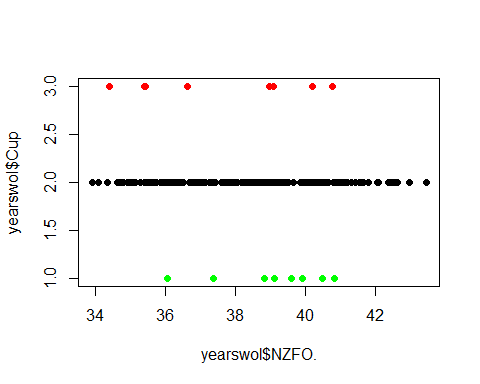
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## Levels: -1 0 1

Now, lets do some logistic regression

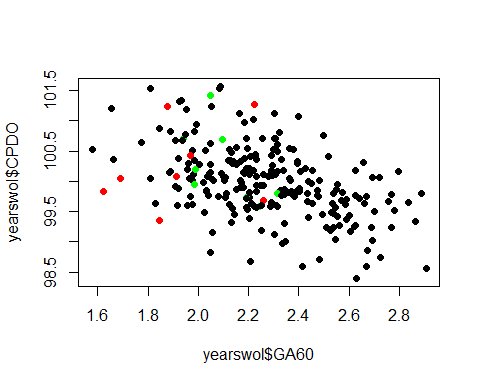
col<-rep("black", nrow(yearswol))  
winners<-which(yearswol[,24]=="1")  
runners<-which(yearswol[,24]=="-1")  
col[winners]<-"red"  
col[runners]<-"green"  
glm.fit<-glm(Cup~ GF + GA + GF60 + GA60 + GF. + SF + SA + Sh. + Sv. + PDO + CF + CA + CF60 + CA60 + CF. + CPDO + OZFO. + DZFO. + NZFO., data=yearswol, family=binomial)  
summary(glm.fit)

##   
## Call:  
## glm(formula = Cup ~ GF + GA + GF60 + GA60 + GF. + SF + SA + Sh. +   
## Sv. + PDO + CF + CA + CF60 + CA60 + CF. + CPDO + OZFO. +   
## DZFO. + NZFO., family = binomial, data = yearswol)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4259 0.0122 0.0583 0.1621 0.8875   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.405e+06 1.255e+06 1.917 0.0553 .  
## GF 3.452e-01 1.285e+00 0.269 0.7882   
## GA -2.617e+00 1.352e+00 -1.935 0.0530 .  
## GF60 3.704e+01 9.839e+01 0.377 0.7065   
## GA60 2.208e+02 9.911e+01 2.228 0.0259 \*  
## GF. -8.437e-01 2.416e+00 -0.349 0.7269   
## SF -1.633e-01 6.701e-02 -2.437 0.0148 \*  
## SA 2.902e-02 4.778e-02 0.607 0.5436   
## Sh. -4.198e+03 2.854e+03 -1.471 0.1414   
## Sv. -4.166e+03 2.851e+03 -1.461 0.1440   
## PDO 4.160e+03 2.852e+03 1.459 0.1446   
## CF 1.138e-01 8.157e-02 1.395 0.1631   
## CA 5.123e-02 4.201e-02 1.219 0.2227   
## CF60 -8.778e+00 5.553e+00 -1.581 0.1139   
## CA60 -1.430e+00 3.584e+00 -0.399 0.6899   
## CF. 1.022e+01 5.201e+00 1.966 0.0493 \*  
## CPDO 4.004e+01 1.970e+01 2.032 0.0422 \*  
## OZFO. -2.409e+04 1.255e+04 -1.919 0.0550 .  
## DZFO. -2.409e+04 1.255e+04 -1.919 0.0550 .  
## NZFO. -2.409e+04 1.255e+04 -1.919 0.0550 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 70.149 on 239 degrees of freedom  
## Residual deviance: 39.853 on 220 degrees of freedom  
## AIC: 79.853  
##   
## Number of Fisher Scoring iterations: 9

#this doesn't seem like many factors are significant, and i want to see if logistic regression is appropriate for all these. so, i will plot everything against Cup  
plot(yearswol$NZFO., yearswol$Cup, col=col, pch=16)



plot(yearswol$GA60, yearswol$CPDO, col=col, pch=16)



glm.small<-glm(Cup~ GA60 + CPDO, data=yearswol, family=binomial)  
summary(glm.small)

##   
## Call:  
## glm(formula = Cup ~ GA60 + CPDO, family = binomial, data = yearswol)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7622 0.1764 0.2343 0.2914 0.5543   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 34.6764 67.9628 0.510 0.610  
## GA60 2.2475 1.6299 1.379 0.168  
## CPDO -0.3616 0.6643 -0.544 0.586  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 70.149 on 239 degrees of freedom  
## Residual deviance: 66.667 on 237 degrees of freedom  
## AIC: 72.667  
##   
## Number of Fisher Scoring iterations: 6

library(MASS)  
#get an error that the variables are collinear, so take away some of the most collinear ones  
lda.all<-lda(Cup~ GF + GA + GF60 + GA60 + GF. + SF + SA + Sh. + Sv. + PDO + CF + CA + CF60 + CA60 + CF. + CPDO + OZFO. + DZFO. + NZFO., data=yearswol)

## Warning in lda.default(x, grouping, ...): variables are collinear

#still get that collienarity error  
lda.small<-lda(Cup~ GA60 + GF. + SF + SA + Sv. + PDO + CA60 + CF. + OZFO. + NZFO., data=yearswol)  
lda.small

## Call:  
## lda(Cup ~ GA60 + GF. + SF + SA + Sv. + PDO + CA60 + CF. + OZFO. +   
## NZFO., data = yearswol)  
##   
## Prior probabilities of groups:  
## -1 0 1   
## 0.03333333 0.93333333 0.03333333   
##   
## Group means:  
## GA60 GF. SF SA Sv. PDO CA60  
## -1 2.100375 53.23281 1894.500 1747.250 92.39705 100.40860 51.08002  
## 0 2.276772 49.70231 1833.388 1848.603 92.16941 99.96846 53.91362  
## 1 1.923700 55.34944 2012.000 1733.250 92.86335 100.45695 49.75610  
## CF. OZFO. NZFO.  
## -1 51.67340 31.06661 39.01436  
## 0 49.81083 30.81777 38.21270  
## 1 53.99481 33.37744 37.60102  
##   
## Coefficients of linear discriminants:  
## LD1 LD2  
## GA60 -2.159423223 4.13651499  
## GF. -0.678098592 1.70543837  
## SF 0.001850156 -0.02868421  
## SA -0.001191267 0.02288307  
## Sv. -1.329778824 1.45432098  
## PDO 2.144557532 -5.75327334  
## CA60 0.031003037 0.10772896  
## CF. 0.127457207 0.25470077  
## OZFO. 0.083272820 0.41911784  
## NZFO. 0.044941951 -0.14760230  
##   
## Proportion of trace:  
## LD1 LD2   
## 0.8339 0.1661

lda.tiny<-lda(Cup ~ GF. + GA + SF + SA + Sh. + Sv. + PDO + CF + CA + OZFO. + DZFO. + NZFO., data = yearswol)

## Warning in lda.default(x, grouping, ...): variables are collinear

lda.tiny$class

## NULL

names(team16)

## [1] "Rank" "Team" "GP" "TOI" "GF" "GA" "GF60" "GA60"   
## [9] "GF." "SF" "SA" "Sh." "Sv." "PDO" "CF" "CA"   
## [17] "CF60" "CA60" "CF." "CPDO" "OZFO." "DZFO." "NZFO." "Cup"   
## [25] "Year"

str(scale(team16[,c(6,9,10,11,12,13,14,15,16,21,22,23)]))

## num [1:30, 1:12] -2.772 -0.775 -0.855 -0.296 -0.935 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:12] "GA" "GF." "SF" "SA" ...  
## - attr(\*, "scaled:center")= Named num [1:12] 115.7 49.86 1502.47 1502.47 7.69 ...  
## ..- attr(\*, "names")= chr [1:12] "GA" "GF." "SF" "SA" ...  
## - attr(\*, "scaled:scale")= Named num [1:12] 12.52 5.52 94.2 104.23 1.03 ...  
## ..- attr(\*, "names")= chr [1:12] "GA" "GF." "SF" "SA" ...

lda.pred<-predict(lda.tiny, data=yearswol)  
lda.class<-lda.pred$class  
table(lda.class, yearswol$Cup)

##   
## lda.class -1 0 1  
## -1 0 0 0  
## 0 8 223 4  
## 1 0 1 4

mean(lda.class==yearswol$Cup)

## [1] 0.9458333

lda.pred<-predict(lda.tiny, data=team16)  
lda.pred$class

## [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [141] 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [176] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## Levels: -1 0 1

table(predict(lda.small, data=yearswol)$class, yearswol$Cup)

##   
## -1 0 1  
## -1 0 0 0  
## 0 8 222 6  
## 1 0 2 2

mean(predict(lda.small, data=yearswol)$class == yearswol$Cup)

## [1] 0.9333333

table(predict(lda.tiny, data=yearswol)$class, yearswol$Cup)

##   
## -1 0 1  
## -1 0 0 0  
## 0 8 223 4  
## 1 0 1 4

mean(predict(lda.tiny, data=yearswol)$class == yearswol$Cup)

## [1] 0.9458333

predict(lda.tiny, data=team16)$class

## [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [141] 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [176] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## Levels: -1 0 1

lda.small<-lda(Cup~ GA60 + GF. + SF + SA + Sv. + PDO + CA60 + CF. + OZFO. + NZFO., data=team16)  
lda.small

## Call:  
## lda(Cup ~ GA60 + GF. + SF + SA + Sv. + PDO + CA60 + CF. + OZFO. +   
## NZFO., data = team16)  
##   
## Prior probabilities of groups:  
## -1 0 1   
## 0.03333333 0.93333333 0.03333333   
##   
## Group means:  
## GA60 GF. SF SA Sv. PDO CA60 CF.  
## -1 2.091900 57.42970 1475.00 1565.00 93.2268 102.92200 58.49420 47.35350  
## 0 2.292793 49.07176 1504.75 1503.75 92.1786 99.71865 54.90211 50.05067  
## 1 1.598800 64.47370 1466.00 1404.00 94.2308 104.25800 52.82070 50.99800  
## OZFO. NZFO.  
## -1 31.07060 35.86630  
## 0 32.10269 35.83093  
## 1 32.37590 37.44680  
##   
## Coefficients of linear discriminants:  
## LD1 LD2  
## GA60 -8.82961048 10.71195943  
## GF. 2.04950987 1.91441971  
## SF -0.02189419 -0.03291906  
## SA 0.04572073 0.01810548  
## Sv. -3.11813102 4.42182257  
## PDO -7.38059788 -6.77424039  
## CA60 -0.15734832 -0.03540070  
## CF. -0.56411605 0.40089672  
## OZFO. -0.16001949 -0.20651101  
## NZFO. -0.28609394 0.49356153  
##   
## Proportion of trace:  
## LD1 LD2   
## 0.9055 0.0945

lda.pred<-predict(lda.small, data=team16)  
lda.pred$class

## [1] 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## [24] 0 0 0 0 0 0 0   
## Levels: -1 0 1

team16[which(predict(lda.small, data=team16)$class == 1),2]

## [1] Washington  
## 30 Levels: Anaheim Arizona Boston Buffalo Calgary Carolina ... Winnipeg

team16[which(predict(lda.small, data=team16)$class == -1),2]

## [1] Minnesota  
## 30 Levels: Anaheim Arizona Boston Buffalo Calgary Carolina ... Winnipeg

names(team16)

## [1] "Rank" "Team" "GP" "TOI" "GF" "GA" "GF60" "GA60"   
## [9] "GF." "SF" "SA" "Sh." "Sv." "PDO" "CF" "CA"   
## [17] "CF60" "CA60" "CF." "CPDO" "OZFO." "DZFO." "NZFO." "Cup"   
## [25] "Year"

summary(team16)

## Rank Team GP TOI GF   
## Min. : 1.00 Anaheim : 1 Min. :61 Min. :2923 Min. : 83.0   
## 1st Qu.: 8.25 Arizona : 1 1st Qu.:63 1st Qu.:3031 1st Qu.:103.0   
## Median :15.50 Boston : 1 Median :64 Median :3060 Median :112.5   
## Mean :15.50 Buffalo : 1 Mean :64 Mean :3071 Mean :115.7   
## 3rd Qu.:22.75 Calgary : 1 3rd Qu.:65 3rd Qu.:3099 3rd Qu.:129.5   
## Max. :30.00 Carolina: 1 Max. :66 Max. :3217 Max. :147.0   
## (Other) :24   
## GA GF60 GA60 GF.   
## Min. : 81.0 Min. :1.685 Min. :1.599 Min. :36.40   
## 1st Qu.:109.5 1st Qu.:2.054 1st Qu.:2.106 1st Qu.:47.30   
## Median :114.0 Median :2.230 Median :2.244 Median :49.21   
## Mean :115.7 Mean :2.259 Mean :2.263 Mean :49.86   
## 3rd Qu.:125.2 3rd Qu.:2.438 3rd Qu.:2.432 3rd Qu.:52.64   
## Max. :145.0 Max. :2.902 Max. :2.943 Max. :64.47   
##   
## SF SA Sh. Sv.   
## Min. :1354 Min. :1325 Min. : 6.028 Min. :90.39   
## 1st Qu.:1439 1st Qu.:1410 1st Qu.: 7.024 1st Qu.:91.86   
## Median :1492 Median :1515 Median : 7.678 Median :92.33   
## Mean :1502 Mean :1502 Mean : 7.695 Mean :92.28   
## 3rd Qu.:1570 3rd Qu.:1577 3rd Qu.: 8.396 3rd Qu.:92.66   
## Max. :1698 Max. :1707 Max. :10.027 Max. :94.23   
##   
## PDO CF CA CF60   
## Min. : 96.96 Min. :2425 Min. :2496 Min. :47.67   
## 1st Qu.: 99.25 1st Qu.:2686 1st Qu.:2682 1st Qu.:52.36   
## Median :100.00 Median :2783 Median :2808 Median :54.75   
## Mean : 99.98 Mean :2813 Mean :2813 Mean :54.95   
## 3rd Qu.:100.75 3rd Qu.:2984 3rd Qu.:2909 3rd Qu.:58.00   
## Max. :104.26 Max. :3162 Max. :3206 Max. :61.04   
##   
## CA60 CF. CPDO OZFO.   
## Min. :48.64 Min. :45.61 Min. : 98.06 Min. :28.25   
## 1st Qu.:52.85 1st Qu.:48.10 1st Qu.: 99.61 1st Qu.:31.05   
## Median :54.31 Median :50.47 Median :100.02 Median :31.97   
## Mean :54.95 Mean :49.99 Mean : 99.99 Mean :32.08   
## 3rd Qu.:57.43 3rd Qu.:51.11 3rd Qu.:100.52 3rd Qu.:33.17   
## Max. :62.32 Max. :55.22 Max. :102.25 Max. :36.83   
##   
## DZFO. NZFO. Cup Year   
## Min. :27.35 Min. :33.77 -1: 1 Min. :201617   
## 1st Qu.:30.73 1st Qu.:35.44 0 :28 1st Qu.:201617   
## Median :32.11 Median :35.82 1 : 1 Median :201617   
## Mean :32.04 Mean :35.89 Mean :201617   
## 3rd Qu.:33.42 3rd Qu.:36.65 3rd Qu.:201617   
## Max. :36.87 Max. :37.45 Max. :201617   
##