

Morphometrics

Sara Michele Schaal

6/28/2018

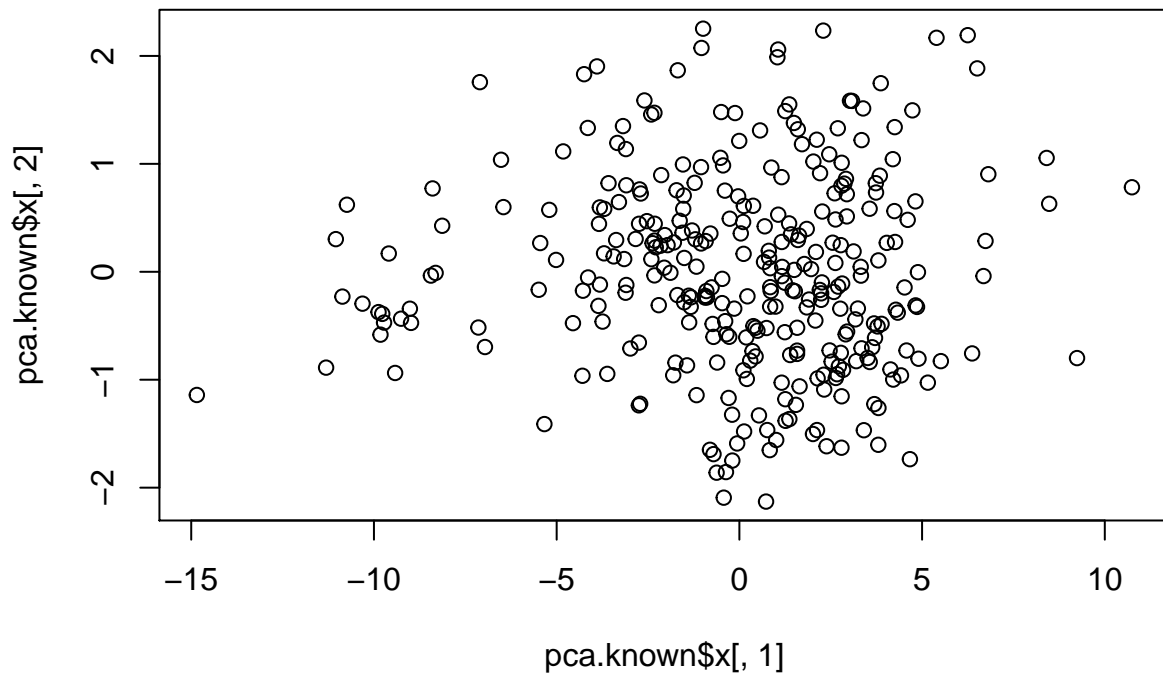
The following analyses are done to determine the spawning group for samples of unknown origin. I begin by creating a Linear Discriminate Analysis model for a dataset of individuals with known spawning origin. Using that model I then apply it to my dataset of unknown spawning origin to get posterior probabilities of assignment to each group.

```
#####  
##### Data & Package Loading #####  
#####  
  
#install.packages("MASS")  
library(MASS)  
  
## Warning: package 'MASS' was built under R version 3.4.3  
  
# Load unknown data  
unknown.morph <- read.csv("../data/Morphometrics/morphPlusMetadata20180628.csv")  
  
# Load known data  
known.morph <- read.csv("../data/Morphometrics/GrahamData.csv")  
  
#####  
#### Data Manipulation ####  
#####  
  
# convert unknown data to mm  
unknown.morph[,4:26] <- 25.4*unknown.morph[,4:26]  
  
# add column with red or olive for unknown samples  
unknown.morph$ColorType <- NA  
for(i in 1:nrow(unknown.morph)){  
  if(unknown.morph$RedGreenRatio[i] >= 1.3){  
    unknown.morph$ColorType[i] <- "Red"  
  } else {  
    unknown.morph$ColorType[i] <- "Olive"  
  }  
}  
  
# convert to cm and take out only the natural log transformed morph data  
unknown.morphlog <- log(unknown.morph[,4:26])  
known.morphlog <- log(known.morph[,c(16:38)]) # already in mm  
#colnames(known.morphlog)[23] <- "TotalLength"  
  
# remove data from landmark 11 - excludes gut fullness from affecting results  
unknown.morphlog <- unknown.morphlog[,c("TotalLength", "D1", "D2", "D3", "D4", "D5",  
                                         "D6", "D7", "D8", "D10", "D13", "D14", "D16",  
                                         "D17", "D18", "D19", "D21", "D22")]  
known.morphlog <- known.morphlog[,c("D1", "D2", "D3", "D4", "D5", "D6", "D7", "D8",
```

```
"D10", "D13", "D14", "D16", "D17", "D18", "D19",  
"D21", "D22"]]
```

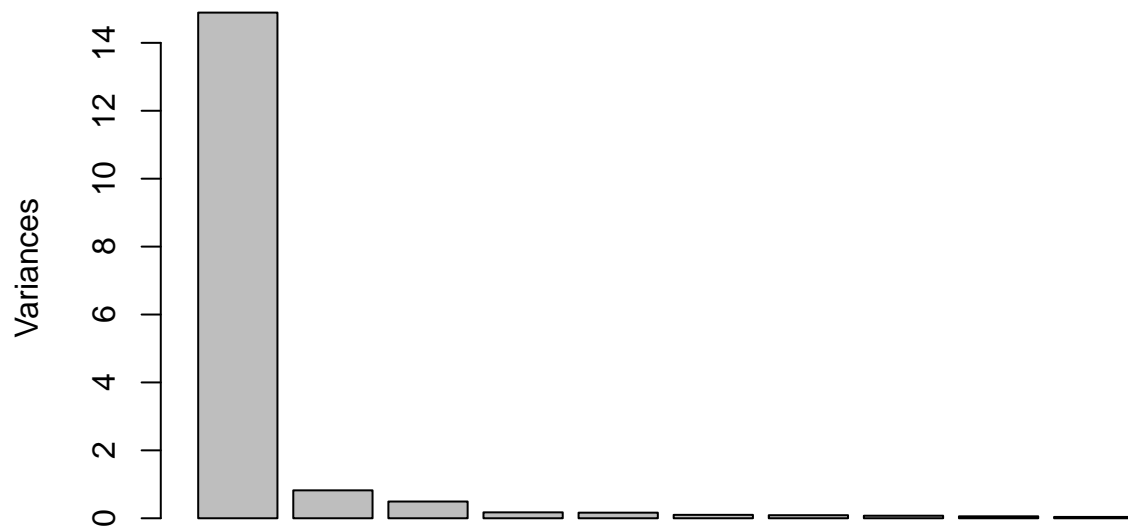
Perform Statistical Analysis on Known Data

```
#####  
#### Run PCA ####  
#####  
# Principle Components Analysis to Account for Total Length of Individuals  
pca.known <- prcomp(known.morphlog, scale. = TRUE)  
pca.known.noScale <- prcomp(known.morphlog)  
summary(pca.known)  
  
## Importance of components%s:  
##  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 3.8588 0.90691 0.70229 0.41778 0.40647 0.31715  
## Proportion of Variance 0.8759 0.04838 0.02901 0.01027 0.00972 0.00592  
## Cumulative Proportion 0.8759 0.92428 0.95329 0.96356 0.97328 0.97920  
## PC7 PC8 PC9 PC10 PC11 PC12  
## Standard deviation 0.30079 0.27433 0.23872 0.19824 0.18315 0.15678  
## Proportion of Variance 0.00532 0.00443 0.00335 0.00231 0.00197 0.00145  
## Cumulative Proportion 0.98452 0.98894 0.99230 0.99461 0.99658 0.99803  
## PC13 PC14 PC15 PC16 PC17  
## Standard deviation 0.10246 0.10043 0.07596 0.06592 0.05322  
## Proportion of Variance 0.00062 0.00059 0.00034 0.00026 0.00017  
## Cumulative Proportion 0.99865 0.99924 0.99958 0.99983 1.00000  
  
max(pca.known$x[,1])  
  
## [1] 10.74289  
  
min(pca.known$x[,1])  
  
## [1] -14.84626  
  
# Plot PCA  
plot(pca.known$x[,1], pca.known$x[,2])
```



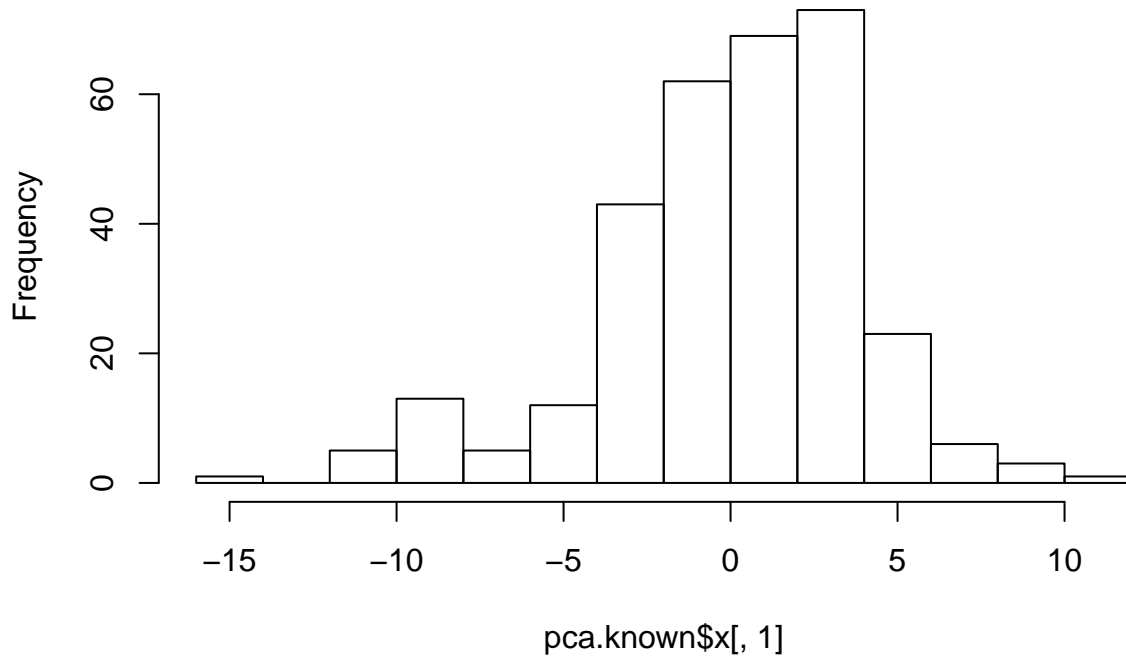
```
screeplot(pca.known)
```

pca.known



```
hist(pca.known$x[,1])
```

Histogram of pca.known\$x[, 1]



```
#####
#### Regress PC1 on Log Transformed Data ####
#####
# Because PC1 corresponds to variance in morphology due to total length of the
# fish we can regress PC1 on our data to account for that variance in downstream
# analyses

# for loop to regress PC1 onto log transformed data and pull out residuals
regress.known <- matrix(nrow = nrow(pca.known$x), ncol = ncol(pca.known$x))
for(i in 1:ncol(pca.known$x)){
  regress.known[,i] <- lm(pca.known$x[,1]~known.morphlog[,i])$residuals
}

# Create dataframe with Spawning Season and Residuals of Regressed Data for DFA
df.knowndata <- data.frame(known.morph$Season, regress.known)

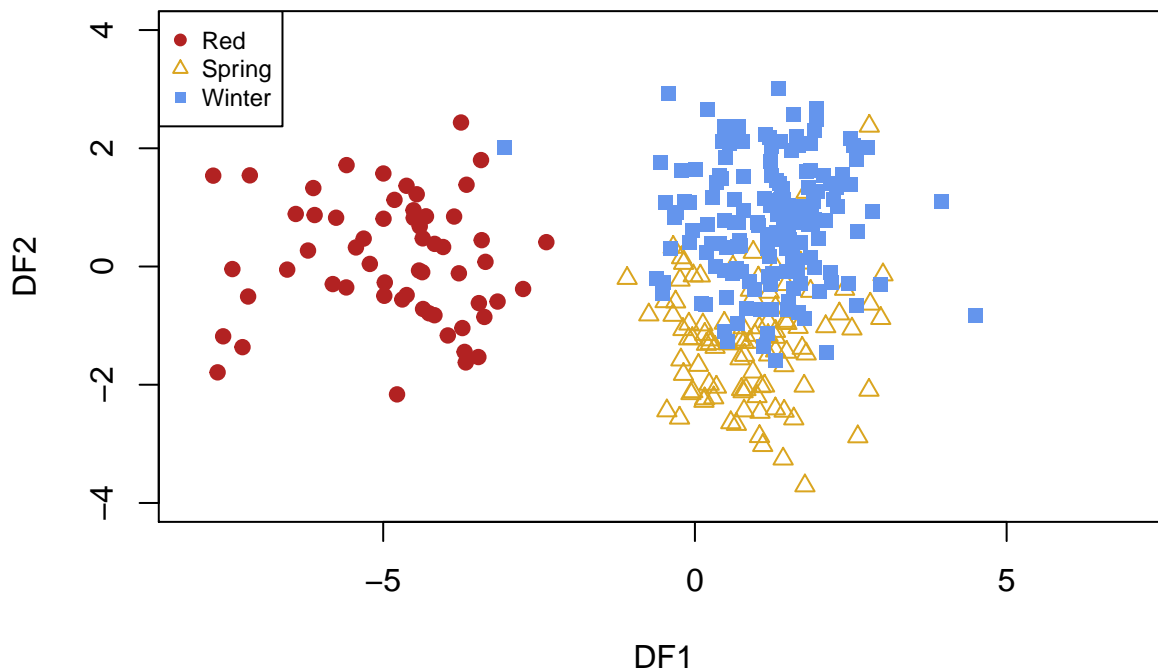
#####
#### Perform Linear Discriminate Analysis ####
#####
# Run DFA
LDAknown.results <- lda(known.morph.Season ~ ., data = df.knowndata)
LDAknown.predict <- predict(LDAknown.results)
str(LDAknown.predict)

## List of 3
## $ class : Factor w/ 3 levels "Red","Spring",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ posterior: num [1:316, 1:3] 0.999 1 1 1 1 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:316] "1" "2" "3" "4" ...
## .. ..$ : chr [1:3] "Red" "Spring" "Winter"
```

```
## $ x      : num [1:316, 1:2] -3.47 -3.43 -4.97 -4.41 -4.37 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:316] "1" "2" "3" "4" ...
## .. ..$ : chr [1:2] "LD1" "LD2"
```

```
# Plot DFA
plot.data <- data.frame(LDAknown.predict$x, df.knowndata$known.morph.Season)
colnames(plot.data)[3] <- "Season"
plot(plot.data$LD1[plot.data$Season == "Red"],
      plot.data$LD2[plot.data$Season == "Red"], col = "firebrick",
      ylim = c(-4,4), xlim = c(-8, 7), pch = 19,
      ylab = "DF2", xlab = "DF1", main = "Linear Discriminate Analysis of Cod Ecotype Morphology")
points(plot.data$LD1[plot.data$Season == "Spring"],
        plot.data$LD2[plot.data$Season == "Spring"], col = "goldenrod",
        pch = 2)
points(plot.data$LD1[plot.data$Season == "Winter"],
        plot.data$LD2[plot.data$Season == "Winter"], col = "cornflowerblue",
        pch = 15)
legend("topleft", legend = c("Red", "Spring", "Winter"),
       pch = c(19, 2, 15), cex = 0.75,
       col = c("firebrick", "goldenrod", "cornflowerblue"))
```

Linear Discriminate Analysis of Cod Ecotype Morphology



```
# LD1: D13 D14 D8 D7
# LD2: D22 D13 D8 D4
```

Results from SPSS

Essentially the same as R just has an inverted sign for DF2 for some reason... haven't figured that out yet

```
SPSS.data <- read.csv("/Users/saraschaal/Desktop/GrahamMorph/GrahamSPSSData.csv")
SPSS.LDA <- read.csv("~/Desktop/GrahamMorph/GrahamDataLDA.csv")
SPSS.Log <- SPSS.data[,grep("Ln", names(SPSS.data))] # Same as R
regress.known.SPSS <- SPSS.data[,grep("RES", names(SPSS.data))] # Same as R when PCA is NOT scaled

head(SPSS.LDA$LDA1)

## [1] -3.40 -3.42 -5.02 -4.32 -4.29 -7.00

head(SPSS.LDA$LDA2)

## [1] 1.35 -1.69 0.29 -0.73 -0.49 0.64

head(LDAknown.predict$x[,1])

##          1          2          3          4          5          6
## -3.473184 -3.430827 -4.974498 -4.413652 -4.365435 -7.166811

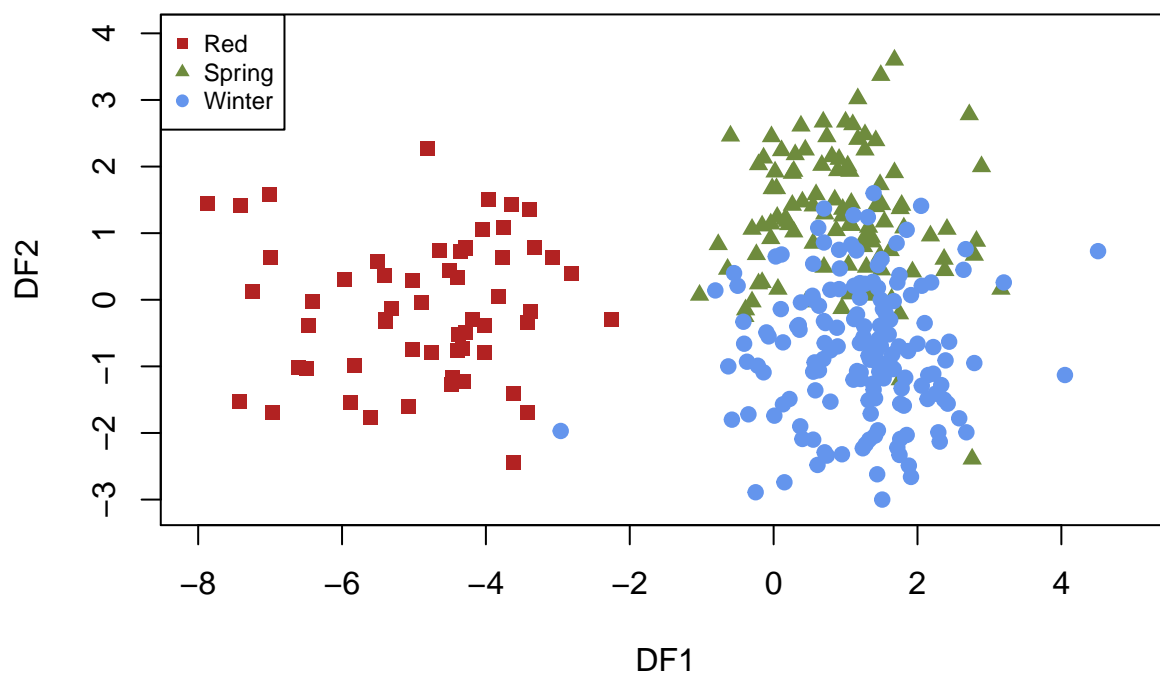
head(LDAknown.predict$x[,2])

##          1          2          3          4          5          6
## -1.5317829 1.8005379 -0.2702528 0.6791140 0.4759889 -0.5073993

# sign is flipped and numbers are SLIGHTLY different from SPSS
# R and SPSS may calculate the LDA slightly differently,
# but doesn't have large impacts on the results

plot(SPSS.LDA$LDA1[SPSS.LDA$Season == "Spring"],
     SPSS.LDA$LDA2[SPSS.LDA$Season == "Spring"], col = "darkolivegreen4",
     pch = 17, xlim = c(-8, 5), ylim = c(-3.1, 4),
     xlab = "DF1", ylab = "DF2", main = "SPSS")
points(SPSS.LDA$LDA1[SPSS.LDA$Season == "Winter"],
       SPSS.LDA$LDA2[SPSS.LDA$Season == "Winter"], col = "cornflowerblue",
       pch = 19)
points(SPSS.LDA$LDA1[SPSS.LDA$Season == "Red"],
       SPSS.LDA$LDA2[SPSS.LDA$Season == "Red"], col = "firebrick",
       pch = 15)
legend("topleft", legend = c("Red", "Spring", "Winter"),
      pch = c(15, 17, 19), cex = 0.75,
      col = c("firebrick", "darkolivegreen4", "cornflowerblue"))
```

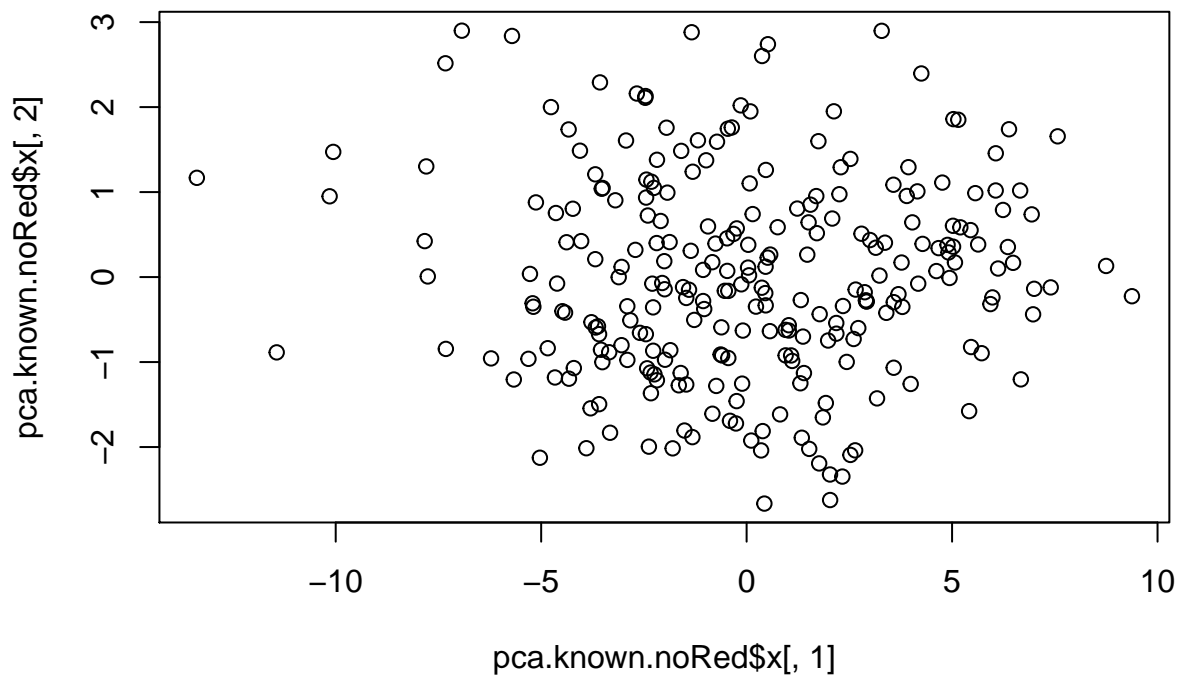
SPSS



Remove Red cod and reanalyze

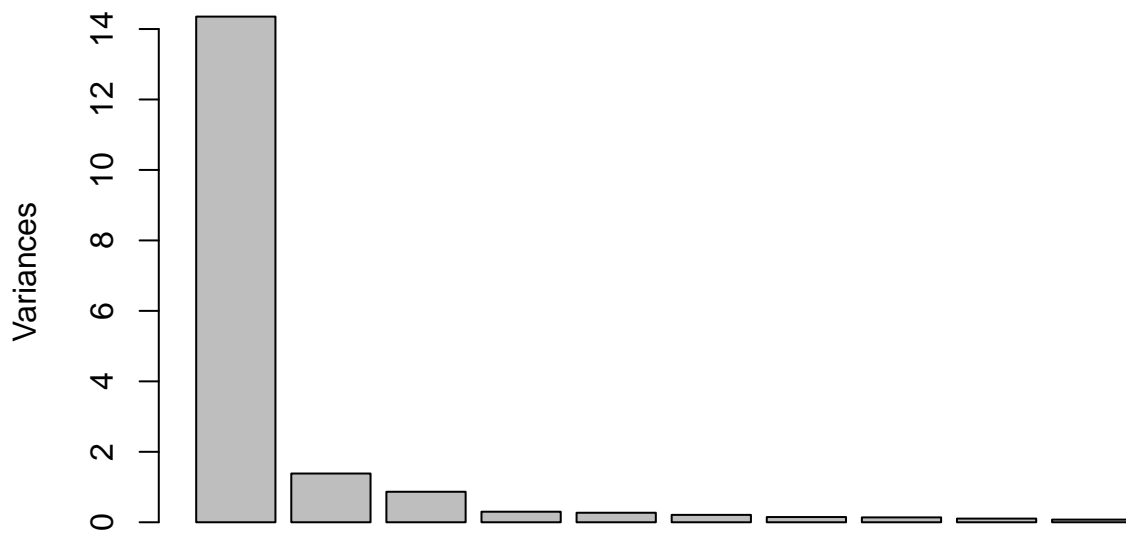
The red cod in the known dataset did not have any length measurements so I had to exclude total length from that analysis. Here I removed red cod from both the known dataset and the unknown dataset to specify just spring or winter spawners in my data. This results in a single LD axis.

```
#####  
#### Run LDA for only Spring and Winter cod ####  
#####  
library(randomForest)  
  
## Warning: package 'randomForest' was built under R version 3.4.4  
## randomForest 4.6-14  
## Type rfNews() to see new features/changes/bug fixes.  
  
# remove Red cod data  
known.noRed <- known.morph[known.morph$Season != "Red", ]  
unknown.noRed <- unknown.morph[unknown.morph$ColorType != "Red", ]  
unknown.lognoRed <- log(unknown.noRed[,c("TotalLength", "D1", "D2", "D3", "D4", "D5",  
                                         "D6", "D7", "D8", "D10", "D13", "D14", "D16",  
                                         "D17", "D18", "D19", "D21", "D22")])  
  
colnames(unknown.lognoRed)[1] <- "D23"  
unknown.lognoRed <- unknown.lognoRed[,c(2:18, 1)]  
known.noRed$Season <- factor(known.noRed$Season)  
known.lognoRed <- log(known.noRed[,c(16:23, 25, 28:29, 31:34, 36:38)])  
  
# Run a Principle Components Analysis on both datasets  
pca.known.noRed <- prcomp(known.lognoRed, scale. = TRUE)  
summary(pca.known.noRed)  
  
## Importance of components%  
##  
##          PC1      PC2      PC3      PC4      PC5      PC6  
## Standard deviation  3.7885 1.17616 0.9305 0.54720 0.51997 0.45794  
## Proportion of Variance 0.7974 0.07685 0.0481 0.01664 0.01502 0.01165  
## Cumulative Proportion 0.7974 0.87421 0.9223 0.93895 0.95397 0.96562  
##          PC7      PC8      PC9     PC10     PC11     PC12  
## Standard deviation  0.38550 0.3722 0.32288 0.27569 0.23523 0.20398  
## Proportion of Variance 0.00826 0.0077 0.00579 0.00422 0.00307 0.00231  
## Cumulative Proportion 0.97387 0.9816 0.98736 0.99158 0.99466 0.99697  
##          PC13     PC14     PC15     PC16     PC17     PC18  
## Standard deviation  0.13744 0.10235 0.09400 0.08306 0.07746 0.05865  
## Proportion of Variance 0.00105 0.00058 0.00049 0.00038 0.00033 0.00019  
## Cumulative Proportion 0.99802 0.99860 0.99909 0.99948 0.99981 1.00000  
  
pca.unknown.noRed<- prcomp(unknown.lognoRed, scale. = TRUE)  
  
# Plot PCA  
plot(pca.known.noRed$x[,1], pca.known.noRed$x[,2])
```

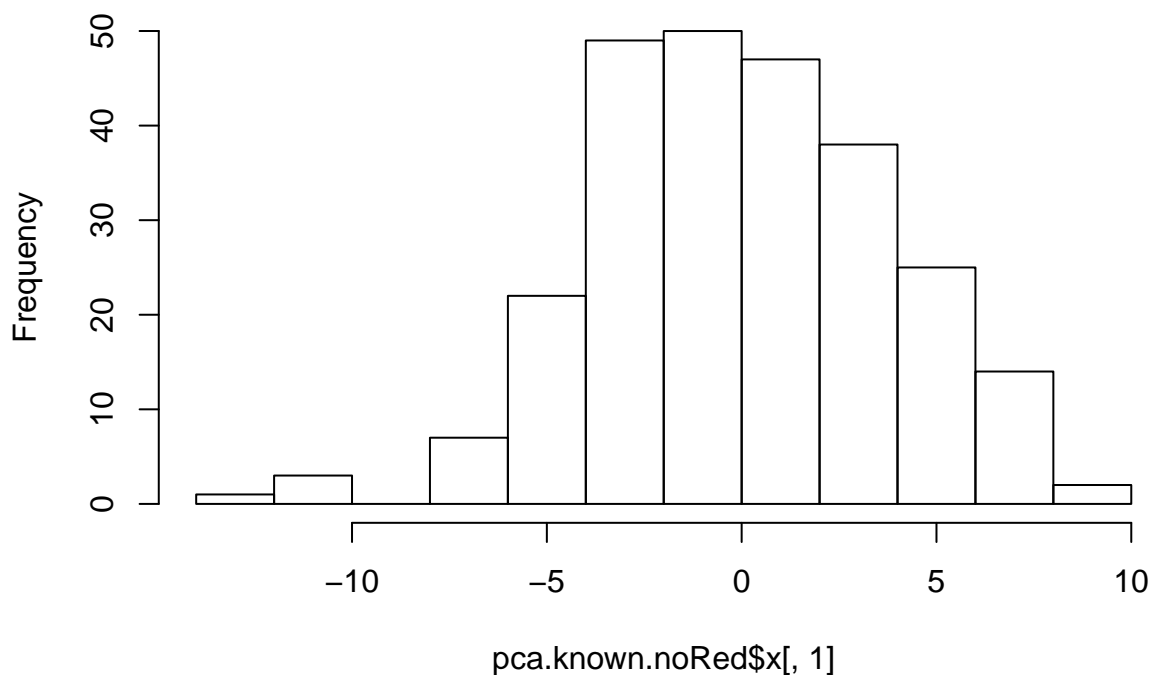
```
screeplot(pca.known.noRed)
```

pca.known.noRed



```
hist(pca.known.noRed$x[,1])
```

Histogram of `pca.known.noRed$x[, 1]`

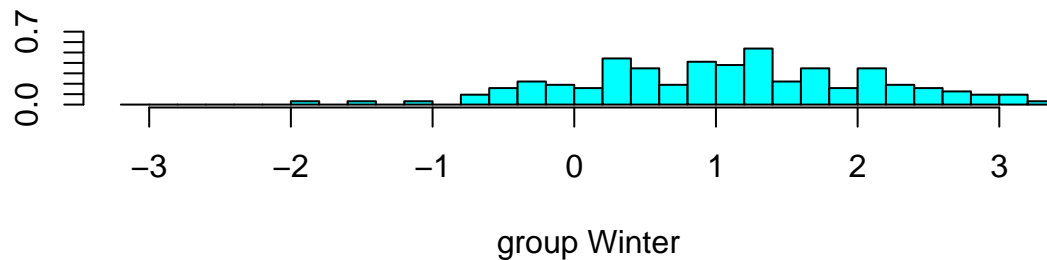
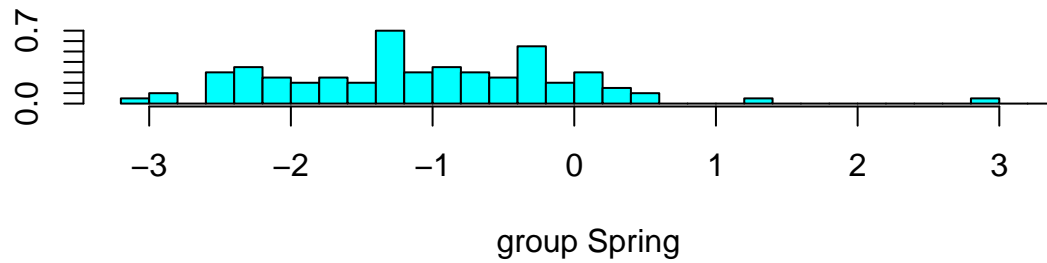


```
# for loop to regress PC1 onto log transformed data and pull out residuals
#Grahams Data
regress.known.noRed <- matrix(nrow = nrow(pca.known.noRed$x),
                             ncol = ncol(pca.known.noRed$x))
for(i in 1:ncol(pca.known.noRed$x)){
  regress.known.noRed[,i] <- lm(pca.known.noRed$x[,1]~known.lognoRed[,i])$residuals
}

#Sara Data
regress.unknown.noRed <- matrix(nrow = nrow(pca.unknown.noRed$x),
                                ncol = ncol(pca.unknown.noRed$x))
for(i in 1:ncol(pca.unknown.noRed$x)){
  regress.unknown.noRed[,i] <- lm(pca.unknown.noRed$x[,1]~unknown.lognoRed[,i])$residuals
}

# Create dataframe with data for Linear Discriminate Function Analysis
df.known.noRed <- data.frame(known.noRed$Season, regress.known.noRed)

# Run DFA
LDA.known.noRed <- lda(known.noRed.Season ~ ., data = df.known.noRed)
colnames(regress.unknown.noRed) <- c("X1", "X2", "X3", "X4", "X5", "X6",
                                     "X7", "X8", "X9", "X10", "X11", "X12",
                                     "X13", "X14", "X15", "X16", "X17", "X18")
regress.unknown.noRed <- as.data.frame(regress.unknown.noRed)
LDApredict.known.noRed <- predict(LDA.known.noRed)
plot(LDA.known.noRed)
```



```
str(LDApredict.known.noRed)
```

```
## List of 3
## $ class      : Factor w/ 2 levels "Spring","Winter": 2 2 2 2 1 1 2 2 2 2 ...
## $ posterior: num [1:258, 1:2] 0.208211 0.000604 0.03184 0.035521 0.715746 ...
## .. attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:258] "1" "2" "3" "4" ...
## .. ..$ : chr [1:2] "Spring" "Winter"
## $ x          : num [1:258, 1] 0.167 2.994 1.134 1.082 -0.884 ...
## .. attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:258] "1" "2" "3" "4" ...
## .. ..$ : chr "LD1"
```

```
# Evaluate Performance of LDA on Known Data
```

```
df.prediction <- as.data.frame(cbind(as.character(df.known.noRed$known.noRed.Season),
                                     as.character(LDApredict.known.noRed$class),
                                     LDApredict.known.noRed$posterior, LDApredict.known.noRed$x))
colnames(df.prediction)[1:2] <- c("Known", "Predicted")
```

```
incorrect <- NULL
for(i in 1:nrow(df.prediction)){
  if(df.prediction[i,1] != df.prediction[i,2]){
    incorrect <- as.data.frame(rbind(incorrect, df.prediction[i,]))
  }
}
incorrect
```

	Known	Predicted	Spring	Winter
## 5	Winter	Spring	0.715746478295315	0.284253521704685
## 6	Winter	Spring	0.539324272221415	0.460675727778585
## 37	Winter	Spring	0.680815830430658	0.319184169569342
## 46	Winter	Spring	0.557338879673612	0.442661120326388
## 52	Spring	Winter	0.406328758060528	0.593671241939472

## 54	Spring	Winter	0.323005235740977	0.676994764259023
## 60	Spring	Winter	0.382153303347702	0.617846696652298
## 63	Spring	Winter	0.00133651242893634	0.998663487571064
## 67	Spring	Winter	0.222031105243401	0.777968894756599
## 77	Spring	Winter	0.370903918124947	0.629096081875053
## 78	Spring	Winter	0.332252128665705	0.667747871334295
## 84	Spring	Winter	0.185394158899518	0.814605841100483
## 94	Spring	Winter	0.375055238423931	0.624944761576069
## 99	Spring	Winter	0.357380266153132	0.642619733846868
## 102	Spring	Winter	0.031738228220128	0.968261771779872
## 115	Spring	Winter	0.285542029149455	0.714457970850545
## 116	Spring	Winter	0.399638019806927	0.600361980193073
## 117	Spring	Winter	0.23488778633572	0.76511221366428
## 118	Spring	Winter	0.484120287316	0.515879712684
## 122	Spring	Winter	0.154766537483756	0.845233462516244
## 129	Spring	Winter	0.448792063006006	0.551207936993994
## 149	Winter	Spring	0.968255048968999	0.0317449510310011
## 150	Winter	Spring	0.719239632554336	0.280760367445664
## 152	Winter	Spring	0.874407331862631	0.125592668137369
## 159	Winter	Spring	0.592615952117972	0.407384047882028
## 172	Winter	Spring	0.594393267054408	0.405606732945592
## 176	Winter	Spring	0.930951607174555	0.0690483928254448
## 186	Winter	Spring	0.706465602062873	0.293534397937127
## 192	Winter	Spring	0.672942393135026	0.327057606864974
## 193	Winter	Spring	0.694381154962197	0.305618845037803
## 211	Winter	Spring	0.588104486232286	0.411895513767714
## 227	Winter	Spring	0.635504986779821	0.364495013220179
## 238	Winter	Spring	0.605504650636466	0.394495349363534
##	LD1			
## 5	-0.884102565132994			
## 6	-0.527753482806401			
## 37	-0.806891448242201			
## 46	-0.561602838456431			
## 52	-0.277983443554045			
## 54	-0.110085925177766			
## 60	-0.230868622223722			
## 63	2.62422382996081			
## 67	0.129023148036207			
## 77	-0.208569989885309			
## 78	-0.129618780447428			
## 84	0.234345875487061			
## 94	-0.216829651885938			
## 99	-0.181390575081886			
## 102	1.13599888743628			
## 115	-0.0276615144195706			
## 116	-0.265043045439139			
## 117	0.0950769514647321			
## 118	-0.42484491514902			
## 122	0.335538411733237			
## 129	-0.358765663489633			
## 149	-2.04471832463799			
## 150	-0.89212145993589			
## 152	-1.3573395291277			
## 159	-0.628803029003236			

```
## 172 -0.632230900878384
## 176 -1.6648594913396
## 186 -0.863080089121929
## 192 -0.790140333085543
## 193 -0.836279639196352
## 211 -0.620122597915824
## 227 -0.713077930079367
## 238 -0.653773570370474
```

```
(nrow(regress.known.noRed) - nrow(incorrect))/nrow(regress.known.noRed)*100
```

```
## [1] 87.2093
```

```
# Run DFA on my samples to determine which spawning groups they belong to
LDAPredictnoRed <- predict(LDA.known.noRed, newdata = regress.unknown.noRed)

#Create dataframe for plotting
df.finalunknown <- cbind(regress.unknown.noRed, LDAPredictnoRed$posterior, LDAPredictnoRed$x)

# Based on posterior probabilities determine which spawning group samples are from
df.finalunknown$Spawning <- NA
for(i in 1:nrow(df.finalunknown)){
  if(df.finalunknown$Spring[i] > 0.5){
    df.finalunknown$Spawning[i] <- "Spring"
  } else {
    df.finalunknown$Spawning[i] <- "Winter"
  }
}

df.finalunknown <- data.frame(df.finalunknown, unknown.noRed$Sample)
colnames(df.finalunknown)[19:23] <- c("LDAMorphSpring", "LDAMorphWinter", "LD1Morph", "LDAMorphAssign")
write.csv(df.finalunknown, "../data/Morphometrics/SamplingSpawningAssignment.csv")

colnames(df.known.noRed)[1] <- "SEASON"
#Subset data into training and validation datasets to test how well it works on known data
trainMorph <- sample(nrow(df.known.noRed), 0.7*nrow(df.known.noRed), replace = FALSE)
TrainMorphData <- df.known.noRed[trainMorph,]
ValidMorphData <- df.known.noRed[-trainMorph,]

#Create RFmodel on Training Data
RFMorphModel <- randomForest(SEASON ~ ., data = TrainMorphData, importance = TRUE)

#Run RFmodel on the Training Dataset
predTrainMorph <- predict(RFMorphModel, TrainMorphData, type = "class")
table(predTrainMorph, TrainMorphData$SEASON)

##
## predTrainMorph Spring Winter
##          Spring      63      0
##          Winter      0     117

#Run RFmodel on the Validation Dataset
predValidMorph <- predict(RFMorphModel, ValidMorphData, type = "class")
table(predValidMorph, ValidMorphData$SEASON)
```

```
##
## predValidMorph Spring Winter
##      Spring      29      6
##      Winter      8     35

# 86% accuracy on validation dataset

# Run Model from Training Data on full dataset
RFAssignMorph <- predict(RFMorphModel, df.known.noRed, type = "class")
RFMorphPred <- predict(RFMorphModel, df.known.noRed, type = "prob")
table(RFAssignMorph, df.known.noRed$SEASON)

##
## RFAssignMorph Spring Winter
##      Spring      91      6
##      Winter      9     152

# 96% on full dataset

df.RFMorphGraham <- as.data.frame(cbind(df.known.noRed, RFMorphPred, RFAssignMorph))

# Check accuracy of the models
RFMorphcorrect <- NULL
RFMorphincorrect <- NULL
for(i in 1:nrow(df.RFMorphGraham)){
  if(df.RFMorphGraham$SEASON[i] == df.RFMorphGraham$RFAssignMorph[i]){
    RFMorphcorrect <- c(RFMorphcorrect, as.character(df.RFMorphGraham$SEASON[i]))
  } else {
    RFMorphincorrect <- c(RFMorphincorrect, as.character(df.RFMorphGraham$SEASON[i]))
  }
}
100-(length(RFMorphincorrect)/length(RFMorphcorrect)*100)

## [1] 93.82716

# Use model to predict unknown samples
RFMorphSara <- predict(RFMorphModel, regress.unknown.noRed, type = "prob")
colnames(RFMorphSara) <- c("RFMorphSpring", "RFMorphWinter")
RFMorphAssignSara <- as.character(predict(RFMorphModel, regress.unknown.noRed))

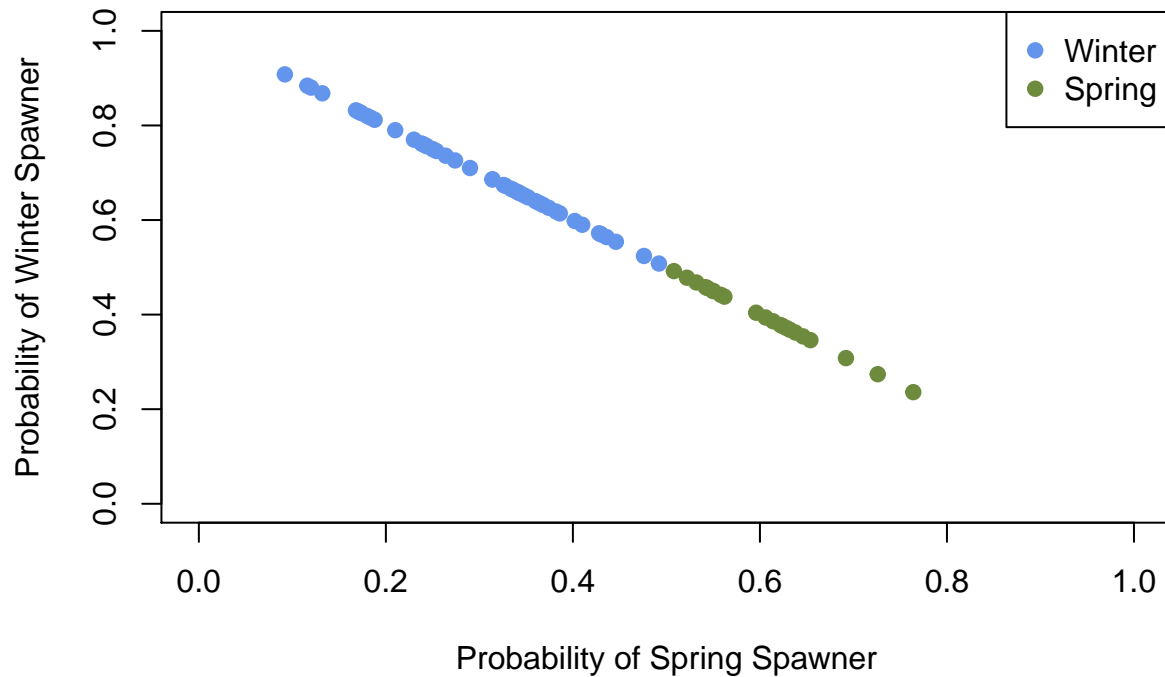
df.RFMorphSara <- as.data.frame(cbind(RFMorphSara, RFMorphAssignSara, unknown.noRed,
                                     df.finalunknown[19:23]))

write.csv(df.RFMorphSara, "../data/Morphometrics/MorphAnalyses.csv")

## Plot
plot(df.RFMorphSara$RFMorphSpring[df.RFMorphSara$RFMorphSpring > 0.5],
     df.RFMorphSara$RFMorphWinter[df.RFMorphSara$RFMorphWinter < 0.5],
     col = "darkolivegreen4", pch = 19, ylim = c(0,1), xlim = c(0,1),
     xlab = "Probability of Spring Spawner", ylab = "Probability of Winter Spawner",
     main = "Random Forest Posterior Probabilities")
points(df.RFMorphSara$RFMorphSpring[df.RFMorphSara$RFMorphSpring < 0.5],
       df.RFMorphSara$RFMorphWinter[df.RFMorphSara$RFMorphWinter > 0.5],
       col = "cornflowerblue", pch = 19, ylim = c(0,1), xlim = c(0,1),
       xlab = "Probability of Spring Spawner", ylab = "Probability of Winter Spawner")
legend("topright", legend = c("Winter", "Spring"),
```

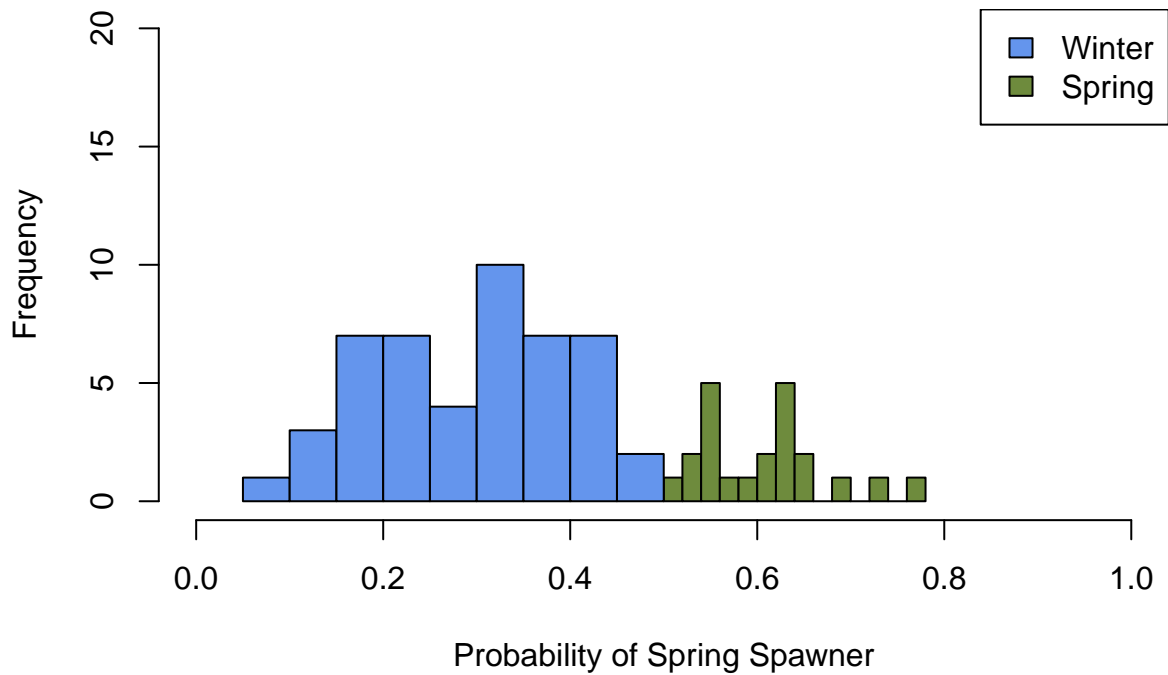
```
col = c("cornflowerblue", "darkolivegreen4"), pch = 19)
```

Random Forest Posterior Probabilities



```
hist(df.RFMorphSara$RFMorphSpring[df.RFMorphSara$RFMorphSpring > 0.5], breaks = 10,
     col = "darkolivegreen4", xlim = c(0,1), ylim = c(0, 20),
     xlab = "Probability of Spring Spawner",
     main = "RF Probability of Assignment to Spring Spawning Cod")
hist(df.RFMorphSara$RFMorphSpring[df.RFMorphSara$RFMorphSpring < 0.5], breaks = 10,
     col = "cornflowerblue", add = TRUE)
legend("topright", legend = c("Winter", "Spring"),
     fill = c("cornflowerblue", "darkolivegreen4"))
```

RF Probability of Assignment to Spring Spawning Cod



```
pdf(file = "../figures/PosteriorProbabilityOfAssignment.pdf")
par(mfrow = c(1,1))
plot(df.finalunknown$LDAMorphSpring[df.finalunknown$LDAMorphSpring > 0.5],
     df.finalunknown$LDAMorphWinter[df.finalunknown$LDAMorphWinter < 0.5],
     col = "darkolivegreen4", pch = 19, ylim = c(0, 1), xlim = c(0,1),
     xlab = "Probability of Spring Spawner", ylab = "Probability of Winter Spawner",
     main = "LDA Posterior Probabilities")
points(df.finalunknown$LDAMorphSpring[df.finalunknown$LDAMorphSpring < 0.5],
       df.finalunknown$LDAMorphWinter[df.finalunknown$LDAMorphWinter > 0.5],
       col = "cornflowerblue", pch = 19)
legend("topright", legend = c("Winter", "Spring"),
      col = c("cornflowerblue", "darkolivegreen4"), pch = 19)
#dev.off()

pdf(file = "../figures/SpawningAssignment.pdf")

hist(df.finalunknown$LDAMorphSpring[df.finalunknown$LDAMorphSpring > 0.5], breaks = 10,
     col = "darkolivegreen4", xlim = c(0,1), ylim = c(0, 20),
     xlab = "Probability of Spring Spawner",
     main = "LDA Probability of Assignment to Spring Spawning Cod")
hist(df.finalunknown$LDAMorphSpring[df.finalunknown$LDAMorphSpring < 0.5], breaks = 10,
     col = "cornflowerblue", add = TRUE)
legend("topright", legend = c("Winter", "Spring"),
      fill = c("cornflowerblue", "darkolivegreen4"))
#dev.off()
```