Lab 8 Discriminant Analysis (DA)

The goal of this lab is to become familiar with **Discriminant Analysis (DA)**. DA allows you to determine what variables are important in separating groups. It is an eigenanalysis that reduces dimensionality of multivariate data into new “canonical axes”. Canonical in statistics is a term describing the analysis of variables that are not directly observed (i.e., latent variables) that represent multiple variables that are directly observed.

# Set up R session

## Data

Today you will be using the palmerpenguins data set found in the palmerpenguins package in R. These data were collected by the Palmer Antarctica Long Term Ecological Research (LTER) site that is part of the LTER network. To access this data set, simply type:

# Note this install.packages line is commented out, so that the RMarkdown  
# document would knit. install.packages('palmerpenguins')  
library(palmerpenguins)

## Warning: package 'palmerpenguins' was built under R version 4.2.3

penguins <- penguins

The dataset contain data for 344 penguins. There are three different species of penguins (Adelie, Chinstrap, Gentoo) in this dataset, collected from three islands (Biscoe, Dream, Torgersen) in the Palmer Archipelago, Antarctica. To learn more about the data set:

`?`(penguins)

## Install and load packages

We will be using the following packages:

library(dplyr)  
library(MASS)  
library(broom)  
library(candisc)  
library(ade4)  
library(vegan)  
library(ggplot2)  
library(mvnormtest)  
library(MVN)

# Testing the assumptions of DA

## Homogeneity of within-group variance-covariance matrices

First, lets look at these groups in NMDS space. We’ll select just the bill, flipper, and body mass traits.

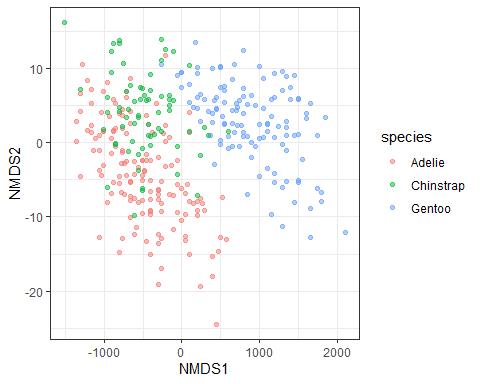
# Remove rows with NA values prior to calculating distance matrix. Remember  
# one assumption of DA is that no missing data are allowed.  
penguins\_noNA <- na.omit(penguins)  
# Calculate Euclidean distance matrix on morphological traits of penguins  
dis <- vegdist(penguins\_noNA[, 3:6], "euclidean")  
# Run NMDS on the Euclidean distance matrix made above  
penguin\_nmds <- metaMDS(dis)

## Warning in metaMDS(dis): stress is (nearly) zero: you may have insufficient  
## data

Now lets plot but using ggplot2 this time.

First build a data frame with the first two axes and the groups (i.e., species names):

NMDS = data.frame(NMDS1 = penguin\_nmds$point[, 1], NMDS2 = penguin\_nmds$point[, 2],  
 species = penguins\_noNA$species)  
  
ggplot(NMDS, aes(x = NMDS1, y = NMDS2)) + geom\_point(data = NMDS, aes(x = NMDS1,  
 y = NMDS2, color = species), alpha = 0.5) + viridis::scale\_fill\_viridis() + theme\_bw()



Next, we will use the Fligner-Killeen test of homogeneity of variances. This test has been shown to be robust to departures from normality (which most of our data does to some extent)

`?`(fligner.test)

You want to test the variance of each variable (n=4) across all three groups (i.e., species). **Remember, you don’t want there to be significant differences**

fligner.test(penguins\_noNA$bill\_length\_mm, penguins\_noNA$species)  
fligner.test(penguins\_noNA$bill\_depth\_mm, penguins\_noNA$species)  
fligner.test(penguins\_noNA$flipper\_length\_mm, penguins\_noNA$species)  
fligner.test(penguins\_noNA$body\_mass\_g, penguins\_noNA$species)

We will also use a Multivariate test of homogeneity of variance called ‘betadisper’ in the vegan package. **Remember, you don’t want there to be significant differences**

`?`(betadisper)

Calculate multivariate dispersions:

# Euclidean distances between samples  
dis <- vegdist(penguins\_noNA[, 3:6], "euclidean")  
  
# groups are the three different species  
groups <- penguins\_noNA$species  
  
# multivariate dispersions  
MVdisp <- betadisper(dis, groups)  
  
  
# Perform parametric test  
disp\_aov <- anova(MVdisp)  
  
# Tukey's Honest Significant Differences  
MVdisp.HSD <- TukeyHSD(MVdisp)  
MVdisp.HSD  
  
## non-parametric test: Permutation test for F  
perm\_MVdisp <- permutest(MVdisp, permutations = 99, pairwise = TRUE)  
perm\_MVdisp

## Question 1: Should you transform the penguin variables based on the tests for homogeneity of variance? (10 pts)

If you think yes, use the code below. If you think no, skip ahead.

log <- cbind.data.frame(apply(penguins\_noNA[, 3:6] + 1, 2, log), penguins\_noNA$species)  
names(log)[5] <- "Species"

Re-run the Fligner-Killeen test on the transformed data:

fligner.test(log$bill\_length\_mm, log$Species)  
fligner.test(log$bill\_depth\_mm, log$Species)  
fligner.test(log$flipper\_length\_mm, log$Species)  
fligner.test(log$body\_mass\_g, log$Species)

Re-run multivariate homogeneity of variance test

# Euclidean distances between samples  
log\_dis <- vegdist(log[, 1:4], "euclidean")  
  
  
# groups are the three different species  
groups <- penguins\_noNA$species  
  
## Calculate multivariate dispersions  
log\_MVdisp <- betadisper(log\_dis, groups)  
  
  
# Perform parametric test  
disp\_aov <- anova(log\_MVdisp)  
  
  
# Tukey's Honest Significant Differences  
MVdisp.HSD <- TukeyHSD(log\_MVdisp)  
MVdisp.HSD  
  
# non-parametric test: Permutation test for F  
perm\_MVdisp <- permutest(log\_MVdisp, permutations = 99, pairwise = TRUE)  
perm\_MVdisp

## Question 2: Based on the results of the tests for the DA assumption of equality of group dispersions, would you consider the interpretation of any results of a DA with these data to be descriptive or inferential? (10 pts)

## Multivariate Normality

To look for multivariate normality, let’s look at the distribution of each variable for each group and run (overly conservative) mulitvariate normality tests for both the log and untransformed penguin data.

# filter for each species: e.g., setosa below  
  
Adelie <- dplyr::filter(penguins\_noNA, species == "Adelie")  
Adelielog <- dplyr::filter(log, Species == "Adelie")  
  
  
# untransformed  
  
mshapiro.test(t(Adelie[, 3:6]))  
  
mvn(Adelie[, 3:6], mvnTest = "mardia")  
  
# log  
  
mshapiro.test(t(Adelielog[, 1:4]))  
  
mvn(Adelielog[, 1:4], mvnTest = "mardia")

## Question 3: Write some code to assess multivariate normality for the other two species (i.e., Chinstrap and Gentoo) of penguins in the dataset. (10 pts)

## Multicolinearity

To test for multicolinearity, you should look at all pairwise correlations. Remember, correlations > 0.7 are often considered problematic.

cor(penguins\_noNA[, 3:6])

## bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## bill\_length\_mm 1.0000000 -0.2286256 0.6530956 0.5894511  
## bill\_depth\_mm -0.2286256 1.0000000 -0.5777917 -0.4720157  
## flipper\_length\_mm 0.6530956 -0.5777917 1.0000000 0.8729789  
## body\_mass\_g 0.5894511 -0.4720157 0.8729789 1.0000000

## Question 4: We have some co-linearity. Based on the results from other assumption checks, which variable might be a good one to leave out of this analysis? Perform the remainder of the lab without that variable. Feel free to check with me to make sure you are on the right track before proceeding. (10 pts)

## Outliers

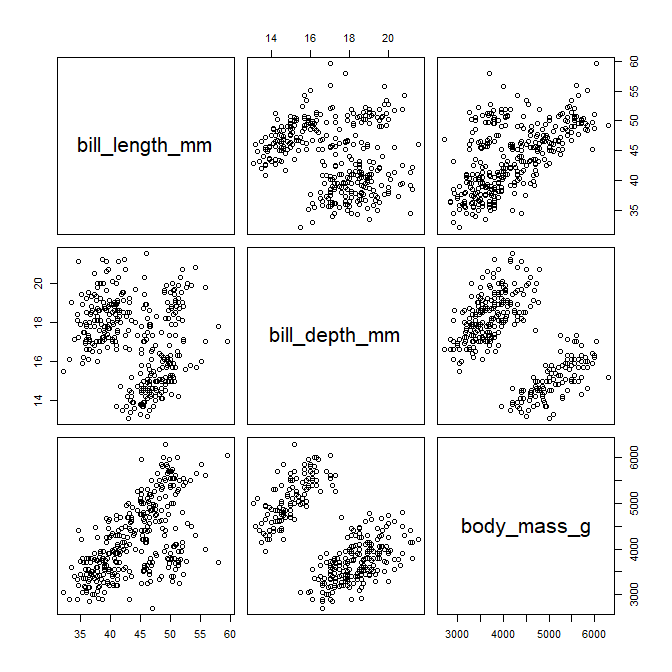
Outliers can have a large influence on canonical axes in DA. You are going to take a multivariate approach to identifying outliers for the untransformed data (we did this in lab 2). Note we are proceeding with the untransformed data because transformation didn’t help to meet any of the assumptions of DA thus far, but we should keep in mind that we might want to used the log transformed data if we have trouble with other assumptions.

# Calculate a within-group distance matrix: Adelie  
  
eucDist <- vegdist(penguins\_noNA[penguins\_noNA$species == "Adelie", c(3:4, 6)], "euclidean")  
  
# Calculate the average distance of each sample to all other samples (i.e.  
# column average) and turn the means into z-scores:  
  
multOut <- scale(colMeans(as.matrix(eucDist)))  
  
  
# Now look at a histogram of these data to identify samples that are > 3 sd  
# from the mean:  
  
hist(multOut)  
  
  
# and get the number of those samples:  
  
Adelie\_out <- multOut[multOut > 3, ]  
Adelie\_out  
  
# Repeat for the other two groups (i.e.,species):  
  
# Chinstrap  
eucDist <- vegdist(penguins\_noNA[penguins\_noNA$species == "Chinstrap", c(3:4, 6)],  
 "euclidean")  
multOut <- scale(colMeans(as.matrix(eucDist)))  
hist(multOut)  
Chinstrap\_out <- multOut[multOut > 3, ]  
Chinstrap\_out  
  
# Gentoo  
  
eucDist <- vegdist(penguins\_noNA[penguins\_noNA$species == "Gentoo", c(3:4, 6)], "euclidean")  
multOut <- scale(colMeans(as.matrix(eucDist)))  
hist(multOut)  
Gentoo\_out <- multOut[multOut > 3, ]  
  
# Finally, make a vector of the outliers to pull out of the data set later:  
  
Outliers <- c(Adelie\_out, Chinstrap\_out, Gentoo\_out)  
Outliers  
  
# Remove the outliers from the data set for the analysis moving forward.  
penguins\_noNA <- penguins\_noNA[-c(96, 104, 38, 39, 18, 40), ]

## Linearity

Next we test for linear relationship between variables. This is key to making sure that the variables change in a linear fashion along underlying gradients (i.e. canonical axes):

pairs(penguins\_noNA[, c(3:4, 6)])



Believe it or not, those pass as linear relationships!

# Discriminant analysis

You will use the LDA function in the *MASS* package to conduct DA and will also use the candisc function in the package *candisc*.

`?`(lda)  
  
`?`(candisc)

Based on the assumption checks we performed, we will run the analysis on the untransformed data and exclude the body mass variable. We are going to split our data set into “training” data for fitting the DA and “testing” data for validating the DA. We randomly select 75 samples from the penguins data set for the training data. Note that we set the seed for the random number generator below to ensure that we all get comparable results from a consistent randomly selected set of training data.

# Set the seed for the random number generator  
set.seed(11)  
# Create a vector whose length is one half of the number of rows in the  
# penguins data set called train that will indicate the rows randomly selected  
# for the training data.  
train <- sample(seq(1:dim(penguins\_noNA)[1]), round(dim(penguins\_noNA)[1] \* 0.66))

## Question 5: What does the *dim* function do in the code chunk above? Why do we have to use the round function? (10 pts)

# Determine the counts of each species in the training data  
priorfreq <- table(penguins\_noNA$species[train])  
# Convert those counts to probabilites that sum to one  
prior <- (as.vector(priorfreq)/round(dim(penguins\_noNA)[1] \* 0.66))

Next run the DA for the training data:

penguinsLDA <- lda(species ~ bill\_length\_mm + bill\_depth\_mm + body\_mass\_g, penguins\_noNA,  
 prior = prior, subset = train)

**Pro Tip: if you wanted to use all variables in a dataset, then “Species ~ .” would be the model and the “dot” stands for all of the variables (so you don’t have to type them all)**

We will now assess what the results tell us about the number of meaningful canonical axes and the absolute contribution of each variable to those axes.

# Assessing and interpreting canonical axes

## Relative % Criterion

The “proportion of trace” from the lda output is the **Relative % criterion**.

penguinsLDA

## Call:  
## lda(species ~ bill\_length\_mm + bill\_depth\_mm + body\_mass\_g, data = penguins\_noNA,   
## prior = prior, subset = train)  
##   
## Prior probabilities of groups:  
## Adelie Chinstrap Gentoo   
## 0.4500000 0.1954545 0.3545455   
##   
## Group means:  
## bill\_length\_mm bill\_depth\_mm body\_mass\_g  
## Adelie 38.71717 18.28081 3680.556  
## Chinstrap 48.91163 18.54884 3735.465  
## Gentoo 47.53974 14.95385 5108.333  
##   
## Coefficients of linear discriminants:  
## LD1 LD2  
## bill\_length\_mm 0.11273857 -0.407717650  
## bill\_depth\_mm -1.00433350 -0.005141873  
## body\_mass\_g 0.00194526 0.001892466  
##   
## Proportion of trace:  
## LD1 LD2   
## 0.8259 0.1741

## Question 6: According to the relative percent criterion, which axis or axes are meaningful for separating the species? (10 pts)

## Canonical Correlation Criterion

Here, you want to test the correlation between the canonical scores and the grouping variable. First you need to predict the membership (i.e. calculate canonical scores) of each sample in the training set. You will use the predict.lda function from the *MASS* package:

`?`(predict.lda)

penguinsLDA.p <- predict(penguinsLDA, penguins\_noNA[train, ])

You want to test the canonical scores (penguinsLDA.P$x) with the grouping variable (i.e. species):

corTest <- lm(penguinsLDA.p$x ~ penguins$species[train])  
summary(corTest)

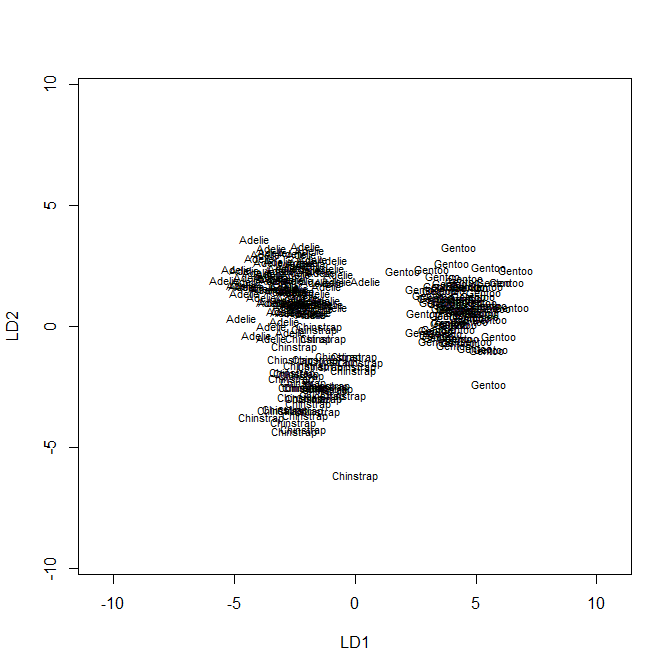
## Response LD1 :  
##   
## Call:  
## lm(formula = LD1 ~ penguins$species[train])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.8312 -0.5911 0.0347 0.7461 8.1378   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.5583 0.1602 -15.965 < 2e-16 \*\*\*  
## penguins$species[train]Chinstrap 0.9457 0.3065 3.086 0.00229 \*\*   
## penguins$species[train]Gentoo 6.5047 0.2402 27.080 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.61 on 217 degrees of freedom  
## Multiple R-squared: 0.7822, Adjusted R-squared: 0.7802   
## F-statistic: 389.7 on 2 and 217 DF, p-value: < 2.2e-16  
##   
##   
## Response LD2 :  
##   
## Call:  
## lm(formula = LD2 ~ penguins$species[train])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.5377 -0.6779 0.0456 0.7245 2.7484   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.0819 0.1121 9.648 < 2e-16 \*\*\*  
## penguins$species[train]Chinstrap -3.9929 0.2145 -18.618 < 2e-16 \*\*\*  
## penguins$species[train]Gentoo -1.0653 0.1681 -6.338 1.33e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.127 on 217 degrees of freedom  
## Multiple R-squared: 0.615, Adjusted R-squared: 0.6115   
## F-statistic: 173.3 on 2 and 217 DF, p-value: < 2.2e-16

## Question 7: How does the correlation and significance look for each canonical axis? (10 pts)

## Plot canonical scores and axes

Now visually asses the discrimination of the three species of penguins by the canonical axes:

plot(penguinsLDA, xlim = c(-11, 11), ylim = c(-6, 6))



## Classification accuracy

You now want to measure how well the axes discriminate. The higher the correct classification rate, the greater degree of group discrimination achieved by the canonical axes. We will classify the training data first. Create a classification matrix:

ct <- table(penguins\_noNA[train, ]$species, predict(penguinsLDA, penguins\_noNA[train,  
 ])$class)  
  
# Change to a table of proportions:  
  
pct <- prop.table(ct)  
pct

##   
## Adelie Chinstrap Gentoo  
## Adelie 0.45000000 0.00000000 0.00000000  
## Chinstrap 0.01363636 0.18181818 0.00000000  
## Gentoo 0.00000000 0.00000000 0.35454545

# Calculate classification rate by summing the diagonal:  
sum(diag(pct))

## [1] 0.9863636

## Interpreting canonical axes (raw coefficients, standardized weights, and structure coefficients)

Unfortunately, lda doesn’t provide an easy way to calculate the standardized weights, and structure coefficients. For this, you will use candisc in the *candisc* package.

`?`(candisc)

For candisc, you build a linear model using lm:

penguinsmod <- lm(cbind(bill\_length\_mm, bill\_depth\_mm, body\_mass\_g) ~ species, data = penguins\_noNA[train,  
 ])

And run the candisc function:

penguinscan <- candisc(penguinsmod, data = penguins\_noNA[train, ])

Next pull the raw coefficients, standardized weights, and structure coefficients from the output:

penguinscan$coeffs.raw

## Can1 Can2  
## bill\_length\_mm 0.11273857 0.407717650  
## bill\_depth\_mm -1.00433350 0.005141873  
## body\_mass\_g 0.00194526 -0.001892466

penguinscan$coeffs.std

## Can1 Can2  
## bill\_length\_mm 0.3532702 1.277597455  
## bill\_depth\_mm -1.1549278 0.005912869  
## body\_mass\_g 0.9067458 -0.882136932

penguinscan$structure

## Can1 Can2  
## bill\_length\_mm 0.5865915 0.73466391  
## bill\_depth\_mm -0.8436517 0.16793812  
## body\_mass\_g 0.8589550 -0.08022041

Note that the raw coefficients are the same as the lda output:

penguinsLDA$scaling

## LD1 LD2  
## bill\_length\_mm 0.11273857 -0.407717650  
## bill\_depth\_mm -1.00433350 -0.005141873  
## body\_mass\_g 0.00194526 0.001892466

*notice that the signs can sometimes be flipped but coefficients are the same*

## Question 8: Looking at the structure coefficients, how would you “define” the first canonical axis? (10 pts)

# Validating canonical axes

A DA is only as good as its ability to classify new data correctly. Here, you are going to use the “testing” data from your split sample to see how well the canonical axes predict group membership of “new” samples. You are going to use the predict.lda function again. This time with the testing data (i.e., penguins\_noNA[-train, ])

penguinsLDAnew <- predict(penguinsLDA, penguins\_noNA[-train, ])

Next, make a classification table and calculate the classification rate:

ynew.table <- table(penguins\_noNA[-train, ]$species, penguinsLDAnew$class)  
ynew.table

##   
## Adelie Chinstrap Gentoo  
## Adelie 47 0 0  
## Chinstrap 0 25 0  
## Gentoo 0 0 41

sum(diag(prop.table(ynew.table)))

## [1] 1

## Question 9: How did the canonical axes handle the new data? (10 pts)

# MANOVA, *the other side of the coin of DA*

In MANOVA, we are interested if groups differ in their measured variables. In DA, we are interested in a linear combination of variables that maximize differences between groups. In this case, MANOVA can provide a test of whether or not the groups are different.

`?`(manova)

Y <- as.matrix(penguins\_noNA[, c(3:4, 6)])  
Sp <- factor(penguins\_noNA$species)  
  
fit <- manova(Y ~ Sp)  
summary(fit, test = "Wilks")

## Post-hoc tests:

YAdelie <- as.matrix(penguins\_noNA[penguins\_noNA$species == "Adelie", c(3:4, 6)])  
YChinstrap <- as.matrix(penguins\_noNA[penguins\_noNA$species == "Chinstrap", c(3:4,  
 6)])  
YGentoo <- as.matrix(penguins\_noNA[penguins\_noNA$species == "Gentoo", c(3:4, 6)])  
Sp <- factor(penguins\_noNA$species)  
  
fit1 <- manova(rbind(YAdelie, YChinstrap) ~ c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap",  
 dim(YChinstrap)[1])))  
summary(fit1, test = "Hotelling-Lawley")

## Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 1  
## Residuals 212  
## Hotelling-Lawley  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3.7556  
## Residuals   
## approx F  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 262.89  
## Residuals   
## num Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3  
## Residuals   
## den Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 210  
## Residuals   
## Pr(>F)  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) < 2.2e-16  
## Residuals   
##   
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

fit2 <- manova(rbind(YAdelie, YGentoo) ~ c(rep("Adelie", dim(YAdelie)[1]), rep("Gentoo",  
 dim(YGentoo)[1])))  
summary(fit1, test = "Hotelling-Lawley")

## Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 1  
## Residuals 212  
## Hotelling-Lawley  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3.7556  
## Residuals   
## approx F  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 262.89  
## Residuals   
## num Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3  
## Residuals   
## den Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 210  
## Residuals   
## Pr(>F)  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) < 2.2e-16  
## Residuals   
##   
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

fit3 <- manova(rbind(YChinstrap, YGentoo) ~ c(rep("Chinstrap", dim(YChinstrap)[1]),  
 rep("Gentoo", dim(YGentoo)[1])))  
summary(fit1, test = "Hotelling-Lawley")

## Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 1  
## Residuals 212  
## Hotelling-Lawley  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3.7556  
## Residuals   
## approx F  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 262.89  
## Residuals   
## num Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 3  
## Residuals   
## den Df  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) 210  
## Residuals   
## Pr(>F)  
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) < 2.2e-16  
## Residuals   
##   
## c(rep("Adelie", dim(YAdelie)[1]), rep("Chinstrap", dim(YChinstrap)[1])) \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Question 10: Consider your own dataset and whether or not DA would be appropriate and if the data would meet the DA assumptions. (10 pts)