Lab 4

Discriminant Analysis
Multivariate Analysis of Variance

Just like principal component analysis, discriminant analysis is a rotation-based technique and can simply be used to visualize your data (literally to look at your data from different angles to reduce complexity). Now, rather than maximizing the total variance explained as the PCA criteria, discriminant analysis *maximizes the total variance between groups*.

Therefore, you need groups that are defined by a class variable. Another question that discriminant analysis answers is: are there significant differences between groups? And thirdly you can ask: to which group does a new observation belong? These last two questions are often asked in the fields of taxonomy, paleontology, or anthropology (e.g. did we find a new species? To which ancestor does that jaw-bone belong?). Discriminant analysis is a poplar analytical tool in these fields.

One important note, though: You need an *a priori* classification system to use discriminant analysis. This analysis does not help you to define classes in the first place (this is done by different techniques covered in the next lab).

4.1. Discriminant Analysis in R

- Let's start with a botanical classification example. Download the dataset "iris.csv" from the website, import it into R and check whether the import was successful by opening the file with the fix() command. The dataset contains 150 flower measurements (petal and sepal length and width) of three iris species (50 specimens from each species).

```
iris=read.csv("iris.csv")
fix(iris)
attach(iris)
```

- To run a discriminant analayis we have to call an optional R package called (MASS). This may already installed on the machines in the lab (check with the command below). If not, select from the main menu: Packages > Install Packages > Select a download site > Select the package: MASS.

```
library(MASS)
```

- The syntax for the linear discriminant analysis is lda(classvariable~., dataset). The dot means "all other variables", but you could list them individually as well lda(classvariable~var1+var2+…+varN, dataset). The results are written into an output file that we can subsequently query:

```
out1=lda(Species~., iris)
```

- Let's first get the discriminant analysis scores, i.e. the new coordinate position of points after the matrix rotation. This is exactly equivalent to PCA. The second command plots the first two discriminant functions (equivalent to PC1, PC2) and colors the points by species. The option asp=1 forces the two axes to have the same scale, so we can better see which discriminant function is most effective in separating the groups:

```
scores=predict(out1, iris)$x
plot(scores, col=rainbow(3)[iris$Species], asp=1)
```

- Let's call the output file (simply execute the name "out1"), and you can confirm that it's all up to discriminant function one. "Proportion of trace" means variance explained.

- Here is an alternative r-package for discriminant analysis that allows for a nice biplot with group centroids and labels, custom coloring and scaling of vectors, gives you loadings, but that does not work for classifying new observations, which we cover in the next section:

```
library(candisc)
x=lm(cbind(PetLength,SepLength,PetWidth,SepWidth)~Species,data=iris)
out2=candisc(x, term="Species")
summary(out2)
out2$structure     # discriminant function loadings
plot(out2, which=c(1,2), scale=8, var.col="#777777", var.lwd=1,
     col=c("red","green","blue"))
```

4.2. Classifying new observations

- Let’s say we collected 6 new iris specimen that need to be classified. We take our measurements of sepals and petals, arrange the data in the same format (but leave the species field blank) and then carry out the classification as above. This assumes that you still have R open and that you are finished with the previous exercise, which gives us the “out1” file:

```
unknown=read.csv("iris_unknown.csv")
fix(unknown)
predict(out1, unknown)$class
```

- We can also add these points to the previous discriminant plot. If you have the graph window from the previous exercise still open you can skip the first line:

```
plot(scores, asp=1, col=rainbow(3)[iris$Species])
out1p=predict(out1, unknown)
scores_unknown=out1p$x
points(scores_unknown, pch=19)
```

4.3. Testing for significant differences between groups

To address the question: “do we really have different species?” we can carry out a Multivariate Analysis of Variance (MANOVA). This is in fact mathematically exactly the same thing as discriminant analysis, above. We are just asking a different question

- First we have to do some data preparation for the manova function to work properly. This involves splitting the dataset into the class variable and into measurements, and then defining those as a matrix of numbers, and the species variable also as a factor (or class) variable:

```
species=(as.matrix(iris[,5])
measurements=as.matrix(iris[,1:4])
```

- We run the MANOVA, and write the result into an output file that we can subsequently query. The subsequent queries ask for the Wilk’s MANOVA test statistic, and you can also query the result of univariate ANOVAs for each of the measurements separately.

```
out2=manova(measurements~species)
summary(out2, test="Wilks")
summary.aov(out2)
```

- An important assumption for MANOVA just as for ANOVA is normality and homogeneity of variances. You therefore first have to check each individual measurement for normality and homogeneity, e.g. by making boxplots or plotting ANOVA residuals for each variable:

```
boxplot(SepLength~Species)
plot(residuals(lm(SepLength~Species)))
```

4.4. Discriminant analysis – climate change projections

For the next example, we analyze spatial data with discriminant analysis. We can recycle the previous dataset AB_Climate.csv because it also has a class variable: ECOSYS. The ecosystems have already been classified, so that's our predetermined class variable. Now what we want to do is make ecosystem predictions for a new set of observations: climate change predictions for the 2020s, 2050s, and 2080s from the Canadian atmosphere-ocean coupled general circulation model CGCM2-B2. I would rate them as middle of the road projection of for Alberta.

- Download the zip file of climate data and additional ecosystem information. We first look at reference climate for the 1961-1990 normal period "AB_Climate.csv":

```
ab6190=read.csv("AB_Climate.csv")
head(ab6190)
attach(ab6190)
```

- The "AB_Climate.csv" file includes an abbreviated ecosystem code, ECOSYS, but we also want full names and a color code for nice maps. The "AB_Ecosystems.csv" file contains that information, and we import names and colors into separate variables. Note that the hexadecimal color codes need to be converted to a character variable in order to work (i.e. they are enclosed by quotes if you query "ecol")

```
ecosys=read.csv("AB_Ecosystems.csv")
head(ecosys)
ecol=as.character(ecosys$ECOL)
ename=ecosys$ENAME
```

- OK, let's display the mapped ecosystems from the file "AB_Climate.csv" with the imported color scheme. We can do it with the plot command from the previous lab. Adjust the size of the window to get the aspect ratio of Alberta correct if necessary:

```
plot(Y~X, pch=21, col=ecol[ECOSYS])
```

- Next comes the actual analysis. We want to predict ecosystem classes from the 1961-1990 reference climate, and then use this relationship to predict what ecosystems might be supported under future climate projections for Alberta. First, we build the discriminant functions as in the example with the iris data.

```
library(MASS)
out1=lda(ECOSYS~MAT+MWMT+MCMT+MAP+MSP+AHM, ab6190)
pred6190=predict(out1, ab6190)$class
```

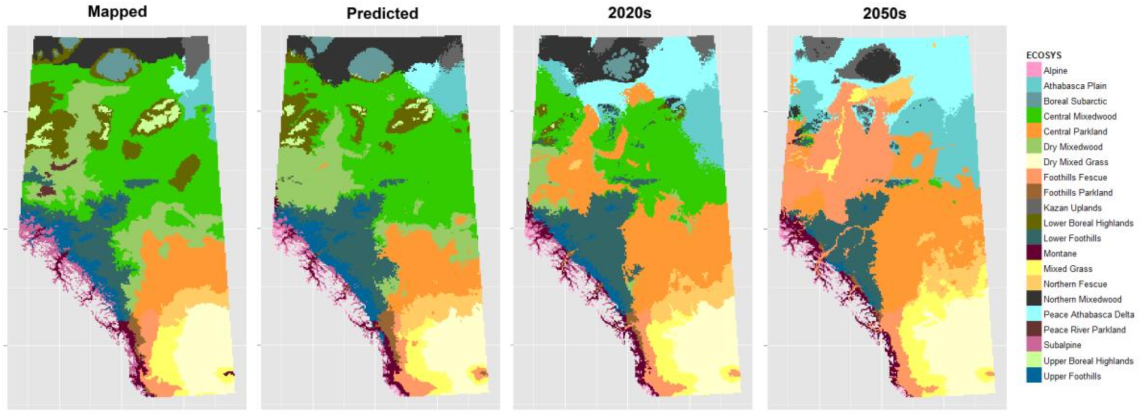
- Now plot the predicted ecosystems for Alberta using the 1961-1990 climate reference data. This is a mapped ecosystems versus modeled ecosystems comparison. The model is just based on climate and it will therefore not be a perfect fit to the mapped delineations.

```
plot(Y~X, pch=21, col=ecol[pred6190])
```

- Now the interesting part: Let's do the same predictions for the 2020s with the code below, and make adjustments to the code below for the 2050s, and 2080s climate projections yourself.

```
ab2020 = read.csv("AB_Climate_2020s.csv")
pred2020 = predict(out1, ab2020)$class
plot(Y~X, pch=21, col=ecol[pred2020])
```

4.5. Better graphics with ggplot2 (if you have time)



The same can be done with an advanced graphics package for display, that we will explore later more thoroughly:

```
ab6190=read.csv("AB_Climate.csv")
head(ab6190)
attach(ab6190)
```

- Read in the Alberta Ecossytems file with includes the color code and ecosystems name for the legend

```
ecosys=read.csv("AB_Ecosystems.csv")
head(ecosys)
ecol=as.character(ecosys$ECOL)
bcol=as.character(ecosys$BCOL)
ename=ecosys$ENAME
bname=ecosys$BNAME
```

- Install the ggplot2 package to plot Alberta ecosystems

```
install.packages('ggplot2')
library(ggplot2)
ggplot(aes(x=X, y=Y, fill=ECOSYS), data=ab6190) +
scale_fill_manual(values=ecol,  labels=ename) +
geom_tile()
```

- Run discriminant analysis

```
library(MASS)
out1=lda(ECOSYS~MAT+MWMT+MCMT+MAP+MSP+AHM, ab6190)
ecosys6190=predict(out1, ab6190)$class
# GGPLOT NEEDS THE DATA IN A DATAFRAME FORMAT!
ecosys6190.df=as.data.frame(ecosys6190)
```

- Check if ecosystems got dropped so that you can adjust the color code by excluding these colors/ecosystems

```
a=levels(ecosys$ECOSYS)
b=levels(droplevels(ecosys6190))
setdiff(a,b)
```

- Now plot the predicted ecosystems for Alberta using the 6190 climate normal data

```
ggplot(aes(x=X, y=Y, fill=ecosys6190), data=ecosys6190.df) +
scale_fill_manual(values=ecol[-c(18)], labels=ename[-c(18)]) +
geom_tile()
```

- How about future climate scenarios, e.g. 2050s?

```
ab2050 = read.csv("AB_Climate_2050s.csv")
ecosys2050 = predict(out1, ab2050)$class
ecosys2050.df = as.data.frame(ecosys2050)

a = levels(ecosys$ECOSYS)
b = levels(droplevels(ecosys2050))
setdiff(a,b)

ggplot(aes(x = X, y = Y, fill = ecosys2050), data =
ecosys2050.df) +
scale_fill_manual(values = ecol[-c(11,20)],  labels = ename[-
c(11,20)]) +
geom_tile()
```

4.6. Spatial maps and plots of discriminant functions (do it yourself)

Just as we did for principal components, you can create plots of your disciminant functions in the rotated coordinate system. To get the component scores for each datapoint manually, we need to use a slightly convoluted code. Oddly, :

```
library(candisc)
x=lm(cbind(MAT,MWMT,MCMT,TD,MAP,AHM,SHM,DD0,DD5,NFFD,FFP,
     PAS)~ECOSYS,data=ab)
out3=candisc(x, term="ECOSYS")
scores=out3$scores
```

- Recycle code from the lab on PCA to create maps of and plots of discriminant functions (i.e. create a map of DF1 and DF2 and a plot of DF2 over DF1 instead of PC1 and PC2).
- How are these component scores different from the PCA maps of the previous lab? What questions could you answer based on these maps (as opposed to PCA maps and plots)?

4.7. Discriminant Analysis in SAS (if you have time)

- Why don't we see if you can get discriminant analysis to work in SAS by yourself and get the same results as in R. Can you apply what you previously learned to any software packge? I provide some code to get you started, and it's your job to explain to me (or the TA) what the output means. Can you determine what the loadings are?

```
PROC CANDISC DATA=iris OUT=scores;
CLASS Species;
VAR SepLength SepWidth PetLength PetWidth;
RUN;

PROC GPLOT DATA=scores;
PLOT CAN1*CAN2=Species;
RUN;
```

- To make predictions, you have to use a different discriminant analysis procedure in SAS and simply append the unknown observations at the end (copy and paste them into the CSV in Excel, leaving the species field blank). Open the resulting “predicted” file and check the predictions in the last column. Notice that the predictions are also made for the training data and that they are not always correct. Why? Note that it there are three columns that provide probability values reflecting the confidence of the classification. Can you interpret those?

```
PROC DISCRIM DATA=iris_all OUT=predicted;
CLASS Species;
VAR SepLength SepWidth PetLength PetWidth;
RUN;
```

4.8. Ecosystem projections in SAS (if you have time)

We can use the previous dataset AB_Climate.csv because it also has a class variable: ECOSYS. The ecosystems have already been classified, so that’s our predetermined class variable. Now what we want to do is make ecosystem predictions for a new set of observations: climate change predictions for the 2020s, 2050s, and 2080s from the Candian atmosphere-ocean coupled general circulation model CGCM2-B2. I would rate them as middle of the road projection of for Alberta.

- Import the file AB_Climate from Lab 1 and assign the name “CURRENT” to the imported dataset. Unzip the file AB_Future_Climate.zip and import any one of those three files to SAS and assign it the name “FUTURE”.
- Open the files in the SAS editor to confirm that they have imported correctly.
- In SAS the training dataset (CURRENT) and the unknown dataset to be classified (FUTURE) have to be in a single file. SAS can distinguish between the training data and the unknown datapoints to be predicted by the by the fact that they have missing values for the class variable.
- `PROC APPEND` joins the two datasets, and `PROC DISCRIM` makes the predictions and calls this variable _INTO_. It also makes predictions for the training dataset, which gives you a sense of the accuracy of predictions. I used only a subset of the climate variables to make the predictions below (runs faster and gives more reliable projections):

```
PROC APPEND BASE=CURRENT DATA=FUTURE; RUN;

PROC DISCRIM DATA=CURRENT OUT=PREDICTED;
CLASS ECOSYS;
VAR MAT MCMT MWMT TD MAP MSP AHM SHM;
RUN;
```

- Open the output file `PREDICTED`. On the far right are the predictions. Now we have to separate the training data and the future data again and export them for display in ArcMap.

```
DATA MAPPED_VS_MODELED;
SET PREDICTED;
IF ECOSYS = "" THEN DELETE;
KEEP X Y ECOSYS _INTO_;
RUN;

DATA FUTURE;
SET PREDICTED;
IF ECOSYS NE "" THEN DELETE;
KEEP X Y _INTO_;
RUN;
```