Biol 206/306 – Advanced Biostatistics
Lab 6 – Multivariate ANOVA and Discriminant Function Analysis

By Philip J. Bergmann

0. Laboratory Objectives

1. Learn when it is appropriate to use Multivariate Analysis of Variance (MANOVA)
2. Learn about the assumptions and interpretation of a MANOVA
3. Learn how to do a MANOVA in R
4. Learn about Discriminant Function Analysis (DFA) and when to use it.
5. Learn to do a DFA in R

1. The Multivariate Analysis of Variance Background

Today we start to explore multivariate statistics. Multivariate techniques have multiple response variables, hence the name. Multivariate Analysis of Variance (MANOVA) is the first such technique we will learn. Consider the typical case of an ANOVA where you have one or more categorical explanatory variables with two or more treatment levels each. However, you now also have multiple response variables that you wish to consider simultaneously instead of separately. Instead of a null hypothesis that group means are not significantly different from one another, you are testing a null hypothesis that **group centroids** are not significantly different form one another. Group centroids are simply multivariate means – means in two or more dimensions.

MANOVA works by finding a linear combination of the multiple response variables that maximizes between group differences relative to within group differences, hence maximizing discriminating power between the groups. To do this, MANOVA takes the variation present in all of the response variables and identifies the axis of the multivariate data with the most between group variation. This is termed the first discriminant function (DF-1). Subsequent DFs are independent of the first and one another. Each DF has a linear model associated with it that gives weights of how strongly each original response variable contributes to the DF, as follows:

$$z_{ik} = a + c_1 y_{i1} + c_2 y_{i2} + \cdots + c_p y_{ip}$$

where z_{ik} is the discriminant function value for individual i and DF k, a is an intercept, just like in linear regression, c_p is the weight of the p^{th} variable on the DF, and y_{ip} is the value of the p^{th} variable for individual i. In this context, you get a value, z_{ik}, for each individual in discriminant space, called a **factor score**. The c_p values for each DF are called **loadings**. The loadings on DF-1 tell you how useful each original variable is to discriminating between your groups/treatments.

A MANOVA produces DFs, but really is simply a test of whether there are significant differences between group centroids. It calculates a number of statistics to test this hypothesis, including the **Pillai Trace**, which is viewed as most robust, and **Wilk's Lambda**, which is the most popularly implemented in statistical software. Since these values do not have test distributions, they are converted to F-statistics, with typical significance tests then applied (we will not go into how this is done – R will do the conversion automatically).

The assumptions of MANOVA are multivariate normality of the data, multivariate homoscedasticity, no multicollinearity, and that there are no multivariate outliers. Unfortunately, multivariate normality and homoscedasticity are difficult to test, and testing one variable at a time tells you nothing of whether the assumptions are met from a multivariate perspective. Fortunately, MANOVA is robust to violations of these assumptions. We have used correlation analysis to test for multicollinearity in past labs. Multivariate outliers can be evaluated by calculating Mahalanobis distances of the points, which is essentially the distance from the multivariate origin in DF space.

What is meant by "homoscedasticity"?

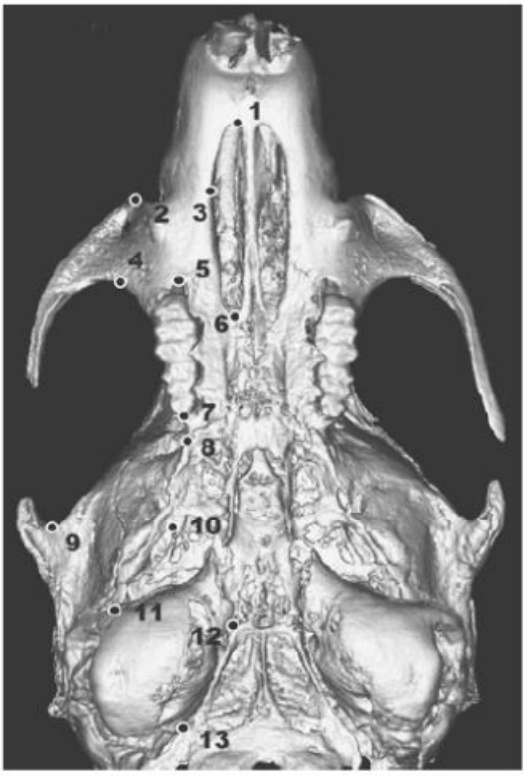
What is meant by "multicollinearity"?

The linear model weights for a given DF, or loadings, together form the **eigenvector** for the data. each DF has an eigenvector, which is simply the values $c_1 - c_p$. Each eigenvector is related to the original data by a number called the **eigenvalue**, abbreviated λ. The eigenvalue is useful because the eigenvalue for a given DF, divided by the sum of all eigenvalues for all the DFs is equal to the proportion of total dataset variance explained by each DF.

2. Discriminant Function Analysis Background

Discriminant Function Analysis (DFA), also called Linear Discriminant analysis (LDA), is simply an extension of MANOVA, and so we deal with the background of both techniques first. This is also done because different software packages provide different amounts of the results along with their MANOVA output or their DFA output. For example, in R, MANOVA gives you only the test of significant differences between group centroids. All other output (eigenvectors, eigenvalues, factor scores) are provided with the DFA/LDA output.

In considering DFA, it does exactly the same thing as described above for the MANOVA, but provides all of the remaining output. This allows you to determine not only which variables are best at discriminating groups, but also which groups are different and which are not. One useful tool to help you do this is classification. DFA uses the discriminant functions to determine which group it would assign each individual in your dataset to. It does this using a cross-validation technique called **jackknifing**, where the DFs are calculated while excluding one individual from the dataset and then using the DFs to classify that individual to a group. This is repeated with all individuals in the dataset, and you keep track of which classifications are done correctly and incorrectly. When complete, the analysis returns the proportion of correctly classified individuals to each group. Groups with a high rate of correct classifications are easy to discriminate, while those with lower rates are more difficult.



3. Doing a MANOVA in R

MANOVA is implemented as a function in the "stats" package in R, which is automatically loaded when you open R. DFA is implemented as a function (called **lda**, or linear discriminant analysis – the alternate name for DFA) in the "MASS" package, which comes automatically installed with R, but not loaded. ***Load the MASS package in R.***

The dataset we will be using to learn how to do MANOVA and DFA in R is a little more complex than most of our previous datasets and requires some explanation. The dataset quantifies the three-dimensional skull shape of the common mouse, *Mus musculus*, and compares this between wild mice, wildtype laboratory strain mice, and various laboratory mice that have mutations that affect cranial (head) shape. The table below (modified from *Jamniczky and Hallgrimsson. 2009. A comparison of covariance structure in wild and laboratory muroid crania. Evolution 63: 1540-1556.*) provides you with the strains, genotype, and a description of each strain. It also assigns strains into Lab (wildtype), Mutant, and Wild.

Category	Strain	Genotype	Description
Lab	AWS	Wildtype	Normal skull, but some have spontaneous cleft lip
Lab	CBA	Wildtype	Normal skull
Lab	DBA	Wildtype	Normal skull
Lab	FVB	Wildtype	Normal skull
Mutant	Crf4	Crf4/Crf4	Reduced size of the face
Mutant	LTL	Ghrhr-/Ghrhr-	Small size mice due to prolactin deficiency
Mutant	Mceph	Mceph/Mceph	25-30% larger brain than normal
Mutant	Nipbl	Nipbl(+/-)	Reduced cranial size and alteration of face shape
Mutant	PTN	Cre (fl/fl)	Increased length of both face and base of the skull
Wild	Mus	Wildtype	Normal skull

The data include twenty-four variables that quantify skull shape, derived from a geometric morphometric analysis. Such an analysis involves digitizing landmarks on an object (a skull) to get 3-D coordinates (see figure for landmarks used in this study, shown from a ventral view of the skull – also from Jamniczky & Hallfrimsson, 2009). These coordinates are then transformed into what are called "partial warp scores" through some complex mathematics. So, your dataset contains unique individual identifiers for each of 239 mice, identifies each as lab, mutant, or wild, identifies the strain or mutation that each belongs to, and 24 derived variables that together quantify skull shape. We are interested in whether these various categories and strains of mice can be distinguished simply based on cranial shape.

The dataset is already in tab-delimited text form. ***Download the dataset and import it into R, as object "mus_data". Examine the dataset using the str() and summary() functions. If we start by doing a MANOVA, the syntax is as follows:***

> manova(Y~M)

Where Y is a matrix of the response variables, and M is a model of categorical variables. Note that Y must be a matrix and not a data frame, so if you just specify the columns containing your skull shape characters from their data frame, you will get an error. ***Make a new object, called "mus_shape", that is a matrix containing just the skull shape variables listed in the "mus_data" data frame. You can use the function "as.matrix()" to do this, referring to the appropriate columns using the square bracket reference system. Then use srt() to ensure that you got what you wanted. Next, fit the manova model, using "mus_shape" as the response variables, and the variable "Category" is the factor, saving the model as "mus_manova1". Take a look at the new object. What does it contain?***

Next, it is time to get the summary of your model that will test the null hypothesis. Do this as follows, specifying the Pillai Trace, which is the most robust test statistic, and save the results to an object:

> summary(manova_object, test=c("Pillai", "Wilks", "Hotelling-Lawley", "Roy"))

What is your null hypothesis for this MANOVA?

The analysis uses the Pillai statistic, but then converts it to an F-statistic, which is used to conduct the test, because the distribution of F is known, but that of the Pillai Trace is not. The degrees of freedom are calculated as:

$$df_{numerator} = (\#\text{ of response variables})*(\text{factor df})$$

$$df_{denominator} = (\text{total df})*(\text{factor df})-(df_{numerator})$$

What do you conclude from the analysis?

The nice thing about MANOVA (but not DFA) is that you can also use more complicated models, with multiple factors. In the mouse skull shape dataset you have two factors. ***What are these two factors? Describe the study design based on how these factors relate to one another. (Hint: What sort of design do you get from considering both Category and Strain in one analysis?)***

Implement your more complicated MANOVA, including both Category and Strain as factors (related correctly), save the fitted model as an object, and then get the summary table with the Pillai Trace and associated F-test. Complete the table below – more rows are included than you might need.

Effect	df	Pillai	F_{approx}	df_{num}	df_{den}	P

What are the null hypotheses being tested by the above analysis?

What do you conclude, biologically from your analysis?

4. Doing a DFA in R

Now that you have done a MANOVA and have a significant result, the most pressing question is what to do next. You have multiple groups and multiple variables and you know there is a difference. Questions of interest now, are which variables are best at discriminating groups and which groups are different from which other groups (and how different are they)?

The **manova** function allows one way of doing this, advocated by some in the literature: doing an ANOVA on each response variable in turn with the same factors as in your MANOVA. This will allow you to see which of the variables differ significantly among groups, but it reduces the analysis from multivariate to univariate – you lose the information from considering all the response variables together. Nevertheless, we will do this to see how it works. Try the following function on your second MANOVA model and save the output to a new object:
> summary.aov(manova_object)

View the resulting object. What does it give you? What would you conclude from this analysis?

A multivariate approach to discovering which groups differ and which variables are most important at discrimination is DFA (also called LDA). ***What do each of these acronyms stand for?***

Start by doing a DFA on mus_shape with Category as the grouping variables, using the following function, and save the output to an object, mus_dfa1:
> lda(X~Y, CV=F)
In this function, X is your grouping variable, Y is your matrix of response variables, and CV refers to cross-validation. When CV=FALSE, you get the important parts of a DFA. The output includes (in order) the prior probabilities of the groups, which is the proportion of individuals belonging to each group; the mean value of each response variable for each group; the eigenvectors for the DFs; and the "proportion of trace", which are the eigenvalues for each DF.

In general, what information do the eigenvectors give you? What about the eigenvalues?

For the DFA you just did, calculate the proportion of variance explained by DF-1 (called LD1 in R). What is it?

For the DFA that you just did, what are the five most important response variables for discriminating wild, lab, and mutant mice? List them in order from most important to least.

When you repeat the DFA/LDA with CV=T, you get completely different, but still useful, output. ***Do this analysis as well and save it as a new object.*** The output is a list. Component $class is a factor that lists what group each individual would be classified to using the DFA with a jackknifing approach. Component $posterior is the probability that each individual belongs to each group. If you look at the first individual, it has a 0.72 probability of being a wildtype lab mous, 0.275 probability of being a mutant, and virtually no chance of being a wild mouse. This result would mean that the first individual would be classified to wildtype lab. If you look at the first entry in the $class vector, you see that this is the case. We will use this output to calculate how frequently individuals are correctly classified to each group. If your DFA with CV=T is stored in object **mus_dfa1a**, then try the following (otherwise change the name of the object):
> mus_dfa1a_table <- table(mus_data$Category,mus_dfa1a$class)
View the object. It compares what each individual is classified as by the DFA to what group the individual actually belongs. The rows represent the group to which individuals actually belong.
How many individuals are correctly and incorrectly classified to the Lab group?

You can turn this table into proportion of successful classification using the following code:
> diag(prop.table(mus_dfa1a_table,1))

Do this and bind the resulting vector as a column to the table you just made using cbind. Reproduce the table below.

	Lab	Mutant	Wild	Prop
Lab				
Mutant				
Wild				

You can also calculate the overall proportion of correct classifications with the following line. Put this number beside the table above.
> sum(diag(prop.table(mus_dfa1a_table)))

What remains to be done is extracting the factor scores from the DFA. These give you the value for each DF that each individual has, and allows you to plot the individuals in a discriminant function space. This may be useful because it shows you which individuals may look most alike in a multivariate sense. It also allows you to check for outliers by calculating the Mahalanobis distance (distance from the multivariate origin) for each individual, a key assumption of DFA.

R allows you to plot factor scores very easily:
> plot(mus_dfa1)

Which of the three groups are most similar to one another?

However, this approach doesn't give you the actual factor score values. ***Save the following line, which extracts the factor scores as a new object, scores1:***
> predict(mus_dfa1)$x

The **predict** function actually returns a list that contains the same thing as a DFA with CV=T, but includes the factor scores as well, saved as item $x. ***Plot your new object to confirm that it is gives you what you expect.***

It is now time to check for outliers. ***First calculate the Mahalanobis distance for each point using the following function:***
> mahalanobis(x, center, cov)
Where **x** is your data, the object containing factor scores. **center** is a vector of length *p* identifying the origin, where *p* is the number of dimensions in the dataset (this is the number of columns in your factor score object). Since the factor scores for DF1 and DF2 are each centered around zero, you can substitute **c(0,0)** for center. Finally, **cov** is a *p x p* variance-covariance matrix relating your DFs. You can simply substitute the function **cov(x)**, where, again, **x** is the object in which you stored your factor scores. The output to the mahalanobis distance function is a vector of distances from origin for each individual in your dataset. ***Try doing this now, storing the Mahalanobis distances as a new object. Then make a box plot of the distances. Does it look like there are any outliers? How many? Since DFA is sensitive to multivariate outliers, if this analysis were for publication, it would be advisable to remove the outliers or deal with them in some way and redo the analysis. Don't worry about doing this here.***

5. MANOVA and DFA – brining it all together.

Although MANOVA can handle complex designs that include interactions and nesting, DFA cannot. The mouse skull shape dataset includes not only the general category that each mouse strain belongs to, but also the strain that each individual belongs to. You have done a MANOVA already that takes both Stain and Category into account, and hopefully found that both factors are highly significant. ***Repeat the complete DFA that you did with Category as grouping variable, but now use Strain as the grouping variables. Report all of your results below, using tables as needed. Be sure to include your interpretation of the results. Do not bother doing the individual ANOVAs for each skull shape variables (they aren't a good way of getting a handle on what is going on). You should have all the guidance you need in the earlier parts of this lab. One tip that may help you is that you should plot factor scores two at a time. Try it the way we did it before first to see what you get, though.***