Applied Ecological Statistics - ECO 636

Lab 3: More Than Two Samples

Introduction

The aim of this lab practical is to demonstrate a statistical analysis in which the objective is to test for statistical differences in the means of three or more groups using linear models in a one-way analysis of variance-*esque* analysis. While this particular lab focuses on mutiple groups, these methods are just simplified for a two groups! In this lab, we first work through an example highlighting the key features of the analysis. Then, you will use the data set provided to conduct a similar analysis as as a take-home assignment.

First worked example

ANOVA background

As always, let us first review the structure of a general linear model for data with a continuous response variable, y, and a single categorical explanatory variable with g groups or levels:

$$y_i = \beta_0 + \beta_{1(g)} X_{1i(g)} + e_i$$

remember that, although X is single categorical variable, in the model it is technically treated in the model matrix as an array of binary indicator variables identifying which group each observation (i) belongs to. Let's say we have three groups: Group 1, Group 2, and Group 3 (we will assume Group 1 is the reference group). This model would look like this:

$$y_i = \beta_0 + \beta_{1(2)} X_2 i + \beta_{1(3)} X_3 i + e_i$$

To make things a little less abstract and to motivate some thinking, let's generate a synthetic data set:

```r
my_dataset <- data.frame("skull" = rnorm(6,10,2),
                          "group" = factor(rep(c('A','B','C'),each=2)))
my_dataset
```

```
##       skull group
## 1 13.036252     A
## 2  6.390362     A
## 3  7.827642     B
## 4 10.195304     B
## 5  9.159034     C
## 6  8.102832     C
```

Quiz yourself: What would the *model matrix* (how **R** codes the variables) look like for the `my_dataset` data if I were investigating group specific variation in skull size (i.e., $\text{skull}_i = \beta_0 + \beta_{1(g)}\text{group}_i + e_i$ or `glm(skull ~ group)`)?

Back to the breakdown of the statistical model. Just like when *g=2*, it has an intercept term (β_0) which, once again, represents the mean of the reference group (Group 1). Now instead of just one slope parameter (β_1) we have $g-1$ slopes. And, just like the two-sample model, these parameters represent the differences (or contrasts) in group means compared to the reference group. So, in this example, there are $g-1=2$ slopes: $\beta_{1(1)}$ is the estimated difference between the mean of Group 1 and estimated mean of Group 2, while $\beta_{1(2)}$ represents the estimated difference between the mean of Group 1 and the estimated mean of Group 3 (make sure you can explain what this means to your classmates!).

Quiz yourself: Why are there $g-1$ slopes instead of g 'slopes'? Because the slopes, or differences, represent contrasts to the reference group which is the estimated intercept, β_0, so there is no difference to be estimated for the reference group.

In the *two-sample* class exercise, X was a two level factor which `R` converted into a binary indicator variable to construct the design matrix: $X=0$ denoting membership to the first group, and $X=1$ denoting membership to the second. In the case of $g>2$ `R` constructs a design matrix with g columns (compare your model matrix to the one below).

```r
model.matrix(skull~group,my_dataset)
```

```
##   (Intercept) groupB groupC
## 1           1      0      0
## 2           1      0      0
## 3           1      1      0
## 4           1      1      0
## 5           1      0      1
## 6           1      0      1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

For this model, which has a three-group categorical predictor variable (groups `A`, `B`, and `C`). The first column of the design matrix is the intercept, and contains all 1's, and the subsequent columns indicate group membership. Group `A` is the reference level, or `Intercept` (**Quiz yourself:** why `A` and not `C`?). Also, notice that the values of the `GroupB` and `GroupC` columns for the observations in group `A` are all 0's.

Worked example - Garlic mustard

Now let's work through an analysis using some data from a study in Harvard Forest on garlic mustard *Alliaria petiolata*. This data was collected, and kindly given to us, by one of our faculty colleagues, Dr. Kristina Stinson.

Garlic Mustard is native to Eurasia and North Africa but is an invasive weed in North America where it can invade and dominate the understory of native deciduous forests and displace native wildflowers. Dr. Stinson has been studying the invasion success of Garlic Mustard using a variety of approaches including population-level experiments. The data we will use in this example come from multiple plants grown in three treatment habitats (`treat.hab`):

1. forest interiors: `Forest`
2. sunny, open, or edge habitats: `Sun`

Figure 1: Garlic Mustard flowers (www.herbierimages.be).

3. intermediate habitats `Int`

Several measurements were recorded from each of the plants within each treatment. For this example we are going to investigate whether the height of plants, measured from base to tip of a dried plant, differs among treatments. How would we write a linear model designed to answer this question?

$$\text{height}_i = \beta_0 + \beta_{1\text{habitat}_i} + e_i$$

or, because `treat.hab` is a 3-level categorical variables, we have two different β_1s: $\beta_{1(2)\text{habitat}_{2i}}$ and $\beta_{1(3)\text{habitat}_{3i}}$

As in the *two-sample* model, the null hypothesis is still that there is no difference in the group means, or that $\beta_{1(2)} = \beta_{1(3)} = 0$. The alternative hypothesis, therefore, is that the differences do *not* equal zero.

Let's read in our data and see what it looks like.

```r
setwd("<your directory>")
mustard.full <- read.table(file = "HF.data.2005.txt",
                           header = TRUE)
head(mustard.full)
str(mustard.full)
```

As is usually the case, our data contain many more variables than we are interested in, and many missing values (NA's). The variables we will focus on for this analysis are treat.hab, the categorical treatment habitat variable, and dryheight, the height of the dried plant. Lets subset the data to just the information we are interested it, namely the two columns treat.hab and dryheight which we can do many ways. Here are two alternatives, one using the subset() function and the other using indexing.

```
mustard <- subset(mustard.full,!is.na(dryheight),
                  c(site,treat.hab,dryheight))
mustard1 <- mustard.full[which(!is.na(mustard.full$dryheight)),
                         c("site","treat.hab","dryheight")]
str(mustard)
str(mustard1)
```

Now let's take a closer look at the data. We can inspect the categorical variable treat.hab using levels() function which returns the name of each level and, importantly, the order in which **R** recognizes them. We could also use table() to see the frequency of observations in each level. There are three levels: Forest (forest interior), Int (intermediate habitats), and Sun (sunny, open, or edge habitats). We also see that we have an unbalanced experimental design with Forest having approximately 57% of our observations. While we are at it, let's calculate the means and standard deviations of each treatment habitat using tapply().

```
levels(mustard$treat.hab)
table(mustard$treat.hab)
(means <- tapply(mustard$dryheight, mustard$treat.hab, mean))
(sds <- tapply(mustard$dryheight, mustard$treat.hab, sd))
```

As is good practice, we will do some data exploration. Let's make the following plots: a *scatter plot* of the raw data, a *conditional boxplot* using habitat as a conditioning factor, a *Cleveland dotplot* by treatment factor, and a *histogram* of dry height.

```
par(mfrow=c(2,2))

#Scatter plot
plot(mustard$dryheight, ylim=c(0,2), pch=21,
     bg=unique(as.numeric(mustard$treat.hab))+2, cex=1.5,
     ylab="Dry Height (cm)")
```

```
#Conditional boxplot
plot(dryheight~treat.hab, data=mustard, ylim=c(0,2), pch=21,
     bg=unique(as.numeric(mustard$treat.hab))+2,
     col=unique(as.numeric(mustard$treat.hab))+2,
     ylab="Dry Height (cm)")

#Cleveland dotchart
dotchart(mustard$dryheight, groups=mustard$treat.hab,
         xlab="Dry Height (cm)")

#Histogram
hist(mustard$dryheight, breaks=seq(0,2,0.05),
     xlab="Dry Height (cm)", main="")
```

Quiz yourself: What are some things you notice about our raw data? How well do you think our data will meet the assumptions of the linear model (we will again assume our data are independent and measured without error)?

Well, our conditional boxplot suggests that the variances among habitats are not equal while our histogram suggests our response variable is not normally distributed. Remember though, that our model assumptions apply to the residuals. Let's fit a model and examine the residuals before proceeding (reflecting the iterative nature of statistical analyses). Doing so will provide the opportunity to deal with any major assumption violations. We can fit and plot the model to evaluate the structure and distribution of the residuals. NOTE: I used `glm(family = "gaussian")` instead of `lm()` - we will transition to using `glm()` as it is more flexible in the types of models we can make, adding `family = "gaussian"` means we are assuming a normal distribution and produces the same thing as `lm()`.

```
mod1 <- glm(dryheight~treat.hab, data=mustard, family = "gaussian")
par(mfrow=c(2,2))
plot(mod1)
par(mfrow=c(1,2))
hist(resid(mod1),breaks=seq(-2,2,0.1))
plot(resid(mod1)~mustard$treat.hab,main="")
```

The pattern of residuals indicates that our model assumptions are not well met. In particular, the Normal Q-Q plot indicates a departure from normality while several other plots suggest the variance differs among treatment habitats (non-homogeneity). What can we do to remedy violations of normality and homogeneity of variances? For now we can apply a transformation to our response variable (but we'll talk about other options later in the semester :)). A log transformation is often helpful in these situations. We can create a new column to our `mustard` data frame called `logheight` and plot the data again:

```
mustard$logheight <- log(mustard$dryheight) #LOG transform
par(mfrow=c(2,2))
plot(mustard$logheight, ylim=c(-3,2), pch=21,
     bg=unique(as.numeric(mustard$treat.hab))+2, cex=1.5,
     ylab="log(Dry Height (cm))")#LABEL CHANGE!

plot(logheight~treat.hab, data=mustard, ylim=c(-3,2), pch=21,
     bg=unique(as.numeric(mustard$treat.hab))+2, cex=1,
     col=unique(as.numeric(mustard$treat.hab))+2,
     ylab="log(Dry Height (cm))") #LABEL CHANGE!

dotchart(mustard$logheight, groups=mustard$treat.hab,
         xlab="log(Dry Height (cm))")#LABEL CHANGE!

hist(mustard$logheight, breaks=seq(-3,2,0.25),
     xlab="log(Dry Height (cm))", main="")#LABEL CHANGE!
```

Our log-transformed data appear more normally distributed with variances that are less heterogeneous. Now let's refit our model and plot the diagnostic plots of the residuals using the log-transformed data.

```
log.mod <- glm(logheight~treat.hab, data=mustard, family = "gaussian")
par(mfrow=c(2,2))
plot(log.mod)
par(mfrow=c(1,2))
hist(resid(log.mod),breaks=seq(-2,2,0.1))
plot(resid(log.mod)~mustard$treat.hab,main="")
```

The transformed data appear to meet our model's assumptions of normality and homogeneity of variance much better! Since we now have more confidence in making inferences from our model, we can begin to interpret the model output and make statistical statements about how garlic mustard height varies by habitat type. If we were only interested in testing our hypothesis that garlic mustard height varies by habitat type, we could use `summary()` to look at our model's $\hat{\beta}$'s and p-values. But what if we had other competing hypotheses we wished to evaluate? Well, we could then use a model-selection framework to compare different models representing our different hypotheses. What would be a logical competing hypothesis? That plant height does *not* vary among treatment habitats seems like a reasonable alternative. This, of course, is the intercept-only model. Let's fit and compare these models using AIC and the `aictab()` function.

```
library(AICcmodavg)
log.mod0 <- glm(logheight~1,mustard, family = "gaussian")
log.modT <- glm(logheight~treat.hab,mustard, family = "gaussian")
models <- list()
models[[1]] <- log.mod0
models[[2]] <- log.modT
names(models) <- c("Null","Habitat")
aictab(models)
```

We can see that the null-model with just an intercept estimate for plant height receives none of the model support. We would therefore be justified in making our inferences from the treatment model. So let's look at our model's $\hat{\beta}$s (estimates) and p-values using `summary()`.

```
summary(log.modT)
```

We can now see the estimates of the intercept, and the two differences (the slopes), as well as the associated standard errors, and p-values. Remember that the $\hat{\beta}$s for the intermediate (`Int`) and sunny (`Sunny`) habitats are the differences between their means and the mean of the Forest habitat (`Forest`), which **R** has used as the reference level. The p-values, therefore, test the null hypothesis that the differences are equal to zero (i.e., no difference between forest and intermediate habitat types and no difference between forest and sunny habitats).

Quiz yourself: Based on this output, what can we say about our null hypothesis of no difference in plant height among treatment habitats? Are the mean plant heights for intermediate and sunny habitats larger or smaller than the mean plant height in the forest habitat? And are those differences significant?

GLM note:: Here you may notice that the bottom of the summary output for `glm()` is different from that of `lm()`! The summary output for `glm()` produces `Null deviance` and `Residual deviance` rather than `R-squared` or `F-statistic`. This is an important difference! *Deviance* is a measure of badness of fit, where higher numbers indicate a worse fit. The `Null deviance` shows how well the response variable is predicted by the null model and the `Residual deviance` shows the deviance for the full (or global) model. If the `Residual deviance` is quite a bit lower than your `Null deviance`, you know the global model is the better one!

But what if we wanted to know if there was a significant difference between the intermediate and sunny habitats? We can't know this information given our model's current structure but we can easily re-level `treat.hab` to make one of these habitats the reference level. Let's use `as.factor()` to create a new column of our re-leveled treatment habitats. Alternatively, see the help page for `levels()` to see how to re-level an existing factor. We can then re-fit our model and examine the `summary()` output. How does mean dry height in sunny habitats compare to mean dry height in intermediate and forest habitats?

```
mustard$new.hab <- factor(mustard$treat.hab,c("Sun","Int","Forest"))
levels(mustard$new.hab)
new.mod <- glm(logheight~new.hab,mustard,family = "gaussian")
summary(new.mod)
```

While re-leveling and re-fitting your models it may be convenient for comparing group means across all groups, this approach becomes tedious when you have many groups. Furthermore, when we start making multiple pair-wise comparisons the probability of committing a Type I error increases. Many multiple comparison corrections exist to reduce this error, like the Tukey's 'Honest Significant Difference' (HSD) method.

Now that we have confirmed any statistical differences in plant height among treatment habitats, let's plot the expected value (i.e., mean) plant height and its 95% Confidence Interval (CI) for each habitat. We can calculate these values using `predict()` and setting `interval="confidence"`. `predict()` requires the creation of a new data frame containing values of our independent variable(s) that will be plugged in to the model to make predictions. Importantly, the new data frame must have the same column names as data used to fit the model. The treatment habitat data was stored in a column called `treat.hab` so we need to use the column name `treat.hab` when creating our new data frame. Because our independent variable was a factor with three levels, we simply create a data frame of a factor with 3 observations, one of each of the three levels (`A`,`B`, and `C`). We can use this opportunity to order the levels in a way that makes sense for our application. Notice that we can also compute the expected values by hand by appropriately adding the $\hat{\beta}$'s as demonstrated below. Also notice that our expected means are identical to the means of our log-transformed raw data.

```
df <- data.frame(treat.hab=factor(c("Forest","Int","Sun")))
CI <- as.data.frame(predict(log.modT, newdata=df, se.fit=TRUE))
CI$lower <- CI$fit-1.96*CI$se.fit
CI$upper <- CI$fit+1.96*CI$se.fit
CI$Habitat <- c("Forest","Intermediate","Sunny")
CI

coef(log.modT)                              #Group A
coef(log.modT)[1] + coef(log.modT)[2] #Group B
coef(log.modT)[1] + coef(log.modT)[3] #Group C
(means <- tapply(mustard$logheight, mustard$treat.hab, mean))
```

By now you may have noticed that our expected values are all negative. This doesn't make sense given that our dependent variable is dry plant height in centimeters. Our negative values are the result of our log transformation which allowed us to better meet our model's assumptions. But for reporting our results it is important to *back-transform* our parameter estimates and CI's to the scale of our original data. We can easily do this by exponentiating (`exp()`) our expected values and CI's. These back-transformed expected means are similar to, but not identical to, our observed means (due to the transformation).

```
CI[,1:5] <- exp(CI[,1:5])
CI
(means <- tapply(mustard$dryheight, mustard$treat.hab, mean))
```

Finally, let's plot our back-transformed expected means and their 95% CI to visualize differences in plant height among habitats on the scale of the original data. Because X is a factor, `plot()` will automatically use a numeric sequence when labeling the x-axis (e.g., 1, 2, 3). In order to make the x-axis display our treatment habitats, we need to remove the x-axis label in `plot()` using the `xaxt="n"` argument. We can use `axis()` to add a new axis with specific labels afterwards. We can also add error bars to the plot showing the 95% CIs; there are many functions available to do this, but one relatively simple function is `errbar()` from the package Hmisc. Within errbar() we need to provide four numeric vectors specifying the x and y values and the upper and lower interval values. This information is already contained within CI so we can simply supply the appropriate columns to each argument. Because our x-axis is categorical instead of numeric, the vector specifying the x-axis values can be a numeric sequence from one to three.

```r
library(Hmisc)
plot(CI$fit,ylab='Dry height (cm)',ylim=c(0,1),
     xlab='Treatment habitat',xaxt='n',cex=1,pch=15)
axis(side=1,at=1:nrow(CI),labels=CI$Habitat,
     las=1,cex.axis=1.075)
errbar(seq(1,3),CI$fit,CI$upper,CI$lower,add=TRUE)

# Try ggplot!
library(ggplot2)

m_plot <- ggplot(CI, aes(Habitat, fit,colour=Habitat,
      fill = factor(Habitat))) + geom_bar(stat="identity",
      show.legend=FALSE)

m_plot <- m_plot + geom_errorbar(data=CI, aes(ymin=lower,
,ymax=upper), linetype=1, color = "black",
alpha=1, width = 0.5, show.legend=FALSE) +
  xlab("Treatment habitat") + ylab("Dry height (cm)") +
  geom_hline(yintercept = 0, colour = "black", linetype=3)
```

Take-home exercise

Now it's your turn to fit a one-way ANOVA! The data you will use comes from a graduate student's masters research studying population fluctuations of birds in an urban-rural study system in Columbus, Ohio. The project included spot map surveys to measure bird territory density at 17 2-ha sites from 2005-2010. Please feel free to work on this with your fellow classmates! I suggest using Rmd to write your code and your write up at the same time. Please be sure to upload your pdf report and code to Moodle by the end of the week.

```r
colrip <- read.csv(file = "colrip.csv", header = TRUE)
# change year from integer to factor
colrip$year <- as.factor(colrip$year)
str(colrip) # look at data
```

Looking at the data, you can see it has one continuous variable, the mean number of territories over all years for each species (`n.terr`), and two factors, three migratory categories (`mig.stat`), and three species of birds (`spp`). What we are interested in is whether the number of territories at a site differs between our three species, Acadian flycatcher (`ACFL`), northern cardinal (`NOCA`), and American robin (`AMRO`). You are a state ecologist working in Franklin County, Ohio, and are interested in seeing if populations of these three bird species differ in sites throughout the county. Alternatively, the differences might be explained by regional climate differences between years (2005 - 2010). You formulate a *null* hypothesis (H_0) and two alternate hypotheses (H_{spp}, H_{yr}):

- H_0: "There are no differences in number of territories between species, or year."
- H_{spp}: "The number of breeding territories at a site differs significantly between species."
- H_{yr}: "The number of breeding territories at a site differ significantly between years."

Using the process we went through as a class: formulate a statistical model for each hypothesis, fit the models, evaluate the fit of those models, and interpret the model. In the process, you should follow these steps:

- Formulate and fit a statistical model for each hypothesis (H_{spp},H_{year},H_0)
- Evaluate the *fit* of these models. Do they fulfill the assumptions of a linear model? If not, what can you do about it?

- Using model selection, with AIC (`aictab`), compare the three competing hypotheses and interpret the output, which model *best* explains the variation in your data?

- Is there a significant effect of species or year on number of breeding territories in this study system?
- Fully interpret your *best* model and "report" your findings.
- What about differences between species in each year? That also seems like a reasonable hypothesis, doesn't it? Do any of these models test that?

Work together in groups, think outloud, talk, ask questions - good luck and have fun!

Figure 2: Tweet Tweet! Acadian Flycatcher hanging out on her nest!