Biol 206/306 – Advanced Biostatistics
Lab 2 – Experimental Design
Fall 2016

By Philip J. Bergmann

0. Laboratory Objectives

1. Continue familiarizing yourself with the R environment
2. Learn the basics of what experimental design is
3. Design some experiments/studies in a robust way
4. Learn what power analysis is
5. Do a basic power analysis
6. Learn about correcting for multiple comparisons

1. Introduction to Experimental Design

Experimental design sounds like it is intuitive and straightforward, which is why it is often a neglected part of learning science and statistics. In fact experimental design requires considerable thought and care, especially when the resulting data are to be analyzed statistically. The goal of experimental design is to **maximize power**.

What is power? Define it verbally and also give the equation for calculating it.

One can maximize power by explicitly considering what statistical analysis to use, what sample size to use, and what to fix the type I error rate at. This is referred to as power analysis, which we will do later in the lab. Power is also affected by the decisions you make in deciding what sorts of data to collect, how you select your sampling units, how you define your sampling units, and what biological questions you what to ask. So, a part of experimental design is very situation-specific, and consideration of the underlying biology is important, and a part of experimental design is very statistical (power analysis). We will consider both of these in today's lab, but keep in mind that they are inter-related. For example, you should be thinking about sample sizes you will need to obtain to do a robust statistical analysis while you are planning how to collect the data in the field. You should also be thinking about what statistical analyses you will conduct on the data before you even collect them.

Start an experimental design by thinking about the biological question(s) that you want to ask. If you can phrase these as hypotheses, it is even better. Identify what the important variables are in the study you are designing. These are really determined by your biological question and by the biological system you are studying. What is/are the **dependent** or **response** variable(s)? What are the **independent** or **explanatory** or **manipulated variables**? What are possible variables that you are not interested in, but might be important and could confound the study? These are

called **nuisance variables** or **covariates**. At this point, you should also consider whether a **control** of some sort is required, and, if so, what it should be.

Once you know your biological question and the variables involved, it is time to think about how to sample data for all of the variables in such a way that the data are as random as possible, as independent as possible, and as robust as possible. Start this stage by **identifying sampling units**. What is it that will make up the individual observations of your dataset? Are these units independent of one another? If not, how will you account for their non-independence in a statistical analysis? Will these sampling units be collected randomly? Will treatments be applied randomly? How many sampling units will you sample?

Define pseudoreplication. How does it relate to observation independence?

2. Practice with Experimental Design

In this section there are two scenarios described, which identify the biological question of interest and some of the variables that you would collect. For each scenario, answer the questions and design a study that would address the biological question as robustly as possible (no one design is necessarily correct).

Scenario A: Prey processing times in lizards

Most lizards are insectivores, some eating a wide range of insects, ranging from hard beetles to leggy grasshoppers to soft caterpillars. The amount of time spent processing the prey varies with various characteristics of the prey, especially size and hardness. Processing time is the time from subduing the prey (killing it) to when it is swallowed; it involves the physical break down of the prey into something that can be swallowed. The size of the lizard can also influence processing time because a larger lizard tends to be stronger and can break down the prey more quickly. You are interested in how insect type influences how long it takes a zebra-tailed lizard (*Callisaurus draconoides*) to process a prey item. You are specifically interested in four insect types that these lizards eat naturally: beetle, grasshopper, caterpillar, and ant. ***Design an experiment that robustly investigates this. Answer the following questions to help you accomplish this.***

What is the biological question?

What is/are your response/dependent variable(s)?

What is/are your manipulated/explanatory variable(s)?

Are there any nuisance variables or covariates? If so, what are they?

What is a good control, if any?

Use the rest of the space provided on this page to design your study. Use point form, but provide enough detail to make as robust a study as you can. Try to consider things like number of trials and number of individuals/subjects to include in your study. Pretend that this is a project for which you have three months (the summer), and so, you cannot just have a massive sample size of 200.

How would you modify your design if you knew that on some occasions lizards do not process their food as quickly as possible, lacking motivation?

Scenario B: Grazing, plant diversity and biomass
Grazing cattle on land can influence the plant community growing in an area because grazing may be selective, and it shortens plants, so can influence competition among species for sunlight. A large area (say 5,000 acres) of Chihuahuan grassland in New Mexico belonging to the National Forest Service has been intensively grazed by cattle for the last five years. The NFS is considering whether to renew a lease of grazing rights to a rancher that has been using the area. They hire you to investigate how grazing has influenced species richness (number of species) and total plant biomass in the area. ***Design a study that robustly investigates this. Answer the following questions to help you accomplish this.***

What is the biological question?

What is/are your response/dependent variable(s)?

What is/are your manipulated/explanatory variable(s)?

Are there any nuisance variables or covariates? If so, what are they?

What is a good control, if any?

Use the space provided on this page and the next to design your study. Use point form, but provide enough detail to make as robust a study as you can. Try to also consider things like number of plots to include in your study. Pretend that this is a project for which you have three months (the summer), and so, you cannot just have a massive sample number of plots.

How would you modify your design if you were not interested in the 5000 acre area, but instead where interested in how grazing influences the response variables in a Chihuahuan grassland in general? Hint: In this case, the grazing has not already taken place.

3. What is Power Analysis?

Power analysis is an *a priori* approach used to help design an experiment. Power analysis considers how sample size (n), effect size (d), power (1-β), and type I error rate (α) interact with one another for a given statistical test. It is an *a priori* analysis because it is conducted during the experimental design stage, before collecting any data. Some people use power analysis in an *a posteriori* way, where they attempt to calculate the power of their analysis after it is done, but this has been criticized as circular, and therefore, unjustified. A good approach to power analysis is to fix two of the four variables listed above (n, d, 1-β, and α), and then manipulate one of them to see the effect of the manipulation on the fourth variable. In most statistical applications, α is fixed at 0.05, although in some cases, one may want to modify it to be more or less stringent. Scientists also often view a power of 0.8 to be sufficiently high to identify differences, yet not so high to require massive sample sizes. If α and 1-β are fixed, one can investigate what sample size is needed to be able to identify an effect of a given size (d is manipulated, n is response), or to see how small of an effect can be detected with a given sample size (n is manipulated, d is response).

Define type I error rate (α).

How does changing α influence power (1-β)?

Under what circumstances would you be interested in knowing how small of an effect you can detect, given a specific sample size?

Under what circumstances would you be interested in knowing what sample size you would need to detect a particular effect size?

4. Doing a Power Analysis

Last week, you learned how to do a t-test in R and tried out your new skill on a sample dataset. Today, you will do a power analysis for a two-sample t-test in R. To do so, you will need the R package **pwr**. This package allows you to do power analysis for a range of common statistical tests in addition to the t-test – you can explore these if you like by reading the usage manual for the package, which you can download from the CRAN website. ***Open R, then install the package 'pwr', then load the package.***

We will learn a few new functions today, one for the actual power analysis, and one to make the power analysis run more efficiently. For the power analysis, you will need the following function:

```
> pwr.t.test(n=NULL, d=NULL, sig.level=0.05, power=NULL,
  type=c("two.sample", "one.sample", "paired"),
  alternative=c("two.sided", "less", "greater"))
```

Where, n is sample size, d is effect size, and the remaining arguments should be self-explanatory. for **type** and **alternative**, type the option you would like to use in double quotes. The default is (always) the first option listed: two-sample and two-sided. **NULL** indicates that the argument is not assigned a value by default. How this function works is that you must specify values for all but one of the arguments listed, and it then calculates the unspecified argument. ***Give it a try, what is the sample size needed that would give you a power of 0.8 for an effect size of 1.0?***

How about for an effect size of 0.5?

We can explore the effects of manipulating any of these parameters on any one of the other parameters by repeatedly adjusting the manipulated parameter (in the example above, d) and recording the resulting response parameter (above, it is n). However, a more efficient way of doing this is to automate the process, something that R is good at (but you have to write the code). To do this, we will use a **for loop**. The basic syntax for a for loop is:

```
> for (i in <sequence>) {statements}
```

Where **i** is a variable that takes a different number after each iteration of the loop, **<sequence>** is a sequence of integers (this could be something like 1:10, or something more complex), and **{statements}** is simply one or multiple lines of code that are repeated with each iteration of the for loop. Note that the repeated steps are encapsulated in **{}** – it will not work without these.

To address the question of how different effect sizes influence what sample size you need to maintain a power of 0.8, you will first create a vector of effect sizes that you wish to sample. Then we will use a for loop to calculate the sample size for each effect size, outputting the sample sizes to another vector. ***Try the following, and when you define a new object, type its name in to view what it looks like before proceeding. Also use the str() function to see what the structure of the new object is. Get into the habit of doing this automatically to make sure you are getting what you expect and need.***

```
> ds <- seq(from=0.1, to=2.0, by=0.1)
```
Creates a vector of effect sizes.

```
> ns <- NULL
```
Creates an empty vector that will hold your calculated sample sizes.

```
> for (i in 1:length(ds))
```
Sets up the for loop. **i** will start at 1 and go to the length of the **ds** vector, so there will be as many iterations as entries in **ds**. Note that this is just a counter, it does not actually look at the values in the **ds** vector.

```
+ {ns[i] <- pwr.t.test(d=ds[i],power=0.8)$n}
```
This is an action-packed line of code! It calls the object called **ns** and the line fills the i^{th} entry of **ns** with the result from the other side of the <-. We will fill each entry with a t-test power analysis, where d is specified as the i^{th} entry of **ds**, with a specified power of 0.8. Also notice the **$n**, which takes the sample size from the output of the **pwr.t.test** function and places it alone in the i^{th} entry of **ns**. The beauty of this is that the order in which the sample sizes are in **ns** corresponds to the order in which the effect sizes are in **ds**. (Note that the "+" symbol is the prompt when a function's syntax is not completed on one line – in this case **for** is the function.)

The next step, aesthetically, is to put the effect and sample sizes together. ***Try this:***
```
> n_pwr <- cbind(ds,ns)
```

View the new object with your ds and ns columns. Note that, as it currently stands, it is a matrix. To be able to plot the data using Deducer, it needs to be a data frame. You can coerce an object of one class to another class using a number of functions. To convert your matrix to a data frame try this:

```
> n_pwr <- as.data.frame(n_pwr)
```
This simply overwrote the original matrix object with a new one that is a data frame. If you wanted the data both in a matrix and a data frame, you could call this new object something else.

The advantage of doing that is that you would have all the steps preserved – you cannot undo what you've done. Note that in a data frame, the rows are now simply numbered and the numbers do not appear in […], as they do for a matrix. Another way to produce the data frame without the intermediate step is:

```
> n_pwr <- data.frame(ds,ns)
```

Plot the data that you have stored in n_pwr. What do they tell you about how the sample size needed to get a power of 0.8 changes with effect size?

Assignment (2 points): Save the plot you created, and insert it at the end of this document. Add a figure caption under the plot and hand it in with this worksheet.

You may also be interested in turning what you just did around slightly, by asking what power you get for a fixed effect size with different sample sizes. If you were to do a pilot study and found that you might get an effect size of 0.65, you might be interested in how big of a sample size you need in your actual study to detect such a difference. ***Repeat the analysis that you did above, but this time, specify effect size as 0.65, and consider what sample sizes are needed to obtain certain levels of power. Power ranges from zero to one, so take a look at power ranging from 0.2 (which is pretty low) to 1.0 at intervals of 0.1.***

If you increase the power that you require from an analysis, what happens to the sample size that you need to make that goal?

Why is the sample size needed to get a power of 1.0 so high?

Trim off the last row of the object so that the data are easier to visualize:

```
> object <- object[1:8,]
```

Here **object** is whatever you called your data frame, and we use the square brackets to specify that we want to keep the first 8 (out of 9) rows.

What sample size would you need to get a power of 0.8? This sort of analysis can be useful if each sample costs a certain amount of money and/or time and you need to understand what sort of power you can get, given the resources that you have at your disposal.

Assignment (2 points): Save the plot you created, and insert it at the end of this document. Add a figure caption under the plot and hand it in with this worksheet.

5. Multiple Comparisons

The last component of today's lab relates to experimental design and power analysis in that it involves adjusting the type I error rate (α) so that your chance of making a type I error does not increase when you re-analyze the same data multiple times. The reason that this is an issue to consider is that, in general, the probability of an event happening when there are repeated chances of it happening is the sum of the probabilities for each chance. Consider taking a die and rolling a six – the probability is of this happening is 1/6=0.667. If you roll the die three times, the probability of rolling one six is 1/6 + 1/6 + 1/6 = 0.5. Likewise, if you do a statistical test and set α=0.05, as is traditionally done, but then do another one on the same dataset, α increases, possibly as high as 0.1. Suddenly you have a 10% chance of erroneously thinking there is a significant difference between two samples when there is not, instead of a 5% chance! Because of this, it is important to correct α so that it remains at about 0.05 (5%) over your entire experiment/study.

The approach often taken, and taught in many introductory stats classes is the Bonferroni correction. This simply involves dividing your experiment-wise α by the number of comparisons, so $\boldsymbol{\alpha_{comp} = \alpha_{expt} / n_{comps}}$. If you have ten comparisons, then α_{comp} would be 0.005, meaning that you would need a p-value of 0.005 for a test to be considered significant. However, this approach has been criticized as being too conservative. The sequential Bonferroni correction is less conservative and justified because once you do one of your ten comparisons, only nine remain. To do the sequential Bonferroni correction, you first put the p-values for your ten comparisons in order from lowest to greatest. Then you calculate α_{comp} for each comparison, starting with the lowest p-value and working your way up. In this case, $\boldsymbol{\alpha_{comp} = \alpha_{expt} / i}$, where i is the rank of the comparison (the order in which the comparison is when they are ordered from greatest to lowest p-value, so the comparison with the lowest p-value would have the highest rank). With the sequential Bonferroni correction, note that α_{comp} changes for each comparison.

Although widely in use in publications, the sequential Bonferroni correction has also been criticized for being too conservative/stringent, on the grounds that it does not consider False Discovery Rate. Essentially, what this means is that it does not take into account both the number of comparisons made and the rank of the comparisons simultaneously. A method developed by Benjamini & Hochberg in 1995 (Journal of the Royal Statistical Society B 57: 289-300) takes this into account, but the approach has not yet been widely used in biology. The B-H method does the following calculation: $\boldsymbol{\alpha_{comp} = i\, \alpha_{expt} / m}$, where m is the total number of comparisons, and i is the rank of the comparison, when the comparisons are ordered lowest to greatest in p-value (note that this is the reverse order from the sequential Bonferroni correction!).

You will compare how you interpret results from a correlation analysis of stickleback morphometrics using the different approaches to correcting for multiple comparisons. First, you will calculate pairwise Pearson correlations for five variables. Then you will calculate their p-values. Finally, you will calculate α_{comp} for each comparison using each method of correction. It is easiest to do some of this in R and some in Excel. Do the following steps:

1. ***Open the lab 2 dataset in Excel, take a look at it, including the explanation of variables, and save it as a tab-delimited text file.***
2. ***Import your text file into R, as an object called "data", which should be a data frame.***

3. ***Calculate pairwise correlations using the following function and save them as an object "pearson":***
   `> cor(x, y=NULL, method=c("pearson", "kendall", "spearman"))`
   - You can either input two variables, x and y, or a data frame with multiple variables, x. Pearson correlation is the default method, so the argument can be skipped.

Looking at all pairwise correlations, how many are there? This is the number of comparisons that you have done for this dataset.

The next step is to calculate the p-values. Unfortunately, we will simply use a primitive way of getting the pairwise p-values that is not very elegant. However, a few lines of code, using for loops could automate this. If you implement the for loop approach and hand in the code, it is worth 5 bonus points on this week's assignment! Calculate the p-values using the following function:

```
> cor.test(x, y, alternative=c("two.sided", "less", "greater"),
method=c("pearson", "kendall", "spearman"), conf.level=0.95)
```

This function works similarly to the cor function, but does not do all pairwise comparisons, so both x and y must be specified and each must be a vector.

For example, try:

```
> cor.test(data$SL,data$WT)          # The other arguments are default.
```

Another way to make this easier is using the **`attach(data.frame)`** ***function. This function allows you to attach a data frame to the workspace, allowing you to directly call the variables instead of needing to use the $ sign. This is convenient because it cuts the number of key strokes. For example, try:***

```
> attach(data)
> cor.test(SL,WT)
```

However, note that if you then want to use a variables from a different object, you ned to either use the $ approach or use **`detach(data.frame)`** ***and then attach a different object.***

Take a look at the output. Notice that it also gives you the correlation and the details of the t-test done to test for a significant correlation. If you use the "str" function with the above code, you will see the structure of the output and how you can even specify how to get only the p-value.

Assignment (6 points):

Next, open Excel and make a spreadsheet with the following columns: comp, R, p, Rank_SB, Rank_BH, no_corr, bonferroni, seq_bonf, and BH. Comp is the variables being compared (for example, "SL-WT"), R is the correlation, p is the p-value, Rank_SB is the rank of each comparison when doing a sequential Bonferroni correction, Rank_BH is the rank of each comparison for a Benjamini-Hochberg correction, and the remaining columns will be the α_{comp} values for each method (no correction, Bonferroni, sequential Bonferroni, and Benjamini-Hochberg).

Fill out the R and p columns from the output you get in R. Then sort the spreadsheet by p-value, and fill in the two Rank columns based on the descriptions of the approaches.

Calculate the α_{comp} values for each comparison and each method.

Finish up by making bolding the α_{comp} values for each method that show that a given correlation and associated p-value are significant (non-significant comparisons should stay unbolded).

Insert your table at the end of this worksheet and hand it in.

Describe how your conclusion about which morphometric variables are significantly correlated differs between the different approaches to correcting for multiple comparisons, and how they compare to when no correction is made.

You can get 3 bonus points in this lab if you write some code that will automatically do the correlation analyses for all pairwise combinations of the variables and compile the results (R and p values) in a single object. The code should use the cor.test function, but cannot call on it multiple times.