Frameworks for Statistical Analysis

Objective: The purpose of this session is to use R to compare the density of ant nests in two different habitats using a frequentist approach (via a parametric analysis of variance, or ANOVA), a Monte Carlo approach (via a simple randomization test), and a Bayesian approach. For the two last sections of this demo, we will assume that the data are compatible with a normal distribution.

Part I. Loading the Data Into R

The tab-delimited text file `Ant_Nest_Data.txt` contains the ant nest density dataset shown in **Table 5.1** of Gotelli & Ellison (2004, p. 108). Browse to the course website and right-click on this file to save it to a folder.

Now start R and type the following commands to load the ant nest dataset:

```
file_name <- "Ant_Nest_Data.txt"
nd <- read.table(file_name, header=T)
```

You can calculate the mean nest density for each habitat using the following code to filter each observation by its habitat type, either "Forest" or "Field":

```
mean(nd$NestsPerQuadrat[nd$Habitat=="Forest"
]) [1] 7
mean(nd$NestsPerQuadrat[nd$Habitat=="Field"]
) [1] 10.75
```

The mean density of ant nests in the sample from the "Field" habitat was 10.75 and the mean of the sample from the "Forest" habitat was 7.00. In the next three parts of this demo, we will use Frequentist, Monte Carlo, and Bayesian approaches to determine whether there is evidence that these nest densities are clearly different from each other.

Part II. Parametric Approach: Analysis of Variance (under Frequentitst Approach)

Analysis of variance (ANOVA) is a common parametric test used when your predictor variable (or variables) is categorical and your response variable is continuous. In Methods I you already learned about ANOVA in detail. You can also review Chapter 10 of the Gotelli & Ellison (2004).

ANOVA belongs to a family of statistical tests known as *linear models*. The R function *lm* is used to fit linear models (Chapter 9 in Gotelli & Ellison). We will use the *lm* function to specify a linear model:

```
model <- lm(NestsPerQuadrat ~ Habitat, data= nd)
```

This command creates a linear model that relates our response variable (NestsPerQuadrat) to our categorical predictor variable (Habitat) and stores the result in a new object named model. The tilde character (~) is a special R operator used in the specification of models. Once we have defined our model, it is a simple matter to use the built-in *anova* function to summarize the analysis of variance on our data:

```
> anova(model)
Analysis of Variance Table

Response: NestsPerQuadrat
          Df Sum Sq Mean Sq F value  Pr(>F)
Habitat    1 33.75   33.750  8.7805 0.01806 *
Residuals  8 30.75    3.844
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You should be familiar with details of this output (which is known as an *ANOVA table*). The F-ratio of approximately 8.78 is the same as that reported in Gotelli & Ellison (2004, p. 119), as is the *P*-value of approximately 0.018 (see output highlighted in red above). We do not recommend the use of *P*-values neither ANOVA summary tables for inference since they provide limited information and have been widely criticized. We instead encourage you to use information theory and to plot your models and data when evaluating your model outputs and to provide their average effects sizes and a measure of their uncertainty (see below).

Part III. Monte Carlo Approach: Randomization Test

One of the major disadvantages of ANOVA and many other parametric analyses is that they assume that the data are sampled from normal distributions. Yet, an examination of the box plots for the nest density data (grouped by habitat type) suggest that these data may not be normally distributed. Verify this for yourself using the *boxplot* function.

One of the major advantages of the Monte Carlo approach to statistical analysis is that it doesn't require you to assume that the data are sampled from a normal distribution (or any other specific probability distribution, for that matter). It only assumes that the data are drawn from random, independent samples (Gotelli & Ellison 2004, p. 115).

Another critical assumption of the Monte Carlo method is that the test statistic adequately represents the pattern we are interested in testing. For our current example, we define our test statistic as the difference between the means of the "Forest" and "Field" samples. We can calculate and display the observed value of this test statistic with the following lines of code:

```
mean_forest<- mean(nd$NestsPerQuadrat[nd$Habitat=="Forest"])
mean_field <- mean(nd$NestsPerQuadrat[nd$Habitat=="Field"])
diff_obs <- mean_forest - mean_field
diff_obs [1]
-3.75
```

Our observed test statistic value of -3.75 matches that reported for DIF_{obs} on p. 111 of Gotelli & Ellison (2004).

Now we need to construct a "null distribution" of our test statistic by reshuffling the habitat labels ("Field" or "Forest") and then randomly reassigning them to the nest density observations. The following line of code uses the *sample* function (without replacement) to randomly re-order the habitat labels:

```
> Habitat_random <- sample(nd$Habitat, length(nd$Habitat))
```

The following pair of commands will create a new data frame object, random_data, containing the original nest density observations, but with the randomly shuffled habitat labels assigned to them:

```
> Nests_random <- nd$NestsPerQuadrat
> random_data <- data.frame(Habitat_random, Nests_random)
```

We can compare this to the original data in **Table 5.1** (Gotelli & Ellison, p. 108) and verify that the labels have indeed been randomized (NOTE: Due to random sampling, your data will most likely be different than that shown here. There is also a very small chance that your data may have the same labels as the original data in Table 5.1):

```
random_data
   Habitat_random Nests_random
1          Forest            9
2          Forest            6
3          Forest            4
4          Forest            6
5           Field            7
```

```
6          Forest          10
7           Field          12
8           Field           9
9          Forest          12
10          Field          10
```

Next, we calculate our test statistic again by taking the value of the difference between the mean density in the "Forest" and "Field" habitats (the value you will obtain may differ from the realization in the example):

```
mean_forest <- mean(Nests_random[Habitat_random=="Forest"])
mean_field <- mean(Nests_random[Habitat_random=="Field"])
```

```
mean_forest - mean_field [1]
1.666667
```

To build our null distribution, however, we need to repeat this process many times, usually 5000 or more. Doing this by hand would be extremely time-consuming and tedious, but we can use a *for* loop inside an R script to do all the hard work for us. Download the `Ant_Nests_2019` from the course website and save it to your folder. Open the script and examine the block of code at lines after "## Part I Monte Carlo Analysis":

```
## create an empty array to hold the test statistic for each iteration
iterations <- 5000
diffs <- numeric(iterations)

## for each iteration, randomize the data and compute new test statistic
for (i in 1:iterations) {

       ## randomize the nest data
       Habitat_random <- sample(nd$Habitat, length(nd$Habitat))
       Nests_random <- nd$NestsPerQuadrat
       random_data <- data.frame(Habitat_random, Nests_random)

       ## compute the group means for the randomized data
       mean_forest <- mean(Nests_random[Habitat_random=="Forest"])
       mean_field <- mean(Nests_random[Habitat_random=="Field"])

       ## compute and save the test statistic
       diffs[i] <- mean_forest - mean_field
}
```

This block of code creates a null distribution for our test statistic of size `iterations` (which is defined earlier in the script and is initially set to a value of 5000) and stores it in the vector `diffs`. The following lines of code will display the distribution of our test statistic as a histogram and compute and display the tail probability, `P_Mc`:

```
## show a histogram of the test statistic
hist(diffs, xlab="DIF")
abline(v=diff_obs, col="red")

## calculate the tail probability
P_Mc <- length(diffs[diffs <= diff_obs])/iterations
P_Mc
```

If you run the script several times, you will get different values reported for `P_Mc`, the tail probability. One of the objections to Monte Carlo approach is that different analyses of the same dataset can lead to slightly different statistical results. But as the number of iterations approaches infinity, the tail probability will converge on a single number. Verify this yourself by, for example, changing the `iterations` variable to a value of 10000 and then running the script several times. What happens to the computed tail probability values? Notice the consistence between `P_Mc = 0.02`, the probability of values in the randomly generated distribution equal or more extreme than our `statistic`, and the `P-value = 0.02` of the frequentist ANOVA. What distribution is used for the generation of the "null" in the later approach? We find these two approaches disappointing because they focus on the comparison against a null and do not provide much information on the comparison of interest, the difference of nest density between habitats.

Part IV Bayesian Analysis.

In this course we encourage you to consider a different framework. We assume that you are ready to do statistical inference in a fresh way. We are convinced that Bayesian analysis helps to recognize and reduce many of the limitations of current scientific discovery. McElreath (2016) provides a great discussion on this topic.

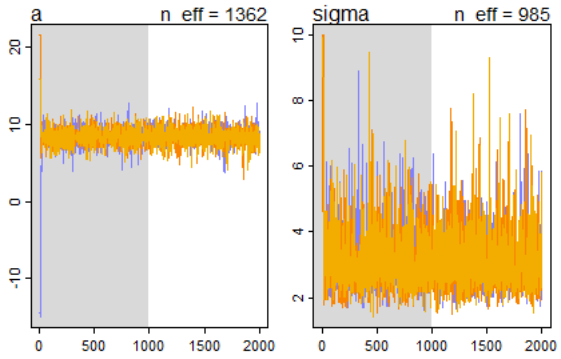
Here, we are interested on evaluating evidence for differences in the density of ant nests between two habitats. We will address this question through two complementary approaches using Bayesian analysis. We start by building two models. We will use the *Stan* program and the *rstan* and *rethinking* libraries in *R*. The code for the first model is below:

```
model1 <- map2stan(
  alist(
    NestsPerQuadrat ~ dnorm(mu,sigma),
    mu <- a,
    a ~ dnorm(0,10),
    sigma ~ dunif(0,10)
  ),
  data = nd, chains =3,
)
```

All Bayesian models have three parts, the likelihood distribution of the data, the model specification and the priors of the parameters. In our first model we do not include any factors explaining the variation among the data. We assume a normal likelihood *dnorm(mu,sigma)* for these data, describing the mean and variance of our single parameter

"*a*". We propose diffuse priors for these parameters. These priors summarize any knowledge that we have on our data. In this case, we assume very little information. This is not the best strategy as we see in other demos of this class. The priors and the likelihood are used to estimate the posterior distribution of the parameters of our model. The prior for the mean is normally distributed and has a very large variance, the prior for sigma must have a distribution that does not include negative numbers and it is also diffuse.

After implementing this model, we require to check our generated parameters. We use two sources to do this. First the plot of the generating process indicates that the chains mixed fine and the estimates appear reasonable.



Second, we use the function *precis* to request a summary of the results. The value of "1" of the Rhat confirms the consistency between the generating chains. We will use this model to compare with another model that includes habitat as an explanatory variable. This model estimates an average of 8.5 nests overall that is commensurate to the mean for all the sites.

precis(model1, digits=2)

```
      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
a     8.48   1.06       6.65       9.96  1362    1
sigma 3.10   0.91       1.93       4.36   985    1
```

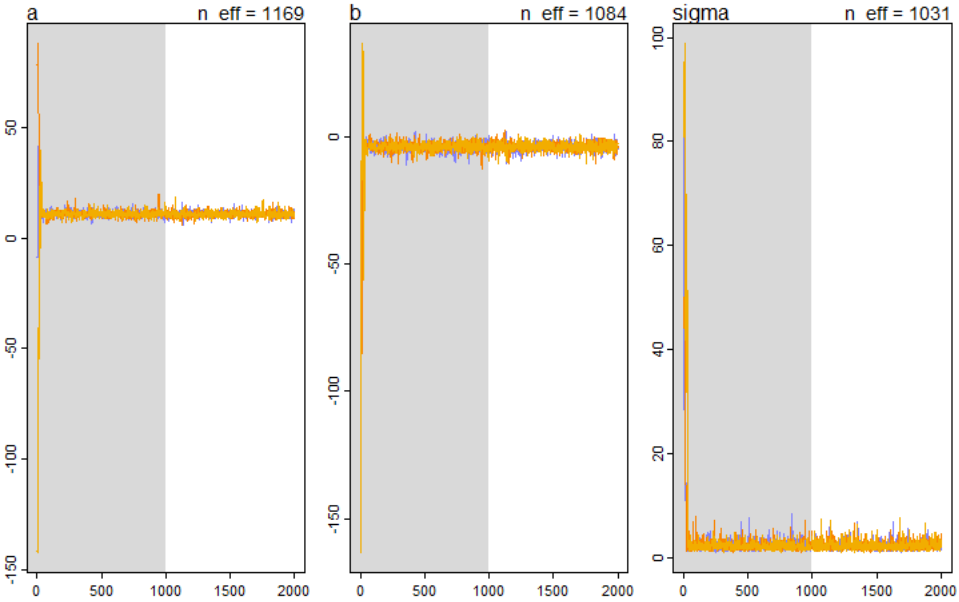
For the second model, we create a dummy variable and specify the linear model as follows. This time we use habitat as an explanatory variable for the variation of ant density.

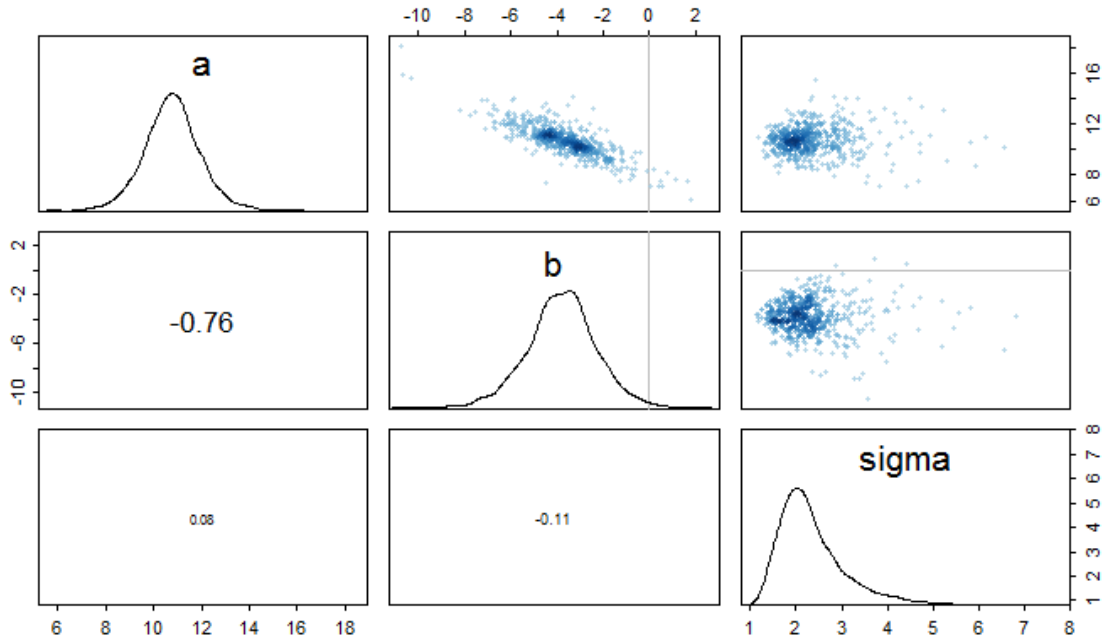
```
nd$Forest <- 0
nd$Forest[nd$Habitat=="Forest"] <- 1

model2 <- map2stan(
  alist(
    NestsPerQuadrat ~ dnorm(mu,sigma),
    mu <- a + b*Forest,
    a ~ dnorm(0,100),
```

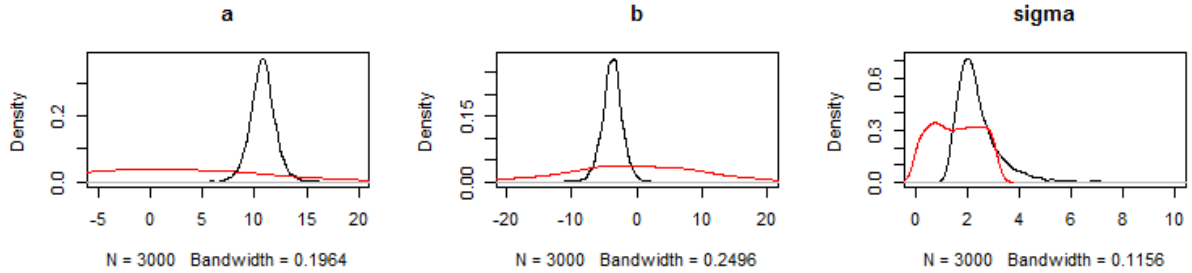
```
    b ~ dnorm(0,100)
    sigma ~ dunif(0,100)
  ),
  data = nd,chains =3,
)
```

This time we have three parameters, the average ant density for the field, the difference with the density in the forest and the variance of the data. We do not see any problem on the generating process.





We re-plot the distribution of the priors (in red) and the posterior distributions (in black).



We use the function precis to call the summary parameters:

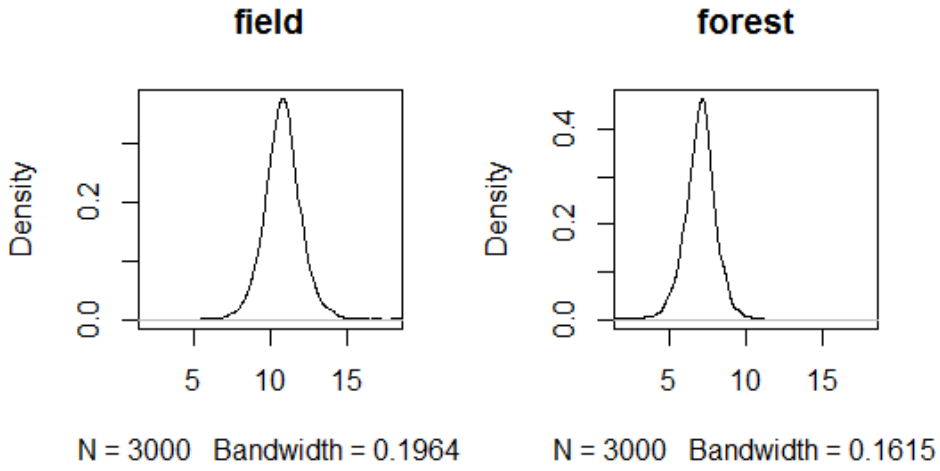
```
precis(model1, digits=2)
```

```
       Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
a     10.79   1.23       8.89      12.66  1169    1
b     -3.82   1.59      -6.37      -1.45  1084    1
sigma  2.39   0.81       1.30       3.42  1031    1
```

When we compare the two models using information theory, we found that model 2 is much more likely to explain the variation in ant density (0.92 vs 0.08).

```
> compare(model1,model2)
       WAIC pWAIC dWAIC weight   SE  dSE
model2 46.1   2.0   0.0   0.92 2.70   NA
model1 51.0   1.3   4.8   0.08 2.84 3.38
```

Given the posterior distribution for the probability of the difference ("*b*"; values lower than 0 are most probable = 0.99) it clearly indicates differences in ant nests between habitats. We can also ask ourselves, given the circumstances of our sampling and our model what is the probability that there are 4 nests less in the forest compared to the field. The answer is $P = 0.44$. We can also generate predictions of the average density of ant nests per habitat.



NOTE: all the materials for this demo can be found at:

https://sciences.ucf.edu/biology/d4lab/methods-2/

References

Gotelli and Ellison. 2004. A Primer of Ecological Statistics. Sinauer.

McElreath, R.M. 2016. Statistical Rethinking: a Bayesian course with examples in R and Stan. Chapman and Hall.