Week 09: Comparative Methods

Due April 3^{rd}

I. General

Type up all of your work in a text editor.

Before you begin, make a new folder in Tasks called Task_08, and save an empty file named task08.r in that folder.

You must also make a separate "Project" folder within Tasks. Inside your Project folder, make "code", "data", and "text" subfolders. By Friday, you should have some data in the data folder, some code to read the data in to `R` in the code folder, and a document with a *single sentence* describing your hypothesis.

When you're **done** with this assignment, turn it in by (1) saving your text document, (2) opening your Terminal or GitBash, (3) navigating to the appropriate directory using `cd` and (4) typing:

`git add -A` (enter)
`git commit -m "Task 08"` (enter)
`git push -u origin master` (enter)

The main goal this week is a bit meta. Almost everyone is going to end up analyzing data along a phylogeny for their final project. So we're going to walk through some functions that allow you to do that clearly and cleanly.

II. Your Code!

```r
# We're going to unite data and phylogenies into one grand thing

# First, set your working directory to your working folder
# Now, we need to tell R that we want to use functions from the phytools library
# Next, read the phylogeny into R using the read.tree() function (see the last R assignment for how to do this)

plot(tree, type="fan")
# QUESTION 1: How many tips are there in the tree, and are branch lengths present?

# Now, let's read in the data. It's in the file svl.csv. SVL stands for "snout-vent length", and it's a standard measure of how long a vertebrate is
data <- read.csv("https://jonsmitchell.com/data/svl.csv", stringsAsFactors=F, row.names=1)

# QUESTION 2: What kind of object is ‘‘data’’? What are its dimensions?

# We're going to convert the object data into a vector
svl <- setNames(data$svl, rownames(data))

# Now, we have an evolutionary tree of these lizards, and the body size of
   the living species. Let's RECONSTRUCT the ancestral states. That is, let'
   s estimate how large the ancestors were using the phylogeny and the data.

Ancestors <- fastAnc(tree, svl, vars=TRUE, CI=TRUE)

# Examine the Ancestors object and the help file for fastAnc
# QUESTION 3: Where are the estimated values stored? What is the CI95
   element?
# QUESTION 4: What are two assumptions made in the estimation of the
   ancestral states using fastAnc?

# Let's plot all of this together!

# First, we'll plot the tree
par(mar=c(0.1,0.1,0.1,0.1))
plot(tree, type=”fan”, lwd=2, show.tip.label=F)

# Now we'll put points instead of names at the tips. The size of the points
   will be proportional to the size of the lizard. To do this, we'll use
   tiplabels(), but tell it to plot points instead of words (using the pch
   parameter), and to scale the points by the size of each lizard using the
   parameter cex. We'll make sure the tips match the points by reorganizing
   the svl vector using the tip labels of the tree using square brackets.
tiplabels(pch=16, cex=0.25*svl[tree$tip.label])

# Now, we'll add the ancestral states to the tree using nodelabels(), as
   each node
nodelabels(pch=16, cex=0.25*Ancestors$ace)

# Looking at the points, you can see SOME variation in size, but it's hard
   to tell how large or small any given tip is relative to its ancestors or
   even to other tips. So let's try visualizing this data in a different way
   .

obj <- contMap(tree, svl, plot=F)
plot(obj, type=”fan”, legend=0.7*max(nodeHeights(tree)), sig=2, fsize=c(0.7,
    0.9))

# Now, we're going to add some fossils in using data from ‘‘Amber fossils
   demonstrate deep-time stability of Caribbean lizard communities’’ by Emma
    Sherratt et al. 2015 in PNAS

# We'll take the measurements and the node closest to each fossil species
fossilData <- data.frame(svl=log(c(25.4, 23.2, 17.7, 19.7, 24, 31)), tip1=c(
   ”aliniger”, ”aliniger”, ”occultus”, ”christophei”, ”cristatellus”, ”
```

```
    occultus”), tip2=c(”chlorocyanus”, ”coelestinus”, ”monticola”, ”cybotes”,
     ”angusticeps”, ”angusticeps”))

# What we'll do is: for each fossil, find what node corresponds to the Most
    Recent Common Ancestor (MRCA) of the pair of tips in the dataframe.

# QUESTION 5: I've set up most of the for loop. You'll need to add the
    actual for() line, and close the loop in the approprioate places.
fossilNodes <- c()
nodeN <- c()

Node <- fastMRCA(tree, fossilData[i, ”tip1”], fossilData[i, ”tip2”])
fossilNodes[i] <- fossilData[i, ”svl”]
nodeN[i] <- Node

names(fossilNodes) <- nodeN

# Now we estimate the ancestral states, but this time use the fossils to
    constrain the estimates at the relevant nodes
Ancestors_withFossils <- fastAnc(tree, svl, anc.states=fossilNodes, CI=TRUE,
     var=TRUE)

# QUESTION 7: How do fossils, in general, change the estimated ancestral
    sizes? (Hint, making a plot of the two different ancestral state
    estimates against one another will help!)

# QUESTIONS 8 - 10: Now, I want you to determine if the assumptions you
    identified in question 4 hold for these data. The way you'll want to go
    about this is by installing a library called ‘‘geiger’’, and use a
    function in that package called ‘‘fitContinuous’’. I want you to 1) fit
    several different models of evolution to these data, 2) determine which
    model fits the data best, and 3) describe whether or not that model is
    different from what fastAnc assumes (look at the help page for fastAnc).

# This will require that you read the help pages for the pertinent functions
    , and possible some other functions. Google will be your friend here.
    Akaikie Information Criteria (AIC) will be an important facet. We'll talk
     about what AIC is in class, but for this assignment all you need to know
     is that the model with the LOWEST AIC is the best-fitting model.
```