

Comparative Methods Workshops

Chris Mitchell

Contents

1	Welcome	5
1.1	For students	5
2	Introduction	7
2.1	What is the comparative method?	7
2.2	Tree thinking	8
3	R recap	11
3.1	Basics	11
3.2	Errors	15
3.3	Google	15
3.4	Stealing	16
4	Phylogenetic trees and where to find them	17
4.1	Phylogeny	17
4.2	Building trees	18
4.3	Locating trees	18
5	Plotting trees in R	19
5.1	Phylogenies in R	19
5.2	Importing your tree	20
5.3	ggtree	21
5.4	Adding images to trees	25
5.5	Further info	32
6	Bibliography	33

Chapter 1

Welcome

Welcome to the online support materials for the Comparative Research Group at the University of Liverpool. The CRG is made up of staff and students engaging in comparative research across various areas of evolutionary biology.

1.1 For students

The materials here are intended to support you through your LIFE363 honours project. For this project you will be performing a comparative study (see **chapter 1** for more information) on an area of your choosing. At first, this is a daunting task but developing your own research here is excellent experience and gives you the opportunity to research an area that really interests you.

The vast majority of statistics here are performed in R [R Core Team, 2019]. You were introduced to R in LIFE223 as a powerful and flexible tool for statistical analysis. **Chapter 2** contains a brief refresher on some of the basics of R in case you need it. For more detailed recaps, please revisit your materials from LIFE223 as some of the code you wrote is likely to be useful this year!

Throughout this book you will see examples of R code and output like this.

```
print(hidden.message)
```

```
[1] "Forty-two"
```

The code can be copied and pasted into your own version of R as you see fit. However, I would recommend that for the first time you are using a piece of code, type it out for yourself. This will help you get to grips with what each argument means.

You will also see some interactive R windows where you can enter your code

directly into this book and an online version of R will run it. This should give you an opportunity to learn more complex things and develop your R skills dramatically.

The rest of the book is populated with workshops and materials to help you learn specific comparative statistical methods. Some of these will be extensions of what you already met in LIFE223. **Chapter 6** looks at phylogenetically controlled ANOVA and **chapter 14** is all about phylogenetic regression.

Other methods may be entirely new to you such as ancestral state reconstruction (**chapters 7 - 10**) or path analysis (**chapter 15**). Don't be intimidated by this. All the code you need is gathered here and will remain available as long as you need it.

Chapter 2

Introduction

This chapter contains a very brief overview of the research we do in the **Comparative Research Group**. Taxonomically, the work done by group members is extremely broad. We've had projects on primates, octopuses, domestic mammals, birds and more! Here is a sample of titles from previous students.

- Identification of a cognitive niche in benthic octopods and possible areas for future study on cephalopod intelligence.
- Evolutionary precursors for the domestication of Artiodactyla.
- You are not what you eat: Lack of morphological convergence in beak and body size between the nectarivorous avian families Trochilidae and Meliphagidae.
- Investigating how lifestyle factors affect lifespan in reptiles.
- Ecological processes causing encephalisation in Madagascan lemurs.

2.1 What is the comparative method?

The comparative method is a catch-all term for a suite of approaches that involve using comparisons to answer scientific questions. In evolutionary biology, the comparative method refers to making comparisons between species or populations in order to identify patterns and relationships between traits of interest. Used correctly, this approach can be very powerful and allows us to ask large-scale questions about evolutionary patterns, adaptive processes and coevolutionary relationships.

The most basic kind of comparative study is comparing one species or lineage to another. For example, a recent paper made waves in the paleontology community by demonstrating (after years of debate) that *Spinosaurus aegypticus* lived an aquatic lifestyle [Ibrahim et al., 2020]. The analysis centered around some newly recovered tail vertebrae with extremely long (1m!) spines. The tail of

Spinosaurus was compared to other animals including terrestrial theropods like *Allosaurus* and semi-aquatic tetrapods such as the crocodile. This comparison showed that the *Spinosaurus* tail was indeed specialised for powerful propulsion through the water (like a crocodile), seemingly settling the debate over whether any non-avian dinosaurs invaded the water.

Other comparative studies take data gathered from many species and search for patterns within that group. Studies like this rely a great deal on work done by others. For example, Simon Reader and colleagues [2011] carried out an extensive literature search looking for examples of five behavioural traits in many species of primate in over 4000 articles published over 75 years. The resulting database included examples of innovation, social learning, tool use, extractive foraging and tactical deception and was used to demonstrate a correlation between these behaviours and brain size, providing evidence of a general intelligence factor in primates similar to that in humans.

2.2 Tree thinking

Comparative studies can be great but there is a problem. In LIFE223 you learned about statistical assumptions. One of the most common and important assumptions of most statistical tests is that data are independent. To run a good comparative study we need to know that the data points we have are independent of each other. In evolutionary biology, we know that this isn't the case!

All living things exhibit a pattern of relatedness which depends on how much shared evolutionary history they have. For example, chimpanzees and human beings diverged about 6-7 million years ago. This means that they have much more shared evolutionary history than chimpanzees and *Spinosaurus* which are separated by hundreds of millions of years.

The best way of visualising this pattern of relatedness is with a phylogenetic tree.

The extant species are displayed on the **tips** of the tree and are connected to each other according to the degree of relatedness by the **branches**. Figure 2.1 shows us the pattern of relatedness of 42 cetacean species. If we wanted to use these species in a comparative study to investigate the evolutionary history of the group, we would not have independent data points. This means that the assumptions of most statistical tests would be violated and we couldn't trust the results!

This is where phylogenetics comes to the rescue. We can use the pattern of relatedness described by the phylogeny to control for the non-independence of data points. To show you what I mean, let's consider body size in those 42 species of cetacean. If we were to show the distribution of body size in the

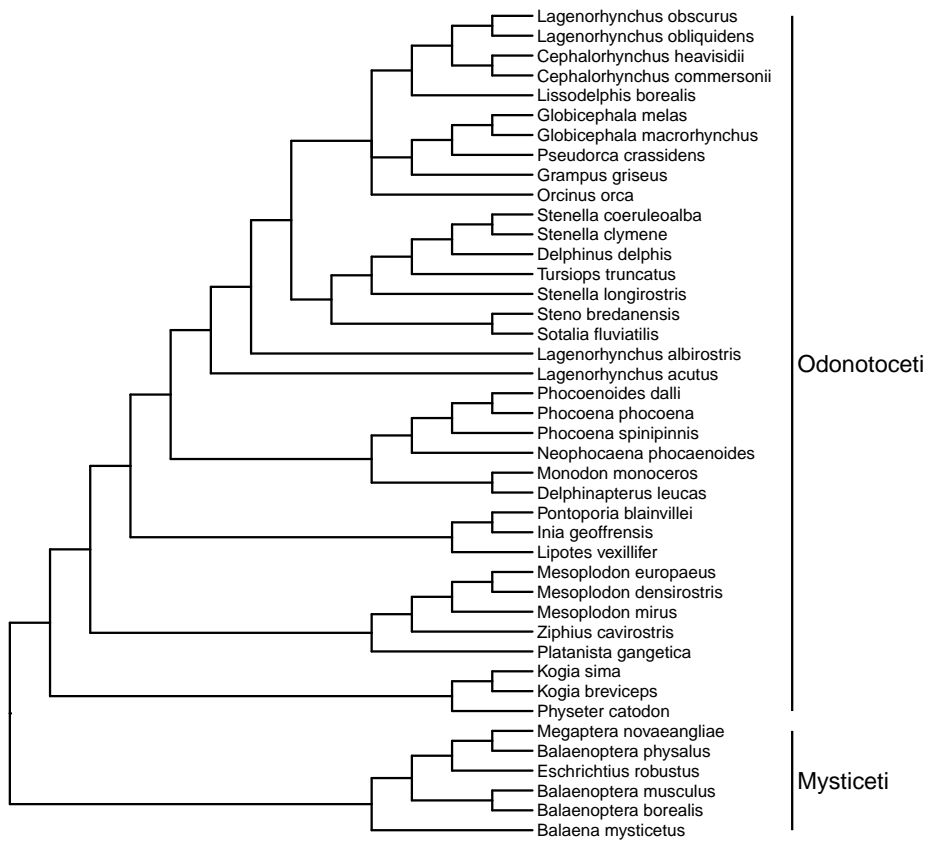


Figure 2.1: A cladogram of 42 cetacean species.

group, we would see that the vast majority of the largest sizes are found in the mysticetes whereas the smaller species tend to be odontocetes. If we viewed these data points as all independent we might say that very large bodies have evolved 7 times in the group (once for each mysticete and once for the sperm whale) whilst small body size has evolved in all the other species (35 times).

In fact, the close relatedness of 6 of the large bodied species suggests that large body size evolved once and not independently for each of these species. Their shared evolutionary history explains why their traits (body size in this case) are so similar. The seventh example of a large body (sperm whales) does not share very much history with the other 6 and this may be of some interest to us. It suggests an independent evolution of large body size and potentially something of interest to us as researchers.

So hopefully you can see how taking phylogeny into account can be illuminating. For a broader (and much more useful) introduction to phylogenetics and its use in evolutionary biology, check out these sources:

- Tree Thinking: An Introduction to Phylogenetic Biology [Baum and Smith, 2012]
- The Comparative Approach in Evolutionary Anthropology and Biology [Nunn, 2011]

Chapter 3

R recap

In LIFE223, we taught you how to use R for statistical analysis and visualising data. This chapter will contain a basic overview of some of the things from 223 that you may find useful as we proceed. You only need to bother with this if you are new to R or have blocked it from your memory since you last used it.

3.1 Basics

R works well as a calculator.

```
6*7
```

```
[1] 42
```

However, R is capable of a great deal more than just simple mathematical operations like multiply and divide. It also has functions that can calculate some common descriptive stats like mean and standard deviation.

```
mean(x)
```

```
[1] 41.97773
```

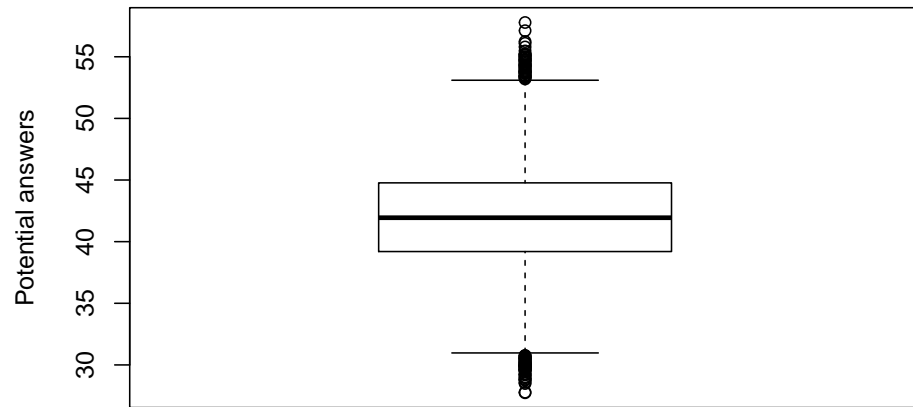
```
sd(x)
```

```
[1] 4.185262
```

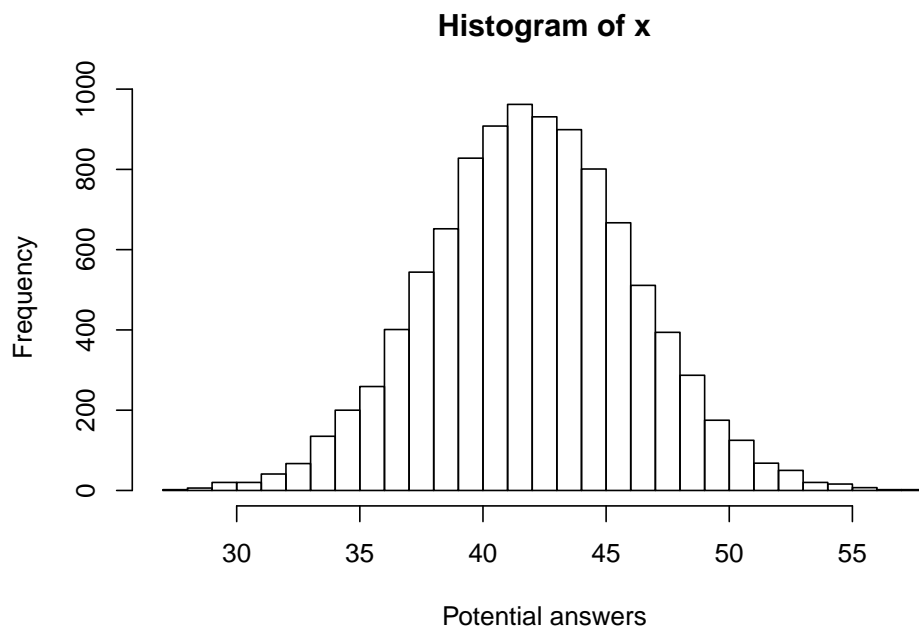
3.1.1 Plotting

R is also a very flexible graphical tool. From LIFE223, you probably remember a few basic plotting functions. Each function in R has arguments that can be added to label axes or change point size as you can see in these plots.

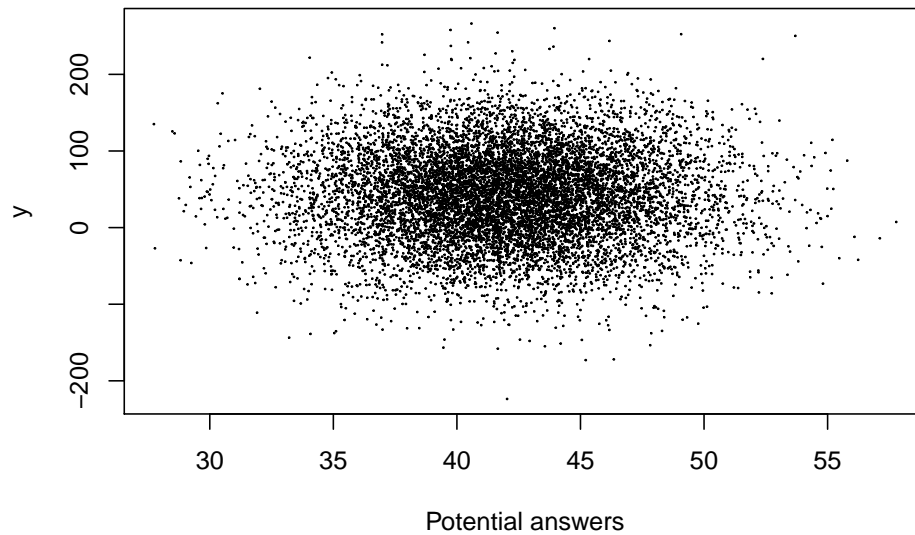
```
boxplot(x, ylab = "Potential answers")
```



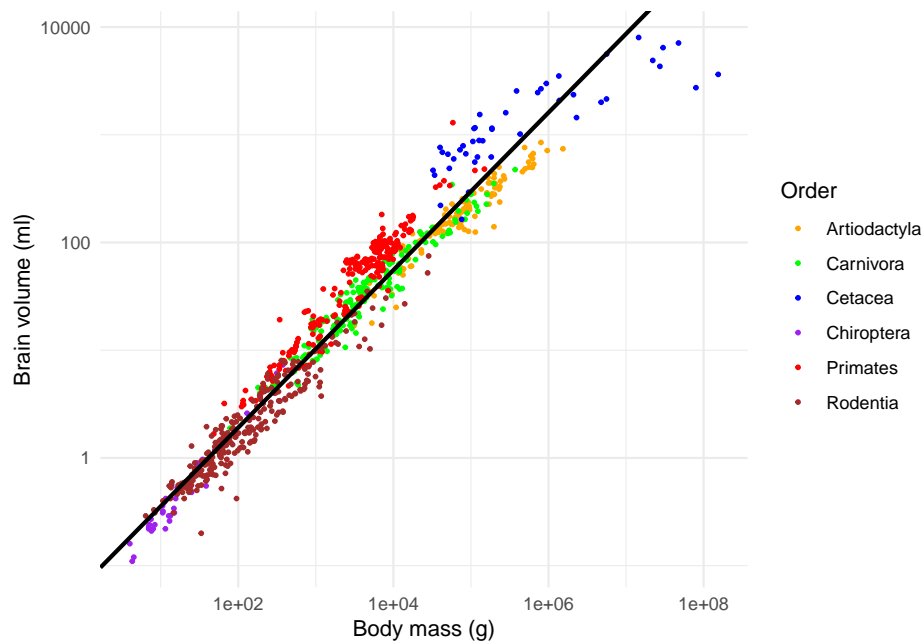
```
hist(x, xlab = "Potential answers", breaks = 25)
```



```
plot(x, y, xlab = "Potential answers", pch = 19, cex = 0.1)
```



For much of this book, I will actually be doing most plotting in a package called **ggplot2**. This package has a slightly different syntax to get used to but the increased flexibility you have will be a good payoff. Plus the plots look quite nice.



3.1.2 The working directory

The working directory is the folder on your computer where R's attention is focused. This is where you should store any files you need R to open. You can find out the path of the current working directory using the function `getwd()`

```
getwd()
```

```
[1] "/Users/chrismitchell/Google Drive/University of Liverpool/GitHub Stuff/bookdownCR"
```

If this isn't the folder we want as our working directory, we can just as easily change it with `setwd()`

```
setwd("~/Desktop/My R Folder")
```

If you are using RStudio there is also a shortcut to do this in the *Files* pane (usually bottom right). Use this pane to navigate to your chosen directory and then use the drop down menu under *More* (look for a blue cog) to set the current folder as your working directory.

If you aren't using RStudio, I'd strongly suggest you start. It's much more user friendly than base R.

3.1.3 Loading data

Data comes in many forms and R is capable of reading most of them if you know the correct functions. One of the most common formats is *comma separated values*. This has the file extension `.csv` at the end of the filename. If you open a `.csv` file with MS Excel or Numbers, you will see that it usually looks much like a spreadsheet. To load a `.csv` data file into R, use the function `read.csv()` as shown here.

```
data <- read.csv(file = "DATAFILE.csv")
```

For other data formats, you may require a different function. For example, data may be provided as a text file (extension `.txt`). In this case, you need `read.table()`. Note that with this function you need to specify that your data has a header (a top row with names for columns) whereas `read.csv()` assumes this by default.

```
data <- read.table(file = "DATAFILE.txt", header = TRUE)
```

3.1.4 Subsetting

Let's say I want to subset my data based on a certain condition. I can achieve this multiple ways but one of the simplest is the function `subset`.

```
newdata <- subset(data, species == "Homo sapiens")
```

This function takes a subset of the object *data* and applies the rule that the value of each row in the column *species* be *Homo sapiens*. Thus it extracts the lines of data that are from human beings.

3.2 Errors

Error messages are a part of life with R. You are not expected to be able to interpret every single one immediately and you definitely shouldn't panic or give up when you get one.

Here's a basic error message:

```
data <- read.csv("mydaat.csv")
```

```
Warning in file(file, "rt"): cannot open file 'mydaat.csv': No such file or
directory
```

```
Error in file(file, "rt"): cannot open the connection
```

The message tells me that R “cannot open the connection” and no such file exists. This means that R cannot find the file I was looking for in the current working directory. It could be because I haven't set the correct working directory or the file is there but in a different format. In this case, the error has appeared because I have spelled the name incorrectly. I have sent R looking for a file called *mydaat.csv* instead of *mydata.csv*. Always remember that R is a useful idiot and will only do exactly what you tell it to do!

3.3 Google

The most important skill you need for using R is the ability to use Google (other search engines are available). It may seem odd but almost any problem you will ever encounter with R can be solved by a quick Google search.

If you come up against a confusing error message, copy and paste the message into Google. You will quickly land on one of the forums where someone else has asked about the same error message. The odds are pretty good you'll discover an explanation for the problem there.

If you don't know how to do something, pop the name of what you want to do into Google and add “in R” at the end and there will almost always be a tutorial on the first page of results with exactly what you need.

Seriously, Google is your strongest ally here. The community of R users has populated the internet with endless advice and guidance for every level from beginner to the most advanced of users. That brings me to my next point...

3.4 Stealing

If imitation is the greatest form of flattery then learning to code in R is just about the most flattering thing you can do. The internet is teeming with examples of R code for all kinds of purposes including in this very book. Take it without thinking twice.

You will have achieved a pretty good level of skill in R when you can take someone else's code and edit it for your own purposes. This is the **core skill** of R and once you can do that, you'll be unstoppable.

Chapter 4

Phylogenetic trees and where to find them

This chapter is a brief overview of some key concepts that may be useful when performing comparative research.

4.1 Phylogeny

Phylogeny is the term used to describe the evolutionary history of a group of species. The most common representation of phylogeny is a phylogenetic tree. There is a lot of terminology around phylogenetic trees. Here we will start with the very basics that will come up a lot in this book.

The **tips** of the tree represent the species/populations/individuals described by the tree. The **branches** of the tree represent the pattern of relationships between species. The **nodes** of a tree represent the most recent common ancestor of the lineages that diverge from that node. A **clade** is a monophyletic grouping of lineages. A grouping is **monophyletic** only if all members of that group descend from a common ancestor to the exclusion of others. for example, humans and apes form a monophyletic grouping but humans, apes and parrots do not.

Here is an example of a phylogenetic tree displaying the relationships of modern dog breeds taken from a nice paper investigating the evolutionary history of the domestic dog [Parker et al., 2017]

The tree shown above is a **cladogram** meaning that the lengths of the branches do not carry any real meaning. This tree is only useful for interpreting the relatedness of the species. It does not give us any information about the amount of evolutionary change or the amount time between nodes.

By contrast, the following tree has **branch lengths**. The tree is based on genetic analysis of 173 species of hymenoptera (bees, ants and wasps) [Peters et al., 2017]. In this tree, the branch lengths represent time (in millions of years) as calculated from analysis of over 3,000 genes and calibrated using fossils.

Branch lengths do not always represent evolutionary distance as time. In some cases, evolutionary distance is represented as the amount of change on each branch. The next tree was built based on a brain development gene (MCPH1) in cetaceans (whales, dolphins and porpoises) [McGowen et al., 2011]. On a tree like this, longer branches indicate more character changes along the branch. In this case the character changes will be changes in genetic sequence but for other trees it may be morphological characters, protein sequences characters or a combination.

4.2 Building trees

Developments in the field of phylogenetics have meant that there are many ways to construct a phylogeny. Many of the modern methods are highly sophisticated and for now, these are not the subject of this book. However, it may help you to have a brief introduction to the logic behind building a phylogeny.

4.3 Locating trees

4.3.1 File format

Chapter 5

Plotting trees in R

5.1 Phylogenies in R

From LIFE223, you know R as a powerful statistical tool. You will also be aware that it is an incredibly flexible tool for plotting data. In this workshop, we will be working with phylogenies in R and manipulating them to produce informative plots.

5.1.1 Packages used

In this section we'll mostly be using a package called `ggtree` [Yu et al., 2017, 2018]. To install it, we need another package called **BiocManager** [Morgan, 2019].

```
install.packages("BiocManager")
BiocManager::install("ggtree")
library(ggtree)
```

```
library(ggtree)
```

We will also need to use `phylobase` [R Hackathon et al., 2019], `ggimage` [Yu, 2019] and it would help to have the tidyverse packages loaded [Wickham, 2017] since we'll be using the syntax of `ggplot2`. If you get an error message, make sure the packages are installed first.

```
library(tidyverse)
library(phylobase)
library(ggimage)
```

5.2 Importing your tree

Let's start by importing a tree. Make sure your working directory is set to wherever you have saved the `tree_newick` file. If you run this line, you should see an object called "tree" appear in your global environment.

```
tree <- read.tree("tree_newick.nwk")
```

If we take a look at the structure of our tree object using the `str` function we can see that the tree is stored as an object of class **phylo**. If you are using a block of trees (more on this subsequent chapters) it will be an object of class **multiphylo**.

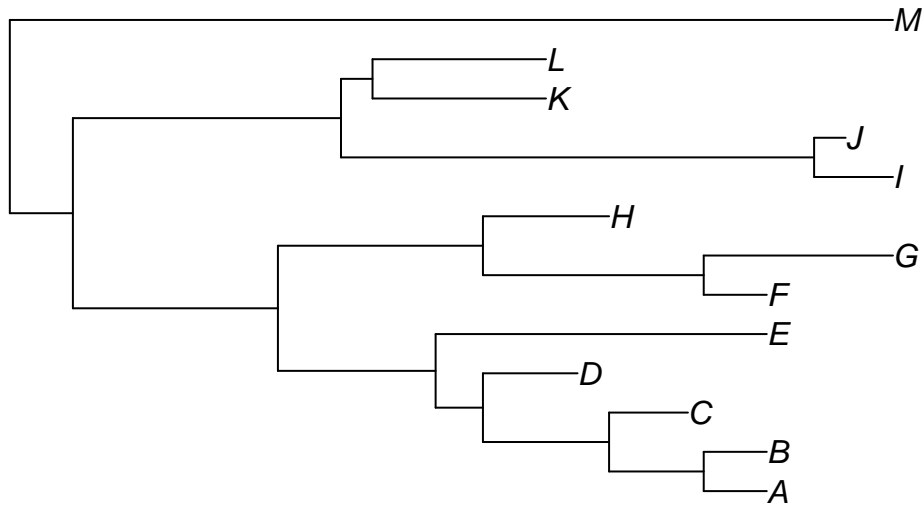
```
str(tree)
```

```
List of 4
 $ edge      : int [1:24, 1:2] 14 15 16 17 18 19 20 20 19 18 ...
 $ edge.length: num [1:24] 4 13 10 3 8 6 4 4 5 6 ...
 $ Nnode      : int 12
 $ tip.label   : chr [1:13] "A" "B" "C" "D" ...
- attr(*, "class")= chr "phylo"
- attr(*, "order")= chr "cladewise"
```

We can see a list of 4 elements of the tree object. The first (**edge**) contains the edges (also known as branches) of the phylogeny and their labels. The next is **edge.length** which contains the lengths of the branches if present (see **chapter 3** for more details). **Nnode** specifies the number of nodes and finally **tip.label** contains the labels of the tips. In this case, we just have letters for tip labels.

Things are often clearer when we plot them. We can do this for trees with the **plot** function in base R. This function is incredibly versatile and you should recognise it from LIFE223. Here we are using very different arguments.

```
plot(tree)
```



This plot is fine for a quick check to make sure the tree looks as we expected it to. Let's look at making a more attractive plot with `ggtree`.

5.3 ggtree

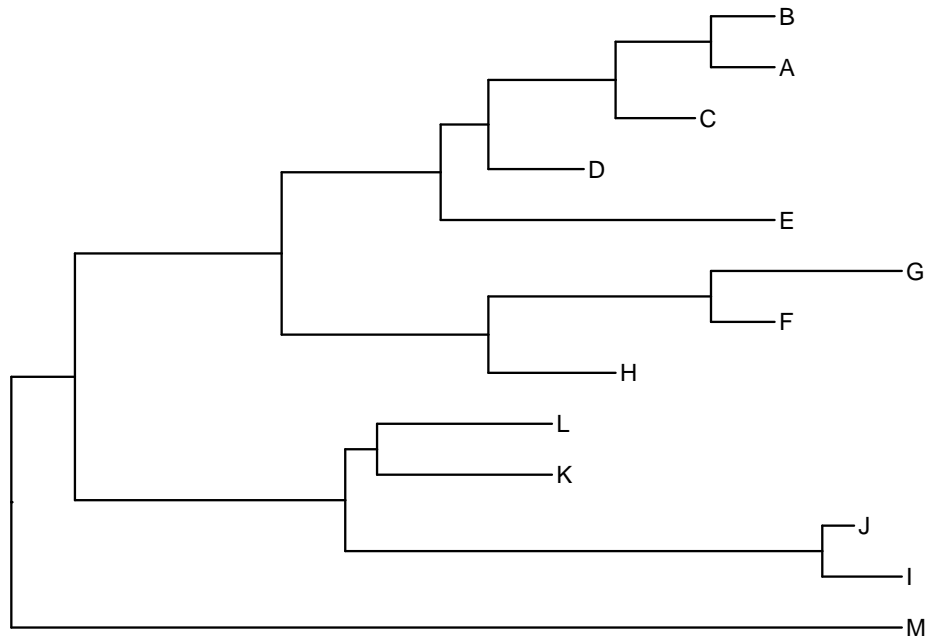
The package `ggtree` is an extension of **ggplot2**, a popular plotting package from the **tidyverse** family of packages. The syntax we'll be using here is a little different than what you may be used to so don't get intimidated. **ggtree** uses the same syntax as **ggplot2**. This works by creating layers (known as **geoms**) and plotting them over each other to build up the plot.

We'll start by using `ggtree` to plot our tree. Below is the base layer of the plot. There are many other options we can include to customise our tree. Try some out in this R window to see how they effect your plot.

5.3.1 Geoms

Geoms are new layers to plot on or alongside your tree. Now let's try plotting it whilst adding new layers. These geoms can be combined as you see fit. This gives you a lot of flexibility in how you plot your trees. For example, we can add a geom to include the tip labels for our tree.

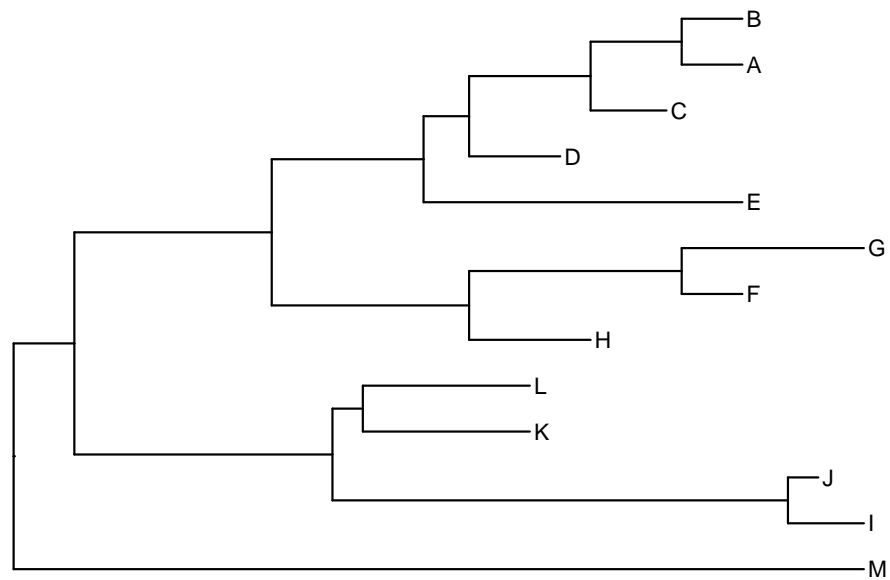
```
ggtree(tree) +  
  geom_tiplab()
```



And we can add a title

```
ggtree(tree) +  
  geom_tiplab() +  
  ggtitle("A phylogeny of letters. For some reason...")
```

A phylogeny of letters. For some reason...



There are many geoms you can use to add more information to your plot. Here are just a few that you may want to investigate.

```
geom_tiplab() #adds tiplables
geom_tippoint() #adds points at the tips
geom_nodepoint() #adds points at the nodes
geom_nodelab() #adds labels for nodes
geom_cladelabel() #adds labels for clades
```

5.3.2 Labelling clades

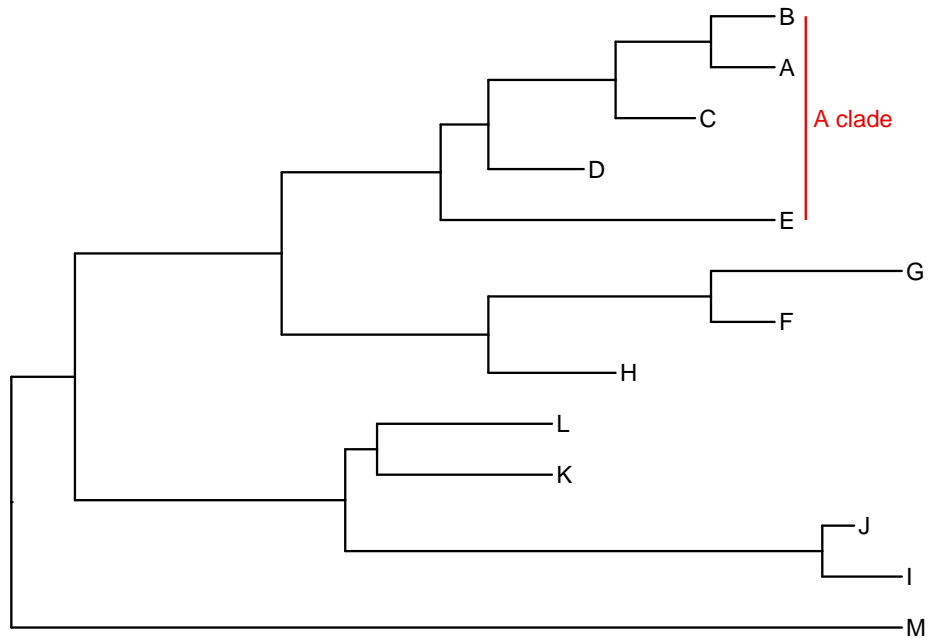
As an example of what you might like to do with ggtree, let's have a look at adding some labels to identify some clades on our tree. To label clades, we need to be able to identify the node of the most recent common ancestor. The function **MRCA** in the package **phylobase** [R Hackathon et al., 2019] tells us that the common ancestor of C and E is node 17.

```
phylobase::MRCA(tree, tip = c("C", "E"))
```

```
[1] 17
```

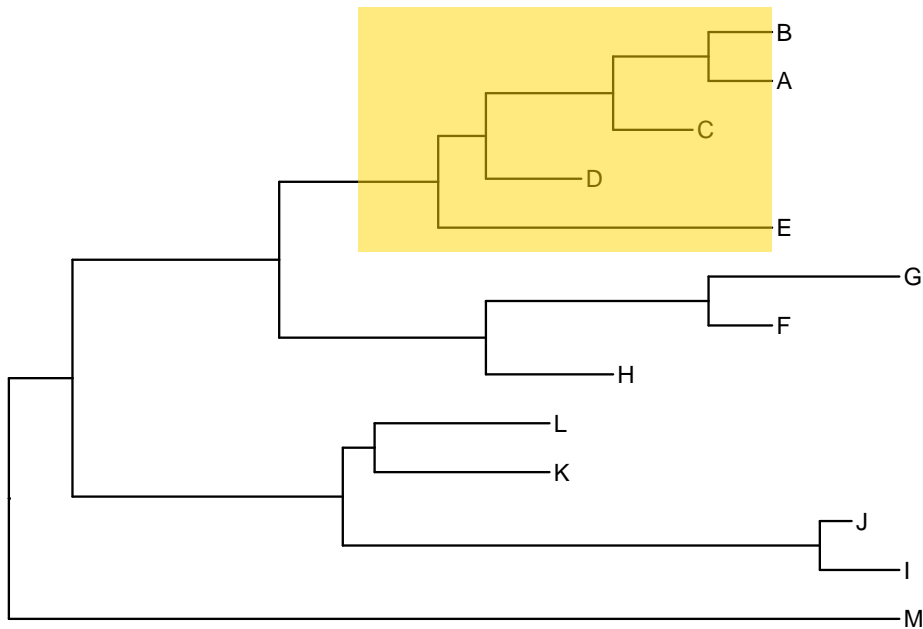
We can now use the **geom_cladelabel** geom to add a simple label for the clade descended from the appropriate node. Take note of the arguments I've added to customise the geom. You may want to play around with these options yourself to see how they work.

```
ggtree(tree) +
  geom_tiplab() +
  geom_cladelabel(node=17, label="A clade",
                  color="red2", offset=1)
```



Pretty good but there are other options. This is a matter of personal preference. You may prefer to overlay a translucent rectangle over your clade of interest.

```
ggtree(tree) +  
  geom_tiplab() +  
  geom_highlight(node=17, fill="gold")
```

Use the R window below to experiment with the available geoms in `ggtree`. Find a combination that suits you and your tree.

5.4 Adding images to trees

As you probably noted in chapter 3, adding images to a plot is an excellent way to annotate your tree. The `ggtree` package can do this as you can see here.

This phylogeny is annotated in a number of useful ways. The tip labels describe cephalopod families. The superorders (octopodiformes and decapodiformes) are highlighted by gold and red rectangles as well as a bar across the tips. this demonstrates how multiple geoms can combine to make a plot easy to interpret.

The most interesting thing for our purposes are the silhouettes at the root of each superorder. The octopodiformes have an octopus and the decapodiformes have a squid as example taxa from within the superorder.

5.4.1 Phylopic

The silhouettes used for that plot are from a website called Phylopic. Phylopic provides open source biological silhouettes that are free to use. We're now going to look at how to get these silhouettes and use them to annotate our trees.

Let's start with loading an example tree. This one is a primate tree courtesy of Randi Griffin. You'll notice that I'm loading this tree using a url. This is

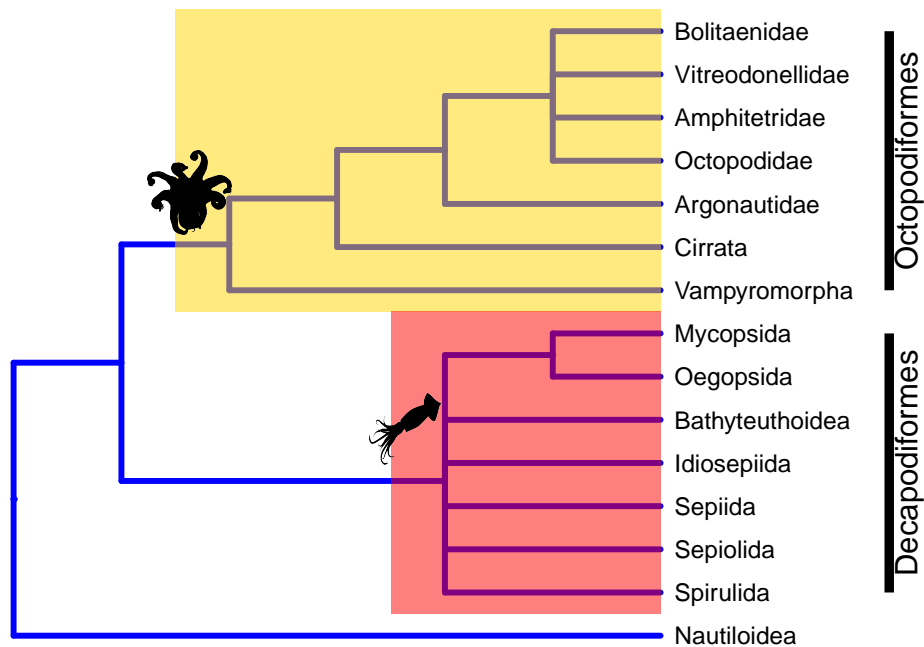


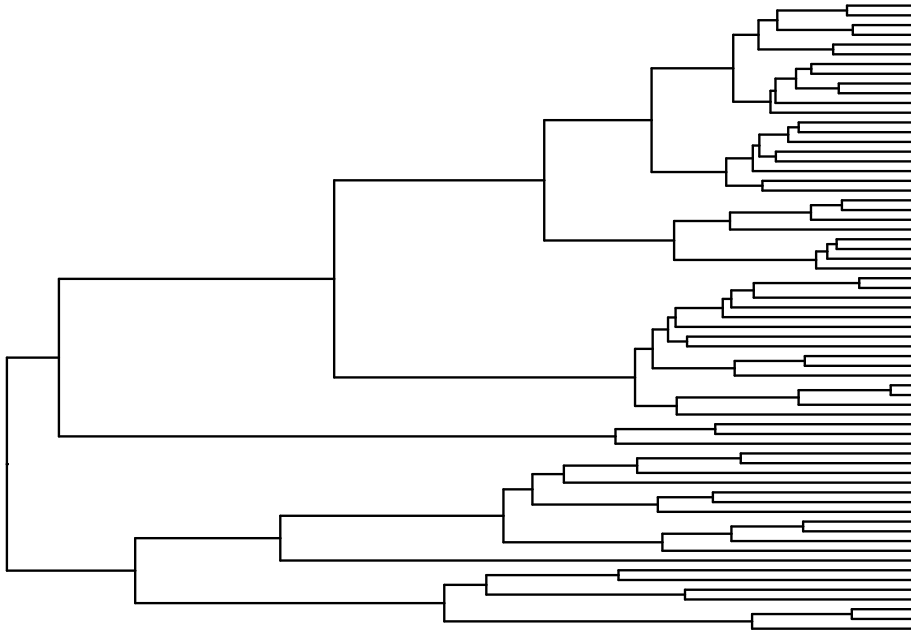
Figure 5.1: Plot of cephalopod families annotated using `ggtree` and `Phylopic`.

because I'm loading a file directly from GitHub, a repository for all sorts of code and the host of this site! Randi (and many other coders) make some of the things they produce freely available through GitHub. This can be data, files or code.

```
primates <- read.nexus("https://raw.githubusercontent.com/rgriff23/Dissertation/master/
```

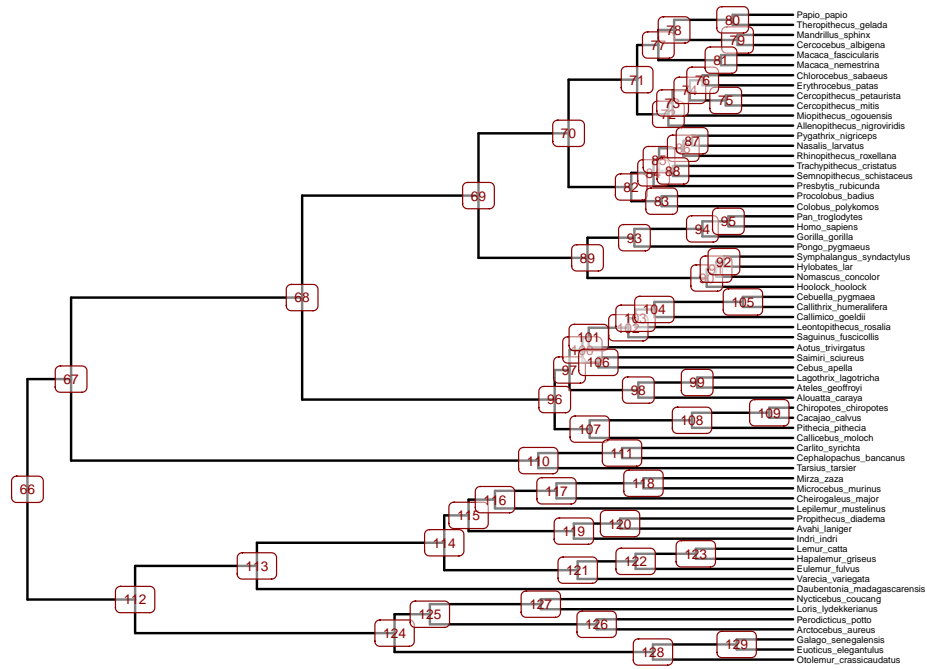
Let's plot the new tree first. Here I'm assigning the plot to a named object (`p1`) in R. This means that instead of immediately printing out the plot, R stores it in the working directory. The reason for doing this will become clear as we go on. It saves us typing out every line of code each time we want to add a new geom!

```
p1 <- ggtree(primates)
p1
```



Let's use what we already know about `ggtree` to customise this plot into something more useful. In particular, this plot is quite useful because it tells us the numbers of each node and we will need that later on.

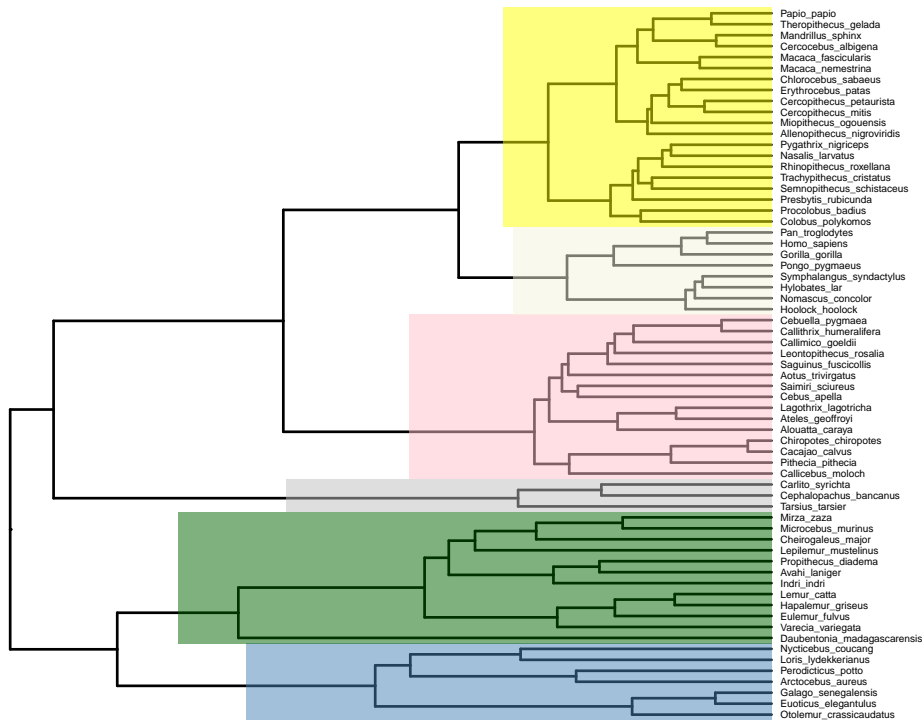
```
ggtree(primates) +  
  xlim(0,90) +  
  geom_tiplab(size=1.5) +  
  geom_label2(aes(subset=!isTip, label=node), size=2, color="darkred", alpha=0.5)
```



Let's label the 6 primate superfamilies using the node numbers I have extracted from the previous plot. You can choose whatever colours you prefer here. I've also added some useful features to this code. the use of `xlim()` can be very useful when plotting a tree with some extra space for more details. Here I've set the limits of the x dimension (the horizontal) to be between 0 and 100. This gives me space for later annotations.

```
p2 <- ggtree(primates) +
  xlim(0,100) +
  geom_tiplab(size=1.5, offset=0.5) +
  geom_highlight(node=124, fill="steelblue", alpha=0.5) +
  geom_highlight(node=113, fill="darkgreen", alpha=0.5) +
  geom_highlight(node=110, fill="gray", alpha=0.5) +
  geom_highlight(node=96, fill="pink", alpha=0.5) +
  geom_highlight(node=89, fill="beige", alpha=0.5) +
  geom_highlight(node=70, fill="yellow", alpha=0.5)
```

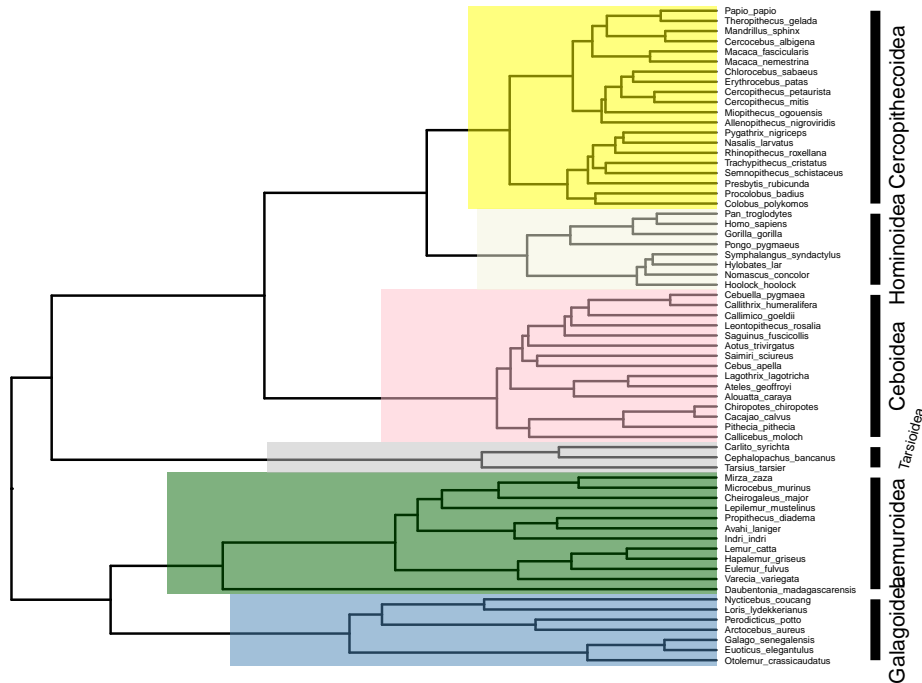
p2



So far so good. Let's add on bars to label the superfamilies like I did for the cephalopod version. This time, I'll add the new details to the object p3 to save retyping. Take note of the arguments in each label. You may need to twist these with some trial-and-error to make sure they suit your plot window.

```
p3 <- p2 +
  geom_cladelabel(124, "Galagoidea", offset=15, barsize=2, angle=90,
    offset.text=1.5, hjust=0.5, fontsize=3) +
  geom_cladelabel(113, "Lemuroidea", offset=15, barsize=2, angle=90,
    offset.text=1.5, hjust=0.5, fontsize=3) +
  geom_cladelabel(110, "Tarsiidae", offset=15, barsize=2, angle=75,
    offset.text=2.5, hjust=0.2, fontsize=2) +
  geom_cladelabel(96, "Ceboidea", offset=15, barsize=2, angle=90,
    offset.text=1.5, hjust=0.5, fontsize=3) +
  geom_cladelabel(89, "Hominoidea", offset=15, barsize=2, angle=90,
    offset.text=1.5, hjust=0.5, fontsize=3) +
  geom_cladelabel(70, "Cercopithecoidea", offset=15, barsize=2, angle=90,
    offset.text=1.5, hjust=0.5, fontsize=3)
```

p3



There are some helpful details here, such as the fact that the label for Tarsiioidea is off at an angle to avoid overlapping with other labels ($angle = 75$). The extra arguments in these options demonstrate how much control you can exercise over each geom.

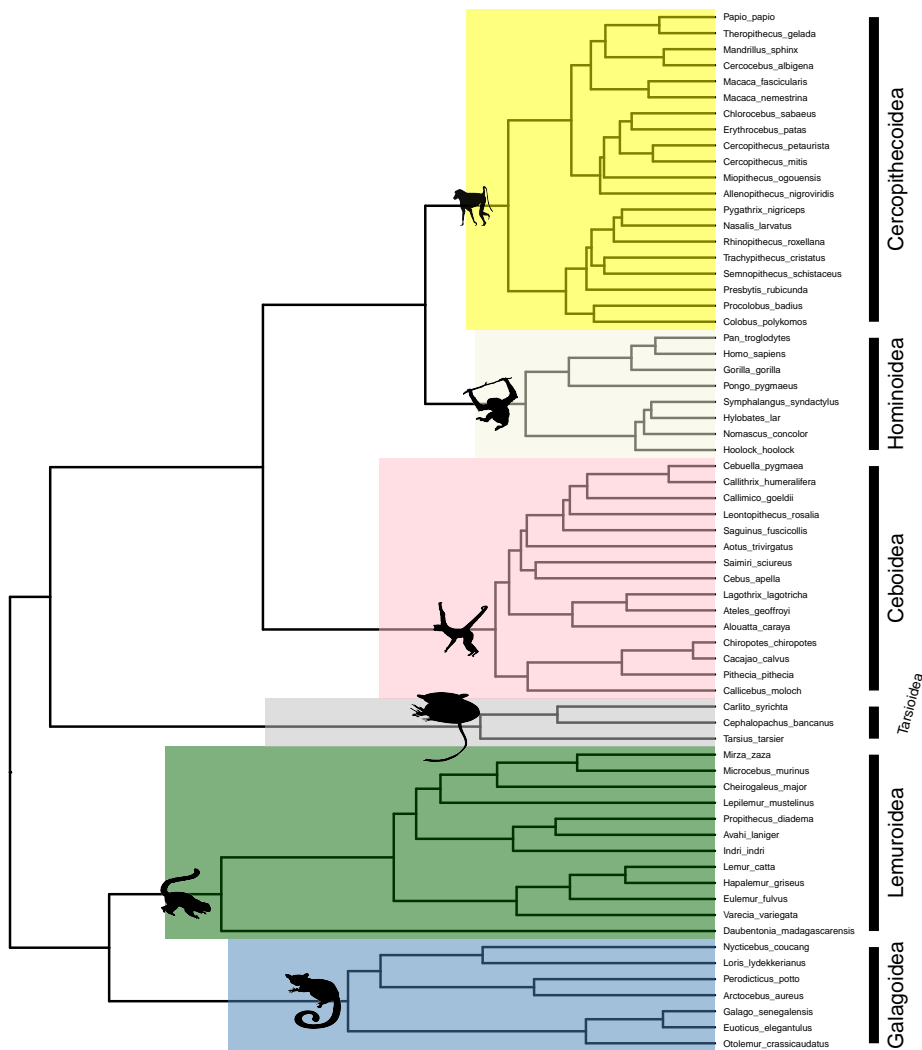
Now let's get to adding images. The way to do this is a little awkward but I think it's worth the hassle. The first thing we have to do is gather the links for each image we want to use. I've chosen to do this by building a small data frame containing the urls to the images on phylopic, the names of the super families I want to label and the nodes I want to plot the images on.

```
images <- data.frame(node = c(124,113,110,96,89,70),
  phylopic = c("http://phylopic.org/assets/images/submissions/
7fb9bea8-e758-4986-afb2-95a2c3bf983d.512.png",
"http://phylopic.org/assets/images/submissions/
bac25f49-97a4-4aec-beb6-f542158ebd23.512.png",
"http://phylopic.org/assets/images/submissions/
f598fb39-facf-43ea-a576-1861304b2fe4.512.png",
"http://phylopic.org/assets/images/submissions/
aceb287d-84cf-46f1-868c-4797c4ac54a8.512.png",
"http://phylopic.org/assets/images/submissions/
0174801d-15a6-4668-bfe0-4c421fbe51e8.512.png",
"http://phylopic.org/assets/images/submissions/
72f2f854-f3cd-4666-887c-35d5c256ab0f.512.png"),
```

```
species = c("Galagoidea", "Lemuroidea", "Tarsioida",
            "Ceboidea", "Hominoidea", "Cercopithecoidea"))
```

Once we have the urls we need in a nice dataframe, we can pipe them into the `geom_nodelab` geom and the end product should appear.

```
p3 %<+% images +
  geom_nodelab(aes(image = phylopic), geom = "image", size = .04, nudge_x = -4)
```



As you can probably tell, the images don't have to be from Phylopic. You can use any images you have the rights to in exactly the same way!

5.5 Further info

This chapter barely scratches the surface of what ggtree is capable of. For much more detail, have a look at Guangchuang Yu's very own Bookdown covering the topic. You can access the book by clicking [here](#) or by running the following code in R once you have ggtree installed.

```
vignette("ggtree", package = "ggtree")
```


Chapter 6

Bibliography

Bibliography

- D.A. Baum and S.D. Smith. *Tree Thinking: An Introduction to Phylogenetic Biology*. Macmillan Learning, 2012. ISBN 9781936221165. URL https://books.google.co.uk/books?id=zW_ApwAACAAJ.
- Nizar Ibrahim, Simone Maganuco, Cristiano Dal Sasso, Matteo Fabbri, Marco Auditore, Gabriele Bindellini, David M. Martill, Samir Zouhri, Diego A. Mattarelli, David M. Unwin, Jasmina Wiemann, Davide Bonadonna, Ayoub Amame, Juliana Jakubczak, Ulrich Joger, George V. Lauder, and Stephanie E. Pierce. Tail-propelled aquatic locomotion in a theropod dinosaur. *Nature*, 581(7806):67–70, 2020. doi: 10.1038/s41586-020-2190-3. URL <https://doi.org/10.1038/s41586-020-2190-3>.
- Michael R. McGowen, Stephen H. Montgomery, Clay Clark, and John Gatesy. Phylogeny and adaptive evolution of the brain-development gene microcephalin (mcp1) in cetaceans. *BMC Evolutionary Biology*, 11(1):98, 2011. doi: 10.1186/1471-2148-11-98. URL <https://doi.org/10.1186/1471-2148-11-98>.
- Martin Morgan. *BiocManager: Access the Bioconductor Project Package Repository*, 2019. URL <https://CRAN.R-project.org/package=BiocManager>. R package version 1.30.9.
- C.L. Nunn. *The Comparative Approach in Evolutionary Anthropology and Biology*. University of Chicago Press, 2011. ISBN 9780226608983. URL <https://books.google.co.uk/books?id=qj4cSzJGQJAC>.
- Heidi G. Parker, Dayna L. Dreger, Maud Rimbault, Brian W. Davis, Alexandra B. Mullen, Gretchen Carpintero-Ramirez, and Elaine A. Ostrander. Genomic analyses reveal the influence of geographic origin, migration, and hybridization on modern dog breed development. *Cell Reports*, 19(4):697 – 708, 2017. ISSN 2211-1247. doi: <https://doi.org/10.1016/j.celrep.2017.03.079>. URL <http://www.sciencedirect.com/science/article/pii/S2211124717304564>.
- Ralph S. Peters, Lars Krogmann, Christoph Mayer, Alexander Donath, Simon Gunkel, Karen Meusemann, Alexey Kozlov, Lars Podsiadlowski, Malte Petersen, Robert Lanfear, Patricia A. Diez, John Heraty, Karl M. Kjer, Seraina Klopstein, Rudolf Meier, Carlo Polidori, Thomas Schmitt, Shan-

- lin Liu, Xin Zhou, Torsten Wappler, Jes Rust, Bernhard Misof, and Oliver Niehuis. Evolutionary history of the hymenoptera. *Current Biology*, 27(7):1013–1018, 2020/06/27 2017. doi: 10.1016/j.cub.2017.01.027. URL <https://doi.org/10.1016/j.cub.2017.01.027>.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.
- R Hackathon et al. *phylobase: Base Package for Phylogenetic Structures and Comparative Data*, 2019. URL <https://CRAN.R-project.org/package=phylobase>. R package version 0.8.6.
- Simon M. Reader, Yfke Hager, and Kevin N. Laland. The evolution of primate general and cultural intelligence. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1017–1027, 2011. doi: 10.1098/rstb.2010.0342. URL <https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2010.0342>.
- Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. URL <https://CRAN.R-project.org/package=tidyverse>. R package version 1.2.1.
- Guangchuang Yu. *ggimage: Use Image in 'ggplot2'*, 2019. URL <https://CRAN.R-project.org/package=ggimage>. R package version 0.2.4.
- Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, and Tommy Tsan-Yuk Lam. ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution*, 8:28–36, 2017. doi: 10.1111/2041-210X.12628. URL <http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12628/abstract>.
- Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, and Yi Guan. Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35:3041–3043, 2018. doi: 10.1093/molbev/msy194. URL <https://doi.org/10.1093/molbev/msy194>.