Lab 6: Diversification Methods

Comparative Biology and Macroevolution

5/10/2019

In this lab, we will be exploring methods of diversification

1: The gamma statistic

2: The Magallon and Sanderson method

3: MEDUSA: modeling evolutionary diversification using stepwise AIC

```
library(ape)
library(phytools)
library(geiger)
library(apTreeshape)
```

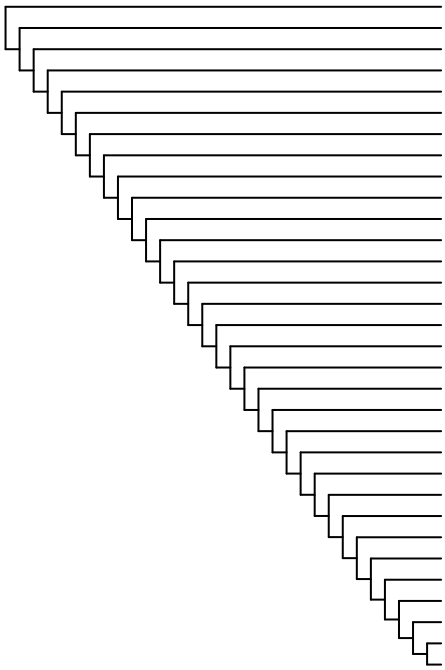
Diversification Methods

In order to understand different patterns of diversification, we will be using methods that look at the shape of a tree.

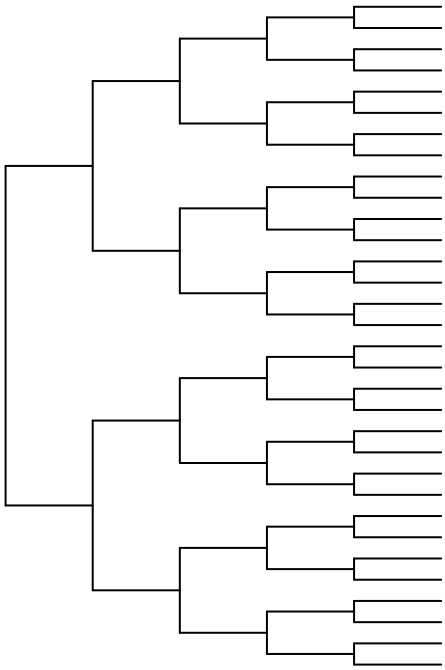
One aspect of tree shape is **Tree Balance**. Tree balance allows us to look at the extent to which nodes define subgroups of equal size. Balanced trees are also called **symmetrical**, while unbalanced trees are also called **pectinate**.

The most widely used statistic for tree balance is the Colless' index of imbalance I_c (Colless 1982). I_c sums the difference in the number of tips in the right-hand and left-hand branches at each node over all $(n-1)$ nodes in a tree with n tips. It then normalizes the imbalance score by dividing by the largest possible score.

$$I_C = \frac{\sum_{allnodes} (N_L - N_R)}{\frac{(N-1)(N-2)}{2}}$$



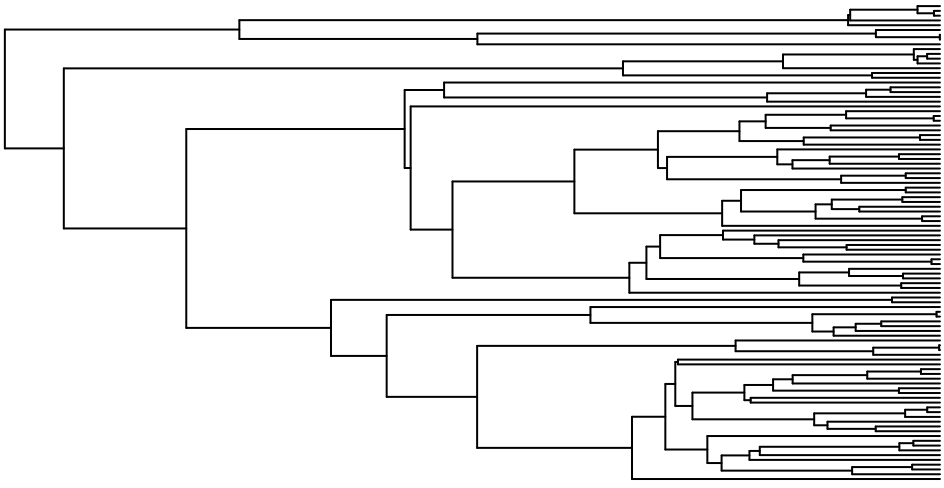
Colless' index = 1



Colless' index = 0

We can calculate the Colless' index of a tree using the package *apTreeshape*. One thing to note, the *apTreeshape* package uses `treeshape` objects instead of `phylo` objects. Therefore, we will need to change convert them using `as.treeshape()`.

```
my_tree <- sim.bdtree(b = 2, d = 0.5, stop = "taxa", n = 100, extinct = FALSE)
my_tree <- drop.extinct(my_tree)

plot(my_tree, show.tip.label = F)
```



```
# Turn phylo into treeshape
my_treeshape <- as.treeshape(my_tree)

# This calculates the sum of the number of left - right
# (the top part of the equation)
my_colless <- colless(my_treeshape, norm = NULL)
my_colless
```

```
## [1] 499
```

```
# this calculates the total score (bottom part of eqn)
ntips <- length(my_tree$tip.label)
total_score <- ((ntips-1)*(ntips-2))/2
total_score
```

```
## [1] 4851
```

```
my_colless/total_score
```

```
## [1] 0.1028654
```

We can get a Colless' index for our tree. However, how can we use interprete this number?

apTreeshape can do an analysis to assess whether the tree has the amount of imbalance one would expect from a pure birth (Yule) tree. We can therefore simulate replicate Yule trees, and estimate a p-value on this Monte Carlo method using the `colless.test` function. In the function we need to specify the number of random trees to be generated `n.mc`.

The alternative "less" should be used to test whether the tree is more balanced (less umbalanced) than predicted by the null model. This gives you the p-value, or the probability of the tree being more balanced given the null model predictions.

If the p-value was 0.5, for example, then our tree would be right in the middle of the distribution expected for Yule trees. If the p-value was 0.01, however, it would mean that very few Yule trees are as balanced as our tree, which would make it hard to believe that our tree is a Yule tree.

```
colless.test(tree = my_treeshape, model = "yule", alternative ="less", n.mc = 1000)
```

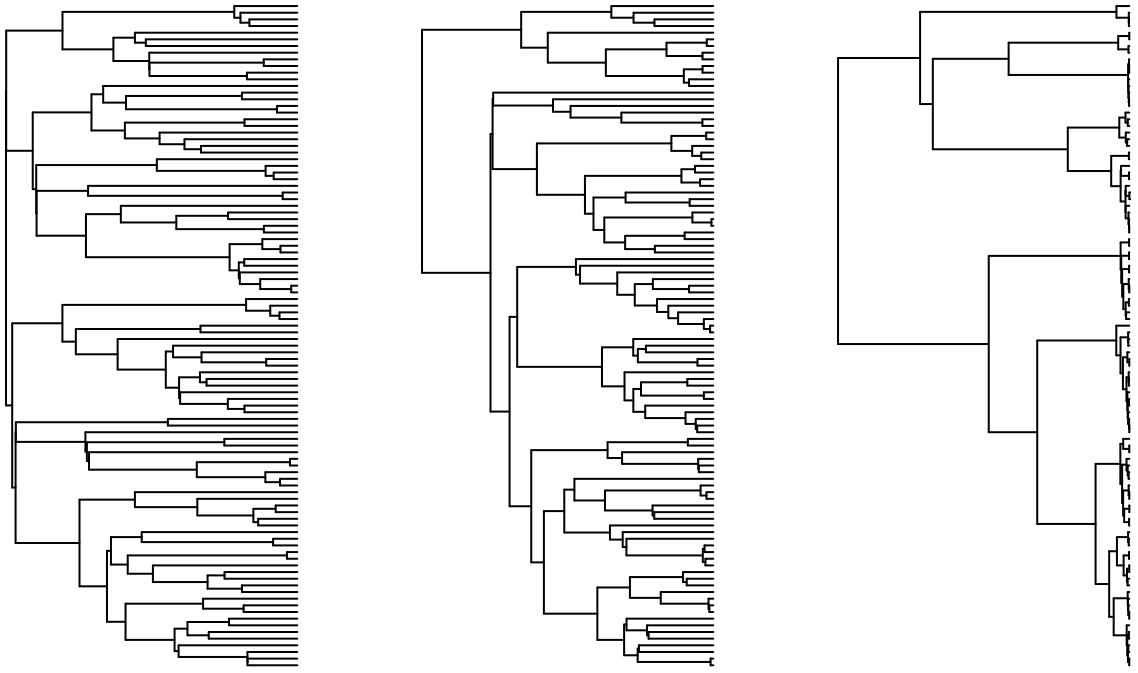
```
##
##
##  Test of the yule hypothesis using the Colless index
## Statistic = 499
## Standardized Statistic = 1.500761
## p-value = 0.953
## alternative hypothesis: the tree is more balanced than predicted by the yule model
##
## Note : the p-value was computed using a Monte-Carlo method
```

We can also check to see if our tree is more unbalanced than predicted using `alternative = "greater"`.

Another way to understand tree shape is by looking at the distribution of waiting times (i.e. the nodes). One common statistic to measure the relative position of internal nodes is the γ**-statistic**. The gamma statistic (γ) was developed by Pybus and Harvey (2000).

$$\gamma = \frac{(\frac{1}{n-2}\sum_{i=2}^{n-1}(\sum_{j=2}^{i} kg_k)) - \frac{T}{2}}{T\sqrt{\frac{1}{12(n-2)}}}, T\sum_{j=2}^{n} jg_j$$

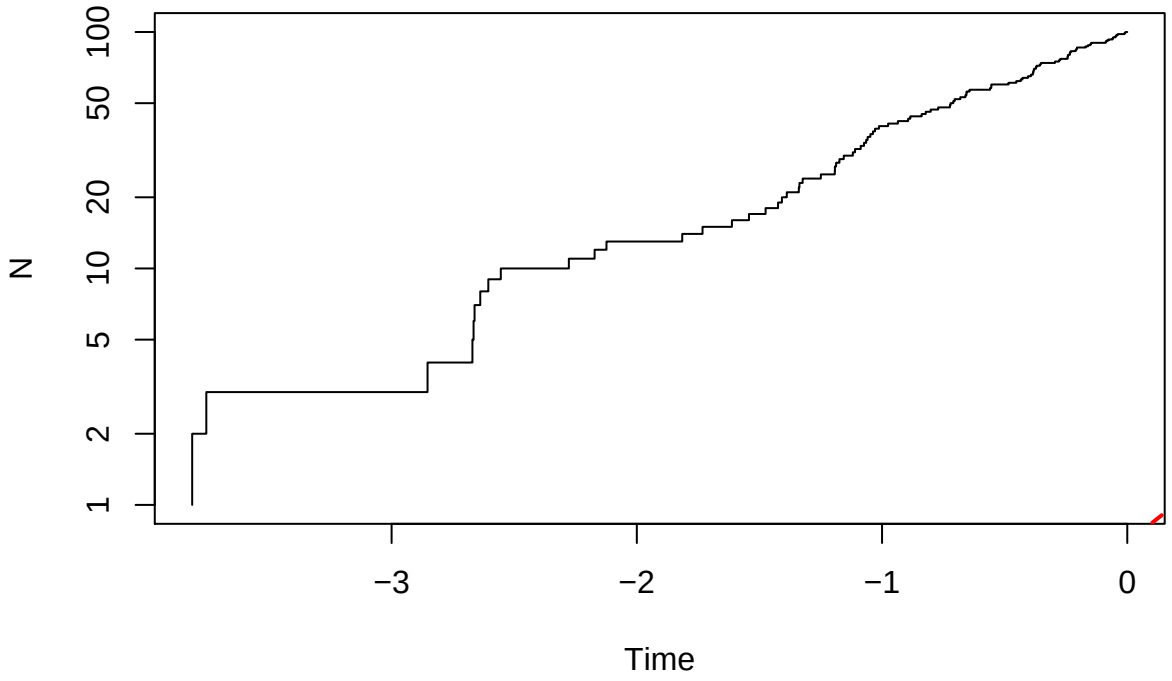
γ has useful properties if we are thinking about diversification under a birth death process. Under a pure birth process, γ-values of completely reconstructed phylogeneis follow a standard normal distribution centered around zero. If $\gamma > 0$, then the internal nodes are closer to the tips, while if $\gamma < 0$, then the internal nodes are closer to the root. Therefore, the null hypothesis of constant b and d is rejected at the 5% level if $\gamma < -1.645$ (one-tailed test). We call this the constant-rates (CR) test.



Exploring using the gamma statistic

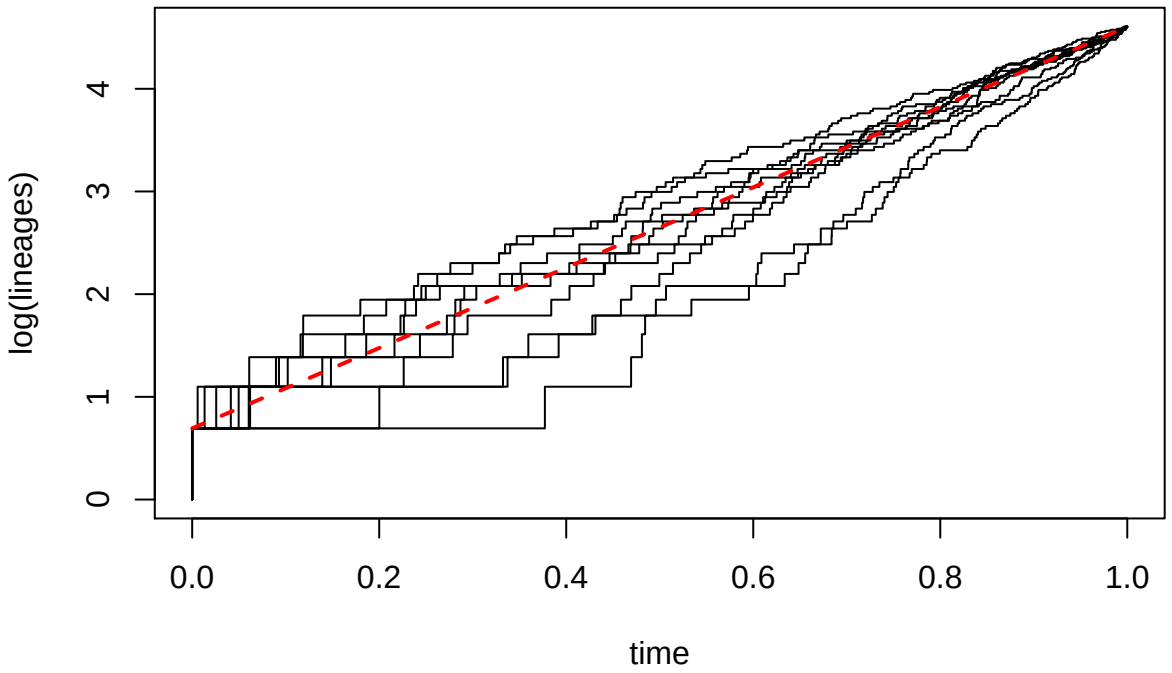
One way to visualize the distribution of waiting times on a tree is to use a lineage though time (LTT) plot. There are multiple functions to generate a LTT plot. In this lab, we will introduce `ltt.plot()` and `ltt()` Lets look at a LTT plot for a single pure birth tree.

```
tree <- pbtree(n = 100)
ltt.plot(tree, log = "y") # log transform the y axis
lines(c(0, 1), c(log(2), log(100)), lty = "dashed", lwd = 2, col = "red")
```



Cool, now lets try with multiple realizations

```
trees <- pbtree(n = 100, nsim = 10, scale = 1)
ltt_plots <- ltt(trees)
lines(c(0, 1), c(log(2), log(100)),
      lty = "dashed", lwd = 2, col = "red")
```




```
# If we give ltt multiple trees
# Then apples the ltt function to all of the trees
# It then returns a list of "ltt" objects
```

Not only does the function `ltt()` generates a LTT plot, but it also calculates the γ statistic for our tree.

Remember to look inside of a list, we use `[[i]]` to get the *ith* element in the list.

```
ltt_plots[[1]]
```

```
## Object of class "ltt" containing:
##
## (1) A phylogenetic tree with 100 tips and 99 internal nodes.
##
## (2) Vectors containing the number of lineages (ltt) and branching times (times) on the tree.
##
## (3) A value for Pybus & Harvey's "gamma" statistic of -0.9996, p-value = 0.3175.
```

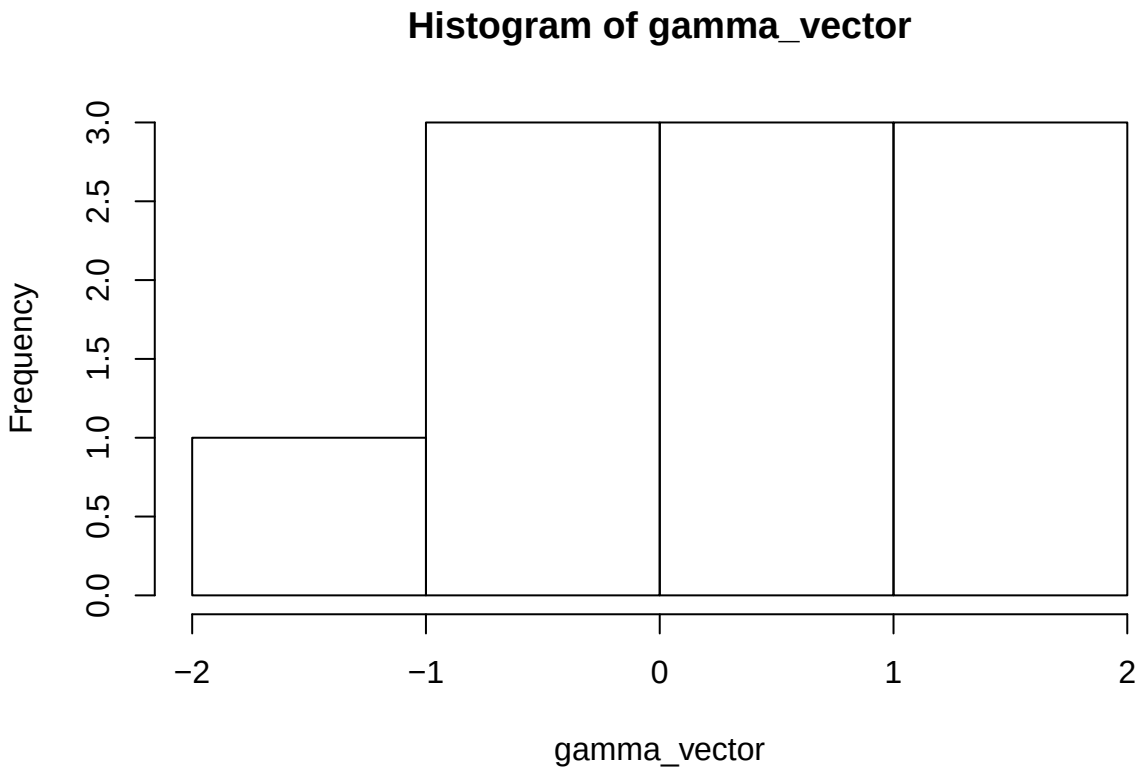
Now lets pull out the γ statistic for `ltt_plots[[1]]`

```
ltt_plots[[1]]$gamma
```

```
## [1] -0.9996246
```

Lets pull out all the γ statistic values and put them in a new vector.

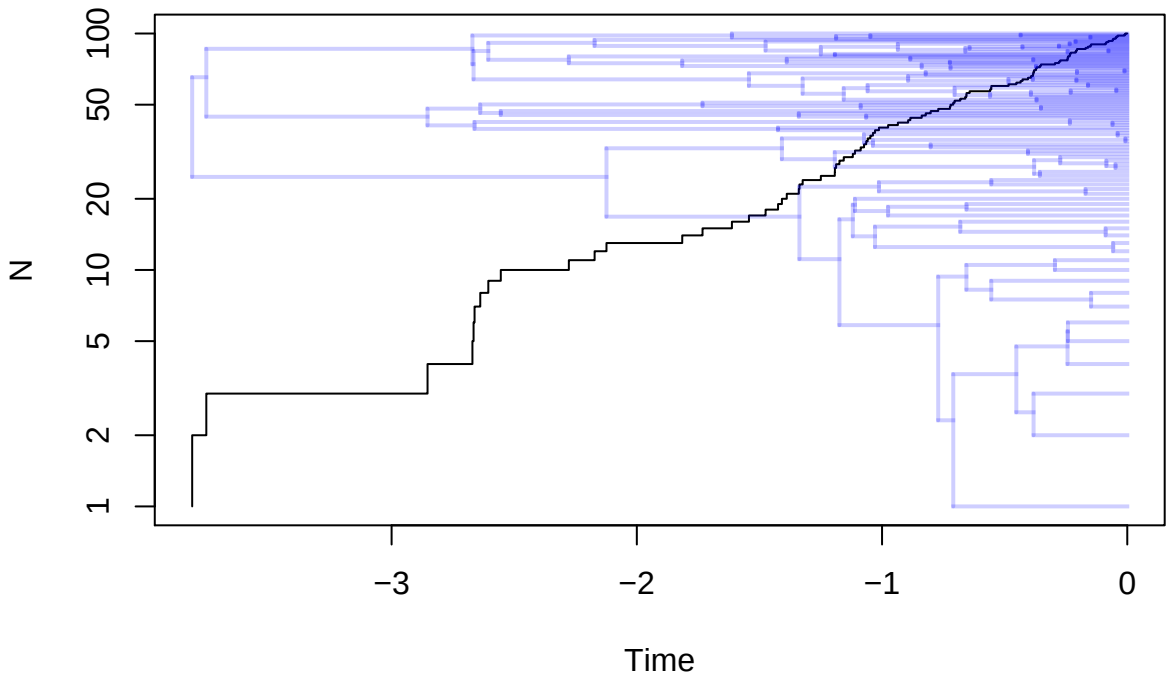
```
n <- length(ltt_plots)

# Create an empty vector
gamma_vector <- numeric(n)

# fill the vector with a for loop
for (i in 1:n){
  gamma_vector[i] <- ltt_plots[[i]]$gamma
}

# Histogram of all the gamma values
hist(gamma_vector)
```



** Question: What value is the histogram centered around?**

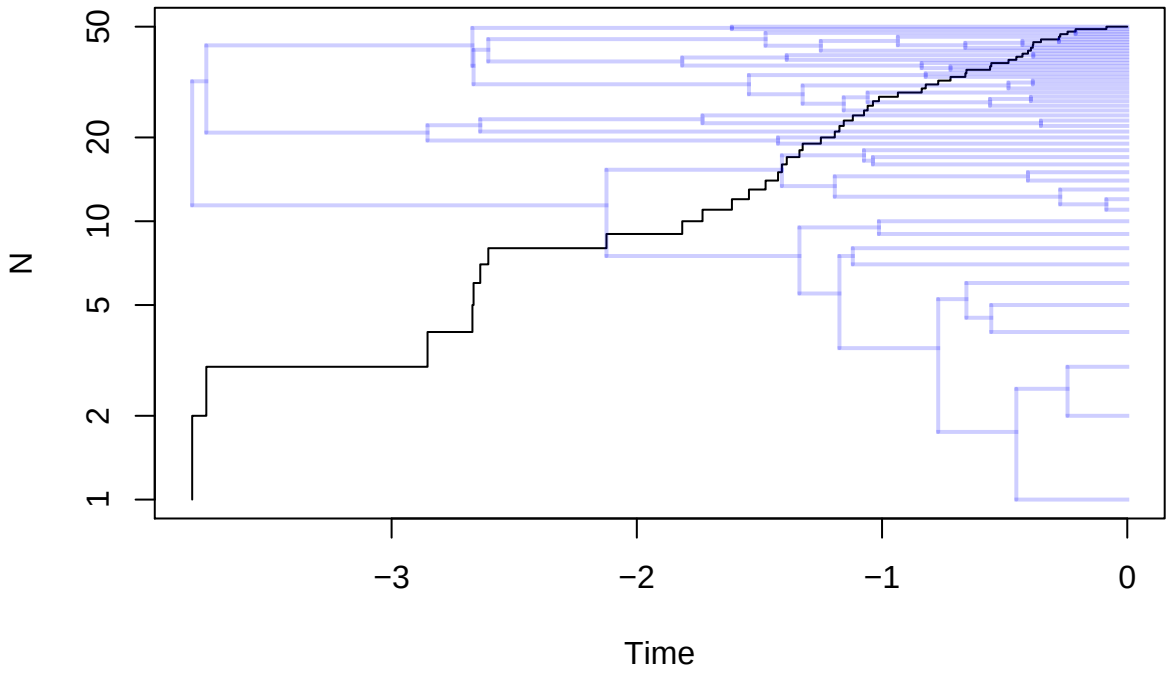
Plotting tree and LTT on the same plot

```
# Our make transparent function
makeTransparent<-function(someColor,alpha=10){
    newColor<-col2rgb(someColor)
    apply(newColor,2,function(curcoldata){
    rgb(red=curcoldata[1],
        green=curcoldata[2],
        blue=curcoldata[3],
        alpha=alpha,
        maxColorValue=255)
    })
}

ltt.plot(tree, log = "y")
plotTree(tree,
         color=makeTransparent("blue",alpha=50),
         ftype="off",
         add=TRUE,
         mar=par()$mar)
```



Lets look at what happens when you incompletely sample a tree

```
# sample 50% of the tree
incomplete_tree <- drop.random(phy = tree, n = 50)

ltt.plot(incomplete_tree, log = "y")
plotTree(incomplete_tree,
         color=makeTransparent("blue",alpha=50),
         ftype="off",
         add=TRUE,
         mar=par()$mar)
```



What would happen to the γ statistic if we incompletely sampled a tree?

```
# Another function to calculate the gamma statistic
gammaStat(tree)
```

```
## [1] -0.5743943
```

```
gammaStat(incomplete_tree)
```

```
## [1] -1.995553
```

Diversification in Homalopsid snakes

Question: Has the family Homalopsidae evolved under a constant rate through time?

```
snake_tree <- read.tree("homalops.tre")
snake_tree
```

```
##
## Phylogenetic tree with 21 tips and 20 internal nodes.
##
## Tip labels:
##  Enhydris_plumbea_A, Enhydris_matannensis, Lake_Towuti, Enhydris_chinensis, Enhydris_subtaeniata, Enh
##
## Rooted; includes branch lengths.
```

```
snake_gamma <- gammaStat(snake_tree)
snake_gamma
```

```
## [1] -3.241085
```

The negative value for the snake tree suggests that cladogenetic events are disproportionately distributed towards the root of the tree. This is possibly evidence for rapid initial diversification in the clade. However, incomplete taxonomic sampling has been shown to bias the gamma statistic towards negative values (see Pybus and Harvey 2000 for a discussion). One solution is to create a null distribution for the gamma statistic

on incompletely sampled trees using simulation. This is called the Monte Carlo constant rates (MCCR) test. To accomplish this we will use functions in *geiger* and *ape* to

- Using a birth-death model, simulate trees of size equal to the known (or estimated) species richness of the clade.
- Prune the simulated trees down to the size of the original by randomly deleting taxa.
- Recalculate the gamma statistic.

If the observed gamma statistic falls into the tail of this null distribution we can conclude that even with incomplete taxonomic sampling, there are more cladogenetic events early in the history of our tree than expected under a constant rates model. To perform these simulations, we need to know or estimate the following parameters for the clade:

- Birth rate
- Total species richness
- Sampled species richness

There are approximately 34 species of homalopsid snakes but we have sampled only 21 of them in our tree. We can estimate the speciation rate as

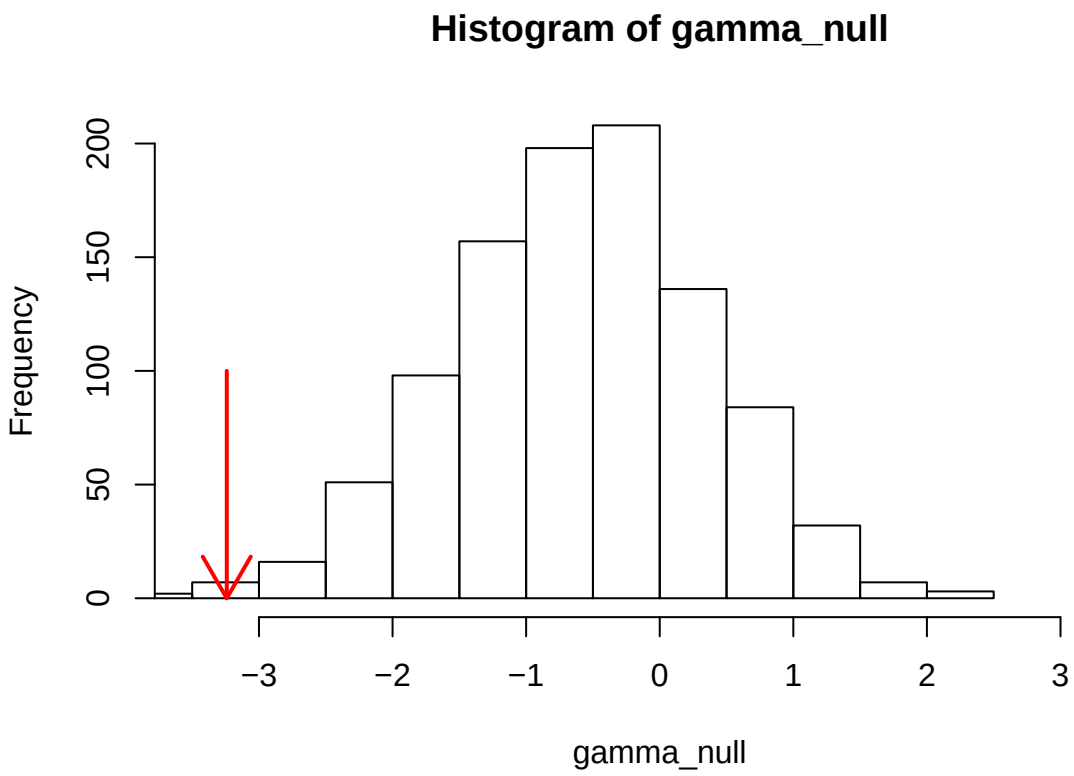
$$\frac{ln(richness) - ln(2)}{age}$$

```
# Get the age of the clade by finding the branching time at the root.
age <- branching.times(snake_tree)[1]
# Manually specify total richness
richness <- 34
snakebirth =  (log(richness) - log(2))/age
snakebirth # speciation rate estimate for all species in the tree
```

```
##        22
## 0.1300637
```

```
num_simulations <- 1000 #number of simulations
gamma_null <- numeric(num_simulations)
# gamma_null will hold the simulated gamma values
# for the trees that have been pruned down
for (i in 1:num_simulations) {
   sim_tree <- sim.bdtree(b = snakebirth, d=0, stop = "taxa", n=34)
  prune <- drop.random(sim_tree, 13)
  # Here we drop 13 species randomly from the tree
  gamma_null[i] <- gammaStat(prune)
  # this stores the gamma values from the pruned trees
}
```

```
hist(gamma_null, xlim = c(-3.5, 3.5))
arrows(x0 = snake_gamma, y0 = 100,
       x1 = snake_gamma, y1 = 0,
       col="red", lwd=2, xlab = "null gammas",
       main = "Incomplete Sampling")
```




```
mean(gamma_null) # Find the mean gamma of nulls
```

```
## [1] -0.6108845
```

Which of the null values are smaller (more negative) than the data?

```
smallerNull<- gamma_null <= snake_gamma
smallerNull
```

```
##    [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [34] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##   [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [122] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [144] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [155] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [166] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [177] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [188] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [199] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [210] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [221] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [232] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
##  [243] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [254] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [276] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [287] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [298] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [309] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [320] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [331] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [342] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [353] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [364] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [375] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [386] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [408] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [419] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [430] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [441] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [452] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [463] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [474] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [485] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [496] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [507] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [518] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [540] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [551] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [562] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [573] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [584] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [595] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [606] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [617] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [628] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [639] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [650] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [672] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [683] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [694] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [705] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [716] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [727] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [738] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [749] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [760] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [771] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [782] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [793] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [804] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [815] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [826] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
##  [837] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [848] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [859] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [870] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [881] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [892] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [903] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [914] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [925] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [936] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [947] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [958] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [969] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [980] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [991] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# How many TRUEs are there?
count<-sum(smallerNull)
count
```

```
## [1] 5
```

We can now calculate a p-value for our emperical γ statistic using the simulations.

$$\frac{nsmaller + 1}{nsims + 1}$$

```
mccr_pval <- (count+1)/(num_simulations+1)
mccr_pval
```

```
## [1] 0.005994006
```

If the null hypothesis is true that clade evolved at a constant rate, the probability of us observing a γ statistic of -3.2410845 is 0.005994.

Can we reject the hypothesis of a constant rate?

Magallon and Sanderson method

We can also use the Magallon and Sanderson (2000) method to calculate net diversification rate for a clade given extant diversity and age. The Magallon and Sanderson method allows us to ask whether a clade is exceptionally diverse or depauperate.

Lineage diversification is characterized by two parameters that are transformations of speciation (λ) and extinction (μ). Diversification rate is defined as $r = \lambda - \mu$, and relative extinction rate is defined as $\epsilon = \frac{\mu}{\lambda}$.

What are the bounds of r and ϵ?

The maximum-likelihood estimate of diversification rate if extinction is negligible is

$$\hat{r} = \hat{\lambda} = \frac{ln(n)}{t}$$

For a stem group age and

$$\hat{r} = \hat{\lambda} = \frac{ln(n) - ln(2)}{t}$$

Are hyenas especially species poor?

From our observation, we can see that there are only 4 species of hyenas. However, is the family Hyenidae especially depapurate compared to other families in the order Carnivora? We will use the Nyakatura and Bininda-Emonds (2012) Carnivora tree, which include all 286 species of Carnivora.

```
carnivora <- read.tree("full_carnivoran_tree.tre")
carnivora
```

```
##
## Phylogenetic tree with 286 tips and 264 internal nodes.
##
## Tip labels:
##  Acinonyx_jubatus, Puma_concolor, Puma_yagouaroundi, Lynx_canadensis, Lynx_lynx, Lynx_pardinus, ...
##
## Rooted; includes branch lengths.
```

```
n <- length(carnivora$tip.label)
t <- max(nodeHeights(carnivora)) # Total age of tree

r_stem_age <- log(n)/t
r_crown_age <- (log(n)-log(2))/t
```

Note: There is a function in *geiger* that will calculate the net diversification rate for the stem and crown age for you

```
r_stem_age
```

```
## [1] 0.08714933
```

```
bd.ms(time = t, n = n, crown = FALSE, epsilon = 0)
```

```
## [1] 0.08714933
```

```
r_crown_age
```

```
## [1] 0.0764691
```

```
bd.ms(time = t, n = n, crown = TRUE, epsilon = 0)
```

```
## [1] 0.0764691
```

Here we can see that the above equations give us the same result when extinction $= 0$, which would also force $\epsilon = 0$. For our net diversfication rate for the rest of our analysis, we will use `r_crown_age` since we are dealing with the crown carnivorans. However, since high extinction can impact our rate estimation, we need to also estimate the net diversification rate under high extinction $\epsilon = 0.9$.

```
net_r_e0 <- r_crown_age
net_r_e9 <- bd.ms(time = t, n = n, crown = TRUE, epsilon = 0.9)
```

Now we need to find the stem and crown age for the Hyenas

```
crown_node <- getMRCA(phy = carnivora, tip = c("Proteles_cristata", "Crocuta_crocuta"))

stem_node <- getMRCA(phy = carnivora, tip = c("Proteles_cristata", "Salanoia_concolor"))

# nodeheight computes the height above the root for a node
# Therefore, to get the age, you need to subtract it from the total height.

crown_node_age <- t - nodeheight(tree = carnivora, node = crown_node)
```

```
stem_node_age <- t - nodeheight(tree = carnivora, node = stem_node)
```

Functions `crown.p` and `stem.p` calculate the probability of obtaining a clade with at least `n` species given a net diversification rate (`r`), extinction fraction (`epsilon`), and time interval. Therefore, to check if a clade is exceptionally depauperate, look at `1 - crown.p`

```
# Prob of getting a clade as big as n
crown.p(time = crown_node_age, n = 4, r = net_r_e0, epsilon = 0)
```

```
## [1] 0.2591003
```

```
# Prob of getting a clade as small as n
1 - crown.p(time = crown_node_age, n = 4, r = net_r_e0, epsilon = 0)
```

```
## [1] 0.7408997
```

```
# Prob of getting a clade as big as n
stem.p(time = stem_node_age, n = 4, r = net_r_e9, epsilon = 0.9)
```

```
## [1] 0.9063514
```

```
# Prob of getting a clade as small as n
1 - stem.p(time = stem_node_age, n = 4, r = net_r_e9, epsilon = 0.9)
```

```
## [1] 0.09364863
```

Are hyenas especially species poor?

We can try visualizing the Magallon and Sanderson method

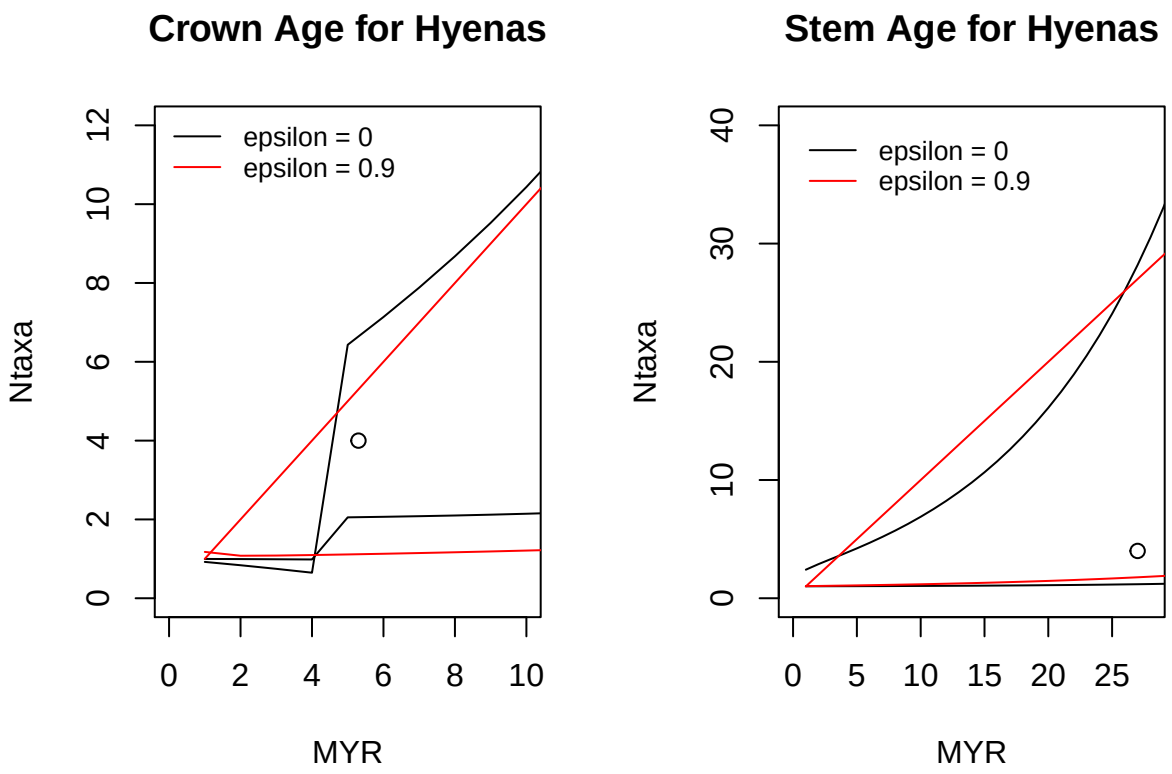
```
crown_bounds <- function(max_age = 100, r = 0.1, epsilon = 0, CI = 0.95){
  times <- 1:max_age
  lower_bound <- numeric(max_age)
  upper_bound <- numeric(max_age)
  for (i in 1:max_age){
    lower_bound[i] <- crown.limits(time = times[i],
                                   r = r,
                                   epsilon = epsilon,
                                   CI = CI)[1]
    upper_bound[i] <- crown.limits(time = times[i],
                                   r = r,
                                   epsilon = epsilon,
                                   CI = CI)[2]
  }
  crown_bounds <- data.frame(lower_bound, upper_bound)
  return(crown_bounds)
}

stem_bounds <- function(max_age = 100, r = 0.1, epsilon = 0, CI = 0.95){
  times <- 1:max_age
  lower_bound <- numeric(max_age)
  upper_bound <- numeric(max_age)
  for (i in 1:max_age){
    lower_bound[i] <- stem.limits(time = times[i],
                                  r = r,
                                  epsilon = epsilon,
                                  CI = CI)[1]
    upper_bound[i] <- stem.limits(time = times[i],
```

```
                                    r = r,
                                    epsilon = epsilon,
                                    CI = CI)[2]
  }
  stem_bounds <- data.frame(lower_bound, upper_bound)
  return(stem_bounds)
}
```

The `crown_bounds` and `stem_bounds` function return a dataframe. To access the columns in the dataframe, use `$` followed by the name of the column (here being `lower_bound` and `upper_bound`)

```
crown_no_extinction <- crown_bounds(max_age = t, r = net_r_e0, epsilon = 0)
crown_high_extinction <- crown_bounds(max_age = t, r = net_r_e9, epsilon = 0.9)

stem_no_extinction <- stem_bounds(max_age = t, r = net_r_e0, epsilon = 0)
stem_high_extinction <- stem_bounds(max_age = t, r = net_r_e9, epsilon = 0.9)
```

```
# one row, two columns
par(mfrow = c(1,2))

plot(x = crown_node_age, y = 4,
     xlim = c(0, 10), ylim = c(0, 12),
     main = "Crown Age for Hyenas",
     xlab = "MYR", ylab = "Ntaxa")
lines(1:t, crown_no_extinction$lower_bound, col = "black")
lines(1:t, crown_no_extinction$upper_bound, col = "black")
lines(1:t, crown_high_extinction$lower_bound, col = "red")
lines(1:t, crown_high_extinction$upper_bound_bound, col = "red")
legend(x = -0.5, y = 12.5, legend=c("epsilon = 0", "epsilon = 0.9"),
       col=c("black", "red"), lty=1, cex=0.8, bty = "n")

plot(x = stem_node_age, y = 4,
     xlim = c(0, 28), ylim = c(0, 40),
     main = "Stem Age for Hyenas",
     xlab = "MYR", ylab = "Ntaxa")
lines(1:t, stem_no_extinction$lower_bound, col = "black")
lines(1:t, stem_no_extinction$upper_bound, col = "black")
lines(1:t, stem_high_extinction$lower_bound, col = "red")
lines(1:t, stem_high_extinction$upper_bound_bound, col = "red")
legend(x = -0.5, y = 40.5, legend=c("epsilon = 0", "epsilon = 0.9"),
       col=c("black", "red"), lty=1, cex=0.8, bty = "n")
```



Where does the hyena fall in relation to the 95% confidence level?

MEDUSA: Modeling evolutionary diversification using stepwise AIC

A final method that you will be briefly introduced to is MEDUSA (Alfaro et al. 2009). Briefly, this method finds the likelihood of obtaining the particular combination of phylogenetic relationships (the tree with branch lengths) and taxonomic data (ages and species richnesses of extant groups) given particular values of b and d.

Next, the method fits a series of alternative models of increasing complexity, stopping when the improvement in AIC score is >4 (as the stopping criteria). However, the AICc-threshold can vary based on the number of tips in the tree.

In other words, the method fits piecewise birth-death models to ultrametric phylogenetic tree(s) according to phylogenetic (*edge-length*) and taxonomic (*richness*) likelihoods. Optimal model size is determined via a stepwise AIC approach.

Therefore, using this stepwise-AIC approach, we can see if there has been a rate shift (or multiple rate shifts) for in our tree.

Have different carnivoran clades evolved at different rates?

What is nice about MEDUSA is that it only needs a phylogeny (and an optional taxonomic richness matrix). Since we are using the Nyakatura and Bininda-Emonds (2012) species level Carnivoran tree, we will not use the optional taxonomic richness matrix argument.

```
run1 <- medusa(phy =  carnivora)
```

```
## Appropriate  aicc-threshold for a tree of 286 tips is: 7.123497.
##
## Step 1: lnLik=-843.5725; aicc=1691.167; model=bd
## Step 2: lnLik=-835.7751; aicc=1677.594; shift at node 466; model=yule; cut=node; # shifts=1
## Step 3: lnLik=-829.502; aicc=1669.114; shift at node 121; model=yule; cut=stem; # shifts=2
```

```
##
## No significant increase in aicc score. Disregarding subsequent piecewise models.
##
##   Model.ID Shift.Node Cut.At Model Ln.Lik.part         r epsilon     r.low
## 1        1        287   node  yule   -806.4043 0.1193610      NA 0.1053969
## 2        2        466   node  yule   -23.09765 0.0267955      NA 0.0000000
## 3        3        121   stem  yule           0 0.0000000      NA 0.0000000
##      r.high
## 1 0.1344988
## 2 0.0576023
## 3 0.0360902
```

Note: Make sure to note the AICc criteria given for your tree

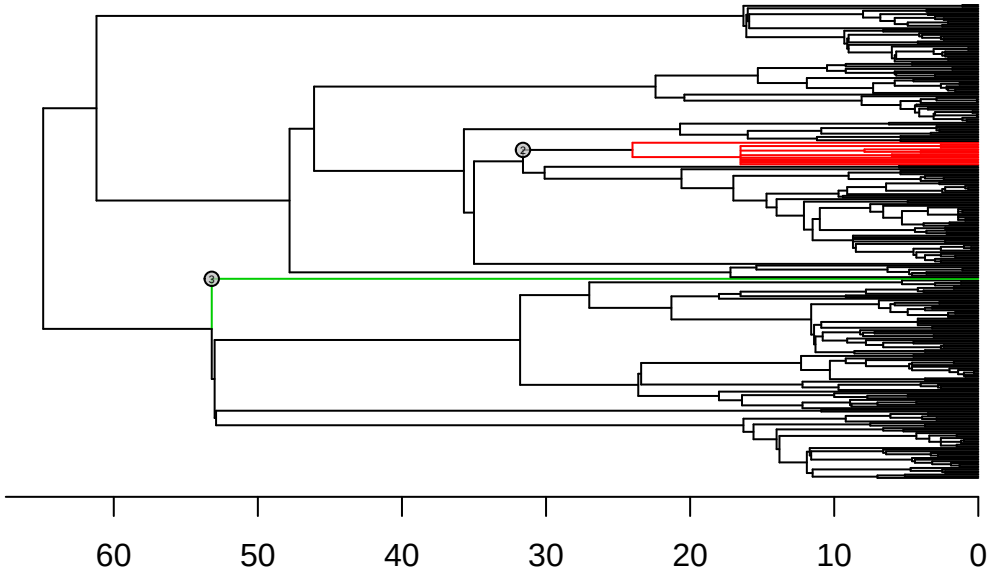
```
run1
```

```
##
## Optimal MEDUSA model for tree with 286 taxa.
##
##   Model.ID Shift.Node Cut.At Model Ln.Lik.part         r epsilon     r.low
## 1        1        287   node  yule   -806.4043 0.1193610      NA 0.1053969
## 2        2        466   node  yule   -23.09765 0.0267955      NA 0.0000000
## 3        3        121   stem  yule           0 0.0000000      NA 0.0000000
##      r.high
## 1 0.1344988
## 2 0.0576023
## 3 0.0360902
##
## 95% confidence intervals on parameter values calculated from profile likelihoods
```

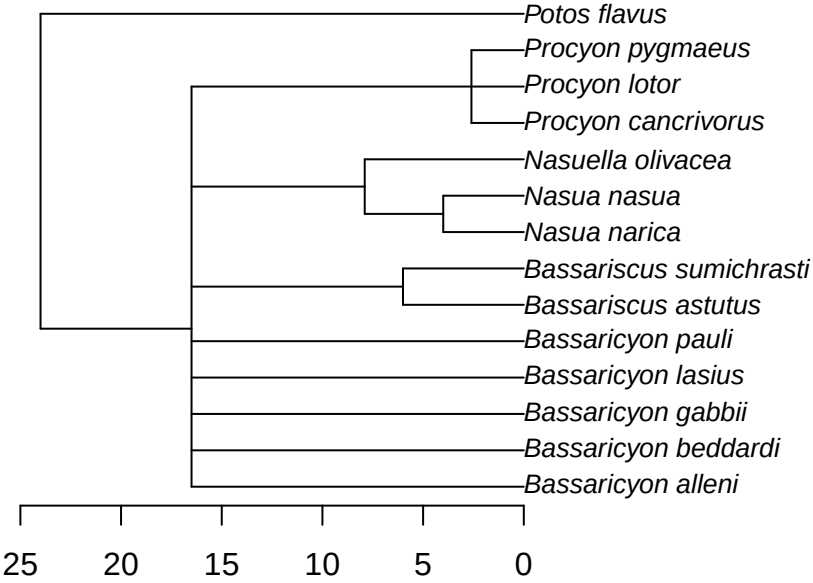
```
# r is the per-lineage net diversification rate
# epsilon is the relative extinction rates
shift_nodes <- run1$model$split.at # get nodes with rate shifts
shift_nodes
```

```
## [1] 287 466 121
```

```
# plot medusa rate shifts
plot(run1, show.tip.label = F)
```




```
low_div_tree <- extract.clade(phy = carnivora, node = shift_nodes[2])
plot(low_div_tree, cex = 0.8)
axisPhylo()
```

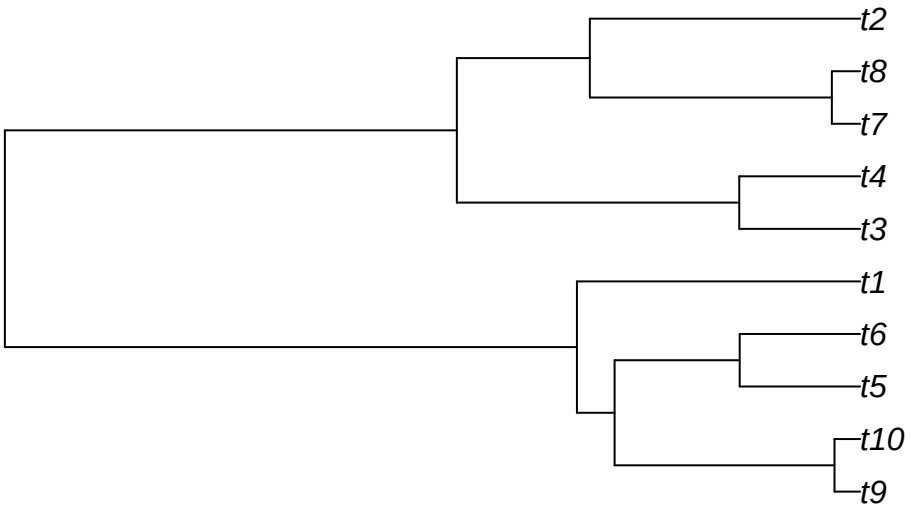


There appears to be a rate shift in the family Procyonidae.

Using richness data for incomplete trees

The `medusa` function can also take a matrix of taxonomic richness as an optional argument. The code below will demonstrate how this works.

```
# Simulate a 10 taxon tree
test_tree <- pbtree(b = 1, d = 0, nsim = 1, n = 10)
plot(test_tree)
```



Lets say t1 represents a clade with 50 species. Here, every tip will represent 1 taxa, except t1, which will represent 50 species.

Where would we expect a rate shift?

```
sort(test_tree$tip.label)
```

```
##  [1] "t1"  "t10" "t2"  "t3"  "t4"  "t5"  "t6"  "t7"  "t8"  "t9"
```

```
taxon_richness = data.frame(taxon = sort(test_tree$tip.label),
                            n.taxa = c(50, rep(1, 9)))
taxon_richness
```

```
##    taxon n.taxa
## 1     t1     50
## 2    t10      1
## 3     t2      1
## 4     t3      1
## 5     t4      1
## 6     t5      1
## 7     t6      1
## 8     t7      1
## 9     t8      1
## 10    t9      1
```

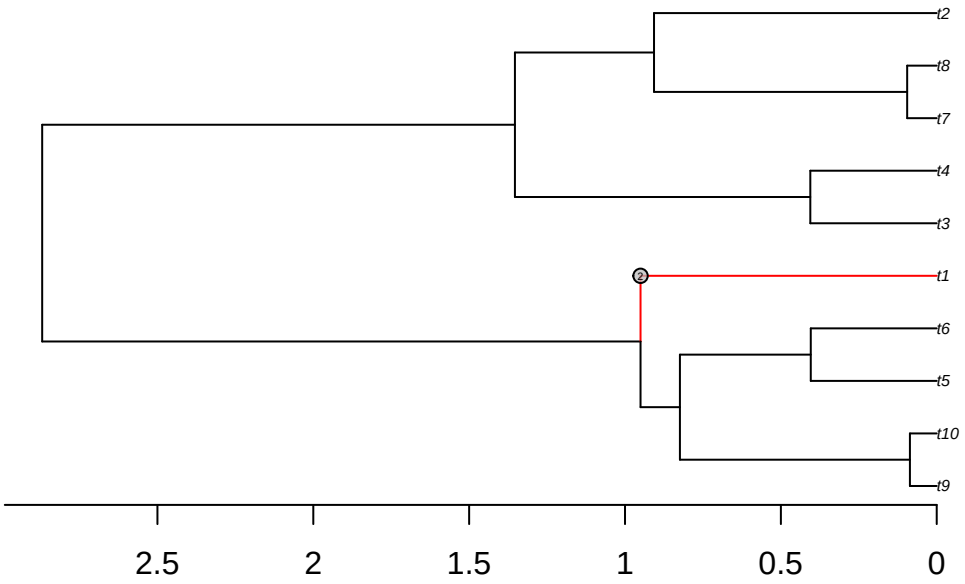
We can now give the taxon_richness as an argument to richness in the medusa function.

```
run2 <- medusa(phy = test_tree, richness = taxon_richness)
```

```
## Appropriate  aicc-threshold for a tree of 10 tips is: 0.
##
## Step 1: lnLik=-21.1075; aicc=46.96499; model=bd
## Step 2: lnLik=-14.53688; aicc=36.67376; shift at node 5; model=yule; cut=stem; # shifts=1
##
## No significant increase in aicc score. Disregarding subsequent piecewise models.
##
##   Model.ID Shift.Node Cut.At Model Ln.Lik.part        r epsilon     r.low
## 1        1         11   node  yule   -9.634925 0.815164      NA 0.3725519
## 2        2          5   stem  yule   -4.901956 4.116900      NA 2.5929860
##     r.high
## 1 1.517214
```

```
## 2 7.119733
```

```
# Plot medusa rate shifts
plot(run2, show.tip.label = T)
```



Homework

Explore the diversity dynamics of your clade. Use at least two of the three methods we discussed in lab. Make sure to give the evolutionary questions as well as the hypotheses you want to test. Remember, you want your project to be question-based, not method based. Write up a lab report in the usual format for next week. Make sure to cite the primary literature in the correct format. Include at least *two* figures, and give p-values, likelihood scores and/or AIC/AICc scores and cutoff values in your results.

The work you do for this lab assignment can be used in your final project as well. Therefore, conducting a literature search on the diversification dynamics of you clade (different from trait evolution) first will be helpful in identifying potential interesting subclades. Make sure to state your hypotheses in your lab report.