Introduction to Brownian Motion

Comparative Biology and Macroevolution

April 19, 2019

You have already been introduced to Brownian Motion in lecture. Today we will go through how to simulate Brownian Motion in R, as well as see how this can be used to understand continuous trait evolution.

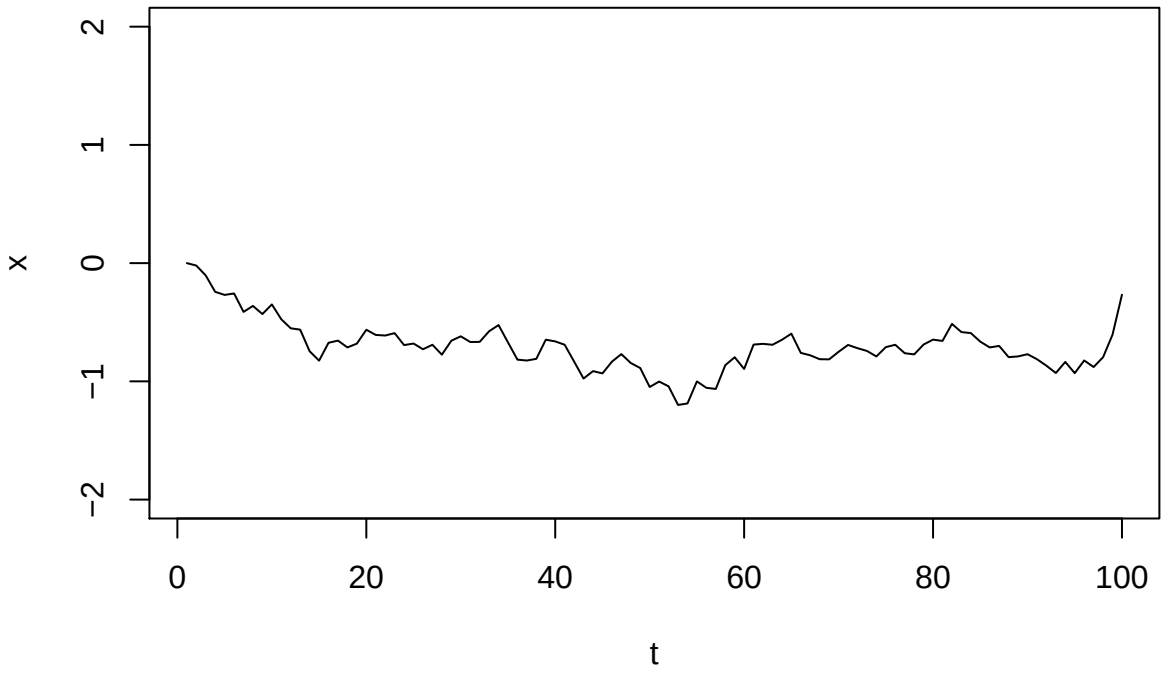
```
library(ape)
library(phytools)
library(car)
```

Lab

Simulating Brownian Motion

Lets start by simulating a single realization of the BM process in R. In this lab, I have tried to accomplish all the procedures in this lab using for loops. Therefore, there are definitely more efficient ways of producing the same results.

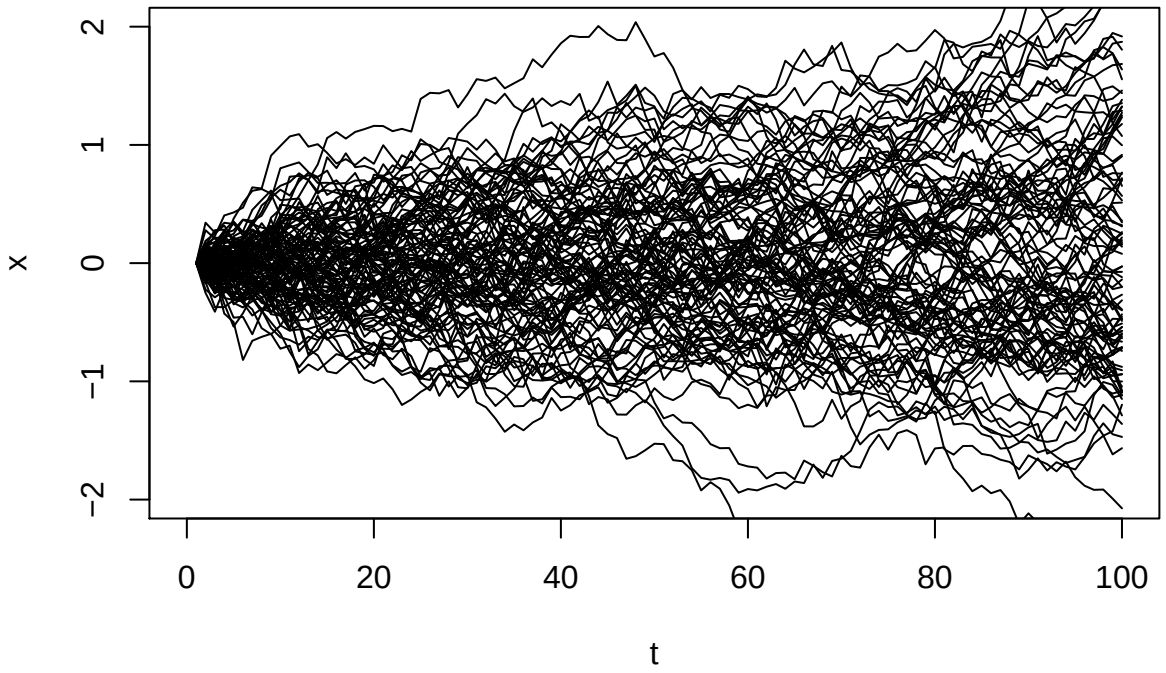
```
# Setting up our parameters
sig2 <- 0.01 # sigma^2
n_steps <- 1:100 # number of steps (time)

# Fill a vector with a bunch of small steps
vec <- numeric(100)  # Going to fill this with each time step
for (i in n_steps){
  small_step <- rnorm(n = 1, sd = sqrt(sig2))
  vec[i+1] <- vec[i] + small_step
}
plot(n_steps, vec[1:100], type = "l", ylim = c(-2, 2), ylab = "x",xlab = "t")
```



Notice that there is no tree involved.

Now lets try to plot multiple runs

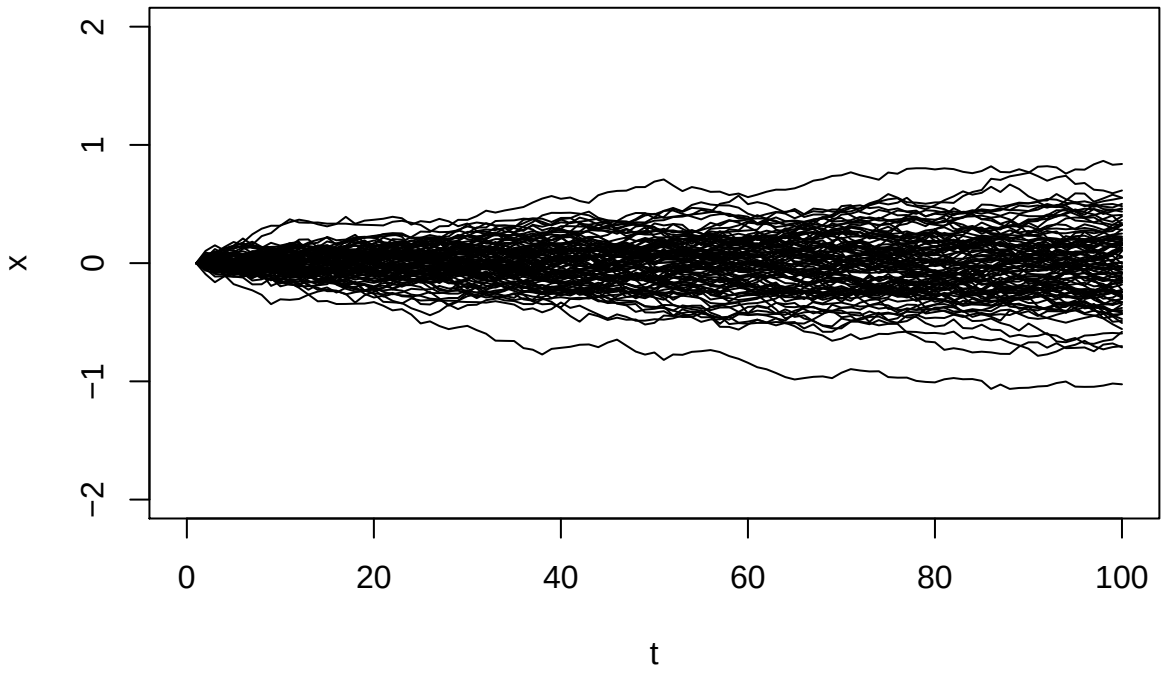
```
# Plot an empty plot
plot(NULL, xlim = c(0, 100), ylim = c(-2, 2), ylab = "x",xlab = "t")
# Put the entire BM for loop inside another for loop
for (j in 1:100) {
    vec <- numeric(100)
  for (i in n_steps){
    small_step <- rnorm(n = 1, sd = sqrt(sig2))
    vec[i+1] <- vec[i] + small_step
  }
  lines(n_steps, vec[1:100])
}
```



This plot shows 100 realizations of a trait evolving under a BM model. If the species tree was a star phylogeny, then we would have just simulated character evolution on that tree for the given time and σ^2.

Question: What would happen if you made the rate parameter smaller?

```
# Plot an empty plot
plot(NULL, xlim = c(0, 100), ylim = c(-2, 2), ylab = "x",xlab = "t")
sig2 <- 0.001
# Put the entire BM for loop inside another for loop
for (j in 1:99) {
    vec <- numeric(100)
  for (i in n_steps){
    small_step <- rnorm(n = 1, sd = sqrt(sig2))
    vec[i+1] <- vec[i] + small_step
  }
  lines(n_steps, vec[1:100])
}
```



We can see that the variance grows with time under BM.

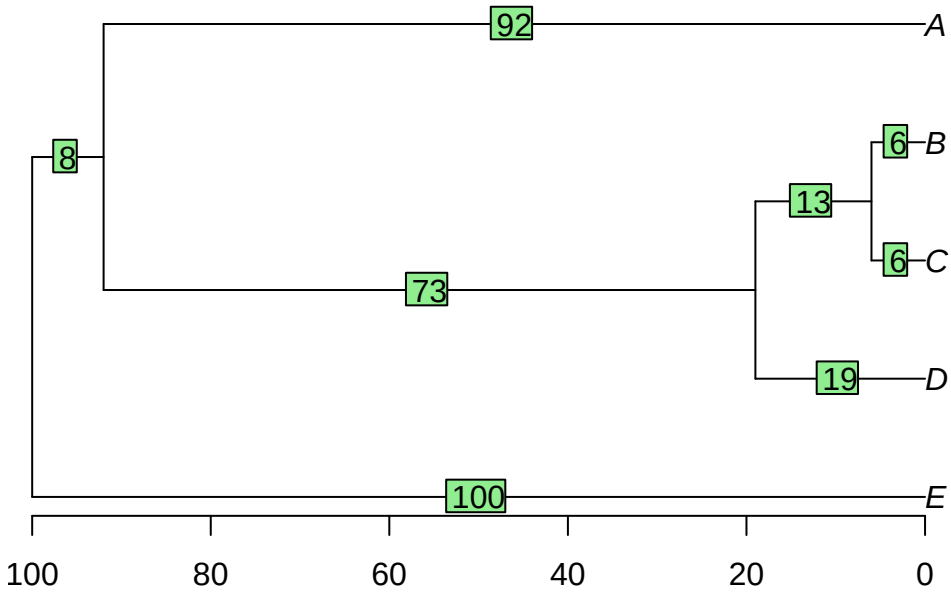
Simulating BM on a Tree

Remember, that we want to use BM as a way of modeling continuous trait evolution. In comparative biology, we usually do not work with star phylogenies. Instead, we have a tree that shows some lineages being more closely related to each other. Therefore, this phylogenetic relatedness will influence the way characters are expected to evolve under BM. Here lets begin to explore using simulatins in R.

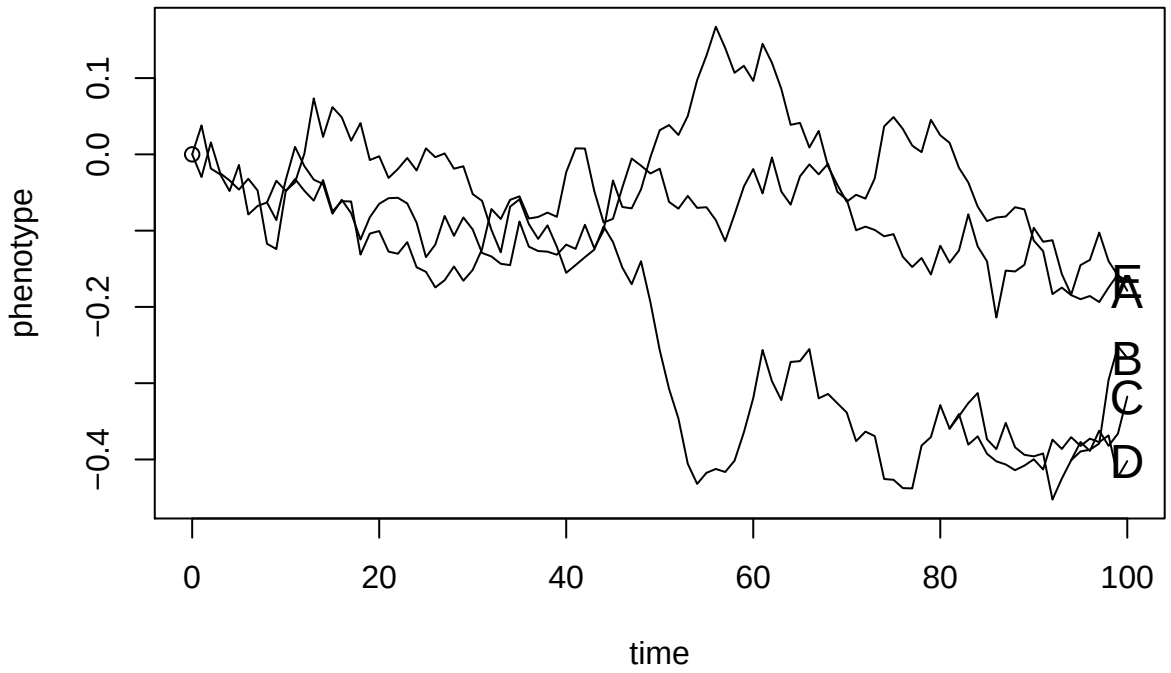
```
# Here we want to simulate a tree that is 100 myrs old and has 5 taxa

t <- 100  # total time
n <- 5  # total taxa
b <- (log(n) - log(2))/t
tree <- pbtree(b = b, n = n, t = t, type = "discrete", tip.label = LETTERS[5:1])
```

```
## simulating with both taxa-stop (n) and time-stop (t) is
## performed via rejection sampling & may be slow
##
##   1 trees rejected before finding a tree
```

```
plot(tree)
axisPhylo()
edgelabels(round(tree$edge.length, 4), cex = 1)
```




```
## From Liam Revell
## simulate evolution along each edge
X <- lapply(tree$edge.length, function(x) c(0, cumsum(rnorm(n = x, sd = sqrt(sig2)))))
## reorder the edges of the tree for pre-order traversal
cw <- reorder(tree)
## now simulate on the tree
ll <- tree$edge.length + 1
for (i in 1:nrow(cw$edge)) {
    pp <- which(cw$edge[, 2] == cw$edge[i, 1])
    if (length(pp) > 0)
        X[[i]] <- X[[i]] + X[[pp]][ll[pp]] else X[[i]] <- X[[i]] + X[[1]][1]
}
## get the starting and ending points of each edge for plotting
H <- nodeHeights(tree)
## plot the simulation
plot(H[1, 1], X[[1]][1], ylim = range(X), xlim = range(H), xlab = "time", ylab = "phenotype")
for (i in 1:length(X)) lines(H[i, 1]:H[i, 2], X[[i]])
## add tip labels if desired
yy <- sapply(1:length(tree$tip.label), function(x, y) which(x == y), y = tree$edge[,
    2])
yy <- sapply(yy, function(x, y) y[[x]][length(y[[x]])], y = X)
text(x = max(H), y = yy, tree$tip.label, cex = 1.5)
```



This simulation tracks the displacement of the phenotype through every point in time from the root to the tips. However, if we don't care about this history of displacement, we can easily simulate BM evolution on any tree following the steps given in lecture.

For each branch:

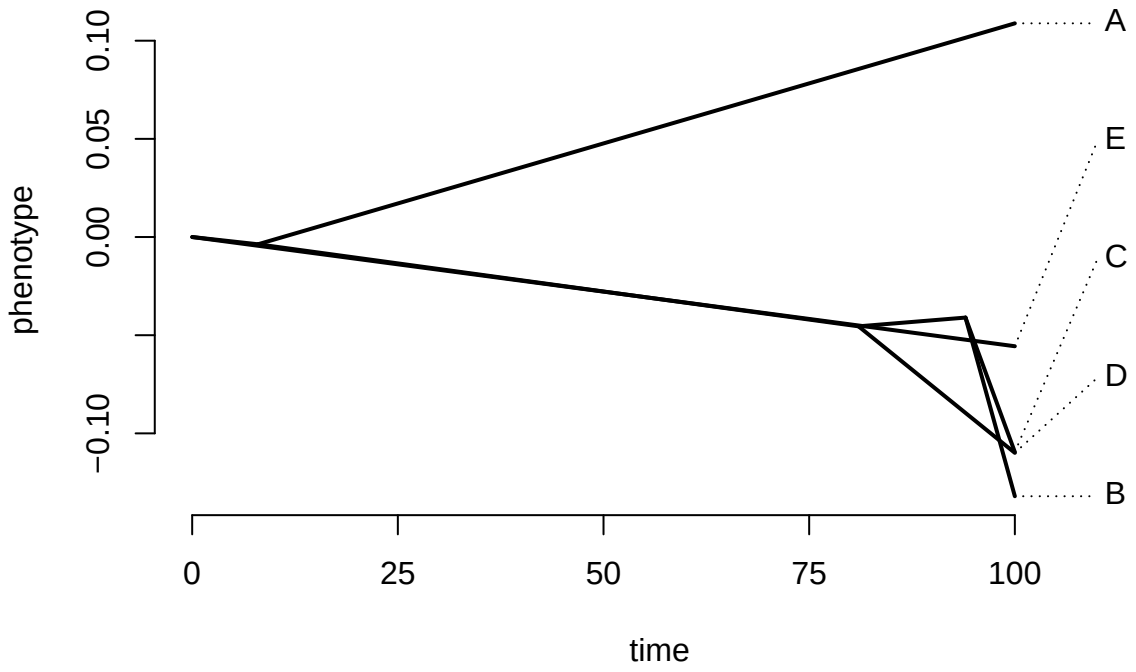
1. Get the length of the branch
2. Draw from a normal distribution with mean at 0 and variance = σ^2 * branch length. This is the amount of trait evolution that has occured on the branch.

Then for each tip, you sum the values you drew along the path from the root to that tip.

The function fastBM (below) is fast because it uses the algorithm above instead of simulating every time point (like we did in the earlier code block).

Note: fastBM will sometimes choke on small trees so if you see a weird graphic below try reexcuting the code.

```
## simulate Brownian evolution on a tree with fastBM
x <- fastBM(tree, sig2 = sig2, internal = TRUE)
## visualize Brownian evolution on a tree
phenogram(tree, x, spread.labels = TRUE, spread.cost = c(1, 0))
```



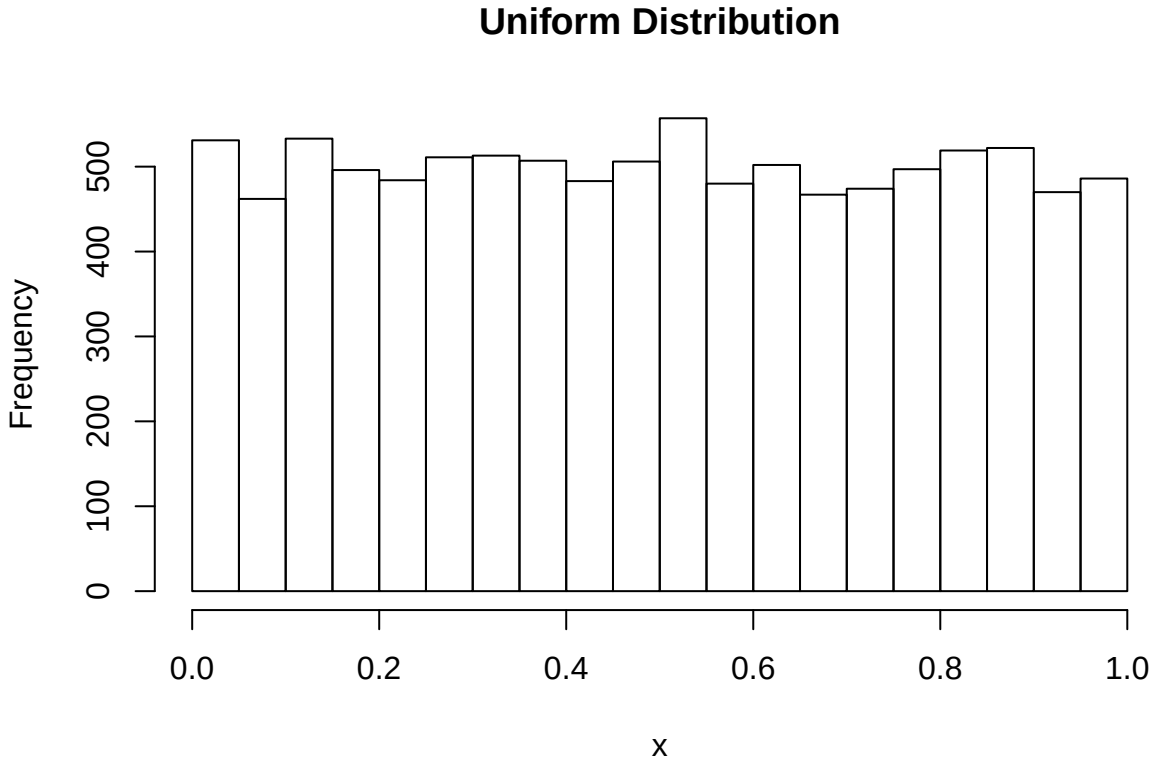
Cool! We were able to quickly simulate BM on a tree!

An aside on the CLT

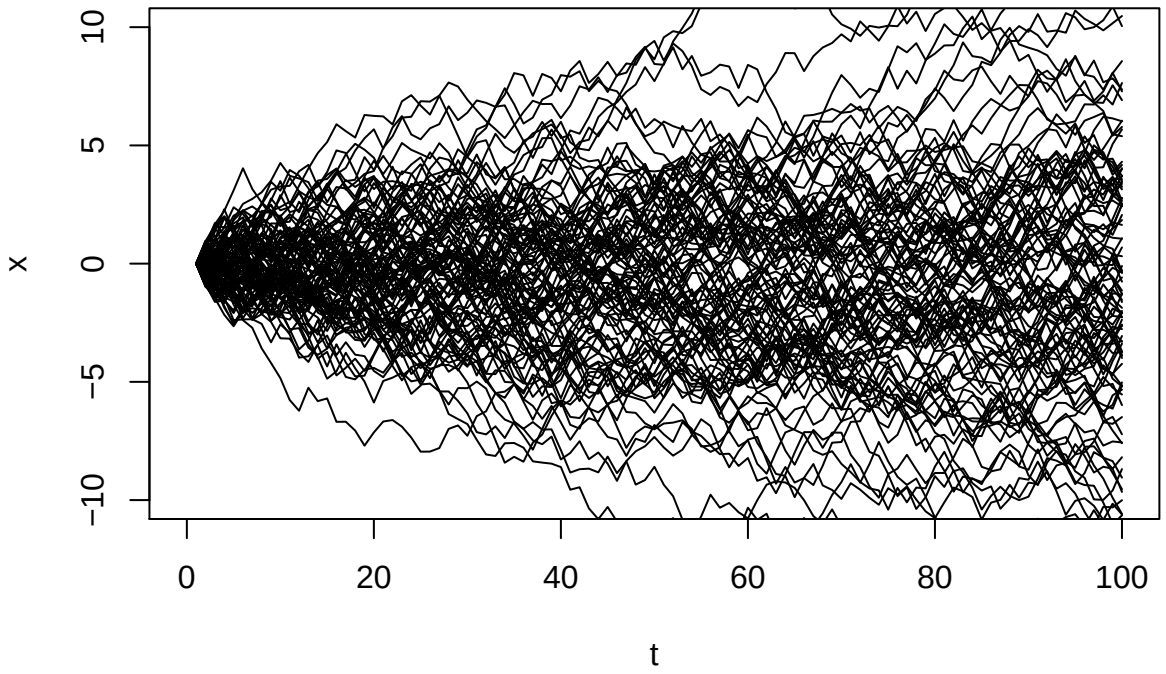
What if at each step, the traits were evolving under different distributions?

Here we will simulate the traits evolving under a uniform distribution

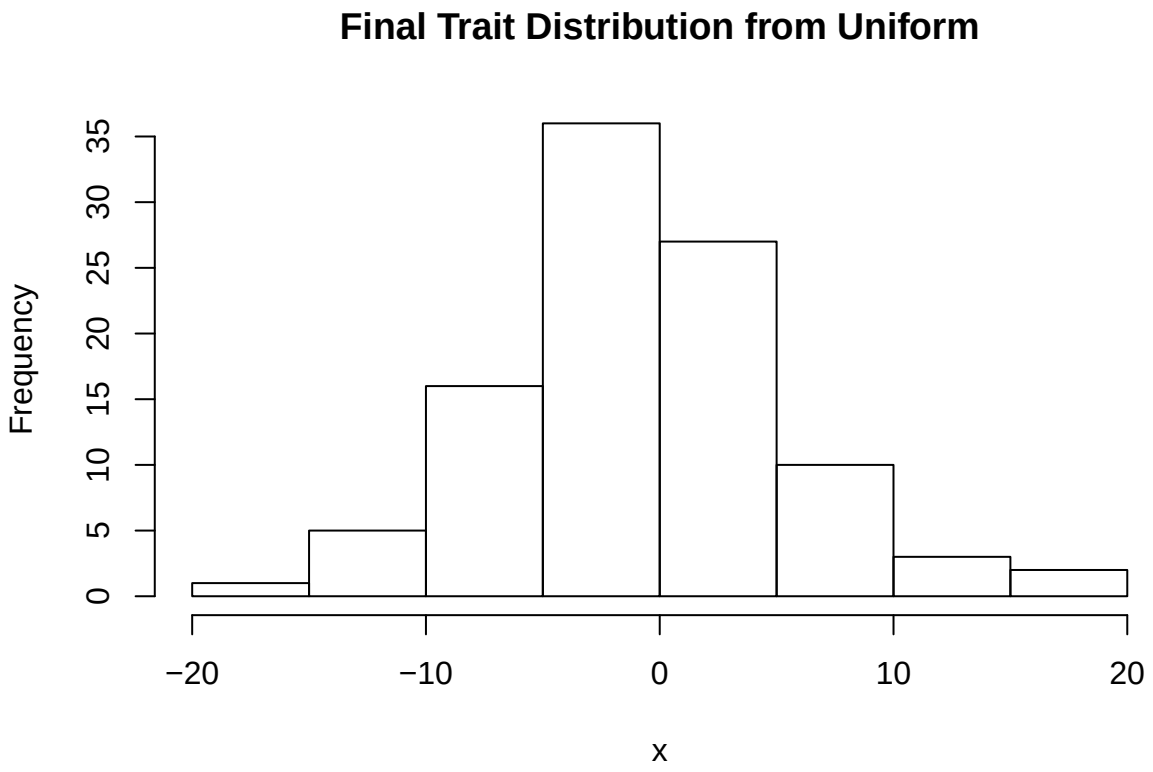
```
hist(runif(10000), main = "Uniform Distribution", xlab = "x")
```



```r
# Plot an empty plot
plot(NULL, xlim = c(0, 100), ylim = c(-10, 10), ylab = "x",xlab = "t")
final_vec <- numeric(100)

# Put the entire BM for loop inside another for loop
for (j in 1:100) {
    vec <- numeric(100)
  for (i in n_steps){
    # Have each step be drawn from a
    # uniform distribution instead of a normal distribution
    small_step <- runif(n = 1, min = -1, max = 1)
    vec[i+1] <- vec[i] + small_step
  }
  lines(n_steps, vec[1:100])
  final_vec[j] <- vec[100] # Save last value of each realization
}
```



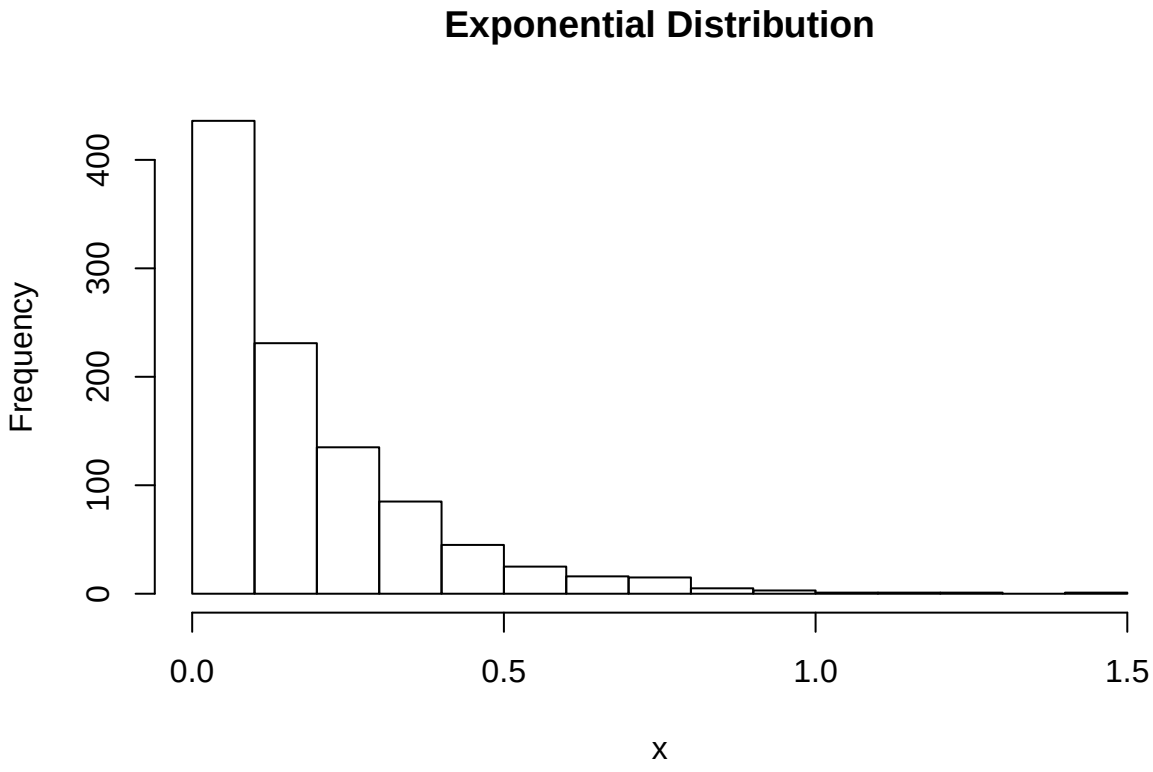

```r
hist(final_vec, xlab = "x", main = "Final Trait Distribution from Uniform")
```



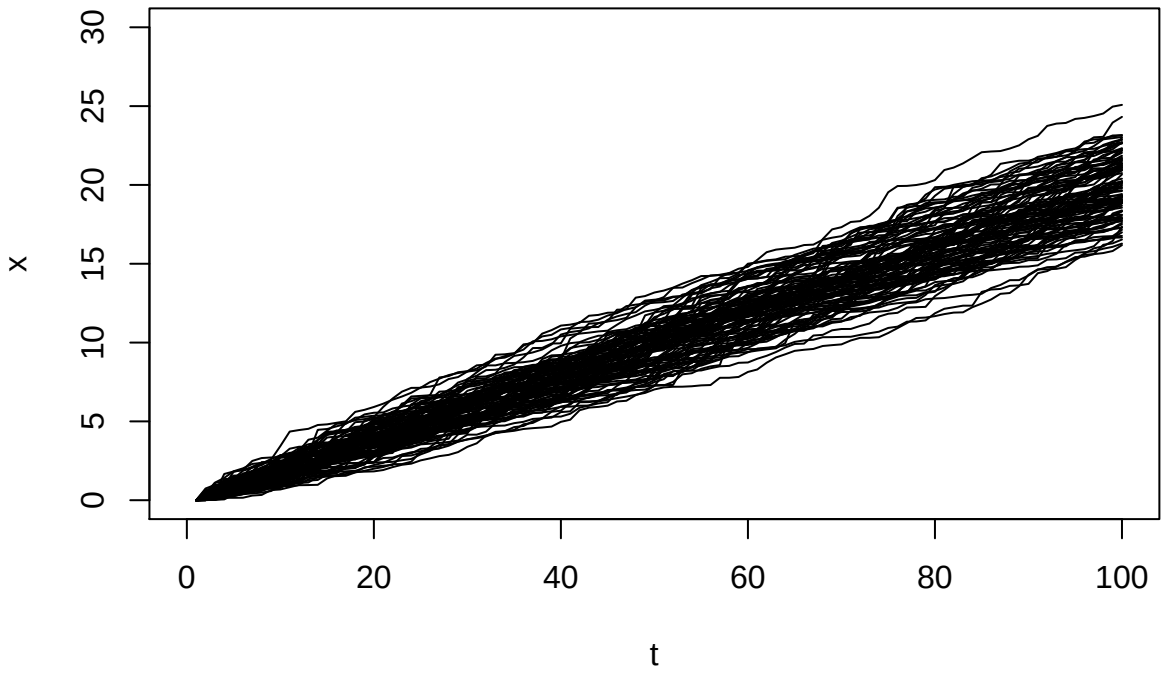
** Question: What is the distribution of the realizations of the traits?**

Here we will simulate traits evolving under an exponential distribution

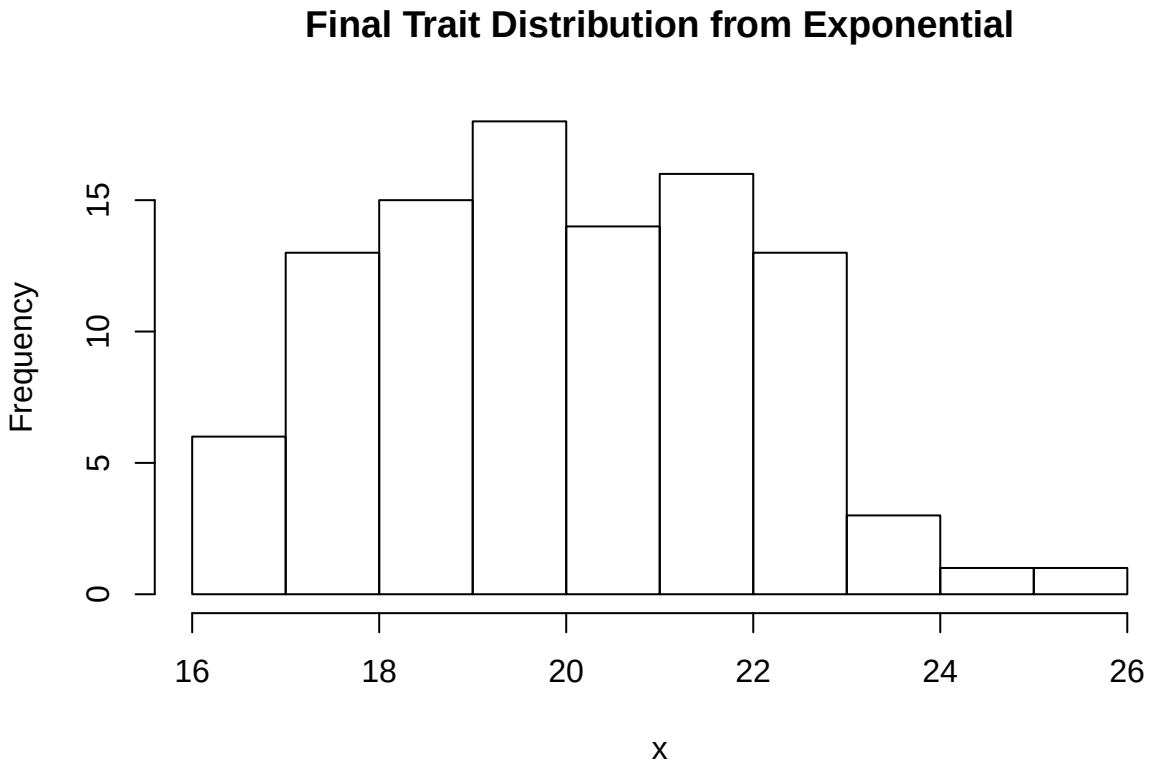
```
hist(rexp(1000, rate = 5), main = "Exponential Distribution", xlab = "x")
```



```r
# Plot an empty plot
plot(NULL, xlim = c(0, 100), ylim = c(0, 30), ylab = "x",xlab = "t")
final_vec <- numeric(100)
# Put the entire BM for loop inside another for loop
for (j in 1:100) {
    vec <- numeric(100)
  for (i in n_steps){
    # Have each step be drawn from a
    # uniform distribution instead of a normal distribution
    small_step <- rexp(1, rate = 5)
    vec[i+1] <- vec[i] + small_step
  }
  lines(n_steps, vec[1:100])
  final_vec[j] <- vec[100] # Save last value of each realization
}
```




```r
hist(final_vec, breaks = 7, main = "Final Trait Distribution from Exponential", xlab = "x")
```



Question: What is the distribution of the realizations of the traits?

Trait Covariance and the Phylogenetic Variance - Covariance Matrix

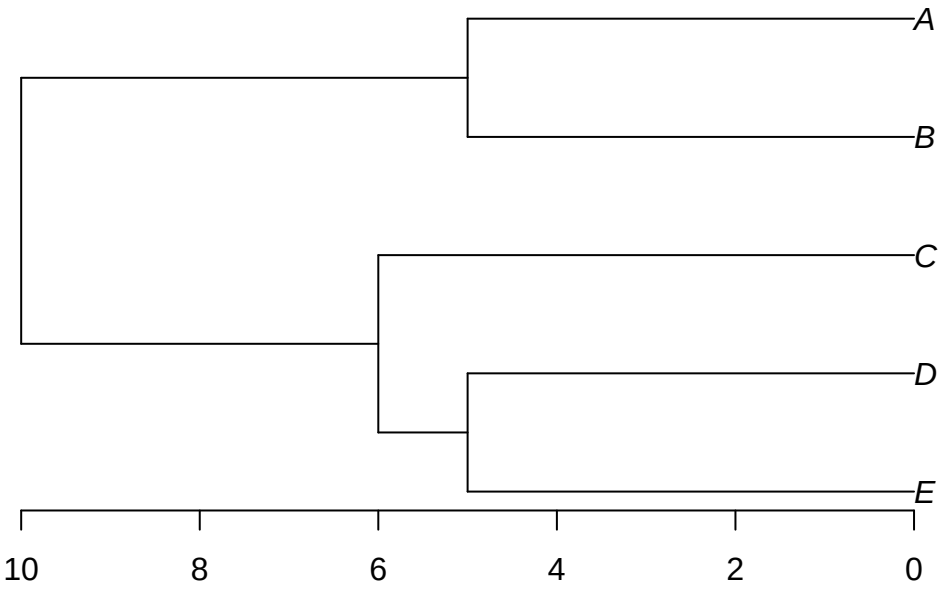
The phylogeny introduces co-variation among traits. We can explore this in simulation by evolving traits over the same phylogeny repeatedly and plotting the covariation in values. Independent branches will tend to evolve trait values that are uncorrelated with other values on the tree. Closely related species will tend to evolve similar values.

In order to see this lets simulate tree, and then simulate 500 realizations of BM on it.

```
# Simulate a tree 10 myrs old with 5 taxa
t <- 10  # total time
n <- 5  # total taxa
b <- (log(n) - log(2))/t
tree <- pbtree(b = b, n = n, t = t, type = "discrete", tip.label = LETTERS[5:1])
```

```
## simulating with both taxa-stop (n) and time-stop (t) is
## performed via rejection sampling & may be slow
##
##   6 trees rejected before finding a tree
```

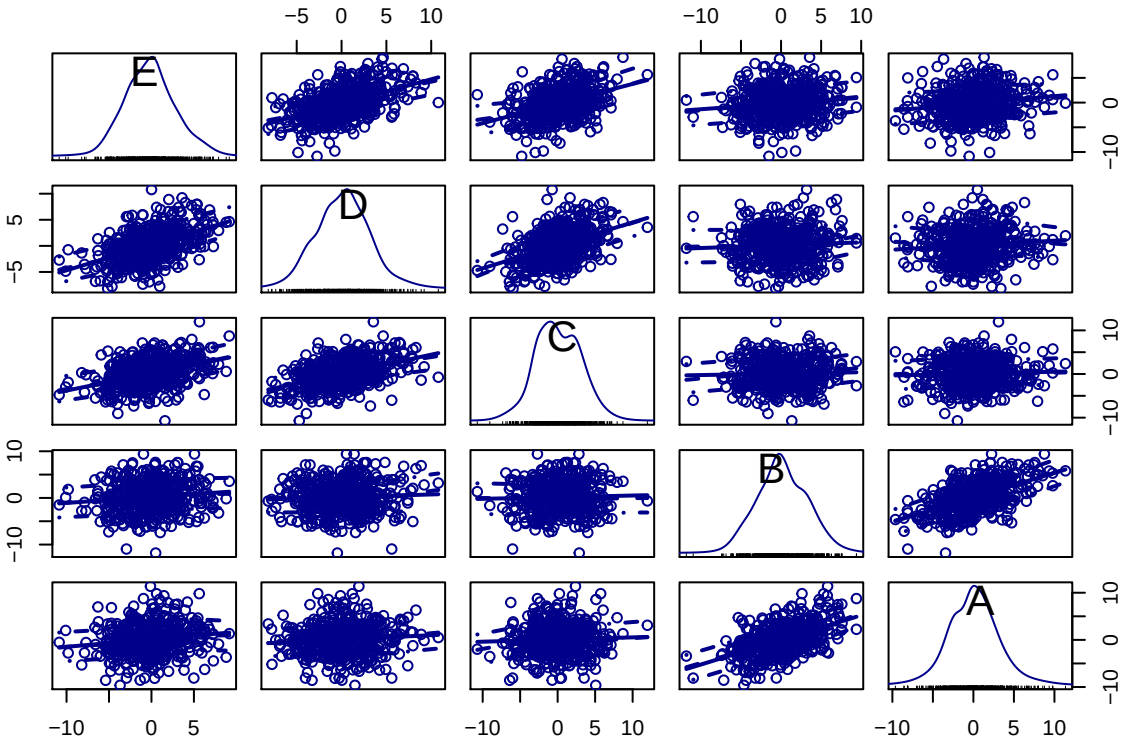
```
plot(tree)
axisPhylo()
```



Question: The traits for which taxa would you expect to covary more?

Now to simulate BM on the tree

```
X <- fastBM(tree, nsim = 500)
X <- t(X)
colnames(X) <- tree$tip.label
scatterplotMatrix(X, col = "darkblue")
```

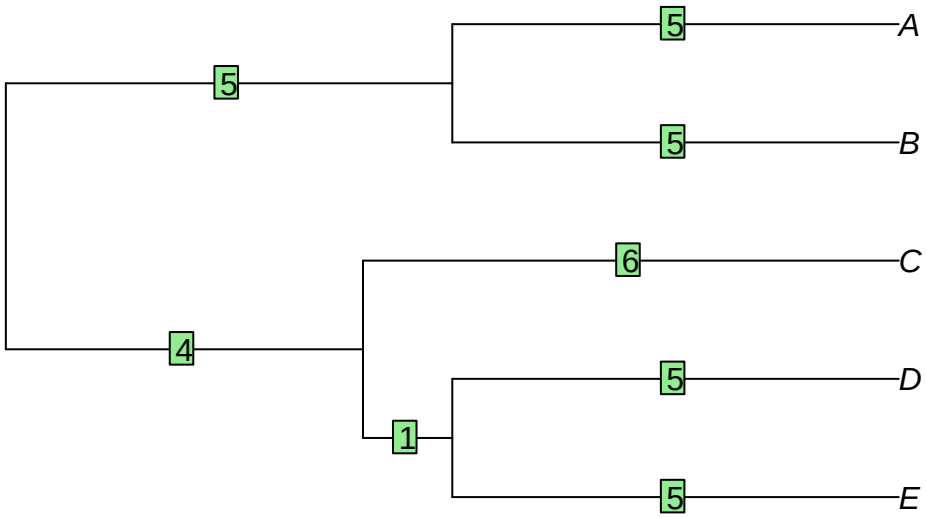


Here we will introduce the phylogenetic variance - covariance matrix.

For a phylogenetic tree with n taxa, the phylogenetic variance - covariance matrix will be an n by n matrix. Each row and column of the matrix corresponds to one of the n taxa in the matrix.

The diagonals of the matrix are the total distances from that tip to the root of the tree. The off diagonals are the total branch lengths shared by a pair of taxa.

```
plot(tree)
edgelabels(round(tree$edge.length, 4), cex = 1)
```




```
vcv.phylo(tree, anc.nodes = FALSE)
```

```
##    E  D  C  B  A
## E 10  5  4  0  0
## D  5 10  4  0  0
## C  4  4 10  0  0
## B  0  0  0 10  5
## A  0  0  0  5 10
```

Fitting BM Models

So far we have focused on explaining the evolutionary process under BM. However, what we really want to know is, given trait data for species, and a phylogeny for those species, what are the **parameters of the BM model**. Remember that BM has θ (ancestral value or phylogenetic mean) and σ^2 (rate). Our best estimate of θ turns out to be exactly what we would get if we use the contrasts algorithm to infer the value of the trait at the root.

In order to get the best estimate for σ^2, we will use maximum likelihood. You will go over maximum likelihood in class, but today we will be introduced to the concept.

Likelihood: Probability of obtaining the observed data given the model and its parameter estimates

Pr(Data | Model) : Viewed as a function of the data, model is fixed

L(Model | Data) : Viewed as a function of the model, data is fixed

The maximum-likelihood (ML) estimator of the rate of evolution (σ^2) under Brownian motion is given to us from O'Meara et al. 2006.

Eq. 1

$$\hat{\sigma}^2 = \frac{[X - E(X)]' C^{-1} [X - E(X)]}{N}$$

Where X is a vector of the tip values, $E(X)$ is the expected values at the root, C is the phylogenetic variance-covariance matrix, and N is the number of tips.

Additionally:

- $[X - E(X)]'$ is the *transpose* of the tip values minus the root value
- C^{-1} is the inverse of the phylogenetic variance-covariance matrix

We can also get the **likelihood** for any value of σ^2, also from O'Meara et al. 2006

Eq. 2

$$log(L) = log\left[\frac{exp\left\{-\frac{1}{2}[X - E(X)]'V^{-1}[X - E(X)]\right\}}{((2\pi)^N * det(V)}\right]$$

V here is called the **rate matrix**. It is simply the proposed value of σ^2 multiplied by the phylogenetic variance-covariance matrix. So, **Eq. 2** is saying that the likelihood (or rather the log of the likelihood) is a function of the difference between the **tip** and **root** values and the **rate matrix**, V (which is simply the vcv matrix multiplied by some value of σ^2 that we wish to evaluate).

We use can use **Eq 2** to write a function in R that will allow us to compare likelihoods given some tree and and a σ^2.

The arguments of our function:

- The phylogenetic variance-covariance matrix
- The inverse of the vcv matrix
- σ^2
- The root state
- A vector of the tips values for the trait

```
BMlk <- function(C, inv.C, sigmasq, root.state, data) {

    N <- length(data); # the number of tips
    EX <- rep(root.state, N) # creates a vector of the expected trait value - which under BM is the roo
    V <- C * sigmasq; # multiply the entries in C by the BM rate
    inv.V <- inv.C * sigmasq ^-1; # do the same for the inverted matrix using the inverse of the rate

    lnlNum<- -0.5*(data - EX) %*% inv.V %*% (data - EX)
    lnlDen<- log(sqrt((2*pi)^N*det(V)))
    L<-lnlNum-lnlDen
    return(L);
}
```

Note that we are going to have to do a little work before we can use our function. We need to pass it a vcv matrix and the inverse of the matrix, we need to pass in a root state value, and we need to pass a value of σ^2.

1. simulate a 5 taxon tree
2. simulate BM on that tree with a known σ^2 of 0.5
3. compare the likelihood of the generating σ^2 to a smaller σ^2 of 0.005.

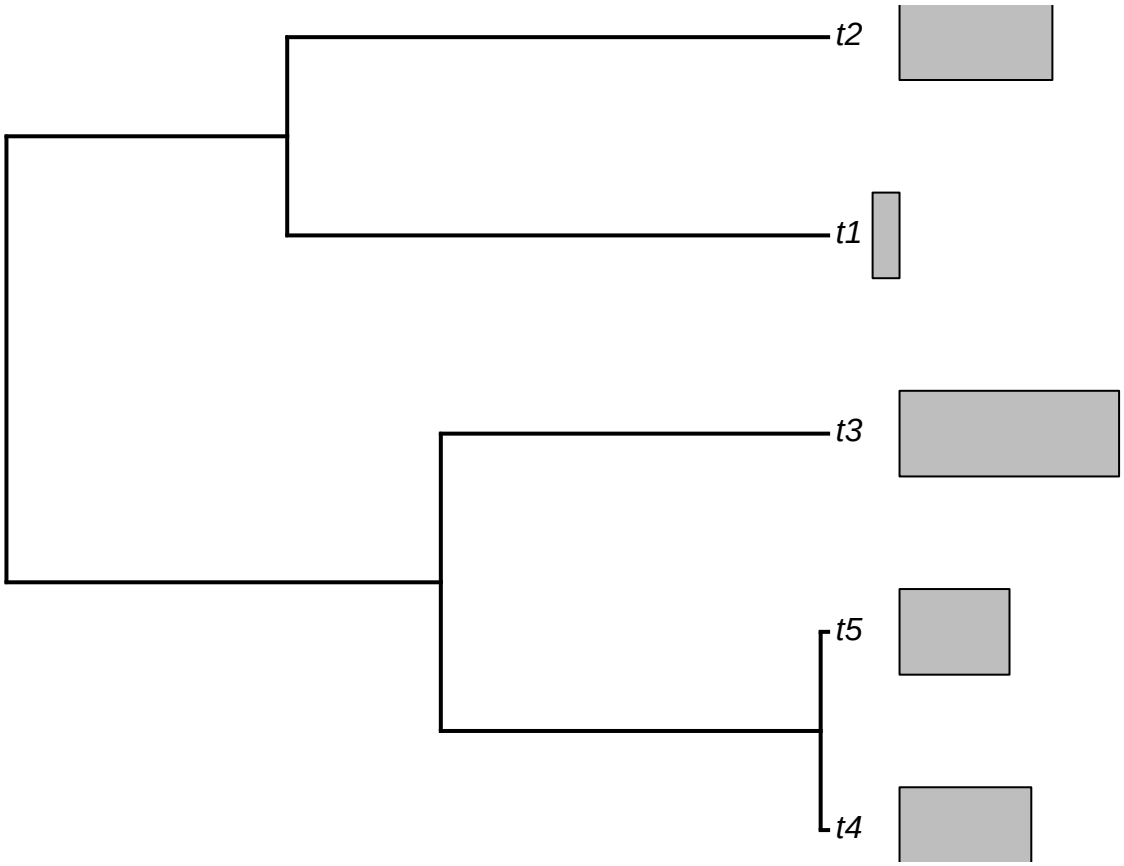
```
t <- 100  # total time
n <- 5  # total taxa
b <- (log(n) - log(2))/t
tree <- pbtree(b = b, n = n, t = t)
```

```
## simulating with both taxa-stop (n) and time-stop (t) is
## performed via rejection sampling & may be slow
##
##   1 trees rejected before finding a tree
```

```
# Simulate BM on the tree with a sig2 of 0.5 and z0 of 0
data <- fastBM(tree, sig2 = 0.5, a = 0)
data
```

```
##        t4        t5        t3        t1        t2
##  5.155342  4.303189  8.590954 -1.054632  5.980693
```

```
plotTree.wBars(tree, data, tip.labels = TRUE)
```



We need to calculate the phylogenetic variance-covariance matrix.

```
c <- vcvPhylo(tree, anc.nodes = FALSE)

# Get inverse of phylogenetic variance-covariance matrix
inv.c <- solve(c)
```

```
estimate_1 <- BMlk(c, inv.c, sigmasq = 0.5, root.state = 0, data  = data) # the true value for sigsq
estimate_1
```

```
##           [,1]
## [1,] -13.7461
```

```
estimate_2 <- BMlk(c, inv.c, sigmasq = 0.005, root.state = 0, data  = data)
estimate_2
```

```
##           [,1]
## [1,] -161.1126
```

Question: Which parameter value fits better?

Remember when comparing likelihoods, the higher the number, the better the likelihood (because the likelihood of a hypothesis is proportional to the probability of the data given that the hypothesis is true). **Smaller** (more negative) log likelihoods are also smaller likelihoods (raise e to the power of the competing likelihoods to check this) so we can see that our data prefers the the generating σ^2 value to the alternative.

```
exp(estimate_1)
```

```
##             [,1]
## [1,] 1.07188e-06
```

```
exp(estimate_2)
```

```
##              [,1]
## [1,] 1.070716e-70
```

We can also try other values of σ^2 and calculate the log likelihood.

```
sigma_0.3 <- BMlk(c, inv.c, sigmasq = 0.3, root.state = 0, data  = data)
sigma_0.4 <- BMlk(c, inv.c, sigmasq = 0.4, root.state = 0, data  = data)
sigma_0.5 <- BMlk(c, inv.c, sigmasq = 0.5, root.state = 0, data  = data)
sigma_0.6 <- BMlk(c, inv.c, sigmasq = 0.6, root.state = 0, data  = data)
sigma_0.3
```

```
##           [,1]
## [1,] -13.53893
```

```
sigma_0.4
```

```
##           [,1]
## [1,] -13.58945
```

```
sigma_0.5
```

```
##          [,1]
## [1,] -13.7461
```

```
sigma_0.6
```

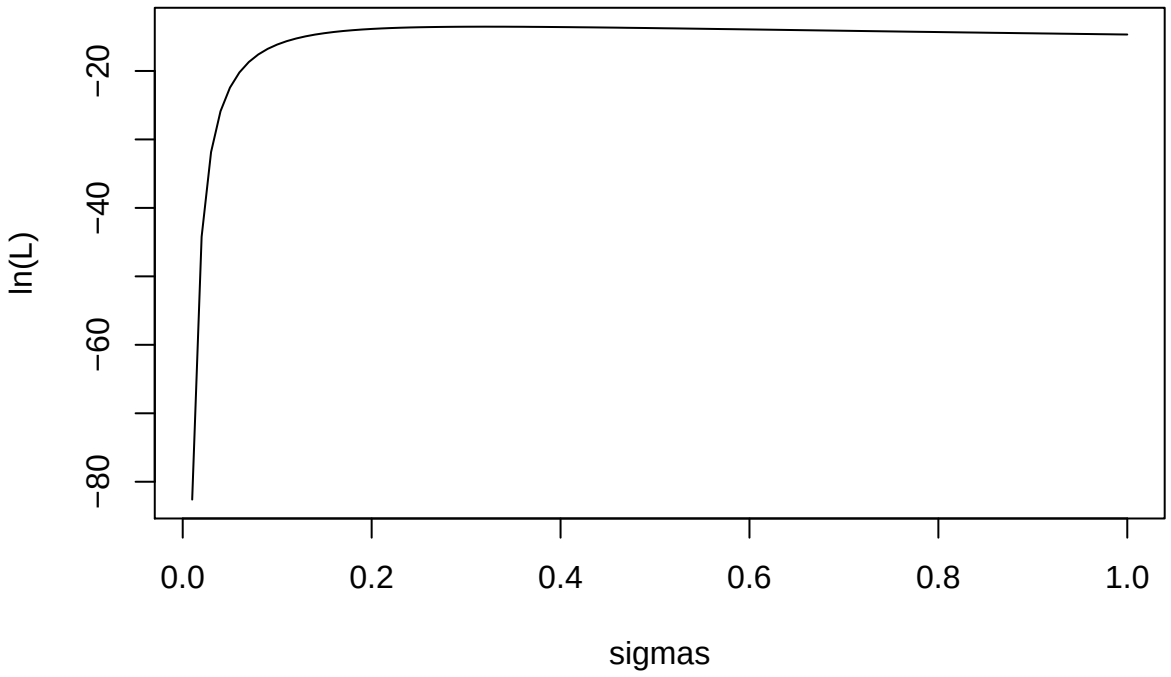
```
##           [,1]
## [1,] -13.93443
```

Which value has the highest log likelihood?

Going through each value of σ^2 would be tedious. Therefore, it might be better visualize the likelihood surface.

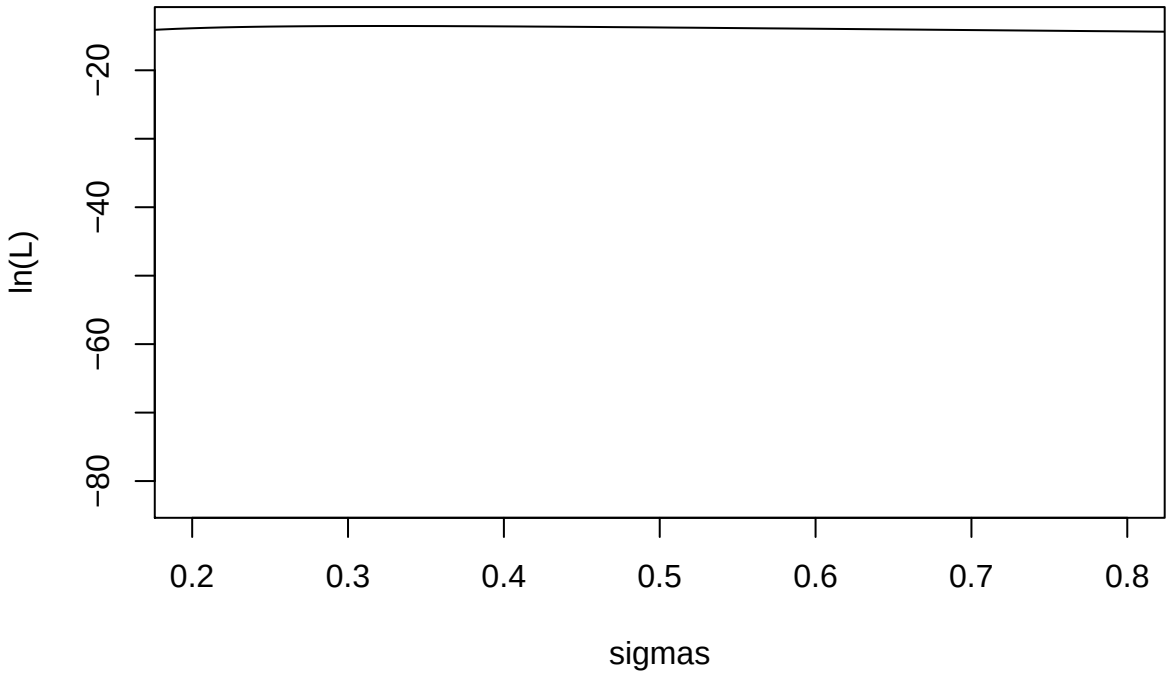
We can look at the likelihood surface around the generating value.

```
vals <- numeric(100)
sigmas <- (1:100)/100
for(i in 1:100){
  vals[i] <- BMlk(c, inv.c, sigmasq = sigmas[i], root.state = 0, data  = data)
}
plot(sigmas, vals,type = "l", ylab = "ln(L)")
```



Zoom in around the generating σ^2 value.

```
plot(sigmas,vals, xlim = c(0.2, 0.8), type = "l", ylab = "ln(L)")
```



The likelihood surface is quite flat around the generating value!

Bonus

Is the log likelihood value we calculated actually the maximum log likelihood value?

```
library(geiger)
bm_fit <- fitContinuous(tree, data, model = "BM")
bm_fit
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'BM' model parameters:
##  sigsq = 0.211426
##  z0 = 4.422059
##
##  model summary:
##  log-likelihood = -12.489426
##  AIC = 28.978853
##  AICc = 34.978853
##  free parameters = 2
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  frequency of best fit = 1.00
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
ML_z0 <- bm_fit$opt$z0
ML_sig2 <- bm_fit$opt$sigsq
BMlk(c, inv.c, sigmasq = ML_sig2, root.state = ML_z0, data  = data)
```

```
##           [,1]
## [1,] -12.48943
```

Homework

This weeks homework will not be in a lab format. Instead, please answer the following questions. Make sure to include your name and lab section (1A or 1B). Also make sure to label your figures in your HW.

1. Read in a time-calibrated tree from your clade. Plot your clade with a time scale and tip labels (make it look clean!).
2. Simulate a data set under Brownian motion with your tree and a rate value that you choose (use the `fastBM()` function) with a σ^2 and root state value of your choosing. Make sure to report the generating rate value as well as the root state value in your HW.
3. Visualize the tree with the trait values you have simulated. Provide the resulting plot in your HW (Hint, there are multiple ways of doing this, we went over more than one way in lab to visualize this data).
4. Fit 5 different Brownian rates to your tree (including the true rate) and compare the log likelihoods using the `BMlk()` function. Provide the log likelihood values for each rate in your HW as a table. Which rate fit the best? How can you tell?
5. Visualize the likelihood surface around your true rate. Provide the plot of the likelihood surface in your HW. Here, keep the root state value constant. Describe the likelihood surface.

6. Find a continuous trait that you might be able to use for your final project. Write a paragraph justifying why the evolution trait you chose is of interest to your clade. In your paragraph, give examples of taxa within you clade that vary in your chosen trait.

Hint: Think about traits that spans the diversity of your clade. For example, if I was studying felids (lions and tigers), skull shape would be a poor choice, since they all pretty much have the same skull shape compared to the more phenotypically diverse canids.