Just How Well Sampled Were the Dicynodonts Anyway?

DWB

How often would we expect ancestors given rates from fossilized-birth-death models?

Instantaneous rates per lineage * million-years: - birth/origination ~= death/extinction = ~0.06 - sampling events = 0.259

This is a suspiciously high sampling rate for a terrestrial vertebrate clade. For comparison look at table in Bapst & Hopkins: this is comparable to sampling rate for global record of bryozoans or cephalopods across the Phanerozoic.

For example, look at the high taxonomic completeness this predicts:

```r
library(paleotree)
```

```
## Loading required package: ape
```

```r
qsRate2Comp(q=0.06, r = 0.259)
```

```
## [1] 0.8119122
```

Nevertheless, to use Foote (1996) equations, we need the per-interval probability of being sampled at least once.

```r
pqsRate2sProb(p=0.06, q=0.06, r = 0.259, int.length = 1)
```

```
## [1] 0.2287588
```

(Note this is for the default scenario where we are estimating per-interval sampling probability for 1 million year intervals.)

We can then input this value in `probAnc` and calculate the probability of getting indirect ancestors under budding using Foote 1996 equations.

(Why budding? Because its probably the predominant way that speciation among morphotaxa occurred in most groups, and why not dicynodonts?)

```r
probAnc(p = 0.06, q = 0.06, R = 0.22876,
    mode = "budding",
    analysis = "indirectDesc", Mmax = 85)
```

```
## Treat result with caution:
##  if p = q, then prob of a taxon being an
##  ancestor should be no greater than 0.5.

## Values higher than 0.5 result from
##  limits of finite calculates, particularly
##  with high sampling probabilities.

## See documentation.
## [1] 0.8531633
```

HmmmmmmmMMMMmmmmmmm. This number seems high, as it realistically should never be calculated above 0.5. Foote's approximation isn't working out too hot for the dicynodonts. Maybe the FBD is estimating too high of a sampling rate?

Calculating Sampling Rates from FAD/LAD Info from the PBDB

Species-level Data

```
# species
searchURL <- url("https://paleobiodb.org/data1.2/taxa/list.txt?taxon_name=dicynodontia&rank=species&tax

dicynData <- read.csv(file = searchURL)
```

Yay, data on some dicynodont species.

For Foote's interval-duration-frequency methods to work (Foote & Raup, 1996; Foote, 1997) then we need to put the intervals into some sort of system where they are of roughly even length. This turns out not to be too tough, just staring at IUGS chronostrat. Each 'unit' of duration is about as long as the longer stages, about 5 million years.

```
# remove taxa with "" as earliest interval
dicynData <- dicynData[dicynData$early_interval != "",]

intervals <- c(
    dicynData$early_interval, dicynData$late_interval)
unqInt <- unique(intervals)

intTranslateTable <- cbind(
    sort(unqInt),
    c( NA, # ""
    4, # "Anisian"
    1, # "Capitanian"
    5, # "Carnian"
    2, # "Changhsingian"
    3, # "Early Triassic"
    0, # "Guadalupian"
    3, # "Induan"
    4, # "Ladinian"
    7, # "Late Triassic"
    2, # "Lopingian"
    4, # "Middle Triassic"
    6, # "Norian"
    3, # "Olenekian"
    7, # "Rhaetian"
    9, # "Sinemurian"
    0, # "Wordian"
    2  # "Wuchiapingian"
    ))

# translate intervals to numbers
dicynData$firstInt <- sapply(dicynData$early_interval,
    function(x) intTranslateTable[
        x == intTranslateTable[,1] ,2]
    )

dicynData$lastInt <- sapply(dicynData$late_interval,
    function(x) intTranslateTable[
        x == intTranslateTable[,1] ,2]
    )

dicynData$lastInt[
    is.na(dicynData$lastInt)] <- dicynData$firstInt[
    is.na(dicynData$lastInt)]

anyNA(dicynData$firstInt)
```

```
## [1] FALSE
```

```
anyNA(dicynData$lastInt)
```

```
## [1] FALSE
```

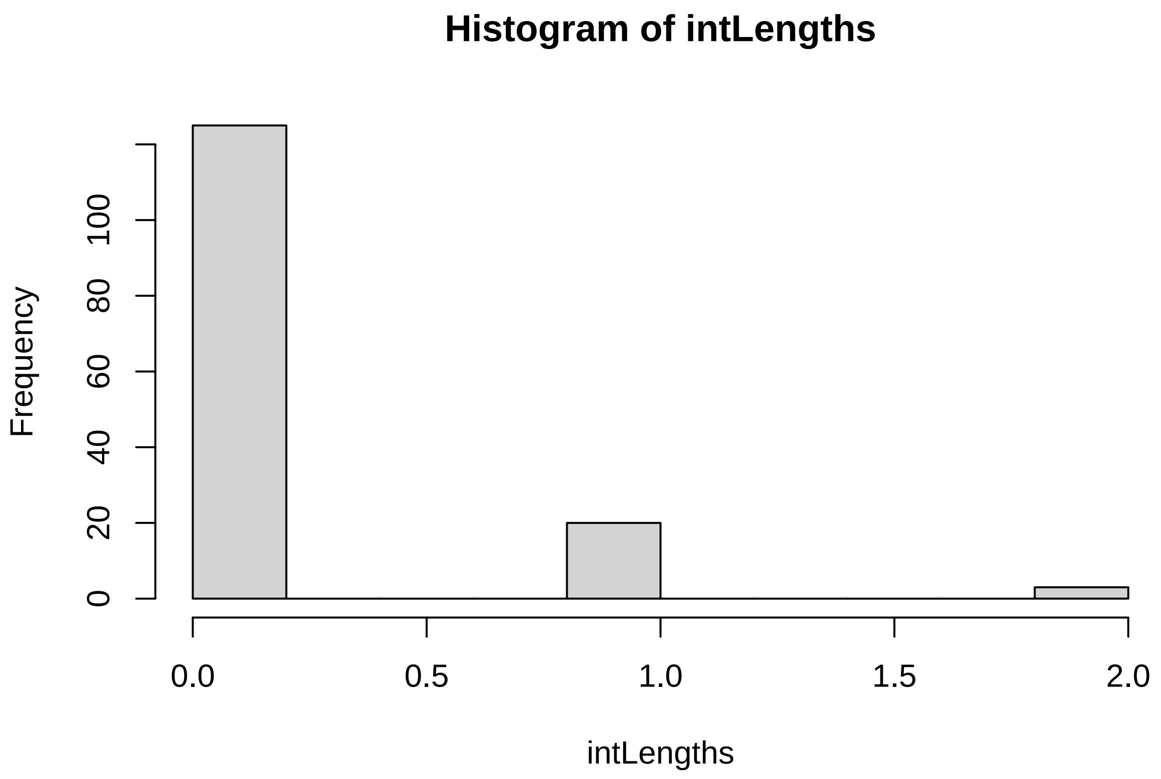
If there are no NAs, we're in business. (NA was used a placeholder for taxa that go extinct in the same interval… or for FADs that are simply unplaceable).

What does the resulting durations look like?

```
intLengths <- as.numeric(dicynData$lastInt) - as.numeric(dicynData$firstInt)
table(intLengths)
```

```
## intLengths
##   0   1   2
## 125  20   3
```

```
#  intLengths
#   0   1   2
# 124  20   3

hist(intLengths)
```



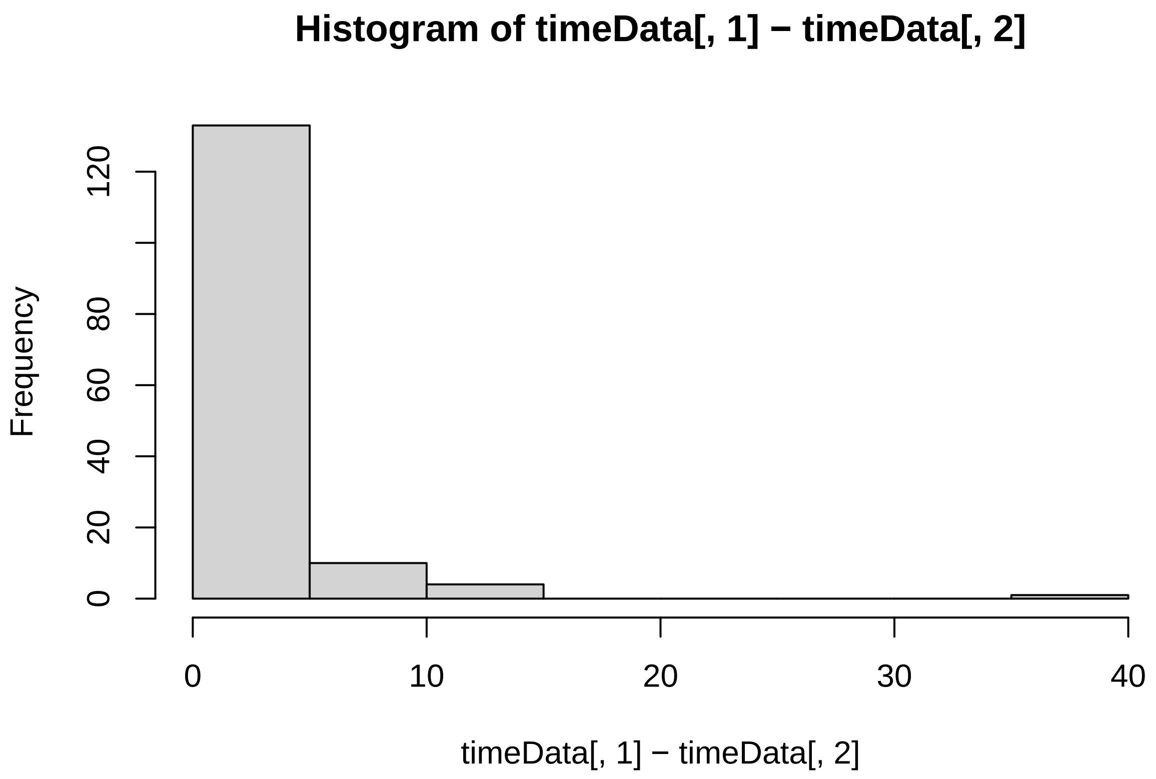
Hmm, okay, does it work for FreqRat?

```
#freqRat
f1 <- table(intLengths)["0"]
f2 <- table(intLengths)["1"]
f3 <- table(intLengths)["2"]
freqRat <- (f2^2)/(f1 * f3)
# 1.075 - violates the model assumptions
```

Nope, returns a probability that exceeds 1. How could that be?

Let's try using the FADs/LADs themselves (mid-point dates from each interval) and fit the continuous-duration model from Foote (1997).

```
# cont time model
timeData <- data.frame(
    FADmean = (dicynData$firstapp_min_ma + dicynData$firstapp_max_ma)/2,
    LADmean = (dicynData$lastapp_min_ma + dicynData$lastapp_max_ma)/2
    )
row.names(timeData) <- dicynData$taxon_name

hist(timeData[,1]-timeData[,2])
```



Okay, a little more structure than the discrete intervals. So let's fit the model!

```
likFun <- make_durationFreqCont(timeData)
optim(parInit(likFun),
      likFun,
      lower = parLower(likFun),
      upper = parUpper(likFun),
      method = "L-BFGS-B",
      control = list(maxit = 1000000)
      )
```

```
## $par
##        q.1        r.1
## 0.17473047 0.05014877
##
## $value
## [1] 169.1083
##
## $counts
## function gradient
##       41       41
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

So the maximum-likelihood model says q = 0.17, r = 0.05 Lmy-1.

Would the answer change much if we were looking at genera instead?

Genera level data

Let’s get genera data.

```
# genera

searchURL <- url("https://paleobiodb.org/data1.2/taxa/list.txt?taxon_name=dicynodontia&rank=genus&taxon

dicynData <- read.csv(file = searchURL)

# remove taxa with "" as earliest interval
dicynData <- dicynData[dicynData$early_interval != "",]

intervals <- c(
    dicynData$early_interval, dicynData$late_interval)
unqInt <- unique(intervals)

intTranslateTable <- cbind(
    sort(unqInt),
    c( NA, # ""
    4, # "Anisian"
    1, # "Capitanian"
    5, # "Carnian"
    2, # "Changhsingian"
    3, # "Early Triassic"
    0, # "Guadalupian"
    3, # "Induan"
    4, # "Ladinian"
    7, # "Late Triassic"
    2, # "Lopingian"
    4, # "Middle Triassic"
    6, # "Norian"
    3, # "Olenekian"
    7, # "Rhaetian"
    9, # "Sinemurian"
    0, # "Wordian"
    2  # "Wuchiapingian"
    ))

# translate intervals to numbers
dicynData$firstInt <- sapply(dicynData$early_interval,
    function(x) intTranslateTable[
        x == intTranslateTable[,1] ,2]
    )

dicynData$lastInt <- sapply(dicynData$late_interval,
    function(x) intTranslateTable[
        x == intTranslateTable[,1] ,2]
    )

dicynData$lastInt[
    is.na(dicynData$lastInt)] <- dicynData$firstInt[
    is.na(dicynData$lastInt)]

anyNA(dicynData$firstInt)
```

```
## [1] FALSE
```

```
anyNA(dicynData$lastInt)
```

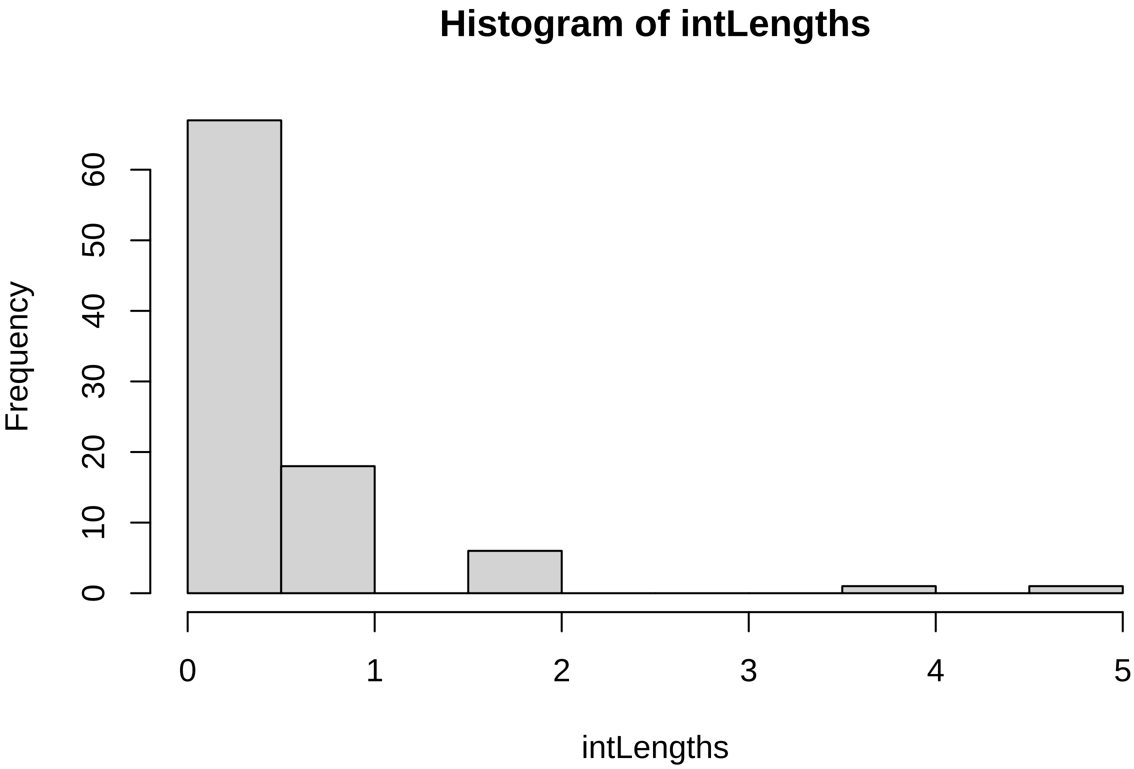
```
## [1] FALSE
```

No NAs? Looks good.

```
intLengths <- as.numeric(dicynData$lastInt) - as.numeric(dicynData$firstInt)
table(intLengths)
```

```
## intLengths
##  0  1  2  4  5
## 67 18  6  1  1
```

```
#  intLengths
#   0   1   2
# 124  20   3

hist(intLengths)
```



Okay…

```
#freqRat
f1 <- table(intLengths)["0"]
f2 <- table(intLengths)["1"]
f3 <- table(intLengths)["2"]
freqRat <- (f2^2)/(f1 * f3)
freqRat
```

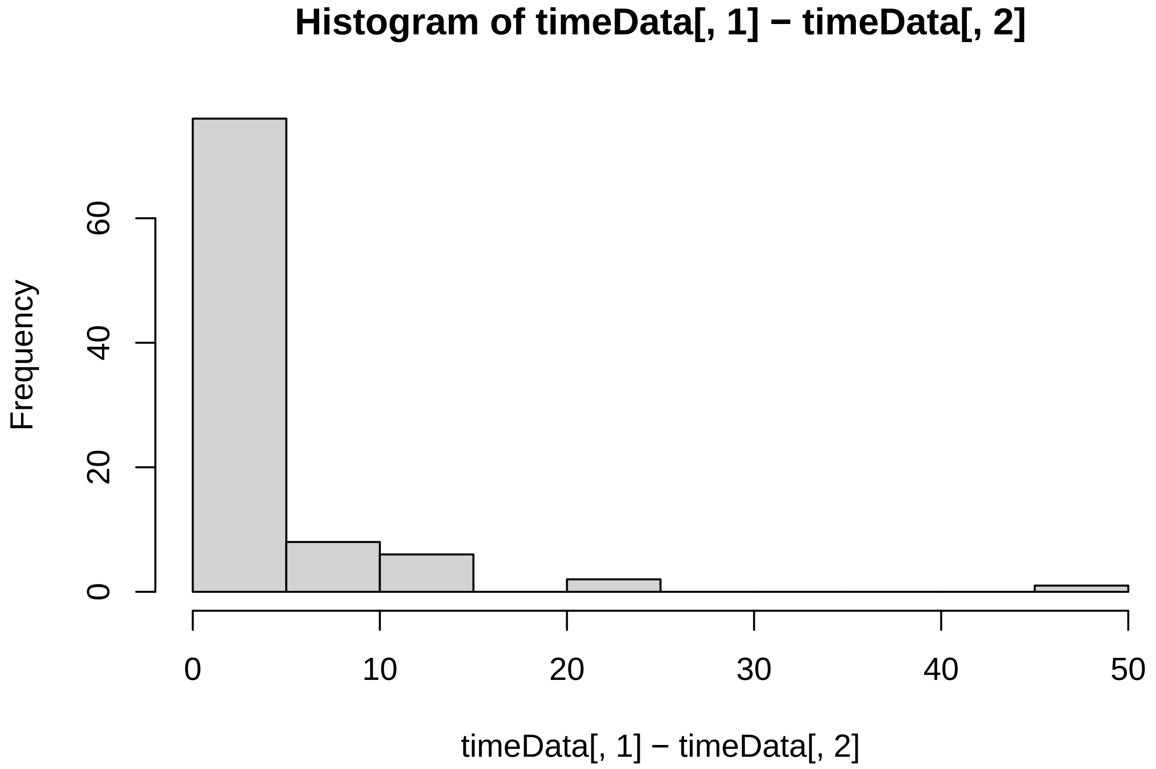
```
##         1
## 0.8059701
```

freqRat of 0.8 is pretty high! What does the continuous-duration model say?

```r
# cont time model
timeData <- data.frame(
    FADmean = (dicynData$firstapp_min_ma + dicynData$firstapp_max_ma)/2,
    LADmean = (dicynData$lastapp_min_ma + dicynData$lastapp_max_ma)/2
    )
row.names(timeData) <- dicynData$taxon_name
timeData[,1] >=timeData[,2]
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [76] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [91] TRUE TRUE TRUE
```

```r
hist(timeData[,1]-timeData[,2])
```



```r
library(paleotree)
likFun <- make_durationFreqCont(timeData)
optim(parInit(likFun),
      likFun,
      lower = parLower(likFun),
      upper = parUpper(likFun),
      method = "L-BFGS-B",
      control = list(maxit = 1000000)
      )
```

```
## $par
##        q.1        r.1
## 0.12780782 0.07365782
```

```
##
## $value
## [1] 165.0098
##
## $counts
## function gradient
##       58       58
##
## $convergence
## [1] 52
##
## $message
## [1] "ERROR: ABNORMAL_TERMINATION_IN_LNSRCH"
```

So q (ext) = 0.125, r (sampling) = 0.068 Lmy-1. This is pretty close to what we got for species level data from the PBDB, although extinction has come down some.

What happens if we feed these in to get sampling rate, assuming birth rate == death rate?

```
sRate2sProb(r = 0.06, int.length = 1)
```

```
## [1] 0.05823547
```

```
pqsRate2sProb(p = 0.13, q = 0.13, r = 0.06, int.length = 1)
```

```
## [1] 0.05830878
```

This R is much smaller than the above R. If we input this into `probAnc` like earlier…

```
probAnc(p = 0.13, q = 0.13, R = 0.058,
    mode = "budding",
    analysis = "indirectDesc", Mmax = 85)
```

```
## Treat result with caution:
##  if p = q, then prob of a taxon being an
##  ancestor should be no greater than 0.5.
```

```
## Values higher than 0.5 result from
##  limits of finite calculates, particularly
##  with high sampling probabilities.
```

```
## See documentation.
```

```
## [1] 0.5181893
```

Well, it's only a little above 0.51. But note that Foote's model involves the use of R, which makes it dependent on the length of intervals in question.

For example, what if we had longer, like 10 million year stages like what we've been applying the freqRat to?

```
sRate2sProb(r = 0.06, int.length = 10)
```

```
## [1] 0.4511884
```

```
pqsRate2sProb(p = 0.13, q = 0.13, r = 0.06, int.length = 10)
```

```
## [1] 0.4953339
```

We get much bigger R values. Overall, the discrepancy in the dicynodont sampling rates/probabilities as estimated by different methods seems to reflect a discrepancy suggesting the Permo-triassic stages are about 20 million years long.

```r
sProb2sRate(R = 0.8, int.length = 20)
```

```
## [1] 0.0804719
```

```r
sRate2sProb(r = 0.05, int.length = 20)
```

```
## [1] 0.6321206
```

But those stages definitely aren't 20 million years old. The models just don't have a lot of observables with good data so its noisy. More detailed methods need to be applied, and there may be heterogeneities structuring the data that is further complicating things. PyRate would be good to try.

What if we put these higher, say 10 million year probabilities of sampling a lineage at least once?

```r
probAnc(p = 0.06, q = 0.06, R = 0.45,
    mode = "budding",
    analysis = "indirectDesc", Mmax = 85)
```

```
## Treat result with caution:
##  if p = q, then prob of a taxon being an
##  ancestor should be no greater than 0.5.

## Values higher than 0.5 result from
##  limits of finite calculates, particularly
##  with high sampling probabilities.

## See documentation.

## [1] 0.9051003
```

Way too high!

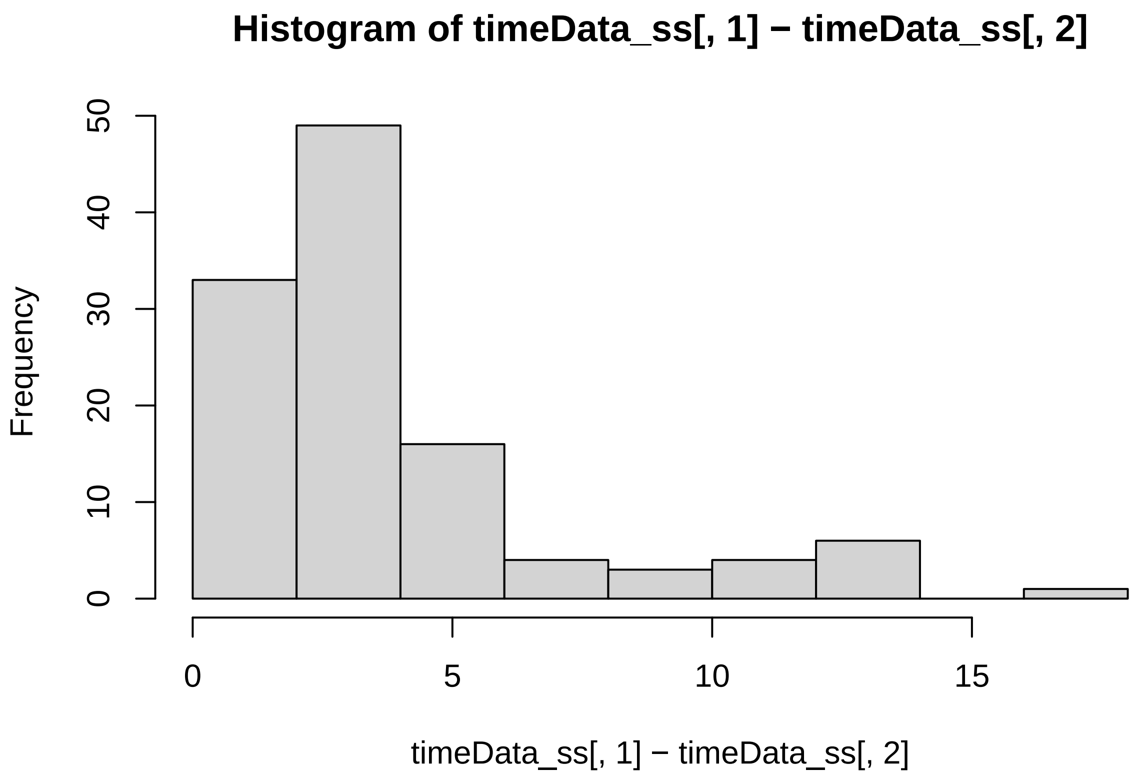
Is it an effect of taxon sub-sampling?

look at original data, what is sampling rate on those age ranges (uncertainties???)

```r
# cont time model
#sampledSpec_timeData <- read.csv("Therapsid_Ages_CFK_2023.csv")
sampledSpec_timeData <- read.table(
    "~/workspace/Continuous_CharacterTesting/data/Therapsid_Ages_CFK_2023.tsv",
    header = TRUE)
#str(sampledSpec_timeData)

timeData_ss <- sampledSpec_timeData[,2:3]
row.names(timeData_ss) <- dicynData$taxon

# as.numeric(timeData_ss[,1])
```

Can we calculate sampling rate from this? Are there one-hit taxa? or no?

```r
hist(timeData_ss[,1]-timeData_ss[,2])
```



Are there any negative values?

```
sampledSpec_timeData[(timeData_ss[,1]-timeData_ss[,2]) < 0,]
```

```
## [1] taxon   min_age max_age
## <0 rows> (or 0-length row.names)
```

No, good.

```
sum((timeData_ss[,1]-timeData_ss[,2]) == 0)
```

```
## [1] 0
```

No one-hits. Ruh-roh. Cannot fit the model!

Take species list see how many are in PBDB data

Get the list of species they use

```
sampledSpec_timeData <- read.table(
    "~/workspace/Continuous_CharacterTesting/data/Therapsid_Ages_CFK_2023.tsv",
    header = TRUE)
analyzedSpecies <- sampledSpec_timeData$taxon

# clean analyzed Species
analyzedSpecies <- sort(sub(pattern = "_", replacement = " ", x = analyzedSpecies))
```

This is an odd mix of genera and species. So we need to combine both from the PBDB.

```
# get species-level data from PBDB
searchURL <- url("https://paleobiodb.org/data1.2/taxa/list.txt?taxon_name=dicynodontia&rank=species&taxo
dicynDataSpecies <- read.csv(file = searchURL)
```

```
# remove bad duplicate D. latericeps

if(any(dicynDataSpecies$orig_no == "56871")){
   dicynDataSpecies <- dicynDataSpecies[dicynDataSpecies$orig_no != "56871",]
}

# get generic level data
searchURL <- url("https://paleobiodb.org/data1.2/taxa/list.txt?taxon_name=dicynodontia&rank=genus&taxon_
dicynDataGenera <- read.csv(file = searchURL)

# remove NA ages
dicynDataSpecies <- dicynDataSpecies[!is.na(dicynDataSpecies$firstapp_max_ma),]
dicynDataGenera <- dicynDataGenera[!is.na(dicynDataGenera$firstapp_max_ma),]

pbdbTaxa <- rbind(dicynDataSpecies, dicynDataGenera)
pbdbTaxonNames <- pbdbTaxa$taxon_name
```

how many matches do we have??

```
getMatches <- sapply(analyzedSpecies, function(x)
    if(any(x == pbdbTaxonNames)){
        which(x == pbdbTaxonNames)
    }else{
        NA
        }
    )

typeof(getMatches)
```

```
## [1] "integer"
```

Do any have more than one match?

```
sum(sapply(getMatches,length) > 1)
```

```
## [1] 0
```

No. How many have no matches?

```
noMatchNames <- analyzedSpecies[is.na(getMatches)]
length(noMatchNames)
```

```
## [1] 37
```

So, the phylogenetic analysis contains 116 taxa (species and genera), of which 79 have matches (0.68 match proportion).

> 01-14-24: So, the phylogenetic analysis contains 116 taxa (species and genera), of which 80 have matches (0.69 match proportion)
>
> This is now old data. We have one fewer matches due to NAs.

Can we get better?

NOTE DO NOT DO THE FOLLOWING. BROOM NAMED LOTS SPECIES THE SAME SPECIES NAME. WHAT A DAMN NIGHTMARE.

let's get dates from the PBDB for the taxa that do match

Need to construct `timeData`.

```r
matchedPBDBtaxonInfo <- pbdbTaxa[getMatches[!is.na(getMatches)],]

#timeData_4date <- matchedPBDBtaxonInfo[,
#    c("firstapp_max_ma", "firstapp_min_ma", "lastapp_max_ma", "lastapp_min_ma")]

timeData <- data.frame(
    FADmean = (matchedPBDBtaxonInfo$firstapp_min_ma + matchedPBDBtaxonInfo$firstapp_max_ma)/2,
    LADmean = (matchedPBDBtaxonInfo$lastapp_min_ma + matchedPBDBtaxonInfo$lastapp_max_ma)/2
    )

str(timeData)
```

```
## 'data.frame':    79 obs. of  2 variables:
##  $ FADmean: num  235 245 257 264 257 ...
##  $ LADmean: num  232 245 256 262 257 ...
```

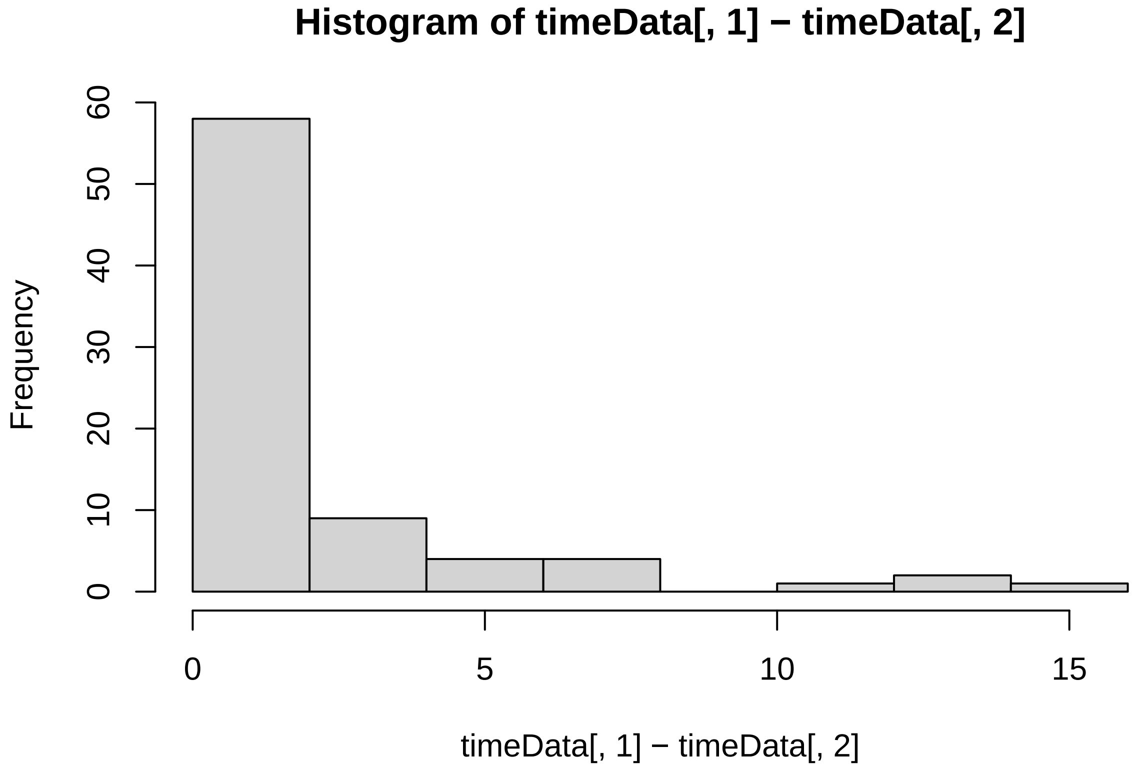
```r
row.names(timeData) <-matchedPBDBtaxonInfo$taxon_name

#matchedPBDBtaxonInfo[duplicated(matchedPBDBtaxonInfo$taxon_name),]
```

Does it look good?

```r
all(timeData[,1] >=timeData[,2])
```

```
## [1] TRUE
```

```r
hist(timeData[,1]-timeData[,2])
```



How many one-hits do we have?

```r
sum(timeData[,1]-timeData[,2] == 0)
```

```
## [1] 50
```

That's... okay. That's good.

```
likFun <- make_durationFreqCont(timeData)
optim(parInit(likFun),
      likFun,
      lower = parLower(likFun),
      upper = parUpper(likFun),
      method = "L-BFGS-B",
      control = list(maxit = 1000000)
      )
```

```
## $par
##       q.1       r.1
## 0.2135806 0.1238817
##
## $value
## [1] 125.7032
##
## $counts
## function gradient
##       22       22
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

What does that mean for completeness?

```
q_est <- 0.21
r_est <- 0.12

r_est / (q_est + r_est)
```

```
## [1] 0.3636364
```