Haplorhini Diversity, Speciation, and Morphology

Youstina Salama

Abstract

In discussion of primate evolution, different of opinions have persisted regarding the palaeobiology and phylogenetic relationships of early primates due to disparities in the fossil record (Ni *et al.* 2013). In this paper I will explore Haplorhini speciation and diversification, along with their morphology using data from an online database named PaleoBiology Database. Using RStudio and Python methods, I will attempt to reconstruct some graphs generated by an R package called PaleoBioDB. I will also use a python program called PyRate which estimates speciation, extinction, and preservation rates from fossil occurrence data using a Bayesian framework (Silvestro *et al.* 2014). In addition to providing further support for the morphological difference between different clades, the analysis shows an increase in speciation richness but a decrease in speciation rate through time.

Introduction

Haplorhini is a clade which includes the tarsiers and the simians or anthropoids. The simians include catarrhines, which are Old World monkeys, apes, and humans, and the platyrrhines or New World monkeys. Haplorhini diverged about 60-65 million years ago. About 60 million years ago, the infraorder Tarsiiformes diverged from the haplorhines. The New World monkeys branched off from catarrhines about 40 mya, while the Hominoidea split from Old World monkeys about 25 mya. Both of the geographically separated lineages are characterized by distinct nose shape, overall structure, and mannerism with which they move. The nostrils of the New World monkeys are wide and open. The nostrils of the Old World Monkeys are narrower. Some species have long tails for climbing balance while others have shorter tails or no tails at all such as apes. Some have equal length limbs, while other do not. Some walk on four limbs such as monkeys, while others walk on two only such as Humans. The apes are the closest relatives to humans, but the exact location of humans in terms of classification remains disputable.

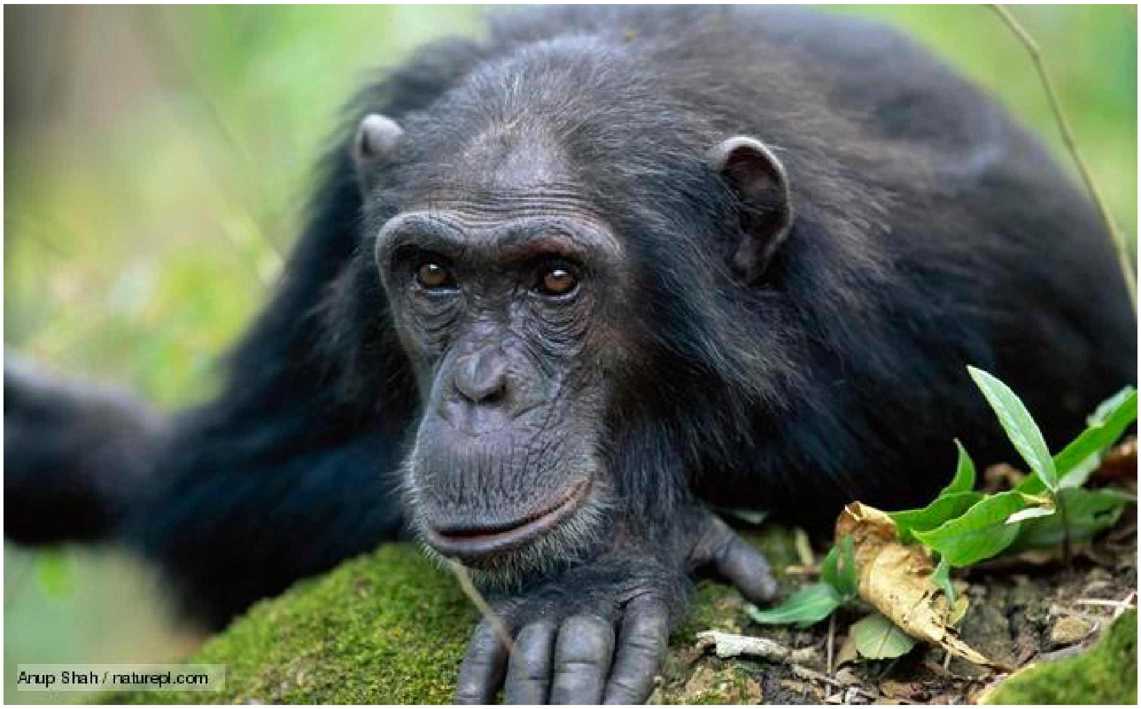


Figure 1: Haplorhini Image

Methods

Eploring Occurrence Data Using PaleoBioDB Package

I will be using an R package called the PaleoBioDB Package to visualize the speciation, the richness, the count, and the location of Haplorhini:

```
## Loading required package: raster
```

```
## Loading required package: sp
```

```
## Loading required package: maps
```

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

```
##
## Attaching package: 'ape'
```

```
## The following objects are masked from 'package:raster':
##
##     rotate, zoom
```

I used pbdb_occurrences to download all the fossil data that belongs to Haplorhini:

```
haplorhini <-pbdb_occurrences (limit="all", vocab= "pbdb", base_name="haplorhini", show=
head(haplorhini)
```

```
pbdb_map(haplorhini)
#pbdb_map returns a map with the species occurrences.
```

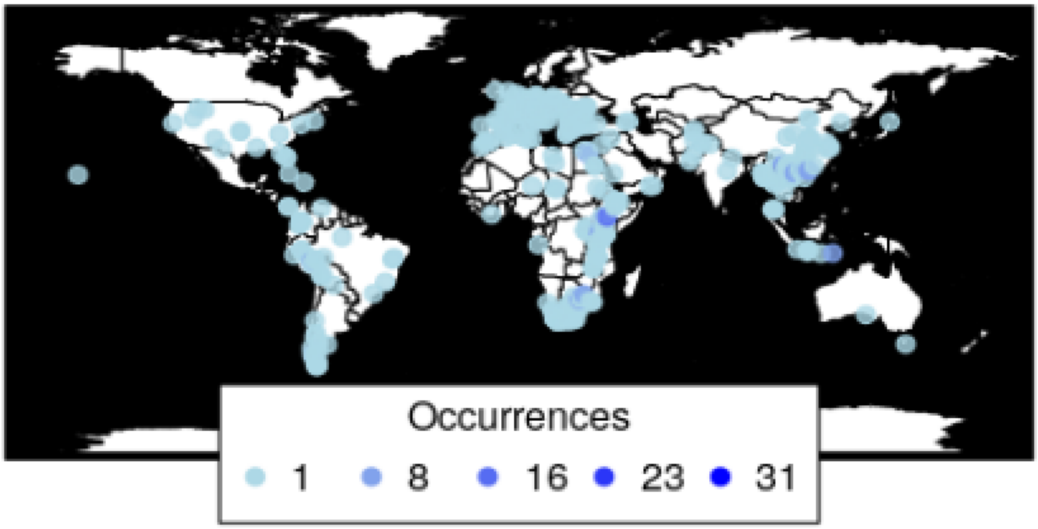


Figure 2: Haplorhini Occurrence Map

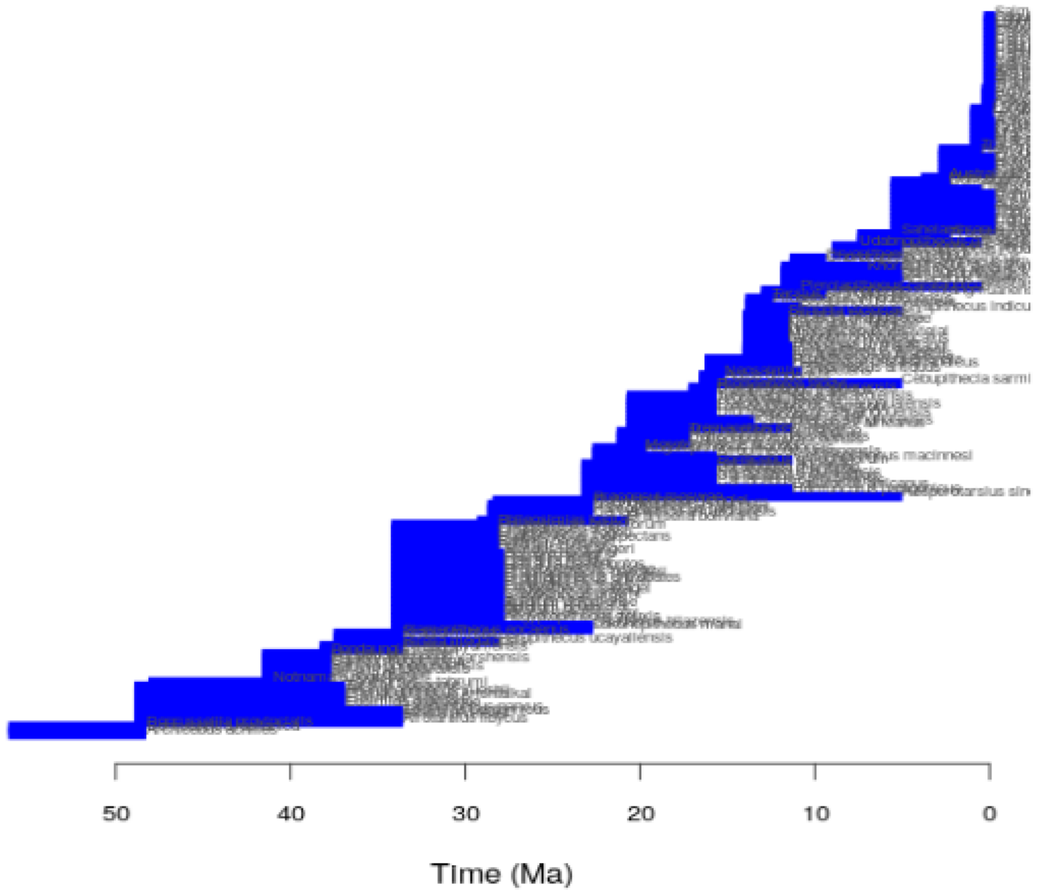


Figure 3: Haplorhini Species

```r
pbdb_temp_range (haplorhini, rank="species")
#This returns a dataframe and a plot with the time span of species,etc..
```

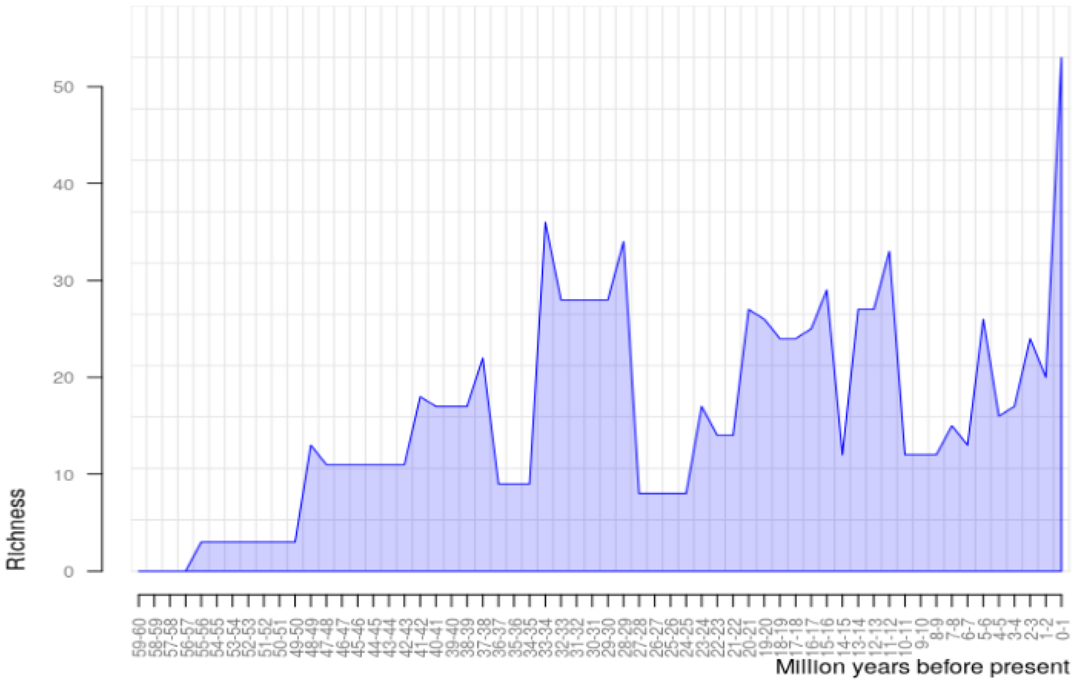


Figure 4: Haplorhini Richness

pbdb_richness returns a dataframe and a plot with the number of species (or genera, families, etc.) across time. I will be setting the temporal extent and the temporal resolution appropriately to show all species:

```r
pbdb_richness (haplorhini, rank="species", temporal_extent=c(0,60), res=1)
```

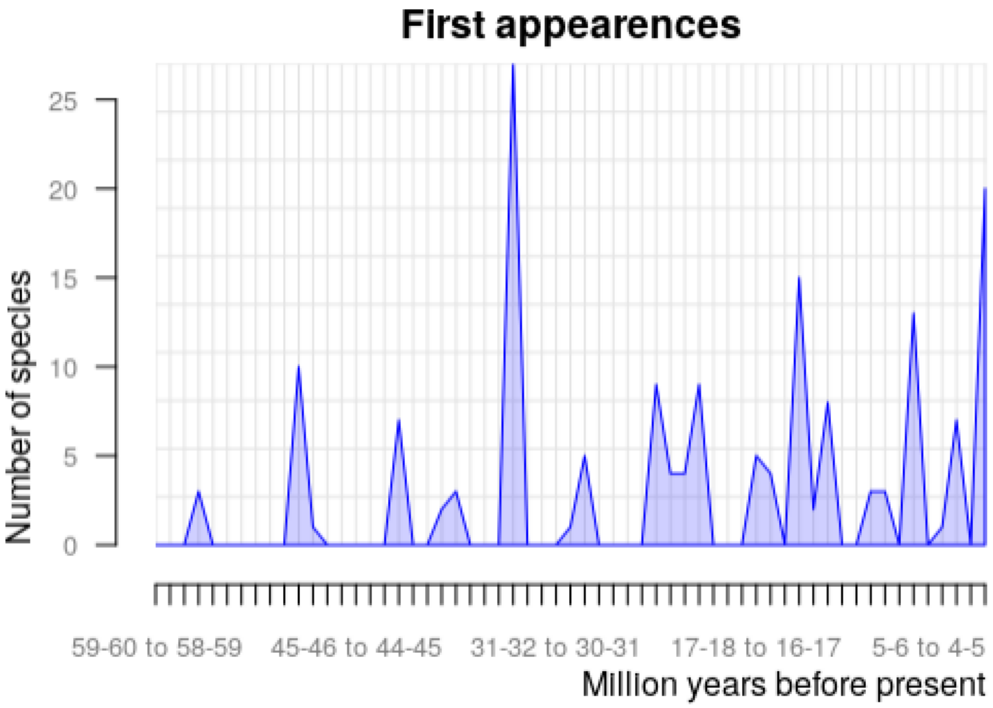


Figure 5: Haplorhini First Appearance

pbdb_orig_ext returns a dataframe and a plot with the number of new appearances and last appearances of species, genera, families, etc.

```
# evolutionary rates= orig_ext=1
pbdb_orig_ext (haplorhini, rank="species", orig_ext=1, temporal_extent=c(0,60), res=1)
pbdb_orig_ext(haplorhini, rank="species", orig_ext=2, temporal_extent=c(0,60), res=1)
```

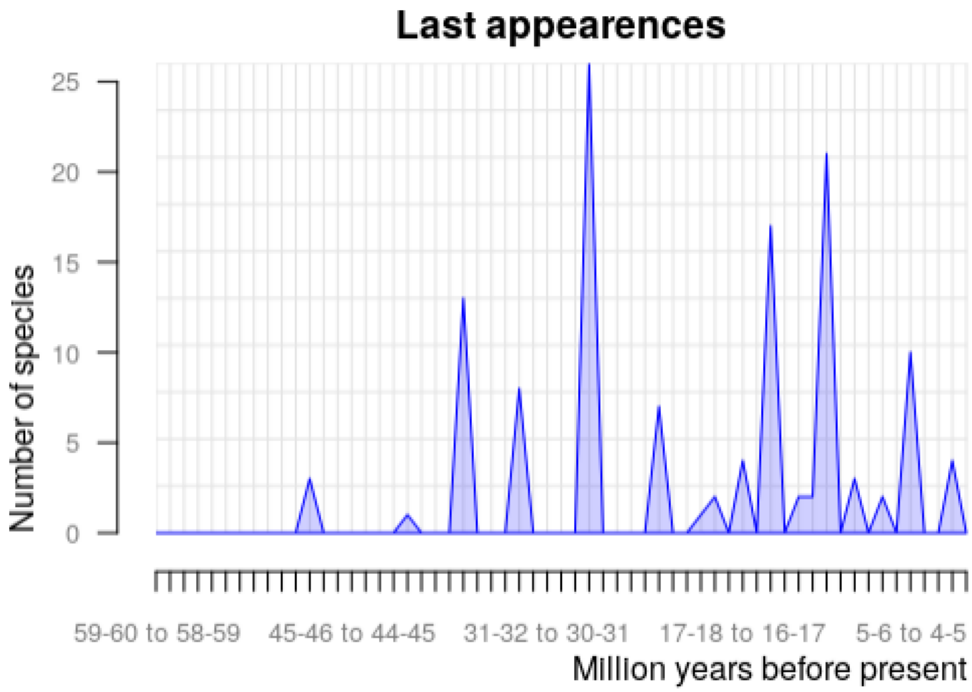


Figure 6: Haplorhini Last Appearance

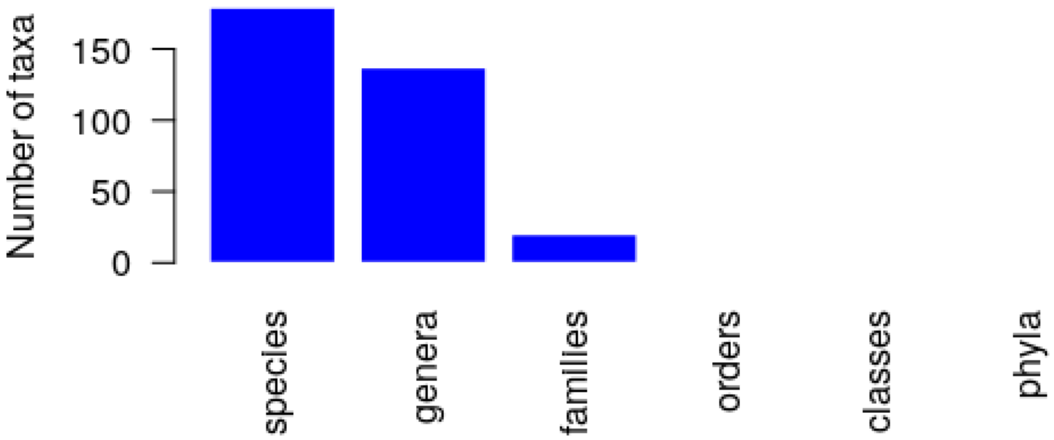
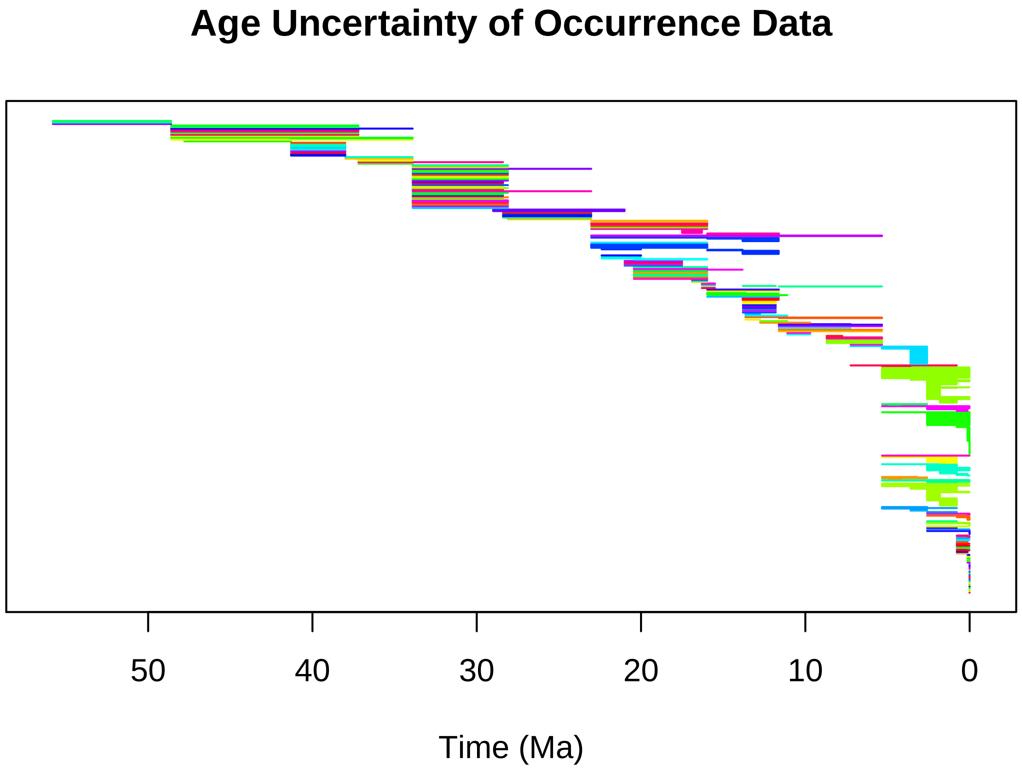


Figure 7: Haplorhini Number of Species

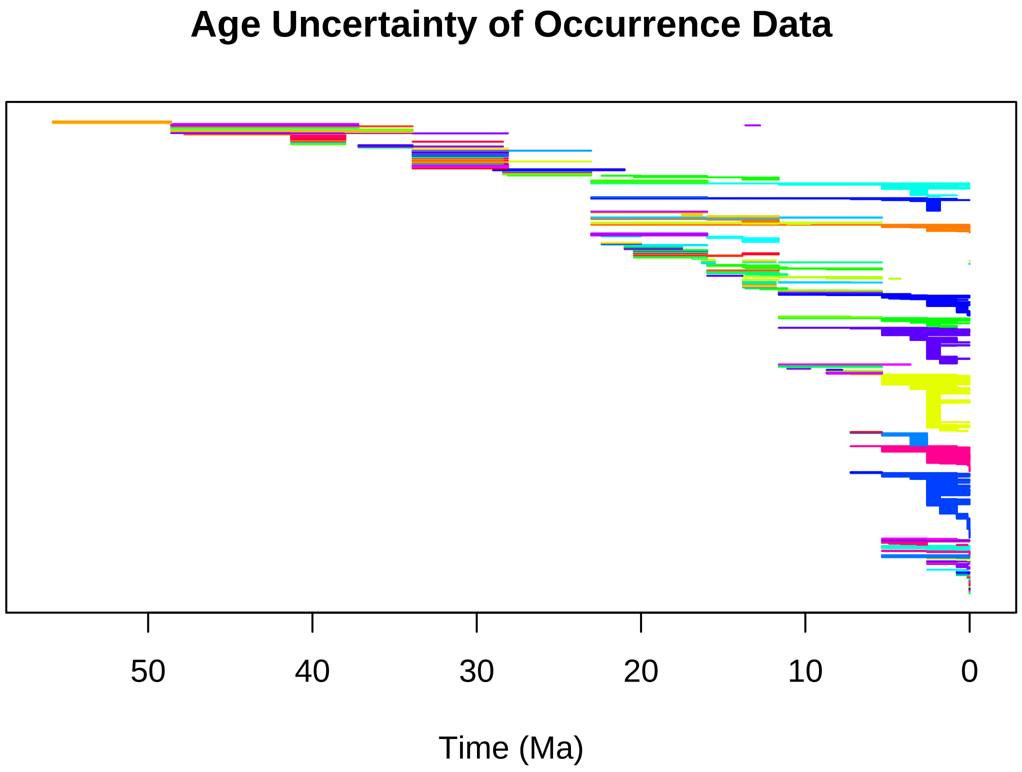
```
#this returns a plot and a dataframe with the number of species etc..
pbdb_subtaxa (haplorhini, do.plot=TRUE)
```

```
## Warning in data(haplorhini): data set 'haplorhini' not found

##     occurrence_no record_type collection_no                taxon_name
## 1:1        138735  occurrence         11798 Lufengpithecus lufengensis
## 1:2        138736  occurrence         11798    Laccopithecus robustus
## 1:3        138831  occurrence         11800      Sivapithecus indicus
## 1:4        150075  occurrence         13293        Hylobatidae indet.
## 1:5        150076  occurrence         13293            Pongo pygmaeus
## 1:6        150077  occurrence         13293                Macaca sp.
##     taxon_rank taxon_no           matched_name matched_rank matched_no
## 1:1    species    89270         Lufengpithecus        genus      89270
## 1:2    species    58366 Laccopithecus robustus      species      58366
## 1:3    species   133688   Sivapithecus indicus      species     133688
## 1:4     family    40889            Hylobatidae       family      40889
## 1:5    species   183261         Pongo pygmaeus      species     183261
## 1:6      genus    40869                 Macaca        genus      40869
##         early_interval early_age late_age reference_no      lng      lat
## 1:1           Turolian     8.700   5.3330    4196;4196 102.0667 25.01667
## 1:2           Turolian     8.700   5.3330  11394;11394 102.0667 25.01667
## 1:3       Late Miocene    11.608   5.3330    4199;4199  72.0000 33.00000
## 1:4 Middle Pleistocene     0.781   0.0117  29343;29343 111.5667 22.76667
## 1:5 Middle Pleistocene     0.781   0.0117    4412;4412 111.5667 22.76667
## 1:6 Middle Pleistocene     0.781   0.0117    4412;4412 111.5667 22.76667
##              genus genus_no          family family_no    order order_no
## 1:1 Lufengpithecus    89270       Hominidae     40899 Primates    40700
## 1:2  Laccopithecus    58365  Pliopithecidae     40884 Primates    40700
## 1:3   Sivapithecus    40898       Hominidae     40899 Primates    40700
## 1:4           <NA>       NA     Hylobatidae     40889 Primates    40700
## 1:5          Pongo    40897       Hominidae     40899 Primates    40700
## 1:6         Macaca    40869 Cercopithecidae     40860 Primates    40700
##        class class_no   phylum phylum_no     genus_name species_name
## 1:1 Mammalia    36651 Chordata     33815 Lufengpithecus  lufengensis
## 1:2 Mammalia    36651 Chordata     33815  Laccopithecus     robustus
## 1:3 Mammalia    36651 Chordata     33815   Sivapithecus      indicus
## 1:4 Mammalia    36651 Chordata     33815    Hylobatidae       indet.
## 1:5 Mammalia    36651 Chordata     33815          Pongo     pygmaeus
## 1:6 Mammalia    36651 Chordata     33815         Macaca          sp.
##     reid_no    late_interval species_reso subgenus_name genus_reso
## 1:1      NA             <NA>         <NA>          <NA>       <NA>
## 1:2      NA             <NA>         <NA>          <NA>       <NA>
## 1:3      NA             <NA>         <NA>          <NA>       <NA>
## 1:4   22506 Late Pleistocene         <NA>          <NA>       <NA>
## 1:5      NA Late Pleistocene         <NA>          <NA>       <NA>
## 1:6      NA Late Pleistocene         <NA>          <NA>       <NA>
##     subgenus_reso
## 1:1          <NA>
## 1:2          <NA>
## 1:3          <NA>
## 1:4          <NA>
## 1:5          <NA>
## 1:6          <NA>
```

```
occSpecies<-taxonSortPBDBocc(haplorhini, rank="species")
#plot it!
plotOccData(occSpecies)
```



```
#formal grapt species in the PBDB
#genera is messier...
#get formal genera
occGenus<-taxonSortPBDBocc(haplorhini, rank="genus")

#plot it!
plotOccData(occGenus)
```



Formatting and Visulizing Data Using ggplot

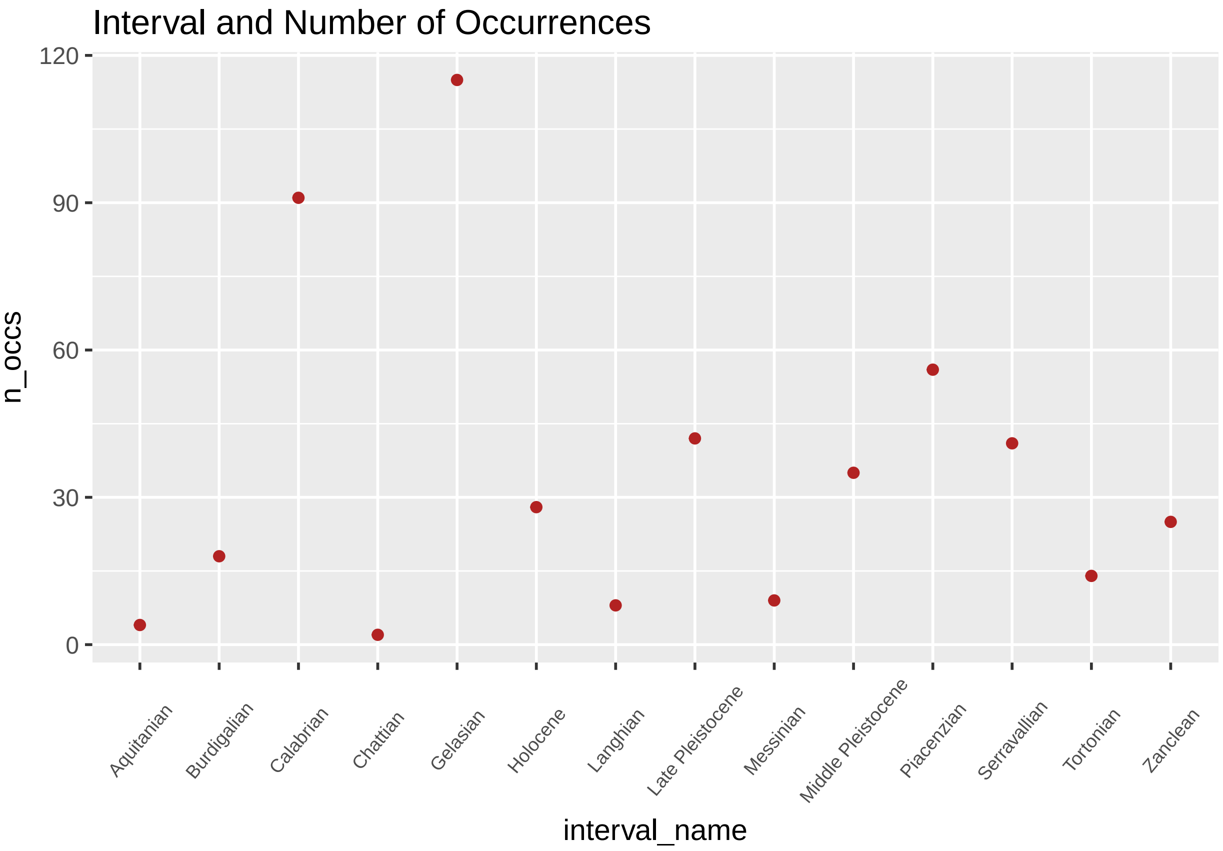
After downloading the occurrence data and diversity data from PaleoBioDB, I formatted my files from the shell using the following commands:

```
tail -n+18 haplolist.csv > haploformatted.csv
tail -n+21 haplodiversity.csv > haplodiversity21.csv
```

```
library(ggplot2)
haplodiv<-read.csv("~/Desktop/eeb-177/eeb-174-final-project/haplodiversity21.csv", as.is

head(haplodiv)
```

```
##   interval_no      interval_name max_ma min_ma X_Ft X_bL X_FL X_bt
## 1        3002           Holocene 0.0117 0.0000    0    1    0    0
## 2         922   Late Pleistocene 0.1260 0.0117    0    1    2    1
## 3         923 Middle Pleistocene 0.7810 0.1260    1    1    2    1
## 4         740         Calabrian 1.8060 0.7810    1    1    0    1
## 5         741           Gelasian 2.5880 1.8060    0    0    0    2
## 6          96         Piacenzian 3.6000 2.5880    1    1    0    1
##   sampled_in_bin n_occs
## 1              1     28
## 2              4     42
## 3              5     35
## 4                3     91
## 5                2    115
## 6                3     56
```

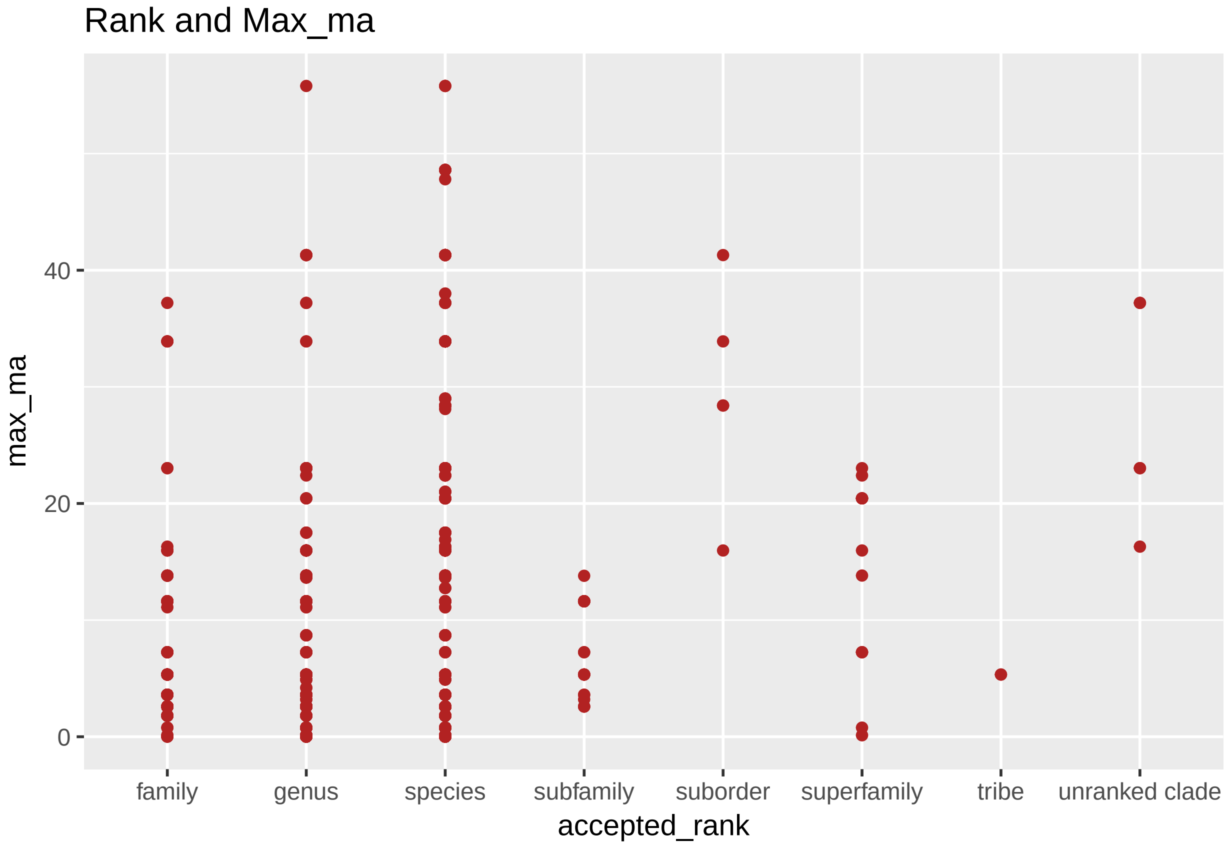
```
g<-ggplot(haplodiv, aes(interval_name, n_occs))+geom_point(color="firebrick")
g<-g+ggtitle('Interval and Number of Occurrences')
g + theme(axis.text.x=element_text(angle=50, size=7, vjust=0.5))
```



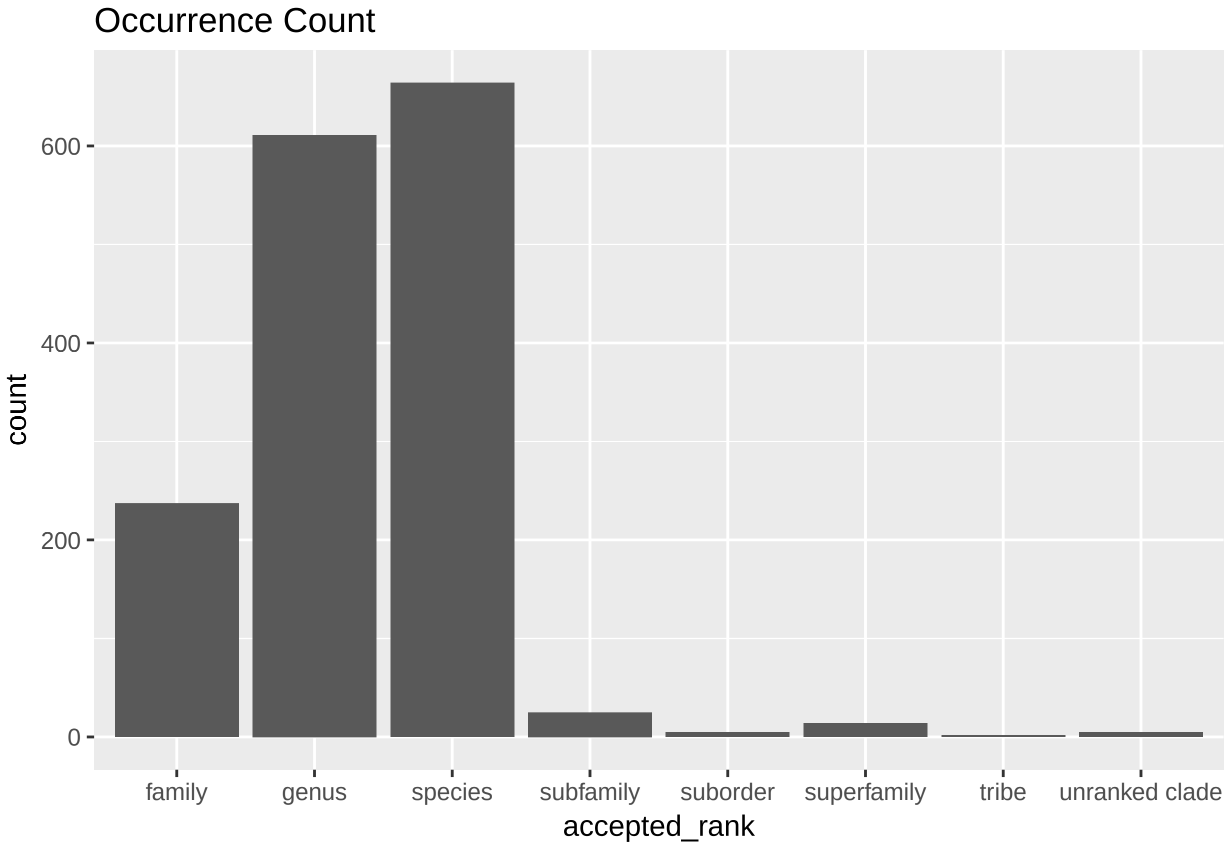
```
library(ggplot2)
haplogg<-read.csv("~/Desktop/eeb-177/eeb-174-final-project/haploformatted17.csv", as.is=
head(haplogg)
```

```
##   occurrence_no record_type reid_no flags collection_no
## 1        138735         occ      NA    NA         11798
## 2        138736         occ      NA    NA         11798
## 3        138831         occ      NA    NA         11800
## 4        150075         occ   22506    NA         13293
## 5        150076         occ      NA    NA         13293
## 6        150077         occ      NA    NA         13293
##               identified_name identified_rank identified_no
## 1 Lufengpithecus lufengensis         species         89270
## 2     Laccopithecus robustus         species         58366
## 3       Sivapithecus indicus         species        133688
## 4         Hylobatidae indet.          family        40889
## 5             Pongo pygmaeus         species       183261
## 6                 Macaca sp.           genus        40869
##           difference          accepted_name accepted_rank accepted_no
## 1 species not entered         Lufengpithecus         genus       89270
## 2                       Laccopithecus robustus       species       58366
## 3                         Sivapithecus indicus       species      133676
## 4                                  Hylobatidae        family       40889
## 5                               Pongo pygmaeus       species      232338
## 6                                       Macaca         genus       40869
##        early_interval    late_interval max_ma min_ma reference_no
## 1            Turolian                   8.700 5.3330         4196
## 2            Turolian                   8.700 5.3330        11394
## 3         Late Miocene                 11.608 5.3330         4199
## 4 Middle Pleistocene Late Pleistocene  0.781 0.0117        29343
## 5 Middle Pleistocene Late Pleistocene  0.781 0.0117         4412
## 6 Middle Pleistocene Late Pleistocene  0.781 0.0117         4412
```

```r
g<-ggplot(haplogg, aes(accepted_rank, max_ma))+geom_point(color="firebrick")
g<-g+ggtitle('Rank and Max_ma')
g
```



```
plot1 <- ggplot(haplogg, aes(x=accepted_rank)) + geom_bar() + labs(title="Occurrence Cou
print(plot1)
```



Visulizing Haplorhini Occurrence Using ggplot2 and forcats

```
##          genus                      species minage maxage
## 1   Archicebus          Archicebus achilles   55.8   48.6
## 2 Donrussellia     Donrussellia lusitanica   55.8   48.6
## 3 Donrussellia Donrussellia provincialis   55.8   48.6
## 4  Afrotarsius         Afrotarsius libycus   48.6   33.9
## 5      Biretia           Biretia piveteaui   48.6   33.9
## 6     Eosimias        Eosimias centennicus   48.6   33.9
```

```
haplo_occ <- ggplot(haplo, aes( species, ymin = maxage, ymax=minage, colour = genus))
haplo_occ <- haplo_occ + geom_linerange()
haplo_occ
haplo_occ <- haplo_occ + theme(legend.position="none")
haplo_occ
haplo_occ <- haplo_occ + coord_flip()
haplo_occ
haplo_occ <- haplo_occ + theme(legend.position="none")
```

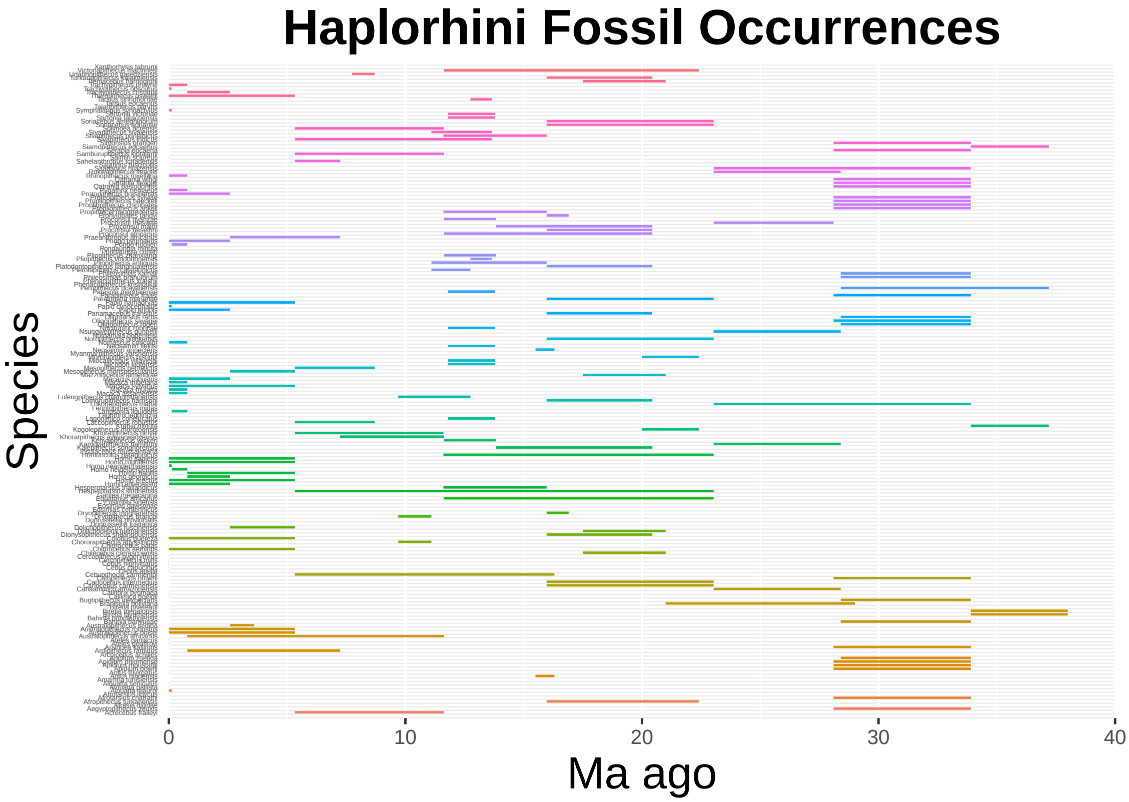


Figure 8: Haplorhini Occurrences

```
haplo_occ
haplo_occ <- haplo_occ +  theme(axis.text.y = element_text(size=3))
haplo_occ
haplo_occ <- haplo_occ + theme(axis.ticks.y=element_blank())
haplo_occ
haplo_occ <- haplo_occ + scale_y_continuous(limits=c(0, 40), expand = c(0, 0), breaks=c
haplo_occ
haplo_occ <- haplo_occ + labs(title = "Haplorhini Fossil Occurrences", x = "Species", y
haplo_occ
```

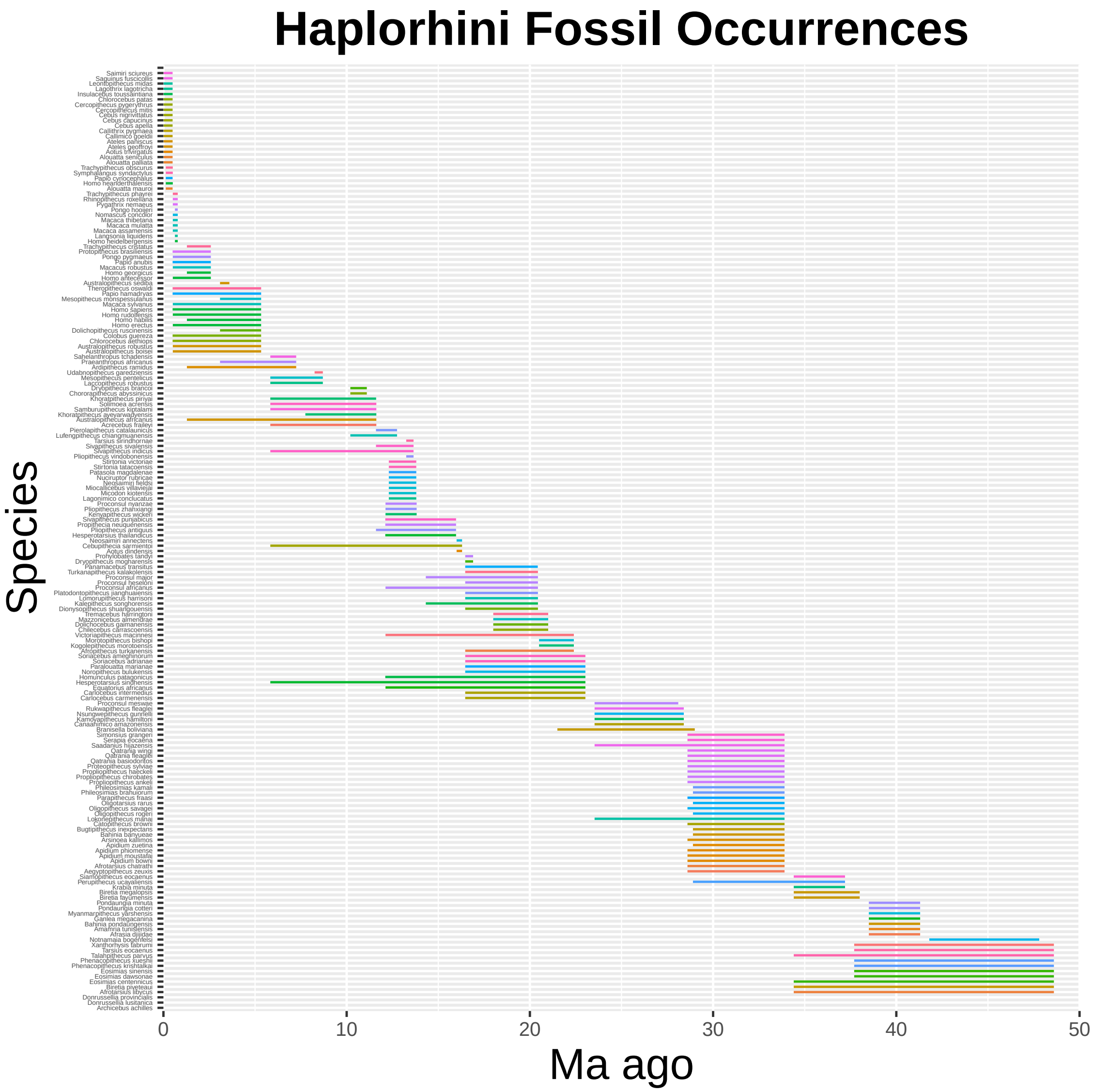


Figure 9: Haplorhini Ordered Occurrences

I am going to use an R package called forcats to reorder the species by their minimum age. This will force ggplot to arrange the ranges so that the youngest are on top and the oldest are on the bottom.

```
#install.packages("forcats")
library(ggplot2)
library(forcats)
haplo_occ <- ggplot(haplo, aes( x = fct_reorder(species, minage, .desc = T), maxage, col

haplo_occ2 <- haplo_occ + geom_linerange(aes(ymin = minage, ymax = maxage + 0.5)) + the
haplo_occ2
```

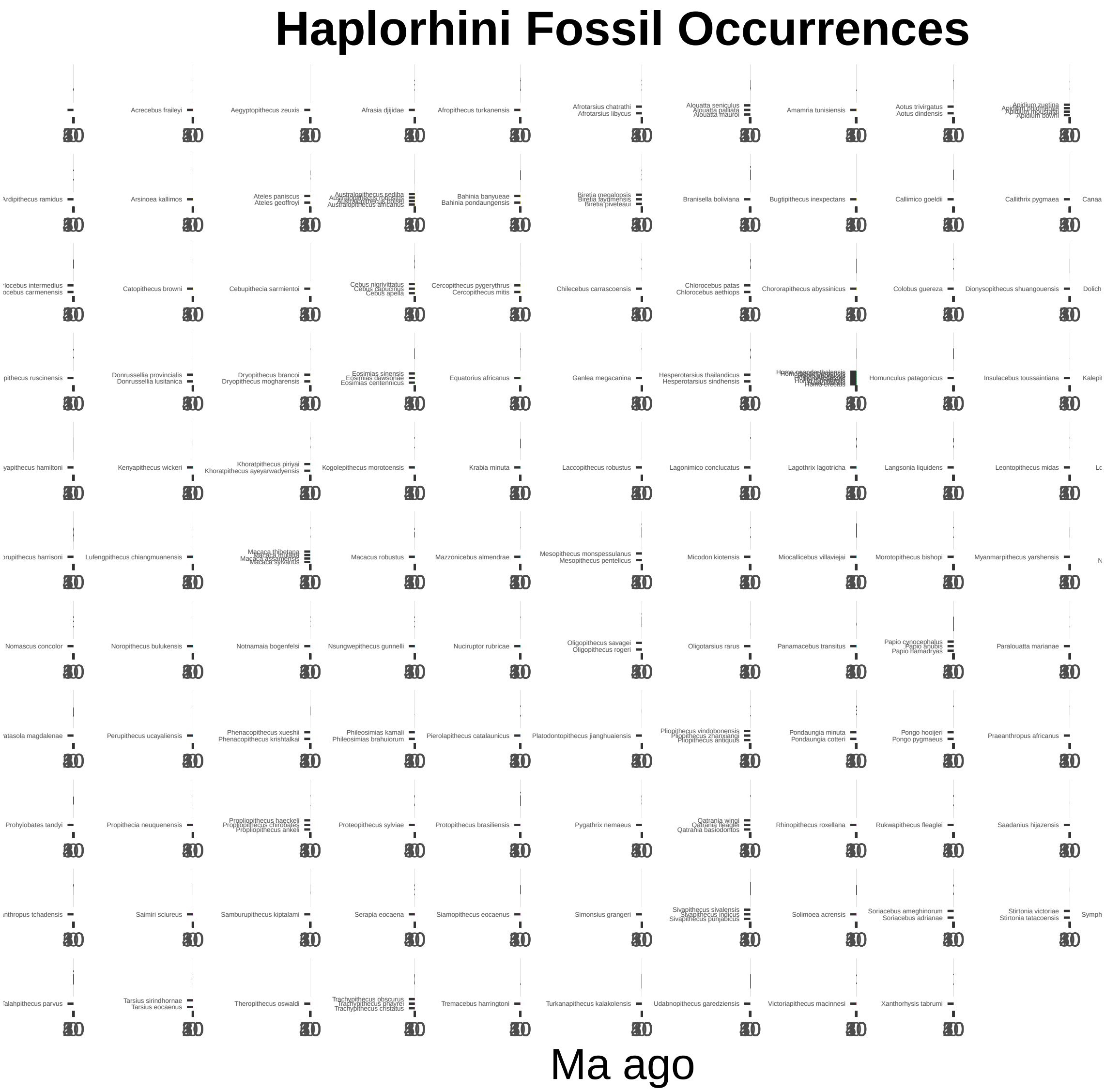


Figure 10: Haplorhini Faceted Ranges

```
# faceted
haplo_occ2 + geom_linerange(aes(ymin = minage, ymax = maxage + 0.5)) + theme(legend.pos
library(tidyr)
```

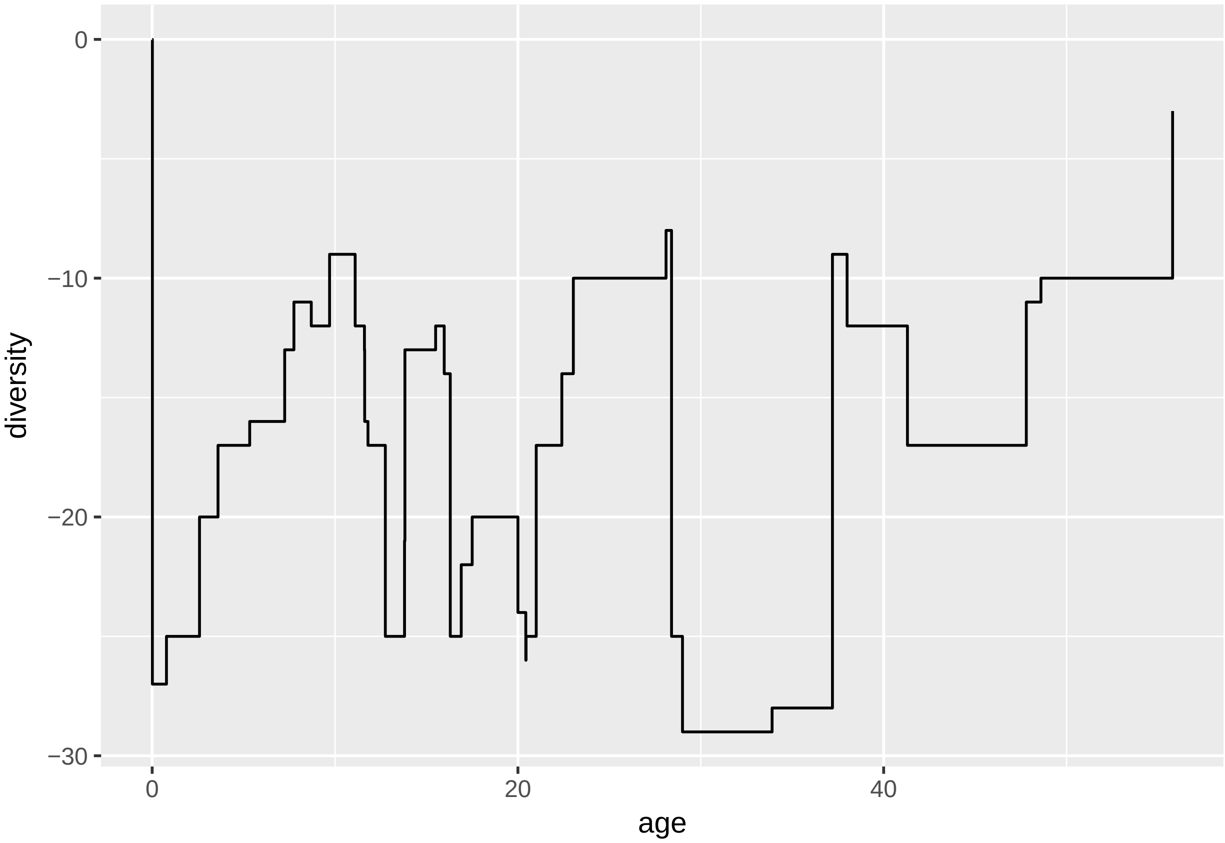
```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:raster':
##
##     extract
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
diversity <- haplo %>% gather(key = type, value = age, minage, maxage) %>% mutate(count

ggplot(diversity, aes(x = age, y = diversity)) + geom_step()
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Reproducing PaleobioBD Graphs using matplot and ggplot

I will now attempt to reproduce the richness graph, but first I need to extract some data:

```
def haplofunc(filename):
    # this function opens the data file and extracts the name and maxage as a dictiona
    haploDict = {}
    fobj = open(filename,"r", encoding = "ISO-8859-15") # this opens the file
    reader = fobj.readlines()  # read the file
    for line in reader:
        line = line.split(",")
        name = line[5][1:-1]
        rank = line[6][1:-1]
        min_age = line[15][1:-1]
        max_age= line[14][1:-1]
        haploDict[name] = max_age
    return haploDict
haplofunc('haploformatted.csv')
```

```
import csv
with open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haploformatted.csv
    reader = csv.reader(infh)
    for row in reader:
        if row[6] == "species":
            species = row[5]
            genus = row[5].split(" ")[0]
            print(row[5] + "," + row[15] + "," + row[14])
```

```
in_file = open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haplominmax3.
# read in the first line
# The first line is a header that we don't want in dictionaries
in_file.readline()
# set up empty dictionaries

species_richness = {}

# iterate through every line in the file
for line in in_file:
    #print(line)

    # It's a tab delimited file, so we can split the contents up into
    # the component rows
    items = line.split(",")

    # Save some important pieces of information as variables
    species_name = items[1]
    max_age = float(items[3])

    # Use the variables to populate our three empty lists
    species_richness[species_name] = max_age
```

```
import matplotlib.pyplot as plt

# Most plotting happens on lists, not dictionaries-
# so, let's convert our data into a list
max_age = list(species_richness.values())

# We're ready to make a histogram!
plt.hist(max_age)
plt.xlabel('Time Interval (ma)')
# a y-axis label
plt.ylabel('Species Richness (frequency)')
# A main title for the graph
plt.title("Haplorhini Spcies Richness")
plt.show()
```

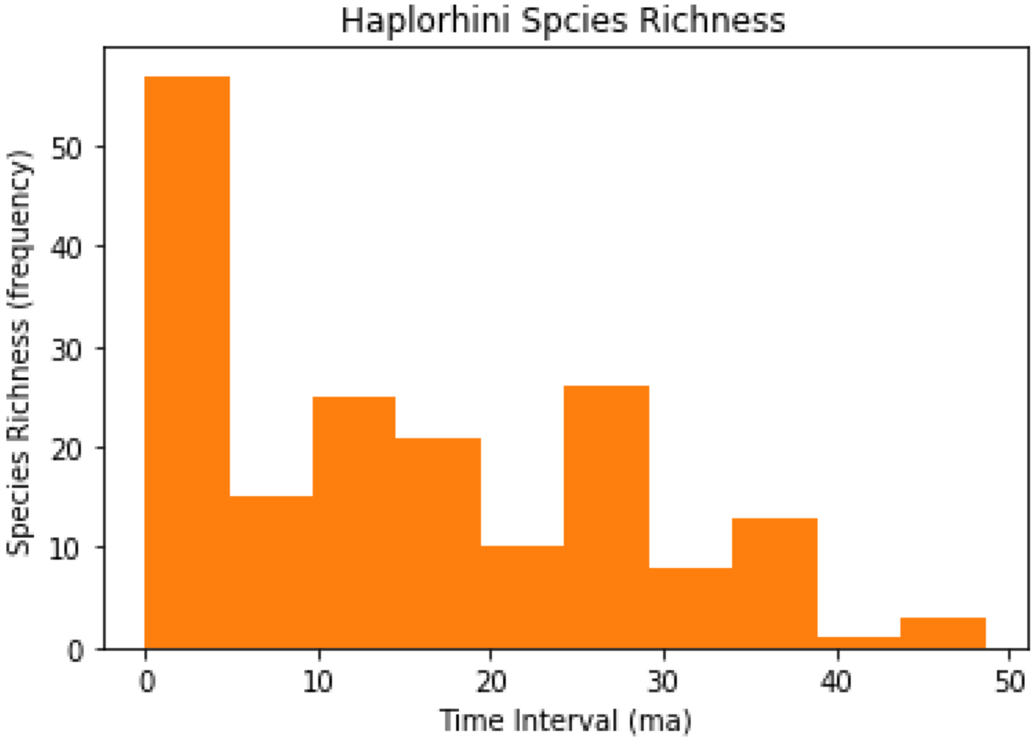
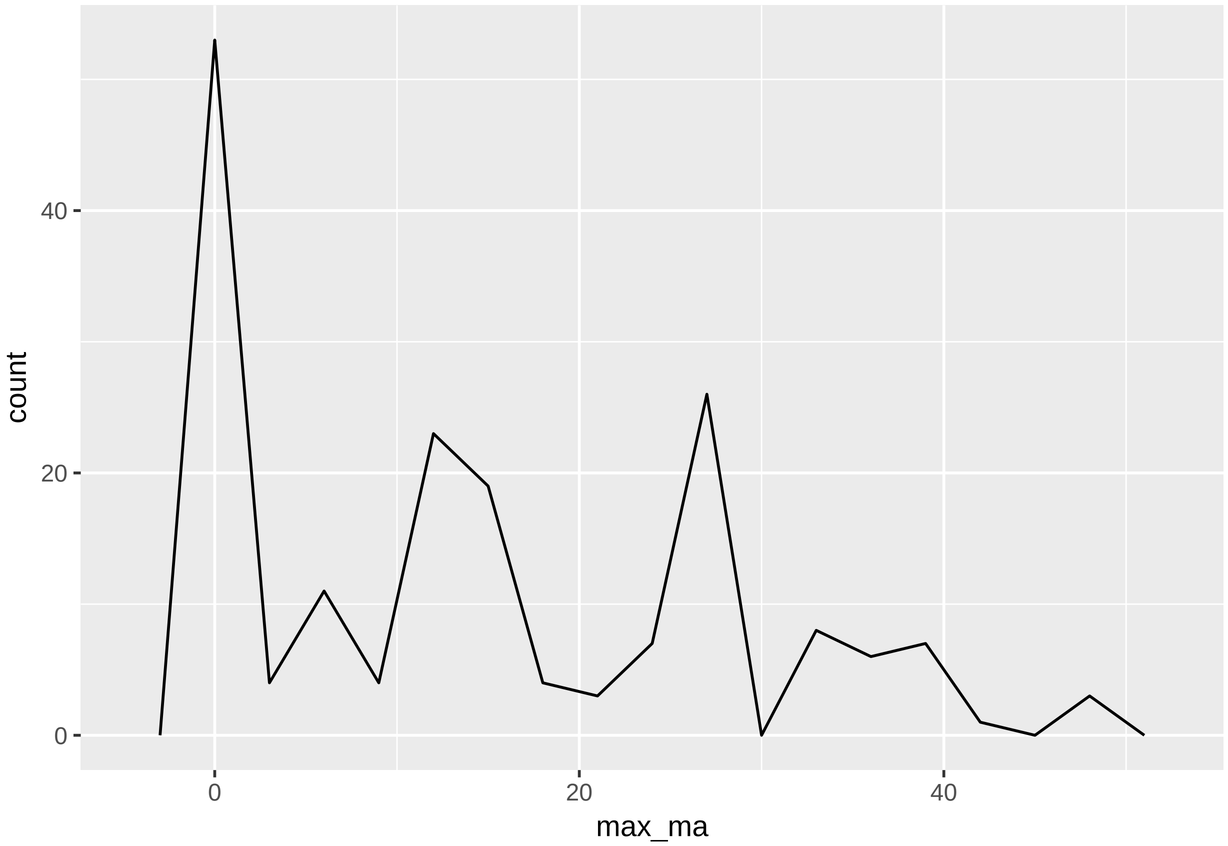


Figure 11: Haplorhini Richness

```r
# I could also produce the same graph using ggplot2
library(ggplot2)
haplo_rich<-read.csv("~/Desktop/eeb-177/eeb-174-final-project/haplominmax3.csv", as.is=T
head(haplo_rich)
```

```
##   xx              accepted_name min_ma max_ma
## 1 ##       Archicebus achilles    55.8   48.6
## 2 ##   Donrussellia lusitanica    55.8   48.6
## 3 ## Donrussellia provincialis    55.8   48.6
## 4 ##       Afrotarsius libycus    48.6   33.9
## 5 ##         Biretia piveteaui    48.6   33.9
## 6 ##      Eosimias centennicus    48.6   33.9
```

```r
plot1 <-ggplot(haplo_rich, aes(max_ma)) + geom_freqpoly(binwidth=3)
print(plot1)
```



Exploring Specimen Measurements Using Matplot and ggplot2

I will extract average width and length measurements and count them in order to produce a density graph:

```
def haplomeasure(filename):
    haploDict = {}
    fobj = open(filename,"r", encoding = "ISO-8859-15") # this opens the file
    reader = fobj.readlines()  # read the file
    for line in reader:
        line = line.split(",")
        measurement = line[5][1:-1]
        measure_number = line[6][1:-1]
        haploDict[measurement] = measure_number
    return haploDict

import csv
with open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haplomeasurements1
    reader = csv.reader(infh)
    counter= 1
    for row in reader:
        if row[5] == "width":
            print(str(counter) + "," + row[6] + ",")
            counter += 1
```

```
in_file = open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haplomeasurew

in_file.readline()
# set up empty dictionaries
sizes_width = {}
sizes_mass = {}
sizes_length = {}

# iterate through every line in the file
for line in in_file:
    #print(line)
    # It's a tab delimited file, so we can split the contents up into
    # the component rows
    items = line.split(",")
    # Save some important pieces of information as variables
    specimen_id = items[0]
    width_avg = float(items[1])
    # Use the variables to populate our three empty lists
    sizes_width[specimen_id] = width_avg
```

```
import matplotlib.pyplot as plt


# Most plotting happens on lists, not dictionaries
width_avg = list(sizes_width.values())
```

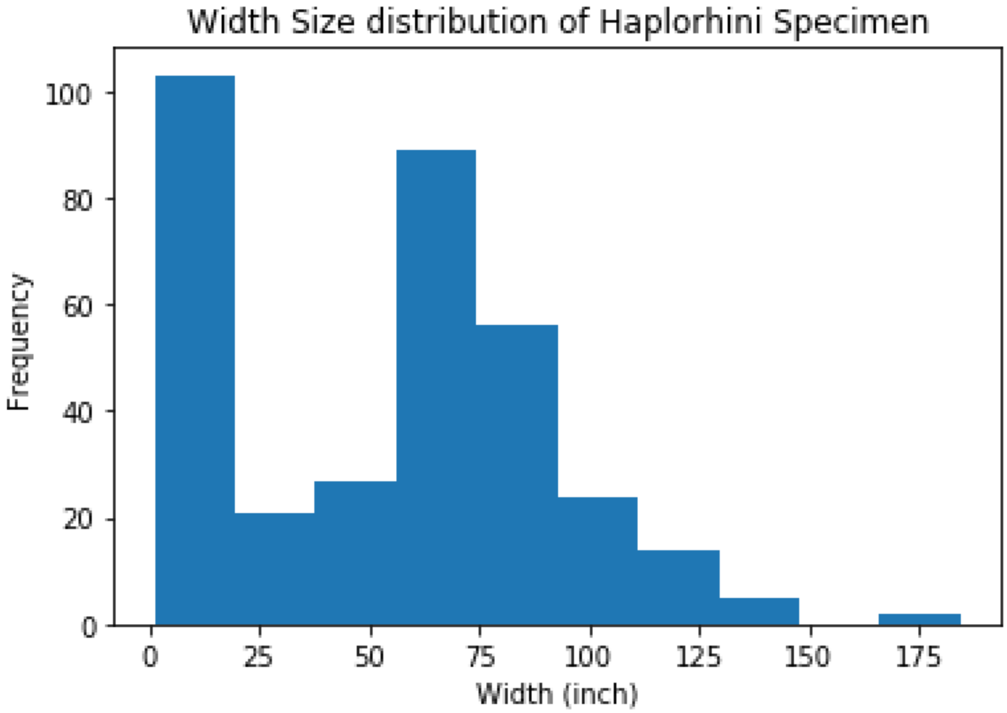


Figure 12: Average Width

```
# We're ready to make a histogram!
plt.hist(width_avg)
plt.xlabel('Width (inch)')
# a y-axis label
plt.ylabel('Frequency')
# A main title for the graph
plt.title("Width Size distribution of Haplorhini Specimen")
plt.show()
```

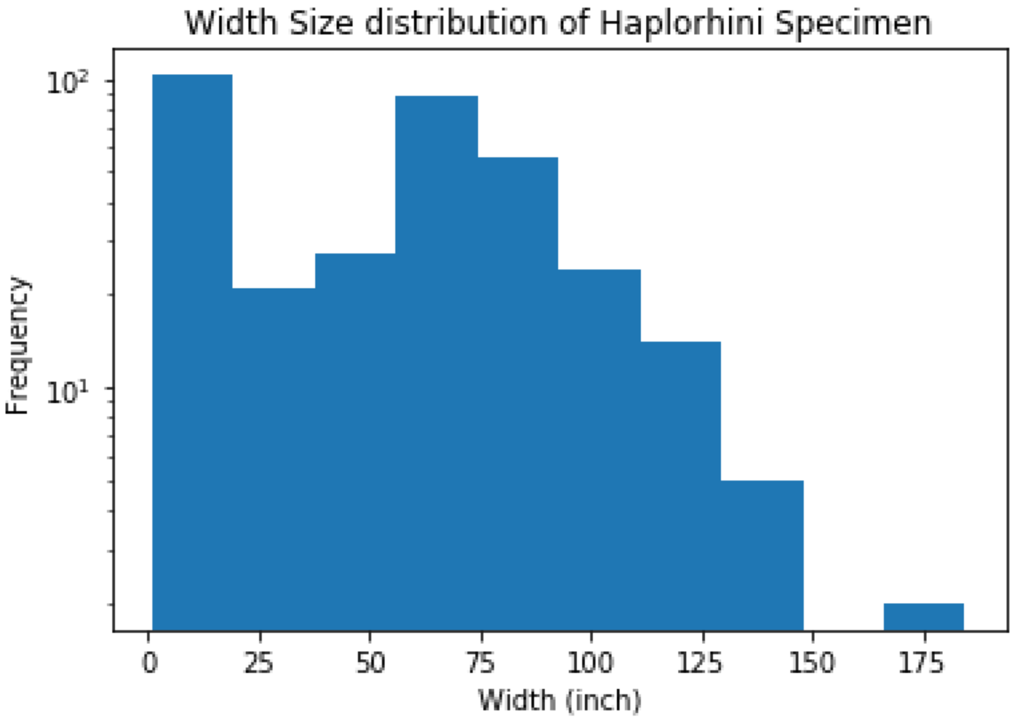


Figure 13: Log of Width Average

```
# Add a log-axis to our histogram
plt.hist(width_avg, log = True)
# an X-axis label
plt.xlabel('Width (inch)')
# a y-axis label
plt.ylabel('Frequency')
# A main title for the graph
plt.title("Width Size distribution of Haplorhini Specimen")
# Let's look!
plt.show()
```

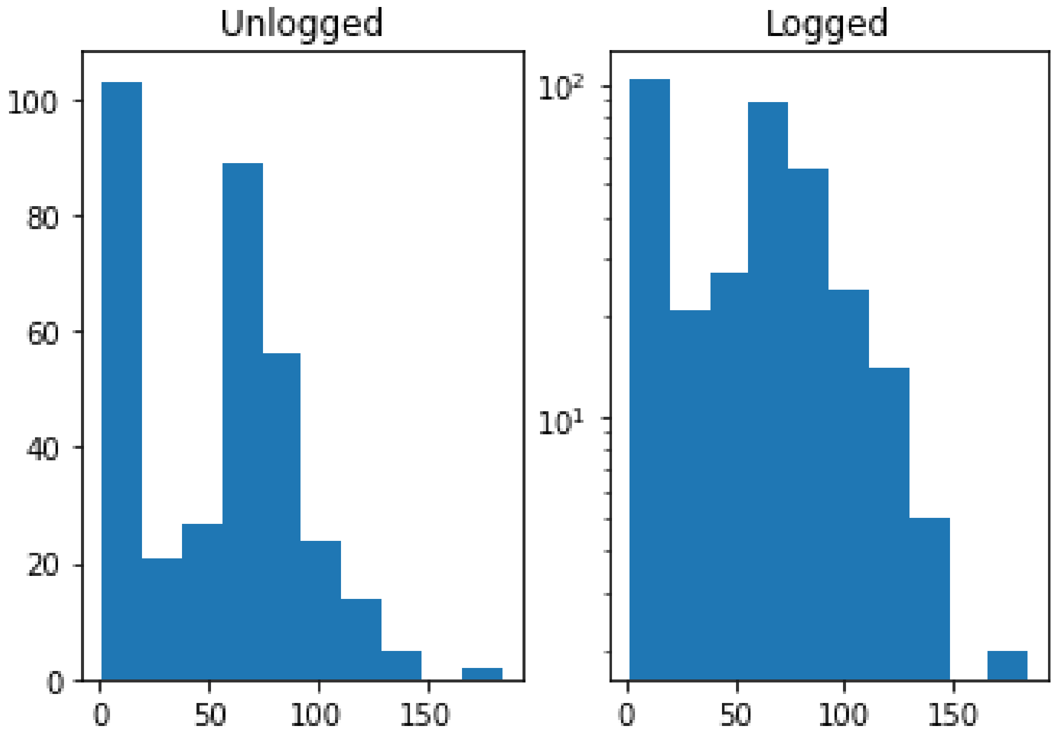


Figure 14: Logged and Unlogged Graphs Side-by-side

```
f, (ax1, ax2) = plt.subplots(1, 2,sharex = True)
ax1.hist(width_avg, log = False)
ax1.set_title('Unlogged')
ax2.hist(width_avg, log = True)
ax2.set_title('Logged')
plt.show()
```

```
#I would like to extract length measurements
import csv
output = open("haplomeasurelength.csv", "w")
with open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haplomeasurements1
    reader = csv.reader(infh)
    counter= 1
    for row in reader:
        if row[5] == "length":

            print(str(counter) + "," + row[6] + ",")
            counter += 1
```

```
in_file = open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haplomeasurel
# read in the first line
# The first line is a header that we don't want in dictionaries
in_file.readline()
# set up empty dictionaries
sizes_length = {}
# iterate through every line in the file
for line in in_file:
    #print(line)

    # It's a tab delimited file, so we can split the contents up into
    # the component rows
    items = line.split(",")

    # Save some important pieces of information as variables
    specimen_id = items[0]
    length_avg = float(items[1])

    # Use the variables to populate our three empty lists
    sizes_length[specimen_id] = length_avg
```

```
import matplotlib.pyplot as plt

# Most plotting happens on lists, not dictionaries
length_avg = list(sizes_length.values())

# We're ready to make a histogram!
plt.hist(length_avg)
plt.xlabel('Length (inch)')
# a y-axis label
plt.ylabel('Frequency')
# A main title for the graph
plt.title("Length distribution of Haplorhini Specimen")
```

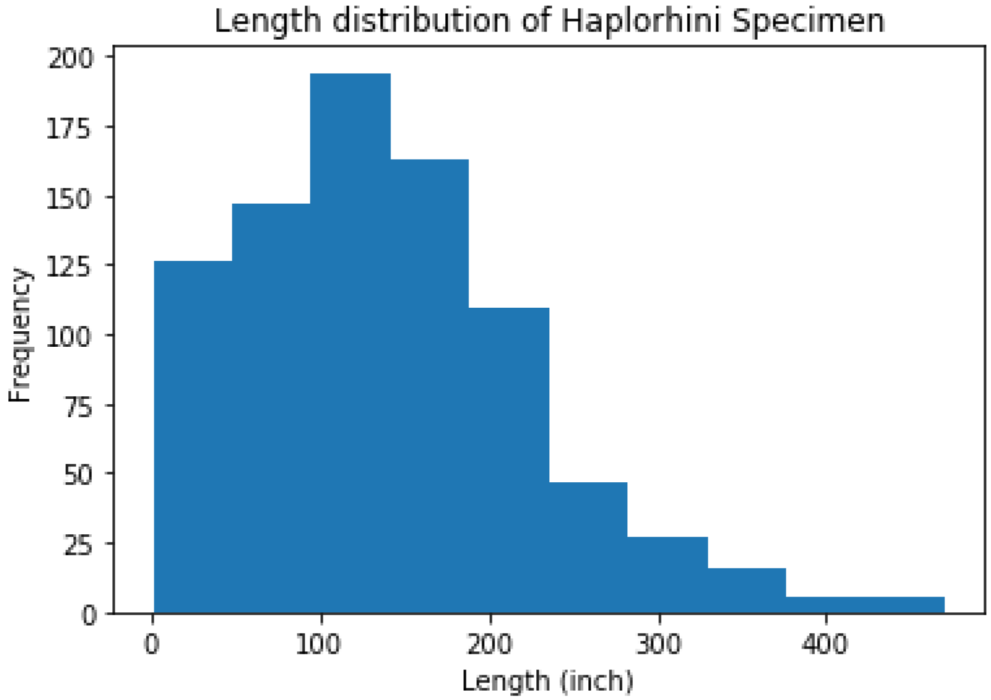


Figure 15: Average Length Distribution

```
plt.show()
```

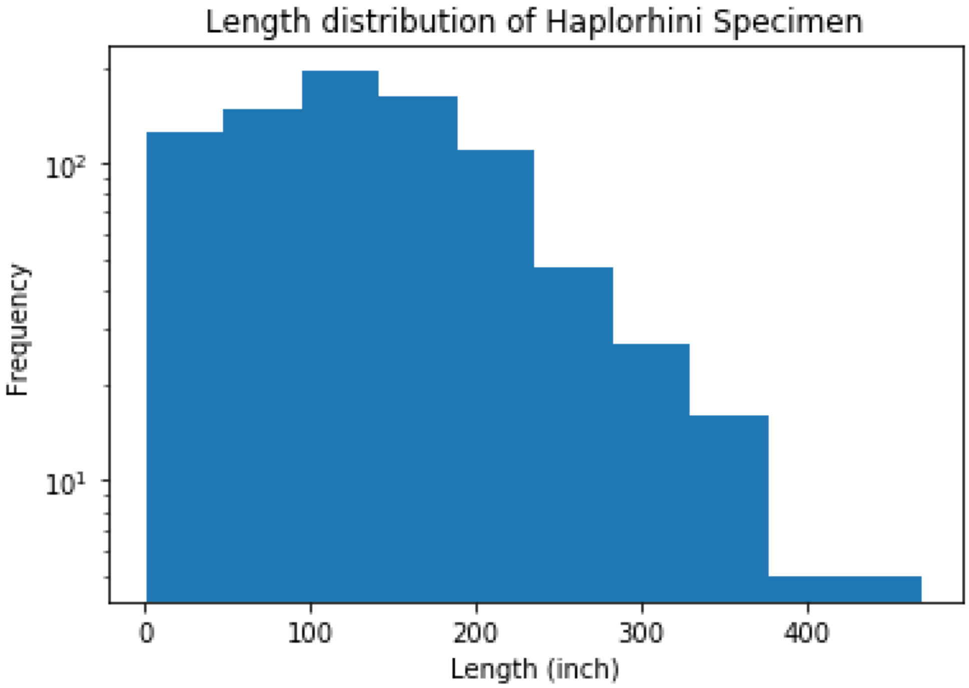


Figure 16: Length Average Logged

```
# Add a log-axis to our histogram
plt.hist(length_avg, log = True)
# an X-axis label
plt.xlabel('Length (inch)')
# a y-axis label
plt.ylabel('Frequency')
# A main title for the graph
plt.title("Length distribution of Haplorhini Specimen")
plt.show()
```

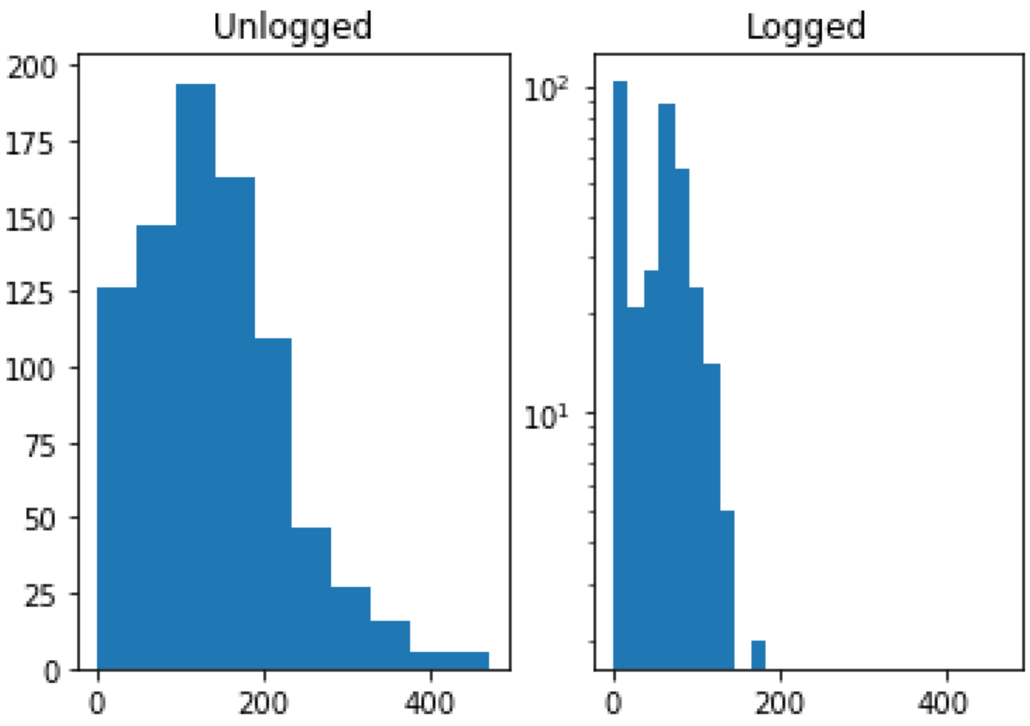
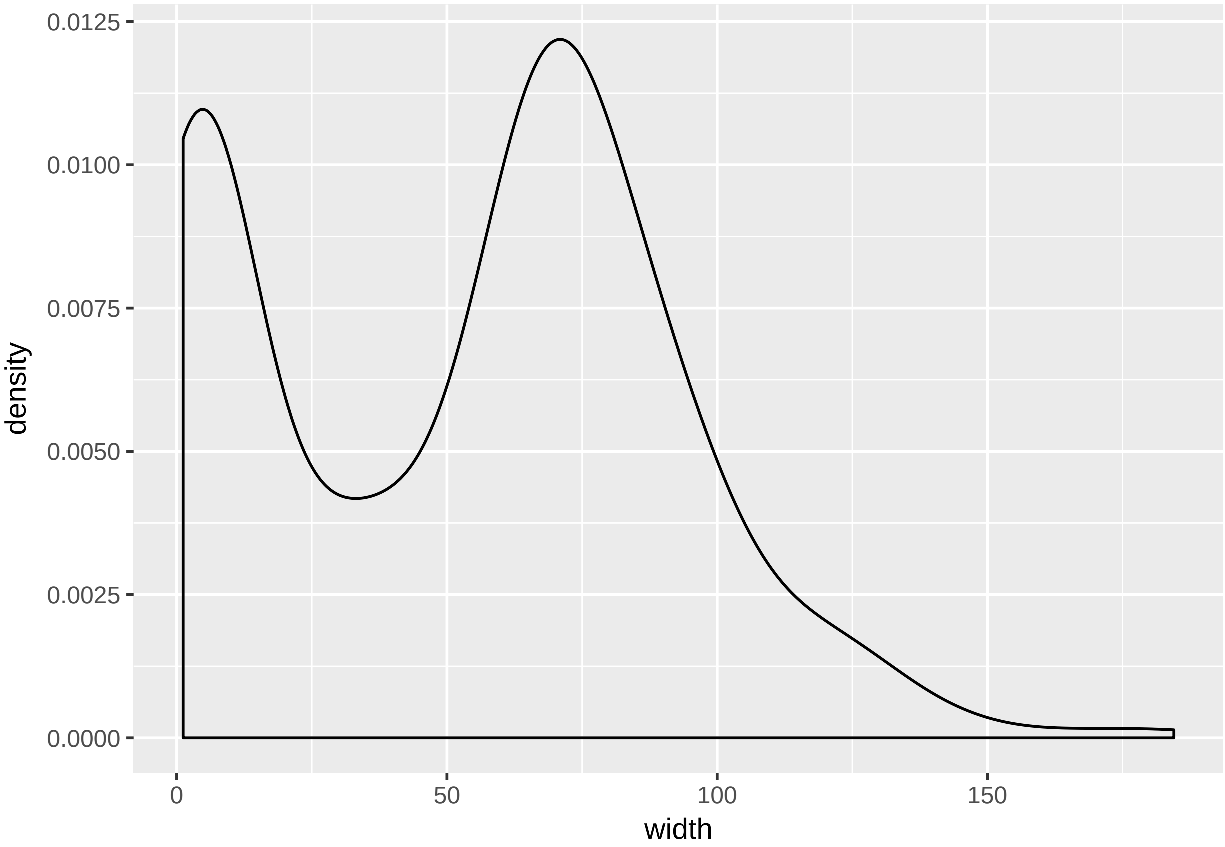


Figure 17: Logged and Unlogged Graphs Side-by-side

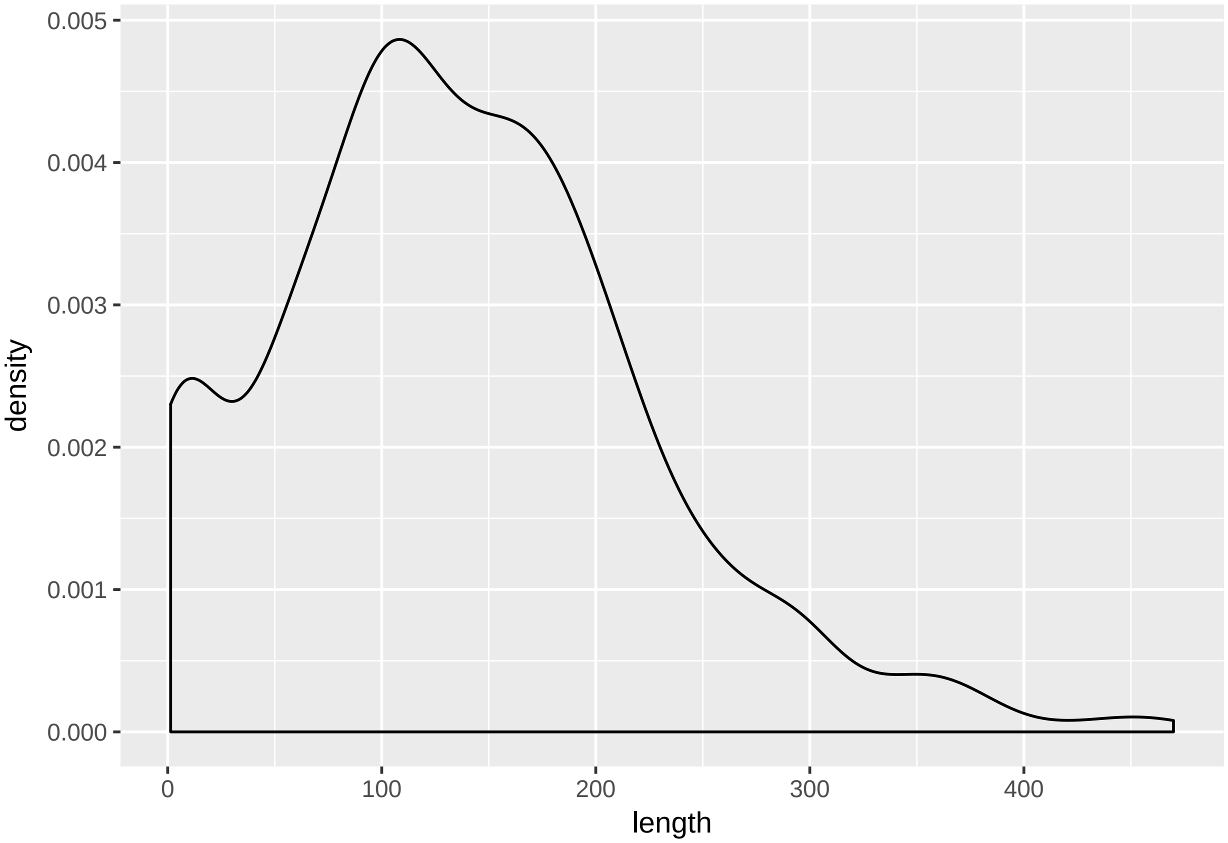
```
f, (ax1, ax2) = plt.subplots(1, 2,sharex = True)
ax1.hist(length_avg, log = False)
ax1.set_title('Unlogged')
ax2.hist(width_avg, log = True)
ax2.set_title('Logged')
plt.show()
```

Now using ggplot, I will produce the same graphs however I will create a gaussian graph in order to better visualize the modes of the graph:

```
library(ggplot2)
haplo_width<-read.csv("~/Desktop/eeb-177/eeb-174-final-project/haplomeasurewidth.csv", a

#head(haplo_width)
plot11 <-ggplot(haplo_width, aes(width)) + geom_density(kernel= "gaussian")
  #geom_freqpoly(binwidth=20)
print(plot11)
```

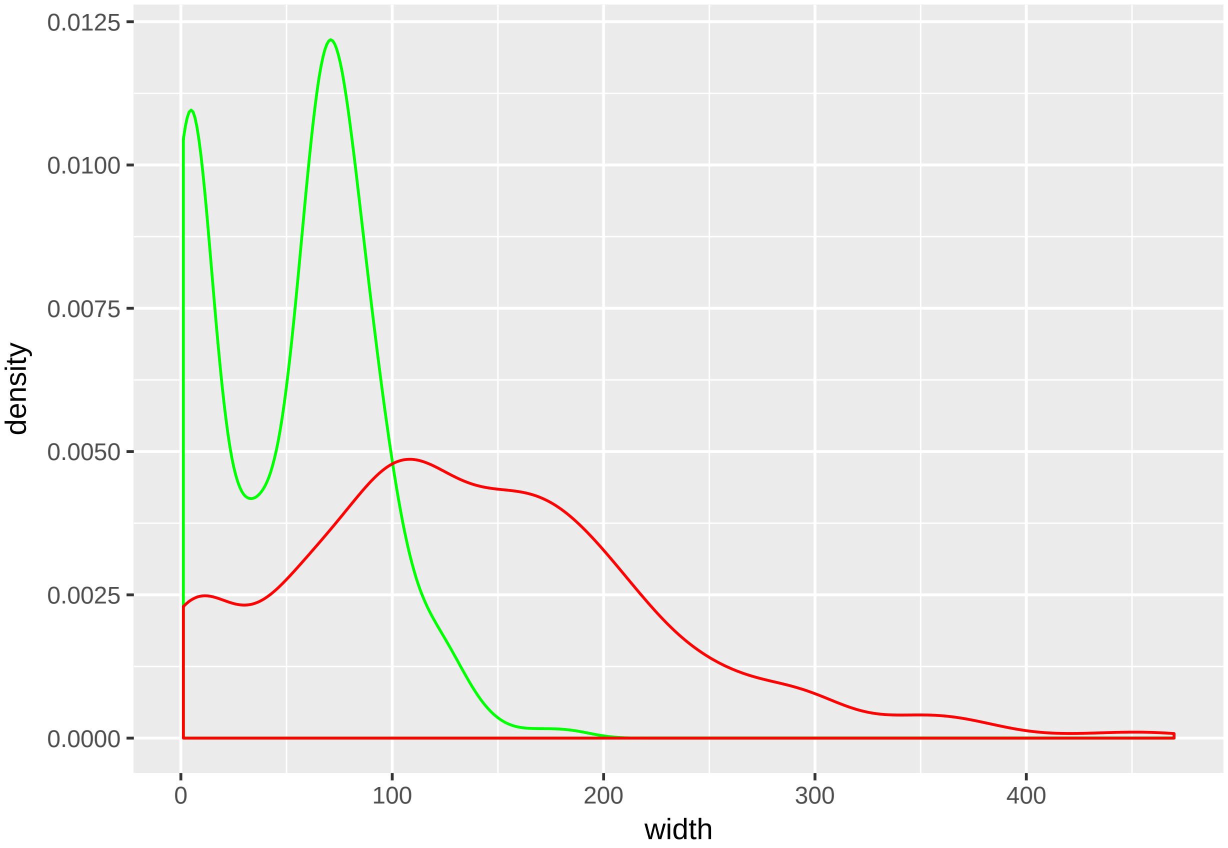


```
library(ggplot2)
haplo_length<-read.csv("~/Desktop/eeb-177/eeb-174-final-project/haplomeasurelength.csv",

#head(haplo_length)
plot12 <-ggplot(haplo_length, aes(length)) + geom_density(kernel= "gaussian")
  #geom_freqpoly(binwidth=20)
print(plot12)
```



I can now overlay/superimpose the width and length graphs on top of each other:

```
plot13 <- ggplot() +
geom_density(data=haplo_width, aes(width), color='green') +
geom_density(data=haplo_length, aes(length), color='red')
plot13
```



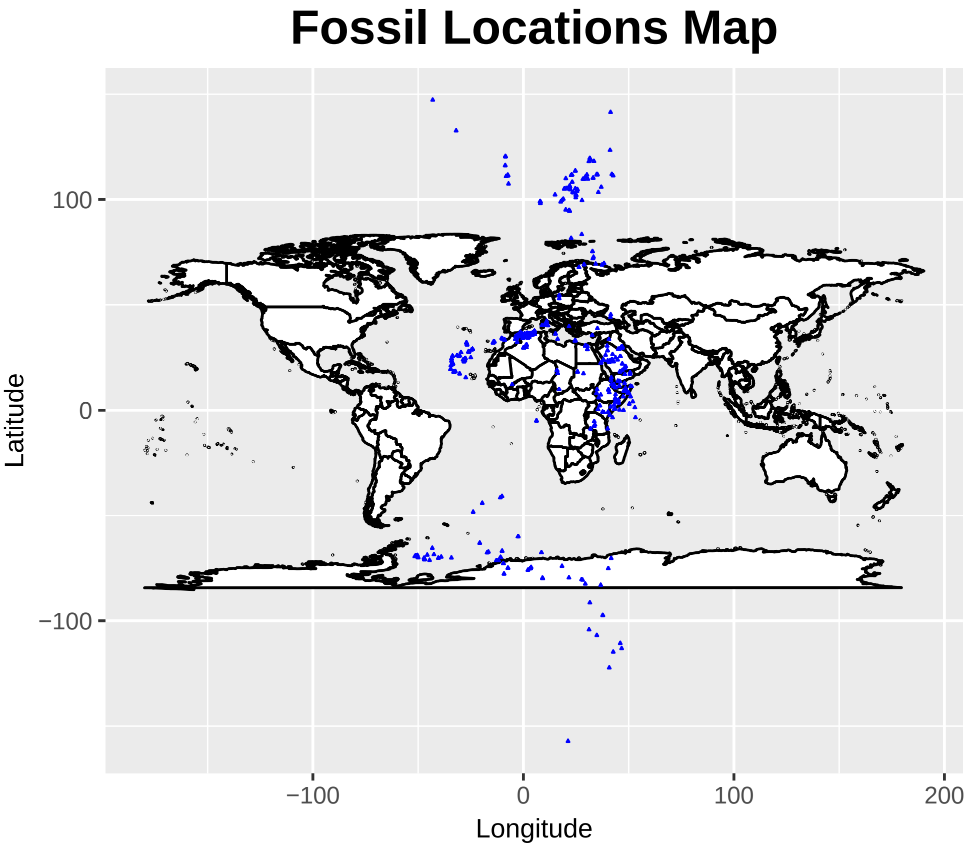
Reproducing PaleoBioDB Fossil Map

After downloading the data from PaleoBioDB, I isolated the lonitude and latitude columns directly from the shell using the following command:

```
tail -n+15 haplolonglat.csv | cut -d "," -f18,19 > haplolonglatoutput.csv
#I removed all double quotes around the numbers using this command:
sed 's/\"//g' haplolonglatoutput.csv > haplolonglatoutput2.csv
```

```
library(ggplot2)
library(maps)

world_map <- map_data("world")

mymap <- ggplot() + coord_fixed() +
  xlab("") + ylab("")

my_world_map <- mymap + geom_polygon(data=world_map, aes(x=long, y=lat, group=group), co

#my_world_map

#my longitude data and latitude data
```

```
haplolonglat <- read.csv("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/hap
names(haplolonglat)<- c("lat", "long")
#head(haplolonglat)

haplo_map <- my_world_map + geom_point(data=haplolonglat,aes(x=long, y=lat), colour="Blu

haplo_map
```



Pyrate

```
#First I need to determine which species are still extant today
# this function returns the specie name and their min age
def haploextant(filename):
# this function opens data file and extracts name and minage as a dict
    haploDict = {}
    fobj = open(filename,"r", encoding = "ISO-8859-15")
    # this opens the file
    reader = fobj.readlines()  # read the file
    for line in reader:
        line = line.split(",")
        name = line[5][1:-1]
        min_age = line[15][1:-1]
        haploDict[name] = min_age
    return haploDict
# However I would like only those that are extant today, with age 0
import csv
output = open("haploextant.csv", "w")
with open("/home/eeb177-student/Desktop/eeb-177/eeb-174-final-project/haploformatted.csv
    reader = csv.reader(infh)
    for row in reader:
        if row[15] == "0":
            species = row[5]
            genus = row[5].split(" ")[0]
            output.write(row[5])
output.close()
```

```
# After writing to output file
# I need to extract unique species to ensure lack of repetitions
less haploextant.csv | cut -d "," -f1 | sort | uniq >haploextant2.csv
less haploextant2.csv | cut -d "," -f1 | sort | uniq | wc -l
# I created a haplo_pyrate directory where I downloaded the data using:
wget -O canid_occ.csv "https://paleobiodb.org/data1.2/occs/list.csv?base_name=Haplorhini
```

```
#write an R script that uses the created R script on our datafile
source("~/PyRate/pyrate_utilities.r")
extant_haplorhini = c("Chlorocebus aethiops","Colobus guereza","Homo
                      sapiens","Papio hamadryas","Theropithecus
                      oswaldi","Papio anubis","Protopithecus
                      brasiliensis","Alouatta mauroi","Papio
                      cynocephalus","Alouatta palliata","Alouatta
                      seniculus","Aotus trivirgatus","Ateles
                      geoffroyi","Ateles paniscus","Callimico
                      goeldii","Callithrix pygmaea","Cebus
                      apella","Cebus capucinus","Cebus
                      nigrivittatus","Cercopithecus
                      mitis","Cercopithecus pygerythrus","Chlorocebus
                      patas","Insulacebus toussaintiana","Lagothrix
                      lagotricha","Leontopithecus midas","Saguinus
                      fuscicollis","Saimiri sciureus")
extract.ages.pbdb(file= "haplopyr_occ.csv",extant_species=extant_haplorhini)
```

```
#Let's run Pyrate!
python ~/PyRate/PyRate.py haplopyr_occ_PyRate.py -n 2500000
```

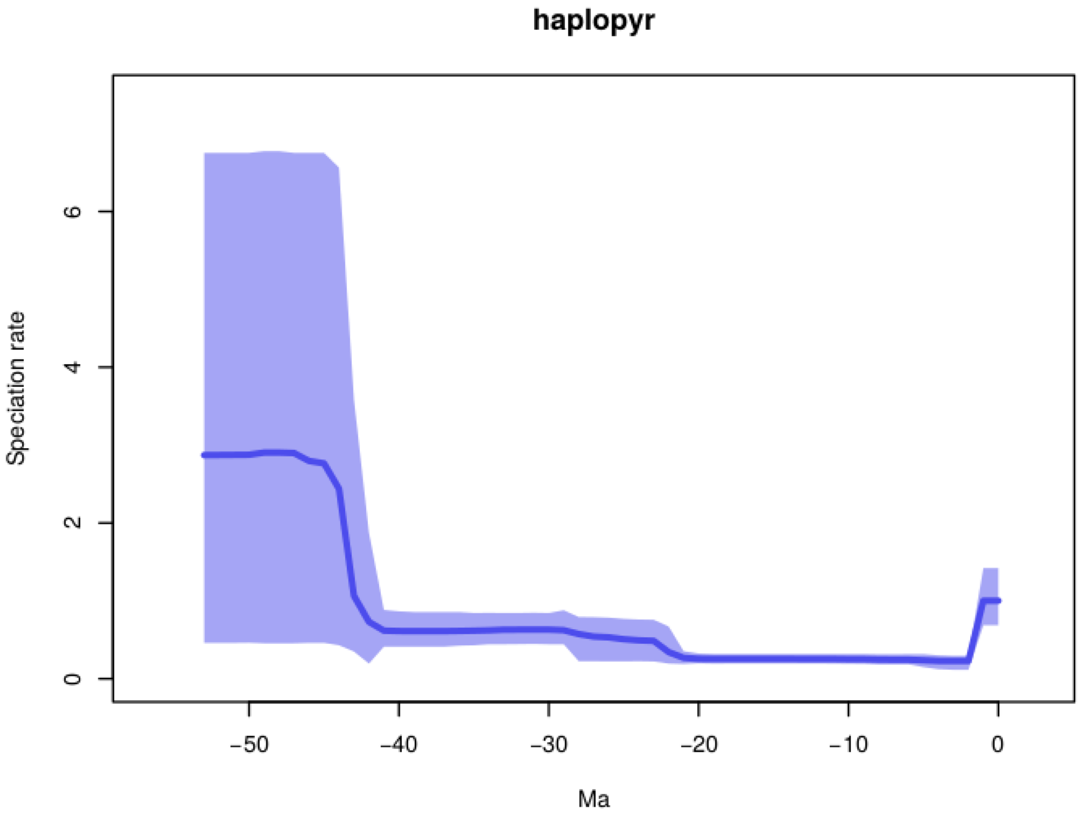


Figure 18: Speciation Rate

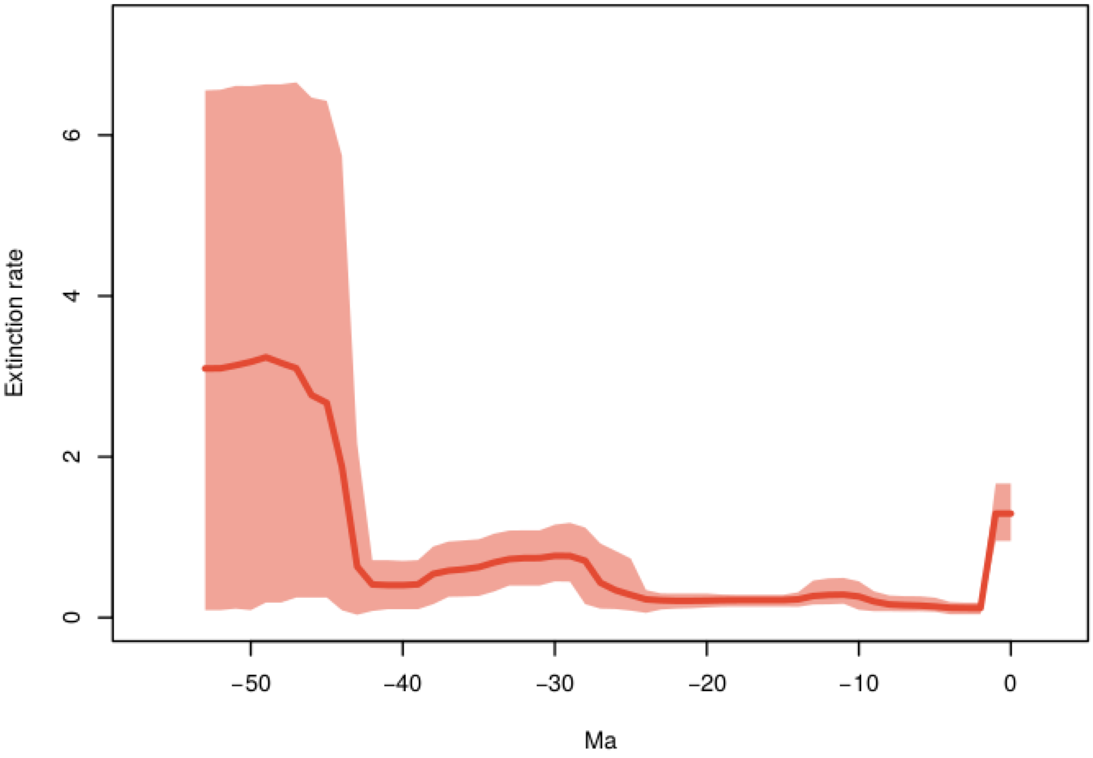


Figure 19: Extinction Rate

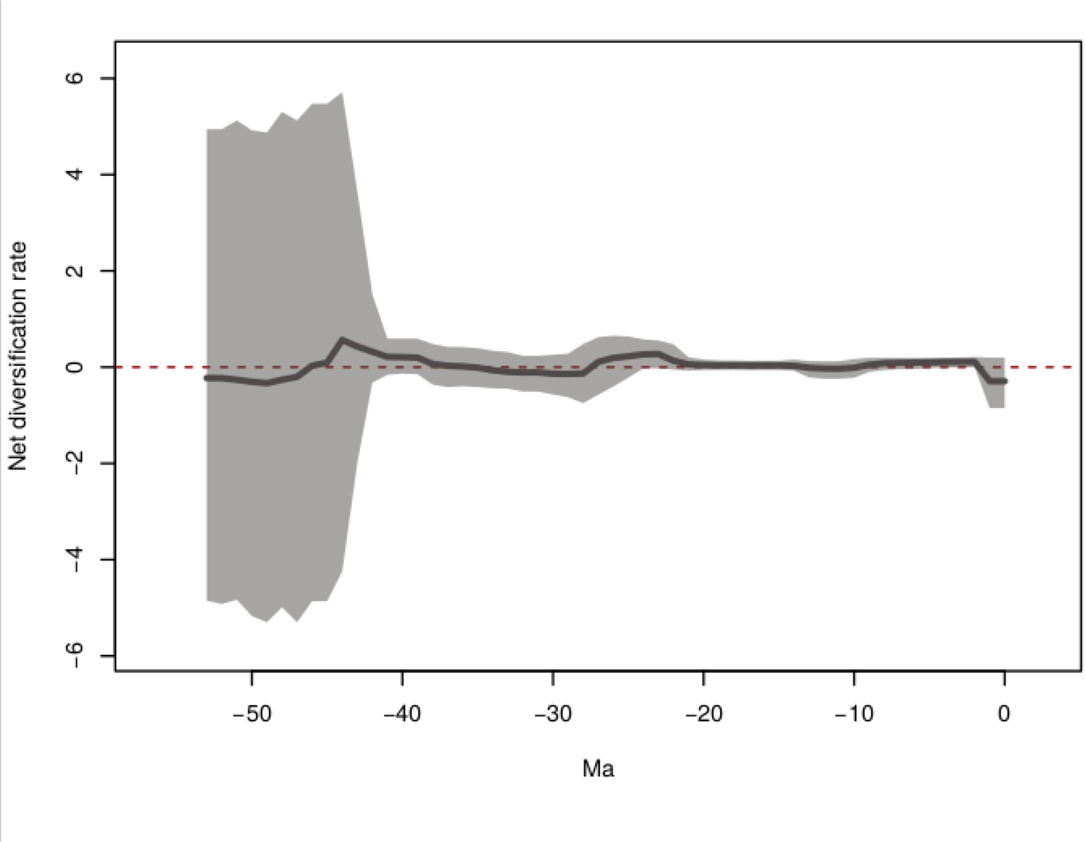


Figure 20: Net Diversification Rate

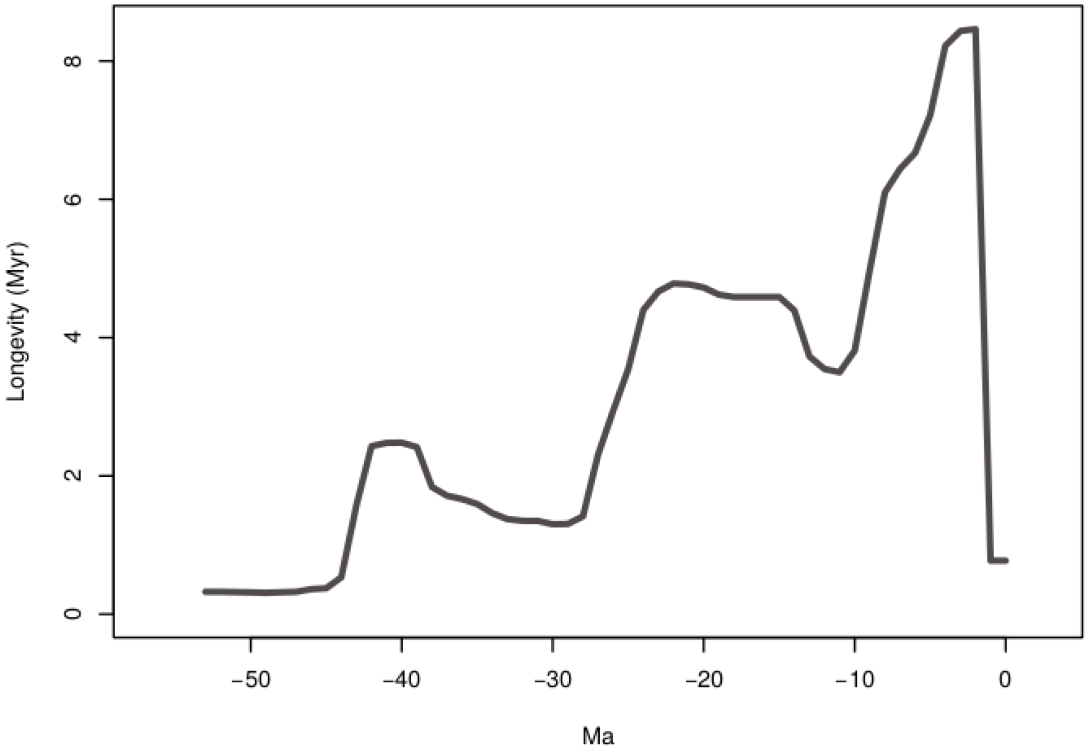


Figure 21: Longevity

Results

The Pyrate generated Speciation Rate graph shows a speciation rate of approximately 3 from -50 ma to -40 ma. The rate decreases to 1 from -40 ma to -20 ma, and decreases again to 0 from -20 ma to -2 ma. The rate increases again to 1 from -2 ma to 0 ma. The Pyrate generated Extinction Rate graph shows an extinction rate of approximately 3 from -50 ma to -40 ma. The rate decreases to 1 from -40 ma to -20 ma, and decreases again to 0 from -20 ma to -2 ma. The rate increases again to 1 from -2 ma to 0 ma. The Pyrate generated Net Diversification Rate shows a net diversification rate approximately -0.5 from -50 ma to -35 ma. The rate increases to 0.5 from -35 ma to -40 ma. The rate drops to 0 from -40 ma to -30 ma and increases to 0.5 from -32 ma to -20 ma. The rate drops again to 0 from -20 ma to -2 ma. From -2 ma to 0 ma, the rate drops to -0.5. The Pyrate generated Longevity graph shows a longevity of 1 myr at -50 ma, 2.5 myr at -40 ma, 1.5 myr at -30 ma, 5 myr at -20 ma, and 4 myr at -10 ma. There a spike at 0 ma to 8 myr and and a drop to 1 myrs. The ggplot and matplot graphs portraying the density of the Average Width Measurements portray bimodal gaussian graphs. The Average Length gaussian graph portrays a bimodal or possibly trimodal graph. In general, both the the width and length measurements vary greatly. Speciation richness graph shows a gradual increase in richness over time. Mapped locations of occurrences include Africa, South America, and Asia.

Discussion

Haplorhine primates are great examples of adaptive radiation or the evolution of a group into a large range of types suited for certain living styles or locations (Fleming & Kress 2013, Fabre *et al.* (2009)). Adaptive radiations can be seen in related groups which evolved in a somewhat short period of time and it includes the utilization of accessible resources and habitat expanse by a specific clade. Adaptive radiation first includes a rapid occupation of habitat living space which is then followed by a decrease of diversification as habitat living space fills up slowly. The decrease of diversification indicates that the rate of habitat occupation is influenced by diversity. Generally, there is a negative relationship between diversification rate, or the difference between speciation rates and extinction rates, and species richness (Fleming & Kress 2013). The discussed relationship suggests that adaptive radiations will halt when accessible habitat space is full and that clade size is more likely to be independent of clade age. As habitat space gets occupied, it is expected that there would be an increase in speciation and species within clades as a consequence of competition for resources and co-evolution with food plants (Fleming & Kress 2013). In addition to a general increase in species richness, there is an increase in morphological diversity, also driven by competition for resources and coevolution leading to a variety of morphologically distinct species (Fleming & Kress 2013). Therefore, adaptive radiation involves both the production of new species and morphologies or traits (Fleming & Kress 2013). The bimodal graphs produced could be referring to the distinct morphologies of the two clades which diverged from Haplorhini. In general, both the width and length measurements varied greatly.

Conclusion and Future Directions

Regarding primate evolution, varying opinions have persisted regarding the palaeobiology and phylogenetic relationships of early primates due to disparities in the fossil record. In this paper I have explored Haplorhini speciation and diversification, along with their morphology using data from an online database named PaleoBiology Database. Utilizing RStudio and Python methods, I have reconstructed some graphs generated by an R package called PaleoBioDB. I have also used a python program called PyRate which estimates speciation, extinction, and preservation rates from fossil occurrence data. In addition to providing further support for the morphological difference between different clades, the analysis shows an increase in speciation richness but a decrease in speciation rate through time. In the future, an exploration of plant and animal coevolution could shed light on the morphologies of closely interacting species and their evolution.

References

Fabre, P.-., Rodrigues, A. & Douzery, E. (2009). Patterns of macroevolution among primates inferred from a supermatrix of mitochondrial and nuclear DNA. *Molecular Phylogenetics and Evolution*, **53**, 808–825.

Fleming, T.H. & Kress, W.J. (2013). *The ornaments of life.* University of Chicago Press.

Ni, X., Gebo, D.L., Dagosto, M., Meng, J., Tafforeau, P., Flynn, J.J. & Beard, K.C. (2013). The oldest known primate skeleton and early haplorhine evolution. *Nature*, **498**, 60–64.

Silvestro, D., Salamin, N. & Schnitzler, J. (2014). PyRate: A new program to estimate speciation and extinction rates from incomplete fossil data (L. Revell, Ed.). *Methods in Ecology and Evolution*, **5**, 1126–1131.