

**Computer Science in Modern Biology**  
**Conservation Biology**  
Aug 12-13, 2020 1:30-4:00 PM

**Instructor:** Hayley Lanier  
**TAs:** Karen Castillioni  
Waldir Miron Berbel Filho

## **Class Overview**

This course is intended to provide you with an opportunity to apply computing to conservation biology questions and data, with an emphasis on mammal conservation. Computing skills emphasized include basic plotting and interpretation of results (day 1), running an analysis to predict population health (day 2), and learning how to use Google to develop an understanding of what to code and how to interpret resulting output (both days)

## **Learning Objectives**

- 1) Plot species conservation assessment data from the IUCN Red List to assess which groups of mammals are the most threatened
- 2) Plot population census data to evaluate changes in population size over time (i.e., population increases or decreases)
- 3) Describe why the absolute number of individuals in a population is not sufficient for determining population health, and identify alternatives that may be more accurate
- 4) Use a model of population viability to predict the probability of extinction in the next 50 years and determine which species to prioritize for conservation actions

## **Day 1 Overview**

- 1) Introductions and expectations
- 2) Discussion & activity: What is conservation biology?
- 3) Break (10 minutes)
- 4) Hands-on in R: digging into the IUCN's conservation threat data
  - a. How do we categorize and discuss which species are at greater threat?
  - b. Predict which groups of mammals are the most threatened
  - c. Evaluate based on the data
  - d. Using population trends to inform threat status
- 5) Break (10 minutes)
- 6) Hands-on in R: exploring population census data to understand population trends
  - a. Case study: Yellowstone grizzly bears
  - b. Application to real data for other species
  - c. Report and interpretation
- 7) Recap and discuss plan for tomorrow

## **Day 2 Overview**

- 1) Welcome back and recap
- 2) Discussion & activity: Probability and how it can help us with conservation planning
- 3) Break (10 minutes)
- 4) Hands-on in R: Population Viability Analyses (PVA) and predicting extinction
  - a. Case study: running a PVA on Yellowstone grizzly bears
  - b. Discussion of output and results

- 5) Break (10 minutes)
- 6) Hands-on in R: PVAs and predicting extinction probabilities for other datasets
  - a. Application to real data for other species
  - b. Small group discussion and interpretation
- 7) Full group recap and conservation prioritization
- 8) Course recap and “next steps”

### **Day 1: Hands on in R - plotting species threat data and population size**

First we will start off working with conservation threat assessment data. These data summarize all of the factors that impact the threat level of a population, and establish a ranking system that allows us to compare the level of threats between species.

You can either type the commands yourself, paste the **commands shown in courier** into R, or run the commands directly from the script. Editing existing code, either pasted in or from a script, is one of the easiest way to ensure that the command you are using is correctly formatted and learn about how the code works. Consider also trying to ‘break’ the code, by removing elements – it’s a great way to figure out what each portion of the code does!

Code shown **in yellow** should be replaced with information based upon your own system.

Anything after a # is commented out, and will not be read by R. **Questions for your consideration** are shown in orange *italics*.

1.1 First, set up your working directory. This is where R will read and write files from – so it’s a pretty essential first set. You can do this by navigating there using the Session tab in R studio, and then selecting ‘Set working directory’ and ‘Choose directory’ and then navigating to that file, or you can use the setwd command below.

```
setwd("~/Desktop/Computing/ConsBio") # Swap the part in yellow for the correct path.
```

1.2 Read in dataset that has all orders of mammals and the number of species found in each Red List category. IUCN Red List Categories: EX - Extinct, CR - Critically Endangered, EN - Endangered, VU - Vulnerable, NT - Near Threatened, DD - Data Deficient, LC - Least Concern. Species in ‘threatened’ categories include those listed as CR, EN, or VU. Original data are located at <https://www.iucnredlist.org/resources/summary-statistics>

```
mammals<-read.table("mammals_IUCN.txt",row.names=1,header=T,sep="\t")
```

1.3. The first thing to do is to look at the dataset. R has lots of ways to do this – three of my favorites are the following.

```
View(mammals) # look at the dataset
names(mammals) # list the column names for the mammals dataset
```

```
row.names(mammals) # list the names for each row of data in the mammals dataset
```

*Q1.3. Looking at the dataset as a table, what can you see? What do the column names refer to?  
(Hint: you may have to paste some of them into google)*

1.4. Some housekeeping before we start plotting

```
iucn.status<-row.names(mammals) # creates a vector of names for categories of  
conservation status from the row names in the dataset  
iucn.colors<=  
c("grey5", "red", "orange", "yellow", "grey", "green", "darkgreen") # creates a  
vector of colors associated with each conservation status rank
```

1.5 How much conservation risk are mammals under? Plot the data for all mammals by conservation status to find out about mammals as a whole.

```
pie(mammals$Total, labels=iucn.status, col=iucn.colors, main="All  
mammals")
```

1.5.1 Count the number of mammal species in each category

```
rowSums(mammals[, 1:27]) # sums across all rows, for all columns all orders (first 27  
columns of the dataset); Should be the same results as typing: mammals$Total to read out  
the total number column (which is column 28)
```

1.5.2 Calculate that number as a % of total mammals

```
# This calculates the sum within each row by conservation status, divides it by the total number  
of mammals (sum(mammals)) and then rounds the result to just two significant digits)  
round(rowSums(mammals)/sum(mammals), 2)
```

*Q1.5. What % of mammals is threatened (CR, EN, or VU)? (Hint: R can work like a calculator and add things up!).*

1.6 Data-driven decision making. Which group of mammals is at more risk?

To start out we will barplot all of the data.

```
barplot(as.matrix(mammals[1:27]), col=iucn.colors, ylab="number of  
species", horiz=T, las=2, cex.names=0.6)
```

*Q1.6. Based on these results, which group of mammals would you argue is in the most need of conservation?*

1.7 Admittedly that is whole lot of data. Let's just compare a subset of the IUCN data, to see how groups compare.

```
comp.table<-cbind(mammals$Carnivora, mammals$Lagomorpha,  
mammals$Primates, mammals$Pholidota) # Binds together columns representing four  
Orders of mammals to compare
```

*Q1.7. What are each of these groups? Google each of the column names to learn what species are within the group.*

1.7.1 Plot the result together and add a legend

```
barplot(comp.table,col=iucn.colors,names=c("Carnivora","Lagomorphs","Primates","Pangolins"),ylab="number of species",main="Which group is at higher risk?")
legend(4,470,iucn.status,pch=22,pt.bg=iucn.colors)
```

*Q1.7.1. Based on these results, which group of mammals would you argue is in the most need of conservation?*

1.8 Let's look at this one more way – with pie charts. They don't capture the total number, but they can show you the % of each group that falls into each category.

1.8.1 First, set up it up to have two plots in window (1 row, 1 column)

```
par(mfrow=c(1,2)) # Set up the plot background to show 2 plots in 1 row & 2 columns
```

*Q1.8.1. Can you guess what you would need to do to re-set the plot to only show one plot per window?*

1.8.2 Plot the conservation status for each group as a separate pie chart

```
pie(mammals$Carnivora,labels=iucn.status,col=iucn.colors,main="Carnivores")
pie(mammals$Lagomorpha,labels=iucn.status,col=iucn.colors,main="Lagomorphs")
pie(mammals$Primates,labels=iucn.status,col=iucn.colors,main="Primates")
pie(mammals$Pholidota,labels=iucn.status,col=iucn.colors,main="Pangolins")
```

*Q1.8.2. Based upon those data, which group of mammals do you think is at higher risk of extinction? Which group do you think should be more highly prioritized as far as conservation efforts?*

1.9 Now we're going to zoom into species-level data for just one of these groups: Lagomorphs (rabbits, hares, and pikas). Not only are they a manageable size, but as a lagomorph researcher I already had the data handy.

1.9.1. First, read in dataset and take a look at it

```
lago<-read.csv("lagomorph-status.csv")
View(lago)
```

*Q1.9.1. Both this dataset and the previous one for all mammals provide information on conservation ranks. How do the datasets differ as far as structure? What are Leporidae and Ochotonidae?*

1.9.2 Organize dataset and set colors for plotting (similar to before)

```
lago$status<-factor(lago$status,levels=c("CR","EN","VU","DD",
"NT","LC")) # sets the order for the level of conservation threat (high to low; otherwise R
will simply alphabetize these ranks
```

```
lago$Trend<-factor(lago$Trend,levels=c("Unknown","Decreasing",
"Stable","Increasing")) # set the order for the level of conservation threat (high to low;
otherwise it will just be alphabetized)
```

```
lago.colors<-c("red","orange","yellow","grey","green","darkgreen") #
same colors as earlier, but without including a color for extinct species, as they are not included
in this dataset
```

```
lago.col.trend<-c("grey","darkorange","lightblue","darkgreen")
```

1.9.3 Because these data aren't already summarized, we will need to ask R to summarize them for us so that we can plot them to match the previous plots

```
ls<-table(lago$Family,lago$status) # summarizes data by family and conservation
threat status
```

1.10 Plot the summarized data as pie charts. Note, these plots call each row of the table separately using the ls[1,] and ls[2,].

```
pie(ls[1,], col=lago.colors, main="Rabbits & hares")
pie(ls[2,], col=lago.colors, main="Pikas")
```

1.11 Plot the summarized data as a table

```
barplot(t(ls),col=lago.colors,ylab="number of species",
names=c("rabbits & hares","pikas"),main="Conservation Status")
```

*Q1.11. Which group (rabbits & hares or pikas) appears to be at greater risk of extinction?*

1.12 How do population trends inform our understanding of threats in lagomorphs? Luckily, we also have information on how populations are changing in these species.

```
trends<-table(lago$Family, lago$Trend)
barplot(t(trends), col=lago.col.trend, ylab="number of species",
names=c("rabbits & hares","pikas"), main="Population Trend")
```

1.13 Plot population trends as pie charts

```
pie(trends[1,],col=lago.col.trend,main="Rabbits & hares")
pie(trends[2,],col=lago.col.trend,main="Pikas")
```

*Q1.13. Which group (rabbits & hares or pikas) shows more evidence of population declines?
Does this mean they are at a greater level of risk?*

## 1.14 Working with census data to understand population trends.

Population trend information (increasing, decreasing, stable) is only as good as the underlying evidence for those trends. For species where we are lucky enough to have people doing that work we can begin to look at how population sizes are changing over time.

### 1.14.1 Getting set up. Install the `popbio` package. Packages are developed sets of tools (and often test data). This package includes useful functions for conservation/population biology.

```
install.packages("popbio") #  
  
library(popbio) # turns on the popbio [ackage]  
data(grizzly) # loads the Yellowstone grizzly dataset that comes with the popbio package
```

### 1.14.2 Let's learn more about the grizzly dataset!

```
View(grizzly) # look at the dataset  
?grizzly # learn more about the dataset from the package  
names(grizzly) # learn what the column names are for this dataset
```

#### *Q1.14.2. What do the data in each of the columns correspond to?*

### 1.14.3 Some housekeeping commands to make everything downstream run more smoothly.

```
attach(grizzly) # makes dataset run without using the grizzly$  
par(mfrow=c(1,1)) # plot one plot per window  
par(bty="n") # gets rid of the default box around a plot (ugh, chart junk!)
```

## 1.15 How do we know if a population is growing or shrinking? Use census (count) data!

```
# Plot 25 years of grizzly population changes in Yellowstone  
plot(year[1:25], N[1:25]) # a basic, no-frills plot (not quite as easy to interpret)
```

```
# A better looking plot of the same data, but a few additional commands to make it look  
somewhat more attractive.
```

```
plot(year[1:25], N[1:25], type='o', pch=16, las=1, xlab="Year",  
ylab="Adult females", main="Yellowstone grizzly bears")
```

#### *Q1.15.1 Why does this plot focus on adult females?*

#### *Q1.15.2 Based upon these data, is the population growing, shrinking, or relatively stable?*

```
# export your plot as a pdf  
dev.print(pdf,file="grizzlyBears_1960-1983.pdf") # Handy command to print  
out a plot as a pdf as shown on the screen
```

1.16. It turns out those weren't the only data we have for this population. Now let's plot the full range of the Yellowstone data for all the years we have data.

```
plot(year, N, type='o', pch=16, las=1, xlab="Year", ylab="Adult females", main="Yellowstone grizzly bears")
```

*Q1.16 Based upon these data, is the population growing, shrinking, or relatively stable?*

1.17. Okay, let's apply what we have done so far to some other datasets from wild populations. The goal is to have different groups of students work on different species – but you are welcome to plot as many different datasets as you want.

1.17.1 First, some housekeeping to detach the grizzly data and load a different dataset.

```
detach(grizzly)
```

```
# Read in your dataset by either modifying the command below or changing which line is commented out
```

```
popDat<-read.table("yourFileNameHere",header=T,sep="\t")
```

```
# Desert Yellowhead (a flowering plant from Wyoming)
```

```
#popDat<-read.table("DesertYellowhead.txt",header=T,sep="\t")
```

```
# Collared pikas (from the Yukon Territory, Canada)
```

```
#popDat<-read.table("collared_pikas.txt",header=T,sep="\t")
```

```
# Red-cockaded woodpeckers (North Carolina population OR Central Florida – your pick)
```

```
#popDat<-read.table("woodpecker_NC.txt",header=T,sep="\t")
```

```
#popDat<-read.table("woodpecker_CF.txt",header=T,sep="\t")
```

```
# Vancouver Island marmot
```

```
#popDat<-read.table("Vl_marmot.txt",header=T,sep="\t")
```

```
# Olympic marmot
```

```
#popDat<-read.table("Oly-Marmots.txt",header=T,sep="\t")
```

```
attach(popDat)
```

1.17.2 Plot your data and look at the trends in the dataset.

```
# Change 'Your Dataset' to the name of the species/population you are plotting
```

```
plot(year, N, type='o', pch=16, col="darkblue", las=1, xlab="Year", ylab="population size", main="Your Dataset")
```

*Q1.17 Based upon these data, is the population growing, shrinking, or relatively stable? Do some quick Googling – what can you find out about the species and why it might be changing?*

## Day 2: Hands on in R – predicting extinction probability with a population viability analysis

We're going to start where we left off yesterday, working with population census (count) data. These data are generally some variant on how many individuals there are in a population. Often for mammals these estimates will focus on adult females, but sometimes the only data available don't distinguish between males and females. Or, for many species there is no different (e.g., plants).

2.1 If you have not yet done so, set your working director to the class folder. Also, you should have installed the package popbio yesterday, but if not you can use the install.packages command.

```
#setwd("~/Desktop/Computing/ConsBio")  
#install.packages("popbio") # should have been installed on day 1  
  
library(popbio) # loads the popbio package  
data(grizzly) # loads the Yellowstone grizzly dataset  
attach(grizzly) # re-attach the dataset, so that just the column names can be used
```

2.1.1 Re-plot the Yellowstone data to take another look at it. It should look the same as yesterday, ending with a relatively large population increase.

```
plot(year, N, type='o', pch=16, las=1, xlab="Year", ylab="Adult  
females", main="Yellowstone grizzly bears")
```

2.2 Running a Population Viability Analysis (PVA) on census (count) data

The input for this type of PVA is literally census/count data for a population. The more times a population has been surveyed the better the data are, and the better the PVA.

First, we need to pull some important information about your population from the census data. The code below calculates the rate of change associated with each sampling interval (i.e., how much does the population size change between one census and the next). While these calculations are things you could easily calculate by hand computers make it faster and more repeatable.

```
nt<-length(N) # counts the number of times your population has been surveyed by  
# counting the number of rows in your dataset  
logN<-log(N[-1]/N[-nt]) # determines change in population size in one year relative to  
# the change across all years (on a log scale)  
x<-sqrt(year[-1]-year[-length(year)]) # units of time (accounts for sampling  
# events that are >1 year apart)  
y<-logN/x # change in population size relative to time between sampling events
```

2.3 Plot the *cumulative distribution function* which is the probability of extinction at some point in the future. In essence this asks the question ‘what is the probability that this population will have gone effectively extinct by a certain time in the future?’

A few things to know about this approach. First, changes in population growth rate (`mu`) are calculated directly from your data. However, you can set the current number of individuals in your population (`Nc`); right now this is set at the most recent number of individuals in your population (`N[nt]`). You can also set the 'effective extinction' threshold (`Ne`), which is the number of individuals at which the population is considered 'effectively extinct'. Right now `Ne` is set to 20, which is a very small population for grizzly bears and not an unreasonable cut off.

```
countCDFxt(mu=mean(logN), sig2=var(logN), nt= length(year)-1, tq=
max(year)-min(year), Nc=N[nt], Ne=20)
```

2.3.1 The output you are looking at should include both a plotted result and a table of values (in the R window to the left). The table includes 3 parameters estimates (taken from your model) projected across the next 50-years: the best estimate for the probability of your population going extinct (`Gbest`), as well as the lower (`Glo`) and upper (`Gup`) confidence limits on that estimate. You may notice that these values aren't very human-readable (often a number followed by e- and another number; e.g., `1.724290e-01`). This is scientific notation; `R` uses it because numbers can easily become quite small or really large. An easy way to convert most number close to 0 back to something readable is to copy it and paste it in the `R` console window and hit return.

*Q2.3.1 At the end of 50 years, what is the best estimate for the probability of quasi-extinction for the grizzly? What is the high-end estimate (`Gup`)?*

2.3.2 The plotted result is a smoothed plot showing values from the table you just looked at. It shows the best parameter estimate (solid blue line), as well as the lower and upper confidence limits (dashed red lines).

*Q2.3.1 Looking at this graph, how does the cumulative probability of quasi-extinction change over time? Why do you think it changes the way it does?*

2.4 Applying a PVA to the populations you looked at yesterday.

2.4.1 First, cleanup the digital workspace

```
detach(grizzly) # some housekeeping to detach the grizzly data
rm(logN,nt,x,y) # remove all of the calculated parameters from the grizzly dataset
```

2.4.2 Read in your dataset by either modifying the command below or changing which line is commented out

```
popDat<-read.table("yourFileNameHere",header=T,sep="\t")
```

```
# Desert Yellowhead (a flowering plant from Wyoming)
```

```
#popDat<-read.table("DesertYellowhead.txt",header=T,sep="\t")
```

```
# Collared pikas (from the Yukon Territory, Canada)
```

```
#popDat<-read.table("collared_pikas.txt",header=T,sep="\t")
```

```
# Red-cockaded woodpeckers (North Carolina population)
```

```
#popDat<-read.table("woodpecker_NC.txt",header=T,sep="\t")
```

```

#popDat<-read.table("woodpecker_CF.txt",header=T,sep="\t")
# Vancouver Island marmot
#popDat<-read.table("Vl_marmot.txt",header=T,sep="\t")
# Olympic marmot
popDat<-read.table("Oly-Marmots.txt",header=T,sep="\t")

attach(popDat)

```

2.4.3 plot your data (just to check that you have read in the right data). Don't forget to change the 'Your Dataset' to the name of the species/population you are plotting

```

plot(year, N, type='o', pch=16, col="darkblue", las=1, xlab="Year",
ylab="population size", main="Your Dataset")

```

2.4.4 We will again pull some important information about your population from the census data. As above, this calculates the rate of change associated with each sampling interval (i.e., how much does the population size change between one headcount and the next)

```

nt<-length(N) # counts the number of times your population has been surveyed by
counting the number of rows in your dataset
logN<-log(N[-1]/N[-nt]) # determines change in population size in one year relative to
the change across all years (on a log scale)
x<-sqrt(year[-1]-year[-length(year)]) # units of time (accounts for sampling
events that are >1 year apart)
y<-logN/x # change in population size relative to time between sampling events

```

2.4.5 Now plot the cumulative distribution function for the probability of extinction in the future. Remember, this asks the question 'what is the probability that this population will have gone effectively extinct by a certain time in the future?' While changes in population growth rate ( $\mu$ ) are calculated directly from your data, there are other parameters here you need to think about.

- 1) The current number of individuals in your population ( $N_c$ ) is set as the most recent number of individuals in your population ( $N_e=N[nt]$ ), but you can replace  $N[nt]$  with any value. You could do some quick googling to see if you can come up with a more accurate, and recent estimate for population size, or just play around with parameters to see what their effect is.
- 2) Consider also what the appropriate 'effective extinction' threshold ( $N_e$ ) should be, and set it at that number. Again, this is the number of individuals at which the population would be considered to be 'effectively extinct' (i.e., too low to recover from) so generally this number is more than 0.

```

countCDFxt(mu=mean(logN), sig2=var(logN), nt= length(year)-1, tq=
max(year)-min(year), Nc=N[nt], Ne=20)

```

*Q2.4.5 At the end of 50 years, what is the best estimate for the probability of quasi-extinction for your population? What is the high-end estimate (Gup)? How do these estimates change when you switch  $N_e$ ?*

```
# When you're all done clean up your workspace
detach(popDat); rm(logN,nt,x,y)

#####
# UNDER THE HOOD - Bonus information on PVA parameters for curious folks #
#####

First, if you haven't had stats or an ecology course yet this section may seem somewhat confusing. That is totally okay, and completely normal! If you like this stuff, and you develop a bit more knowledge background this section will make far more sense.
```

This type of PVA relies on a parameter population biologists call 'mu' ( $\mu$ ) - the instantaneous rate of increase in a population. It does this by calculating the difference between each population during each sampling interval, and then combining those values to create one estimate for the value. To understand some of how it works you can actually unpack a bit more to see how it is estimated from your data.

First, we pull the same parameters as before from your data:

```
nt<-length(N)
logN<-log(N[-1]/N[-nt])
x<-sqrt(year[-1]-year[-length(year)])
y<-logN/x
```

Then we can then plot the time between sampling events (x) against the change per sampling unit (y). This plot will look pretty darn boring in most cases, but by running a regression and setting the intercept to 0 we can use the slope of the regression line to estimate  $\mu$ , the rate of population growth in your sample. This captures whether your population has been growing or shrinking over time.

```
mod<-lm(y~0 + x) # Runs the linear regression (think y = mx + b but the intercept (b) has been replaced by 0), and saves the output as 'mod'
```

Plot your data to look at the relationships between change per sampling period (generally year) and overall population growth rate

```
plot(x,y, xlim=c(0,1.8), ylim=c(-.3,.3), pch=16, las=1,
      xlab=expression((t[t+1]-t[i])^{1/2}),
      ylab=expression(log(N[t+1]/N[t]) / (t[t+1]-t[i])^{1/2}) ,
      main=expression(paste("Estimating ", mu, " and ", sigma^2, " using regression")))
abline(mod)           #code to add the trendline to your plot
# if the trendline is positive your population is growing
# if the trendline is negative your population is shrinking
# if the trendline is flat your population is stable
```

Estimate the slope from the regression line in your previous graph to calculate  $\mu$ , the average change in your population.

```
mu<- coef(mod)
```

The mean square residual value from the analysis of variance table is an estimator for  $\sigma^2$ . If you haven't learned what an analysis of variance or a mean square residual is yet, don't worry – just know that this is a way to estimate how variable your dataset is.

```
sig2<-anova(mod)[["Mean Sq"]][2]
```

Display both mean population growth rate ( $\mu$ ) and variance ( $\sigma^2$ )

```
c(mean= mu , var= sig2)
```

Calculate the confidence interval for the population growth rate

```
confint(mod,1)
```

Calculate a confidence interval for the variance ( $\sigma^2$ )

```
df1<-length(logN)-1  
df1*sig2 /qchisq(c(.975, .025), df= df1)
```

# Notes

