

# Population & Community Ecology Homework 1

Samantha Apgar & Henry Frye

## Problem 1

In this problem we look to find the density of duckweed plants per  $m^2$  after 47 days. We will estimate the future population using the exponential growth formula,  $N_t = N_0 e^{rt}$  where the initial population,  $N_0$ , is 11.1 duckweed plants per  $m^2$ . We can calculate the rate of unitless population change,  $r$ , by using the above equation.

```
r <- (log(23.5/11.1))/20  
r
```

```
## [1] 0.03750277
```

Thus with  $r = 0.03750$ , we can calculate  $N_{47}$  by using the equation  $N_t = N_0 e^{rt}$  again.

```
N47 <- 11.1*(exp(0.0375*47))  
N47
```

```
## [1] 64.67955
```

Thus after a total of 47 days, the population density will be 64.680 duckweed plants per  $m^2$ .

---

## Problem 2

In this problem we want to find the time at which both Ethiopia and Mexico will have the same population size, while starting at different population sizes and growing at different rates. We will determine this time by setting the exponential growth formula,  $N_t = N_0 e^{rt}$  for each population equal to each other and solving for  $t$  (time). The initial population,  $N_0$ , for Mexico is 121,005,000 and  $r$  is 1.2% and the initial population for Ethiopia,  $N_0$ , is 90,076,000 and  $r$  is 2.9%.

*#SO. basically this problem has the same variable on both sides of the equation and I'm not sure how to*

```
t <- -0.29518/-1.7  
t
```

```
## [1] 0.1736353
```

In 0.1736 years the populations of Mexico and Ethiopia will be equal.

---

## Problem 3

### Part a

Given the information we would like to solve for  $N_7$ , the population in seven years.

```
n_7 <- (1.037^7)*112
n_7
```

```
## [1] 144.434
```

Thus the population density will be 144 (rounded, no units given) at year 7.

## Part b

Using the exponential growth formula,  $N_t = N_0 e^{rt}$ , we would like to solve for an  $r$  value where  $N_0 = 112$  and  $N_t = 144$ . Thus we solve for  $r$  as  $\ln(\frac{N_t}{N_0})/t = r$ .

```
r <- log(144/112)/7
r
```

```
## [1] 0.03590206
```

Thus  $r$  is equal to .036 in the exponential model to achieve the same results as the discrete population growth model.

---

## Problem 4

In this problem, we are given values for the number of collared doves monitored in Great Britain each year. We are asked to estimate  $r$  by plotting the natural log of the number of doves by the time in years.

```
Time <- c(1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964)
Time
```

```
## [1] 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964
```

```
NumDoves <- c(4, 16, 45, 100, 205, 675, 1900, 4650, 10200, 18885)
NumDoves
```

```
## [1] 4 16 45 100 205 675 1900 4650 10200 18885
```

```
LnNumDoves <- log(NumDoves)
LnNumDoves
```

```
## [1] 1.386294 2.772589 3.806662 4.605170 5.323010 6.514713 7.549609
## [8] 8.444622 9.230143 9.846123
```

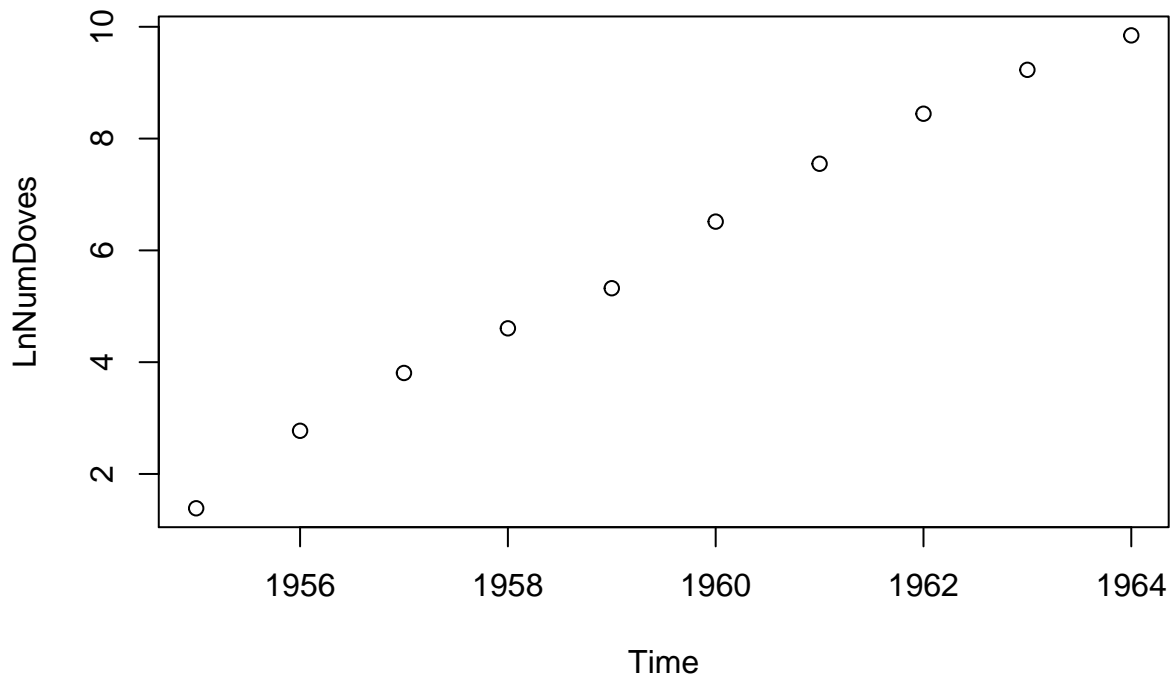
```
lm(LnNumDoves~Time)
```

```
##
## Call:
## lm(formula = LnNumDoves ~ Time)
##
## Coefficients:
## (Intercept)      Time
## -1829.5236      0.9367
```

```
r = 0.9367
```

In the second part of this question, we are asked to plot  $\ln(N)$  versus  $t$ :

```
plot(Time, LnNumDoves)
```



We are also asked to calculate what the population sizes would be each year, assuming exponential growth, given that the initial population size in 1955 is 4 doves.

```
TimeSinceStart <- c(1,2,3,4,5,6,7,8,9)
TimeSinceStart
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
ExponEqu <- 4*exp(0.9367*TimeSinceStart)
ExponEqu
```

```
## [1] 10.20619 26.04158 66.44632 169.54093 432.59171 1103.77827
## [7] 2816.34257 7186.03157 18335.50020
```

```
for(val in Time)
  print(val)
```

```
## [1] 1955
## [1] 1956
## [1] 1957
## [1] 1958
## [1] 1959
## [1] 1960
## [1] 1961
## [1] 1962
## [1] 1963
## [1] 1964
```

The population sizes would be as follows: 1955: 4 doves 1956: 10.20619 doves 1957: 26.04158 doves 1958: 66.44632 doves 1959: 169.54083 doves 1960: 432.59171 doves 1961: 1103.77827 doves 1962: 2816.34257 doves 1963: 7186.03157 doves 1964: 18335.50020 doves

## Problem 5

$l_x$  and  $p_x$  are solved below and placed into the following life history table:

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.3.2
```

```
age <- 0:4
sx <- c(740,280,105,32,0)
bx <- c(0,0.4,1.3,3.8,NA)
lx <- sx/740

px <- vector(mode = 'numeric',length = 5)
for(i in 1:length(lx)){
  px[i] <- (lx[i+1]/lx[i])
}

lifetable <- cbind(age,sx,lx,px,bx)
colnames(lifetable) <- c('Age','S_x','l_x','p_x','b_x')
lifetable <- as.data.frame(lifetable)
kable(lifetable)
```

Age	S_x	l_x	p_x	b_x
0	740	1.0000000	0.3783784	0.0
1	280	0.3783784	0.3750000	0.4
2	105	0.1418919	0.3047619	1.3
3	32	0.0432432	0.0000000	3.8
4	0	0.0000000	NA	NA

We solve for  $R_0$  in the following:

```
r0 <- sum(lifetable$l_x*lifetable$b_x, na.rm= TRUE)
r0
```

```
## [1] 0.5001351
```

Thus  $R_0 = .5$

We solve for G as

```
(with(lifetable, sum(age*l_x*b_x, na.rm = TRUE))/r0)
```

```
## [1] 2.025939
```

Thus we find the generation time, G, to be about 2.03.

---

## Problem 6

In this problem we are asked to use the life table from problem 5 to create a leslie matrix and project the number in each age class for the next two years. First, we provide the completed life table below:

```
library(knitr)
age <- 0:4
```

```

sx <- c(740,280,105,32,0)
bx <- c(0,0.4,1.3,3.8,NA)
lx <- sx/740
lxbx <- lx*bx
agelxbx <- age*lx*bx

px <- vector(mode = 'numeric',length = 5)
for(i in 1:length(lx)){
  px[i] <- (lx[i+1]/lx[i])
}

pxbx1 <- vector(mode = 'numeric',length = 5)
for(i in 1:length(px)){
  pxbx1[i] <- (px*bx[i+1])
}

## Warning in pxbx1[i] <- (px * bx[i + 1]): number of items to replace is not
## a multiple of replacement length

## Warning in pxbx1[i] <- (px * bx[i + 1]): number of items to replace is not
## a multiple of replacement length

## Warning in pxbx1[i] <- (px * bx[i + 1]): number of items to replace is not
## a multiple of replacement length

## Warning in pxbx1[i] <- (px * bx[i + 1]): number of items to replace is not
## a multiple of replacement length

## Warning in pxbx1[i] <- (px * bx[i + 1]): number of items to replace is not
## a multiple of replacement length

pxbx1

## [1] 0.1513514 0.4918919 1.4378378      NA      NA

lifetable2 <- cbind(age,sx,lx,px,bx,lxbx,agelxbx,pxbx1)
colnames(lifetable2) <- c('Age','S_x','l_x','p_x','b_x', 'lx_bx', 'x_lx_bx', 'px_bx+1')
lifetable2 <- as.data.frame(lifetable2)
kable(lifetable2)

```

Age	S_x	l_x	p_x	b_x	lx_bx	x_lx_bx	px_bx+1
0	740	1.0000000	0.3783784	0.0	0.0000000	0.0000000	0.1513514
1	280	0.3783784	0.3750000	0.4	0.1513514	0.1513514	0.4918919
2	105	0.1418919	0.3047619	1.3	0.1844595	0.3689189	1.4378378
3	32	0.0432432	0.0000000	3.8	0.1643243	0.4929730	NA
4	0	0.0000000	NA	NA	NA	NA	NA

Next, we will construct the leslie matrix using  $m_0$ ,  $m_1$ ,  $m_2$ ,  $m_3$ ,  $p_0$ ,  $p_1$ , and  $p_2$ . Then we will multiply the leslie matrix by the projection matrix two separate times to receive the projected number of individuals at each age class over the next two years.

```

leslie <- matrix(c(0.15136, 0.4875, 1.15824, 0, 0.3784, 0, 0, 0, 0, 0.3750, 0, 0, 0, 0, 0.3048, 0), nrow = 5, ncol = 16)
leslie

##           [,1]  [,2]    [,3] [,4]

```

```
## [1,] 0.15136 0.4875 1.15824 0
## [2,] 0.37840 0.0000 0.00000 0
## [3,] 0.00000 0.3750 0.00000 0
## [4,] 0.00000 0.0000 0.30480 0
```

```
leslie <- as.data.frame(leslie)
kable(leslie)
```

V1	V2	V3	V4
0.15136	0.4875	1.15824	0
0.37840	0.0000	0.00000	0
0.00000	0.3750	0.00000	0
0.00000	0.0000	0.30480	0

```
leslie
```

```
##      V1      V2      V3 V4
## 1 0.15136 0.4875 1.15824 0
## 2 0.37840 0.0000 0.00000 0
## 3 0.00000 0.3750 0.00000 0
## 4 0.00000 0.0000 0.30480 0
```

```
projection1 <- matrix(c(70, 40, 20, 10), nrow = 4, byrow = T)
projection1
```

```
##      [,1]
## [1,]   70
## [2,]   40
## [3,]   20
## [4,]   10
```

*# HELP ME MULTIPLY MATRICES IN R UGH- basically the leslie matrix needs to be multiplied by the project*

## Problem 7

The following is the setup for the population project matrix:

```
stages <- c('seedling','juvenile','small adult', 'medium adult', 'large adult')
A <- matrix(c(
  0, 0,.015,.063,.189,
  .294,.25, 0 , 0 , 0 ,
  .176,.35,.385,.059,.012,
  0,.35,.461,.568,.171,
  0, 0 ,.108,.356,.805
), nrow = 5, byrow = TRUE,
  dimnames = list(stages,stages)
)
```

## Population Projection matrix

```
library(knitr)
kable(A)
```

	seedling	juvenile	small adult	medium adult	large adult
seedling	0.000	0.00	0.015	0.063	0.189
juvenile	0.294	0.25	0.000	0.000	0.000
small adult	0.176	0.35	0.385	0.059	0.012
medium adult	0.000	0.35	0.461	0.568	0.171
large adult	0.000	0.00	0.108	0.356	0.805

### Part a

To calculate  $\lambda$  from the popbio package we do the following:

```
library(popbio)
lambda(A)
```

```
## [1] 1.037396
```

Thus  $\lambda = 1.04$ .

### Part b

To calculate the stable stage distribution we do the following:

```
stable.stage(A)
```

```
##      seedling      juvenile  small adult medium adult  large adult
## 0.10633963 0.03970536 0.08489524 0.28816968 0.48089009
```

### Part c

Calculating sensitivity table as such:

```
sense <- sensitivity(A)
kable(sense)
```

	seedling	juvenile	small adult	medium adult	large adult
seedling	0.0429644	0.0160422	0.0343003	0.1164293	0.1942942
juvenile	0.0926033	0.0345765	0.0739290	0.2509457	0.4187716
small adult	0.0985556	0.0367989	0.0786809	0.2670757	0.4456890
medium adult	0.1097745	0.0409879	0.0876374	0.2974777	0.4964231
large adult	0.1208039	0.0451061	0.0964426	0.3273663	0.5463005

Calculating elasticity table

```
elast <- elasticity(A)
kable(elast)
```

	seedling	juvenile	small adult	medium adult	large adult
seedling	0.0000000	0.0000000	0.0004960	0.0070706	0.0353979
juvenile	0.0262440	0.0083325	0.0000000	0.0000000	0.0000000
small adult	0.0167205	0.0124153	0.0292002	0.0151894	0.0051555
medium adult	0.0000000	0.0138286	0.0389445	0.1628763	0.0818283
large adult	0.0000000	0.0000000	0.0100403	0.1123413	0.4239189

The largest elasticity value is the probability of staying a large adult tree, with a value of .42.

## Part d

Cutting the value of adult tree fecundities by half:

```
B <- A
.189/2 -> B[1,5]
.063/2 -> B[1,4]
.015/2 -> B[1,3]
lambda(B)
```

```
## [1] 1.013429
```

Thus the value of  $\lambda$  is now 1.013 and the stable-age distribution is:

```
stable.stage(B)
```

```
##      seedling      juvenile  small adult medium adult  large adult
## 0.06086339 0.02343876 0.06868347 0.29963870 0.54737569
```

When the fecundities for adult trees were cut by half  $\lambda$  decreased by .024. The stable-age distribution was higher for medium and adult trees when fecundities of all adult trees was reduced suggesting a forest with stable ages skewed towards older trees.

## Part e

Cutting the probabilities that trees stay in the same class by half:

```
C <- A
.385/2 -> C[3,3]
.568/2 -> C[4,4]
.805/2 -> C[5,5]
lambda(C)
```

```
## [1] 0.7464096
```

Thus the value of  $\lambda$  is reduced quite a bit more than the previous case with a value of .746. The stable-age distribution is:

```
stable.stage(C)
```

```
##      seedling      juvenile  small adult medium adult  large adult
## 0.12167932 0.07206492 0.12567264 0.31502130 0.36556182
```

Now the stable age distribution has now increased in all classes but large adult trees. This suggests that large adult trees have a large role in dictating the age distribution within the population.



## Problem 8

Running the stochastic simulation 50 times with 50 steps,  $N_0 = 10$ ,  $b=.55$  and  $d=.5$  yields the following vector.

```
sim10runs <- vector(mode = 'numeric',length = 50)

for(i in 1:50){
  sim10runs[i] <- StochasticSim(steps = 50,b=.55,d=.5, N0=10)
}

propextinction10 <- length(which(sim10runs == 0))/length(sim10runs)
paste('Extinction rate is:',propextinction10)

## [1] "Extinction rate is: 0.06"
```

```
sim10runs

## [1] 8 16 10 0 14 4 12 8 10 28 12 10 12 10 22 4 0 14 20 8 14 12 10
## [24] 12 14 10 16 0 6 10 4 16 12 6 22 22 14 12 8 12 12 14 8 14 14 8
## [47] 16 10 22 12
```

Running the same stochastic simulation, but with  $N_0 = 50$  yields

```
sim50runs <- vector(mode = 'numeric',length = 50)

for(i in 1:50){
  sim50runs[i] <- StochasticSim(steps = 50,b=.55,d=.5, N0=50)
}

propextinction50 <- length(which(sim50runs == 0))/length(sim50runs)
paste('Extinction rate is:',propextinction50)

## [1] "Extinction rate is: 0"
```

```
sim50runs

## [1] 56 44 46 42 48 52 42 50 54 48 46 62 48 50 44 52 48 52 50 46 46 60 48
## [24] 64 54 50 50 46 56 42 54 46 46 56 60 50 56 50 54 54 52 56 52 56 50 56
## [47] 40 52 50 64
```

## Problem 9

In problem number 9 we are asked to calculate the average weight of trees in Plot B and the total weight of plants in both Plots A and B. We are given the following values: Plot A: 16 seedlings/ $m^2$ , 1 tree = 24kg, 1 tree/ $m^2$ , Plot B: 4 seedlings/ $m^2$  and 0.5 trees/ $m^2$ . We will use the equation  $w = cN^k$  to first calculate  $c$  using the values from Plot A and then use the value of  $c$  to calculate the value of  $w$  in Plot B.

```
c <- 24/1^(-3/2)
c

## [1] 24

w <- 24*(0.5^(-3/2))
w
```

```
## [1] 67.88225
```

The average tree in Plot B weighs 67.88225 kg

The total weight of plants in each plot is 24 kg.