Practical: Matrix Models

Life-History 1: smurfs

Draw an age-based life-cycle of "Smurfs" based on the description below:

- The population was censured immediately before breeding.
- Smurfs live for 4 years; no individual has ever been recorded to reach the age of 5.
- Smurfs cannot remain at the same age more than one year.
- Smurfs become reproductively mature (papa smurf) at the age of 4
- This is a male-only model, we ignore the smurfette.
- **Hint**: write the age of each age-class in the appropriate square/circle

Life-History 2: great tits

Draw the great tit life-cycle according to the description below

- The population was censused immediately before breeding.
- Great tits can live up to 9 years of age (and perhaps even older).
- Data is sparse when great tits become older than 5; pool these age-classes together.
- Great tits are able to reproduce after their first year.

Life-History 3: teasel

Draw a stage-based life-cycle of teasel based on the description below (reproduced from Werner 1977 *Ecology* **58**: 840-849):

- Teasel is commonly classified as a biennial, producing a low vegetative rosette up to 60cm in diameter, which overwinters and is followed in a succeeding growing season by a stout flowering stem 0.5m to 2.5m in height.
- In actuality, the year of the flowering is not a function of the plant's chronological age, but rather depends upon the attainment of a minimum rosette size (~0.3m diameter), which may require several years of growth.
- An individual plant dies after flowering; there is no vegetative reproduction and all stages of the life-cycle are easily recognizable.
- A cohort of teasel seeds spreads its germination over more than one growing season.

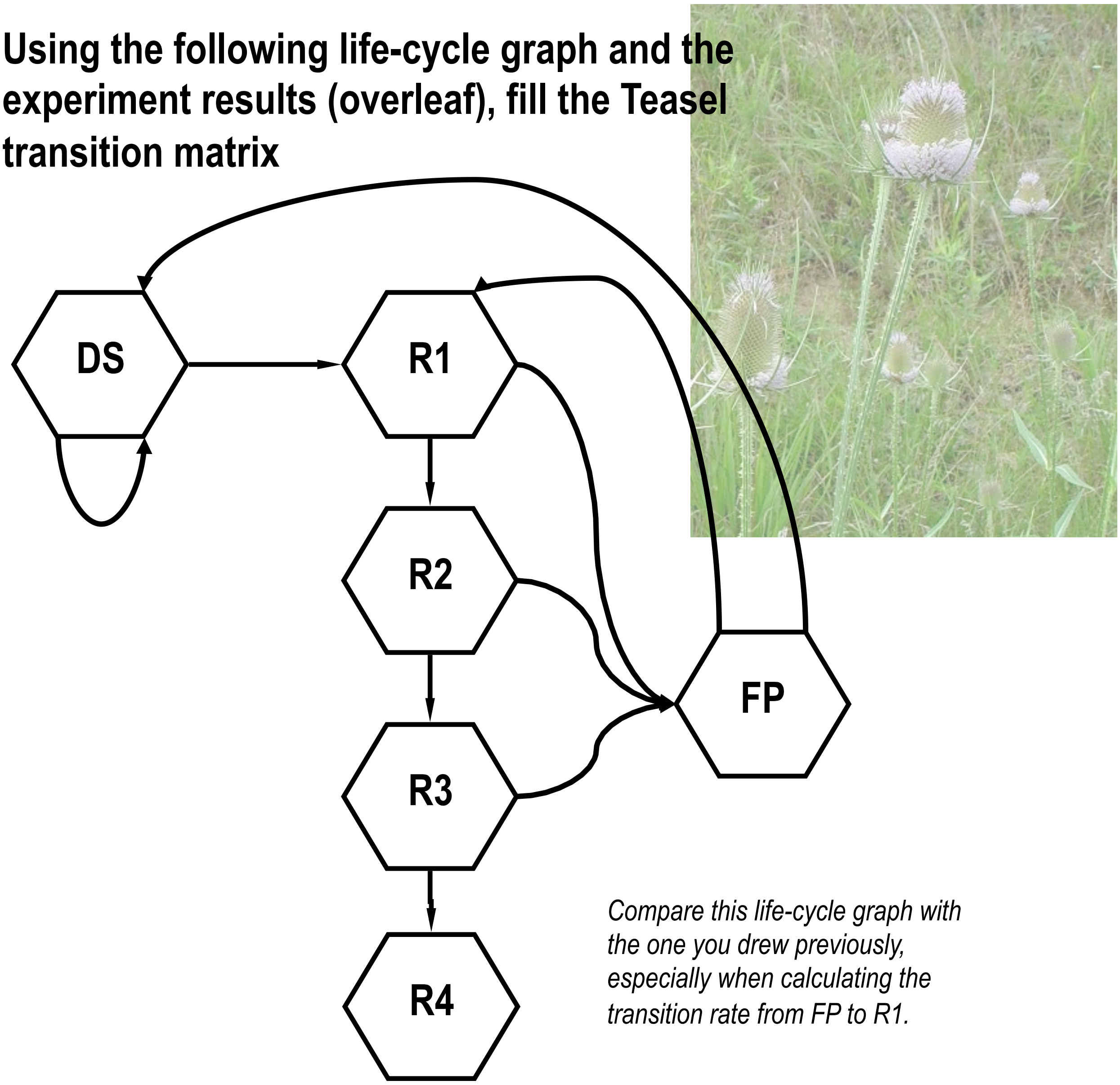
	Yr 1	Yr 2
Age 1	90	160
Age 2	60	27
Age 3	100	24
Age 4	80	90

Using the data from two years (above), estimate the matrix elements.

Using your parameterized matrix and the population structure of each village in year 1 (right), estimate the population structure in year 2, and estimate the population growth rate from year 1 to year 2.

	Village A		Village B		Village C		Village D	
Year	1	2	1	2	1	2	1	2
Age 1	100		0		0		100	
Age 2	100		100		0		0	
Age 3	0		100		100		0	
Age 4	0		0		100		100	
Sum								
$\frac{N_2}{N_1}$								

Using the following life-cycle graph and the experiment results (overleaf), fill the Teasel transition matrix



Compare this life-cycle graph with the one you drew previously, especially when calculating the transition rate from FP to R1.

Experiment Results:

1000 teasel seeds were introduced into a field, which did not have any teasels before. Each individual was followed for each of 5 subsequent years.

The table shows the fate of the seeds. They are classified according to different developmental patterns as described in the left-hand side of the table (DS= dormant seeds; R=rosettes; F=flowering stalk; x=death).

Developmental pattern					Number of individuals from cohort from 1000 seeds
Census/Year					
1	2	3	4	5	
Rx					222
R	Rx				8
R	R	Rx			2
R	R	R	Rx		1
R	R	R	R	Rx	2
R	F				1
R	R	F			12
R	R	R	F		4
R	R	R	R	F	0
DS	Rx				21
DS	R	Rx			2
DS	R	R	Rx		1
DS	R	R	R	Rx	2
DS	R	F			0
DS	R	R	F		1
DS	R	R	R	F	2

Reading the table.

The 1st row of the table tells us that 222 seeds, which germinated before census 1, died before census 2.

The 7th row tells us that 12 seeds germinated before census 1, remained as rosettes at census 2 and were classified as a flowering stalk at census 3.

Obtaining transition rates from the table.

Out of 1000 seeds, 222+8+2+1+2+1+12+4+0 = 252 germinated in their first year (before census 1). The transition rate for seed to rosette year 1 (S > R) is 252/1000 = 0.252. *How many seeds remain dormant? What is the transition rate from seeds to dormant seeds?*

Consider the transition rate of rosette year 1 to flowering stalk. 252 germinated before census 1; 29 (21+2+1+2+0+1+2) germinated from DS at census 2 as well though. Therefore, the total number of rosette year 1 is 252+29 = 281. Of these 281, 1 (1+0) is a flowering stalk at the next census. R1 > F is therefore (0+1)/(252+29) = 1/281 = 0.004.

As a final example, consider the transition rate from rosette year 1 to rosette year two. As above, there are 281 rosette year 1 individuals. Of these, (8+2+1+2+12+4) + (2+1+2+1+2)=37 are rosette year 2 at the subsequent census. The rate is therefore 37/281=0.132

All required information to fill the matrix is contained within the table, except that each flowering plant produces 431 seeds. *What is the transition rate from flowering plants to rosette year 1? Why is 431 important?*

Computer analysis:

R may seem intimidating, but it is freeware (means if you change institutions you don't need to learn a new program like SAS, MATLAB, etc…), updated incredibly regularly by the finest academic minds working at the cutting edge (it reaches theory that commercial products do not) and has an extensive online support network (who report bugs, etc. faster than elsewhere). It really is worth spending some time getting to grips with it. **Some tips**:

> R is case-senstive, i.e. R is not r.
> If you type the up arrow, you get the last line you typed (this is especially useful for correcting typos)
> Enter commands in a text editor (R has one, or use notepad, etc..) and then cut and paste everything in.
> Use # to comment out descriptions of your code, so you can remember what each bit does.

Questions to be answered or things to do are in italics, `with lines to enter into R given in Courier New.`

Entering the matrix in R

Define a matrix filled with zeros and 6 rows and 6 columns using

```
M <- matrix(0, nrow=6, ncol=6)
fix(M)
```

This opens a window where you can change 0s as appropriate, ending up with your transition matrix. Once all the matrix elements are correct, close the window and look at the matrix

```
M
```

Calculating population growth

What quantity is the asymptotic population growth rate (λ)? This can be obtained using the function eigen():

```
eigen(M)
```

This opens a list containing eigenvectors and eigenvalues. *What is the eventual fate of this population? Why? Write down λ. Store this information in* `eigM` *by typing*

```
eigM <- eigen(M)
```

There are many eigenvectors; each is associated with an eigenvector. *Which is the stable-(st)age distribution?*

Calculating the stable-age distribution

Store the stable-(st)age distribution as `sadM` *by typing*

```
sadM <- eigM$vectors[,1]
sadM
```

The square brackets are used to pick out particular parts of a matrix. So `M[1,1]` is the element of M in the first row and first column. `M[6,3]` is the element of M in the 6th row and 3rd column. Leaving nothing before or after the comma selects all elements in that column or vector, respectively. *Why is the number after the comma 1 to calculate the stable-(st)age distribution?* In a vector (a matrix that has only one column), only one reference point is required.

The stable-(st)age structure is scaled as the proportion of individuals in each (st)age. This can be obtained by simply dividing `sadM` by the total number of individuals, I.e. `sum(sadM)`:

```
sadM <- sadM/sum(sadM)
sadM
```

Note that this overwrites the previous `sadM`. *Write down the stable-(st)age structure. Which (st)age class eventually contains most individuals? You may see* `0i` *; this is the imaginary part of the number; we have no need for them here! If they're annoying, retain just the real parts using* `Re()`

Calculating the reproductive value

The stable-(st)age distribution is the right eigenvector associated with the dominant eigenvalue; reproductive value is the left eigenvector associated with the dominant eigenvalue. To calculate this, take the eigenvector of the `t`ranspose of `M`.

```
rvM <- eigen(t(M))$vectors[,1]
```

What is a transpose? Have a look. The reproductive value is scaled to the size of the first element:

```
rvM <- rvM/ rvM[1]
rvM
```

Write down the reproductive value of each stage. Which (st)age class eventually has the highest reproductive value? Why is that not a surprise?

Calculating sensitivities and elasticities

Copy and paste the function eigen.analysis in R:

```
eigen.analysis <-
function(A){
 ev <- eigen(A)
 lmax <- which(Re(ev$values)==max(Re(ev$values)))
 lambda <- Re(ev$values[lmax])
 W <- ev$vectors
 w <- abs(Re(W[,lmax]))
 V <- Conj(solve(W))
 v <- abs(Re(V[lmax,]))
 s <- v%o%w
 s[A == 0] <- 0
 class(s) <- "leslie.matrix"
 e <- s*A/lambda
 rho <- lambda/abs(Re(ev$values[2]))
eigen.analysis <- list(lambda1=lambda,
rho=rho,sensitivities=s,elasticities=e, stable.age=w/
sum(w),repro.value=v/v[1])
eigen.analysis
}
```

And, execute the function on the matrix M:

```
eigen.analysis(M)
```

Which demographic rate has the highest sensitivity? Which demographic rate has the highest elasticity? Why are these different? What other information (that you've already calculated) can be gleaned using this function?

Population Dynamics

We want to initialize a population with 1000 seeds. *Which (st)age is seeds? Initialize a vector called v0 with as many stages as given by the life-cycle graph.*

Matrix (dot) multiplication is achieved using %*% in R. If `v0` is the initial vector and `M` the transition matrix, `v1` can be calculated and viewed as follows:

```
v1 <- M %*% v0
v1
```

The one-step population growth rate is: `sum(v1)/sum(v0)`
Is this the same as λ? Adapt these lines to calculate the population vector at time 2 and the one-step growth rate between time 2 and time 1.

We want to project the population forward across multiple time steps. The easiest way is using a loop. Begin by initializing where data will be stored. *You've already got v0 above, so define a vector to receive the population size at each time step and the number of time steps*:

```
nTime <- 20                #the number of time steps
popSize <- numeric(nTime)#to receive each population size
pV <- v0                   #the initial population vector
```

A `for` loop repeats commands contained within `{ }` a specified number of times (`nTime`).

```
for (i in 1:nTime)
          {
          popSize[i] <- sum(pV)  #store total popSize
          pV1 <- M %*% pV        #calculate the new pV
          pV <- pV1              #overwrite old pV with new pV
          }
```

Plot the population trajectory: `plot(popSize,type=‘b’)`
Is the population increasing exponentially? `plot(log(popSize),type=‘b’)`

Adapt the above to store the population structure at each time step and plot the population structure:

```
popStruct <- matrix(0,nrow=dim(M)[1],ncol=nTime)
pV <- v0
for (i in 1:nTime)
          {
          popStruct[,i] <- pV/sum(pV)      #store popStruct
          pV1 <- M %*% pV
          pV <- pV1           }
barplot(popStruct)
```

Has the stable-age distribution converged? If not, change `nTime` *and try again!*

The projections above were initialized with 1000 seeds. What happens if the population was instead initialized by one invasive plant? *Write down v0 for this scenario, and use your loops to project the population forward. Does λ change? Is the stable-age distribution reached in more or fewer generations than in the previous example? Why?*