Age structure

Age structured population dynamics

When we introduced variation in demographic rates, we did so because we wanted to recognize that individuals may differ in their demographic rates as a function of time (**temporal variation** or **environmental variation**) or simply through **demographic variability**. But let's consider a population of individuals that reproduce sexually. It takes time before newborn individuals are sexually mature, limiting reproduction rate by age class. Also, mortality varies as a function of individual age, or through grouping individuals into age classes (e.g., 1-5 year olds). Acknowledging that different age classes have fundamentally different contributions to mortality and reproduction is important, as the ability to accurately predict population dynamics may require this knowledge for many species. Think of game species management, for example. The ability to predict deer populations in the next year necessitates information not only on the current number of deer, but an approximation of their age structure, as the manager must assume something about mortality and population growth rates of the population. A miscalculation here could result in important changes to the ecosystem.

Defining some terms

Originally designed for insurance companies, life tables are a way to track demographic rates through time, partitioning things by the age of the organism.

Table 1: Life table

age (x)	s_x	l_x	b_x
0	1	1	0
1	0.5	0.5	0.5
2	0.25	0.4	0.25
3	0.8	0.1	1
4	0.2	0.08	2

The term s_x refers to the survival rate of each individual age class. That is, s_2 would be the probability of an individual surviving to reach age 3. We can define $s(x)$ for each age class x by considering the value to be $1 - d(x)$, where $d(x)$ is the death rate of age class x.

The term $l(x)$ represents the probability of surviving from age 0 to age $x + 1$. To understand how survivorship of a single age class corresponds to the ability of a newborn to survive through multiple age classes, we have to consider the product of our survival rates (s_0, s_1, $\ldots s_x$). That is, think of survivorship as a probability of surviving to that age class x. To combine probabilities, we multiply them. For instance, the probability of a newborn (age class 0 in the table above) surviving is 1, meaning all newborns survive. However, if we wanted to calculate the probability of an individual surviving until they were 3 years old in the above life table, we would have to consider each probability separately.

$$l_x = s_0 * s_1 * s_2 ... * s_{x-1}$$

or

$$l_x = \Pi_{i=1}^{x-1} s_i$$

Table 2: A Leslie matrix – also called a *population projection matrix* describing the survival and fecundity relationships between age classes 1,2, and 3.

	1	2	3
1	F_1	F_2	F_3
2	S_1	0	0
3	0	S_2	0

So for the above example of an individual surviving until 3 years old, we would multiply $1 * 0.8 * 0.4$, for a survivorship probability of $l_3 = 0.32$.

The term $b(x)$ represents the per-capita birth rate for females of age x. This is the number of offspring generated from one individual of age x. All offspring are newborns, so that all contributions of b_x go into the newborn age class n_0, which Case goes into how we don't commonly model this age group, because we rarely care about newborns, treating them as being some collective sum of reproduction from other age classes multiplied by some constant mortality term s_0.

This gives us a good idea of how age or life stage can influence reproductive output and survival.

Population growth rate in age structured populations

Elements of the square matrix correspond to the production of (row) by (column). These are transitions between lifestages. This matrix is called a *Leslie matrix*.

Here, we have fecundity F_i, and stage transition rates (S_i). It is important to note that fecundity is different from birth rates discussed earlier. Here, fecundity captures both survival *and* birth rate.

So we can use the transition matrix to simulate age or stage-structured population dynamics. How we do this is by using matrix multiplication, as follows. We have a 1 column matrix containing the initial population sizes for all life stages.

$$\mathbf{n}^0 = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}$$

We can simply multiply this one column matrix by the transition matrix,

$$\begin{bmatrix} n_{1,t+1} \\ n_{2,t+1} \\ n_{3,t+1} \end{bmatrix} = \mathbf{N} * \mathbf{L} = \begin{bmatrix} n_{1,t} & n_{2,t} & n_{3,t} \end{bmatrix} * \begin{bmatrix} F_1 & F_2 & F_3 \\ S_1 & 0 & 0 \\ 0 & S_2 & 0 \end{bmatrix}$$

where $\mathbf{N}$ is the population size matrix and $\mathbf{L}$ is the transition matrix in Table 2, to yield the resulting population size at time $t + 1$.

$$\begin{bmatrix} n_1 * F_1 & + & n_2 * F_2 & + & n_3 * F_3 \\ n_1 * S_1 & + & n_2 * 0 & + & n_3 * 0 \\ n_1 * 0 & + & n_2 * S_2 & + & n_3 * S_3 \end{bmatrix} \tag{1}$$

Example: This tracks a three-stage population, in which individuals go through juvenile (J), teen (T), and adult (A) stages.

Here, juveniles transition to teenagers with rate 0.2, and teenagers to adults at rate 0.3. Meanwhile, all stages contribute to the population of juveniles at different per capita rates, with juveniles producing 0.3 new juveniles for every 1 juvenile at time t, teenagers producing 0.5 juveniles for every teenager at time t, and adults producing 2 juveniles for every adult at time t.

Let's simulate the model 1 timestep forward, beginning with the abundance matrix

Table 3:

	J	T	A
J	0.3	0.5	2.0
T	0.2	0	0
A	0	0.3	0

$$\mathbf{N}_t = \begin{bmatrix} n_{J,t} = 20 & n_{T,t} = 10 & n_{A,t} = 0 \end{bmatrix}$$

$$\mathbf{N}_t * \mathbf{L} = \begin{bmatrix} (20 * 0.3) \ + \ (10 * 0.5) \ + \ (0 * 2) \\ (20 * 0.2) \ + \ (10 * 0) \ + \ (0 * 0) \\ (20 * 0) \ + \ (10 * 0.3) \ + \ (0 * 0) \end{bmatrix} = \begin{bmatrix} n_{J,t+1} = 11 \\ n_{T,t+1} = 4 \\ n_{A,t+1} = 3 \end{bmatrix} \tag{2}$$

It is important to consider the t in the equations above (and introduced previously). We will largely treat time as discrete, so we can consider population growth at some biologically realistic timescale (e.g., a year). Case spends some time discussing the importance of considering species biology, birth pulses, etc. when modeling age or stage structured populations.

Estimating age/stage structured vital rates

All of the models above assume that we have knowledge of structured demographic rates (e.g., $s(x)$). These are often difficult parameters to estimate. There are a number of ways that researchers go about estimating them, with Case largely focusing on *segment-based* estimation and *cohort analysis*. The difference between them is that *segment-based* approach surveys a population at a given time to estimate demographic rates, assuming that researchers can reliably estimate individual age or stage. The *cohort analysis* follows a cohort from birth to estimate demographic rates as individuals age, allowing the estimation of time to maturity, and individual variation in demographic rates that may covary through time (i.e., individuals that mature quickly may produce more/less offspring more/less often).

Age versus stage structure

Up to this point, we've treated age and stage structure as sort of the same thing, while mainly providing examples of age structured populations. The underlying idea here is that stage structure can provide a way to simplify the model, especially if we can assume that groups of individuals of different ages can still have the same demographic rates (e.g., individuals past a certain age will contribute very little to fecundity). I will continue to use these two together. You may hear folks discussing size-structured populations, where age or stage is assumed to vary with individual body size, or at a minimum we assume that individual size drives resulting demographic rates (e.g., large individuals have more offspring). This is *generally* a bad idea, as we assume that size can be a proxy for age or stage, which is perhaps an untenable assumption.

Stable age distributions and some eigenvalue fun

The earlier models we introduced without age/stage structure had easily determined equilibrium values. But the introduction of age structure changes the transient dynamics of the system. That is, there is an distribution of individuals at different age classes that results in stable dynamics, but we may not know this information and it is very unlikely that we will start with the stable age distribution. Knowing this distribution is important though. The dominant eigenvalue (λ) of the Leslie matrix tells us some important information, as we will show that it is the ultimate geometric growth rate of the population. How does it do this? First, we'll start with 'what is an eigenvalue?' An eigenvalue is

To start to explore this, we will consider exponential growth we discussed earlier.

$$N_t = \lambda N_{t+1}$$

or to project to any time T, we can consider

$$N_t = N_0 \lambda^T$$

In terms of age-structured populations, N_t becomes a vector (Case refers to this as a scalar) and λ is the Leslie matrix ($\mathbf{L}$). Note that the notation is such that scalars and matrices are now bolded.

$$\mathbf{n_t} = \mathbf{n_0} \mathbf{L}^T$$

In this case, if we wanted to know the population at time $T = 2$, we would multiply the Leslie matrix L by itself. What we ultimately want to estimate is λ, the maximum achievable population growth rate. That is, can we find single value λ that satisfies the following equation?

$$\mathbf{L} * \mathbf{x} = \lambda * \mathbf{x}$$

Note that on the left, we have the Leslie matrix, and on the right we have λ, which is a vector the length of the number of age classes x. λ is what we care about, and we will estimate it as the dominant eigenvalue of the Leslie matrix. The details of why this works are presented in Case, and we can go into this if folks are interested, but it is a bit advanced. Instead, I will show how to do this in *R*.

```
projMatrix <- matrix(
  c(
    0.2,      1,     0.5,
    0.8,      0,     0,
    0,        0.5,   0
  )
  ,nrow=3,ncol=3,byrow=T
)

abund0 <- matrix(c(20,20,20), ncol=1)
```

Simulate one generation into the future.

```
abund1 <- projMatrix  %*% abund0
```

Simulate one more generation

```
projMatrix %*% abund1
```

```
##      [,1]
## [1,] 27.8
## [2,] 27.2
## [3,]  8.0
```

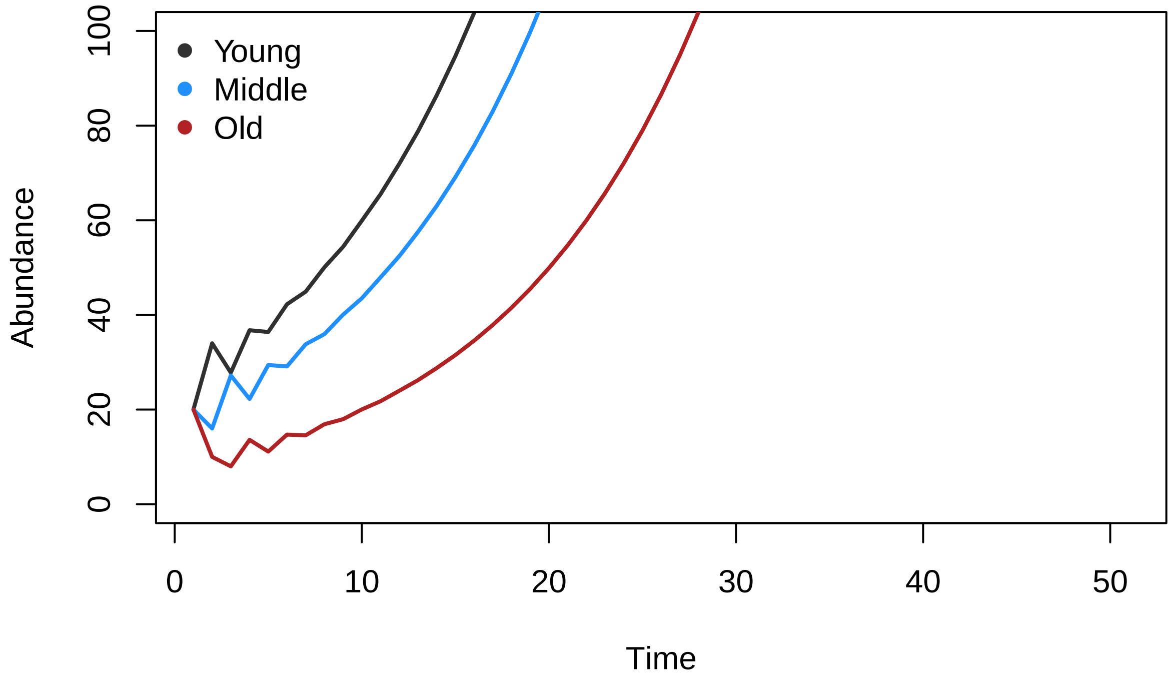
Simulate many generations to examine dynamics

```
getStageDynamics <- function(projMatrix, abund, steps=100){
  ret <- matrix(0, ncol=3, nrow=steps+1)
  ret[1,] <- abund
  for(i in 1:steps){
    ret[i+1, ] <- projMatrix %*% matrix(ret[i,],ncol=1)
  }
  return(ret)
}
```

```
# make some colors for plotting
colz <- c(grey(0.1,0.9), 'dodgerblue', 'firebrick', 'forestgreen')

stageDynamics <- getStageDynamics(projMatrix, abund0, steps=50)

plot(stageDynamics[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,100),
  ylab='Abundance', xlab='Time')
lines(stageDynamics[,2], lwd=2, col=colz[2])
lines(stageDynamics[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
```



What is the age-structured populations ultimate geometric growth rate λ?

In the Case text, there is detail provided on how to calculate eigenvalues. We will not be calculating them by hand, as it gets a bit much once we go past 2x2 matrices, but the underlying idea is important. That is, that

$$N_T = N_0 \lambda^T$$

where we want to know λ. All of the information contained in L is what is needed to determine λ, so let's just place L directly in place of λ.

$$N_T = N_0 L^T$$

But then we notice that we are exponentiating a matrix L by some value, which will still lead to a matrix of values of the same size as L. What we want is to find some scalar (n x 1) set of values that contains all of the information contained in the matrix L, such that

$$x\lambda = xL$$

The way we can approximate λ and the stable age distribution is by taking the eigenvalue of the matrix L (the underlying reasons for this are a bit beyond the class, but check out Case page 64). Here, λ is equal to the dominant eigenvalue of the Leslie matrix L, where dominant eigenvalue means the largest real numbered eigenvalue. Each eigenvalue (of which there are the same as the number of rows or columns in the input matrix L) corresponds to an eigenvector, each of which has the same as the number of rows or columns in L. So in our case, we have a 3x3 Leslie matrix, so we will obtain 3 eigenvalues, and each of these eigenvalues will have corresponding eigenvectors of length 3.

```
eigProj <- eigen(projMatrix)
eigProj
```

```
## eigen() decomposition
## $values
## [1]  1.0962158 -0.5835879 -0.3126279
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.7800647 -0.4845817  0.2028684
## [2,] 0.5692782  0.6642794 -0.5191308
## [3,] 0.2596561 -0.5691339  0.8302696
```

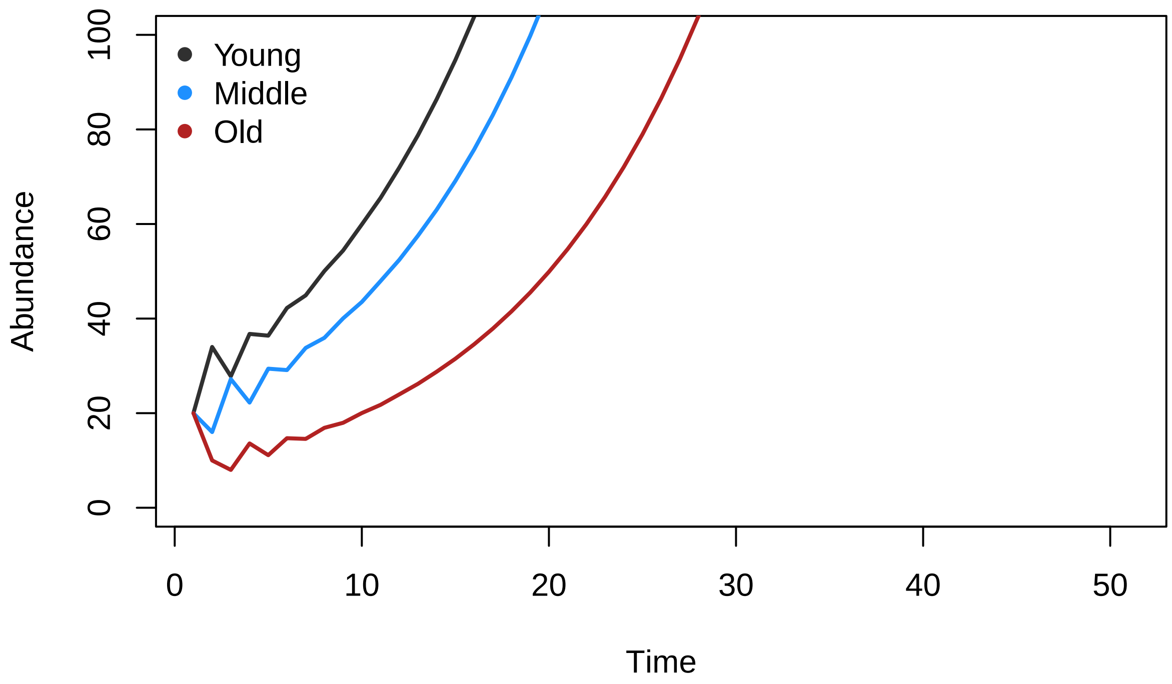
This suggests that for our Leslie matrix (`projMatrix`), the maximum geometric growth rate λ is 1.0962158, as this is the largest real-valued eigenvalue (i.e., the *dominant eigenvalue*). Each eigenvalue has an associated eigenvector. The eigenvector associated with the dominant eigenvalue describes the stable age distribution.

```
eigStable <- eigProj$vectors[,1]
eigStable <- eigStable / sum(eigStable)
```

This suggests that when the different age/stage classes reach abundance values proportional to `eigStable`, that this is the stable age distribution (i.e., the equilibrium for the system). Let's explore this through simulation.

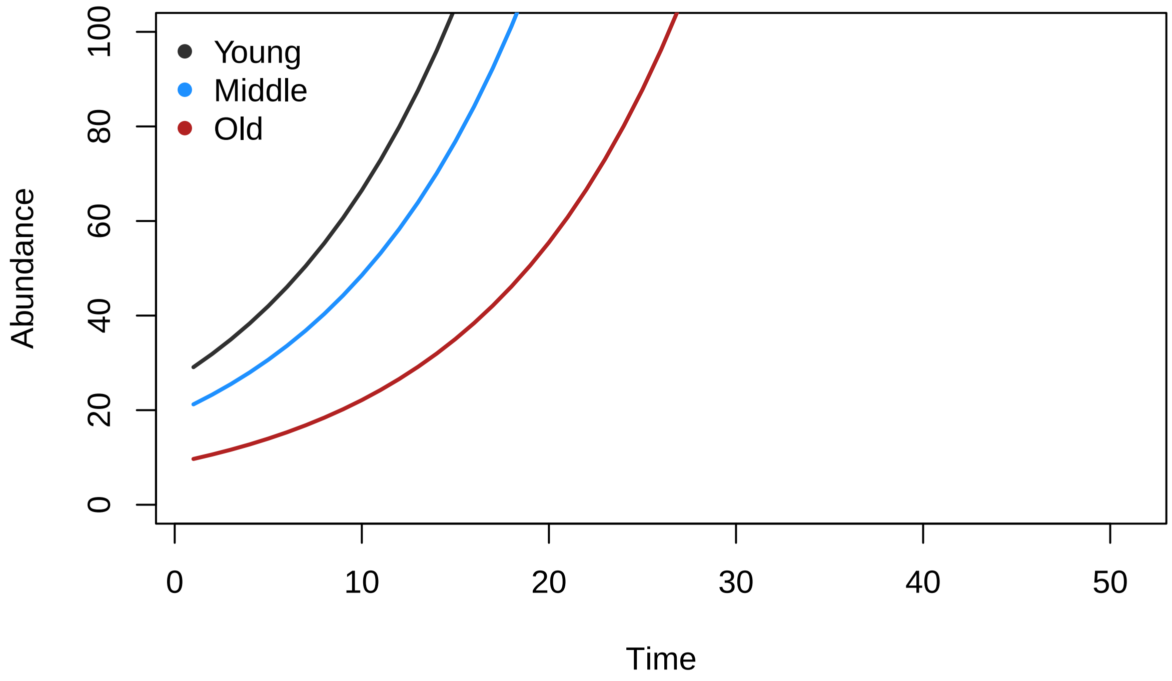
```
stageDynamics <- getStageDynamics(projMatrix, abund0, steps=50)

plot(stageDynamics[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,100),
  ylab='Abundance', xlab='Time')
lines(stageDynamics[,2], lwd=2, col=colz[2])
lines(stageDynamics[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
```



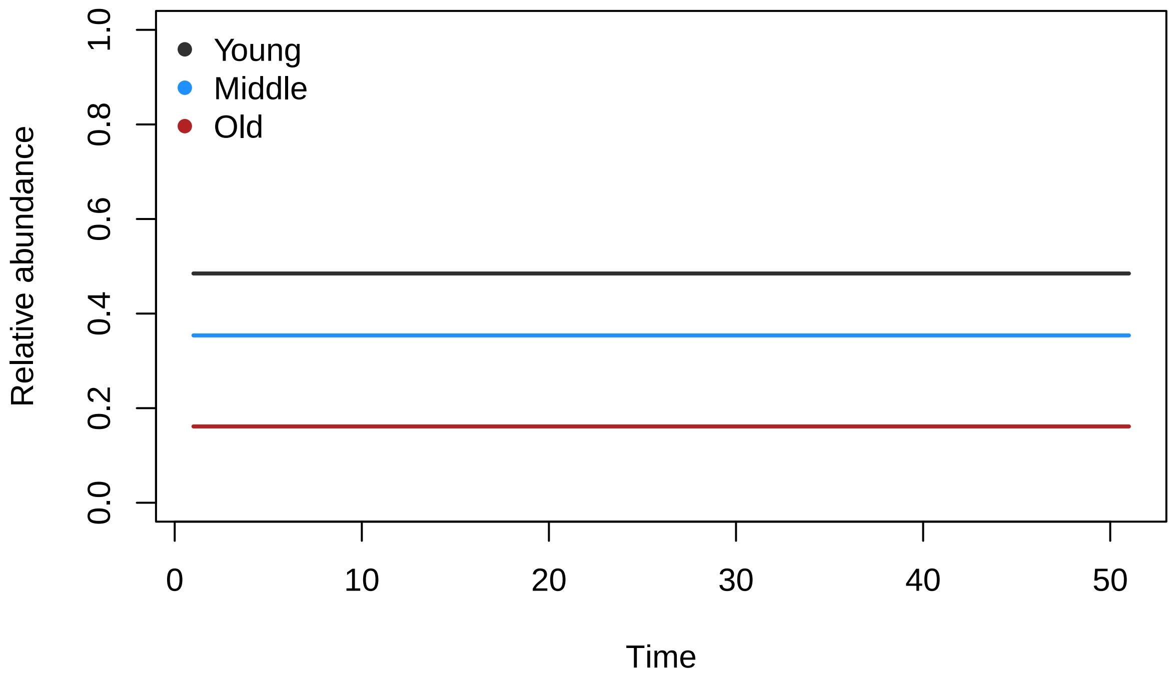
```
stageDynamics <- getStageDynamics(projMatrix, 60*eigStable, steps=50)

plot(stageDynamics[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,100),
  ylab='Abundance', xlab='Time')
lines(stageDynamics[,2], lwd=2, col=colz[2])
lines(stageDynamics[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
```



I honestly thought that would look a bit better, but perhaps it would look better if we looked at the actual age distribution?

```
stageDynamics2 <- stageDynamics / rowSums(stageDynamics)
plot(stageDynamics2[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,1),
  ylab='Relative abundance', xlab='Time')
lines(stageDynamics2[,2], lwd=2, col=colz[2])
lines(stageDynamics2[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
```



So we see here that the relative distribution of individuals across age classes stabilizes very quickly when we start it at the stable age distribution. The system will naturally go towards the stable age distribution, which is why we observed similar dynamics when initiating the system with 20 individuals in each age/stage class.

So we have calculated λ for the population and determined the stable age structure. In the case we knew λ beforehand, we could estimate the stable age distribution given information on birth and survivorship.

$$c(x) = bl_x exp(-rx)$$

To use this approach, we need to know the gross birth rate (b), survivorship l_x of each stage, and the instanteous rate of increse (r), where $r = log(\lambda)$. This starts to become a bit of a side point, as so much of the focus of estimating the Leslie matrix L is to get at λ.

Sensitivity of each age class

Recall how we determined stability of equilibria in the geometric growth model. There was a single equilibrium value $N^* = 0$. We determined the stability of this equilibrium value by perturbing it away from that point (by adding some number of individuals) and exploring where the system went (either back to $N = 0$ or away to $N =$infinity). We can do similar things with age-structured models, by characterizing the Leslie matrix. There are at least two things we could care about with respect to the importance of individual life stages/ages.

Sensitivity: what effect does an absolute change in a vital rate have on λ? **Elasticity**: what effect does a proportional change in a vital rate have on λ?

Sensitivity is defined as the partial derivative of λ with respect to L_{ij}, and elasticity is defined as the sensitivity weighted by the L_{ij}/λ.

The result of both of these is a matrix of the same dimensions as the Leslie matrix L, with larger values corresponding to a larger effect of that rate on overall dynamics. Specifically, sensitivity values would indicate the effect of an additive change in some Leslie matrix entry L_{ij}, corresponding to the stable age distribution scaled by the reproductive value of that age class (i.e., what happens if an entry of the Leslie matrix is increased by some small constant amount?). Elasticities address a similar question, but ask about proportional increase (i.e., what happens if an entry of the Leslie matrix is increased by some small percentage?).

```
sens <- demogR::eigen.analysis(projMatrix)

sens$sensitivities
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5077744 0.3705653 0.1690202
## [2,] 0.5688442 0.0000000 0.0000000
## [3,] 0.0000000 0.1690202 0.0000000
## attr(,"class")
## [1] "leslie.matrix"
```

```
sens$elasticities
```

```
##            [,1]       [,2]       [,3]
## [1,] 0.09264132 0.33804045 0.07709259
## [2,] 0.41513305 0.00000000 0.00000000
## [3,] 0.00000000 0.07709259 0.00000000
## attr(,"class")
## [1] "leslie.matrix"
```

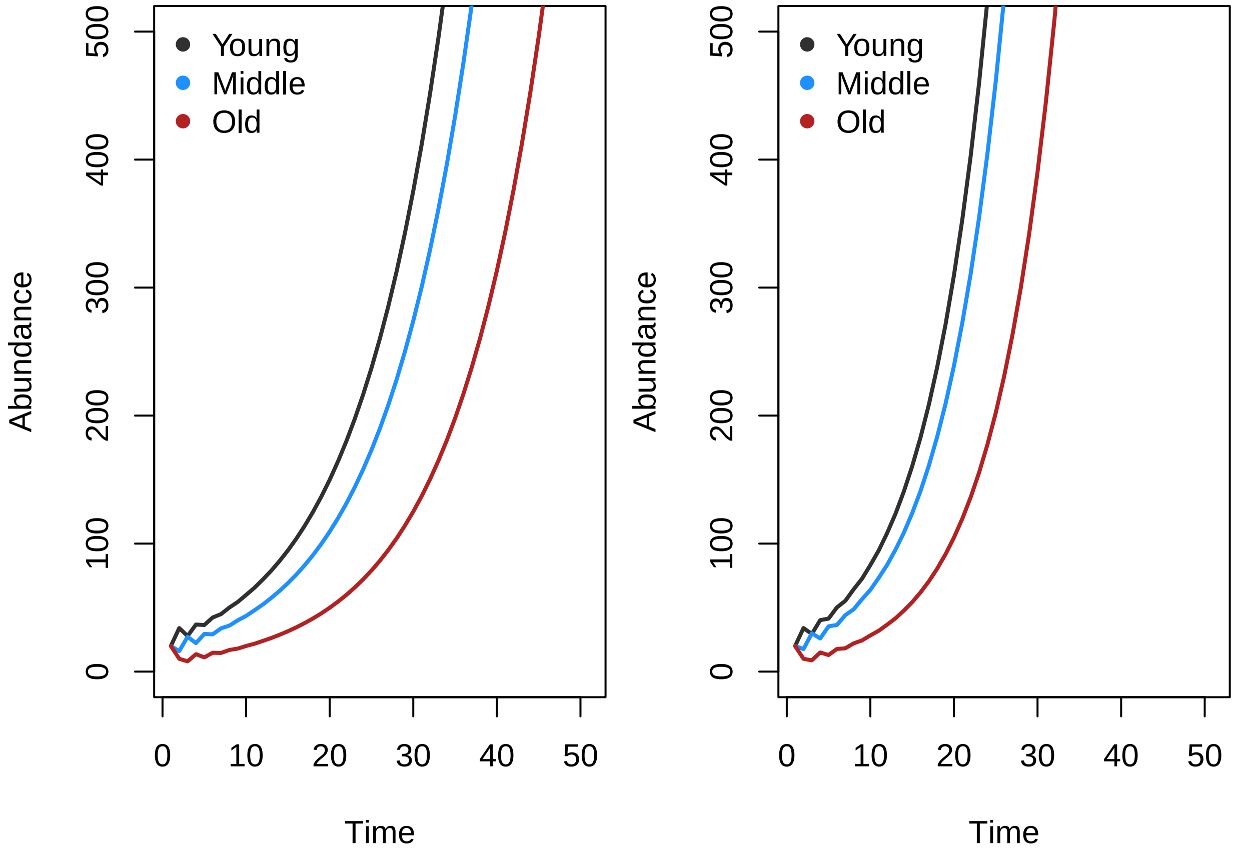
Given these values, which age classes and/or demographic rates would we want to target to have the largest effect on overall λ?

```
projMatrix2 <- projMatrix
projMatrix2[2,1] <- projMatrix2[2,1] * 1.1
stageDynamics <- getStageDynamics(projMatrix, abund0, steps=50)
stageDynamicsP <- getStageDynamics(projMatrix2, abund0, steps=50)

layout(matrix(1:2,ncol=2))
par(mar=c(4,4,0.5,0.5))
plot(stageDynamics[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,500),
  ylab='Abundance', xlab='Time')
lines(stageDynamics[,2], lwd=2, col=colz[2])
lines(stageDynamics[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
eigen(projMatrix)
```

```
## eigen() decomposition
## $values
## [1]  1.0962158 -0.5835879 -0.3126279
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.7800647 -0.4845817  0.2028684
## [2,] 0.5692782  0.6642794 -0.5191308
## [3,] 0.2596561 -0.5691339  0.8302696
```

```
plot(stageDynamicsP[,1], type='l', lwd=2,
  col=colz[1], ylim=c(0,500),
  ylab='Abundance', xlab='Time')
lines(stageDynamicsP[,2], lwd=2, col=colz[2])
lines(stageDynamicsP[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
  c('Young', 'Middle', 'Old'), bty='n')
```



```
eigen(projMatrix2)
```

```
## eigen() decomposition
## $values
## [1]  1.1406145 -0.6385642 -0.3020503
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.7648055 -0.4960781  0.1747491
## [2,] 0.5900581  0.6836411 -0.5091180
## [3,] 0.2586580 -0.5352955  0.8427702
```

Alright. Let's work through chapter 3, homework problem 3 together (page 64 of Case)

> Make a life table for the fictional shrewd shrew. Two consecutive breeding censuses produce the following data for it.

Age x	Females (yr 1)	Females (yr 2)	Number female births from age x
1	1000	1110	0
2	600	700	600
3	540	540	1620