

# 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.

## Life table

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

Table 1: Life table

age (x)	$S(x)$	$l(x)$	$b(x)$	$l(x)b(x)$	$l(x)b(x)x$
0	500	1	0	0	0
1	400	0.8	0.5	0.4	0.4
2	200	0.4	0.25	0.3	0.6
3	50	0.2	1	0.2	0.6
4	10	0.1	2	0.2	0.8

The term  $s(x)$  refers to the number of individuals from a particular cohort that are still alive at age  $x$ . 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  $x$  to age  $x + 1$ . This is called survival rate or **survivorship**. 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 these survival rates. That is, think of survivorship as a probability of surviving that age class. 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.

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

or

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

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  $x$ .

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

$R_0$  is the net reproductive rate, which corresponds to the average number of offspring that would be born to each member of a population going from birth to death following to life table.

$$R_0 = 0 + 0.4 + 0.3 + 0.2 + 0.2 = 1.1$$

$G$  is the generation time, which is quantified as the average age of parents of all offspring produced in a single cohort. It can be calculated as

$$G = \frac{\sum_{x=0}^k l(x)b(x)x}{\sum_{x=0}^k l(x)b(x)}$$

$$G = (0.4 + 0.6 + 0.6 + 0.8)/(0.4 + 0.3 + 0.2 + 0.2) = 2.182$$

From  $R_0$  and  $G$ , we can compute  $r$  and  $\lambda$

$$r = \frac{\ln(R_0)}{G} = \frac{\ln(1.1)}{2.182} = 0.0437$$

$$\lambda = e^r = e^{0.0437} = 1.045$$

What is the relationship between  $R_0$  and  $\lambda$ ?

So how do we model these structured populations? We could break the populations down into stages, and use the models described above for each life stage. Here is an example for a stage-structured population consisting of juveniles ( $J$ ), teenagers ( $T$ ), and adults ( $A$ ). Here, we can track the dynamics of each stage independently, as below. What's wrong with this? It doesn't explicitly consider the inherent connections between the different stages. So it tracks population growth, but not the transitions between classes. \

$$J_{t+1} = \lambda_J * J_t$$

$$T_{t+1} = \lambda_T * T_t$$

$$A_{t+1} = \lambda_A * A_t$$

Then,

$$N_t = J_t + T_t + A_t$$

We can account for population flow explicitly by having some survival term which tracks the transition of juveniles to teenagers, and teenagers to adults, but what else does this fail to account for? The contribution of different stage classes is not to its own class, but to the first stage class, right? How do we incorporate this? We could set up a system of equations, or we could use *matrix modeling*, which essentially sets up a system of equations, but in a nice way.

Table 2: A Leslie matrix – also called a *population projection matrix* describing the survival and fecundity relationships between life stages ( $J$ ,  $T$ , and  $A$ )

	J	T	A
J	$F_J$	$F_T$	$F_A$
T	$S_J$	-	-
A	-	$S_T$	$S_A$

### Matrix modeling

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 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.

$$\mathbf{n}^0 = \begin{bmatrix} n_J & n_T & n_A \end{bmatrix}$$

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

$$\begin{bmatrix} n_{J,t+1} \\ n_{T,t+1} \\ n_{A,t+1} \end{bmatrix} = \mathbf{N} * \mathbf{M} = \begin{bmatrix} n_{J,t} & n_{T,t} & n_{A,t} \end{bmatrix} * \begin{bmatrix} F_J & F_T & F_A \\ S_J & 0 & 0 \\ 0 & S_T & S_A \end{bmatrix}$$

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

$$\begin{bmatrix} n_J * F_J + n_T * F_T + n_A * F_A \\ n_J * S_J + n_T * 0 + n_A * 0 \\ n_J * 0 + n_T * S_T + n_A * S_A \end{bmatrix} \quad (1)$$

**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  $t$ .

Table 3:

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

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

$$\mathbf{N}_t = [n_{J,t} = 20 \quad n_{T,t} = 10 \quad n_{A,t} = 0]$$

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

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 (but see the JTA example above). 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 ( $\lambda$ ) 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  $\lambda$  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  $\lambda$ , the maximum achievable population growth rate. That is, can we find single value  $\lambda$  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  $\lambda$ , which is a vector the length of the number of age classes  $x$ .  $\lambda$  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,
    0.6,    0,    0,
    0,      0.3, 0.6
  )
  ,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,] 20.4
## [3,] 14.4
```

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)
}

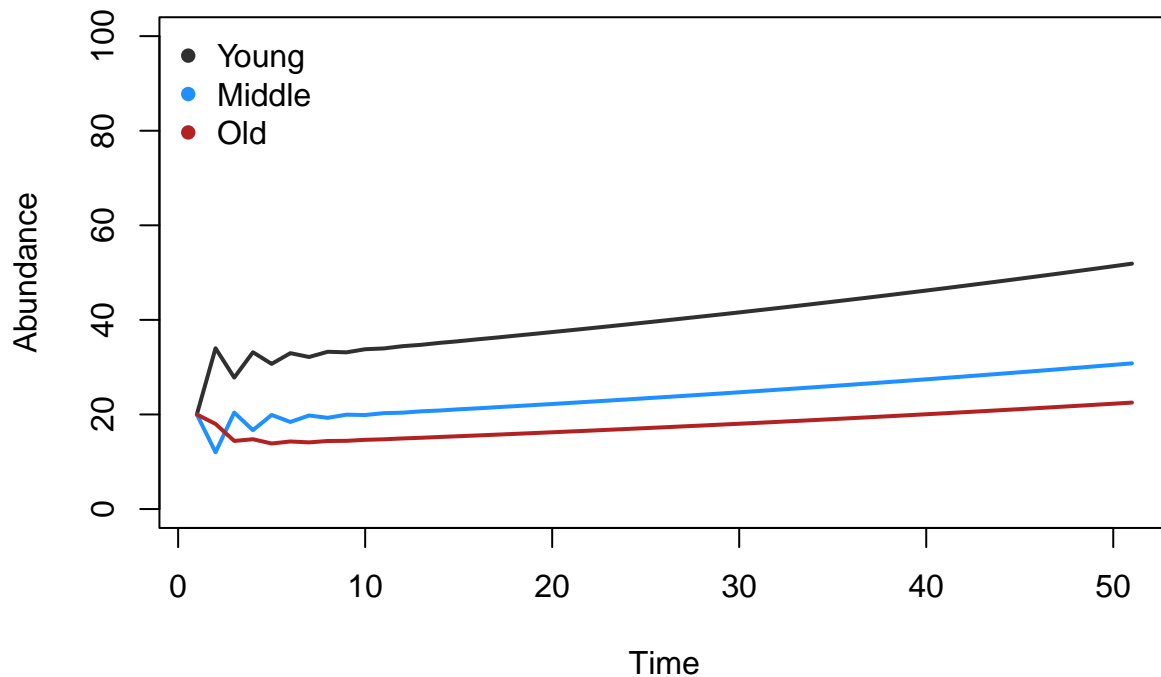
# make some colors for plotting
colz <- c(grey(0.1,0.9), 'dodgerblue', 'firebrick', 'forestgreen')

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  $\lambda$ ?

```

eigProj <- eigen(projMatrix)
eigProj

```

```

## eigen() decomposition
## $values
## [1] 1.0105996 -0.6327997 0.4222001
##
## $vectors
##      [,1]      [,2]      [,3]
## [1,] 0.8056510 0.7157003 -0.3376894
## [2,] 0.4783206 -0.6786036 -0.4798995
## [3,] 0.3494796 0.1651372 0.8097298

```

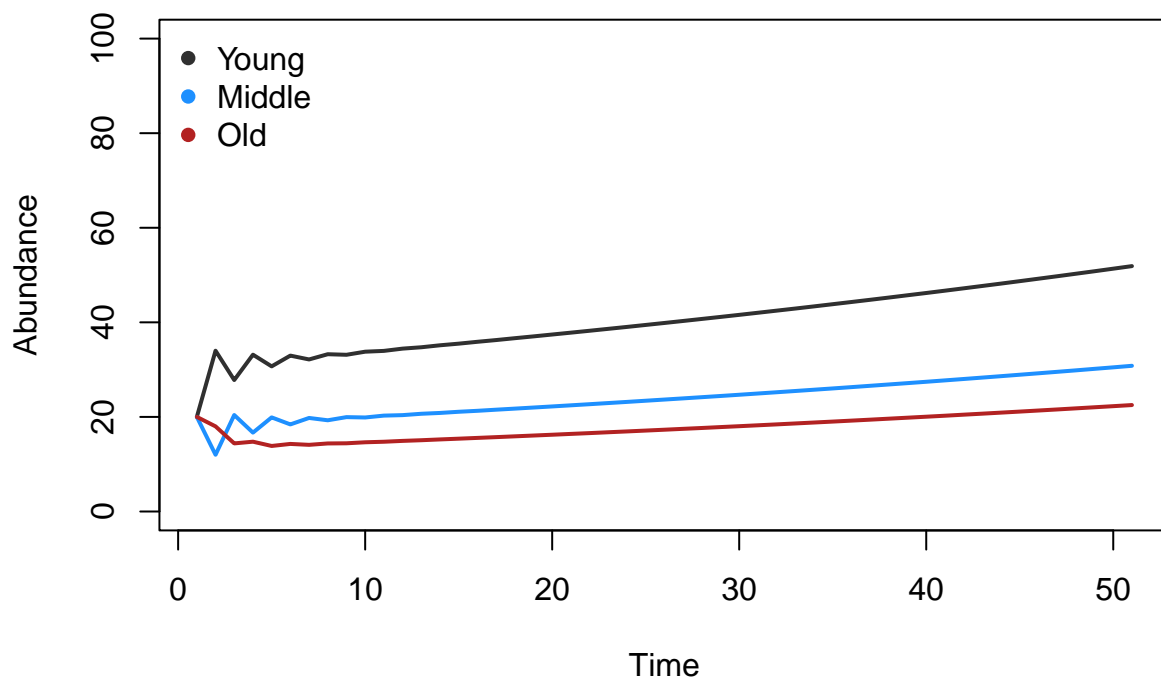
This suggests that for our Leslie matrix (`projMatrix`), the maximum geometric growth rate  $\lambda$  is 1.0105996, as this is the largest real-valued eigenvalue (i.e., the *dominant eigenvalue*). Each eigenvalue has an associated eigenvector. The eigenvector associated with the dominant eigenvalue describes the stable age distribution.

```
eigStable <- eigProj$eigenvectors[,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.

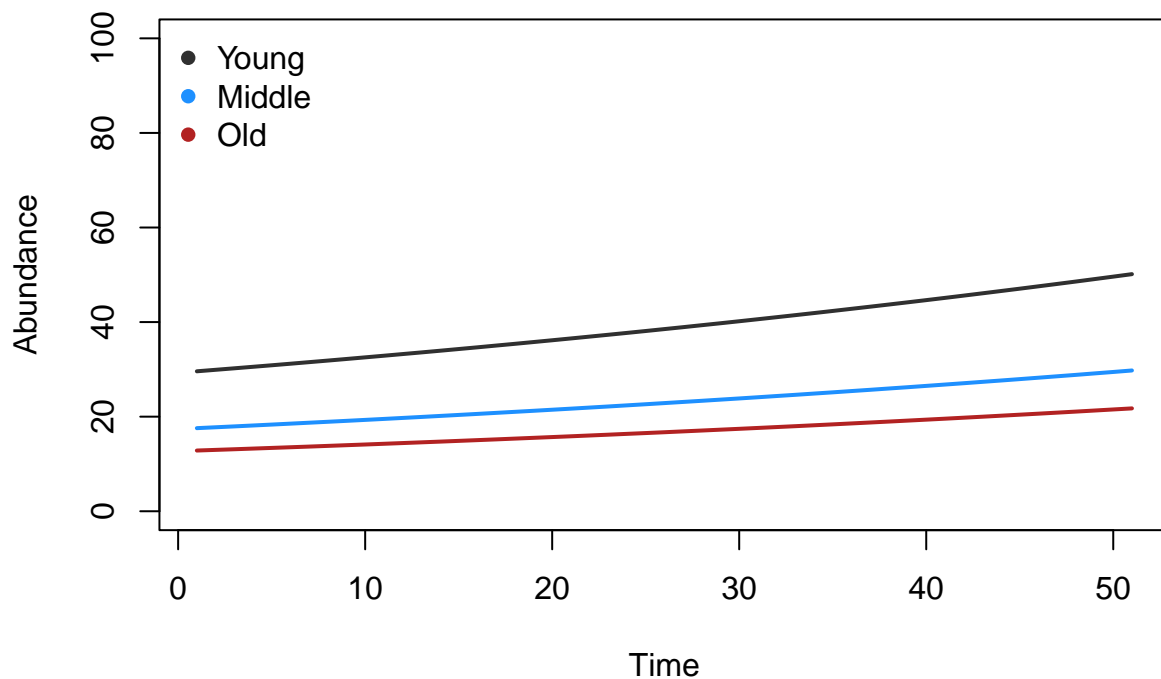
```
stageDynamics <- getStageDynamics(projMatrix, abund0, steps=50)
```

```
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)
```

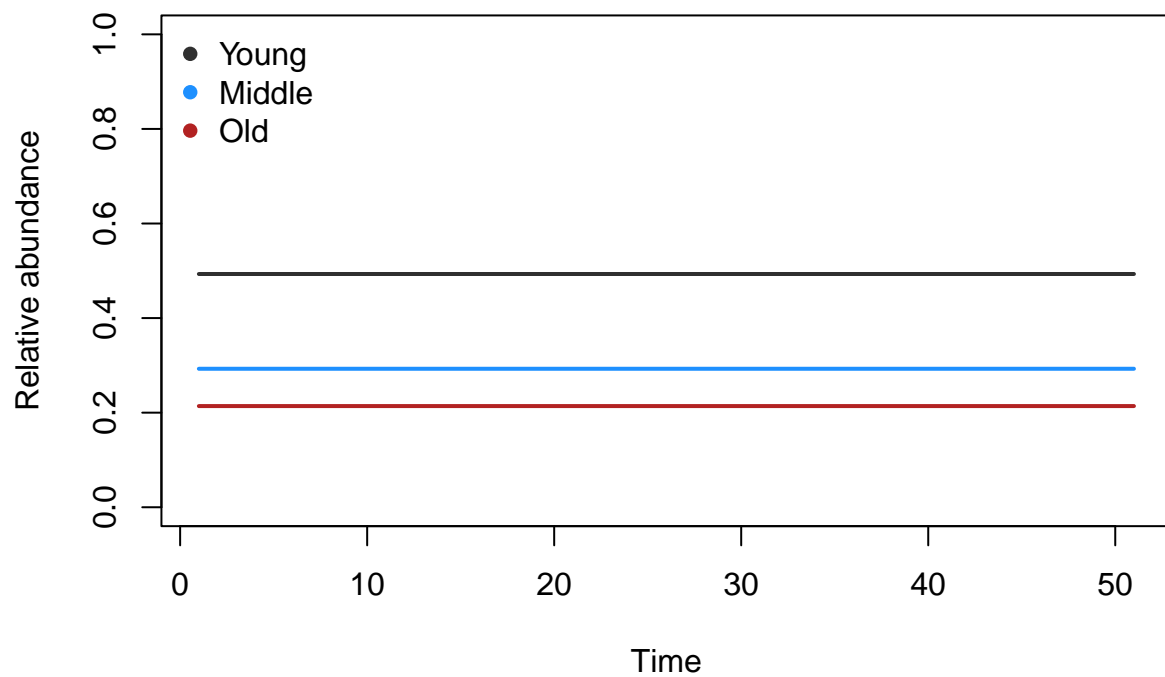
```
plot(stageDynamics[,1], type='l', lwd=2,
     col=colz[1], ylim=c(0,100),
     ylab='Abundance', xlab='Time')
lines(stageDynamics[,2], lwd=2, col=colz[2])
lines(stageDynamics[,3], lwd=2, col=colz[3])
legend('topleft', pch=16, col=colz[1:3],
     c('Young', 'Middle', 'Old'), bty='n')
```



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.