BIO311: Population Ecology
Practical 7 (Unstructured populations)

Timothée Bonnet &* Koen van Benthem

timothee.bonnet@ieu.uzh.ch
koen.vanbenthem@ieu.uzh.ch

Spring 2016

Contents

---

1 Discrete Population Growth

The botanical garden created a new pond in 2009 and introduced one plant of pond-lily. This population produces offspring each spring and grows with a discrete growth factor λ of 3, $N_t = N_0 \lambda^t$: after the first year, there were three individuals; $3 = 1 \cdot (3)^1$, by the second year the botanical garden had six more and in 2013, the total population size was 81.

Let us look at the evolution of population size over time. First we define the values of our different parameters.

```
N0 <- 1
lambda <- 3
t <- 0:4
```

Now, we get the population size over time and we plot this in a graph. The code `axis` sets the label below the x-axis to be 2009, 2010, 2011, 2012, 2013. The ^ sign means "raise at the power".

```
Nt <- N0*(lambda)^t
plot(t, Nt, xlab="time", ylab="N", xaxt = "n")
axis(1, at=0:4, labels=2009:2013)
```

Change the initial population size to 100. What happens if $\lambda < 1$ and $\lambda = 1$? Remember that λ can only be a positive number. Try plotting $\lambda < 0$ and explain why the behaviour in this graph is not very likely to occur in reality.
Hint Use `par(mfrow=...)` to see all your graphs in the same window. If you do not remember how to use this function, type ?`par` for help.
The botanical garden is curious to know how this pond-lily will do 10 years from now. Change the time parameter in the former code to project the population in the future.

2   Average Growth Rate

In nature, population growth rates between different year are rarely identical. Let us examine a hypothetical population of crocodiles in Egypte. The annual growth rate (R) of that population has varies from year to year. For example between 2009 and 2013; $R_{2009} = 1$, $R_{2010} = 0.9$, $R_{2011} = 0.7$, $R_{2012} = 1.4$. In that case, to get the finite growth rate, we need to take the average over the years.

Let us first see what happens when we use the arithmetic average.

```
R <- c(1, 0.9, 0.7, 1.4)
Rari <- sum(R) / 4
Rari
```

The Arithmetic average here is 1 so you would expect a population with this growth rate to remain constant. Check what is happenning:

```
R2009<-1
N2009<-100
N2010<-R2009*N2009

R2010<-0.9
N2011<-R2010*N2010

R2011<-0.7
N2012<-R2011*N2011

R2012<-1.4
N2013<-R2012*N2012
```

What is the population size in 2010, 2011, 2012 and 2013? Is your population constant on the long term as you would expect from a population with a growth rate of 1?
There is obviously something wrong with the arithmetic average. This is because with λ we are multiplying the R with each other: $N_4 = R_{2009} \cdot R_{2010} \cdot R_{2011} \cdot$

$R_{2012} \cdot N_0$. Instead of the arithmetic average, we need a average such that

$$\bar{R}^4 = R_{2009} \cdot R_{2010} \cdot R_{2011} \cdot R_{2012} \tag{1}$$

So we just have to solve for $\bar{R}$. To do so, we take the fourth root on each side of our expression:

$$\bar{R} = \sqrt[4]{R_{2009} \cdot R_{2010} \cdot R_{2011} \cdot R_{2012}}. \tag{2}$$

But $R_{2009} = \frac{N_{2010}}{N_{2009}}$, $R_{2010} = \frac{N_{2011}}{N_{2010}}$, $R_{2011} = \frac{N_{2012}}{N_{2011}}$ and $R_{2012} = \frac{N_{2013}}{N_{2012}}$. So we can re-write

$$\bar{R} = \sqrt[4]{\frac{N_{2010}}{N_{2009}} \cdot \frac{N_{2011}}{N_{2010}} \cdot \frac{N_{2012}}{N_{2011}} \cdot \frac{N_{2013}}{N_{2012}}}. \tag{3}$$

Which can be simplified as

$$\bar{R} = \sqrt[4]{\frac{N_{2013}}{N_{2009}}}. \tag{4}$$

We can generalise this to

$$\bar{R} = \sqrt[t]{\frac{N_t}{N_0}}. \tag{5}$$

where t is the last time step. Calculate the geometric mean, does this value make more sens than the arithmetric mean that we calculated before?

3 Continous Population Growth

So far, we looked at discrete population growth. This type of growth is suitable for describing the growth of populations which reproduce punctually, for example during one annual breeding period. To describe the growth of population such as bacteria, or humans, a continuous growth model is more appropriate. Instead of having a population punctually increasing with time, the population is constantly growing by an infinitesimally small change in population size. The change in population size over time is thus a constant (r) times the population size:

$$\frac{dN}{dt} = r \cdot N \tag{6}$$

We have seen during the lecture that the solution to this differential equation is:

$$N_t = N_0 e^{rt} \tag{7}$$

where r is the instantaneous rate of increase. If we know the population instantaneous rate of increase and the initial population size, we can project the population in the future.

Let us look at a rabbit population in the UK. Rabbits have been introduced to the UK by the Romans and have since then proliferated. The rabbit population we consider breeds all year round and can be considered as continuously growing. The population currently is composed of 300 individuals with an instantaneous growth rate of 0.7. The wildlife management organisation want to know what the population will look like in 10 years.

To project this population with equation 7, we need the function `exp()`. This function returns the exponential of its input. Try `exp(1)` and `exp(2)`. These results correspond to e^1 and e^2 respectively. If you input a vector instead of a scalar into `exp()`, it will return a vector in which the first entry is the exponential of the first entry of the input vector. The second entry of the output vector will be the exponential of the second entry of the input vector:

```r
exp(c(1, 2))
```

The next step is to set the initial values of the UK rabbit population and write the equation for the expontial population growth in `R`.

```r
N0 <- #insert initial value here
r <-  #insert instantaneous rate of increase here
t <- 0:10
Nt <- ...*exp(...)
```

And then we plot the results. Because we are now considering a continuous growth, we are assuming that the population is not punctually increasing but constantly increasing. Thus instead of just plotting points, we can now plot a continuous line. This is what the last part (`type="l"`) of the code does.

```
plot(t, Nt, xlab="time", ylab="N", type="l")
```

Do you think this is a realistic outcome? What mechanisms would prevent the population to grow like this in reality?

During the brief introduction to mathematics last week, we mentioned that the natural logarithm is the inverse function of the exponential. Plot $\ln(N_t)$ over time. The command in R for ln is `log()`. Do not forget to change the label of the y-axis. What is the slope of that line?

** Exercise 1   doubling time

Solve on paper at what time step the rabbit population has doubled in size. For this you need to derive the formula for doubling time in a population with exponential growth. Start from equation 7. Compare your result with the graph in R to check your answer.

3.1   The link between continuous and discrete population growth

Both the continuous and the discrete population growth describe the population size in the future, but what is the link between them? Let us state each equation again:

continuous

$$N_t = N_0 e^{rt} \tag{9}$$

discrete

$$N_t = N_0 \lambda^t \tag{10}$$

Both equations 9 and 10 give the population growth at time t so we can write:

$$N_0 \lambda^t = N_0 e^{rt} \tag{11}$$

Solving for λ we get:

$$\lambda = e^r \tag{12}$$

This means that if you have λ, the finite rate of increase, you can calculate the instantaneous growth rate (r).

4   Data Fitting

So far, we have worked on simulated data. In this section we will work on real biological data. We will extract key population growth parameters and plot the data assuming 1) density-independence, 2) unlimited resources, 3) no environmental stochasticity and 4) no demographic stochasticity. The data you collected in the lab on the rotifer is not well suited for the type of population model that we covered during this practical.

Instead, we will examine the variation of a beaver population from Alberta, Canada from 1919 to 1981. The data is from NERC Centre for Population

Biology, Imperial College (1999) The Global Population Dynamics Database.
`http://www.sw.ic.ac.uk/cpb/cpb/gpdd.html` and was published in *Furbearer harvests in North America, 1600-1984* by Novak, M., Obbard, M., Jones, J., Newman, R., Booth, A., Satterthwaite, A. & Linscombe, G. in 1987.

First you need to take the `beaver_pop.cvs` that we already put on OLAT and put it in a meaningful folder (for example the Pop_Ecol folder on your desktop we created last time). As you now know, we 1) need to set the working directory and 2) open the file. Change the directory according to the location of your file.

```
setwd("C:/Desktop/Pop_Ecol")
```

```
Bea <- read.csv("beaver_pop.csv", sep=";", header=T)
head(Bea)
str(Bea)
```

The next step is to plot your data. Try to play around with the value of `scipen` to see how it effects the graph. What happens if it has a value of -1? If we set it once, it is remembered throughout the rest of the `R`-session.

```
options(scipen=10)
plot(Bea$Year, Bea$Population, xlab="year", ylab="N")
```

It looks like this population may be decribed by an exponential function!

4.1 From the graph

Let us try to find correct model parameters from the graph. To do so, we plot the natural logarithm of the population to see if there may be a linear relationship.

```
plot(Bea$Year, log(Bea$Population), xlab="year", ylab="ln(N)")
```

Remember that the straight line of the plot of the natural logarithm of the population is r, the instantaneous rate of increase. So we need to find the line that fits the points we just plotted. That means we want to fit a line for which the deviations of the actual data points to the line are minimized. This is a called a **linear regression**. The `lm` function fits a regression line through the points. If you type `lm(N~t)` it finds the line of the form $N = at + b$ that best describes the relation between N and t. We however, want to fit a line through the natural logarithm of our points so we first transform them with the `log` function (we are thus looking for a line of the form $\ln(N) = at + b$).

```
reg1 <- lm(log(Bea$Population) ~ Bea$Year)
reg1 # Use this to take a look at reg1
```

And then we add the regression line on the graph.

```
abline(reg1)
```

By running `reg1`, you see the outcome of the regression line. You get two coefficients; the intercept (b) and the slope (a, the value below `Bea$Year`). These are the two values we want to find our exponential growth equation for this population.We have now an equation of the form

$$\ln(N) = at + b \tag{13}$$

Solving for N, we need to take the exponential on both sides of the equation which gives us $N = e^{at+b}$. This can also be writen as $N = e^b \cdot e^{at}$. Compare this to equation 7, and find N_0 and r.

Why is the initial population size for this equation not close to 5685, the population size in 1919?

So we have found the values of the function:

$$N_t = N_0 e^{rt} \tag{14}$$

Let us now find the population size that would be predicted by the intercept and the slope we extracted from the linear regression. For this, use the commands `reg1$coef[[1]]` and `reg1$coef[[2]]`.

```
Bea$sim <- ... * exp(... * (Bea$Year))
```

And plot it on the same graph as the original population.

```
plot(Bea$Year, Bea$Population, xlab="year", ylab="N")
lines(Bea$Year,  Bea$sim)
```
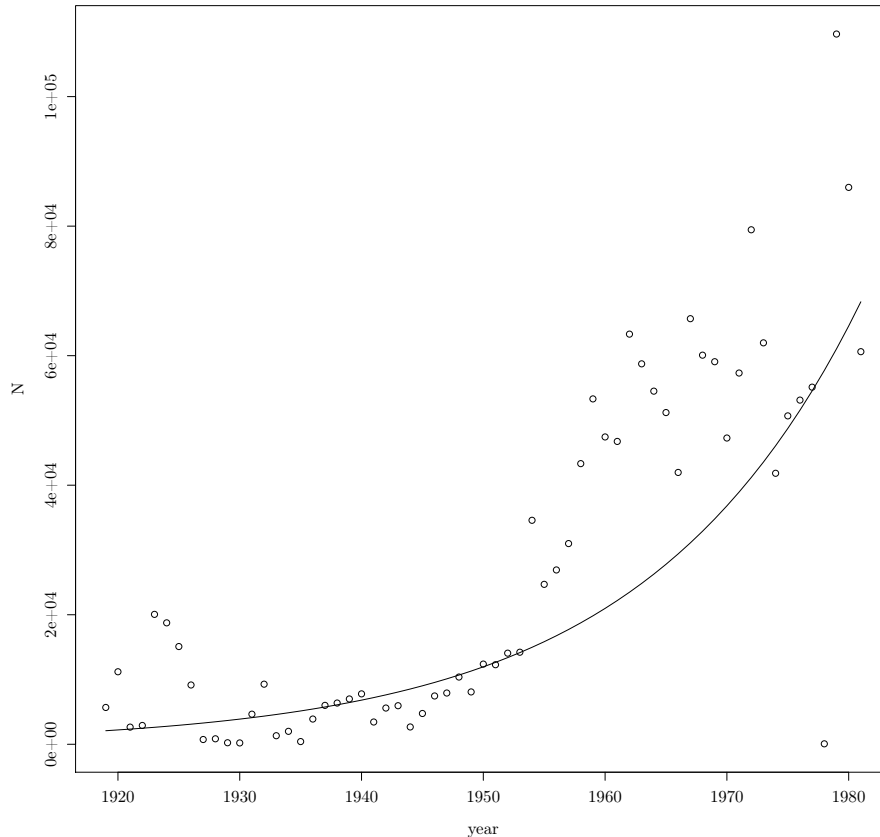
The fit is actually not bad at all!

** Exercise 2   fitting data from graph

As an exercise, write down the exponential growth equation of this beaver population with the parameter values. Knowing the relationship between the finite rate of increase λ and the instantenous growth rate r, find λ and write the discrete population growth equation down.

Show your equations to one of the instructors.

4.2   More about the average growth rate

An alternative to find the function describing the growth of this population is to go from the average growth rate which we will call R. In the section *Average growth rate*, we showed that the average growth rate is given by $R = \sqrt[t]{\frac{N_t}{N_0}}$. The code `length` gives the number of elements of the object it is applied to. We take the length of our population variable -1 because among n points, there are

always $n - 1$ transitions. Run separately the different parts of the code below to find out what they do.

```
N0 <- Bea$Population[1]
Nt <- Bea$Population[length(Bea$Population)]

lambda <- (Nt / N0) ^ (1 / (length(Bea$Population) - 1))
lambda
```

** Exercise 3   Comparing the λs

Compare this value of lambda with the value of lambda you found in the former exercise. Knowing the relationship between the finite rate of increase λ and the instantenous growth rate r, find r and write the discrete and continuous population growth equations with the appropriate parameters
Show your equations to one of the instructors.

Let us compare the two curves we got with the different value of λ. For this we again use the same plotting code and an extra line.

```
plot(Bea$Year, Bea$Population, xlab="year", ylab="N")
options(scipen=10)
lines(Bea$Year,  Bea$sim) #line from the linear regression
lines(Bea$Year,Bea$Population[1]*
        lambda^(0:(length(Bea$Population)-1))
      ,col='red') #line from the average growth rate
```

You can see that the two curves are not identical. This is because of the methods we used to estimate λ. The curve from the linear regression is a fit. The average distance between the data points and the line is minimized. You can see that the curve neither starts nor ends at one of the data points. However the curve obtained from the average growth rate is a mean and does starts at the first data point and ends on the last data point of the serie. To convince you of this, let us repeat both analyses on a subset of the dataset where the last three points are excluded.

```
Beasub<-Bea[1:60,]

#From the fit
reg2 <- lm(log(Beasub$Population)~Beasub$Year)
Beasub$sim<-exp( reg2$coef[[1]])*exp( reg2$coef[[2]]*(Beasub$Year))

#From the average growth rate
N02<-Beasub$Population[1]
Nt2<-Beasub$Population[length(Beasub$Population)]
```

```
lambda2<-(Nt2/N02)^(1/(length(Beasub$Population)-1))
lambda2

## [1] 0.9279561

#The plots
plot(Bea$Year, Bea$Population, xlab="year", ylab="$N$")
options(scipen=10)
lines(Bea$Year,  Bea$sim, col='black')
lines(Bea$Year,Bea$Population[1]*lambda^(
  0:(length(Bea$Population)-1)),col='red')

lines(Beasub$Year,  Beasub$sim, col='black', lty=2)
lines(Beasub$Year,Beasub$Population[1]*lambda2^(
  0:(length(Beasub$Population)-1)),col='red', lty=2)
legend("topleft",
       c("fit full ","fit subset", "avg full", "avg subset"),
       bty = "n",
lty=c(1,2,1,2),
col=c("black", "black", "red", "red"))
```
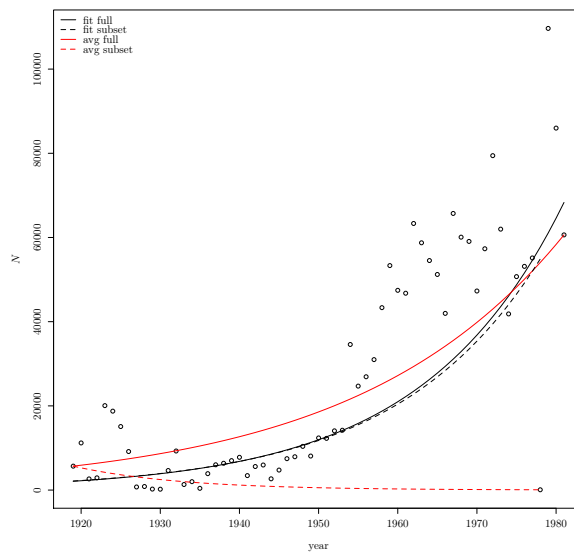

When the last three points have been removed, the last point is a particularily low value. You can see in this case that the λ from the average growth rate causes the projected population to be far from the actual population trajectory because the curve goes through both the first and the last point. In contrast, the two curves issued from the projections from the λ from the linear regressions are not very different from each other. The last and first data points have much

less influence on their behaviour.

5 Logistic Growth

5.1 Continuous Logistic Model

So far we have assumed that the rabbits have a constant growth rate (where we found that $r = 0.7$), regardless of the population size. In reality however, this growth rate is the maximum possible growth rate. In larger populations the actual growth rate is smaller than this maximal value (due to competition for food). Below we show how the growth rate depends on the population size:
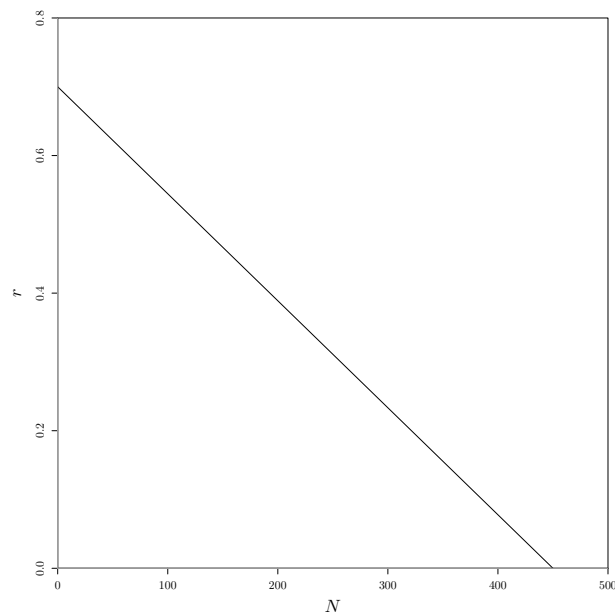
From this graph, find the carrying capacity (K) of this population and set it in R:

```
K <-
```

What is the population growth rate when the population size equals the carrying capacity?

$$r(K) = \ldots$$

Think about what this means for the population size when it reaches the carrying capacity.

In the lecture we saw that when the growth rate decreases linearly with the population size, the population can be described by logistic growth:

$$N = \frac{K}{1 + \frac{K - N_0}{N_0} e^{-r_{max}t}}. \tag{15}$$

12

Now we will use R to see how changes in K, N_0 and r_{max} influence the population size over time. First we need to create a vector in R that contains the time values at which we want to evaluate $N(t)$. For this exercise, let us use the time vector:

$$\texttt{time} = \texttt{c}\,(0, 0.1, 0.2, 0.3, \ldots 10)$$

Use the function `seq()` to make this vector in R. Type `?seq` in the console if you do not know how the function `seq()` works.

```
time<-seq(    ,    ,    )
```

Check your vector by outputting the first and the last few entries of `time`:

```
head(time)

## [1] 0.0 0.1 0.2 0.3 0.4 0.5

tail(time)

## [1]  9.5  9.6  9.7  9.8  9.9 10.0
```

Now we will calculate the values of N at these times. To do so we need the function `exp()`. Like before, we assume that the population size starts with 300 individuals. In order to find the population size N at all the time points in `time`, complete the following code:

```
K <-      # Input your value of K here
rmax <-   # Input the value of rmax here
N0 <- 300
N <- K / (... * exp(... * time))
```

Now we plot the behaviour of N against time:

```
plot(time, N, ylab="N", xlab="t", type="l", ylim=c(200,500))
```
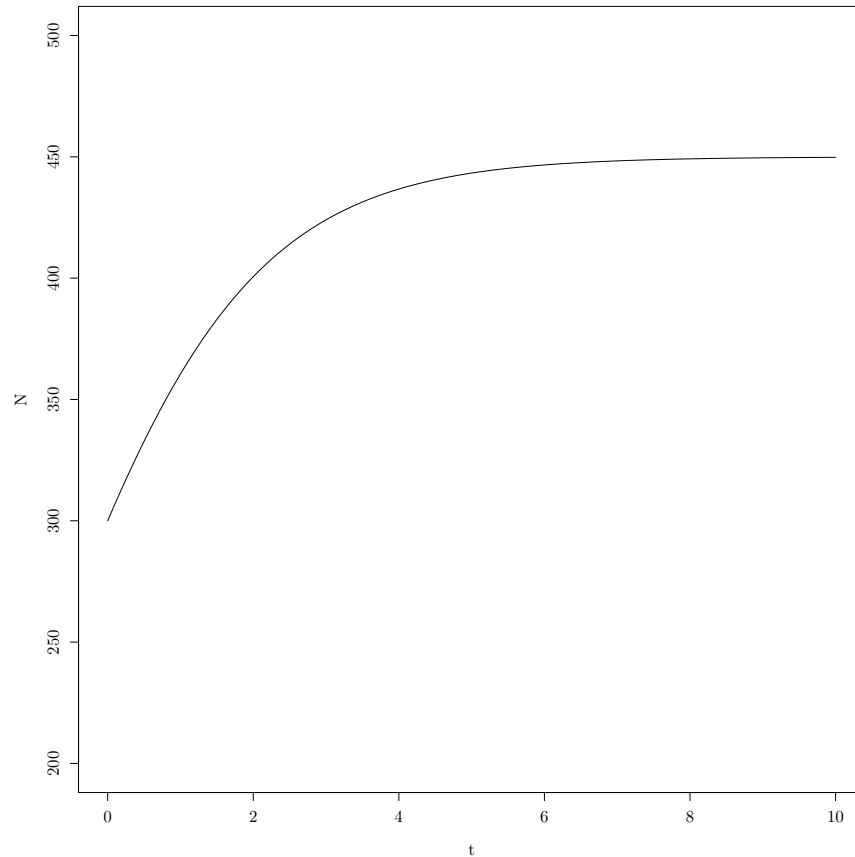
Now change one of the parameters $(K, r_{max}, N0)$ and calculate the new values for N. Next, you can add a line with these new values to your current plot by using the function `lines()`. Below the code is illustrated for a change in the initial population size (N_0):

```
NO <- 350
N <- K / (... * exp(... * time))
lines(time,N,col="red")
```

Now change r_{max} and K and see how they influence the behaviour of the population. **If you want to compare the changed situation to the original one, do not forget to set N_0 back at its original value. You may consider writing a function that returns N.** What do the different parameters (N_0, r_{max}, K) determine? What happens if $K < N_0$? And what do you see if $N_0 < \frac{K}{2}$? *Make sure to adjust the parameter `ylim` in your plot if necessary.*
We conclude this exercise on continuous logistic growth with a *pen & paper* exercise:

** Exercise 4   Logistic Growth

In the lecture Arpat talked about density dependent population growth. He showed that if the growth rate of the population depends linearly on the number of individuals, the following equation holds:

$$\frac{dN}{dt} = r_{max} N \left(1 - \frac{N}{K} \right). \tag{16}$$

He also showed that the solution to this equation is:

$$N = \frac{K}{1 + \frac{K-N_0}{N_0} e^{-r_{max} t}}. \tag{17}$$

From this equation calculate separately $\frac{dN}{dt}$ and $r_{max} N \left(1 - \frac{N}{K} \right)$ and show that the two are equal. Hereby you show that equation 17 is indeed the solution to equation 16.

5.2 Discrete Logistic Growth Model

Now we will consider the discrete logistic growth model (Ricker's version, see lecture):

$$N_{t+1} = N_t e^{\left(r_d \left[1 - \frac{N_t}{K} \right] \right)}. \tag{18}$$

Let us consider again the pond-lily example. The botanical garden decided to create a second, larger pond where they put 300 individuals ($N_1 = 300$). The growth rate of this new population is 0.2 and has a carrying capacity at $K = 400$. Let us set these variable in `R`:

```
N1 <- 300
rd <- 0.2
K <- 400
```

Now from these numbers, calculate the population size at $t = 2$ (that is, we want to calculate N_2). Now also calculate N_3 and N_4. The answers are shown below, so you can check them:

$N_2 = 315.4$

$N_3 = 329$

$N_4 = 340.9$

If we want to know what the population size is at $t = 100$, it is a bit inconvenient to manually calculate N_2 up to N_{100} manually. So let's try to automatize that calculation. First we create a vector N with only zeros, in the end we will put the values of N_1 up to and including N_{100} in this vector. Now create a vector with 100 entries where every entry is zero. For this use the function rep(), if you do not know how this function works, use ?rep to obtain more information on the function

Verify that the vector indeed has the desired length and that it contains zeros:

```
length(N)

## [1] 100

head(N)

## [1] 0 0 0 0 0 0
```

We already know the population size at the first position, so we can fill in the value of the first value:

```
N[1] <- 300

head(N)

## [1] 300   0   0   0   0   0
```

Now we need to calculate the other entries of N, to do so we use a *for-loop*. Below is an incomplete version of the code, please complete it and run the code to fill N with the correct values:

```
for(i in 2:..){
  N[i] <- N[i-..] * exp(....)
}
```

Check the final vector to see that the values that were calculated in this way for the first few entries equal the values that you found earlier by manual calculation. Now we can easily retrieve the population size at $t = 100$. Now plot how the population develops over time.

```
N[100]
```

```
## [1] 400
```
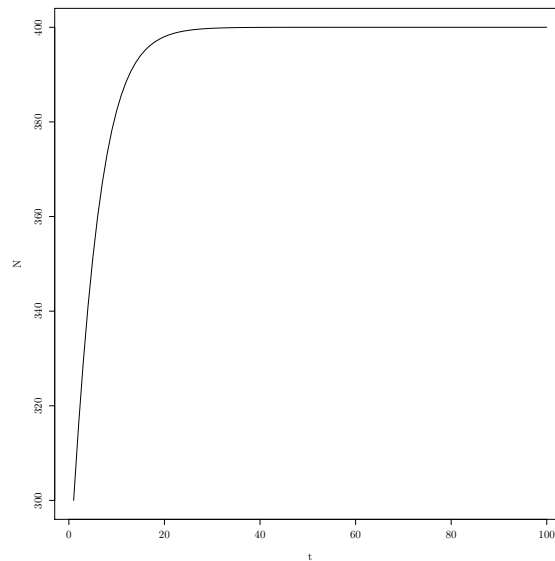

Run the model again with different values of r_d, can you explain why the model behaves strangely for large values of r_d (e.g. $r_d = 6$)?

5.3   Environmental Stochasticity

So far we have assumed that the growth rate r_d and K remain constant over time. In reality however, environmental influences will lead to these rates being different every year. We will incorporate this stochasticity in the model for the second pond-lily population and see how this influences the predictions of the model.
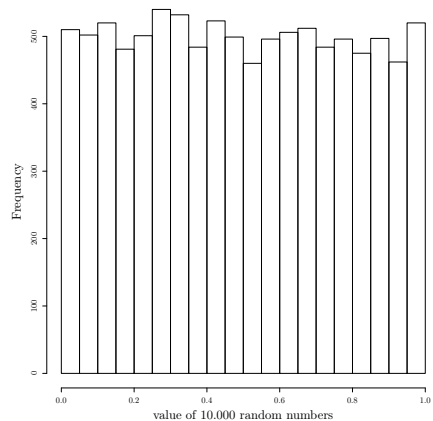
5.3.1   Random numbers

In order to incorporate this, we need random numbers. In R functions are available to generate pseudo random numbers from different distributions. One such function is `runif(N)`, this function generates N random numbers between 0 and 1, where every number between 0 and 1 is equally likely to be drawn (the distribution is uniform).

```r
runif(7)
```

```
## [1] 0.74391237 0.77038642 0.75648707 0.40006851 0.03212594
## [6] 0.80518831 0.68901597
```

```r
hist(runif(10000),n=20,main="",
     xlab="value of $10.000$ random numbers", cex.lab=1.5)
```


A computer works in a deterministic way, it is therefore not able to generate truly random numbers. Some algorithms have been developed however that generate series of numbers that very strongly resemble random numbers. They are not truly random: they depend on an initial number which is called the seed. If the seed is set to the same value, R will generate the same random numbers. The seed can be set by using the function set.seed(). When you use R for generating a random number, without explicitly setting the seed during an R-session, the seed is generated from the current time on your computer and the process ID. Compare the following lines of code:

```r
set.seed(24)
runif(7)
```

```
## [1] 0.2925740 0.2248911 0.7042230 0.5188971 0.6626196
## [6] 0.9204438 0.2797356
```

```r
runif(7)
```

```
## [1] 0.7638205 0.8016306 0.2547251 0.6048889 0.3707349
## [6] 0.6716903 0.6729823
```

```r
set.seed(24)
runif(7)
```

```
## [1] 0.2925740 0.2248911 0.7042230 0.5188971 0.6626196
## [6] 0.9204438 0.2797356
```

More random numbers than this can be obtained from `www.random.org`. This website generates random numbers based on atmospheric noise. For most simulation purposes, however, pseudo random number generators are good enough. A second function for generating random numbers is the function `rnorm()`, use this function to create a histogram for 10.000 random numbers with a standard deviation and mean of your choice. If you are not familiar with this function, use `?rnorm` to obtain more information on this function.

5.3.2 Environmental Stochasticity in the Logistic Growth Model

We will now use such random numbers to simulate environmental stochasticity in the logistic growth model. Before we assumed that both r and K were the same every timestep. Instead of assuming this, we will now assume that every timestep these parameters are drawn from a normal distribution. Back to the pond-lily population, for r_d the mean of this distribution is $\mu = 0.2$ and the standard deviation $\sigma = 0.1$. For K we take a normal distribution with mean $\mu = 400$ and standard deviation $\sigma = 25$. Now adapt the code for the discrete logistic growth model to draw 1 new value for both K and r_d every time step and using these values to calculate the population numbers. We start with a population size of 300. Plotting the final result may look like this (depending on your random numbers of course...):
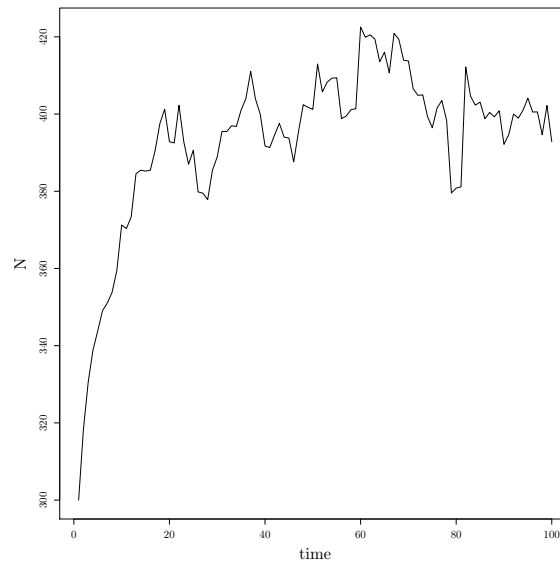
Now we will make a plot that shows multiple possible outcomes of the logistic growth:
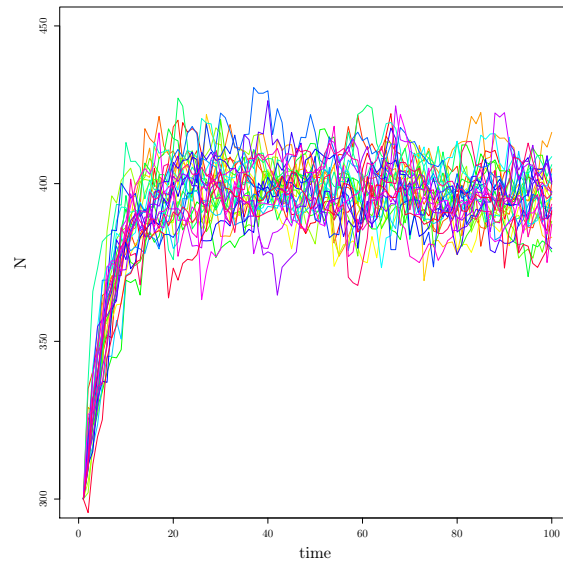
To create this plot, we need to perform the model multiple times: to do so we put it in a for-loop. This procedure is outlined below:

```r
# First we create an empty plot that will later
# contain the different lines. We set the range
# of the axes (xlim,ylim) and by setting type=
# "n", we specify that no data should be
# plotted yet.

plot(0,0,xlim=c(0,100),ylim=c(300,450),
     type="n",xlab="time",ylab="N")

# We define 30 colors that we will use to plot
linecols=rainbow(30)

# Now we start the loop, in this example,
# we created 30 realisations of the stochastic
# model:
for(rep in 1:30){

  ...
  ...

  # Here you have to put your code for creating
  # 1 realisation of the stochastic growth model
  # (this includes a loop)
```

```
  # Finally we add a line with this new realisation to
  # the current plot
  lines(1:100,N,col=linecols[rep])
}
```

If you want you can repeat this exercise with different values for the standard
deviations of K and r_d to see how this changes the output.

6   Demographic Stochasticity

We finally do a short exercise on demographic stochasticity. Assume we have a
small population of 10 bears in the Swiss Alpes. In this population the chance
of survival is $p = 0.6$.
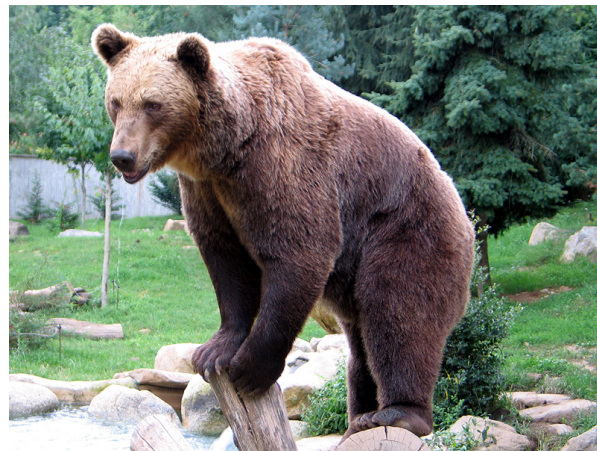
First we take a look at the individual. So how do we decide whether it survives
or not? We draw a random number between 0 and 1 from a random distribution:

```
a <- runif(1)

## 0.7186776
```

If this random number (a) is larger than 0.6, the individual dies, if it is smaller
than 0.6, it survives. Because every number between 0 and 1 is equally likely
to be drawn, this is the same thing as having a survival chance of 0.6. In this
case the individual thus dies.

We do not have 1 individual, but we have 10 individuals, so we need 10 random
numbers between 0 and 1 to determine which individuals survive:

```
a <- runif(10)

##  [1] 0.02316389 0.90885056 0.71608529 0.23073700 0.74916228
##  [6] 0.88252136 0.67751503 0.38407123 0.79324897 0.45452743

a < 0.6

##  [1]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
## [10]  TRUE

sum(a < 0.6)

## [1] 4
```

Here we have used that `sum(a < 0.6)` counts the number of times that `a < 0.6` contains the value `TRUE`. The proportion of surviving individuals is thus not 0.6, but instead:

```
sum(a < 0.6) / 10

## [1] 0.4
```

Note that the reason for this is not that the chances of survival are different, over time, but instead that this year more individuals were unlucky and as a result of that died. Hence, this is demographic stochasticity and not environmental stochasticity.

To summarise this code, we can find the proportion of surviving individuals as follows:

```
sum( runif(10) < 0.6) / 10
```

Use a for loop to repeat this calculation 1000 times and store all the results. You can plot them using `hist`. What happens to the proportions when the number of individuals increases? (assume that the chance of survival remains constant).

```
Np <- 1000
outcome <- ... # Create empty vector
for(i in 1:...){
  outcome[i] <- ...
}
hist(outcome, xlim=c(0,1))
```

7   Additional Exercise

This exercise is extra!
In case you finish the exercises fast and don't want to go home yet: can you

adapt the discrete logistic growth model to incorporate a delayed density effect? (try it first without the stochasticity, then you can include the stochasticity on r and K.)

8 Additional information: The continuous logistic growth model

Extra: Logistic Growth: Solving the differential equation (for those who wish to know why the solution is the solution...)

$$\frac{dN}{dt} = r_{max} N \left(1 - \frac{N}{K} \right). \tag{19}$$

With solution:

$$N = \frac{K}{1 + \frac{K - N_0}{N_0} e^{-r_{max} t}}. \tag{20}$$

We have seen in the exercise that if you have equation 20, one can show that it obeys equation 19, but how to find equation 20 from equation 19? Let us first, for simplicity set $K = 1$ and $r_{max} = 1$ (if you want, you can repeat the exercise without doing that):

$$\frac{dN}{dt} = N(1 - N). \tag{21}$$

First write the equation in a way such that all terms that contain N are on the left side of the equation and all terms containing t on the right side:

$$\frac{dN}{N(1 - N)} = dt \tag{22}$$

Now we need to realise that in general:

$$\frac{1}{x} + \frac{1}{1 - x} = \frac{1 - x}{x(1 - x)} + \frac{x}{x(1 - x)} = \frac{1 - x + x}{x(1 - x)} = \frac{1}{x(1 - x)} \tag{23}$$

Using this we can write:

$$\frac{dN}{N} + \frac{dN}{1 - N} = dt \tag{24}$$

Now we take the primitive on the left and the right side of the equation:

$$\ln(N) + C_1 - \ln(1 - N) + C_2 = t + C_3 \tag{25}$$

Let us group all these constants together in a new constant:

$$C = C_2 + C_3 - C_1 \tag{26}$$

Now:

$$\ln(N) - \ln(1 - N) = t + C \tag{27}$$

We take the exponential on both sides:

$$e^{(\ln(N)-\ln(1-N))} = e^{t+C} \tag{28}$$

Next we use that $e^{a+b} = e^a e^b$ and that $e^{-x} = \frac{1}{e^x}$:

$$\frac{e^{(\ln(N))}}{e^{(\ln(1-N))}} = e^C e^t \tag{29}$$

Now we use that $e^{\ln(x)} = x$ and we define a new constant: $A = e^C$

$$\frac{N}{1-N} = Ae^t \tag{30}$$

Rewriting this equation:

$$N = (1-N)Ae^t = Ae^t - ANe^t \tag{31}$$

And thus:

$$N(1 + Ae^t) = Ae^t \tag{32}$$

$$N = \frac{Ae^t}{1 + Ae^t} \tag{33}$$

Now we divide the numerator and the denominator on the right side by Ae^t to get the solution to the equation:

$$N = \frac{1}{1 + \frac{1}{A}e^{-t}} \tag{34}$$

Next, we know that at $t = 0$, $N = N_0$, so we plug this in the equation:

$$N_0 = \frac{1}{1 + \frac{1}{A}e^0} = \frac{1}{1 + \frac{1}{A}} \tag{35}$$

Now:

$$\left(1 + \frac{1}{A}\right)N_0 = 1 \tag{36}$$

So:

$$1 + \frac{1}{A} = \frac{1}{N_0} \tag{37}$$

$$\frac{1}{A} = \frac{1}{N_0} - 1 = \frac{1}{N_0} - \frac{N_0}{N_0} = \frac{1 - N_0}{N_0} \tag{38}$$

Plugging this into the solution that we found:

$$N = \frac{1}{1 + \frac{1-N_0}{N_0}e^{-t}} \tag{39}$$

This information is extra!

9 Answers to the *pen & paper* exercises

Answer of exercise 1

We want to derive an equation for t_{double} that is the time at which the population is twice the size of the initial population.

$$N_{t_{double}} = 2N_0 \tag{8}$$

Substituting this expression into equation 7 we get:

$$2N_0 = N_0 e^{rt_{double}}$$

Now we need to solve for t. We start by dividing by N_0 which is on both sides of the equation.

$$2 = e^{rt_{double}}$$

Then we take the natural logarithm on each side to get ride of the exponential.

$$\ln(2) = rt_{double}$$

From there, the equation can be rearranged.

$$\frac{\ln(2)}{r} = t_{double}$$

You can use R as a calculator to find the value of t_{double} from the value of r given at the begining of the section.

```r
r<- 0.7
tdouble<-log(2)/r
tdouble
```

Even before a full year has passed, the population has already double in size.

Answer of exercise 2

$N_t = e^{-100.248} \cdot e^{0.056t}$

```r
lambda1<-exp(reg1$coef[[2]])
lambda1

## [1] 1.057835
```

$\lambda = 1.058$
$N_t = e^{-100.248} \cdot 1.058^t$

Answer of exercise 3

$N_t = 5685 \cdot 1.039^t$
$\lambda = 1.039$

```r
r1<-log(lambda)
r1
```

```
## [1] 0.03817303
```

$N_t = 5685 \cdot e^{0.038t}$

Answer of exercise 4

Let us first consider $\frac{dN}{dt}$. First we write:

$$N(u) = \frac{K}{u} = Ku^{-1} \qquad\qquad u(t) = 1 + \frac{K - N_0}{N_0}e^{-r_{max}t}$$

From this we can find:

$$N'(u) = -Ku^{-2} \qquad\qquad u'(t) = -\frac{K - N_0}{N_0}r_{max}e^{-r_{max}t}$$

We know the chain rule: $N'(t) = N'(u)u'(t)$, so:

$$\frac{dN}{dt} = Ku^{-2}\frac{K - N_0}{N_0}r_{max}e^{-r_{max}t}$$

Substituting u:

$$\frac{dN}{dt} = \frac{\frac{K-N_0}{N_0}Kr_{max}e^{-r_{max}t}}{(1 + \frac{K-N_0}{N_0}e^{-r_{max}t})^2}$$

Now we consider the righthand side of the equation: $r_{max}N\left(1 - \frac{N}{K}\right)$. We fill in the solution for N:

$$r_{max}N\left(1 - \frac{N}{K}\right) = r_{max}\frac{K}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\left(1 - \frac{1}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\right)$$

$$= r_{max}\frac{K}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\left(\frac{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}} - \frac{1}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\right)$$

$$= r_{max}\frac{K}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\left(\frac{\frac{K-N_0}{N_0}e^{-r_{max}t}}{1 + \frac{K-N_0}{N_0}e^{-r_{max}t}}\right)$$

$$= \frac{Kr_{max}\frac{K-N_0}{N_0}e^{-r_{max}t}}{(1 + \frac{K-N_0}{N_0}e^{-r_{max}t})^2}$$

Comparing this outcome with the expression for $\frac{dN}{dt}$, we see that the two are equivalent and thus that equation 17 satisfies equation 16. Arpat was thus correct when he told us this!