

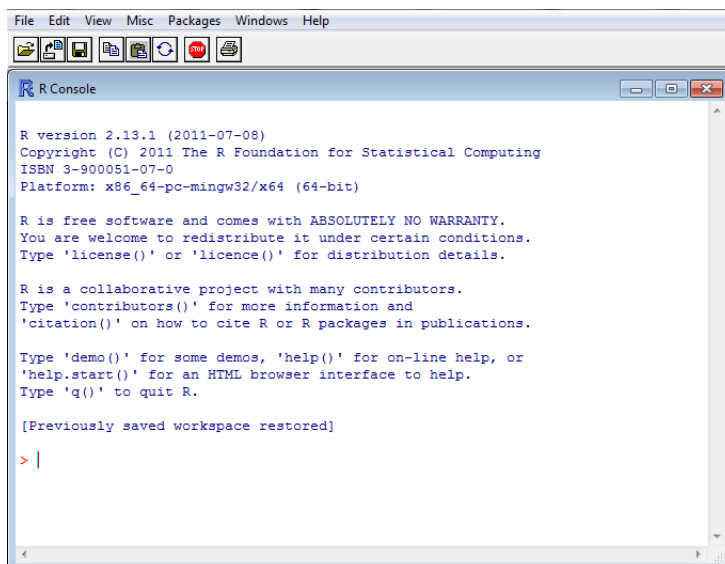
1 A simple model of fishery dynamics

We now demonstrate the dynamics of a fishery by developing a simple dynamic bioeconomic model. We will examine the behaviour of this model in two ways. The first way is to derive the equilibrium conditions of the model, as well as its behaviour when it deviates from the equilibrium, algebraically. The second way is to develop a simple numerical model in R.

R is a language and environment for statistics and mathematical modelling. It is used a lot in statistics, biology, and the social sciences. For instance, some of the most widely used fisheries models are written in R. Admittedly, there are many other languages out there that can do similar tasks as R, such as Matlab, Stata, SAS, Scilab, C, Python, etc. Stata is most widely used in econometrics; Matlab is very widely used in computational economics. Scilab and Python are open-source software. Compared to these programmes, however, R has two important advantages. First, R is open source. Unlike Matlab, Stata, and SAS, you can download, install, and run R, and all its add-ins, for free. Second, R has more and better functionalities than other open-source software such as Scilab and Python.

1.1 Getting started with R

You can download R from <http://www.r-project.org> and install it on your own computer. When you open R you see the following:



```
File Edit View Misc Packages Windows Help
R Console
R version 2.13.1 (2011-07-08)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> |
```

Figure 1: The R console

The one window that is open now is the console. This is your prime way of communicating with R, but to write computational models you will need to use the so-called script editor. First, however, you need to change the working directory of R to your directory of choice. You can check (and, if you prefer, change) the working directory as follows:

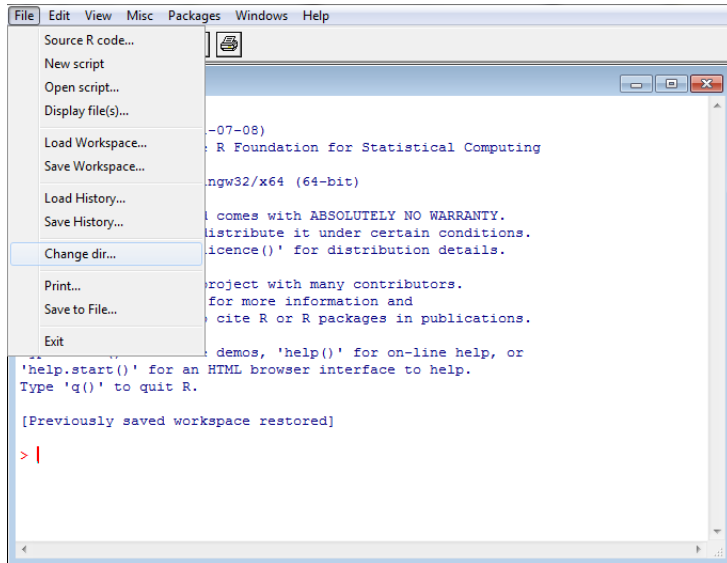


Figure 2: Changing the working directory in R

You will see that the default working directory is 'My Documents'. If you want to keep 'My Documents' as your working directory, simply click 'OK'. If you prefer another existing directory, click on that directory and click 'OK'. If you want to make a new directory (preferred: call it 'R practical' or something like that) then open the directory where you want this new directory and click 'Make New Folder'. *Be aware that R always starts with the same working directory so you always need to change it to your preferred working directory when you start R!*

1.2 Biological growth

Imagine a petri dish with one bacteria. As long as there are no external limitations, the number of bacteria will show what is called *exponential growth*: every given length of time Δt (say, 30 minutes), the number of bacteria doubles. The first bacteria splits in two after Δt ; after another Δt these two have split, bringing the number to four; after another Δt we have eight bacteria; and so on. In other words, the growth *rate* of the bacteria colony is constant, whereas the absolute growth, i.e. the number of bacteria added each Δt , depends on the size of the colony:

$$X_t = X_{t-1} + rX_{t-1} \quad (1)$$

where r is the growth rate as a fraction of current biomass. Let's plot this function in R. First we click 'File' → 'New Script':

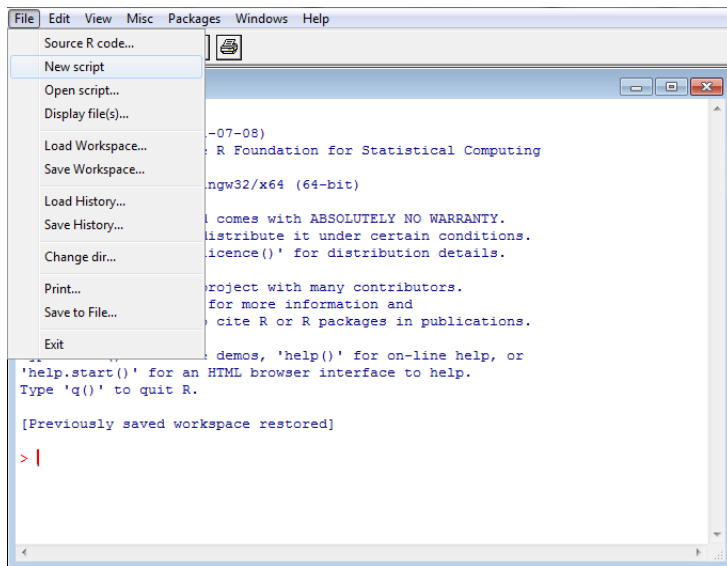


Figure 3: Opening a new script file in R

In this file we first define the variables used:

```
# Define parameter indicating number of years
numberOfYears <- 10

# Define vector for stock size
stockSize <- vector(mode="numeric", length=numberOfYears)

# Define stock growth parameter
stockGrowth <- 0.05
```

Let's go through this code step by step. The first thing you see is the hash ("#"). R ignores everything in a line starting with a hash. So why use it? Because the sentence "Define parameter indicating number of years" may not be understood by R, but all the better by you. The hash is the R way of inserting comments. Comments are very helpful for yourself or other users to understand what the code is doing. Would you still understand your code in, say, 5 years' time if you did put in some explanation of what it is doing? Therefore, one of the rules of good coding is *Always add comments to explain what the code is doing*.

After the comment we see the name of the parameter, `numberOfYears`, with the sign "`<-`" behind it. "`<-`" is a typical R oddity. Here it simply says: "assign the value 10 to `numberOfYears`". More recent versions of R also accept "`=`" to assign values to parameters, but here we stick to the convention of using "`<-`".

Also note that the parameter for the number of years is fairly lengthy. "`T <- 10`" would have worked just as well. But again, would you be able to understand 5 years later what `T` stands for? Therefore, another rule of good coding is *Always use descriptive names*.

Next we encounter another comment; in the line below the comment, the definition of a vector. This is the vector of stock sizes `X`; it has as many entries as there are years `t`. The vector is created with a function called `vector`, and this function contains two arguments:

- The type of data stored in the vector (`mode`);
- The length of the vector (`length`).

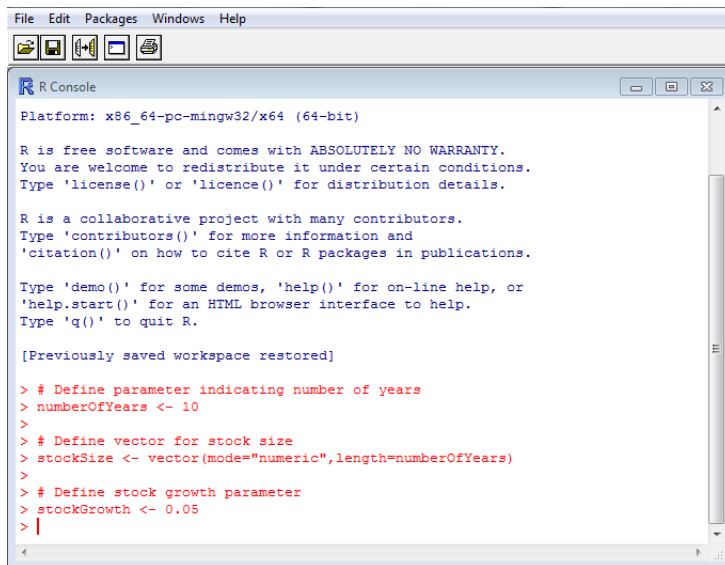
The mode of this vector is `numeric`, because we want to store numbers in it. Vectors can also contain string data (i.e. words), but we will not use that function here. The length of the vector is the number of years.

Lastly, we encounter another comment, and then the definition of the stock growth parameter `stockGrowth`, which we assume to be equal to 0.05.

First save this script by clicking 'File' → 'Save'. R will ask you to give a name to the file. Again, it is a good idea to give a descriptive name to the file. More importantly, however, *add the extension*

".r" to the name. It's a bit silly, but R does not automatically add the extension ".r" to file names of scripts. So your file should be called "expgrowth.r", or "bioeconmodel.r", or something like that - *just don't forget the ".r" part.*

Then we run the script. To run it, click "Edit" → "Run All". R will show the script in the console:



```
File Edit Packages Windows Help
R Console
Platform: x86_64-pc-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

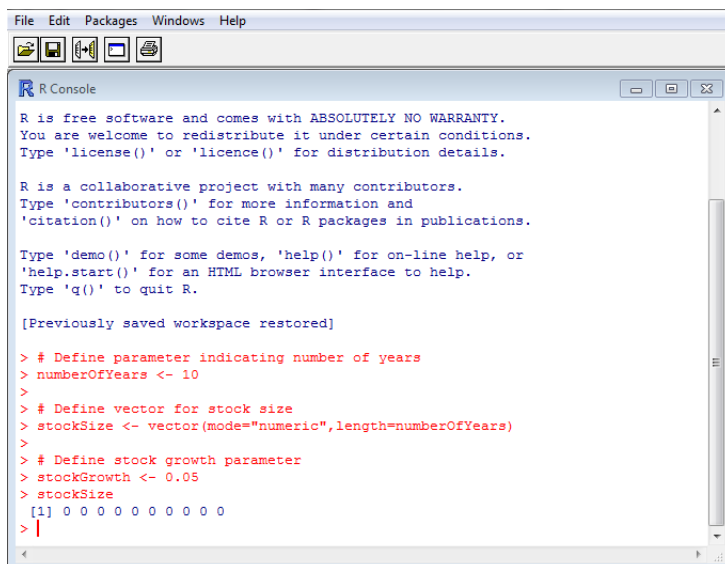
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Define parameter indicating number of years
> numberOfYears <- 10
>
> # Define vector for stock size
> stockSize <- vector(mode="numeric",length=numberOfYears)
>
> # Define stock growth parameter
> stockGrowth <- 0.05
> |
```

Figure 4: R output after running the script file

R has now added two variables and a vector to its *workspace*. The workspace is the part of your computer's memory where R stores its data, like variables, vectors, matrices, and so on. If you want to see the value of a variable, go to the R console, type in the name of the variable you want to see, and press enter. The output on the console will look like this:



```
File Edit Packages Windows Help
R Console
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Define parameter indicating number of years
> numberOfYears <- 10
>
> # Define vector for stock size
> stockSize <- vector(mode="numeric",length=numberOfYears)
>
> # Define stock growth parameter
> stockGrowth <- 0.05
> stockSize
[1] 0 0 0 0 0 0 0 0 0 0
> |
```

Figure 5: R output after checking the value of stockSize

Nevermind, for the time being, the "[1]" before the actual values. You see that `stockSize` has indeed 10 entries: they are all zeros because we have not yet calculated the values of `stockSize`. To calculate those values we first set the size of the stock in the first year to 10:

```
# Initiate stock size in first year
stockSize[1] <- 10
```

The number between the square brackets indicates the entry number of the vector. We have now set the first entry of the vector \mathbf{X} to 10.

Now that we have set the value of the stock in the first year to 10 ($X_1 = 10$) we can calculate the value of the stock in the subsequent years according to Equation (1). We do this with the following code:

```
# Start loop over years
for(iYear in 2:numberOfYears) {
  stockSize[iYear] <- stockSize[iYear-1]+
    stockGrowth*stockSize[iYear-1]
}
```

There is a lot happening in this code, so let's go over it step by step. After the comment we find the **for** command. This command is used to create a so-called *loop*: a piece of code that is repeated a number of times. In this case we want to repeat a calculation for the years 2 to 10:

$$\begin{aligned} X_2 &= X_1 + rX_1 \\ X_3 &= X_2 + rX_2 \\ &\dots \\ X_{10} &= X_9 + rX_9 \end{aligned} \tag{2}$$

Note that we already have a value for X_1 , which is why count from 2 to 10 with the expression `2:numberOfYears`. The colon ":" is used to make a sequence in R. Typing, for instance, `2:10` creates a sequence of numbers from 2 to 10:

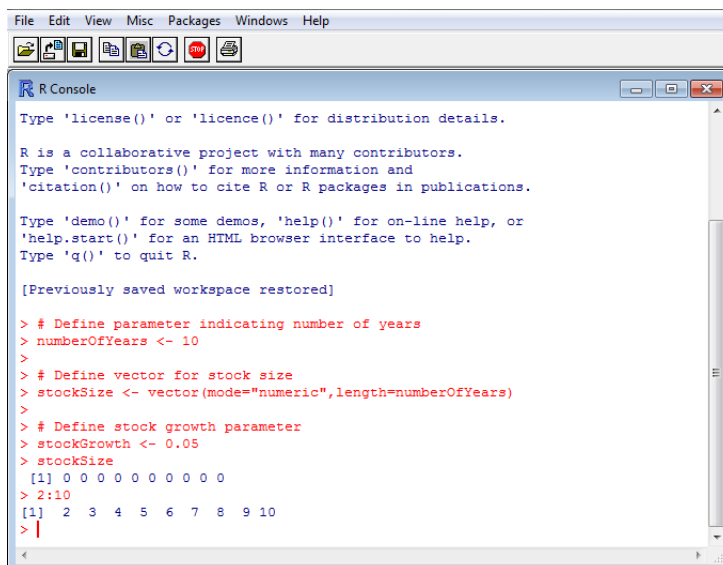


Figure 6: R output after making a sequence

So the expression `"iYear in 2:numberOfYears"` tells R *"Go through a sequence from 2 to numberOfYears, and for each value give iYear that value and do the following:"*.

Next we encounter, between accolades ("`{`" and "`}`"), the code that we want to repeat. In this case we have an expression that is based on Equation (1): `stockSize[iYear] <- stockSize[iYear-1]+stockGrowth*stockSize`

If we now run our script and check again the value of `stockSize` we get the following:

```

File Edit Packages Windows Help
R Console
[1] 0 0 0 0 0 0 0 0 0 0
> 2:10
[1] 2 3 4 5 6 7 8 9 10
> # Define parameter indicating number of years
> numberOfYears <- 10
>
> # Define vector for stock size
> stockSize <- vector(mode="numeric",length=numberOfYears)
>
> # Define stock growth parameter
> stockGrowth <- 0.05
>
> # Initiate stock size in first year
> stockSize[1] <- 10
>
> # Start loop over years
> for(iYear in 2:numberOfYears) {
+   stockSize[iYear] <- stockSize[iYear-1]+
+     stockGrowth*stockSize[iYear-1]
+ }
> stockSize
[1] 10.00000 10.50000 11.02500 11.57625 12.15506 12.76282 13.40096 14.07100
[9] 14.77455 15.51328
>

```

Figure 7: R output after running the exponential growth script

Wouldn't you just love to show that in a plot? Add the following code to your script and run the script again:

```

# Print stock size
plot(stockSize)

```

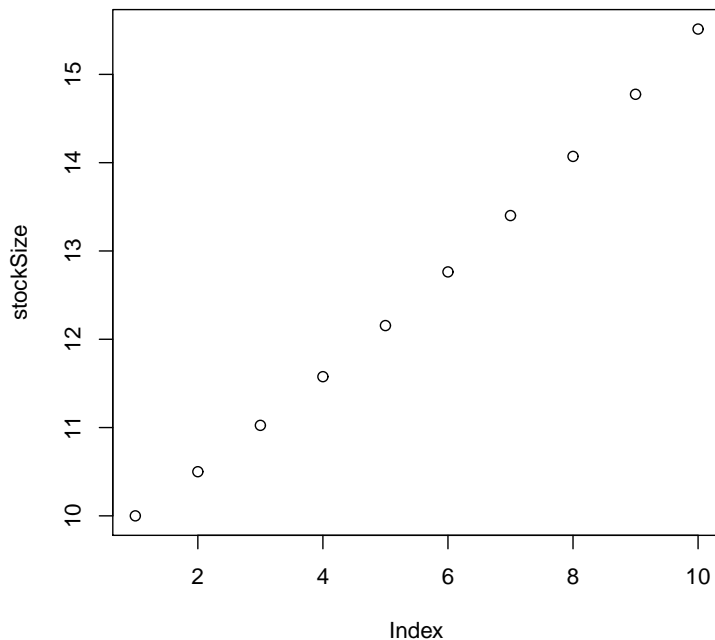


Figure 8: R plot of exponential growth

Exercise 1

In your model, set $r = 0.1$, let it run for 100 years, and plot the result.

No stock, however, can grow this way indefinitely. Eventually it will reach the maximum size that can be maintained in the ecosystem. We can easily modify Equation (1) to reflect the carrying capacity of the ecosystem:

$$X_t = X_{t-1} + rX_{t-1} \left(1 - \frac{X_{t-1}}{m}\right) \quad (3)$$

where m denotes the *carrying capacity* of the ecosystem, or the largest biomass of the species it can sustainably maintain. The growth function in Equation (3) is also called a Gordon-Schaefer growth function after the biologists who developed it.

Exercise 2

Introduce the variable m in your model and give it the value 100. Also change the growth function to the function in Equation (3) and run the script to plot this function. Don't forget to give m a descriptive variable name, and to include a comment line to explain what it is.

1.3 The impact of fishing

We now add fishing to the model:

$$X_t = X_{t-1} + rX_{t-1} \left(1 - \frac{X_{t-1}}{m}\right) - Y_{t-1} \quad (4)$$

where Y_t denotes the amount of fish taken out of the fish stock.

Exercise 3

Assume a fixed harvest $Y_1 = Y_2 = \bar{Y} = 2$ and introduce this harvest into your model. What happens if $X_1 = 10$? Don't forget to give \bar{Y} a descriptive variable name, and to include a comment line to explain what it is.

You see in Exercise 3 that the fishing pressure causes stock size to become negative. This is of course not realistic. It would be better to define the model such that if $\bar{Y} > X_t$, the stock in the following year is zero ($X_{t+1} = 0$). The simplest way to do this is to use the `max` function. A statement like `max(5,10)` returns the value 10; likewise, the statement `max(0,variable)` returns the value of `variable` if that variable is positive, but it returns the value 0 if `variable` is negative. So mathematically we want the following growth function:

$$X_{t+1} = \max \left\{ 0, X_t + rX_t \left(1 - \frac{X_t}{m}\right) - Y_t \right\} \quad (5)$$

Exercise 4

Implement Equation 5 in your model and run the model. What happens if $X_1 = 10$? And what if $X_1 = 50$?

You see in Exercise 4 that under these assumptions the outcome of the model depends on its starting point: if we harvest 2 units every year from a small stock we deplete it, but harvesting the same amount from a larger stock leads to a steady state where we can harvest this amount indefinitely.

Exercise 5

What is the difference in long-term stock if you start from $X_1 = 50$ or $X_1 = 80$?

In reality harvesting a given amount of fish will be easier from an abundant stock than from a stock that is almost depleted. For this and other reasons fisheries economists use harvest functions where the harvest depends on two variables, namely the stock size (X_t) and the *fishing effort* (E_t). The latter needs some explanation. Fishers use many inputs when they fish, such as labour, capital (like nets, fishing vessels, and so on) and fuel. For simplicity we lump these inputs together in one variable: effort. Although this

is a gross simplification, fishing effort is often indeed measured in days-at-sea, or in horsepower-days, where the number of days-at-sea is multiplied by the capacity of the vessel's motor.

A common function that relates fishing effort and fish stock to fish harvest is the following:

$$Y_t = qE_tX_t \quad (6)$$

where q is called the *catchability parameter*. It is almost the same as the catch per unit of effort (CPUE), except that CPUE varies with stock size, whereas q is assumed to be constant. Indeed, you can derive CPUE from Equation (6) as follows:

$$\text{CPUE} = \frac{Y_t}{E_t} = qX_t \quad (7)$$

You can interpret q as the fraction of the stock that you catch with one unit of effort.

Exercise 6

Introduce the parameter q and the variable \bar{E} in your R script. Assume $q = 0.1$.

As for catch, replace the (exogenous) variable \bar{Y} by a vector \mathbf{Y} . Include Equation (6) in your script to calculate Y_t for each year. Run your script for a number of different values of \bar{E} and check what happens to X_t and Y_t . Don't forget to use descriptive variable names, and to include comments.

To finally add some economics, let p be the price of fish and c the costs of fishing effort. We then define the fishing revenues B_t , fishing costs C_t , and net resource rents R_t as follows:

$$B_t = pY_t \quad (8a)$$

$$C_t = c\bar{E} \quad (8b)$$

$$R_t = B_t - C_t \quad (8c)$$

Exercise 7

Introduce p , c , B_t , C_t , and R_t in your script. Assume that $p = 5$ and $c = 10$. Calculate B_t , C_t , and R_t for each year in your model. Run your script for a number of different values of \bar{E} and check what happens to B_t , C_t , and R_t . Don't forget to use descriptive variable names, and to include comments.

Hint 1: To add another graph to an existing plot, use the command `points` (for point plots) or `lines` (for lines) instead of `plot`. Note that this only works when you have already made a plot with `plot`.

Hint 2: The dimensions of your plot depend on the data in the `plot` command. If your data in `points` or `lines` exceeds those dimensions you can also plot those data first.

1.4 Dynamics of the fishing fleet

We now add the behaviour of the fishers to our model. A very simple but plausible assumption is the following:

- If net resource rents are positive, new fishers enter the fishery;
- If net resource rents are negative, some of the current fishers leave the fishery.

A simple way to approach is to define an entry/exit parameter α that denotes the number of fishers entering or leaving the fishery per unit of net resource rents. Hence we now have a dynamic system with two state variables:

$$X_{t+1} = \max \left\{ 0, X_t + rX_t \left(1 - \frac{X_t}{m} \right) - qE_tX_t \right\} \quad (9a)$$

$$E_{t+1} = \max \{ 0, E_t + \alpha (pE_tX_t - cE_t) \} \quad (9b)$$

Exercise 8

We will now include the behaviour of the two state variables X_t and E_t in Equation (9). Instead of the fixed effort variable \bar{E} , introduce a vector \mathbf{E} . Assume $\alpha = 0.1$; $X_1 = 100$; and $E_1 = 0.2$. Calculate the fishing revenues of each year (B_t) and the fishing costs of each year (C_t), and plot these in one graph. Explain the dynamics of this system.

Hint 1: If you want to change the line type, use the option `lty` in either `plot` or `lines`: `lines(variable,lty=2)`, where 1 indicates a solid line (default); 2 indicates a dashed line; 3 indicates a dotted line; 4 indicates a dot-dashed line; 5 indicates a line with long dashes; and 6 indicates a two-dashed line.

Hint 2: Feel free to change the number of years if you want to see the model converge.