

Contents

1	Thurs Sept 10: Syllabus	3
1.1	Instructor Information	3
1.2	Course Information	3
1.3	Method of Evaluation	4
1.4	Additional Policies	5
1.5	Additional Supports	5
1.6	Tentative course schedule	6
1.7	Handing in your work	6
2	Friday Sept 11: What is a population?	7
2.1	Questions	7
2.2	Resources	7
3	Tues Sept 15: Exponential growth - discrete time	9
3.1	Required reading	9
3.2	Questions	9
4	Thurs Sept 17: Getting started with R	11
4.1	Required reading	11
4.2	Questions	11
4.3	Just for fun	11
5	Fri Sept 18: Protection Island 1	13
5.1	Questions	13
6	Tues Sept 22: Protection Island 2	15
6.1	Questions	16
7	Thurs Sept 24: Protection Island 3	17
7.1	Questions	17
8	Fri Sept 25: Exponential growth (continuous time)	23
8.1	Required reading	23
8.2	Discrete or continuous time formulations	24

8.3 Questions	24
9 Tues Sept 29: Doubling time	27
9.1 Questions	34
10 Thurs Oct 1: Density dependence and logistic growth	35
10.1 Required reading	35
10.2 Questions	35
11 Tues Oct 6: Solving the logistic growth equation using a computer 1	37
11.1 What is an ordinary differential equation?	38
11.2 Numerically integrating an ordinary differential equation in R . .	39
11.3 Writing custom functions	39
11.4 Questions	41
12 Thurs Oct 8: Solving the logistic growth equation using a computer 2	43
12.1 Writing the func argument for ode()	43
12.2 Questions	45
13 Fri Oct 9 and Thurs Oct 15: Data and the logistic equation	47
13.1 Questions	47
13.2 The data	48
13.3 The intrinsic growth rate	49
13.4 References	51
14 Tues Oct 20: Density-yield and density dependence in births versus deaths	53
15 Thurs Oct 22: Stage-structured models	55
16 Fri Oct 23: Stage-structured dynamics	57
17 Tues Oct 27: Yellow columbine	59
18 Thurs Oct 29-Fri Oct 30: Midterm [due Fri Nov 6 at 5pm]	61
19 Thurs Nov 5: Evolutionary ecology	63
20 Fri Nov 6: Evolutionary ecology	65
21 Tues Nov 10: Evolutionary ecology	67
22 Thurs Nov 12: Evolutionary ecology	69
23 Fri Nov 13: Evolutionary ecology	71

<i>CONTENTS</i>	3
24 Tues Nov 17: Evolutionary ecology	73
25 Thurs Nov 19: Evolutionary ecology	75
26 Fri Nov 20: Evolutionary ecology	77
27 Final project ideas	79
27.1 How to read scientific papers	80
27.2 Sources for datasets	80
27.3 Grading Rubric: Review	80
27.4 Grading Rubric: Analysis	80

Chapter 1

Thurs Sept 10: Syllabus

1.1 Instructor Information

Instructor: Dr. Amy Hurford

Office: Teaching remotely

Email: ahurford@mun.ca

I will try to reply to emails within 24 hours (excluding evenings, weekends and holidays). I am always available during the lecture times. Please email to request a meeting for a different time. Please check my schedule and suggest a time I am free that works for you.

1.2 Course Information

TR 12.00-12.50pm

F 1-1.50pm

WebEx links for lecture times are on the course Brightspace under Announcements.

Course description:

Population and Evolutionary Ecology is an introduction to the theory and principles of evolutionary ecology and population dynamics. Pre-requisites: BIOL 2600; at least one of BIOL 2010, 2122 or 2210.

Course format:

The course has been re-designed for online delivery. Specifically, no exams that require invigilation are part of the grading scheme because these are challenging to deliver remotely. Pre-recorded lectures limit my ability to interact with

students. Therefore, I have elected to dedicate all lecture time to interacting with students. For each class there is a list of questions you are required to answer and hand-in. Prior to some classes there may be a *Required Reading*, that if completed will allow you to answer the day's assignment questions. Prior to the day of class you should complete the *Required Reading*. In addition, I can most effectively help you if you have read over the questions ahead of time.

Course expectations:

Any students that are disruptive, violating university policies, or acting in a potentially unsafe way will be warned and asked to leave.

Learning goals:

I consider your completed assignments to be a portfolio of your knowledge in population and evolutionary ecology. You will also get some exposure to coding in R. It takes time to become proficient in a programming language, but the time you will spend coding in this class will help you towards becoming more proficient. The course content emphasizes a deeper understanding of fewer concepts. You have the opportunity to further explore a topic of interest to you for the final project.

Required Text and Resources:

The course materials are online at <https://ahurford.github.io/BIOL-3295-Fall-2020/>. In addition you will need a computer to install R and RStudio. This will be covered on Thursday Sept 17 (see Chapter 4). Class announcements and WebEx links will be provided on the course BrightSpace and your assignments are to be submitted to BrightSpace.

1.3 Method of Evaluation

- 27 assignments (equal weighting) - 50%
- Midterm (due Fri Nov 6 at 5pm) - 15%
- Final Project (due Monday Dec 14 at 9am) - 35%

You should aim to complete each assignment before the next class, but assignments will be accepted, without penalty, up to a week later.

Late assignments, labs, and missed midterms, and final exams will be accommodated as described by University Regulation 6.7.3 and 6.7.5 (see <https://www.mun.ca/regoff/calendar/sectionNo=REGS-0474> for Regulations).

1.4 Additional Policies

1.4.1 Accommodation of students with disabilities

Memorial University of Newfoundland is committed to supporting inclusive education based on the principles of equity, accessibility and collaboration. Accommodations are provided within the scope of the University Policies for the Accommodations for Students with Disabilities see www.mun.ca/policy/site/policy.php?id=239. Students who may need an academic accommodation are asked to initiate the request with the Glenn Roy Blundon Centre at the earliest opportunity (see www.mun.ca/blundon for more information).

1.4.2 Academic misconduct

Students are expected to adhere to those principles, which constitute proper academic conduct. A student has the responsibility to know which actions, as described under Academic Offences in the University Regulations, could be construed as dishonest or improper. Students found guilty of an academic offence may be subject to a number of penalties commensurate with the offence including reprimand, reduction of grade, probation, suspension or expulsion from the University. For more information regarding this policy, students should refer to University Regulation 6.12.

1.4.3 Equity and Diversity

A safe learning environment will be provided for all students regardless of race, colour, nationality, ethnic origin, social origin, religious creed, religion, age, disability, disfigurement, sex (including pregnancy), sexual orientation, gender identity, gender expression, marital status, family status, source of income or political opinion.

You should not photograph or record myself, teaching assistants, or other students in the class without first obtaining permission. Accommodation will be made for students with special needs.

The sound should be turned off on phones and computers during class.

1.5 Additional Supports

Resources for additional support can be found at:

- www.mun.ca/currentstudents/student/
- <https://munsu.ca/resource-centres/>

1.6 Tentative course schedule

The course schedule is found in the toolbar of the class materials, see <https://ahurford.github.io/BIOL-3295-Fall-2020/>.

The last day to drop the course without academic prejudice is Wednesday Nov. 4.

1.7 Handing in your work

1.7.1 Making figures to hand-in

The graphs you hand in need to have descriptive axes and a figure caption. You may put these elements together using a word processing software such as *Microsoft Word*. Elements of a good figure caption:

- Has a label, i.e., “Figure 1”,
- The first sentences provides a summary of what the figure shows, i.e., “The price of oranges has increased steadily since 1964”,
- Provide all necessary information to understand everything in the figure, i.e., if the figure has no legend, but multiple line types/symbols, be sure to indicate what is represented by the different symbols. If the axes labels are overly brief due to space constraints in the graph, provide a more thorough description in the figure caption. If any assumptions have been made in making the figure, disclose these, i.e., a point that was excluded from the analysis due to being considered an outlier.

1.7.2 Writing R scripts to hand-in

To write your own R scripts follow the guidelines described in Chapter 7 Best Practices of *Quantitative training in Biology*. If you are asked to hand in your R script this means you need to submit an `.R` file on Brightspace.

Chapter 2

Friday Sept 11: What is a population?

For the questions below, submit your answers to Brightspace ideally before the next class. The deadline to submit your answers is Friday Sept 18.

The *Resources* below are sufficient to answer all the questions, however, you are encouraged to find your own textbooks or peer-reviewed articles to answer the questions if you feel comfortable. You can search for textbooks using and the library catalogue and peer-reviewed articles using Web of Science.

2.1 Questions

1. Give a definition of a population from a textbook or peer-reviewed publication. Provide the citation. [2 marks]
2. Find a peer-reviewed paper where a population is studied. Write 1 paragraph discussing how a population is defined for the study and how this compares to your definition of a population given in Question 1. [5 marks]

2.2 Resources

Vandermeer, J.H., Goldberg, D.E., 2013. Population Ecology: First Principles (Second Edition). Princeton University Press, Princeton, United States. [Link](#)

The Princeton Guide to Ecology, edited by Simon A. Levin, et al., Princeton University Press, 2009. ProQuest Ebook Central, [Link](#)

Sacchi, R., Gentili, A., Razzetti, E., Barbieri, F., 2002. Effects of building features on density and flock distribution of feral pigeons *Columba livia* var. *domestica* in an urban environment. *Can. J. Zool.* 80, 48-54. [Link](#)

Chapter 3

Tues Sept 15: Exponential growth - discrete time

Please submit your answers before the next class. You have until Tues Sept 22 to submit your answers.

3.1 Required reading

Vandermeer, J.H., Goldberg, D.E., 2013. Population Ecology: First Principles (Second Edition). Princeton University Press, Princeton, United States. p1-3. [Link](#)

3.2 Questions

1. Suppose $\lambda = 5$ in equation (3) (see the required reading). Explain in 1-2 sentences the meaning of $\lambda = 5$. [1 mark]
2. Suppose the number of lilypads during week 7 is 78,125. Let $\lambda = 5$, and assume that the units of t are weeks. Use equation (3) to calculate the number of lilypads in week 8. Show your calculations. [2 marks]
3. Use your answer to question 2. to calculate the number of lilypads in week 9. Show your calculations. [2 marks]
4. Equation (4) for the required reading assumes that $N_0 = 1$, however, this formula can be generalized such that

$$N_t = N_0 \lambda^t$$

where N_t is the population size at time, t . Define time such that $t = 0$ is week 7 and t is then the number of weeks since week 7. Use the equation above to answer question 3 and confirm that the answer is the same (i.e., find the population size for week 9, when $\lambda = 5$ and the population size for week 7 is 78,125). Show your calculations. [2 marks]

5. Use the formula from question 4 to find the population size for week 15, where $N_0 = 1$ and $\lambda = 5$. Define time as the number of weeks since week 0. Show your calculations [2 marks]
6. It is important to note that all mathematical formulas should have the same units on both sides of the equals sign, and for each term that is added or subtracted. The units of the population size, N_t , at time, t , are number. The geometric growth rate, λ is unitless. Choose an equation from the required reading and give the units for each of the terms to show that both sides of the equals have the same units. For example, for the equation that appears in question 4, we have:

$$\begin{aligned} N_t &= N_0 \lambda^t \\ (\text{number}) &= (\text{number})(\text{unitless})^{\text{weeks}} \\ (\text{number}) &= (\text{number}) \end{aligned}$$

Note that:

$$\begin{aligned} (\text{unitless}) \times (\text{quantity with units}) &= (\text{quantity with units}) \\ (\text{unitless})^{(\text{quantity})} &= (\text{unitless}) \end{aligned}$$

[2 marks]

7. Although not stated in the reading, $\lambda = 1 + b - d$ where b is the per capita birth rate over one time step (i.e. one week for this example), and d is the fraction of the lilypad population that dies over one time step. The number 1 is considered unitless, what must the units of b and d be? [1 mark]
8. In the reading, $\lambda = 2$. Given that $\lambda = 1 + b - d$, what are some possible values of b and d . Note that d is a fraction and must be $1 \geq d \geq 0$ and $b > 0$. [1 mark]
9. [True or False] For discrete time exponential growth (as per the reading), the change in population size from one week to the next depends not so much on the per capita birth rate, but on the difference between the per capita birth rate and the per capita death rate. [1 mark]

Chapter 4

Thurs Sept 17: Getting started with R

You need to have the **R** and **RStudio** softwares installed before you can proceed with the next class. Your answer to the question is due Thursday Sept 24.

4.1 Required reading

You are required to read and complete all the exercises in Chapters 1 *Introduction*, 3 *R and RStudio*, and 4 *Finding your way around RStudio* of:

Quantitative skills for biology <https://ahurford.github.io/quant-guide-all-courses/>

When you are finished you should have **R** and **RStudio** installed on your computer, or you should be familiar with running **RStudio Cloud**.

4.2 Questions

1. Write 1 paragraph describing your experience completing the the exercises. [5 marks. You will receive full marks for attempting the question and submitting an answer.]

4.3 Just for fun

Type into the R Console:

```
install.packages("praise")  
require("praise")  
praise()
```

Chapter 5

Fri Sept 18: Protection Island 1

Today's exercise will be challenging. You should aim to complete this exercise before the next class. This exercise is due by Fri Sept 25. There will be a lot of carryover to the next exercise.

The information below is taken from the following source: Newcomb, HR. 1940. Ring-necked pheasant studies on Protection Island in the Strait of Juan de Fuca, Washington. MS thesis. Oregon State University.

- a. Pheasant chicks are born during the summer.
- b. In May 1937, 10 pheasants were introduced to the island. Before the next breeding season there were 35 pheasants.
- c. November 10, 1938 a census estimated 110 pheasants.
- d. October 13, 1939 a census estimated 400 pheasants.

5.1 Questions

1. Read and complete all the exercises in Chapters 4.3 *Variables and assignment* to 4.10 *R packages* and 9 *Making graphs in R* of *Quantitative skills for biology*
 - Answer all questions marked HAND IN in the reading [5 marks]
2. To make a graph of the data listed in b.-d., we need to learn how to work with dates. We will consider two possible approaches:
 - i. Use a built-in R function to convert dates to a format that can be plotted (today's class); and

- ii. Convert the dates to number of days since a reference date. Now the dates are numbers and these values can be plotted on the x-axis of a graph (next class).

In this question, we will proceed with option i. The function we will use is `as.Date()`. You can learn how to use this function using an internet search or by typing the following into your **Console**:

```
?as.Date
```

These files can be difficult to understand (see R Help files. A good way to proceed is to experiment with the function in the **Console**. Try these:

```
as.Date(2012-01-31, format = %Y-%m-%d)
as.Date("2012-01-31", format = "%Y-%m-%d")
```

Note that only the second command is error-free. The first command fails because the date argument for the `as.Date()` function must be a character string, i.e., must be enclosed in `" "` (see `?character`).

It is also possible to omit the format argument and just code: `as.Date("2012-01-31")`. The help file notes that when the format argument is not specified, that formats will be tried one by one and an error will be returned if none work. It is advisable to specify the format, as allowing the function to infer the format could introduce errors.

Chapter 6.9 Data structures describes how to make a vector (note a vector is a list of numbers rather than just a single number). We need to make a vector of the dates so that we can make our plot. For example,

```
x <- as.Date(c("2012-01-31", "2012-03-05", "2013-01-11"), format = "%Y-%m-%d")
```

Having completed Chapter 11 Making graphs in R, and having learned how to work with dates, you should now be able to write an R script to make plot using the information in b.-d. above.

HAND IN

- A graph and figure caption. The graph should have dates on the x-axis and the pheasant population size on the y-axis drawing from the information provided in b.-d. You will need to guess the date of ‘before the breeding season’ as stated in b. and you should disclose the value of this guess in the figure caption. See 1.7.1 for more information. The solutions to this problem have figure caption that is 2 sentences long. *[Note that the graph will likely have years, not months on the x-axis - changing this is a quite fiddle-y and not worth it at this stage of your R journey]* [10 marks]
- An R Script that produces the figure described above. See 1.7.2 for more information. [5 marks]

Chapter 6

Tues Sept 22: Protection Island 2

Continuing from last week, today we will try approach ii. to make the graph of the number of pheasants on Protection Island as it changes with time. Under approach ii. we will work with the dates by converting them to the number of days since a reference date. To do this we will use the `julian()` function, which is part of the `chron` package.

Today's questions are due by Tues Sept 29, however, ideally you will have them completed for the next class.

Read Section 4.4 of Quantitative skills for biology regarding installing packages. Install the package `chron` using either the **Install** button on the **Packages** tab, or by using the command `install.packages("chron")` in the **Console** window. Note that the package is only available for use once you check the box on the **Packages** tab or by running the following command in the **Console**:

```
require("chron")
```

After the `chron` package is loaded, we can then query the `julian` function,

```
?julian
```

or use an internet search to better understand how to use it. As the help files can be difficult to understand, another approach to is to try out the function. Try the following:

```
julian(1,1,1970)
julian(1,2,1970)
julian(2,1,1970)
julian(1,1,1971)
julian(1,1,1969)
```

Which argument position of the `julian()` function corresponds to the month? Note also that by default the origin (the origin is when the returned value is 0) is set to January 1, 1970. Experiment by running the following lines of code:

```
julian(1,1,2000)
julian(1,1,2000, origin = c(1,1,1970))
julian(1,1,2000, origin = c(1,1,2000))
```

Finally, we need to make our figure. Recall that the `plot` function requires vectors of equal length for the x- and y-axes. Make a vector of the days since a reference date as follows:

```
ref.day = c(1,1,2000)
x = c(julian(1,1,2000, origin = ref.day), julian(1,1,2002, origin = ref.day))
```

If you run into problems you can query the value of `x` in your console, and you can use `length(x)` to check the length of `x`.

6.1 Questions

1. You need to hand in a graph with descriptive axes and with a figure caption. The y-axis on your graph is population size and the x-axis will be created using the `julian()` function. Be sure to label the x-axis differently than you did for the previous assignment. See 1.7.1 for more information. The figure caption in the solutions for this problem is 2 sentences long. [10 marks]
2. You also need to produce an R Script that makes the figure described above. See 1.7.2 for more information. Save this file as *protection-island.R* [5 marks]

Chapter 7

Thurs Sept 24: Protection Island 3

DUE DATE: Thurs Oct 1

Here is some additional information also taken from: Newcomb, HR. 1940. Ring-necked pheasant studies on Protection Island in the Strait of Juan de Fuca, Washington. MS thesis. Oregon State University.

- a. Pheasant chicks are born during the summer.
- b. In May 1937, 10 pheasants were introduced to the island. Before the next breeding season there were 35.
- c. November 10, 1938 a census estimated 110 pheasants.
- d. October 13, 1939 a census estimated 400 pheasants.
- e. Between the 1938 and 1939 censuses, Newcomb observed that 17 adult birds died.
- f. During the 1938 nesting season there were 5.86 eggs/nest. 83.57% of eggs hatched.
- g. During the 1939 nesting season there were 8.73 eggs/nest. 64.58% hatched.
- h. During the 1939 nesting season: Average number of chicks per clutch was 6.93.¹
- i. You can assume the sex ratio is 50:50 male to female. Pheasants are a sexually reproducing species.

¹ Note that g. and h. appear to be contradictory.

7.1 Questions

1. Let d be the fraction of population that dies each year. Estimate d for the ring-tailed pheasant population on Protection Island? Write down any

assumptions you have made. [3 marks]

2. b is the per capita number of births each year. What is the value of b ? Write down any assumptions you have made. [3 marks]
3. Recall that $\lambda = 1 + b - d$. What is the value of λ ? Is this population expected to grow over time? [2 marks]
4. Lets assume that the pheasant population on Protection Island grows geometrically (i.e. exponentially but for a discrete time model) where the geometric growth rate, λ , is the value that you estimated in 3. Lets predict the population size each May beginning with May 1937. Let $N_0 = 10$ and let t be the number of years since May 1937. Recall that when a population grows geometrically,

$$N_t = N_0 \lambda^t$$

You can use R to do this calculation as follows (you should use your value of λ from question 5, rather than `lambda <- 3` as in the example below):

```
t <-1
N0 <-10
lambda <-3
N0*lambda^t
```

The result of `N0*lambda^t` is N_{t+1} , and with $t = 1$ then $N_{t+1} = N_2$: the population size two years after May 1937. You can change the value of t and repeat the calculation. Unless you have cleared your workspace it won't be necessary to re-input $N_0 = 10$ and $\lambda = 3$. As such, you can calculate N_3 with the following commands:

```
t <-2
N0*lambda^t
```

HAND IN

Use R to predict the value of the pheasant population size every year up until May 1940. You only need to hand in the values that you get, not an R Script. [2 marks]

The approach to calculating the pheasant population size in Question 4 is not very organized. In this question, we will learn how to make a data frame, use a for loop, and use the function `rbind()`.

Read Data structures in *Quantitative training for biology*.

Create a one row dataframe called `df`:

```
df <- data.frame(time = 0, popn.size = 10)
```

Query `df` in your **Console** to see the data frame you have created. We would like to add successive values of the population size that we calculate to the data frame. To do this we use the `rbind()` function, which binds rows together.

```
new.result <- data.frame(time = 1, popn.size = 20)
df <- rbind(df, new.result)
```

Here the `rbind()` function takes the `df` dataframe and adds the `new.result` data frame as a new row onto the bottom. Note that the code above *overwrites* the value of `df`: that is, `new.result` is added to the bottom of the `df` dataframe (containing only one row), and the result is called `df` (which now has two rows), and the old dataframe `df` (with one row) is overwritten. As such, each time you run the command `df <- rbind(df, new.result)` another row is added to `df`. Try the following:

```
new.result <- data.frame(time = 1, popn.size = 20)
df <- rbind(df, new.result)
df <- rbind(df, new.result)
df <- rbind(df, new.result)
```

If you query the value of `df` you can see that the several rows, all with identical values have been added because we have run the command `df <- rbind(df, new.result)` multiple times while the value of `new.result` is unchanged. Now let's change the value of `new.result` between each time we run the `df <- rbind(df, new.result)` command.

```
new.result <- data.frame(time = 1, popn.size = 20)
df <- rbind(df, new.result)
new.result <- data.frame(time = 2, popn.size = 30)
df <- rbind(df, new.result)
```

Type `df` into the **Console** to see the resulting dataframe. Finally, when we do calculations for a sequence of values, it is easier to code this using a `for` loop.

```
lambda <- 1.2
NO <- 10
df <- data.frame(time = 0, popn.size = 10)
for(t in seq(1,4,1)){
  val <- NO*lambda^t
  new.result <- data.frame(time = t, popn.size = val)
  df <- rbind(df, new.result)
}
```

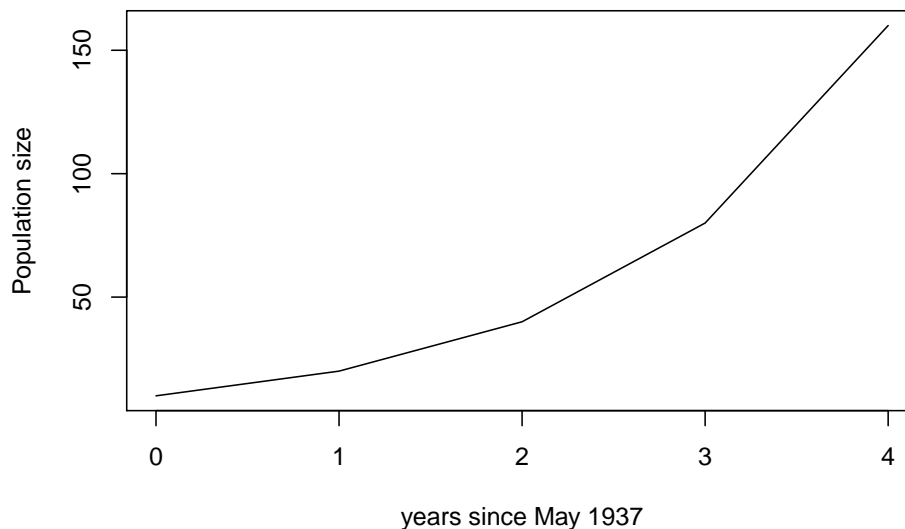
To understand the above code, after copy and pasting it into your **Console**, and running it by clicking **Return**, query the value of `df`: you should see predicted population sizes up until 4 years after May 1937. Now, let's try to understand `seq(1,4,1)`. Let's learn about the `seq()` function by trying it out in the **Console**. What is the result of each of these?

```
seq(-10,10)
seq(-10,5,0.1)
```

The `for` loop works by beginning with `t` equal to the first value of the sequence and stepping through each value until the final value. The code is written so that quantities that depend on `t` are inside the `for` loop (i.e., enclosed with in the `{}` and those that do not depend on `t` are outside the `for` loop). Note that `val` changes for different values of `t`, `new.result` changes for different values of `t` (because `new.result` has `time = t` and `pop.size = val`, where `val` depends on `t`). Finally, `df` also depends on `t`, because `new.result` depends on `t`. In contrast, `N0` and `lambda` do not change with `t`, so it is more efficient to place the allocated values for these parameters outside of the loop.

We can also plot the results of our calculations:

```
lambda <- 2
N0 <- 10
df <- data.frame(time = 0, popn.size = 10)
for(t in seq(1,4,1)){
  val <- N0*lambda^t
  new.result <- data.frame(time = t, popn.size = val)
  df <- rbind(df, new.result)
}
plot(df$time, df$popn.size, typ = "l", xlab = "years since May 1937", ylab = "Population size")
```



If you already have an existing plot you can add new lines using `lines()`. For example,

```
plot(seq(1,4), c(1,3,4,2), ylab = "y-axis", xlab = "x-axis")
lines(seq(1,4), seq(1,4))
```



HAND IN

5. Write an R script that builds on the file you have previously made *protection-island.R*. Use the `lines()` command to add the predicted population size assuming geometric growth using the commands described in this section. If you have written the code correctly the result should look something like this:



Note that to get years on the x-axis for data from the *protection-island.R* file, I needed to plot `plot(x/365,y)` because with `plot(x,y)` the units on my x-axis were days. Your graph may look slightly different since you may have calculated a slightly different λ value if you made different assumptions to me in questions

1 and 2.

You are to hand in your R Script. See 1.7.2 for more information. [10 marks]

Chapter 8

Fri Sept 25: Exponential growth (continuous time)

DUE DATE: Tues Oct 6

8.1 Required reading

Vandermeer, J.H., Goldberg, D.E., 2013. Population Ecology: First Principles (Second Edition). Princeton University Press, Princeton, United States. p4-8. [Link](#)

We now have two ways of describing how population size changes with time whereby each individual has the same average number of offspring per unit time and the same probability of dying.

- 1) Discrete time geometric growth:

$$N_t = N_0 \lambda^t \quad (8.1)$$

and,

- 2) Continuous time exponential growth:

$$N(t) = N(0)e^{rt} \quad (8.2)$$

Notably, for both these models the per capita birth and death rates do not change over time, and do not change with density or age.

The notation N_t and $N(t)$ is conventional for discrete time versus continuous time formulations respectively, however, these notations both mean the same: the population size at a particular time, t . When $t = 0$ we have the population size at time 0: N_0 or $N(0)$.

8.2 Discrete or continuous time formulations

Whether it is appropriate to model a population in a discrete time or continuous time formulation depends on whether births and deaths are overlapping or neatly partitioned into a distinct time period. For example, for many animals there is a distinct breeding season: a short proportion of the year when offspring are born. As such, there is very little temporal overlap between the times of year when births and deaths occur. Humans are an example of a species that might reasonably be modelled as continuous time because babies are born year round.

8.3 Questions

Please submit the answer to these questions to brightspace, ideally for the next class, but you have until Sept 30 to complete them without penalty.

1. In equation (8.2), what is e ? [1 mark]
2. For what values of r does the population size increase over time? [1 mark]
3. As described in the reading, b is a per capita birth rate, and d is a per capita death rate. For continuous time exponential growth, both b and d must be non-negative and can take values bigger than 1. Note that this differs from the discrete time model formulation where $0 \leq d \leq 1$. When $d > 1$ in the continuous time formulation, this means that the average lifespan is less than one time step. For example, when $d = 2$ this means that the average life expectancy for an individual is $1/2$ a time step (i.e., days or year, however, the time unit is defined in the model). When the population size increases over time, what is true of b relative to d ? [1 mark]
4. For what value of r does $N(t)$ not change over time? Hint: if $N(t)$ is not changing then $N(t) = N(0)$ for all t . [1 mark]
5. Consider the equation:

$$\frac{dN(t)}{dt} = rN(t).$$

As described in the reading, this is an alternative way to write the continuous time exponential growth equation. The quantity $\frac{dN(t)}{dt}$ can be understood as the slope of a graph where population size is on the vertical axis and time is on the horizontal axis. As such, if the slope is zero, $\frac{dN(t)}{dt} = 0$, then the population size is not changing. If $\frac{dN(t)}{dt} < 0$, then the



Figure 8.1: An example of a very well thought out data visualization where the data might demonstrate exponential growth

population size is decreasing. For what value of r does the population size decrease? What is true about b relative to d in this instance? [2 marks]

6. Which population would be more appropriate to be modelled as a continuous time formulation: *E. coli* bacteria or moose? [1 mark]
7. Calculate the formula for the doubling time for continuous time exponential growth (equation (8.2)). This is the time for the population to double in size. The value of $N(0)$, the population size at $t = 0$ doesn't matter as long as it is a positive number. When the population has doubled, $N(t) = 2N(0)$. To answer this question you need to find t such that $N(t) = 2N(0)$. You may need to revisit some rules about working with logarithms to complete this question (i.e. see here, specifically the *Product*, *Quotient*, *Power*, and *Root* table. Also recall, $\ln(e^x) = x$). Please show your work. [2 marks]
8. Consider the following plot:

In this question, we will find the equation for 'doubles every day'. Our model is equation (8.2). This means that we will calculate the equation for 'doubles every day' under the assumption that cumulative COVID-19 deaths grow exponentially over time.

To begin, we choose the time step for our model as days. The first problem we should consider is what is the value of r if the population doubles every 1 day? Make the appropriate substitutions and re-arrange equation (8.2) to find the value of r .

Now, note that the y-axis on the above graph is on a logarithmic scale. We will assume this is a base e logarithmic scale. Take the natural logarithm of both sides of equation (8.2). Note that the result should be the formula for a straight line, i.e., $y = mx + c$, but where y is $\ln(N(t))$ and x is t (to match the axes of Figure 8.1). The questions we are left to solve are: what is the slope of the line (i.e., m , or the value of the coefficient that multiplies t), and what is the value of the intercept (i.e., c or the remaining terms on the opposite side of the equals to $\ln(N(t))$ and which are not multiplied by t)?

Show all your work. Your answer is complete when you provide the numerical values of the slope and the intercept for the straight line corresponding to a doubling time of 1 day. Assume $N(0) = 10$. Your answers must be numbers - you will need to substitute the value of r that you calculated in the early part of this question. [2 marks]

9. Provide the values of the slope and the intercept for doubling times of 2, 3, and 7 days. Again assume $N(0) = 10$. You do not need to show your work. [2 marks]

Chapter 9

Tues Sept 29: Doubling time

DUE DATE: Thurs Oct 8

Today we will make a graph of the current COVID-19 data for Ontario. When only a small fraction of the population is resistant to an infection, the number of daily cases, in absence of any interventions such as physical distancing, is expected to grow exponentially over time. At the end of this exercise you will have calculated the doubling time for the second wave of COVID-19 in Ontario.

Visit the website: <https://art-bd.shinyapps.io/covid19canada/>, this site is the *front end* for the data we will use which are archived here (also referred to as the *back end*). These data can easily be pulled into R using this command:

```
COVID.data <- data.table::fread('https://raw.githubusercontent.com/ishaberry/Covid19Canada/master
```

However, first you will need to install the `data.table` package:

```
install.packages("data.table")
```

When asked: *Do you want to install from sources the package which needs compilation? (Yes/no/cancel)* enter `no`.

To view the data type:

```
head(COVID.data)
```

##	province	date_active	cumulative_cases	cumulative_recovered	cumulative_deaths
## 1:	Alberta	25-01-2020	0	0	0
## 2:	Alberta	26-01-2020	0	0	0
## 3:	Alberta	27-01-2020	0	0	0
## 4:	Alberta	28-01-2020	0	0	0
## 5:	Alberta	29-01-2020	0	0	0

```
## 6:  Alberta  30-01-2020          0          0          0
##      active_cases active_cases_change
## 1:           0           0
## 2:           0           0
## 3:           0           0
## 4:           0           0
## 5:           0           0
## 6:           0           0
```

`head()` is a command that shows the first 6 lines of the quantity inside the round brackets. `COVID.data` is a very large table of data. If I type `COVID.data` into the Console I will get more information than I want, so I instead use `head()`.

So far we have pulled data from an online repository. However, it is good practice to save these data in case later the repository is removed. We can make copy of the data for our records using the following commands:

Note that the above contains the path to folders on my computer and saves `COVID.data` as the file *COVID-data-save.csv*. You will need to replace the path above with correct path to the folder on your own computer where you want to save these data. Setting the path is computer-specific, and so we always have problems with this in class because the solutions vary by student. Some tricks that work for me are:

- type `getwd()` into the Console. This will tell you the current working directory and can give you clues on the format of the path for your computer.
- clicking the **Source** button on a .R file that you have made will print the path to that .R file into the Console.
- elect to save the file somewhere very simple, for example, on your Desktop and then move it to a more organized place later.

Navigate to the folder where you expected to save the data as a .csv file. You should be able to open the .csv file in *Microsoft Excel*. [THIS PART DOESN'T WORK - SORRY - JUST SKIP]

Now, the data table `COVID.data` contains data for all provinces and we would like to *subset* the data so that only the information pertaining to Ontario is left. We do this as follows:

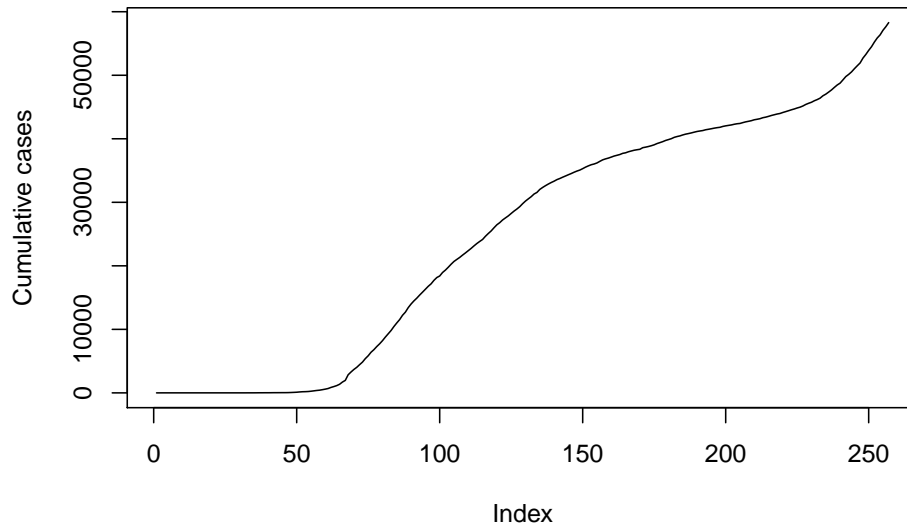
```
data.ON <- COVID.data[province == "Ontario"]
```

We should understand what variables are contained in our data set. Query the following in your Console:

```
colnames(data.ON)
```

We will plot confirmed positive cases over time:

```
plot(data.ON$cumulative_cases, ylab = "Cumulative cases", typ = "l")
```



Note that we haven't provided an x-axis argument above. The x-axis should be `date_active`, however, from our previous exercise (Chapter 6), we recognize that dates need to be reformatted to be an appropriate x-axis.

```
head(data.ON$date_active)
```

```
## [1] "25-01-2020" "26-01-2020" "27-01-2020" "28-01-2020" "29-01-2020"
## [6] "30-01-2020"
```

The earliest date is January 25, 2020. We will use this as the origin and calculate the number of days since that time.

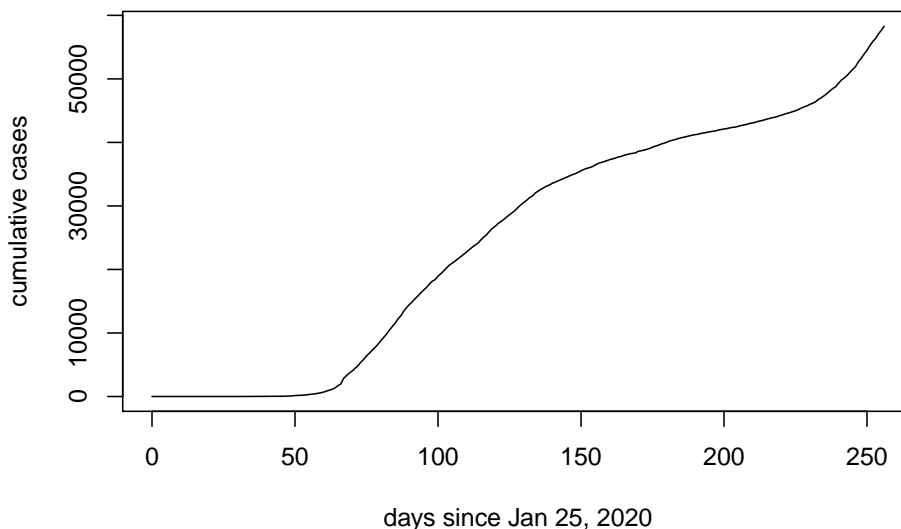
```
require(chron)
days.since <- julian(as.Date(data.ON$date_active, format = "%d-%m-%Y"), origin=as.Date("25-01-2020"))
```

If you get an error this may be because you have not installed the `chron` package. You can do so as `install.packages("chron")`.

On my first attempt at the above code, I tried `julian(data.ON$date_active)`. This generated an error because our dates from the COVID-19 dataset are formatted as a character string "01-25-2020", but the `julian()` function is expecting three numerical arguments `julian(m,d,y)`. As such, `julian(data.ON$Date)` fails to provide three numerical arguments to the function and instead provides a single character string. For this reason the function `as.Date()` needs to be applied to `data.ON$Date` inside the `julian()` function.

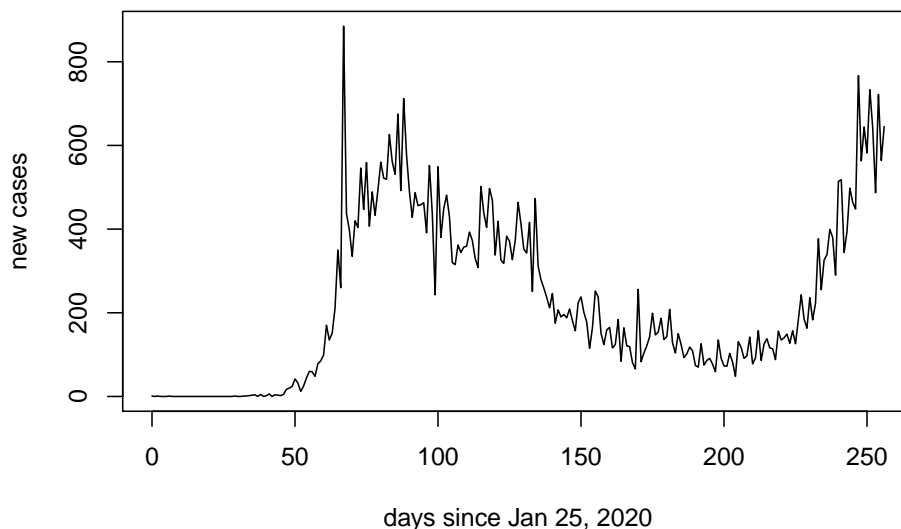
We now have a quantity `days.since` that is appropriate for the x-axis of our graph:

```
plot(days.since, data.ON$cumulative_cases, ylab = "cumulative cases", xlab = "days since Jan 25, 2020", type = "l")
```



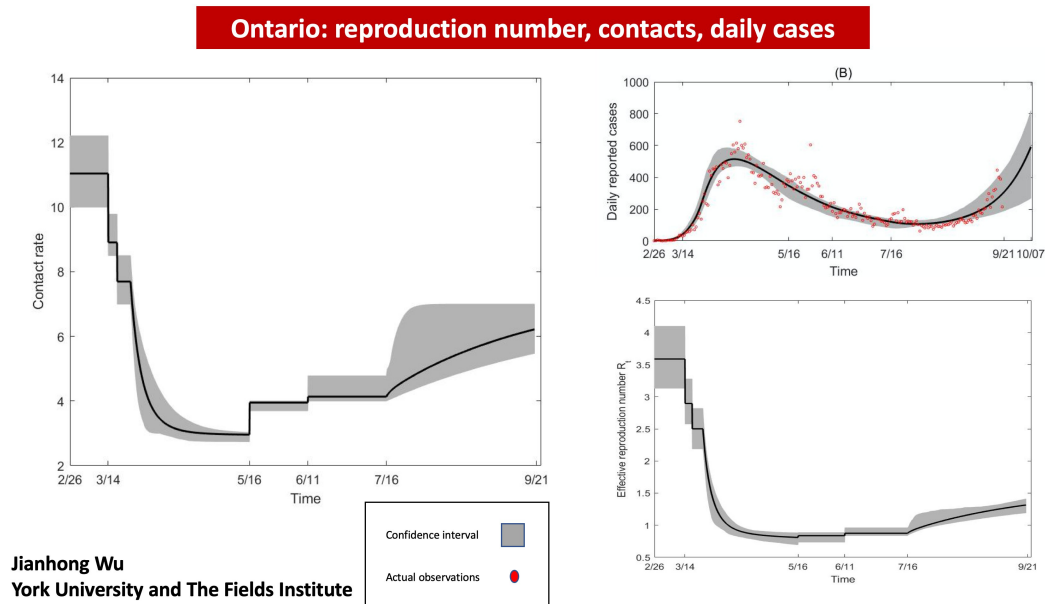
However, rather than calculating cumulative cases suppose that we might like to plot daily new cases. We do this by subtracting the cumulative cases for the previous day. The function `diff()` subtracts the element previous to the entry of a list and because the first element of the list has not preceding value we add in a 0.

```
new.cases <- diff(c(0, data.ON$cumulative_cases))
plot(days.since, new.cases, ylab = "new cases", xlab = "days since Jan 25, 2020", type = "l")
```



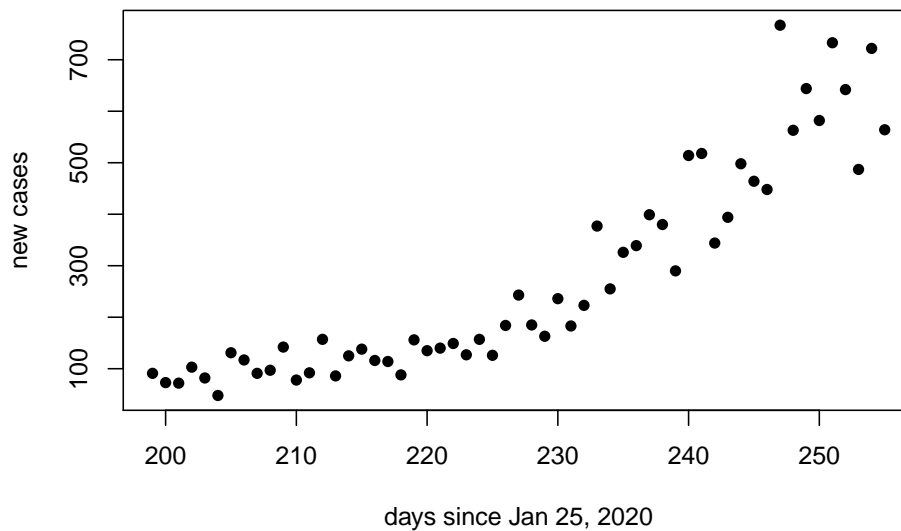
If we have done a good job of our work, it should be consistent with the official

graph labelled as (B) below:



We can also focus in on just the second wave by adjusting the `xlim` argument:

```
plot(days.since[200:max(days.since)], new.cases[200:max(days.since)], ylab = "new cases", xlab =
```



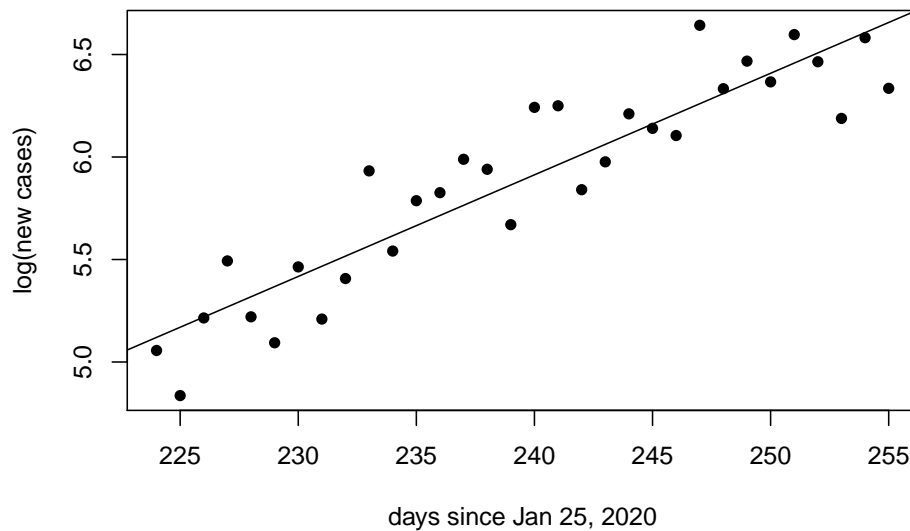
Note that the second wave of COVID-19 in Ontario appears to be consistent with exponential growth. Recall that when the y-axis of the graph is on a logarithmic scale, exponential growth will be represented as a straight line.



Figure 9.1: Straight lines illustrate exponential growth with different doubling times when the y-axis is plotted on a logarithmic scale

We can estimate the doubling time of the number of new cases for the second wave of COVID-19 in Ontario by performing a linear regression on the natural logarithm of the number of new cases. Below, we consider only the data 224 days after January 25 and onwards. The function `lm()` performs the linear regression and `abline()` plots the regression.

```
since.225 <- days.since[225:max(days.since)]
log.cases.225 <- log(new.cases[225:max(days.since)])
plot(since.225, log.cases.225, ylab = "log(new cases)", xlab = "days since Jan 25, 2020")
reg <- lm(log.cases.225~since.225)
abline(reg)
```



```
summary(reg)
```

```
##
## Call:
## lm(formula = log.cases.225 ~ since.225)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36897 -0.10648 -0.02282  0.12287  0.38261
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.981132   0.944973  -6.329 5.57e-07 ***
## since.225    0.049559   0.003943  12.570 1.74e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2059 on 30 degrees of freedom
## Multiple R-squared:  0.8404, Adjusted R-squared:  0.8351
## F-statistic: 158 on 1 and 30 DF, p-value: 1.735e-13
```

From the summary of the linear regression we are interested in the slope, because in the previous class we estimated the relationship between this slope and the doubling time.

```

Call:
lm(formula = log.cases.225 ~ since.225)

Residuals:
    Min       1Q   Median       3Q      Max
-0.34767 -0.12710  0.03244  0.10027  0.33219

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.612982    2.026825  -5.236 8.15e-05 ***
since.225     0.069584    0.008715   7.984 5.69e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1918 on 16 degrees of freedom
Multiple R-squared:  0.7994,    Adjusted R-squared:  0.7868
F-statistic: 63.75 on 1 and 16 DF,  p-value: 5.691e-07

```

Figure 9.2: The above output shows the slope of the regression as 0.0695 log cases per day

9.1 Questions

1. How many observations and how many variables are there in the COVID.data and the data.ON files? (see the *Environment* tab) [1 mark]
2. What command did you use to save the data as a .csv file? - IGNORE THIS QUESTION - INSTRUCTIONS WERE BAD [1 mark]
3. What do you estimate the doubling time of the second wave of COVID-19 in Ontario to be? Show your calculations [3 marks]
4. Write an R Script that performs the linear regression as illustrated above. Your script must make the figure including the regression line [7 marks]

Chapter 10

Thurs Oct 1: Density dependence and logistic growth

DUE DATE: Friday Oct 9

10.1 Required reading

Vandermeer, J.H., Goldberg, D.E., 2013. Population Ecology: First Principles (Second Edition). Princeton University Press, Princeton, United States. p9-17.
[Link](#)

10.2 Questions

1. What is the equation for continuous time logistic growth in its *classic form*? Define all the symbols in the equation by writing their meanings in words. Can K be negative? [3 marks]
2. What does dN/dt mean? [1 mark]
3. Assume that $N < K$. For what values of r will N increase over time? [1 mark]
4. Assume that $r > 0$ and $K > N$. Will N increase or decrease in size over time? [1 mark]

5. Assume that $r, K \neq 0$. For what values of N is the population size constant (i.e., not changing over time)? [2 marks]
6. What is the main difference between exponential and logistic growth? [1 mark]

Chapter 11

Tues Oct 6: Solving the logistic growth equation using a computer 1

ANNOUNCEMENT: Friday Oct 2 is a catch-up day

DUE DATE: Tues Oct 13

Recall that for continuous time exponential growth there were two ways that the equation might be written. Firstly, as a *change* in population size over time,

$$\frac{dN(t)}{dt} = rN(t), \quad (11.1)$$

such that if $\frac{dN(t)}{dt} > 0$, the population size, $N(t)$ is increasing over time, and if $\frac{dN(t)}{dt} < 0$, the population size is decreasing over time. Alternatively, we can write continuous time exponential growth as,

$$N(t) = N(0)e^{rt}, \quad (11.2)$$

where the lefthand side of equation (11.2) states the *size* of the population, $N(t)$, at time t , rather than the *change* in the size of the population, as it did in equation (11.1).

For logistic growth, we have:

$$\frac{dN(t)}{dt} = rN(t) \left(1 - \frac{N(t)}{K} \right). \quad (11.3)$$

How might we find the equivalent of equation (11.2) for logistic growth? i.e., for logistic growth, what is the equation for the *size* of the population at a particular time, t ? If you are great at math, you might integrate equation (11.3) using separation of variables. However, another option, that we will build towards for next class, is to numerically integrate equation (11.3) using a computer.

11.1 What is an ordinary differential equation?

In an introductory calculus course that covers integration, most often you are solving problems of the type:

$$\frac{dy(x)}{dx} = x^2, \quad (11.4)$$

$$\int \frac{dy(x)}{dx} = \int x^2 dx, \quad (11.5)$$

$$y(x) = \frac{1}{3}x^3 + c. \quad (11.6)$$

Note that in the above calculations, we are able to use integration to find $y(x)$ when we are given the rate of change equation, $\frac{dy(x)}{dx}$. In the above equations, x , is the independent variable, $y(x)$ is the dependent variable, the derivative of $y(x)$ is with respect to x , and x appears on the opposite side of the $=$ to the derivative of the dependent variable. In the notation of population biology, where $N(t)$ is conventional for the dependent variable, a similar problem might look like this:

$$\int \frac{dN(t)}{dt} = \int rt dt, \quad (11.7)$$

$$N(t) = \frac{1}{2}rt^2 + c.$$

We have not seen any equations like equation (11.8) in BIOL 3295 so far: usually, our equations for the *change* in the population size, $\frac{dN(t)}{dt}$, have a dependency on the population size, $N(t)$ (the dependent variable), rather than a sole dependency on t (the independent variable).

The equations that are models for population biology are sometimes ODEs. While some ODEs can be solved using math, commonly the ODEs that are used to model population dynamics need to be integrated numerically; that is, using a computer.

11.2 Numerically integrating an ordinary differential equation in R

Numerical integration in R can be performed by installing this package:

```
install.packages("deSolve")
```

To integrate our ODE, we use the function `ode()` from the `deSolve` package. Let's look at the structure of this function. The mandatory arguments to the function are:

```
ode(y, times, parms, func)
```

where,

`y` is a vector of the initial values of the dependent variable (i.e., those which correspond to the initial time).

`times` is a sequence of time points that we will calculate the values of the variables; the first value of `times` is the initial time.

`func` is a function that computes the values of the derivatives of the dependent variables. `func` must be defined as: `func <- function(t, y, parms,...)`. `t` is the current time point in the integration, and `y` is the current estimate of the variables in the ODE system.

Given what we have learned in BIOL 3295 so far, `y` and `times` are not so tricky, but the `func` argument requires us to write our own function, and so far we have only used functions written by other people: we have yet to write our own custom functions. In preparation for next class, when we will numerically integrate the logistic growth equation, we will next learn how to write custom functions.

11.3 Writing custom functions

Generally, your code can still work without writing custom functions, however writing functions makes your code modular and more organized.

Consider the following code:

```
# Temperatures in Celcius
temp.min.C <- 10
temp.max.C <- 20

# Temperatures in Farenheit
temp.min.F <- temp.min.C*9/5+32
temp.max.F <- temp.max.C*9/5+32
```

The above code is fine, but since the same calculation is performed twice (conversion of Celcius to Farenheit), just with a different number, we might consider writing a custom function. Let's re-write the above code so that it now uses a custom function:

```
# A function that converts Celcius to Farenheit
C.to.F <- function(C){
  F <- C*9/5 +32
  return(F)
}

# Temperatures in Celcius
temp.min.C <- 10
temp.max.C <- 20

# Temperatures in Farenheit
temp.min.F <- C.to.F(temp.min.C)
temp.max.F <- C.to.F(temp.max.C)
```

Note that the above code consists of the *function definition* and the *function call*. The function definition uses the function `function()`.

Function Definition

Description

These functions provide the base mechanisms for defining new functions in the R language.

Usage

```
function( arglist ) expr
return(value)
```

Arguments

arglist Empty or one or more name or name=expression terms.
expr An expression.
value An expression.

In the example, note the following:

- **arglist** is just one value: C. Generally, several values might be provided to the function where each should be separated by commas and enclosed within the ().

- `expression` can be any set of commands and should be enclosed within the `{}`.
- The function is assigned a name: `C.to.F`. We give the function a name so that we can use it later during a function call.

11.4 Questions

1. The following is an equation for exponential population growth with a constant immigration rate, $m > 0$, into the population:

$$\frac{dN(t)}{dt} = rN(t) + m$$

where r is the net reproductive rate, and $N(t)$ is the population size at time, t .

Is this equation an ordinary differential equation (ODE)? Give 1 sentence explaining your reasoning [2 marks].

2. Write a function that adds 123 to a user supplied argument, and then write the line of code that calls the function to evaluate $123+1$, and assigns the result the variable name `y`. You are to hand in your R script. [5 marks]

Chapter 12

Thurs Oct 8: Solving the logistic growth equation using a computer 2

DUE DATE: Thursday Oct 15

Last class we learned how to make a custom function. Today we would like to write a custom function that satisfies the requirements of the `func` argument that is supplied to `ode()` so that numerical integration can be performed.

12.1 Writing the `func` argument for `ode()`

The `func` argument of `ode()` has the following constraints:

- The `arglist` must include `t`, `y`, and `parms`.
- The value returned must be a *list* of the rates of change in the dependent variable(s).

Note that if your ODE comprises of more than one variable that is changing over time, then `y` will be a vector. Below is an example of how `func` might be coded for exponential growth. Remember that the ODE for exponential growth is:

$$\frac{dN(t)}{dt} = rN(t) \quad (12.1)$$

Below equation (12.1) is coded in the appropriate syntax for eventual use in the `ode()` function:

```
# Write a function that returns a list of the rate of change in your dependent variable.
model = function(t,y,parms){
  # It's a personal preference of mine to switch the symbols
  # to be consistent with the equation.
  N <- y[1]
  # This is this equation for dN/dt
  dN <- r*N
  # The function returns the value of the change in N.
  return(list(c(dN)))
}
```

Let's try out the function to make sure it works. Copy and paste the function into your Console and then type:

```
r<-1
model(1,2,parms=NULL)
```

Did you get 2? Does it make sense to you that the returned value is 2? Try some other values. Are the values making sense?

Now that the `func` argument of `ode()` has been demonstrated, the remaining arguments for the `ode()` function are not so bad.

```
# The initial value of N
yini<- c(N = 1)

# the times for the numerical integration
times <- seq(0, 1, by = .1)
```

Remember to assign a value for `r`:

```
# Assign a value to the parameter:
r <- 2
```

And then once all the arguments are set, call the function, and plot the results:

```
# performing the numerical integration
out <- ode(y = yini, parms = NULL, times = times, func = model)
out <- data.frame(out)
head(out)

# this line overrides a multipanel plot. I had tried plot(out) # and was having conflicts
# laptop. Generally, the below line isn't necessary.
par(mfrow=c(1,1))

# Make the graph
plot(out$time, out$N, typ = "l")
```

12.2 Questions

1. For the code `model(1,2,parms=NULL)` (where the function `model` is defined above), what do 1 and 2 correspond to? One of these quantities does not affect the value that the `model()` function returns. Which quantity does not affect the value the function returns? [2 marks]
2. Write an R Script that numerically integrates the logistic growth function (equation (11.3)). Choose a value of r such that the population size increases over time. Remember to use good practices for writing your script - see 1.7.2. Your script should produce a graph with $N(t)$ on the y-axis and time on the x-axis [10 marks]
3. Add to your script from 2. such that another graph is produced, this time where the population size decreases over time. Note that this should just require adding new lines to the bottom of your question 1 script such that you: a) set a new value of `r`; b) run the `ode()` command; and c) make the graph. In other words, it is not necessary to define the logistic growth function, `model`, again because that has not changed. Further, if you are happy to use the same values of `times` as for question 1, you do not need to restate this either. Answer this question by adding a few lines of code to your answer to 1. You are to hand in your R script. You should answer both questions 1. and 2. in the same R script. [5 marks]

Chapter 13

Fri Oct 9 and Thurs Oct 15: Data and the logistic equation

DUE DATE: Thurs Oct 22

13.1 Questions

1. Question 1.9 on p12 of Vandermeer and Gordon (but completed following the instructions below). Hand-in your graph with a figure caption. Your graph should have the intrinsic growth rate on the y-axis and population size (in millions) on the x-axis. In response to ‘what kind of function would describe the data reasonably well’, you just need to answer an increasing or decreasing function. [10 marks]
2. Based on your work in question 1, what is the value of the intrinsic growth when $t = 1955, 2013$ and 2014 ? [2 marks]
3. If a population is growing exponentially, and the intrinsic growth rate is plotted on the y-axis, and population size is plotted on the x-axis, what is the slope of the line? What does the intercept of the graph tell us? [2 marks]

13.2 The data

Question 1.9 in Vandermeer and Goldberg involves a large data table. It will be very boring to have to type all those numbers in by hand. Fortunately, here is a website, with a link to a similar data set. Let's load in the data:

```
pop.data <- read.csv('https://ourworldindata.org/uploads/2013/05/WorldPopulationAnnual
```

Let's take a look at our data by entering the following in the console:

```
head(pop.data)
```

```
##      year World.Population..Spline.Interpolation.until.1950.
## 1 -10000                                     2431214
## 2 -9999                                       2432196
## 3 -9998                                       2433179
## 4 -9997                                       2434162
## 5 -9996                                       2435145
## 6 -9995                                       2436129
```

Let's find out the column names of our data. Type the following into your console:

```
colnames(pop.data)
```

```
## [1] "year"
## [2] "World.Population..Spline.Interpolation.until.1950."
```

The second column name is weird. Let's replace it with a better name:

```
colnames(pop.data)<-c("year", "size")
```

The data we have loaded starts in 10,000 B.C., but the Vandermeer and Goldberg data just starts in 1955. The code below identifies which rows correspond to 1955 onwards.

```
ind <- which(pop.data$year>=1955)
```

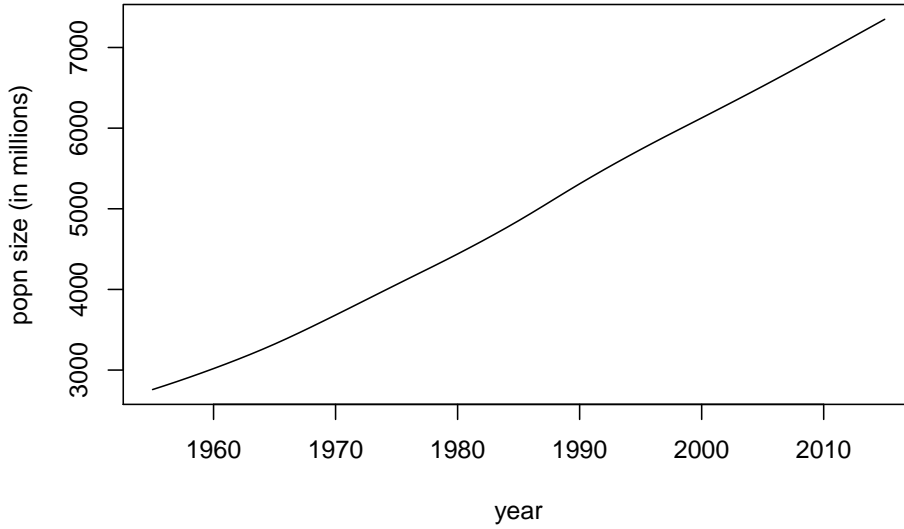
Let's make a new dataset that only contains the data for 1955 onwards:

```
pop.data.1955 <- pop.data[ind,]
```

In the above code the square brackets are the `[rows,columns]` that we are extracting. Note that the rows we would like, we have assigned the variable name `ind`. When the column argument is left blank this is understood to be all columns (or in this case, *both* columns: `year` and `size`).

Let's plot our data, but to make things easier, let's change the population size into millions (where `1e6` is scientific notation for millions).

```
pop.data.1955$size <- pop.data.1955$size/1e6
plot(pop.data.1955$year, pop.data.1955$size, typ = "l", xlab = "year", ylab = "popn si
```



13.3 The intrinsic growth rate

Now, consider the equation,

$$N_{t+1} = r(N_t)N_t. \quad (13.1)$$

This equation states that the population size in the next time step (N_{t+1}) is equal to the population size at time t (which is N_t), multiplied by the intrinsic growth rate, $r(N_t)$, where the intrinsic growth rate might depend on population size (which is why $r(N_t)$ is written that way: you can read it as ‘ r is a function of N_t ’). For example, with exponential growth, the intrinsic growth rate does not change with population size, so $r(N_t) = \lambda$. However, when population growth demonstrates negative density dependence, then $r(N_t)$ would be a decreasing function of population size, N_t .

Consider the Beverton-Holt equation which is equation 28 on p29 of Vandermeer and Goldberg:

$$N_{t+1} = \frac{\lambda N_t}{1 + \alpha N_t}.$$

Note that this could be written in the form of equation (13.1) where,

$$r(N_t) = \frac{\lambda}{1 + \alpha N_t}. \quad (13.2)$$

Therefore, the Beverton-Holt equation demonstrates negative density dependence since the intrinsic growth rate, $r(N_t)$, is a decreasing function of population size, N_t .

By re-arranging equation (13.1), we can see that the intrinsic growth rate is calculated as:

$$r(N_t) = \frac{N_{t+1}}{N_t},$$

such that the intrinsic growth rate can be understood as a ratio of population sizes at consecutive time points. How might we calculate the intrinsic growth rate for our data?

First, we should note that Question 1.9 in Vandermeer and Goldberg asks us to plot the intrinsic growth rate (y-axis) versus the population size (x-axis). Let's take $N_{1955} = 2758$: this is our first x-axis value, but what is the corresponding y-value? The answer is:

$$r(N_{1955}) = \frac{N_{1956}}{N_{1955}} = \frac{N_{t+1}}{N_t} \text{ where } t = 1955.$$

Note that the numerator values will be the sequence of values

$[N_{1956}, N_{1957}, \dots, N_{2015}]$,

and the denominator values will be the sequence of values

$[N_{1955}, N_{1956}, \dots, N_{2014}]$.

Therefore, the numerator values are the list of all population sizes except the first value (N_{1955}), and the denominator values are the list of all population sizes except the last value (N_{2015}). The functions `head()` and `tail()` can be used to remove values from a list:

```
numerator <- tail(pop.data.1955$size,-1)
denominator <- head(pop.data.1955$size,-1)
```

And then we can calculate the list of intrinsic growth rates, $r(N_t)$:

```
r <- numerator/denominator
```

Now consider the values we need for our x-axis. The last value of $r(N_t)$ that we calculated was $\frac{N_{2015}}{N_{2014}} = \frac{N_{t+1}}{N_t}$ when $t = 2014$, which means that the last value of N_t we have a corresponding $r(N_t)$ value for is N_{2014} . Therefore, our x-axis needs to be:

```
Nt <- head(pop.data.1955$size,-1)
```

Finally, we need to make the plot as required by Question 1.9.

```
plot(Nt, r, typ = "l")
```

It will be your job to put appropriate axes labels on the plot.

13.4 References

Vandermeer, J.H., Goldberg, D.E., 2013. Population Ecology: First Principles (Second Edition). Princeton University Press, Princeton, United States. [Link](#)

Chapter 14

**Tues Oct 20: Density-yield
and density dependence in
births versus deaths**

Chapter 15

Thurs Oct 22: Stage-structured models

- The idea of stage-structured models.
- Multiplying matrices.
- Eigenvalues of 2×2 matrix

Chapter 16

Fri Oct 23: Stage-structured dynamics

- Eigenvalues and eigenvectors
- Diagrams

Chapter 17

**Tues Oct 27: Yellow
columbine**

Chapter 18

**Thurs Oct 29-Fri Oct 30:
Midterm [due Fri Nov 6 at
5pm]**

Chapter 19

**Thurs Nov 5: Evolutionary
ecology**

Chapter 20

Fri Nov 6: Evolutionary ecology

Chapter 21

**Tues Nov 10: Evolutionary
ecology**

Chapter 22

Thurs Nov 12:
Evolutionary ecology

Chapter 23

**Fri Nov 13: Evolutionary
ecology**

Chapter 24

Tues Nov 17: Evolutionary
ecology

Chapter 25

Thurs Nov 19:
Evolutionary ecology

Chapter 26

**Fri Nov 20: Evolutionary
ecology**

Chapter 27

Final project ideas

[DRAFT]

For your final project you may choose either:

1. Review paper: Read and synthesize information on a new topic in population and evolutionary ecology that we have not considered in class.

or,

2. Analysis: Download and visualize a dataset relevant to population and evolutionary ecology. Discuss your graphs in the context of a principle in population and evolutionary ecology.

Use Dias et al. 1996 *Sources and sinks in population biology* as a guide for what a final project that is a *Review* should aspire to look like.

If you choose to do a final project that is an *Analysis* you should use some of the exercises we have completed for class as a guide (for example, the Doubling time, Protection Island, Yellow columbine, or balsam fir analyses). Note that you will need to write your final project as a report, rather than a series of questions.

Some examples of final project topics are:

Review paper: - Population dynamics in warming environments. - Population dynamics in seasonal environments. - Spatial population dynamics. - Metapopulations.

Analysis:

27.1 How to read scientific papers

Please consult *How to Read a Scientific Paper* (2014) for the recommended approach to ‘reading’ journal articles. Steps 1,2, and 4 are good advice, but step 3 may not be relevant for some readings.

27.2 Sources for datasets

27.3 Grading Rubric: Review

27.4 Grading Rubric: Analysis