BIO 423 - Lab 3

Benjamin Blonder

Spring 2019

Learning outcomes

For single species:

- Apply matrix/eigenvalue operations in R
- Calculate elasticities and population trajectories for stage-structured models
- Use elasticities to propose management interventions for species

For multiple species:

- Model the temporal dynamics of competing species
- Predict the outcome of competition using nullcline analysis

Stage-structured models

We will investigate stage structured models from the COMADRE database (`https://www.compadre-db.org/Data/Comadre`). This resource is the largest global database of animal and plant matrix models and reflects the efforts of hundreds of investigators. It is freely available for our use.

We are going to use the database to calculate some basic demographic parameters for different species. Parameters will include the future abundances of each life stage (using a `project_population` function), the long-term growth rate of the population (using the largest eigenvalue of the transition matrix, via the `lambda` function), the elasticity at each life stage (via the `elasticity` function), and the stable age distribution (via eigenvector analysis and the `stable_age` function).

```
# load in animal data - when using the load() command,
# R can directly import data into a range of different variable names
# as opposed to read.csv, which requires you to save its output in a
# certain variable
load("comadre_v.2.0.1.RData")

# look to see the names of available variables
ls()
```

```
## [1] "comadre"
```

```
# explore the structure of the data
names(comadre)
```

```
## [1] "metadata"    "matrixClass" "mat"         "version"
```

```
str(comadre$metadata)
```

```
## 'data.frame':    1927 obs. of  50 variables:
##  $ SpeciesAuthor         : chr  "Acipenser_fulvescens" "Acipenser_fulvescens" "Acipenser_fulvescens"
##  $ SpeciesAccepted       : chr  "Acipenser_fulvescens" "Acipenser_fulvescens" "Acipenser_fulvescens"
##  $ CommonName            : chr  "Lake sturgeon" "Lake sturgeon" "Lake sturgeon" "Lake sturgeon" ...
##  $ CoLCheckOK            : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ CoLCheckDate          : chr  "08082016" "08082016" "08082016" "08082016" ...
```

```
##  $ Infraspecific         : chr  NA NA NA NA ...
##  $ SpeciesEpithetAccepted: chr  "fulvescens" "fulvescens" "fulvescens" "fulvescens" ...
##  $ GenusAccepted         : chr  "Acipenser" "Acipenser" "Acipenser" "Acipenser" ...
##  $ GenusAuthor           : chr  "Acipenser" "Acipenser" "Acipenser" "Acipenser" ...
##  $ Family                : Factor w/ 193 levels "Accipitridae",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Order                 : Factor w/ 95 levels "Accipitriformes",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Class                 : Factor w/ 27 levels "Actinopterygii",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Phylum                : Factor w/ 10 levels "Annelida","Arthropoda",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Kingdom               : Factor w/ 2 levels "Animalia","Bacteria": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OrganismType          : Factor w/ 28 levels "Actinopterygii",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Authors               : chr  "Velez-Espino; Koops" "Velez-Espino; Koops" "Velez-Espino; Koops" "Ve
##  $ Journal               : chr  "N Am J Fish Manag" "N Am J Fish Manag" "N Am J Fish Manag" "N Am J I
##  $ YearPublication       : chr  "2009" "2009" "2009" "2009" ...
##  $ DOI.ISBN              : chr  "10.1577/M08-034.1" "10.1577/M08-034.1" "10.1577/M08-034.1" "10.1577/
##  $ AdditionalSource      : chr  NA NA NA NA ...
##  $ StudyDuration         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ StudyStart            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ StudyEnd              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ AnnualPeriodicity     : chr  "1" "1" "1" "1" ...
##  $ NumberPopulations     : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ MatrixCriteriaSize    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MatrixCriteriaOntogeny: Factor w/ 4 levels "Based on age",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ MatrixCriteriaAge     : Factor w/ 3 levels "Based on size",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ MatrixPopulation      : chr  "Western Hudson Bay; Saskatchewan River; Nelson river main stem; Red
##  $ Lat                   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Lon                   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Altitude              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Country               : Factor w/ 74 levels "1","1a","1b",..: 16 16 16 16 16 16 16 16 16 16 ...
##  $ Continent             : Factor w/ 8 levels "Africa","Asia",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ Ecoregion             : chr  NA NA NA NA ...
##  $ StudiedSex            : Factor w/ 5 levels "A","F","H","M",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ MatrixComposite       : Factor w/ 4 levels "Individual","Mean",..: 2 1 1 1 1 1 1 1 1 1 ...
##  $ MatrixTreatment       : chr  "Unmanipulated" "Unmanipulated" "Unmanipulated" "Unmanipulated" ...
##  $ MatrixCaptivity       : Factor w/ 4 levels "C","CW","W","W; CW": 3 3 3 3 3 3 3 3 3 3 ...
##  $ MatrixStartYear       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixStartSeason     : Factor w/ 4 levels "1","2","3","4": NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixStartMonth      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixEndYear         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixEndSeason       : Factor w/ 4 levels "1","2","3","4": NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixEndMonth        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MatrixSplit           : Factor w/ 2 levels "Divided","Indivisible": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MatrixFec             : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Observation           : Factor w/ 355 levels "0.024cm^3 glyphosate herbicide + distilled water to
##  $ MatrixDimension       : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ SurvivalIssue         : num  0.904 0.916 0.861 0.931 0.898 ...
```

```
# the 'mat' element (comadre$mat) is a list of many demographic matrics,
# with the [[i]] entry corresponding to the ith species in the metadata table
```

Define functions for demography

```
project_population = function(A,xstart,nsteps)
{
```

```
    stopifnot(nsteps >=2)

    x = matrix(nrow=nrow(A),ncol=nsteps)
    x[,1] = xstart

    for (i in 2:nsteps)
    {
        x[,i] = A %*% x[,i-1] # replace x with the old value of x
    }

    # transpose matrix for easier plotting
    return(t(x))
}

sensitivity = function(A) {
  w = eigen(A)$vectors # right eigen vectors
  v = Conj(solve(w)) # complex conjugate of left eigen vectors
  # output of eigenvalues is decreasingly sorted.
  return(Re(v[1,] %*% t(w[,1])))
}

elasticity = function(A)
{
    s = sensitivity(A)

    ev = eigen(A)
    lmax = which.max(Re(ev$values))
    lambda = Re(ev$values[lmax])

    return(s*A / lambda)
}

lambda = function(A)
{
    ev = eigen(A) # get eigensystem
    lmax = which.max(Re(ev$values)) # find the biggest eigenvalue
    lambda = Re(ev$values[lmax]) # store the real part of the largest eigenvalue

    return(lambda)
}

stable_age = function(A)
{
    # find right eigenvector corresponding to maximum eigenvalue
    ev <- eigen(A)
    lmax <- which.max(Re(ev$values))
    W <- ev$vectors
    w <- abs(Re(W[, lmax]))

    # return normalized vector
    return(w/sum(w))
}
```

Apply functions to real data

```
# choose a species by running View(comadre$metadata)
index = 1749 # the spectacled caiman
comadre$metadata[index,]
```

```
##          SpeciesAuthor   SpeciesAccepted        CommonName CoLCheckOK
## 1749 Caiman_crocodilus Caiman_crocodilus Spectacled caiman       TRUE
##      CoLCheckDate Infraspecific SpeciesEpithetAccepted GenusAccepted
## 1749     08082016          <NA>             crocodilus        Caiman
##      GenusAuthor        Family      Order    Class   Phylum  Kingdom
## 1749      Caiman Alligatoridae Crocodylia Reptilia Chordata Animalia
##      OrganismType Authors Journal YearPublication          DOI.ISBN
## 1749     Reptilia Tucker     Book            2001 978-0-949324-89-4
##                                                  AdditionalSource
## 1749 Thorbjanarson 1991, 1992; Ouboter 1996; Snider & Bowler 1992
##      StudyDuration StudyStart StudyEnd AnnualPeriodicity NumberPopulations
## 1749            NA         NA       NA                 1                 1
##      MatrixCriteriaSize MatrixCriteriaOntogeny MatrixCriteriaAge
## 1749                 No                    Yes               Yes
##      MatrixPopulation Lat Lon Altitude Country Continent Ecoregion
## 1749             <NA>  NA  NA       NA    <NA>      <NA>      <NA>
##      StudiedSex MatrixComposite MatrixTreatment MatrixCaptivity
## 1749          F            Mean   Unmanipulated            <NA>
##      MatrixStartYear MatrixStartSeason MatrixStartMonth MatrixEndYear
## 1749              NA              <NA>               NA            NA
##      MatrixEndSeason MatrixEndMonth MatrixSplit MatrixFec Observation
## 1749            <NA>             NA     Divided       Yes       FRANK
##      MatrixDimension SurvivalIssue
## 1749               5         0.875
```

```
comadre$mat[[index]]
```

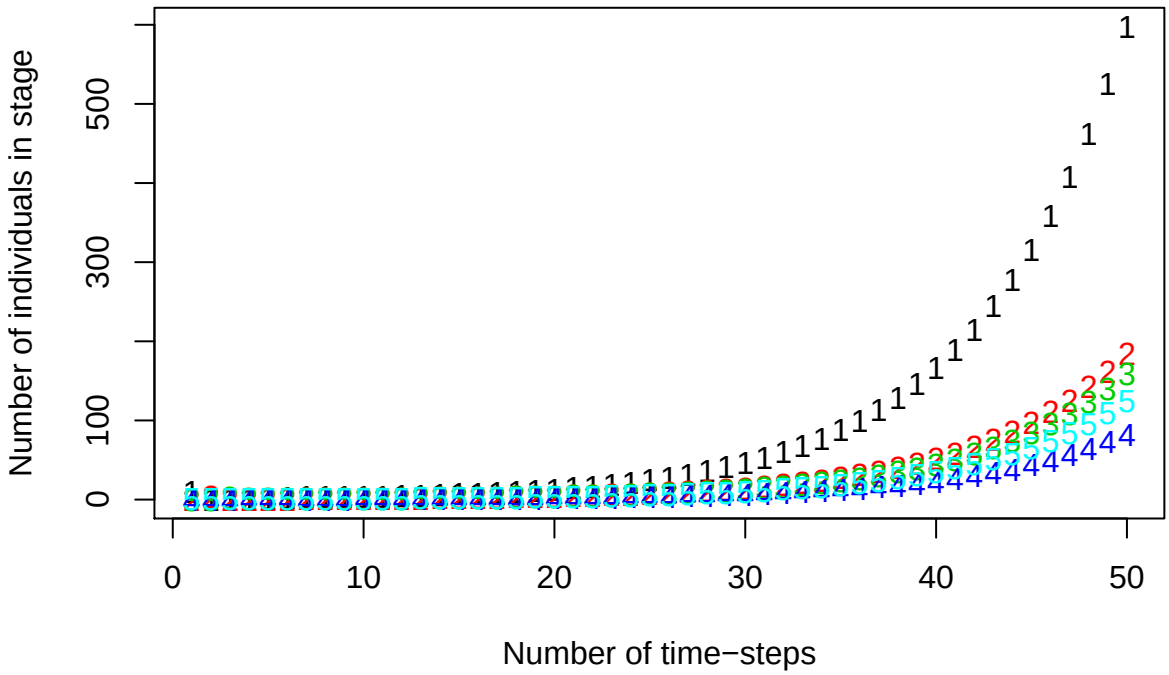
```
## $matA
##        A1  A2      A3      A4      A5
## [1,] 0.00 0.0 0.00000 1.49000 4.50000
## [2,] 0.35 0.0 0.00000 0.00000 0.00000
## [3,] 0.00 0.6 0.44072 0.00000 0.00000
## [4,] 0.00 0.0 0.34728 0.46667 0.00000
## [5,] 0.00 0.0 0.00000 0.40833 0.86426
##
## $matU
##        U1  U2      U3      U4      U5
## [1,] 0.00 0.0 0.00000 0.00000 0.00000
## [2,] 0.35 0.0 0.00000 0.00000 0.00000
## [3,] 0.00 0.6 0.44072 0.00000 0.00000
## [4,] 0.00 0.0 0.34728 0.46667 0.00000
## [5,] 0.00 0.0 0.00000 0.40833 0.86426
##
## $matF
##      F1 F2 F3   F4  F5
## [1,]  0  0  0 1.49 4.5
## [2,]  0  0  0 0.00 0.0
## [3,]  0  0  0 0.00 0.0
```

```
## [4,]  0  0  0 0.00 0.0
## [5,]  0  0  0 0.00 0.0
##
## $matC
##      C1 C2 C3 C4 C5
## [1,]  0  0  0  0  0
## [2,]  0  0  0  0  0
## [3,]  0  0  0  0  0
## [4,]  0  0  0  0  0
## [5,]  0  0  0  0  0
```

```
# we can get the integrated 'A' matrix (incorporating all of growth, fecundity, etc.) as
A = comadre$mat[[index]]$matA
print(A)
```

```
##        A1  A2      A3      A4      A5
## [1,] 0.00 0.0 0.00000 1.49000 4.50000
## [2,] 0.35 0.0 0.00000 0.00000 0.00000
## [3,] 0.00 0.6 0.44072 0.00000 0.00000
## [4,] 0.00 0.0 0.34728 0.46667 0.00000
## [5,] 0.00 0.0 0.00000 0.40833 0.86426
```

```
# now 'A' is a variable we can use in all the code below...

# simulate population forward
nsteps = 20 # define the number of time steps
x1 = rep(0,dim(A)[1])
x1[1] <- 10
# set the initial numbers in each population stage to 10 juveniles
# make sure to use the right number of entries to conform to the 'A' matrix!

population_future = project_population(A, xstart=x1, nsteps = 50)

# plot the population over time
matplot(population_future,ylab='Number of individuals in stage',xlab='Number of time-steps')
```



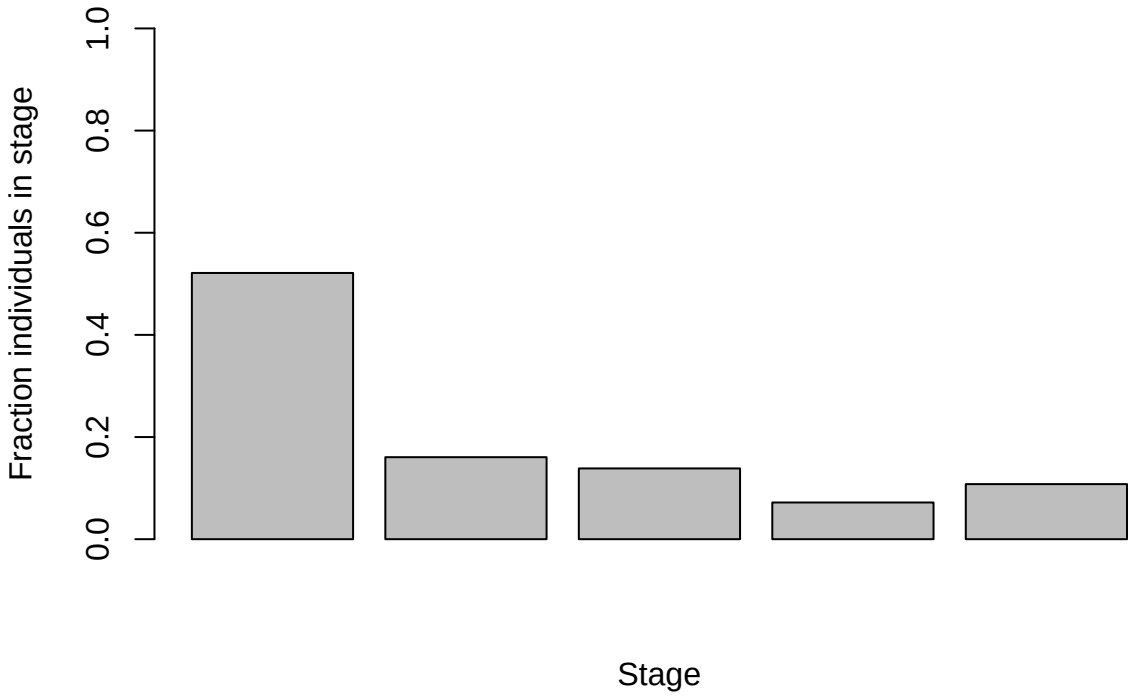

```
# find the long-term growth rate of the population
lambda(A)
```

```
## [1] 1.136268
```

```
# calculate elasticities for each transition
elasticity(A)
```

```
##             A1        A2         A3         A4         A5
## [1,] 0.0000000 0.0000000 0.00000000 0.02064545 0.09360089
## [2,] 0.1142463 0.0000000 0.00000000 0.00000000 0.00000000
## [3,] 0.0000000 0.1142463 0.07238988 0.00000000 0.00000000
## [4,] 0.0000000 0.0000000 0.11424634 0.07962289 0.00000000
## [5,] 0.0000000 0.0000000 0.00000000 0.09360089 0.29740098
```

```
# calculate stable age structure
populationratio = stable_age(A)
barplot(populationratio,xlab='Stage',ylab='Fraction individuals in stage',ylim = c(0,1))
```



Questions

1. Explore the data to find a species that has `lambda < 1` and a species with `lambda > 1`. Report the names of each, the `index` value in the metadata for each, and the value of `lambda` you calculate.
2. For your species with `lambda < 1`, plot the number of individuals in each stage over time, assuming that the population starts out with 100 individuals in the smallest class. How many time steps are needed before the population becomes extinct?
3. For your species with `lambda > 1`, calculate the stable age distribution. Which stage is predicted to be most abundant, and which most rare? Based on your knowledge of this species (use Google), does this prediction seem realistic?
4. Calculate the elasticity matrix for the `lambda > 1` species. Based on the elasticity matrix, if you were managing this species, what intervention could you take (& at which life stage) to most rapidly increase the population?
5. For your species with `lambda > 1`, the model suggests the population will eventually become infinite. We know this won't occur - give at least one explanation for why the model yields this incorrect prediction.

Optional question for graduate students

- Write your own code to project the matrix models forward using matrix multiplication (you will need the `%*%` operator) and verify it produces the same results as my code.
- Write your own code to find the eigenvalues of A.
- For the species you found with `lambda > 1`, identify a change in the A matrix you could make (e.g. `A[3,2]` becomes 0.9) to yield `lambda=1`. You can either do this via numerical experimentation, or if you have had some linear algebra, you can do an eigendecompositon of the matrix and explore the consequences of manipulating the eigenvalues directly.

Predator-prey dynamics

We can also explore the behavior of a predator-prey system. Consider a system of rabbits and foxes (N_{rabbit} and (N_{fox})), where the rabbits have exponential growth that is checked only by predation determined by the number of foxes, and the foxes have exponential growth that increases with the number of rabbits, and is checked by density-independent mortality. Such a model can be expressed as:

$$\frac{dN_{rabbit}}{dt} = k_{rabbit\ growth\ rate}N_{rabbit} - k_{prey\ capture\ rate}N_{rabbit}N_{fox}$$

$$\frac{dN_{fox}}{dt} = k_{predator\ conversion\ efficiency}N_{rabbit}N_{fox} - k_{fox\ mortality\ rate}N_{fox}$$

where $k_{rabbit\ growth\ rate}$ is the intrinsic growth rate of rabbits, $k_{prey\ capture\ rate}$ is the rate at which foxes kill rabbits, $k_{predator\ conversion\ efficiency}$ is the rate at which rabbit kills are converted into new foxes, and $k_{fox\ mortality\ rate}$ is the per-capita mortality rate of foxes.

We will investigate its behavior both by solving the equations for the equilibrium state, and also by simulating the differential equation.

We can simulate how the numbers of rabbits and foxes using numerical integration techniques, by approximating differentials as finite difference. Suppose we have a generic differential equation,

$$\frac{dN(t)}{dt} = f(N(t), t)$$

We can approximate this equation as

$$\frac{\Delta N(t)}{\Delta t} \approx f(N(t), t)$$

rearranging yields

$$\Delta N(t) \approx f(N(t), t)\Delta t$$

and thus we can find an approximate solution for $N(t)$ after a single time step from t to $t + \Delta t$ as

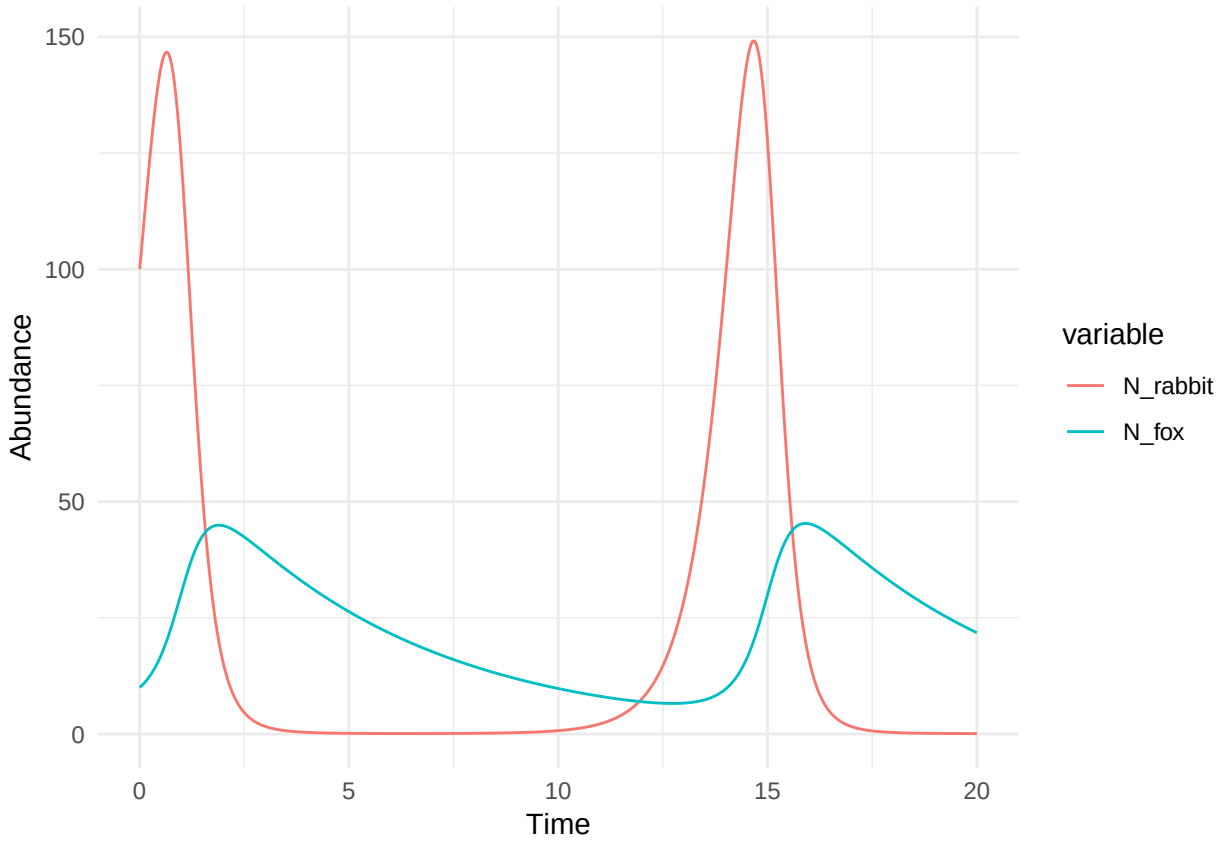
$$N(t + \Delta t) \approx N(t) + \Delta N(t) = N(t) + f(N(t), t)\Delta t$$

That is, we evaluate the value of f at each time point, multiply it by the time step Δt, and add this difference to the existing value of N to get the value at time $t + \Delta t$. As the value of Δt, becomes increasingly small, the approximation becomes increasingly good. This is called **Euler integration**. (Better numerical integration methods exist, but this is the simplest one to explain in a lab!)

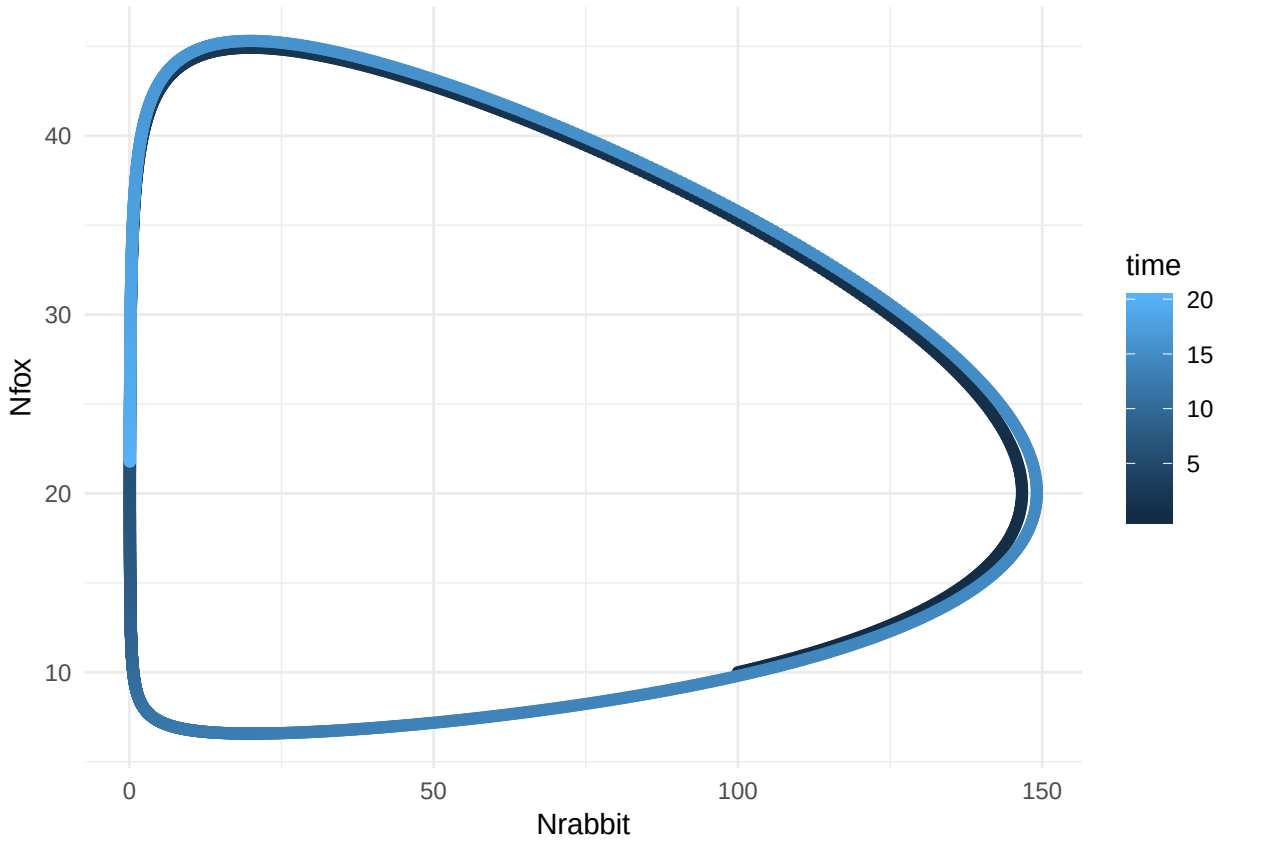
The below code implements this Euler integration scheme for you. A few key parameters are `n_steps`, the total number of integration steps, each of which has size `delta_t` (Δt) (i.e. a total integration time of `n_steps * delta_t`). The initial population sizes are set as `N_rabbit[1] = 100` and `N_fox[1] = 10`. Parameters are set a bit below as, e.g. `reproduction_rate_rabbit = 2`. There is one small trick in the code at the end - we use the `melt()` function from the `reshape2` package to help transform the dataframe output from 'wide' format (multiple columns for the abundance of each species) to 'long' format (single column for the abundance of all species, with an additional identifier column). This is helpful for plotting the data in `ggplot2`.

Let's run the code and see the model's predictions for predator and prey for one set of parameters:

```r
# make sure both libraries are installed
library(ggplot2)
library(reshape2)

# number of time steps in the simulation
n_steps = 4000
delta_t = 0.005 # the smaller delta_t, the more accurate the simulation

# allocate vectors for the predicted abundances over time
N_rabbit = rep(NA, n_steps)
N_fox = rep(NA, n_steps)
time_intervals = (1:n_steps)*delta_t

# initial population sizes for each species
N_rabbit[1] = 100
N_fox[1] = 10

# key parameters for species interactions
reproduction_rate_rabbit = 2
prey_capture_rate = 0.1
predator_conversion_efficiency = 0.01
mortality_rate_fox = 0.2

for (i in 2:n_steps)
{
  delta_rabbit = reproduction_rate_rabbit*N_rabbit[i-1] -
      prey_capture_rate * N_rabbit[i-1] * N_fox[i-1]

  delta_fox = predator_conversion_efficiency * N_rabbit[i-1] * N_fox[i-1] -
      mortality_rate_fox * N_fox[i-1]

  N_rabbit[i] = N_rabbit[i-1] + delta_rabbit * delta_t
  N_fox[i] = N_fox[i-1] + delta_fox * delta_t
}

# combine results in a data frame
predictions_predprey = data.frame(time=time_intervals,N_rabbit,N_fox)
# convert the data frame from 'wide' to 'long' format for easier grouped plotting
predictions_predprey_melted = melt(predictions_predprey, id.vars="time")

# plot predictions as time-series using the melted dataframe
ggplot(predictions_predprey_melted,aes(x=time,y=value,col=variable)) +
  geom_line() + theme_minimal() +
  xlab("Time") + ylab("Abundance")
```




```
# plot predictions as phase space plot (now use the un-melted dataframe)
ggplot(predictions_predprey,aes(x=N_rabbit,y=N_fox,col=time)) +
  geom_point() + theme_minimal() +
  xlab("Nrabbit") + ylab("Nfox")
```




```
# (the small deviation on the right side of the plot is not biologically
# real, but reflects numerical error from the Euler integration scheme -
# you can reduce it by decreasing delta_t)
```

Questions

6. In the equation version of the model, find the equilibrium number of foxes and rabbits using nullcline analysis. Interpret the resulting equations in 1-2 sentences.
7. In the numerical simulation, do increases in the number of rabbits lead (come before) or lag (come after) increases in the numbers of foxes? Give a one-sentence explanation for why this occurs.
8. Find a combination of parameters that results in zero foxes and zero rabbits after a long amount of time. Show a time-series plot and report the parameter combination. How can you explain this outcome based on the parameter values?

Optional questions for graduate students

- The equilibrium state in the predator-prey model with a finite number foxes and rabbits exists in principle, but is never actually obtained as you can see in these simulations. Give a verbal explanation for why not, or (if mathematically inclined), a quantitative explanation based on a linear stability analysis.
- (For modeling inclined students) Incorporate a rabbit carrying capacity into the model, e.g. by extending the rabbit differential equation to include a negative quadratic term. How does this affect the model predictions? Explore a few parameter combinations.

- (For computationally inclined students) - the numerical integration scheme used here accumulates large errors if you allow `n_steps` to become large - the errors compound. You can largely fix this problem by using a more robust numerical integration method, for example a Runge-Kutta 4th-order method (`http://lpsa.swarthmore.edu/NumInt/NumIntFourth.html`. Implement this method and show the error rates are much lower (you can do so by showing no wiggles in the phsae space plot - the model should produce stable limit cycles).

What to hand in

- A single Word Document including:
  - written answers (1-2 sentences) and figures for each question above
  - A copy of your R script (the contents of your `.R` file pasted into the Word document)
  - Author contribution statement