A Tour of the Generalized Lotka-Volterra Model

Stefano Allesina

2020-02-05

Contents

Chapter 1

Introduction

These lectures were prepared for the ICTP-SAIFR/IFT-UNESP **"School on Community Ecology: from patterns to principles"**, held in São Paulo, Brazil, January 20-25, 2020.

The four 90-minute lectures provide a semi-rigorous introduction to the Generalized Lotka-Volterra model, along with its applications to community ecology. Throughout the text, I have included code that can be used to simulate the processes that are introduced from a mathematical point of view.

The emphasis is squarely on models for large communities, composed of $n \gg 1$ species. In particular, I often refer to "random" communities, i.e., dynamical systems with random parameters. This allows me to derive interesting results that are meant to describe the "typical" or "expected" community.

At the end of each lecture, I refer to my own work in the area, highlighting some of the strongest contributions from my laboratory at the University of Chicago.

1.1 Prerequisites

The lectures are fairly self-contained, but some working knowledge of calculus, dynamical systems, linear algebra, and game theory would be beneficial.

The code is written in `R`, and meant to be run using a fairly recent version of `R` along with the packages `tidyverse`, `deSolve`, `MASS`, `lpSolve` and `igraph`, which need to be installed.

1.2 Sources

These notes were prepared by summarizing and interpolating a variety of sources, including Hadeler et al. (2017), Hofbauer and Sigmund (1998), and Baigent (2016).

Chapter 2

Generalized Lotka-Volterra

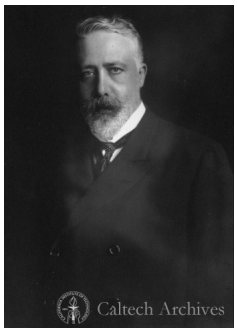
2.1 History

Alfred J. Lotka (1880-1949) was born to French-speaking American parents in Lemberg (then part of the Habsburg empire, now Lviv, Ukraine). He studied in France, Germany and England, receiving a BSc in 1901 and a DSc in 1912 from Birmingham university. He moved to the US in 1902, and worked at the US Patent office, as an editor of Scientific American, and as a statistician at the Metropolitan Life Insturance Company in NYC. He wrote more than a hundred papers and five books, spanning a large range of topics. He's best known for the book *Elements of Physical Biology*, his contributions to demography, and one of the first studies dealing with bibliometrics (Lotka, 1926).

Starting in 1910 (reprinted as Lotka (2002)) he investigated coupled differential equations relating to chemical as well as ecological dynamics. In Lotka (1920) he studied a system of two ODEs that gave rise to perpetual oscillations: *"It was, therefore, with considerable surprise that the writer, on applying his method to certain special cases, found these to lead to undamped, and hence indefinitely continued, oscillations."* He went on to describe *"1. A species of organism* S_1*,*

a plant species, say, deriving its nourishment from a source presented in such large excess that the mass of the source may be considered constant during the period of time with which we are concerned. 2. A species S_2, for example a herbivorous animal species, feeding on S_1."

The equations he had derived (and then studied later in more detail) are now termed Lotka-Volterra equations.



Vito Volterra (1860-1940) was born in Ancona (then part of the Papal State) in a poor jewish family. The financial situation precipitated with the death of his father, when Vito was two. Vito and his mother went to live with relatives in Turin and then Florence. Volterra showed amazing mathematical talent at a very young age. Antonio Roiti, professor of physics in Florence, noticed the budding mathematician and hired him as his assistant, so that he could continue his studies. He went on to enroll at the Scuola Normale in Pisa, receiving a degree in Physics in 1882. At age 23 he was made full professor of Rational Mechanics in Pisa, and then in 1900 of Mathematical Physics in Rome. For thirty years, he contributed important studies in mathematics, and enriched academic life in Italy (for example, he was the first director of the National Center for Research). In 1931 he refused to take an oath of loyalty to the fascist regime (only 12 professors out of 1250 refused), and was therefore forced to resign (his take on the fascist enterprise: *"Empires die, but Euclid's theorems keep their youth forever"*).

His interest in mathematical ecology is due to Umberto D'Ancona (his son-in-law), who had studied the trends in fisheries in the Adriatic sea before and immediately after WWI. In 1914-1918 fisheries in the Adriatic had stopped completely because of the conflict. D'Ancona had noticed that, while herbivorous fish had remained about constant, the piscivorous fish had increased dramatically in numbers. The problem piqued Volterra who immediately published a sophisticated study, proposing the same equations studied by Lotka. In a short letter to Nature (Volterra, 1926a), he stated the so-called "Volterra's Effect" (which he termed "Law III"): *"a complete closure of the fishery was a form of 'protection' under which the voracious fishes were much the better and prospered accordingly, but the ordinary food-fishes, on which these are accustomed to prey, were worse off than before."* This brief paper was a summary of the

much extensive Volterra (1926b).

2.1.1 Lotka-Volterra interactions

In 1927, Lotka wrote to Nature to raise the issue that the equations studied by Volterra and the figures presented in Volterra's brief article were identical to those found in *Elements of Physical Biology* (published in 1925). He concluded: *"It would be gratifying if Prof. Volterra's publication should direct attention to a field and method of inquiry which apparently has hitherto passed almost unnoticed."*

Volterra graciously conceded *"I recognize his priority, and am sorry not to have known his work, and therefore not have been able to mention it."* He however listed a few points in which the two authors had pursued different directions, and concluded *"Working independently the one from the other, we have found some common results, and this confirms the exactitude and the interest in the position of the problem. I agree with him in his conclusions that these studies and these methods of research deserve to receive greated attention from scholars, and should give rise to important applications."*

2.2 Basic formulation

We can write the Generalized Lotka-Volterra model in a compact form as:

$$\frac{dx(t)}{dt} = D(x(t))(r + Ax(t))$$

where $x(t)$ is a (column) vector of length n containing the densities of all populations $1, \ldots, n$ at time t, r is a vector of "intrinsic growth rates" (or death rates, when negative), measuring the growth (decline) of population i when grown alone at low density, and A is a $n \times n$ matrix of interaction coefficients. We use $D(x)$ to denote the diagonal matrix with x on the diagonal.

2.3 A single population

The simplest case to study is that of a single population, in which case the equation becomes that of the logistic growth:

$$\frac{dx(t)}{dt} = x(t)(r + ax(t))$$

This is a separable ODE, with solution:

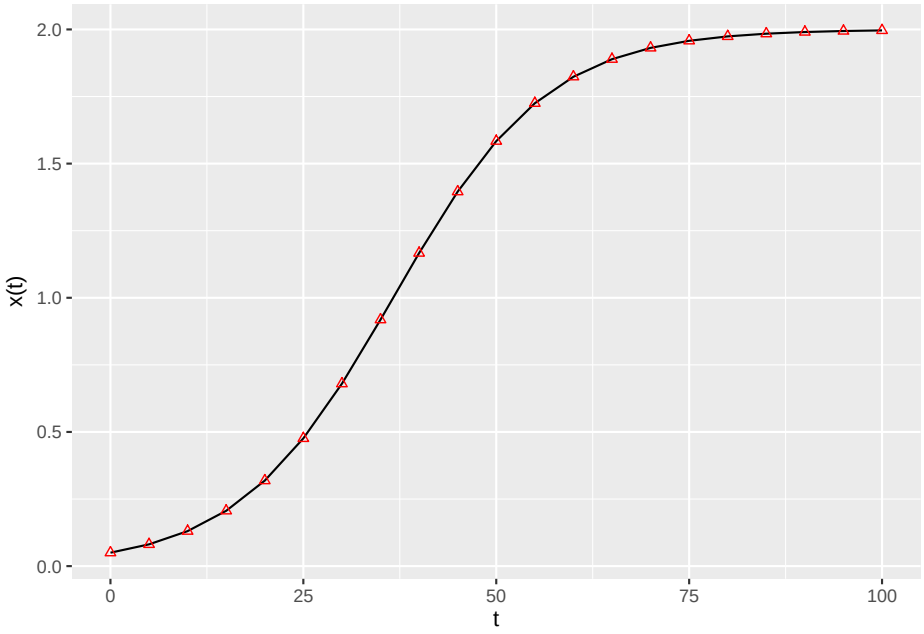
$$x(t) = \frac{r}{e^{-r(k+t)} - a}$$

where k is a constant. Setting $x(0) = x_0$ (i.e., providing an initial condition), solving for the constant and substituting, we obtain:

$$x(t) = \frac{rx_0e^{rt}}{r - ax_0\left(e^{rt} - 1\right)}$$

As such, provided with the parameters r and a, as well as an initial condition, we can determine the population size for any time t. For example, in R:

```
library(deSolve) # integrate ODEs
library(tidyverse) # plotting and wrangling
# define the differential equation
logistic_growth <- function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    dxdt <- x * (r + a * x)
    list(dxdt)
  })
}
# define parameters, integration time, initial conditions
times <- seq(0, 100, by = 5)
x0 <- 0.05
r <- 0.1
a <- -0.05
parameters <- list(r = r, a = a)
# solve numerically
out <- ode(y = x0, times = times,
           func = logistic_growth, parms = parameters,
           method = "ode45")
# now compute analytically
solution <- r * x0 * exp(r * times) / (r - a * x0 * (exp(r * times) - 1))
# use ggplot to plot
res <- tibble(time = out[,1], x_t = out[,2], x_sol = solution)
ggplot(data = res) + aes(x = time, y = x_t) +
  geom_line() +
  geom_point(aes(x = time, y = x_sol), colour = "red", shape = 2) +
  ylab(expression("x(t)")) + xlab(expression("t"))
```



If $a < 0$ and $r > 0$, the population started at any positive value eventually reaches an equilibrium, which we can find by setting $dx(t)/dt = 0$ and considering $x \neq 0$:

$$(r + ax) = 0 \rightarrow x = -\frac{r}{a}$$

2.3.1 Metapopulation dynamics

Consider a fragmented landscape in which habitable patches are connected by dispersal (for simplicity, suppose that all patches are reachable from any other). Call $p(t)$ the proportion of patches occupied by the species of interest at time t, and assume that a) an empty patch (the proportion of empty patches is $1-p(t)$) is colonized by the species with rate $cp(t)$, where c is the "colonization rate", and b) that occupied patches become empty at rate $ep(t)$ ("extinction rate"). We want to model the proportion of patches occupied by the population at time t (Levins, 1969):

$$\frac{dp(t)}{dt} = cp(t)(1 - p(t)) - ep(t) = p(t)((c - e) - cp(t))$$

which is equivalent to the logistic equation above with $r = c - e$ and $a = -c$. As such, asymptotically the proportion of patches occupied by the population will be $-r/a = (c - e)/c$.

2.3.2 S-I-S model

Consider a population of individuals, each of which can be in one of two states: susceptible to a disease, or infective/infected. Call $S(t)$ the proportion of susceptible individuals at time t, and $I(t)$ the proportion of infected individuals, with $S(t)+I(t)=1$. When individuals meet, an infected individual can transmit the disease to susceptibles with rate β; infected individuals recover from the disease with rate γ, and return susceptible. We can write the system of equations:

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)I(t) + \gamma I(t) \\ \dfrac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \end{cases}$$

take the second equation, and substitute $S(t) = 1 - I(t)$; rearranging:

$$\frac{dI(t)}{dt} = \beta(1 - I(t))I(t) - \gamma I(t) = I(t)(\beta - \gamma - \beta I(t))$$

which is again the equation for the logistic growth with $r = \beta - \gamma$ and $a = -\beta$. As such, provided that $\beta - \gamma > 0$, asymptotically a fraction $(\beta - \gamma)/\beta$ of individuals will be infected. The condition $\beta - \gamma > 0 \to \beta > \gamma \to \beta/\gamma > 1$ is often written as $\mathcal{R}_0 = \beta/\gamma > 1$.

2.4 Multi-species dynamics

2.4.1 Existence of an equilibrium

Returning to the multi-species system, and in analogy with the single species, we can look for stationary points (equilibria). If the matrix A is not singular, then we can look for a solution of $r + Ax$ that has positive components (called a **feasible equilibrium**). **If such point exists, it is unique** and is the solution of $Ax^\star = -r$, $x^\star = -A^{-1}r$.

Suppose that the GLV has no feasible equilibrium. Then all trajectories (if bounded; some could grow to infinity) reach the boundary of $\mathbb{R}^n_{0+}$. Practically, this means that **to ensure coexistence of all species, it is necessary to have an equilibrium in the interior** $\mathbb{R}^n_+$.

For a proof, see Theorem 5.2.1 in Hofbauer and Sigmund (1998).

2.4.2 Stability of an equilibrium

Suppose that a feasible equilibrium $x^\star$ exists. Then we can ask whether it is **attractive**, i.e. if trajectories started at initial condition $x(0)$ will eventually

reach $x^\star$. This problem is in general difficult to solve (but see below); as an alternative, we can test for **local asymptotic stability**, i.e., ask whether **the system will return to the equilibrium if perturbed infinitesimally away from it**. In general, whenever we describe an ecological community as a system of nonlinear, autonomous ODEs:

$$\frac{dx_i(t)}{dt} = f_i(x(t)) \ ,$$

we define an equilibrium $x^\star$ as a vector of densities such that:

$$\left.\frac{dx_i}{dt}\right|_{x^\star} = f_i(x^\star) = 0 \quad \forall i$$

A given system might have a multitude of equilibria. When the system is resting at an equilibrium point, it will remain there unless it is perturbed away from it. Local stability analysis is a method to probe whether a system that is perturbed infinitesimally away from an equilibrium will eventually return to it, or rather move away from it.

Suppose that the system is resting at an equilibrium $x^\star$, and that it is slightly perturbed away from it. $\Delta x(0) = x(0) - x^\star$ is the state of the system immediately after the perturbation. We Taylor-expand around $x^\star$:

$$f(\Delta x(0)) = f(x^\star) + J|_{x^\star} \, \Delta x(0) + \dots$$

Where J is the Jacobian matrix of the system, whose elements are defined as:

$$J_{ij} = \frac{\partial f_i(x)}{\partial x_j}$$

Each element of this matrix is therefore a function, whose value depends on x. When we evaluate the Jacobian matrix at an equilibrium point $x^\star$, we obtain the so-called "community matrix" M:

$$M = J|_{x^\star}$$

Note that, although each system has a unique Jacobian matrix, there are as many community matrices as there are equilibria. The community matrix details the effect of increasing the density of one species on any other species around the equilibrium point.

We can therefore write the differential equation:

$$\frac{d\Delta x(t)}{dt} \approx M \Delta x(t)$$

with solution:

$$\Delta x(t) = \Delta x(0) e^{Mt} = \Delta x(0) Q e^{\Lambda t} Q^{-1}$$

Where Q is the matrix containing the (unit) eigenvectors of M, and Λ is a diagonal matrix containing the eigenvalues of M. As such, the eigenvalues of M determine the stability of the equilibrium $x^\star$: if all the eigenvalues have negative real part, then the system will eventually return to the equilibrium after sufficiently small perturbations; conversely, if any of the eigenvalues have positive real part, the system will move away from the equilibrium whenever perturbed. Therefore, depending on the sign of the "rightmost" eigenvalue of M, λ_1, we can determine the stability of $x^\star$:

$$\mathrm{Re}(\lambda_1) \begin{cases} < 0 \to x^\star & \text{is stable} \\ > 0 \to x^\star & \text{is unstable} \end{cases}$$

Local asymptotic stability means that the equilibrium is stable with respect to infinitesimal perturbations ("local"), and that returning to the equilibrium could take a long time ("asymptotic"). Ecologists have also studied stronger forms of stability (e.g., "global stability", in which all trajectories started at positive densities lead to the equilibrium).

For the GLV model, the Jacobian is easy to compute:

$$J_{ij} = \frac{\partial f_i}{\partial x_j} = a_{ij} x_i$$

and

$$J_{ii} = \frac{\partial f_i}{\partial x_i} = r_i + \sum_j a_{ij} x_j + a_{ii} x_i$$

At equilibrium $r_i + \sum_j a_{ij} x_j = 0$, and therefore:

$$M = J|_{x^\star} = D(x^\star) A$$

2.4.3 Types of dynamics

For a single population, there are only three types of dynamics that can be displayed by a GLV model: either the population grows to infinity, shrinks to zero, or asymptotically reaches a steady state.

Smale (1976) and Hirsch (1982) showed that limit cylces are possible for three or more species, and that any dynamics can be found for competitive GLV systems with five or more species.

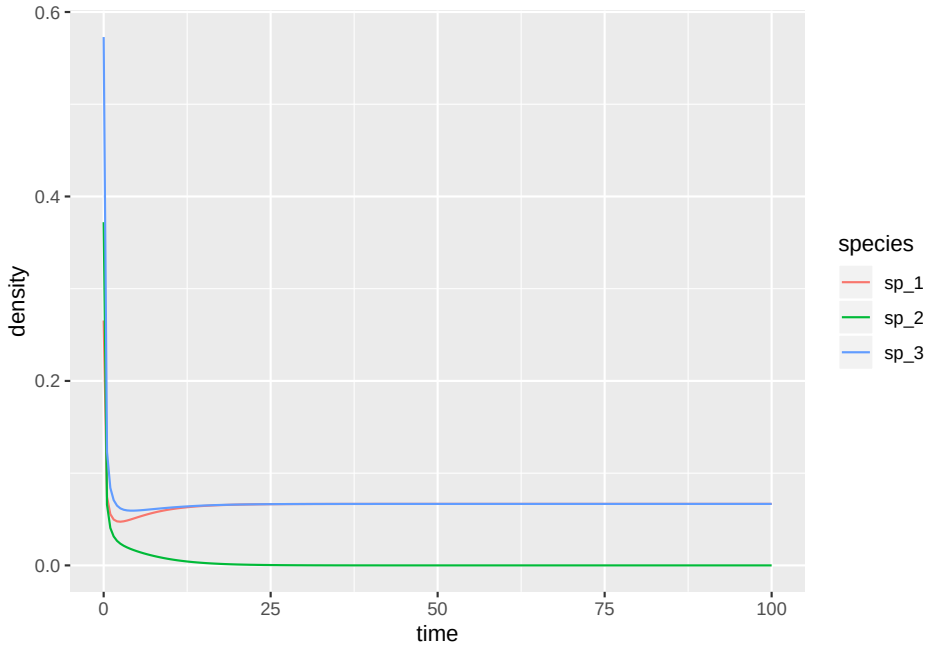
The code necessary for some numerical explorations:

```
# Generalized Lotka-Volterra model
GLV <- function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    x[x < 10^-8] <- 0 # prevent numerical problems
    dxdt <- x * (r + A %*% x)
    list(dxdt)
  })
}
# function to plot output
plot_ODE_output <- function(out){
  out <- as.data.frame(out)
  colnames(out) <- c("time", paste("sp", 1:(ncol(out) -1), sep = "_"))
  out <- as_tibble(out) %>% gather(species, density, -time)
  pl <- ggplot(data = out) +
    aes(x = time, y = density, colour = species) +
    geom_line()
  show(pl)
  return(out)
}
# general function to integrate GLV
integrate_GLV <- function(r, A, x0, maxtime = 100, steptime = 0.5){
  times <- seq(0, maxtime, by = steptime)
  parameters <- list(r = r, A = A)
  # solve numerically
  out <- ode(y = x0, times = times,
             func = GLV, parms = parameters,
             method = "ode45")
  # plot and make into tidy form
  out <- plot_ODE_output(out)
  return(out)
}
```

A few examples taken from Barabás et al. (2016). First, a competitive system in which a feasible equilibrium does not exist, leading to the extinction of a species:

```
set.seed(1) # for reproducibility
r_1 <- rep(1, 3)
A_1 <- -matrix(c(10, 9, 5,
                 9, 10, 9,
                 5, 9, 10), 3, 3, byrow = TRUE)
```

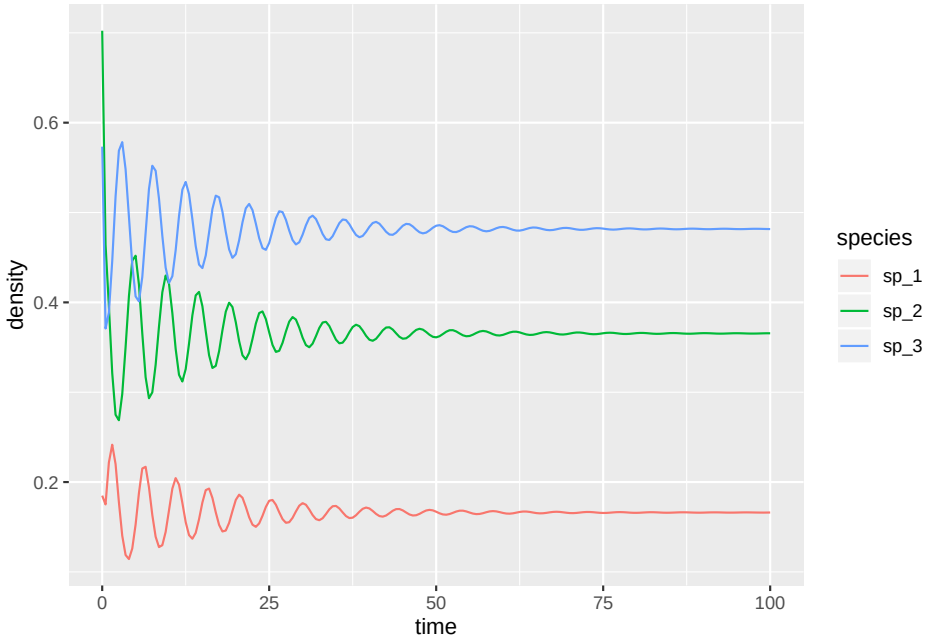
```r
# check the existence of feasible equilibrium
print(solve(A_1, -r_1)) # not feasible
x0_1 <- runif(3)
res_1 <- integrate_GLV(r_1, A_1, x0_1)
```




```
# [1] -0.08333333  0.25000000 -0.08333333
```

Then, a case in which the equilibrium exists, and is attractive (stable):

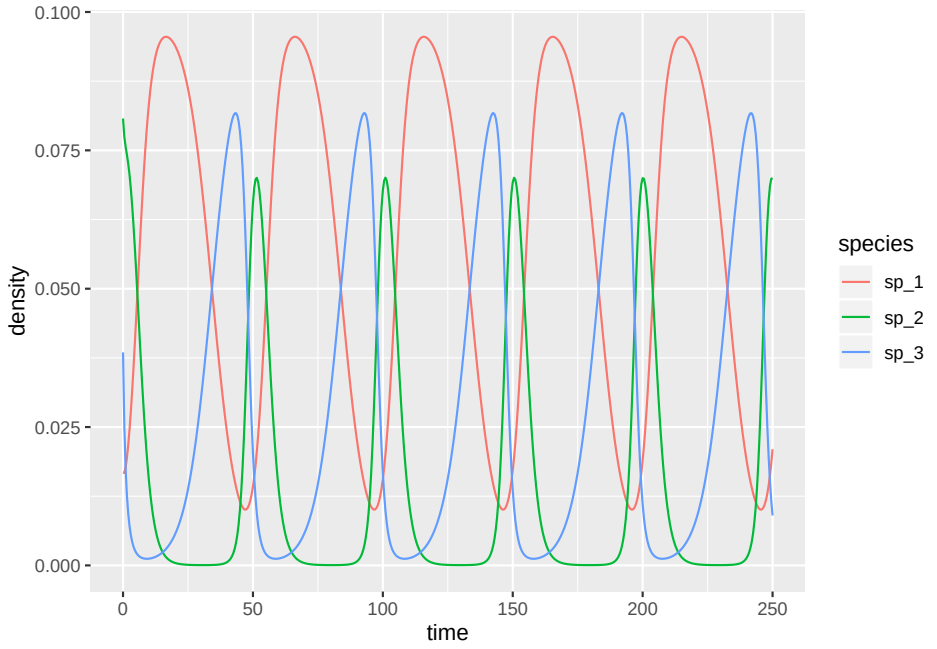
```r
set.seed(2) # for reproducibility
r_2 <- rep(10, 3)
A_2 <- -matrix(c(10, 7, 12,
                 15, 10, 8,
                 7, 11, 10), 3, 3, byrow = TRUE)
# check the existence of feasible equilibrium
print(solve(A_2, -r_2)) # feasible
x0_2 <- runif(3)
res_2 <- integrate_GLV(r_2, A_2, x0_2)
```




```
# [1] 0.1661130 0.3654485 0.4817276
```

With three competitors we can find stable limit cycles:

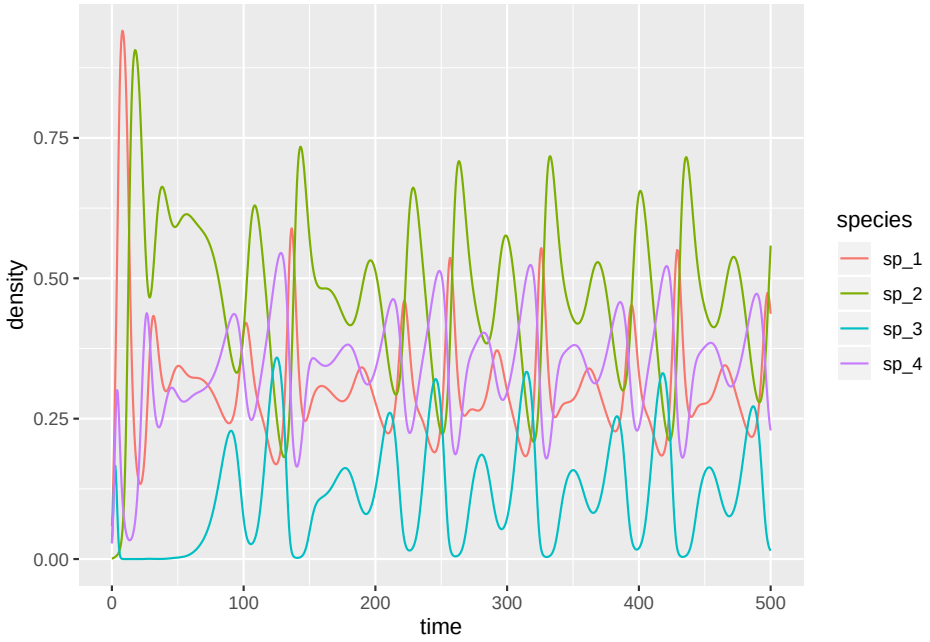
```
set.seed(3) # for reproducibility
r_3 <- rep(1, 3)
A_3 <- -matrix(c(10, 6, 12,
                 14, 10, 2,
                 8, 18, 10), 3, 3, byrow = TRUE)
# check the existence of feasible equilibrium
print(solve(A_3, -r_3)) # feasible
x0_3 <- 0.1 * runif(3)
res_3 <- integrate_GLV(r_3, A_3, x0_3, maxtime = 250)
```




```
# [1] 0.05714286 0.01428571 0.02857143
```

And with four or more species we can have chaos:

```
set.seed(4) # for reproducibility
r_4 <- c(1, 0.72, 1.53, 1.27)
A_4 <- -matrix(c(1, 1.09, 1.52, 0,
                 0, 0.72, 0.3168, 0.9792,
                 3.5649, 0, 1.53, 0.7191,
                 1.5367, 0.6477, 0.4445, 1.27), 4, 4, byrow = TRUE)
# check the existence of feasible equilibrium
print(solve(A_4, -r_4)) # feasible
x0_4 <- 0.1 * runif(4)
res_4 <- integrate_GLV(r_4, A_4, x0_4, maxtime = 500)
```




```
# [1] 0.3013030 0.4586546 0.1307655 0.3557416
```

2.4.4 The equilibrium is the time-average

Suppose that $x(t)$ has a periodic orbit of period T (we assume $x(0) = x(T)$). Further, assume that the GLV has an interior equilibrium $x^\star$. We want to calculate the average density for each species:

$$\frac{1}{T}\int_0^T x(t)dt$$

We perform the change of variable $y = \log(x(t))$, finding $\frac{dx(t)}{dt} = \frac{dx(t)}{d\log(x(t))}\frac{d\log(x(t))}{dt} = x(t)\frac{d\log(x(t))}{dt}$. Then:

$$\frac{d\log(x(t))}{dt} = r + Ax(t)$$

Compute the average on both sides:

$$\frac{1}{T}\int_0^T \frac{d\log(x(t))}{dt}dt = \frac{1}{T}\int_0^T (r + Ax)\,dt$$

yielding:

$$\frac{1}{T}(\log(x(T)) - \log(x(0))) = 0 = r + A\left(\frac{1}{T}\int_0^T x(t)dt\right)$$

multiplying by the matrix inverse and rearranging:

$$-A^{-1}r = x^\star = \frac{1}{T}\int_0^T x(t)dt$$

showing that **the average density is** in fact **the equilibrium**. With a similar argument, one can prove that if the trajectory stays in a compact space (i.e., in case of chaotic attractors), then long-time average is still $x^\star$.

For example, let's compute the average for the system with the limit cycle above (discarding the transients in the first part of the time series):

```
res_3 %>% filter(time > 50) %>%
  group_by(species) %>%
  summarize(average = mean(density))
# compare with equilibrium
solve(A_3, -r_3)
```

```
# # A tibble: 3 x 2
#   species average
#   <chr>     <dbl>
# 1 sp_1     0.0568
# 2 sp_2     0.0147
# 3 sp_3     0.0284
# [1] 0.05714286 0.01428571 0.02857143
```

Repeat for the chaotic system:

```
res_4 %>% filter(time > 50) %>%
  group_by(species) %>%
  summarize(average = mean(density))
# compare with equilibrium
solve(A_4, -r_4)
```

```
# # A tibble: 4 x 2
#   species average
#   <chr>     <dbl>
# 1 sp_1      0.303
# 2 sp_2      0.463
# 3 sp_3      0.126
# 4 sp_4      0.354
# [1] 0.3013030 0.4586546 0.1307655 0.3557416
```

2.4.5 Lyapunov diagonal stability and global stability

Suppose that there is a positive diagonal matrix C such that $CA + A^tC$ is negative definite (i.e., has only negative eigenvalues; the eigenvalues are real because the matrix is symmetric). Then A is **Lyapunov-diagonally stable**. If this is the case, then A is stable, and any DA with D positive is also stable (called $D-$stability).

Further, suppose that the GLV system with parameters A and r has a feasible equilibrium point $x^\star$. Then the function:

$$V(x(t)) = 1^tC\left(x(t) - D(x^\star)\log x(t)\right)$$

is a Lyapunov function for the GLV system.

To prove this point, we start from $r = -Ax^\star$. Substituting, we can write the GLV system as $dx(t)/dt = D(x)A(x-x^\star)$; similarly, we can write $d\log x(t)/dt = A(x-x^\star)$. Taking the derivative of V with respect to time:

$$\begin{aligned}
\frac{dV(x(t))}{dt} &= 1^tC\left(\frac{dx(t)}{dt} + D(x^*)\frac{d\log x(t)}{dt}\right)\\
&= 1^tC\left(A(x-x^\star) + D(x^*)A(x-x^\star)\right)\\
&= 1^tC\left(D(x-x^\star)A(x-x^\star)\right)\\
&= 1^t\left(D(x-x^\star)CA(x-x^\star)\right)\\
&= (x-x^\star)^tCA(x-x^\star)\\
&= \frac{1}{2}(x-x^\star)^t(CA + A^tC)(x-x^\star)
\end{aligned}$$

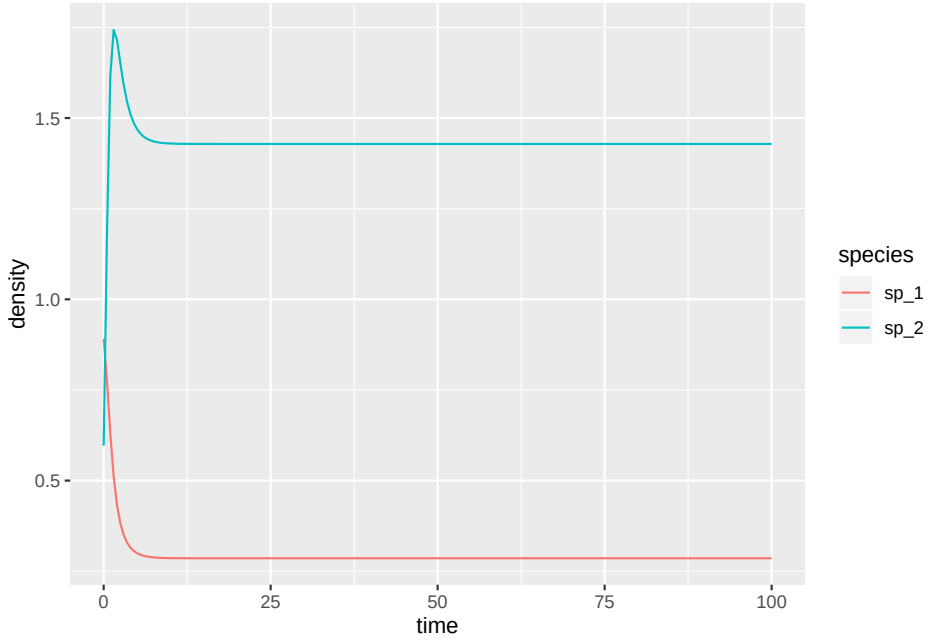
A matrix B is negative definite if $y^tBy < 0$ for all $y \neq 0$. As such $\frac{dV(x(t))}{dt} \leq 0$, i.e., will decrease in time (starting from any $x(0)$) until the equilibrium $x^\star$ is reached.

The results above show that **if A is Lyapunov diagonally-stable and a feasible equilibrium $x^\star$ exists, than all trajectories starting at a positive density will converge to the equilibrium**. This property is often used to prove **global stability**.

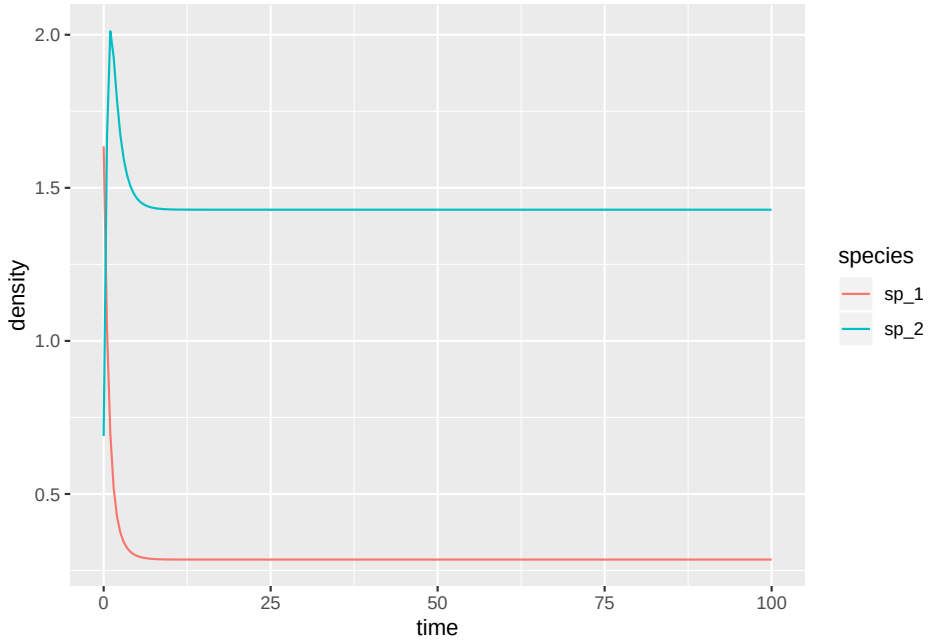
A simple numerical example with two species:

```
r <- c(1, 1)
A <- matrix(c(-1, 3/2, -1/2, -1), 2, 2)
# the eigenvalues of A + A^t are -1 and -3, hence A is Lyapunov Diagonally-stable
eigen(A + t(A))$values
# now integrate from random initial condition---this should always
# converge to the equilibrium
```

```
solve(A, -r)
res <- integrate_GLV(r, A, abs(rnorm(2)))
```




```
res <- integrate_GLV(r, A, abs(rnorm(2)))
```



```
# [1] -1 -3
# [1] 0.2857143 1.4285714
```

2.5 Stability of large random communities

As we have seen above, an equilibrium $x^\star$ is stable if the community matrix for the equilibrium has all eigenvalues with negative real part. In general, to determine the equilibrium and its stability, we would need to specify all the growth rates (r, n values), as well as the matrix of interactions (A, n^2 values). This is impractical to do for large systems (though we will try this out later). But can something quite general be said about the limit in which many species are in the community?

May (1972) attempted to answer this question by considering a **random community matrix**. In a GLV system, the diagonal elements $m_{ii} = a_{ii}x_i^\star$ are influenced by self-regulation (i.e., as in a carrying capacity), while the off-diagonal elements $m_{ij} = a_{ij}x_i^\star$ model the effect of species j on the equilibrium of species i. May considered the following algorithm to build a random community matrix. Take a large community, resting at an unspecified, feasible equilibrium; we build the community matrix by setting:

- $m_{ij} = 0$ with probability $(1 - C)$; with probability C we draw m_{ij} from a distribution with mean zero and variance σ^2. C is the proportion of realized connections, termed the "connectance" of the system.
- the diagonal elements are set to $-d$, modeling self-regulation.

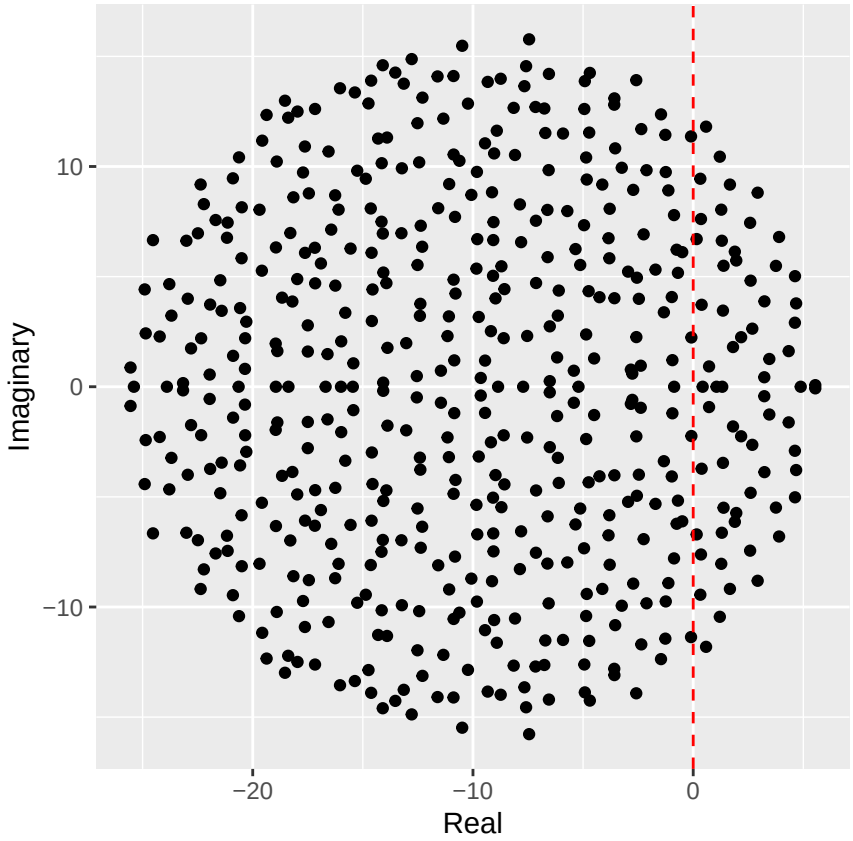
For example, the code uses a normal distribution:

```
build_May_normal <- function(n, C, d, sigma){
  # fill the whole matrix
  M <- matrix(rnorm(n * n, mean = 0, sd = sigma), n, n)
  # remove connections
  M <- M * matrix(runif(n * n) <= C, n, n)
  # set diagonals
  diag(M) <- -d
  return(M)
}
```

Note that the average of the eigenvalues of a matrix A is given by the average of its diagonal elements $\frac{1}{n}\sum_i \lambda_i = \frac{1}{n}\text{Tr}(A) = \frac{1}{n}\sum_i A_{ii}$. As such, if $A = dI + B$, the eigenvalues of A will be those of B shifted by d.

We want to determine whether the equilibrium will be stable, given n, C, d and σ^2. To do so, we need to find the location of the "rightmost" eigenvalue of M. For example, let's plot the eigenvalues of a large matrix:

```
plot_eigenvalues <- function(M, prediction = NULL){
  eig <- eigen(M, only.values = TRUE)$values
  dt <- tibble(Real = Re(eig), Imaginary = Im(eig))
  pl <- ggplot(dt) + aes(x = Real, y = Imaginary) +
    geom_point() +
    coord_equal() +
    geom_vline(xintercept = 0, colour = "red", linetype = 2)
  if (is.null(prediction) == FALSE) {
    pl <- pl + geom_vline(xintercept = prediction, colour = "black", linetype = 2)
  }
  show(pl)
}
set.seed(100) # for reproducibility
# parameters
n <- 500
C <- 0.5
d <- 10
sigma <- 1
M <- build_May_normal(n, C, d, sigma)
plot_eigenvalues(M)
```



The eigenvalues fall into an almost perfect circle! Turns out, that this is the behavior we should expect, as stated by the so-called "Circular Law", one of the most beautiful results in random matrix theory.

Circular law: Take a non-symmetric, $S \times S$ random matrix in which all coefficients X_{ij} are i.i.d. random variables with $\mathbb{E}[X_{ij}] = 0$ and $\mathbb{E}[X_{ij}] = 1$. Then, as $S \to \infty$, the e.s.d. of $X/\sqrt{S}$ converges to the circular law:

$$\mu(\lambda) = \begin{cases} \frac{1}{\pi} & \text{if } (\mathrm{Re}(\lambda))^2 + (\mathrm{Im}(\lambda))^2 \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

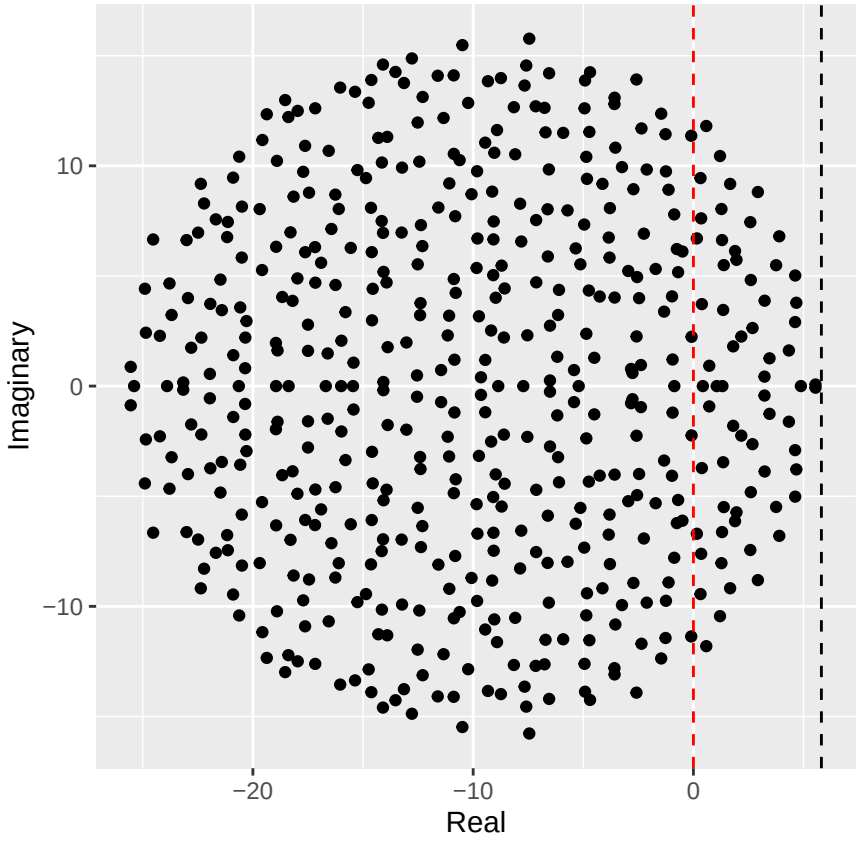
This result can be used to calculate the radius of the eigenvalue distribution of the matrices studied by May: when the off-diagonal coefficients M_{ij} are 0 with probability $1 - C$ and are sampled independently from a distribution with mean 0 and variance σ^2 with probability C, we have that $\mathbb{E}[M_{ij}] = 0$ and $\mathbb{E}[M_{ij}^2] = C\sigma^2$. This means that if we were to divide the coefficients of M by $\sqrt{C\sigma^2}$ we would recover the unit variance, and the matrix would follow the circular law when S is large. Armed with this, we can calculate the radius: if the

radius of $M/\sqrt{SC\sigma^2}$ converges to 1 when the matrix is large, then the radius of M is approximately $\sqrt{SC\sigma^2}$. For stability, we need a sufficiently negative diagonal (setting the center of the circle), yielding May's stability criterion:

$$\sqrt{SC\sigma^2} < d$$

We can try this on our matrix:

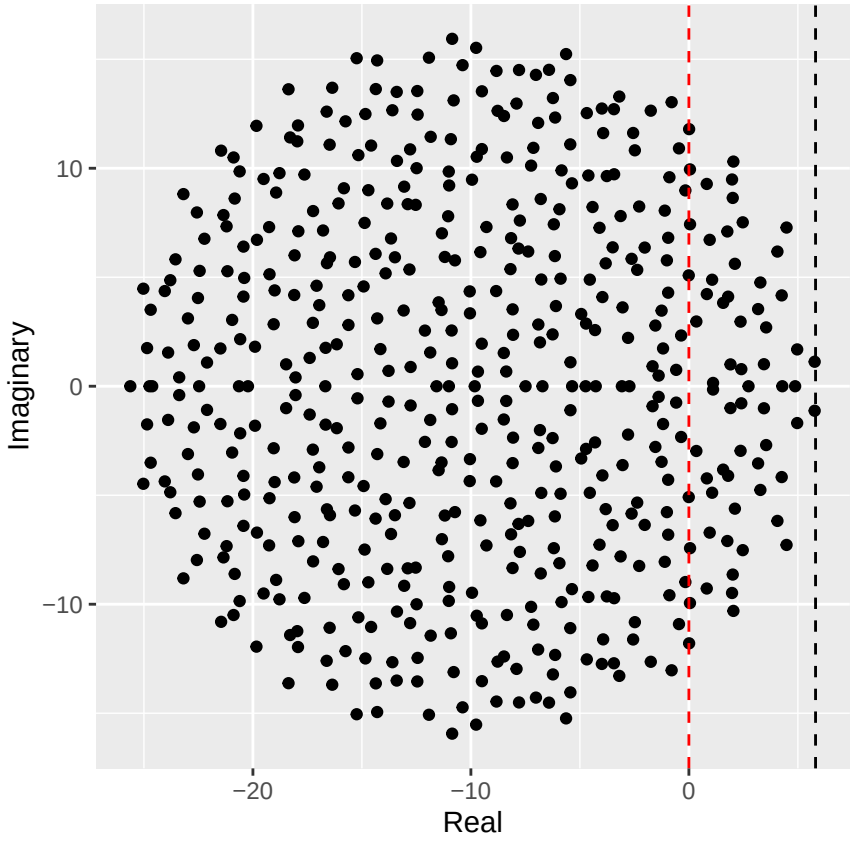
```
prediction <- sqrt(n * C * sigma^2) - d
plot_eigenvalues(M, prediction)
```



Showing that we accurately approximate the location of the rightmost eigenvalue. Note that, in the case of large n, whenever the circle crosses zero, some eigenvalues will be positive, determining the instability of the equilibrium.

Importantly, the distribution from which the coefficients are sampled does not matter—only that the mean is zero and that the variance is σ^2. For example, build the matrix using coefficients from a uniform distribution:

```r
build_May_uniform <- function(n, C, d, sigma){
  # fill the whole matrix (sqrt(3) to ensure var = sigma^2)
  M <- matrix(runif(n * n, min = -sqrt(3) * sigma, max = sqrt(3) * sigma), n, n)
  # remove connections
  M <- M * matrix(runif(n * n) <= C, n, n)
  # set diagonals
  diag(M) <- -d
  return(M)
}
# parameters
n <- 500
C <- 0.5
d <- 10
sigma <- 1
M <- build_May_uniform(n, C, d, sigma)
prediction <- sqrt(n * C * sigma^2) - d
plot_eigenvalues(M, prediction)
```



This property is called **universality** in random matrix theory.

In ecological communities, the effect of species i on j and that of j on i are typically not independent (as assumed above): in the case of competition between species, we expect them both to be negative; for consumption, if one is positive, the other is negative, and so forth. A more refined model of a random matrix would therefore sample interactions in pairs from a bivariate distribution. The elliptic law can deal with this case:

Elliptic law: Take a non-symmetric, $S \times S$ random matrix in which the pairs of coefficients (X_{ij}, X_{ji}) are sampled independently from a bivariate distribution defined by a vector of means $m = (0, 0)^t$ and a covariance matrix $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. Then, as $S \to \infty$, the e.s.d. of $X/\sqrt{S}$ converges to the elliptic law:

$$\mu(\lambda) = \begin{cases} \frac{1}{\pi(1-\rho^2)} & \text{if } \frac{(\mathrm{Re}(\lambda))^2}{(1+\rho)^2} + \frac{(\mathrm{Im}(\lambda))^2}{(1-\rho)^2} \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that when $\rho = 0$, the elliptic law reduces to the circular law. Using the elliptic law, Allesina and Tang (2012) were able to extend May's criterion to ecological networks with different mixtures of interaction types. In particular, the stability criterion becomes:

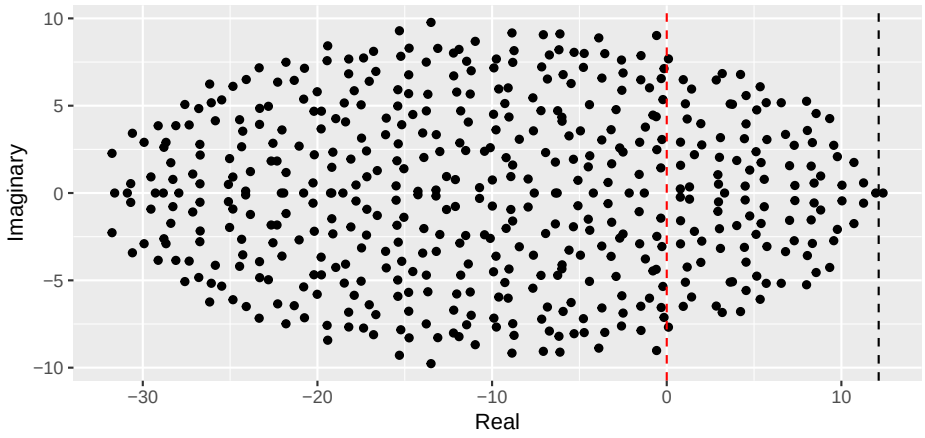
$$\sqrt{SC\sigma^2}(1 + \rho) < d$$

To see the elliptic law in action, we can build matrices in which we sample the coefficients in pairs from a bivariate normal distribution:

```
build_Allesina_Tang_normal <- function(n, C, d, sigma, rho){
  # sample coefficients in pairs
  pairs <- MASS::mvrnorm(n = n * (n-1) / 2,
                         mu = c(0, 0),
                         Sigma = sigma^2 * matrix(c(1, rho, rho, 1), 2, 2))
  # build a completely filled matrix
  M <- matrix(0, n, n)
  M[upper.tri(M)] <- pairs[,1]
  M <- t(M)
  M[upper.tri(M)] <- pairs[,2]
  # determine which connections to retain
  Connections <- (matrix(runif(n * n), n, n) <= C) * 1
  Connections[lower.tri(Connections)] <- 0
  diag(Connections) <- 0
  Connections <- Connections + t(Connections)
  M <- M * Connections
  diag(M) <- -d
  return(M)
}
```

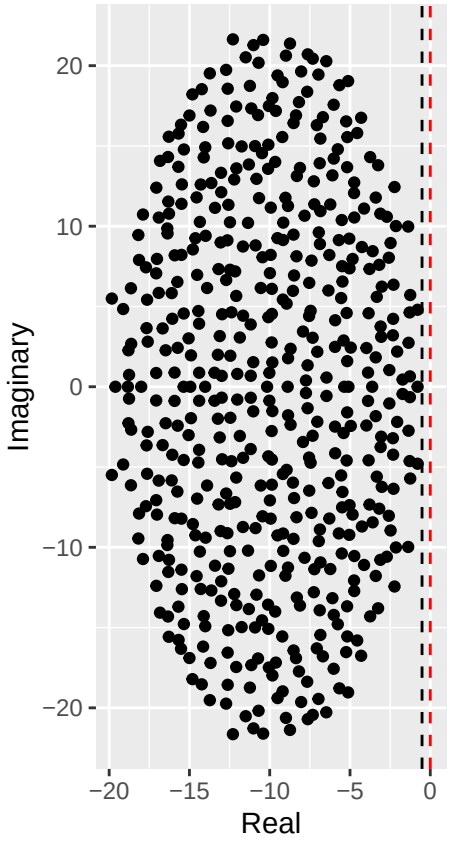
We can see that a positive connectance leads to an eigenvalue distribution that describes an horizontally-stretched ellipse (and hence, more difficult to stabilize than the circle):

```
# parameters
n <- 500
C <- 0.5
d <- 10
sigma <- 1
rho <- 0.4
M <- build_Allesina_Tang_normal(n, C, d, sigma, rho)
prediction <- sqrt(n * C * sigma^2) * (1 + rho) - d
plot_eigenvalues(M, prediction)
```



Similarly, a negative correlation (e.g., as in predator-prey) would make the system easier to stabilize:

```
# parameters
n <- 500
C <- 0.5
d <- 10
sigma <- 1
rho <- -0.4
M <- build_Allesina_Tang_normal(n, C, d, sigma, rho)
prediction <- sqrt(n * C * sigma^2) * (1 + rho) - d
plot_eigenvalues(M, prediction)
```



2.6 Further readings

On the theory of GLV:

- Strogatz (2018) is a simple introduction to dynamical systems and bifurcation theory.
- Hofbauer and Sigmund (1998) is a wonderful introduction to dynamical systems in ecology and population genetics, with a nice introduction to evolutionary game theory.
- Hadeler et al. (2017) contains a more mathematically-oriented treatment of the material covered in the first part of this lecture.
- Baigent (2016) is a mathematical introduction to Lotka-Volterra dynamics.

On random matrices and stability:

- Allesina and Tang (2015) is an opinionated review on applications of random matrices in ecology.

Chapter 3

Multispecies GLV: assembly and invasibility

We have seen in the previous lecture that a large ecological community will almost invariably be unstable, possibly leading to extinctions. It is therefore natural to ask what happens when either a feasible equilibrium does not exist, or it is unstable. In particular, we want to know how many species can persist.

3.1 Assembly

We call **assembly the process by which a community is constituted by adding species sequentially**. Imagine starting with a bare environment, and introducing the first species. If the species can grow in the new environment, it will establish, and if not, it will go extinct. Mathematically, we assume that **new species are introduced at very low abundance**, so that they do not experience self-limitation due to crowding, and that **invasion are spaced in time** so that the dynamics can play out before the next invasion happens. For example, consider the first species entering the system, and write the equation for the per-capita growth rate:

$$\frac{1}{x_i(t)}\frac{dx_i(t)}{dt} = r_i + A_{ii}x_i(t)$$

If $x_i(0) \ll 1$, we can set $A_{ii}x_i(0) \approx 0$, obtaining

$$\frac{1}{x_i(t)}\frac{dx_i(t)}{dt} \approx r_i$$

That is, the species will establish if it has a positive growth rate in the new environment. Suppose that this is the case: then species i will grow to its equilibrium abundance $x_i^\star = -r_i/A_{ii}$. Now add a second species. Its initial per-capita growth rate is going to be:

$$\frac{1}{x_j(t)}\frac{dx_j(t)}{dt} = r_j + \sum_k A_{jk}x_k(t) \approx r_j + A_{ji}x_i^\star$$

Species j can therefore growth when rare if $r_j + A_{ji}x_i^\star > 0$, i.e., $r_j > -A_{ji}x_i^\star$. We call this type of inequality an **"invasion criterion"**.

If the species j can grow when rare, in general, it could a) grow initially, but then go extinct; b) displace species i, sending it to extinction; c) coexist with species i. By reiterating invasions with different species, we can assemble a large ecological community.

3.2 Invasion in multispecies communities

Now consider a pool of species (e.g., a metacommunity) and an environment (e.g., an island) in which some of the species are present, and coexisting at an equilibrium. We have n species in the pool, and k species in the environment. We want to write conditions for the invasibility of the equilibrium. To this end, we can re-arrange the rows and columns of A, and the elements of r to obtain two blocks: one for the k species already in the community, and one for the $n-k$ potential invaders.

The fixed point $\bar{x}$ can be written as

$$\bar{x} = \begin{pmatrix} x^{(k)} \\ 0^{(n-k)} \end{pmatrix}$$

where $x^{(k)}$ contains the density of the coexisting species. Similarly, the growth rates are

$$r = \begin{pmatrix} r^{(k)} \\ r^{(n-k)} \end{pmatrix}$$

and the interaction matrix

$$A = \begin{pmatrix} A^{(k,k)} & A^{(k,n-k)} \\ A^{(n-k,k)} & A^{(n-k,n-k)} \end{pmatrix}$$

where each block $A^{(a,b)}$ contains the effects of the species in set b on the growth of the species in a. For feasibility we need:

$$x^{(k)} = -r^{(k)} \left(A^{(k,k)} \right)^{-1} > 0$$

Now we want to write the condition for the non-invasibility of the resident community by the other species in the metacommunity. For each species in $(n-k)$, we need to have a negative growth rate when invading:

$$r^{(n-k)} + A^{(n-k,k)} x^{(k)} < 0$$

an equilibrium $\bar{x}$ for which a) the $x^{(k)}$ is feasible and stable (when considering only the species in k) and b) no other species can invade when rare is called the **feasible, stable, non-invasible solution** or the **saturated equilibrium**.

3.3 Lyapunov stability and saturated equilibria

We have seen in the previous lecture that whenever there exists a matrix C such that $CA + A^t C$ is negative definite and we have a feasible equilibrium $x^\star$ then the equilibrium is globally stable. Now we consider the case in which a feasible equilibrium does not exist. However, we can prove that a **saturated equilibrium** exists and is unique: we have an equilibrium $\bar{x}$ in which some components are positive and some are zero such that k species coexist at a globally stable equilibrium, and the remaining $(n-k)$ species cannot invade it.

Following Hofbauer and Sigmund (1998) (section 15.3), we write:

$$V(x) = -\sum_i c_i (\bar{x}_i \log x_i - x_i)$$

yielding:

$$\begin{aligned}
\frac{V(x)}{dt} &= -\sum_i c_i \left(\bar{x}_i \frac{d \log x_i}{dt} - \frac{dx_i}{dt} \right) \\
&= -\sum_i c_i \left(\bar{x}_i - x_i \right) \left(r_i + \sum_j A_{ij} x_j \right) \\
&= -\sum_i c_i \left(\bar{x}_i - x_i \right) \left(r_i + \sum_j A_{ij} (x_j - \bar{x}_j) + \sum_j A_{ij} \bar{x}_j \right) \\
&= \sum_{i,j} \left(\bar{x}_i - x_i \right) c_i A_{ij} \left(\bar{x}_j - x_j \right) - \sum_i c_i \left(\bar{x}_i - x_i \right) \left(r_i + \sum_j A_{ij} \bar{x}_j \right)
\end{aligned}$$

The first term is negative for all $x \neq \bar{x}$, and is zero at the saturated equilibrium point. The second term is zero for all $\bar{x}_i > 0$: $r_i + \sum_j A_{ij}\bar{x}_j = 0$, given that $\bar{x}$ is an equilibrium for the species at positive density; for the remaining species (for which $\bar{x}_i = 0$), this amounts to the invasion criterion above, and must therefore be negative. As such $V(x)$ is a Lyapunov function for the system, assuming negative values everywhere but at $\bar{x}$.

3.4 How many species will coexist?

Here we investigate the simplest case in which A is symmetric and stable (and therefore Lyapunov Diagonally stable) matrix with random coefficients, and r is a vector of growth rates with random coefficients. For simplicity, we take the off-diagonal elements of A and the growth rates from a normal distribution centered at zero (the same would be found for any distribution symmetric about zero). This case was studied by Serván et al. (2018).

First, we are going to consider a trivial case, then simulate a more complex one, and finally outline the proof in Serván et al. (2018).

3.4.1 A trivial case

Suppose that the growth rates are drawn from a normal distribution centered at zero, and that the matrix A is diagonal and stable (i.e., species do not interact with each other; each species is self regulating). As such, a species will persist if and only if $r_i > 0$. If we draw the growth rates from any distribution centered at zero, then the probability of having a positive growth rate is $1/2$. Therefore, the number of coexisting species will follow the binomial distribution with parameters n and $1/2$.

3.4.2 Simulating final composition

Now, let's simulate cases in which species do interact with each other. First, we load the functions we've written before:

```
library(tidyverse)
library(deSolve)
# Generalized Lotka-Volterra model
GLV <- function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    x[x < 10^-8] <- 0 # prevent numerical problems
    dxdt <- x * (r + A %*% x)
    list(dxdt)
  })
```

```
}
# function to plot output
plot_ODE_output <- function(out){
  out <- as.data.frame(out)
  colnames(out) <- c("time", paste("sp", 1:(ncol(out) -1), sep = "_"))
  out <- as_tibble(out) %>% gather(species, density, -time)
  pl <- ggplot(data = out) +
    aes(x = time, y = density, colour = species) +
    geom_line()
  show(pl)
  return(out)
}
# general function to integrate GLV
integrate_GLV <- function(r, A, x0, maxtime = 100, steptime = 0.5){
  times <- seq(0, maxtime, by = steptime)
  parameters <- list(r = r, A = A)
  # solve numerically
  out <- ode(y = x0, times = times,
             func = GLV, parms = parameters,
             method = "ode45")
  # plot and make into tidy form
  out <- plot_ODE_output(out)
  return(out)
}
```

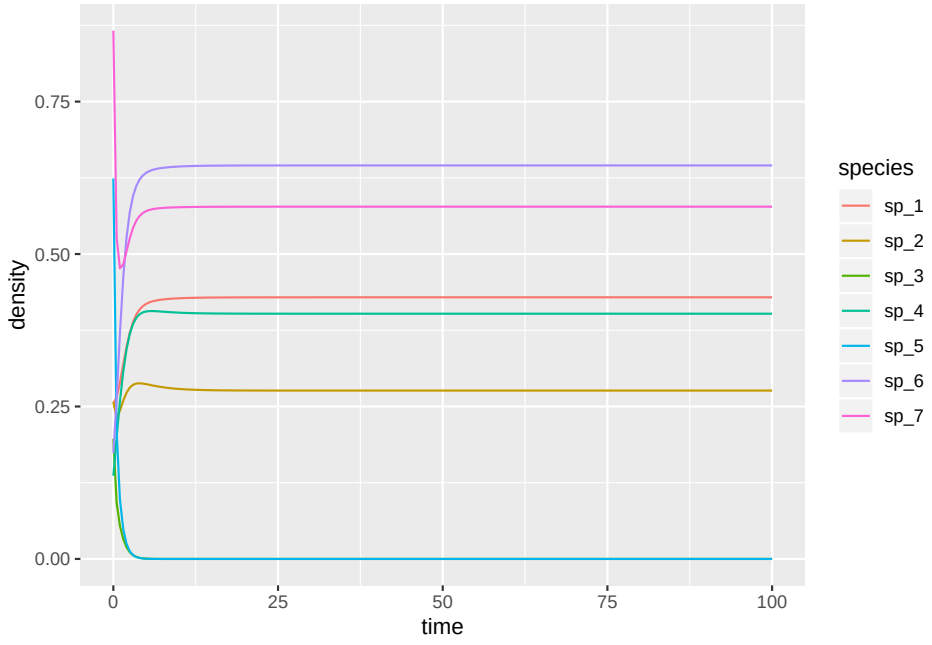
And then write functions to give us a (barely) stable, symmetric matrix A, and random growth rates:

```
# function to build symmetric, Lyapunov Diagonally-stable matrix
build_LDstable <- function(n){
  A <- matrix(0, n, n)
  A[upper.tri(A)] <- rnorm(n * (n - 1) / 2)
  # make symmetric
  A <- A + t(A)
  # now find the largest eigenvalue
  l1A <- max(eigen(A, only.values = TRUE, symmetric = TRUE)$values)
  if (l1A > 0){
    # set the diagonal to make it stable
    diag(A) <- diag(A) - l1A - 0.01
  }
  return(A)
}

# function to get random growth rates
build_randomr <- function(n){
  return(rnorm(n))
```

```
}
```

Now, we build a random system with seven species, and integrate the dynamics:

```
set.seed(5) # for reproducibility
n <- 7
A <- build_LDstable(n)
r <- build_randomr(n)
x0 <- runif(n)
out <- integrate_GLV(r, A, x0)
```

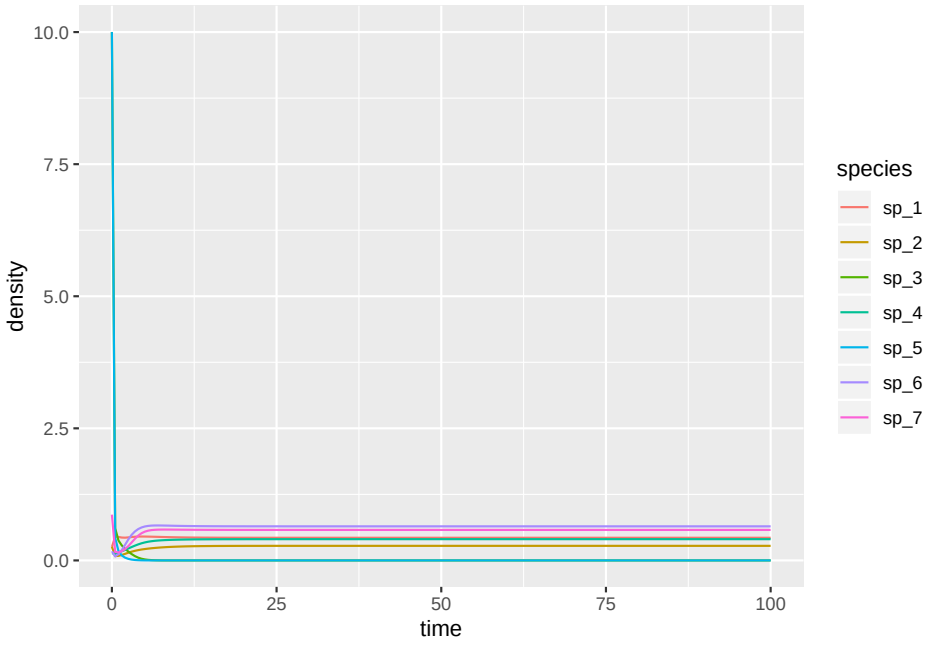



```
out %>%
  filter(out$time == max(out$time)) %>%
  mutate(density = ifelse(density < 10^-6, 0, density))
```

```
# # A tibble: 7 x 3
#    time species density
#   <dbl> <chr>     <dbl>
# 1   100 sp_1      0.429
# 2   100 sp_2      0.276
# 3   100 sp_3      0
# 4   100 sp_4      0.402
# 5   100 sp_5      0
# 6   100 sp_6      0.645
# 7   100 sp_7      0.578
```

As you can see, in this case, two species (3 and 5) go extinct, while the other ones reach a feasible equilibrium. Let's try to start the system with species 3 and 5 at high abundance, to show that the equilibrium is indeed globally stable:

```
x0[3] <- 10
x0[5] <- 10
out <- integrate_GLV(r, A, x0)
```




```
out %>%
  filter(out$time == max(out$time)) %>%
  mutate(density = ifelse(density < 10^-6, 0, density))
```

```
# # A tibble: 7 x 3
#    time species density
#   <dbl> <chr>     <dbl>
# 1   100 sp_1      0.429
# 2   100 sp_2      0.276
# 3   100 sp_3      0
# 4   100 sp_4      0.402
# 5   100 sp_5      0
# 6   100 sp_6      0.645
# 7   100 sp_7      0.578
```

As you can see, the system still goes to the same equilibrium, with 3 and 5 extinct. Turns out, for this type of system, one does not even need to integrate dynamics: the Lemke–Howson algorithm can be adapted to solve the problem efficiently (Serván et al., 2018). I have coded up the algorithm already, and you

can load the function `get_final_composition(A, r)` by typing:

```
source("https://raw.githubusercontent.com/StefanoAllesina/Sao_Paulo_School/master/code/
get_final_composition(A, r) # use LH instead of integrating dynamics
```

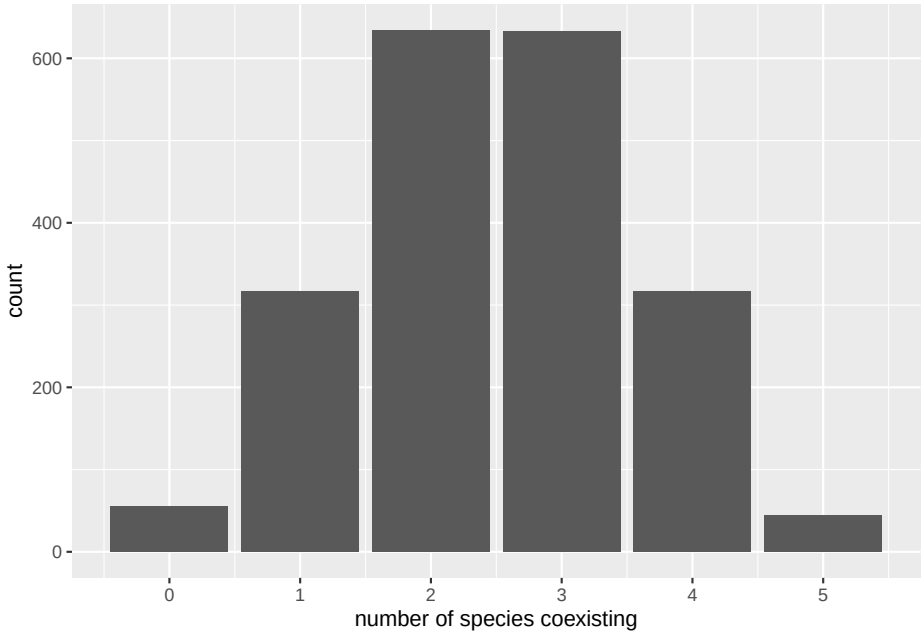
```
# [1] 0.4290300 0.2761030 0.0000000 0.4021761 0.0000000 0.6453202 0.5776840
```

3.5 A random zoo

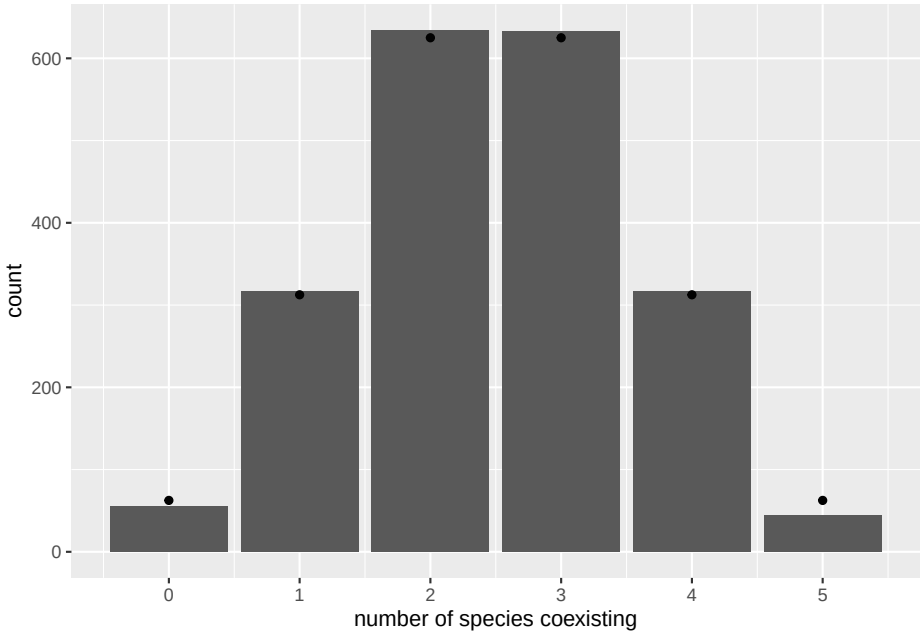
In the case above, 5 species persisted. We can ask how many species will coexist in general under this parametrization. A good metaphor is that of the random zoo: take a large zoo, open all the cages, and return after fifty years. How many species will you find?

To simulate the random zoo, we take random matrices and random growth rates with n species, and tally the number of coexisting species over several simulations:

```
set.seed(1)
n <- 5
nsim <- 2000
results <- data.frame(simulation = 1:nsim, ncoexisting = rep(NA, nsim))
for (i in 1:nsim){
  # build the matrix and vector
  A <- build_LDstable(n)
  r <- build_randomr(n)
  xstar <- get_final_composition(A, r)
  results$ncoexisting[i] <- sum(xstar > 0)
}
pl <- ggplot(data = results) +
  aes(x = ncoexisting) +
  geom_bar() +
  scale_x_continuous("number of species coexisting", breaks = 0:10)
show(pl)
```




```
# add binomial distribution
tbinom <- data.frame(ncoexisting = 0:n,
                     expectation = nsim * dbinom(0:n, n, 0.5))
pl <- pl + geom_point(data = tbinom, aes(x = ncoexisting, y = expectation))
show(pl)
```



We find again that the number of coexisting species follows the binomial distribution with parameters n and $1/2$ (i.e., exactly the same result found for non interacting species). To prove this fact, let's begin with showing that the probability of having all species coexisting is $1/2^n$. For this, we need $x^\star = -rA^{-1}$ to be feasible. For each possible A (invertible) and r, we have that the vector $-rA^{-1}$ will display a certain pattern of signs. Proving that the probability that all n species coexist is $1/2^n$ amounts to proving that all sign patterns are equally probable. To this end, define the matrix $D_k = (-1)^{\delta_{ik}}\delta_{ij}$. This is like the identity matrix, but with element k of the diagonal set to -1 instead of 1. We have that:

$$(D_k A D_k) D_k x^\star = -D_k r$$

That is, by setting the k^{th} element of D_k to -1, we flip the sign of the k^{th} component of $x^\star$. For example:

```
set.seed(2)
A <- build_LDstable(3)
r <- build_randomr(3)
# the equilibrium is not feasible
solve(A, -r)
# build matrix D_1
D_1 <- diag(c(-1, 1, 1))
# the multiplication changes the sign of a component of the solution
# making the equilibrium feasible
```

```
solve(D_1 %*% A %*% D_1, -D_1 %*% r)
```

```
# [1] -11.75292  26.47508  22.73981
#           [,1]
# [1,] 11.75292
# [2,] 26.47508
# [3,] 22.73981
```

Because of the symmetry assumption (i.e., we sampled all growth rates and off-diagonal elements from a distribution centered at zero), $(D_k A D_k)$ has the same distribution as A, and $D_k r$ the same distribution as r. In fact, one can see that $(D_k A D_k)$ is also a similarity transformation, implying that the eigenvalues of A and $(D_k A D_k)$ are the same (i.e., the operation preserves stability):

```
eigen(A)$values
eigen(D_1 %*% A %*% D_1)$values
```

```
# [1] -0.010000 -1.601076 -3.665781
# [1] -0.010000 -1.601076 -3.665781
```

We can repeat the operation several times, connecting each possible starting point to the feasible solution. Therefore, the probability of all species coexisting amounts to the probability of having chosen the "right" sequence of D_k, which happens with probability $1/2^n$. Notice that the same proof holds when the coefficients of A are sampled in pairs from a bivariate distribution (rather than having symmetric matrices), as long as the distribution is centered at zero, and the matrix is Lyapunov-Diagonally stable (Serván et al. (2018)). Using the same argument, one can prove that, under this parametrization, the probability of observing k species coexisting and $n-k$ not being able to invade is exactly $\binom{n}{k}\frac{1}{2^n}$ (Serván et al. (2018)).

3.6 Assembly and saturated equilibrium

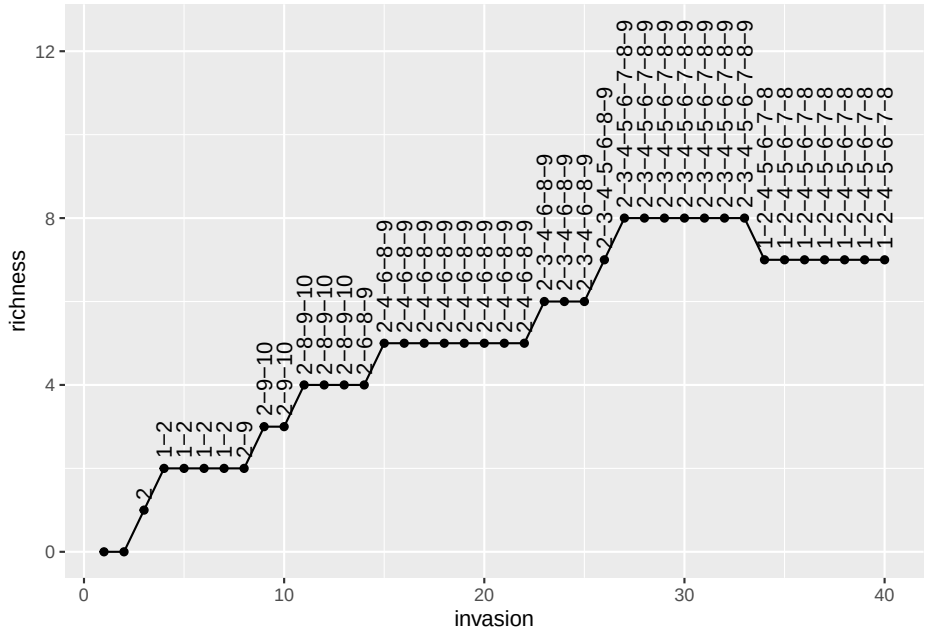
Now let's make it more complicated: we assemble an ecological community from the ground up. At each step, we introduce a species at low abundance, starting from an empty community. We then compute the new equilibrium, completing a step of the assembly:

```
assembly_one_step <- function(x, r, A){
  n <- nrow(A)
  invader <- sample(1:n, 1)
  x[invader] <- 0.001 # introduce the invader at low abundance
  present <- x > 0 # these are the species present now
  # compute new equilibrium
  y <- get_final_composition(A[present, present, drop = FALSE], r[present])
  x[present] <- y
```

```
  return(x)
}
```

Now we can take a species pool along with their parameters, and try to assemble the system until we can no longer add any more species (i.e., until we reach a saturated equilibrium):

```
set.seed(7)
n <- 10
A <- build_LDstable(n)
r <- build_randomr(n)
# start with no species
x <- rep(0, n)
# assemble for 40 steps and keep track of richness and composition
ninvasions <- 40
results <- tibble(invasion = 1:ninvasions,
                  richness = rep(NA, ninvasions),
                  composition = rep(NA, ninvasions))
for (i in 1:ninvasions){
  x <- assembly_one_step(x, r, A)
  results$richness[i] <- sum(x > 0)
  results$composition[i] <- paste((1:n)[x>0], collapse = "-")
}
print(head(results))
pl <- ggplot(results) + aes(x = invasion, y = richness, label = composition) +
  geom_point() + geom_line() + geom_text(hjust = 0, nudge_y = 0.25, angle = 90)+
  ylim(c(0, n * 1.25))
show(pl)
```

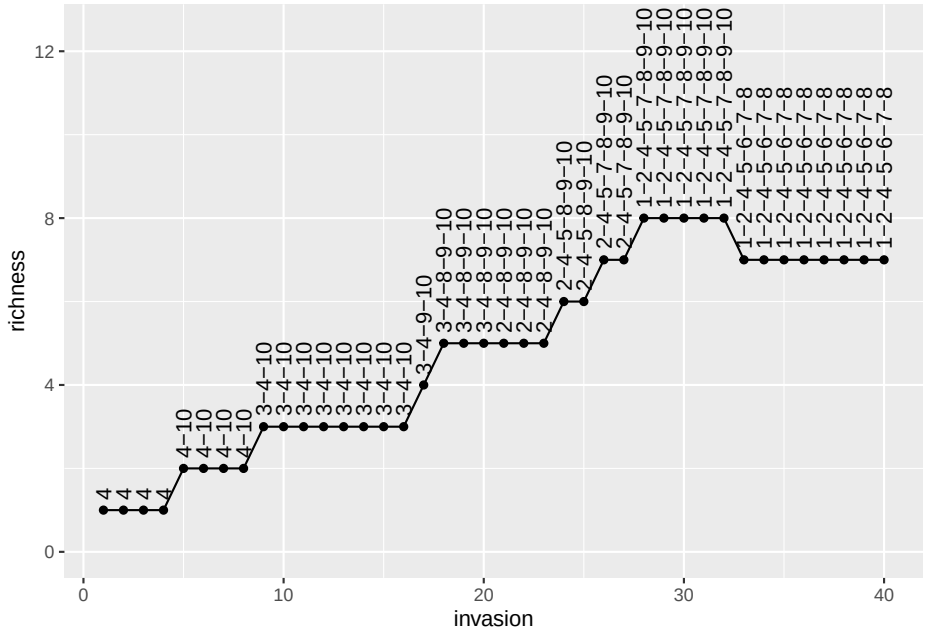



```
# # A tibble: 6 x 3
#   invasion richness composition
#      <int>    <int> <chr>
# 1        1        0 ""
# 2        2        0 ""
# 3        3        1 2
# 4        4        2 1-2
# 5        5        2 1-2
# 6        6        2 1-2
```

Now let's roll back history and assemble again:

```
# start with no species
x <- rep(0, n)
# assemble for 40 steps and keep track of richness
ninvasions <- 40
results <- tibble(invasion = 1:ninvasions,
                  richness = rep(NA, ninvasions),
                  composition = rep(NA, ninvasions))
for (i in 1:ninvasions){
  x <- assembly_one_step(x, r, A)
  results$richness[i] <- sum(x > 0)
  results$composition[i] <- paste((1:n)[x>0], collapse = "-")
}
print(head(results))
pl <- ggplot(results) + aes(x = invasion, y = richness, label = composition) +
```

```
  geom_point() + geom_line() + geom_text(hjust = 0, nudge_y = 0.25, angle = 90)+
  ylim(c(0, n * 1.25))
show(pl)
```




```
# # A tibble: 6 x 3
#   invasion richness composition
#      <int>    <int> <chr>
# 1        1        1 4
# 2        2        1 4
# 3        3        1 4
# 4        4        1 4
# 5        5        2 4-10
# 6        6        2 4-10
```

As you can see, despite taking a different assembly history, we reach the same final composition. In fact, this is exactly what we would expect if we were to throw all species in the environment at the same time:

```
get_final_composition(A, r)
x
```

```
#  [1] 1.7588344 2.5440743 0.0000000 0.7860178 1.8730535 2.5502055 0.6961259
#  [8] 1.6956878 0.0000000 0.0000000
#  [1] 1.7588344 2.5440743 0.0000000 0.7860178 1.8730535 2.5502055 0.6961259
#  [8] 1.6956878 0.0000000 0.0000000
```

As such, given enough time, any assembly history for a symmetric, stable matrix A will eventually reach the final composition represented by the saturated equilibrium (Serván, unpublished).

Interestingly, this is not the case when the matrix is not symmetric. Serván et al. (2018) conjectured however that the probability of finding a system whose final composition cannot be assembled one species at a time decreases rapidly with the size of the pool, as long as A is Lyapunov Diagonally stable.

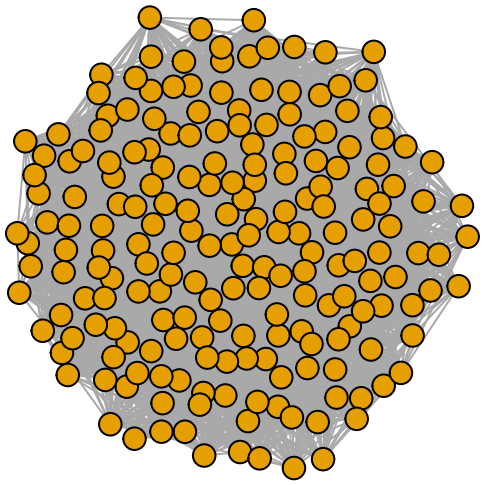
3.7 Network spandrels

What is the network structure of the assembled community vs. that of the initial pool? Can we detect a signature of the forces acting on the community such that some species can persist, while other go extinct? To answer these questions, we start by considering a larger pool of species:

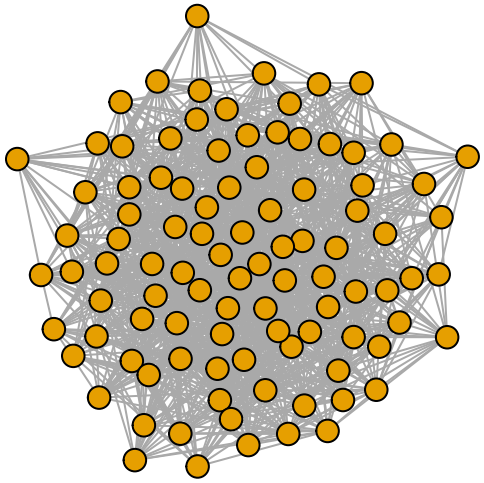
```
set.seed(1)
# initial pool
A <- build_LDstable(200)
r <- build_randomr(200)
# final composition
xstar <- get_final_composition(A, r)
A_pruned <- A[xstar > 0, xstar > 0]
r_pruned <- r[xstar > 0]
```

Are the properties of matrix A different from those of the pruned version $\tilde{A}$? We probe this in two ways: first, we build a graph with the strongest interactions, and plot it.

```
library(igraph)
plot_graph_strong <- function(B, quantile = 0.75){
  Bstrong <- abs(B)
  diag(Bstrong) <- 0
  Bstrong[Bstrong < quantile(Bstrong, quantile)] <- 0
  gr <- graph_from_adjacency_matrix((Bstrong > 0) * 1, mode = "undirected")
  plot(gr, vertex.size=10, vertex.label=NA, layout=layout_with_fr)
}

plot_graph_strong(A)
```

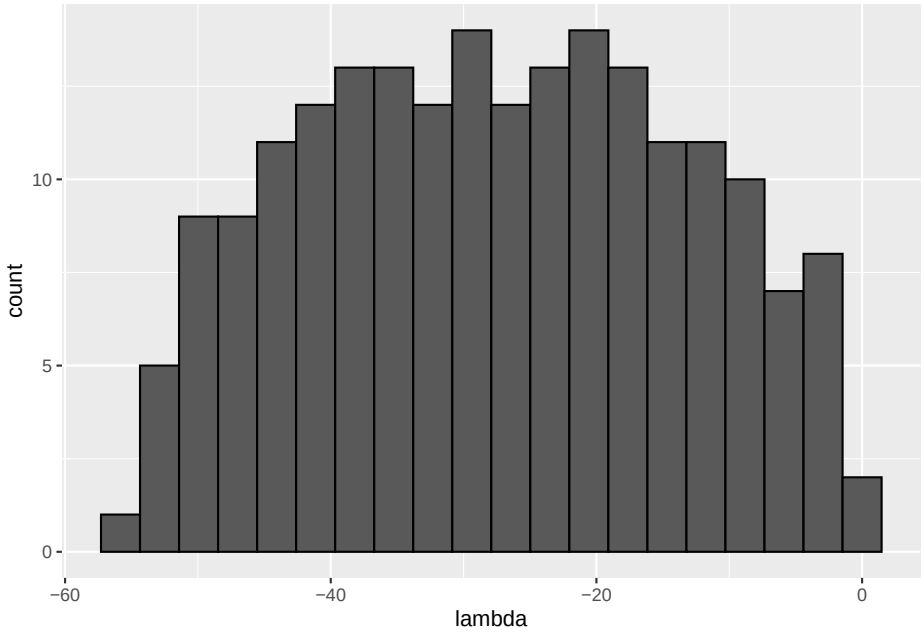


```
plot_graph_strong(A_pruned)
```

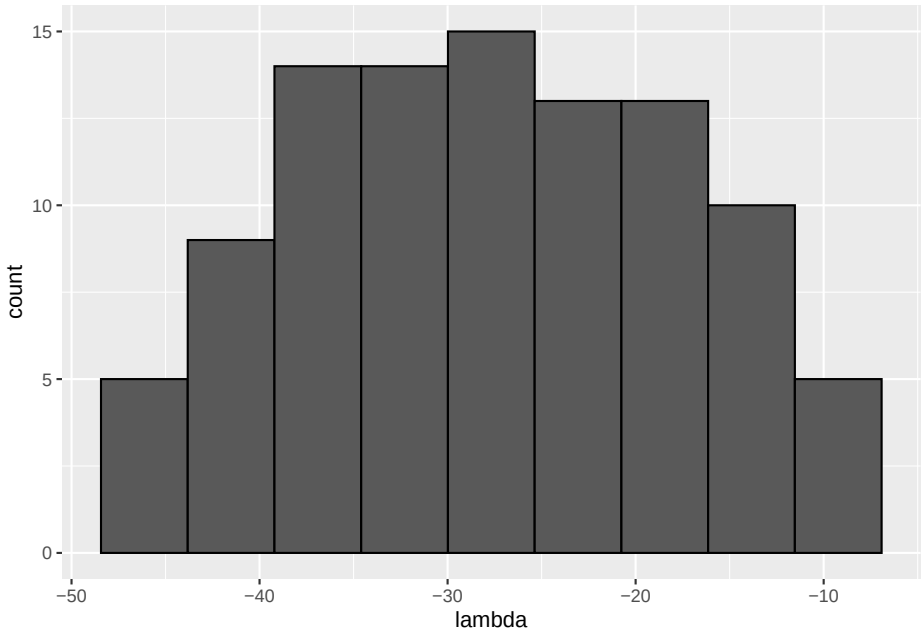


There seems to be no special structure. A more powerful way to show the same is to plot the eigenvalues of A and $\tilde{A}$. For a symmetric matrix with off-diagonal elements centered at zero, the eigenvalues should follow Wigner's semicircle law:

```
plot_eigen <- function(B){
  evals <- data.frame(lambda = eigen(B, only.values = TRUE, symmetric = TRUE)$values)
  ggplot(data = evals) +
    aes(x = lambda) +
    geom_histogram(bins = as.integer(nrow(evals) / 10), colour = "black")
}
plot_eigen(A)
```



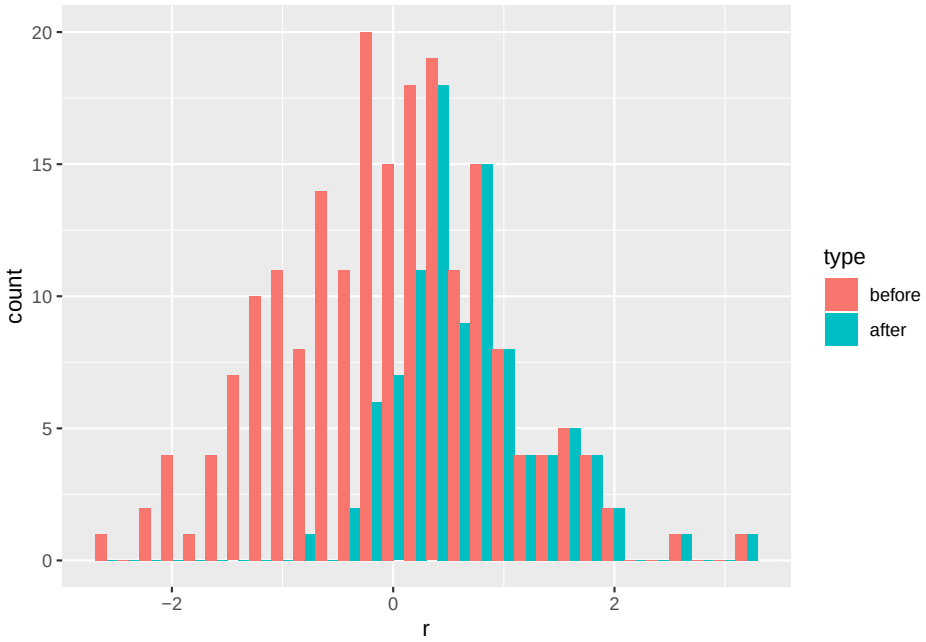

```
plot_eigen(A_pruned)
```



As such, the matrix of interactions before/after dynamics seem to have the same properties. However, as shown in Serván et al. (2018), the distribution of the

growth rates changes in a non-trivial way:

```
toplot <- data.frame(r = r, type = "before")
toplot <- rbind(toplot, data.frame(r = r_pruned, type = "after"))
ggplot(toplot) + aes(x = r, fill = type) + geom_histogram(position = "dodge")
```

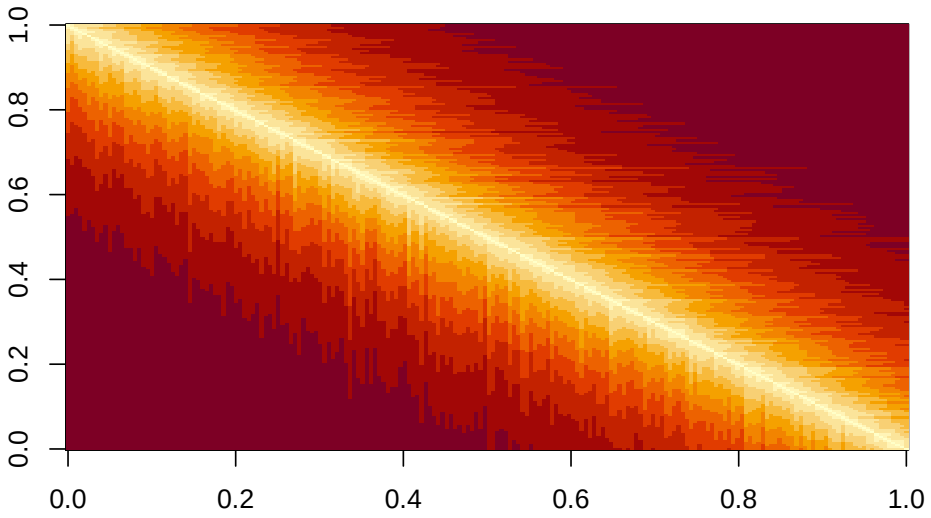


But what if species were to be related to each other? For example, suppose $r_i = 1$ for all species, and build a matrix in which the interactions of species $i + 1$ are obtained by mutating slightly those of species i:

```
n <- 200
r <- rep(1, 200)
A <- matrix(0, n, n)
# set first species
A[1, 1] <- -1
# now each species is obtained by mutating the previous one
for (i in 2:n){
  ai <- A[i -1,] * (1 - 0.05 * runif(n))
  A[i, ] <- A[i, ] + ai
  A[, i] <- A[, i] + ai
  A[i, i] <- -1
}
# make LD-stable
l1 <- max(eigen(A, only.values = TRUE, symmetric = TRUE)$values)
if (l1 > 0) diag(A) <- diag(A) - l1 * 1.01
```

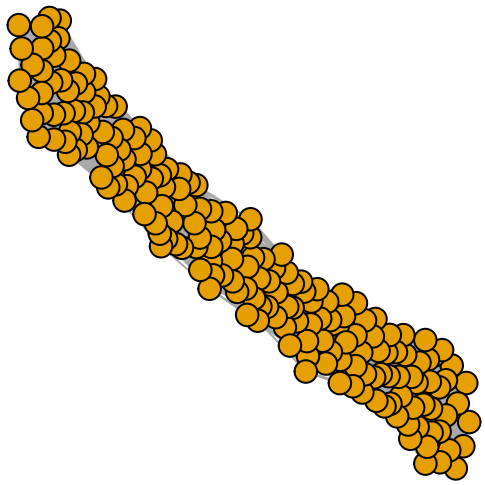
Now each species is similar to the previous one:

```
image(A[1:n, n:1])
```

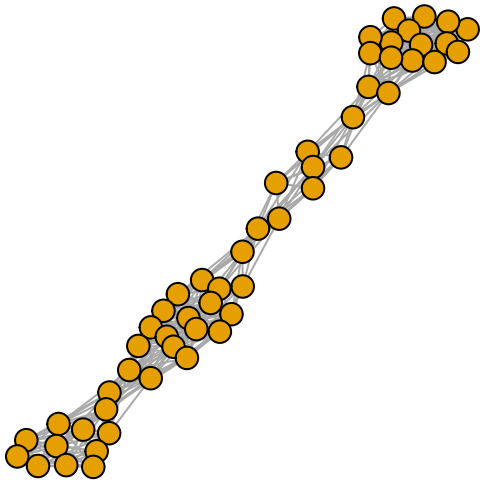


Perform the pruning, and plot networks and eigenvalues:

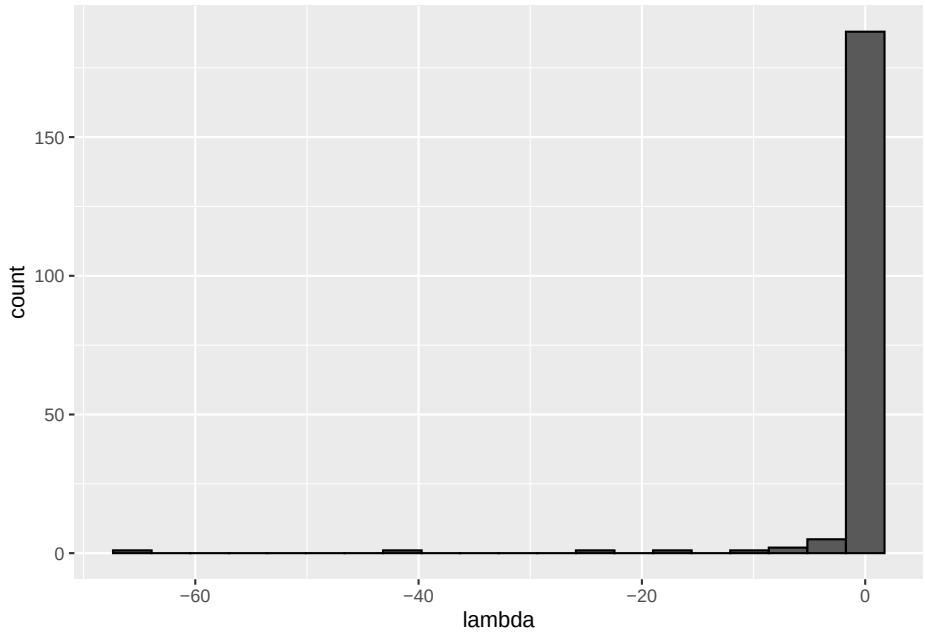
```
xstar <- get_final_composition(A, r)
A_pruned <- A[xstar > 0, xstar > 0]
r_pruned <- r[xstar > 0]

plot_graph_strong(A)
```



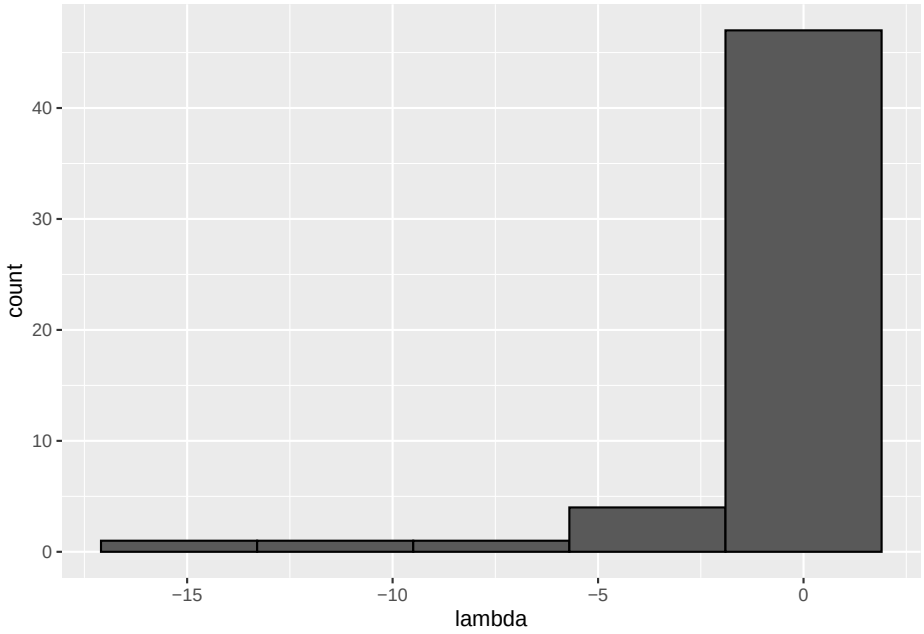
```
plot_graph_strong(A_pruned)
```



```
plot_eigen(A)
```




```
plot_eigen(A_pruned)
```



Meaning that if our matrix A is structured, we will recover a structured matrix after pruning, while if A is unstructured, we will recover an unstructured matrix. Maynard et al. (2018) showed that a well-defined network structure could be a "network spandrel" (cfr. Gould and Lewontin (1979) and Solé and Valverde (2006)) arising from the way new species are introduced, rather than a "signature of stability".

Chapter 4

Evolutionary Game Theory

4.1 Game theory

We briefly introduce the study of mathematical models of games, and their relations to the GLV model. While the origin of game theory can be traced back to the 1700s, modern game theory started with von Neumann's paper *On the Theory of Games of Strategy*, published in 1928, and with the subsequent 1944 book with Morgensten *Theory of Games and Economic Behavior*. John Nash's paper (Nash, 1950) introduced the notion of a **Nash Equilibrium** (NE). Originally, game theory was studied by economists and mathematicians, but with the 1973 book *Evolution and the Theory of Games* (Maynard Smith, 1982), John Maynard Smith showed how the same tools could be used to study evolutionary dynamics, introducing the influential concept of an **Evolutionary Stable Strategy** (ESS). Evolutionary game theory was greatly advanced through the introduction of the **Replicator Equation** (RE), which, as we will see later, has strong connection with the GLV model. For a detailaed introduction to evolutionary game theory see Hofbauer and Sigmund (1998).

4.2 Two-player matrix games

We start by anayzing games in which two players face each other, each choosing one strategy out of a set. Importantly, we consider static games in which each player makes their decision without having knowledge of the decision of the other player. Player 1 chooses a "pure" strategy s_1 from the set of pure strategies S_1, while player 2 chooses $s_2 \in S_2$. We call $\pi_k(s_1, s_2)$ the **payoff** for player k when player 1 chooses s_1 and player 2 s_2. In a matrix game, we can arrange the payoffs for each player in a matrix. We call A the matrix of payoffs for player one and B that for player two.

4.2.1 Example: the prisoner's dilemma

Two prisoners alleggedly committed a crime, but the police do not have enough evidence to convict them. They therefore approach each prisoner separately, proposing a deal: if the prisoner confesses ("defect" the other prisoner), then their sentence will be reduced. In particular: i) if both confess (both "defect"), they are sentenced to 4 years in prison; ii) if one confesses ("defect") and the other keeps quiet ("cooperate" with the other prisoner), then the one who has confessed is let free, and the other sentenced to 5 years; iii) if both keep quiet ("cooperate"), then they are both sentenced to two years of detention for some accessory crime.

In matrix form, we have (rows for pl 1 strategy C/D; cols for pl 2 strategy):

$$A = \begin{pmatrix} -2 & -5 \\ 0 & -4 \end{pmatrix} \quad B = \begin{pmatrix} -2 & -5 \\ 0 & -4 \end{pmatrix}$$

What is the best strategy player 1 can play—without knowing whether player 2 will confess or not? If player 2 were to keep quiet, player 1 should confess and be let free; if player 2 confesses, on the other hand, player 1 should also confess and get a reduced sentence. As such, each player would rationally choose to confess, thereby getting sentenced to four years in prison; note that if they could trust the other player, they could cooperate to get a much reduced sentence.

The defect/defect is called a **Nash Equilibrium**: no player can improve their payoff by changing only their own strategy.

4.3 Mixed strategies

Above, the player could choose one out of two strategies. A generalization of this situation is one in which players play **mixed** strategies, i.e., play a given strategy at random according to a set of probabilities. Call x_i the probability that player 1 plays strategy i; then $\sum_i x_i = 1$. Similarly, y_i is the probability that player 2 plays strategy i. A natural choice for the payoff of player 1 is therefore:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j \pi_1(s_1, s_2) = x^t A y$$

Similarly, we have $y^t B x$ for player 2.

A mixed strategy x is dominated by mixed strategy $\tilde{x}$ if $\tilde{x}^t A y \geq x A y$ for every y. The condition can be written as $(\tilde{x}^t - x) A y \geq 0$.

4.4 Nash Equilibria

A pair of mixed strategies $\tilde{x}$ and $\tilde{y}$ is called a Nash Equilibrium for a two person matrix game if:

$$\tilde{x}^t A\tilde{y} \geq x^t A\tilde{y} \quad \text{for all } x$$
$$\tilde{y}^t B\tilde{x} \geq y^t B\tilde{x} \quad \text{for all } y$$

Nash proved that every two-person game has at least one Nash Equilibrium.

4.5 Evolutionary stable strategies

In the context of evolution, one can consider x to be the proportion of individuals (e.g., of different species, or phenotypes) displaying different characteristics. When playing against each other, they win or lose, and their payoffs are invested in reproduction. Because the different "strategies" (populations, phenotypes) play against each other, the payoff matrix is the same for all players, and encoded in matrix A.

In this context, a strategy x is called an evolutionary stable strategy if two conditions are met:

$$x^t Ax \geq y^t Ax \quad \text{for all } y$$

meaning that x plays against itself not worse than any other strategy, and

$$\text{If } y \neq x \text{ then } y^t Ax = x^t Ax \text{ implies } x^t Ay > y^t Ay$$

meaning that if y plays as well as x against x, then x plays against y better than y against itself.

We next connect NE and ESS with dynamical systems.

4.6 Replicator dynamics

The replicator equation for this type of games can be written as:

$$\frac{dx_i}{dt} = x_i \left(\sum_j A_{ij} x_j - \sum_{jk} x_j A_{jk} x_k \right)$$

If we define $f = Ax$ as a vector reporting the "fitness"" of each population at time t, and $\bar{f} = \sum_{jk} x_j A_{jk} x_k = x^t At$ as the average fitness at time t we can write the replicator equation more compactly as:

$$\frac{dx_i}{dt} = x_i(f_i - \bar{f})$$

The RE is essentially equivalent to a GLV model in which we track frequencies instead of abundances.

Importantly, we can see x as a "mixed strategy" for the symmetric game encoded in A. In this context, **an equilibrium of the replicator equation is a Nash Equilibrium for the game**; similarly, **a stable equilibrium for the replicator equation is an Evolutionary Stable Strategy**.

4.6.1 Invariants

Adding a constant to each column of A does not alter the dynamics. We have $B = A + eb^t$, where e is a vector of ones. Then:

$$\begin{aligned}
x_i\left(\sum_j B_{ij}x_j - \sum_{kj} x_k B_{kj}x_j\right) &= x_i\left(\sum_j (A_{ij}+b_j)x_j - \sum_{kj} x_k(A_{kj}+b_j)x_j\right) \\
&= x_i\left(\sum_j A_{ij}x_j + \sum_j b_j x_j - \sum_{kj} x_k A_{kj}x_j - \sum_{kj} x_k b_j x_j\right) \\
&= x_i\left(\sum_j A_{ij}x_j + \sum_j b_j x_j - \sum_{kj} x_k A_{kj}x_j - \sum_j b_j x_j\right) \\
&= x_i\left(\sum_j A_{ij}x_j - \sum_{kj} x_k A_{kj}x_j\right)
\end{aligned}$$

Similarly, multiplying each column of A by a (possibly different) positive constant does not alter dynamics (it just rescales time). As such, if $A_2 = A_1 D + eb^t$ the replicator equations formed using A_1 and A_2 are equivalent.

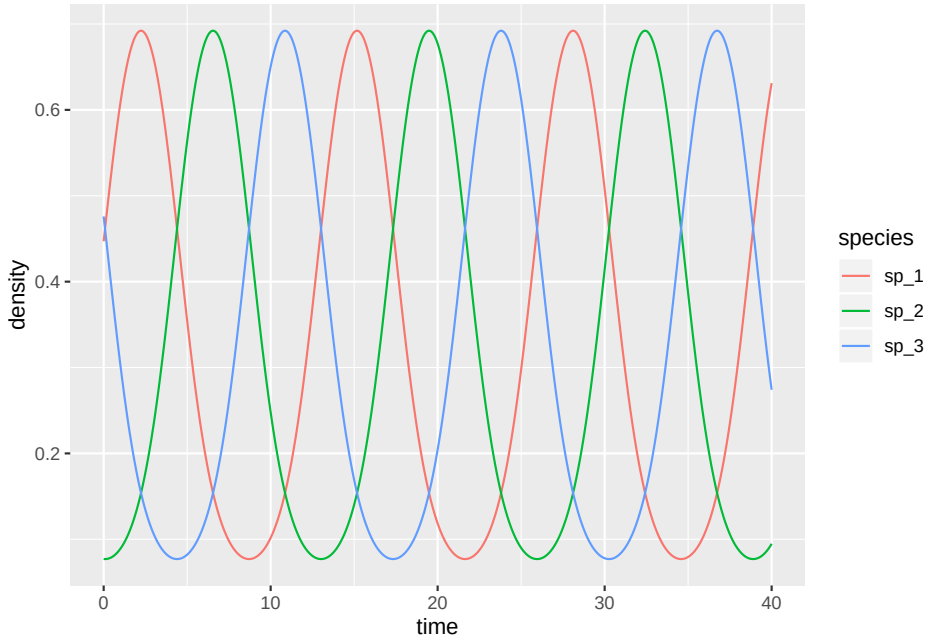
4.6.2 Rock-paper-scissor

Let's try our hand with a simple zero-sum (i.e., $A = -A^t$) replicator equation. We have three populations ("rock", "paper", and "scissors") with payoff matrix:

$$A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

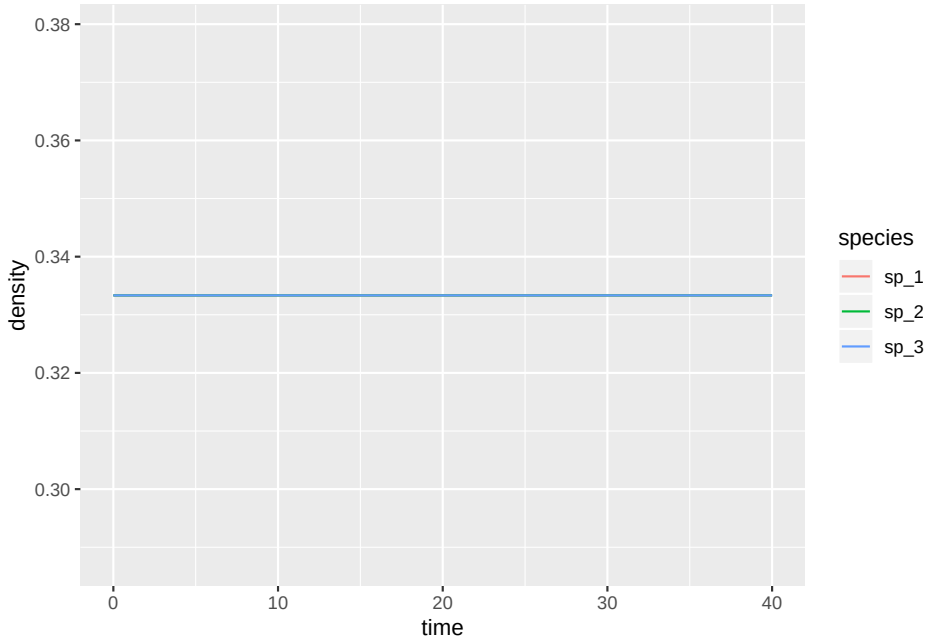
We start the population at a random initial condition, and track dynamics:

```
library(deSolve) # integrate ODEs
library(tidyverse) # plotting and wrangling
# define the differential equation
RE <-function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    x[x < 10^-8] <- 0 # prevent numerical problems
    x <- x / sum(x) # keep on simplex
    dxdt <- x * (A %*% x - sum(x * A %*% x))
    list(dxdt)
  })
}
# function to plot output
plot_ODE_output <- function(out){
  out <- as.data.frame(out)
  colnames(out) <- c("time", paste("sp", 1:(ncol(out) -1), sep = "_"))
  out <- as_tibble(out) %>% gather(species, density, -time)
  pl <- ggplot(data = out) +
    aes(x = time, y = density, colour = species) +
    geom_line()
  show(pl)
  return(out)
}
# general function to integrate RE
integrate_RE <- function(A, x0, maxtime = 40, steptime = 0.05){
  times <- seq(0, maxtime, by = steptime)
  parameters <- list(A = A)
  # solve numerically
  out <- ode(y = x0, times = times,
             func = RE, parms = parameters,
             method = "ode45")
  # plot and make into tidy form
  out <- plot_ODE_output(out)
  return(out)
}
# payoff matrix
A <- matrix(c(0, -1, 1,
              1, 0, -1,
              -1, 1, 0), 3, 3, byrow = TRUE)
# initial conditions
x0 <- runif(3)
x0 <- x0 / sum(x0)
rps <- integrate_RE(A, x0)
```



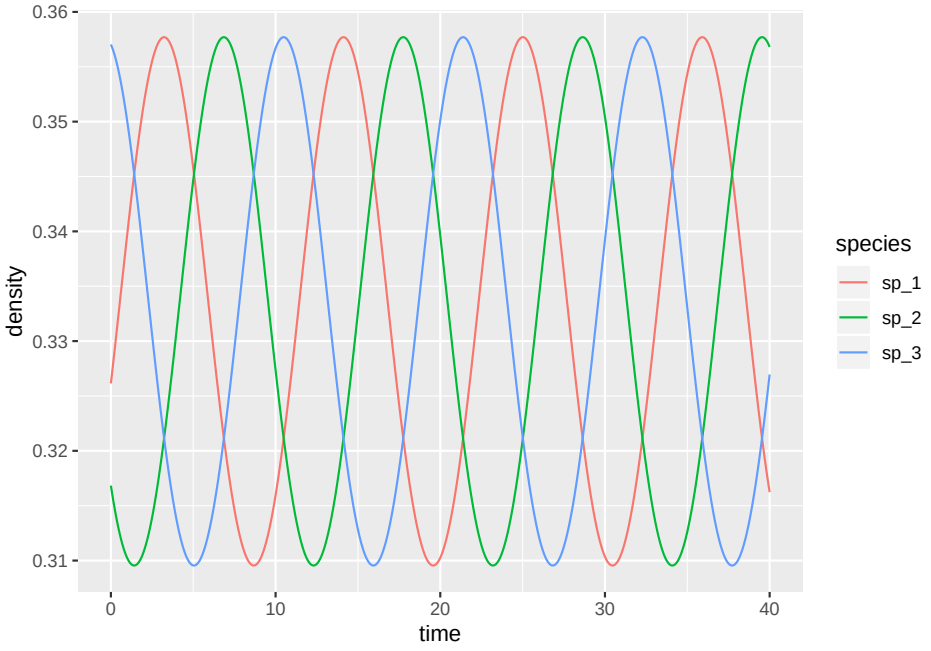
What if we start all populations at the same density?

```
x0 <- rep(1 / 3, 3)
rps <- integrate_RE(A, x0)
```



And if they are close to the 1/3?

```
x0 <- rep(1/3, 3) + 0.05 * runif(3)
x0 <- x0 / sum(x0)
rps <- integrate_RE(A, x0)
```



4.6.3 Equivalence with GLV

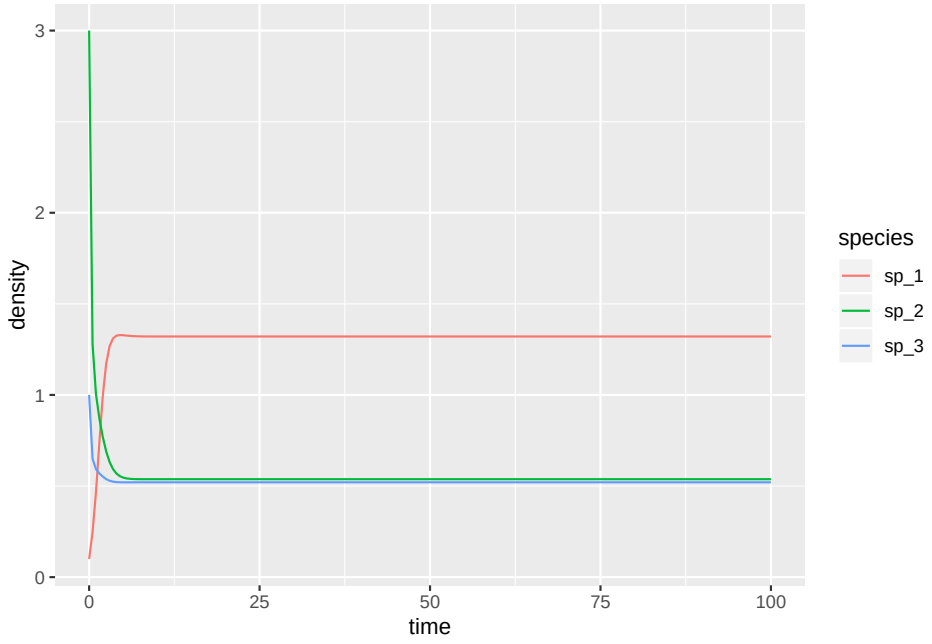
For a given $n-$species GLV system, there is an equivalent $(n+1)-$dimensional replicator equation with zeros in the last row of the matrix. To show this, let's take our functions for integrating GLV:

```
# Generalized Lotka-Volterra model
GLV <- function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    x[x < 10^-8] <- 0 # prevent numerical problems
    dxdt <- x * (r + A %*% x)
    list(dxdt)
  })
}
# general function to integrate GLV
integrate_GLV <- function(r, A, x0, maxtime = 100, steptime = 0.5){
  times <- seq(0, maxtime, by = steptime)
  parameters <- list(r = r, A = A)
  # solve numerically
```

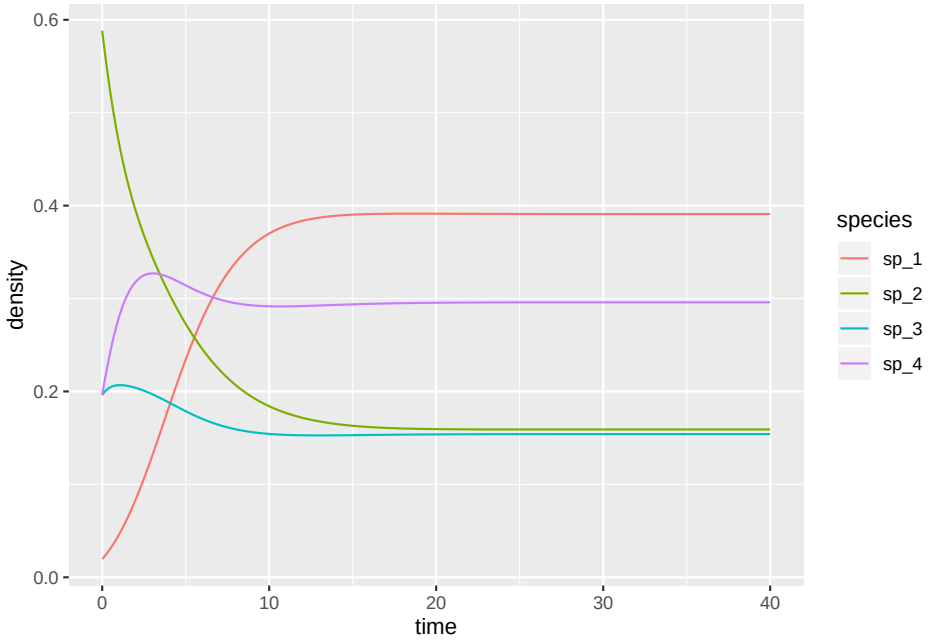
```
  out <- ode(y = x0, times = times,
             func = GLV, parms = parameters,
             method = "ode45")
  # plot and make into tidy form
  out <- plot_ODE_output(out)
  return(out)
}
```

And integrate the simple system:

```
r <- c(1, 2, 3)
A <- matrix(c(-1, 0.5, 0.1,
              -0.7, -2, 0,
              -0.3, 0, -5), 3, 3, byrow = TRUE)
x0 <- c(0.1, 3, 1)
glvex <- integrate_GLV(r, A, x0)
```




```
B <- matrix(0, 4, 4)
B[1:3, 1:3] <- A
B[1:3, 4] <- r
B
y0 <- c(x0, 1)
y0 <- y0 / sum(y0)
reex <- integrate_RE(B, y0)
```




```
#      [,1] [,2] [,3] [,4]
# [1,] -1.0  0.5  0.1    1
# [2,] -0.7 -2.0  0.0    2
# [3,] -0.3  0.0 -5.0    3
# [4,]  0.0  0.0  0.0    0
```

Now let's look at the equilibrium of the GLV model:

```
glvex %>% filter(time == 100)
```

```
# # A tibble: 3 x 3
#    time species density
#   <dbl> <chr>     <dbl>
# 1   100 sp_1      1.32
# 2   100 sp_2      0.538
# 3   100 sp_3      0.521
```

And recover it from the RE equation (just divide by the density of the extra "species"):

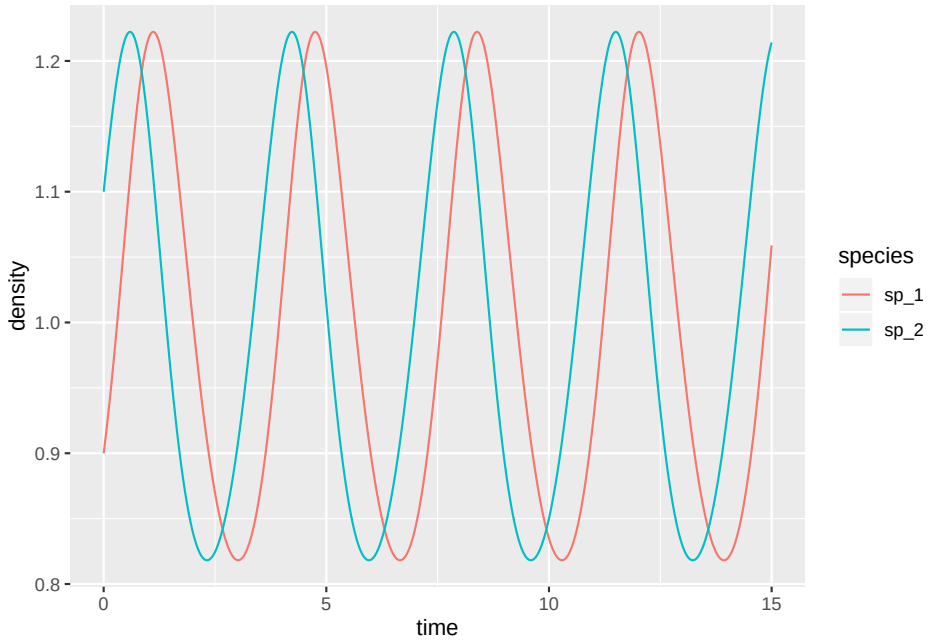
```
reex %>% filter(time == 40)
reex %>% filter(time == 40) %>% mutate(density = density / tail(density, 1))
```

```
# # A tibble: 4 x 3
#    time species density
#   <dbl> <chr>     <dbl>
# 1    40 sp_1      0.391
```

```
# 2    40 sp_2      0.159
# 3    40 sp_3      0.154
# 4    40 sp_4      0.296
# # A tibble: 4 x 3
#    time species density
#   <dbl> <chr>     <dbl>
# 1    40 sp_1      1.32
# 2    40 sp_2      0.538
# 3    40 sp_3      0.521
# 4    40 sp_4      1
```

Not only the equilibria are the same, but also the dynamics are the same once time has been properly rescaled. Similarly, for each RE system we can always recover a matrix with zero in the last row by applying the transformations detailed above, and therefore recover the corresponding GLV. For example, take the matrix for the RPS above, and make each coefficient in the last row zero by adding the appropriate constant to each column. Then one recovers some sort of a predator-prey system:

```
r <- c(-1, 1)
A <- matrix(c(-1, 2,
              -2, 1), 2, 2, byrow = TRUE)
x0 <- c(0.9, 1.1)
glvex <- integrate_GLV(r, A, x0, maxtime = 15, steptime = 0.01)
```



in which the species oscillate around one.

4.6.4 Hypertournament games

The rock-paper-scissor game above is a simple case of a "hypertournament" game. Take the zero-sum payoff matrix $A = -A^t$. Then, we have

$$\sum_i \sum_j A_{ij} x_i x_j = 0$$

and the RE simplifies to

$$\frac{dx}{dt} = D(x)Ax$$

At equilibrium, either some elements of x are zero, or

$$Ax^\star = 0$$

meaning that if a feasible equilibrium $x^\star$ exists, it is an eigenvector of A corresponding to a zero eigenvalue.

4.6.4.1 Number of coexisting species

We now show how the equations above can arise when modeling ecological dynamics. Suppose that a forest is composed of a fixed number of trees. Each time a tree dies (with rate $d = 1$ for all species), a gap in the canopy opens, and species will compete to colonize it. Let's assume that two seeds (sampled with probability proportional to the density of the species) land in the empty patch, and that they compete to fill the gap. Call H_{ij} the probability that i wins when competing with j; we have $H_{ij} + H_{ji} = 1$. We can write the dynamics (Grilli et al., 2017) as:

$$\begin{aligned}
\frac{dx}{dt} &= x_i \left(\sum_j 2H_{ij} x_j - 1 \right) \\
&= x_i \sum_j (2H_{ij} x_j - x_j) \\
&= x_i \sum_j (H_{ij} x_j + (1 - H_{ji}) x_j - x_j) \\
&= x_i \sum_j (H_{ij} - H_{ji}) x_j \\
&= x_i \sum_j A_{ij} x_j
\end{aligned}$$

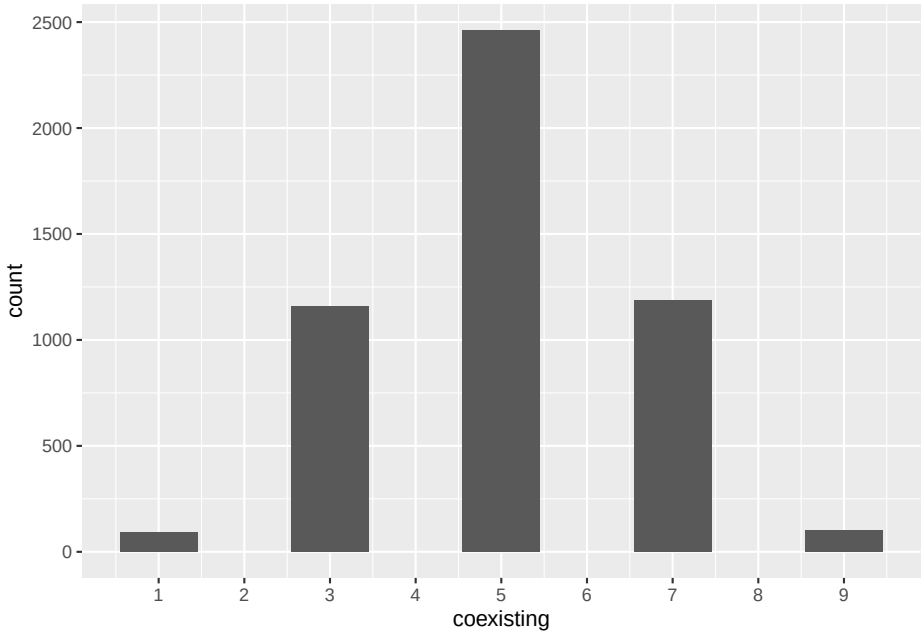
I.e., we recover the RE for a zero-sum game. What happens if we draw H (and therefore A) at random? Allesina and Levine (2011) and Grilli et al. (2017) applied the results of Fisher and Reeves (1995) and Brandl (2017) to show that, when n species compete, the probability of observing k coexisting is $p(k|n) = \binom{n}{k} 2^{1-n}$ when k is odd, and $p(k|n) = 0$ when k is even.

Importantly, to find the set of coexisting species we do not need to integrate dynamics. One can use linear programming to solve for the set of species that will coexist.

```
library(lpSolve)
# Build a random matrix H such that H_ij + H_ji = 1
random_H <- function(n){
  # build random hypertournament H
  H <- matrix(runif(n * n), n, n)
  return(H / (H + t(H)))
}
# Find the optimal strategy for the two-person game encoded in H
# using linear programming.
# This is also the coexistence equilibrium of the dynamical system.
find_optimal_strategy <- function(H){
  n <- dim(H)[1]
  f.obj <- rep(1, n)
  f.con <- H
  f.rhs <- rep(1, n)
  f.dir <- rep("<=", n)
  z <- lp ("max", f.obj, f.con, f.dir, f.rhs)
  return(z$solution / sum(z$solution))
}
```

Now let's try to count how many species suvive when starting with 10:

```
n <- 10
num_simulations <- 5000
results <- tibble(simulation = 1:num_simulations, coexisting = NA)
for (i in 1:num_simulations){
  H <- random_H(n)
  coexisting <- find_optimal_strategy(H)
  results[i,"coexisting"] <- sum(coexisting > 0)
}
# and plot
ggplot(data = results) +
  aes(x = coexisting) +
  geom_bar() +
  scale_x_continuous(breaks = 0:10)
```



4.6.5 Lyapunov function

In the rock-paper-scissor example above, species cycled neutrally around the unique equilibrium point. To show that this is in fact the behavior of this type of RE, we write a Lyapunov function. By finding a constant of motion we can show that the species will follow closed orbits.

Suppose $x_i^\star > 0$ is the equilibrium for the system. We write:

$$V(x) = -\sum_i x_i^\star \log \frac{x_i}{x_i^\star}.$$

Because of Gibbs' inequality, $V(x) \geq 0$ for any x, and is equal to zero only if $x = x^\star$. Note also that at equilibrium $2\sum_j H_{ij} x_j^\star = 1$. We write:

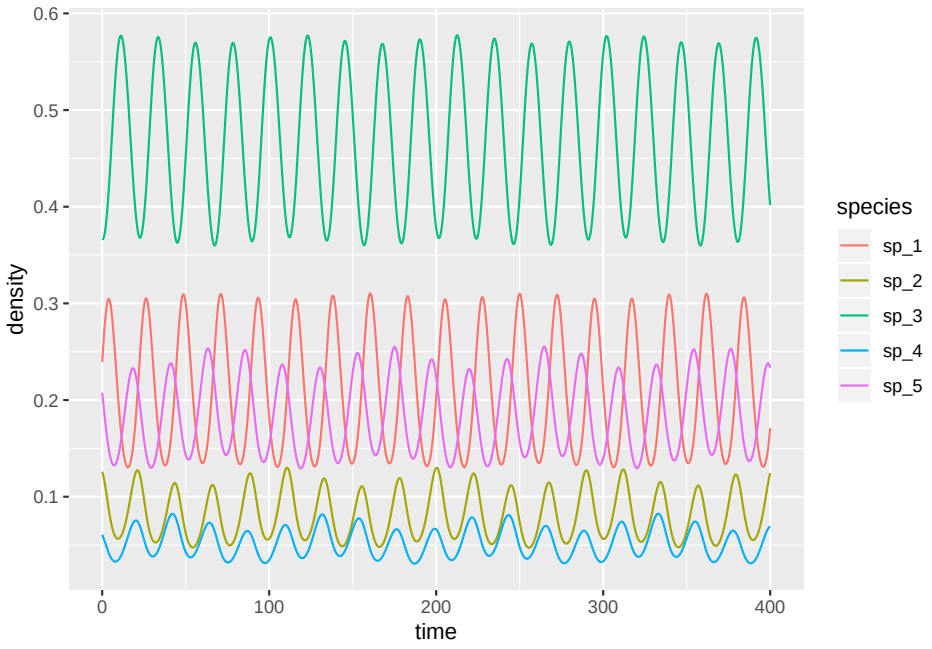
$$
\begin{aligned}
\frac{dV}{dt} &= \sum_i \frac{\partial V}{\partial x_i}\frac{dx_i}{dt} \\
&= -\sum_i \frac{x_i^\star}{x_i}\frac{dx_i}{dt} \\
&= -2\sum_{i,j} x_i^\star H_{ij} x_j + \sum_i x_i^\star \\
&= -2\sum_{i,j} x_i^\star H_{ij} x_j + 1 \\
&= \sum_j \left(-2\sum_i H_{ij} x_i^\star\right) x_j + 1 \\
&= \sum_j \left(-2\sum_i (1 - H_{ji}) x_i^\star\right) x_j + 1 \\
&= \sum_j \left(-2\sum_i x_i^\star + 2\sum_i H_{ji} x_i^\star\right) x_j + 1 \\
&= \sum_j (-2+1)\, x_j + 1 \\
&= -\sum_j x_j + 1 \\
&= 0
\end{aligned}
$$

We have found a constant of motion, meaning that the system will follow closed orbits. Hence, unless we start the system precisely at $x^\star$, the abundances will cycle neutrally around the equilibrium.

Let's try with a larger system:

```
n <- 5
# search for random H yielding all species coexisting
i <- 0
while(TRUE){
  i <- i + 1
  set.seed(i)
  H <- random_H(n)
  x_star <- find_optimal_strategy(H)
  if (all(x_star) > 0) break
}
# payoff matrix
A <- H - t(H)
# initial conditions close to equilibrium
x0 <- x_star + runif(n) * 0.2
```

```
x0 <- x0 / sum(x0)
fivespp <- integrate_RE(A, x0, maxtime = 400, steptime = 0.1)
```



4.6.6 Higher-order interactions

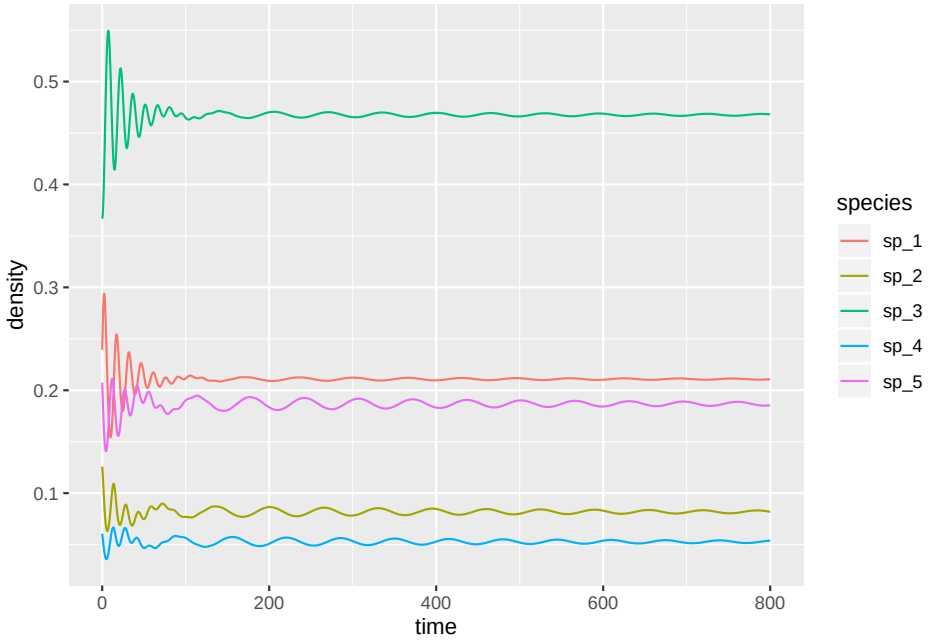
We can extend the game above to the case in which three (or more) seeds compete to fill each patch. Grilli et al. (2017) showed that in this case, one can write the replicator equation:

$$\frac{dx_i}{dt} = x_i \sum_{j,k} A_{ijk} x_j x_k$$

where the tensor A (a three-dimensional matrix) encodes the effect of a pair of species (j and k) on the density of i. Importantly, one can choose the tensor such that the equilibrium is the same as for the two-player replicator equation: take $A_{ijk} = 2H_{ij}H_{ik} - H_{ji}H_{jk} - H_{ki}H_{kj}$, which can be derived from first principles by writing the stochastic dynamics. What is surprising is that, while the equilibrium is unchanged, the dynamics are now globally stable:

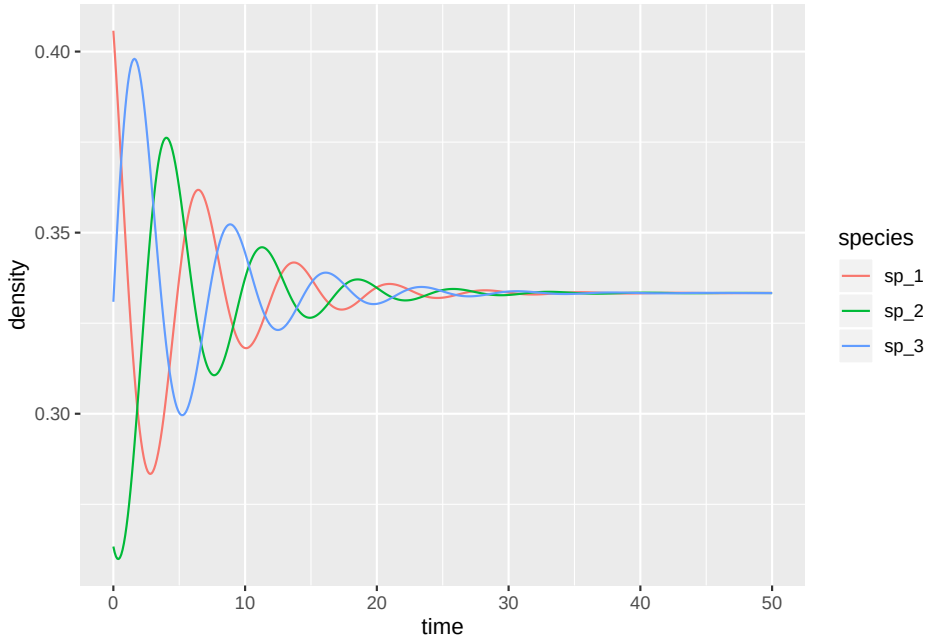
```
# Now payoff is a tensor
RE_3 <- function(t, x, parameters){
  with(as.list(c(x, parameters)), {
    x[x < 10^-8] <- 0 # prevent numerical problems
    x <- x / sum(x) # keep on simplex
```

```
    n <- length(x)
    dxidt <- rep(0, n)
    for (i in 1:n){
      dxidt[i] <- x[i] * x %*% P3[i,,] %*% x
    }
    list(dxidt)
  })
}
# general function to integrate RE_3
integrate_RE_3 <- function(H, x0, maxtime = 40, steptime = 0.05){
  times <- seq(0, maxtime, by = steptime)
  n <- nrow(H)
  P3 <- array(0, c(n, n, n))
  for (i in 1:n){
    for (j in 1:n){
      for (k in 1:n){
        P3[i,j,k] <- 2 * H[i,j] * H[i,k] -  H[j,i] * H[j,k] - H[k,i] * H[k,j]
      }
    }
  }
  parameters <- list(P3 = P3)
  # solve numerically
  out <- ode(y = x0, times = times,
           func = RE_3, parms = parameters,
           method = "ode45")
  # plot and make into tidy form
  out <- plot_ODE_output(out)
  return(out)
}
# integrate the system above
fivespp <- integrate_RE_3(H, x0, maxtime = 800, steptime = 0.1)
```



And the rock-paper-scissors:

```
H <- matrix(c(1/2, 1, 0,
              0, 1/2, 1,
              1, 0, 1/2), 3, 3, byrow = TRUE)
x0 <- runif(3)
x0 <- x0 / sum(x0)
rps <- integrate_RE_3(H, x0, maxtime = 50, steptime = 0.1)
```


0.40
0.35
0.30
density
0
10
20
30
40
50
time
species
sp_1
sp_2
sp_3

Chapter 5

Simple models for synthetic communities

Ecologists have performed large experiments in which different assemblages of species are co-cultured. These experiments have been conducted with plants (for example, Biodiversity Ecosystem Functioning experiments e.g., (Hector et al., 1999) (Tilman et al., 2001) (Cadotte, 2013)) and in the laboratory using protozoan, algae or bacteria. Two commonly-encountered problems in this type of experiments have to to with the scaling of the number of experiments with the number of species, and with the probability of coexistence.

Scale: How many communities can we form from a pool of n species? We can culture a single species in isolation (n possibilities), two species in pair ($n(n-1)/2$ possibilities), and so on. The total is therefore:

$$\sum_{j=1}^{n} \binom{n}{j} = 2^n - 1$$

And this is only considering the presence/absence of each species! Moreover, we might want to vary the initial conditions (e.g., starting two species at low/high abundance, equal abundance, high/low abundace), etc. Clearly, this makes trying all possible combinations unfeasible when n is large enough. For example, for 10 species we can form 1023 assemblages, while with 20 more than a million!

Coexistence: even if we could try all possible experiments, many assemblages would collapse to smaller communities because of extinctions. For example, pairs could become monocultures, triplets become pairs or monocultures, etc. As such, even if we were to try all possible combinations, we would end up observing a smaller set of "final communities".

To solve these two issues, we would need to find a good way to navigate the enormous space of possibilities, thereby suggesting "good" experiments that yield a large probability of coexistence.

5.1 Synthetic communities using the GLV model

To set the stage for deriving a statistical model that can deal with the problems above, we start by simulating the communities that can be formed from a pool of n species using the GLV model. First, we need a way to index our communities. We take a vector p reporting the presence/absence of a given species in a community. To this end, we associate a label $k \in \{1, \dots, 2^n - 1\}$ with each possible community. If only taxon 1 is present ($p = [1, 0, 0, \dots, 0]$), then $k = 1$, if only taxon 2 is present ($p = [0, 1, 0, \dots, 0]$) then $k = 2$, if both taxa 1 and 2 are part of the community, $k = 3$ ($p = [1, 1, 0, \dots, 0]$); in general, $k = \sum_{i=1}^{n} p_i 2^{(i-1)}$ where $p_i = 1$ if taxon i is in the community, and $p_i = 0$ otherwise.

We can then cycle through each of the $2^n - 1$ assemblages that can be formed, and determine whether the species will coexist or not in our experiment. For simplicity, we report as coexisting any set of species for which a) the GLV equilibrium is feasible, and b) it is locally stable. We collect all the communities that are coexisting along with the abundance of all species in the matrix E.

In R:

```
set.seed(3)
# take a random matrix of interactions:
# -Aij <- U[0,1]; -Aii <- Sqrt(2) + U[0,1]
# all interactions are competitive
n <- 5
A <- -matrix(runif(n * n), n, n)
diag(A) <- diag(A) - sqrt(2)
print(A)
# choose positive growth rates
r <- runif(n) + 0.5
print(r)
```

```
#            [,1]       [,2]       [,3]       [,4]        [,5]
# [1,] -1.5822551 -0.6043941 -0.5120159 -0.8297087 -0.22820188
# [2,] -0.8075164 -1.5388470 -0.5050239 -0.1114492 -0.01532989
# [3,] -0.3849424 -0.2946009 -1.9482489 -0.7036884 -0.12898156
# [4,] -0.3277343 -0.5776099 -0.5572494 -2.3117018 -0.09338193
# [5,] -0.6021007 -0.6309793 -0.8679195 -0.2797326 -1.65109857
# [1] 1.291147 1.099732 1.410148 1.060425 1.255705
```

Now go through all possible combinations:

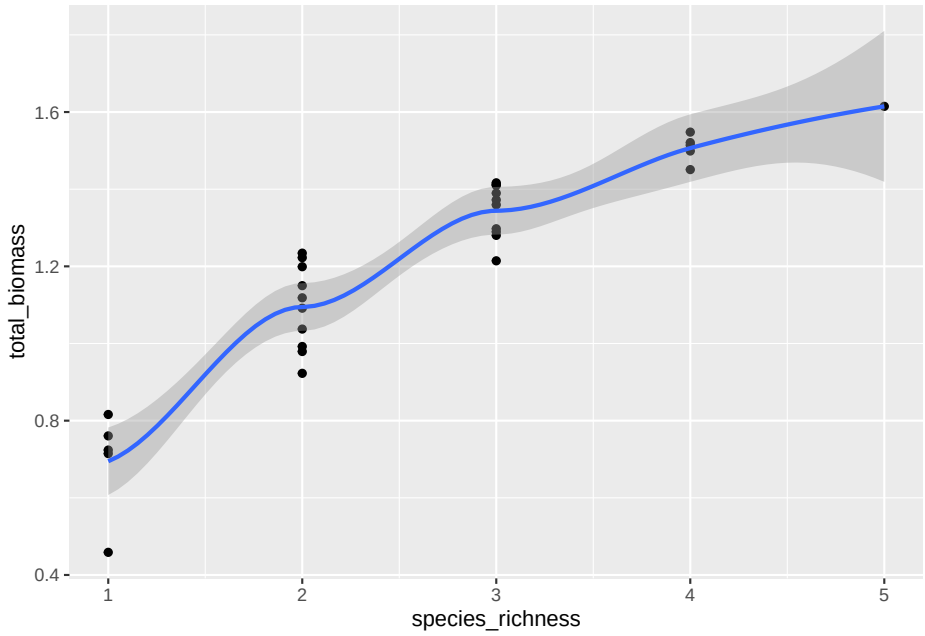
```
E <- matrix(0, 0, n) # matrix containing all communities
x_template <- rep(0, n) # template for row of E
for (k in 1:(2^n - 1)){
  p <- as.integer(intToBits(k)[1:n])
  presence <- p > 0
  A_k <- A[presence, presence, drop = FALSE]
  r_k <- r[presence]
  # check if equilibrium is feasible
  x_k_star <- solve(A_k, -r_k)
  if (all(x_k_star > 0)){
    # check if equilibrium is locally stable
    # a) build community matrix
    M <- x_k_star * A_k
    # b) compute eigenvalues
    eM <- eigen(M, only.values = TRUE)$values
    # c) check stability
    if (all(Re(eM) < 0)){
      # we have a feasible, stable equilibrium
      # add to the set
      tmp <- x_template
      tmp[presence] <- x_k_star
      E <- rbind(E, tmp)
    }
  }
}
rownames(E) <- NULL
head(E) # show the first few communities
tail(E) # show the last few communities
```

```
#           [,1]      [,2]      [,3] [,4] [,5]
# [1,] 0.8160172 0.0000000 0.0000000    0    0
# [2,] 0.0000000 0.7146465 0.0000000    0    0
# [3,] 0.6791728 0.3582477 0.0000000    0    0
# [4,] 0.0000000 0.0000000 0.7238026    0    0
# [5,] 0.6215352 0.0000000 0.6009974    0    0
# [6,] 0.0000000 0.5020196 0.6478907    0    0
#            [,1]      [,2]      [,3]      [,4]      [,5]
# [26,] 0.0000000 0.6907495 0.0000000 0.2679023 0.4511633
# [27,] 0.4476631 0.4567147 0.0000000 0.2658808 0.3776961
# [28,] 0.0000000 0.0000000 0.5887242 0.3006418 0.4001216
# [29,] 0.4749396 0.0000000 0.5198836 0.2551261 0.2708252
# [30,] 0.0000000 0.5139182 0.5636548 0.1848845 0.2365139
# [31,] 0.4000232 0.3200119 0.5151320 0.1902248 0.1893434
```

5.2 Total biomass

Our simulated communties have much in common with BEF experiments. For example, plotting the species richness on the x-axis and the total biomass on the y-axis, we recover the typical pattern found in real experiments ((Hector et al., 1999) (Tilman et al., 2001) (Cadotte, 2013)):

```
library(tidyverse) # plotting and wrangling
total_biomass <- rowSums(E)
species_richness <- rowSums(E > 0)
BEF <- tibble(species_richness = species_richness, total_biomass = total_biomass)
ggplot(data = BEF) +
  aes(x = species_richness, y = total_biomass) +
  geom_point() + geom_smooth()
```



showing a modest increase in total biomass with increasing species richness.

5.3 Hyperplanes

We can rewrite the GLV model as:

$$\begin{aligned}\frac{dx(t)}{dt} &= D(x(t))(r + Ax(t)) \\ &= D(r \circ x(t))(1 + Bx(t))\end{aligned}$$

where we have $b_{ij} = a_{ij}/r_i$. The equilibrium for each subset of species k (if it exists) can be found as $(B^{(k)})^{-1}1^{(k)}$, where $(B^{(k)})^{-1}$ is the inverse of the $B^{(k)}$ sub-matrix of B in which only the rows/columns belonging to k are retained, and $1^{(k)}$ is a vector of ones with length equal to the number of species in k.

We now want to link the matrix B with the results of the experiments, stored in matrix E. Call $x_i^{(k)}$ the recorded density of taxon i in community k (i.e., stored in one of the rows of E). Then, we can build a matrix P where each row represents an equation $\sum_{j \in k} b_{ij} x_j^{(k)} = 1$ and we have a column for each coefficient in B. For example, take three taxa, and suppose that we can culture each in isolation, and that the all assemblages coexist when co-cultured. We can write 12 equations: monocultures ($k \in \{1, 2, 4\}$) give rise to a single equation; co-cultures of two taxa to two equations each ($k \in \{3, 5, 6\}$); and communities with three taxa to three independent equations ($k = 7$). Here are the 12 equations we can write:

$$\begin{aligned}
b_{11}x_1^{(1)} &= 1 \\
b_{22}x_2^{(2)} &= 1 \\
b_{11}x_1^{(3)} + b_{12}x_2^{(3)} &= 1 \\
b_{21}x_1^{(3)} + b_{22}x_2^{(3)} &= 1 \\
b_{33}x_3^{(4)} &= 1 \\
b_{11}x_1^{(5)} + b_{13}x_3^{(5)} &= 1 \\
b_{31}x_1^{(5)} + b_{33}x_3^{(5)} &= 1 \\
b_{22}x_2^{(6)} + b_{23}x_3^{(6)} &= 1 \\
b_{32}x_2^{(6)} + b_{33}x_3^{(6)} &= 1 \\
b_{11}x_1^{(7)} + b_{12}x_2^{(7)} + b_{13}x_3^{(7)} &= 1 \\
b_{21}x_1^{(7)} + b_{22}x_2^{(7)} + b_{23}x_3^{(7)} &= 1 \\
b_{31}x_1^{(7)} + b_{32}x_2^{(7)} + b_{33}x_3^{(7)} &= 1
\end{aligned}$$

We can summarize the equations in a more compact form as $Pv(B) = 1$:

$$\begin{pmatrix} x_1^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2^{(2)} & 0 & 0 & 0 & 0 \\ x_1^{(3)} & x_2^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1^{(3)} & x_2^{(3)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3^{(4)} \\ x_1^{(5)} & 0 & x_3^{(5)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^{(5)} & 0 & x_3^{(5)} \\ 0 & 0 & 0 & 0 & x_2^{(6)} & x_3^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_2^{(6)} & x_3^{(6)} \\ x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^{(7)} & x_2^{(7)} & x_3^{(7)} \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \\ b_{33} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

where $v(B)$ is a vectorized version of B. Because we have $n2^{n-1} = 12$ equations, and $n^2 = 9$ variables, in principle the system can be solved. One can recognize the equation above as a linear regression, and choose $\hat{v}(B) = P^+1$ (where P^+ is the Moore-Penrose pseudo-inverse of P). Before doing that, however, we can try to simplify the calculation by rearranging the rows of P:

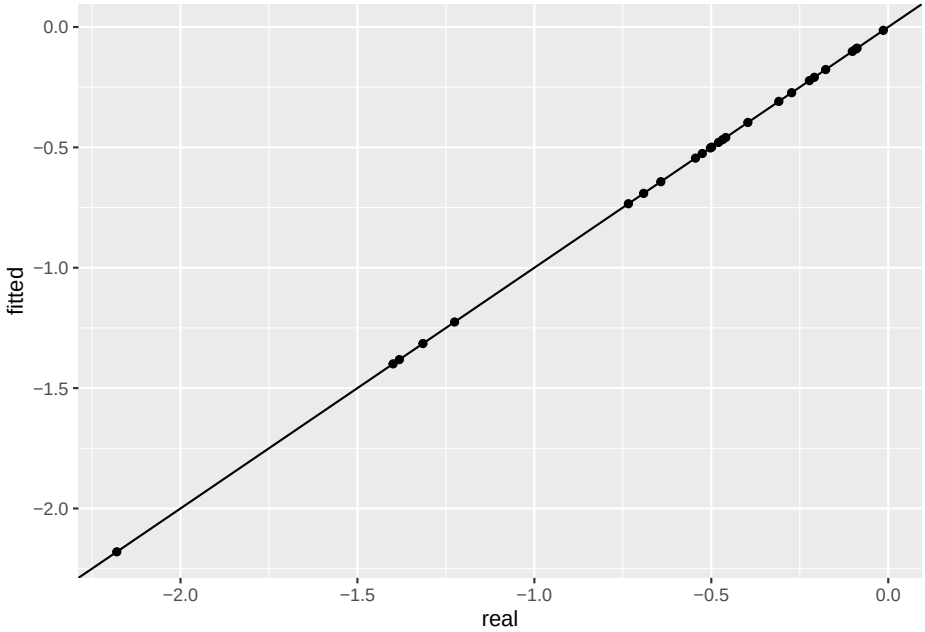
$$\begin{pmatrix} x_1^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1^{(3)} & x_2^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1^{(5)} & 0 & x_3^{(5)} & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1^{(3)} & x_2^{(3)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2^{(6)} & x_3^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3^{(4)} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^{(5)} & 0 & x_3^{(5)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_2^{(6)} & x_3^{(6)} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^{(7)} & x_2^{(7)} & x_3^{(7)} \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \\ b_{33} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Showing that the matrix P is block-diagonal, with blocks $P_i = E_i$, where E_i is the matrix E in which only the rows in which species i is present are retained. As such, we can solve for the matrix B one row at a time:

$$\begin{pmatrix} x_1^{(1)} & 0 & 0 \\ x_1^{(3)} & x_2^{(3)} & 0 \\ x_1^{(5)} & 0 & x_3^{(5)} \\ x_1^{(7)} & x_2^{(7)} & x_3^{(7)} \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

For example:

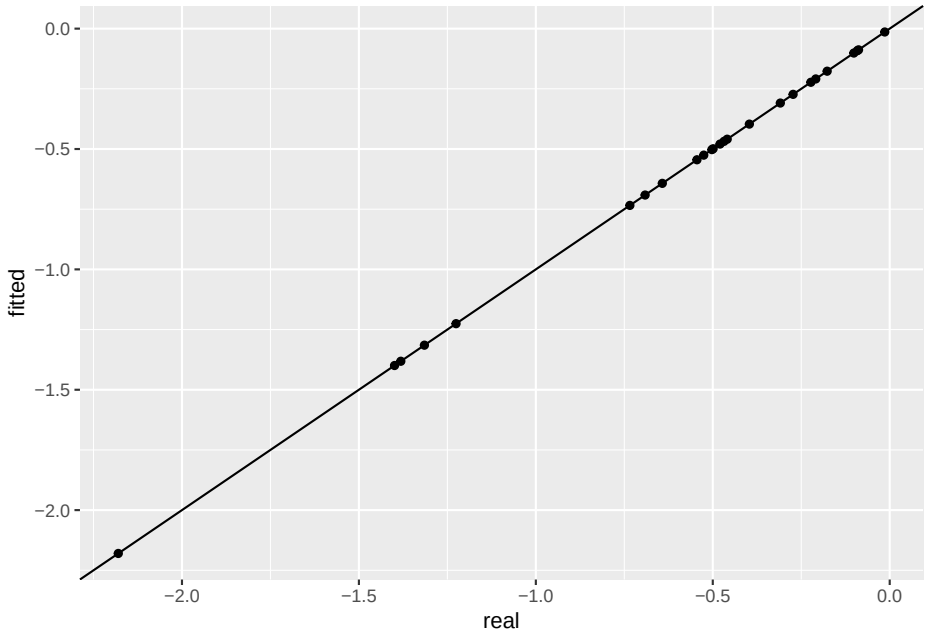
```
# rescaled matrix
B <- 1 / r * A
# now try to recover B from matrix E
fitted_B <- matrix(0, n, n)
for (i in 1:n){
  # take the matrix Ei
  Ei <- E[E[,i] > 0, , drop = FALSE]
  # solve for the row (MASS::ginv computes the pseudoinverse)
  fitted_B[i,] <- -rowSums(MASS::ginv(Ei))
}
ggplot(data = tibble(real = as.vector(B),
                     fitted = as.vector(fitted_B))) +
  aes(x = real, y = fitted) + geom_point() +
  geom_abline(slope = 1, intercept = 0)
```



Notice that in general the system of equations is over-determined. As such, we can fit using a subset of the experiments, and then predict the rest of the ex-

periments out-of-fit. For example, let's fit again using the information on monocultures and pairs only, and then predict experiment 31, in which all species should coexist:

```
E_partial <- E[rowSums(E> 0) < 3, ]
fitted_B <- matrix(0, n, n)
for (i in 1:n){
  # take the matrix Ei
  Ei <- E_partial[E_partial[,i] > 0, , drop = FALSE]
  # solve for the row
  fitted_B[i,] <- -rowSums(MASS::ginv(Ei))
}
ggplot(data = tibble(real = as.vector(B),
                     fitted = as.vector(fitted_B))) +
  aes(x = real, y = fitted) + geom_point() +
  geom_abline(slope = 1, intercept = 0)
```




```
print("predicted")
solve(fitted_B, - rep(1, n))
print("observed")
E[(2^n -1), ]
```

```
# [1] "predicted"
# [1] 0.4000232 0.3200119 0.5151320 0.1902248 0.1893434
# [1] "observed"
```

```
# [1] 0.4000232 0.3200119 0.5151320 0.1902248 0.1893434
```

The equation $E_i B_i = -1$ shows that all the equilibria of species i in a GLV system are arranged on a hyperplane.

5.4 Fitting real data

Having played with synthetic data, we can start thinking about data stemming from real experiments. In particular, consider the linear, additive model:

$$x_i^{(k)} = \gamma_i - \sum_{j \neq i} \theta_{ij} x_j^{(k)}$$

where $x_i^{(k)}$ is the density of species i in community k, γ_i models the density of i when grown in isolation, and the parameters θ_{ij} model how the density of i is influenced by the other species in community k. Dividing both sides by γ_i and rearranging:

$$\sum_{j \neq i} (\theta_{ij}/\gamma_i) x_j^{(k)} + (1/\gamma_i) x_i^{(k)} = \sum_j b_{ij} x_j^{(k)} = 1$$

which is exactly the same system of equations found above. As such, fitting a linear, addittive model for the biomass of each species in each plot is equivalent to finding the equilibrium strcuture of a GLV model that approximates the data.

One thing to consider when modeling real data, contrary to simulated data, is the structure of the errors. For example, empirical data often contain replicates that reach slightly different densities. Moreover, what is typically done in these types of experiments is to sort and measure biomass for a section of each plot, and then to multiply to extrapolate the biomass for the whole plot. As such, if the error is normally distributed for the sub-plot, then it's going to be lognormally distributed for the whole plot. Furthermore, we need to link the rows of matrix P above, because when we make an error in measuring $x_j^{(k)}$, the value is reported over several values. Following Maynard et al. (2020), we do the following:

- Propose a matrix B
- Compute the predicted abundance for all species in all observed assemblages (call them $\tilde{x}_j^{(k)}$)
- Compute the SSQ: $\sum_{j,k} (\log x_j^{(k)} - \log \tilde{x}_j^{(k)})^2$
- Search numerically for the matrix B minimizing the SSQ

```
# constant to penalize solutions with negative
# abundances
penalization <- 100000
```

```
# main fitting function
# input:
# - matrix E as above (the data)
# - proposed matrix B
# output:
# - predicted matrix \tilde{E}
# - SSQ
compute_SSQ <- function(B, E){
  tildeE <- E
  ones <- rep(1, ncol(B)) # vector of 1s
  for (i in 1:nrow(E)){
    presence <- E[i, ] > 0
    # use internal calls to speed up calculation
    predicted_x <- .Internal(La_solve(B[presence, presence, drop = FALSE],
                                      ones[presence],
                                      10^-6))
    tildeE[i, presence] <- predicted_x
  }
  # now compute SSQ
  observed <- E[E > 0]
  # penalize negatives
  predicted <- tildeE[E > 0]
  predicted[predicted < 0] <- penalization
  return(list(B = B,
              E = E,
              tildeE = tildeE,
              SSQ = sum((log(predicted) - log(observed))^2)
              ))
}
# for numerical minimization
to_minimize <- function(b, E){
  n <- ncol(E)
  return(compute_SSQ(
    matrix(b, n, n),
    E
  )$SSQ)
}
```

To test this method, we are going to use the data from Kuebbing et al. (2015), who co-cultured four species of plant in 14 out of 15 possible combinations. Let's load the data

```
url <- "https://github.com/dsmaynard/endpoints/raw/master/data/Kuebbing_plants/natives
E <- as.matrix(read_csv(url))
head(E)
tail(E)
```

```
#          as fa la po
# [1,] 1.4711  0  0  0
# [2,] 1.7968  0  0  0
# [3,] 1.8993  0  0  0
# [4,] 2.0634  0  0  0
# [5,] 2.2665  0  0  0
# [6,] 2.3062  0  0  0
#            as     fa     la     po
# [135,] 1.1219 1.8011 0.2390 1.4065
# [136,] 1.4751 1.3486 0.2474 1.5491
# [137,] 0.6021 1.9389 0.1817 2.7487
# [138,] 0.8973 1.6002 0.1156 3.2293
# [139,] 3.0384 1.4442 0.4590 1.5490
# [140,] 2.9240 2.3903 0.2379 4.3230
```

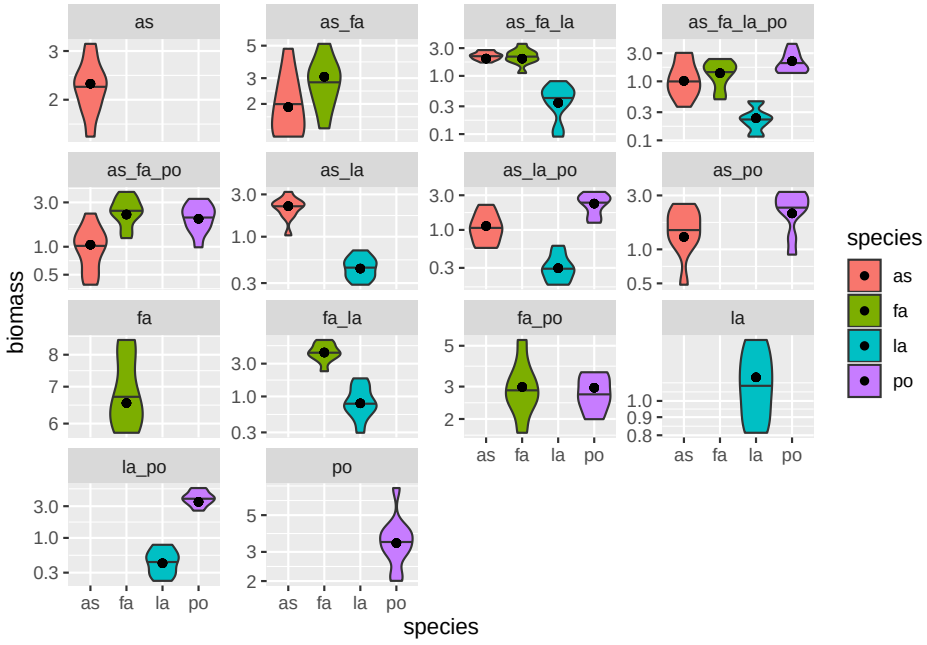
And let's try to find a good numerical solution starting from the identity matrix:

```
set.seed(2)
B <- matrix(0, 4, 4)
diag(B) <- 1
tmp <- optim(par = as.vector(B),
             fn = to_minimize,
             E = E,
             method = "BFGS")
best_B <- matrix(tmp$par, 4, 4)
solution <- compute_SSQ(best_B, E)
```

Now let's write code to show a nice plot of the predicted vs. observed values for all communities:

```
plot_solution <- function(solution, oof = NULL){
  # add a column with community composition
  spnames <- colnames(solution$E)
  community <- apply(solution$E, 1, function(x) paste(spnames[x > 0], collapse = "_"))
  # work with observed data
  E <- as_tibble(solution$E) %>%
    add_column(community = community) %>%
    mutate(experiment = row_number())
  toplot <- E %>% gather(species, observed_biomass, -experiment, -community) %>%
    filter(observed_biomass > 0)
  # now the predicted data
  if (is.null(oof)){
    tildeE <- as_tibble(solution$tildeE) %>%
      add_column(community = community) %>%
      gather(species, predicted_biomass, -community) %>%
      distinct()
  } else {
```

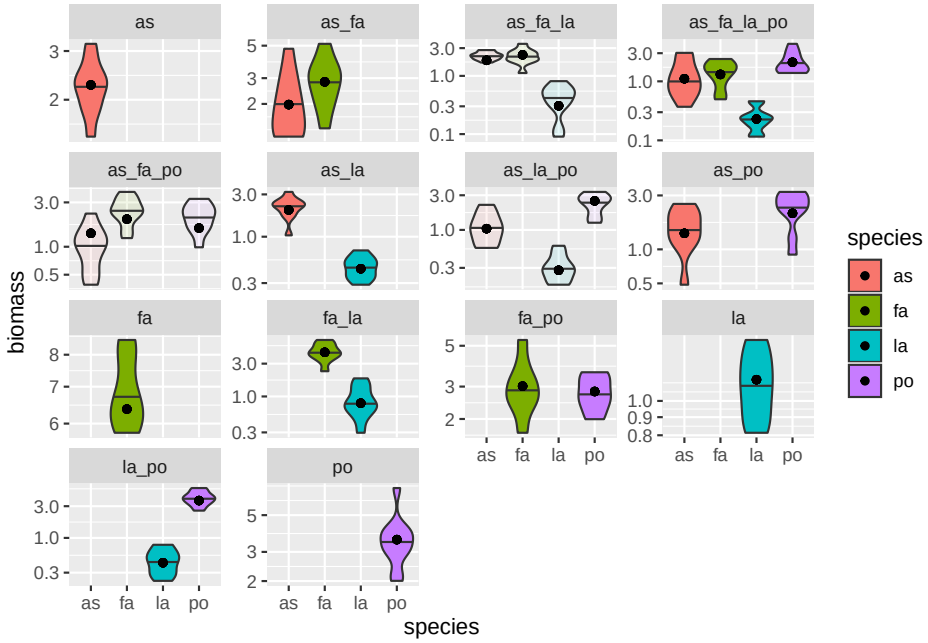
```
    tildeE <- as_tibble(solution$tildeE) %>%
      add_column(community = community, oof = oof) %>%
      gather(species, predicted_biomass, -community, -oof) %>%
      distinct()
  }
  # join the two data sets
  toplot <- toplot %>%
    inner_join(tildeE, by = c("community", "species")) %>%
    filter(observed_biomass > 0)
  if (is.null(oof)){
    pl <- ggplot(data = toplot) +
    aes(x = species, fill = species) +
    geom_violin(aes(y = observed_biomass), draw_quantiles = 0.5, scale = "width") +
    geom_point(aes(y = predicted_biomass)) +
    facet_wrap(~community, scales = "free_y") +
    scale_y_log10("biomass")
  } else {
    pl <- ggplot(data = toplot) +
    aes(x = species, fill = species) +
    geom_violin(aes(y = observed_biomass, alpha = I(1 - 0.9 * oof)),
                draw_quantiles = 0.5, scale = "width") +
    geom_point(aes(y = predicted_biomass)) +
    facet_wrap(~community, scales = "free_y") +
    scale_y_log10("biomass")
  }
  return(pl)
}
plot_solution(solution)
```



5.5 Predicting real data out of fit

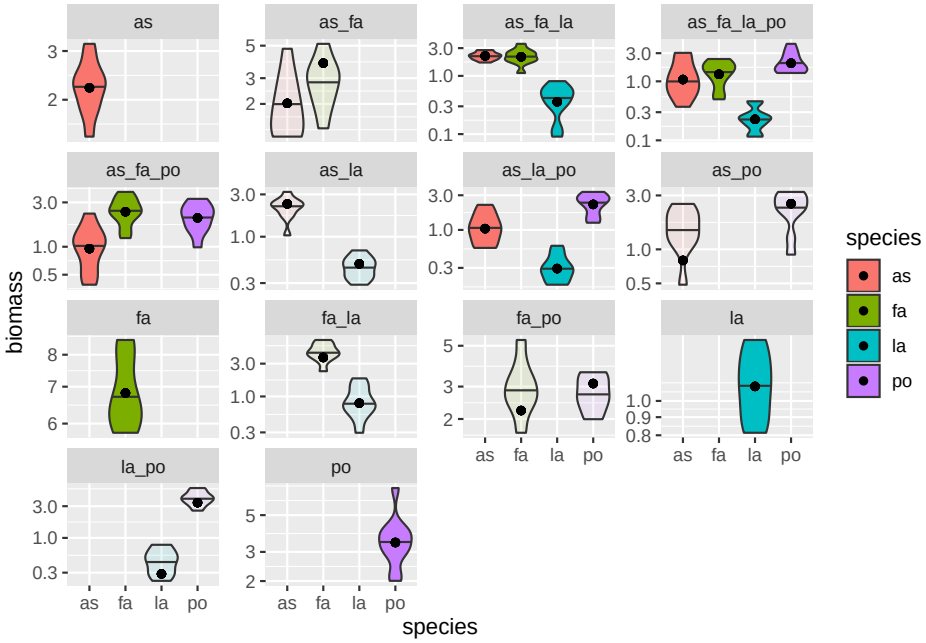
Exactly as done for the GLV simulated data, we can attempt fitting the matrix B using only part of the experimental data, and then predict the rest out-of-fit. This is a powerful test to make sure that a) we have chosen a good structure for the statistical model, and b) we are not overfitting. For example, let's try to predict all triplets out of fit. First, we fit the model using only the information about monocultures, pairs, and the quadruplet (11 experiments); then, we use the fitted matrix to predict all of the data and plot:

```
set.seed(1)
B <- matrix(0, 4, 4)
diag(B) <- 1
Enotriplets <- E[rowSums(E > 0) != 3, ]
oof <- rowSums(E > 0) == 3
tmp <- optim(par = as.vector(B),
             fn = to_minimize,
             E = Enotriplets,
             method = "BFGS")
best_B_oof <- matrix(tmp$par, 4, 4)
solution_oof <- compute_SSQ(best_B_oof, E)
plot_solution(solution_oof, oof)
```



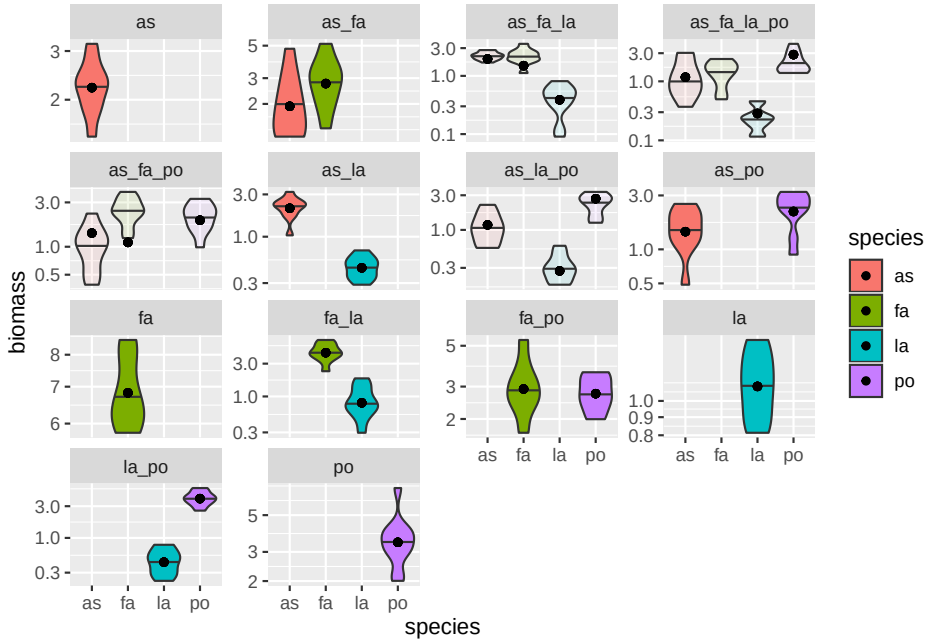
Repeat excluding all the pairs:

```
set.seed(1)
B <- matrix(0, 4, 4)
diag(B) <- 1
Enopairs <- E[rowSums(E > 0) != 2, ]
oof <- rowSums(E > 0) == 2
tmp <- optim(par = as.vector(B),
             fn = to_minimize,
             E = Enopairs,
             method = "BFGS")
best_B_oof2 <- matrix(tmp$par, 4, 4)
solution_oof2 <- compute_SSQ(best_B_oof2, E)
plot_solution(solution_oof2, oof)
```



Of course, we might push this too far. For example, a popular way to attempt parameterizing models is to consider only monocultures and pairs:

```
set.seed(1)
B <- matrix(0, 4, 4)
diag(B) <- 1
Eonlymonopairs <- E[rowSums(E > 0) < 3, ]
oof <- rowSums(E > 0) >= 3
tmp <- optim(par = as.vector(B),
             fn = to_minimize,
             E = Eonlymonopairs,
             method = "BFGS")
best_B_oof3 <- matrix(tmp$par, 4, 4)
solution_oof3 <- compute_SSQ(best_B_oof3, E)
plot_solution(solution_oof3, oof)
```



You can see that we predict a lack of coexistence for the system with all four species, while experimentally we have observed coexistence in all ten replicates. Maynard et al. (2020) have demonstrated that experimental designs mixing high- and low-abundance experiments provide the best fit while minimizing the number of experiments.

Bibliography

Allesina, S. and Levine, J. M. (2011). A competitive network theory of species diversity. *Proceedings of the National Academy of Sciences*, 108(14):5638–5642.

Allesina, S. and Tang, S. (2012). Stability criteria for complex ecosystems. *Nature*, 483(7388):205.

Allesina, S. and Tang, S. (2015). The stability–complexity relationship at age 40: a random matrix perspective. *Population Ecology*, 57(1):63–75.

Baigent, S. A. (2016). Lotka-volterra dynamics: an introduction. World Scientific.

Barabás, G., J. Michalska-Smith, M., and Allesina, S. (2016). The effect of intra- and interspecific competition on coexistence in multispecies communities. *The American Naturalist*, 188(1):E1–E12.

Brandl, F. (2017). The distribution of optimal strategies in symmetric zero-sum games. *Games and Economic Behavior*, 104:674–680.

Cadotte, M. W. (2013). Experimental evidence that evolutionarily diverse assemblages result in higher productivity. *Proceedings of the National Academy of Sciences*, 110(22):8996–9000.

Fisher, D. C. and Reeves, R. B. (1995). Optimal strategies for random tournament games. *Linear Algebra and its Applications*, 217:83–85.

Gould, S. J. and Lewontin, R. C. (1979). The spandrels of san marco and the panglossian paradigm: a critique of the adaptationist programme. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 205(1161):581–598.

Grilli, J., Barabás, G., Michalska-Smith, M. J., and Allesina, S. (2017). Higher-order interactions stabilize dynamics in competitive network models. *Nature*, 548(7666):210.

Hadeler, K. P., Mackey, M. C., and Stevens, A. (2017). *Topics in mathematical biology*. Springer.

Hector, A., Schmid, B., Beierkuhnlein, C., Caldeira, M., Diemer, M., Dimitrakopoulos, P., Finn, J., Freitas, H., Giller, P., Good, J., et al. (1999). Plant diversity and productivity experiments in european grasslands. *science*, 286(5442):1123–1127.

Hirsch, M. W. (1982). Systems of differential equations which are competitive or cooperative: I. limit sets. *SIAM Journal on Mathematical Analysis*, 13(2):167–179.

Hofbauer, J. and Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge university press.

Kuebbing, S. E., Classen, A. T., Sanders, N. J., and Simberloff, D. (2015). Above-and below-ground effects of plant diversity depend on species origin: an experimental test with multiple invaders. *New Phytologist*, 208(3):727–735.

Levins, R. (1969). Some demographic and genetic consequences of environmental heterogeneity for biological control. *American Entomologist*, 15(3):237–240.

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, 6(7):410–415.

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of the Washington academy of sciences*, 16(12):317–323.

Lotka, A. J. (2002). Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry*, 14(3):271–274.

May, R. M. (1972). Will a large complex system be stable? *Nature*, 238(5364):413–414.

Maynard, D. S., Miller, Z. R., and Allesina, S. (2020). Predicting coexistence in experimental ecological communities. *Nature Ecology & Evolution*, 4:91–100.

Maynard, D. S., Serván, C. A., and Allesina, S. (2018). Network spandrels reflect ecological assembly. *Ecology letters*, 21(3):324–334.

Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge university press.

Nash, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49.

Serván, C. A., Capitán, J. A., Grilli, J., Morrison, K. E., and Allesina, S. (2018). Coexistence of many species in random ecosystems. *Nature ecology & evolution*, 2(8):1237.

Smale, S. (1976). On the differential equations of species in competition. *Journal of Mathematical Biology*, 3(1):5–7.

Solé, R. V. and Valverde, S. (2006). Are network motifs the spandrels of cellular complexity? *Trends in ecology & evolution*, 21(8):419–422.

Strogatz, S. H. (2018). *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press.

Tilman, D., Reich, P. B., Knops, J., Wedin, D., Mielke, T., and Lehman, C. (2001). Diversity and productivity in a long-term grassland experiment. *Science*, 294(5543):843–845.

Volterra, V. (1926a). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560.

Volterra, V. (1926b). Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Memor. Accad. Lincei.*, 6:31–113.