Stability and Extinction

By Marcellinus Jerricho

During the seminar, you explored the world of Foxes and Rabbits. In this assignment, we take a look at other population models.

Too Many Rabbits

Because of the increased demand for Fox hides and a lack of action by WWF, the Foxes were hunted to extinction. Without the predators, the Rabbit growth rate is given by

$$\dot{R}(t) = rR(t)\,,$$

allowing the population to grow without bound. In reality, however, there is a point when the numbers become so large that Rabbits start cometing with each other for resources. Once this happens, the increased number of Rabbits will actually slow down the population growth.

Recall that during the seminar the deleterious effect of Foxes on the Rabbit population was captured by a term of the form $F(t)R(t)$. In this case, it is the Rabbits that hinder the growth, so we will introduce a term $R(t) \times R(t)$. From this, we have

$$\dot{R}(t) = rR(t) - \frac{r}{N}\left[R(t)\right]^2 = rR(t)\left[1 - \frac{R(t)}{N}\right]\,.$$

You might be wondering why we wrote the coeffecient of $[R(t)]^2$ as r/N. To understand that, take a look inside the last brackets. What is the growth rate if $R(t) = N$? Is it positive, negative, or zero? What happens if $R(t) > N$? Can you describe in your words the meaning of N? Don't worry if you cannot quite see it as we will explore this shortly.

Exercise 1

Write a function that takes R, N, and r and returns $\dot{R}$.

```
r_dot <- function(R, N, r) {
  r * R * (1 - (R / N))
}
```

Next, write a function that takes the initial Rabbit population, r, N, and a list of times to return the Rabbit population for these times (refer to **Exercise 2** from the seminar).

```
r_pop_time <- function(R0, N, r, times) {
  # Get the number of entries in the times list
  num_steps = length(times)
  # Initialize list to hold R(t)
  Rs <- matrix(0, 1, num_steps)
  # Set the initial values
  Rs[1] <- R0
  # Write the portion of the function that fills Rs with
  # the correct values:
```

```
  for (i in 2:num_steps) {
    delta <- times[i] - times[i - 1]
    Rs[i] <- Rs[i - 1] + (delta * r_dot(Rs[i - 1], N, r))
  }

  return(Rs)
}
```

Exercise 2

Using the same r and $R(0)$ as during the seminar, plot the Rabbit population for 48 months. Play around with the value of N by setting it above and below $R(0)$.

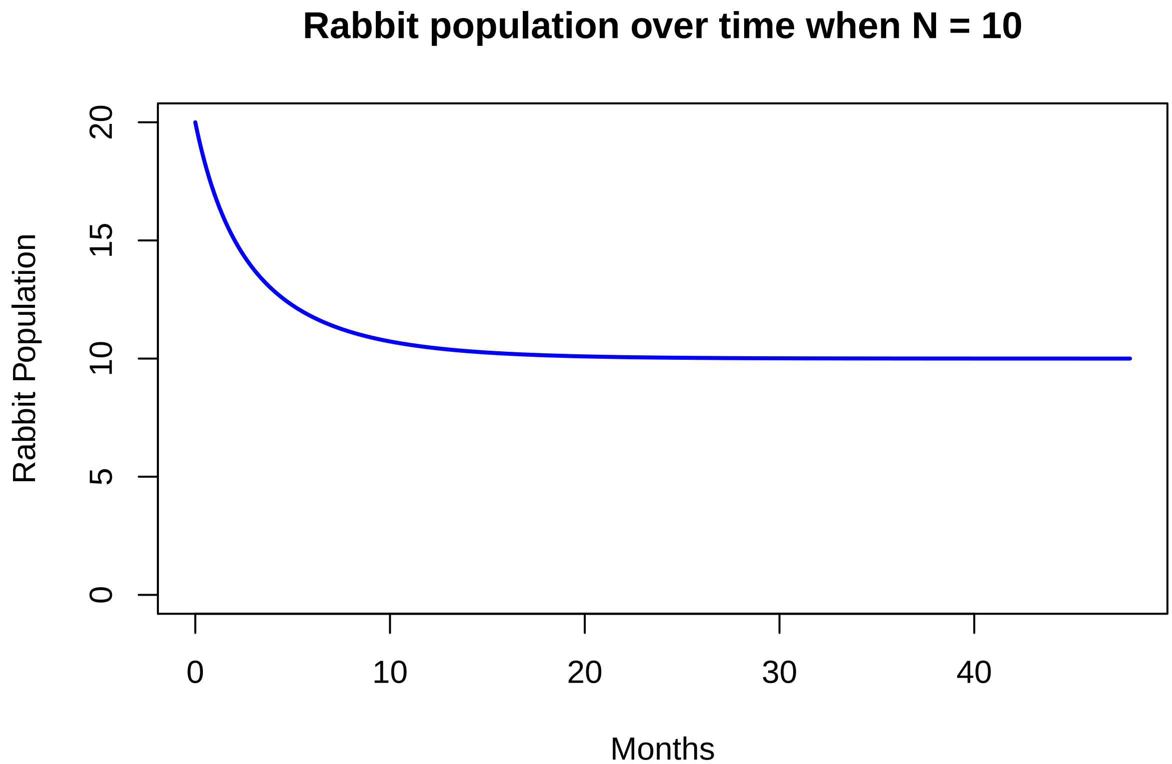
From the graphs, you can identify N as the *carrying capacity*–the maximum population size that the environment can support.

Important! Graph the plots of **Exercise 2** using different numbers of time steps. You will notice that as this number gets smaller, you population starts oscillating around N instead of *monotonically* approaching it. This is an artefact (defect)! It arises because we are using a discrete approximation (time steps) for a continuous-time problem.

By looking at the expression for $\dot{R}(t)$, you can see that crossing the carrying capacity ($R(t) = N$) sets $\dot{R}(t)$ to zero, which means that the rate of change vanishes. Make sure you understand this point!

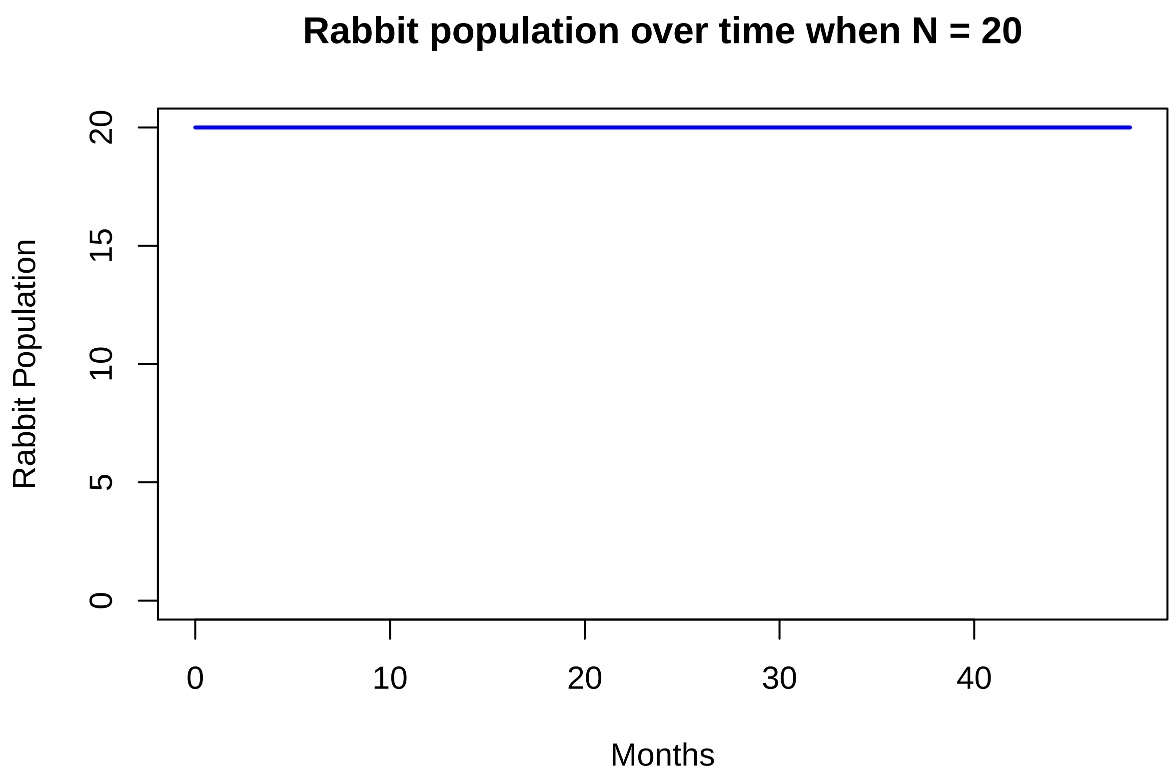
This oscillatory behavior is an example of spurious effects that one needs to be aware of when modeling systems. Always check whether your numerical results makes sense physically.

```
t_max <- 48
num_steps <- 10000
times <- seq(0, t_max, length.out = num_steps)
r <- 0.2
R0 <- 20

N <- 10

r_pop <- r_pop_time(R0, N, r, times)

plot(times, r_pop,
     type = "l", col = "blue" ,
     xlab = "Months", ylab = "Rabbit Population",
     lwd = 2, ylim = c(0, max(r_pop)),
     main = "Rabbit population over time when N = 10")
```



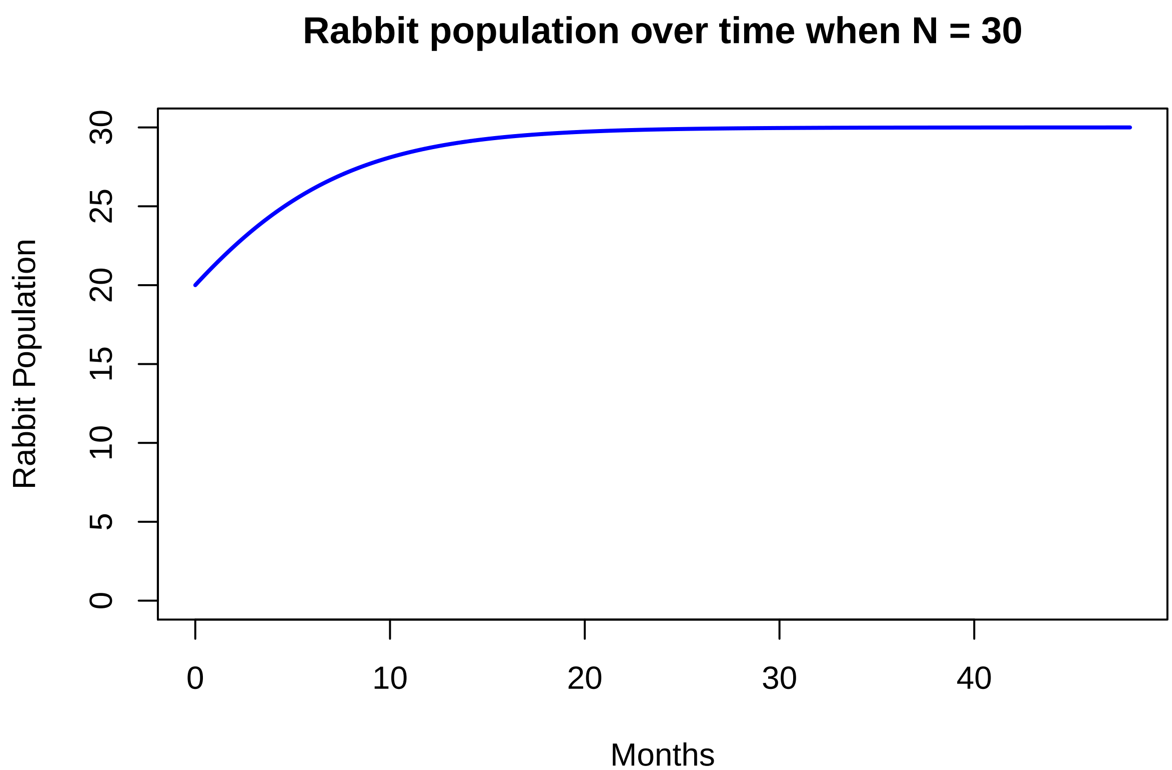
When $R(t) > N$, the graph is a decreases at a decreasing rate over time to plateau at N.

```
N <- 20

r_pop <- r_pop_time(R0, N, r, times)

plot(times, r_pop,
     type = "l", col = "blue" ,
     xlab = "Months", ylab = "Rabbit Population",
     lwd = 2, ylim = c(0, max(r_pop)),
     main = "Rabbit population over time when N = 20")
```



When $R(t) = N$, the graph is a straight horizontal line. This means there is no change in the population of the rabbit.

```
N <- 30

r_pop <- r_pop_time(R0, N, r, times)

plot(times, r_pop,
     type = "l", col = "blue" ,
     xlab = "Months", ylab = "Rabbit Population",
     lwd = 2, ylim = c(0, max(r_pop)),
     main = "Rabbit population over time when N = 30")
```



When $R(t) < N$, the graph increases at a decreasing rate over time to plateau at N.

This means that N is the the number at which the population of the rabbit stabilizes.

Rabbits and Sheep

With the Foxes out of the picture, Sheep join Rabbits in their land. If only one of the species were present, the population would grow to its carrying capacity and then flatten out, guided by

$$\dot{R}(t) = rR(t)\left[1 - \frac{R(t)}{N_R}\right]$$

and

$$\dot{S}(t) = sS(t)\left[1 - \frac{S(t)}{N_S}\right],$$

where r and s are the respective growth rates and N_R and N_S are the carrying capacities for the two species.

Because Sheep and Rabbits desire the same limited resources, there will be interspecies *competition*. Unlike the Fox-Rabbit scenario where the interaction was beneficial to one of the species and highly traumatic for the other one, here the interaction is harmful to both Sheep and Rabbits. This can be written as

$$\dot{R}(t) = rR(t)\left[1 - \frac{R(t)}{N_R}\right] - aR(t)S(t)$$

$$\dot{S}(t) = sS(t)\left[1 - \frac{S(t)}{N_S}\right] - bR(t)S(t)\,.$$

The terms a and b do not have to be equal. After all, one of the species might be harmed more by the competition than the other (for example, Sheep not only eat the food, but also trample on Rabbit burrows).

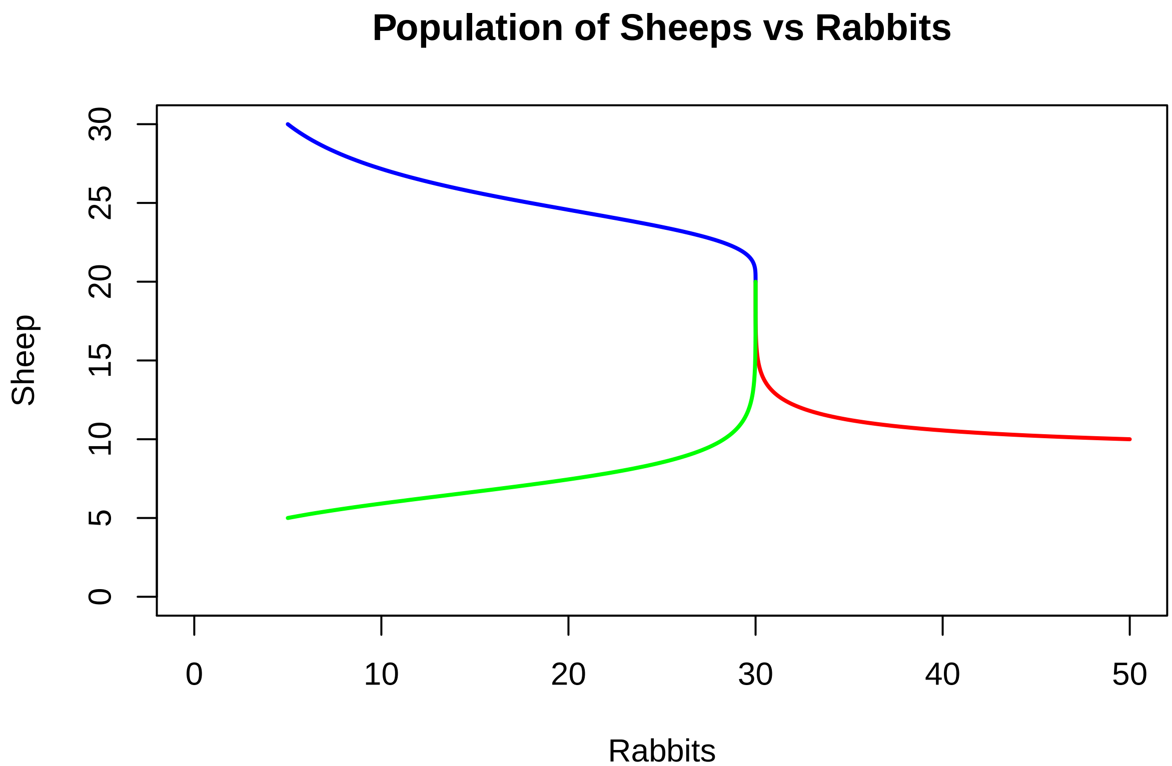
Exercise 3

Repeat **Exercises 1** and **2** from the seminar for this system. You can just copy and modify the code. Code recycling is an important skill. Do not do the same thing multiple times!

There is a great way of illustrating population dynamics of two species. As an example, let us take the case where the animals do not compete ($a = b = 0$). Here, we expect the Rabbit and Sheep populations to reach their own respective carrying capacities. Instead of plotting the populations versus time, we plot the Sheep population versus the Rabbit population as follows:

```r
# Copying the code from previous seminar

change_rate <- function(num_R, num_S, r, s, Nr, Ns, a, b) {
  R_dot <- (r * num_R * (1 - (num_R / Nr))) - (a * num_R * num_S)
  S_dot <- (s * num_S * (1 - (num_S / Ns))) - (b * num_R * num_S)

  return(c(R_dot, S_dot))
  }

animals_time <- function(R0, S0, r, s, Nr, Ns, a, b, times) {
  # Get the number of entries in the times list num_steps = length(times)
  num_steps = length(times)
  # Initialize the lists to hold R(t) and F(t)
  Rs <- matrix(0, 1, num_steps)
  Ss <- matrix(0, 1, num_steps)
  # Set the initial values
  Rs[1] <- R0
  Ss[1] <- S0
  # Write the portion of the function that fills Rs and Fs with
  # the correct values:
  for (i in 2:num_steps) {
    delta <- times[i] - times[i - 1]
    change_rate_ <- change_rate(Rs[i - 1], Ss[i - 1], r, s, Nr, Ns, a, b)
    Rs[i] <- Rs[i - 1] + (delta * change_rate_[1])
    Ss[i] <- Ss[i - 1] + (delta * change_rate_[2])
  }

  return( rbind(Rs, Ss) )
  }

t_max <- 240
num_steps <- 1000
times <- seq(0, t_max, length.out = num_steps)

r <- 0.2; s <- 0.05; Nr <- 30; Ns <- 20; a <- 0; b <- 0

R0 <- 5; S0 <- 30
animal_pop1 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 50; S0 <- 10
animal_pop2 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 5; S0 <- 5
animal_pop3 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)
```

```r
plot(animal_pop1[1, ], animal_pop1[2, ], type = "l", col = "blue"
    , xlab = "Rabbits", ylab = "Sheep", lwd = 2
    , xlim = c(0, 50)
    , ylim = c(0, 30)
    , main = "Population of Sheeps vs Rabbits"
    )
lines(animal_pop2[1, ], animal_pop2[2, ], col = "red", lwd = 2)
lines(animal_pop3[1, ], animal_pop3[2, ], col = "green", lwd = 2)
```



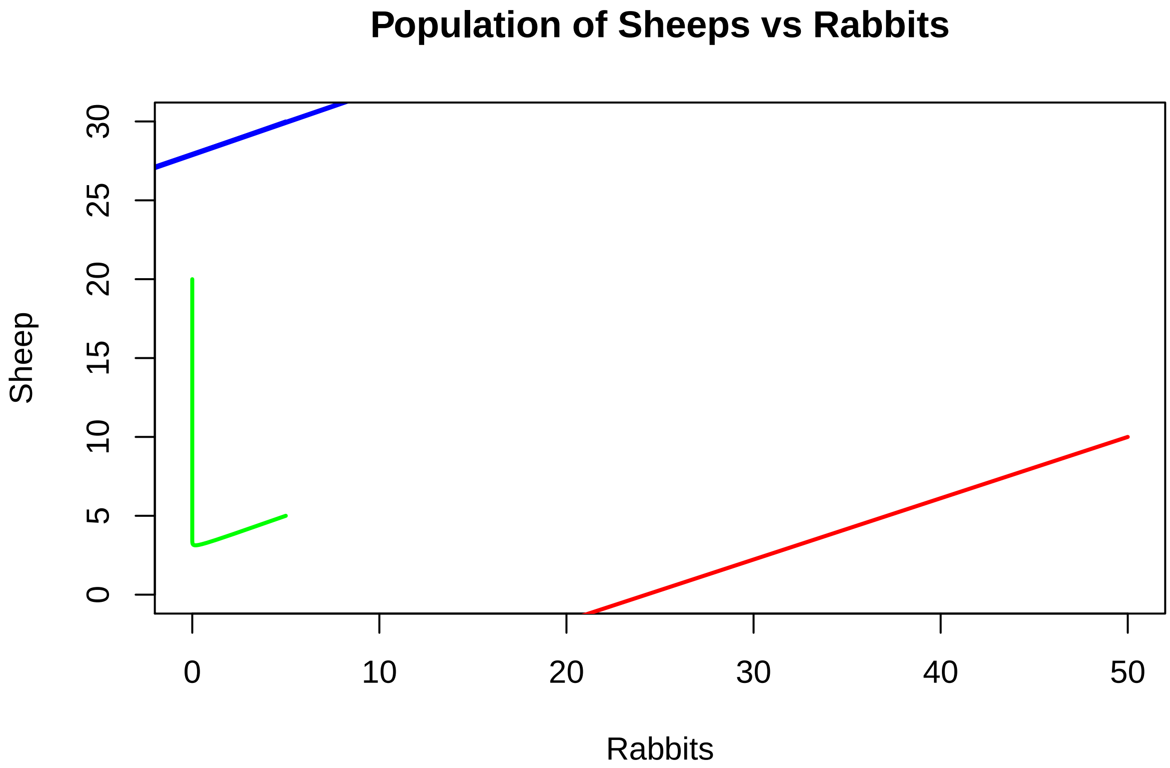
Plotting the populations for several different *initial conditions*, we see that they all arrive to the same numbers given by the carrying capacity.

Exercise 4

Reuse the code above and start dialing up the a and b parameters. Keep all other parameters the same. You should see that for large enough values of a and b the solutions show one of the species going extinct and another reaching its carrying capacity.

```r
r <- 0.2; s <- 0.05; Nr <- 30; Ns <- 20; a <- 0.5; b <- 0.2

R0 <- 5; S0 <- 30
animal_pop1 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 50; S0 <- 10
animal_pop2 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 5; S0 <- 5
animal_pop3 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

plot(animal_pop1[1, ], animal_pop1[2, ], type = "l", col = "blue"
```

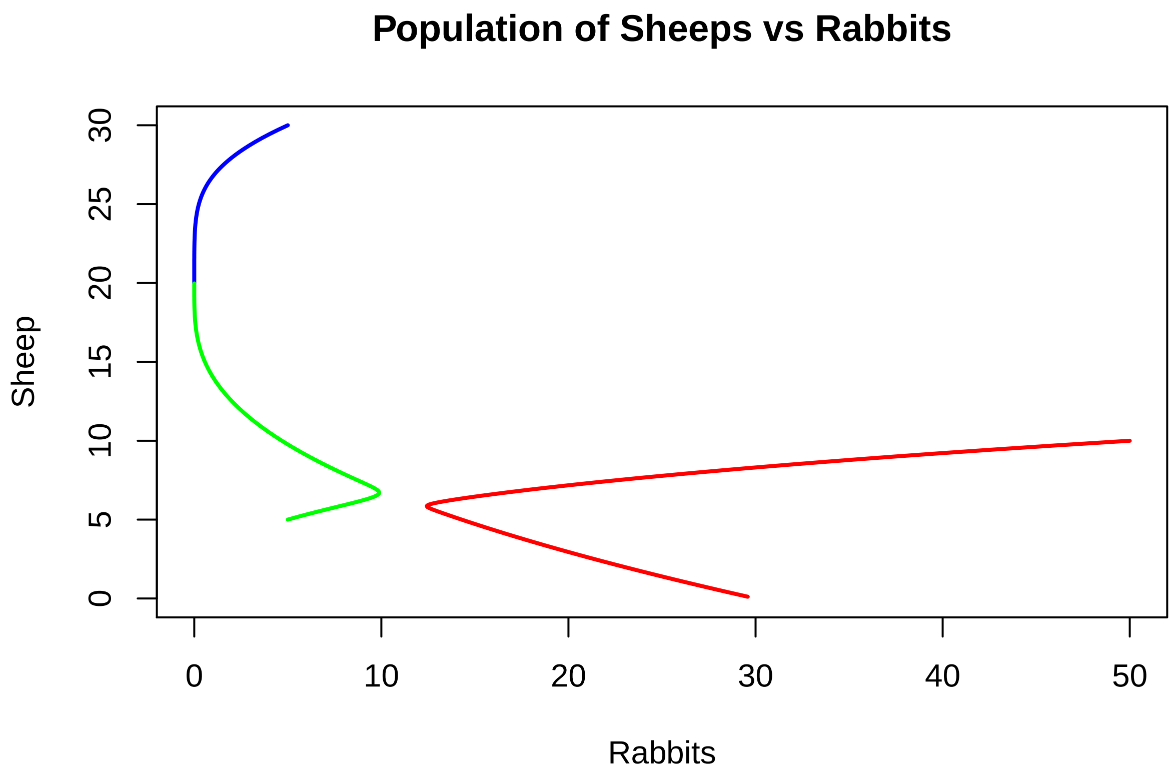
```
    , xlab = "Rabbits", ylab = "Sheep", lwd = 2
    , xlim = c(0, 50)
    , ylim = c(0, 30)
    , main = "Population of Sheeps vs Rabbits"
    )
lines(animal_pop2[1, ], animal_pop2[2, ], col = "red", lwd = 2)
lines(animal_pop3[1, ], animal_pop3[2, ], col = "green", lwd = 2)
```



After you've played with a and b to get some feel of what's happening, explore the vicinity of $a = 0.02$ and $b = 0.003$. Here, you will see that a small variation in a and b favors the survival of a particular species based on the initial conditions. You solutions will appear to "jump" from one point to another as you tune the parameters.

```
r <- 0.2; s <- 0.05; Nr <- 30; Ns <- 20; a <- 0.02; b <- 0.003

R0 <- 5; S0 <- 30
animal_pop1 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 50; S0 <- 10
animal_pop2 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

R0 <- 5; S0 <- 5
animal_pop3 <- animals_time(R0, S0, r, s, Nr, Ns, a, b, times)

plot(animal_pop1[1, ], animal_pop1[2, ], type = "l", col = "blue"
    , xlab = "Rabbits", ylab = "Sheep", lwd = 2
    , xlim = c(0, 50)
    , ylim = c(0, 30)
    , main = "Population of Sheeps vs Rabbits"
    )
```

```r
lines(animal_pop2[1, ], animal_pop2[2, ], col = "red", lwd = 2)
lines(animal_pop3[1, ], animal_pop3[2, ], col = "green", lwd = 2)
```



The graphs show that the initial population of the sheeps and rabbits have enormous impact on how the population will change over time.

In addition to being a basic introduction to farming, the point of this exercise is to show the incredible complexity that can arise even in simple systems. Small changes to parameters or initial conditions lead to drastically different results. These effects are magnified for more complex systems, such as Earth's atmosphere.