Exploring modeling with data and differential equations using R

John M. Zobitz

Version 2.0.0

Contents

Welcome {#welcome}

This book is written for you: the student learning about modeling and differential equations. Perhaps you first encountered models, differential equations, and better yet, building plausible models from data in your Calculus course.

This book sits "at the intersection" of several different mathematics courses: differential equations, linear algebra, statistics, calculus, data science - as well as the partner disciplines of biology, chemsitry, physics, business, and economics. An important idea is one of *transference* where a differential equation model applied in once context can also be applied (perhaps with different variable names) in a separate context.

I intentionally emphasize models from biology and the environmental sciences, but throughout the text you can find examples from the other disciplines. I hope you see the connections of this content to your own intended major.

This book is divided into 4 parts:

1. Modeling with differential equations and data.
2. Model parameter estimation.
3. Stability analysis for differential equations.
4. Modeling with stochastic differential equations.

This may seem like a different order than traditionally presented. This is a "modeling first" paradigm that first introduces models, and equally important, how to estimate parameters for a model using data. This conversation between models and data are important to help build plausibility. Stability analysis helps to solidify the connection between models and parameters (which may change the underlying dynamical stability). Finally the notion of *randomness* is extended with the introduction of stochastic differential equations.

0.1 Computational code

This book makes heavy use of the R programming language, and unabashedly develops programming principles using the `tidyverse` syntax and programming approach. This is intentional to facilitate direct connections to courses in introductory data science or data visualization. Throughout the years learning (and teaching) different programming languages I have found R to be the most versatile and adaptable. The `tidyverse` syntax has also been transformational for me in my own work and as my students - the barrier to compute and write code is lowered.

There is a companion R package available to run programs and functions in the text. Instructions to do so are given in Section **??**. The minimum version of R Version 4.0.2 (2020-06-22) and RStudio is Version 1.4.1106.

0.2 Acknowledgments

This book has been developed over the course of several years and has been written across different continents.

- **Augsburg University:** You have been my professional home for over 14 years and given me the space to be intellectually creative in my teaching. I have great colleagues to work with.

- **Augsburg University students:** Thank you for your interest in this topic, providing honest and insightful feedback about the course. This has been a work in progress (albeit bumpy at times).

- **My family:** Shannon, Colin, Grant, and Phoebe for humoring me while this project has been completed.

- **Waterparks, coffee shops, soccer practices:** Many times this was written "in the spaces" between work and home, and especially during downtimes when my kids could play. Turns out my kids love waterparks. Who knew?

0.3 Copyright

Part I

Stochastic Differential Equations

Chapter 1

Stochastic Biological Systems

So far in this course we have studied *deterministic* differential equations. Given information about the solution or certain values of the parameters we could characterize the long term behavior of the system. In this part we will study *stochastic* differential equations (or SDEs for short) - which means that the differential equation is subject to randomness - either in the parameters (which may cause a change in the stability for a time) or in the variables themselves.

Stochastic differential equations are a neat field of study because the field can be studied using computational approaches with some interesting mathematical approaches. In this part we will give you a taste of how to incorporate stochastics into your differential equation and focus on some solution techniques. Some of the ideas we will learn can be applied in other contexts. For example, Brownian motion is the foundation to modern physics or can be applied to modeling changes in the stock market.
Hopefully the previous sections in the textbook imparted a sense (1) that parameter values may change according to the data used to estimate them (parameter estimation) and (2) changing a parameter may drastically alter the long-term dynamics of a system (bifurcation). Learning about stochastic differential equations will extend your knowledge.

Understanding how to model SDEs requires learning some new mathematics as well as ways to numerically simulate this mathematics. A lot of SDEs rely on numerical simulation, so we will build up our understanding of some of the results of numerical simulation first.

1.1 A discrete system

Let's focus on an example that involves discrete populations. Moose are large animals (part of the deer family) weighing 1000 pounds that can be found in Northern Minnesota link. Here is a picture of a moose: link. While in the early 2000's the population of the moose was 8000, recent estimates have the numbers at 3000, although that seems to have stabilized link

A starting model that describes their population dynamics is the discrete dynamical system:

$$M_{t+1} = M_t + b \cdot M_t - d \cdot M_t, \tag{1.1}$$

where M_t is the population of the moose in year t, and b the birth rate and d the death rate. This equation can be reduced down to $M_{t+1} = rM_t$ where $r = 1 + b - d$ is the net birth/death rate. This model states that the population of moose in the next year equals the current population, added to any fraction of births and taking away any deaths.

This discrete dynamical system is a little bit different from a continuous dynamical system, but can be simulated pretty easily by defining a function.

```r
M0 <- 3000  # Initial population of moose
N <- 5  # Number of years we simulate

moose <- function(r) {
  out_moose <- array(M0,dim=N)
  for (i in 1:(N-1)) {
    out_moose[i+1] <- r*out_moose[i]
  }
  return(out_moose)
}
```

Notice how the function `moose` returns the current population of moose after N years with the net birth rate r. Let's take a look at the results for different values of r:

```
moose_rates <- tibble( years = 1:N,
                     r0.4 = moose(0.4),
                     r0.8 = moose(0.8),
                     r1.1 = moose(1.1))


ggplot(data = moose_rates) +
  geom_line(aes(x=years,y=r0.4),color='red') +
  geom_line(aes(x=years,y=r0.8),color='blue') +
  geom_line(aes(x=years,y=r1.1),color='green') +
  labs(x='Years',
       y='Moose')
```

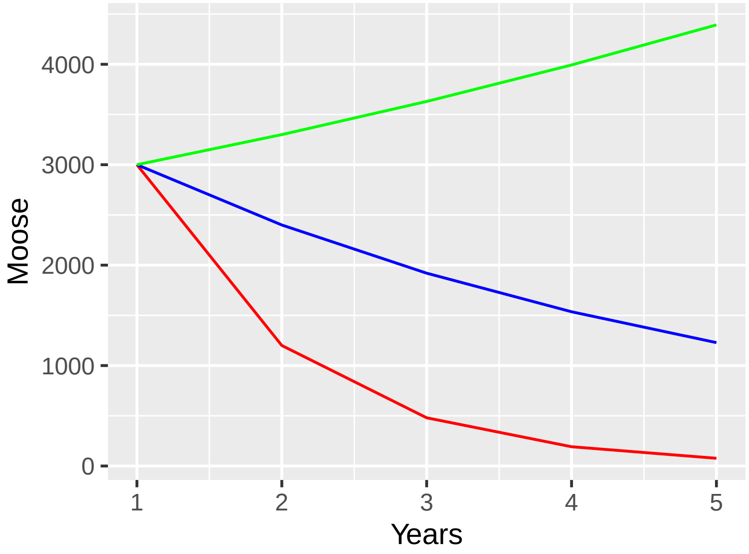

Figure 1.1: Simulation of the moose population with different birth rates.

Let's remind ourselves of what is going on in the code.

- `out_moose <- tibble...` creates a data frame that we can use for plotting. We call several different instances of the `moose` code for different net birth rates. I've decided to label each of those instances with the value of the birth rate r for reference.
- The command `geom_line` makes a line plot. Remember that we need to specify both the x (horizontal) and y (vertical) axes. I specified the different colors to distinguish the different birth rates.

Notice how for some values of r the population starts to decline, stay the same, or increase. Let's analyze this system a little more. Just like with a continuous differential equation we want to look for solutions that are in steady state, or ones were the population is staying the same. In other words this means that $M_{t+1} = M_t$, or $M_t = rM_t$. If we simplify this expression this means that $M_t - rM_t = 0$, or $(r-1)M_t = 0$. Assuming that M_t is not equal to zero, then this equation is consistent only when $r = 1$. This makes sense: we know $r = 1 - b - d$, so the only way this can be one is if $b = d$, or the births balance out the deaths.

Ok, so we found our equilibrium solution. What is the general form of this solution? Just like when we were solving continuous systems and we assumed an exponential solution we will do the same here, but this time we represent the solution as $M_t = A \cdot v^t$, which is an exponential equation. Since we have $M_0 = 3000$, then $A = 3000$. Plugging this expression into our equation we have:

$$100 \cdot v^{t+1} = r3000 \cdot v^{t} \tag{1.2}$$

Our goal is to figure out a value for v that is consistent with this expression. Just like we did with the equilibrium solution we can arrange to the following equation, using the fact that $v^{t+1} = v^{t} \cdot v$:

$$3000v^{t}(v - r) = 0 \tag{1.3}$$

Since we assume $v \neq 0$, the only possibility is if $v = r$. Aha! Our general solution then is

$$M_t = 3000r^{t} \tag{1.4}$$

We know that if $r > 1$ we have exponential growth and $r < 1$ exponential decay, consistent with our results above.

There is some comfort here: just like in continuous systems we find the eigenvalues that determine the stability of the equilibrium solution. For discrete systems the stability is based on the value relative to one (not zero) - so it is good to be aware of what type of system you are looking at!

Now that we have an understanding of how this system works we can begin to look at stochasticity.

1.2 Environmental Stochasticity

It may be the case that environmental effects drastically change the net birth rate from one year to the next. For example it is known that in snowy winters the net birth rate changes because it is difficult to find food @ref(carroll_modeling_2013). For our puruposes, let's say that in snowy winters r changes from 1.1 to 0.7. This would be a pretty drastic effect on the system - as one case is associated with exponential growth and the other exponential decay.

Because the years when this snowfall could occur is random we need to account for this in our model. One way is to create a conditional statement based on the probability of the moose to adjust to a deep snowpack. We will define this probability to be on a scale from 0 to 1, where 0 means the moose cannot adjust to a deep snowpack, and 1 they are able to adjust to a deep snowpack. How we implement that is writing a function that draws a uniform random number each year and adjusting the birth rate:

```
# We use the snowfall_rate  as an input variable

moose_snow <- function(snowfall_prob) {
  out_moose <- array(M0,dim=N)
   for (i in 1:(N-1)) {
    r = 1.1  # Normal net birth rate
    if (runif(1)>snowfall_prob) {
     r = 0.7   # Decreased birth rate
    }
    out_moose[i+1] <- r*out_moose[i]
   }
 return(out_moose)
 }
```

Let's take a look at some solutions for different realizations of the moose population over time when the probability to adjust to the snowfall rate varies.

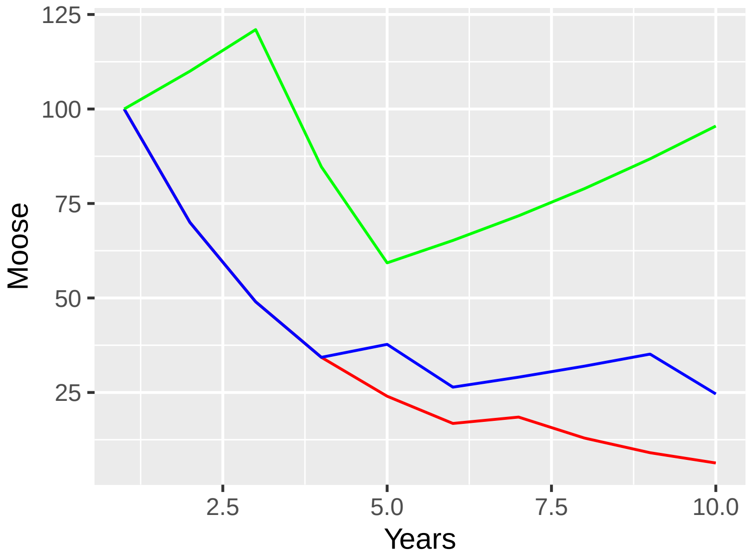

Figure 1.2: Moose populations with different probability of adjusting to deep snowpacks.

Run the above code on your computer. Do you obtain the same result? I hope not! We are drawing random numbers for each year, so you should have different trajectories. While this may seem like a problem, one key thing that we will learn in Section **??** is when we compute *multiple* simulations and then compute an ensemble average we see patterns in the results.

As you can see when the probability of adjusting to deep snowpacks is very high ($p = 0.75$), the population continues exponentially. If that probability is lower, it can still increase, but one bad year does knock the population down.

The moose example introduced how random effects into the dynamical system. Both contexts (environmental or demographic stochasticity) affected the net reproduction rate r, but in different ways. The important lesson is that the *type* of stochasticity matters just as much as *how* it is implemented.

1.3 Discrete systems of equations

Finally, one way that we can extend the moose model is to account for both adult and juvenile moose populations. Here is one possible model:

$$\begin{cases} J_{t+1} & = F_M \cdot M_t \\ M_{t+1} & = G_J \cdot J_t + P_M \cdot M_t, \end{cases} \tag{1.5}$$

Equation (1.5) is a little different from (1.1) because it includes juvenile and adult moose populations, which has the following parameters:

- F_M: represents the birth rate of new juvenile moose
- G_J: represents the maturation rate of juvenile moose
- P_M: represents the survival probability of adult moose

We can code up this model in the following way:

```
M0 <- 900  # Initial population of adult moose
J0 <- 100  # Initial population of juvenile moose

N <- 10    # Number of years we run the simulation
moose_two_stage <- function(f,g,p) {

  # f: birth rate of new juvenile moose
  # g: maturation rate of juvenile moose
  # p: survival probability of adult moose

  # Create a data frame of moose to return
  out_moose <- tibble(years = 1:N,
                      adult = array(M0,dim=N),
                      juvenile = array(J0,dim=N)
  )

  # And now the dynamics
  for (i in 1:(N-1)) {
    out_moose$juvenile[i+1] <- f*out_moose$adult[i]
    out_moose$adult[i+1] <- g*out_moose$juvenile[i] + p*out_moose$adult[i]
  }

  return(out_moose)
}
```

To simulate the dynamics we just call the function `moose_two_stage` and plot:

```
moose_two_stage_rates <- moose_two_stage(f = 0.5,
                                         g = 0.6,
                                         p = 0.7
                                         )

ggplot(data=moose_two_stage_rates) +
  geom_line(aes(x=years,y=adult),color='red') +
  geom_line(aes(x=years,y=juvenile),color='blue') +
  labs(x='Years',
     y='Moose')
```

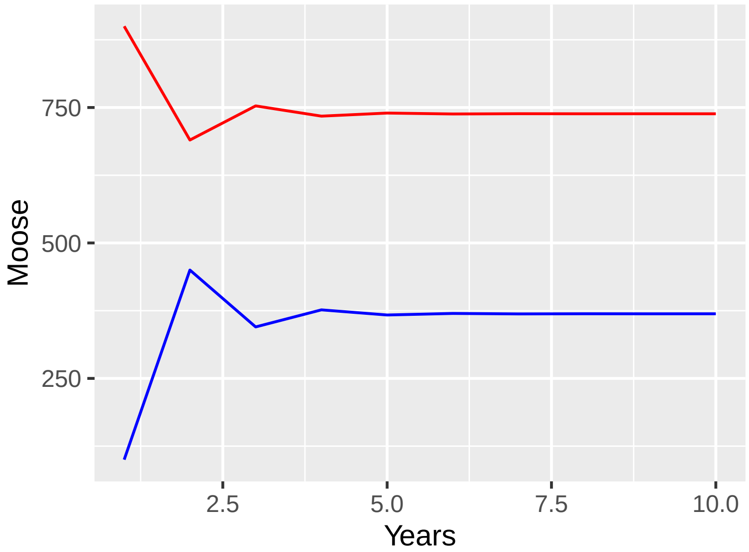

Figure 1.3: Simulation of a two stage moose population model. Adults are in red, juveniles in blue.

Looking at Figure 1.3, it seems like both populations stabilize after a few years. We could further analyze this model for stable population states (in fact, it would be similar to determining eigenvalues as in Section **??**).

1.4 Exercises

Exercise 1.1. Re-run the moose population model with probabilities of adjusting to the deep snowpack at $p = 0,\ 0.1,\ 0.9,\ 1$. How does that adjusting the probability affect the moose population after 10 years?

Exercise 1.2. Modify the two stage moose population model (Equation (1.5)) with the following parameters and plot the resulting adult and juvenile populations:

a. $f = 0.6$, $g = 0.6$, $p = 0.7$

b. $f = 0.5$, $g = 0.6$, $p = 0.4$

c. $f = 0.3$, $g = 0.6$, $p = 0.5$

Exercise 1.3. An animal reproduces two, one, or no offspring. The chance it produces one offspring is 0.50, two offspring 0.25, and no offspring 0.25. This animal does not survive after reproducing. Use the function `population` in the `MAT 369Code` library to produce 1000 realizations of this stochastic process over 20 generations. Assume the initial population size is 10 individuals. Comment on the long term dynamics of the population.

Exercise 1.4. You are playing a casino game. If you win the game earn a dollar. If you lose the game you lose one dollar. The probability of winning or losing is 50-50 (0.50). You start the game with $100. You play the game 200 times. Use the function `casino` in the `MAT369Code` library to produce 1000 realizations of this stochastic process. Comment on the long term dynamics of your earnings. Then assuming the house win probability is 0.52 figure out how long the game, on average, runs before you are broke. Finally, adjusting the house win probability, hypothesize a function of the length of game as a function of the house win probability.

Exercise 1.5. Modify the two stage moose population model (Equation (1.5)) to account for years with large snowdepths. In normal years, $f = 0.5$, $g = 0.6$, $p = 0.7$. However for snowy years, $f = 0.3$, $g = 0.6$, $p = 0.5$. Generate code that can account for these variable rates (similar to the moose population model). Plot simulations when $N = 10$ and $N = 30$ and comment on the long-term dynamics of the moose.

Exercise 1.6. A population grows according the the growth law $x_{t+1} = r_t x_t$.

a. Determine the general solution to this discrete dynamical system.

b. Plot a sample growth curve with $r_t = 0.86$ and $r_t = 1.16$, with $x_0 = 100$. Show your solution for $t = 50$ generations.

c. Now consider a model where $r_t = 0.86$ with probability 1/2 and $r_t = 1.16$ with probability 1/2. Write a function that will predict the population after $t = 50$. Show three or four different realizations of this stochastic process.

Exercise 1.7. A “patch’ ’ has area a, perimeter s, and a strip (band) of width w inside the boundary of a from which the animals disperse. Only those in the strip disperse. Let u_t be the number of animals in a at any time t. The growth rate of all the animals in a is r. The rate at which animals disperse from the strip is proportional to the fraction of the animals in the strip, with proportionality constant ϵ, which is the emigration rate for those in the strip.

a. Draw a picture of the situation described above.

b. Explain why the equation that describes the dynamics is

$$u_{t+1} = ru_t - \epsilon \left(\frac{w \cdot s}{a} u_t \right)$$

c. Determine conditions on the parameters r, w, s, ϵ, and a under which the population is growing.