

Exercise 10 - Simulation modeling with ordinary differential equations (ODEs)

As usual begin this week, by picking a partner and forking the TA's Exercise 10 Github repo. Then provide collaborative access to the other partner. Clone the forked repo so that you have the required files. Be sure to commit regularly to show how you arrived at your solutions and demonstrate coding effort by both partners.

1. Imagine you've been tasked with giving a lecture to teach high school students about density-dependent population growth. You've been told to generate the three figures described below to demonstrate the influence of growth rate (r), carrying capacity (K), and initial population size (N_0) on temporal dynamics of a population. Write a script with all code necessary to generate the three figures below.

- A plot of population size as a function of time with five populations (one line per population) possessing different maximum growth rates ($r = -0.1, 0.1, 0.4, 0.8, \text{ and } 1.0$). Use $K = 100$ and $N_0 = 10$ (these were somewhat arbitrarily chosen to make an informative figure).
- A plot of population size as a function of time with three populations (one line per population) possessing different carrying capacities ($K = 10, 50, \text{ and } 100$). Use $r = 0.2$ and $N_0 = 1$ (these were somewhat arbitrarily chosen to make an informative figure).
- A plot of population size as a function of time with three populations (one line per population) that shows the effect of initial population size ($N_0 = 1, 50, \text{ and } 100$). Use a $r = 0.1$ and $K = 50$ for these simulations.

Think about how for-loops might help make your code more efficient!

2. Often simulation models are used to develop an intuition about how a system works. We describe the system using some simple differential equations and then we try a variety of parameter values to explore how the system responds. All the while we are making the strong assumption that the real system we care about behaves somewhat like our simple model. Let's use a simple model of disease transmission to learn about this process! One simple model used for simulating disease transmission is called an SIR model because it considers three sub-groups of a population - susceptible (S), infected (I), and resistant (R). This model doesn't actually model the disease causing agent at all, but rather focuses on the host of the disease. The model does, however, include parameters that describe the transmission of the disease causing agent and the rate a host recovers from the disease. The simplest SIR model assumes a constant population size and looks like this:

$$\frac{dS}{dt} = -\beta IS$$

$$\frac{dI}{dt} = \beta IS - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

When considering dynamics of a disease epidemic we can quantify:

- *incidence* - the number of new infections occurring over a defined time interval ($I_t - I_{t-1}$)
- *prevalence* - the fraction of the population that is infected at a given time ($\frac{I}{S+I+R}$)
- *percent affected* - the fraction of the population that was sick at any time during an epidemic ($\frac{I+R}{S+I+R}$)
- *basic reproduction number* ($R_0 = \frac{\beta(S+I+R)}{\gamma}$) - the number of cases one case generates on average over its infectious period in an otherwise uninfected population

Write a custom function describing the SIR model above, and for each row in the table below, use the pair of parameter values (β and γ) to simulate the model for 500 days. Start each simulation with 999 susceptible, 1 infected, and 0 resistant individuals. For each simulation, calculate the maximum daily incidence and maximum daily prevalence. Also calculate the percent affected over the simulation (use the last time step of the simulation for this) and the basic reproduction number (you can actually do this without simulating, since we give you β , γ , and the initial $S + I + R$).

β	γ
0.0005	0.05
0.005	0.5
0.0001	0.1
0.00005	0.1
0.0001	0.05
0.0002	0.05
0.0001	0.06

What patterns do you see in these results? Try additional parameter combinations to identify key values or trends with β , γ , or R_0 (the basic reproduction number). Add text (a paragraph or two) to the readme file in your GitHub repository summarizing what you've learned. Include the code you generated to learn about the patterns you summarize as a file in your Github repository as well.

Devise a plan for splitting up the work and generating the required code. Do this in parallel, not sequentially. Don't forget to check and edit each other's code. Remember to frequently **add-commit** locally and **push-pull** to GitHub to avoid conflicts. Also, remember you don't have to be in the same place at the same time to work on this collaboratively thanks to GitHub!!!

Turning in your assignment via GitHub

Once you have committed all changes to your local Git repos and pushed all of those commits to the forked repo on GitHub, you can "turn in" your assignment using a **pull request**. This can be done from the GitHub repo website. When viewing the forked repo, select "Pull requests" in the upper middle of the screen, then click the green "New pull request" button in the upper right. You'll then see a screen with a history of commits for you and your collaborator, select the green "Create pull request button". In the text box next to your user icon near the top of the page, remove whatever text is there and add "owner's last name - collaborator's last name submission", but obviously substitute your last names. If I and Ann Raiho worked on the project together the text would read "jones-raiho submission". Then click the green "Create pull request" button. **Only one of you will need to create a pull request.**