Problem set 8

Exercise 1. The SIR model (R and Python)

The SIR model describes the spread of an epidemic. Let S(t), I(t), and R(t) be the number of Susceptible, Infected, and Recovered individuals at time t. We consider these functions as continuous quantities, even though they are necessarily discrete in practice. Suppose that each Infected individual becomes Recovered at rate β, and each Infected individual infects each Susceptible at rate α, then we have:

$$\frac{dS}{dt} = -\alpha \cdot I(t) \cdot S(t)$$
$$\frac{dI}{dt} = \alpha \cdot I(t) \cdot S(t) - \beta \cdot I(t)$$
$$\frac{dR}{dt} = \beta \cdot I(t)$$

By looking at I'(0), give a condition for the epidemic to die out quickly, in terms of α, and S(0). Test your condition by solving the system of equations numerically, for a variety of initial conditions. Write your code in R and Python.

Exercise 2. The Lanchester combat (R and Python)

The Lanchester combat model was first used to study air combat during World War I. Let A(t) be the number of planes on the side A and B(t) the number of planes on the other side B. We treat A(t) and B(t) as continuous differentiable functions, even though they are discrete.

a) The basic Lanchester model supposes that the rate of decrease in A at any given time is proportional to B and that the rate of decrease in B is proportional to A. Express the dynamics of A and B as a system of ordinary differential equations (ODEs).
b) Suppose now that side A has A0 reinforcements, that arrive at rate r1 (until they are used up), and that side B has B0 reinforcements, that arrive at rate r2. In addition, suppose that individual aircraft break down at a rate b1 for side A and b2 for side B. Incorporate these additional components into your model.
c) Write a program to simulate your model in both R and Python. Consider appropriate parameter values and plot the solutions.

Exercise 3. Monte Carlo Estimation of Pi in Python

The Monte Carlo method is a statistical technique that allows for numerical approximation of complex problems. One common use case is the approximation of the value of Pi.

Write a Python script using NumPy's random number generator to estimate the value of Pi using the Monte Carlo method. Here's the process:

- Generate a large number of random points in a 2D space (for example, 1 million points).
- Determine the proportion of points that lie within a unit circle (centered at the origin).
- Use this proportion to estimate the value of Pi.

Exercise 4. Simulating a Lottery Draw

Lotteries rely on random numbers to ensure fairness. Write a Python program that simulates a lottery draw. The lottery draw should select 6 unique numbers from 1 to 49. Use the NumPy random number generator **numpy.random.choice** function with **replace=False** to ensure that the numbers are unique.

Exercise 5. Distribution for a random variable

You have a distribution for a random variable that has a bell shape like that of a Gaussian. Looking at the distribution near the tails, you notice that 0.01 % of the points are more than 5σ (standard deviation) away from the average. Can you say that the distribution is Gaussian? Write a program to simulate 10^5 numbers for a normal distribution N(0,1) and calculate the number of points lying more than 5 away from 0.