

# Introduction to Mathematics for Data Science

## Personal Assignment 1

Zehao Qian

November 2, 2023

### 1 Question 1

#### 1.1 Modelling Bird Population Decline Due to Invasive Snakes

In the modeling of bird populations after snake invasion, I utilized **Formula 1**. This choice was made to satisfy the requirements specified in the problem statement, where the population starts decreasing gradually after the snake invasion, and then, over time, the bird population experiences a sharp decline. Additionally, the modeling process should also account for the principles of **population dynamics** (in the later stages, due to a reduced availability of birds for the snakes to prey on, the snake population would also decrease, resulting in a slowdown in the rate of decline of the bird population).

$$P(t) = \frac{A}{e^{kt^2}} \quad (1)$$

Where:

- **P(t)** is the population of birds at time 't'
- **A**: Maximum capacity, representing the constant bird population before snake invasion.
- **k** is a constant that determines the rate of decline.

Here is a brief proof where I applied SymPy to differentiate Formula 1. The code is provided below, and the result is (I preset A=1000, k=0.1):

$$P'(t) = -200.0te^{-0.1t^2} \quad (2)$$

```
1 import sympy
2
3 # Define symbolic variable
4 t = sympy.symbols('t')
```

```

5
6 # Define constants
7 # Define the maximum capacity A, the value of k,
8 # and the invasion time
9 A = 1000
10 k = 0.1
11
12 # Define the function
13 P = A * sympy.exp(-k * t**2)
14
15 # Calculate the derivative with resympyect to t
16 P_derivative = sympy.diff(P, t)
17
18 # Print the elegant mathematical expression
19 P_derivative

```

After I computed the derivative of  $P(t)$  using SymPy, I employed the Python library Matplotlib to create a graphical representation of the derivative (**Figure 1**). I found that it meets the specified criteria, with the derivative values always being less than or equal to 0. Initially, the derivative is far from the x-axis, indicating an increasing rate of decline, while in the later portion, it starts to exhibit a deceleration in the decline rate.

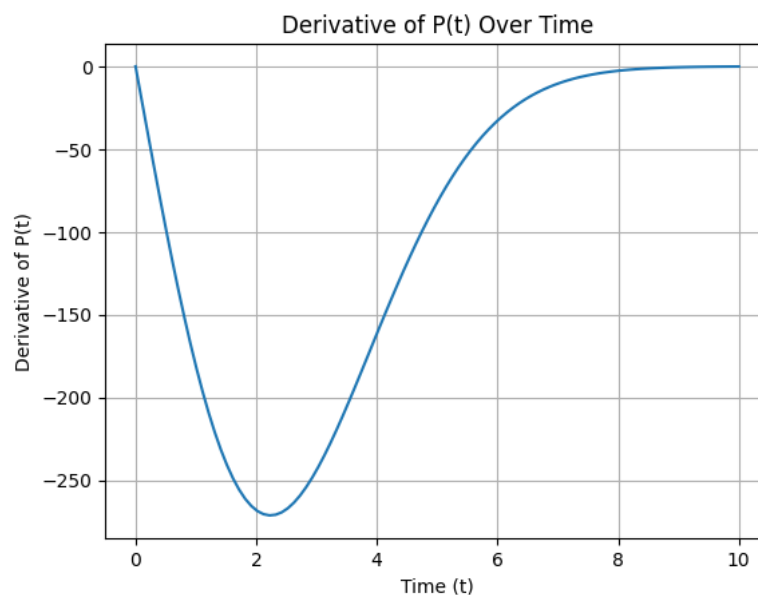


Figure 1: Derivative of  $P(t)$

Finally, I took into account the event of snake invasion by introducing an invasion event. Before this event, the bird population remains constant.

```
1 # import numpy as np
2 # import matplotlib.pyplot as plt
3 import numpy as np
4
5 def bird_population_model(t, A, k, invade_time):
6     P = np.zeros_like(t) # Initialize the population array
7
8     for i in range(len(t)):
9         if t[i] < invade_time:
10             P[i] = A # Bird population remains constant before
                       # invasion
11         else:
12             P[i] = A * np.exp(-k * (t[i] - invade_time)**2)
13
14     return P
15
16
17 invade_time = 5 # Time of snake invasion
18
19 # Generate a time range for plotting
20 t = np.linspace(0, 15, 100) # Adjust the time range as needed
21
22 # Calculate the population over time
23 P = bird_population_model(t, A, k, invade_time)
24
25 # Plot the population over time
26 plt.plot(t, P)
27 plt.xlabel("Time")
28 plt.ylabel("Bird_Population")
29 plt.title("Bird_Population_Over_Time")
30 plt.grid(True)
31 plt.show()
```

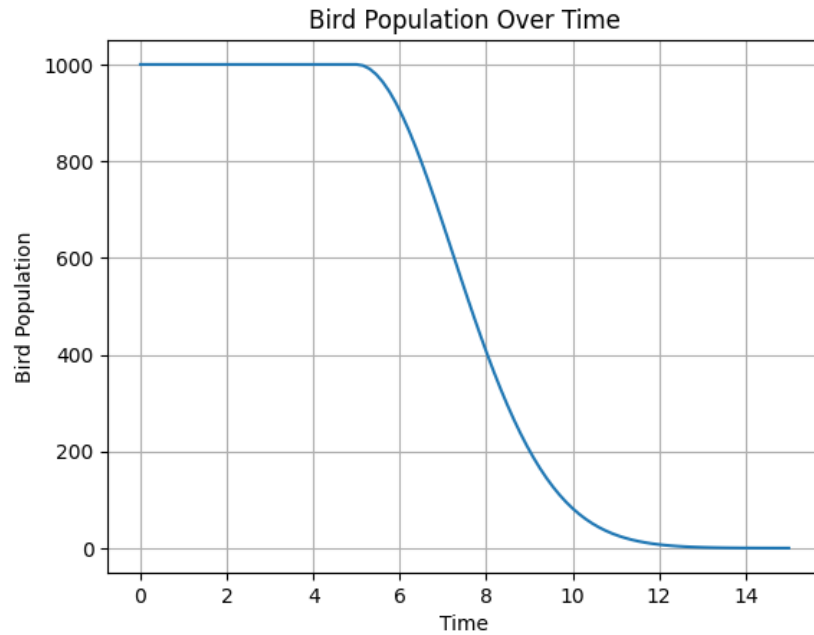


Figure 2: Figure of Bird Popullation Model

**Figure 2** represents the final graph I created.

## 1.2 Seasonal Epidemiological Models

In this code, the abscissa shows the summer and winter of each year, with the number of cases increasing in the summer and remaining the same in the winter. To achieve this, we use the integer part to determine the year in order to switch between summer and winter. Custom labels were also created for the x-axis to represent the seasons of the year.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def pandemic_model(t):
5     # Define the peak year and the duration of the pandemic
6     peak_year = 5
7     duration = 10
8
9     # Calculate the number of cases based on the given
10    conditions
11    cases = np.zeros_like(t)
12    for i in range(len(t)):

```

```

12     year = int(t[i])
13     if 0 <= year < peak_year:
14         cases[i] = (year + t[i] - year) / peak_year * (
15             duration / 2)
16     elif peak_year <= year < peak_year + duration / 2:
17         cases[i] = (1 - (year + t[i] - year - peak_year) /
18             (duration / 2)) * (duration / 2)
19
20     return cases
21
22 # Generate a time range for plotting (a decade)
23 t = np.linspace(0, 10, 100)
24
25 # Calculate the number of cases over the decade
26 cases = pandemic_model(t)
27
28 # Create custom labels for x-axis to represent summer and
29     winter
30 x_labels = [f'Year_{int(year)}_{if int(year) % 1 == 0
31     else "Winter"}' for year in t]
32
33 # Plot the number of cases over time
34 plt.plot(x_labels, cases)
35 plt.xlabel("Time_(Year_and_Season)")
36 plt.ylabel("Number_of_Cases")
37 plt.title("Pandemic_Cases_Over_a_Decade_with_Seasonal_Variation")
38
39 plt.grid(True)
40 plt.xticks(rotation=45)
41 plt.show()

```

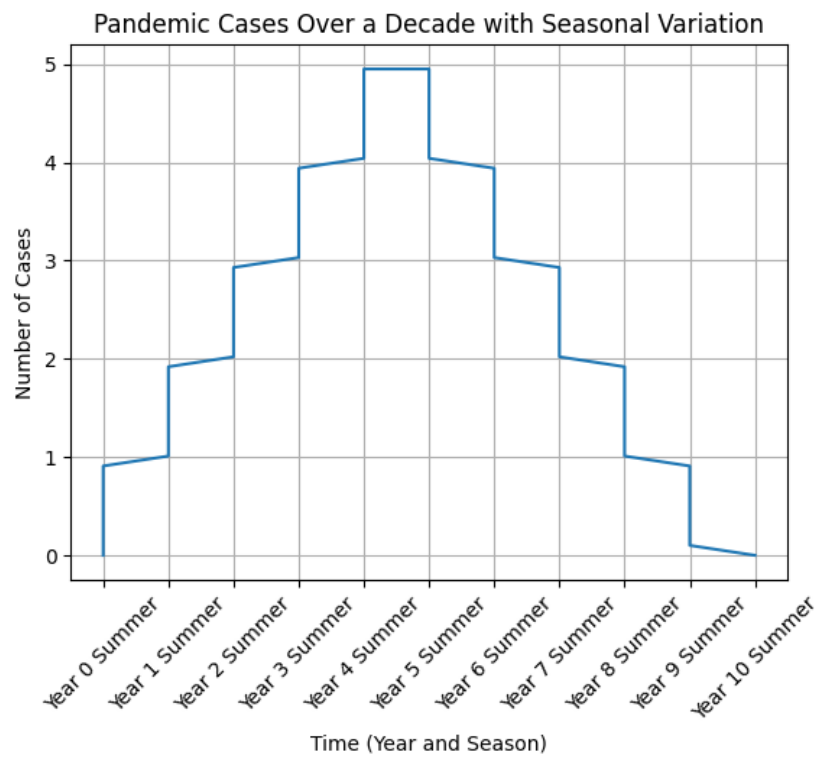


Figure 3: Figure of Pandemic Cases

The result of this code is **Figure 3**. The number of cases changes only when the Season is summer.

## 2 Question 2

### 2.1 Derivative $f'(x)$ at $x = 7$

From the definition of the derivative, it can be seen that:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3)$$

In the diagram I plotted at  $x = 7, 7.2$ , while  $y = 3.8, 4.4$ . So we let  $\Delta x = 0.2$

$$\begin{aligned} f'(x) &= \frac{f(7 + \Delta x) - f(7)}{\Delta x} \\ &= \frac{f(7.2) - f(7)}{0.2} \\ &= \frac{4.4 - 3.8}{0.2} \\ &= 3 \end{aligned}$$

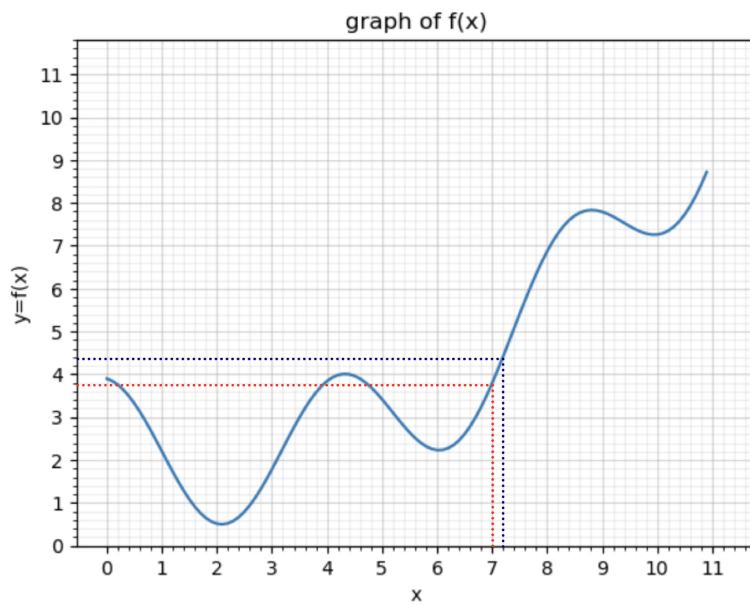


Figure 4:  $f(x)$ 's value when  $x = 7, 7.2$

## 2.2 The Estimated Value of $\int_2^6 f(x)dx$

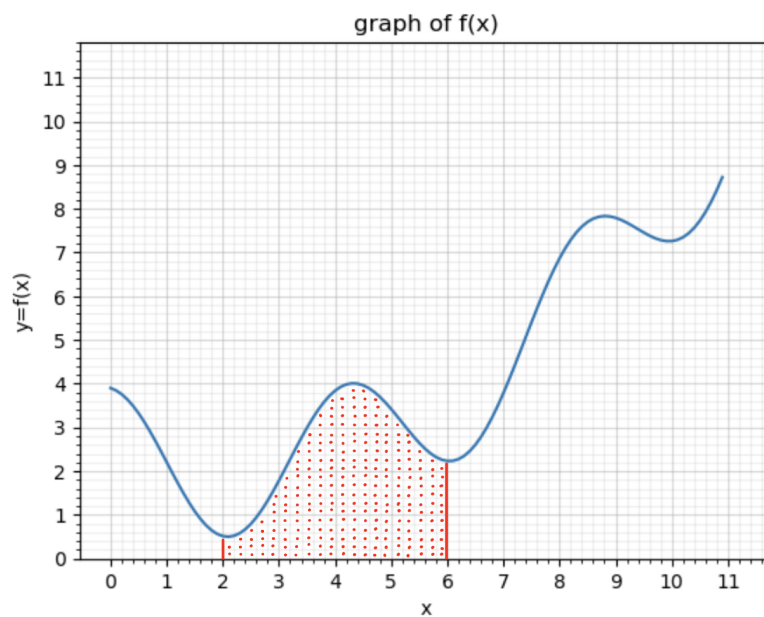
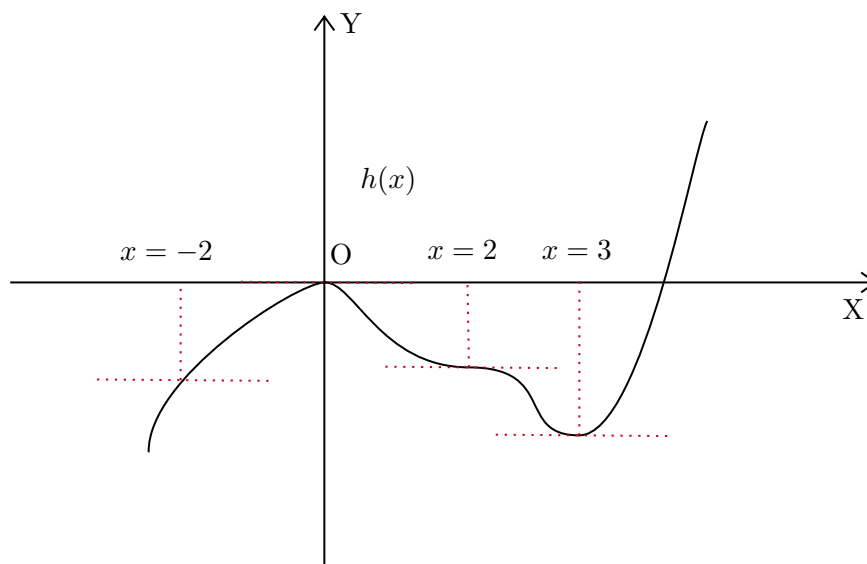


Figure 5: It is about 258 blocks from  $x = 2$  to 6

The area of each block is:  $0.2 \times 0.2 = 0.04$ , so the value of  $\int_2^6 f(x)dx$  is  $253 \times 0.04 = \mathbf{10.12}$

## 2.3 Draw the plot of function h





**Analytics:**

- For  $x < 0$ , the graph increases.
- At  $x = 0$ , there is a local maximum.
- For  $0 < x < 2$ , the graph decreases.
- At  $x = 2$ , there is a local minimum.
- For  $2 < x < 3$ , the graph decreases.
- At  $x = 3$ , there is an inflection point.
- For  $x > 3$ , the graph increases.

### 3 Question 3