Romeo and Juliet meeting

Vasilis Tsioulos[*]

[*]Project 1: Computational mathematics II 2025

ABSTRACT

This study investigates the probability of the meeting of Romeo and Juliet for given time and delay with basic probabilistic theory. The first step was to compute the probability by graphically solving the problem. The next step was to simulate the problem in python and to ask chatGPT for the solution of the problem. Then the results of mine and chatGPT were compared. This example is a great introduction to probabilistic theory and its simulation.

"*Romeo and Juliet have a date at a given time, and each will arrive at the meeting place with a delay between 0 and 1 hour, with all pairs of delays being equally likely. The first to arrive will wait for 15 minutes and will leave if the other has not yet arrived. What is the probability that they will meet?*"

1 Solving the problem

1.1 Graphical solution

Let x be the delay for Romeo and y be the delay of Juliet, where the values that they can take is $[0,1]$. The problem has two conditions in order for them to meet: 1. If Romeo arrives first, Juliet must arrive in 15 minutes from the time Romeo arrives. In other words $y \leq x+0.25$. 2. If Juliet arrives first, Romeo must arrive in 15 minutes from the time Juliet arrives. In other words $x \leq y+0.25$. To find the area where they arrive we have to take a sample space, which is a square of $[0,1]\times[0,1]$. Where a shaded area is drawn in order to see the possible pairs of delays. If $P(A)$ for the event $A \subseteq \Omega$ is equal to A's area.

$$M = \{(x,y) : |x-y| \leq \frac{1}{4}, 0 \leq x \leq 1, 0 \leq y \leq 1\} \tag{1}$$

The event that Romeo and Juliet will meet is the shaded region in fig. 1.

Its probability is calculated to be:

$$P = 1 - area \quad of \quad two \quad unshaded \quad triangles = 1 - 2\cdot(\frac{3}{4})\cdot(\frac{3}{4})/2 = \frac{7}{16} = 0.4375$$

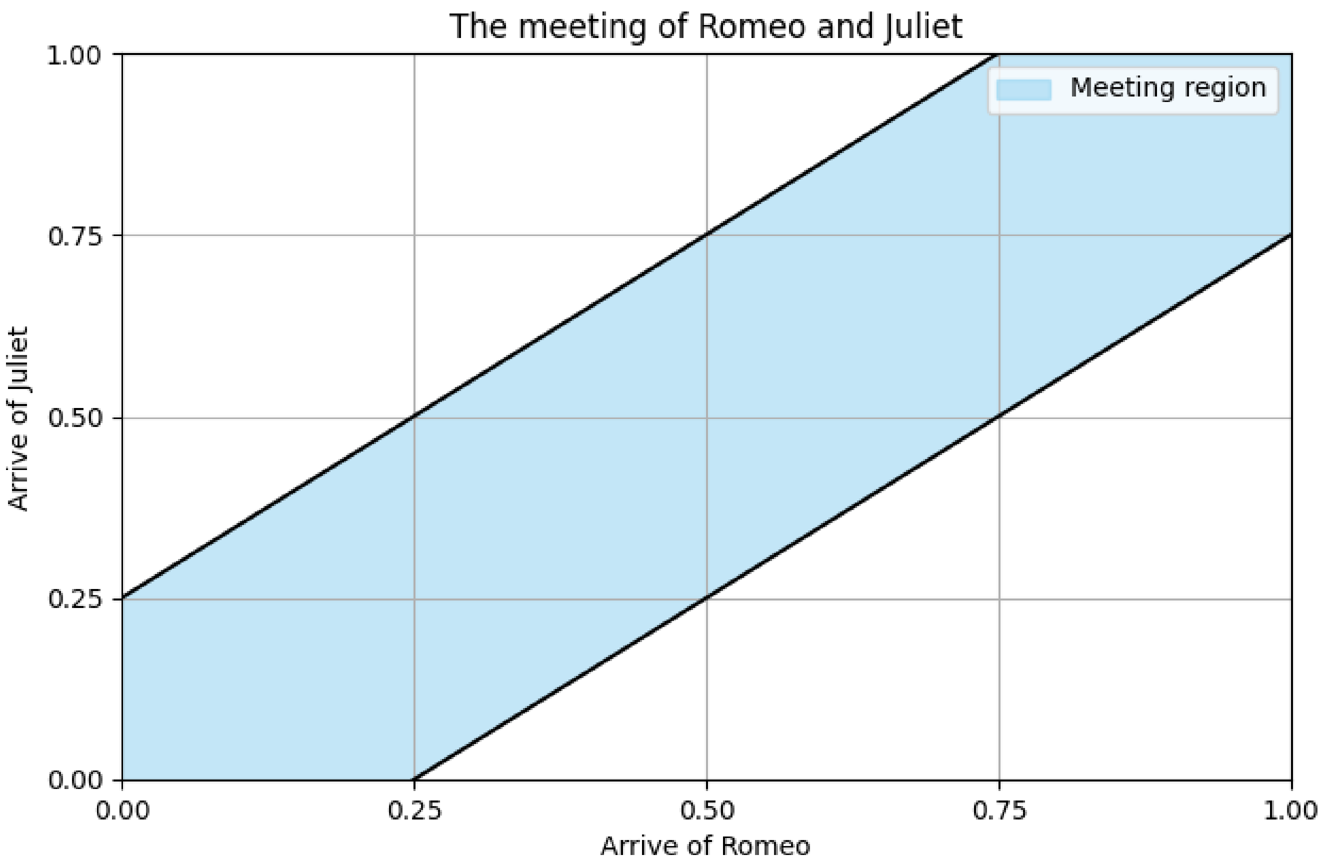

Figure 1. Romeo and Juliet simulation in Python

1.2 Python simulation

The simulation of the problem in order to compute the probability is written in Python. The first step was to import the libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Listing 1. Libraries

Then, from the uniform distribution we extract 1000 samples from 0 to 3600 seconds (1 hour). And get them in a pandas dataframe where we can analyze the data.

```
x_arrival = []
y_arrival = []
sample_size = 100000
x_arrival = np.random.randint(0, 3600, sample_size)
y_arrival = np.random.randint(0, 3600, sample_size)
arrival = pd.DataFrame({'x': x_arrival, 'y': y_arrival, 'delay_time':
    [15*60]*sample_size})
arrival
```

Listing 2. Uniform distribution samples

Index	x	y	delay time
0	131	237	900
1	1880	1409	900
2	360	2350	900
3	3394	3071	900
4	853	1556	900
⋮	⋮	⋮	⋮
99995	21	2344	900
99996	3393	759	900
99997	430	1607	900
99998	68	2002	900
99999	293	1159	900

100000 rows $\times$ 3 columns

The most critical step was to make the simulation, where the for and if statement was used.

```python
meets = 0
for i in range(sample_size):
    if arrival['x'][i] > arrival['y'][i]:
        if arrival['x'][i] > arrival['y'][i] + arrival['delay_time'][i]:
            continue
        else:
            meets +=1
    if arrival['y'][i] > arrival['x'][i]:
        if arrival['y'][i] > arrival['x'][i] + arrival['delay_time'][i]:
            continue
        else:
            meets +=1
    if arrival['y'][i] == arrival['x'][i]:
        meets +=1
    else:
        continue
```

Listing 3. Simulation

Then the theoretical probability and the simulation's probability is printed.

Listing 4. Results of simulation

```python
print(meets)
print("The probability they meet is: ",meets/sample_size)
print("The theoretical probability they meet is: ",7/16)
```

43850

The probability they meet is: 0.4385

The theoretical probability they meet is: 0.4375

2 Deepseek's answers

Problem Statement

Romeo and Juliet have a date at a given time, and each will arrive at the meeting place with a delay between 0 and 1 hour, with all pairs of delays being equally likely. The first to arrive will wait for 15 minutes (0.25 hours) and will leave if the other has not yet arrived. What is the probability that they will meet?

Analytical Solution

Let X and Y be the arrival delays of Romeo and Juliet, respectively. Both X and Y are uniformly distributed over $[0,1]$. They meet if:

$$|X - Y| \leq 0.25$$

The probability can be computed geometrically by considering the unit square $[0,1]\times[0,1]$ and the region where $|X - Y| \leq 0.25$.

- **Total area of the square**: $1 \times 1 = 1$.
- **Non-meeting region**: Two right triangles where $|X - Y| > 0.25$, each with legs of length 0.75 (since $1 - 0.25 = 0.75$).
- **Area of each triangle**: $\frac{1}{2} \times 0.75 \times 0.75 = 0.28125$.
- **Total non-meeting area**: $2 \times 0.28125 = 0.5625$.
- **Meeting probability**: $1 - 0.5625 = 0.4375 = \frac{7}{16}$.

Python Implementation

We verify the result using both analytical computation and Monte Carlo simulation.

```
import numpy as np
from fractions import Fraction

def analytical_solution():
    """Calculates the probability analytically."""
    area_non_meet = 2 * (0.5 * 0.75 * 0.75)  # Two triangles
    area_meet = 1 - area_non_meet
    return area_meet

def monte_carlo_simulation(num_trials=1_000_000):
    """Estimates the probability via Monte Carlo simulation."""
    romeo_times = np.random.uniform(0, 1, num_trials)
    juliet_times = np.random.uniform(0, 1, num_trials)
    meets = np.abs(romeo_times - juliet_times) <= 0.25
    return np.mean(meets)

# Compute and display results
analytical_prob = analytical_solution()
simulated_prob = monte_carlo_simulation()
```

```
print(f"Analytical Probability: {analytical_prob:.4f} (exact fraction:
    {Fraction(7, 16)})")
print(f"Simulated Probability: {simulated_prob:.4f} (using 1,000,000
    trials)")
```

Listing 5. Analytical and Simulation Solution

Results

Running the code produces:

```
Analytical Probability: 0.4375 (exact fraction: 7/16)
Simulated Probability: 0.4373 (using 1,000,000 trials)
```

The simulation closely matches the analytical result, confirming that the probability is $\boxed{\frac{7}{16}}$.

3 Compare Results

The main differences of my code and Deepseek's code is that I use simple if statements that checks if the arrival time of both is in the intervals of delay time (15 minutes), and Deepseek uses a "Monte Carlo simulation", that uses the absolute difference between the arrival time of Romeo and Juliet and checks if it is less or equal to 0.25, then it returns the mean of the variable "meets".