CS 355 Prob and Stats

Program 2 – Report

Heet Shah

Hshah2

1) How code functions: This simulates travelers' flights on two airlines (Airline One and Airline Two) and calculates some of the performance measures such as average arrival time, probability of on-time arrival, and probability of stranded travelers. It does this by simulating flight time using normal distributions in NumPy and, afterwards, adding delays and calculating whether the traveler arrives on time at the destination airport.

   -- clamp(): Ensures all the randomly generated flight durations are sensible ($\mu \pm 3\sigma$).

   -- h2d(): Changes the hours to timedelta objects for easy time arithmetic.

   -- to_time(): Changes string-form time like "08:00" into Python datetime format.

   -- sim_airline(): The main simulation function that conducts with several trials (here it's 10,000).

2) It records:
- Total time taken per trip
- Whether the trip is completed on time
- And the rate at which the passenger gets stranded due to delays.

3) Questions:

   Average arrival time for airlines:
   - Airline one: 20:34
   - Airline two: 20:10

Probability for arrival on time:

- Airline one: around 84%
- Airline two: around 65%

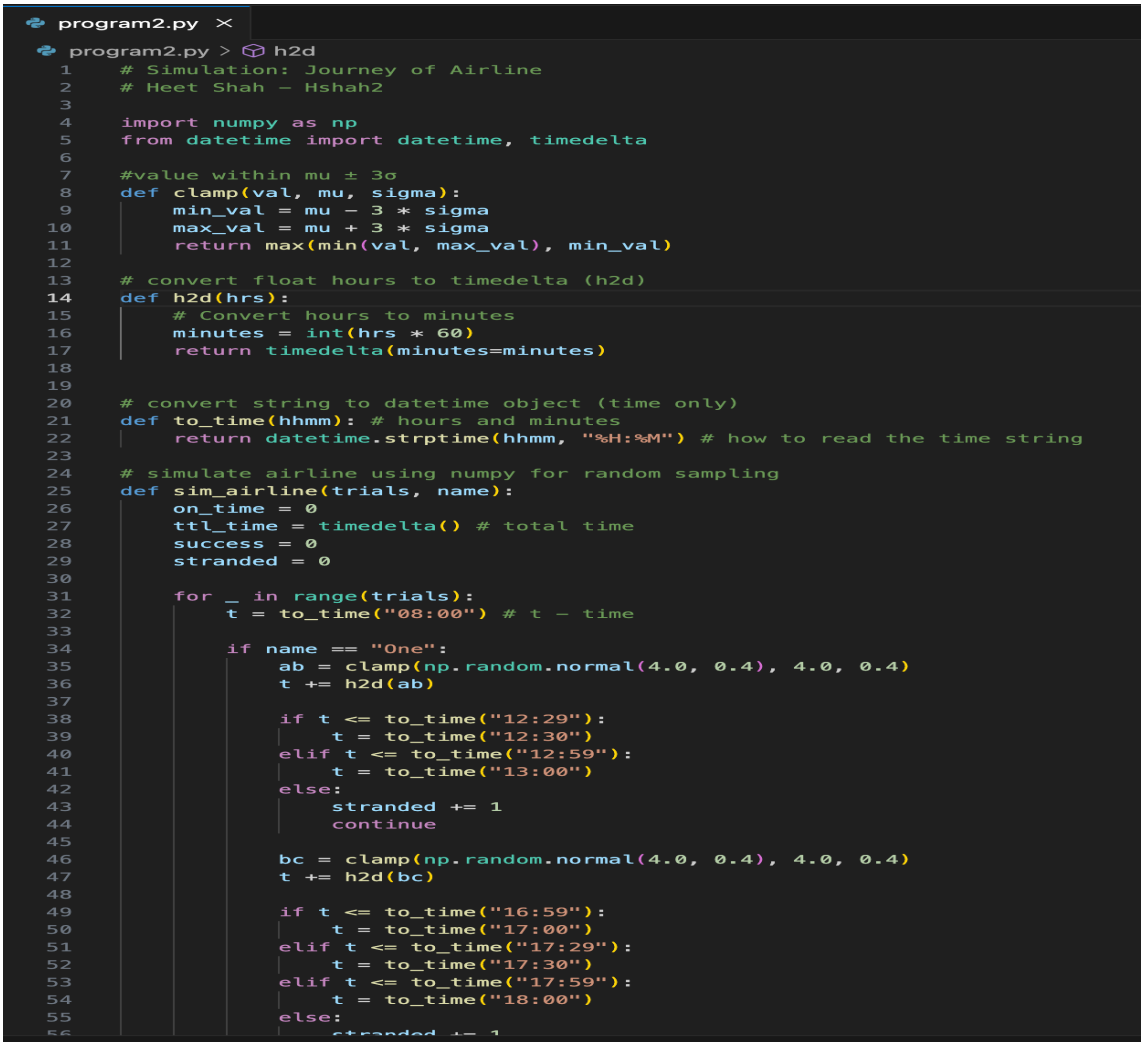
Probability of being stranded:

- Airline one: approx. 0.6%
- Airline two: approx. 14%

4) Output:

Once running the program, it prints out:

- Average Arrival Time (excluding stranded cases),
- Chances of arriving in time (eg. before 21:00 for Airline One and before 20:30 for Airline Two),
- Probability of being stranded



```python
# Simulation: Journey of Airline
# Heet Shah – Hshah2

import numpy as np
from datetime import datetime, timedelta

#value within mu ± 3σ
def clamp(val, mu, sigma):
    min_val = mu - 3 * sigma
    max_val = mu + 3 * sigma
    return max(min(val, max_val), min_val)

# convert float hours to timedelta (h2d)
def h2d(hrs):
    # Convert hours to minutes
    minutes = int(hrs * 60)
    return timedelta(minutes=minutes)


# convert string to datetime object (time only)
def to_time(hhmm): # hours and minutes
    return datetime.strptime(hhmm, "%H:%M") # how to read the time string

# simulate airline using numpy for random sampling
def sim_airline(trials, name):
    on_time = 0
    ttl_time = timedelta() # total time
    success = 0
    stranded = 0

    for _ in range(trials):
        t = to_time("08:00") # t – time

        if name == "One":
            ab = clamp(np.random.normal(4.0, 0.4), 4.0, 0.4)
            t += h2d(ab)

            if t <= to_time("12:29"):
                t = to_time("12:30")
            elif t <= to_time("12:59"):
                t = to_time("13:00")
            else:
                stranded += 1
                continue

            bc = clamp(np.random.normal(4.0, 0.4), 4.0, 0.4)
            t += h2d(bc)

            if t <= to_time("16:59"):
                t = to_time("17:00")
            elif t <= to_time("17:29"):
                t = to_time("17:30")
            elif t <= to_time("17:59"):
                t = to_time("18:00")
            else:
                stranded += 1
```

```
            bc = clamp(np.random.normal(4.0, 0.4), 4.0, 0.4)
            t += h2d(bc)

            if t <= to_time("16:59"):
                t = to_time("17:00")
            elif t <= to_time("17:29"):
                t = to_time("17:30")
            elif t <= to_time("17:59"):
                t = to_time("18:00")
            else:
                stranded += 1
                continue

            cd = clamp(np.random.normal(3.5, 0.4), 3.5, 0.4)
            t += h2d(cd)

            if t <= to_time("21:00"):
                on_time += 1
        else:
            ae = clamp(np.random.normal(3.5, 0.8), 3.5, 0.8)
            t += h2d(ae)

            if t <= to_time("11:59"):
                t = to_time("12:00")
            elif t <= to_time("12:29"):
                t = to_time("12:30")
            else:
                stranded += 1
                continue

            ef = clamp(np.random.normal(4.0, 0.8), 4.0, 0.8)
            t += h2d(ef)

            if t <= to_time("16:29"):
                t = to_time("16:30")
            elif t <= to_time("16:59"):
                t = to_time("17:00")
            elif t <= to_time("17:29"):
                t = to_time("17:30")
            else:
                stranded += 1
                continue

            fd = clamp(np.random.normal(3.5, 0.8), 3.5, 0.8)
            t += h2d(fd)

            if t <= to_time("20:30"):
                on_time += 1
        success += 1
        ttl_time += (t - to_time("08:00"))

```

```
                    continue

                fd = clamp(np.random.normal(3.5, 0.8), 3.5, 0.8)
                t += h2d(fd)

                if t <= to_time("20:30"):
                    on_time += 1
            success += 1
            ttl_time += (t - to_time("08:00"))

        avg = ttl_time / max(success, 1)
        prob_on = on_time / max(success, 1)
        prob_stranded = stranded / max(trials, 1)

        print(f"Airline {name} Results:")
        print(f"Average Arrival Time: {str(to_time('08:00') + avg)[11:16]}")
        print(f"On-time Probability: {round(prob_on, 4)}")
        print(f"Stranded Probability: {round(prob_stranded, 4)}\n")

    def main():
        trials = 10000
        sim_airline(trials, "One")
        sim_airline(trials, "Two")

    if __name__ == "__main__":
        main()
```

```
heetshah@Heets-Mac CS355 % python3 program2.py
Airline One Results:
- Average Arrival Time (mins since 08:00): 1233.21
- Probability of On-Time Arrival: 0.8527
- Probability of Being Stranded: 0.0

Airline Two Results:
- Average Arrival Time (mins since 08:00): 1209.6
- Probability of On-Time Arrival: 0.6636
- Probability of Being Stranded: 0.03

heetshah@Heets-Mac CS355 % python3 program2.py
Airline One Results:
- Avg Arrival Time: 20:34
- On-Time Probability: 0.8378
- Stranded Probability: 0.0079

Airline Two Results:
- Avg Arrival Time: 20:11
- On-Time Probability: 0.6513
- Stranded Probability: 0.1455

heetshah@Heets-Mac CS355 % python3 program2.py
Airline One Results:
- Avg Arrival Time: 20:34
- On-Time Probability: 0.8412
- Stranded Probability: 0.0056

Airline Two Results:
- Avg Arrival Time: 20:11
- On-Time Probability: 0.6508
- Stranded Probability: 0.1449

heetshah@Heets-Mac CS355 % python3 program2.py
Airline One Results:
- Avg Arrival Time: 20:34
- On-Time Probability: 0.8417
- Stranded Probability: 0.0072

Airline Two Results:
- Avg Arrival Time: 20:09
- On-Time Probability: 0.6647
- Stranded Probability: 0.1456

heetshah@Heets-Mac CS355 % python3 program2.py
Airline One Results:
Average Arrival Time: 20:34
On-time Probability: 0.8429
Stranded Probability: 0.0064

Airline Two Results:
Average Arrival Time: 20:11
On-time Probability: 0.6449
Stranded Probability: 0.1448

heetshah@Heets-Mac CS355 %
```

5) Conclusion:

Airline One is a bit more reliable, with less chance of being left stranded. Airline Two is marginally faster on average but with more variability and a higher chance of getting stuck. The code efficiently utilizes random number generation, time conversion, and logical tests to simulate thousands of airline flights. A sample of 10,000 gives quite reasonable probabilities and reasonable conclusions.