HW – Quantitative Risk Analysis

Student: Bhupinder Singh (bsingh55)

Course: SER 416 – Software Enterprise: Projects & Process

Assignment: HW – Quantitative Risk Analysis

1. Problem Summary

The project has encountered a technical challenge, and the team is considering two possible solutions (Option A and Option B).

Each option has probabilistic impacts on both additional **cost** and **schedule delay**.

The goal of this homework is to:

- Use Python to compute the expected additional cost and expected schedule delay for both options.
- Select a preferred option based on delay only and based on cost only.
- Re-evaluate the decision when a $5,000 bonus is paid if the project finishes with **no delay**.
- Use a Monte Carlo simulation to estimate the expected delay for Option A and show convergence of the estimate.

All numerical results in this document come directly from the Python script quant_risk_analysis.py (no manual calculations).

2. Given Data

Cost Impact

- **Option A**
    - 60% chance of **$6,000** additional cost

- 40% chance of **$9,000** additional cost

- **Option B**
  - 20% chance of **$4,000**
  - 25% chance of **$6,000**
  - 40% chance of **$8,000**
  - 15% chance of **$12,000**

Schedule Impact

- **Option A**
  - 40% chance of **1-week** delay
  - 60% chance of **2-week** delay
- **Option B**
  - 20% chance of **0-week** delay
  - 25% chance of a **1-week** delay
  - 40% chance of **2-week** delay
  - 15% chance of **3-week** delay

3. Expected Cost and Schedule Delay (Task 1)

All expected values below were computed in Python by multiplying each outcome by its probability and summing the results.

3.1 Expected Additional Cost

Option A

$$E[Cost_A] = 0.60\ x\ 6000 + 0.40\ x\ 9000 = 7200$$

Option B

$E[Cost_B] = 0.20 \, x \, 4000 + 0.25 \, x \, 6000 + 0.40 \, x \, 8000 + 0.15 \, x \, 12000 = 7300$

Python output:

- Expected additional cost (A) = **$7,200.00**
- Expected additional cost (B) = **$7,300.00**

So, based on expected cost alone, Option A is slightly cheaper on average.

3.2 Expected Schedule Delay

Option A

$E[Delay_A] = 0.40 \, x \, 1 + 0.60 \, x \, 2 = 1.6 \, weeks$

Option B

$E[Delay_B] = 0.20 \, x \, 0 + 0.25 \, x \, 1 + 0.40 \, x \, 2 + 0.15 \, x \, 3 = 1.5 \, weeks$

Python output:

- Expected schedule delay (A) = **1.600 weeks**
- Expected schedule delay (B) = **1.500 weeks**

So, based on expected delay alone, Option B has slightly less expected delay.

4. Choice Based on Schedule Delay Only (Task 2)

When we only care about **schedule**, we choose the option with the smaller expected delay.

From the Python script:

- Expected delay (A): **1.600 weeks**
- Expected delay (B): **1.500 weeks**

Because 1.5 weeks < 1.6 weeks, the preferred choice **based purely on delay** is:

Option B – it has the lower expected schedule delay.

5. Choice Based on Cost Only (Task 3)

When we only care about **cost**, we choose the option with the smaller expected additional cost.

From the Python script:

- Expected cost (A): **$7,200.00**
- Expected cost (B): **$7,300.00**

Because $7,200 < $7,300, the preferred choice **based purely on cost** is:

Option A – it has the lower expected additional cost.

6. Choice with $5,000 Bonus for No Delay (Task 4)

Now there is a **$5,000 bonus** if the project finishes **without delay**.

Only Option B has a non-zero probability of zero delay:

- Probability of **no delay (A)** = 0.0
- Probability of **no delay (B)** = 0.20

Expected bonus for Option B:

$E[Bonus_B] = 0.20\ x\ 5000 = 1000$

We adjust the expected cost:

- Net expected cost (A) with bonus = **$7,200.00** (no chance of bonus)
- Net expected cost (B) with bonus

  = 7300 - 1000 = 6300

From the Python output:

- Expected net cost (A) with bonus: **$7,200.00**
- Expected net cost (B) with bonus: **$6,300.00**

Because $6,300 < $7,200, when the bonus is considered, the best choice is:

Option B – it has the lowest expected net cost once we include the bonus for on-time delivery.

7. Monte Carlo Simulation for Option A Delay (Task 5)

To verify the analytical expected delay for Option A, the Python script performs a **Monte Carlo simulation**:

- At each iteration, it randomly samples a delay for Option A using the probabilities
  - 40% → 1 week
  - 60% → 2 weeks
- It keeps a running average of the simulated delays.
- We run the simulation for **1,000 iterations** (more than the required 100).

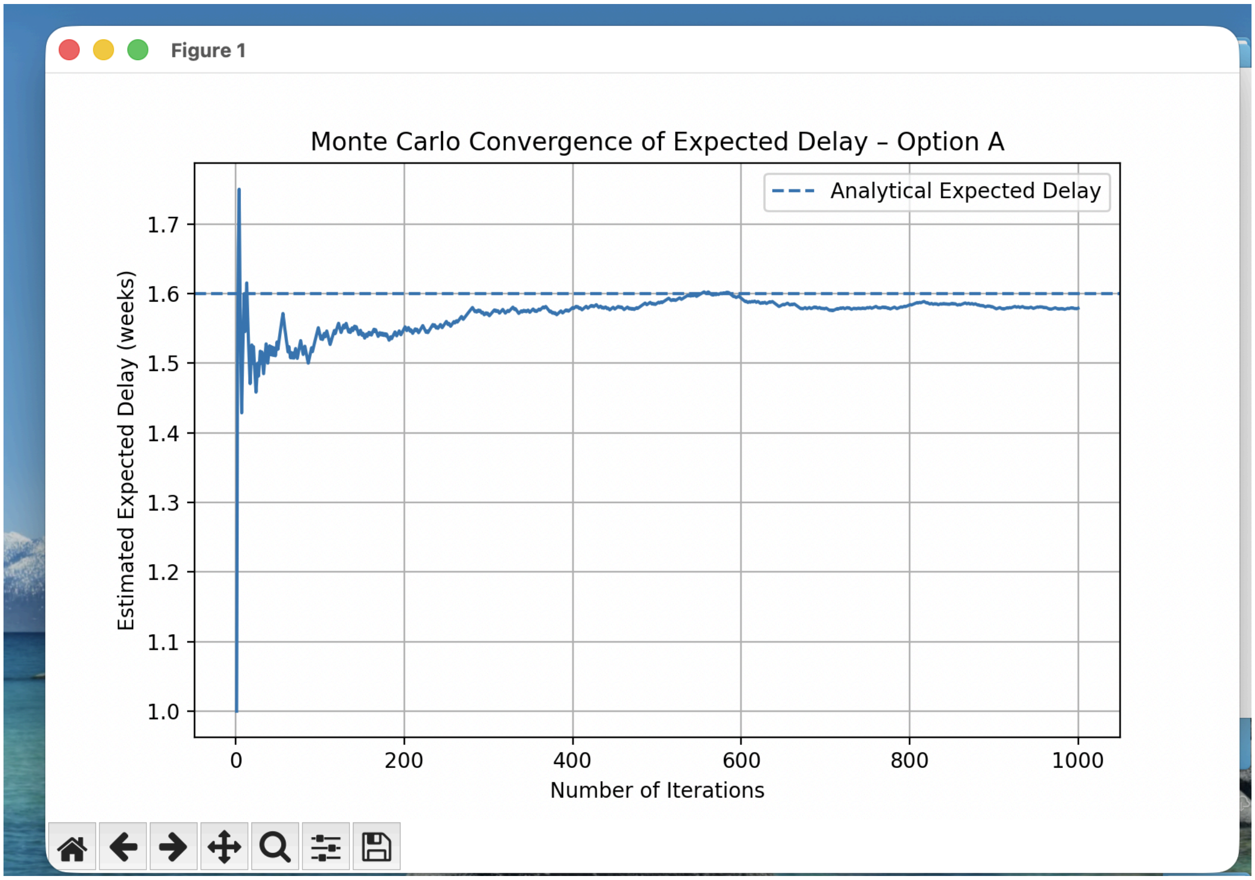
From one run of the script, the terminal output showed:

- Final estimated expected delay from simulation: **≈ 1.579 weeks**
- Analytical expected delay from Task 1: **1.600 weeks**

These two values are very close, which confirms the simulation is converging to the analytical expected delay.

Convergence Chart

The script also generates a plot (quant_risk_mc_convergence.png) that shows how the running average of the simulated delay approaches 1.6 weeks as the number of iterations increases.



8. Summary of Recommendations

- **Expected delay only:**
  - Option B is better (1.5 weeks vs 1.6 weeks).
- **Expected additional cost only:**
  - Option A is better ($7,200 vs $7,300).
- **Including a $5,000 bonus for on-time completion:**
  - Only Option B can finish with no delay, so it gets an expected bonus of $1,000.
  - Net expected costs become:
    - Option A: $7,200

        - Option B: $6,300
    - Option B is clearly preferred once the bonus is considered.
- **Monte Carlo validation:**
    - The simulation for Option A's delay converges near 1.6 weeks, matching the analytical expected value and confirming the correctness of the calculations.