HW - PERT and Monte Carlo Simulation

Start Assignment

- Due Dec 3 by 11:59pm
- Points 33
- Submitting a file upload
- File Types zip

HW - PERT and Monte Carlo Simulation

NOTE: All coding problems must be done via code - no hard coding of any type is allowed for credit. Any crash or execution errors that are not gracefully handled will result in a mandatory 20% reduction. Also, all output requested should be clear with labels and easy to find, or a penalty of 10% will occur.

In this assignment, you will analyze the uncertainty in project scheduling using **Program Evaluation and Review Technique (PERT)** and **Monte Carlo simulation**. You are provided with an Excel file, `Critical Path Data.csv`, which contains the task data for the **critical path** of a project. For each task, you will see three estimates: **Optimistic (O)**, **Most Likely (ML)**, and **Pessimistic (P)** durations. Your goal is to work entirely in Python, using this data to compute PERT-based task durations, simulate uncertainty using triangular distributions, and analyze the distribution of total project durations.

To begin, write a Python script that reads and validates the Excel file. Each row must contain valid O, ML, and P numerical entries. If any invalid values are found, they must be handled gracefully. Once validated, store the cleaned data to a CSV file named `critical_path_clean.csv`.

Next, calculate the **PERT duration** for each task using the formula `(O + 4×ML + P) / 6`, and record the range for each task as the minimum (Optimistic) and maximum (Pessimistic) durations. These values should be stored alongside the original inputs. After completing these calculations, compute the total project duration for each of the following estimates: **Optimistic**, **Most Likely**, **Pessimistic**, and **PERT** (i.e., summing across all tasks). Write all of this data into a file named `pert_summary.csv`.

Once your summary file is generated, simulate the project 1,000 times using a **Monte Carlo approach**. For each task, generate 1,000 random values based on a **triangular distribution**, where the minimum is the optimistic value, the mode is the most likely value, and the maximum is the pessimistic value. Calculate the total project duration for each simulation iteration by summing one sample value from each task. Store the full matrix of generated values and the 1,000 total durations in a file named `monte_carlo_raw.csv`.

To verify your simulation logic, create a **histogram** of the sampled durations for **Task 1**. This plot should reflect the triangular shape expected from the distribution. Save the histogram as `task1_histogram.png`. This visual check ensures your random number generation logic behaves as expected.

Using the 1,000 simulated total project durations, compute a **confidence table** that reports the project duration at each percentile from **60.0% to 99.9%**, in increments of 0.1%. Save this percentile-duration table to a CSV file called `confidence_curve.csv`. You should also generate and save a line plot of this curve as `confidence_plot.png`.

Finally, analyze your results and provide answers to the following management questions:

- What is the minimum project duration that gives **70%**, **80%**, and **90%** confidence of completion?
  Record these values in a clearly written file named `confidence_answers.txt`, or alternatively, print them cleanly in your terminal output.

Supporting Files

You must download and use the csv file **Critical Path Data.csv (https://canvas.asu.edu/courses/234472/files/112833205?wrap=1) (https://canvas.asu.edu/courses/234472/files/112833205/download?download_frd=1)** , which contains the base task data. Here are some references you can use to understand triangular distribution generation using Python :-

- **Using Scipy (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.triang.html)**
- **Using Numpy (https://numpy.org/doc/2.1/reference/random/generated/numpy.random.Generator.triangular.html#numpy.random.Generator.triangular)**

Submission Instructions

Submit a single `.zip` file containing all of the following:

- Your well-commented Python script named `pert_mc_simulation.py`, which performs all parts of the assignment.
- The cleaned input file `critical_path_clean.csv`.
- The PERT and range summary output file `pert_summary.csv`.
- The raw Monte Carlo simulation data file `monte_carlo_raw.csv`.
- The confidence percentile table `confidence_curve.csv`.
- Your answers to the managerial insight questions in `confidence_answers.txt`.
- Two visual files: the Task 1 histogram (`task1_histogram.png`) and the confidence curve plot (`confidence_plot.png`).

All filenames must match exactly. Ensure that your code is robust, well-documented, and outputs are clearly labeled to avoid penalties. Any script that crashes or contains hard-coded logic will incur deductions.

HW Monte Carlo Simulation

Criteria	Ratings			Pts
Input File Processing and Validation Input Excel read and validated, cleaned data saved to critical_path_clean.csv.	**3 pts** **Full Marks** Correct parsing, validation, and CSV generation.	**0 pts** **No Marks** Has errors.		3 pts
PERT Duration Estimate Calculation Uses (O + 4×ML + P)/6 correctly for each task.	**2 pts** **Value correct**	**0 pts** **No Marks**		2 pts
Min and Max Duration Calculation Calculates correct optimistic and pessimistic sum of project.	**2 pts** **Full Marks** Both values accurate.	**0 pts** **No Marks**		2 pts
Variable Task Count Handling Works for any number of tasks in input.	**2 pts** **Full Marks** No hardcoded task limits.	**0 pts** **No Marks**		2 pts
Monte Carlo Iterations (1000 runs) Runs 1000+ simulations.	**10 pts** **Full Marks** Exactly 1000 with correct random logic.	**5 pts** **Partial Marks** ≤500 iterations or inconsistent generation.	**0 pts** **No Marks**	10 pts
Task 1 Histogram task1_histogram.png included and clearly shows triangular shape.	**2 pts** **Full Marks** Graph generated, labeled.	**0 pts** **No Marks**		2 pts
Total Duration per Simulation Total project duration calculated for each simulation.	**2 pts** **Full Marks** monte_carlo_raw.csv contains task-wise and total simulation values.	**0 pts** **No Marks**		2 pts
Confidence Interval table Calculate values for the percentile values from 60% to 99% in increments of 0.1 to get good resolution.	**5 pts** **Full Marks** Percentiles from 60% to 99.9% at 0.1% resolution in confidence_curve.csv.	**3 pts** **Partial** Incomplete range or incorrect percentile logic.	**0 pts** **No Marks**	5 pts
Confidence Plot Line plot of percentiles saved as confidence_plot.png.	**2 pts** **Full Marks** Labeled and clear.	**0 pts** **No Marks**		2 pts
Confidence Analysis (70/80/90%) Clear and accurate 3 answers in confidence_answers.txt or terminal output.	**3 pts** **3/3 correct** All 3 Correct	**0 pts** **No Marks**		3 pts
				Total Points: 33