Assignment 1: The TwoPlus Earbuds Problem

Context:

You are the product manager of TwoPlus Earbuds X, a Bluetooth headset for TwoPlus 7T smart phone. Your CEO has announced the release date of the earbuds as 1st January 2023. The CEO calls you privately and shares that this product is to be offered only for one month until the company has a more advanced version, the TwoPlus Earbuds Z, ready to be shipped to stores. You must decide on the number of TwoPlus Earbuds X to manufacture.

Your Business Analyst forecasts that the demand for TwoPlus Earbuds X will be normally distributed with mean 150 and standard deviation 20. The total cost of manufacturing and logistics for each unit is $28.50 and you sell the earbuds at a retail price of $150.00. Any earbuds not sold within one month need to be disposed in an environmentally sustainable way which will cost the company $8.50 per unit.

You face the following dilemma –

- If you manufacture too many, you may not sell all of them and make a loss of $37 (sum of manufacturing, logistics and disposing costs, $28.50 + $8.50) for each earbud left unsold
- If you manufacture too less, you may not have enough for all customers and let go of a profit of $121.5 (difference of retail price and cost $150 – $28.50) for each customer leaving the store without the earbuds

You want to know what the optimal number of TwoPlus Earbuds Xs is to manufacture.

In this assignment, you will code a simulation in Python to resolve this dilemma.

In this assignment, you will use the Numpy package, a powerful tool for generating and manipulating arrays. It will help streamline your simulations and prepare you for the second half of the course, where we will dive deeper into Numpy. After installing the package, import it using the following:

import numpy as np

Below is a recommendation for simulation (teams are free take their own approaches) –

Write a function with input parameters as unit price, retail price, cost of disposing and the number of earbuds manufactured.

Next, within the function –

1. create a list of length 1000 and randomly generate demand instances from N(150,20) at each index.

`np.random.normal(mean, std)`: This function generates random numbers from a normal distribution with the specified mean and standard deviation. In your simulation, it is used to simulate the demand for TwoPlus Earbuds X, which follows a normal distribution with a mean of 150 and a standard deviation of 20. Example: `np.random.normal(150, 20)` generates random values for demand with an average of 150 and variability of 20.

2. compute the profit for a given number of earbuds manufactured at each index and save it in another list

3. Compute mean and standard deviation of the profit

```
np.mean(): This function computes the average value of a list of numbers. In the simulation, it calculates the average profit over 1000 simulations. Example: np.mean(profits) computes the mean of the list of profits obtained from multiple simulations.
```

4. Repeat steps 1 – 3 by inputting several possible values of the number of earbuds manufactured and identify the best value. A natural place to start is the mean.

You are asked the following questions –

1) Question 1 (24 Pts)**:**

a) How many earbuds will you manufacture and why? Please restrict your answer to at most 50 words.

Hint: Determine the number of earbuds that maximizes profit by iterating through all possible production quantities, from 0 to 230.[1]

2) Question 2 (6 Pts)**:**

Please upload a single .pdf file that with three sections -

(i) Your python code appropriately commented

(ii) additional instructions for running your code, if necessary.

(iii) snapshots of the output of your implementation.

3) **Bonus Question** (10 Pts)**:** Your Policy analysts tell you there is a 50% chance of government imposing additional taxes on disposing electronic waste which will increase the cost of disposing unsold earbuds to $17.

a) How many earbuds will you manufacture and why? Please restrict your answer to at most 50 words.

Hint: To simulate the uncertainty of tax, you can "flip a coin" in each of your 1000 simulation runs, so approximately 500 instances will include a tax while the other 500 will not. Use `additional_tax = np.random.choice([True, False], p=[0.5, 0.5])` to simulate this coin flip, which gives a 50% chance of either outcome. You can then use this result in an if statement to determine the effect of the additional tax when it is imposed.

[1] 230 = 150 + 4 * (20), meaning 230 is four standard deviations above the mean of the normally distributed demand. This implies an almost 0 probability that demand will exceed 230, making it the upper bound of potential demand. Therefore, your supply should always be less than this upper bound

np.random.choice([True, False], p=[0.5, 0.5]): This function selects random elements from a specified list based on the given probabilities. In your code, it is used to simulate whether additional taxes will be imposed on unsold earbuds, with a 50% chance for each outcome.
Example: np.random.choice([True, False], p=[0.5, 0.5]) randomly decides whether additional taxes apply.