Lab 1 - Basic Probability Questions

Note - Please read the instructions mentioned in the questions carefully. We have provided boilerplate code for each question. Please ensure that you make changes in the areas marked with TODO.

Question 1 - Random number generator

Complete the function **generate_uniform** to generate n numbers sampled from a uniform distribution on the interval [0,1] and save the generated numbers to a file named “uniform.txt”. Each number must be on a new line.
You will need to use the function to generate random numbers from the **np.random** module. You will also need to set the seed in numpy before starting the random number generation.

Function signature - **def generate_uniform(seed: int, num_samples: int)**

seed - The seed for random number generation that needs to be set
num_samples - The number of samples you need to generate

Question 2 - Inverse transform sampling

Complete the function **inv_transform** which generates samples from a given probability distribution using uniform random samples from [0,1].
The function takes a file_path, target distribution name and some extra keyword arguments (which store the target distribution parameters) as input.

The file corresponding to the file_path will contain 100 numbers sampled from a uniform distribution on [0,1]. Each number will be separated by a newline character.

The second argument will be the target distribution name which will be one out of - “categorical”, “exponential” and “cauchy”.

The kwargs (parameters) will depend on the second argument

Function signature - **def inv_transform(file_name: str, distribution: str, **kwargs)**

For “categorical”, the kwargs will be of the form -

{
“values” : <list-of-numbers>,
“probs” : <list-of-probability-values-associated-with-the-numbers>
}

For “exponential”, the kwargs will be of the form -

{
“lambda” : <float>
}

For “cauchy”, the kwargs will be of the form -

{
“peak_x” : <float>,
“gamma” : <float>
}

Question 3 - Find the best distribution!

Complete the function **find_best_distributions** to find the distributions (from the options given below) which are most likely to have generated the given data.
The distributions are -

0. Gaussian distribution with $\mu = 0$, $\sigma = 1$
1. Gaussian distribution with $\mu = 0$, $\sigma = 0.5$
2. Gaussian distribution with $\mu = 1$, $\sigma = 1$

0. Uniform distribution on [0, 1]
1. Uniform distribution on [0, 2]
2. Uniform distribution on [-1, 1]

0. Exponential distribution with $\lambda = 0.5$
1. Exponential distribution with $\lambda = 1$
2. Exponential distribution with $\lambda = 2$

The function takes a list of numbers as the only argument. It must return the **three** indices corresponding to the best distribution from each type which is the most likely to have generated the data.

Function signature - **def find_best_distributions(samples: list)**

Hint - Be wary of floating-point underflow

Question 4 - Confidence intervals

Complete the function **marks_confidence_intervals.** The marks of students enrolled in CS337 follows an unknown distribution with unknown mean μ and variance equal to 5. We have a list of marks scored by 50 students in the class. Given n values of confidence interval $\in$, $\{\in_i\}$, you need to first find the sample mean û and also report the probabilities δ_i such that

$$P(|\hat{u} - \mu| >= \in_i) <= \delta_i$$

Function signature - **def marks_confidence_intervals(samples: list, variance: float, epslions: list)**

The function returns a tuple containing the sample mean and the n probabilities δ_i.

Submission instructions

Complete the functions in **assignment.py.** Keep the file in a folder named **<ROLL_NUMBER>_L1** and compress it to a tar file named **<ROLL_NUMBER>_L1.tar.gz** using the command

tar -zcvf <ROLL_NUMBER>_L1.tar.gz <ROLL_NUMBER>_L1

Submit the tar file on Moodle.

The directory structure should be -

<ROLL_NUMBER>_L1
| - - - - assignment.py

Replace ROLL_NUMBER with your own roll number. If your Roll number has alphabets, they should be in “small” letters.