MA 202 ASSIGNMENT 2
SEMESTER II, AY 2021–2022

Problem 1 Mini-project 1: Writing a MATLAB/Python code to compute the PDF/PMF of a Random Variable

Write a MATLAB/Python function titled `ComputePDF(Arg1,Arg2,...)`, which does the following:

- **Arguments:** The function should take an array of numerical values of size $m \times n$ representing a variable as the first argument (`Arg1`). These values might be randomly generated, may represent data from an experiment, or variations in some physical quantity in any process. For instance, this may be the velocity magnitude in a fluid flow.

  You may also pass other arguments - whether to do this is for you to decide. For instance, in the classroom demo, the function '`FindProbabilityDistribution(Arg1,Arg2)`' was shown, which has two arguments. The first one is the numerical values, representing the data or the variable and the second one was the relative interval, which is used to find the probability density. You may choose to do the same or, you may choose a different path - this is for your to decide.
- **What should the function do?** The function should decide on its own whether the sample/realization values passed through `Arg1` represent a discrete or a continuous random variable. Accordingly, the function should then compute the PDF or the PMF of this random variable.
- **Output:** The function should output the following:

  (a) **The sample space** (`x`, say): This should be a 1D array encompassing all the possible values that the random variable may take. If the RV is discrete, this array should contain all the discrete values that the RV may assume, based on the data provided through `Arg1`. If the RV is continuous, the size of the array would then depend on the choice of the interval being used to compute the PDF.

  (b) **The PDF or the PMF** (`p`, say): Should be an 1D array of the same size as that of `x` and should contain the magnitude of the PDF or the PMF for each value in the sample space as represented by `x`.

  (c) **A flag variable showing whether the RV is discrete or not** (`isDiscrete`, say): If the RV is discrete the the function should output `isDiscrete = 1`; otherwise it should yield `isDiscrete = 0`.

If you are using MATLAB, then the function may be called like this:

```
[x,p,isDiscrete] = ComputePDF(Arg1,Arg2,...).
```

You are allowed to use necessary in-built functions of MATLAB/Python, except for those which can explicitly find PDFs/PMFs (otherwise the purpose of this entire exercise will be beaten).

Problem 2 Mini Project-2: Verify the function you have written in Problem 1

Now write another MATLAB script to ensure that the `ComputePDF` function that you have written in Problem 1 actually gives correct output. The best way to do this is to ensure that it gives correct outputs for random variables whose PDF/PMFs are known. MATLAB has many types of random number generators. For example, **randn, rand, randi,** etc. have already been shown in the class. You can also generate exponential RVs, Poisson RVs, Binomial RVs, Chi Square RVs and many more using MATLAB. Their ideal PDFs/PMFs are also available in MATLAB and you may use those in your script. Search Random numbers in the help of MATLAB and you will find a comprehensive summary. The present script should do the following:

- Generate multiple realizations of an RV using the various random number generators available in MATLAB.
- Use `ComputePDF` to generate the PDF/PMF of the RV. Plot the PDF - i.e., plot p vs `x`. If the RV is continuous, you may directly use the MATLAB function `plot`. For plotting the PMFs of discrete RVs, you may use the MATLAB function `stem`.
- Now, also plot the ideal PDF/PMF of the RV, for which realizations have been generated, in the same figure. For instance, if you are generating a normally distributed rv $\sim N(\mu, \sigma^2)$, then the ideal PDF should have the following expression:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

This will directly indicate the accuracy of the function you have written to compute PDFs.

Your script should carry out the above comparison for at least one discrete rv and at least two continuous rvs. Accordingly you may choose any of the random number generators available in MATLAB. There should be only one script, structured as follows:

```
% Start of MATLAB Script
% Section-1:  Test for RV with <Specify Distribution> Distribution -----------
--- Code ---
--- Code ---
etc.
% Section-2:  Test for RV with <Specify Distribution> Distribution -----------
--- Code ---
--- Code ---
etc.
% Section-3:  Test for RV with <Specify Distribution> Distribution -----------
--- Code ---
--- Code ---
etc.
% End of MATLAB Script
```

The above script will be checked (on our end) one section at a time. That is, if say, '`Section 1`' is being tested, the other two 'Section's will be commented out. We may also test using other arbitrary random variables.

The script should be named as follows: `TestPDF<YOUR ROLL NUMBER>.m` in MATLAB. If you are using Python, it should be named: `TestPDF<YOUR ROLL NUMBER>.py`.

Submission: You should submit a single 'zip' file containing both the MATLAB/Python programs using the link provided in google classroom. Additionally, it should also contain a 1-2 page write up on how you does the function `ComputePDF(...)` work and how to use it. In this report, you should also include the possible limitations of this function and the potential scenarios where it might fail. Think of it as a manual for the future users.