Assignment #3: Probability Functions, Moments, and Monte Carlo Methods (total 10 points), due by 11:59 pm Friday, 29 October 2024

Exercise 1: Probability density function features. Given the triangular distribution defined as follows:

$$p_T(x \mid \alpha, \beta) = \begin{cases} \frac{2(x-\alpha)}{(\beta-\alpha)^2} & \alpha \leq x \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

(a) Calculate its first, second, third, and fourth moments. **(0.5 points per correct moment)**

(b) Determine the median of p_T. **(1 point)**

(c) Write a script to plot $p_T(x|\alpha, \beta)$ as a function of x and include vertical lines that mark the mean, the mode, and the median of this probability density function. You can choose the values of α and β. **(1 point)**

You can write your answers for this question using any text editor of your preference, but, please, set the document layout with the following features:

- Page orientation: portrait;
- Page size: 8.5" × 11" (letter);
- Margins: 2.54 cm (top, bottom, left, right);
- Line spacing: between 1.0 and 1.15 pt;
- No double-column format. Write your answers as a 1-column text;
- Paragraph alignment can be 'align left' or 'justify';
- Font type and size: Times New Roman, 12 pt, non-italic, non-bold;
- Use *italic*/**bold**/<u>underline</u> only to emphasize brief pieces of text if needed.
- ONLY `pdf` FILES ARE ACCEPTED. Do not submit editable files such as `.docx`.

Alternatively, you can write your answers on paper and scan your solutions. However, **make sure your answers can be read clearly and the handwriting is legible.** The scanned file to be submitted needs to be in `pdf` format.

*****

Exercise 2: Monte Carlo Integration

(a) Perform a numerical calculation of the number π using the traditional Monte Carlo method of integration in which we can approximate an integral I as

$$I \simeq \frac{(b-a)}{N} \times \sum_{i=1}^{N} g(x_i) \tag{1}$$

with $[a, b]$ defining the interval of a uniform distribution and g is a function outlining a circle of radius 1 (or part of it). The equation above can be implemented by drawing N random uniform samples x_i, then calculating the values $g(x_i)$ and evaluating the sum. Determine how many samples are sufficient to achieve an average precision of 0.1%. **(1 point)**

(b) Perform a numerical calculation of the number π using the hit-or-miss Monte Carlo method. Determine how many samples are sufficient to achieve an average precision of 0.1%. **(1 point)**

*****

Exercise 3: Markov Chain Monte Carlo

Consider the simplified Hubble data in the file `hubbleHumasonData.dat` containing four columns and just a few rows of data. The first column carries the name of the nebula, the second column carries the mean velocity (v) in km/s, the third column carries the number of velocities collected, and the fourth column carries the apparent magnitude (m), related to the distance via a logarithmic relationship already discussed in class. For this exercise, consider that m is in arbitrary units. This data can be modelled with the following linear relationship

$$\log(v) = a + b\,m \tag{2}$$

with a and b being the intercept and the slope parameter, respectively.

(a) Construct a Monte Carlo Markov Chain for the fit to the linear model above with 10,000 iterations. Use uniform distributions for the prior/proposal distributions of the two model parameters a and b, the latter with widths of 0.2 and 0.04, respectively, for a and b in the neighbourhood of the current value. Start the chain at values $a = 0.2$, and $b = 0.9$. **(2 points)**

(b) After completion of the chain, plot the sample distribution of the two model parameters and analyze your results. **(2 points)**

*****

Submission Information

IMPORTANT: POINTS WILL BE DEDUCTED IF THESE SUBMISSIONS PROCEDURES ARE NOT FOLLOWED.

Name the `pdf` file you generated for question 1 as `question1_assignment_3.pdf` (if you have handwritten solutions to present), the code for question 1 as `code1_assignment_3.ipynb`, the code for question 2 as `code2_assignment_3.ipynb`, and the code for question 3 as `code3_assignment_3.ipynb`. **The solution code for each exercise needs to be placed in different files as described. Do not combine codes of different exercises on the same Jupyter Notebook file.** Note that only Jupyter Notebook files are accepted for the coding part. You will upload the `pdf` file of your written answer and your code to the Gradescope platform. **It is important that you log in to Gradescope.ca and not Gradescope.com.** Your code needs to be documented, i.e., introduce comment lines to explain their main procedures. Pure lines of code without explanatory comments will have reduced marks. You can upload multiple files to the Gradescope platform and you can resubmit your work until the due date. Upload all files at once. We will test-run your submitted codes for syntax errors and check if they generate the requested outcomes. We will also check if the results/figures presented in your assignment are 'paper-like' quality and that the quantitative predictions are scientifically/mathematically founded.

ONLY `pdf` (for the written reports), `jpg/jpeg` (for figures), and `.ipynb`, (for codes) FILES ARE ACCEPTED. This means we are only accepting Python Notebooks.

IMPORTANT: In your codes include instructions on how to run it and, if applicable, include testing values for the initial conditions or settings that you attempted so we can reproduce the results. We will test other initial conditions and parameter variations to evaluate the robustness of your code. But we need a place to start. Include also information about the version of your coding platform. If this information is not included, points will be deducted.

IMPORTANT: Your codes need to output all required numbers of the exercises and they should be printed clearly. The printed values should be properly identified with text saying what is being printed. We can only assess and consider marks for numerical answers printed by your code (we cannot give marks to what we cannot see). We cannot debug your code and/or search the answers inside your code for you. This is a graded component of the course, therefore, we can ONLY evaluate what is uploaded and see printed on the screen. We will test other parameter variations and settings in your code, but we will not alter its structure and logistics. Interpret your code as any other graded component as 'in paper', meaning what you submitted is your final answer and what is not provided (or printed) in terms of answers, we cannot consider for marks.

IMPORTANT: (Follow-up from the item above) If an exercise asks you to write a code to compute or calculate a value, we will need to see the calculated output value and not simply a number manually printed using the `print` command. For instance, if an exercise asks you to write a code to obtain the velocity of a particle, we need to see the output velocity of the algorithm you implemented and not simply `print(2.34, 'km/s')`.

*****