E2-243
Programming Exercise - 2

Course Instructor: Prof. R. Vittal Rao

Instructions:

- **Attempt the following programming exercises using MATLAB.**
- **Do not submit your code, output files, etc.**
- **Please read all the questions carefully. All the questions are self explanatory.**

---

1 Coin-Tossing Experiment and the Law of Large Numbers

1. **Write a MATLAB function FlipCoin(p) that emulates the flipping of a coin with probability of 'H' = p as follows:**

   i. Use the MATLAB function rand() to generate a random number distributed uniformly in the interval (0,1).

   ii. return 'H' (or 1) if the generated number is less than p, and return 'T' (or 0) otherwise.

2. **Write a MATLAB function Flips(n) that emulates flipping of this coin n times. You will make n calls to your FlipCoin(p) function.**

3. **Write a MATLAB function EmpProb(face, n) (where face is either 'H' (or 1), or 'T' (or 0)) that returns the empirical probability of the coin turning up Heads or Tails when it is tossed n times. You will implement this by calling Flips(n) and noting the number of 'H' (or 1) and 'T' (or 0) outcomes and returning the needed probability as a fraction of n tosses.**

4. **Call your function EmpProb(face, n) with face = 'H' (or 1) n times for $n = 10, 20, 30, \ldots\ldots., 10000$. You should be able to observe that as the value of n increases, the empirical probability computed using EmpProb('H', n) converges to the true probability p(Law of Large numbers).**

2 Expectation, Variance and Standard Deviation

Let the following 30 integers represent the score of 30 students in an exam in which the minimum possible marks was 0 and the maximum possible marks was 10.

6, 8, 2, 7, 8, 1, 8, 5, 4, 9, 2, 10, 6, 7, 8, 4, 6, 2, 10, 8, 9, 8, 10, 8, 2, 7, 7, 8, 4,10.

1. **Store these numbers in the array Scores.**
2. **Write MATLAB functions to compute the average, variance and standard deviation of these numbers.**
3. **Write a MATLAB function Freq(array, x) that returns the count of data elements in array having value x. Call this function with parameters Scores, x ($x \in 0, 1, \ldots, 10$). That is, Freq(Scores,x) returns the frequency of the score x in the array Scores.**
4. **Write a MATLAB function NormFreq(array,x) that returns the normalized frequency of the data element x in the array array (where normalization is done as follows: $\sum_{x=0}^{10}$ NormFreq(array,x) = 1). You can think of NormFreq(Scores,x) (for $x \in 0, 1, \ldots, 10$) as the PMF of the integer random variable X that represents the scores of the students. In other words, $p_X(x)$ = NormFreq(Scores,x). Plot this normalized frequency.**
5. **Calculate the expected value, variance and standard deviation of the random variable X (defined in (4) above) using the standard formulae and compare with the average, variance and standard deviation you computed in (2) above. Also, verify your result with the output from the MATLAB built-in functions mean(), var(), std().**

3 Conditional Probability

With reference to the test score data in Problem 2, consider only the test scores with values greater than or equal to 5. Write a MATLAB routine CondNormFreq(x) where $x \in 5, 6, \ldots, 10$, and returns the normalized frequency of the score x as $\frac{\textbf{Freq}(x)}{\sum_{x=5}^{10} \textbf{Freq}(x)}$. Define E to be event where the test score is greater than or equal to 5. You can think of CondNormFreq(x) (for $x \in 5, 6, \ldots, 10$) as the conditional probability of the event $X = x$ (where the random variable X is as defined in Problem 1(4)) given that the event E has occurred.

4 Generating Binomially Distributed Random Numbers

Consider the experiment of flipping a biased coin with p = 0.7, for n = 50 times. You can use the Flips(n) to do this. Count the number of 'H' (or 1) in this experiment and add this number to an array HeadCount. Repeat this experiment 104 times and append the count of 'H' (or 1) from each experiment into HeadCount. As a result, HeadCount will become an array of size 104. Use the function NormFreq(HeadCount,x) for $x \in 0, 1, ..50$in a loop, to calculate the distribution (PMF) of the data. Compare this with the PMF of Binomial distribution with parameters $p = 0.7$ and $n = 50$, generated using the in-built MATLAB function binopdf().

5 Generating Geometrically Distributed Random Numbers

Consider the experiment of flipping a biased coin with $p = 0.7$. The experiment is to keep on tossing the coin till the first 'H' (or 1) appears, and counting the number of tosses it took for the first 'H' (or 1) to appear (including the last toss that resulted in 'H' (or 1)). Store this count in an array FirstHead. Repeat this experiment 104 times. Follow the same steps used in Problem 4 to obtain the normalized frequency for the data stored in FirstHead. Compare this with the PMF of Geometric distribution with parameter $p = 0.7$, generated using the in-built MATLAB function geopdf().

6 Central limit theorem

1. **Write a MATLAB function FlipCoin(p) that emulates the flipping of a coin with probability of 'H' = p as follows:**
   (a) **Use the MATLAB function rand() to generate a random number distributed uniformly in the interval (0,1)**
   (b) **Return 'H' (or 1) if the generated number is less than p, and return 'T' (or 0) otherwise.**
2. **Write a MATLAB function Flips(n)that emulates flipping of this coin 'n' times. You will make 'n' calls to your FlipCoin(p)function. The Flips(n) should return the number of 'H' from 'n' trials**
3. **Repeat the call to Flips(n), M number of times. Let us consider every call as an iteration of the experiment. The array count$_n$ would record the number of 'H' from each iteration.**

4. **Plot the distribution for count$_n$ for M=100, 1000, 10000 and n=50.**
5. **Calculate the mean and standard deviation for count$_n$ in each case.**
6. **Plot a normal distribution with mean** $= np$ **and sigma** $= \sqrt{np(1-p)}$ **using the inbuilt function in MATLAB. Compare the distribution in (4) to the distribution in (6).**