**MEMO Number** CMPE320-S22-0101

**DATE:** 12 January 2021

**TO: CMPE320 Students**

**FROM:** EFC LaBerge

**SUBJECT: Histograms, PDFs and PMFs**

# Introduction

This project will explore how a histogram approaches a probability density function (pdf or PDF) or a probability mass function (pmf or PMF) for a random variable and illustrate how to use the pmf/pdf to compute probabilities.

Warning #1: Thinking is required!

Warning #2: Follow directions!

Warning #3: It isn't on the web, so don't bother looking. You may, however, look up background material, such as the definition of terms, etc. All of the terms are well-defined in your textbook and the lectures.

Warning #4: ***Do NOT let this wait until the last minute. Debugging the multitude of plots will take time. There are four other projects coming, so I can’t give any extensions.***

This project involves concepts spread across the various lectures in Module 1, so you might not be able to do the entire thing when assigned. It is, however, perfectly acceptable to do what you can and then go back and do more as the course content expands.

Remember, there are no exams in S22 CMPE320, so I’m looking for you to develop and *explain* concepts we have developed in class. Good technical writing is expected. So is independent thinking and explanation of the concepts.

For this project, I have provided a detailed skeleton of MATLAB code, which is posted along with the assignment. I have also provided the MATLAB function randx for use in Section 3.3. Although you may program in any language you desire, you should look at these for some hints.

# BackGround

By now, you should know something about histograms; after all, we discussed them in ENES101! *Hint: At the very least, take a look at help histogram in MATLAB, even if you are programming in Python or something else.* The process we’ll investigate in this lab is shown in Figure 1.



Figure 1 Histogram to PMF to PDF

Referring to the top box of Figure 1, we take a collection of *events* from multiple *independent* trials of a specific experiment, assign numbers to those events by means of a *random variable*, and then sort the *values of the random variable* into bins, counting the number in each bin. This creates our “N-point Histogram”. The bin width is  and the bin centers are .

Referring to the middle box of Figure 1, if we increase the number of trials in our sample to a very large number, and then divide the number of occurrences in each bin (i.e., the value of the raw histogram in that bin) by the *total number of independent trials*, we form a naïve estimate of the probability that the value of the random variable falls in that bin. Clearly, if we sum up all of these estimates, we should compute exactly 1.000…, because we will have accounted for all of the trials. Thus, in a simple view, this scaled histogram represents the probability mass function (PMF) of discrete events where the value of our random variable, , “rounds” to the value at the center of the bin, .

Referring to the lowest box, if we then also let the size of the bin, *decrease* as we make the number of trials larger, the histogram should more and more closely approach the analytical probability density  (PMF), where  represents the center of the bin. We write this infinitesimally small bin width as , just like in Calc II.

The following experiments will illustrate this process.

# Experiments

## PMF for a single fair die

Using the MATLAB function randi(imax, m, n)[[1]](#footnote-1), model the number of dots showing on a fair six-sided die. In this case the *number* of dots – that is, resulting random integer – is the random variable. Each element of the returned  matrix of values is *one* trial. Generate histograms using 120; 1200; 12,000; 120,000 trials and generate the unnormalized and normalized histograms where the y-value associated with each bin is . In each case, compute and report the (sample) mean and (sample) variance of your trials. *Hint: use the appropriate MATLAB functions for this!*  Discuss your observations as the number of points increases. How do the histograms vary (or not) from what you expect? Note that your histogram is an estimate of the Probility Mass function, because “number of dots” is a discrete random variable. It doesn’t matter how small you make the bins, you will still get values only at the integers from 1…6.

For this problem, the analytical expected value, or mean, is , and the analytical variance 2.9167. How do the (sample) mean and (sample) variance compare with the analytical values? What do you observe as the value of *N* increases?

This section requires four plots, one for each number of trials. You only need plot the normalized histogram. *Hint: Look at the skeleton solution.*

## PMF for binary strings

Now generate a series of strings of 100 binary values, where each value can be either 0 or 1. First, let the probability of a value of 1 be . See the hints in the accompanying MATLAB script. Generate a large number of these strings. For this problem, the value of the random variable is the *index* of the first 1 in the string, not the string itself. This is a mapping from a random event (the string) to a[n] (integer) number and thus, a random variable. For each string of 100 binary values, compute the value of the random variable and then create a histogram of these values.

Follow the same process as in 3.1: scale the histograms to compute a PMF for this geometrically distributed random variable. Determine the value of the analytical, or population, PMF based on your value of , and plot the analytical values on the same axis of the scaled histogram or PMF.

Compute your sample mean and variance, and the analytical or population, mean and variance,  and  for each value of  and each value of 

Repeat this for .

Then do the entire process over again, including for various values of  for.

For each of the nine cases (Product Rule!  ) answer the following questions. How do the histograms vary (or not) from what you expect? How do the (sample) mean and (sample) variance compare with the analytical values? *Hint: A table might be a good way to summarize your answers.*

This section requires nine plots. MATLAB subplots are recommended.

## PDF for an exponentially distributed random variable.

Using the provided MATLAB function randx(n, k, lambda)[[2]](#footnote-2), generate histograms for  independent trials of , first using a raw (i.e unscaled) histogram and then using the ‘Normalization’,’pdf’ option in the MATLAB function histogram. Plot the resultant normalized histograms of each set of trials. On the same set of axes, plot the values value of the pdf, where  is the value at the center of each bin. Go to the text (or lecture slides) and review the definition of the probability density function. In each case, compute the sample mean and variance from your experiments) and the analytical (population) mean and variance. For this problem, the analytical expected value is  and the analytical variance is .

Answer and discuss these questions in your report: What scale factor creates the normalized histogram from the raw histogram? Discuss this scaling in terms of the two-step process shown in Figure 1. How does that inform your understanding of the meaning of the pdf? Comment on why the scaling was necessary. *Hint: The answer “to make it fit” is not acceptable.* How do the (sample) mean and (sample) variance compare with the analytical (population) values? What trend do you observe?

This section requires three plots, each of which has the normalized histogram and the analytical (population) pdf on the same axes. Each plot represents one set of trials.

## PDF for a unit variance normal or Gaussian distributed random variable.

Using the built-in MATLAB function randn(n,k)[[3]](#footnote-3) generate histograms for 10, 1000, and 100,000 independent trials of a zero mean, unit variance, Gaussian (Normal) random variable.

Create the raw and normalized histograms as in Section 3.3. Plot the scaled histograms of each set of trials, and, on the same axis, plot the analytical PDF evaluated at the bin centers. For each set of trials, compute the sample mean and variance and the population mean and variance, as in Section 3.3 above. For this problem, the analytical expected value is  and the analytical variance is .

As in Section 3.3, answer and discuss these questions in your report: What scale factor creates the normalized histogram from the raw histogram? Discuss this scaling in terms of the two-step process shown in Figure 1. How does that inform your understanding of the meaning of the pdf? Comment on why the scaling was necessary. *Hint: The answer “to make it fit” is not acceptable.* How do the (sample) mean and (sample) variance compare with the analytical (population) values? What trend do you observe?

This section requires three plots, each of which has the normalized histogram and the theoretical pdf on the same axes. Each plot represents one set of trials.

## PDF for a normal or Gaussian distributed random variable.

Repeat all of 3.4 with a samples from , that is, a normal (Gaussian) random variable with  and . *Hint: You will have to modify the output of the MATLAB function* randn(n,k)*to get the desired pdf.*

For this problem, the analytical expected value is  and the analytical variance is .

As in Section 3.4, answer and discuss these questions in your report: What scale factor creates the normalized histogram from the raw histogram? Discuss this scaling in terms of the two-step process shown in Figure 1. How does that inform your understanding of the meaning of the pdf? Comment on why the scaling was necessary. *Hint: The answer “to make it fit” is not acceptable.* How do the (sample) mean and (sample) variance compare with the analytical (population) values? What trend do you observe? This section requires three plots.

## Computing probabilities from the pdf

Using the unscaled histogram from Section 3.5, count the number of trials that fall between  and  . Scale this to be a probability by dividing by the total number of trials. Then use your normalized histogram, which models the probability density function, to compute the sample  probability that the random variable falls between 1.0 and 3.0. Finally, numerically integrate the true probability density function  to find the probability that . Compare your results and discuss any differences. How might the width of the bins affect your answer?

There are no plots required in Section 3.6.

# Instructions for PROJECT REport

## Report Format

The project report shall be in the same form as this document, with an introduction, simulation and discussion section, and a "what I learned" section. Each section shall contain the content identified in Section 2 and described in more detail below subsection below. The report shall be in Times New Roman 11-point font. MATLAB pictures shall be pasted in-line in the report (this is a useful skill to know!); shall be numbered consecutively (use the Word Caption function)[[4]](#footnote-4); shall be appropriately titled; the axes shall be appropriately labeled; the curves shall be appropriately identified by an appropriate legend. I’ll provide suggestions on professional-looking plots on Blackboard. Please follow them.

## Section 2 Content

Section 2 of the report shall be titled "Simulation and Discussion" and shall contain the simulation plots and a discussion of each plot. The discussion shall address the points identified in Section 2. and any other interesting observations that occur to you. Remember, I know this stuff: you don't. The whole purpose is to have you look at the plots and tell me what you see and what it means to you. A large part of the grade is based on what you observe, so take your time!

The subsections of Section 2 in your report should match the subsections of Section 2 of this document.

There are a total of 22 plots required.

## Section 3 Content

Section 3 of the report shall be titled "What I learned" and shall contain a summary of what information you observed, what insights you gained, etc. Section 3 shall also contain a subsection critiquing the project and suggesting improvements that I could institute for next spring. Finally, Section 3 shall contain an estimate of how much time you spent on the project, including reading, research, programming, writing, and final preparation.

## Questions

I will accept questions regarding the project via the Ask The Professor Forum on Blackboard, so that I can reply to the entire class, and so that no student has an advantage by clever questioning. I will not do the project for you. I will not be answering (or even acknowledging) individual e-mails, so don’t ask.

I will *stop* responding to questions at 12 Noon on the day before the project is due. If you don’t start early, don’t ask for clarification, or read the earlier clarifications.

## Project Grading

The project shall be graded in the following way:

75% of the project score shall depend on the technical, theoretical, and graphical presentations of the tasks set out in Section 2 of this document.

25% of the project score shall be based on an evaluation of the technical writing against the Rubric on Technical Writing, posted on Blackboard, including grammar, clarity, organization, etc. For the purpose of this document, you can assume that the intended audience consists of your CMPE320 classmates.

I will be assisted in grading by the TA/grader staff for this course, but I will personally grade two projects for every student, spread over the five total projects.

## Project Delivery

The project shall be delivered by 11:59 PM on the date indicated in the Detailed Schedule spreadsheet.

Delivery shall be by submission of a PDF file as a Blackboard assignment. This is an individual assignment. You should also publish and deliver your MATLAB files in a single PDF file in the same assignment.

## Academic Integrity

The academic integrity provisions you signed at the beginning of class are in effect. You may discuss the interpretation of the assignment and approaches to solve the various problems amongst yourselves. You MAY NOT share MATLAB code, plots, text, etc. Do your own work.

I *will* be looking at source files for similarities, so please do not even attempt to copy work.

You ***may not*** ask for assistance on the project from the TA/grader staff, although you may ask for help with the various concepts. So, for example, if you don’t understand a Gaussian pdf you may ask for help understanding pdfs in general and Gaussian pdfs in particular. You may **not** ask for help completing the tasks assigned in this document.

You ***may* ask *me*** for assistance via the Ask The Professor forum. You may ask ***me***for help on the concepts during my open office hours.

**An important note about Chegg!**

Chegg and other cheating websites are anathema to learning this content. In fact, you have a far better way to get out of trouble if you are concerned: **ask *me*!** If I find content from this project on Chegg, there will be repercussions for the entire class. There’s simply no reason to resort to a cheating site (and be reported for an academic integrity violation) when **my** office hours and review sessions are specifically to answer questions, but not to do the work for you.

***Final word: Do NOT let this wait until the last minute. Debugging the multitude of plots will take time.***

1. For those programming in a language other than MATLAB, randi(imax, m, n) creates an  array of random integers in the range of 1 to imax. [↑](#footnote-ref-1)
2. For those programming in a language other than MATLAB, randx(n,k,lambda) creates an  array of random values from the distribution . You may look at the MATLAB code for randx to see how to create your own. [↑](#footnote-ref-2)
3. For those programming in languages other than MATLAB, randn(n, k)generates independent samples from a zero mean, unit variance Gaussian pdf, . [↑](#footnote-ref-3)
4. For those who are NOT in CMPE349, Figures are numbered at the bottom. Remember “Figures at the foot, Tables at the top.” [↑](#footnote-ref-4)