BIOE 198MI Biomedical Data Analysis. Spring Semester 2018.
Lab 4: Introduction to Probability and Statistics

A. Discrete Distributions: The probability of specific event E occurring, $\Pr(E)$, is given by

$$\Pr(E) = \lim_{N\to\infty} \frac{n(E)}{N} \quad \text{for } E \in \mathrm{S} \quad , \tag{1}$$

where $n(E)$ is the number of events occurring and N is the number of all possible events within event space S. For example, throwing one fair die we find the probability of event E=3 to be $\Pr(3) = 1/6$. The event space for this experiment is S ∈ {1,2,3,4,5,6} has size 6. This result occurs because throwing the die N times results in $\Pr(3) = (N/6)/N = 1/6$ as $N \to \infty$.

Repeating the calculation for each of the six possible events in S, we find these *discrete events* are equally probable, i.e., $\Pr(E) = 1/6$ for all E (Fig 1 left). This is a *uniform distribution*, and since the events are also mutually exclusive, we can sum the individual probabilities to find the total probability of obtaining a number between 1 and 6 to equal one. That is, $\sum_{E\in S} \Pr(E) = \sum_{E=1}^{6} \Pr(E) = \frac{6}{6} = 1$. We can also ask, "What is the probability of finding a 2 or 3 after one roll?" The answer is found by summing the probabilities over both events, $\sum_{E=2}^{3} \Pr(E) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$. Asking "2 or 3" is the union of mutually exclusive events, and so we sum their probabilities. To find "the chance we roll a 2 and 3 for two throws," we are looking at the intersection of events and so multiply their probabilities, in this case $\frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$, a rarer event. Because photon absorptions, like die rolls, are mutually-exclusive discrete events, it can be argued that the probability of the union of events is simply the sum of probabilities. Note that all I need to know for a uniform discrete distribution in the range of events because the individual probabilities are equal and sum to one, as all probabilities must.

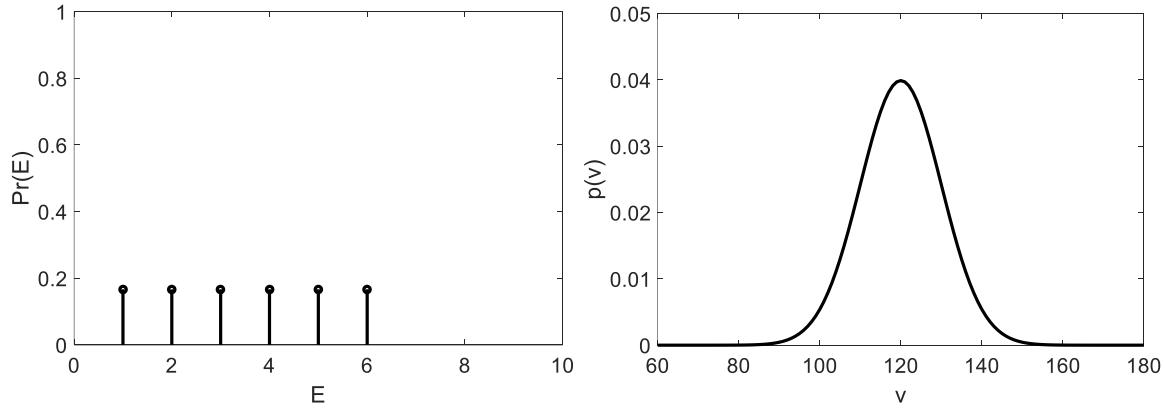

Figure 1. (left) The individual probabilities for throwing a fair die are plotted. Events E are **discrete** (therefore the stem plot), and the individual probabilities are equal and thus **uniformly distributed**, summing to one. (right) A **normally distributed** variable v (voltage) is plotted. This distribution is a probability density function (pdf) with mean voltage 120V and standard deviation 10V. Voltage values are **continuous** events. Also the integral of p(v) over all v equals one.

B. Continuous Distributions. Consider a noisy voltage measurement over time, $v(t)$ (See Fig 2). This voltage is measured at an outlet in your house and a histogram of the random fluctuations over time is shown. We see the time-average voltage $\bar{v}(t) = 120$V and the spread of values over time

about the mean value is given by the standard deviation σ = 10V. A normalized histogram of voltages over time, as $t \to \infty$, is a *continuous univariate distribution*, for example, Fig 1 right. This is the normal probability density function pdf, $p(v)$. Note that pdfs are not probabilities! In fact, the equivalence between an individual discrete probability and an individual continuous probability is $\Pr(E) \sim \Delta v\, p(v)$.

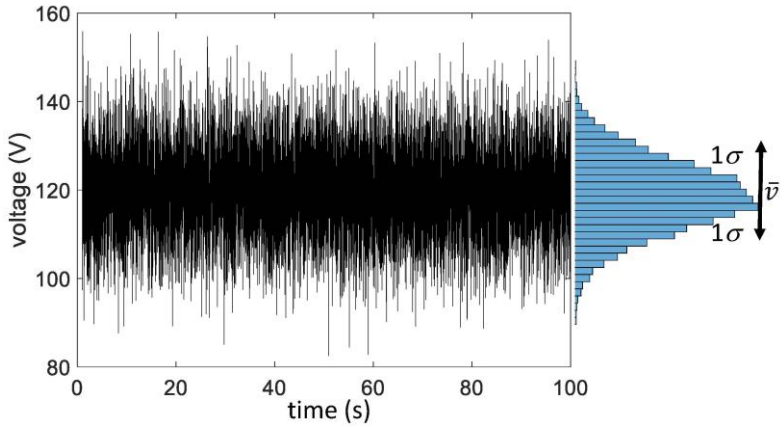

Fig. 2. Noisy voltage trace on left and a histogram of 10^4 voltage values to the right over 100 s of measurements (Δt=0.01s).

The pdf for a normal random process is

$$p(v;\bar{v},\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(v-\bar{v})^2/(2\sigma^2)} \quad (2)$$

Here we indicate a pdf of voltage that is *parameterized* by terms following the semicolon: the mean voltage $\bar{v}$ and standard deviation σ (or variance σ^2). The factor out front is used to make sure $\int_{-\infty}^{\infty} dv\, p(v) = 1$, which is true for all probabilities.

C. Simulating a Uniformly-Distributed Discrete Random Variable. The following script simulates `j` samples drawn from a uniform distribution of integer samples over the range `1:im` and places them in an `MxN` array named `g`. The three distributions simulated are called *sample distributions*.

```
%
k=0;
for j=[100 10000 1000000]
    k=k+1;                          %set counter
    g = randi(6,j,1);
    subplot(3,1,k);histogram(g,'Normalization','probability')
end
%
```

This script generates and histograms random samples while properly normalizing the bins so bin values sum to one.

Questions:

1. Can you see how the FOR loop repeats the process three times?

2. If the line `g = randi(6,j,1);` is changed to `g = randi(6,j,j);` how must the FOR loop change to give the same result as the script above?

D. Simulating a Normally-Distributed Continuous Random Variable. The following script parallels the script above but to simulate a normal pdf. Since we know parameters that generate each of the three *sample data distributions*, we can apply Eq (2) to display the true *parent pdf*.

```
%
k = 0; s = 10; m = 120; v = 60:0.1:180; %set counter, parameters & v range
for j = [10 100 1000]                   %vary the sample size
    k = k+1;                            %index the counter
    g = s*randn(j)+120;                 %modify stand norm for problem
    gp = exp(-(v-120).^2/(2*10^2))/(10*sqrt(2*pi));      %parent pdf
```

```
    subplot(3,1,k);histogram(g,'Normalization','pdf');hold on;
    plot(v,gp);hold off
end
fprintf('Mean voltage is %0.1f\n', mean(mean(g)));
str = ['Mean voltage is ' num2str(mean(mean(g))) ' volts.'];disp(str)
fprintf('Standard deviation of voltage is %0.5f\n', mean(std(g)));
%
```

Matlab provides a standard normal pseudo-random number generator `randn(N)` that has zero mean and unit variance. The output is an N x N array. To generate a column vector of N samples, use `randn(N,1)`. Row vector?

To modify the standard normal results for a specific problem, we multiply by `s` $= \sigma$ and `m` $= \bar{v}$. You see, the exponential of a specific Gaussian kernel is $(v - \bar{v})^2/(2\sigma^2)$ and that of the standard normal is $z^2/2$. Equating the two quantities and solving for v, we find $= \sigma z + \bar{v}$. I also changed the normalization to estimate a pdf. Note that we are NOT fitting data to a model. Instead, we are plotting the pdf for the parent population via Eq (2) and comparing it to normalized sample-data histograms from simulations. Finally, notice how once the histograms were plotted, we applied the `hold on` command, plotted the parent distribution, and then applied the `hold off` command. We also printed out the mean and standard deviation to see that the sample means and standard deviation for 10^6 samples is pretty close to the population values input to the model.

Another tool of interest is `rng` used to "seed" the random number generator.

```
rng('default');c=randn(3,1)  %resets seed to reproduce c w/ea call
rng('shuffle');c=randn(3,1)  %seed randn with clock time to vary
```

Assignment:
Adapt the scripts above to histogram samples of the Poisson random process (discrete random variable whose variance equals to its mean) along with its parent population. You can generate individual probabilities using `g=poissrnd(5,1000,1000);` and the parent distributions using `p=poisspdf(x,5);`

Solution Here.

E. Central Limit Theorem
Normal pdfs are often used because of the central limit theorem (CLT). Roughly, the CLT tells us that when independent random variables are summed, the appropriately normalized result has a distribution that tends toward a normal distribution. Don't believe me? Good, let's see if we can show this using numerical simulations.

Assignment:
Let me suggest the following steps to see if you can demonstrate the CLT using samples drawn from a uniform distribution.

1. Generate a 2-D array of samples drawn from a uniform random number generator. Let the array be 10,000 x 20.
2. Histogram the first column and place it in the first location of a 4x1 subplot array.
3. Sum first two rows and histogram, placing it in the second subplot location.
4. Sum first 10 rows and histogram, placing it in the third subplot location.
5. Sum first 20 rows and histogram, placing it in the fourth subplot location.

Be sure to title and label plots appropriately. Describe what do you see.

Solution here.

F. Bivariate distributions. When two measurements are made on a patient, let's call them x_1, x_2, and each is a normal random variable, we can described the pair by a bivariate normal pdf,

$$p(\boldsymbol{x};\boldsymbol{\theta}) \;=\; p(x_1,x_2;\boldsymbol{\theta}) \;=\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\exp\left[-\left(\frac{\sigma_2^2(x_1-\bar{x}_1)^2-2\rho\sigma_1\sigma_2(x_1-\bar{x}_1)(x_2-\bar{x}_2)+\sigma_1^2(x_2-\bar{x}_2)^2}{2\sigma_1^2\sigma_2^2(1-\rho^2)}\right)\right] . \qquad (3)$$

Wow, this equation is messy, huh? Let's look at the details. $\boldsymbol{x} = (x_1, x_2)$ is a two-element measurement vector, $\boldsymbol{\theta} = (\bar{x}_1, \bar{x}_2, \sigma_1^2, \sigma_2^2, \rho)$ is a parameter vector with measurement means, variances, and the correlation coefficient ρ that describes the dependence between the two measurements. Boldfaced quantities indicate vectors. When measurements x_1 and x_2 are uncorrelated, then Eq (3) simplifies to

$$p(\boldsymbol{x};\boldsymbol{\theta}) \;=\; \frac{1}{2\pi\sigma_1\sigma_2}\exp\left[-\left(\frac{(x_1-\bar{x}_1)^2}{2\sigma_1^2}+\frac{(x_2-\bar{x}_2)^2}{2\sigma_2^2}\right)\right] \;=\; p(x_1;\bar{x}_1,\sigma_1^2)\,p(x_2;\bar{x}_2,\sigma_2^2) . \qquad (4)$$

Uncorrelated bivariate normal measurements $\rho = 0$ are also statistically independent

$$p(\boldsymbol{x};\boldsymbol{\theta}) \;=\; p(x_1;\bar{x}_1,\sigma_1^2)\,p(x_2;\bar{x}_2,\sigma_2^2) . \qquad (5)$$

Simulating and displaying bivariate data. The script below describes four sets of flow cytometry data. Four sets help illustrate how parameters affect the data appearance. Two sets have correlated forward and side scatter intensity measurements, i.e., $\rho \neq 0$, while two are uncorrelated, $\rho = 0$. Copy and paste the script below into the Command Window of Matlab to see the results.

```
% The following script simulates four flow cytometry data sets that are
% each bivariate normal.  However, the parameter vectors, theta, for
% each data set varies.
%%
close all;
rho=-0.4;s1=[100 40];m1=[500 180];N1=50; %First data set
z=randn(N1); %N1^2 is the number of data points simulated
X1=s1(1)*z+m1(1);      %X1 and Y1 convert from standard normal pdf
Y1=s1(2)*(rho*z+sqrt(1-rho^2)*randn(N1))+m1(2);
plot(X1,Y1,'k.');axis([0 1000 0 1000]);axis square; %plot on fixed axis
text(730,270,'S1');hold on   %label the first group S1
%%
rho=0;s2=[40 100];m2=[500 600];N2=20;   %Second data set
z=randn(N2);
X2=s2(1)*z+m2(1);
Y2=s2(2)*(rho*z+sqrt(1-rho^2)*randn(N2))+m2(2);
plot(X2,Y2,'ro');text(620,880,'S2')
%%
rho=0;s3=[30 30];m3=[200 700];N3=30;    %Third data set
z=randn(N3);
X3=s3(1)*z+m3(1);
Y3=s3(2)*(rho*z+sqrt(1-rho^2)*randn(N3))+m3(2);
plot(X3,Y3,'bx')
```

```
text(180,850,'S3')
%%
rho=0.98;s4=[40 40];m4=[800 600];N4=50; %Fourth data set
z=randn(N4);
X4=s4(1)*z+m4(1);
Y4=s4(2)*(rho*z+sqrt(1-rho^2)*randn(N4))+m4(2);
plot(X4,Y4,'bx');hold off
title('Bivariant Normal Flow Cytometry Data');
xlabel('Side Scatter Intensity');ylabel('Forward Scatter Intensity')
text(850,580,'S4');hold off
% save('FN1','X1','Y1','X2','Y2','X3','Y3') %use these for the assignment
%load('FN1.mat'); %then use whos command to see what is loaded
%%
[rho1]=corr(X1,Y1);R1=trace(rho1)/N1;          %Here we estimate Pearson's
str = ['rho_1 = ' num2str(R1)];disp(str)       %correlation and average for
[rho2]=corr(X2,Y2);R2=trace(rho2)/N2;          %all points along diagonal
str = ['rho_2 = ' num2str(R2)];disp(str)
[rho3]=corr(X3,Y3);R3=trace(rho3)/N3;
str = ['rho_3 = ' num2str(R3)];disp(str)
[rho4]=corr(X4,Y4);R4=trace(rho4)/N4;
str = ['rho_4 = ' num2str(R4)];disp(str)
%
```

Notice how the distribution parameters affect the appearance of the data in the x_1, x_2 plane. Also compare the set ρ values in the models with their estimates, i.e., `rho1 - rho4` values output. If the script runs a few times, `rho` change a bit but remain fairly close to the corresponding ρ values.

The N x N matrices `rho1` through `rho4` display ***Pearson's correlations coefficients*** that measure how dependent x_2 is on the value of x_1. The assumption for Pearson's correlations to be meaningful is that both measurements are normally distributed. A correlation coefficient of `1` means that x_2 is ***positively correlated*** with x_1; as one measurement changes, so does the other. A value of `-1` also indicates a strong dependence, however, the two measurements are ***anti-correlated***. A value of zero indicated no linear dependence exists between the two measurements; they are ***uncorrelated***. You should create images out of the `rho1` through `rho4` arrays to see how they are structured. This structure might then tell you why I computed the trace of the N x N matrix, $tr(R) = \sum_{n=1}^{N} R_{nn}$ and divided by N.

You should see from the plots that

1. measurements x_1 and x_2 are uncorrelated for S2 and S3.
2. the variances σ_1^2 and σ_2^2 are equal for S3 and fairly small, while those for S2 are unequal.
3. S1 data are weakly and negatively correlated and S4 data are strongly positively correlated.

Assignment:
Break up into four groups. Each group should use the Matlab script above to form two groups of bivariate normal data, call them S1 and S2. Write the data arrays created to files using the commented-out SAVE command in the script above. Exchange with another group without telling them anything about your data and let them estimate the parameters of both groups, i.e., mean, variance or both groups and the correlation coefficient. Your job it to think up challenging distributions to make the other group work hard. Remember that $-1 \leq \rho \leq 1$.

G. Inferential Statistics. Sections A-F above explain descriptive statistics. These are equation models offering parameters as descriptive summaries of random distributions. You can't predict data exactly but you can know data statistically by estimating distribution parameters. Powerful idea!

Inferential statistic uses these statistical descriptions to ask questions like "is a group of data well represented by a normal distribution?" After all, visually inspecting plots of measurement data provides no quantitative summary measures. Matlab provides the function `[H,P] = chi2gof(X)`, which is a ***chi-square goodness-of-fit test*** that the data in vector `X` are drawn from a normal distribution with mean and variance estimated from the data in vector `X`. The result is `H=0` if the null hypothesis cannot be rejected at the 5% significance level, $p<0.05$ (`H=0` means `X` is described well by a normal distribution). The result is `H=1` if the null hypothesis can be rejected at the 5% level, $p>0.05$ (`H=1` means `X` is not well represented by a normal distribution.) `P` is the probability of observing this result by chance if the null hypothesis is true.

Let's try it! Below, two univariate Poisson-distributed (not quite[1]) sample groups with different means are simulated to see how well each is approximated as a normal distribution. A third normal distribution, with mean and variance equal to the larger Poisson distribution, is also simulated.

```
%%
close all
g1=poissrnd(5,1000,1)+190; %mean = 190+5, var = 5
histogram(g1);hold on;text(191,160,'g1') %histogram and label
g2=poissrnd(25,1000,1)+180; %mean = 180+25, var = 25
histogram(g2);text(201,160,'g2')
g3=5*randn(1000,1)+205; %normal dist with same parameters as poiss
histogram(g3);text(222,10,'g3')
title('Total Cholesterol');
xlabel('mg/dL');ylabel('Number of Patients');hold off;
[H1,P1]=chi2gof(g1);   % chi2gof test how well g1 is rep by norm pdf
str = ['H1 = ' num2str(H1),' P1 = ' num2str(P1)];disp(str)
[H2,P2]=chi2gof(g2);
str = ['H2 = ' num2str(H2),' P2 = ' num2str(P2)];disp(str)
[H3,P3]=chi2gof(g3);
str = ['H3 = ' num2str(H3),' P3 = ' num2str(P3)];disp(str)
%
```

The script above applies the ***χ^2 goodness-of-fit test for normality***. If you run it a few times, you will see that group 1 samples `g1` are never considered normal. That is, `H` is always 1 and `P` is always very small, viz., `P<0.05`. In contrast, `g3` is always normal, i.e., `H = 0` as it should be, and `P>0.05`. The interesting one is `g2` for which can go either way. Note that a Poisson distribution is well approximated by a normal distribution as its parameter becomes large, as $\lambda \rightarrow \infty$.

H. Comparing Groups of Data Samples. There are several types of ***t-tests*** but all of them ask some version of the question of whether two data vectors are from the same distribution. The assumption is that if they are from the same distribution, then there is no difference between their means and so they come from the same source. This is important if you want to test a diagnostic indicator and want to make a statistical argument that healthy and diseased cohorts of patients are clearly differentiated by the test.

[1] Inspecting the code, you might notice that I added constants to the Poisson variables to shift values near the serum cholesterol levels one might measure from patients. These are then a constant plus a Poisson RV.

`H = ttest(X,Y)` performs a ***paired t-test*** of the hypothesis that two matched samples in vectors `X` and `Y` come from distributions with equal means. The difference `X-Y` is assumed to come from a normal distribution with unknown variance. `X` and `Y` must have the same length.

Modeling cell growth. Let's simulate an experiment where MCF-7 human mammary carcinoma cells are inoculated into 150-sq cm flasks at 3×10^5 cells/flask. After 48 hrs, these cells grow exponentially for 5 days with a mean population doubling time of about 24 hrs. Assume on day 2 the cell number is $N(2) = 4 \times 10^5$ cells in an area 150 cm^2. Assuming cells grow exponentially during days 2-7, which is $M = 6$ days, then the growth equation is

$$N(t) = N(2)\exp\big(k(t-2)\big) \quad for\ t = 2,3,\dots,7. \tag{6}$$

The growth rate k depends on the doubling time t_2 according to $k = \ln(2)/t_2$. Assume you have 20 flasks prepared identically, $L = 10$ without treatment and $L = 10$ treated with an agent that accelerates the growth rate. So there are potentially two growth rates k_1 and k_2 one for each of the groups.

```
%%
close all
L=10;M=6;t=2:M+1;t2=1;k1=log(2)/t2;N0=4e5; %Set parameters, units = [days]
k2=1.2*k1;N1b=zeros(L,M);N2b=zeros(L,M);   %Set parameter and initial
%                      % Note that growth rate k2 is some fraction of k1.
N1=N0*exp(k1*(t-t(1)));N2=N0*exp(k2*(t-t(1)));%Modeled data
figure; hold on                        %Initiate plot outside FOR
for j=1:M                              %simulate M pts at each time point (j)
   N1b(:,j)=N0*randn(10,1) + N1(j); %simulating noisy measurement data
   plot(t(j),N1b(:,j),'ko')
   N2b(:,j)=N0*randn(10,1) + N2(j);
   plot(t(j),N2b(:,j),'r*')
end
hold off;
title('Cell growth');xlabel('time (days) after plating');ylabel('Cell
count/150 cm^2')
ax = gca; % current axes
ax.FontSize = 16;
%
disp('H=0 indicates mean(N1b)-mean(N2b) = zero.') %give user options
[H,P] = ttest2(N1b,N2b,'alpha',0.05);   %apply paired t-test at each time
str = ['H = ' num2str(H),' and P = ' num2str(P)];disp(str)
figure;errorbar(t,N1,std(N1b),'linewidth',2);hold on;
errorbar(t,N2,std(N2b),'linewidth',2);hold off;
title('Cell growth');xlabel('time (days) after plating');ylabel('Cell
count/150 cm^2')
ax = gca; % current axes
ax.FontSize = 16;
%
```

Notice that the data are plotted two ways. I did not control the axes so unfortunately they are not directly comparable.

Assignment:

(a) Adjust `k2` (fraction of `k1`) to find the threshold growth rate where all 6 days clearly have different means at the p=0.05 level. Also find the `k2` threshold where all 6 days have the same means. Those two values span the range where the growth rates are clearly the same and different. The agent concentrations corresponding to those two threshold cell growth rates give the region of ambiguous effect.
(b) Examine other data parameters and analyze what is important for determining the separability between the two types of MDF 7 cells. I really want you to analyze the script and its parameters to see what conclusions you can draw about cell growth as determined by this numerical model.

Solution to part (a) here.