CALIFORNIA STATE UNIVERSITY, LONG BEACH

EE 381 - Probability and Statistics with Applications to Computing

Central Limit Theorem

Simulating Continuous Random Variables with Various Distributions

0. Introduction and Background Material

0.1. Simulating a R.V. with Uniform Probability Distribution

The Python function "`numpy.random.uniform(a,b,n)`" will generate n random numbers with uniform probability distribution in the open interval $[a,b)$.

The PDF of a random variable uniformly distributed in $[a,b)$ is defined as following:

$$f(x) = \begin{cases} \frac{1}{(b-a)}, & a \le x \le b \\ 0, & \text{otherwise} \end{cases} \quad ; \quad \text{and} \quad P(X \le x) = F(x) = \begin{cases} 0, & x < a \\ \dfrac{(x-a)}{(b-a)}, & a \le x < b \\ 1, & x \ge b \end{cases}$$

It is noted that the mean and variance of a uniformly distributed random variable X are given by:

$$\mathrm{E}(X) = \mu_X = \frac{a+b}{2} \quad ; \quad \mathrm{Var}(X) = \sigma_X^2 = \frac{(b-a)^2}{12}$$

0.2. Simulating a R.V. with Exponential Probability Distribution

The Python function "`numpy.random.exponential(beta,n)`" will generate n random numbers with exponential probability distribution.

The PDF of a random variable exponentially distributed is defined as following:

$$f_T(t\ ;\ \beta) = \begin{cases} \frac{1}{\beta}\, exp(-\frac{1}{\beta}t), & t \geq 0 \\ 0, & t < 0 \end{cases}$$

From the above definition, the CDF of T is found as:

$$P(T \leq t) = F(t) = \begin{cases} 0, & t < 0 \\ 1 - \exp(-\frac{1}{\beta}t), & t \geq 0 \end{cases}$$

It is noted that the mean and standard deviation of the exponentially distributed random variable T are given by:

$$\mu_T = \beta \quad ; \quad \sigma_T = \beta$$

0.3. Simulating a R.V. with Normal Probability Distribution

The Python function "`numpy.random.normal(mu, sigma, n)`" will generate n random numbers from a Gaussian probability distribution (also called normal probability distribution) with mean μ_X = mu and standard deviation σ_X = sigma.

The PDF of a normal random variable X with mean μ_X and standard deviation σ_X is defined as following:

$$f(x) = \frac{1}{\sigma_X\sqrt{2\pi}} \exp\{-\frac{(x-\mu_X)^2}{2\sigma_X^2}\}$$

It is noted that the mean and variance of the normally distributed random variable X are given by:

$$E(X) = \mu_X \quad ; \quad \text{Var}(X) = \sigma_X^2$$

0.4. Central Limit Theorem

If $X_1, X_2, \ldots X_n$ are independent random variables having the same probability distribution with mean μ and standard deviation σ, consider the sum $S_n = X_1 + X_2 + \ldots X_n$.

This sum S_n is a random variable with mean $\mu_{S_n} = n\mu$ and standard deviation $\sigma_{S_n} = \sigma\sqrt{n}$.

The Central Limit Theorem states that as $n \to \infty$ the probability distribution of the R.V. S_n will approach a normal distribution with mean μ_{S_n} and standard deviation σ_{S_n}, *regardless of the original distribution* of the R.V. $X_1, X_2, \ldots X_n$.

It is noted that the PDF of the normally distributed R.V. S_n is given by:

$$f(s_n) = \frac{1}{\sigma_{S_n}\sqrt{2\pi}} \exp\{-\frac{(x-\mu_{S_n})^2}{2\sigma_{S_n}^2}\}$$

PROBLEMS

1. Simulate continuous random variables with selected distributions

1.1 Simulate a Uniform Random Variable.

a) Create a random variable X with a uniform distribution. Use the Python function "`numpy.random.uniform(a,b,n)`" to generate n values of the R.V. X of with uniform probability distribution in the open interval $\left[a,b\right)$.

b) Use the histogram function to plot a bargraph of the experimental values of the R.V. X. On the same graph plot the probability density function for the R.V. X, given by
$f(x) = \begin{cases} \frac{1}{(b-a)}, & a \le x \le b \\ 0, & \text{otherwise} \end{cases}$ and compare to the bargraph plot.

c) Calculate the expectation and standard deviation of the R.V. X, using the Python functions numpy.mean and numpy.std . Compare to the theoretical values given by $\mu_X = \frac{a+b}{2}$; $\sigma_X^2 = \frac{(b-a)^2}{12}$

d) Your report should contain the graph require in (b) and the values required in (c) tabulated in the following table. The graph should be properly labeled.

e) A sample code for (a-c) is given below.

Table 1: Statistics for a Uniform Distribution			
Expectation		**Standard Deviation**	
Theoretical Calculation	Experimental Measurement	Theoretical Calculation	Experimental Measurement

Table 1. Calculations for a Uniform distribution

```
# The following code provides a way to create the bar graph of a
# uniform probability distribution in the interval [a,b)
# where  a=1, b=3.
# The code generates n=10000 values of the random variable.
# This is only a sample code. Your project has different values
# of a and b. You must use the correct values for your project
#
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

# Generate the values of the RV X
a=1; b=3; n=10000;
x=np.random.uniform(a,b,n)

# Create bins and histogram
nbins=30; # Number of bins
edgecolor='w';      # Color separating bars in the bargraph
#
bins=[float(x) for x in np.linspace(a, b,nbins+1)]
h1, bin_edges = np.histogram(x,bins,density=True)
# Define points on the horizontal axis
be1=bin_edges[0:np.size(bin_edges)-1]
be2=bin_edges[1:np.size(bin_edges)]
b1=(be1+be2)/2
barwidth=b1[1]-b1[0] # Width of bars in the bargraph
plt.close('all')

# PLOT THE BAR GRAPH
fig1=plt.figure(1)
plt.bar(b1,h1, width=barwidth, edgecolor=edgecolor)

#PLOT THE UNIFORM PDF
def UnifPDF(a,b,x):
    f=(1/abs(b-a))*np.ones(np.size(x))
    return f

f=UnifPDF(1,3,b1)
plt.plot(b1,f,'r')

#CALCULATE THE MEAN AND STANDARD DEVIATION
mu_x=np.mean(x)
sig_x=np.std(x)
```

1.2 Simulate an Exponentially distributed Random Variable.

a) Create a random variable T with an exponential distribution. Use the Python function "`numpy.random.exponential(beta,n)`" to generate n values of the R.V. T of with exponential probability distribution.

b) Use the histogram function to plot a bargraph of the experimental values of the R.V. T. On the same graph plot the probability density function for the R.V. T, given by

$f_T(t\ ;\ \beta) = \begin{cases} \frac{1}{\beta}\, exp(-\frac{1}{\beta}t), & t \geq 0 \\ 0, & t < 0 \end{cases}$ and compare to the bargraph plot.

c) Calculate the expectation and standard deviation of the R.V. X, using the Python functions numpy.mean and numpy.std . Compare to the theoretical values given by $\mu_T = \beta \quad ; \quad \sigma_T = \beta$

d) Your report should contain the graph require in (b) and the values required in (c) tabulated in the following table. The graph should be properly labeled.

e) Modify the sample code for (a-c) given previously.

Table 2: Statistics for Exponential Distribution			
Expectation		**Standard Deviation**	
Theoretical Calculation	Experimental Measurement	Theoretical Calculation	Experimental Measurement

Table 2. Calculations for Exponential distribution

1.3 Simulate a Normal Random Variable.

a) Create a random variable X with a normal distribution. Use the Python function "`numpy.random.normal(mu,sigma,n)`" to generate n values of the R.V. X of with normal probability distribution.

b) Use the histogram function to plot a bargraph of the experimental values of the R.V. X. On the same graph plot the probability density function for the R.V. X, given by

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

and compare to the bargraph plot.

c) Calculate the expectation and standard deviation of the R.V. X, using the Python functions numpy.mean and numpy.std . Compare to the theoretical values given by $\mu_X = \mu \quad ; \quad \sigma_X = \sigma$

d) Your report should contain the graph required in (b) and the values required in (c) tabulated in the following table. The graph should be properly labeled.

e) Modify the sample code for (a-c) given previously.

Table 3: Statistics for Normal Distribution			
Expectation		**Standard Deviation**	
Theoretical Calculation	Experimental Measurement	Theoretical Calculation	Experimental Measurement

Table 3. Calculations for Normal distribution

2. The Central Limit Theorem

Central Limit Theorem.

Consider a collection of books, each of which has thickness W. The thickness W is a RV, uniformly distributed between a minimum of a and a maximum of b cm. Use the values of a and b that were provide to you, and calculate the mean and standard deviation of the thickness. Use the following table to report the results. Points will be taken off if you do not use the table to report .

Mean thickness of a single book (cm)	Standard deviation of thickness (cm)
μ_w =	σ_w=

The books are piled in stacks of $n = 1, 5, 10, \text{ or } 15$ books. The width S_n of a stack of n books is a RV (the sum of the widths of the n books). This RV has a mean $\mu_{S_n} = n\mu_w$ and a standard deviation of $\sigma_{S_n} = \sigma_w \sqrt{n}$.

Calculate the mean and standard deviation of the stacked books, for the different values of $n = 1, 5, 10, \text{ or } 15$. Use the following table to report the results. Points will be taken off if you do not use the table to report.

Number of books n	Mean thickness of a stack of n books (cm)	Standard deviation of the thickness for n books
n=1	μ_w =	σ_w=
n=5	μ_w =	σ_w=
n=15	μ_w =	σ_w=

Perform the following simulation experiments, and plot the results.

a) Make $n = 1$ and run $N = 10,000$ experiments, simulating the RV $S = W_1$.

b) After the N experiments are completed, create and plot a probability histogram of the RV S

c) On the same figure, plot the normal distribution probability function and compare the probability histogram with the plot of $f(x)$

$$f(x) = \frac{1}{\sigma_S\sqrt{2\pi}} \exp\{-\frac{(x-\mu_S)^2}{2\sigma_S^2}\}$$

d) Make $n = 5$ and repeat steps (a)-(c)

e) Make $n = 15$ and repeat steps (a)-(c)

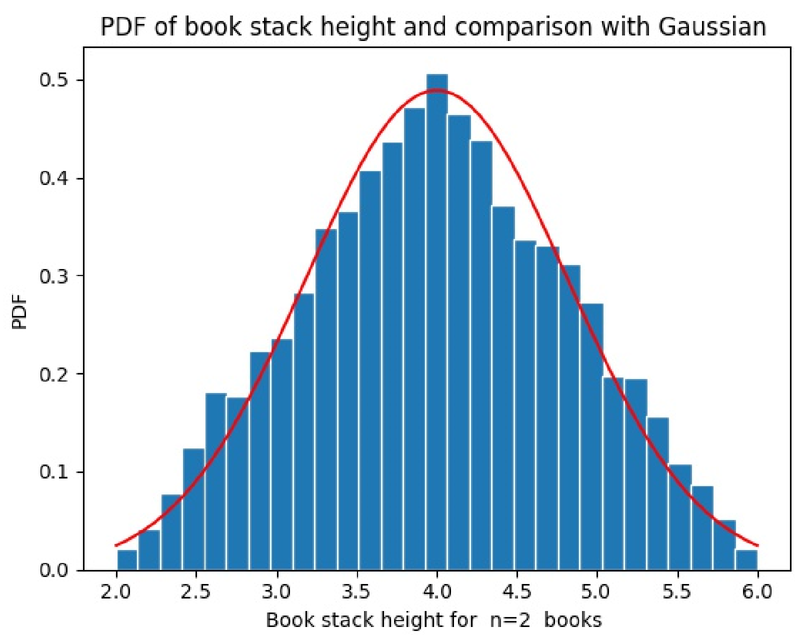
SUBMIT a report following the guidelines as described in the syllabus.
The report should include, among the other requirements:

- The above tables

- The histograms for $n = \{1, 5, 15\}$ and the overlapping normal probability distribution plots.
- The Python code, included in an Appendix.
- Make sure that the graphs are **properly labeled**.

An example of creating the PDF graph for $n = 2$ is shown below.
The code below provides a suggestion on how to generate a bar graph for a continuous random variable X , which represents the total bookwidth for $n = 2$, and $a = 1$, $b = 3$.

Note that the value of "barwidth" is adjusted as the number of bins changes, to provide a clear and understandable bar graph.
Also note that the "density=True" parameter is needed to ensure that the total area of the bargraph is equal to 1.0.



```python
# The code provides a way to create the bar graph at the end
# for n=2 and a=1, b=3
# This is only a sample code. Your project has different values
# of a and b. You must use the correct values for your project
#
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

# Generate the values of the RV X
N=100000; nbooks=2; a=1; b=3;
mu_x=(a+b)/2  ;  sig_x=np.sqrt((b-a)**2/12)
X=np.zeros((N,1))
for k in range(0,N):
    x=np.random.uniform(a,b,nbooks)
    w=np.sum(x)
    X[k]=w

# Create bins and histogram
nbins=30; # Number of bins
edgecolor='w';      # Color separating bars in the bargraph
#
bins=[float(x) for x in np.linspace(nbooks*a, nbooks*b,nbins+1)]
h1, bin_edges = np.histogram(X,bins,density=True)
# Define points on the horizontal axis
be1=bin_edges[0:np.size(bin_edges)-1]
be2=bin_edges[1:np.size(bin_edges)]
b1=(be1+be2)/2
barwidth=b1[1]-b1[0] # Width of bars in the bargraph
plt.close('all')


# PLOT THE BAR GRAPH
fig1=plt.figure(1)
plt.bar(b1,h1, width=barwidth, edgecolor=edgecolor)

#PLOT THE GAUSSIAN FUNCTION
def gaussian(mu,sig,z):
       f=np.exp(-(z-mu)**2/(2*sig**2))/(sig*np.sqrt(2*np.pi))
       return f

f=gaussian(mu_x*nbooks,sig_x*np.sqrt(nbooks),b1)
plt.plot(b1,f,'r')
```

3. Distribution of the Sum of Exponential RVs

This problem involves a battery-operated critical medical monitor. The lifetime (T) of the battery is a random variable with an exponentially distributed lifetime. A battery lasts an average of β days, which has been provided to you. Under these conditions, the PDF of the battery lifetime is given by:

$$f_T(t\,;\beta) = \begin{cases} \frac{1}{\beta}\exp(-\frac{1}{\beta}t), & t \geq 0 \\ 0, & t < 0 \end{cases}$$

As mentioned before, the mean and variance of the random variable T are:

$$\mu_T = \beta \quad ; \quad \sigma_T = \beta$$

When a battery fails it is replaced immediately by a new one. Batteries are purchased in a carton of 24. The objective is to simulate the RV representing the lifetime of a carton of 24 batteries, and create a histogram. To do this, follow the steps below.

a) Create a vector of 24 elements that represents a carton. Each one of the 24 elements in the vector is an exponentially distributed random variable (T) as shown above, with mean lifetime equal to β. Use the same procedure as in the previous problem to generate the exponentially distributed random variable T.

b) The sum of the elements of this vector is a random variable (C), representing the life of the carton, i.e.

$$C = T_1 + T_2 + \cdots T_{24}$$

where each T_j , $j = 1, 2, \cdots 24$ is an exponentially distributed R.V. Create the R.V. C, i.e. simulate one carton of batteries. This is considered one experiment.

c) Repeat this experiment for a total of N=10,000 times, i.e. for N cartons. Use the values from the N=10,000 experiments to create the experimental PDF of the lifetime of a carton, $f(c)$.

d) According to the Central Limit Theorem the PDF for one carton of 24 batteries can be approximated by a normal distribution with mean and standard deviation given by:

$$\mu_C = 24\mu_T = 24\beta \quad ; \quad \sigma_C = \sigma_T\sqrt{24} = \beta\sqrt{24}$$

Plot the graph of a normal distribution with

$$\text{mean} = \mu_C \text{ and (standard deviation)} = \sigma_C,$$

over plot of the experimental PDF on the same figure, and compare the results.

e) Create and plot the CDF of the lifetime of a carton, $F(c)$. To do this use the Python "`numpy.cumsum`" function on the values you calculated for the experimental PDF. Since the CDF is the integral of the PDF, you must multiply the PDF values by the `barwidth` to calculate the areas, i.e. the integral of the PDF.

If your code is correct the CDF should be a nondecreasing graph, starting at 0.0 and ending at 1.0.

Answer the following questions:

1. Find the probability that the carton will last longer than three years, i.e. $P(S > 3\times 365) = 1 - P(S \le 3\times 365) = 1 - F(1095)$. Use the graph of the CDF $F(t)$ to estimate this probability.

2. Find the probability that the carton will last between 2.0 and 2.5 years (i.e. between 730 and 912 days): $P(730 \le S \le 912) = F(912) - F(730)$. Use the graph of the CDF $F(t)$ to estimate this probability.

3. **SUBMIT a report following the guidelines as described in the syllabus.** The report should include, among the other requirements:

- The numerical answers using the table below. Note: You will need to replicate the table, in order to provide the answer in your report. Points will be taken off if you do not use the table.
- The PDF plot of the lifetime of one carton and the corresponding normal distribution on the same figure.
- The CFD plot of the lifetime of one carton
- Make sure that the graphs are **properly labeled**.
- The code in an Appendix.

QUESTION	ANS.
1. Probability that the carton will last longer than three years	
2. Probability that the carton will last between 2.0 and 2.5 years	