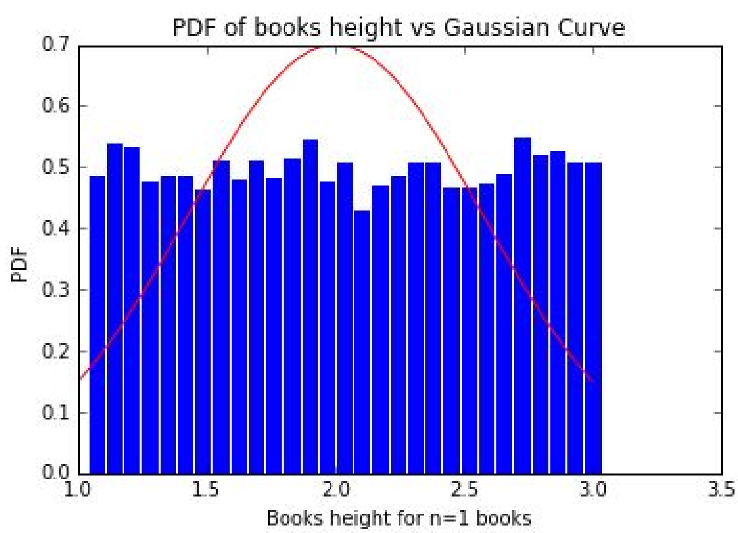
Roberto Sanchez (014587792)
October 31, 2017
EE 381 – Probability and Statistics with Applications to Computing
Lab 4 - Central Limit Theorem Simulating RVs with Exponential Distribution

1) The Central Limit Theorem
    a) **INTRODUCTION:**
        i) This problem we go through and generate a list of books with numbers and generating a distribution graph. On top of that we are going superimpose a gaussian distribution graph on top of it. As we increase the number of books we see it slowly takes shape of the gaussian curve.
    b) **METHODOLOGY:**
        i) We declare all given information on the program such as the sigma, variance, and mu. We generate our list from the number of rounds that is stated in the document, in this case it's 10,000. We plot the distribution on the graph and graph the gaussian curve on top of it from the given equation.
    c) **RESULTS AND CONCLUSIONS:**
        i) The results for the following booklist:

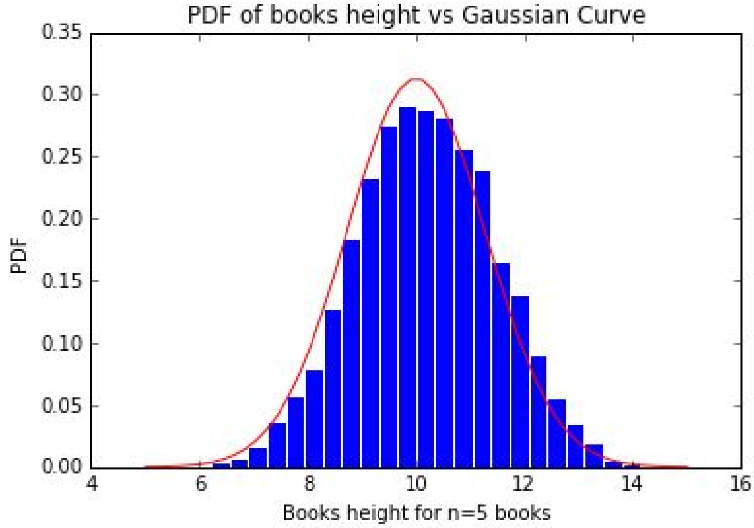
n = 1:

```
In [79]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
[ 2.01090649  1.36441834  1.13950956 ...,  1.69962672  1.29967564
  2.07817485]
```



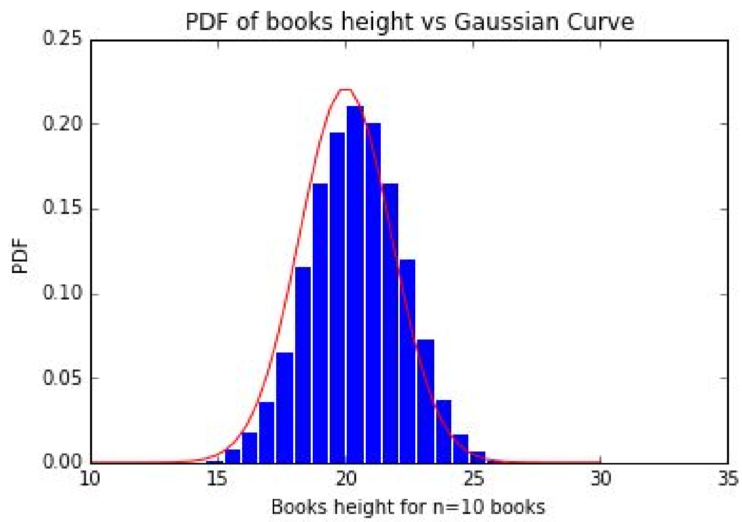
n = 5:

```
In [40]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
```



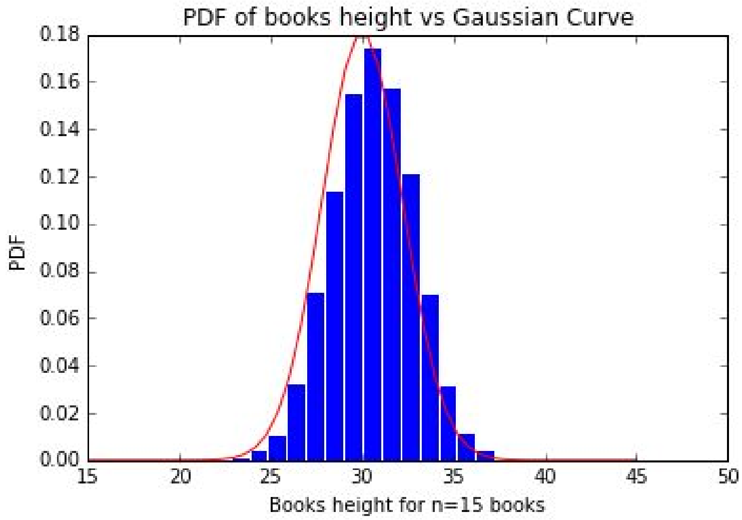
n = 10:

```
In [41]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
```



n=15:

```
In [42]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
```



Conclusion: As you increase n the histogram starts to resemble the function

2) Exponentially Distributed Random Variables

   a) INTRODUCTION:

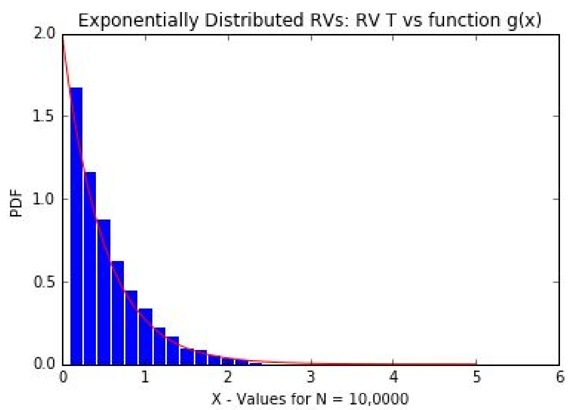
      i) This section we plot the histogram of one RV with the curve of the same RV but as a function. We superimpose the curve on top of the histogram to see how they are related.

   b) METHODOLOGY:

      i) For the histogram we generate random data to save to a list and generate the probability histogram from f(t). Then we plot the function g(x) on top of the other function and compare both graphs.

   c) RESULTS AND CONCLUSIONS:

```
In [15]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
```



Conclusion: The curve and the histogram almosts matches

3) Distribution of the Sum of RVs

   a) INTRODUCTION:

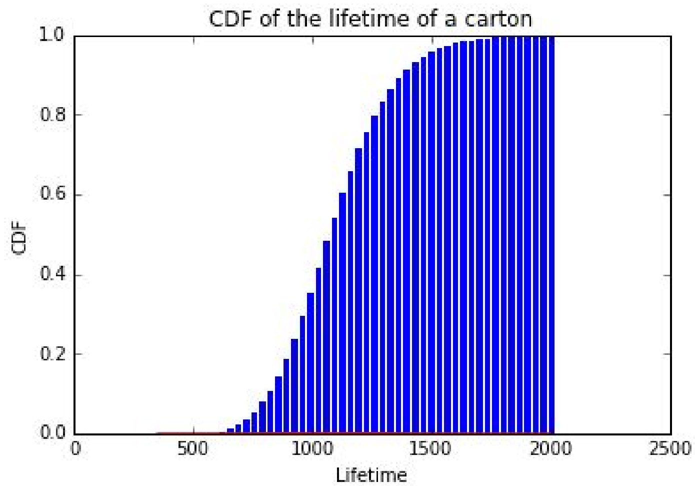
      i) We graph the following equation f_t(t;B) on the plot and try to find the probability based on that graph to answer the following questions.

   b) METHODOLOGY:

      i) We first generated 24 elements in a list. Each element is a distributed random variable known as (T). We sum them up to get the R.V. We repeat this experiment multiple times to create an experimental PDF of the lifetime of a carton.

   c) RESULTS AND CONCLUSIONS:

```
In [44]: runfile('/home/beryl/.spyder2-py3/temp.py', wdir='/home/beryl/.spyder2-py3')
```



Question	**ANS.**
1. Probability that the carton will last longer than three years	~0.4
2. Probability that the carton will last between 2.0 and 2.5 years	~0.2

4) Appendix

a) Code for #1

```
# Assignment 4 - Central Limit Theorem
# Part 1: The Central Limit Theorem
# ------------------------------------------------------------------
import math
import numpy as np
import matplotlib.pyplot as plt
def CLT(rounds,books):
    #Declaring Variables given in assignment
    mu_w = 2 # (1+3)/2 = 2
    sigma_w = 0.57 # The square root of sigma_w
    sigma_sn = sigma_w * math.sqrt(books)
    var = sigma_sn * sigma_sn
    mu_sn = books * mu_w
    #Creating list full of zeros
    BookList = np.zeros((rounds))
    #BookList = [0]*books
    # Creating Book list and filling it with random ints
    x = books * 1
    y = books * 3
    for i in range(rounds):
        for j in range(books):
            temp = np.random.uniform(x,y)
            BookList[i] = BookList[i] + temp
        BookList[i] = BookList[i] * 1/books
    print(BookList)
    # Begins plotting curves
    fig1 = plt.figure(books+1)
    bins = [float(BookList) for BookList in np.linspace(x,y, 30)]
    h1, bin_edges = np.histogram(BookList,bins,density=True)
    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]
    plt.bar(b1,h1, width = barwidth, edgecolor='w')
    # Creating Gaussian Curve
    x0 = np.linspace( books * 1, books * 3)
    fact = 1/(np.sqrt(2*np.pi*var))
    y = fact*np.exp(-(x0-mu_sn) ** 2/ (2*var))
    # Plotting
    plt.plot(x0,y, color='red')
    plt.title('PDF of books height vs Gaussian Curve')
    plt.xlabel('Books height for n=' + str(books) + ' books')
    plt.ylabel('PDF')
    fig1.show()
CLT(10000,1)
```

b) Code for #2

```
#---------------------------------------------------------------------
# Name: Roberto Sanchez
# Date: Oct 30,17
# Assignment 4 - Central Limit Theorem
# Part 2: Exponentially Distributed RVs
#  In order to run: make n = 1 and run N = 10,000 simulating RV S = W1
# -------------------------------------------------------------------
import numpy as np
import matplotlib.pyplot as plt
def ExpoDisRV(rounds):
    beta = 0.5
    # Exponentially Distributed R.V.
    T = np.zeros((rounds))
    #create Random ints
    for i in range(rounds):
        T[i] = np.random.exponential(beta)
    # Creating Bins
    bins = [float(T) for T in np.linspace(0, 5, 31)]
    # Creating Curve
    h1, bin_edges = np.histogram(T, bins, density = True)
    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]
    # Create Other Curve
    plt.bar(b1, h1, width = barwidth, edgecolor = 'w')
    x = np.linspace(0,5)
    gx = (1 / beta) * np.exp(( -1 / beta) * x)
    plt.plot(x, gx, color='red')
    plt.title('Exponentially Distributed RVs: RV T vs function g(x)')
    plt.xlabel('X - Values for N = 10,0000')
    plt.ylabel('PDF')
    plt.show()
ExpoDisRV(10000)
```

c) Code for #3

```
# Assignment 4 - Central Limit Theorem
# Part 1: The Central Limit Theorem
#  In order to run: make n = 1 and run N=10,000 and plot
#   Make n = 5, n = 10, n = 15 and plot each one
#
# ----------------------------------------------------------------------
import math
import numpy as np
import matplotlib.pyplot as plt
def DisSumRV(rounds,n):
    #Declaring Variables given in assignment
    beta =45
    mu_c = beta * n # (1+3)/2 = 2
    sigma_c = beta* math.sqrt(n) # The square root of sigma_w
    #Creating list full of zeros
    BookList = np.zeros((rounds))
    #BookList = [0]*n
    # Creating Book list and filling it with random ints
    for i in range(rounds):
        for j in range(n):
            temp = np.random.exponential(beta)
            BookList[i] = BookList[i] + temp
    # Begins plotting curves
    fig1 = plt.figure(n+1)
    bins = [float(BookList) for BookList in np.linspace(350, 2000, 50)]
    h1, bin_edges = np.histogram(BookList,bins,density=True)
    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]
    plt.bar(b1,h1, width = barwidth, edgecolor='w')

    g = np.linspace(350,2000)
    fact = 1/(np.sqrt(2*np.pi*sigma_c * sigma_c))
    y = fact*np.exp(-(g-mu_c)**2/(2*sigma_c * sigma_c))
    # Plotting
    plt.plot(g,y, color='red')
    plt.title('CDF of the lifetime of a carton')
    plt.xlabel('Lifetime')
    plt.ylabel('CDF')
    H1 = np.cumsum(h1)*barwidth
    plt.bar(b1,H1,width=barwidth,edgecolor = 'w')

    fig1.show()
DisSumRV(10000,24)
```