Ajay Kc
013213328
Project 4 Report

1. The Central Limit Theorem

Introduction

The experiment deals with random variable S, the sum of the widths of n books. The thickness W of each book ranges from 1 to 3 cm. The mean of the thickness is 2 and the standard deviation of the thickness is 0.57. The values of n are 1,5,10,and 15.

Methodology

For each value of n, 10,000 uniform numbers between 1 and 3 is generated randomly. The histogram of the random variable is generated. Finally, the normal distribution probability function, $f(x) = \frac{1}{\sigma s\sqrt{2\pi}} \exp\{-\frac{(x-\mu s)^2}{2\sigma s^2}\}$ is plotted on the same graph.

Source code

```
"""
Ajay Kc
013213328
EE381
Project 4 Part 1

The problem plots a probability distribution function of S,
where S is a Random Variable of sum
of the widths of n books. The value of n is 1,5,10,and 15.
For each value of n, the experimental
PDF and normal distribution function is calculated.
"""
import numpy as np
import math
import matplotlib.pyplot as plt
mean = (1+3)/2
standard_deviation = math.sqrt(((3-1)**2)/12)

def getRVMean(n):
    return n*mean

def getRVSd(n):
    return standard_deviation*math.sqrt(n)

def simulateRV(n):

    RV = []
```

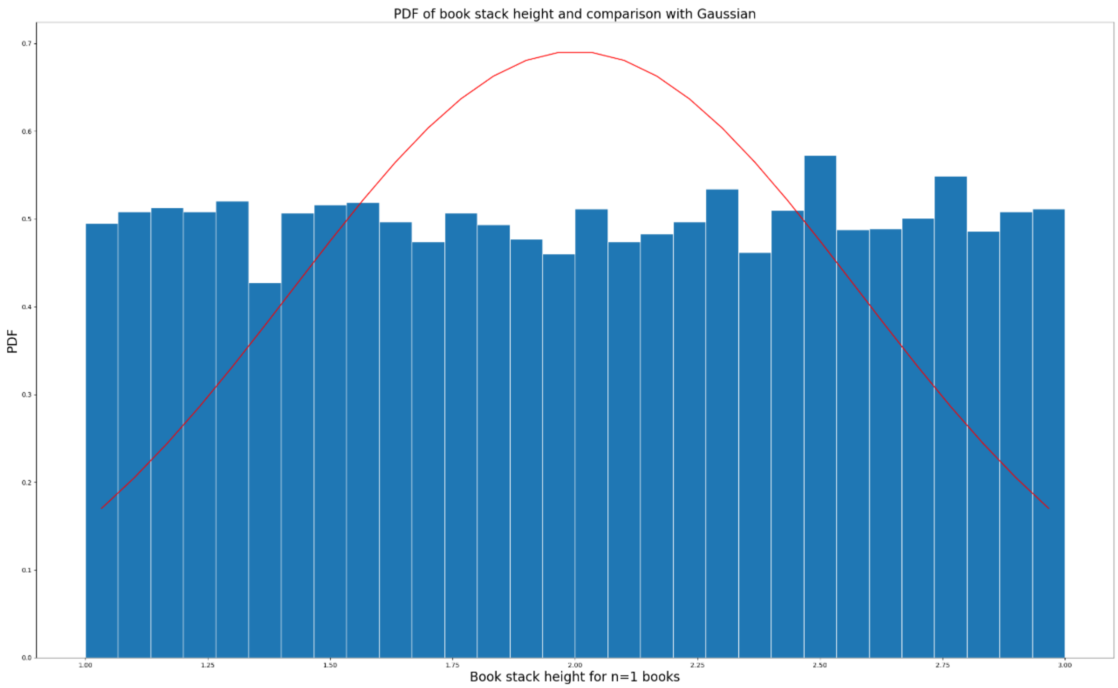
```
    for i in range(0,10000):
        thickness = np.random.uniform(1,3,n)
        RV.append(thickness.sum())
    return RV

def plotFunctions(RV,n):
    a = 1; b =3 #a = min width; b = max width
    nbooks =n; nbins=30; #Number of books; Number of bins
    edgecolor = 'w';     #Color seperating bars in the
bargraph
    #
    #Create bins and histogram
    bins = [float(x) for x in np.linspace(nbooks*a,
nbooks*b, nbins+1)]
    h1, bin_edges = np.histogram(RV, 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

    figureOne = plt.figure(1)
    xlabel = "Book stack height for n="+str(n)+" books"
    plt.xlabel(xlabel, fontsize = 20)
    plt.ylabel("PDF", fontsize = 20)
    plt.title("PDF of book stack height and comparison with
Gaussian",fontsize=20)
    plt.bar(b1,h1,width = barwidth, edgecolor=edgecolor)

    normalDist = []
    for i in range(0,len(b1)):
        normalDist.append((1 / (getRVSd(n) * math.sqrt(2 *
math.pi))) * math.exp(
            -((b1[i] - getRVMean(n)) ** 2) / (2 *
getRVSd(n) ** 2)))

    plt.plot(b1,normalDist,color='r')
    plt.show()

#For 1 book
RV = simulateRV(1)
plotFunctions(RV,1)

#For 5 books
RV = simulateRV(5)
plotFunctions(RV,5)
```

```
#For 10 books
RV = simulateRV(10)
plotFunctions(RV,10)

#For 15 books
RV = simulateRV(15)
plotFunctions(RV,15)
```

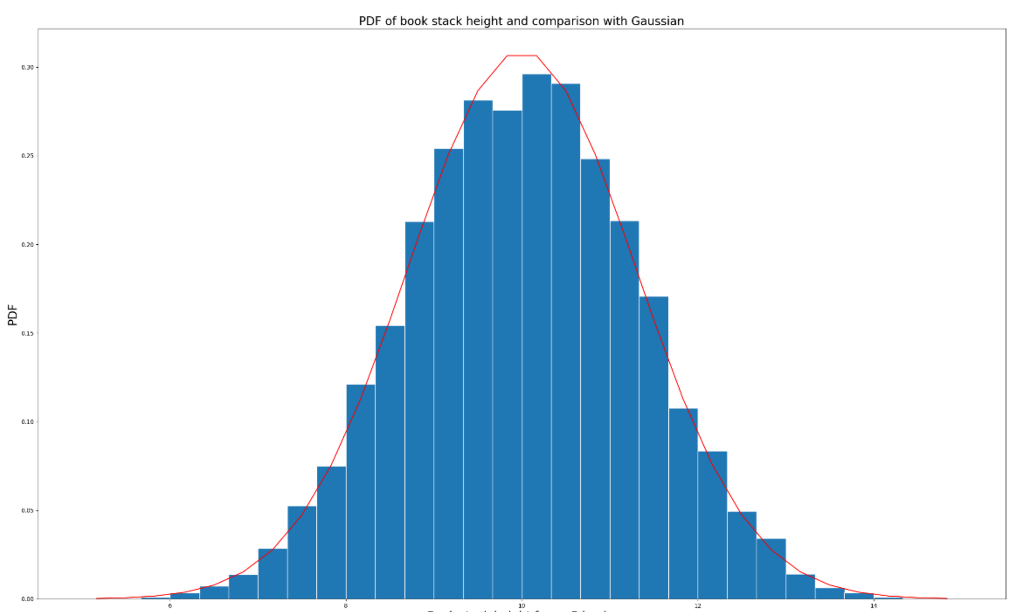
Ajay Kc
013213328
Project 4 Report

Histogram Plot with overlapping normal probability distribution

When n = 1

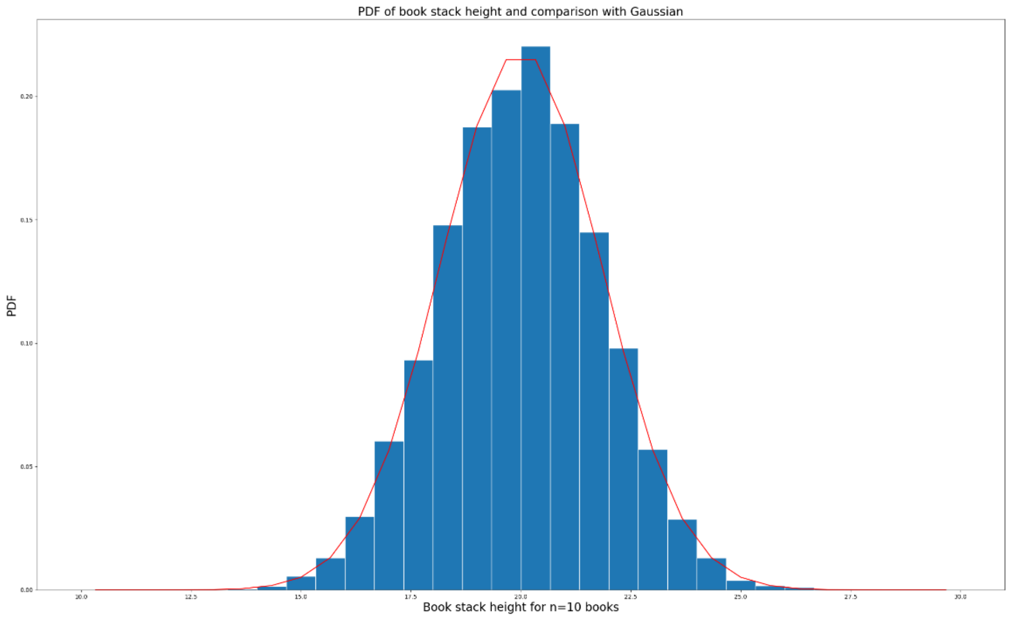


When n = 5

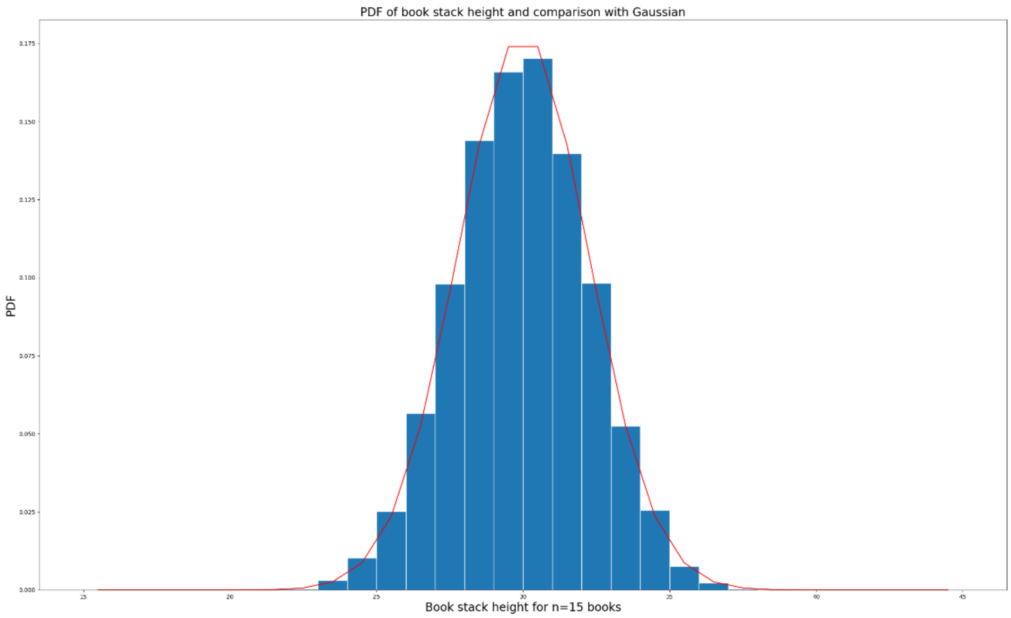


Ajay Kc
013213328

When n = 10



When n =15



Ajay Kc
013213328
Project 4 Report

2. Exponentially Distributed Random Variable

Introduction

The objective of the experiment is to simulate an exponentially distributed random variable with the pdf, $f(T) = 2\exp(-2T)$.

Methodology

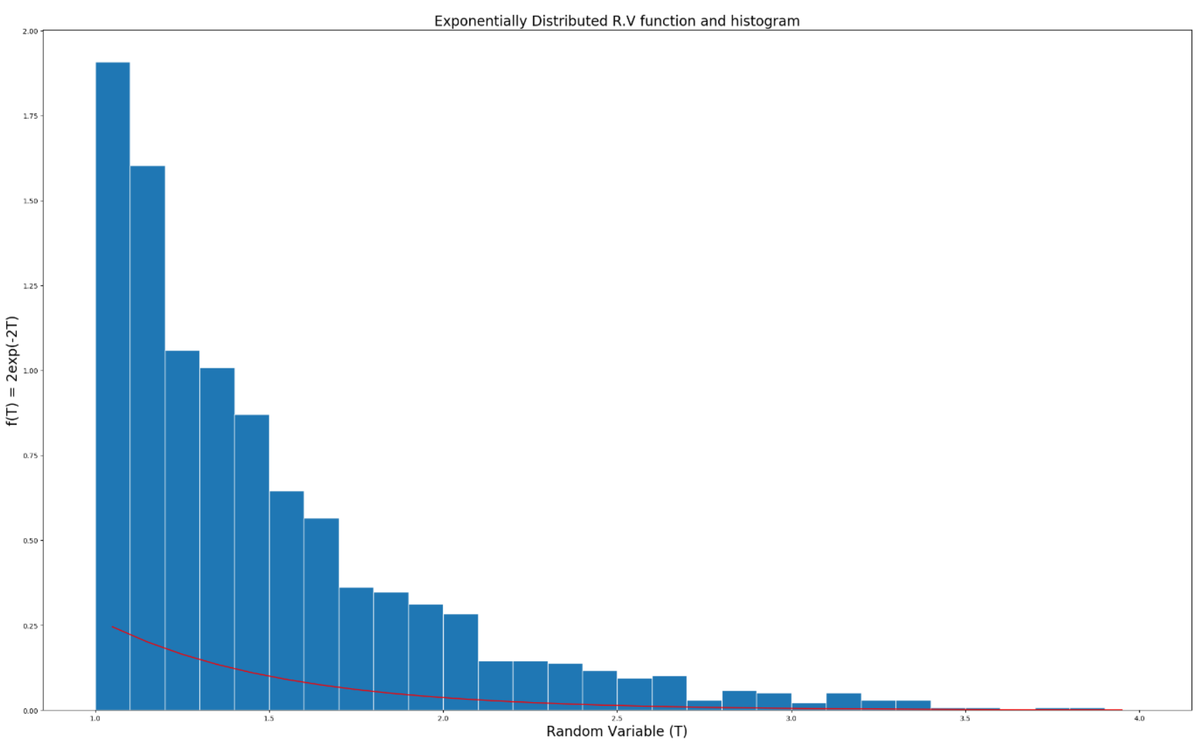
The experiment will randomly generate 10,000 values with the scalar factor of 0.5 and the probability histogram of the random variable T will be plotted. On the same graph, the function, $f(T) = 2\exp(-2T)$ will be plotted.

Source Code

```
"""
Ajay Kc
013213328
EE381
Project 4 Part 2

The problem simulates the exponentially distributed RV and
plots the histogram for the simulated
results. Finally, it plots the plot for exponentially
distributed RV
"""

import numpy as np
import math
import matplotlib.pyplot as plt
def simulateRV():

    RV = np.random.exponential(0.5,10000)
    return RV

def plotFunctions(RV):
    nbins=30; # Number of bins
    edgecolor = 'w';    #Color seperating bars in the
bargraph
    #
    #Create bins and histogram
    bins = [float(x) for x in np.linspace(1, 4, nbins+1)]
    h1, bin_edges = np.histogram(RV, 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

    figureOne = plt.figure(1)
    xlabel = "Random Variable (T)"
    plt.xlabel(xlabel, fontsize = 20)
    plt.ylabel("f(T) = 2exp(-2T)", fontsize = 20)
    plt.title("Exponentially Distributed R.V function and
histogram",fontsize=20)
    plt.bar(b1,h1,width = barwidth, edgecolor=edgecolor)

    exponentialDist = []
    for i in range(0, len(b1)):
        exponentialDist.append(2*math.exp(-2*b1[i]))
    plt.plot(b1, exponentialDist, color='r')

    plt.show()

RV = simulateRV()
plotFunctions(RV)
```

<u>Histogram of R.V with the overlapping function for exponential distribution</u>



3. Distribution of the Sum of RVs

Introduction

The problem deals with the lifetime of a carton with 24 batteries. The lifetime of the battery is a RV with an exponentially distributed lifetime. The mean of the battery is 45. The objective of the experiment is to simulate the RV representing the lifetime of a carton.

Methodology

For each 24 batteries, a exponentially distributed random variable (T) is simulated. The random variable C is the sum of each T for 24 batteries. The whole experiment is repeated 10,000 times and the experimental PDF of the lifetime of a carton is plotted. Moreover, the normal distribution is also plotted in the same graph. Finally, the CDF of the lifespan of a carton is also plotted in a different graph.

Source code

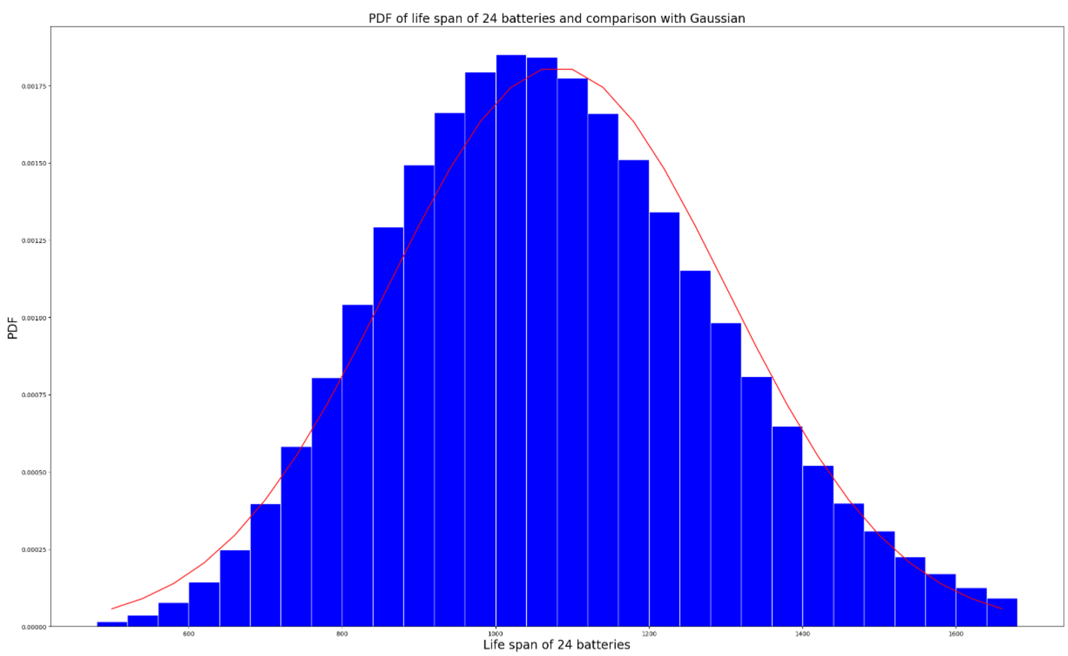
```
"""
Ajay Kc
013213328
EE381
Project 4 Part 3

The problem simulates the lifespan of a carton with 24
batteries and plots the histogram
for the simulated results.
"""
import numpy as np
import matplotlib.pyplot as plt
import math as math
mean = 45
standard_deviation = 45

def getRVMean(n):
    return n*mean

def getRVSd(n):
    return standard_deviation*math.sqrt(n)

def simulateExpRV():
    expRV = np.random.exponential(45,100)
    return expRV
```

Ajay Kc
013213328


```
def simulateSumExpEV():
    carton = []
    sumRV = []

    for i in range (0,24):
        carton.append(simulateExpRV())

    sumRV = [sum(x) for x in zip(*carton)]

    return sumRV

def plotFunctions(RV):
    nbins=30; # Number of bins
    nbatteries = 24
    a=20
    b=70
    edgecolor = 'w';    #Color seperating bars in the
bargraph
    #
    #Create bins and histogram
    bins = [float(x) for x in np.linspace(nbatteries*a,
nbatteries*b, nbins+1)]
    h1, bin_edges = np.histogram(RV, 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

    figureOne = plt.figure(1)
    xlabel = "Life span of 24 batteries"
    plt.xlabel(xlabel, fontsize = 20)
    plt.ylabel("PDF", fontsize = 20)
    plt.title("PDF of life span of 24 batteries and
comparison with Gaussian",fontsize=20)
    plt.bar(b1,h1,width = barwidth, edgecolor=edgecolor,
color = 'b')

    normalDist = []
    for i in range(0, len(b1)):
        normalDist.append((1 / (getRVSd(24) * math.sqrt(2 *
math.pi))) * math.exp(
            -((b1[i] - getRVMean(24)) ** 2) / (2 *
getRVSd(24) ** 2)))

    plt.plot(b1, normalDist, color='r')
    plt.show()
```

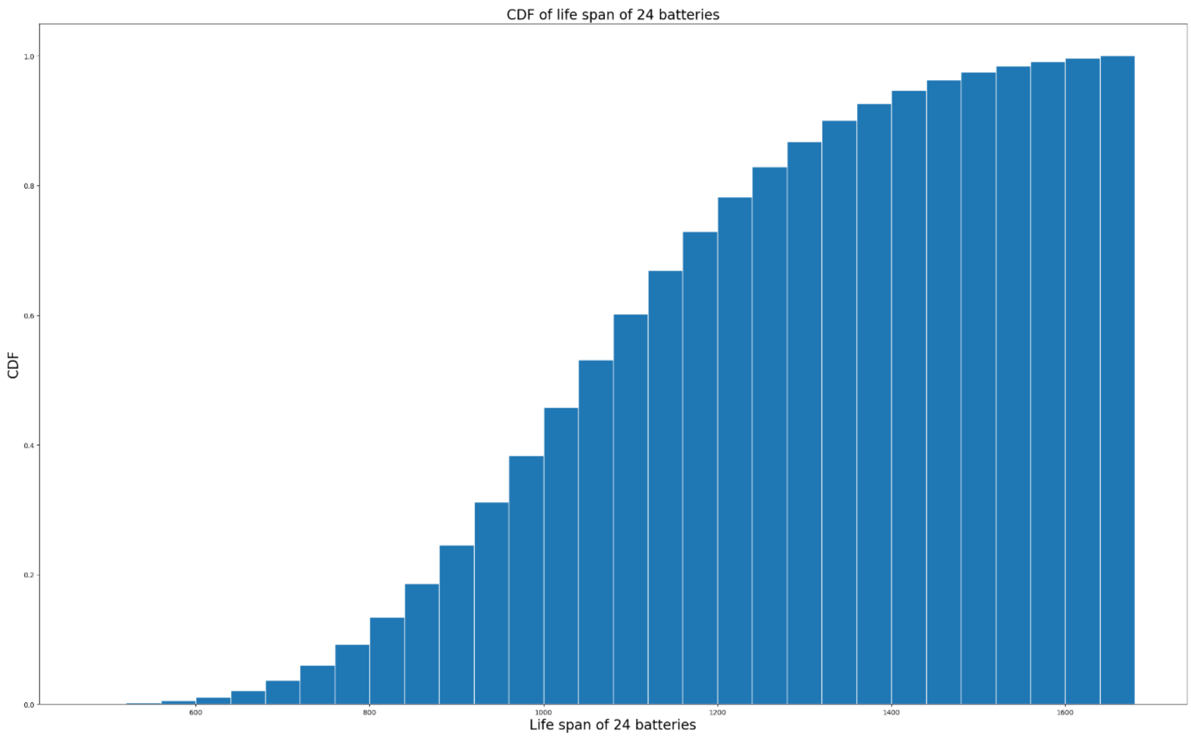
```
    plt.close()

    H1 = np.cumsum(h1)*barwidth
    xlabel = "Life span of 24 batteries"
    plt.xlabel(xlabel, fontsize=20)
    plt.ylabel("CDF", fontsize=20)
    plt.title("CDF of life span of 24 batteries",
fontsize=20)

    plt.bar(b1,H1,width = barwidth, edgecolor=edgecolor)
    plt.show()
    plt.close()

mergedList = []
for i in range(0,10000):
    mergedList += simulateSumExpEV()

plotFunctions(mergedList)
```

<u>PDF Plot with the overlapping normal distribution</u>



Ajay Kc
013213328
Project 4 Report

CDF Plot



QUESTION	ANS
1. Probability that the cartoon will last longer than three years	0.4
2. Probability that the carton will last between 2.0 and 2.5 years	0.19