|  |
| --- |
| **Project 4, Normal (Gaussian) Distribution** |
| EE381, Probability and Statistics Computing |
| Instructor: Duc Tran  Group members: Thanh Nguyen, Alyssa Faiferlick  Due Date: December 3rd, 2020 |

****

|  |
| --- |
| **Question**  **#1**  **Normal Distribution** |

1. **Introduction**

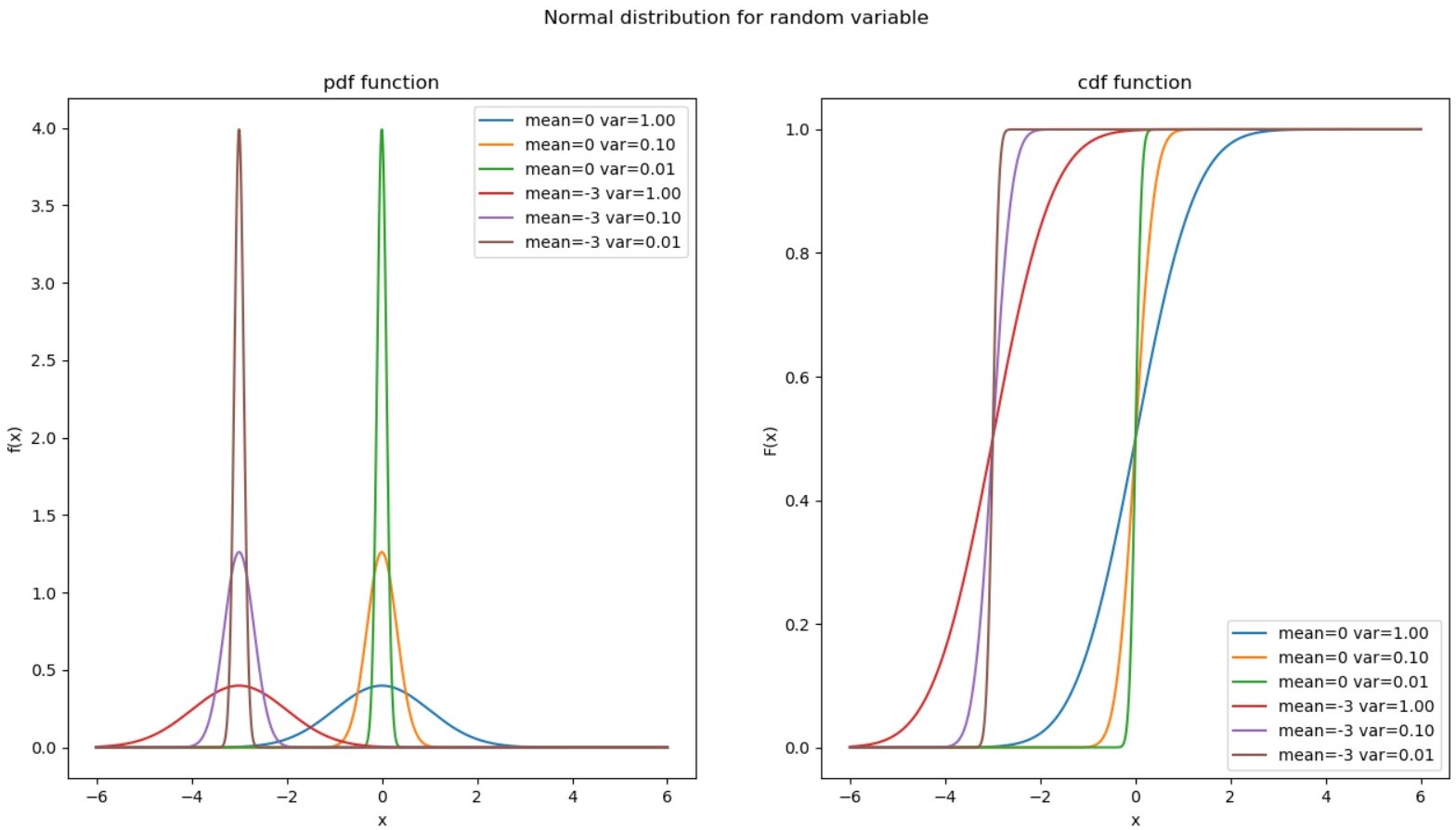
Write two Python functions (the probability density function and cumulative distribution function) of normal distribution the given equations. Then, plot all pdf and cdf curves with different values of mean and standard deviation.

1. **Procedure**

* Generate a list of x from -6 to 6 with the number of elements is 100000 using numpy.linspace()
* For the density function, we used the formula below to calculate f(x) with every mean and standard deviation. The inputs of this function are x, (mu), and (sigma\_2).
* For the cumulative distribution function, we used the error function (erf) from math package in Python to calculate F(x). Then, we used the below equation to get the results. The inputs for this function are x, (mu), and (sigma\_2).
* Then, we plotted these curses by using plot().

1. **Results and Discussion**

* Results:

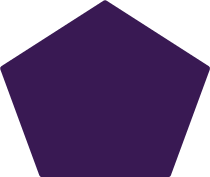


* These curses are what we expected.

1. **Conclusion**

* For two normal distributions with the same mean values but different standard deviations, the larger standard deviations produce distributions that are more spread out. In the cumulative distribution curves, the bigger standard deviations make the value of x which is close to mean changes more quickly => The standard deviation defines the width of the normal distribution. It determines how far away from the mean the values tend to fall.
* For two normal distributions with the same standard deviation but with different means, these curves have the same shapes, but they are located at different positions on the x axis => The mean defines the location of the peak for normal distributions.

1. **Appendix**
2. **import** matplotlib**.**pyplot **as** plt
3. **from** scipy**.**stats **import** norm
4. **import** numpy **as** np
5. **import** math
6. # Create cdf of a normal random variable
7. **def** norm\_cdf**(**x**,** mu**,** var**):**
8. **return** 0.5 **\*** **(**1 **+** math**.**erf**((**x **-** mu**)/**math**.**sqrt**(**2**\***var**)))**
9. # Create pdf of a uniform random variable
10. **def** norm\_pdf**(**x**,** mu**,** sigma\_2**):**
11. f **=** np**.**exp**(-** **(**x **-** mu**)\*\***2 **/** **(**2 **\*** sigma\_2**))** **/** **(**math**.**sqrt**(**2 **\*** np**.**pi **\*** sigma\_2**))**
12. **return** f
13. # Genrating the list of x
14. x **=** np**.**linspace**(-**6**,** 6**,** 100000**)**
15. mean **=** **[**0**,** 0**,** 0**,** **-**3**,** **-**3**,** **-**3**]**
16. variance **=** **[**1**,** 0.1**,** 0.01**,** 1**,** 0.1**,** 0.01**]**
17. # Plotting
18. # divide the frame into 4 part: part a is in row 1, and part b is in row 2
19. fig**,** **(**ax1**,** ax2**)** **=** plt**.**subplots**(**1**,** 2**)**
20. # Set title and label for graphs
21. plt**.**suptitle**(**"Normal distribution for random variable"**)**
22. ax1**.**set\_title**(**"pdf function"**)**
23. ax2**.**set\_title**(**"cdf function"**)**
24. ax1**.**set\_xlabel**(**'x'**)**
25. ax2**.**set\_xlabel**(**'x'**)**
26. ax1**.**set\_ylabel**(**'f(x)'**)**
27. ax2**.**set\_ylabel**(**'F(x)'**)**
28. # Plotting pdf
29. **for** i **in** **range(**6**):**
30. ax1**.**plot**(**x**,** **[**norm\_pdf**(**x\_i**,** mean**[**i**],** variance**[**i**])** **for** x\_i **in** x**],** label **=** 'mean=%.d var=%.2f' **%(**mean**[**i**],** variance**[**i**]))**
31. # Plotting cdf
32. **for** i **in** **range(**6**):**
33. ax2**.**plot**(**x**,** **[**norm\_cdf**(**x\_i**,** mean**[**i**],** variance**[**i**])** **for** x\_i **in** x**],** label **=** 'mean=%.d var=%.2f' **%(**mean**[**i**],** variance**[**i**]))**
34. ax1**.**legend**()**
35. ax2**.**legend**()**
36. plt**.**show**()**

****

**Question**

**#2**

**Central Limit Theorem**

**I. Introduction**

Assuming 𝑋1, 𝑋2, …, 𝑋𝑛 are independent random variables having the same probability distribution with mean 𝜇 and standard deviation 𝜎, consider the sum 𝑆𝑛 = 𝑋1 + 𝑋2 + ⋯ + 𝑋𝑛.

This sum 𝑆𝑛 is a random variable with mean 𝜇𝑆𝑛 = 𝑛𝜇 and standard deviation 𝜎𝑆𝑛 = 𝜎√𝑛.

The Central Limit Theorem states that as the probability distribution of the random variable 𝑆𝑛 will approach a normal distribution with mean 𝜇𝑆𝑛 and standard deviation , regardless of the original distribution of the random variables 𝑋1, 𝑋2, …, 𝑋𝑛.

It is noted that the PDF of the normally distributed random variable 𝑆𝑛 is given by:

**:**

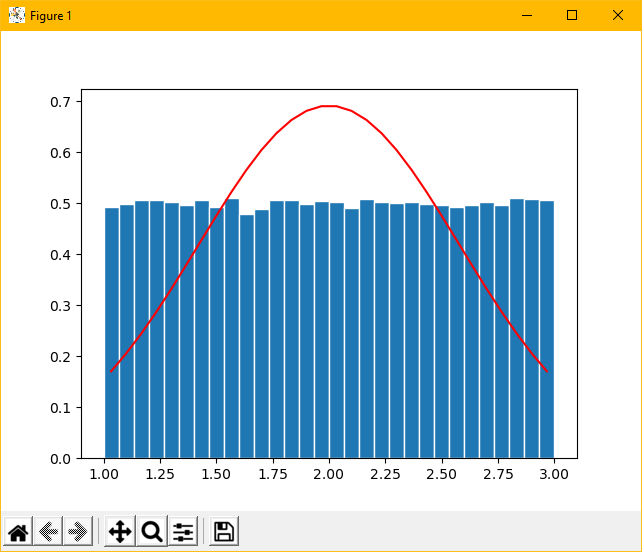
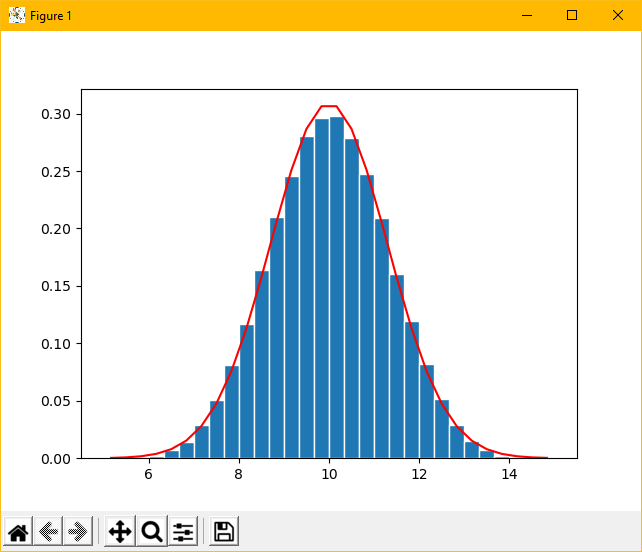
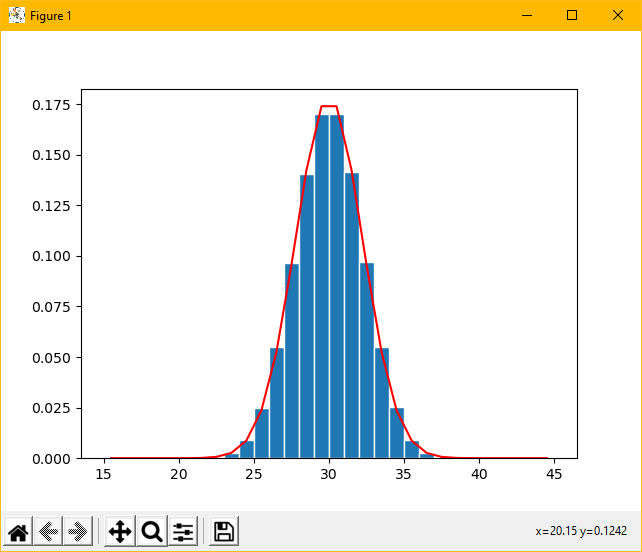
**II. Procedure**

* Generate N = 10,000 different values for the width of the book, which is between [1,3]. Assign these values to S.
* Plot the Histogram of S along with the Normal Probability Function of by using:
* Plot them on the same figure to compare the two results.
* Repeat these steps with n = [1,5,15].

**III. Results and Discussion**

|  |  |
| --- | --- |
| Mean Thickness of a Single Book (cm) | Standard Deviation of Thickness (cm) |
| 2 | 0.577 |

|  |  |  |
| --- | --- | --- |
| Number of Books n | Mean Thickness of a Stack  of n Books (cm) | Standard Deviation of the  Thickness for n Books |
| 1 | = 2 | = 0.577 |
| 5 | = 10 | = 1.290 |
| 15 | = 30 | = 2.234 |

1 Histogram 5 Histogram 15 Histogram

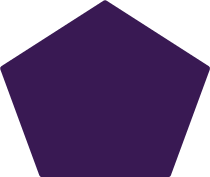
**IV. Conclusion**

This problem helped us better understand the Central Limit Theorem. The results show that the individual distributions of an RV do not look anything like Gaussian, but when enough of the identical RV are together, the result then represents a Gaussian with a mean equal to the sum of the individual of the RV, and a standard deviation equal to the square root of the sum times the individual RV standard deviation.

The code the Professor provided for this question is simple to follow and understand, we had no troubles reading and understanding the code.

**V. Appendix**

****

****

**Question**

**#3**

**Distribution of the Sum of Exponential Random Variables**

1. **Introduction**

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 . The mean and variance of the random variable T are: .

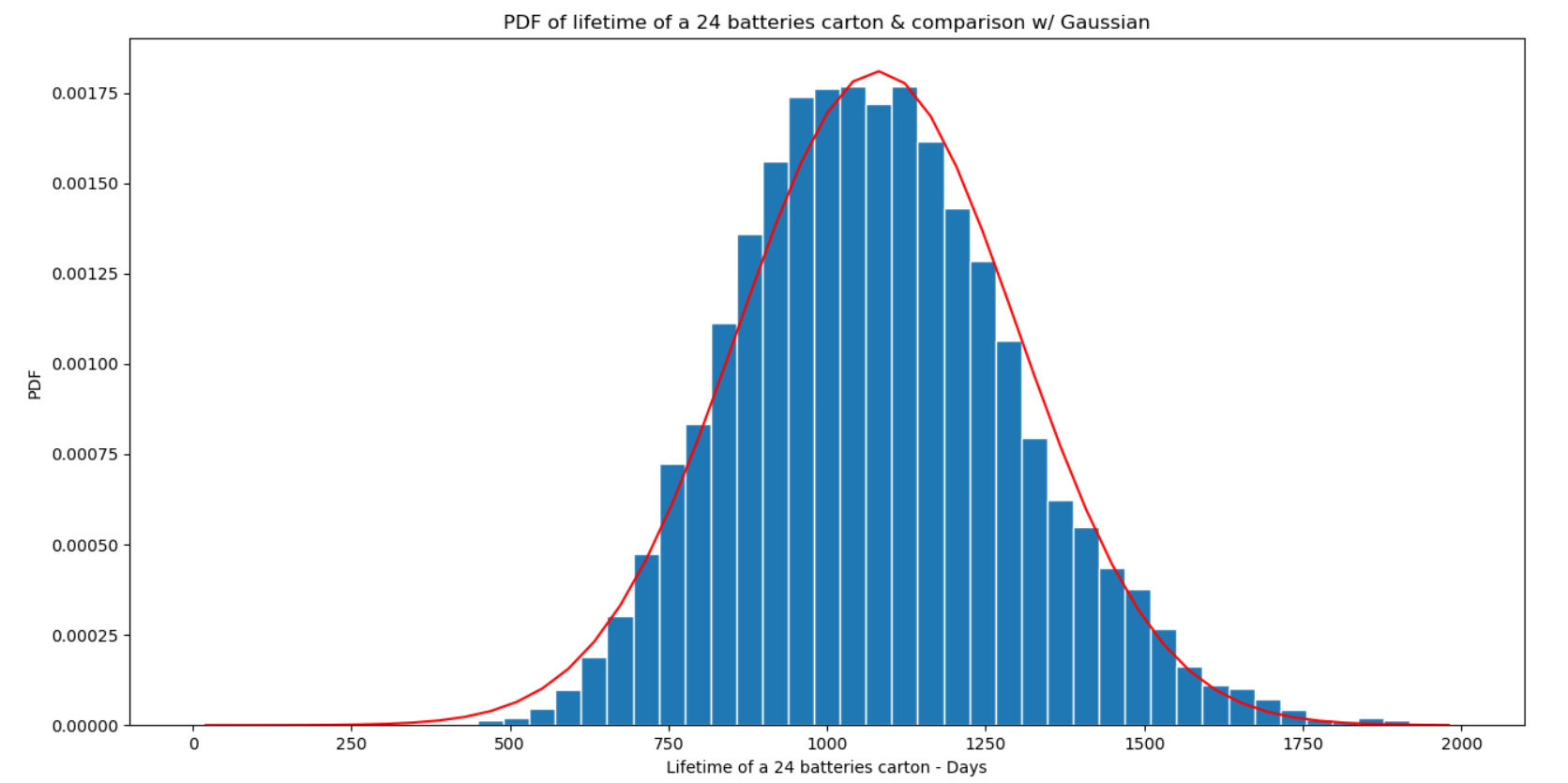
When a battery fails it is replaced immediately by a new one. Batteries are purchased in a carton of 24. The objective is to simulate the RV representing the lifetime of a carton of 24 batteries, and create a histogram.

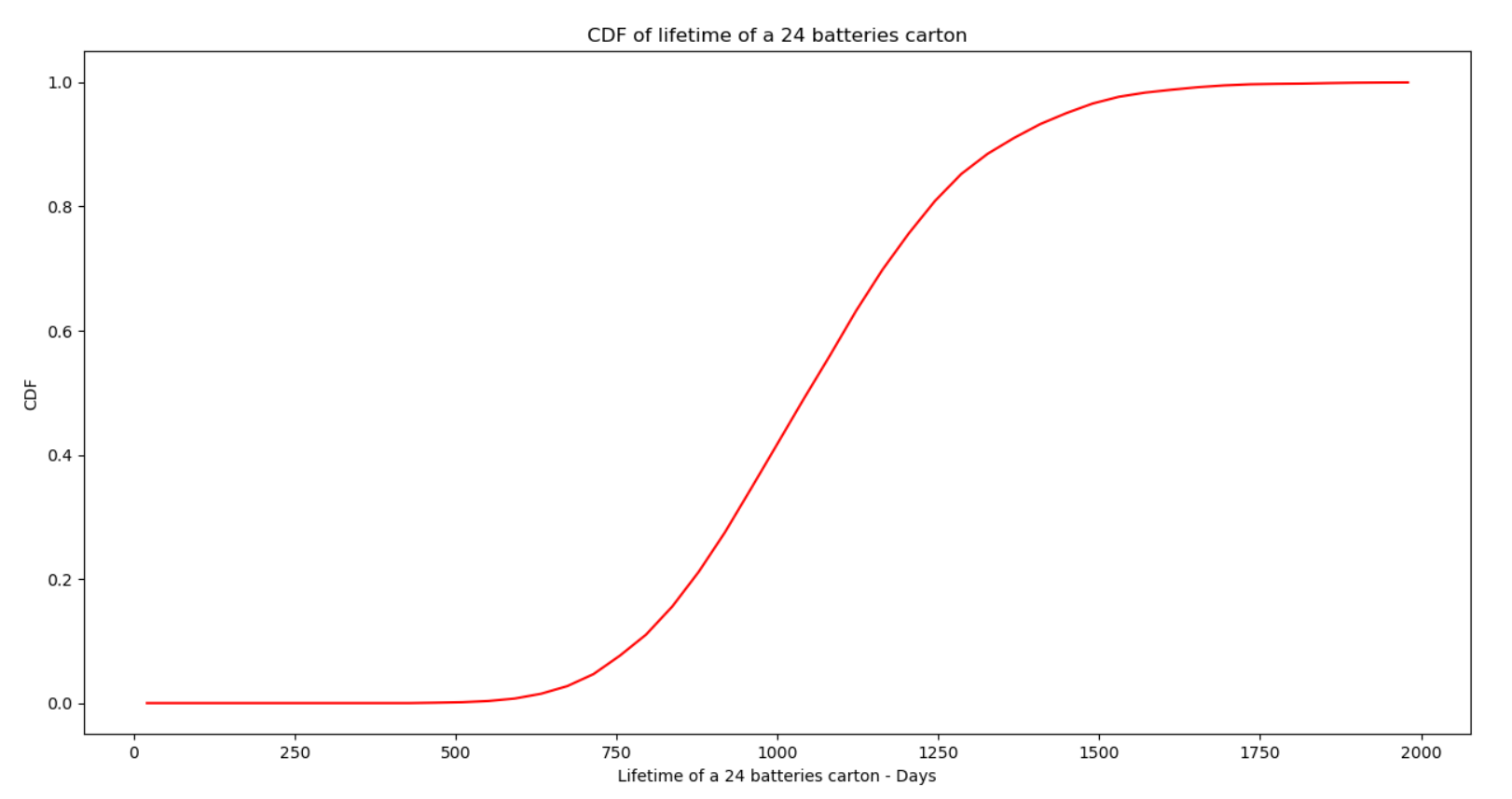
1. **Procedure**

* For part a, b and c:
  + Generate N = 10,000 cartons randomly. Each carton is a vector of 24 elements. Each one of the 24 elements in the vector is an exponentially distributed random variable (T), with mean lifetime equal to . Generate 24 elements randomly by using the Python function ***numpy.random.exponential(beta, n)*** with n = 24.
  + Then, calculate the lifetime (C) of each carton by getting the sum of 24 elements of each vector, and store this value in a list called X.
* For part d: According to the Central Limit Theorem the PDF for one carton of 24 batteries can be approximated by a normal distribution. Plot the histogram of X, and then plot the PDF with using the normal PDF function on the same figure to compare the results.
* For part e: The CDF is plotted by using the cumulative sum of the PDF and multiplying it with barwidth, which produces F(c).

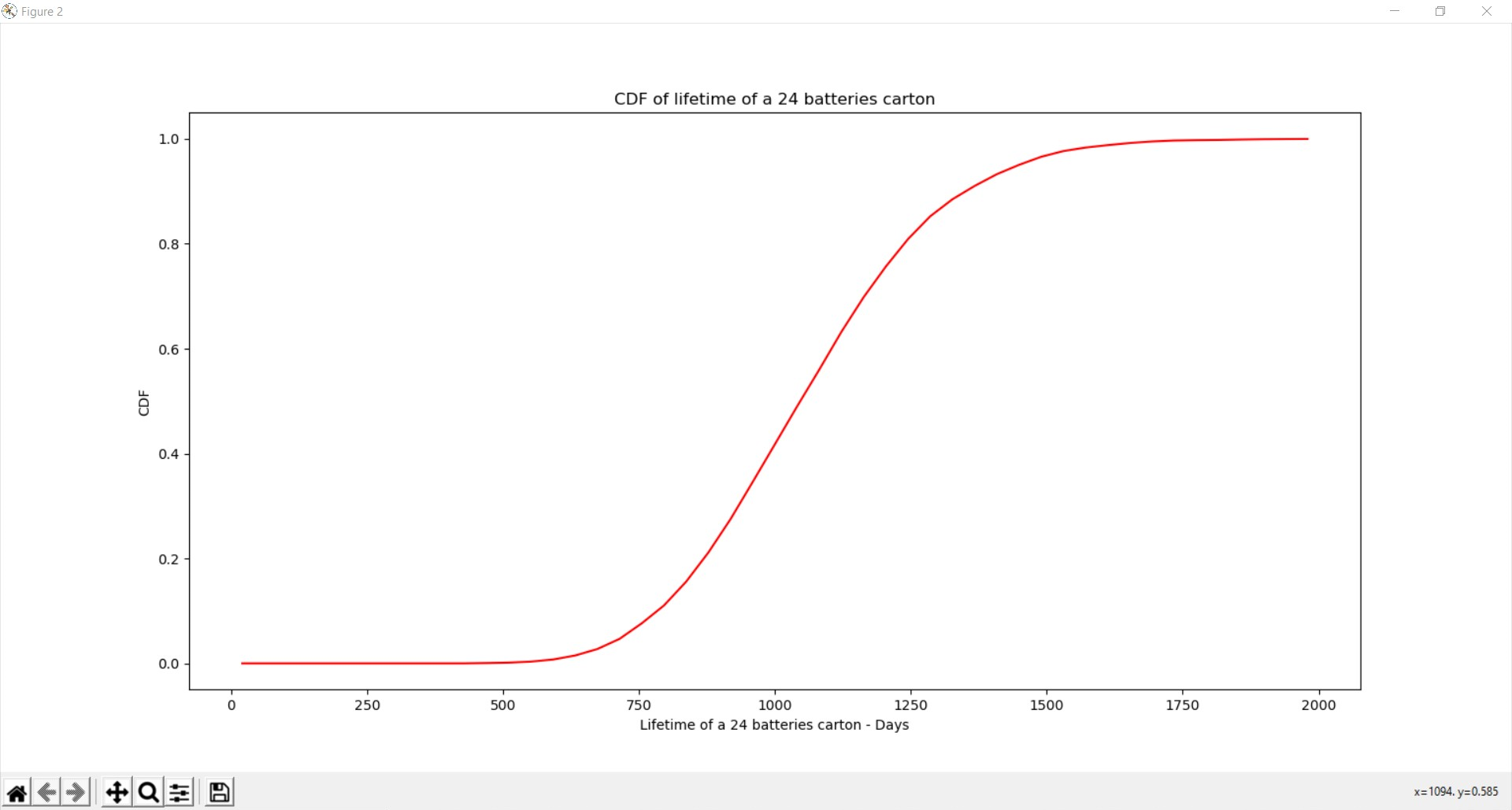
1. **Results and Discussion**

* Results:

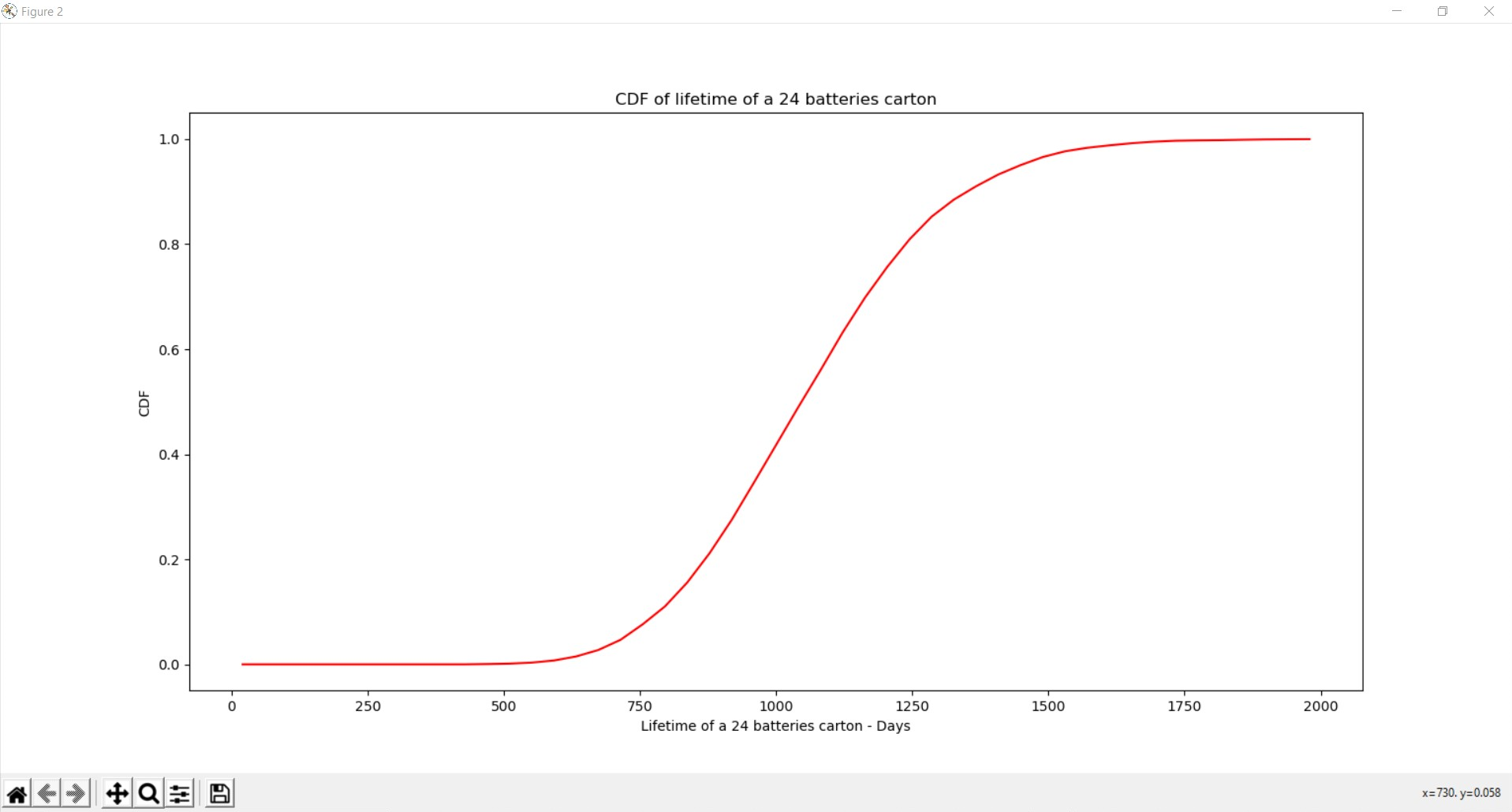


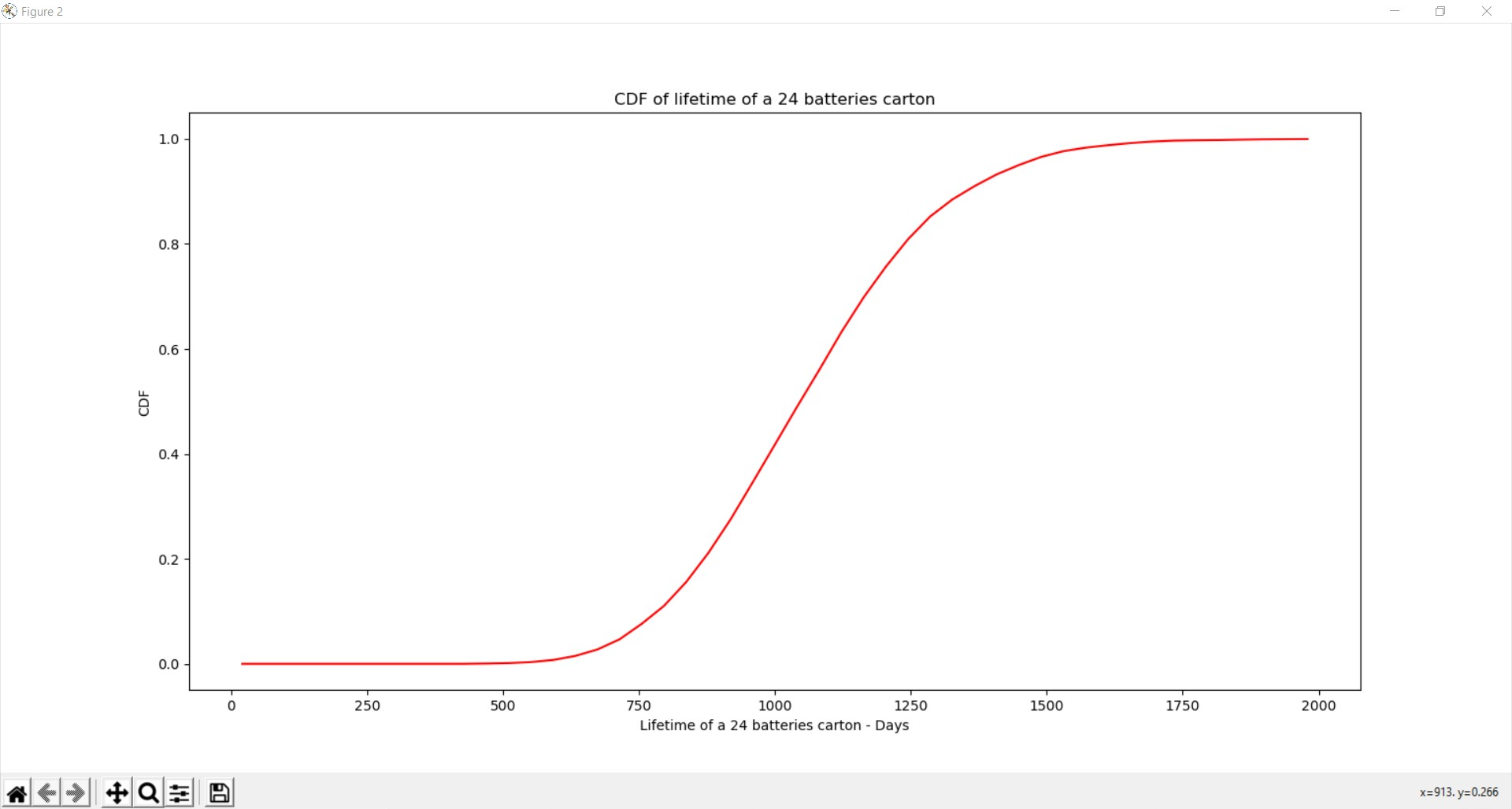


* To answer the two questions, we got:



* The probability that the carton will last longer than three years is:





* The probability that the carton will last between 2.0 and 2.5 years (i.e between 730 and 912 days) is:

1. **Conclusion**

To simulate the RV representing the lifetime of a carton of 24 batteries, we followed the steps that the Professor provided. These steps are clear and easy to follow. We also used the code of part 2 for this problem, just changed a little bit to make the code meet all requirements, and used numpy.random.exponential(beta, n) function in Python to generate the exponentially distributed random variable T.

The results show that the PDF for one carton of 24 batteries is approximated by a normal distribution with mean and standard deviation .

1. **Appendix**
2. **import** numpy **as** np
3. **import** math
4. **import** matplotlib**.**pyplot **as** plt
5. # Generate the values of the RV X
6. N **=** 10000**;**
7. beta **=** 45**;**
8. n **=** 24**;**
9. # Part a, b and c -
10. # Create a vector of 24 elements
11. # Calculate the sum of the elements in this vector
12. # Repeat this experiment for a total 10000 times
13. mu\_T **=** beta**;**
14. sigma\_T **=** beta**;**
15. X **=** np**.**zeros**((**N**,**1**));**
16. **for** i **in** **range(**0**,**N**):**
17. x **=** np**.**random**.**exponential**(**beta**,** n**);**
18. w **=** np**.sum(**x**);**
19. X**[**i**]** **=** w**;**
20. # Part d and e
21. # Calculate the average and standard deviation
22. mu\_c **=** 24 **\*** beta**;**
23. sigma\_c **=** beta **\*** np**.**sqrt**(**24**);**
24. # Create bins and histogram
25. nbins **=** 50**;** # Number of bins
26. edgecolor **=** 'w'**;** # Color separating bars in the bargraph
27. bins **=** **[float(**x**)** **for** x **in** np**.**linspace**(**0**,** 2000**,** nbins**)];**
28. h1**,** bin\_edges **=** np**.**histogram**(**X**,** bins**,** density **=** **True);**
29. # Define points on the horizontal axis
30. be1 **=** bin\_edges**[**0 **:** np**.**size**(**bin\_edges**)** **-** 1**];**
31. be2 **=** bin\_edges**[**1 **:** np**.**size**(**bin\_edges**)];**
32. b1 **=** **(**be1 **+** be2**)** **/** 2**;**
33. barwidth **=** b1**[**1**]** **-** b1**[**0**];** # Width of bars in the bargraph
34. plt**.**close**(**'all'**);**
35. # PLOT THE BAR GRAPH
36. fig1 **=** plt**.**figure**(**1**);**
37. plt**.**bar**(**b1**,** h1**,** width **=** barwidth**,** edgecolor **=** edgecolor**);**
38. # Part d
39. **def** NormPDF**(**mu**,** sigma**,** x**):**
40. f **=** **((**1 **/** **(**sigma **\*** np**.**sqrt**(**2 **\*** math**.**pi**))** **\*** np**.**exp**((-**1 **\*** **((**x**-**mu**)\*\***2**))** **/** **(**2 **\*** **(**sigma**\*\***2**)))** **\*** np**.**ones**(**np**.**size**(**x**))));**
41. **return** f**;**
42. # Plot PDF
43. f **=** NormPDF**(**mu\_c**,** sigma\_c**,** b1**);**
44. plt**.**plot**(**b1**,** f**,** 'r'**);**
45. plt**.**title**(**'PDF of lifetime of a 24 batteries carton & comparison w/ Gaussian'**);**
46. plt**.**xlabel**(**'Lifetime of a 24 batteries carton - Days'**);**
47. plt**.**ylabel**(**'PDF'**);**
48. fig1**.**show**();**
49. # Plot bar graph
50. fig2 **=** plt**.**figure**(**2**);**
51. # Part e
52. F **=** np**.**cumsum**(**h1**)** **\*** barwidth**;**
53. plt**.**plot**(**b1**,** F**,** 'r'**);**
54. #plt.bar(b1, F, width = barwidth, edgecolor = edgecolor);
55. plt**.**title**(**'CDF of lifetime of a 24 batteries carton'**);**
56. plt**.**xlabel**(**'Lifetime of a 24 batteries carton - Days'**);**
57. plt**.**ylabel**(**'CDF'**);**
58. fig2**.**show**();**