Problem 1. Bus Lines (#Probability)

Maria arrives at a bus stop at a uniformly random time. When Maria arrives at the bus stop, either of two independent bus lines (both of which can take her home) may come by. Company A's bus arrival times are exactly 10 minutes apart, whereas the time from one Company B bus to the next is distributed as $Expo(\frac{1}{10})$.

1. ***What is the probability that Company B's bus arrives first?***

We know that Company A's bus always arrives 10 minutes apart. Distributing the arrival times, the probability is uniformly distributed, as we do not know how long it has been since the last bus arrived. Since arrival times are exactly 10 minutes apart, we can use $Unif(0,\ 10)$ to model Company A. This can be plotted using a simulation, shown below:

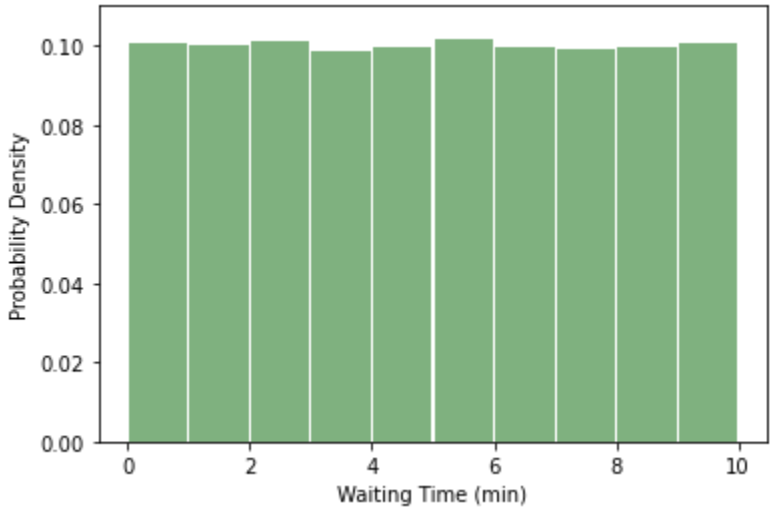


Figure 1: *Histogram of Company A's Expected Waiting Times*. This displays the uniform distribution of the likelihood of arrival during each minute, represented by a bin. Each bin has a probability density of ~0.10 once simulated with 100,000 trials. Code in Appendix.

We know that waiting time for Company B's bus is exponentially distributed as $Expo(\frac{1}{10})$, visualized below using simulated trials on the same scale as Company A:

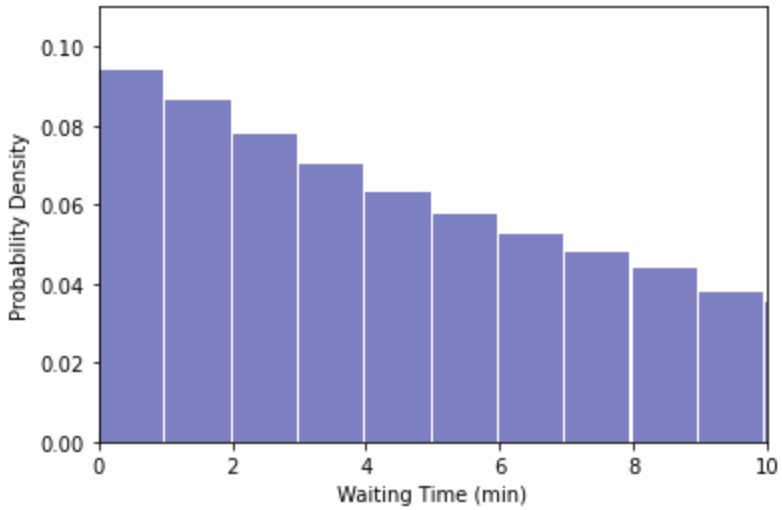


Figure 2: *Histogram of Company B's Expected Waiting Times.* This displays the exponential distribution of Company B on the scale of which Company A's times were plotted above. Though the exponential shape is not extremely visible, this provides scale to visually compare both A and B. Code in Appendix A.

In finding the probability that Company B's bus arrives before Company A's, we are finding $P(B < A)$, or the probability that the waiting time for B is less than the waiting time for A. We can use the continuous Law Of Total Probability to solve this problem, where $A \sim Unif(0,\ 10)$ and $B \sim Expo(\frac{1}{10})$:

$$P(B < A) = \int_{-\infty}^{\infty} P(B < A \mid A = a)\ f_A(a)\, da$$

Plugging in our known, f_A(a) using the Exponential and Uniform property:

$$P(B < A) = \int_{0}^{10} P(B < a \mid A = a) \mid log(Expo(\tfrac{1}{10}))\mid da$$

$$P(B < A) = \int_{0}^{10} P(B < a \mid A = a)\tfrac{1}{10}\, da$$

Deriving $P(B < a \mid A = a)$ to then solve the integral:

$$P(B < a \mid A = a) = 1 - P(B > A) = 1 - Expo(\tfrac{1}{10}) = 1 - e^{-a/10}$$

Substituting $1 - e^{-a/10}$ for $P(B < a \mid A = a)$:

$$P(B < A) = \frac{1}{10} \int_0^{10} (1 - e^{-a/10})\, da$$

Integrating:

$$P(B < A) = \frac{1}{10} \left[(a + 10e^{-a/10})\right]_0^{10} da$$

$$P(B < A) = \frac{1}{10} \left[(a + 10e^{-a/10})\right]_0^{10}$$

$$P(B < A) = \frac{1}{10} (10 + 10e^{-10/10} - 0 - 10)$$

$$P(B < A) = 1 + e^{-1} - 1 = \frac{1}{e}$$

Solution:

$$P(B < A) = \frac{1}{e}$$

Therefore, the probability of Company B's bus arriving before Company A's bus is $\frac{1}{e}$. This is roughly a 36.8 percent chance, making it more likely that Company A's bus will arrive first.

2. ***What is the PDF of Maria's waiting time for a bus? Be sure to state what the support of the distribution is.***

Maria's waiting time will depend on which bus she takes first, which will be whichever is first to arrive. The bus that comes first will have a waiting time of $T = min(A, B)$. We know that $P(A > t, B > t)$ because the random variable t represents the amount of waiting time until an arrival; both the first and second bus will arrive after t minutes. This tells us:

$$P(A > t, B > t) = P(min(A, B) > t) = P(T > t)$$

These knowns above, as well as the A and B distributions, will help us find the PDF of Maria's waiting time by first allowing us to find the CDF:

$$F(t) = P(T \leq t) = 1 - P(T > t)$$

$$F(t) = 1 - P(A > t) * P(B > t)$$

$$F(t) = 1 - (1 - F_A(t)) * (1 - F_B(t))$$

$$F(t) = 1 - (1 - \frac{t}{10}) * (1 - (1 - e^{-t/10}))$$

Above, I worked under the assumption that w remains between 0 and 10, similar to Problem 1.1 where we used the bounds 0 and 10. Therefore, this is only valid when W lies in the range [0, 10]. Now that we have the CDF, we can take its derivative to solve for the PDF:

$$f(t) = (\frac{d(F(t))}{dt})$$

Plugging in our CDF, derived previously:

$$f(t) = \frac{d}{dt}(1 - (1 - \frac{t}{10}) * (1 - (1 - e^{-t/10})))$$

$$f(t) = \frac{d}{dt}[1] - \frac{d}{dt}[(1 - \frac{t}{10}) * (e^{-t/10})]$$

Deriving with respect to t, first simplifying $\frac{d}{dt}[1]$ and then applying the chain rule:

$$f(t) = 0 - (\frac{d}{dt}[1 - \frac{t}{10}] * (e^{-t/10}) + (1 - \frac{t}{10}) * \frac{d}{dt}[e^{-t/10}])$$

$$f(t) = -(0 - \frac{1}{10}\frac{d}{dt}[t]) * (e^{-t/10}) - (e^{-t/10}) * (\frac{d}{dt}[-\frac{t}{10}]) * (1 - \frac{t}{10})$$

$$f(t) = -(-\frac{1}{10}) * (e^{-t/10}) - (-e^{-t/10}) * (-\frac{1}{10}\frac{d}{dt}[t]) * (1 - \frac{t}{10})$$

$$f(t) = \frac{e^{-t/10} * (1 - \frac{t}{10})}{10} - \frac{e^{-t/10}}{10}$$

Simplified:

$$f(t) = -\frac{(t-20) * e^{-t/10}}{100}$$

Using the derivative of the CDF, we have found the PDF of Maria's wait time (T):

$f(t) = -\frac{(t-20)*e^{-t/10}}{100}$.

3. ***Write a simulation to check that your answer in part (1) and your expression for the PDF in part (2) are correct.***

```
Simulated Answer for Part 1: 0.3676
```

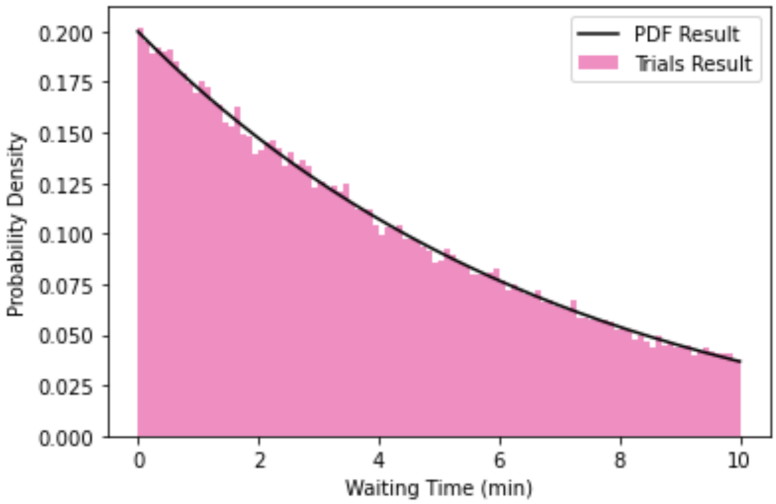


Figure 3. *Checking My Answers Using Simulations*. Above, I have included the output of my code for this part of the problem. I defined a new function to check the $P(B < A)$ calculator in part 1 and I used many trials to simulate the waiting time PDF for Maria's waiting times in the graph above. Code is in Appendix.

Running these simulations, my results display that my answers were on-par. Notably, the simulated answer for part 1 matches $\frac{1}{e}$ and the rough 36.8 percentile; the "PDF result", or the mathematical approach (black line) matches my simulated approach, or "trials result".

Problem 2. Counting Votes (#Distributions)

In a small voting district, $Poisson(\lambda)$ *people arrive to vote during a referendum. Each voter votes for the proposal with probability p and against the proposal with probability* $(1 - p)$ *. Assume everyone casts their vote independently.*

1. ***If is the difference between the number of votes for the proposal and the number of votes against the proposal, determine*** $E[D]$ ***and*** $Var(D)$**.**

Using random variables *F* and *A* to represent the number of votes for and against the proposal, we can define *D* as the following:

$$D = F - A$$

Having defined *D*, we can put our expected values and variances in terms of our r.v.s of *F* and *A* via linearity:

$$E[D] = E[F] - E[A]$$

$$Var(D) = Var(F) + Var(A)$$

Knowing these relationships, we can plug in the known values for expected values and variances for Poisson distributions, displayed in the chicken-and-egg story in the Bliztein Hwang's Introduction to Probability textbook (Blitzstein & Hwang, 2015). Our known distributions are the following:

$$F \sim Poisson(\lambda p)$$

$$A \sim Poisson(\lambda\,(1 - p))$$

Knowing these distributions allows the following to be concluded:

$$E[F] = Var(F) = \lambda p$$

$$E[A] = Var(A) = \lambda\,(1 - p)$$

Pairing the above knowns with the expected values and variances of D in terms of F and A, we can solve for $E[D]$ and $Var(D)$:

$$E[D] = E[F] - E[A] = Var(F) - Var(A)$$

$$E[D] = \lambda p - \lambda(1 - p) = \lambda(2p - 1)$$

$$Var(D) = Var(F) + Var(A) = E[F] + E[A]$$

$$Var(D) = \lambda p + \lambda(1 - p) = \lambda$$

Therefore:

$$E[D] = \lambda(2p - 1)$$

$$Var(D) = \lambda$$

2. ***Implement a simulation to check your answers.***

```
#QUESTION 2: COUNTING VOTES
import matplotlib.pyplot as plt
import scipy.stats as sts
import numpy as np
import random

l = 14 #assigning a random value to lambda
p = 0.45 #assigning a random probability to p

trials = 100000

sample = sts.poisson(l).rvs(trials)

D = []
for x in range(trials):
    #this list appends either True or False values depending on the random variable
    #between 0 and 1. If below p, then True and if above p, then False
    poll = [random.random() < p for _ in range(sample[x])]

    #this calculates the difference between the votes for vs. against the proposal
    diff = poll.count(True) - poll.count(False)

    #appending the diff value for each trial
    D.append(diff)

#Printing out our simulated and declared values to compare
print("Simulated Expected Value:", np.mean(D))
print("Simulated Variance", np.var(D))
print("True Expected Value:", ((2*p-1)*l))
print("True Variance:", l)
```

```
Simulated Expected Value: -1.40818
Simulated Variance 13.9928490876
True Expected Value: -1.3999999999999997
True Variance: 14
```

This code simulated via 100,000 trials, finding the difference between voters that were for versus against the proposal. Both my simulated values and true values (as declared at the beginning of the code) matched very well, within small decimal places. They would approach exactness had I ran more trials.

Problem 3. Hereditary Heights (#Probability)

We model the heights of a family as $Normal(\mu, \sigma^2)$ *random variables. These heights have the same distribution but they are not necessarily independent. Assume there is a mother, a father, and 4 children and that they are all biologically related.*

1. ***This is an oversimplification but assume for this part that the heights are all independent. On average, how many of the 4 children are taller than both parents?***

This problem includes six independently distributed Normal(μ, σ^2) random variables. The mother and father lie within this distribution. If 4 children are randomly placed on this distribution with parents as "markers", we are finding the average number of children that lie to the right (x-axis) of both parent markers. For each child, there are two permutations of being taller than both parents – where the mother is shorter, or where the father is shorter. The probability can be modeled by dividing the number of people by the number of permutations. In this case, we would have an average of 2 children taller than their parents. This value can be confirmed via simulation, where six normally distributed random variables are compared:

```
#QUESTION 3: HEREDITARY HEIGHTS
import matplotlib.pyplot as plt
import scipy.stats as sts
import numpy as np

#setting the values for the standard normal distribution,
#used in normal r.v.s

mu = 0
sigma = 1
n = 1

trials = 100000
results = []
for i in range(trials):
    p1 = np.random.normal(mu, sigma, n)
    p2 = np.random.normal(mu, sigma, n)
    c1 = np.random.normal(mu, sigma, n)
    c2 = np.random.normal(mu, sigma, n)
    c3 = np.random.normal(mu, sigma, n)
    c4 = np.random.normal(mu, sigma, n)

    pval = [p1, p2]
    cval = [c1, c2, c3, c4]

    count = 0
    for x in range(len(cval)):
        if cval[x] > pval[0] and pval[1]:
            count += 1
    results.append(count)

print(sum(results)/trials)
```

```
2.00323
```

2. ***Let X_1 be the height of the mother, X_2 be the height of the father, and $Y_1, \dots, Y_4$ be the heights of the children. Suppose that $(X_1, X_2, Y_1, \dots, Y_4)$ is Multivariate Normal, with $Normal(\mu, \sigma^2)$ marginals and $Corr(X_1, Y_j) = \rho$ for $j \in \{1, 2, 3, 4\}$, with $\rho < 1$. On average, how many of the children are at least 1 centimeter taller than their mother?***

In finding an average number of children 1 centimeter taller than their mother, we create an indicator variable, I_j, describing the j-th child 1 centimeter (or more) taller than their mother.

From the problem, we know that each child follows the same distribution and has the same correlation with their mother. This outlines the expected value that we are finding, being:

$$E[\sum_{j=1}^{4} I_j] = \sum_{j=1}^{4} P(I_j = 1)$$

$$E[\sum_{j=1}^{4} I_j] = 4 * P(I_1 = 1)$$

Using LOTP, we know that:

$$P(I_1 = 1) = P(Y_1 - X_1 \geq 1) = 1 - P(Y_1 - X_1 < 1)$$

Similar to the "counting votes" problem, we can define a new random variable, *D*, to represent the difference – but this time, in height between the first child and the mother:

$$D = Y_1 - X_1$$

Becuase both the child and mother have normally distributed heights to begin with, the difference between the two (our new r.v., *D*) would remain normally distributed with a mean of 0 and a variance to be defined. Using the class textbook, $Var(D)$, or $Var(Y_1 - X_1)$ is equal to the following (Blitzstein & Hwang, 2015):

$$Var(D) = Var(Y_1 - X_1) = Var(Y_1 + X_1) - 2Cov(Y_1, X_1)$$

$$Var(D) = Var(Y_1) + Var(X_1) - 2Corr(Y_1, X_1) * \sqrt{Var(Y_1)\,Var(X_1)}$$

$$Var(D) = \sigma^2 + \sigma^2 - 2\rho\sigma^2$$

$$Var(D) = 2\sigma^2(1 - \rho)$$

Now that we have the variance, we can define the distribution of our random variable, *D*, as we know that the mean is 0 and it is normally distributed:

$$D \sim Normal(0,\ 2\sigma^2(1 - \rho))$$

This defined distribution for *D* allows us to find the expected value:

$$E[\sum_{j=1}^{4} I_j] = 4 * P(I_1 = 1)$$

Knowing that:

$$P(I_1 = 1) = 1 - P(D < 1)$$

We can plug this into our expected value equation:

$$E[\sum_{j=1}^{4} I_j] = 4 * (1 - P(D < 1))$$

From the normal distribution, we know that:

$$D = \frac{b-a}{\sqrt{2\sigma^2}}$$

Plugging this in:

$$E[\sum_{j=1}^{4} I_j] = 4 * (P(\frac{b-a}{\sigma\sqrt{2(1-\rho)}} < \frac{b-a}{\sigma\sqrt{2(1-\rho)}}))$$

$$E[\sum_{j=1}^{4} I_j] = 4 * (P(\frac{D-0}{\sigma\sqrt{2(1-\rho)}} < \frac{1-0}{\sigma\sqrt{2(1-\rho)}}))$$

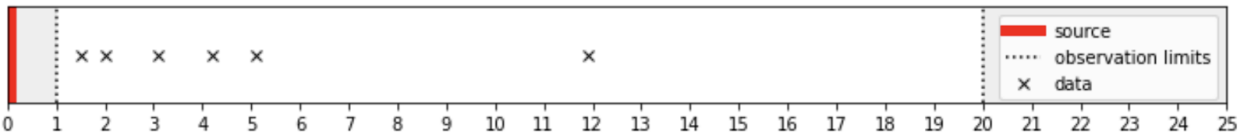
$$E[\sum_{j=1}^{4} I_j] = 4 * (1 - \Phi(\frac{1}{\sigma\sqrt{2(1-\rho)}}))$$

Solution:

$$4 * (1 - \Phi(\frac{1}{\sigma\sqrt{2(1-\rho)}})$$

Problem 4. Radioactive Decay (#ParameterEstimation)

Radioactive particles are emitted from a source and decay independently at a random distance X from the source. The experimental setup allows for decay events to be observed only if they occur at a distance between $x = 1$ *cm and* $x = 20$ *cm from the source.*
As data, use $\{x_i\} = \{1.5,\ 2.0,\ 3.1,\ 4.2,\ 5.1,\ 11.9\}$.



If $X \sim Expo(\lambda)$*, what is* λ*?*

1. ***Provide your best estimate for*** λ ***and motivate in what sense this is the "best" estimate.***

In finding λ , we need to maximize the probability of getting this data set given a value for λ . Thus, this data would be most likely to occur given our found λ value. Defining the probability of getting this data, we use λ as our given: the probability of the data given a specific λ value:

$$P(data \mid \lambda)$$

To maximize this, we need to define probability above by inputting each given independent data points:

$$\prod_{i=1}^{6} P(x_i \mid \lambda)$$

Knowing that this distribution is exponential, we can conclude the following (similar to Company B's bus distribution in problem 1):

$$\prod_{i=1}^{6} P(x_i \mid \lambda) = \prod_{i=1}^{6} \lambda e^{-\lambda x_i}$$

Inputting the data:

$$\lambda^6 e^{-\lambda \sum_{i=0}^{6} x_i} = \lambda^6 e^{-27.8\lambda}$$

Now that we have defined $P(data \mid \lambda)$, we can maximize this using derivatives. Deriving with respect to λ , the variable we are solving for, using the chain rule and solving for λ once set equal to 0:

$$\frac{d}{d\lambda}(\lambda^6 e^{-27.8\lambda})$$

$$\frac{d}{d\lambda}[\lambda^6] * e^{-27.8\lambda} + \lambda^6 * \frac{d}{d\lambda}[e^{-27.8\lambda}]$$

$$6\lambda^5 e^{-27.8\lambda} + \lambda^6 e^{-27.8\lambda} * \frac{d}{d\lambda}[-27.8\lambda]$$

$$6\lambda^5 e^{-27.8\lambda} + \lambda^6 e^{-27.8\lambda} * (-27.8 * \frac{d}{d\lambda}[\lambda])$$

$$6\lambda^5 e^{-27.8\lambda} - 27.8(\lambda^6 e^{-27.8\lambda})$$

Setting equal to 0 and solving for λ :

$$0 = 6\lambda^5 e^{-27.8\lambda} - 27.8(\lambda^6 e^{-27.8\lambda})$$

$$\lambda = \frac{6}{27.8}$$

Solution:

$$\lambda \sim 0.216$$

2. ***Provide a 95% interval for the value of*** λ***. This part can be a bit hand-wavy, meaning you don't have to compute a precise 95% interval but should provide some reasonable justification for how you arrived at the interval you got.***

Here, a confidence interval would be a good option to make λ a bit more generalizable, as there is a small amount of data and our λ value above is a point value. Adding a 95% interval

would give λ a bit of wiggle room dependant on the data collected (not straying far from the data given above, of course). To find this interval, we can use our current λ value and the formula listed on the *Exponential Distribution* Wikipedia page (n.a., 2022):

$$\lambda_{lower} = \widehat{\lambda}\,(1 - \frac{1.96}{\sqrt{n}})$$

$$\lambda_{lower} = \widehat{\lambda}\,(1 - \frac{1.96}{\sqrt{n}})$$

Plugging in our knowns:

$$\lambda_{lower} = 0.216\,(1 - \frac{1.96}{\sqrt{6}})$$

$$\lambda_{upper} = 0.216\,(1 + \frac{1.96}{\sqrt{6}})$$

Solutions:

$$\lambda_{lower} \sim 0.043$$

$$\lambda_{upper} \sim 0.039$$

These intervals seem reasonable, as well as the math behind them! The 1.96 rings familiar, as the formulas add or subtract the z-score of the 95th percentile as it applies to the upper or lower bounds. There is lack of exactness in this answer, as 0.216 was taken from the problem above and data is few. With more data, these estimates would be strengthened and more generalizable for the radioactive-decay-distance application.

References

Blitzstein, J.K. & Hwang, J. (2015). *Introduction to probability*. Crc Press.

Wikipedia (2022). *Exponential Distribution*.

https://en.wikipedia.org/wiki/Exponential_distribution.

Appendix

```
#QUESTION 1: BUS LINES

#1.1
import matplotlib.pyplot as plt
import scipy.stats as sts
import numpy as np
import math

#repeating r.v. generation for x number of trials
trials = 100000

#modeling Company A's distribution using Uniform
A = sts.uniform(scale=10).rvs(trials)

plt.hist(A, density=True, bins=10, color="darkgreen", alpha=0.5, label="Bus A", rwidth=0.965)
plt.xlabel("Waiting Time (min)")
plt.ylabel("Probability Density")
plt.ylim([0.0, 0.11])
plt.show()

#modeling Company B's distribution using Exponential
B = sts.expon(scale=10).rvs(trials)
plt.hist(B, density=True, bins=100, color="darkblue", alpha=0.5, label="Bus A", rwidth=0.965)
plt.xlabel("Waiting Time (min)")
plt.ylabel("Probability Density")
#using these parameters to scale it to Company A for visual comparison
plt.xlim([0.0, 10])
plt.ylim([0.0, 0.11])
plt.show()
```

```python
#1.3
#checking answer for part 1:
#creating a function to find how often B arrives before A
def checkans():
    count = 0
    for x in range(trials):
        if B[x] < A[x]:
            #appending to the count if B is faster than A
            count += 1
    return (count/trials) #the percent of the time that B arrived before A
print("Simulated Answer for Part 1:", checkans())

#checking answer for part 2:
#creating
T = []
T = np.minimum(A, B)

#creating a range of values to use for the PDF we calculated above
xrange = np.linspace(0, 10, 1000)
PDF_time = []

#this loop feeds the values created in the xrange through the PDF, outputting values to plot on top of the trials.
for y in range(1000):
    PDF_time.append(-(xrange[y]-20)*(math.e**(-xrange[y]/10))/(100))

#plotting the trials histogram (using the PDF_time list that we simulated) and our PDF equation result in Part 2
plt.hist(T, density=True, bins=100, color='hotpink', alpha=0.8, label="Trials Result")
plt.xlabel("Waiting Time (min)")
plt.ylabel("Probability Density")
plt.plot(xrange, PDF_time, color='black', label='PDF Result')
plt.legend()
plt.show()
```