Assignment3

February 21, 2024

1 Assignment 3: Solutions

[3]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.stats import nbinom
from scipy.integrate import solve_ivp
import random
```

1.1 1. Museum Collections

Museum collections are extraordinarily valuable in the study of ecology and evolution as they give us access to rare long-term (longitudinal) data that couldn't be collected in the 4 years of a typical PhD project. Consider the number of specimens of European Goldenrod *Solidago virgaurea* present in a herbarium (herbarium: museum for plants). These collections are done by many different researchers, often with years in between. However, given that a researcher is collecting data on goldenrods they are likely to submit more than one accession (accession: submission to a biological database or museum) at the same time. Hence we can model the accumulation of accessions of *S. virgaurea* using a compound Poisson process.

Suppose that research studies on goldenrod occur at a constant rate $\lambda = 0.61$ 1/year and that the number of accessions submitted per study is distributed according to a negative binomial distribution with parameters $r = 8$, $p = 0.75$.

Part A: How many independent research studies are expected to occur during a PhD (e.g., 4 years)? What is the expected time between research studies? How many accessions are submitted by the average research study? Plot the distribution of accessions/study. How many accessions would constitute a 'large' study?

From the Poisson distribution, we have that there is an expected/mean 0.61 research studies per year for a total of 2.44 research studies over the course of a standard PhD.

[1]:
```
4*0.61
```

[1]:
```
2.44
```

The expected waiting time is the mean of the corresponding exponential distribution or $\frac{1}{\lambda} = 1.64$years

[2]:
```
1/0.61
```

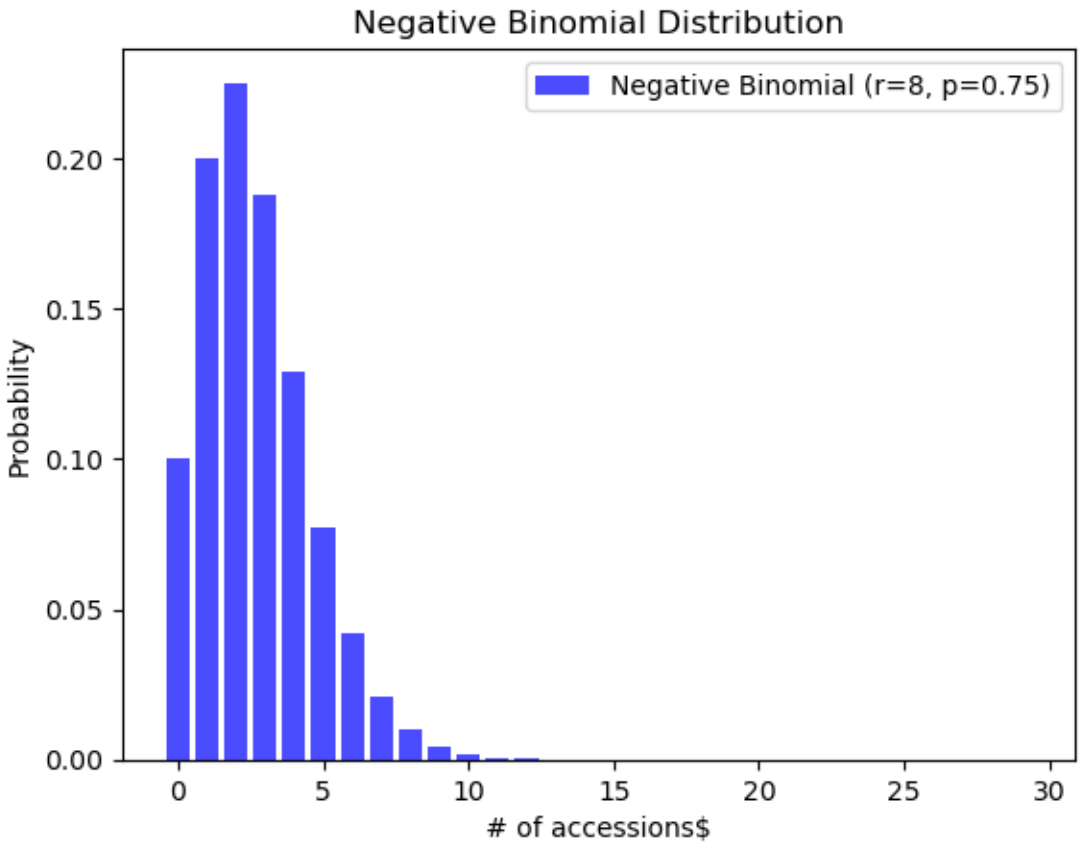
[2]:
```
1.6393442622950822
```

The mean of the negative binomial distribution is $\frac{r(1-p)}{p}$:

[11]:
```
8*0.25/0.75
```

[11]:
```
2.6666666666666665
```

[7]:
```
# Set the parameters for the negative binomial distribution
r = 8  # Number of failures until success
p = 0.75  # Probability of success on each trial

# Generate a range of x values
x_values = np.arange(0, 30)

# Calculate the probability mass function (PMF) for each x value
pmf = nbinom.pmf(x_values, r, p)

# Plot the negative binomial distribution
plt.bar(x_values, pmf, color='blue', alpha=0.7, label=f'Negative Binomial␣
↪(r={r}, p={p})')
plt.title('Negative Binomial Distribution')
plt.xlabel('# of accessions$')
plt.ylabel('Probability')
plt.legend()
plt.show()
```



To evaluate what would be "large" study we need to know the value of x for which 95% of studies are smaller. We can do this with the Percentile Point Function (PPF).

[9]:
```
nbinom.ppf(0.95, 8, 0.75)
```

[9]:
```
6.0
```

Hence a large study is a study that has 7 or more accessions

Part B: What is the expected number of accessions over a 10-year period by all researchers?

The mean of a compound Poisson process is equal to the expected number of events times the expected size of those events. In our case this gives: $\lambda * 10 * 2.66 = 16.226$

[12]:
```
0.61*10*2.66
```

[12]:
```
16.226
```

Part C: What is the variance in the number of accessions submitted over this 10 year period?

We have to use the law of total variance where Y is the number of accessions over the 10 years and N is the number of studies:

$$\text{Var}(Y) = \underbrace{E_N[Var(Y|N)]}_{\text{Variance within}} + \underbrace{Var_N[E[Y|N]]}_{\text{Variance Between}}$$

Let X_i be the number of accessions in the ith study.

$$Var(Y|N) = \sum_{i=1}^{N} Var(X_i) = N * \frac{r(1-p)}{p^2}$$
$$E_N[Var(Y|N)] = 10\lambda \frac{r(1-p)}{p^2}$$

Using the result from part B we have,

$$E[Y|N] = N\frac{r(1-p)}{p}$$
$$Var_N[E[Y|N]] = \left(\frac{r(1-p)}{p}\right)^2 \times 10\lambda$$

So:

$$\text{Var}(Y) = 10\lambda \left(\frac{r(1-p)}{p^2} + \left(\frac{r(1-p)}{p}\right)^2\right)$$

[14]:
```
10*0.61*(8*0.25/0.75**2+8**2*0.25**2/0.75**2)
```

[14]:
```
65.06666666666666
```

1.2 2. Simulating a Stochastic Epidemic

Consider a simplified epidemiological model for the spread of an infectious disease in a small population. The model includes two compartments: susceptible individuals (S) and infected individuals (I). The disease spreads through a single type of interaction with a given infection rate.

Susceptible individuals become infected at a rate βSI Infected individuals recover (immediately become susceptible) at a per-capita rate γI

Part A: Write a system of ODEs describing the dynamics in this system. Solve them numerically for $S(0) = 99, I(0) = 1$ and $\beta = 0.001$ and $\gamma = 0.05$?

$$\frac{dS(t)}{dt} = -\beta S * I + \gamma I \quad S(0) = 99$$
$$\frac{dI(t)}{dt} = \beta S * I - \gamma I \quad I(0) = 1$$

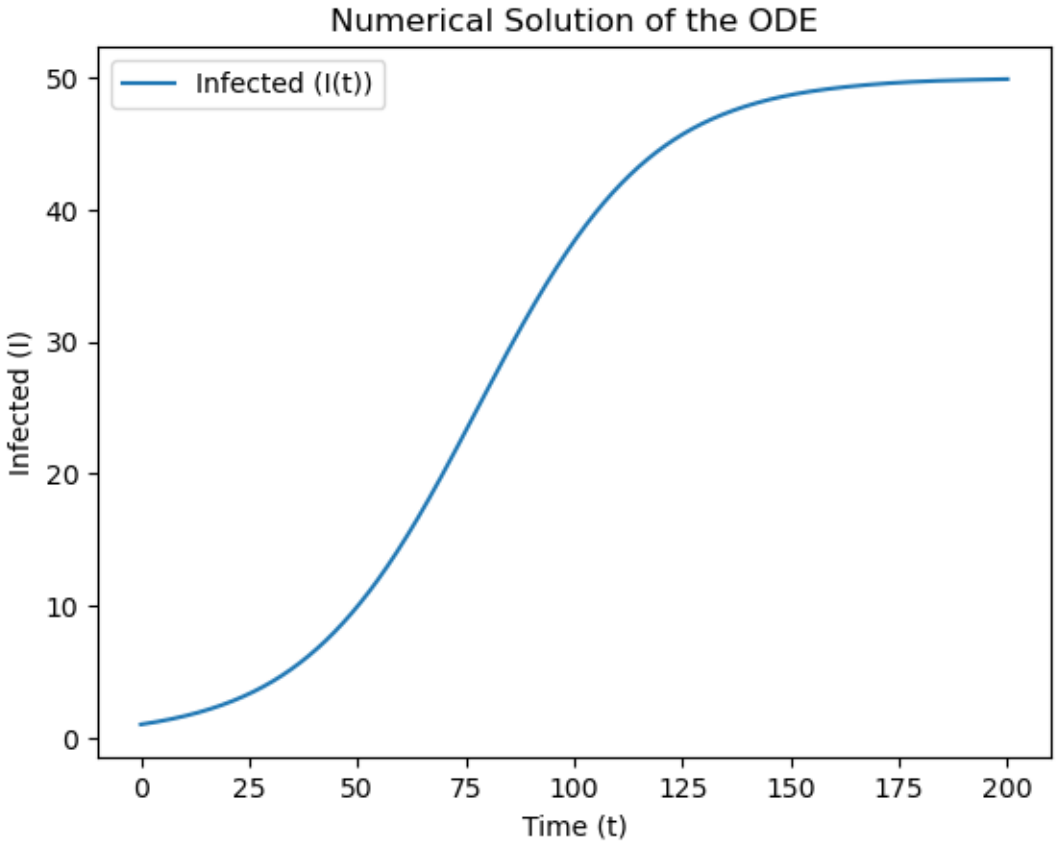
We can rewrite this as a single equation where $N = 100$

$$\frac{dI(t)}{dt} = \beta(N - I) * I - \gamma I \quad I(0) = 1$$

There are two options here, either you can solve the ODE numerically or note that this has a general solution. I am going to use the numerical integration approach.

[18]:
```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

# Define the ODE function
def ode(t, y, beta, gamma, N):
    dIdt = beta * (N - y) * y - gamma * y
    return dIdt

# Set the parameters
beta1 = 0.001
gamma1 = 0.05
N1 = 100

# Initial condition
I0 = [1]

# Time span for integration
t_span = (0, 200)

# Solve the ODE
sol = solve_ivp(
    fun=lambda t, y: ode(t, y, beta1, gamma1, N1),
    t_span=t_span,
    y0=I0,
    method='RK45',
    dense_output=True
)

# Generate time points for plotting
t_plot = np.linspace(t_span[0], t_span[1], 1000)

# Evaluate the solution at the specified time points
I_plot = sol.sol(t_plot)

# Plot the solution
plt.plot(t_plot, I_plot[0], label='Infected (I(t))')
plt.title('Numerical Solution of the ODE')
plt.xlabel('Time (t)')
plt.ylabel('Infected (I)')
```

```
plt.legend()
plt.show()
```



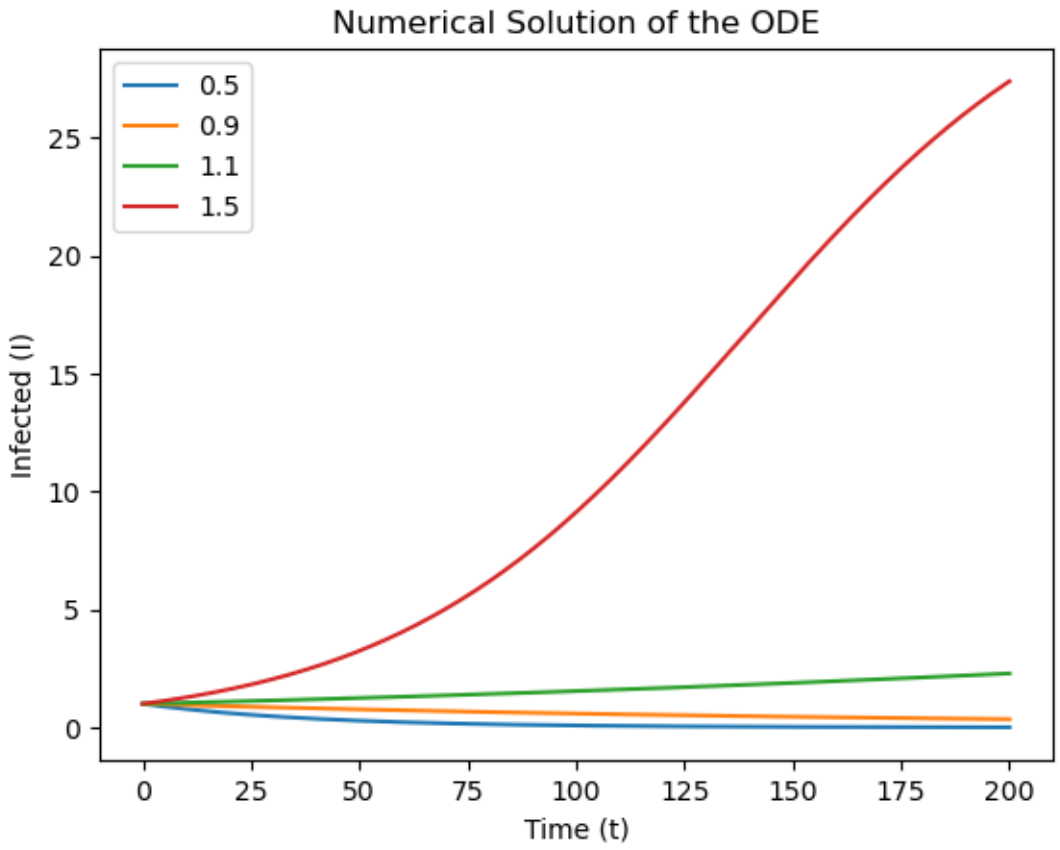
Part B: For this model the value $R_0 = \frac{N\beta}{\gamma}$ is the number of secondary infections that result from a single initial infection in an otherwise susceptible population. If $R_0 < 1$ then the disease is guaranteed to go extinct if $R_0 > 1$ the disease will spread in the deterministic model. What are four parameter combinations that have an R_0 value of $0.5, 0.9, 1.1$ and 1.5 respectively? Plot the dynamics for $I(t)$ in the deterministic model for each of these parameter sets.

Let's keep $\gamma = 0.05$ and $N = 100$ then:

[1]:
```
betaA=0.5*0.05/100
print("Beta A: ",betaA)
betaB=0.9*0.05/100
print("Beta B: ",betaB)
betaC=1.1*0.05/100
print("Beta C: ",betaC)
betaD=1.5*0.05/100
print("Beta D: ",betaD)
```

```
Beta A:  0.00025
Beta B:  0.00045000000000000004
Beta C:  0.00055
Beta D:  0.0007500000000000001
```

[29]:
```python
def solBeta(beta):
    temp=solve_ivp(
        fun=lambda t, y: ode(t, y, beta, 0.05, 100),
        t_span=t_span,
        y0=[1],
        method='RK45',
        dense_output=True
    )
    return temp

# Generate time points for plotting
t_plot = np.linspace(t_span[0], t_span[1], 1000)

# Evaluate the solution at the specified time points
IA_plot = solBeta(betaA).sol(t_plot)
IB_plot = solBeta(betaB).sol(t_plot)
IC_plot = solBeta(betaC).sol(t_plot)
ID_plot = solBeta(betaD).sol(t_plot)

# Plot the solution
plt.plot(t_plot, IA_plot[0], label='0.5')
plt.plot(t_plot, IB_plot[0], label='0.9')
plt.plot(t_plot, IC_plot[0], label='1.1')
plt.plot(t_plot, ID_plot[0], label='1.5')
plt.title('Numerical Solution of the ODE')
plt.xlabel('Time (t)')
plt.ylabel('Infected (I)')
plt.legend()
plt.show()
```



Part C: Write a Gillespie algorithm to simulate the dynamics of a corresponding stochastic epidemic where $R_0 = 1.5$. Describe what the possible events are, their rates, and what is their effect on state space?

Event	Rate	$[\Delta S, \Delta I]$
Transmission	$\beta S * I$	$[-1, +1]$
Recovery	γI	$[+1, -1]$

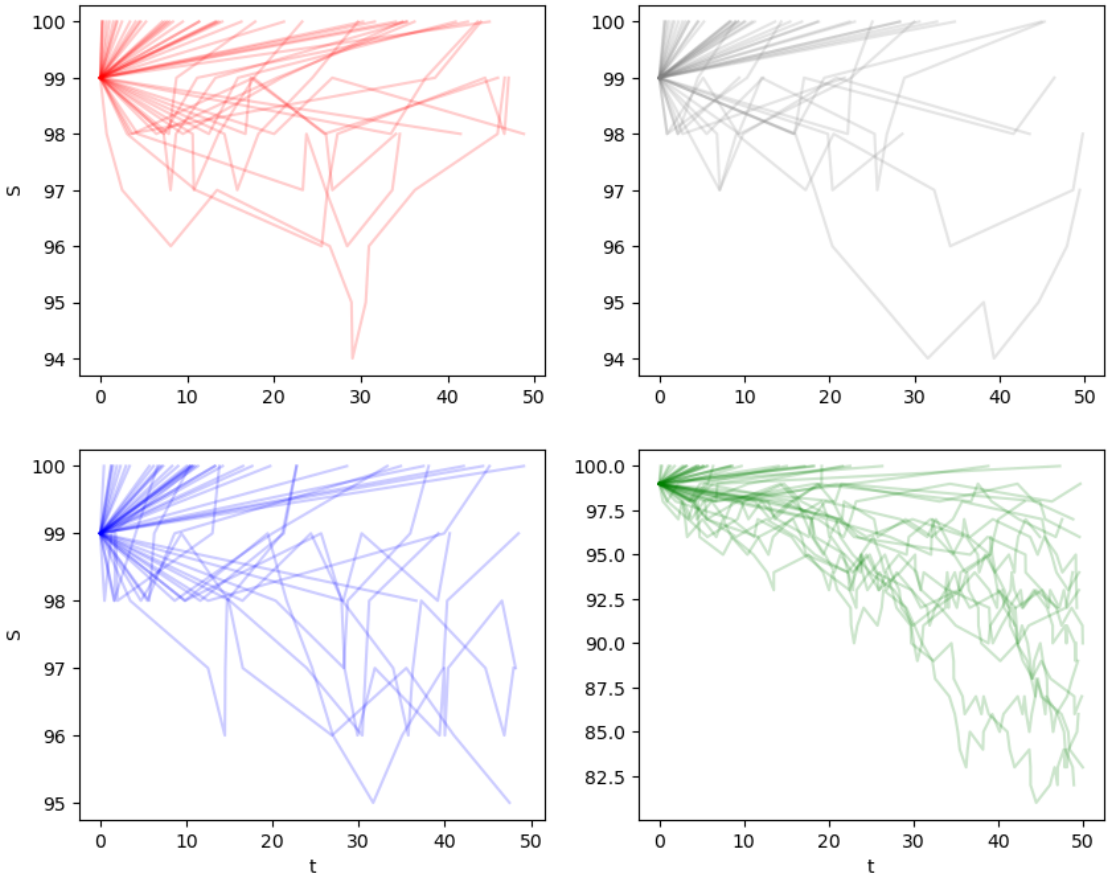
[30]:
```
# Define reaction rates
def rates(state, beta, gamma):
    return np.array([beta*state[0]*state[1],gamma*state[1]])

def gillespie(tMax,beta,gamma):
    # Lists to store results for plotting
    time= np.array([[0]])
    state = np.array([[99,1]])
    events = np.array([[-1,1], [1,-1]])
    temp=rates(state[-1],beta,gamma)
    dt=np.random.exponential(scale=1/np.sum(temp))
```

```
    while time[-1]+dt<tMax:
        time=np.vstack([time, time[-1]+dt])
        # Perform a weighted random choice
        random_event = random.choices(events, temp)
        state=np.vstack([state,state[-1]+random_event])
        # Calculate new rates and new dt
        temp=rates(state[-1],beta,gamma)
        #The case of extinction
        if np.sum(temp)>0:
            dt=np.random.exponential(scale=1/np.sum(temp))
        else:
            dt=tMax-time[-1]
    return [time, state]
```

Part D: Simulate 50 trajectories for each of the four parameter sets you chose in B. How do the stochastic dynamics compare to the deterministic dynamics? Are you surprised by any of the results given R_0?

[48]:
```
# Create a dictionary to store results
sim_dict_A = {}
sim_dict_B = {}
sim_dict_C = {}
sim_dict_D = {}

# Calculate and save results for specified indices
for index in range(50):
    sim_dict_A[index] = gillespie(50,betaA,0.05)
    sim_dict_B[index] = gillespie(50,betaA,0.05)
    sim_dict_C[index] = gillespie(50,betaA,0.05)
    sim_dict_D[index] = gillespie(50,betaD,0.05)
#Plotting
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
for index in range(50):
    axs[0, 1].plot(sim_dict_A[index][0],sim_dict_A[index][1][:
 ↪,0],color='gray',alpha=0.2)
    axs[0, 0].plot(sim_dict_B[index][0],sim_dict_B[index][1][:
 ↪,0],color='red',alpha=0.2)
    axs[1, 0].plot(sim_dict_C[index][0],sim_dict_C[index][1][:
 ↪,0],color='blue',alpha=0.2)
    axs[1, 1].plot(sim_dict_D[index][0],sim_dict_D[index][1][:
 ↪,0],color='green',alpha=0.2)
# Label the outer axes
axs[0, 0].set_ylabel('S')
axs[1, 0].set_ylabel('S')
axs[1, 0].set_xlabel('t')
axs[1, 1].set_xlabel('t')
```

[48]: `Text(0.5, 0, 't')`



The number of susceptible hosts for $R_0 = 0.5$ (red),$R_0 = 0.9$ (gray),$R_0 = 1.1$ (blue),$R_0 = 1.5$ (green).

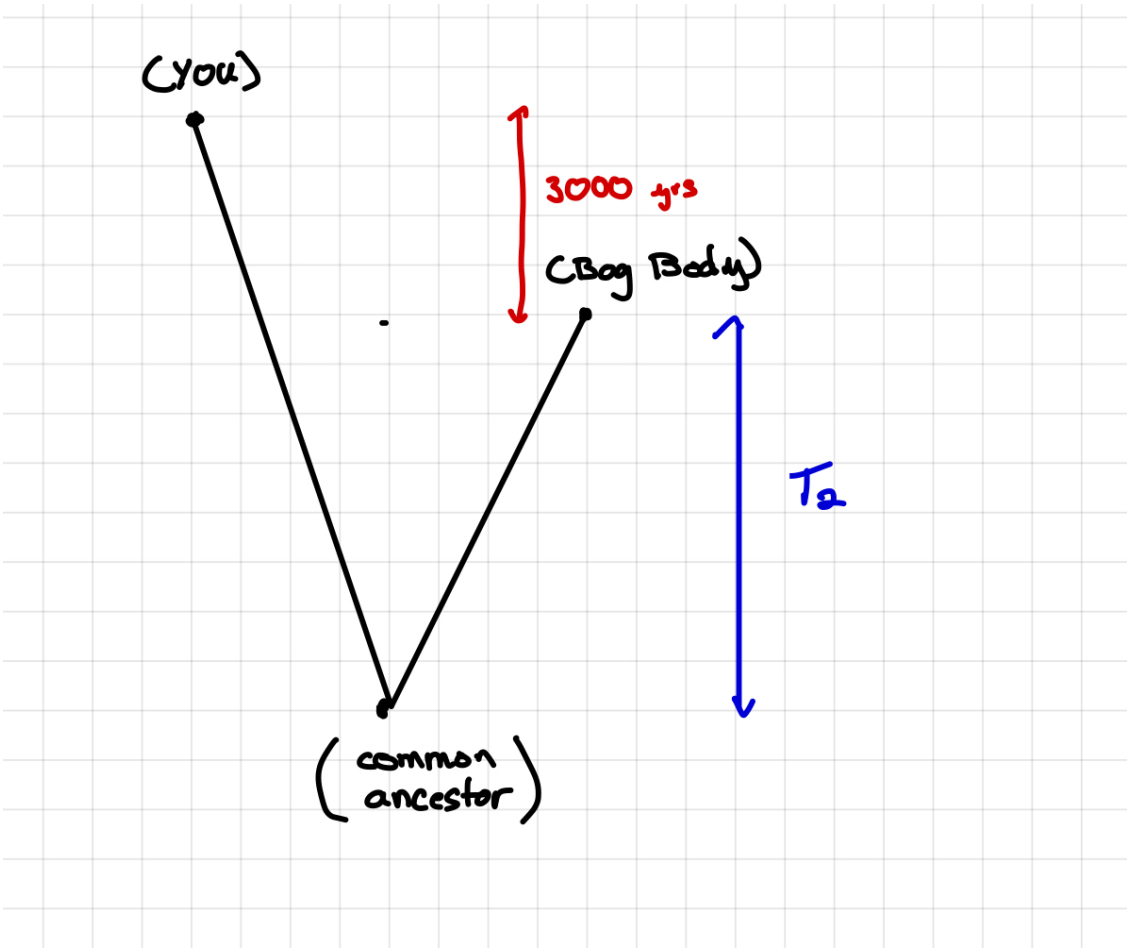
The disease can go extinct if $R_0 > 1$ and can persist for awhile if $R_0 < 1$.

1.3 3. (Optional) Bog Bodies and Ancestral DNA

The chemical conditions of peat bogs are ideal for the natural preservation of human bodies making them a rich source of ancient cadavers known as 'bog bodies' these bodies can range in age but are often from the Iron Age (1300 B.C.E. to 800 C.E.). Over this time period, the effective human population size is approximately 5000.

Part A: Consider a bog body that is 3000 years old. Draw the topology of a gene genealogical between yourself and this bog body.

Topology:



[3]:
```
print(3000/20)
print(150/5000)
```

```
150.0
0.03
```

Note that we should technically divide by $2 * N_e$ since humans are diploid. But for consistency with the notes, I treat them as haploid here.

Part B: Assuming that human generation times are 20 years. How long ago did you and the bog body share a common ancestor? Give the full distribution of times to common ancestry, the expected time, and the variance in times and make sure to note the units of time that each of these answers is measured in. What is the expected time to your common ancestor in units of years, generations, and coalescent time units?

First, let's consider the time over which you could not have possibly shared a common ancestor (Red Arrow).

3000 years=150 generations=0.03 coalescent time units.

Then there is the time over to coalescence (given that the two lineages coexist, shown by the blue arrow) is given by T_2 where:

$$\Pr(T_2 = \tau) = \binom{2}{2} e^{-\binom{2}{2}\tau} = e^{-\tau}$$

The net result is that the probability of sharing a common ancestor x coal. units of time ago is:

$$\Pr(x) = \begin{cases} 0 & x < 0.03 \\ e^{-(x-0.03)} & x > 0.03 \end{cases}$$

The *EXPECTED TIME* to common ancestry is:

$$E[x] = 0.03 + E[\tau] = 1.03 \text{ coal. units } = 5150 \text{ generations } = 103,000 \text{ years}$$

In comparison, Humans and Bonobos diverged approximately 4.5-6MYA (Locke et al. 2011).

[5]:
```
5000*1.03*20
```

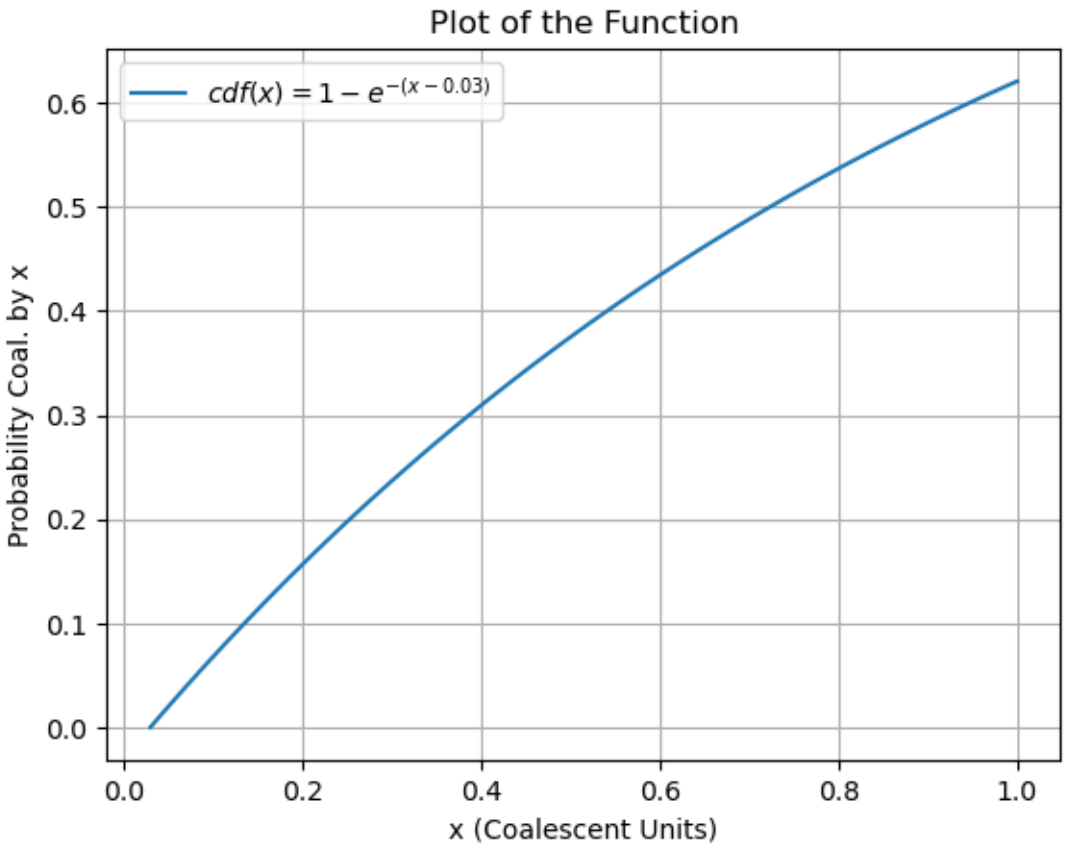
[5]:
```
103000.0
```

The *VARIANCE IN THE TIME* to common ancestry is:

$$Var(x) = Var(0.3 + \tau) = 0.3 + Var(\tau) = 0.3 + \frac{1}{1^2} = 1.3 \text{ generations } = 103,000 \text{ years}$$

Plotting the full CDF (The probability that the lineages have coalesced by time x):

[16]:
```
# Define the function
def f(x):
    return 1-np.exp(-(x - 0.03))

# Generate x values
x_values = np.linspace(0.03, 1, 100)  # Adjust the range as needed

# Compute corresponding y values
y_values = f(x_values)

# Plot the function
plt.plot(x_values, y_values, label=r'$cdf(x) = 1-e^{-(x-0.03)}$')
plt.xlabel('x (Coalescent Units)')
plt.ylabel('Probability Coal. by x')
plt.title('Plot of the Function')
plt.legend()
plt.grid(True)
plt.show()
```



Part C: Assuming an infinite sites model, what is the expected number of pairwise differences between your genome and that of the bog body? Assume a mutation rate of $\theta = 0.8$. How many segregating sites are there in this sample of two genomes (you and the bog body)?

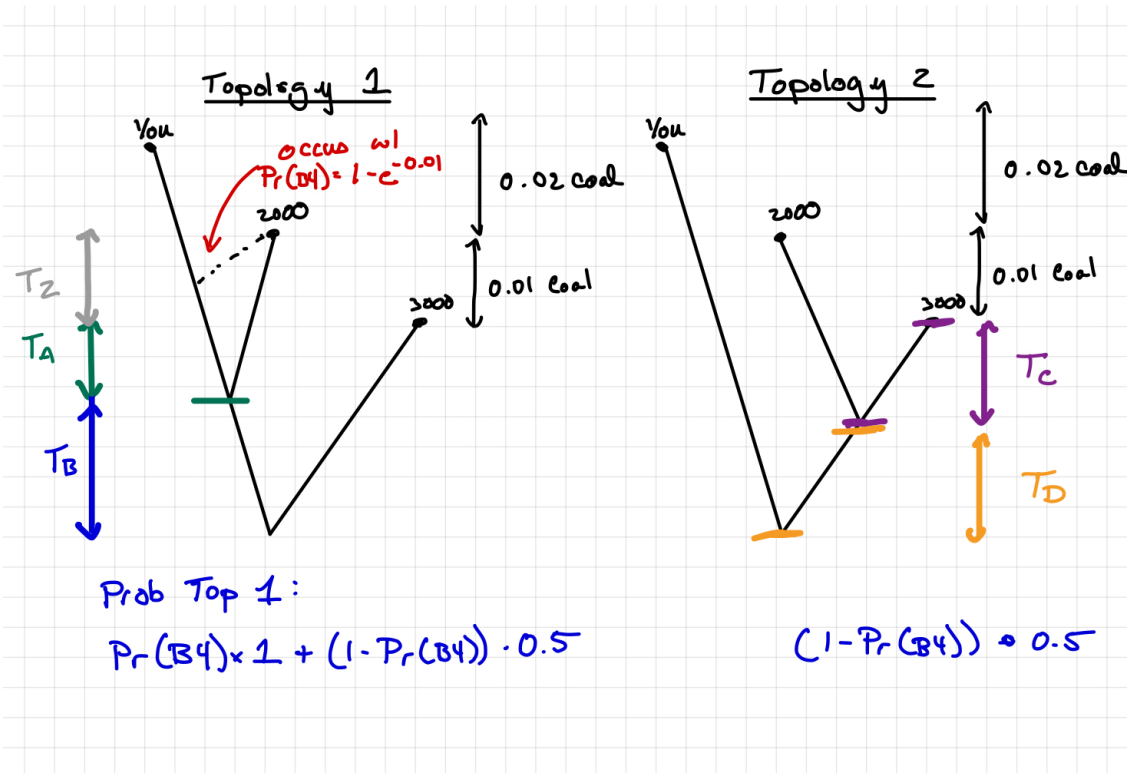
The expected number of pairwise differences between a pair of samples $E[\pi] = \theta = 0.8$, plus we have to add any mutations that may have occurred between the present and 3000 years ago. Recall that the θ is twice the mutation rate/coalescent time unit. So the expected number of mutations in 0.03 coalescent time units is $0.03 * \theta/2 = 0.012$.

The expected number of pairwise differences then is: $E[\pi] = 0.812$

[17]:
```
0.03*0.8/2
```

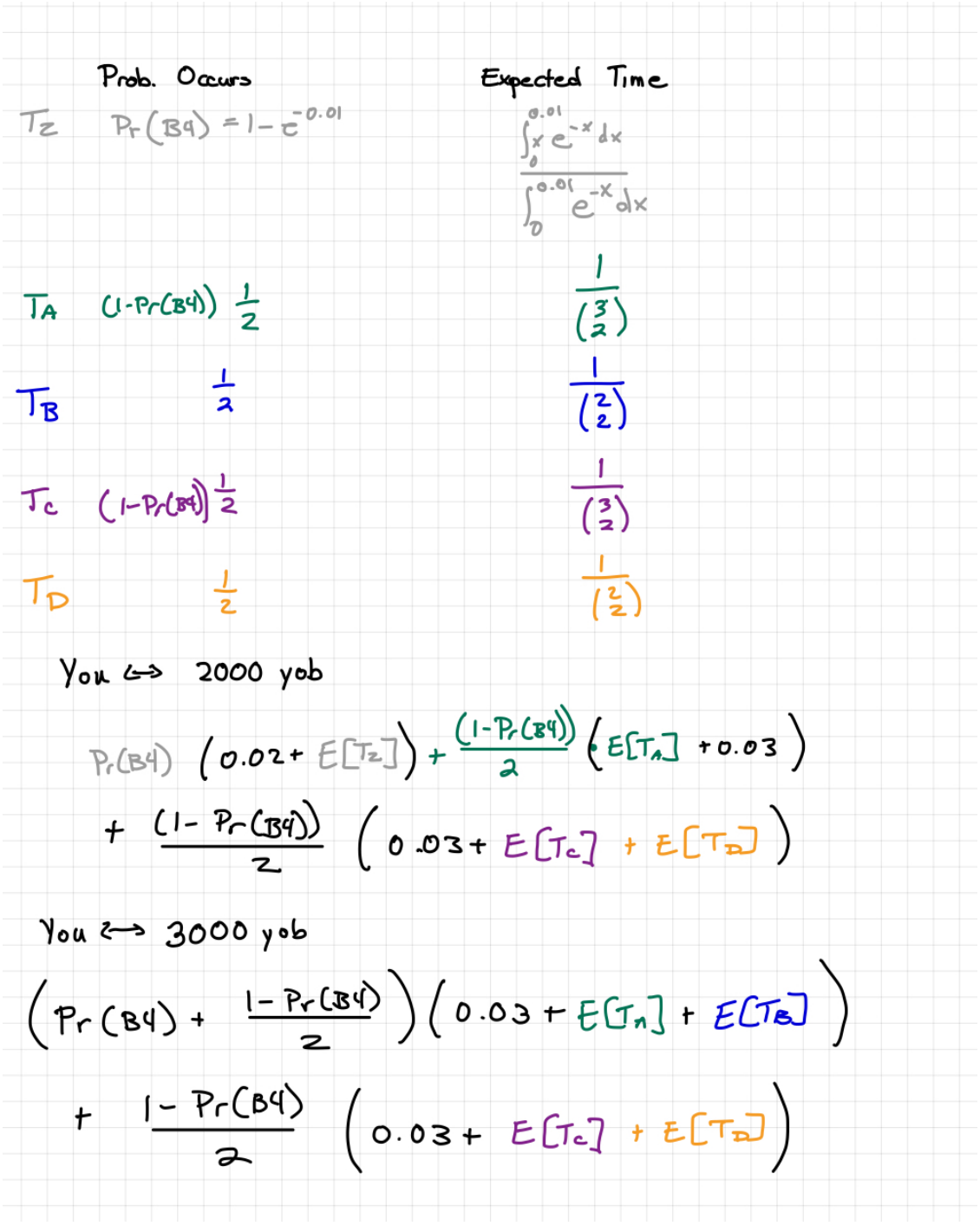
[17]:
```
0.012
```

Part D (Challenge Part for 795): Now consider a sample with three genomes, your genome, the 3000-year-old bog body, and a second 2000-year-old bog body. Draw the two possible genealogical topologies between the three samples. Do these two topologies occur with equal probability, if not what is the probability each occurs? [Hint: What is the probability that you and the 2000-year-old body coalesce before 3000 years ago?]



Part E (Challenge Part for 795): What are the expected times to the common ancestor of a) You and the 2000-year-old body, b) You and the 3000-year-old body, and c) the 2000-year-old body and the 3000-year-old body

See Figure above



1.4 4. Genetic diversity

Consider the following sample of 5 genome sequences. There are several different formats in which DNA sequences can be reported. The following is in the style of a VCF "Variant Call Format" file that reports only sites in which two or more neucleotides are present.

seq:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	A	T	T	A	A	C	G	G	A	G	G	G	C	G	C	G	T	G	T	A	T	T
2	A	A	T	A	A	C	G	G	A	G	G	G	A	G	C	G	T	G	A	A	T	T
3	T	T	A	A	A	C	G	G	A	G	G	A	A	C	C	G	T	G	T	G	A	C
4	A	T	A	G	G	T	C	G	C	C	G	A	G	G	A	A	G	T	A	A	T	T
5	A	T	A	G	G	T	C	A	C	C	G	G	A	G	G	A	T	A	T	A	A	T

Part A: Calculate the number of segregating sites, S, in the sample.

There are 22 SNPs in the genome so there are 21 segregating sites.

Part B: Calculate the number of pairwise differences between each, $\pi_{i,j}$ between each pair of sequences. What is the average number of pairwise differences in this sample?

$$\begin{bmatrix} - & 3 & 7 & 14 & 13 \\ - & - & 8 & 15 & 14 \\ - & - & - & 15 & 14 \\ - & - & - & - & 5 \\ - & - & - & - & - \end{bmatrix}$$

The mean number of pairwise differences is:

[2]:
```
(3+7+13+13+8+15+14+15+14+5)/10
```

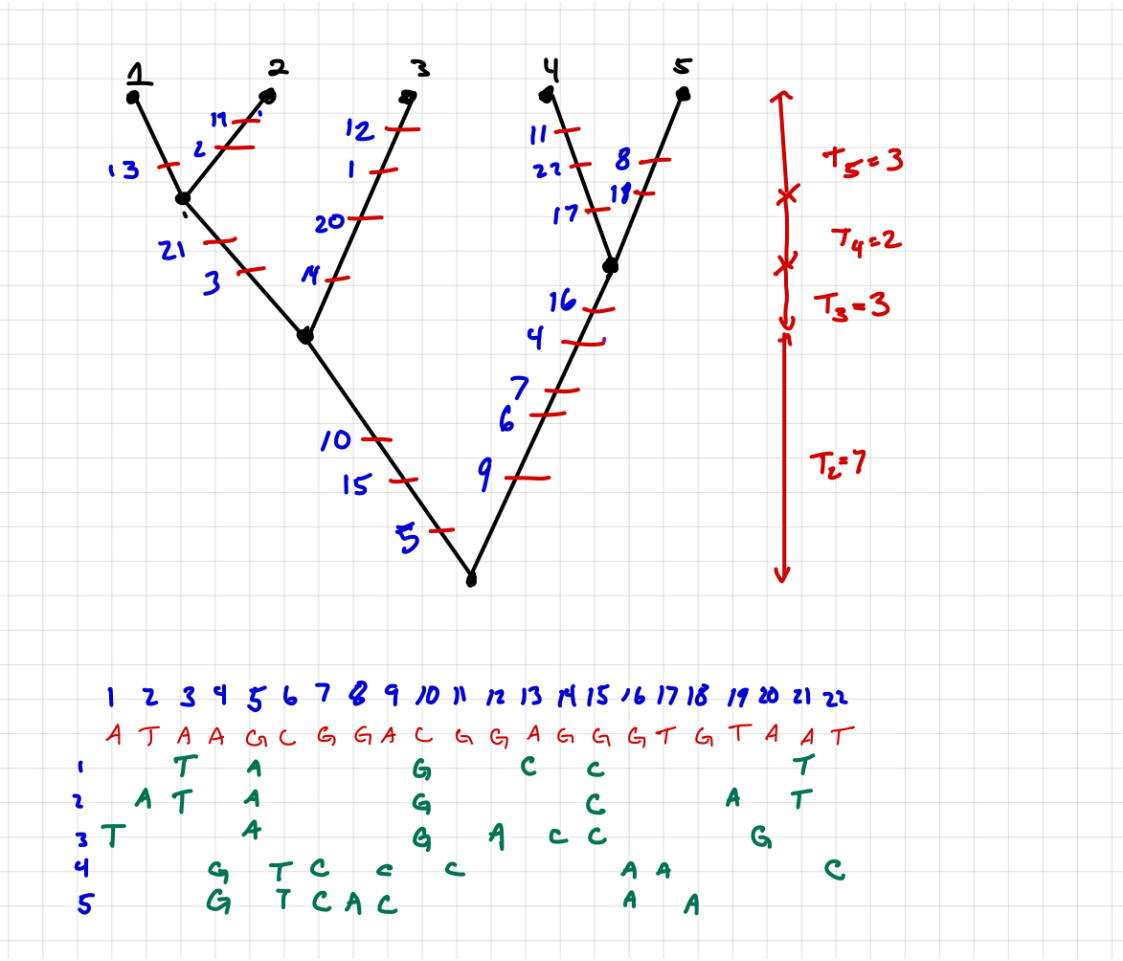
[2]:
```
10.7
```

Part C: Calculate the observed site frequency spectrum, ξ.

$\xi_1 = 12$ & $\xi_2 = 7$ & $\xi_3 = 3$ & $\xi_4 = 0$

Part D (Challenge Part for 795): Assuming genetic diversity evolves according to the infinite sites model, propose a hypothetical genealogy of this sample. What is the likely topology of this genealogy and what are the likely coalescent times?

The tree topology and mutations are given by:



Note that we can't figure out exact coalescent times from the data but they should be approximately proportional to the values shown.

1.5 5. Time to the most recent common ancestor

Consider a population from which you have sampled 4 haploid individuals.

Part A: Draw and label a coalescent history in which the coalescent times shown are proportional to the expected values.

To answer this we need to calculate the expected time to coalescence T_i for $i = \{2, 3, 4\}$

[57]:
```
def ExpT(i):
    return 2/(i*(i-1))
```

[58]:
```
# Evaluating the function for i = 2, 3, and 4
ET2 = ExpT(2)
ET3 = ExpT(3)
ET4 = ExpT(4)

print(f"ET2/T_total: {ET2/(ET2+ET3+ET4)}")
```

```
print(f"ET2/T_total: {ET3/(ET2+ET3+ET4)}")
print(f"ET2/T_total: {ET4/(ET2+ET3+ET4)}")
print(f"ET4+ET3: {ET4+ET3}" )
```

```
ET2/T_total: 0.6666666666666666
ET2/T_total: 0.2222222222222222
ET2/T_total: 0.1111111111111111
ET4+ET3: 0.5
```

There are no restrictions on topology. Time is in coalescent units.

Part B: Draw a $\pm 1SD$ error bar at each internal node indicating the variance in the TOTAL time to that coalescent event.

Recall that the distribution of coalescent events is given by:

$$Pr[T_i = \tau] = \binom{i}{2} \exp\left(-\binom{i}{2} t\right)$$

Which is an exponential distribution with rate parameter $\binom{i}{2}$. The variance in the coalescent time T_i then is:

$$Var[T_i] = \frac{1}{\binom{i}{2}^2}$$

We can use this to calculate the standard deviation ($\sigma = \sqrt{var}$) in the time to the first coalescent event.

[18]:
```
def ExpT(i):
    return 1/math.comb(i, 2)**2
```

[22]:
```
math.sqrt(ExpT(4))
```

[22]:
```
0.16666666666666666
```

To calculate the error in the total time to the next coalescent event, we have to find $Var(T_3 + T_4)$. Because the coalescent times are independent of each other we have:

$$Var(T_3 + T_4) = Var(T_3) + Var(T_4) = \frac{1}{\binom{4}{2}^2} + \frac{1}{\binom{3}{2}^2} = 0.1389$$

Similarly:

$$Var(T_2 + T_3 + T_4) = Var(T_3) + Var(T_4) = \frac{1}{\binom{4}{2}^2} + \frac{1}{\binom{3}{2}^2} = 1.1389$$

Part C: What is the distribution of times until there are exactly 2 lineages in the population, $\Pr(T_4 + T_3)$? Plot this distribution to double check your answers above.

This distribution is a convolution of two exponential distributions the first with a rate of $\lambda_1 = \binom{4}{2}$ and the second with a rate of $\lambda_2 = \binom{3}{2}$

$$\Pr(T_4 + T_3 = \tau) = \int_0^t \lambda_1 e^{-\lambda_1 x} \lambda_2 e^{-\lambda_2 (t-x)} dx = -\frac{\lambda_1 \lambda_2 \left(e^{\lambda_1 (-t)} - e^{\lambda_2 (-t)}\right)}{\lambda_1 - \lambda_2}$$

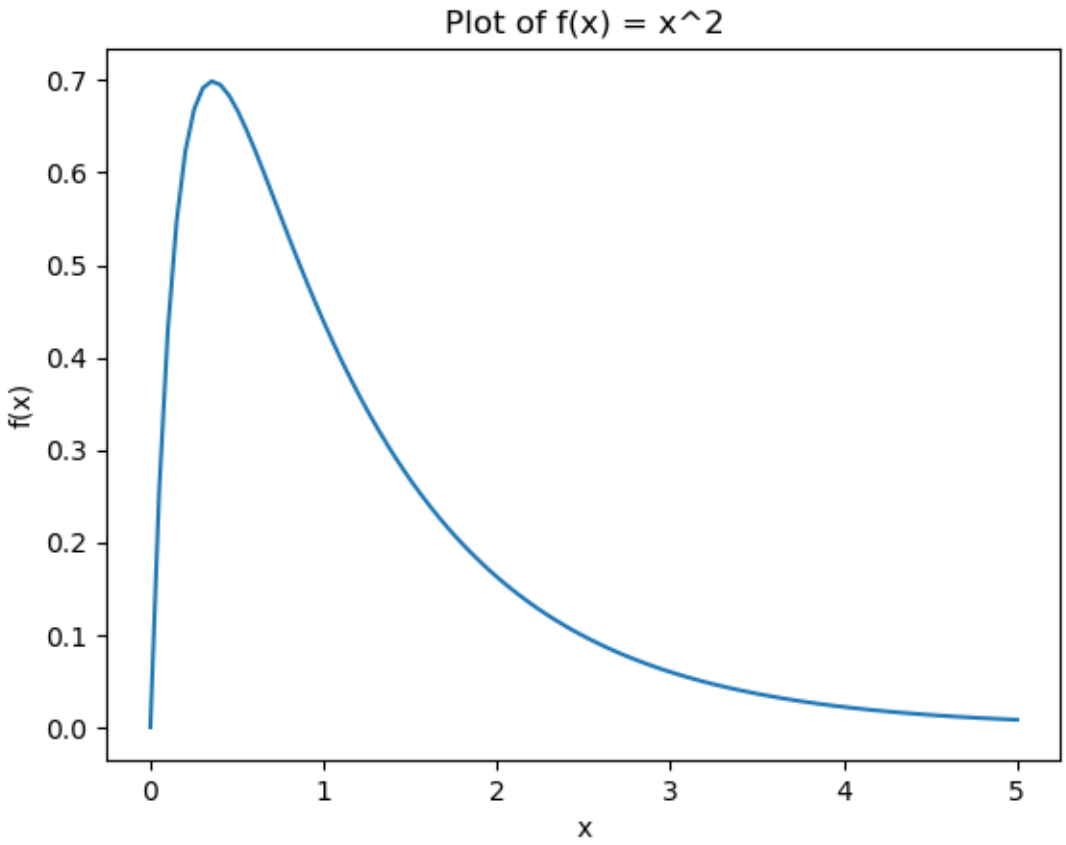
[54]:
```
def fun(t):
    lam1=math.comb(4,2)
    lam2=math.comb(2,2)
    return -lam1*lam2/(lam1-lam2)*(np.exp(-lam1*t)-np.exp(-lam2*t))
```

[52]:
```
fun(0.3)
```

[52]:
```
0.6906231989521575
```

[55]:
```
# Generate x values from 0 to 5
x = np.linspace(0, 5, 100)

# Calculate corresponding y values using the function
y = fun(x)

# Plot the function
plt.plot(x, y, label='f(x) = x^2')

# Add labels and title
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Plot of f(x) = x^2')
```

[55]:
```
Text(0.5, 1.0, 'Plot of f(x) = x^2')
```



$E[X] = 0.5$ and $Var[X] = 0.14$ are not unreasonble.