NUMERICAL ANALYSIS

PROJECT – 4

Name – Mrinal Saini | **Roll No. – BS1920** | **Semester – II**

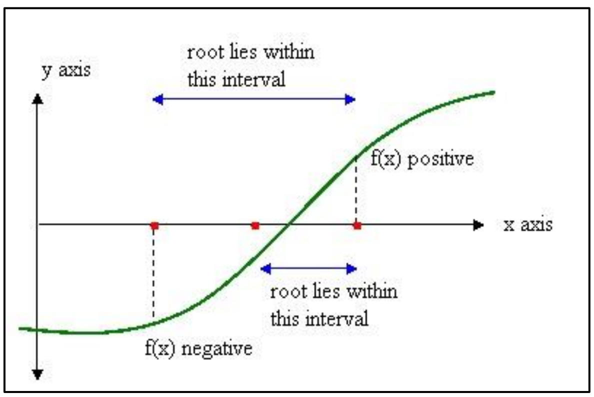
Bisection Method: The Bisection Method is one of the simplest methods for solving equations numerically. It solves the equation $f(x) = 0$, assuming that $f(x)$ is continuous. This method works on the fact that if a function $f(x)$ is real and continuous in an interval $[a, b]$ and $f(a)$ and $f(b)$ are of opposite signs then there is at least one real root of $f(x)$ in $[a, b]$. We start with an appropriate interval $[a_0, b_0]$ satisfying the conditions that $f(x)$ is continuous in $[a_0, b_0]$ and $f(a_0) * f(b_0) < 0$. Now, if

a) $f\left(\frac{a_0+b_0}{2}\right) = 0$, then $\boldsymbol{x} = \frac{\boldsymbol{a_0}+\boldsymbol{b_0}}{\boldsymbol{2}}$ is the root of $f(x)$

b) $f\left(\frac{a_0+b_0}{2}\right) < 0$, then set $\boldsymbol{a_1} = \frac{\boldsymbol{a_0}+\boldsymbol{b_0}}{\boldsymbol{2}}$ and $\boldsymbol{b_1} = \boldsymbol{b_0}$

c) $f\left(\frac{a_0+b_0}{2}\right) > 0$, then set $\boldsymbol{a_1} = \boldsymbol{a_0}$ and $\boldsymbol{b_1} = \frac{\boldsymbol{a_0}+\boldsymbol{b_0}}{\boldsymbol{2}}$

We iterate the process until we get $f\left(\frac{a_i+b_i}{2}\right) = 0$ for some value of i. The geometrical description of the method may be clearly visualized in the figure below:



Project: The file 'data.txt' has $n = 996$ random numbers that are generated from the density

$$f(x; p, a) = \frac{a^p}{\Gamma(p)} x^{p-1} e^{-ax}, x > 0$$

for unknown constants $p, a > 0$. The principle of maximum likelihood estimation suggests estimating p, a by maximizing

$$L(p, a) = \prod_{i=1}^{n} f(x_i; p, a),$$

where $x_1, \dots, x_n$ are the data in the file. Perform this estimation, and check your answer graphically by overlaying the graph of $f(x; p, a)$ on the histogram of the data.

Procedure and Calculations:

1.) Calculate the expression of the Likelihood

$$L(p,a) = \prod_{i=1}^{n} f(x_i; p, a)$$

$$L(p,a) = \left(\frac{a^p}{\Gamma(p)}\right)^n \left(\prod_{i=1}^{n} x_i\right)^{p-1} e^{-p\sum_{i=1}^{n} x_i}$$

2.) Take log on both sides of the equation.

$$\log(L) = (np)\log(a) - a\sum_{i=1}^{n} x_i + (p-1)\log\left(\prod_{i=1}^{n} x_i\right) - (n)\log(\Gamma(p))$$

$$\log(L) = (np)\log(a) - a\sum_{i=1}^{n} x_i + (p-1)\sum_{i=1}^{n}\log(x_i) - (n)\log(\Gamma(p))$$

3.) Calculate the partial derivatives of $\log(L)$ w.r.t a and p

$$\frac{\partial(\log(\mathrm{L}))}{\partial a} = \frac{np}{a} - \sum_{i=1}^{n} x_i$$

$$\frac{\partial(\log(L))}{\partial p} = (n)\log(a) + \sum_{i=1}^{n}\log(x_i) - n\left(\frac{\Gamma'(p)}{\Gamma(p)}\right)$$

4.) Set the partial derivatives of $\log(L)$ equal to zero and solve the equations for **maximum likelihood estimators** of a and p.

$$a = \frac{np}{\sum_{i=1}^{n} x_i}$$

$$\frac{\Gamma'(p)}{\Gamma(p)} = \log(a) + \frac{\sum_{i=1}^{n}\log(x_i)}{n}$$

5.) Substitute the values of a in the second equation.

$$\frac{\Gamma'(p)}{\Gamma(p)} = \log\left(\frac{n}{\sum_{i=1}^{n} x_i}\right) + \log(p) + \frac{\sum_{i=1}^{n}\log(x_i)}{n}$$

6.) Use Bisection Method to solve this equation and find the value of $p(>0)$ which satisfies this equation.

7.) Use this value of p to find the value of a.

8.) Plot the normalized histogram of data such that the total area is 1.

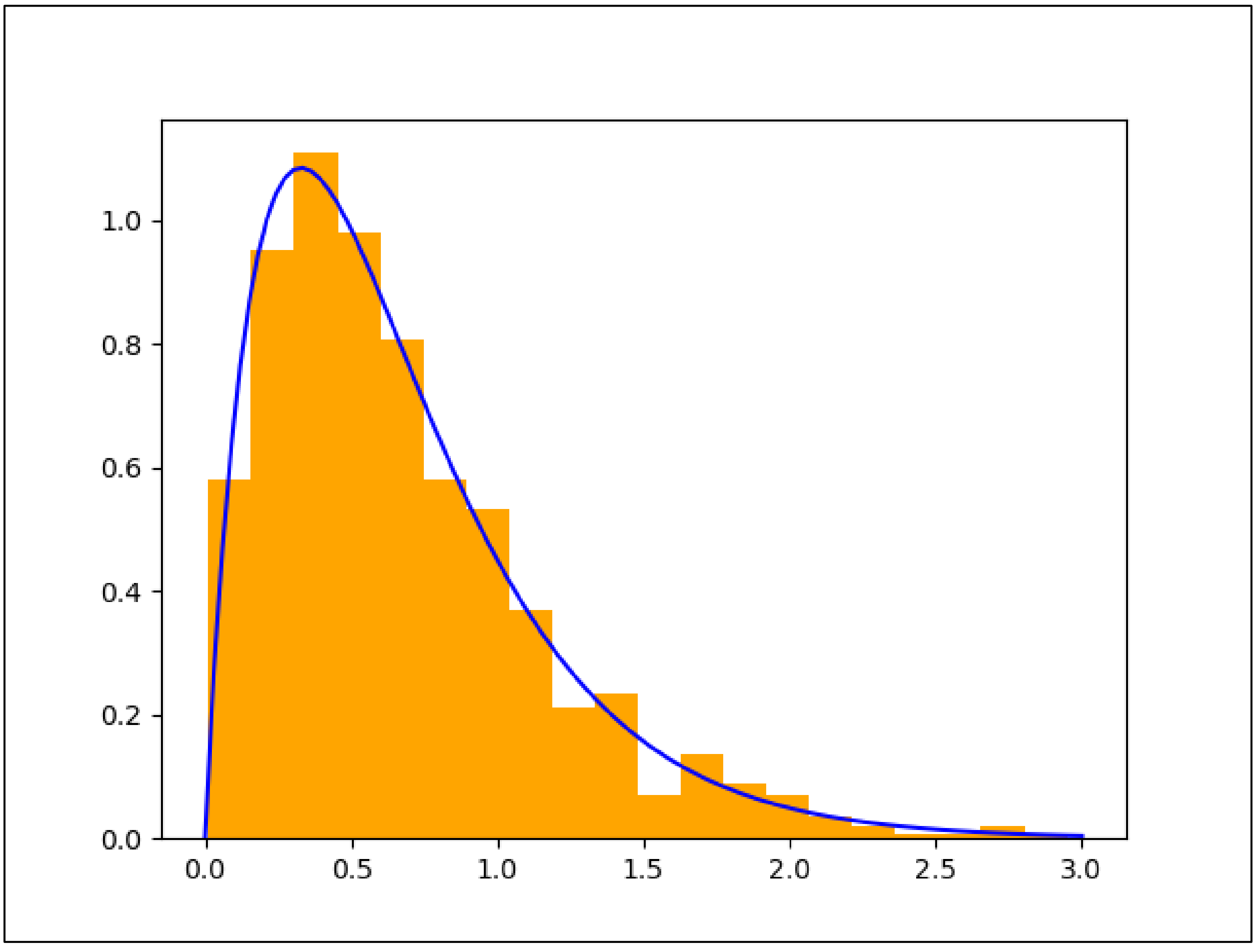
9.) Overlap the graph of $f(x; p, a)$ on this histogram and verify whether the numbers given in the 'data.txt' really follow the given density or not.

Python Code:

```
# PROJECT - 4

## Importing Libraries
import numpy as np
import pandas as pd
from scipy.special import digamma, gamma
from math import log, exp
import matplotlib.pyplot as plt

## Importing the dataset
data=np.array([])
with open("data.txt","r") as f:
    for line in f.readlines():
        f_list=[float(i) for i in line.split(" ") if i.strip()]
        data=np.append(data,f_list)
f.close()

def f(x):
    return digamma(x) - log(np.prod(data))/996 - log(996/sum(data)) - log(x)

## Bisection Method Iterations
def bisection(a,b):
    if f(a)*f(b)>0:
        print("Both f(a\u2080) and f(b\u2080) are of same sign, give different input")
    else:
        tab = pd.DataFrame(columns = ['ai', 'bi', '(ai+bi)/2', 'f(ai+bi)/2'])
        tab = tab.append({'ai':a, 'bi':b, '(ai+bi)/2':(a+b)/2, 'f(ai+bi)/2':f((a+b)/2)}, ignore_index=True)
        while f((a+b)/2)!=0:
            if f((a+b)/2)*f(a)>0:
                a=(a+b)/2
            else:
                b=(a+b)/2
            tab = tab.append({'ai':a, 'bi':b, '(ai+bi)/2':(a+b)/2, 'f(ai+bi)/2':f((a+b)/2)}, ignore_index=True)
        print(tab)
        return (a+b)/2

## Finding the maximum likelihood estimators of a and p
c = float(input("a\u2080 = "))
d = float(input("b\u2080 = "))
if f(c)==0:
    p = c
    print("\np =",p)
    a = 996*p/sum(data)
    print("a =",a)
elif f(d)==0:
    p = d
    print("\np =",p)
    a = 996*p/sum(data)
    print("a =", a)
else:
    p = bisection(c,d)
    if p!=None:
        print("\np =",p)
        a = 996*p/sum(data)
        print("a =", a)

## Visualizing the Results
t = np.linspace(0,3,100)
ft = np.array([])
for i in t:
    ft = np.append(ft, [(a**p)*exp(-a*i)*(i**(p-1))/gamma(p)])
plt.plot(t, ft, color = 'blue')
plt.hist(data, bins = 20, color = 'orange', density = True)
plt.show()
```

Output:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\Notes\Numerical Analysis\Projects\Project4_MLEGamma\Project4_MLEGamma.py
aₒ = 1
bₒ = 2
       ai       bi  (ai+bi)/2  f(ai+bi)/2
0  1.0000  2.00000   1.500000   -0.090619
1  1.5000  2.00000   1.750000   -0.033787
2  1.7500  2.00000   1.875000   -0.011412
3  1.8750  2.00000   1.937500   -0.001376
4  1.9375  2.00000   1.968750    0.003388
5  1.9375  1.96875   1.953125    0.001026
.
.
.
.
.
          ai        bi  (ai+bi)/2    f(ai+bi)/2
49  1.946419  1.946419   1.946419 -1.110223e-16
50  1.946419  1.946419   1.946419  1.110223e-16
51  1.946419  1.946419   1.946419  0.000000e+00

p = 1.9464189208655907
a = 2.8788902855783003
>>>
```



Conclusion: The numbers given in the 'data.txt' file really follow **gamma distribution**.