PROJECT:2

Kirtagya Namdev

21103054

Q1. Monty Hall Paradox: Solve the Monty Hall problem by performing Monte Carlo simulations. Plot the probability of winning for both options for number of simulations ranging between 100 and 1000.

Sol1. Monty Hall asks you to choose one of three doors. One of the doors hides a prize and the other two doors have no prize.

You pick one door out of three door, but you don't open it right away.

Monty opens one of the other two doors, and there is no prize behind it.

At this moment, there are two closed doors, one of which you picked.

The prize is behind one of the closed doors, but you have two choice either you stay with your previous choice or you can switch the door don't know which one. Monty asks you, "Do you want to switch doors?"

So we have to find the probability of winning in with switching the door and and without switch the door.

Using analytical method

Three door are there 1,2,3 in which one door hide a car and other two door hide a goat

Probability of getting car is 1/3 in each door

Now if we switch the door after

Let we P select the door 3

Now monty open the 1 door in which there is a goat

Then probability become

P(m=1/p=1)=0

P(m=1/p=2)=1

P(m=1/p=3)=0.5

Now probability of winning when we switch the door

P(p=2/m=3)= (1*1/3)/(1/2*1/3)+(1*1/3)=2/3

And the probability when we do not switch the door

P(without switching)=1-2/3=1/3

Using Monty carlo simulation

Step 1 import the function in python

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import random
```

step 2 Applying the primary ‘for’ loop in num from range (100,1000)

step 3 First defining the initializing of variable. Applying the ‘for’ loop again inside the primary loop for finding the probability of winning in two condition ‘switching ’ and ‘without switching’.

```
for num in range(100,1000):
    # define the initializing of variable
    win_switch=0
    lose_switch=0
    win=0
    sim_no=0
    lose=0
    for k in range(num):
        doors =["goat","car","goat"]
        sim_no=sim_no+1
        #print("simulation number"+ str(sim_no))
        random.shuffle(doors)
        #print(doors)
        #  if we switch the door winning probability
        n=random.randrange(3)
        if doors[n]=="car":#first choice is car
            lose_switch=lose_switch+1#because u change the choice
            #print("goat")
        if doors[n]=="goat":
            win_switch=win_switch+1
            #print("car")
        # if without swiching the door winning probability
        if doors[n]=="car":#first choice is car
            win=win+1#because u did not change the choice
```

```
            #print("car")
        if doors[n]=="goat":
            lose=lose+1
            #print("goat")
    s=win*100/num
    d=win_switch*100/num
    #print(num,win,lose)
    print(s)
    print(d)
```

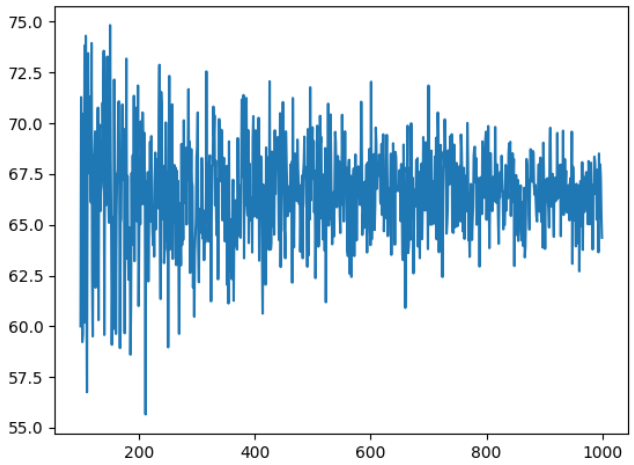
Result

Probability(switching) percentage =(62,67)=around 0.65

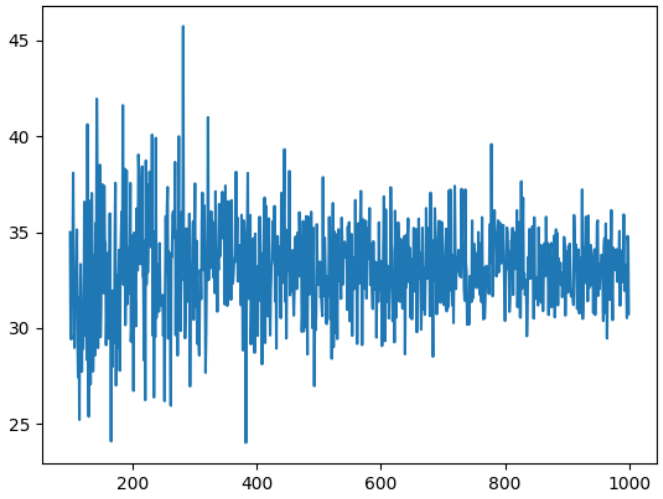
Probability (without switching) percentage=(31,35)=around =.33

step 4. plot the monty calro simulation graph by using import function matplotlib library.

1).Graph of winning probability percentage with switching simulate from100 to 1000



2).Graph of winning probability percentage without switching simulate from100 to 1000.



Comparison as the simulation number increase the result will we more accurate toward the anaytical method answer

Q2. Birthday paradox: How many people must be there is a room before there is 50% chance that two of them were born on the same day of the year. Assume that there are no leap years, i.e. each year has 365 days, and each day of the year is equally probable for a birthday. Solve the problem using Monte Carlo simulations and compare the result with analytical solution.

Sol 2. Analytical Solution

Let suppose 'n' people in a room

Pair of 'n' people by using permutation and combination =$\frac{n(n-1)}{2}$

Probability of 2 people in a room having birthday on same date =$\frac{1}{365}$

Probability of 2 people in a room having birthday not on same date= 1-$\frac{1}{365}$=$\frac{364}{365}$=.9972

The value of n so probability will become 0.5

$.9972^{\frac{n(n-1)}{2}}$ =0.5

Applying log both side

We get $\frac{n(n-1)}{2}\log 0.9972=\log 0.5$

By solving above equation

n(n-1)=497.52

and we get

n=23(we get more than 23)

Monty Carlo simulation method

Step1 import the function in python

```
import statistics
import random
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
```

```
Ymean = []#null list used for plot graph
NumList = []
for num in range(100,1000):# loop on num from 100 to 1000
    runLength = []
    for i in range(num):
        Bday = []
        while True:
            day = random.randint(1, 365)
            if day in Bday:
                break # break when repeatation of number occur while taking random number
(1,365)
            #print(day,Bday)
            Bday.append(day)
        runLength.append(len(Bday))
    print(statistics.mean(runLength))
    Ymean.append(statistics.mean(runLength))

    NumList.append(num)
plt.plot(NumList,Ymean)
plt.show()
```

step 2 Applying the primary ‘for’ loop in num from range (100,1000)

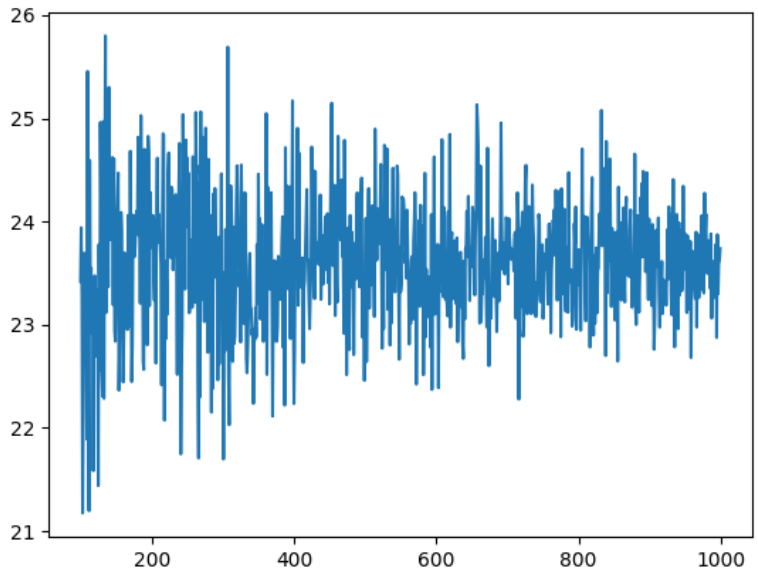
step 3 Again apply ‘for’ loop inside num ‘for’ loop in which we randomly take the value from 365 day

step 4 we will apply ‘while’ loop and break it when the repetition of number occur in randomly selecting the number from 365 days.

Step 5 we print mean of runlength of num so that we get the value mean for which repetition of day occur on such number of selecting in a room.

Result=(23,24)

Step 6 Plotting the graph of the x= number of simulation (100,1000) and y=number of mean people in a room



Q3. Indian Air Force deployed a C-130J Super Hercules aircraft for an emergency evacuation of 300 Indians stuck in a foreign country. It is estimated the weight of the evacuees, including their luggage, is uniformly distributed between 25 kg and 100 kg. If the maximum payload specified for the aircraft is 19,000 kg, estimate the probability that the total weight of the evacuees will exceed the maximum payload.

Sol3. Analytical method

Let x represent weight of an evacuee

X is uniform distributed (25,100)

Mean and variance of an evacuee

Mean = $\frac{25+100}{2}$=62.5

Variance=$\frac{(100-25)^2}{12}$=468.75

Central limit theorem apply for n=300 passenger

Probability exceeding 19000kg

P(s_n>19000)=P$\left(\frac{s_n - n*mean}{\sqrt{n*variance}} > \frac{19000-n*mean}{\sqrt{n*variance}}\right)$

=P($Z > .0667$)

From standard normal table

P($Z > .0667$)=1-=P($Z < .0667$)

1-0.7485

0.2515 answer

Monty carlo method

Step 1 importing the library in python

Step 2 First applying the 'for' loop for simulation from 100 to 1000

Step 3 Then again applying loop on num

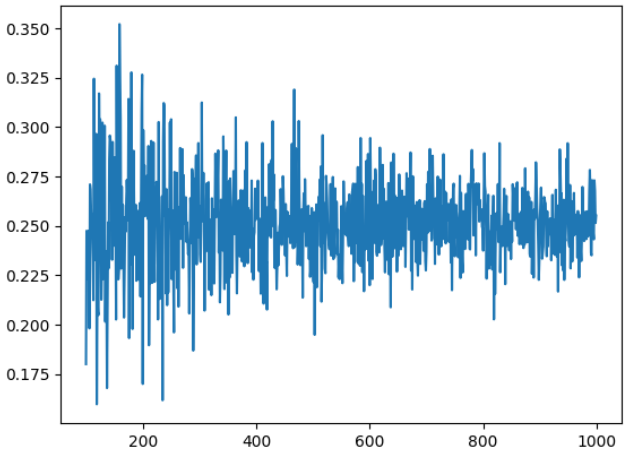
Step 4 we randomly select the 300 number between 0 to 1 and find the X which is the weight of a person and sum the weight of person that we have selected

Step 5 finding the probability that weight sum is less than 19000

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import statistics
import numpy as np


#num = 100
NumList = []
ansList = []
for num in range(100,1000):
    total19000 = 0
    for i in range(num):
        men = stats.uniform.rvs(0,1,size=300)
        x = men*75+25
        weightSum = sum(x)

        if weightSum > 19000:
            total19000 = total19000 + 1

    #print(total19000/num)
    ansList.append(total19000/num)
    NumList.append(num)
plt.plot(NumList,ansList)
plt.show()
```

Result= around 0.25

Step 6. plot the graph between Nsimulation and probability when weight is less than 19000



Q4. The lifetime of an LED floodlight used in a stadium is uniformly distributed in the range 2 to 12 years. The floodlight will be replaced upon failure or when it reaches the age of 10 years, whichever occurs first. What is the expected value of the age of the flood light at the time of replacement?

Sol 4.let the age of the LED flood light be given by X

X is uniform distributed (2,12)

The age of LED flood light at the time of replacement be Y

f_y=$\begin{cases} 0.1\,, 2 \le y \le 10 \\ 0.2\,, \quad y = 10 \end{cases}$

The mean of Y can be calculate

Mean of y = E[y] = $\int_{-\infty}^{\infty} y\, f_y(y) dy$

Now integrate

=$\int_{2}^{10} 0.1 y dy$ +0.2*10

=6.8 year answer

Monty carlo method

Step 1 importing the library in python

Step 2 Number of simulation equal to 10000 ,as no of simulation increase the answer will we more perfect .

Step 3 Randomly selecting the number between 0 to 1 Nsim times.

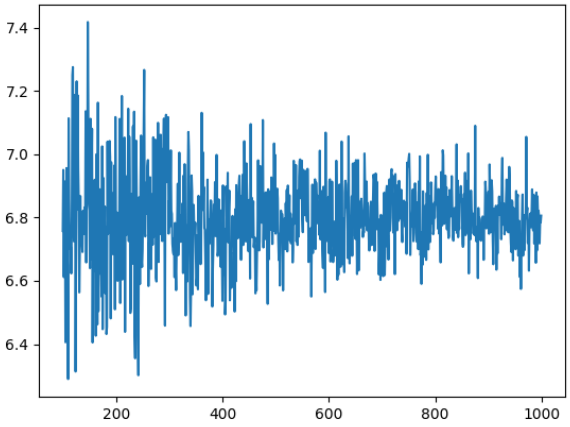
Step 4 Apply ‘for’ loop on x and take the value if x is less than 10 otherwise we will replace after 10 year completed which will we represent by y

Step 5 now taking the mean of y

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import statistics
import numpy as np

Nsim = 10000
u= stats.uniform.rvs(0,1,size=Nsim)
x=u*10+2
y = []
for x1 in x:
    y.append(min(x1,10))

print(statistics.mean(y))
```

Answer mean of y= 6.779165937593668

Step 6 plot the graph between y mean and Nsimulation



Q5. . The pmf of an integer random variable X is given by

f_y=$\begin{cases}\frac{1}{9}, & [-4,4] \\ 0, otherwise\end{cases}$

Derive the pmf for $Y = |X|$. The mean and variance of Y will be

Sol 5.

$Y = |X|$

The sample space for Y={0, 1, 2, 3, 4}

Pmf of Y = $\begin{cases}\frac{2}{9}, if\ y = 1,2,3,4 \\ \frac{1}{9}, if\ y = 0 \\ 0, othrewise\end{cases}$

Y	p_y(y)	yp_y(y)	y^2p_y(y)
0	1/9	0	0
1	2/9	2/9	2/9
2	2/9	4/9	8/9
3	2/9	6/9	18/9
4	2/9	8/9	32/9

Answer

Mean =$\sum yp_y$(y)=20/9

E(y^2)= $\sum y^2p_y$(y)=60/9

Variance=140/81

Monty carlo simulation

Step 1 import the library in python

Step 2 first applying the ‘for’ loop for simulation (100,1000)

Step 3 Now generating the Y null list and applying the for loop in num number of time on Y

Step 4 taking the mean and variance of Y

```
import statistics
import random
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt


Xvals = [ -4 , -3 , -2 , -1 , 0 , 1 , 2 , 3 , 4 ]
Ymean = []
Yvar = []
NumList = []
for num in range(100,1000):
    Yvals = []
    for i in range(num):
        Yvals.append(abs(random.choice(Xvals)))
    #print(statistics.mean(Yvals), statistics.pvariance(Yvals))
    Ymean.append(statistics.mean(Yvals))
    Yvar.append(statistics.pvariance(Yvals))
    NumList.append(num)

plt.plot(NumList,Ymean)
plt.show()
```

Result

Mean=around 2.12

Variance=around 1.73

Step 5 plot the graph of mean of wrt no of simulation