CSC 521

Take Home Midterm

Jonggoo Kang

In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from termcolor import colored
from itertools import combinations
import math
import random
```

Problem 1

In [2]:

```python
df = pd.read_csv('problem1.csv', names = ['X','Y','Z'])
df.head()
```

Out[2]:

	X	Y	Z
0	13.242098	21.650022	27.425479
1	9.434795	20.916445	7.025783
2	15.485839	21.733990	14.963390
3	7.730181	16.332673	4.203992
4	18.166409	21.486609	28.191421

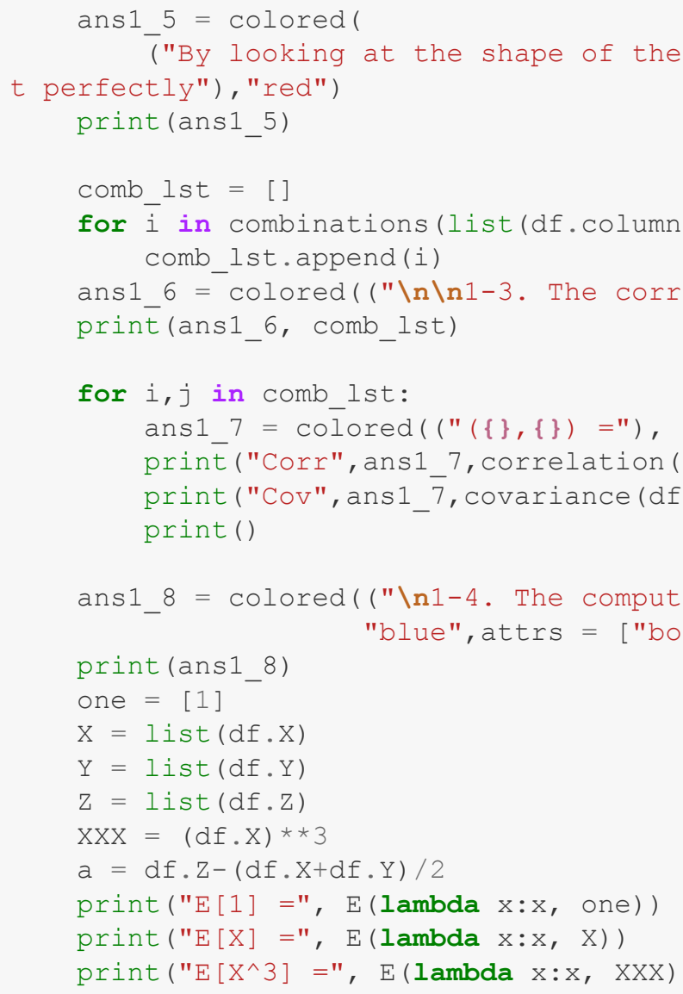
In [3]:

```python
def E(f,S): return float(sum(f(x) for x in S))/(len(S) or 1)
def mean(X): return E(lambda x:x, X)
def variance(X): return E(lambda x:x**2, X) - E(lambda x:x, X)**2
def sd(X): return math.sqrt(variance(X))
def covariance(X,Y):
    return sum(X[i]*Y[i] for i in range(len(X)))/len(X) - mean(X)*mean(Y)
def correlation(X,Y): return covariance(X,Y)/sd(X)/sd(Y)

def prob1(df):
    Mean = []
    Var = []
    Std = []
    for i in list(df.columns):
        Mean.append(mean(df[i]))
        Var.append(variance(df[i]))
        Std.append(sd(df[i]))
    ans1 = colored(("1-1. Mean, Var, Std for column X, Y, and Z are as follow:"),
                   "blue", attrs = ["bold"])
    print(ans1)
    ans1_1 = colored(("Means"), "blue")
    ans1_2 = colored(("Variances"), "blue")
    ans1_3 = colored(("Standard deviations"), "blue")
    print(ans1_1, Mean)
    print(ans1_2, Var)
    print(ans1_3, Std)

    ans1_4 = colored(("\n1-2. The distribution of each columns are as follow:"),
                     "blue",attrs = ["bold"])
    print(ans1_4)
    df.hist(figsize=[8,6])
    plt.show()
    ans1_5 = colored(
        ("By looking at the shape of the histogram, it looks like a normal distribution, but no
t perfectly"),"red")
    print(ans1_5)

    comb_lst = []
    for i in combinations(list(df.columns),2):
        comb_lst.append(i)
    ans1_6 = colored(("\n\n1-3. The correlation and covariance for"), "blue",attrs = ["bold"])
    print(ans1_6, comb_lst)

    for i,j in comb_lst:
        ans1_7 = colored(("({},{}) ="), "blue").format(i,j)
        print("Corr",ans1_7,correlation(df[i],df[j]))
        print("Cov",ans1_7,covariance(df[i],df[j]))
        print()

    ans1_8 = colored(("\n1-4. The computed expectation values are as follow:"),
                     "blue",attrs = ["bold"])
    print(ans1_8)
    one = [1]
    X = list(df.X)
    Y = list(df.Y)
    Z = list(df.Z)
    XXX = (df.X)**3
    a = df.Z-(df.X+df.Y)/2
    print("E[1] =", E(lambda x:x, one))
    print("E[X] =", E(lambda x:x, X))
    print("E[X^3] =", E(lambda x:x, XXX))
    print("E[Z-(X+Y)/2] =",(E(lambda x:x, a)))

    ans1_9 = colored(("\n1-5. The formulars for the mean, varaiance, covariance, and correlatio
n\
    \nin terms of the expectation values are as follow:"), "blue",attrs = ["bold"])
    print(ans1_9)
```

In [4]:

```python
prob1(df)
```

```
1-1. Mean, Var, Std for column X, Y, and Z are as follow:
Means [9.93706828076954, 20.217017423770194, 14.862678823134607]
Variances [27.009135786469884, 24.22520896822499, 37.84152597711147]
Standard deviations [5.197031439819263, 4.9219111093380175, 6.151546632929923]

1-2. The distribution of each columns are as follow:
```



```
By looking at the shape of the histogram, it looks like a normal distribution, but not perfe
ctly


1-3. The correlation and covariance for [('X', 'Y'), ('X', 'Z'), ('Y', 'Z')]
Corr (X,Y) = -0.05040109245653239
Cov (X,Y) = -1.2892260139755933

Corr (X,Z) = 0.3999697204931487
Cov (X,Z) = 12.786944472729914

Corr (Y,Z) = 0.321761269269099
Cov (Y,Z) = 9.742083621691393


1-4. The computed expectation values are as follow:
E[1] = 1.0
E[X] = 9.93706828076954
E[X^3] = 1777.8614071677903
E[Z-(X+Y)/2] = -0.21436402913524902

1-5. The formulars for the mean, varaiance, covariance, and correlation
in terms of the expectation values are as follow:
```

Mean

$$\mu_x = E(X)$$

Variance

$$\sigma_x^2 = E[(X - \mu_x)^2]$$

Covariance

$$cov(X, Y) = E[XY] - E[X]E[Y]$$

Correlation

$$corr(X, Y) = \frac{cov(X, Y)}{(\sigma_x \sigma_y)}$$

Problem 2

Consider a continuous random variable in the range[2,3] with probability mass function propotional to exp(x/2)

Given the condition,

$$p(x) = c * e(\frac{x}{2})$$

, where x is in [2,3]

Now,

$$F(X) = \int_{-\infty}^{x} p(x)dx = \int_{2}^{x} c * e(\frac{x}{2})dx = 2c[e^{\frac{x}{2}} - e]$$

$$F(3) = 1$$

implies

$$c = \frac{1}{2(e^{\frac{3}{2}} - e)} \approx 0.2835$$

Therefore,

$$p(x) = \frac{e^{\frac{x}{2}}}{2(e^{\frac{3}{2}} - e)}$$

To compute with E[1], E[x], E[x^2], E[x^3]

$$E(1) = \int_{2}^{3} c * e(\frac{x}{2})dx = 2c(e^{\frac{3}{2}} - e) = 0.99999 \approx 1$$

$$E[X] = \int_{2}^{3} cxe^{\frac{x}{2}}dx = ce^{\frac{x}{2}}(2x - 4)\Big|_{2}^{3} = 2.54148$$

$$E[X^2] = \int_{2}^{3} cx^2e^{\frac{x}{2}}dx = ce^{\frac{x}{2}}(2x^2 - 8x + 16)\Big|_{2}^{3} = 6.54145$$

$$E[X^3] = \int_{2}^{3} cx^3e^{\frac{x}{2}}dx = ce^{\frac{x}{2}}(2x^3 - 12x^2 + 48x - 96)\Big|_{2}^{3} = 17.0393$$

Problem 3

Consider a continuous various random variable in the range [1,5] with probability mass function proportional to 1/x.

What is the cumulative distribution function?

$$p(x) = c\frac{1}{x}, x \in [1, 5]$$

$$F(x) = \int_{-\infty}^{x} p(x)dx = c\int_{1}^{x} \frac{1}{x}dx = c(ln(x) - ln(1))$$

$$F(5) = 1 = c(ln(5) - ln(1))$$

$$C = 0.62133$$

$$p(x) = \frac{dF(x)}{dx} = 0.62133\frac{1}{x}, x \in [1, 5]$$

$$F(x) = 0.62133ln(x)$$

In [5]:

```python
C = 1/(math.log(5) - math.log(1))
C
def prob3():
    rand_10 = []
    rand_100 = []
    rand_1000 = []

    for i in range(10):
        y = -(float(C*math.log(random.random())))**(-1)
        if y >= 1 and y <= 5:
            rand_10.append(y)

    for i in range(100):
        y = -(float(C*math.log(random.random())))**(-1)
        if y >= 1 and y <= 5:
            rand_100.append(y)

    for i in range(1000):
        y = -(float(C*math.log(random.random())))**(-1)
        if y >= 1 and y <= 5:
            rand_1000.append(y)

    one = [1]
    X = rand_10
    XX = []
    for i in X:
        XX.append(i**2)
    XXX = []
    for i in X:
        XXX.append(i**3)
    title = colored("When range(10)", "blue")
    print(title)
    print("E[1] =", E(lambda x:x, one))
    print("E[X] =", E(lambda x:x, X))
    print("E[X^2] =", E(lambda x:x, XX))
    print("E[X^3] =", E(lambda x:x, XXX))

    one = [1]
    X = rand_100
    XX = []
    for i in X:
        XX.append(i**2)
    XXX = []
    for i in X:
        XXX.append(i**3)
    title = colored("\nWhen range(100)", "blue")
    print(title)
    print("E[1] =", E(lambda x:x, one))
    print("E[X] =", E(lambda x:x, X))
    print("E[X^2] =", E(lambda x:x, XX))
    print("E[X^3] =", E(lambda x:x, XXX))

    one = [1]
    X = rand_1000
    XX = []
    for i in X:
        XX.append(i**2)
    XXX = []
    for i in X:
        XXX.append(i**3)
    title = colored("\nWhen range(1000)", "blue")
    print(title)
    print("E[1] =", E(lambda x:x, one))
    print("E[X] =", E(lambda x:x, X))
    print("E[X^2] =", E(lambda x:x, XX))
    print("E[X^3] =", E(lambda x:x, XXX))
```

In [6]:

```python
prob3()
```

```
When range(10)
E[1] = 1.0
E[X] = 2.204706561233817
E[X^2] = 5.2880367564056305
E[X^3] = 13.800958387233635

When range(100)
E[1] = 1.0
E[X] = 2.403370147475276
E[X^2] = 6.922273762689239
E[X^3] = 23.05510268166969

When range(1000)
E[1] = 1.0
E[X] = 2.3864317535427695
E[X^2] = 6.85573276662736
E[X^3] = 22.636828658646117
```

Problem 4

Assume you receve phone calls with an average of 10 per hour. What is the probablity of receving two phone calls less than 1 minute apart wthin any one day? Write a program that computes the answer. Explain your reasoning and your program.

$$m = 1/6$$

, per minute

In [7]:

```python
def prob4():
    count = 0
    lst = []
    for i in range(500):
        lst.append(random.expovariate(1/6))
    ans4 = colored("TIME GAP", "red")
    ans4_1 = colored("random.expovariate(1/6)", "blue")
    print("Let",ans4_1, "be the", ans4,"between two phone calls\n")
    print("Generate the ENOUGH samples of",ans4_1)
    print("( I made 500 samples of", ans4_1,")")
    ans4_2 = colored("1440", "blue")

    print("\nNow, add each",ans4_1,"repeatly until the total sum equals to",ans4_2,"(= minutes
 in a day)")

    lst1=[]
    summ = 0
    sum_cnt =0
    for i in lst:
        summ += i
        sum_cnt += 1
        lst1.append(i)
        if summ >= 1440:
            break
    ans4_3 = colored(summ, "blue")
    print("By adding them up, the total sum is calculated as", ans4_3,)
    print("and this satisfies to be the minutes in a day\n")

    ans4_4 = colored("each number of the addition","red")
    ans4_5 = colored("TOTAL PHONE CALLS IN A DAY", "red")
    ans4_6 = colored(sum_cnt, "blue")
    print("While adding them up,",ans4_4,"is counted.")
    print("The total count of",ans4_4,"is considered as",ans4_5)
    print("and",ans4_5,"is calculated as", ans4_6)

    cnt = 0
    for i in lst1:
        if i < 1:
            cnt  += 1
    ans4_7 = colored("Number of phone calls less than 1 minute apart from each other","blue")
    ans4_8 = colored(cnt,"blue")
    print("\nAmong",ans4_5,ans4_6)
    print(ans4_7,"is seperately counted and the count of this is",ans4_8)
    ans4_9 = colored(cnt/sum_cnt, "blue")
    print("\nThe probability of recieving two phone calls less than 1 minute apart could be",
        cnt,"/",sum_cnt,"=",ans4_9)
    print("\nTherefore,",ans4_9)
```

In [8]:

```python
prob4()
```

```
Let random.expovariate(1/6) be the TIME GAP between two phone calls

Generate the ENOUGH samples of random.expovariate(1/6)
( I made 500 samples of random.expovariate(1/6) )

Now, add each random.expovariate(1/6) repeatly until the total sum equals to 1440 (= minutes
 in a day)
By adding them up, the total sum is calculated as 1442.0099034458904
and this satisfies to be the minutes in a day

While adding them up, each number of the addition is counted.
The total count of each number of the addition is considered as TOTAL PHONE CALLS IN A DAY
and TOTAL PHONE CALLS IN A DAY is calculated as 239

Among TOTAL PHONE CALLS IN A DAY 239
Number of phone calls less than 1 minute apart from each other is seperately counted and the
 count of this is 43

The probability of recieving two phone calls less than 1 minute apart could be 43 / 239 = 0.
1799163179916318

Therefore, 0.1799163179916318
```

Problem 5

In [9]:

```python
class RandomSource(object):

    def __init__(self,generator=None):
        if not generator:
            import random as generator
        self.generator = generator

    def random(self):
        return self.generator.random()

    def lookup(self,table, epsilon=1e-6):
        if isinstance(table,dict): table = table.items()
        u = self.random()
        for key,p in table:
            if u<p+epsilon:
                return key
            u = u - p
        raise ArithmeticError('invalid probability')

def table(n):
    table = []
    room = 1
    a = 1
    for i in range(n):
        table.append((room, a * 0.5))
        room +=1
        a = a*0.5
    return table

table = table(1000)

def test_lookup(nevents,table):
    g = RandomSource()
    f=[]
    for i in range(nevents):
        f.append(0)

    p_over4 = 0
    p_less4 = 0
    for k in range(nevents):
        p = g.lookup(table)
        #########################
        # This code counts total number of people who entereed the office
        # when 40 people entered the room #4 or greater.
        if p < 4:
            p_less4 += 1
        else:
            p_over4 += 1
            if p_over4 == 40:
                less4 = colored(p_less4, "blue")
                over4 = colored(p_over4, "blue")
                total = colored(p_less4+p_over4, "blue")


                explain_1 = "While generating enough samples (1000 samples) by "
                eq1 = colored("p=g.lookup(1000)", "blue")
                explain_2 = ",\ntwo counting variables, "
                eq2 = colored("p >= 4 ","blue")
                eq3 = colored("p < 3","blue")
                explain_3 = "and "
                explain_4 = " are counted.\n\nAt the moment the counting variable "
                explain_5 = "equals to 40, the program displays"
                explain_6 = " the total counts of the variable "
                explain_7 = "So when the program shows "
                explain_8 = "equals to 40, the total counts of the variable "
                explain_9 = " equals to "
                explain_10 = "\nTherefore, "

                ans5_3_1 = explain_1 + eq1 + explain_2 + eq2 + explain_3 + eq3
                ans5_3_2 = explain_4 + eq2 + explain_5 + explain_6 + eq3
                ans5_3_3 = explain_7 + eq2 + explain_8 + eq3 + explain_9 + less4
                ans5_3_4 = explain_10 +less4 + "+" + over4 +"="+ total

        ################################
        #print(p),
        f[p-1]=f[p-1]+1
    print(f,"\n")
    printt = colored("I just removed the rooms with zero probability", "red")
    print(printt)
    avg_room = []
    for i in range(len(table)):
        f[i]=float(f[i])/nevents
        if f[i] != 0.000000:
            pass #############
            print('\tfrequency[%i]=%f' % (i+1,f[i]))
        avg_room.append((i+1) * f[i]*nevents)

    ans5_2 = colored(sum(avg_room)/len(avg_room), "blue")
    print("\nThe average room number is",ans5_2,
          "calculated by\nsumming up all frequency by room numbers"
          ", which is", sum(avg_room),
          "\nThen, dividing by the total number of frequency, which is"
          ,len(avg_room))
    print(sum(avg_room),"/",len(avg_room),"=", sum(avg_room)/len(avg_room))
    title3 = colored(("\nProblem 5-3"), "blue", attrs = ["bold"])
    q3 = colored("how many people enter rooms with the number greater or equal to 4","blue")
    print(title3,q3)
    print(ans5_3_1, ans5_3_2,ans5_3_3,ans5_3_4)
```

In [10]:

```python
def prob5():
    title1 = colored(("Problem 5-1"), "blue", attrs = ["bold"])
    q1 = colored("the average number of people who entere the first room if 100 people enter",
"blue")
    print(title1, q1)
    print("E[x] = np = 100 * 0.5 =", 100*0.5,"\n")
    title2 = colored(("Problem 5-2"), "blue",attrs = ["bold"])
    q2 = colored("the average room number","blue")
    print(title2,q2)
    test_lookup(1000,table)
    title4 = colored(("\nProblem 5-4"), "blue", attrs = ["bold"])
    print(title4)
prob5()
```

```
Problem 5-1 the average number of people who entere the first room if 100 people enter
E[x] = np = 100 * 0.5 = 50.0

Problem 5-2 the average room number
[496, 236, 139, 53, 44, 17, 6, 4, 3, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

I just removed the rooms with zero probability
	frequency[1]=0.496000
	frequency[2]=0.236000
	frequency[3]=0.139000
	frequency[4]=0.053000
	frequency[5]=0.044000
	frequency[6]=0.017000
	frequency[7]=0.006000
	frequency[8]=0.004000
	frequency[9]=0.003000
	frequency[10]=0.001000
	frequency[12]=0.001000

The average room number is 2.042 calculated by
summing up all frequency by room numbers, which is 2042.0
Then, dividing by the total number of frequency, which is 1000
2042.0 / 1000 = 2.042

Problem 5-3 how many people enter rooms with the number greater or equal to 4
While generating enough samples (1000 samples) by p=g.lookup(1000),
two counting variables, p >= 4 and p < 3  are counted.

At the moment the counting variable p >= 4 equals to 40, the program displays the total coun
ts of the variable p < 3 So when the program shows p >= 4 equals to 40, the total counts of
 the variable p < 3 equals to 287
Therefore, 287+40=327

Problem 5-4
```

Since the probability that entering room 1,2,and3 is

$$0.5 + 0.25 + 0.125 = 0.875$$

The probability that the entering the room with the number greater or equal to 4 is 0.125.

$$40 : 0.125 = x : 0.875$$

$$\therefore x * (0.125) = (40)(0.875)$$

$$\therefore x = (40)(0.875)(\frac{1}{0.125})$$

$$\therefore x = 280$$

Therefore, 280 + 40 = 320 people entered the room

In [14]:

```python
print("done")
```

```
done
```