IC252-Data Science II

Lab 4

By Adarsh Santoria(B21176)

Code is displayed and in terminal output is shown with examples as mentioned in question

1. Independence and conditional probability. Let X1 and X2 be independent random variables, taking values from the set {0, 1}, such that both outcomes are equally likely. Define Z = X1 + X2. Note that Z can take values {0, 1, 2}. Do the following.

• Demonstrate by counting, that the X1 and X2 are independent. Generate X1 and X2 a large number of times (call this N). Count the number of times X1 takes the value 1, X2 takes the value 1, and the number of times both X1 and X2 take the value 1. This should be approximately equal. Compute the probability by hand and compare with the result of the simulation. X1 and X2 can be stored in two arrays.

• Now generate Z, using the already computed values of X1 and X2. Is Z independent of X1? Determine using counting, and by hand.

• Now condition X1 and X2 on Z = 1. Is X1 conditioned on Z, independent of X2 conditioned on Z? In other words, is $P(X1=1, X2=1|Z = 1) = P(X1=1|Z = 1)P(X2=1|Z = 1)$? Demonstrate by counting and calculate by hand

1) (a) $X_1 = \{0,1\}$, $X_2 = \{0,1\}$

$Z = \{0,1,2\}$

$P(X_1=1) = 1/2 = 0.5$

$P(X_2=1) = 1/2 = 0.5$

$P(Z=1) = 2/4 = 1/2 = 0.5$

(b) Let, assume Z is independent of X_2

then, $P(X_2=\alpha) \cdot P(Z=\alpha) = P(X_2=\alpha = Z)$

$\forall$ possible value of α

~~let~~ for $\alpha=1$

$1/2 \times 1/2 = 1/4$

for $\alpha=0$

$1/2 \times 1/4 \neq 1/4$ $\therefore$ Our assumption contradicts

$\therefore$ 'Z' is not independent of X_2

(c) $P(X_1=X_2 \mid Z=1) = 0$ ...(i)

$P(X_1=1 \mid Z=1) = 1/2 \times 1/2 = 1/4$ ...(ii)

$P(X_2=1 \mid Z=1) = 1/2 \times 1/2 = 1/4$ ...(iii)

As product of (ii) & (iii) is not equals to (i),

X_1 & X_2 are not independent conditioning Z.

Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP

C:\Users\HP\untitled20.py

2.py | untitled3.py | untitled14.py | untitled15.py | untitled18.py | untitled19.py | untitled20.py*

```
from random import randint as r #importing modules
Z=[]                            #making lists and appending random values
X1=[]                           #with the help of for loop
X2=[]                           #N is number simulations
N=int(input("number of times the loop needs to run:"))
for i in range(N):
    X1.append(r(0,1))
    X2.append(r(0,1))
    Z.append(X1[-1]+X2[-1])     #Z is sum of outcome of X1 & X2
'''a'''                         #desired probabilities are obtained by using
print("P((X1=1 AND X2=1)):",Z.count(2)/N)     #probability formula
print("P(X1=1)",X1.count(1)/N)
print("P(X2=1)",X2.count(1)/N)
'''b'''                         #checking Z is independent of X1 or not
print("P(X1=1)*P(Z=1):",(X1.count(1)/N)*(Z.count(1)/N))
count=0
for i in range(N):              #taking one case when X1=Z=1
    if X1[i]==1 and Z[i]==1:
        count+=1
print("P(X1=1 AND Z=1):",count/N)   #There is a mismatch
print("Hence,Z is not independent of X1")
'''c'''
c1=0                           #outcomes are counted inside samplespace
c2=0                           #when Z is 1
c3=0
for i in range(N):
    if(X1[i]==1) and (X2[i]==1) and (Z[i]==1):
        c1=c1+1
    if(X1[i]==1) and (Z[i]==1):
        c2=c2+1
    if(X2[i]==1) and (Z[i]==1):
        c3=c3+1
print("P(X1=1,X2=1 | Z=1):",c1/Z.count(1))
print("P(X1=1 | Z=1)*P(X2=1 | Z=1):",(c2/Z.count(1))*(c3/Z.count(1)))
print("Hence,X1 & X2 are not independent under condition of Z")
```

Nam	Type	Size	Value
c1	int	1	0
c2	int	1	23
c3	int	1	23
count	int	1	23
i	int	1	99
N	int	1	100
X1	list	100	[1, 1, 0, 1, 1, 1, 1, 0, 1, 0, ...]
X2	list	100	[1, 0, 0, 0, 0, 1, 0, 1, 1, 1, ...]
Z	list	100	[2, 1, 0, 1, 1, 2, 1, 1, 2, 1, ...]

Help | Variable Explorer | Plots | Files

Console 3/A

```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/HP/untitled20.py', wdir='C:/Users/HP')

number of times the loop needs to run:100
P((X1=1 AND X2=1)): 0.26
P(X1=1) 0.49
P(X2=1) 0.49
P(X1=1)*P(Z=1): 0.22540000000000002
P(X1=1 AND Z=1): 0.23
Hence,Z is not independent of X1
P(X1=1,X2=1 | Z=1): 0.0
P(X1=1 | Z=1)*P(X2=1 | Z=1): 0.25
Hence,X1 & X2 are not independent under condition of Z

In [2]:
```

IPython Console | History

2. Simulate the number of heads obtained in N independent throws of a coin with p(H) = p. Accept N and p form the user. Run the experiment 10,000 times and plot the histograms of the number of heads obtained, for various values of N and p. Each N, p will need a separate histogram.

Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP

C:\Users\HP\untitled21.py

untitled14.py | untitled15.py | untitled18.py | untitled19.py | untitled20.py | untitled21.py*

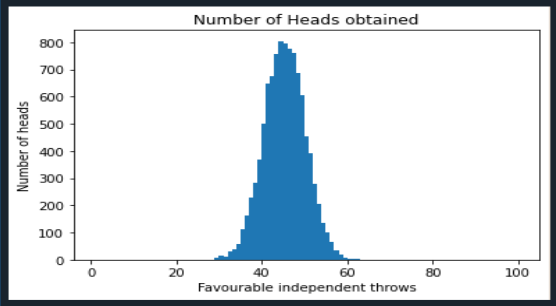
```
1   from matplotlib import pyplot as plt
2   import random as r          #importing modules
3   C=[]
4   #C is a list that consists number of heads on simulating different times
5   N=int(input("number of indipendent throws:"))
6   p=float(input("probability of getting a Head:"))
7   S=[1]*int(100*N*p)+[0]*int(100*N*(1-p))    #adjusting probability of Head
8   for i in range(10000):                     #running 10000simulations
9       x=[]
10      for j in range(N):
11          x.append(r.choice(S))
12      C.append(x.count(1))
13  fig,ax = plt.subplots()
14  #plotting the graph the results of all simuations
15  plt.title('Number of Heads obtained')
16  plt.xlabel('Favourable independent throws')
17  plt.ylabel('Number of heads')
18  ax.hist(C, bins=list(range(1, N+1)))
19
20
```

Nam	Type	Size	
ax	axes._subplots.AxesSubplot	1	AxesSubplot object of matplotlib.axes._
C	list	10000	[49, 41, 46, 37, 40, 44, 42, 42, 46, 42
c1	int	1	0
c2	int	1	23
c3	int	1	23
count	int	1	23
fig	figure.Figure	1	Figure object of matplotlib.figure modu

Help | Variable Explorer | Plots | Files

Console 3/A

```
In [2]: runfile('C:/Users/HP/untitled21.py', wdir='C:/Users/HP')
number of indipendent throws:100
probability of getting a Head:0.45
```




```
In [3]:
```

IPython Console | History

LSP Python: ready | custom (Pyth

Type here to search

26°C Mostly sunny

3. Count the number of times a message needs to be transmitted until it reaches correctly at the destination. The probability for a successful transmission is p. Repeat the experiment 10,000 times and plot the histograms of the count for various values of p. Each p will need a separate histogram. Hint: Generate a random bit b with P(b = 1) = p. Keep adding to the count until you generate a 1. A successful transmission at the 4th try (ie. count = 4) corresponds to the binary string 0001.