IEOR 241 : Homework 9

Arnaud Minondo

November 1, 2022

For each of the exercises you can find the code in the corresponding appendix.

Exercise 1

1. What I obtained from simulating a uniform on $[0;1]$ 5 times is : 0.56, 0.8, 0.78, 0.8, 0.1

2. What I obtained as a mean estimate is 0.51.

Exercise 2

1. Let $U \sim \mathcal{U}([0;1])$ and define $V = 2\pi U$.
$\forall t \in [0; 2\pi]$, $\mathbb{P}(V \leq t) = \mathbb{P}(U \leq \frac{t}{2\pi}) = \frac{t}{2\pi}$.
Hence

$$\boxed{f_V(v) = \begin{cases} 0 & \text{if } t \in]-\infty; 0] \cup [2\pi; \infty[\\ \frac{1}{2\pi} & \text{otherwise} \end{cases} \quad \text{and } V \sim \mathcal{U}([0; 2\pi])}$$

2.

$$\boxed{\text{The mean estimate for } V \text{ is 3.15 which is very close to } \pi}$$

.

Exercise 3

1. The exponential distribution $\mathcal{E}(\frac{1}{2})$ has the distribution function $F(t) = 1-e^{-\frac{1}{2}t}$ with $F^{-1}(x) = -2\ln(1-x)$.
Let $E = F^{-1}(U)$ then $\mathbb{P}(E \leq t) = \mathbb{P}(U \leq F(t)) = F(t)$ and $E \sim \mathcal{E}(\frac{1}{2})$.

$$\boxed{\text{Sampling 5 times I obtained : 0.55, 3.03, 1.91, 0.36, 0.55}}$$

2.

$$\boxed{\text{The estimated mean value of } E \text{ is : 2.02 which is very close to 2}}$$

Exercise 4

1. We can see in figure 1 that $N_1 = \sqrt{E}\cos(V)$ follows a gaussian distribution.

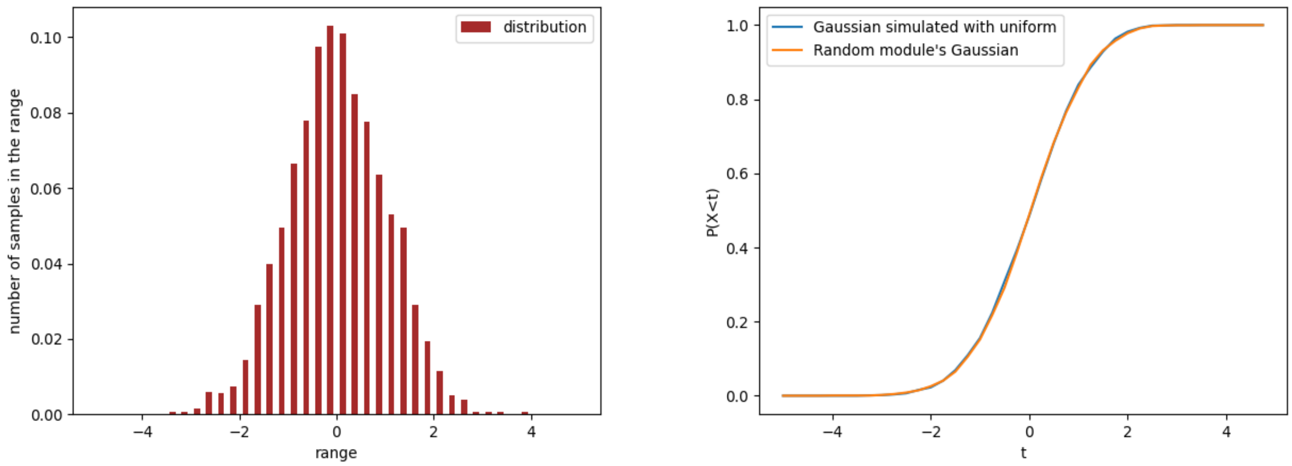


Figure 1: On the left : Density of the simulated gaussian. On the right comparison of the distribution function between the random python's module gaussian and our uniform simulated gaussia.n

A few samples from N_1 are : [-0.05, 0.45, -0.04, 0.32, 0.6] and for N_2 : [0.47, -0.43, 1.77, -0.28, -1.08]

2.

$$\boxed{\text{The two means etimate are 0.2 which is very close to 0}}$$

Exercise 5

1. We need $X = \mu_X + \sigma_X N_1$ and $Y = \mu_Y + \rho\sigma_Y N_1 + \sigma_Y\sqrt{1-\rho^2}N_2$ ie.

$$\boxed{X = 1 + N_1 \text{ and } Y = 1.5 + N_1 + 2\sqrt{0.75}N_2}$$

2. The estimated mean for X is : 1.01 and for Y is : 1.56 both are very close to there respective mean. The estimated correlation is 0.54 which is very close to ρ.

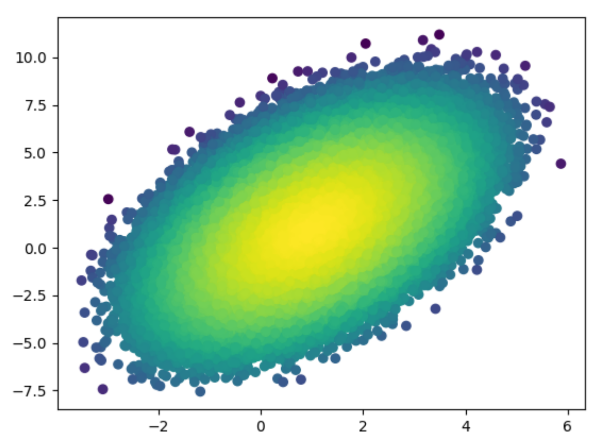


Figure 2: Simulated distribution of correlated gaussians X and Y

A notebook.ipynb

Import cell

We are importing the four modules we will need :
random for generating uniform $\mathcal{U}([0;1])$
math for basic functions such as $\log$ or $\sqrt{}$
numpy for basic functions such as linspace andy for histograms
matplotlib.pyplot for plotting outstanding results

In [1]:

```
import random as rd
import math
import numpy as np
import matplotlib.pyplot as plt
```

Exercise 1 :

In [2]:

```
"""Exercise 1"""
"""Question 1"""

print("Values obtained from sampling the uniform distribution U([0;1]) 5 times :{}".forma
t([round(rd.random(),2) for i in range(5)]))

"""Question 2"""
def mean_estimate(distribution) :
    return round(1/1000*sum([distribution() for i in range(1000)]),2)

print( "The mean values estimated by sampling the uniform distribution U([0;1]) is :{}".f
ormat(mean_estimate(rd.random)))
```

```
Values obtained from sampling the uniform distribution U([0;1]) 5 times :[0.95, 0.01, 0.1
7, 0.42, 0.96]
The mean values estimated by sampling the uniform distribution U([0;1]) is :0.5
```

Exercise 2 :

In [3]:

```
"""Exercise 2"""
"""Question 2"""
def V():
    return 2*math.pi*rd.random()

print("The mean estimated of V = 2piU is :{}. The theoritical one is :{}".format(round(m
ean_estimate(V),2),round(math.pi,2)))
```

```
The mean estimated of V = 2piU is :3.21. The theoritical one is :3.14
```

Exercise 3 :

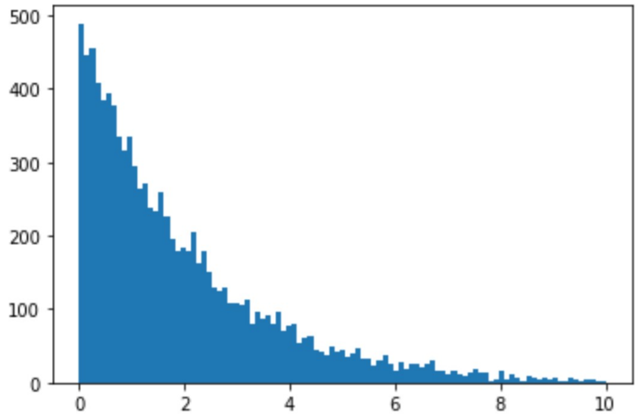
In [4]:

```
"""Exercise 1"""
"""Question 1"""
def E() :
    return -2*math.log(1-rd.random())
print("Values obtained from sampling the exponential distribution E(1/2) 5 times :{}".for
mat([round(E(),2) for i in range(5)]))

plt.close()
```

```
plt.figure()
plt.hist([E() for i in range(10000)],np.linspace(0,10,100))
plt.show()
"""Question 2"""
print("The estimated mean value is : {}".format(mean_estimate(E)))
```

```
Values obtained from sampling the exponential distribution E(1/2) 5 times :[0.29, 0.89, 1
.29, 0.73, 2.72]
```



```
The estimated mean value is : 2.1
```

Exercise 4 :

In [5]:

```
def N1() :
    return math.sqrt(E())*math.cos(V())
def N2() :
    return math.sqrt(E())*math.sin(V())

print("Values obtained from sampling N1 5 times : {}".format([round(N1(),2) for i in ran
ge(5)]))
print("Values obtained from sampling N2 5 times : {}".format([round(N2(),2) for i in ran
ge(5)]))
```

```
Values obtained from sampling N1 5 times : [-0.99, 1.2, -0.41, -0.13, 1.25]
Values obtained from sampling N2 5 times : [-1.71, -1.24, 0.28, -0.36, 1.27]
```

In [6]:

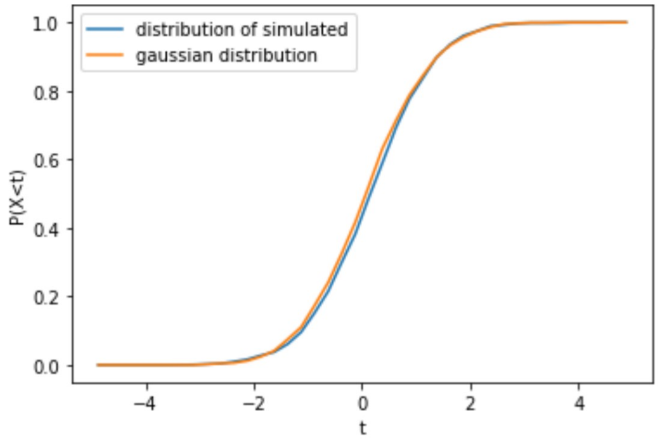
```
L1 = [N1() for i in range(20000)]
L2 = [N2() for i in range(20000)]
L3 = [rd.normalvariate(0,1) for i in range(20000)]


def count(subdivision, samples) :
    return [sum([samples[i]<=subdivision[0] for i in range(2000)])/2000]+[sum([subdivisi
on[j]<samples[i]<=subdivision[j+1] for i in range(2000)])/2000 for j in range(1, len(sub
division)-1)]+[sum([samples[i]>subdivision[-1] for i in range(2000)])/2000]


sub = [-5+0.25*i for i in range(40)]
X = [sub[i]+0.125 for i in range(40)]

F1 = count(sub, L1)
F3 = count(sub, L3)
plt.close()
plt.plot(X, [sum(F1[:i]) for i in range(len(F1))])
plt.plot(X, [sum(F3[:i]) for i in range(len(F3))])
plt.xlabel('t')
plt.ylabel('P(X<t)')
plt.legend(['distribution of simulated','gaussian distribution'])

plt.show()
```



Exercise 5 :

In [7]:

```
"""Exercise 5"""
"""Question 1"""
mu1 = 1
mu2 = 1.5

sigma1 = 1
sigma2 = 2

rho = 0.5

def XY() :
    a=N1()
    b=N2()
    return [mu1+sigma1*a,mu2+sigma2*rho*a+sigma2*np.sqrt(1-rho**2)*b]
def mean() :
    X = [XY()[0] for i in range(1000)]
    Y = [XY()[1] for i in range(1000)]
    return sum(X)/1000, sum(Y)/1000

"""Question 2"""
print("The estimated mean value for X is :{}".format(mean()[0]))
print("The estimated mean value for Y is :{}".format(mean()[1]))
```

```
The estimated mean value for X is :0.9616375828692564
The estimated mean value for Y is :1.4177432011356228
```

In [8]:

```
def correlation() :
    x,y = XY()
    return (x-mu2)*(y-mu2)/sigma1/sigma2

rho_hat = sum([correlation() for i in range(1000)])/1000
print("The estimated correlation is :{}".format(rho_hat))
```

```
The estimated correlation is :0.43293520930236784
```

In [11]:

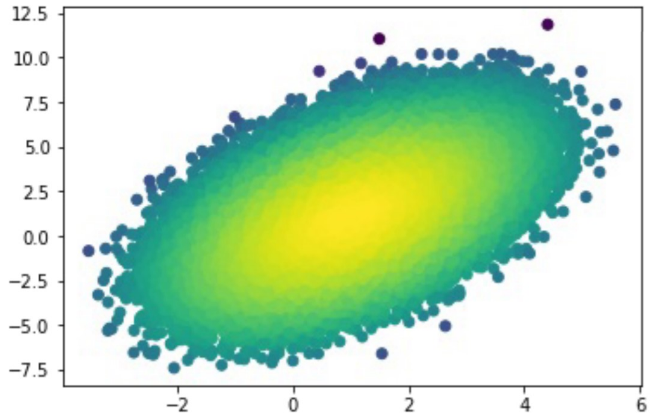
```
L = [XY() for i in range(500000)]
X,Y = [L[i][0] for i in range(len(L))], [L[i][1] for i in range(len(L))]

z = [-(-Y[i]+4*X[i]-3)*(X[i]-1)-(Y[i]-X[i])*(Y[i]-1)for i in range(len(X))]
c= [-z[i]for i in range(len(z))]

plt.scatter(X,Y,c=z)
```

Out[11]:

```
<matplotlib.collections.PathCollection at 0x1a8c0cbbc40>
```



12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−2
0
2
4
6