MA201_ASSIGNMENT 4

Name: Dipean Dasgupta Date:3/11/2022

STD ID: 202151188

Task:

Show graphically for independently generated random variable XX and YY, $f_{xy}(x,y)=f_x(x)f_y(y)$.fxy(x,y)=fx(x)fy(y).

1. Generate X_X and Y_Y from Gaussian distribution with 00 mean and variance 11.
2. find $f_{xy}(x,y)$fxy(x,y) using $f_x(x)$fx(x) and $f_y(y)$fy(y).
   HINT: $[f_{xy}(x,y)]_{no\_bins*no\_bins}=[f_x(x)]_{no\_bins\times 1}\times[f_y(y)]_{1\times no\_bins}$[fxy(x,y)]no_bins∗no_bins =[fx(x)]no_bins×1×[fy(y)]1×no_bins
3. Plot joint PDF.

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

For Histogram:

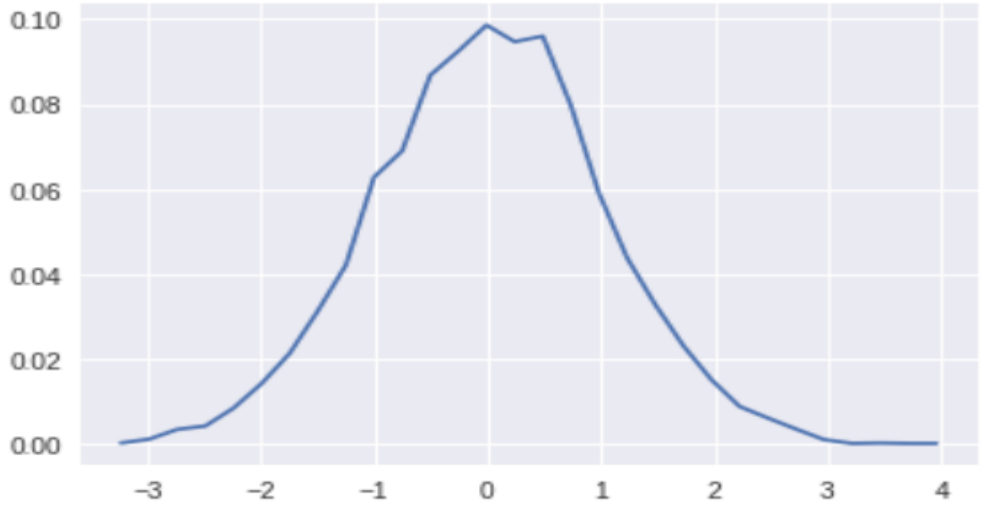
```
def hist(x,no_bins):
    no_samples = np.size(x)
    x_max = max(x)
    x_min = min(x)
    bins = np.linspace(x_min,x_max,no_bins)
    bin_width = abs(bins[1] - bins[0])
    freq = np.zeros(shape=(no_bins))
    for i in range(no_samples):
        for j in range(no_bins):
           if(x[i] <= bins[j]+0.5*bin_width and x[i] > bins[j]-bin_width):
                freq[j] += 1
                break
    freq =freq/(no_samples)
    return bins,freq
M = 10000 # Number of samples
no_bins = 30 #Number of bins for histogram
```

Generating X and Y from Gaussian distribution with 0 mean and variance 1:

Generating X:

```
mu = 0
sigma = 1
X = np.random.normal(mu,sigma,M)
[x,fx]=hist(X,no_bins)
plt.figure()
plt.plot(x,fx)
```

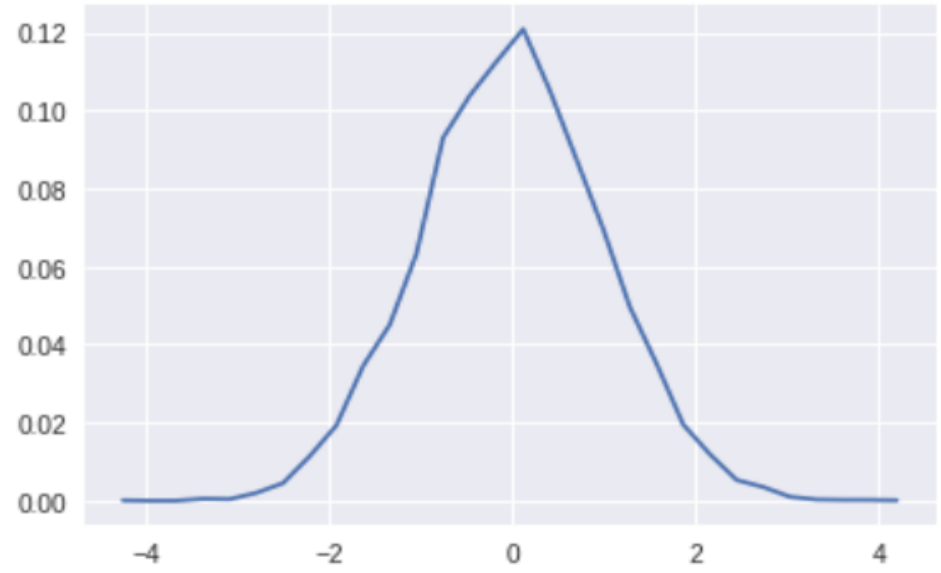
```
[<matplotlib.lines.Line2D at 0x7fc9bcfa2518>]
```



Generating Y:

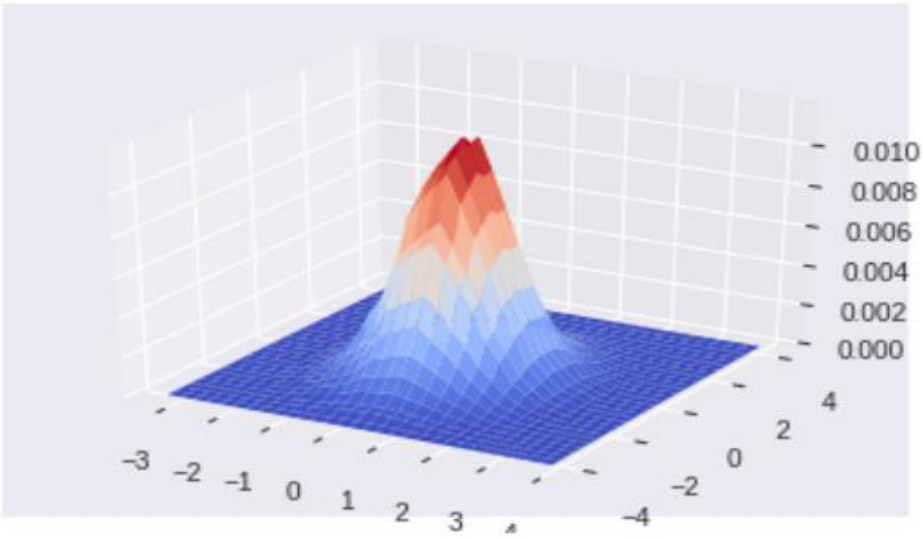
```
Y = np.random.normal(mu,sigma,M)
[y,fy]=hist(Y,no_bins)
plt.figure()
plt.plot(y,fy)
```

```
[<matplotlib.lines.Line2D at 0x7fc9baeb78d0>]
```



Finding $f_{xy}(x,y)$ using $f_x(x)$ and $f_y(y)$

```
fig = plt.figure()
ax = fig.gca(projection='3d')
fx = np.reshape(fx,[np.size(fx),1])
fy = np.reshape(fy,[1,np.size(fy)])
fxy = np.matmul(fx,fy)
x1,y1 = np.meshgrid(x,y)
surf = ax.plot_surface(x1,y1, fxy, cmap=cm.coolwarm)
```



Generating joint histogram of two variable:

```
def hist2(x,y,no_bins):
    no_samples = np.size(x)
    x_max = max(x)
    y_max = max(y)
    x_min = min(x)
    y_min = min(y)
    binsx = np.linspace(x_min,x_max,no_bins)
    binsy = np.linspace(y_min,y_max,no_bins)
    bin_widthx = abs(binsx[1] - binsx[0])
    bin_widthy = abs(binsy[1] - binsy[0])
    freq = np.zeros(shape=(no_bins,no_bins))
    for i in range(no_samples):
      for j in range(no_bins):
        for k in range(no_bins):
          if(x[i] <= binsx[j]+0.5*bin_widthx and x[i] > binsx[j]-
bin_widthx and y[i] <= binsy[k]+0.5*bin_widthy and y[i] > binsy[k]-
bin_widthy):
            freq[j,k] += 1
            break
    freq =freq/(no_samples)
```

```
    return binsx,binsy,freq
fig = plt.figure()
ax = fig.gca(projection='3d')
[x,y,fxy] = hist2(X,Y,no_bins)
x1,y1 = np.meshgrid(x,y)
surf = ax.plot_surface(x1,y1,fxy, cmap=cm.coolwarm)
```