BM20BTECH11001-Lab13

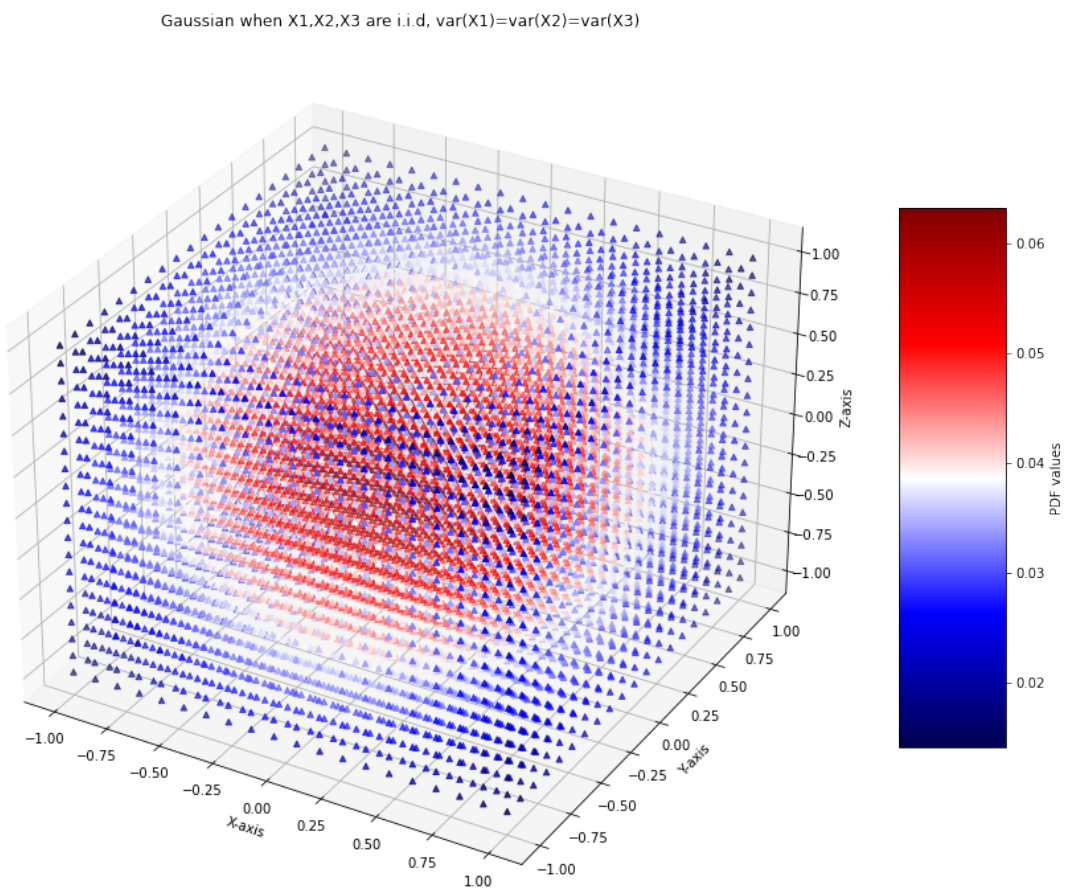
November 23, 2021

[1]:
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
from pylab import *
import matplotlib.colors as colors
```

0.1 When X1,X2,X3 are independent identical distributions

[2]:
```
#Covariance matrix
matrix = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/((2*np.pi)**(3/2))
```

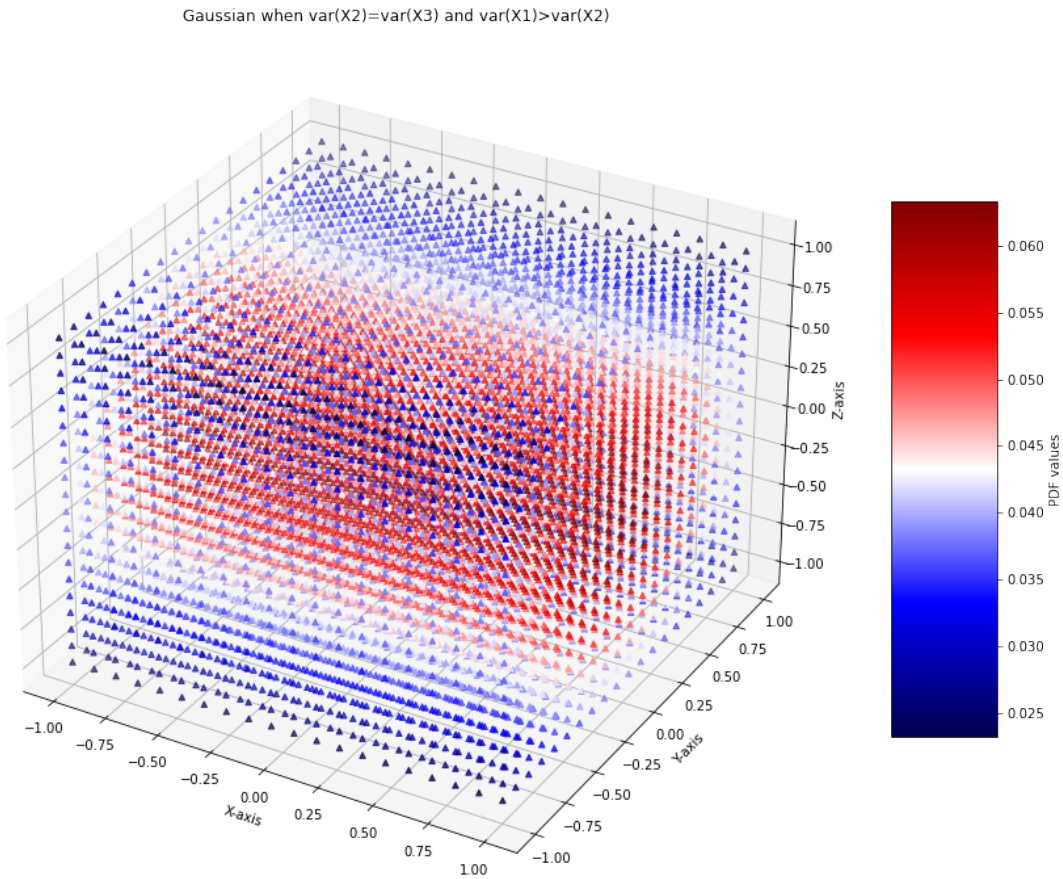
```
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when X1,X2,X3 are i.i.d, var(X1)=var(X2)=var(X3)');
```



- Here, when all the autovariances of x1,x2 and x3 are same, we notice a spherical cloud distribution near the mean.

0.2 When var(X2)=var(X3) and var(X1)>var(X2)

[3]:
```
#Covariance matrix
matrix = [[100, 0, 0], [0, 1, 0], [0, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/((2*np.pi)**(3/2))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when var(X2)=var(X3) and var(X1)>var(X2)');
```

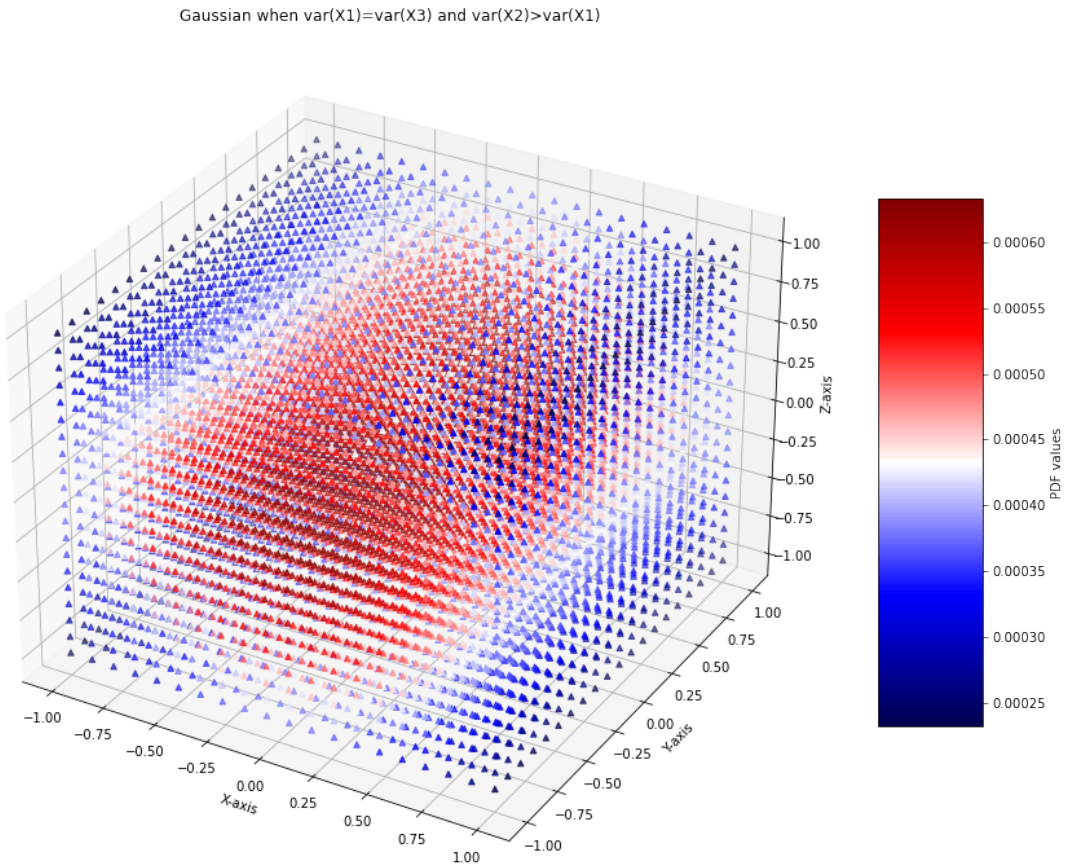


- If autovariance of x1 is greater than x2 with var(x2) = var(x3), we see that the cloud is elongated along the direction of x1.

0.3 When var(X1)=var(X3) and var(X2)>var(X1)

[4]:
```
#Covariance matrix
matrix = [[1, 0, 0], [0, 100, 0], [0, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
```

```
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when var(X1)=var(X3) and var(X2)>var(X1)');
```

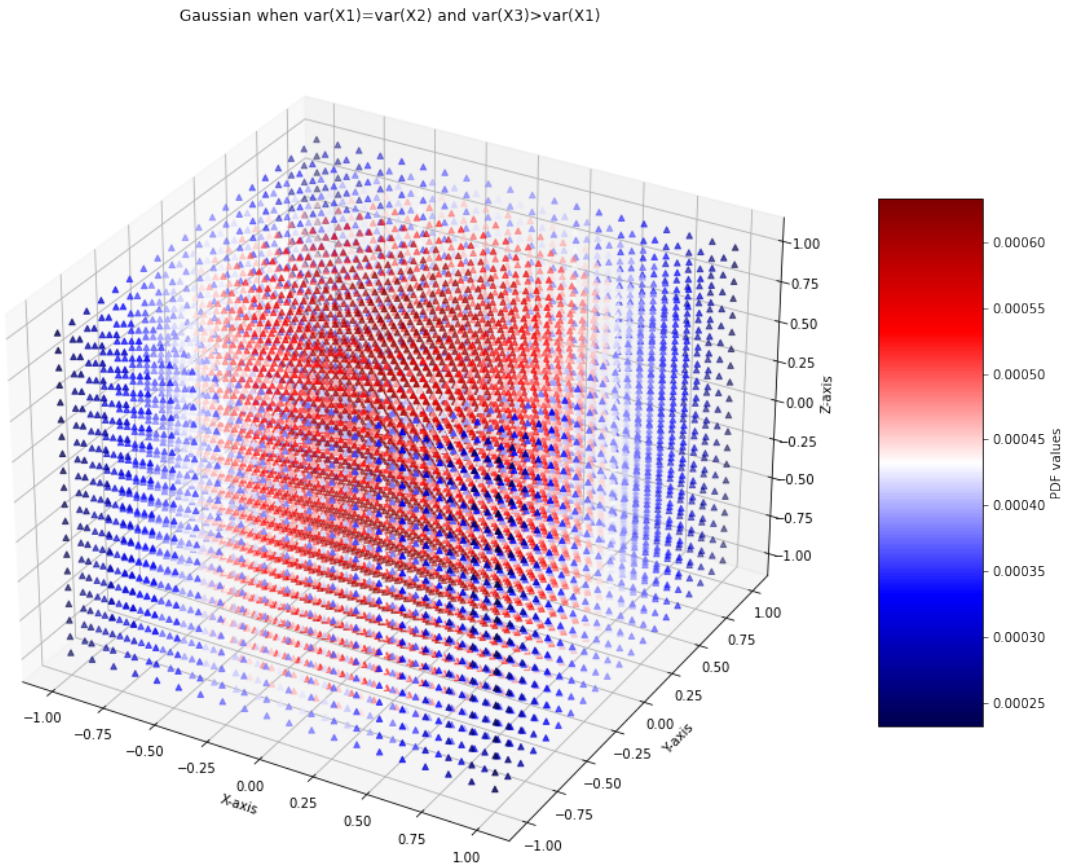


- If autovariance of x2 is greater than x1 with var(x1) = var(x3), we see that the cloud is elongated along the direction of x2.

0.4 When var(X1) = var(X2) , var(X3)>var(X1)

[5]:
```
#Covariance matrix
matrix = [[1, 0, 0], [0, 1, 0], [0, 0, 100]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
```

```
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when var(X1)=var(X2) and var(X3)>var(X1)');
```

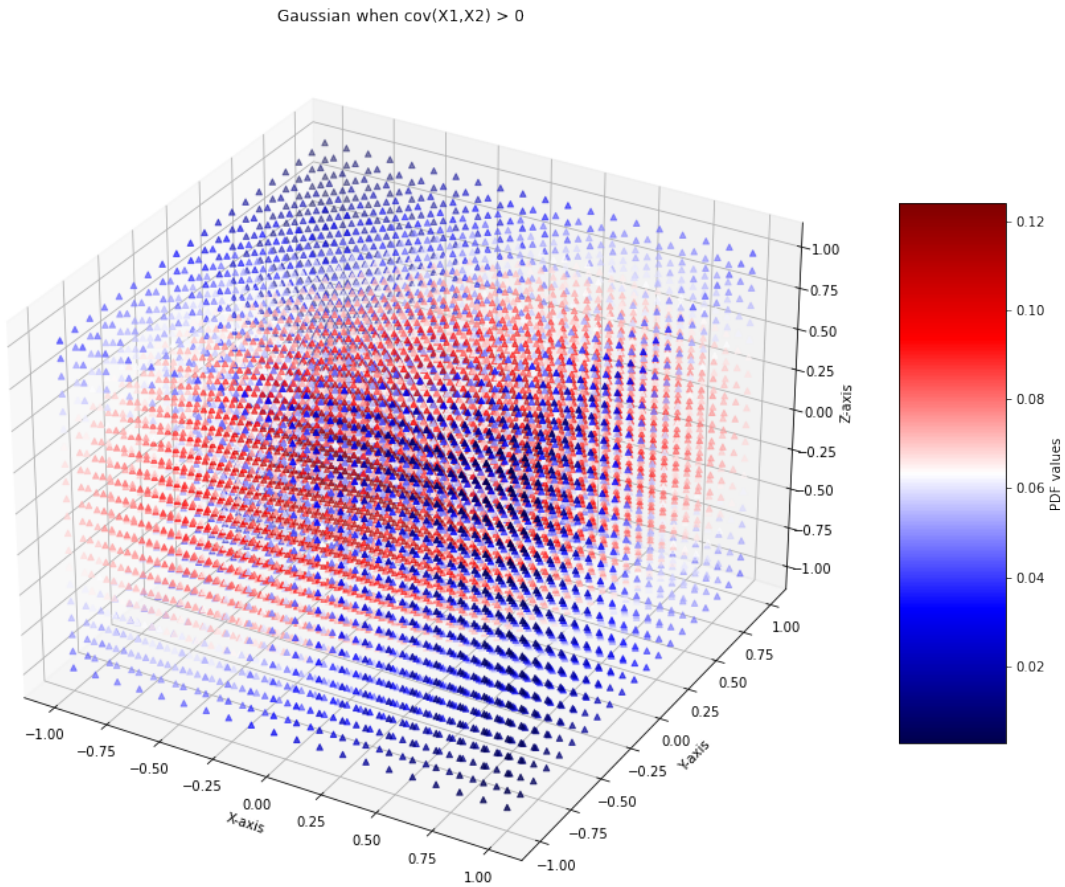


- If autovariance of x3 is greater than x2 with var(x1) = var(x2), we see that the cloud is elongated along the direction of x3.

0.5 When cov(X1,X2) != 0

[6]:
```
#Covariance matrix
matrix = [[1, 0.7, 0], [0.7, 1, 0], [0, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
```

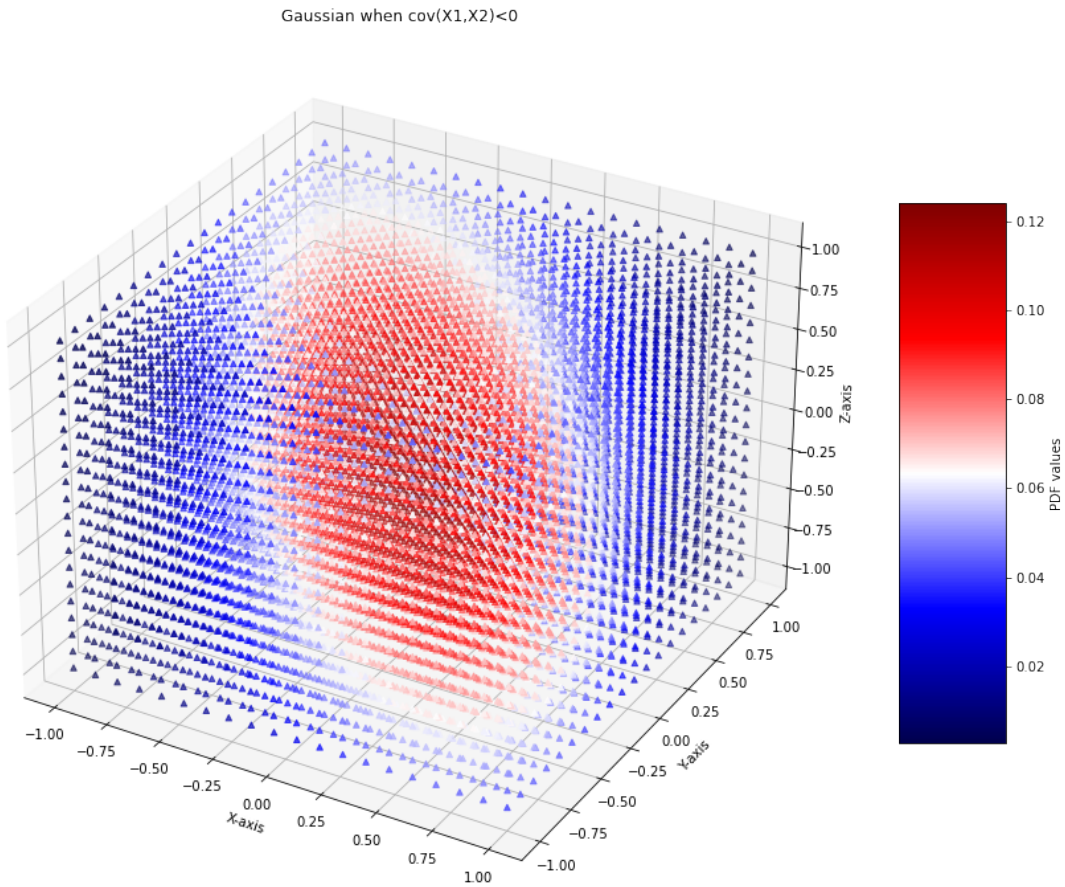
```
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X1,X2) > 0');
```



- If cov(x2,x3)=0 and cov(x1,x3) = 0 and cov(x1,x2)>0, we find the cloud to be elongated along y=x direction.

[7]:
```
#Covariance matrix
matrix = [[1, -0.7, 0], [-0.7, 1, 0], [0, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
```

```
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X1,X2)<0');
```

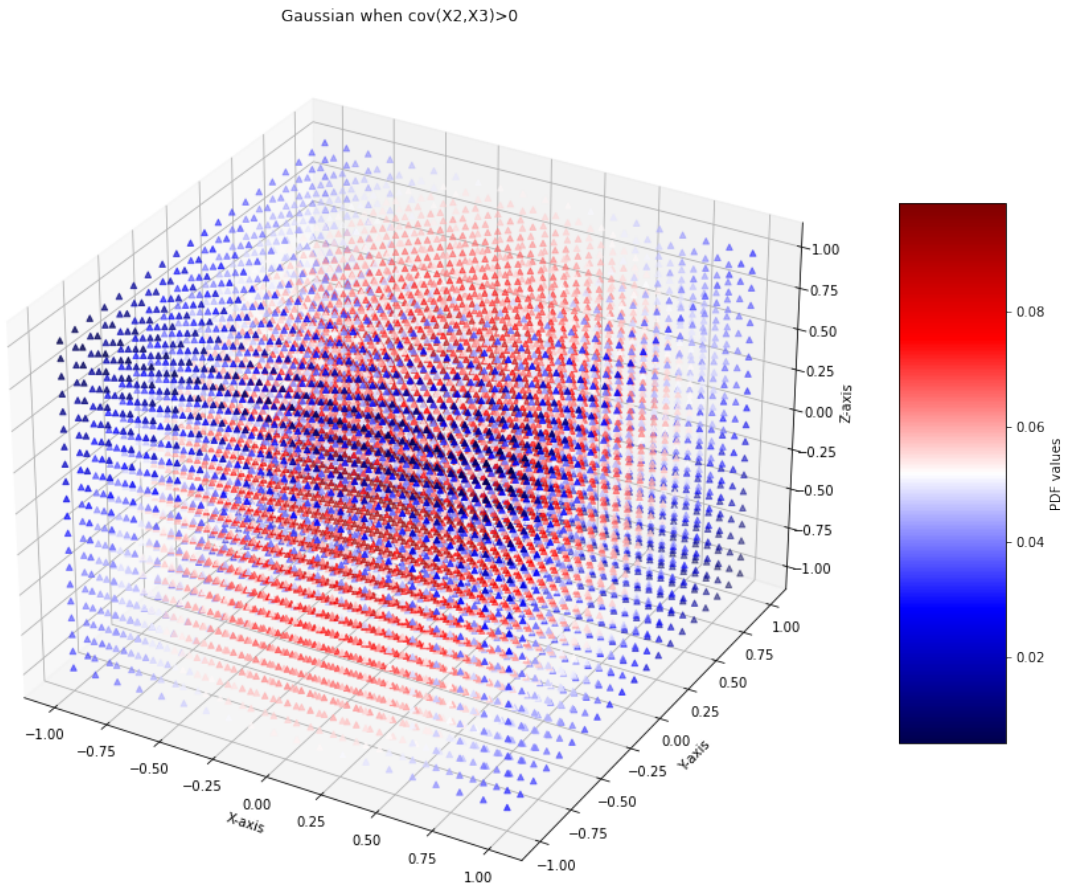


- If cov(x2,x3)=0 and cov(x1,x3) = 0 and cov(x1,x2)<0, we find the cloud to be elongated along y=-x direction.

0.6 When cov(X2,X3)!=0

[8]:
```
#Covariance matrix
matrix = [[1, 0, 0], [0, 1, 0.60], [0, 0.60, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []
```

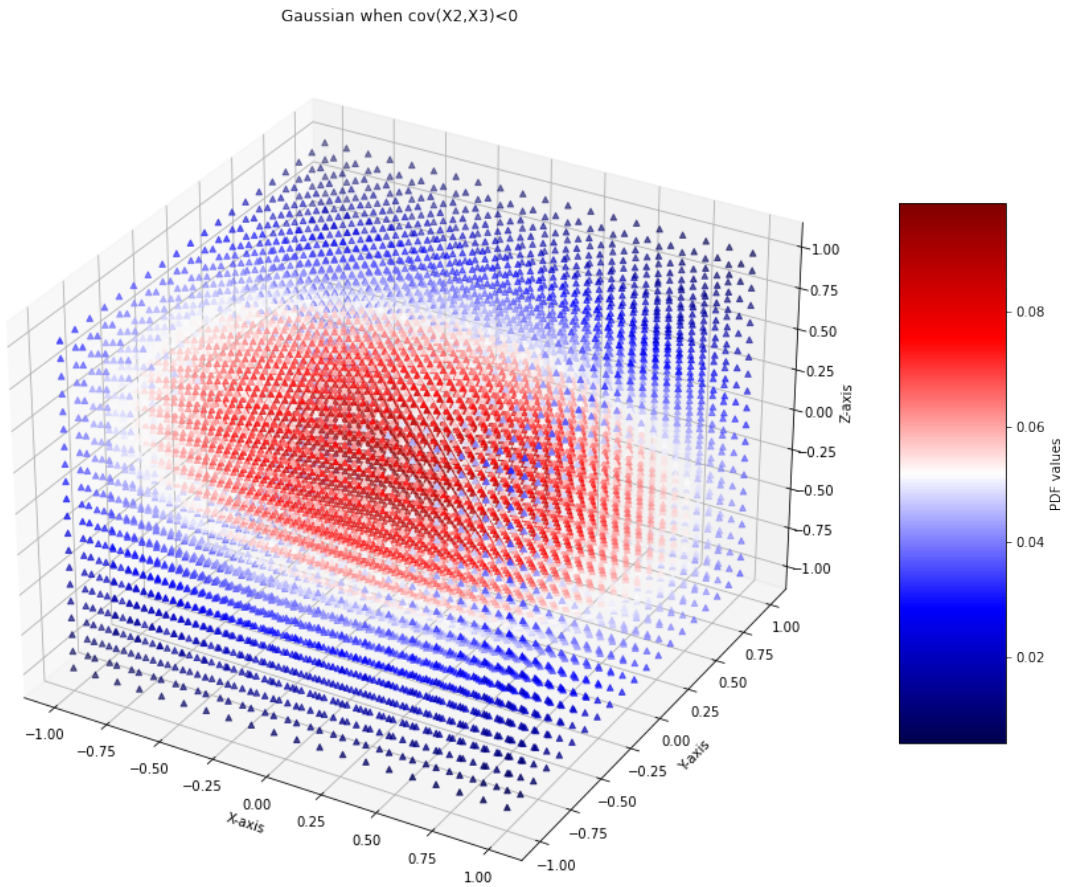
```
x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X2,X3)>0');
```



- If cov(x1,x2)=0 and cov(x1,x3) = 0 and cov(x2,x3)>0, we find the cloud to be elongated along y=z direction.

[9]:
```
#Covariance matrix
matrix = [[1, 0, 0], [0, 1, -0.6], [0, -0.6, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
```

```
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X2,X3)<0');
```

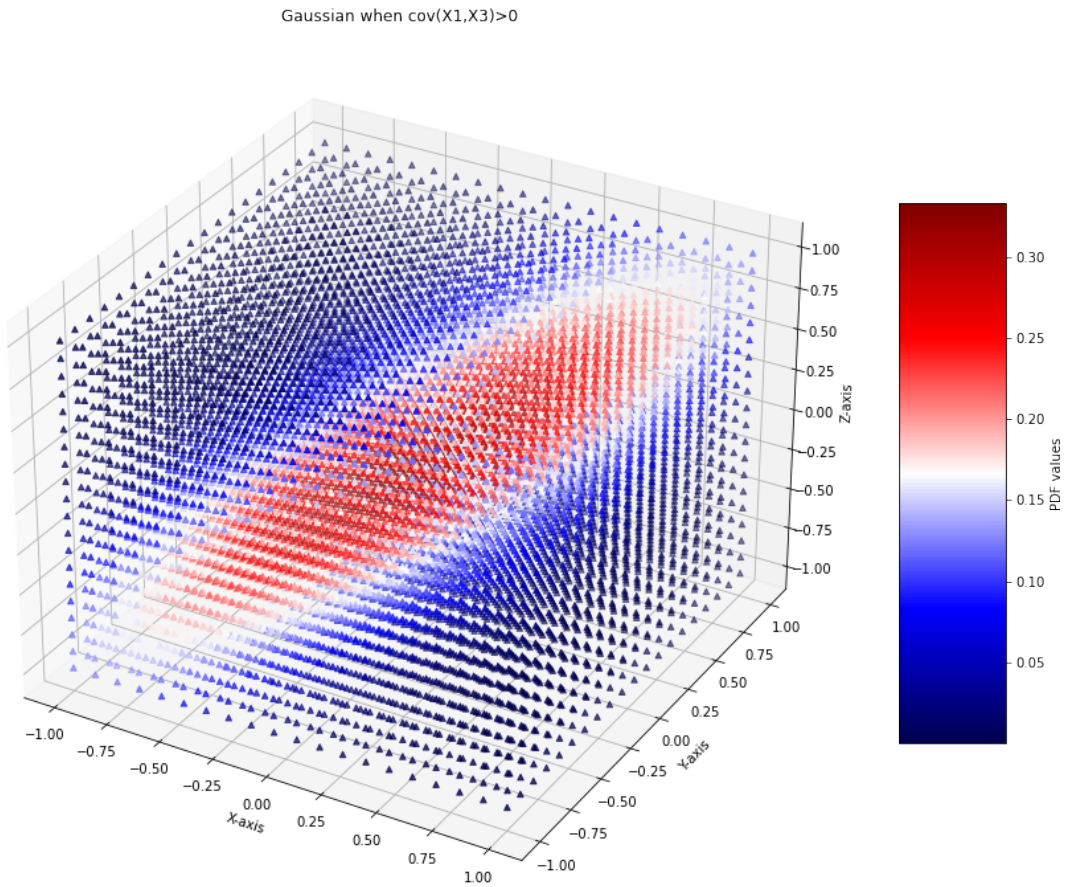


- If cov(x1,x2)=0 and cov(x1,x3) = 0 and cov(x2,x3)<0, we find the cloud to be elongated along y=-z direction.

0.7 When cov(X1,X3)!=0

[10]:
```
#Covariance matrix
matrix = [[1, 0, 0.9], [0, 1, 0], [0.9, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []
```

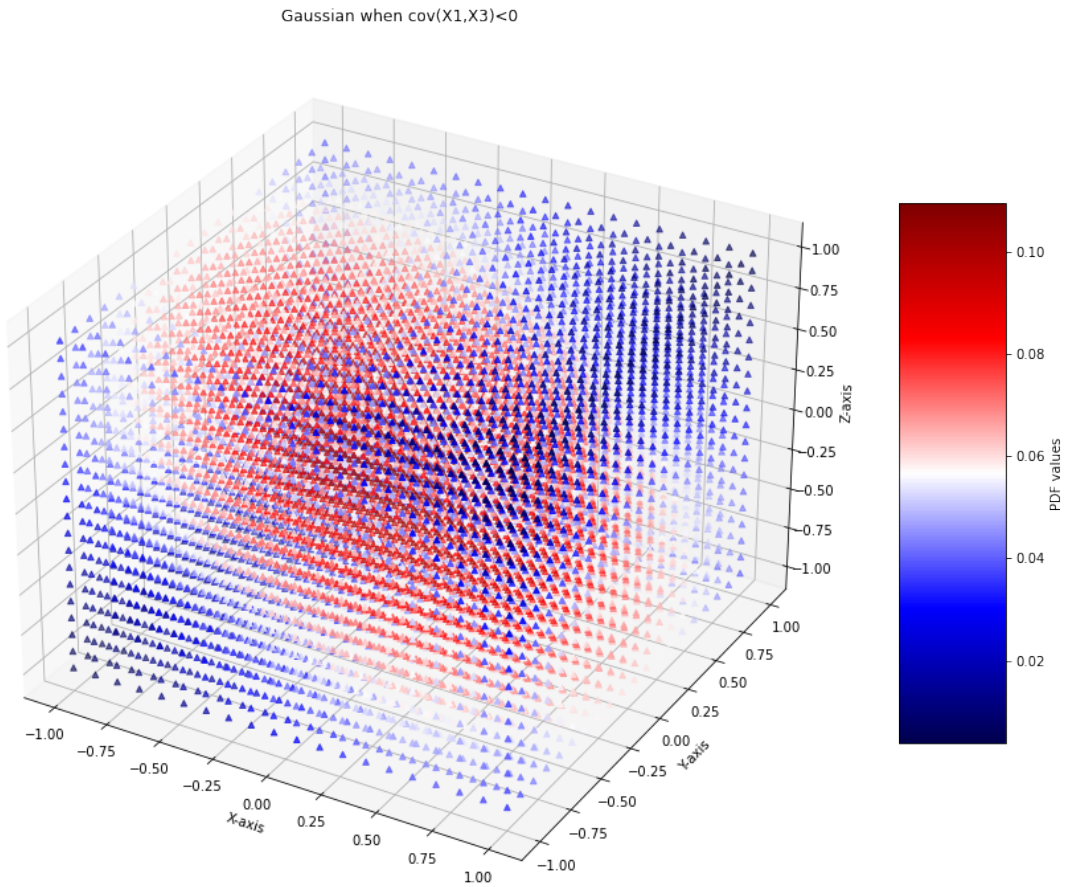
```
x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X1,X3)>0');
```



- If cov(x1,x2)=0 and cov(x2,x3) = 0 and cov(x1,x3)>0, we find the cloud to be elongated along x=z direction.

[11]:
```
#Covariance matrix
matrix = [[1, 0, -0.65], [0, 1, 0], [-0.65, 0, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []


x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
```

```
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X1,X3)<0');
```

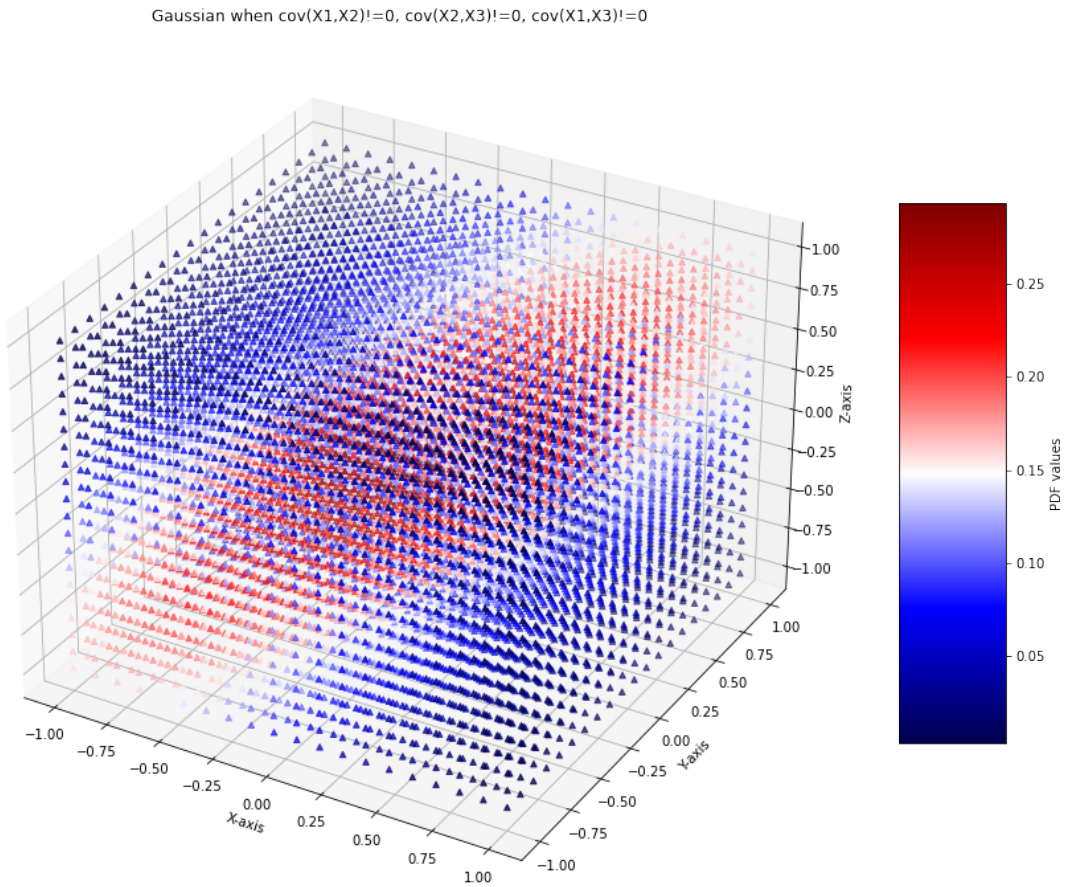


- If cov(x1,x2)=0 and cov(x2,x3) = 0 and cov(x1,x3)<0, we find the cloud to be elongated along x=-z direction.

0.8 When cov(X1,X2)!=0, cov(X2,X3)!=0 and cov(X1,X3)!=0

[12]:
```
#Covariance matrix
matrix = [[1, 0.7, 0.7], [0.7, 1, 0.7], [0.7, 0.7, 1]]

mean = 0

#Y is column vector
Y = lambda x,y,z : np.transpose([[x-mean, y-mean, z-mean]])

#Inverse of covariance matrix
Inv = np.linalg.inv(matrix)
f_z = []
```

```
x = np.linspace(-1, 1, 20)
y = np.linspace(-1, 1, 20)
z = np.linspace(-1, 1, 20)
x_1 = []
y_1 = []
z_1 = []

for d in x:
    for e in y:
        for f in z:
            x_1.append(d)
            y_1.append(e)
            z_1.append(f)
            t = np.dot(np.transpose(Y(d,e,f)), Inv)
            a = np.dot(t, Y(d,e,f))
            gauss = ((np.e)**(-a[0][0]/2))/(((2*np.pi)**(3/2))*(np.linalg.
 ↪det(np.array(matrix))))
            f_z.append(gauss)

my_cmap = plt.get_cmap('seismic')

fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

# creating the heatmap
img = ax.scatter3D(x_1, y_1, z_1,c=f_z, cmap=my_cmap, marker='^')
fig.colorbar(img, ax = ax, shrink = 0.5, aspect = 5, label="PDF values")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Gaussian when cov(X1,X2)!=0, cov(X2,X3)!=0, cov(X1,X3)!=0');
```



- If cov(x2,x3)!=0 and cov(x1,x3) = !0 and cov(x1,x2)!=0, we find the cloud to be elongated along x+y+z=0 direction.