

# Actividad 1 - La Normal Variada

- Frida Cano Falcón - A01752953

## Librerías

```
In [ ]: import numpy as np
from scipy.stats import multivariate_normal
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

## Ejercicio 1

Hallar el procedimiento para el cálculo de probabilidad de que:  $P(X_1 \leq 2, X_2 \leq 3)$  con  $X_1, X_2$  se distribuyen Normal con  $\mu = (\mu_1 = 2.5, \mu_2 = 4)$  y

$\Sigma =$

$$\begin{bmatrix} 1.2 & 0 \\ 0 & 2.3 \end{bmatrix}$$

```
In [ ]: # Definir los parámetros mu y sigma
mu = np.array([2.5, 4])
sigma = np.array([[1.2, 0], [0, 2.3]])

# Crear la distribución normal bivariada
bivariate_normal = multivariate_normal(mean=mu, cov=sigma)

# Definir los límites para X1 y X2
x1_lim = 2
x2_lim = 3

# Calcular la probabilidad P(X1 <= 2, X2 <= 3)
prob = bivariate_normal.cdf([x1_lim, x2_lim])

print("Probabilidad: ", prob)
```

Probabilidad: 0.08257333341548989

## Ejercicio 2

Graficar la anterior distribución bivariada del problema 1.

```
In [ ]: # ----- Distribución del problema 1 -----
xmin = mu[0] - 5
xmax = mu[0] + 5
```

```

ymin = miu[1] - 5
ymax = miu[1] + 5

x = np.arange(xmin, xmax, 0.1)
y = np.arange(ymin, ymax, 0.1)

X, Y = np.meshgrid(x, y)
grid = np.dstack((X,Y))
pdf_values = bivariate_normal.pdf(grid)

# ----- Distribución del problema 1, std =3 -----
std = 3
x_std3 = np.sqrt(sigma[0,0]) * std
y_std3 = np.sqrt(sigma[1,1]) * std

xmin_std3 = miu[0] - x_std3
xmax_std3 = miu[0] + x_std3
ymin_std3 = miu[1] - y_std3
ymax_std3 = miu[1] + y_std3

x_3 = np.linspace(xmin_std3, xmax_std3, 100)
y_3 = np.linspace(ymin_std3, ymax_std3, 100)
X3, Y3 = np.meshgrid(x_3, y_3)
grid3 = np.dstack((X3, Y3))
pdf_values3 = bivariate_normal.pdf(grid3)

# ----- Gráficas -----
fig, axs = plt.subplots(1, 2, figsize=(12, 5), subplot_kw={'projection': '3d'})

axs[0].plot_surface(X, Y, pdf_values, cmap='viridis')
axs[0].set_xlabel('X')
axs[0].set_ylabel('Y')
axs[0].set_zlabel('Densidad de Probabilidad')
axs[0].set_title('Distribución Bivariada')

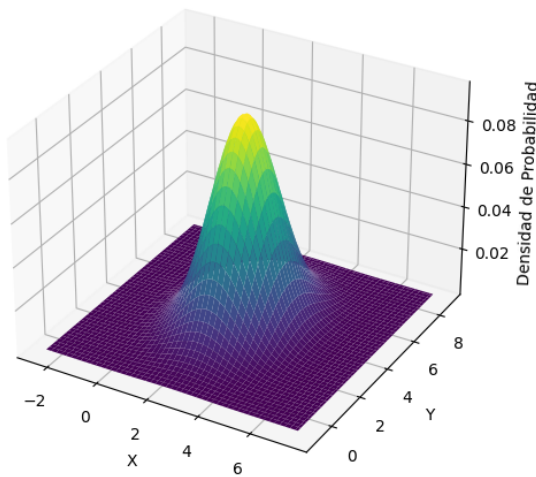
axs[1].plot_surface(X3, Y3, pdf_values3, cmap='viridis')
axs[1].set_xlabel('X')
axs[1].set_ylabel('Y')
axs[1].set_zlabel('Densidad de Probabilidad')
axs[1].set_title('Distribución Bivariada (3 Desviaciones Estándar)')

plt.tight_layout()

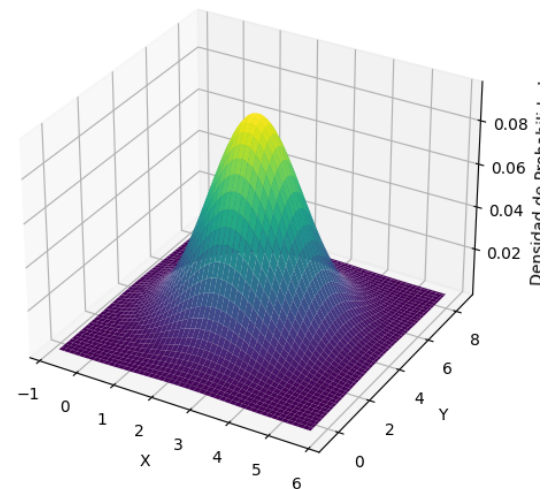
plt.show()

```

Distribución Bivariada



Distribución Bivariada (3 Desviaciones Estándar)



## Ejercicio 3

Grafique los contornos de la anterior distribución normal bivariada correspondiente a las alturas de 0.01, 0.03, 0.05, 0.07, 0.09.

```
In [ ]: # ----- Gráficas de contrornos -----
# Alturas especificadas
levels = [0.01, 0.03, 0.05, 0.07, 0.09]

fig, axs = plt.subplots(1, 2, figsize=(12, 5), subplot_kw={'projection': '3d'})

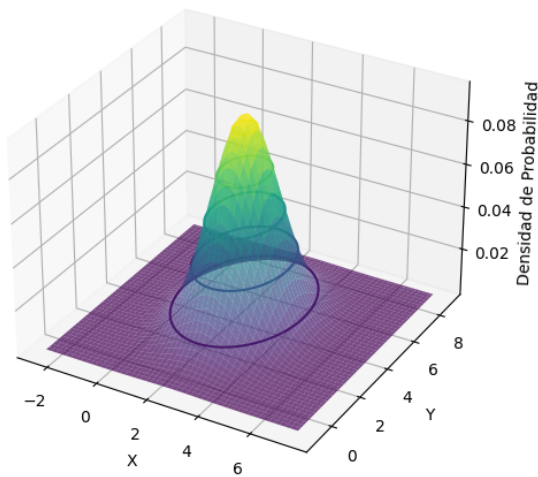
# ----- Distribución del problema 1 -----
axs[0].plot_surface(X, Y, pdf_values, cmap='viridis', alpha= 0.7)
axs[0].contour(X, Y, pdf_values, levels = levels)
axs[0].set_xlabel('X')
axs[0].set_ylabel('Y')
axs[0].set_zlabel('Densidad de Probabilidad')
axs[0].set_title('Distribución Bivariada')

# ----- Distribución del problema 1, std =3 -----
axs[1].plot_surface(X3, Y3, pdf_values3, cmap='viridis', alpha= 0.7)
axs[1].contour(X3, Y3, pdf_values3, levels = levels)
axs[1].set_xlabel('X')
axs[1].set_ylabel('Y')
axs[1].set_zlabel('Densidad de Probabilidad')
axs[1].set_title('Distribución Bivariada (3 Desviaciones Estándar)')

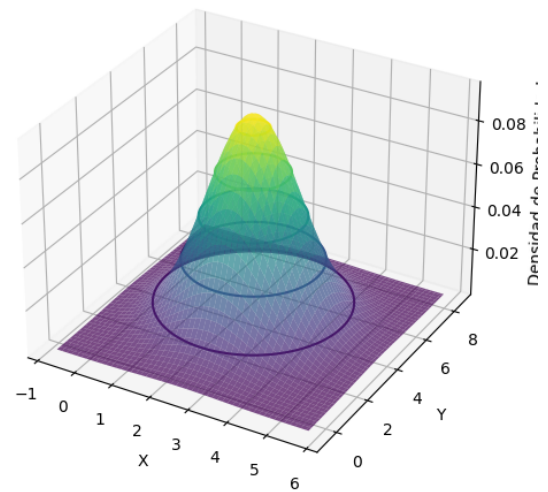
plt.tight_layout()

plt.show()
```

Distribución Bivariada



Distribución Bivariada (3 Desviaciones Estándar)



## Comentarios

La relación entre los ejercicios es que para todos se utiliza una misma distribución bivariada de raíz, solamente ajustando diferentes desviaciones estándar. Las gráficas nos ayudan a visualizar dichas distribuciones, y a visualizar cómo cambian los valores de X y Y para ciertas alturas de acuerdo con las desviaciones estándar.