

# Introducción a la representación gráfica. Tarea 02

Alberto Simón

18/04/2025

## Pregunta 1

A partir del siguiente código, intenta obtener el gráfico del dibujo.

```
import numpy as np
import matplotlib.pyplot as plt

n = 256
X = np.linspace(-np.pi, np.pi, n, endpoint=True)
Y = np.sin(2 * X)

plt.plot(X, Y + 1, color="blue", alpha=1.00)
plt.plot(X, Y - 1, color="blue", alpha=1.00)
plt.fill_between(X, Y + 1, 1, color="blue", alpha=0.1)
plt.fill_between(X, Y - 1, -1, (Y > 0), color="blue", alpha=0.1)
plt.fill_between(X, Y - 1, -1, (Y < 0), color="red", alpha=0.1)
plt.show()
```

## Pregunta 2

A partir del siguiente código, intenta obtener el gráfico del dibujo.

```
import numpy as np
import matplotlib.pyplot as plt

n = 1024
X = np.random.normal(0, 1, n)
Y = np.random.normal(0, 1, n)

T = np.arctan2(Y, X) + np.pi

color_map = plt.cm.Spectral_r
plt.scatter(X, Y, c=T, alpha=1/3, cmap = color_map)
plt.show()
```

## Pregunta 3

A partir del siguiente código, intenta obtener el gráfico del dibujo.

```
import numpy as np
import matplotlib.pyplot as plt

n = 12
X = np.arange(n)
```

```

Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)

plt.bar(X, +Y1, facecolor="#9999ff", edgecolor="white")
plt.bar(X, -Y2, facecolor="#ff9999", edgecolor="white")

for x, y in zip(X, Y1):
    plt.text(x, y + 0.02, "%.2f" % y, ha="center", va="bottom")

for x, y in zip(X, -Y2):
    plt.text(x, y - 0.02, "%.2f" % y, ha="center", va="top")

plt.ylim(-1.25, +1.25)
plt.show()

```

## Pregunta 4

A partir del siguiente código, intenta obtener el gráfico del dibujo.

```

import numpy as np
import matplotlib.pyplot as plt

Z = np.random.uniform(0, 1, 20)
Z = Z / np.sum(Z)
Z_copy = Z.copy()
Z_copy.sort()
Z_scaled = Z_copy/Z_copy.max()

colour_array = np.array([Z_scaled, Z_scaled,Z_scaled])
colour_array2 = np.transpose(colour_array)

explode_array = [0.1] * len(Z)
explode_array[len(explode_array)-1] = explode_array[0] * 2

plt.pie(Z, colors = colour_array2, explode = explode_array, wedgeprops={'edgecolor': 'black'})
plt.show()

```