

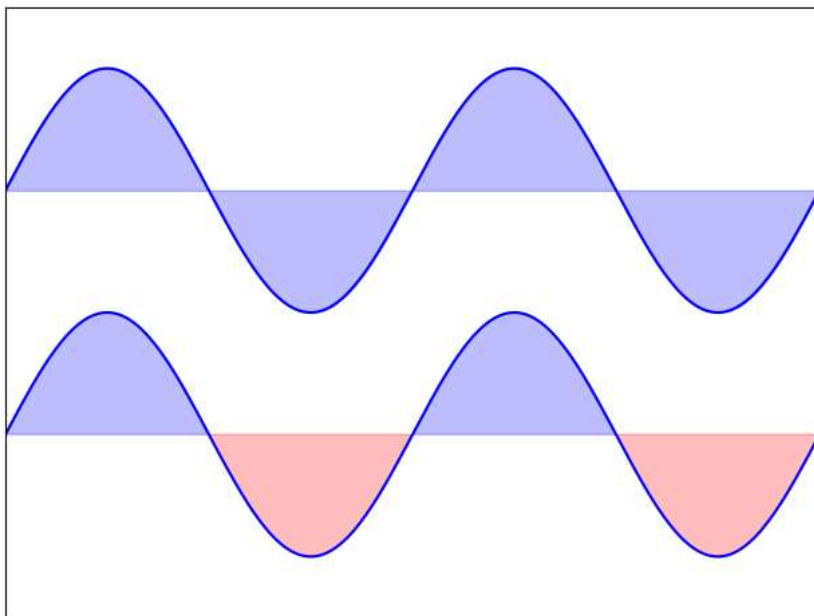
## Gráficos con Python

### Pregunta 1

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



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 n = 256
5 X = np.linspace(-np.pi, np.pi, n, endpoint=True)
6 Y = np.sin(2 * X)
7 plt.plot(X, Y+1, color='blue', alpha=1)
8 plt.plot(X, Y-1, color='blue', alpha=1)
9
10 plt.axes([0.025, 0.025, 0.95, 0.95])
11 plt.plot(X, Y+1, color="blue", alpha=1.00)
12 plt.fill_between(X, 1, Y+1, color="blue", alpha=.25)
13 plt.plot(X, Y-1, color="blue", alpha=1.00)
14 plt.fill_between(X, -1, Y-1, (Y-1)>-1, color="blue", alpha=.25)
15 plt.fill_between(X, -1, Y-1, (Y-1)<-1, color="red", alpha=.25)
16 plt.xlim(-np.pi, np.pi)
17 plt.xticks(())
18 plt.ylim(-2.5, 2.5)
19 plt.yticks(())
20 plt.show()
```



### Pregunta 2

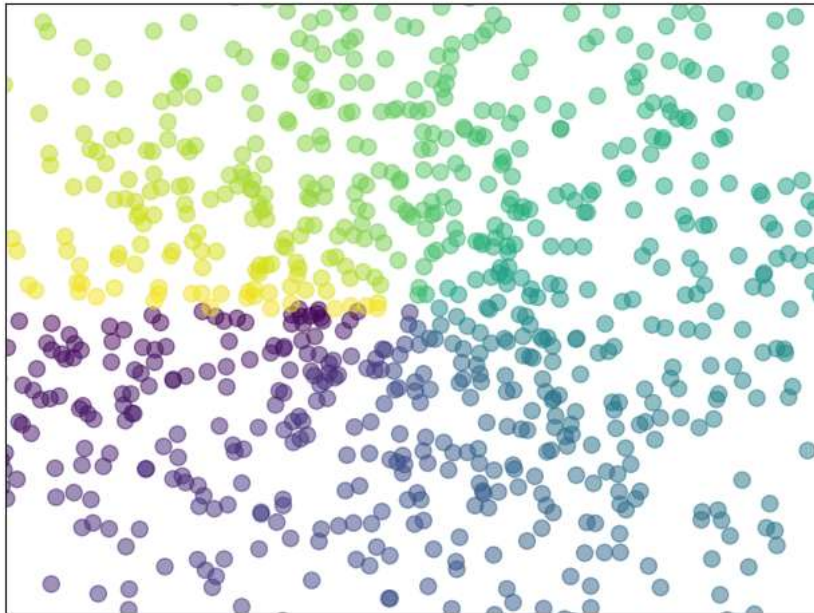
A partir del siguiente código, intenta obtener el gráfico del dibujo



```

1 n=1024
2 X=np.random.normal(0, 1, n)
3 Y=np.random.normal(0, 1, n)
4 T=np.arctan2(Y, X)
5 plt.axes([0.025, 0.025, 0.95, 0.95])
6 plt.scatter(X, Y, s=75, c=T, alpha=.5)
7 plt.xlim(-1.5, 1.5)
8 plt.xticks(())
9 plt.ylim(-1.5, 1.5)
10 plt.yticks(())
11 plt.show()

```



### ▼ Pregunta 3

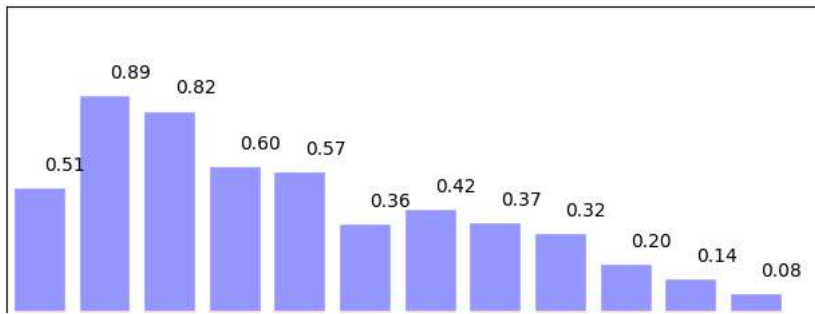
A partir del siguiente código, intenta obtener el gráfico del dibujo



```

1 n=12
2 X=np.arange(n)
3 Y1=(1-X/float(n))*np.random.uniform(0.5, 1.0, n)
4 Y2=(1-X/float(n))*np.random.uniform(0.5, 1.0, n)
5 plt.axes([0.025, 0.025, 0.95, 0.95])
6 plt.bar(X,+Y1, facecolor="#9999ff", edgecolor="white")
7 plt.bar(X,-Y2, facecolor="#ff9999", edgecolor="white")
8 for x, y in zip(X, Y1):
9     plt.text(x+0.4, y+0.05, "%.2f"%y, ha="center", va="bottom")
10 for x, y in zip(X, Y2):
11     plt.text(x+0.4, -y-0.05, "%.2f"%y, ha="center", va="top")
12 plt.xlim(-.5, n)
13 plt.xticks(())
14 plt.ylim(-1.25, 1.25)
15 plt.yticks(())
16 plt.show()

```



#### ▼ Pregunta 4

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

image.png

```

1 n=20
2 Z=np.ones(n)
3 Z[-1]*=2
4 plt.axes([0.025, 0.025, 0.95, 0.95])
5 plt.pie(Z, explode=Z*.05, colors=["%f"%(i/float(n))for i in range(n)])
6 plt.axis("equal")
7 plt.xticks()
8 plt.yticks()
9 plt.show()

```

