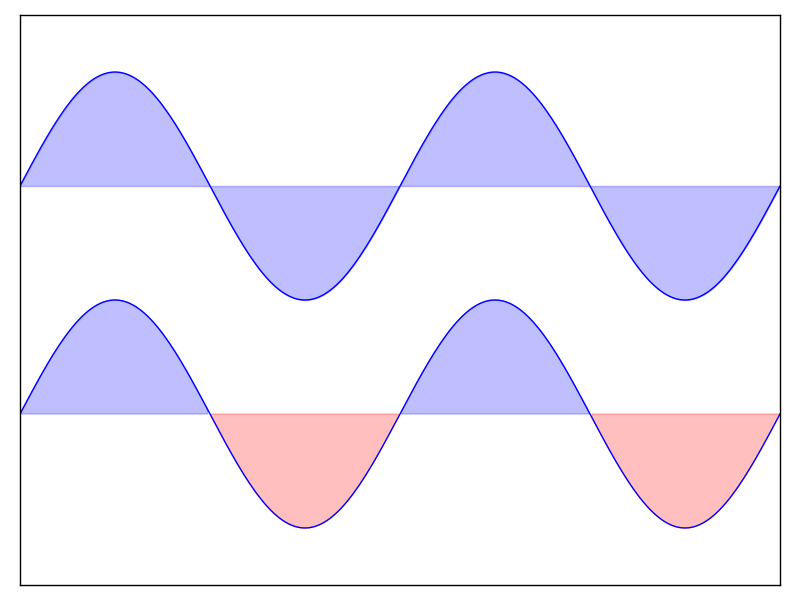
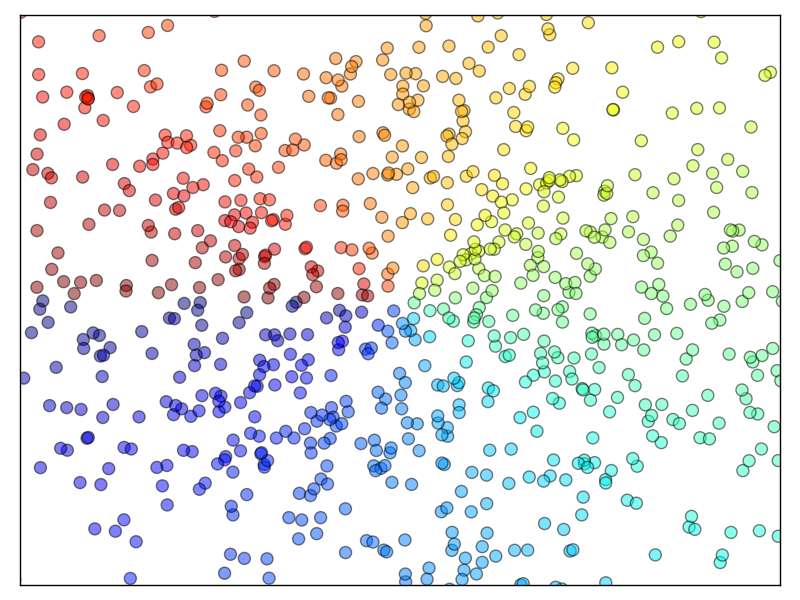
Lee atentamente cada ejercicio e investiga la documentación para sacar el ejercicio adelante. Tienes los gráficos de cada ejercicio a continuación.

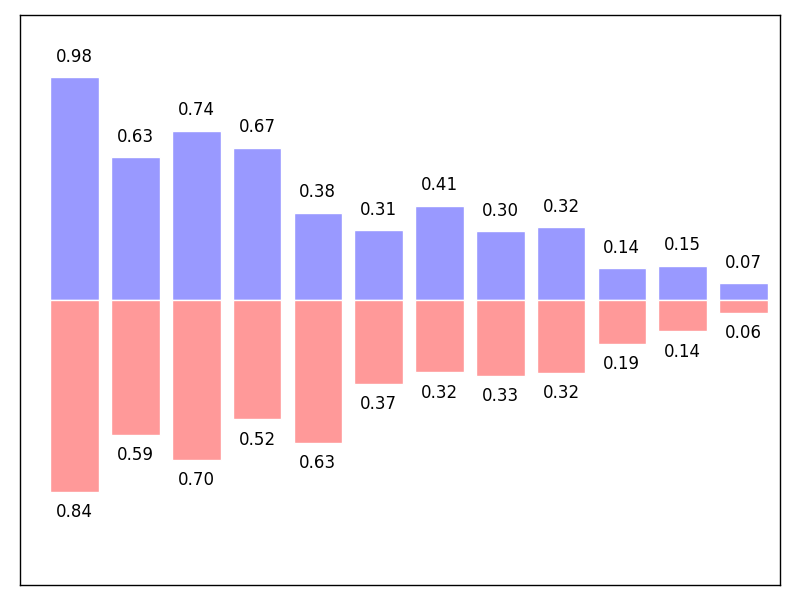
* Ejercicio 1:



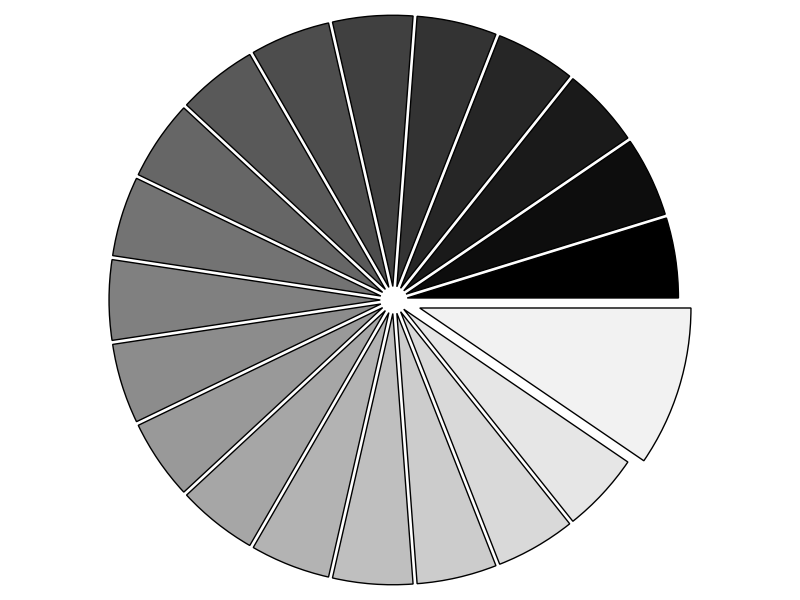
* Ejercicio 2



* Ejercicio 3



* Ejercicio 4:



**Preguntas de esta tarea**

**A partir del siguiente código, intenta obtener el gráfico del dibujo llamado ejercicio 1 en el enunciado.**

**Pista: investiga el parámetro fill\_between de matplotlib.**

**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)**

**A partir del siguiente código, intenta obtener el gráfico del dibujo llamado ejercicio 2 en el enunciado.**

**Pista: el color viene dado por el ángulo que forman x e y con la horizontal**

**n = 1024**

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

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

**plt.scatter(X,Y)**

**A partir del siguiente código, intenta obtener el gráfico del dibujo llamado ejercicio 3 en el enunciado.**

**Pista: cuidado con la alineación del texto en el centro de las barras.**

**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 + 0.4, y + 0.05, '*%.2f*' % y, ha='center', va='bottom')**

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

**A partir del siguiente código, intenta obtener el gráfico del dibujo llamado ejercicio 4 en el enunciado.**

**Pista: tendrás que modificar la Z.**

**Z = np.random.uniform(0, 1, 20)**

**plt.pie(Z)**