María Fernanda Moreno Gómez A01708653

Actividad 4.1: Tablas de frecuencias e histogramas.

Data01.txt:

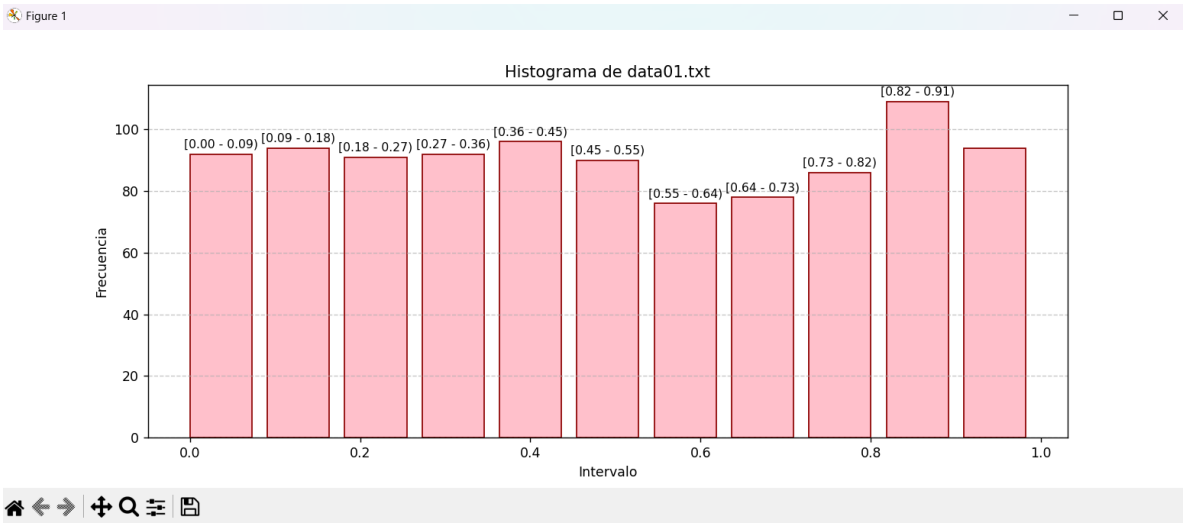


Tabla de frecuencias para data01.txt:

	Intervals	f
0	[0.0002 - 0.0911)	92
1	[0.0911 - 0.1820)	94
2	[0.1820 - 0.2729)	91
3	[0.2729 - 0.3638)	92
4	[0.3638 - 0.4547)	96
5	[0.4547 - 0.5456)	90
6	[0.5456 - 0.6365)	76
7	[0.6365 - 0.7274)	78
8	[0.7274 - 0.8183)	86
9	[0.8183 - 0.9092)	109
10	[0.9092 - 1.0001)	94

Sum of frequencies: 998

Data02.txt:

María Fernanda Moreno Gómez A01708653

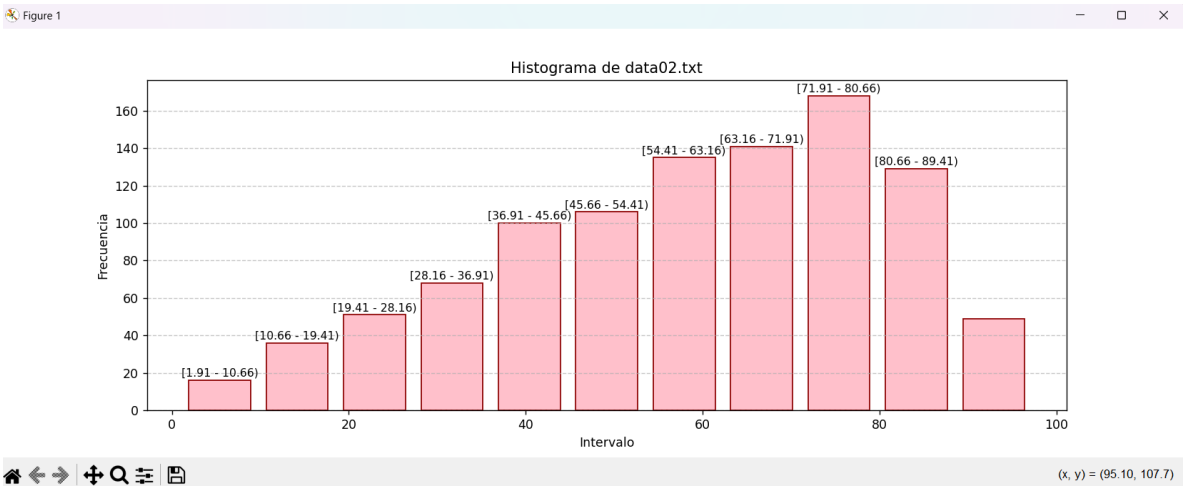


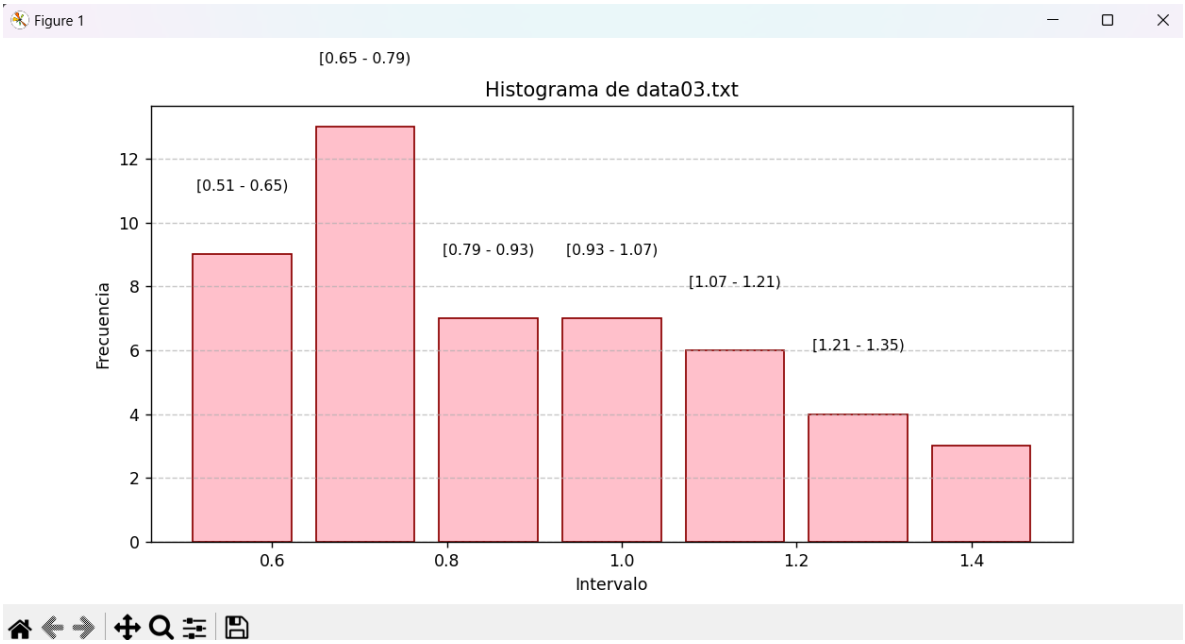
Tabla de frecuencias para data02.txt:

	Intervals	f
0	[1.91 - 10.66)	16
1	[10.66 - 19.41)	36
2	[19.41 - 28.16)	51
3	[28.16 - 36.91)	68
4	[36.91 - 45.66)	100
5	[45.66 - 54.41)	106
6	[54.41 - 63.16)	135
7	[63.16 - 71.91)	141
8	[71.91 - 80.66)	168
9	[80.66 - 89.41)	129
10	[89.41 - 98.16)	49

Sum of frequencies: 999

Data03.txt:

María Fernanda Moreno Gómez A01708653




```
Tabla de frecuencias para data03.txt:
+---+-----------------+----+
|   |    Intervals    | f  |
+---+-----------------+----+
| 0 | [0.509 - 0.650) | 9  |
| 1 | [0.650 - 0.791) | 13 |
| 2 | [0.791 - 0.932) | 7  |
| 3 | [0.932 - 1.073) | 7  |
| 4 | [1.073 - 1.214) | 6  |
| 5 | [1.214 - 1.355) | 4  |
| 6 | [1.355 - 1.496) | 3  |
+---+-----------------+----+

Sum of frequencies: 49
```

Código:

#María Fernanda Moreno Gómez

Actividad 4.1: Tablas de frecuencias e histogramas.

María Fernanda Moreno Gómez A01708653

'''

Instrucciones:

Utilizando Python, de manera individual, escribe un programa que construya la tabla de frecuencias de un conjunto de datos. Además, tu programa deberá graficar el histograma.

'''

```
# Librerías
import matplotlib.pyplot as plt
import numpy as np
import math
import pandas as pd
from tabulate import tabulate

# Función para cargar los datos
def cargar_datos(filename):
    return np.loadtxt(filename)

# Función para redondear hacia arriba con una cantidad de decimales específica
def redondeo_techo(valor, decimales):
    factor = 10 ** decimales
    return math.ceil(valor * factor) / factor

# Función para redondear hacia abajo con una cantidad de decimales específica
def redondeo_piso(valor, decimales):
    factor = 10 ** decimales
    return math.floor(valor * factor) / factor
```

María Fernanda Moreno Gómez A01708653

```
# Función para calcular los parámetros de la tabla de frecuencias
def obtener_parametros(data, decimales):
    N = len(data)
    min_val = np.min(data)
    max_val = np.max(data)

    # Redondear los valores
    min_val = redondeo_piso(min_val, decimales)  # Redondear hacia abajo
    max_val = redondeo_techo(max_val, decimales)  # Redondear hacia arriba
    C = int(math.ceil(1 + 3.30 * np.log10(N)))  # Número de clases
    W = redondeo_techo((max_val - min_val) / C, decimales)  # Ancho de intervalo redondeado hacia arriba

    return N, min_val, max_val, C, W

# Función para calcular la tabla de frecuencias
def calcular_tabla_frecuencias(data, decimales):
    N, min_val, max_val, C, W = obtener_parametros(data, decimales)

    # Crear los límites de los intervalos con redondeo adecuado
    bins = [round(min_val + i * W, decimales) for i in range(C + 1)]
    frecuencias, _ = np.histogram(data, bins=bins)

    # Crear los intervalos como texto para la tabla
    intervalos = [f"[{bins[i]:.{decimales}f} - {bins[i+1]:.{decimales}f})" for i in range(len(bins) - 1)]
```

```
    tabla = pd.DataFrame({"Intervals": intervalos, "f": frecuencias})
    return tabla, bins[:-1], frecuencias

# Función para graficar el histograma
def graficar_histograma(bins, frecuencias, title):
    # Espaciado entre barras
    width = 0.8 * (bins[1] - bins[0])

    plt.bar(bins, frecuencias, width=width, color='pink', edgecolor='darkred',
align='edge')

    # Agregar los intervalos como etiquetas en las barras
    for i in range(len(frecuencias)):
        if i < len(bins) - 1:
            plt.text(bins[i] + width / 2, frecuencias[i] + 2, f'[{bins[i]:.2f} - {bins[i+1]:.2f})',
                     ha='center', color='black', fontsize=9)

    plt.xlabel('Intervalo')
    plt.ylabel('Frecuencia')
    plt.title(title)
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.show()

# Función principal
def main():
```

```
    archivos = ['data01.txt', 'data02.txt', 'data03.txt']
    decimales_dict = {
        'data01.txt': 4,
        'data02.txt': 2,
        'data03.txt': 3
    }

    for archivo in archivos:
        decimales = decimales_dict[archivo]
        print(f"\nTabla de frecuencias para {archivo}:")
        data = cargar_datos(archivo)
        tabla, bins, frecuencias = calcular_tabla_frecuencias(data, decimales)
        print(tabulate(tabla, headers='keys', tablefmt='pretty'))
        print(f"\nSum of frequencies: {sum(frecuencias)}")
        graficar_histograma(bins, frecuencias, f'Histograma de {archivo}')

if __name__ == "__main__":
    main()
```