BUAP

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Materia:

Minería de datos

Alumno:

Mendoza Reyes Jose de Jesus

Matricula:

202263899

Fecha de entrega:

7/Febrero/2025

Tarea:

Practica 3: Probabilidad frecuencial y relativa

Descripción

Durante clase se expusieron las definiciones que tiene la probabilidad, siendo una de ellas la probabilidad frecuencial y relativa. Esta definición de probabilidad se toma cuando no se saben datos importantes relacionados con un evento probabilístico dado. Debido a esto saber con exactitud los elementos involucrados que le dan forma a la probabilidad final es una tarea dificil o imposible en algunos casos. Sin embargo, se propuso una definición que no chocara con lo visto anteriormente sino que la complementara. Según la definición formal dado un evento K que puede aparecer de k maneras su probabilidad final es $\frac{k}{K}$. **En el limite cuando se tiende al infinito esta expresión tenderá a la probabilidad real de dicho límite**.

A partir de esto en la práctica presente vimos un experimento que tiene que ver con dicha probabilidad frecuencial y como se puede ver en distribuciones de rangos de números dados. Se hizo un programa en Python para poder visualizar con tablas y graficas como dada una muestra y una partición de rangos como es que se reparte la muestra en los rangos dados. Entre más aumentemos el tamaño de la muestra la distribución **se irá equilibrando cada vez más**.

Codigo

El codigo hecho en la práctica es el siguiente:

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

sizes = {
        '10': 10,
        '100': 100,
        '1000': 1000,
        'millon': 1000000
}

def generar_muestra(size):
    #Hacemos una muestra del tamaño deseado desde 0 hasta 1500 (1501 para incluir el 1500)
    muestra = np.random.randint(0, 1501, sizes[size])
    muestra.sort() #ordenamos para poder dividir los rangos y que sean excluyentes mas adelante
    print(f"Muestra generada (ordenada): {muestra}")
    return muestra
```

Importamos dos librerias que serán utilizadas en nuestro codigo. Primero, *numpy* que es una librería que nos ayudará para la creación de una muestra random. Despues tenemos la librería pyplot perteneciente al paquete *matplotlib*. Esta nos servirá para hacer un gráfico detallado de la distribución resultante. Tenemos tambien un diccionario sizes global que es el encargado de dar un tamaño dado según la respuesta del usuario. Nuestra primera función en el programa llamada **generar_muestra** está encargada que dado un tamaño de muestra requerido, se hace una lista de números aleatorios en el rango de $[0,1500]$ con **size** números. Después de esto ordenamos la muestra para poder imprimirla.

```python
def creacion_rangos():
    rangos = []
    inicio = 0
    #se generan los 10 intervalos que son excluyentes y que "saltan" de 1500 // 10 en 1500 // 10
    for _ in range(9):
        fin = inicio + (1500 // 10) - 1
        rangos.append((inicio, fin)) #se guarda el inicio del rango y el final del rango en una tupla esto para su posterior impresio
        inicio += 1500 // 10
    rangos.append((inicio, 1500)) #ultimo rango, el que completa los 1500
    return rangos
```

La función **creación_rangos** es la encargada de crear los rangos (10 rangos) los cuales los números caerán en dichos rangos. Primero llenaremos una lista con tuplas de inicio de un rango y final de un rango e incrementamos los rangos en 1500 // 10. En otras palabras,

llenamos la lista rangos con tuplas (inicio,fin) de los rangos que existirán.

```python
def tabla(muestra, rangos):
    total_FA, total_FR = 0, 0
    frecuencias = []

    print("\tRangos\t\tFA\tFR")
    for i, rango_particiones in enumerate(rangos):
        #formulas para el calculo de FA y FR
        #se recorre toda la muestra y se suma 1 por cada numero i que este en el rango (rango_particiones[0] <= i <= rango_particiones[1])
        #es decir, se cuentan los numeros de la muestra cayeron en un intervalo dado y se suma 1 si es asi
        frecuencia_absoluta = sum(1 for valor in muestra if rango_particiones[0] <= valor <= rango_particiones[1])
        #formual general para FR
        frecuencia_relativa = frecuencia_absoluta / len(muestra)

        frecuencias.append(frecuencia_absoluta) #añadimos la frecuencia absoluta para luego imprimirla
        total_FA += frecuencia_absoluta
        total_FR += frecuencia_relativa
        print(f"{i+1:<2} || {rango_particiones[0]:<3} - {rango_particiones[1]:<6} || {frecuencia_absoluta:<6} || {frecuencia_relativa:.2f}")
    print(f"Suma total de Frecuencia Absoluta: {total_FA}")
    print(f"Suma total de Frecuencia Relativa: {total_FR:.1f}")
    return frecuencias
```

La función **tabla** es la encargada en imprimir en terminal una tabla con los rangos, su frecuencia absoluta y su frecuencia relativa. Aquí se calcula tambien ambas frecuencias, la primera recorriendo toda la muestra y buscando números **x** que estén en el rango actual, si es así se suma 1 a la frecuencia absoluta. Posteriormente se calcula la frecuencia relativa dividendo la frecuencia absoluta del rango actual entre el tamaño de la muestra. Posteriormente imprimimos toda la informacion valiosa para el usuario y la suma de ambas frecuencias.

```python
def graficar(rangos, frecuencias, size):
    size = sizes[size]

    #Etiquetas de los intervalos en el eje X
    etiquetas = [f"{r[0]}-{r[1]}" for r in rangos]

    #Configuracion inicial de la grafica
    plt.figure(figsize=(14, 8))
    plt.gca().set_facecolor("#f7f7f7") #blanco

    #Creamos las barras
    barras = plt.bar(etiquetas, frecuencias, color='#4CAF50', edgecolor='black', alpha=0.85) #verde

    # Agregar etiquetas con los valores sobre las barras
    for barra in barras:
        altura = barra.get_height()
        plt.text(barra.get_x() + barra.get_width() / 2, altura + max(frecuencias) * 0.02, str(altura), ha='center', fontsize=12, color='black', fontweight='bold')

    plt.xticks(rotation=45, fontsize=12) #Rotamos etiquetas del eje X
    plt.yticks(fontsize=12)
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    #Etiquetas y titulo
    plt.xlabel("Rangos", color='#2E7D32', fontsize=14, fontweight='bold') #verde
    plt.ylabel("Frecuencia Absoluta", color='#C62828', fontsize=14, fontweight='bold') #rojo
    plt.title(f"Distribucion de Frecuencias en muestra tamaño {size}", color='#C62828', fontsize=20, fontweight='bold') #rojo
    plt.show()
```

La función **graficar** se encarga de mostrar en pantalla una ventana con una gráfica de la informacion de frecuencia relativa y absoluta en

ejes x y. **Esto es útil para poder visualizar cómo se comportan los datos y a que tienden**.

```
def menu():
    print("Dame el tamaño de la muestra: (10, 100, 1000, millon)")
    size = input()
    muestra = generar_muestra(size)
    rangos = creacion_rangos()
    frecuencias = tabla(muestra, rangos)
    graficar(rangos, frecuencias, size)

menu()
```

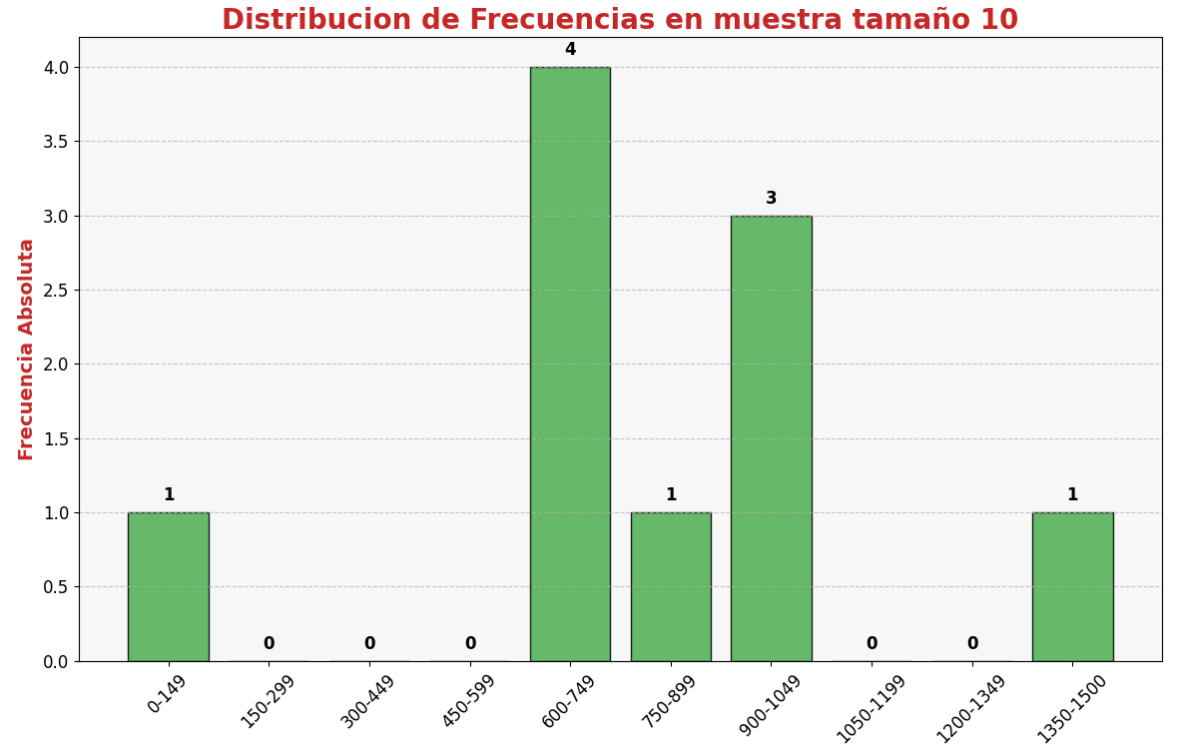
Finalmente en la función **menu** llamamos a todas las demas funciones para poder visualizar todo.

Experimentos/Resultados

Veremos cómo se comportan los datos y los resultados de sus frecuencias con diferentes tamaños:

1. Tamaño 10:

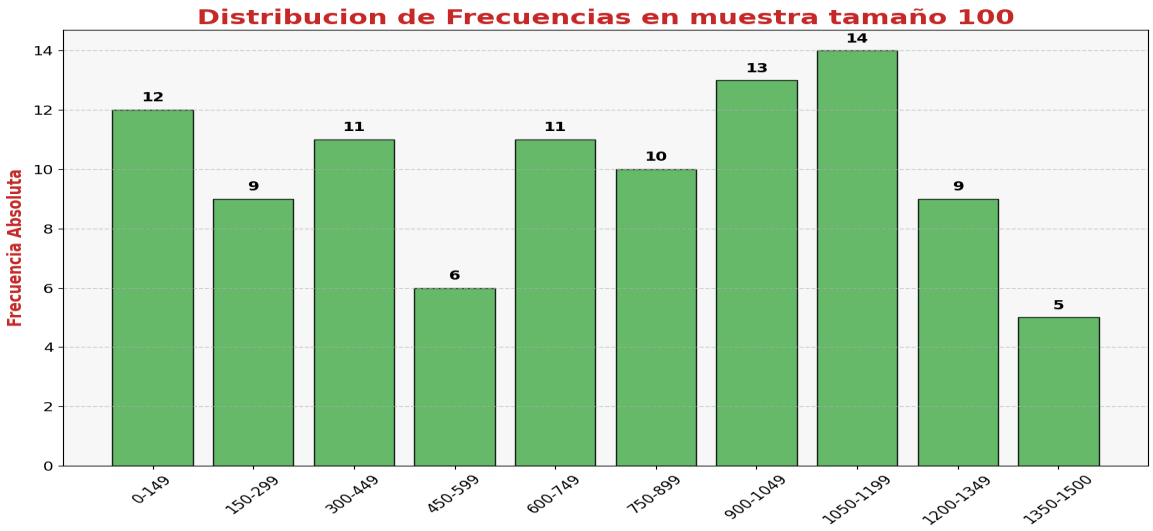
```
Dame el tamaño de la muestra: (10, 100, 1000, millon)
10
Muestra generada (ordenada): [ 104  605  708  711  715  872  905 1017 1043 1353]
        Rangos          FA      FR
0  || 0   - 149    || 1      || 0.10
1  || 150 - 299    || 0      || 0.00
2  || 300 - 449    || 0      || 0.00
3  || 450 - 599    || 0      || 0.00
4  || 600 - 749    || 4      || 0.40
5  || 750 - 899    || 1      || 0.10
6  || 900 - 1049   || 3      || 0.30
7  || 1050 - 1199   || 0       || 0.00
8  || 1200 - 1349   || 0       || 0.00
9  || 1350 - 1500   || 1       || 0.10
Suma total de Frecuencia Absoluta: 10
Suma total de Frecuencia Relativa: 1.0
```



Como vemos el número de números que caen en ciertos rangos es bajo debido al tamaño de la muestra e incluso hay **rangos que estuvieron vacíos**. Tambien vemos que la frecuencia absoluta tiene que ser igual al tamaño de la muestra dada y la frecuencia relativa igual a 1, que son todos los eventos (en este caso rangos) juntos.

2. Tamaño 100:

```
Dame el tamaño de la muestra: (10, 100, 1000, millon)
100
Muestra generada (ordenada): [  35   41   46   51   90  105  122  123  126  126  129  136  168  202
  221  232  245  245  253  268  277  353  362  365  373  374  375  378
  390  419  432  439  467  513  517  555  577  592  637  656  660  670
  671  705  708  713  734  736  746  782  801  803  803  806  831  837
  870  882  898  902  904  907  920  930  937  974  987  993  994 1002
 1035 1046 1058 1059 1065 1068 1077 1084 1106 1109 1116 1131 1135 1171
 1174 1177 1204 1205 1232 1237 1243 1263 1265 1323 1328 1365 1413 1414
 1437 1497]
        Rangos          FA      FR
0  || 0   - 149    || 12     || 0.12
1  || 150 - 299    || 9      || 0.09
2  || 300 - 449    || 11     || 0.11
3  || 450 - 599    || 6      || 0.06
4  || 600 - 749    || 11     || 0.11
5  || 750 - 899    || 10     || 0.10
6  || 900 - 1049   || 13     || 0.13
7  || 1050 - 1199   || 14     || 0.14
8  || 1200 - 1349   || 9      || 0.09
9  || 1350 - 1500   || 5      || 0.05
Suma total de Frecuencia Absoluta: 100
Suma total de Frecuencia Relativa: 1.0
```

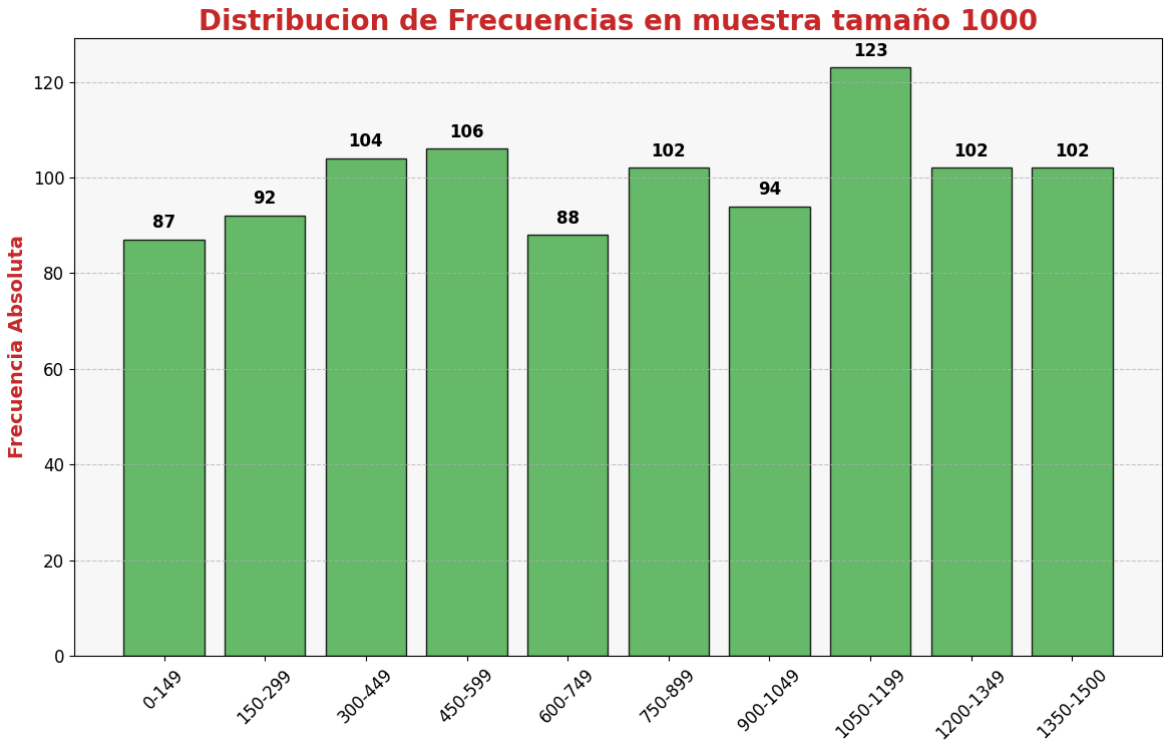


Podemos ver que ahora ya **no hay rangos vacíos** debido a que el tamaño de la muestra es ahora más grande y la probabilidad de que

aparezcan rangos vacíos es más baja. La distribución de los datos no se ve que vaya a estabilizarse o que siga un patrón.

3. Tamaño 1000:

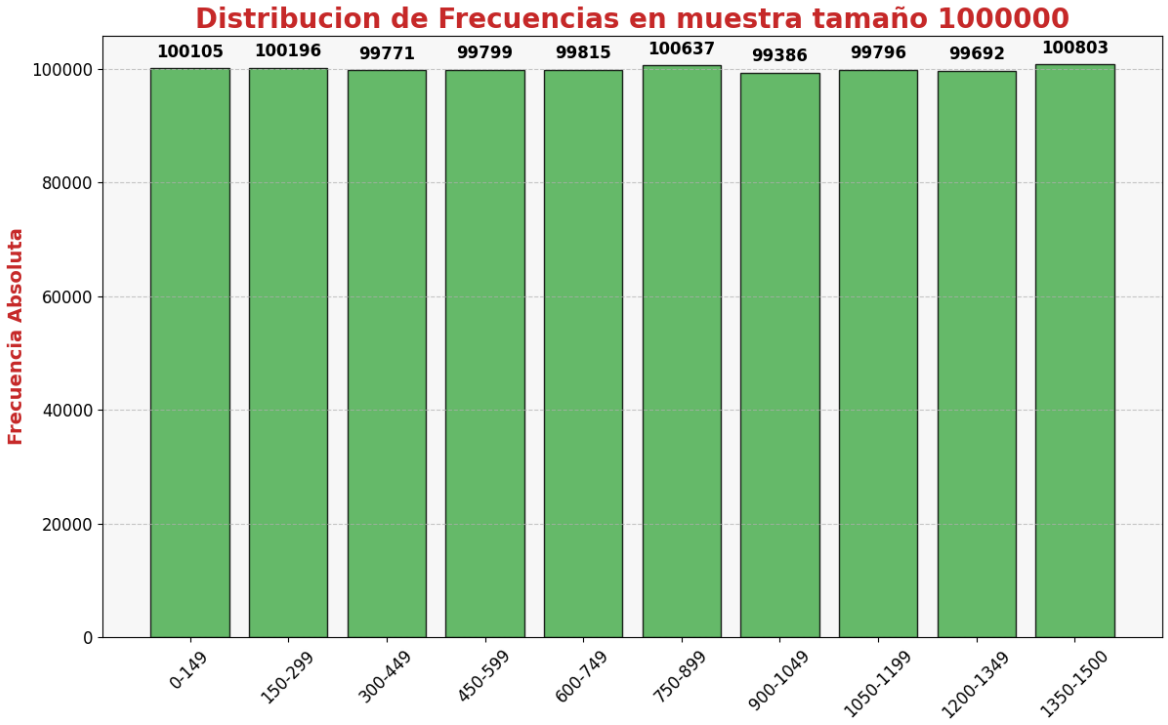
```
 1467 1467 1472 1472 1472 1477 1479 1480
 1495 1495 1497 1500 1500 1500]
        Rangos          FA      FR
0  || 0   - 149    || 87     || 0.09
1  || 150 - 299    || 92     || 0.09
2  || 300 - 449    || 104    || 0.10
3  || 450 - 599    || 106    || 0.11
4  || 600 - 749    || 88     || 0.09
5  || 750 - 899    || 102    || 0.10
6  || 900 - 1049   || 94     || 0.09
7  || 1050 - 1199   || 123    || 0.12
8  || 1200 - 1349   || 102    || 0.10
9  || 1350 - 1500   || 102    || 0.10
Suma total de Frecuencia Absoluta: 1000
Suma total de Frecuencia Relativa: 1.0
```



La muestra ahora se ha vuelto mas grande y la distribución **empieza a estabilizarse teniendo ligeros cambios entre rango**. Lo mismo le pasa a la frecuencia relativa, la cual empieza a estabilizarse alrededor de **0.10**.

4. Tamaño millón:

```
Dame el tamaño de la muestra: (10, 100, 1000, millon)
millon
Muestra generada (ordenada): [   0    0    0 ... 1500 1500 1500]
        Rangos          FA      FR
0  || 0   - 149    || 100105 || 0.10
1  || 150 - 299    || 100196 || 0.10
2  || 300 - 449    || 99771  || 0.10
3  || 450 - 599    || 99799  || 0.10
4  || 600 - 749    || 99815  || 0.10
5  || 750 - 899    || 100637 || 0.10
6  || 900 - 1049   || 99386  || 0.10
7  || 1050 - 1199   || 99796  || 0.10
8  || 1200 - 1349   || 99692  || 0.10
9  || 1350 - 1500   || 100803 || 0.10
Suma total de Frecuencia Absoluta: 1000000
Suma total de Frecuencia Relativa: 1.0
```



Como último experimento podemos ver que **la frecuencia relativa y la distribución de números se estabilizaron.** En este caso en específico la frecuencia relativa de todos los rangos dio **0.10** y su distribución mostrada es muy pareja como se ve en la gráfica.

Conclusión

La manipulación de muestras en cierto tipo de datos puede que tenga un patrón que sigue el cual en tamaños muy grandes de la muestra se hace presente. Parece muy impredecible el como que n cantidad de números aleatorios en un rango dado caigan en varios rangos y estos al final se estabilicen con una cantidad similar de números para cada rango. Estos patrones pueden ayudarnos mucho a la detección de informacion en un espacio muestral y nos puede ser de utilidad en muchos casos, por lo cual saber identificar estos patrones es importante.