

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Genetic Algorithms

Reporte Primera Práctica:

“Graficación de un histograma utilizando el lenguaje c”

Grupo 3CM5

Romero Godinez José Enrique

Introducción:

Un histograma es la representación gráfica de la distribución numérica de una variable, es decir mapea los valores que esta variable puede tomar a lo largo de un rango[1], en estadística es utilizado para obtener una vista general de la distribución de una población o de una muestra respecto a un rasgo en específico, para así observar la tendencia hacia un valor que la variable tiene.

Objetivo:

Desarrollar un programa en lenguaje c que reciba datos referentes a la variable a representar y que grafique su histograma.

Solución:

Dado que el lenguaje c no permite de manera nativa las interfaces gráficas se hará uso del programa gnuplot[2] de linux, el cuál será invocado por un programa escrito en lenguaje c que será el encargado de recibir los datos a graficar.

Implementación:

Codigo del programa en c:

```
#include <stdlib.h>
```

```
1. #include <stdio.h>
2. #include <unistd.h>
3. #include <string.h>
4.
5. #define NUM_COMANDOS 6
6. void ordenar(double*, double*, int);
7. int main(){
8.     int numPuntos=0;
9.     char cadena[100]="set xlabel \"----\";
10.    char valor[30];
11.    // X, Y valores de los puntos a graficar
12.    system("clear");
13.    printf("Ingrese el numero de parejas de datos que desea ingresar al histograma:");
14.    scanf("%d",&numPuntos);
15.    printf("Ingrese el rasgo a representar en el histograma\n");
16.    scanf("%s",valor);
17.    double* valoresX=(double*)malloc(sizeof(double)*numPuntos);
18.    double* valoresY=(double*)malloc(sizeof(double)*numPuntos);
19.    for(int k=0;k<numPuntos;k++)
20.    {
21.        printf("\nIngrese la frecuencia correspondiente a la pareja de datos %i\n",k+1);
22.        getchar();
23.        scanf("%lf",&valoresY[k]);
24.        printf("\nIngrese el valor correspondiente a la pareja de datos %i\n",k+1);
25.        getchar();
26.        scanf("%lf",&valoresX[k]);
27.    }
28.    //Los valores en x tienen que estar ordenados de menor a mayor para una correcta lectura del
    histograma
```

```

29. ordenar(valoresX,valoresY,numPuntos);
30. for(int j=0;j<numPuntos;j++){
31.     printf("%lf\n", valoresX[j]);
32.     /* se crea y se abre el archivo puntosGraficar.txt en modo escritura
33.      * para almacenar los valores de x y y que están declarados en los arreglos
34.      * valoresX y valoresY*/
35.     FILE * archivoPuntos = fopen("puntosGraficar.txt", "w");
36.
37.     /*Guardar los puntos x,y en el archivo de texto creado y abierto previamente*/
38.     for (int i=0;i<numPuntos+1;i++){
39.         if(i==0)
40.         {
41.             fprintf(archivoPuntos, "%lf %d \n", valoresX[0]-10, 0);
42.             /*}else if(i==numPuntos){
43.                 fprintf(archivoPuntos, "%lf %d \n", valoresX[numPuntos-1]+10, 0);*/
44.             }else{
45.                 fprintf(archivoPuntos, "%lf %lf \n", valoresX[i-1], valoresY[i-1]);
46.             }
47.         }
48.         strcat(cadena,valor);
49.         strcat(cadena,"--->\n");
50.
51.         /*lista de comandos para ejecutar y configurar la visualización que tendrán
52.          * los puntos en la gráfica con gnuplot*/
53.         char * configGnuplot[] = {"set title \"Histograma\"",
54.                                     "set ylabel \"----Frecuencia--->\"",
55.                                     cadena,
56.                                     "set style data histogram",
57.                                     "set style histogram cluster gap 1",
58.                                     "plot \"puntosGraficar.txt\" using 1:2 with linespoints"
59.                                     };
60.
61.         /*Se crea una archivo de tipo poen, es una tubería IPC que se usa, para
62.          * ejecutar gnuplot y enviarle el archivo a graficar*/
63.         FILE * ventanaGnuplot = popen ("gnuplot -persist", "w");
64.         // Ejecuta los comandos de configGnuPlot 1 por 1
65.         for (int i=0;i<NUM_COMANDOS;i++){
66.             fprintf(ventanaGnuplot, "%s \n", configGnuplot[i]);
67.         }
68.         fclose(archivoPuntos);
69.         return 0;
70. }
71.
72. //ordena las parejas a graficar en el histograma dependiendo del valor de la variable
73. void ordenar(double *puntosX,double *puntosY,int tamano)
74. {
75.     double temp1,temp2;
76.     for(int i=1; i<tamano;i++){
77.         for(int j=0;j<tamano-1;j++){
78.             if(puntosX[j]>puntosX[j+1])

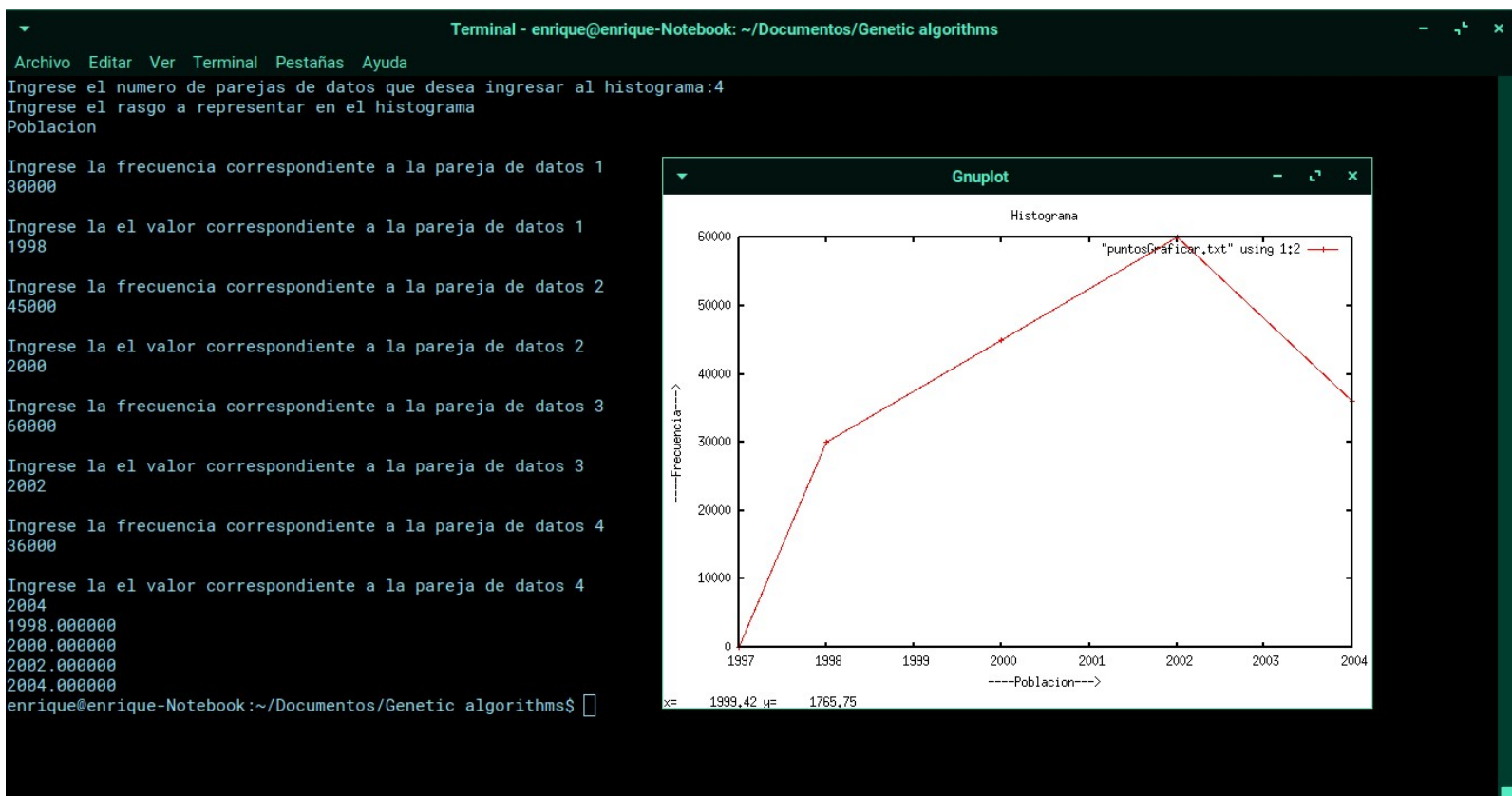
```

```

79.  {
80.    temp1=puntosX[j];
81.    temp2=puntosY[j];
82.    puntosX[j]=puntosX[j+1];
83.    puntosY[j]=puntosY[j+1];
84.    puntosX[j+1]=temp1;
85.    puntosY[j+1]=temp2;
86.  }
87. }
88. }
89.}

```

Ejecucion del programa:



Conclusiones:

El análisis de un histograma es muy útil para cuando queremos saber el comportamiento de alguna variable, por ejemplo el cambio en el número de la población a lo largo de los años, es por eso que implementar un programa que nos dé la herramienta gráfica facilita mucho el tratamiento de los datos.

Referencias:

- [1]<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=23&ved=2ahUKEwiKyN6auqzdAhWi7IMKHV2vCdQQFjAWegQIARAC&url=http%3A%2F%2Fasesorias.cuautitlan2.unam.mx%2FLaboratoriovirtualdeestadistica%2FDOCUMENTOS%2FTEMA%25201%2F7.%2520HISTOGRAMAS.pdf&usg=AOvVaw0DIeBVGFGWlIXwXAOWWDBi>
- [2]<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiA1pLduqzdAhUPPq0KHXYUC3gQFjAAegQICRAB&url=http%3A%2F%2Fwww.gnuplot.info%2F&usg=AOvVaw3VAHCVnADPjiSQMpRM9jNG>