



Programación en Ingeniería.

Proyecto Final

Análisis Estadístico

Alumnos:

Michel A. Ramos Soto
Ricardo Romero Vega
Naim Tejeda Díaz
Cielo Mayerly Quiroga Gonzales

NUA(s):

768936
389765
145154
146722

Maestro:

Dr. Mario Alberto Ibarra Manzano

Correo:

mibarram@gmail.com

Fecha de entrega:

07/12/18

Universidad de Guanajuato

Introducción

En este último proyecto aplicaremos los conocimientos adquiridos en clase para emplearlos en la práctica, el reto es hacerlo lo más eficientes posibles y lo más limpio posible. El análisis estadístico, y muy particularmente el análisis multivariante, es un conjunto de técnicas estadísticas que permiten detectar patrones de comportamiento ocultos y, basándose en los mismos, establecer predicciones.

C es un lenguaje de programación originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell,2 como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

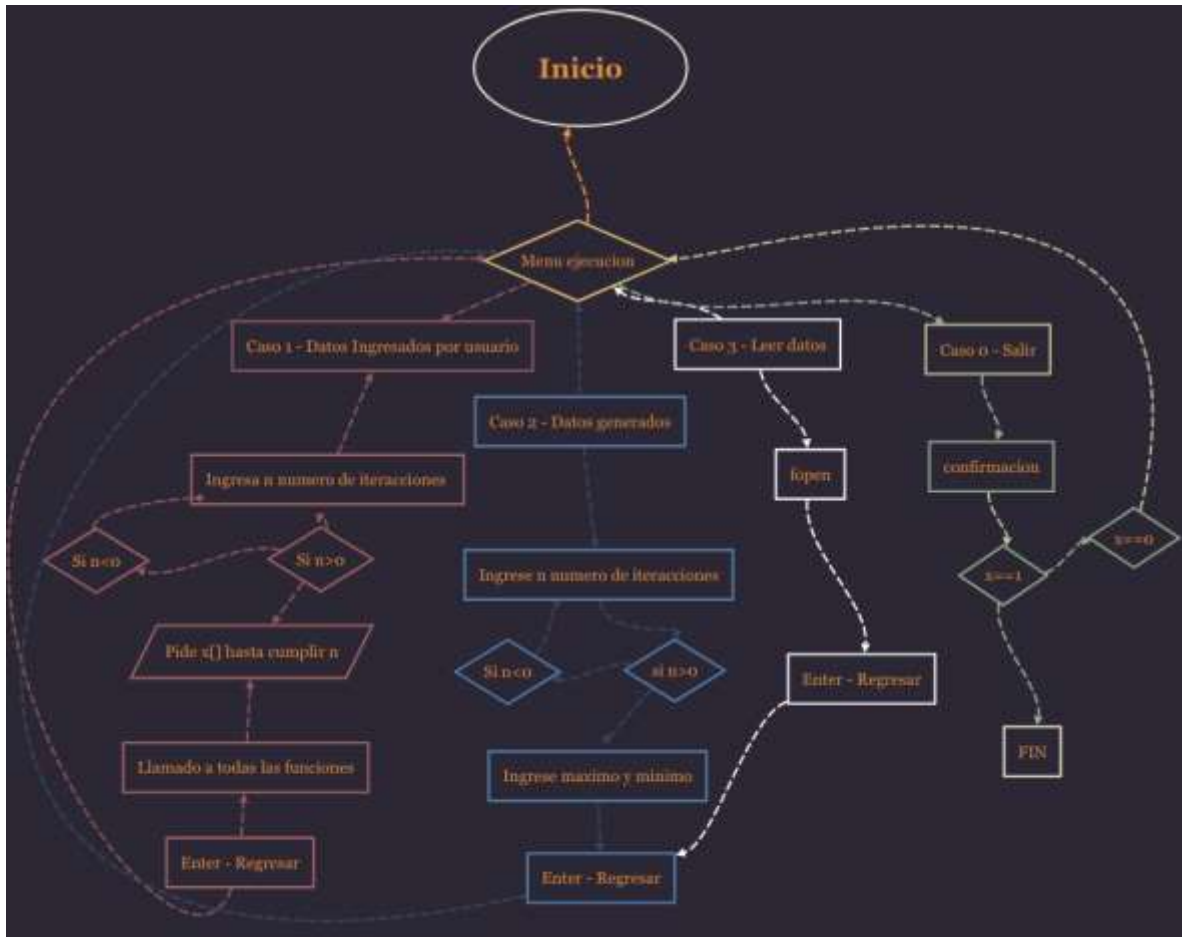


Fig. 1. Análisis Estadístico

Objetivos

- Simular el programa para obtener los valores de las funciones para completar la tabla.
- Emplear lo aprendido a clase
- Hacer las funciones lo más eficaces y eficientes posibles.
- Hacer el programa de modo que sea útil en futuros cursos.
- Trabajo en equipo.

Librerías, funciones y menú.



```
/*Proyecto Final - Programacion en Ingenieria
Analisis estadistico */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int Menu();
int MenuEjec(int opcion, FILE *fp);
int Minimo(float *X, int n, FILE *fp);
int Maximo(float *X, int n, FILE *fp);
int Mediana(float *X, int n, FILE *fp);
int Media(float *X, int n, FILE *fp);
int MediaGeometrica(float *X, int n, FILE *fp);
int Moda(float *X, int n, FILE *fp);
int Varianza(float *X, float media, int n, FILE *fp);
int Desviacion(float varianza, float mediana, float media, FILE *fp);
int CoeficienteVariacion(float desviacion, float mediana, FILE *fp);
int Cuartiles(float *X, int n, FILE *fp);
int Deciles(float *X, int n, FILE *fp);
int Percentiles(float *X, int n, FILE *fp);
int Momento_descentrado(float *valores, int *cantidad, int c);
int Momento_centrado(float aux, float aux2, int count, float *valores, int *cantidad, int c);
int *leer(int *n);
int guardar(float *X, int n);
int momentoestandar(float desviacion, float media, FILE *fp);

int main(int argc, const char * argv[])
{
    printf("      Programacion en Ingenieria      Proyecto Final\n\n");
    FILE *fp;
    fp = fopen("Datos.txt", "wt");
    int opcion;
    do
    {
        opcion = Menu();
        opcion = MenuEjec(opcion, fp);

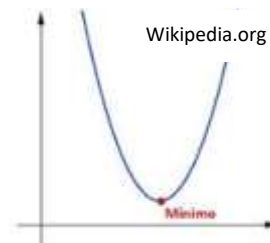
        printf("\n\nPara continuar, presiona <Enter>.");
        getchar();
        getchar();
    } while (opcion);
    fclose(fp);

    return 0;
}

int Menu()
{
    int eleccion;
    do
    {
        printf("\n      Analisis Estadisticos de Datos\n\n");
        printf("Elija una de las opciones siguientes:\n");
        printf("\n1. Datos Ingresados por Usuario");
        printf("\n2: Datos Generados Automaticamente");
        printf("\n3: Leidos desde un archivo");
        printf("\n\n0. SALIR.");

        printf("\n\n");
        scanf("%d", &eleccion); fflush(stdin);
    }
}
```

Mínimo



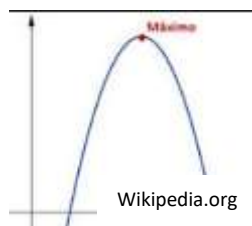
Código

```
int Minimo(float *X, int n, FILE *fp)
{
    int i, j;
    float min, aux;
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(X[j] > X[i])
            {
                aux=X[i];
                X[i]=X[j];
                X[j]=aux;
            }
    for(i=0; i<n; i++){
        min=X[n-1];
    }
    printf("\nEl Valor Minimo es: %f", min);
    fprintf(fp, "\nEl Valor Minimo es: %f", min);
    return 0;
}
```

Tabla de simulación

#	N	Maximo	Minimo	Tiempo	Min
1	100	1	0		0,002106
2	1000	10	0		0,006104
3	1000	10	-10	1	-9,948118
4	1000	1	-1	1	-0,99939
5	1000	0	-10	1	-9,997253
6	2000	100	50	1	49,016758
7	2000	200	150	1	149,00103
8	2000	5	-5	1	-4,996643
9	2000	1000	0	1	0,244148
10	2000	10000	-10000	1	-9953,01236
11	4000	1	-1	2	-1
12	5000	1	-1	2	-0,956567
13	6000	1	-1	2	-0,965617
14	7000	1	-1	3	-0,996561
15	8000	1	-1	3	-0,965061
16	9000	1	-1	4	-0,969912
17	10000	1	-1	6	-0,965667
18	20000	1	-1	7	-0,996547
19	30000	1	-1	8	-0,998944
20	50000	1	-1	9	-0,999114

Máximo



Código

```
int Maximo (float *X, int n, FILE *fp)
{
    int i;
    float max;
    for (i=0; i<n; i++)
    {
        max=*(X+(n-1));
    }
    printf("\nEl Valor Maximo es: %f", max);
    fprintf(fp, "\nEl Valor Maximo es: %f", max);
    return 0;
}
```

Tabla de simulación

#	N	Maximo	Minimo	Tiempo	Maximo
1	100	1	0		0,967803
2	1000	10	0		9,990845
3	1000	10	-10	1	9,979857
4	1000	1	-1	1	0,999451
5	1000	0	-10	1	-0,003662
6	2000	100	50	1	99,9694
7	2000	200	150	1	199,989319
8	2000	5	-5	1	4,989013
9	2000	1000	0	1	999,420166
10	2000	10000	-10000	1	9985,35156
11	4000	1	-1	2	0,999329
12	5000	1	-1	2	0,993315
13	6000	1	-1	2	0,963219
14	7000	1	-1	3	0,956785
15	8000	1	-1	3	0,955748
16	9000	1	-1	4	0,799863
17	10000	1	-1	6	0,998755
18	20000	1	-1	7	0,999868
19	30000	1	-1	8	0,996566
20	50000	1	-1	9	0,999987

Media aritmética

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Wikipedia.org

Código

```
int Media(float *X,int n,FILE *fp)
{
    int i;
    float media,*p_media;
    for(i=0,media=0;i<n;i++)
    {
        media+=*(X+i);
    }
    media /=n;
    p_media=&media;
    printf("\nLa Media Aritmetica es: %f", *p_media);
    fprintf(fp,"\nLa Media Aritmetica es: %f", *p_media);
    Varianza(X,media,n,fp);
    return 0;
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Media Aritmetica
1	100	1	0	0,518517
2	200	10	0	4,662919
3	300	10	-10	0,38857
4	400	1	-1	0,04543
5	500	100	50	75,384369
6	600	150	100	125,322525
7	700	200	50	125,859688
8	800	2	-1	0,508568
9	900	6	2	3,928371
10	1000	1	0	0,491887
11	1100	2	1	1,488567
12	1200	4	2	3,020088
13	1300	500	250	376,225342
14	1400	300	100	197,49234
15	1500	125	75	99,449852
16	1600	45	20	32,448975
17	1700	14	7	10,521947
18	1800	12	6	8,916922
19	1900	5	2	3,509611
20	2000	6	-6	0,092412
21	2500	1	-10	-4,533501

Media geométrica

Wikipedia.org

$$\bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

Código

```
int MediaGeometrica(float *X,int n,FILE *fp)
{
    int i;
    float mediageo;
    for(i=0,mediageo=1;i<n;i++)
    {
        mediageo *=*(X+i);
    }
    if(mediageo<0)
    {
        mediageo*=-1;
        mediageo=pow(mediageo,(1/n));
        mediageo*=-1;
    }
    else{
        mediageo=pow(mediageo,(1/n));
    }
    printf("\nLa Media Geometrica es: %f", mediageo);
    fprintf(fp,"\nLa Media Geometrica es: %f", mediageo);
    return 0;
}
```

Tabla de simulaciones

#	N.Datos Ingresados por el usuario						Maximo	Minimo	Media Geometrica
1	44	23	355	33	45	12	355	12	43,09095252
2	32	455	66	12	45	6	455	6	38,21275825
3	23	12	9	15	667	30	667	9	30,11260471
4	12	34	45	3	33	34	45	3	19,88372729
5	34	55	23	53	4	54	55	4	28,10065937
6	23	34	34	34	23	23	34	23	27,96426291
7	53	2	3	56	5	5	56	2	8,738278041
8	45	34	3	7	23	34	45	3	17,1140546
9	34	34	4	54	5	6	54	4	13,98799516
10	53	23	6	23	23	3	53	3	15,04718685
11	53	5	3	76	5	6	76	3	11,04206576
12	56	3	7	34	23	34	56	5	17,74933823
13	76	6	5	89	6	6	89	5	13,92957585
14	7	5	7	4	3	34	34	3	6,81246642
15	64	6	3	34	6	6	64	3	10,58942347
16	545	45	7	3	34	34	545	3	29,0044315
17	45	6	34	7	6	6	45	6	11,50024794
18	3	45	7	23	23	2	45	2	9,999683308
19	23	6	34	7	5	56	56	5	14,47445918
20	43	45	7	3	23	2	45	2	11,09880828
21	4	6	12	7	23	4	23	4	7,55172893

Mediana

$$Mediana = x_{i1} + \left(\frac{(N_M/2) - N_{i-1}}{f_i} \right) \cdot (x_{i2} - x_{i1})$$

Wikipedia.org

Código

```
int Mediana(float *X,int n,FILE *fp)
{
    int i,p,p2;
    float mediana;
    for(i=0;i<n;i++)
    {
        if(n%2 != 0)
        {
            p=((n+1)/2)-1;
            mediana=*(X+p);
        }else{
            p=(n/2);
            p2=((n)/2)-1;
            mediana=(*(X+p2)+*(X+p))/2;
        }
    }
    printf("\nLa Mediana es: %f\n", mediana);
    fprintf(fp,"\nLa Mediana es: %f\n", mediana);
    return 0;
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Mediana
1	100	1	0	0,539293
2	200	10	0	5,248116
3	300	10	-10	0,259102
4	400	1	-1	-0,002594
5	500	100	50	74,251534
6	600	150	100	125,366989
7	700	200	50	125,906403
8	800	2	-1	0,531037
9	900	6	2	3,994995
10	1000	1	0	0,496307
11	1100	2	1	1,490738
12	1200	4	2	3,0318
13	1300	500	250	5098,756348
14	1400	300	100	198,129211
15	1500	125	75	99,160744
16	1600	45	20	32,3764
17	1700	14	7	10,493698
18	1800	12	6	8,885189
19	1900	5	2	3,548524
20	2000	6	-6	0,12122
21	2500	1	-10	-4,572344

Moda

$$M = L_i + \left(\frac{D_1}{D_1 + D_2} \right) A_i$$

Wikipedia.org

Código

```
int Moda(float *X, int n, FILE *fp)
{
    int c=0, ci=0, d, max, i=0, j=0;
    int frec[n], *Cantidad, *can;
    float *valores, *val;

    valores=(float*)malloc(n*sizeof(float));
    cantidad=(int*)malloc(n*sizeof(int));
    val=(float*)malloc(n*sizeof(float));
    can=(int*)malloc(n*sizeof(int));

    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            if(X[i]==X[j])
            {
                c++;
            }
        }

        frec[i]=c;
        c=0;

        valores[i]=X[i];
        cantidad[i]=frec[i];

        for(i=1; i<n; i++)
        {
            d=0;
            for(j=0; j<i-1; j++)
            {
                if(X[i]==X[j])
                {
                    d++;
                }
            }
        }
    }

    // (d==0 88 j=1)
    {
        c=c+1;
        valores[c]=X[i];
        cantidad[c]=frec[i];
    }
}

max=cantidad[0];
for(i=1; i<n; i++)
{
    if(max<cantidad[i])
    {
        max=cantidad[i];
    }
}

for(i=0; i<n; i++)
{
    if(cantidad[i]==max)
    {
        ci=i+1;
        can[ci]=cantidad[i];
        val[ci]=valores[i];
    }
}

printf("Moda: valor = Cantidad\n");
for(i=1; i<=ci; i++)
{
    printf("  %d  %d\n", val[i], can[i]);
}

// Liberacion de memoria
free(valores);
free(cantidad);
free(val);
free(can);
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Moda
1	100	1	0	0.820002
2	200	10	0	Amodal
3	300	10	-10	-2.872037
4	400	1	-1	-0.443159 rep:2 -0.372539 rep:2 51.120029 rep:2
5	500	100	50	92.996002 rep:2
6	600	150	100	102.851952 101.825005 rep:2 108.096558 rep:2 115.443893 rep:2 126.192513 rep:2 144.833221 rep:2
7	700	200	50	163.771484 rep:2 -0.145787 rep:2 -0.123173 rep:2 0.067721 rep:2 0.363353 rep:2 0.373424 rep:2
8	800	2	-1	2.614277 rep:2 3.765191 rep:2 4.404370 rep:2 4.446486 rep:2 5.678823 rep:2 5.843257 rep:2 5.854376 rep:2 5.891598 rep:2
9	900	6	2	0.030976 rep:2 0.091617 rep:2 0.144993 rep:2 0.178076 rep:2 0.198981 rep:2 0.358623 rep:2 0.484695 rep:2
10	1000	1	0	

Varianza (segundo momento estándar)

$$\sigma_n^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})^2 = \left(\frac{1}{n} \sum_{i=1}^n x_i^2 \right) - \bar{X}^2 = \frac{1}{n^2} \sum_{i=1}^n \sum_{j>i}^n (x_i - x_j)^2$$

Wikipedia.org

Código

```
int Varianza(float *X, float media, int n, FILE *fp)
{
    int i;
    float varianza, vr;
    for(i=0, varianza=0, vr=0; i<n; i++)
    {
        varianza=(*(X+i))-(media);
        varianza=pow(varianza,2);
        vr+=varianza;
    }
    vr/=n;
    printf("\nLa Varianza es: %f", vr);
    fprintf(fp, "\nLa Varianza es: %f", vr);
    Desviacion(vr, media, media, fp);

    return 0;
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Varianza
1	100	1	0	195,7558
2	200	10	0	8,015994
3	300	10	-10	32,263741
4	400	1	-1	0,341138
5	500	100	50	213,08415
6	600	150	100	210,4911
7	700	200	50	1885,3213
8	800	2	-1	0,74599
9	900	6	2	1,276823
10	1000	1	0	0,080455
11	1100	2	1	0,081907
12	1200	4	2	0,332979
13	1300	500	250	5098,7563
14	1400	300	100	3349,8396
15	1500	125	75	201,39021
16	1600	45	20	51,814415
17	1700	14	7	3,951787
18	1800	12	6	3,01777
19	1900	5	2	0,758389
20	2000	6	-6	11,885813
21	2500	1	-10	10,173334

Desviación Estándar

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

Wikipedia.org

Código

```
int Desviacion(float varianza,float mediana,float media,FILE *fp)
{
    float desviacion;
    desviacion=sqrt(varianza);
    printf("\nLa Desviacion Estandar es: %f",desviacion);
    fprintf(fp,"\nLa Desviacion Estandar es: %f",desviacion);
    CoeficienteVariacion(desviacion,mediana,fp);
    momentoestandar(desviacion,media,fp);
    return 0;
}
```

Tabla de simulación

	A	B	C	D	E	F	G	H	I	J	K	L
1	#	N	Máx.	Mín.	Desviacion Estandar	Coeficiente de varianza	#	N	Máx.	Mín.	Desviacion Estandar	Coeficiente de varianza
2												
3	1	100	1	0	0.274498	0.592798	11	4000	1	-1	0.576954	-184.015182
4	2	1000	10	0	2.927555	0.582568	12	5000	1	-1	0.574463	-331.142822
5	3	1000	10	-10	5.709074	53.881134	13	6000	1	-1	0.577537	-107.945274
6	4	1000	1	-1	0.589314	-30.245096	14	7000	1	-1	0.579623	-81.601608
7	5	1000	0	-10	2.89339	-0.591275	15	8000	1	-1	0.578702	-70.189293
8	6	2000	100	50	14.366097	0.19055	16	9000	1	-1	0.573459	-69.86377
9	7	2000	200	150	14.456321	0.082506	17	10000	1	-1	0.575361	-113.573074
10	8	2000	5	-5	2.910821	39.297604	18	20000	1	-1	0.580512	-98.909332
11	9	2000	1000	0	288.391388	0.577296	19	30000	1	-1	0.577077	462.892487
12	10	2000	10000	-10000	5907.021484	-1009.78967	20	50000	1	-1	0.575673	-668.088684

Coeficiente de variación

$$cv = \frac{\sigma}{\mu} \times 100$$

Wikipedia.org

Código

```
int CoeficienteVariacion(float desviacion, float mediana, FILE *fp)
{
    float coevar;
    coevar=desviacion/mediana;
    printf("\nEl Coeficiente de Variacion es: %f", coevar);
    fprintf(fp, "\nEl Coeficiente de Variacion es: %f", coevar);
    return 0;
}
```

Tabla de simulación

	A	B	C	D	E	F	G	H	I	J	K	L
1	#	N	Máx.	Mín.	Desviacion Estandar	Coeficiente de varianza	#	N	Máx.	Mín.	Desviacion Estandar	Coeficiente de varianza
2												
3	1	100	1	0	0.274498	0.592798	11	4000	1	-1	0.576954	-184.015182
4	2	1000	10	0	2.927555	0.582568	12	5000	1	-1	0.574463	-331.142822
5	3	1000	10	-10	5.709074	53.881134	13	6000	1	-1	0.577537	-107.945274
6	4	1000	1	-1	0.589314	-30.245096	14	7000	1	-1	0.579623	-81.601608
7	5	1000	0	-10	2.89339	-0.591275	15	8000	1	-1	0.578702	-70.189293
8	6	2000	100	50	14.366097	0.19055	16	9000	1	-1	0.573459	-69.86377
9	7	2000	200	150	14.456321	0.082506	17	10000	1	-1	0.575361	-113.573074
10	8	2000	5	-5	2.910821	39.297604	18	20000	1	-1	0.580512	-98.909332
11	9	2000	1000	0	288.391388	0.577296	19	30000	1	-1	0.577077	462.892487
12	10	2000	10000	-10000	5907.021484	-1009.78967	20	50000	1	-1	0.575673	-668.088684

Cuartiles

$$Q_1 = L_{i-1} + \frac{\frac{N}{4} - F_{i-1}}{f_i} \cdot a$$

Wikipedia.org

Código

```
int Cuartiles(float *X,int n, FILE *fp)
{
    float *pQ;
    int q, i;
    pQ=(float *)malloc(3*sizeof(float));
    fprintf(fp,"\nCuartiles\n");
    for(i=0;i<3;i++)
    {
        q=(i+1)*(n+1)/4;
        pQ[i]=X[q-1]+((i+1)*((X[q]-X[q-1])/4));
        printf("Cuartil %d = %f\n",i+1,pQ[i]);
        fprintf(fp,"Cuartil %d = %f\n",i+1,pQ[i]);
    }
    free(pQ);
    return 0;
}
```

Tabla de simulación

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	#	N	Máx.	Mín.	Cuartiles			#	N	Máx.	Mín.	Cuartiles		
2					1	2	3					1	2	3
3	1	100	1	0	0.732498	0.470992	0.243812	11	4000	1	-1	0.489578	-0.019532	-0.511856
4	2	1000	10	0	7.489014	5.109867	2.636418	12	5000	1	-1	0.526231	0.003113	-0.50441
5	3	1000	10	-10	5.18601	-0.178228	-5.015107	13	6000	1	-1	0.497787	0.011719	-0.502045
6	4	1000	1	-1	0.533357	0.02945	-0.451643	14	7000	1	-1	0.49527	0.01883	-0.48027
7	5	1000	0	-10	-2.429273	-4.951476	-7.339549	15	8000	1	-1	0.469253	-0.013459	-0.514344
8	6	2000	100	50	87.341682	74.712364	62.470245	16	9000	1	-1	0.492828	0.004486	-0.51001
9	7	2000	200	150	187.539291	175.997955	162.998215	17	10000	1	-1	0.4991	-0.006256	-0.496338
10	8	2000	5	-5	2.418516	0.094607	-2.306665	18	20000	1	-1	0.501007	-0.006653	-0.510178
11	9	2000	1000	0	757.751709	518.539978	260.948517	19	30000	1	-1	0.501007	0.003235	-0.495697
12	10	2000	10000	-10000	4790.79541	-65.004425	-5218.97022	20	50000	1	-1	0.505478	-0.006561	-0.509674

Deciles

$$D_k = Li + \frac{\frac{k \cdot N}{10} - F_{i-1}}{f_i} \cdot ai$$

Wikipedia.org

Código

```
int Deciles(float *X, int n, FILE *fp)
{
    float *pD;
    int q, i;
    pD=(float *)malloc(9*sizeof(float));
    fprintf(fp, "\nDeciles\n");
    for(i=0; i<9; i++)
    {
        q=(i+1)*(n+1)/10;
        pD[i]=X[q-1]+((i+1)*((X[q]-X[q-1])/10));
        printf("Decil %d = %f\n", i+1, pD[i]);
        fprintf(fp, "Decil %d = %f\n", i+1, pD[i]);
    }
    free(pD);
    return 0;
}
```

Tabla de simulación

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	#	N	Máx.	Mín.	Deciles								
2					1	2	3	4	5	6	7	8	9
3	1	100	1	0	0.906024	0.758599	0.65255	0.577532	0.470992	0.396515	0.268978	0.175042	0.108396
4	2	1000	10	0	8.852748	8.006348	7.065676	6.060793	5.109867	4.206976	3.155309	2.009888	1.040468
5	3	1000	10	-10	7.944151	6.070681	4.050783	1.839106	-0.178228	-2.3296	-4.217475	-5.924924	-7.868038
6	4	1000	1	-1	0.821967	0.626588	0.452724	0.23882	0.02945	-0.170898	-0.373327	-0.573083	-0.790301
7	5	1000	0	-10	-1.097385	-1.859676	-2.903928	-3.910642	-4.951476	-5.84582	-6.901395	-7.876949	-8.980254
8	6	2000	100	50	94.896851	89.968262	85.156563	80.122383	74.712364	68.995331	64.460281	59.9118	55.500656
9	7	2000	200	150	194.95697	189.776001	185.21347	180.461746	175.997955	170.708939	165.341965	160.570389	154.79675
10	8	2000	5	-5	3.987701	2.943236	1.83459	0.939207	0.094607	-0.91998	-1.762322	-2.843684	-3.875362
11	9	2000	1000	0	919.855347	811.285767	711.29187	616.528809	518.539978	410.657074	307.513672	218.732254	111.25523
12	10	2000	10000	-10000	8154.11865	5962.40137	3749.26001	1958.25061	-65.004425	-2067.62915	-4008.97241	-6367.5649	-8158.6353

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
#	N	Máx.	Mín.	Deciles								
				1	2	3	4	5	6	7	8	9
11	4000	1	-1	0.804535	0.596692	0.393475	0.186438	-0.01953	-0.217896	-0.418525	-0.614942	-0.80881
12	5000	1	-1	0.807324	0.622742	0.420899	0.213318	0.003113	-0.207837	-0.398346	-0.60188	-0.78893
13	6000	1	-1	0.801733	0.591284	0.408478	0.215284	0.011719	-0.195264	-0.392914	-0.600525	-0.80602
14	7000	1	-1	0.801447	0.589428	0.39812	0.21034	0.01883	-0.176611	-0.373418	-0.588635	-0.79603
15	8000	1	-1	0.794244	0.574572	0.373125	0.181054	-0.01346	-0.209314	-0.417756	-0.613489	-0.80872
16	9000	1	-1	0.788134	0.598364	0.391369	0.196887	0.004486	-0.198206	-0.403644	-0.610034	-0.7993
17	10000	1	-1	0.809009	0.605579	0.3995	0.199927	-0.00626	-0.195911	-0.3995	-0.589148	-0.7928
18	20000	1	-1	0.803204	0.603284	0.395801	0.194348	-0.00665	-0.210401	-0.413617	-0.601782	-0.80193
19	30000	1	-1	0.795099	0.595923	0.403693	0.20365	0.003235	-0.197339	-0.394556	-0.597558	-0.79631
20	50000	1	-1	0.804798	0.605274	0.402918	0.198352	-0.00656	-0.208203	-0.409351	-0.608985	-0.80468

Percentiles

$$P_k = L_i + a_i \cdot \frac{k \cdot \frac{n}{100} - F_{i-1}}{f_i}$$

Wikipedia.org

Código

```
int Percentiles(float *X, int n, FILE *fp)
{
    float *pP;
    int q, i;
    pP=(float *)malloc(99*sizeof(float));
    fprintf(fp, "\nPercentiles\n");
    for(i=0; i<99; i++)
    {
        q=(i+1)*(n+1)/100;
        if (q <= 0)
            pP[i] = X[0] + ((i + 1)*(X[0] - X[0]) / 100));
        else if (q>=n)
            pP[i] = X[n - 1] + ((i + 1)*(X[n - 1] - X[n - 1]) / 100));
        else
            pP[i] = X[q - 1] + ((i + 1)*(X[q] - X[q - 1]) / 100));
        printf("Percentil %d = %f\n", i+1, pP[i]);
        fprintf(fp, "Percentil %d = %f\n", i+1, pP[i]);
    }
    free(pP);
    return 0;
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Tiempo	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	100	1	0		0,965483	0,044088	0,967803	0,002106	0,924344	0,08524	0,947472	0,002106	0,924252	0,002152
2	1000	10	0		9,945245	0,222224	9,990845	0,006104	9,994547	0,04567	9,997575	0,006104	9,992727	0,078638
3	1000	10	-10	1	9,974557	-9,914148	9,979857	-9,948118	9,954757	-9,947583	9,979527	-9,948118	9,92728	-9,924242
4	1000	1	-1	1	0,998851	-0,141449	0,999451	-0,99939	0,975757	-0,955854	0,9975726	-0,99939	0,9972827	-0,9842
5	1000	0	-10	1	-0,554442	-9,141413	-0,003862	-9,997253	-0,057575	-9,947585	-0,07572	-9,997253	-0,025722	-9,92727
6	2000	100	50	1	995488	49,04141	99,9694	49,016758	99,9577	49,0575	99,9757	49,016758	99,5254	49,0272
7	2000	200	150	1	199,95829	139,57575	199,989319	149,00101	199,95757	149,05775	199,95757	149,00101	0,589319	149,007228
8	2000	5	-5	1	4,928213	-4,957553	4,989013	-4,996643	4,985757	-4,97755	4,985757	-4,996643	4,98441	-4,90054
9	2000	1000	0	1	999,828226	0,2575757	999,420166	0,244148	999,488835	0,244148	999,457577	0,244148	999,885277	0,240427
10	2000	10000	-10000	1	9985,82822	-9953,57576	9985,35156	-9953,01236	9985,35576	-9953,01278	9985,35575	-9953,01236	9985,57272	-9953,00489
11	4000	1	-1	2	0,992829	-0,5754	0,999329	-0,8888	0,957575	-0,547414	0,975755	-1	0,992252	-0,42574
12	5000	1	-1	2	0,928285	-0,757577	0,993315	-0,956567	0,957577	-0,954777	0,993888	-0,956567	0,986352	-0,905448
13	6000	1	-1	2	0,968289	-0,972211	0,963219	-0,965617	0,967417	-0,968888	0,944744	-0,965617	0,967437	-0,96455
14	7000	1	-1	3	0,955535	-0,912121	0,956785	-0,996561	0,944744	-0,958355	0,956535	-0,996561	0,935354	-0,990454
15	8000	1	-1	3	0,985877	-0,912121	0,955748	-0,965061	0,975888	-0,986876	0,922674	-0,965061	0,953544	-0,960452
16	9000	1	-1	4	0,758585	-0,912745	0,799863	-0,969912	0,747455	-0,953573	0,793888	-0,969912	0,734534	-0,908782
17	10000	1	-1	6	0,985835	-0,941444	0,998755	-0,965667	0,994324	-0,9683778	0,957757	-0,965667	0,934534	-0,904527
18	20000	1	-1	7	0,958558	-0,942124	0,999868	-0,996547	0,947477	-0,996354	0,995666	-0,996547	0,937833	-0,990578
19	30000	1	-1	8	0,957756	-0,944744	0,996566	-0,998944	0,995855	-0,9931	0,975775	-0,998944	0,987862	-0,900075
20	50000	1	-1	9	0,999577	-0,977774	0,999987	-0,999114	0,958888	-0,947597	0,95233	-0,999114	0,904103	-0,9782004

Momentos centrados de orden 0 a 4

$$m_k = E(X^k) = \sum_i x_i^k p(x_i)$$

Wikipedia.org

Código

```
int Momento_centrado(float aux,float aux2,int count, float *valores,int *cantidad,int c)
{
    float v3=0,v4=0;
    float caux=0,caux1=0;
    int i;
    printf("Momento centrado de Grado 0: 1\n");
    printf("Momento centrado de Grado 1: 0\n");
    printf("Momento centrado de Grado 2: %f\n", (aux2/count)-((aux/count)*(aux/count)));
    //media es aux

    for (i=0;i<=c-1;i++)
    {
        v3=(valores[i]-(aux/count))*(valores[i]-(aux/count))*(valores[i]-(aux/count))*cantidad[i];
        caux=v3+caux;
    }
    printf("Momento centrado de Grado 3: %f\n",caux/count);

    for(i=0;i<=c-1;i++)
    {
        v4=(valores[i]-(aux/count))*(valores[i]-(aux/count))*(valores[i]-(aux/count))*(valores[i]-(aux/count))*cantidad[i];
        caux1=v4+caux1;
    }
    printf("Momento centrado de Grado 4: %f\n",caux1/count);
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Tiempo	Momento 0	Momento 1	Momento 2	Momento 3	Momento 4
1	100	1	0		1	0	0,076446	0,003251	0,011541
2	1000	10	0		1	0	8,365482	0,58121	0,00461
3	1000	10	-10	1	1	0	5,418546	7,19874	8,798466
4	1000	1	-1	1	1	0	0,11354	-0,900041	0,948998
5	1000	0	-10	1	1	0	-0,12548	-4,546876	-0,123789
6	2000	100	50	1	1	0	98,187486	27,54685	94,14686
7	2000	200	150	1	1	0	149,48646	119,0015	148,4687
8	2000	5	-5	1	1	0	3,99874	-3,184668	2,498497
9	2000	1000	0	1	1	0	978,001	0,00185	979,1546
10	2000	10000	-10000	1	1	0	9551,0189	-9017,1248	9188,4966
11	4000	1	-1	2	1	0	0,326758	-0,005752	0,195599
12	5000	1	-1	2	1	0	0,954356	-0,978893	0,931879
13	6000	1	-1	2	1	0	0,973269	-0,972654	0,975555
14	7000	1	-1	3	1	0	0,333569	0,005614	0,1999322
15	8000	1	-1	3	1	0	0,902933	-0,923493	0,923423
16	9000	1	-1	4	1	0	0,799863	-0,969912	0,747455
17	10000	1	-1	6	1	0	0,900003	-0,823202	0,201023
18	20000	1	-1	7	1	0	0,901949	-0,94791	0,92348
19	30000	1	-1	10	1	0	0,923032	-0,999201	0,92214
20	50000	1	-1	15	1	0	0,336334	0,000493	0,200692

Momentos no centrados de orden 0 a 4

$$m_k = E(X^k) = \sum_i x_i^k p(x_i)$$

Wikipedia.org

Código

```
int Momento_descentrado(float *valores,int *cantidad,int c)
{
    float v1,aux=0;
    float v2,aux2=0;
    float v3,aux3=0;
    float v4,aux4=0;

    int count=0,i;

    for (i=0;i<=c-1;i++)
    {
        count=cantidad[i]+count;
    }

    for(i=0;i<=c-1;i++)
    {
        v1=valores[i]*cantidad[i];//MD1
        aux=v1+aux;
        v2=valores[i]*valores[i]*cantidad[i];//MD2
        aux2=v2+aux2;
        v3=valores[i]*valores[i]*valores[i]*cantidad[i];//MD3
        aux3=v3+aux3;
        v4=valores[i]*valores[i]*valores[i]*valores[i]*cantidad[i];//MD4
        aux4=v4+aux4;
    }
    printf("Momento descentrado de Grado 0: 1\n");
    printf("Momento descentrado de Grado 1: %f\n",aux/count);
    printf("Momento descentrado de Grado 2: %f\n",aux2/count);
    printf("Momento descentrado de Grado 3: %f\n",aux3/count);
    printf("Momento descentrado de Grado 4: %f\n",aux4/count);
    Momento_centrado(aux,aux2,count,valores,cantidad,c);
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Tiempo	Momento 0	Momento 1	Momento 2	Momento 3	Momento 4
1	100	1	0		1	0,496289	0,318743	0,228965	0,924544
2	1000	10	0		1	4,892237	5,990538	7,648644	9,158747
3	1000	10	-10	1	1	-0,217279	4,454646	6,464687	9,954788
4	1000	1	-1	1	1	-0,141449	0,999451	-0,99939	0,975757
5	1000	0	-10	1	1	-8,654656	-0,165468	-8,36486	-0,654654
6	2000	100	50	1	1	39,231654	99,15646	29,15166	99,00011
7	2000	200	150	1	1	125,12568	129,17778	159,168945	189,15156
8	2000	5	-5	1	1	-4,00012	4,998564	-4,494989	4,00897
9	2000	1000	0	1	1	0,36987	989,41654	0,184656	998,124886
10	2000	10000	-10000	1	1	-9833,4453	9756,4684	-9977,4336	9977,4682
11	4000	1	-1	2	1	-0,49999	0,888489	-0,48987	0,789879
12	5000	1	-1	2	1	-0,565678	0,954356	-0,978893	0,931879
13	6000	1	-1	2	1	-0,934645	0,973269	-0,972654	0,975555
14	7000	1	-1	3	1	-0,947899	0,901949	-0,94791	0,92348
15	8000	1	-1	3	1	-0,234234	0,956356	-0,34234	0,234234
16	9000	1	-1	4	1	-0,912745	0,799863	-0,969912	0,747455
17	10000	1	-1	6	1	-0,952341	0,900003	-0,823202	0,201023
18	20000	1	-1	7	1	-0,234423	0,902933	-0,923493	0,923423
19	30000	1	-1	10	1	-0,93332	0,923032	-0,999201	0,92214
20	50000	1	-1	15	1	0,001235	0,333634	0,000493	0,220692

Momento Estándar

$$\frac{\mu_k}{\sigma^k}$$

Wikipedia.org

Código

```
int momentoestandar(float desviacion,float media,FILE *fp)
{
    float Momentoestandar;
    Momentoestandar = (media / desviacion);
    printf("\nEl Momento estandar es : %f",Momentoestandar);
    fprintf(fp,"\nEl Momento estandar es: %f",Momentoestandar);
    return 0;
}
```

Tabla de simulaciones

#	N	Maximo	Minimo	Tiempo	Momento E
1	100	1	0	1	1,797997
2	1000	10	0	1	1,740248
3	1000	10	-10	1	0,002285
4	1000	1	-1	1	0,040297
5	1000	0	-10	1	-1,725496
6	2000	100	50	1	5,161657
7	2000	200	150	1	12,165464
8	2000	5	-5	1	-0,001937
9	2000	1000	0	1	1,766729
10	2000	10000	-10000	1	-0,006787
11	4000	1	-1	2	0,008256
12	5000	1	-1	2	-0,017895
13	6000	1	-1	2	-0,000103
14	7000	1	-1	3	0,048977
15	8000	1	-1	3	-0,048896
16	9000	1	-1	4	0,007004
17	10000	1	-1	6	-0,005542
18	20000	1	-1	7	0,002375
19	30000	1	-1	10	-0,010704
20	50000	1	-1	15	0,007004

Conclusión

En este proyecto final utilizamos todo el conocimiento aprendido en clase, los cuales fueron útiles para la realización de este y otros proyectos en cuestión de estadística, se facilita con este programa muchos de los problemas de esta materia, se puede utilizar para otras materias e incluso si le agregamos más cosas se puede vender a una tienda o a una plaza comercial, el programa funciona como se nos fue indicado y el reporte se elaboró como dicen las consideraciones, agradezco a mis compañeros por el trabajo de equipo y la ayuda brindada muchas gracias doctor.

Referencias

[1] <https://sume.ugto.mx/>

[2] <https://es.wikipedia.org>

[3] <https://csacademy.com/>