Alumno: **PAREDES AGUILERA CHRISTIAN LIMBERT.**
C.I.: **6788578 L.P.**
Universidad: **Mayor de San Ándres.**
Carrera: **Matemáticas.**
Asignatura: **Computación Científica II.**
Tarea: **10**
Fecha **22-11-2021**

1. Desigualdades que relacionan distintos tipos de promedios

a) (Tom Apostol, Calculus Vol 1)

Sean $x_1, x_2, ..., x_n$ n números reales positivos. Si p es un entero no nulo, la media de potencias p-énesimas M_p se define como sigue.

$$M_p = \left(\frac{x_1^p + ... + x_n^p}{n} \right)^{1/p}$$

El número M_1 se denomina media aritmética, M_2 media cuadrática y M_{-1} media armónica.

b) Código fuente.

```python
# función media aritmética, media cuadrática y media aritmética.
def promedios(lista):
    sum_arit, sum_cuad, sum_armo = 0,0,0

    ## loop for para sumar los valores de la lista
    for i in lista:
        sum_arit += i
        sum_cuad += i**2
        sum_armo += i**(-1)

    aritmetica = sum_arit / len(lista)
    cuadratica = (sum_cuad / len(lista))**(1/2)
    armonica = (sum_armo / len(lista))**(-1)
    return "Media aritmética: {:.2f} \nMedia cuadrática:{:.2f} \nMedia armónica: {:.2f
        }".format(aritmetica, cuadratica, armonica)

list_ = list(map(float, input("Agregar lista: ").strip().split()))
print(promedios(list_))
```

c) Prueba de la ejecución del programa.

2. Obtención de las estimaciones de mínimos cuadrados ordinarios

a) (George Wooldridge, Introducción a la econometría, capítulo 2).

Sea $\{(x_i, y_i) : i = 1, \ldots, n\}$, una muestra aleatoria de tamaño n tomada de la población. Como estos datos provienen de (2,1) para todo i puede escribirse

$$y_i = \beta_o + \beta_1 x_i + u_i. \tag{1}$$

En tanto el intercepto β_0 aparezca en la ecuación, nada se altera al suponer que el valor promedio de u en la población, es cero. Es decir, $E(u) = 0$.
El supuesto crucial es que el valor promedio de u no depende del valor de x. Este supuesto se expresa como u $E(u\backslash x) = E(u)$. Esta última ecuación indica que el valor promedio de los factores no observables es el mismo en todas las fracciones de la población determinados por los valores de x y que este promedio común es necesariamente igual al promedio al promedio de u en toda la población. Y por lo tanto u es media independiente de x. Combinando la independencia de la media con el supuesto de $E(u) = 0$ se obtiene el supuesto de media condicional cero, $E(u\backslash x) = 0$

En la población, u no está correlacionada con x. Por tanto, se tiene que el valor esperado de u es cero y que la covarianza entre x y y es cero:

$$E(u) = 0 \tag{2}$$

y

$$Cov(x, u) = E(xu) = 0. \tag{3}$$

Covarianza.- Sean $\mu_x = E(X)$ y $\mu_y = E(Y)$ y considere la variable aleatoria $(X - \mu_x)(Y - \mu_y)$. Si X es mayor a su media y Y es mayor a su media, entonces $(X - \mu_x)(Y - \mu_y) > 0$. La covarianza entre dos variables aleatorias X y Y llamada algunas veces covarianza poblacional, para hacer énfasis en que se refiere a la relación entre dos variables que describen una población, está definida como el valor esperado del producto $(X - \mu_x)(Y - \mu_y)$:

$$Cov(X, Y) = E\left[(X - \mu_x)(Y - \mu_y)\right] \tag{4}$$

que también suele denotarse como σ_{XY}.
Algunas expresiones para útiles para calcular $Cov(X, Y)$ son las siguientes

$$Cov(X, Y) = E\left[(X - \mu_x)(Y - \mu_y)\right] = E\left[(X - \mu_X)Y\right] = E\left[X(Y - \mu_Y)\right] = E(XY) - \mu_x\mu_y \tag{5}$$

De donde se sigue que si $E(x) = 0$ o $E(Y) = 0$, entonces $Cov(X, Y) = E(XY)$.

Luego

$$E(y - \beta_0 - \beta_1 x) = 0 \tag{6}$$

y

$$E\left[x(y - \beta_0 - \beta_i x) = 0\right] \tag{7}$$

Como hay que estimar dos parámetros desconocidos, se espera que las dos ecuaciones anteriores puedan servir para obtener buenos estimadores de β_0 y β_1. En efecto estas ecuaciones pueden servir para la estimación de estos parámetros.

Por el método de momentos para la estimación , y las anteriores dos ecuaciones,

$$n^{-1}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0 \tag{8}$$

y

$$n^{-1}\sum_{i=1}^{n}x_i(y_i - \beta_0 - \beta_1 x_i) = 0 \tag{9}$$

Luego por la ecuación (2.9) tenemos que

$$\overline{y} = \hat{\beta}_0 + \hat{\beta}_1\overline{x}. \tag{10}$$

donde $\overline{y} = n^{-1}\sum_{i=1}^{n} y_i$ es el promedio muestral de las y_i, y lo mismo ocurre con $\overline{x}$, así,

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1\overline{x}. \tag{11}$$

Por último empleando (2.10) y (2.12) para sustituir $\hat{\beta}_0$ se obtiene,

$$\sum_{i=1}^{n} x_i\left[y_i - (\overline{y} - \hat{\beta}_1\overline{x}) - \hat{\beta}_1 x_i\right] = 0,$$

de donde, reordenando, tenemos que

$$\sum_{i=1}^{n} x_i(y_i - \overline{y}) = \hat{\beta}_1\sum_{i=1}^{n} x_i(x_i - \overline{x}).$$

en consecuencia por las propiedades de la sumatoria,

$$\sum_{i=1}^{n} x_i(x_i - \overline{x}) = \sum_{i=1}^{n}(x_i - \overline{x})^2 \quad y \quad \sum_{i=1}^{n} x_i(y_i - \overline{y}) = \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

ya que

$$\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) = \sum_{i=1}^{n} x_i y_i - \overline{y}\sum_{i=1}^{n} x_i - \overline{x}\sum_{i=1}^{n} y_i + \overline{yx}\sum_{i=1}^{n} 1,$$

luego ya que $\sum_{i=1}^{n} x_i = n\overline{x}$ entonces

$$\sum_{i=1}^{n} x_i y_i - n\overline{yx} - n\overline{xy} + n\overline{xy} = \sum_{i=1}^{n}(x_i - \overline{x})y_i$$

por lo tanto,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}. \tag{12}$$

Ésta ecuación no es nada mas que la covarianza muestral en x e y dividida entre la variación muestral de x.

b) **Código fuente.**

```
# función de estimación de la mínimos cuadrados ordinarios
def beta01(x,y):
    sumnum, sumden = 0,0
    mediax, mediay = sum(x)/len(x), sum(y)/len(y)
    for i in range(len(x)):
        sumnum += (x[i] - mediax) * (y[i] - mediay)
        sumden += (x[i]-mediax)**2
    beta1 = sumnum / sumden
    beta0 = mediay - beta1*mediax
    return "beta 1: {} \nbeta 0: {}".format(beta1,beta0)

# entrada
x = list(map(float,input("Introducir vector x: ").strip().split()))
y = list(map(float,input("Introducir vector y: ").strip().split()))

# salida
print(beta01(x,y))
```

c) **Prueba de la ejecución del programa**.

```
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 MCO.py
Introducir vector x: 3 1 8 7 3 3
Introducir vector y: 9 7 8 3 1 4
beta 1: 0.0180995475113122
beta 0: 5.257918552036199
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 MCO.py
Introducir vector x: 87 93 63 73 82 74
Introducir vector y: 91 83 72 63 53 33
beta 1: 0.8297266514806378
beta 0: 0.5615034168564819
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 MCO.py
Introducir vector x: 328 732 632 73 632 738
Introducir vector y: 328 732 482 62 173 73
beta 1: 0.42108012191145183
beta 0: 88.31896963459974
```

3. VARIANZA

a) Fórmula

$$Var(x) = \frac{\sum_{i=1}^{n}(x_i - \overline{X})^2}{n}$$

b) Código fuente.

```python
# varianza
def var(vector):
    if len(vector) == 0:
        return 0
    else:
        mean = sum(vector) / len(vector)
        return sum([(x - mean) ** 2 for x in vector]) / len(vector)

vector = list(map(float,input("Introducir vector: ").strip().split()))
print("La varianza es: ", var(vector))
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 var.py
Introducir vector x: 2 3 4 5
La varianza es:  1.25
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 var.py
Introducir vector x: 4 2 6 5 4 1 3 4 2
La varianza es:  2.246913580246914
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 var.py
Introducir vector x: 4 2 8 7 1 3 4 5 1 4 9 8 7 6
La varianza es:  6.494897959183674
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 var.py
Introducir vector x: 3 2 0 9 8 6 83 72 63 73
La varianza es:  1138.8899999999999
```

4. PRODUCTO ESCALAR

a) Definición

Si $A = \{a_1, \ldots a_n\}$ y $B = \{b_1, \ldots, b_n\}$ son dos vectores de V_n su producto escalar se representa con $A \cdot B$ y se define con la igualdad,

$$A \cdot B = \sum_{k=1}^{n} a_k b_k$$

b) Código fuente.

```python
# Producto escalar
def scalar_product(v1, v2):
    return sum(v1[i] * v2[i] for i in range(len(v1)))

# entrada
v1 = list(map(float,input("Introducir vector v1: ").strip().split()))
v2 = list(map(float,input("Introducir vector v2: ").strip().split()))

# salida
print(scalar_product(v1,v2))
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 prod_escalar.py
Introducir vector v1: 3 9 8 2
Introducir vector v2: 3 4 2 9
79.0
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 prod_escalar.py
Introducir vector v1: 98 798 678 89
Introducir vector v2: 89 76 89 67
135675.0
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 prod_escalar.py
Introducir vector v1: 0 98 7 0
Introducir vector v2: 3 9 7 3
931.0
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 prod_escalar.py
Introducir vector v1: 8 9 8 7 6 5 4
Introducir vector v2: 8 7 5 9 1 3 8
283.0
```

5. NORMA DE UN VECTOR

a) Definición

Si A es un vector de V_n, su longitud o norma se designa con $\|A\|$ y se define mediante la igualdad

$$\|A\| = (A \cdot A)^{1/2}$$

b) Código fuente.

```python
# norma de un vector
def norma(v):
    return (v[0]**2 + v[1]**2)**0.5

# entrada
v = list(map(float,input("Introducir vector v: ").strip().split()))

# salida
print(norma(v))
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 norma.py
Introducir vector v: 2 3 4 2 1 4 2
3.605551275463989
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 norma.py
Introducir vector v: 4 2 1 4 2 3
4.47213595499958
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 norma.py
Introducir vector v: 12 43 53 213 43
44.64302857109943
fode@ubuntu:~/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 norma.py
Introducir vector v: 98 88 6 7 6 7 87 63 73 837
131.71180660821565
```