
Alumno: PAREDES AGUILERA CHRISTIAN LIMBERT.
C.I.: 6788578 L.P.
Universidad: Mayor de San Andrés.
Carrera: Matemáticas.
Asignatura: Computación Científica II.
Tarea: 9 / semana 2
Fecha: 10-11-2021

1. Desigualdades que relacionan distintos tipos de promedios

a) (Tom Apostol, Calculus Vol 1)

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

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

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

b) Código fuente.

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

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

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

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

c) Prueba de la ejecución del programa.

```
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 promedios.py
Agregar lista: 2 3 2 42 2 31 31 43 13 81 173 1731 371 137
Media aritmética: 190.14
Media cuadrática:477.72
Media armónica: 6.83
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 promedios.py
Agregar lista: 2 2 2 2 2 2
Media aritmética: 2.00
Media cuadrática:2.00
Media armónica: 2.00
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 promedios.py
Agregar lista: 2 3 4 5 6 7
Media aritmética: 4.50
Media cuadrática:4.81
Media armónica: 3.77
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week3$ python3 promedios.py
Agregar lista: 21 31 32 -34 2 42
Media aritmética: 15.67
Media cuadrática:29.86
Media armónica: 9.91
```

2. Obtención de las estimaciones de mínimos cuadrados ordinarios

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

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

$$y_i = \beta_0 + \beta_1 x_i + u_i. \quad (1)$$

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 $E(u|x) = E(u)$. 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 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, $E(u|x) = 0$

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:

$$E(u) = 0 \quad (2)$$

y

$$Cov(x, u) = E(xu) = 0. \quad (3)$$

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 $(X - \mu_x)(Y - \mu_y)$:

$$Cov(X, Y) = E[(X - \mu_x)(Y - \mu_y)] \quad (4)$$

que también suele denotarse como σ_{XY} .

Algunas expresiones para útiles para calcular $Cov(X, Y)$ son las siguientes

$$Cov(X, Y) = E[(X - \mu_x)(Y - \mu_y)] = E[(X - \mu_x)Y] = E[X(Y - \mu_y)] = E(XY) - \mu_x \mu_y \quad (5)$$

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

Luego

$$E(y - \beta_0 - \beta_1 x) = 0 \quad (6)$$

y

$$E[x(y - \beta_0 - \beta_1 x)] = 0 \quad (7)$$

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,

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

y

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

Luego por la ecuación (2.9) tenemos que

$$\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}. \quad (10)$$

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

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}. \quad (11)$$

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

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

de donde, reordenando, tenemos que

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

en consecuencia por las propiedades de la sumatoria,

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

ya que

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

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

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

por lo tanto,

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

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
    mediac, mediacy = sum(x)/len(x), sum(y)/len(y)
    for i in range(len(x)):
        sumnum += (x[i] - mediac) * (y[i] - mediacy)
        sumden += (x[i]-mediac)**2
    beta1 = sumnum / sumden
    beta0 = mediacy - beta1*mediac
    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 MC0.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 MC0.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 MC0.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. a)

b) Código fuente.

```
# función de coheficiente binomial

def binomial(n, k):
    if k < 0 or k > n:
        return 0
    if k == 0 or k == n:
        return 1
    t = 1
    for i in range(min(k, n - k)):
        t = t * (n - i) // (i + 1)
    return t
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,20))'
231
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,0))'
1
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,4))'
7315
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(2233,34))'
1915992310120937079161553621708885020654666583096334247860059756882159911700
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(6,5))'
6
```

4. a)

b) Código fuente.

```
# función de coheficiente binomial

def binomial(n, k):
    if k < 0 or k > n:
        return 0
    if k == 0 or k == n:
        return 1
    t = 1
    for i in range(min(k, n - k)):
        t = t * (n - i) // (i + 1)
    return t
```

c) Prueba de la ejecución del programa.

```
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,20))'
231
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,0))'
1
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,4))'
7315
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(2233,34))'
1915992310120937079161553621708885020654666583096334247860059756882159911700
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(6,5))'
6
```

5. a)

b) Código fuente.

```
# función de coheficiente binomial

def binomial(n, k):
    if k < 0 or k > n:
        return 0
    if k == 0 or k == n:
        return 1
    t = 1
    for i in range(min(k, n - k)):
        t = t * (n - i) // (i + 1)
    return t
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,20))'
231
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,0))'
1
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,4))'
7315
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(2233,34))'
1915992310120937079161553621708885020654666583096334247860059756882159911700
fode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(6,5))'
6
```