

Problema: Calculador estadístic

Author: Sebastià Vila, Humbert Cuesta

Date: 3-nov-2009

License: Creative Commons By-Nc-Sa

Enunciat

L'objectiu d'aquest exercici és dissenyar i implementar un programa que permeti analitzar mostres estadístiques. A tal efecte seguirem els següents passos:

1. Dissenyeu i implementeu una funció que llegeixi del teclat una seqüència de reals i retorna la llista corresponent. Assumiu que els valors són sempre positius o zero i que el primer negatiu que es llegeix actua de sentinella.
2. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula la suma. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.
3. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula el màxim. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.
4. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula el mínim. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.
5. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula el recorregut. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.
6. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula la mitja. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.
7. Dissenyeu i implementeu una funció tal que donada una llista de reals de mida arbitrària en calcula la variància. Documenteu la funció i afegiu els doctests convenients. Comproveu amb noisetest que es comporta com cal.

A continuació, escriviu un programa que presenti un menú a l'usuari i li permeti executar les següents opcions:

- Llegir dades
- Calcular el recorregut
- Calcular la mitja
- Calcular la variància

Haureu construït el vostre propi calculador estadístic! Com ampliació us proposem que afegiu al calculador les funcionalitats de calcular la moda i la mediana.

Solució

```
# -*- encoding: utf-8 -*-

"""
Aquest mòdul implementa un calculador estadístic. L'objecte principal que
es manipula és una llista de valors reals que representa una mostra. Les
funcions calculen valors estadístics sobre aquesta mostra.
"""

def llegeix_mostra():
    """
    Llegeix una seqüència de reals del teclat acabada en un
    sentinella negatiu. Els els retorna en una llista.
    """
    l=[]
    print("Escriuiu la mostra i acabeu en un valor negatiu")
    entrada=float(input("Valor:"))
    while entrada >= 0:
        l+= [entrada]
        entrada=float(input("Valor:"))
    return l

def suma_mostra(l):
    """
    Retorna la suma dels valors de la mostra l.

    >>> suma_mostra([2.0,3.0,9.5,2.4])
    16.899999999999999
    >>> suma_mostra([9.1,100,25.2,0.1])
    134.39999999999998
    """
    suma=0.0
    for value in l:
        suma += value
    return suma

def maxim_mostra(l):
    """
    Retorna el valor més gran de la mostra.

    >>> maxim_mostra([2.0,3.0,9.5,2.4])
    9.5
    >>> maxim_mostra([9.1,100,25.2,0.1])
    100
    """
    maxim=l[0]
    for value in l:
        if value > maxim:
            maxim = value
    return maxim
```

```

def minim_mostra(l):
    """
    Retorna el valor més petit de la mostra.

    >>> minim_mostra([2.0,3.0,9.5,2.4])
    2.0
    >>> minim_mostra([9.1,100,25.2,0.1])
    0.10000000000000001
    """
    minim=l[0]
    for value in l:
        if value < minim:
            minim = value
    return minim

def recorregut_mostra(l):
    """
    Retorna el recorregut de la mostra l. És a dir la major
    diferència entre valors de la mostra.

    >>> recorregut_mostra([2.0,3.0,9.5,2.4])
    7.5
    >>> recorregut_mostra([9.1,100,25.2,0.1])
    99.900000000000006
    """
    return maxim_mostra(l) - minim_mostra(l)

def mitjana_mostra(l):
    """
    Retorna la mitjana de la mostra l.

    >>> mitjana_mostra([2.0,3.0,9.5,2.4])
    4.2249999999999996
    >>> mitjana_mostra([9.1,100,25.2,0.1])
    33.599999999999994
    >>> mitjana_mostra([2,10])
    6.0
    """
    return suma_mostra(l)/len(l)

def variancia_mostra(l):
    """
    Retorna la variancia de la mostra l.

    >>> variancia_mostra([9.1,100,25.2,0.1])
    1550.5050000000001
    >>> variancia_mostra([600,470,170,430,300])
    21704.0
    """
    suma = 0.0

```

```

m = mitjana_mostra(l)
for v in l:
    suma += (v - m) * (v - m)
return suma/len(l)

def taula_freq_mostra(l):
    """
    Donada una mostra l retorna la taula de freqüències.
    La taula de freqüències es representa com una llista
    de parelles [v,f] on v és un valor i f la freqüència
    d'aparició de v.

    >>> taula_freq_mostra([1,2,3])
    [[1, 1], [2, 1], [3, 1]]
    >>> taula_freq_mostra([1,2,3,1,2])
    [[1, 2], [2, 2], [3, 1]]
    >>> taula_freq_mostra([1,1,2,1,3,1])
    [[1, 4], [2, 1], [3, 1]]
    """
    tf = []
    for v in l:
        # busca v a la taula i actualitza o afegeix
        for e in tf:
            if e[0] == v:
                e[1] += 1
                break
            else:
                tf += [[v,1]]
    return tf

def moda_mostra(l):
    """
    Retorna la moda de la mostra l.

    >>> moda_mostra([100,100,100,0.1])
    100
    >>> moda_mostra([100,100,9.1,9.1])
    100
    >>> moda_mostra([600,470,170,430,430,430])
    430
    """
    t = taula_freq_mostra(l)
    max_f = t[0]
    for ef in t:
        if max_f[1] < ef[1]:
            max_f = ef
    return max_f[0]

def ordena_mostra(l):
    """
    Modifica la mostra i la ordena
    en ordre ascendent. Usa ordenació per selecció.

```

```

>>> l = [100,100,100,0.5]
>>> ordena_mostra(l)
>>> l == [0.5, 100, 100, 100]
True
"""
for i in range(len(l)):
    m = i
    for j in range(i, len(l)):
        if l[j] < l[m]:
            m = j
    aux = l[i]
    l[i] = l[m]
    l[m] = aux

def mediana_mostra(l):
    """
    Retorna la mediana de la mostra.

    >>> mediana_mostra([100,100,100,0.1])
    100.0
    >>> mediana_mostra([89,10.25,105.78,95.0])
    92.0
    >>> mediana_mostra([100,100,9.1,9.1,50])
    50.0
    """
    m = l[:]
    lm = len(m)
    ordena_mostra(m)
    if lm % 2 == 0:
        return (m[lm/2]+m[lm/2-1])/2.0
    else:
        return float(m[lm/2])

def presenta_menu():
    """
    Escriu el menú del calculador estadístic per pantalla.
    Retorna la opció triada (string) per l'usuari que es garanteix
    que està en el conjunt [123456789s].
    """
    v = "."
    while v not in "123456789s":
        print ("OPCIONES DEL CALCULADOR ESTADÍSTIC")
        print ("=====")
        print
        print ("1. Definir la mostra.")
        print ("2. Suma de la mostra.")
        print ("3. Màxim de la mostra.")
        print ("4. Màxim de la mostra.")
        print ("5. Rang de la mostra.")
        print ("6. Mitjana de la msotra.")
        print ("7. Variància de la mostra.")
        print ("8. Moda de la mostra.")

```

```

    print ("9. Mediana de la mostra.")
    print
    v = raw_input("[123456789s]> ")
return v

if __name__=='__main__':
    hi_ha_mostra = False
    while True:
        n = presenta_menu()
        if n == "s":
            break

        if n != "1" and not hi_ha_mostra:
            print "Cal definir una mostra prèviament!"
        elif n == "1":
            l = llegeix_mostra()
            hi_ha_mostra = True
        elif n == "2":
            print "Suma: ", suma_mostra(l)
        elif n == "3":
            print "Màxim: ", maxim_mostra(l)
        elif n == "4":
            print "Mínim: ", minim_mostra(l)
        elif n == "5":
            print "Recorregut: ", recorregut_mostra(l)
        elif n == "6":
            print "Mitjana: ", mitjana_mostra(l)
        elif n == "7":
            print "Variància: ", variancia_mostra(l)
        elif n == "8":
            print "Moda: ", moda_mostra(l)
        elif n == "9":
            print "Mediana: ", mediana_mostra(l)

    print
    raw_input("Pulsa [Ret] per continuar")
    print "\n"*30

```