

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from datetime import datetime
6
7
8  def meanNA(a):
9      """funcion que devuelve la media de una lista, si no hay ningun valor 'NA'"""
10
11     prom = 0
12     for i in range(len(a)):      # suma los valores si son distintos a 'NA'
13         if a[i] == 'NA':
14             return 'NA'
15         prom += a[i]
16     return prom / len(a)        # saca y devuelve el promedio
17
18
19 def meanProm(a):
20     """Devuelve el promedio de a.
21     Si hay 'NA', los valores NA se completan con el promedio de ese sensor para esa ventana.
22     Si todos son NA, el valor es NA"""
23
24     cant_NA = a.count('NA')      # cuenta los elementos 'NA'
25     if cant_NA == len(a):        # chequea si son todos 'NA'
26         return 'NA'
27     elif cant_NA != 0:           # sino, pero hay algun 'NA'
28         while a.count('NA') != 0: # saca todos los 'NA' de la lista, para poder calcular la media
29             a.remove('NA')
30             suma = sum(a)         # saca el promedio de los valores
31             mean = suma / len(a)
32             return ((mean * cant_NA) + suma) / (len(a) + cant_NA) # devuelve el promedio reemplazando los
33                                 # el promedio
34     else:
35         return sum(a) / len(a) # devuelve el promedio
36
37
38 def Mediana(a):
39     """Calcula la mediana de una lista de valores a."""
40
41     a.sort()                      # ordena la lista
42     if len(a) % 2 == 0:          # si la cantidad de elementos es par, devuelve la media de los valores
43         print('mediana_ok')
44         return (int(a[int(len(a)/2)] + a[int((len(a)/2) - 1)])) / 2
45     else:                        # si la cantidad de elementos es impar, devuelve el elemento central
46         print('mediana_ok')
47         return a[int(len(a)/2)]
48
49
50 def meanMed(a):
51     """Devuelve el promedio de a.
52     Si hay 'NA', los valores 'NA' se completaran con la mediana de ese sensor para esa ventana.
53     Si todos son 'NA', el valor es 'NA'."""
54
55     cant_NA = a.count('NA')      # cuenta los elementos 'NA'
56     if cant_NA == len(a):        # chequea si son todos 'NA'
57         return 'NA'
58     elif cant_NA != 0:           # sino, pero hay valores 'NA'
59         while a.count('NA') != 0: # saca todos los 'NA' de la lista, para poder calcular la mediana de
60                                 # valores restantes
61             a.remove('NA')
62             suma = sum(a)
63             mediana = Mediana(a)
64             return ((mediana * cant_NA) + suma) / (len(a) + cant_NA) # devuelve el promedio, reemplazando

```

```

64  los 'NA'
65                                     # por la mediana
66  else:
67      return sum(a) / len(a)      # devuelve el promedio
68
69
70 def listar(a, n):
71     """funcion que arma una lista con los eneeavos elementos de cada elemento de la lista de listas a y
    los transforma a
72     tipo de datos float.
73     ej:
74     a = [['1','2','3'],['4','5','6'],['7','8','9']]
75     listar(a,0) = [1.0, 4.0, 7.0]
76     listar(a,2) = [3.0, 6.0, 9.0]"""
77
78     lista_nueva = []
79     for i in range(len(a)):
80         lista_nueva.append(a[i][n])
81     return lista_nueva
82
83
84 def float_NA(a):
85     """funcion que convierte los numeros (que estan como string) a float y deja los NA como string"""
86
87     for i in range(len(a)):
88         if a[i] != 'NA':
89             a[i] = float(a[i])
90     return a
91
92
93 def listaStr(a):
94     """funcion que redondea a la segunda cifra significativa los elementos float de una lista y los
    convierte en str."""
95
96     for i in range(len(a)):
97         if isinstance(a[i], float):
98             a[i] = round(a[i], 2)
99             a[i] = str(a[i])
100     return a
101
102
103 def main(argumentos):
104     arch_entrada = argumentos[0]      # asigno los argumentos a variables
105     arch_salida = argumentos[1]
106     tam_ventana = int(argumentos[2])
107
108     if len(argumentos) > 3:          # metodo indicado o deefault
109         metodo = argumentos[3]
110     else:
111         metodo = "def"
112
113     print('I:', arch_entrada, 'O:', arch_salida, 'L:', tam_ventana, 'M:', metodo)      # Prueba para
    ver los
114                                     #
    parámetros que llegaron
115
116     # Abre los archivos de entrada (en modo R:Read) y el de salida (en modo W:Write)
117     with open(arch_entrada, 'r') as entrada, open(arch_salida, 'w') as salida:
118
119         lineasDeEntrada = []      # Aquí nos vamos a guardar toda la info del archivo de entrada
120
121         for linea in entrada:
122             linea = linea.strip('\n')      # Elimina el salto de línea del final
123             camposDeLinea = linea.split(',')      # Se parte la cadena de la línea entera y se genera una
    lista
124             # Se agrega la lista de campos de la línea a la lista de líneas completa
125             lineasDeEntrada.append(camposDeLinea)

```

```

126
127     lineasDeSalida = []
128
129     for vent_i in range(len(lineasDeEntrada) - tam_ventana + 1):      # ciclo que recorre las
ventanas
130
131         # TRATAMIENTO DEL TIEMPO
132
133         # crea una lista de entradas con los datos de la ventana
134         entrada_vent = lineasDeEntrada[vent_i:(vent_i+tam_ventana)]
135         # guardo el primer tiempo de la ventana
136         tiempo_i = datetime.strptime(entrada_vent[0][0], '%Y-%m-%dT%H:%M:%S')
137         # guardo el ultimo tiempo de la ventana
138         tiempo_f = datetime.strptime(entrada_vent[tam_ventana - 1][0], '%Y-%m-%dT%H:%M:%S')
139         # saco la diferencia de tiempos
140         delta = tiempo_f - tiempo_i
141         # agrega a la listaDeSalidas, una lista con el delta-t de la ventana
142         lineasDeSalida.append([float(delta.seconds)])
143
144         # TRATAMIENTO DE LAS TEMPERATURAS
145
146         # armo listas por cada sensor, del tamaño de la ventana para sacar los promedios
147         for i in range(1, len(entrada_vent[0])):      # recorre el recorte de la ventana, cada
iteracion contempla
148
149             sensor_i = listar(entrada_vent, i)      # 1(un) sensor
150             sensor_i = float_NA(sensor_i)      # hago una lista con los valores del sensor en
la ventana
151             sensor_i = float_NA(sensor_i)      # paso los numeros a float, y los NA los deja
como str
152
153             # agrega a la lista de la ventana el promedio de temperaturas según el metodo elegido
154             if metodo == 'def':
155                 lineasDeSalida[vent_i].append(meanNA(sensor_i))
156             elif metodo == 'prom':
157                 lineasDeSalida[vent_i].append(meanProm(sensor_i))
158             elif metodo == 'med':
159                 lineasDeSalida[vent_i].append(meanMed(sensor_i))
160             else:
161                 return print("Metodo de promedio no valido. Metodos validos: def prom med")
162
163         for i in range(len(lineasDeSalida)):      # transforma los tipos de dato a str y redondea a la
segunda cifra
164
165             listaStr(lineasDeSalida[i])      # significativa
166
167         for lineaPorCampos in lineasDeSalida:
168             print(",".join(lineaPorCampos[:]), file=salida)      # Guarda en un archivo los campos
originales
169
170 # Sólo si el programa es ejecutado (esto es, no se usa con 'import') se ejecutará lo de abajo
171 if __name__ == "__main__":
172
173     if len(sys.argv) < 4:
174         print("Se esperaban más argumentos:\n taller3.py arch_entrada arch_salida tam_ventana [
metodo_na]")
175         sys.exit(1)
176
177     main(sys.argv[1:])
178

```