```python
#!/usr/bin/env python3

import numpy                as np
import matplotlib.pyplot    as plt
import pandas               as pd
import allantools

def read_data(fname : str):
    df = pd.read_excel(fname, 'Sheet1', parse_datas=['Timet(s)', 'Deviation(Hz)'])
    #  print(df.head())
    var1 = df['Timet(s)'].tolist()
    var2 = df['Deviation(Hz)'].tolist()
    return var1, var2

def partial_average(fract_data: list,
                    step      : int) :
    nparts = (len(fract_data) - 1) // step
    out    = []
    for i in range(nparts) :
        partial_sum = .0
        for j in range(i, i+step + 1):
            partial_sum += fract_data[j]
        out.append(partial_sum / step )
    return out

def calc_deviation(pavgs : list) :
    sum_of_dy2 = .0
    for i in range(len(pavgs) - 1):
        sum_of_dy2 += (pavgs[i+1] - pavgs[i]) ** 2
    return np.sqrt(sum_of_dy2 / (2*len(pavgs) - 2))

def main() :
    #  f_0 = 1
    f_0 = 3e8 / 730e-9 # the frequency of 730nm light
    timet, fract_data = read_data(r'./Ca_clock_transition_data.xls')
    timet.pop()
    fract_data.pop()

    y = []
    x = timet[:len(fract_data)//2]

    for i in range(len(fract_data) // 2):
        pavgs = partial_average(fract_data, i + 1)
        devia = calc_deviation (pavgs)
        y.append(devia / f_0)
```

```
46
47      plt.loglog(x, y)
48      plt.xlabel('$\\tau / sec$')
49      plt.ylabel('$\delta f / f$')
50      plt.savefig('ad.png', dpi=400)
51      #  plt.show()
52
53  def main_use_allantools() :
54      f_0 = 3e8 / 730e-9 # the frequency of 730nm light
55      timet, fract_data = read_data(r'./Ca_clock_transition_data.xls')
56      timet.pop()
57      fract_data.pop()
58
59      for i, elem in enumerate(fract_data) :
60          fract_data[i] = elem / f_0
61      (t2, ad, ade, adn) = allantools.oadev(fract_data, rate=11.2,
62              data_type="freq", taus = [i for i in range(1, len(timet) +
    1)])
63      plt.loglog(t2, ad)
64      plt.savefig('ad.png', dpi=400)
65      #  plt.show()
66
67  def debug_partial_average(step: int) :
68      vec = [i for i in range(1, 21)]
69      pavg = partial_average(vec, step)
70      print('step = {}, vec.size() = {}, pavg.size = {}'
71              .format(step, len(vec), len(pavg)))
72      print(vec)
73      print(pavg, '\n')
74
75  def debug_deviation() :
76      vec = [i for i in range(1, 21)]
77      devia = calc_deviation(vec)
78      print(vec)
79      print(devia)
80
81  if __name__ == "__main__" :
82      #  for i in range(1, 10):
83          #  debug_partial_average(i)
84      #  debug_deviation()
85      main()
86      #  main_use_allantools()
```

链接:https://pan.baidu.com/s/19NBb8A5YvqXYeh0UOqgDxw 密码:qv82