UBS - Code Assessment

Exercise 1 - Longest run of heads in a sequence of coin tosses

In [4]:

```python
import numpy as np
import random
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator
from matplotlib import pyplot as plt
```

1.A) Write an effcient function which generates random sequences of heads and tails for any value of n

In [4]:

```python
def coinToss (n):
    sequence=[]
    for i in range (n):
        toss=random.randint(0,1)  # We generate random numbers between 0 and 1
        if toss==0:
            sequence.append("T")  # If number is 0, we call it Tails and add it to the
        if toss==1:
            sequence.append("H")  # If number is 1, we call it Heads and add it to the
    return sequence

random_toss = coinToss (10)
print (random_toss)
```

```
['T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'T']
```

1.B) Write an efficient function which computes the length of the longest run of heads in an arbitrary sequence of heads and tails.

In [8]:

```python
def headRun (random_toss):
    current_headRun=0
    max_headRun=0
    for i in random_toss: # We iterate through every element of the random toss seque
        if i=='H':
            current_headRun+=1  # Every H found is added to counter of current headRun
            if current_headRun>max_headRun:   # If the current head run larger than ma
                max_headRun=current_headRun   # we set a new max
        else:
            current_headRun=0 # If no H is found, current head run resets back to 0
    return max_headRun

random_toss = coinToss (1000)
max_headRun = headRun (random_toss)
#print (random_toss)
print (max_headRun)
```

```
9
```

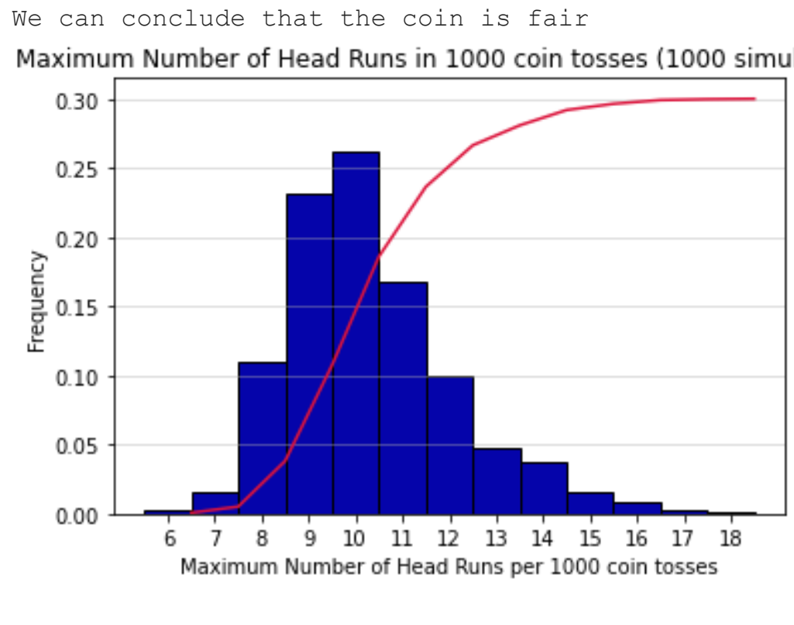
1.C) For a sequence of length n = 1000, we have observed a longest run of heads equal to M = 6.

Basing your decision on this piece of information, do you believe that the coin is fair or not? Please elaborate on your analysis and results.

In [33]:

```python
# Monte-Carlo Simulation
n_simulations=1000
max_headRun_list = []
for i in range(n_simulations):
    random_toss = coinToss (1000) # We generate 1000 random coin tosses
    max_headRun = headRun (random_toss) # We count the maximum number of run-heads
    max_headRun_list.append(max_headRun) # We add the max_headRun to a list

# Histogram - We create a histogram with all the head-runs saved in the previous list
n_bins=13
x, y, patches = plt.hist(max_headRun_list, density=True,
                         bins=n_bins,color='#0504aa',edgecolor="black")
plt.xticks([(patch._x0 + patch._x1)/2 for patch in patches],[i for i in range(6,19)])
plt.grid(axis='y', alpha=0.5)
plt.ylabel('Frequency')
plt.xlabel('Maximum Number of Head Runs per 1000 coin tosses')
plt.title('Maximum Number of Head Runs in 1000 coin tosses (1000 simulations)\n')

# Cumulative Distribution Function
dx = y[1] - y[0]
F1 = np.cumsum(x)*dx*0.3
plt.plot(y[1:], F1,color='#DC143C')

# Hypothesis Testing I - Monte-Carlo Approach
confidenceInterval = 0.99
alpha=round(1-confidenceInterval,3)
p_value=0

# WIn this loop, we calculate the cumulative frequency of 6 (sum of all the frequenci
for i in range (min(max_headRun_list),7):
    p_value += round(max_headRun_list.count(i)/n_simulations,3)   # For a lower-tailed

print ("Hypothesis: \n H0: mu = 6 \n H1: mu < 6")
print ("Significance Level: ",alpha," (99% Confidence Interval)")
print ("p-value (Cumulative Frequency of 6): ",p_value)
if p_value < alpha:
    print ("H0 can be rejected: ",p_value," < ",alpha)
    print ("We can conclude that the coin IS NOT fair")
else:
    print ("H0 CANNOT be rejected: ",p_value," > ",alpha)
    print ("")
    print ("We can conclude that the coin is fair")
```

```
Hypothesis:
 H0: mu = 6
 H1: mu < 6
Significance Level:  0.01  (99% Confidence Interval)
p-value (Cumulative Frequency of 6):  0.017
H0 CANNOT be rejected:  0.017  >  0.01

We can conclude that the coin is fair
```



Exercise 2 - Outlier detection and statistics

2.A) Load the data of the csv file into a data format of your choice

In [5]:

```python
import pandas as pd

df=pd.read_csv('/Users/alvarosanchezfernandez/Documents/Programming/Raw Data/data.csv
df_list = df.values.tolist()

GDP=df_list[0]
GDP=GDP[1:len(GDP)]
GDP = [x for x in GDP if pd.isnull(x) == False]
GDP_vect=[]
GDP_dx=(max(GDP)-min(GDP))/(len(GDP))
for i in range(len(GDP)):
    GDP_vect.append([GDP[i],i*GDP_dx])
print (GDP_vect)

LIBOR=df_list[1]
LIBOR=LIBOR[1:len(LIBOR)]
LIBOR_vect=[]
LIBOR_dx=(max(LIBOR)-min(LIBOR))/(len(LIBOR))
for i in range(len(LIBOR)):
    LIBOR_vect.append([LIBOR[i],i*LIBOR_dx])
print (LIBOR_vect)

Eq1=df_list[2]
Eq1=Eq1[1:len(Eq1)]
Eq1_vect=[]
Eq1_dx=(max(Eq1)-min(Eq1))/(len(Eq1))
for i in range(len(Eq1)):
    Eq1_vect.append([Eq1[i],i*Eq1_dx])
print (Eq1_vect)

Eq2=df_list[3]
Eq2=Eq2[1:len(Eq2)]
Eq2 = [x for x in Eq2 if pd.isnull(x) == False]
Eq2_vect=[]
Eq2_dx=(max(Eq2)-min(Eq2))/(len(Eq2))
for i in range(len(Eq2)):
    Eq2_vect.append([Eq2[i],i*Eq2_dx])
print (Eq2_vect)

Eq3=df_list[4]
Eq3=Eq3[1:len(Eq3)]
Eq3 = [x for x in Eq3 if pd.isnull(x) == False]
Eq3_vect=[]
Eq3_dx=(max(Eq3)-min(Eq3))/(len(Eq3))
for i in range(len(Eq3)):
    Eq3_vect.append([Eq3[i],i*Eq3_dx])
print (Eq3_vect)
```

```
[[179700.0, 0.0], [177070.0, 1547.6404494382023], [174104.0, 3095.2808988764045], [175
146.0, 4642.921348314607], [177800.0, 6190.561797752809], [177186.0, 7738.20224719101
1], [184197.0, 9285.842696629214], [182218.0, 10833.483146067416], [183758.0, 12381.12
3595505618], [185535.0, 13928.76404494382], [187641.0, 15476.404494382023], [206405.1,
17024.044943820227], [185328.0, 18571.685393258427], [188247.0, 20119.325842696628],
[188612.0, 21666.96629213483], [186597.0, 23214.606741573036], [186273.0, 24762.247191
011236], [187357.0, 26309.887640449437], [187976.0, 27857.52808988764], [190720.0, 294
05.168539325845], [193906.0, 30952.808988764045], [194809.0, 32500.449438202246], [196
983.0, 34048.08988764045], [199495.0, 35595.730337078654], [200246.0, 37143.3707865168
54], [199385.0, 38691.011235955055], [198118.0, 40238.651685393255], [198444.0, 41786.
29213483146], [200998.0, 43333.93258426966], [204045.0, 44881.573033707864], [206394.
0, 46429.21348314607], [208764.0, 47976.85393258427], [206300.0, 49524.49438202247],
[205586.0, 51072.13483146067], [207696.0, 52619.77528089887], [210205.0, 54167.4157303
3708], [214120.0, 55715.05617977528], [218430.0, 57262.69662921348], [221421.0, 58810.
33707865169], [223497.0, 60357.97752808989], [222841.0, 61905.61797752809], [228028.0,
63453.25842696629], [226300.0, 65000.89887640449], [228504.0, 66548.53932584269], [232
130.0, 68096.1797752809], [233174.0, 69643.82022471911], [236571.0, 71191.4606741573
1], [239462.0, 72739.10112359551], [244190.0, 74286.74157303371], [248401.0, 75834.382
02247191], [250530.0, 77382.02247191011], [255166.0, 78929.66292134831], [259355.0, 80
477.30337078651], [264256.0, 82024.94382022473], [267935.0, 83572.58426966293], [25764
9.0, 85120.22471910113], [253178.0, 86667.86516853933], [258423.0, 88215.50561797753],
[264910.0, 89763.14606741573], [271309.0, 91310.78651685393], [276413.0, 92858.4269662
9214], [280423.0, 94406.06741573034], [283267.0, 95953.70786516854], [286927.0, 97501.
34831460674], [290526.0, 99048.98876404495], [293493.0, 100596.62921348315], [293488.
0, 102144.26966292135], [294548.0, 103691.91011235955], [294932.0, 105239.5505617977
5], [296587.0, 106787.19101123596], [300947.0, 108334.83146067416], [302091.0, 109882.
47191011236], [302142.0, 111430.11235955056], [308781.0, 112977.75280898876], [309654.
0, 114525.39325842696], [309869.0, 116073.03370786516], [311844.0, 117620.6741573033
8], [308009.0, 119168.31460674158], [308068.0, 120715.95505617978], [308793.0, 122263.
59550561798], [305796.0, 123811.23595505618], [299015.0, 125358.87640449438], [294431.
0, 126906.51685393258], [290892.0, 128454.15730337078], [289127.0, 130001.7977528089
8], [288202.0, 131549.4382022472], [286118.0, 133097.07865168538], [283665.0, 134644.7
191011236], [284722.938, 136192.3595505618]]
[[3.8125, 0.0], [3.625, 0.017664794007490638], [3.5, 0.035329588014981275], [3.4375,
0.052994382022471914], [3.25, 0.07065917602996255], [3.125, 0.08832397003745318], [2.62
5, 0.10598876404494383], [2.875, 0.12365355805243447], [2.46875, 0.1413183520599251],
[2.0, 0.15898314606741573], [2.3125, 0.17664794007490636], [1.8125, 0.1943127340823970
2], [1.76563, 0.21197752808988765], [1.57813, 0.22964232209737828], [1.76563, 0.247307
11610486894], [1.875, 0.26497191011235954], [2.70313, 0.2826367041198502], [2.5, 0.300
30149812734086], [2.5, 0.31796629213483146], [2.11719, 0.3356310861423221], [1.39063,
0.3532958801498127], [1.67969, 0.3709606741573034], [2.125, 0.38862546816479404], [1.8
125, 0.40629026217228464], [1.71094, 0.4239550561797753], [1.75, 0.44161985018726596],
[1.875, 0.45928464419475656], [1.73438, 0.4769494382022472], [0.98828, 0.4946142322097
379], [1.375, 0.5122790262172285], [1.51563, 0.5299438202247191], [1.25391, 0.54760861
42322097], [1.38672, 0.5652734082397004], [1.75, 0.5829382022471911], [1.875, 0.600602
9962546817], [1.25, 0.6182677902621723], [1.0625, 0.6359325842696629], [0.875, 0.65359
73782771536], [1.4375, 0.6712621722846442], [1.47656, 0.6889269662921349], [1.6875, 0.
7065917602996254], [2.16406, 0.7242565543071161], [1.84375, 0.7419213483146068], [1.57
031, 0.7595861423220974], [1.375, 0.7772509363295881], [1.0, 0.7949157303370787], [1.8
1281, 0.8125805243445693], [1.25, 0.83024531835206], [1.19458, 0.8479101123595506],
[1.24333, 0.8655749063670413], [1.19125, 0.8832397003745319], [0.96167, 0.900904494382
0225], [1.0, 0.9185692883895131], [1.17833, 0.9362340823970038], [1.02167, 0.953898876
4044944], [0.96333, 0.9715636704119851], [1.19167, 0.9892284644194758], [1.17333, 1.00
68932584269663], [2.04333, 1.024558052434457], [1.58833, 1.0422228464419476], [2.0683
3, 1.0598876404494382], [2.2, 1.077552434456929], [2.575, 1.0952172284644195], [3.1,
1.1128820224719103], [2.78333, 1.1305468164794008], [3.23667, 1.1482116104868914], [3.
21, 1.1658764044943821], [3.265, 1.1835411985018727], [3.28167, 1.2012059925093634],
[3.21667, 1.218870786516854], [3.45, 1.2365355805243445], [3.36667, 1.254200374531835
3], [3.45833, 1.2718651685393259], [3.43, 1.2895299625468166], [3.365, 1.3071947565543
072], [3.17667, 1.3248595505617977], [3.28, 1.3425243445692885], [3.28333, 1.360189138
576779], [3.25333, 1.3778539325842698], [3.33833, 1.3955187265917604], [2.24667, 1.413
183520599251], [2.14167, 1.4308483146067417], [2.10333, 1.4485131086142322], [1.78833,
1.466177902621723], [1.61333, 1.4838426966292135], [1.705, 1.5015074906367043], [1.51,
1.5191722846441948], [1.52, 1.5368370786516854], [1.17333, 1.5545018726591762], [1.161
67, 1.5721666666666667], [0.76667, 1.5898314606741575], [0.68167, 1.607496254681648],
[0.71667, 1.6251610486891386], [0.675, 1.6428258426966293], [0.73833, 1.6604906367041
2], [0.60667, 1.6781554307116107], [0.59167, 1.6958202247191012], [0.58167, 1.71348501
87265918], [0.295, 1.7311498127340825], [0.26667, 1.748814606741573], [0.24333, 1.7664
794007490638], [0.26333, 1.7841441947565544], [0.23833, 1.801808988764045], [0.24333,
1.8194737827715357], [0.2, 1.8371385767790263], [0.20167, 1.854803370786517], [0.245,
1.8724681647940076], [0.22, 1.8901329588014981], [0.2, 1.907797752808989], [0.22833,
1.9254625468164794], [0.21667, 1.9431273408239702], [0.21833, 1.9607921348314608], [0.
2, 1.9784569288389515], [0.4, 1.996121722846442], [0.40333, 2.0137865168539326], [0.4
2, 2.0314513108614234], [0.595, 2.049116104868914], [0.645, 2.0667808988764045], [0.6
8, 2.0844456928838953], [0.66, 2.102110486891386], [0.71, 2.1197752808988763], [0.7183
3, 2.137440074906367], [0.75, 2.155104868913858], [0.73833, 2.172769662921348], [0.73,
2.190434456928839], [0.72667, 2.2080992509363297], [0.73, 2.2257640449438205], [0.73,
2.243428838951311], [0.74, 2.2610936329588016], [0.75, 2.2787584269662924], [0.91167,
2.2964232209737827], [0.89167, 2.314088014981273], [0.89, 2.3317528089887642], [1.03,
2.349417602996255], [1.14083, 2.3670823970037453], [1.26, 2.384747191011236], [1.31,
2.402411985018727], [1.42167, 2.420076779026217], [1.42, 2.437741573033708], [1.55, 2.
4554063670411987], [1.72083, 2.473071161048689], [1.73, 2.49073595505618], [1.88, 2.50
84007490636706], [2.04583, 2.5260655430711614], [2.08083, 2.5437303370786517], [2.12,
2.5613951310861425], [2.21833, 2.5790599250936332], [2.20667, 2.5967247191011236], [2.
35417, 2.6143895131086143], [2.615, 2.632054307116105], [2.56167, 2.6497191011235954],
[2.7375, 2.667383895131086], [2.53167, 2.685048689138577], [2.28333, 2.702713483146067
7], [2.665, 2.720378277153558], [2.42833, 2.738043071161049], [2.45833, 2.755707865168
5396], [2.61, 2.77337265917603], [2.71833, 2.7910374531835207], [2.40583, 2.8087022471
910115], [2.36, 2.826367041198502], [2.35667, 2.8440318352059926], [2.2725, 2.86169662
92134833], [2.255, 2.879361423220974], [2.8, 2.8970262172284644], [1.855, 2.9146910112
35955], [0.88, 2.932355805243446], [0.34, 2.9500205992509363], [0.31833, 2.96768539325
8427], [0.30167, 2.985350187265918], [0.23167, 3.0030149812734086], [0.21, 3.020679775
280899], [0.20167, 3.0383445692883897], [0.195, 3.0560093632958805], [0.16, 3.07367415
7303371], [0.12, 3.0913389513108616], [0.10083, 3.1090037453183523], [0.09333, 3.12666
85393258427], [0.11, 3.1443333333333334], [0.10667, 3.161998127340824], [0.1, 3.179662
921348315], [0.09167, 3.1973277153558053], [0.08583, 3.214992509363296], [0.0007, 3.23
2657303370787], [0.03667, 3.250322097378277], [0.06417, 3.267986891385768], [0.13167,
3.2856516853932587], [0.12667, 3.303316479400749], [0.1425, 3.32098127340824], [0.1316
7, 3.3386460674157306], [0.14333, 3.3563108614232213], [0.1425, 3.3739756554307117],
[0.13833, 3.3916404494382024], [0.13667, 3.409305243445693], [0.13833, 3.4269700374531
835], [0.14, 3.4446348314606743], [0.12667, 3.462299625468165], [0.12833, 3.4799644194
75656], [0.12583, 3.497629213483146], [0.0, 3.515294007490637], [0.00667, 3.5329588014
981277], [0.02333, 3.550623595505618], [0.03167, 3.568288389513109], [0.03167, 3.58595
31835205996], [0.04583, 3.60361797752809], [0.06083, 3.6212827715355806], [0.07667, 3.
6389475655430714], [0.075, 3.656612359550562], [0.04417, 3.6742771535580525], [0.038,
3.6919419475655433], [0.02, 3.709606741573034], [0.01, 3.7272715355805244], [0.009, 3.
744936329588015], [0.006, 3.762601123595506], [0.005, 3.7802659176029962], [-0.008, 3.
797930711610487], [0.003, 3.815595505617978], [-0.001, 3.8332602996254685], [-0.001,
3.850925093632959], [-0.001, 3.8685898876404496], [-0.001, 3.8862546816479404], [-0.00
3, 3.9039194756554307], [-0.003, 3.9215842696629215], [-0.003, 3.9392490636704123], [-
0.003, 3.956913857677903], [-0.005, 3.9745786516853934], [-0.007, 3.992243445692884],
[-0.005, 4.0099082397003745], [-0.006, 4.027573033707865], [-0.007, 4.04523782771535
6], [-0.007, 4.062902621722847], [-0.007, 4.0805674157303375], [-0.008, 4.098232209737
828], [-0.002, 4.115897003745318], [0.001, 4.133561797752809], [0.002, 4.151226591760
3], [0.0, 4.1688913857677905], [0.002, 4.186556179775281], [-0.003, 4.20422097378277
2], [-0.038, 4.221885767790262], [-0.904, 4.239550561797753], [-0.873, 4.2572153558052
435], [-0.851, 4.274880149812734], [-0.815, 4.292544943820225], [-0.821, 4.31020973782
7716], [-0.832, 4.3278745318352065], [-0.787, 4.345539325842696], [-0.781, 4.363204119
850187], [-0.792, 4.380868913857678], [-0.774, 4.398533707865169], [-0.848, 4.41619850
18726595], [-0.798, 4.43386329588015], [-0.789, 4.451528089887641], [-0.8332, 4.469192
883895131], [-0.7826, 4.486857677902622], [-0.779, 4.5045224719101125], [-0.7746, 4.52
2187265917603], [-0.8286, 4.539852059925094], [-0.7976, 4.557516853932585], [-0.8108,
4.5751816479400755], [-0.802, 4.592846441947565], [-0.7854, 4.610511235955056], [-0.81
78, 4.628176029962547], [-0.799, 4.645840823970038], [-0.7884, 4.6635056179775285], [-
0.8008, 4.681170411985019], [-0.7792, 4.69883520599251]]
[[3017.3, 0.0], [3041.2, 16.350561797752807], [3143.1, 32.701123595505614], [3220.4, 4
9.051685393258424], [3340.6, 65.40224719101123], [3323.7, 81.75280898876403], [3449.9,
98.10337078651685], [3509.4, 114.45393258426965], [3520.2, 130.80449438202245], [3518.
7, 147.15505617977527], [3680.4, 163.50561797752806], [3689.3, 179.85617977528088], [3
752.8, 196.2067415730337], [3752.7, 212.55730337078649], [3718.4, 228.9078651685393],
[3806.0, 245.2584269662921], [3739.2, 261.6089887640449], [3725.6, 277.9595505617977],
[3734.4, 294.31011235955054], [3884.4, 310.66067415730333], [3992.2, 327.011235955056
1], [3948.5, 343.36179775280897], [4038.5, 359.71235955056176], [4118.5, 376.062921348
31455], [4257.8, 392.4134831460674], [4307.1, 408.7640449438202], [4248.1, 425.1146067
4157297], [4445.0, 441.46516853932576], [4562.8, 457.8157303370786], [4728.3, 474.1662
921348314], [4899.3, 490.5168539325842], [4870.2, 506.86741573033703], [5317.1, 523.21
79775280898], [4906.4, 539.5685393258426], [4921.8, 555.9191011235954], [5135.5, 572.2
696629213482], [5599.0, 588.6202247191011], [5820.6, 604.9707865168539], [6017.6, 621.
3213483146067], [6010.3, 637.6719101123595], [5837.9, 654.0224719101122], [5919.9, 67
0.373033707865], [5809.7, 686.7235955056179], [5169.1, 703.0741573033707], [4908.2, 71
9.4247191011235], [5525.5, 735.7752808988763], [5537.4, 752.1258426966291], [5882.6, 7
68.4764044943819], [6012.4, 784.8269662921348], [6060.9, 801.1775280898876], [6330.02,
817.5280898876404], [6552.18, 833.8786516853932], [6250.03, 850.2292134831459], [6488.
9, 866.5797752808987], [6288.29, 882.9303370786515], [6276.18, 899.2808988764044], [59
70.73, 915.6314606741572], [6283.97, 931.98202247191], [6645.97, 948.3325842696628],
[6930.2, 964.6831460674156], [6290.93, 981.0337078651684], [6364.89, 997.384269662921
3], [6462.12, 1013.7348314606741], [6327.43, 1030.0853932584268], [6470.54, 1046.43595
50561796], [6470.43, 1062.7865168539324], [6379.35, 1079.1370786516852], [6795.01, 109
5.487640449438], [6284.46, 1111.8382022471908], [6457.61, 1128.1887640449436], [6170.4
2, 1144.5393258426964], [6222.46, 1160.8898876404494], [6251.83, 1177.2404494382022],
[5908.59, 1193.591011235955], [5618.47, 1209.9415730337078], [5928.02, 1226.2921348314
605], [5809.6, 1242.6426966292133], [5716.68, 1258.9932584269661], [5546.93, 1275.3438
20224719], [5312.12, 1291.6943820224717], [4785.63, 1308.0449438202245], [5071.23, 132
4.3955056179773], [5164.64, 1340.74606741573], [5217.35, 1357.0966292134829], [5189.6
8, 1373.4471910112359], [5169.02, 1389.7977528089887], [5271.76, 1406.1483146067415],
[5125.51, 1422.4988764044942], [5085.07, 1438.849438202247], [4685.76, 1455.1999999999
998], [4044.52, 1471.5505617977526], [4180.88, 1487.9011235955054], [3797.4, 1504.2516
853932582], [3996.98, 1520.602247191011], [4154.27, 1536.9528089887638], [3940.36, 155
3.3033707865166], [3689.36, 1569.6539325842696], [3684.67, 1586.0044943820224], [3684.
78, 1602.3550561797751], [3880.13, 1618.705617977528], [4129.28, 1635.0561797752807],
[3963.92, 1651.4067415730335], [4098.43, 1667.7573033707863], [4204.43, 1684.107865168
539], [4169.19, 1700.4584269662919], [4332.57, 1716.8089887640447], [4410.03, 1733.159
5505617975], [4476.87, 1749.5101123595502], [4381.37, 1765.860674157303], [4537.0, 178
2.211235955056], [4410.71, 1798.5617977528088], [4489.69, 1814.9123595505616], [4422.6
8, 1831.2629213483144], [4424.72, 1847.6134831460672], [4415.73, 1863.96404494382], [4
502.05, 1880.3146067415728], [4659.65, 1896.6651685393256], [4673.79, 1913.01573033707
84], [4735.68, 1929.3662921348312], [4814.3, 1945.716853932584], [4906.21, 1962.067415
7303367], [5000.47, 1978.4179775280895], [4914.03, 1994.7685393258425], [4801.68, 201
1.1191011235953], [5011.04, 2027.4696629213481], [5161.01, 2043.820224719101], [5290.8
5, 2060.1707865168537], [5328.51, 2076.5213483146063], [5501.55, 2092.8719101123593],
[5344.33, 2109.2224719101123], [5486.12, 2125.573033707865], [5618.76, 2141.9235955056
18], [5801.63, 2158.2741573033704], [5844.07, 2174.6247191011234], [6024.28, 2190.9752
80898876], [6023.14, 2207.325842696629], [5749.69, 2223.6764044943816], [5884.44, 224
0.0269662921346], [5880.84, 2256.377528089887], [5949.15, 2272.72808988764], [5957.83,
2289.0786516853927], [6149.62, 2305.4292134831458], [6021.54, 2321.779775280899], [622
0.81, 2338.1303370786513], [6282.23, 2354.4808988764044], [6116.04, 2370.83146067415
7], [6315.53, 2387.18202247191], [6419.61, 2403.5325842696625], [6676.66, 2419.8831460
674155], [6590.55, 2436.233707865168], [6250.57, 2452.584269662921], [6315.22, 2468.93
48314606736], [6506.25, 2485.2853932584267], [6586.08, 2501.6359550561792], [6386.57,
2517.9865168539322], [6456.91, 2534.3370786516853], [6029.18, 2550.687640449438], [581
8.62, 2567.038202247191], [5852.58, 2583.3887640449434], [6087.25, 2599.739325842696
4], [6007.61, 2616.089887640449], [5479.93, 2632.440449438202], [5354.67, 2648.7910112
359546], [5602.82, 2665.1415730337076], [4959.59, 2681.49213483146], [4443.28, 2697.84
2696629213], [4065.49, 2714.1932584269657], [4434.17, 2730.5438202247187], [4077.78, 2
746.8943820224717], [3625.83, 2763.2449438202243], [3955.61, 2779.5955056179773], [424
3.22, 2795.94606741573], [4506.19, 2812.296629213483], [4340.71, 2828.6471910112355],
[4682.46, 2844.9977528089885], [4819.7, 2861.348314606741], [5047.81, 2877.69887640449
4], [5104.5, 2894.0494382022466], [5312.17, 2910.3999999999996], [5412.88, 2926.750561
797752], [5247.41, 2943.101123595505], [5405.94, 2959.4516853932582], [5744.89, 2975.8
02247191011], [5553.29, 2992.152808988764], [5163.3, 3008.5033707865164], [4805.75, 30
24.8539325842694], [5397.11, 3041.204494382022], [5366.41, 3057.555056179775], [5592.
9, 3073.9056179775275], [5694.62, 3090.2561797752805], [5642.5, 3106.606741573033], [5
899.94, 3122.957303370786], [5957.82, 3139.307865168539], [5935.76, 3155.658426966291
7], [6009.92, 3172.0089887640447], [6069.9, 3188.3595505617973], [5928.61, 3204.710112
3595503], [5989.76, 3221.060674157303], [5774.43, 3237.411235955056], [5418.65, 3253.7
617977528084], [5075.5, 3270.1123595505615], [5421.57, 3286.462921348314], [5489.34, 3
302.813483146067], [5572.28, 3319.1640449438196], [5790.72, 3335.5146067415726], [593
1.25, 3351.8651685393256], [5874.89, 3368.215730337078], [5812.23, 3384.566292134831],
[5260.19, 3400.9168539325838], [5640.64, 3417.267415730337], [5712.82, 3433.6179775280
893], [5758.41, 3449.9685393258424], [5820.45, 3466.319101123595], [5861.92, 3482.6696
62921348], [5871.24, 3499.0202247191005], [5897.81, 3515.3707865168535], [6347.24, 353
1.721348314606], [6378.6, 3548.071910112359], [6411.74, 3564.422471910112], [6451.29,
3580.7730337078647], [6525.12, 3597.1235955056177], [6307.78, 3613.4741573033702], [66
81.98, 3629.8247191011233], [6506.19, 3646.175280898876], [6460.01, 3662.52584269662
9], [6734.74, 3678.8764044943814], [6595.33, 3695.2269662921344], [6749.09, 3711.57752
8089887], [6465.66, 3727.92808988764], [6708.35, 3744.2786516853926], [6652.61, 3760.6
292134831456], [6808.87, 3776.9797752808986], [6864.1, 3793.330337078651], [6802.92, 3
809.680898876404], [6679.18, 3826.0314606741567], [6825.31, 3842.3820224719097], [655
7.52, 3858.7325842696623], [6487.97, 3875.0831460674153], [6656.37, 3891.43370786516
8], [6566.09, 3907.784269662921], [6782.55, 3924.1348314606735], [6940.64, 3940.485393
2584265], [6809.5, 3956.835955056179], [6985.95, 3973.186516853932], [6953.58, 3989.53
70786516845], [6608.59, 4005.8876404494376], [6688.62, 4022.2382022471907], [6058.54, 4
038.588764044943], [6072.47, 4054.9393258426962], [6361.8, 4071.289887640449], [6395.6
5, 4087.640449438202], [6242.32, 4103.991011235955], [6060.1, 4120.341573033707], [615
2.88, 4136.69213483146], [6146.05, 4153.0426966292125], [6241.89, 4169.393258426966],
[6191.93, 4185.743820224719], [6577.83, 4202.094382022471], [6693.95, 4218.44494382022
5], [6745.97, 4234.795505617977], [6983.52, 4251.14606741573], [6917.14, 4267.49662921
3482], [6752.93, 4283.847191011236], [7142.83, 4300.197752808988], [7107.65, 4316.5483
14606741], [7382.9, 4332.898876404493], [7282.69, 4349.249438202247]]
[[100.0, 0.0], [83.0, 9.275307692307694], [72.7, 18.550615384615387], [70.95, 27.82592
3076923083], [82.92, 37.101230769230774], [81.19, 46.376538461538466], [70.23, 55.6518
46153846165], [77.56, 64.92715384615386], [104.28, 74.20246153846155], [149.15, 83.477
76923076924], [198.63, 92.75307692307693], [158.05, 102.02838461538462], [185.52, 111.
30369230769233], [165.57, 120.57900000000002], [178.64, 129.8543076923077], [189.73, 1
39.1296153846154], [200.5, 148.4049230769231], [276.79, 157.6802307692308], [322.31, 1
66.95553846153848], [298.52, 176.2308461538462], [321.19, 185.50615384615386], [355.4
6, 194.78146153846157], [423.87, 204.05676923076925], [510.2, 213.33207692307695], [45
3.22, 222.60738461538466], [503.96, 231.88269230769234], [440.82, 241.15800000000004],
[322.91, 250.43330769230772], [396.86, 259.7086153846154], [300.61, 268.9839230769231
3], [323.91, 278.2592307692308], [321.91, 287.5345384615385], [312.37, 296.8098461538
462], [171.71, 306.0851538461539], [145.72, 315.3604615384616], [145.64, 324.6357692307
6926], [65.66, 333.91107692307696], [41.18, 343.18638461538467], [58.38, 352.461692307
6924], [67.74, 361.737], [58.93, 371.01230769230773], [55.09, 380.28761538461544], [7
3.5, 389.56292307692314], [78.7, 398.83823076923085], [91.83, 408.1135384615385], [96.
76, 417.3888461538462], [131.48, 426.6641538461539], [115.77, 435.9394615384616], [10
2.95, 445.2147692307693], [84.5, 454.49007692307697], [97.28, 463.7653846153847], [11
3.41, 473.0406923076924], [175.26, 482.3160000000001], [171.29, 491.59130769230774],
[180.47, 500.86661538461544], [226.22, 510.14192307692315], [226.87, 519.41723076923
09], [194.75, 528.6925384615386], [176.38, 537.9678461538463], [200.37, 547.24315384615
4], [237.15, 556.5184615384616], [203.83, 565.7937692307693], [194.09, 575.0690769230
77], [141.71, 584.3443846153847], [970.93, 593.6196923076924], [170.33, 602.8950000000
001], [161.55, 612.1703076923078], [167.58, 621.4456153846155], [182.53, 630.720923076
9232], [213.47, 639.9962307692308], [214.79, 649.2715384615385], [195.08, 658.546846153
8462], [209.57, 667.8221538461539], [179.9, 677.0974615384616], [204.13, 686.372769230
7693], [227.01, 695.648076923077], [260.05, 704.9233846153847], [288.01, 714.198692307
6925], [305.47, 723.474], [349.77, 732.7493076923078], [386.1, 742.0246153846155], [39
1.37, 751.2999230769232], [361.78, 760.5752307692309], [332.3, 769.8505384615386], [32
9.44, 779.1258461538463], [335.18, 788.401153846154], [357.45, 797.6764615384617], [36
1.01, 806.9517692307693], [359.07, 816.227076923077], [347.06, 825.5023846153847], [39
1.26, 834.7776923076924], [360.66, 844.0530000000001], [422.37, 853.3283076923078], [4
55.66, 862.6036153846155], [502.96, 871.8789230769232], [451.05, 881.1542307692309],
[540.53, 890.4295384615386], [573.85, 899.7048461538462], [538.14, 908.9801538461539],
[530.43, 918.2554615384616], [567.25, 927.5307692307694], [607.87, 936.8060769230771],
[688.72, 946.0813846153848], [748.95, 955.3566923076925], [631.11, 964.6320000000002],
[557.45, 973.9073076923079], [585.15, 983.1826153846155], [545.81, 992.4579230769232],
[583.67, 1001.7332307692309], [648.28, 1011.0085384615386], [665.71, 1020.283846153846
3], [611.07, 1029.559153846154], [614.11, 1038.8344615384617], [640.32, 1048.109769230
7694], [714.1, 1057.3850769230771], [680.38, 1066.6603846153848], [670.36, 1075.935692
3076925], [667.13, 1085.2110000000002], [717.87, 1094.486307692308], [782.4, 1103.7616
153846154], [894.57, 1113.0369230769231], [1038.76, 1122.3122307692308], [942.11, 113
1.5875384615385], [1058.93, 1140.8628461538462], [1125.6, 1150.138153846154], [1339.5
7, 1159.4134615384617], [1450.82, 1168.6887692307694], [1450.88, 1177.964076923077],
[1657.28, 1187.2393846153848], [1459.99, 1196.5146923076925], [1512.04, 1205.790000000
0002], [1544.88, 1215.065307692308], [1620.0, 1224.3406153846156], [1557.23, 1233.6159
230769233], [1644.86, 1242.891230769231], [1780.51, 1252.1665384615387], [1921.92, 126
1.4418461538464], [1894.25, 1270.7171538461541], [1797.1, 1279.9924615384616], [1927.
3, 1289.2677692307693], [1935.51, 1298.543076923077], [1829.5, 1307.8183846153847], [1
911.75, 1317.0936923076924], [1946.34, 1326.3690000000001], [1926.5, 1335.644307692307
8], [2047.09, 1344.9196153846156], [2220.16, 1354.1949230769233], [2200.75, 1363.47023
0769231], [2290.51, 1372.7455384615387], [1968.97, 1382.0208461538464], [2032.29, 139
1.296153846154], [2063.69, 1400.5714615384618], [2122.5, 1409.8467692307695], [2452.7
6, 1419.1220769230772], [2242.74, 1428.397384615385], [1941.73, 1437.6726923076926],
[1666.52, 1446.948], [1189.06, 1456.2233076923078], [802.39, 1465.4986153846155], [63
1.84, 1474.7739230769232], [631.89, 1484.049230769231], [508.25, 1493.3245384615386],
[539.37, 1502.5998461538463], [685.51, 1511.875153846154], [832.87, 1521.150461538461
7], [1167.42, 1530.4257692307694], [977.94, 1539.7010769230772], [1067.98, 1548.976384
6153849], [1073.62, 1558.2516923076926], [1266.85, 1567.5270000000003], [1357.06, 157
6.802307692308], [1416.3, 1586.0776153846157], [1444.61, 1595.3529230769234], [1470.7
9, 1604.628230769231], [1429.77, 1613.9035384615386], [1608.39, 1623.1788461538463],
[1572.84, 1632.454153846154], [1373.87, 1641.7294615384617], [1288.72, 1651.0047692307
694], [1523.16, 1660.280076923077], [1449.99, 1669.5553846153848], [1531.91, 1678.8306
923076925], [1600.36, 1688.1060000000002], [1633.71, 1697.381307692308], [1770.28, 170
6.6566153846156], [1910.01, 1715.9319230769233], [1962.68, 1725.207230769231], [2076.7
8, 1734.4825384615388], [2026.94, 1743.7578461538465], [1876.81, 1753.0331538461542],
[1937.57, 1762.3084615384619], [1985.78, 1771.5837692307696], [1703.86, 1780.859076923
0773], [1291.7, 1790.1343846153848], [1484.98, 1799.4096923076925], [1546.7, 1808.6850
000000002], [1381.87, 1817.9603076923079], [1599.56, 1827.2356153846156], [1723.57, 18
36.5109230769233], [1643.2, 1845.786230769231], [1594.2, 1855.0615384615387], [1227.6
5, 1864.3368461538464], [1357.34, 1873.6121538461541], [1374.98, 1882.8874615384618],
[1406.04, 1892.1627692307695], [1513.81, 1901.4380769230772], [1443.5, 1910.7133846153
85], [1451.69, 1919.9886923076926], [1526.98, 1929.2640000000004], [1628.31, 1938.5393
07692308], [1509.8, 1947.8146153846158], [1445.57, 1957.0899230769232], [1407.21, 196
6.365230769231], [1319.01, 1975.6405384615387], [1276.85, 1984.9158461538464], [1331.7
1, 1994.191153846154], [1291.98, 2003.4664615384618], [1438.04, 2012.7417692307695],
[1475.18, 2022.0170769230772], [1398.74, 2031.292384615385], [1442.73, 2040.5676923076
926], [1293.2, 2049.8430000000003], [1115.06, 2059.118307692308], [1235.74, 2068.39361
53846157], [1155.7, 2077.6689230769234], [1322.52, 2086.944230769231], [1362.1, 2096.2
19538461539], [1212.74, 2105.4948461538465], [1177.66, 2114.7701538461542], [1113.42,
2124.045461538462], [1078.33, 2133.3207692307697], [958.33, 2142.5960769230774], [790.
71, 2151.871384615385], [745.82, 2161.146692307693], [903.56, 2170.4220000000005], [91
0.41, 2179.697307692308], [1029.31, 2188.972615384616], [953.88, 2198.2479230769236],
[930.66, 2207.523230769231], [834.89, 2216.7985384615386], [805.06, 2226.073846153846
3], [781.65, 2235.349153846154], [853.77, 2244.6244615384617], [839.14, 2253.899769230
7694], [757.04, 2263.175076923077], [730.39, 2272.450384615385], [786.36, 2281.7256923
076925], [861.96, 2291.001], [951.11, 2300.276307692308], [890.52, 2309.551615384615
6], [933.32, 2318.8269230769233], [922.53, 2328.102230769231], [944.96, 2337.377538461
5387], [1004.69, 2346.6528461538464], [996.63, 2355.928153846154], [1047.43, 2365.2034
61538462], [1152.33, 2374.4787692307696], [1167.53, 2383.7540769230773], [1109.39, 239
3.029384615385], [1122.98, 2402.3046923076927]]
[[115.39, 0.0], [109.08, 10.319780219780219], [106.25, 20.639560439560437], [100.86, 3
0.959340659340654], [93.96, 41.279120879120875], [92.99, 51.598901098901095], [90.71,
61.91868131868131], [94.07, 72.23846153846154], [92.74, 82.55824175824175], [93.56, 9
2.87802197802196], [92.75, 103.19780219780219], [91.47, 113.5175824175824], [90.55, 12
3.83736263736262], [92.88, 134.15714285714284], [95.72, 144.47692307692307], [100.98,
154.79670329670327], [116.44, 165.1164835164835], [123.67, 175.43626373626373], [126.9
8, 185.75604395604392], [128.29, 196.07582417582415], [131.67, 206.39560439560438], [1
42.76, 216.71538461538458], [142.45, 227.0351648351648], [141.8, 237.35494505494503],
[145.18, 247.67472527472523], [133.01, 257.99450549450546], [127.32, 268.314285714285
7], [131.17, 278.6340659340659], [121.66, 288.95384615384614], [122.36, 299.2736263736
263], [120.78, 309.59340659340654], [118.06, 319.91318681318677], [115.53, 330.2329670
32967], [116.05, 340.5527472527472], [114.22, 350.87252747252745], [117.81, 361.192307
6923077], [117.49, 371.51208791208785], [119.12, 381.8318681318681], [125.06, 392.1516
483516483], [125.02, 402.47142857142853], [122.65, 412.79120879120876], [123.27, 423.1
10989010989], [119.57, 433.43076923076916], [120.48, 443.7505494505494], [121.78, 454.
0703296703296], [117.05, 464.39010989010984], [116.44, 474.70989010989007], [118.57, 4
85.0296703296703], [113.89, 495.34945054945047], [115.36, 505.6692307692307], [114.28,
515.9890109890109], [113.15, 526.3087912087911], [112.56, 536.6285714285714], [109.37,
546.9483516483516], [107.36, 557.2681318681318], [110.58, 567.5879120879121], [110.88,
577.9076923076923], [95.48, 588.2274725274725], [93.66, 598.5472527472526], [100.14, 6
08.8670329670329], [100.2, 619.1868131868131], [99.91, 629.5065934065933], [107.93, 63
9.8263736263735], [92.0, 650.1461538461538], [61.46, 660.465934065934], [70.49, 670.78
57142857142], [900.9, 681.1054945054944], [870.03, 691.4252747252747], [967.37, 701.74
50549450549], [1000.56, 712.0648351648351], [903.4, 722.3846153846154], [872.03, 732.7
043956043955], [844.9, 743.0241758241757], [781.56, 753.3439560439559], [718.64, 763.6
637362637362], [730.56, 773.9835164835164], [638.54, 784.3032967032966], [621.93, 794.
6230769230768], [579.86, 804.9428571428571], [551.22, 815.2626373626373], [510.23, 82
5.5824175824175], [539.47, 835.9021978021977], [566.47, 846.221978021978], [560.73, 85
6.5417582417582], [561.01, 866.8615384615383], [604.5, 877.1813186813185], [636.28, 88
7.5010989010988], [649.48, 897.820879120879], [712.27, 908.1406593406592], [734.84, 91
8.4604395604395], [664.46, 928.7802197802197]]
```

In [ ]:

2.B) Detect potential outliers for each time series based on a simple methodology

In [32]:

```python
from sklearn.metrics.pairwise import euclidean_distances
from sklearn.cluster import DBSCAN
import pandas as pd

def DBSCAN_alg (eps, M, dataset_vector):
    print("Outlier Detection Algorithm:")

    mDistance = euclidean_distances(dataset_vector)
    clustering = density_clustering_DBSCAN(mDistance,eps,M)

    return clustering


def density_clustering_DBSCAN(mDistance, eps, M):
    labels = np.zeros(mDistance.shape[0])
    #print(labels)
    C = 0

    for i in np.arange(mDistance.shape[0]):

        if labels[i] != 0:
            continue

        vecindario = find_vecindario(mDistance, i, eps)
        #print('Vecindario:', vecindario)  # Uncomment to see output

        if len(vecindario)<M:
            # If it is NOT a nuclear point
            labels[i] = -1
        else:
            # If it is a nuclear point, a new C is assigned and we try to expand it
            C += 1
            labels[i] = C
            expand_cluster(mDistance, labels, vecindario, C, eps, M)
            #print(labels)

    return labels

def expand_cluster(mDistance, labels, vecindario, C, eps, M):
    it = 0
    while it < vecindario.size:
        j = vecindario[it]

        if labels[j] == -1:
            # If element has been labelled as no-nuclear
            labels[j] = C
        elif labels[j] == 0:
            # If element has not been labelled yet
            labels[j] = C
            # Calculation of vecindario
            vecindario_j = find_vecindario(mDistance, j, eps)
            if vecindario_j.size >= M:
                # If "j" is a nuclear point, it is mingledd with its neighbours
                # with the neighbourhood that we are analyzing
                vecindario = np.append(vecindario, vecindario_j)
        it += 1

def find_vecindario(mDistance, i, eps):
    vecinos = np.where(mDistance[i,:] < eps)[0]

    vecinos = vecinos[vecinos != i]

    return vecinos


# IMPORTANT! Dataset must have constant volatility
def outlier_detection (dataset, name):
    # I) Configuration of x-axis scale: x=range(y)
    #    between consecutive elements of the dataset
    dx =(max(dataset)-min(dataset))/(len(dataset))

    # II) Calculation of difference (delta)
    #    between consecutive elements of the dataset
    dataset_delta=[]
    for i in range(0,len(dataset)-1):
        dataset_delta.append(abs(dataset[i]-dataset[i+1]))

    # III) Calculation of the mean and standard deviation
    #     of the dataset differences
    dataset_delta_mean=np.mean(dataset_delta)
    dataset_delta_std=np.std(dataset_delta)**(1/2)

    # IV) Calculation of Chebyshev k: nº of std that
    #      contain at least 95% of observations
    k=(1/0.05)**(1/2)

    # V) Calculation of the epsilon: min distance
    #    between 2 points to be considered same cluster:
    #        Eps = 2*(dx^2 + 95% CI of daily variability)^(1/2)
    eps=2*(dx**2+(dataset_delta_mean+dataset_delta_std*k)**2)**(1/2)

    # VI) Determination of M: min 3 points within Eps. distance
    M=3

    # VII) Clustering process
    # Creation of the Dataset vector: [y,i*dx]
    dataset_vect=[]
    for i in range(len(dataset)):
        dataset_vect.append([dataset[i],i*dx])
    #clustering = DBSCAN (eps=eps, min_samples=M).fit(dataset_vect)
    clustering = DBSCAN_alg (eps,M,dataset_vect)

    # VIII) Printing Results
    print (name+" Outliers:")
    print (" - Delta mean:",dataset_delta_mean)
    print (" - Delta std:",dataset_delta_std)
    print (" - k: ",k)
    print (" - Epsilon: ",eps)
    print (" - M: ",M)
    #print (clustering.labels_)
    print(clustering)
    print ()

    return clustering

cluster_GDP   = outlier_detection (GDP, "GDP")
cluster_LIBOR = outlier_detection (LIBOR, "LIBOR")
cluster_Eq1   = outlier_detection (Eq1, "Equities I")
cluster_Eq2   = outlier_detection (Eq2, "Equities II")
cluster_Eq3   = outlier_detection (Eq3, "Equities III")
```

```
Outlier Detection Algorithm:
GDP Outliers:
 - Delta mean: 3080.262931818182
 - Delta std: 56.41750591131117
 - k:  4.47213595499958
 - Epsilon:  7348.799001038649
 - M:  3
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1. -1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.
  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.  3.  3.  3.  3.  3.  3.  3.]

Outlier Detection Algorithm:
LIBOR Outliers:
 - Delta mean: 0.12706601503759402
 - Delta std: 0.44290452184477486
 - k:  4.47213595499958
 - Epsilon:  4.215738544315322
 - M:  3
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1.]

Outlier Detection Algorithm:
Equities I Outliers:
 - Delta mean: 170.56627819548874
 - Delta std: 11.987532911705578
 - k:  4.47213595499958
 - Epsilon:  449.54327669403915
 - M:  3
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1.]

Outlier Detection Algorithm:
Equities II Outliers:
 - Delta mean: 77.20996138996139
 - Delta std: 10.03284466137934
 - k:  4.47213595499958
 - Epsilon:  244.8601224491558
 - M:  3
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1. -1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.]

Outlier Detection Algorithm:
Equities III Outliers:
 - Delta mean: 23.071444444444444
 - Delta std: 9.398752236818474
 - k:  4.47213595499958
 - Epsilon:  131.83354900643172
 - M:  3
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.]
```

In [ ]: