Tarefa 07

Renan Camargo de Castro - RA 147775

Para obter os resultados, transformei os números em pares do tipo média, desvio padrão para cada intervalo de N pontos, onde N é um parâmetro fixo arbitrário que depende fortemente da disposição dos intervalos.

Os resultados não foram muito satisfatórios, pois é necessário calibrar o N para os conjuntos de dados. Nesse laboratório isso foi feito manualmente, mas seria interessante pensar em algum algoritmo que possa explorar bem as características do problema, mas para isso seriam necessárias mais informações sobre os dados.

A abordagem do lab foi simples e consistiu em utilizar o LOF, local outlier factor, que utiliza a densidade de um ponto p e os seus k vizinhos para determinar se o ponto é um outlier local ou não. O número de vizinhos considerado na análise local foi de 20, o que costuma ser um número padrão razoável, de acordo com a literatura do sklearn.

Em baixo, coloquei os gráficos numerados por série de dados. Os pontos em vermelho foram considerados outliers pela análise. O título do gráfico mostra o N utilizado para definir intervalos no conjunto de dados.

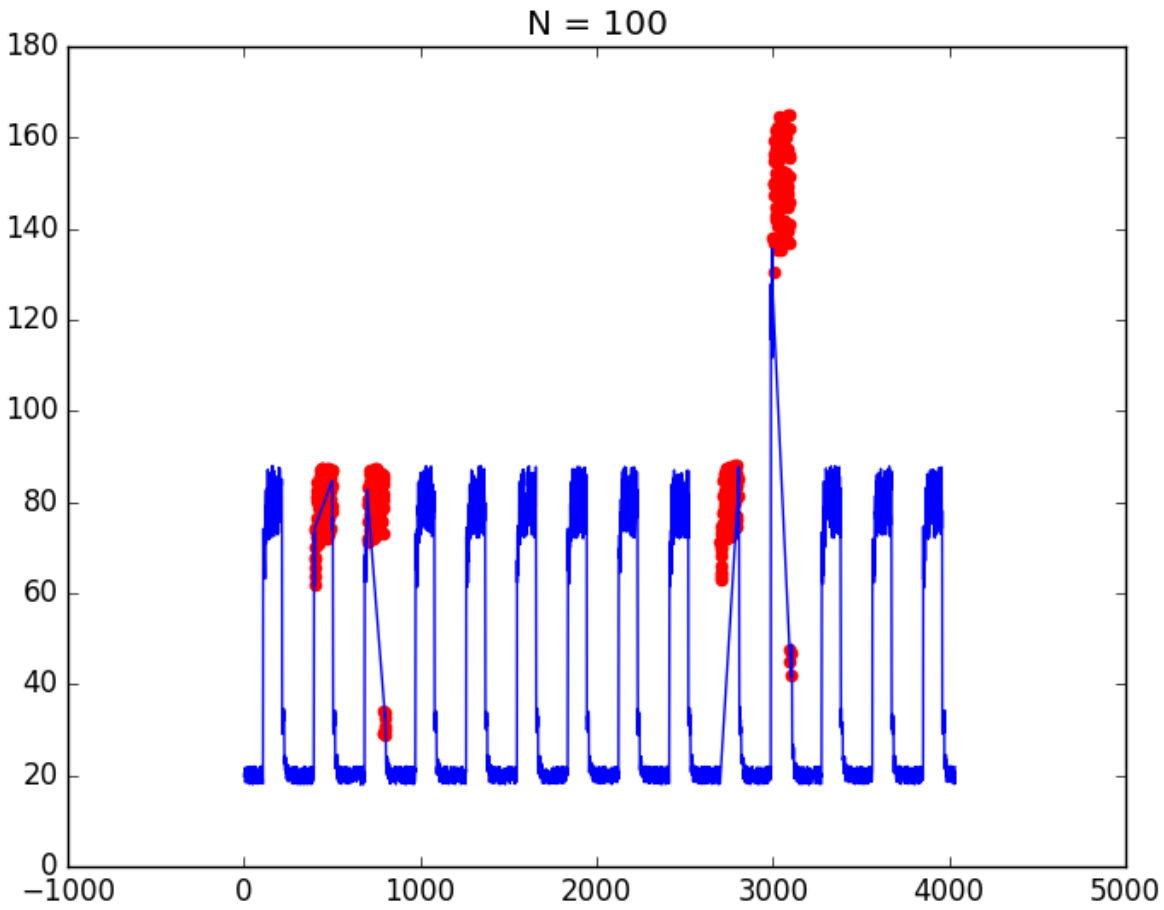
N = 100
180
160
140
120
100
80
60
40
20
0
−1000
0
1000
2000
3000
4000
5000

Figura 1

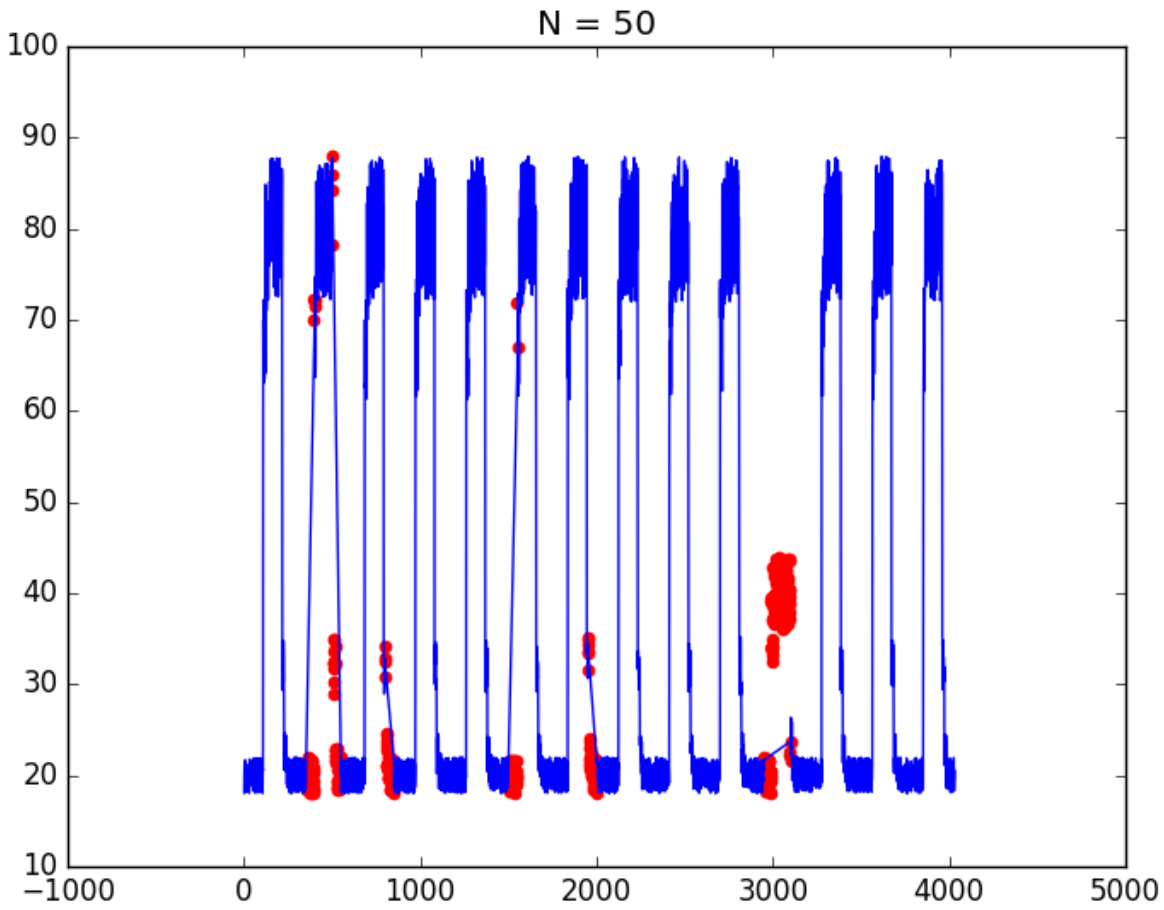

Figura 2

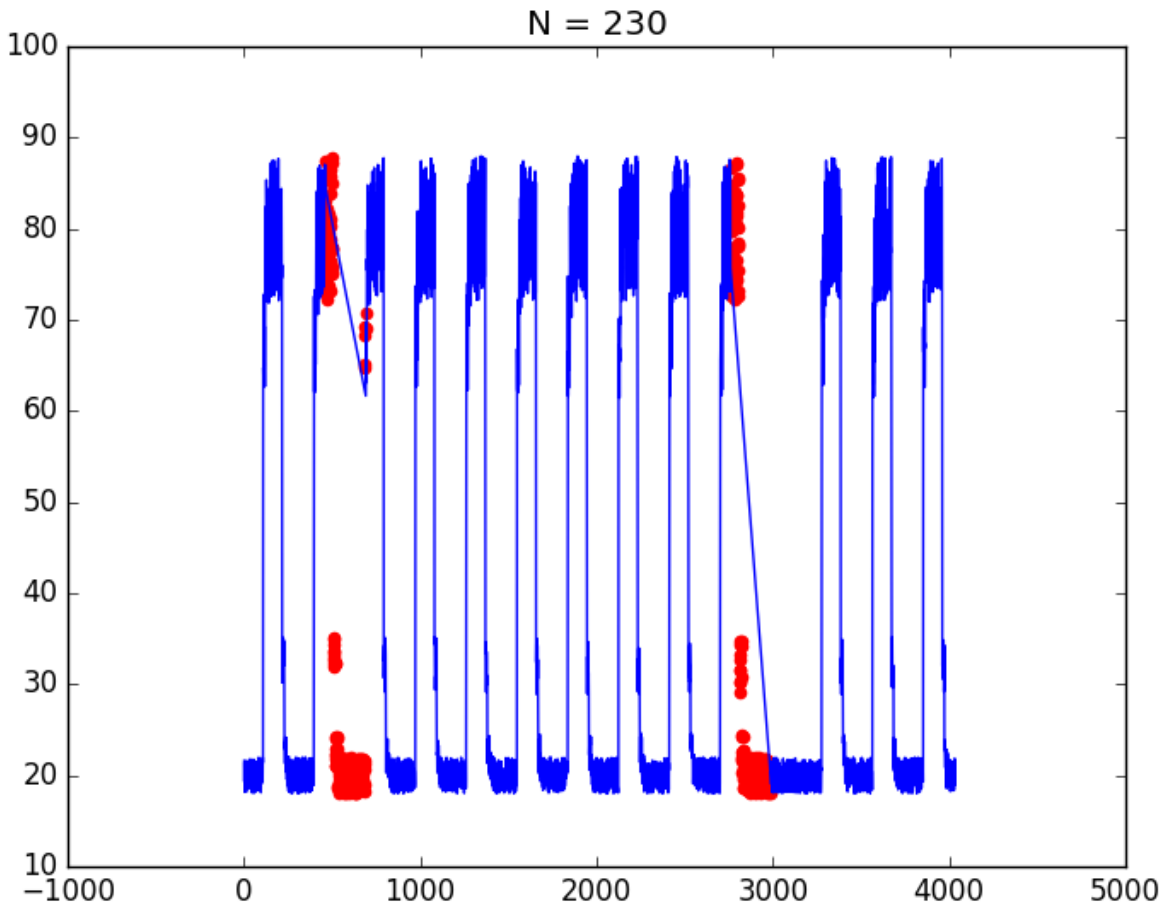
N = 230
100
90
80
70
60
50
40
30
20
10
−1000
0
1000
2000
3000
4000
5000

Figura 3

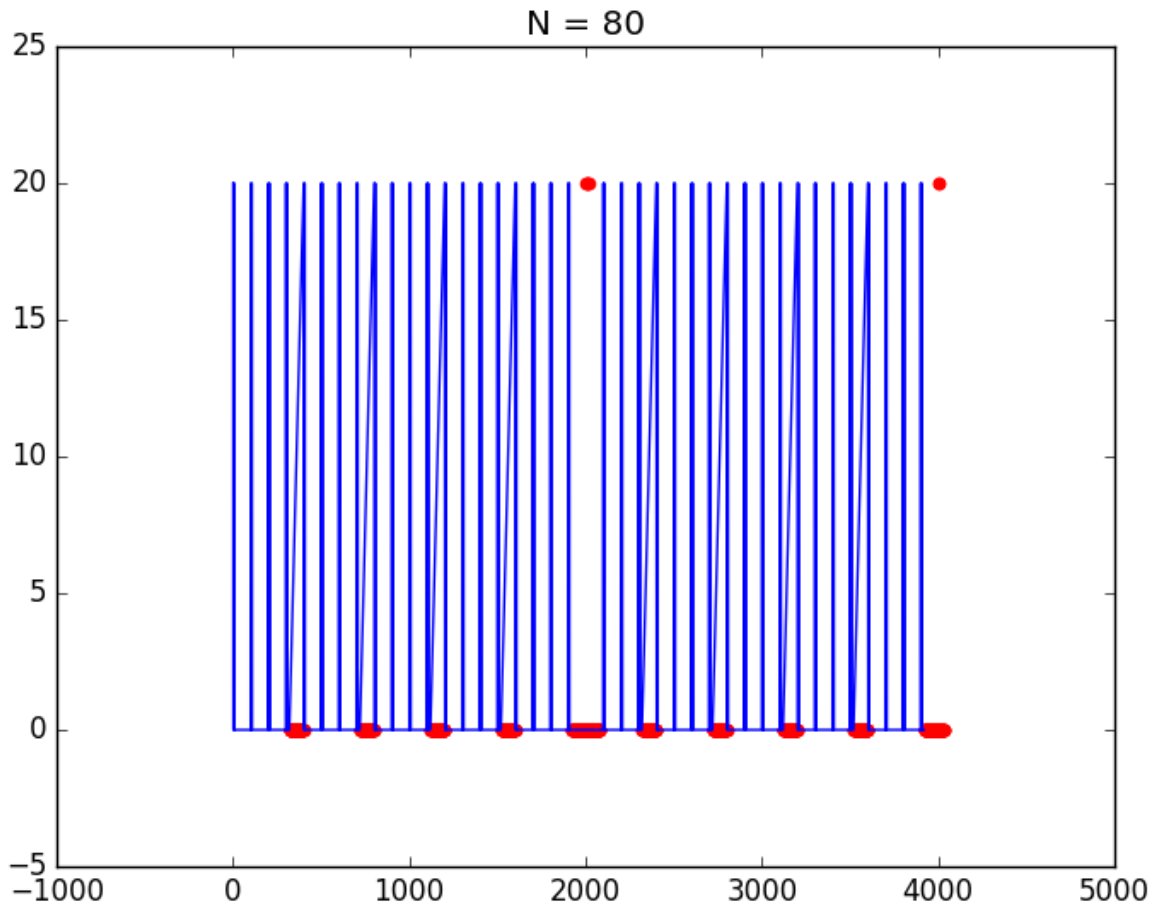

Figura 4

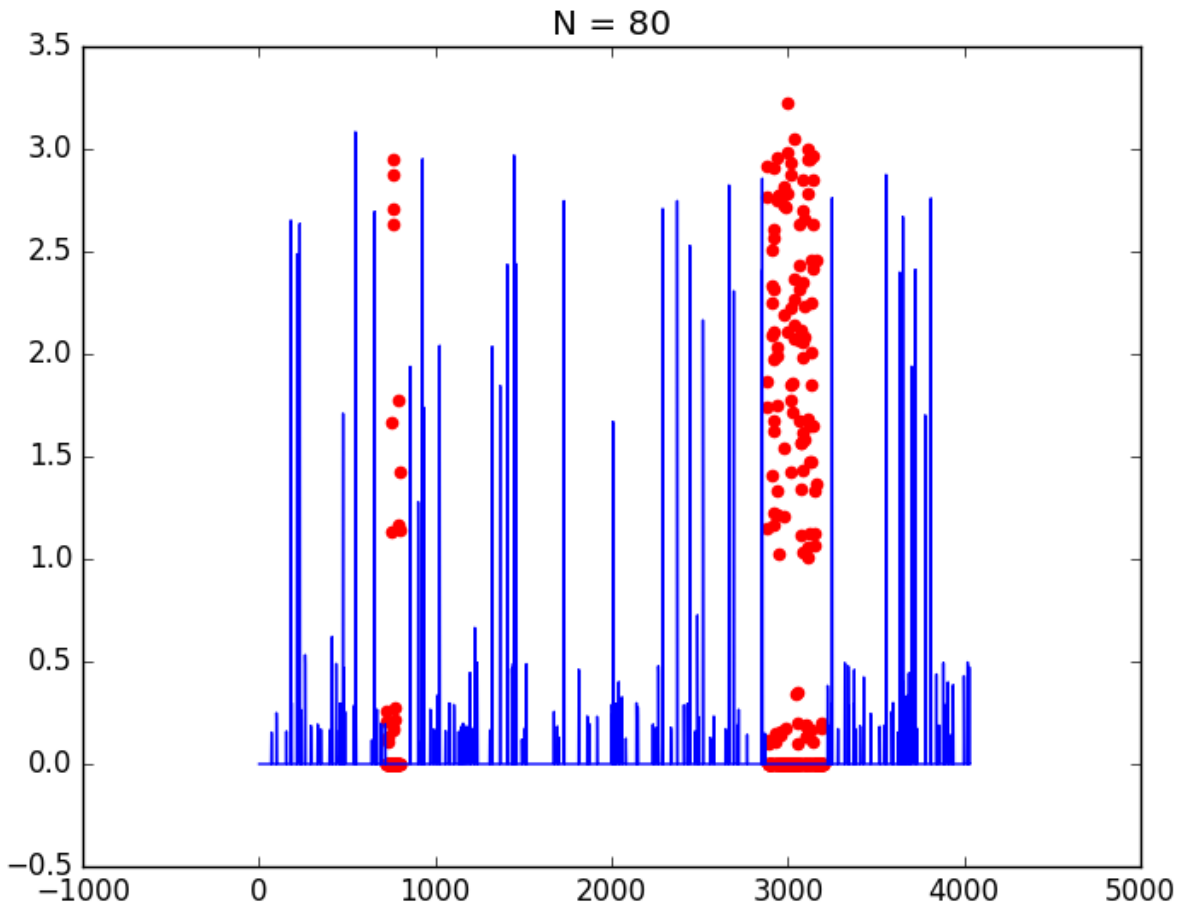

Figura 5

Python

```
from numpy import genfromtxt
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import LocalOutlierFactor


serie1 = genfromtxt('serie1.csv', delimiter=',', dtype=object)[1:]
serie2 = genfromtxt('serie2.csv', delimiter=',', dtype=object)[1:]
serie3 = genfromtxt('serie3.csv', delimiter=',', dtype=object)[1:]
serie4 = genfromtxt('serie4.csv', delimiter=',', dtype=object)[1:]
serie5 = genfromtxt('serie5.csv', delimiter=',', dtype=object)[1:]

N = 80

serie1_trecho = []
i = 0
current_trecho = []
for value in serie5:
    i += 1
```

```
    current_trecho.append(float(value[1]))
    if i % N == 0:
        serie1_trecho.append([np.mean(current_trecho), np.std(current_trecho)])
        del current_trecho[:]
serie1_trecho = np.asarray(serie1_trecho)


clf = LocalOutlierFactor(n_neighbors=20)
y_pred = clf.fit_predict(serie1_trecho)

print y_pred.shape
i = 0
j = 0
outlier = []
not_outlier = []
for value in serie5:
    i += 1

    if y_pred[j] == -1:
        outlier.append([i, value[1]])
    else:
        not_outlier.append([i, value[1]])
    if i % N == 0:
        print j
        j += 1
        j = min(j, y_pred.shape[0]-1)



outlier = np.asarray(outlier)
not_outlier = np.asarray(not_outlier)

# print serie1_trecho
# x = serie1_trecho[:,0]
# y = serie1_trecho[:,1]
# print x
# print y
# plt.scatter(x,y)
x_outlier = outlier[:,0]
y_outlier = outlier[:,1]
print not_outlier.shape
if not_outlier.shape[0] > 0:
    x_not_outlier = not_outlier[:,0]
    y_not_outlier = not_outlier[:,1]
    plt.plot(x_not_outlier, y_not_outlier)

plt.scatter(x_outlier,y_outlier, color="red")
plt.title("N = %d" % N)
```

```
plt.show()
```