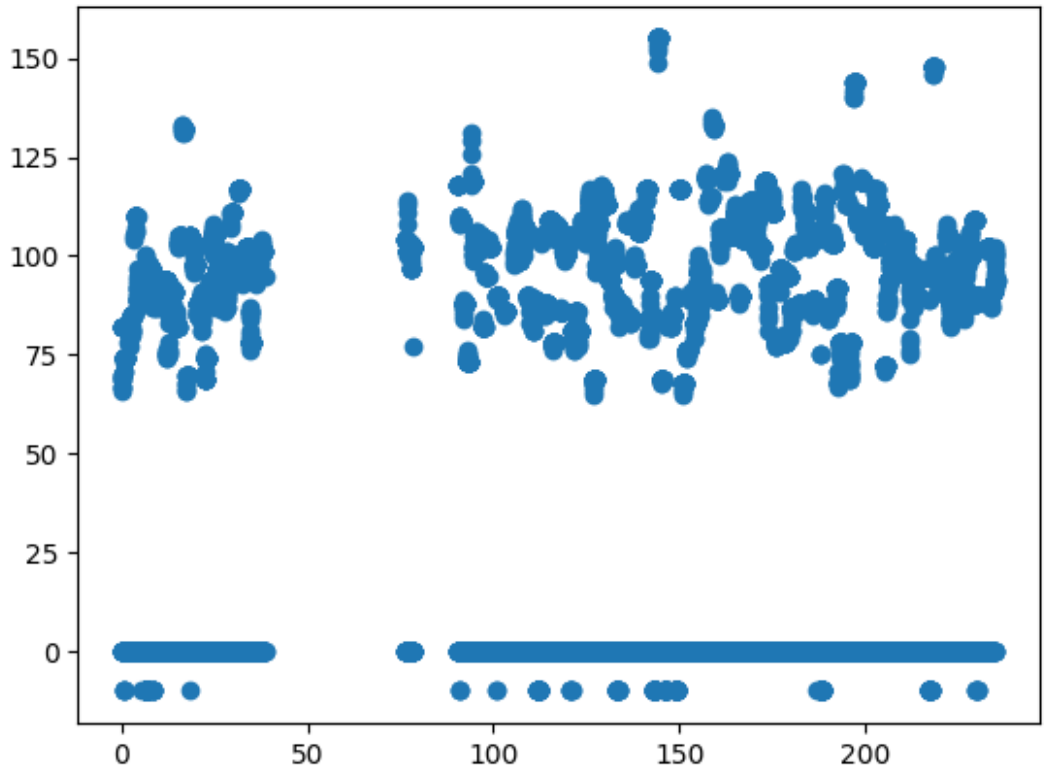
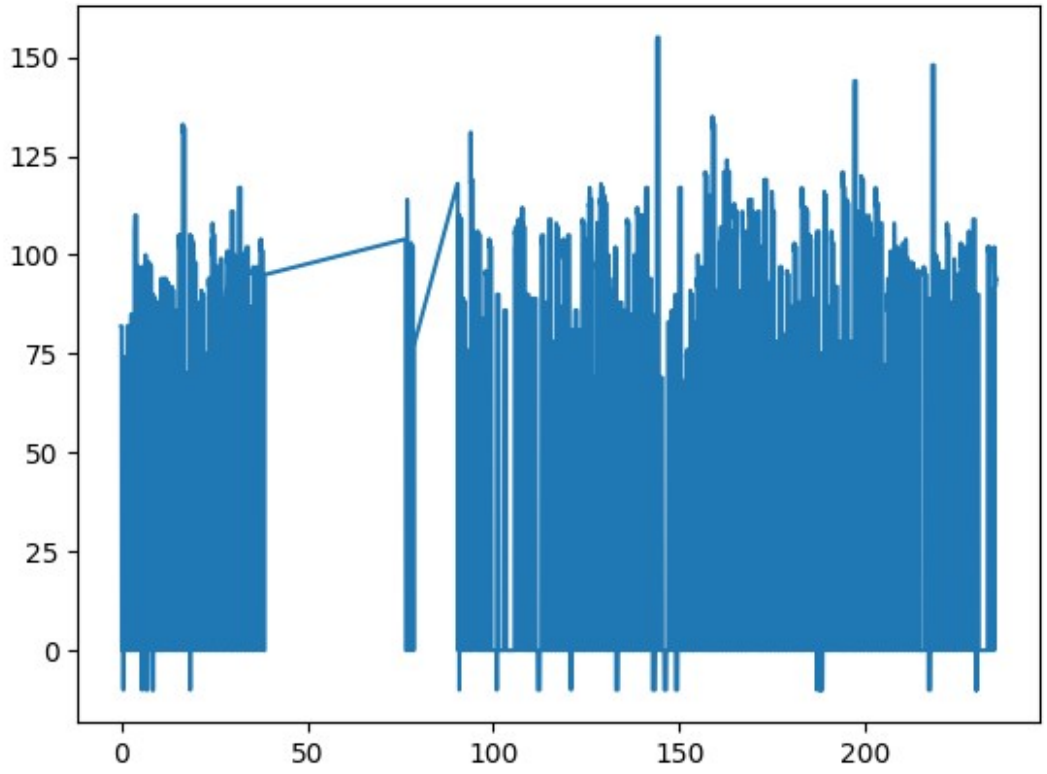
```
import sys
import warnings
import os
#Delete warning message
if not sys.warnoptions:
    warnings.simplefilter("ignore")
    os.environ["PYTHONWARNINGS"] = "ignore"

import matplotlib.pyplot as plt
import pandas as pd


columns = ['timestamp', 'heartRate', 'gyroscopeX', 'gyroscopeY',
'gyroscopeZ', 'gyroscopeRotationX', 'gyroscopeRotationY',
'gyroscopeRotationZ', 'light']

data = {}
data["drink"] = []

def read(htype, name):
    global columns
    temp = pd.read_csv(name + ".csv")
    temp.columns = columns
    temp["timestamp"] = (temp["timestamp"] - temp.loc[0, "timestamp"])
/ 1000 / 60
    data[htype].append({'name':name, 'data': temp})

read("drink","[drink, p1] Health-2019-10-29 23-56-55")
read("drink","[drink, p2] Health 2019-10-29 22_12_32")
read("drink","[drink, p3] Health 2019-10-31 22_50_39")
read("drink","[drink, p4] Health 2019-11-06 20_00_23")
```

New Section

```
temp = data['drink'][0]['data']
plt.scatter(temp["timestamp"], temp["heartRate"])
plt.show()

plt.plot(temp["timestamp"], temp["heartRate"])
plt.show()
```

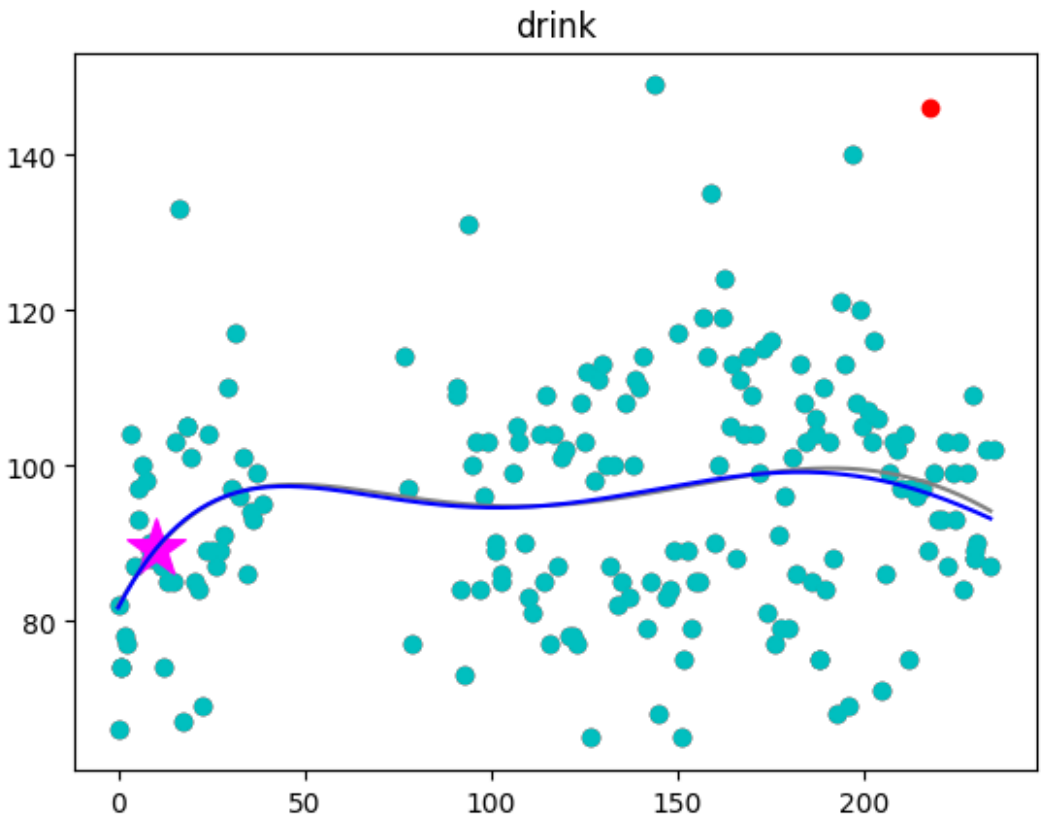

150
125
100
75
50
25
0
0
50
100
150
200


150
125
100
75
50
25
0
0
50
100
150
200

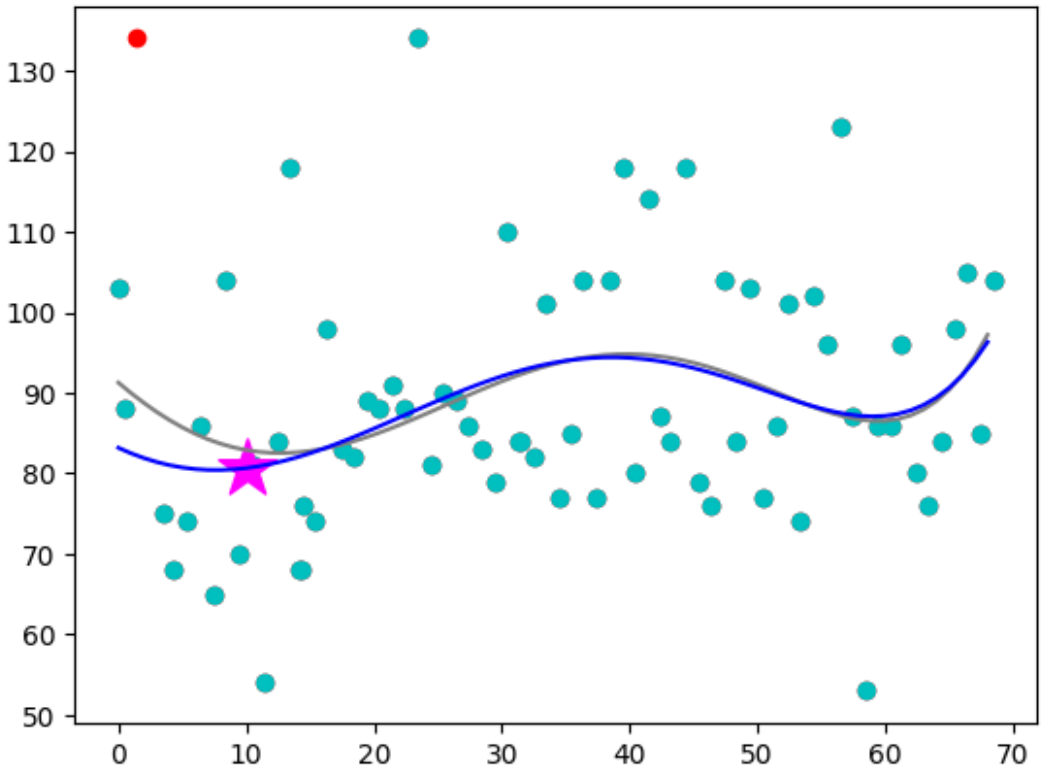
```
import numpy as np

from sklearn.linear_model import Ridge

from sklearn.neighbors import LocalOutlierFactor
from sklearn.cluster import DBSCAN

def Preprocessing(X):
    X = np.array(X)
    X2 = X * X
    X3 = X * X * X
    X4 = X * X * X * X
    X5 = X * X * X * X * X
    return np.column_stack([X, X2, X3, X4, X5])

def HeartRate(data):
    temp = data.copy()


    # Remove duplicated data in hreat rate or null data
    index_list = []
    last = 0
    for index, row in temp.iterrows():
        if (row["heartRate"] > 0 and last <= 0):
            index_list.append(index)

        last = row["heartRate"]

    temp = temp.loc[index_list]


    # Find Linear Curve
    X = np.array(temp["timestamp"]).reshape(-1, 1)
    Y = np.array(temp["heartRate"]).reshape(-1, 1)

    rid = Ridge(alpha=90)
    rid.fit(Preprocessing(X), Y)
    plt.scatter(temp["timestamp"], temp["heartRate"], c='red')

    new_X_range = np.arange(0, temp["timestamp"].max(),
dtype=np.float64)
    new_X = Preprocessing(new_X_range.reshape(-1,1))
    plt.plot(new_X_range, rid.predict(new_X),c='gray')


    # Find Outlier data
    cluster = DBSCAN(eps=23,
min_samples=8).fit_predict(np.column_stack([X,Y]))

    # Remove Outlier data
    temp = temp.loc[cluster != -1]
```

```
    # Fine Linear Curve without outlier data
    X = np.array(temp["timestamp"]).reshape(-1, 1)
    Y = np.array(temp["heartRate"]).reshape(-1, 1)
    rid = Ridge(alpha=90)
    rid.fit(Preprocessing(X), Y)

    new_X_range = np.arange(0, temp["timestamp"].max(),
dtype=np.float64)
    new_X = Preprocessing(new_X_range.reshape(-1,1))

    plt.scatter(temp["timestamp"], temp["heartRate"], c='c')
    plt.plot(new_X_range, rid.predict(new_X), c='blue')

    # In 10 minutes, Display heart rate using marker 'star'
    plt.scatter([10], rid.predict(Preprocessing([10])), c='fuchsia',
marker='*', s=500)

    plt.show()


plt.title("drink")
HeartRate(data['drink'][0]['data'])
HeartRate(data['drink'][1]['data'])
HeartRate(data['drink'][2]['data'])
HeartRate(data['drink'][3]['data'])
```

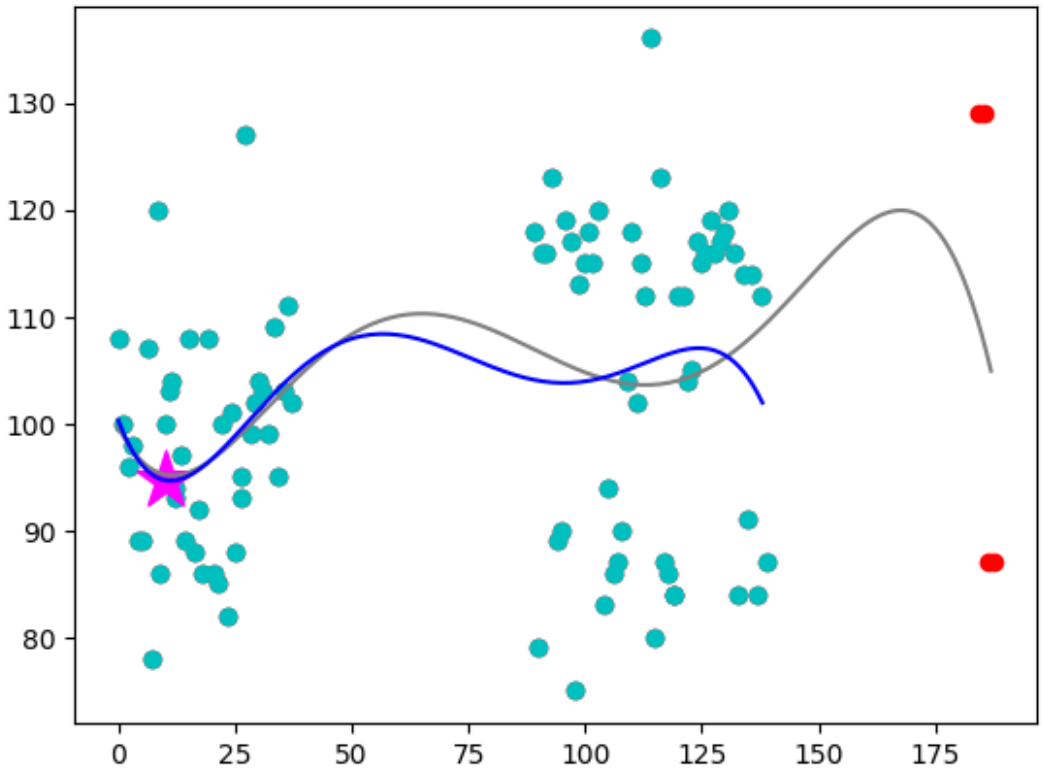
```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/
_ridge.py:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.46427e-
22): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_ridge.py
:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.48713e-22):
result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
```



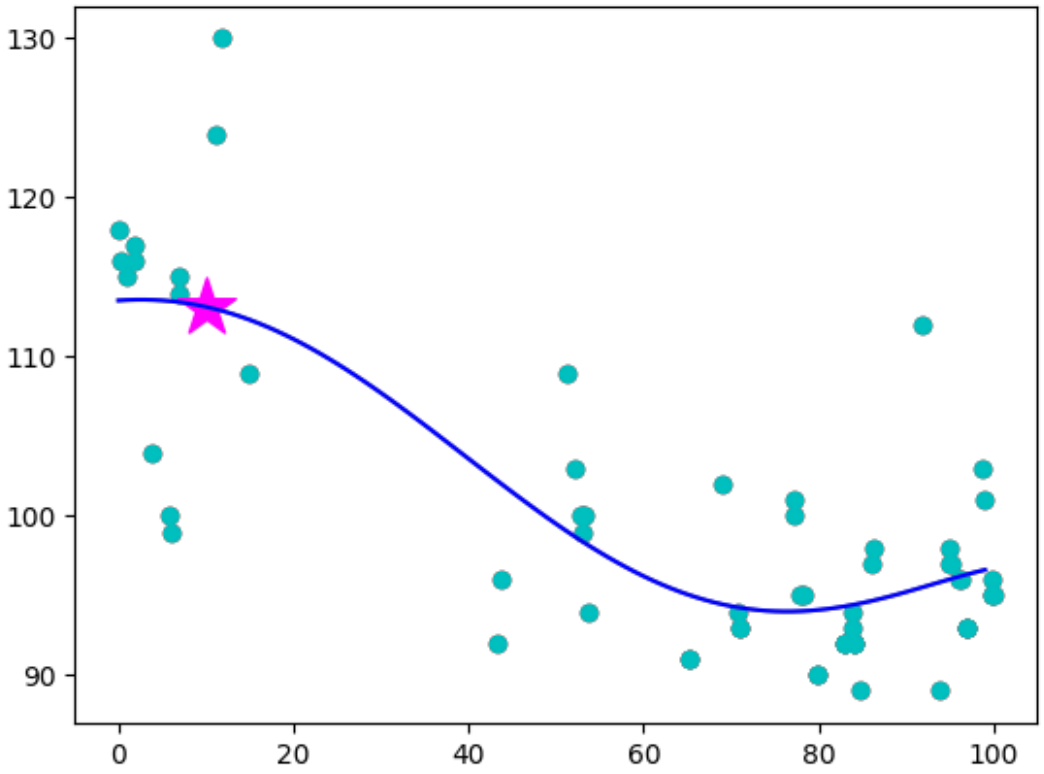

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/
_ridge.py:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.0251e-
17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_ridge.py
:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.01313e-17):
result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
```




```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/
_ridge.py:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.43457e-
21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_ridge.py
:216: LinAlgWarning: Ill-conditioned matrix (rcond=9.3013e-21): result
may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
```

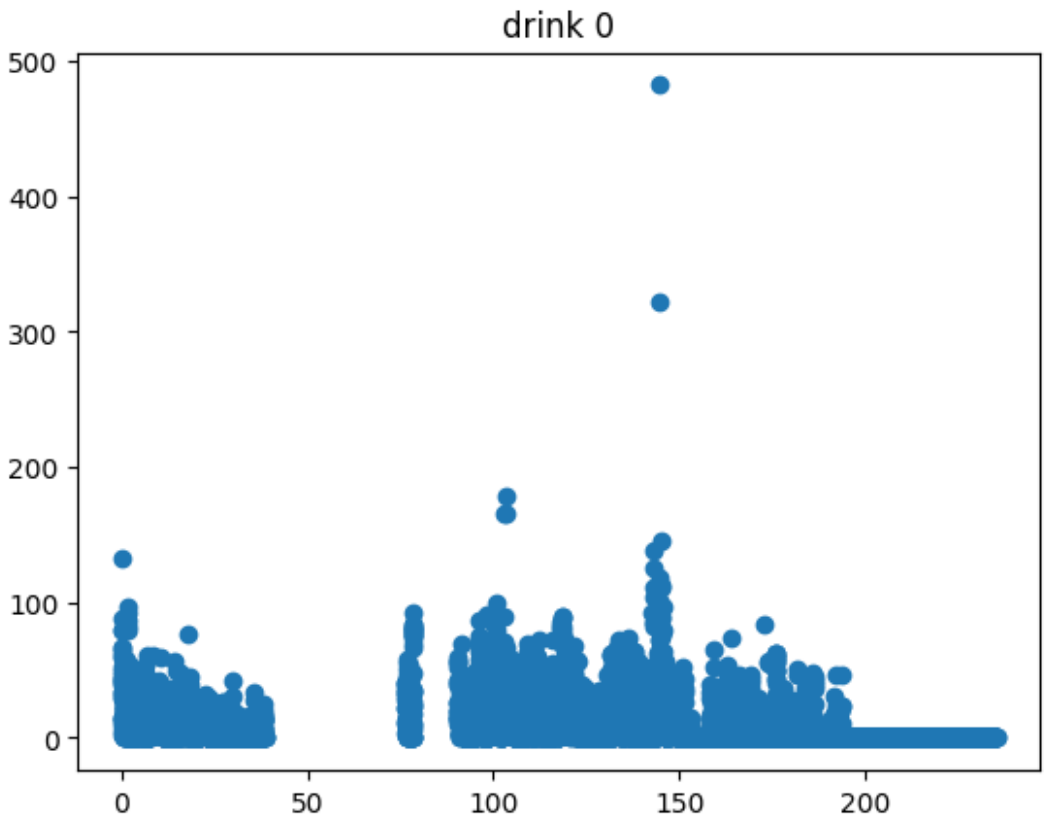


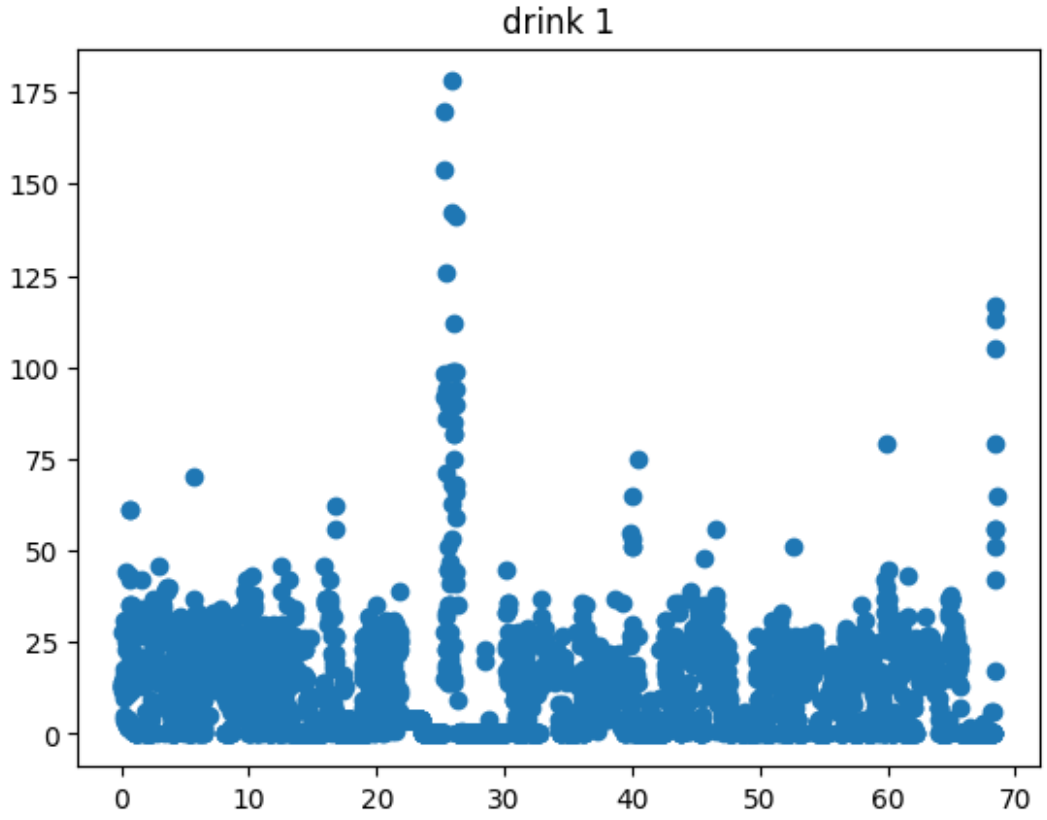

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/
_ridge.py:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.87959e-
19): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_ridge.py
:216: LinAlgWarning: Ill-conditioned matrix (rcond=1.87959e-19):
result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
```

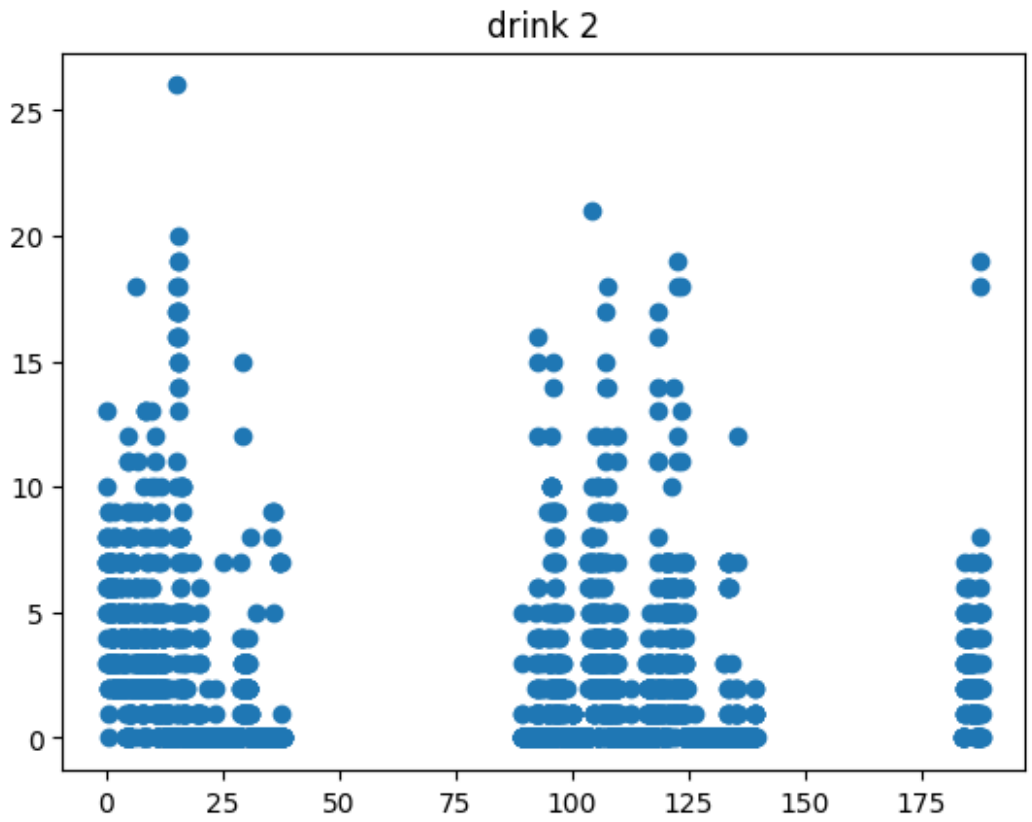


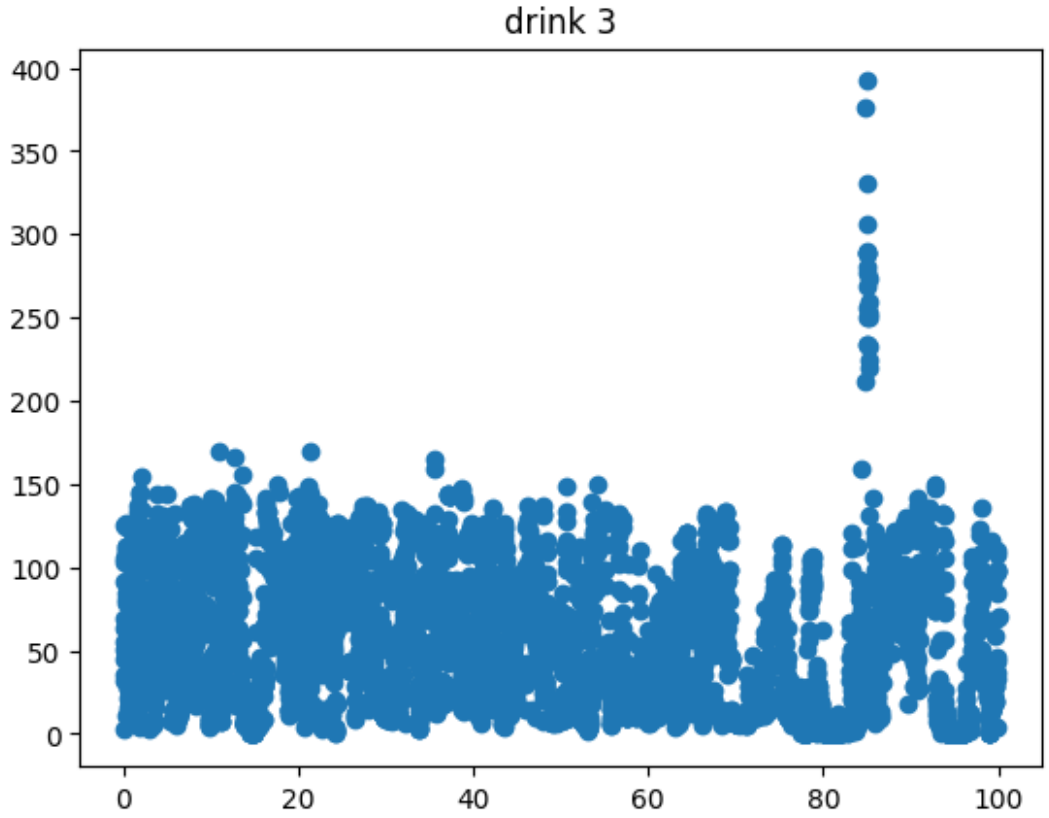
From the polts , we conclude that alcohol can lead to your heart rate temporarily jumping up in speed, and if it goes over 100 beats per minute it may lead to rash driving which may cause accidents.

```
def display_scatter(htype, no):
    temp = data[htype][no]['data']
    plt.scatter(temp["timestamp"], temp["light"])
    plt.title(htype + " " + str(no))
    plt.show()

display_scatter('drink', 0)
display_scatter('drink', 1)
display_scatter('drink', 2)
display_scatter('drink', 3)
```


drink 0
500
400
300
200
100
0
0
50
100
150
200


drink 1
175
150
125
100
75
50
25
0
0
10
20
30
40
50
60
70


drink 2
25
20
15
10
5
0
0
25
50
75
100
125
150
175


drink 3
400
350
300
250
200
150
100
50
0
0
20
40
60
80
100

Light data results

It can be seen that in most drinking environments, light is detected within 0 to 150.

Although this itself cannot be used as a core feature, it is expected that it can at least be used as a feature to assist in alcohol measurement.

(Because there are no cases of exercising in a dark environment, etc.)