```
from __future__ import division
from pandas import read_csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter


def detect_outliers(df, n, features):
    outlier_indices = []
    # iterate over features(columns)
    for col in features:
        # 1st quartile (25%)
        Q1 = np.percentile(df[col], 25)
        # 3rd quartile (75%)
        Q3 = np.percentile(df[col], 75)
        # Interquartile range (IQR)
        IQR = Q3 - Q1
        # outlier step
        outlier_step = 1.5 * IQR
        # Determine a list of indices of outliers for feature col
        outlier_list_col = df[(df[col] < Q1 - outlier_step) | (df[col] > Q3 +
outlier_step)].index
        # append the found outlier indices for col to the list of outlier
indices
        outlier_indices.extend(outlier_list_col)
    # select observations containing more than 2 outliers
    outlier_indices = Counter(outlier_indices)
    multiple_outliers = list(k for k, v in outlier_indices.items() if v > n)
    return multiple_outliers


#  数据预处理
f = open('1.csv', encoding='UTF-8')
names = ['Date', 'OLR', 'F', 'inCOD', 'HLR', 'ALR', 'pH0', 'T', 'TSS', 'VFA',
'outCOD', 'CODrate', 'VFA1', 'pH1', 'T_in', 'T_high', 'T_low', 'T_minus',
'PH_MINUS']
data = read_csv(f, names=names)

dataframe = pd.DataFrame(data)

#  找到缺失值所在列
tmp = dataframe.isnull().any()
print(tmp)

#  中位数填充缺失值
dataframe['T_high'].fillna(dataframe['T_high'].mean(), inplace=True)

tmp2 = dataframe.isnull().any()
print(tmp2)

# detect outliers
Outliers_to_drop = detect_outliers(dataframe, 2, ['OLR', 'F', 'inCOD', 'HLR',
'ALR', 'pH0', 'T', 'TSS', 'VFA', 'outCOD',
```

```
                                                'CODrate', 'VFA1', 'pH1',
'T_in', 'T_high', 'T_low', 'T_minus', 'PH_MINUS'])
print(Outliers_to_drop)
# Drop outliers
dataframe = dataframe.drop(Outliers_to_drop, axis=0).reset_index(drop=True)

plt.scatter(dataframe['Date'], dataframe['outCOD'])
plt.savefig('pic1-1')
plt.show()

plt.hist(dataframe['outCOD'], cumulative=True, density=True)
plt.savefig('pic1-2')
plt.show()

plt.hist(dataframe['outCOD'], cumulative=False, density=True)
plt.savefig('pic1-3')
plt.show()

plt.scatter(dataframe['Date'], dataframe['VFA1'])
plt.savefig('pic1-4')
plt.show()

plt.hist(dataframe['VFA1'], cumulative=True, density=True)
plt.savefig('pic1-5')
plt.show()

plt.hist(dataframe['VFA1'], cumulative=False, density=True)
plt.savefig('pic1-6')
plt.show()
```