

# 1. UVA DATA 画图：散点图

UVA 数据：对每片 WAFER 的 sensor\_value，将特定的一段数据(即 STEP)提取，再计算统计值(最常见的统计值是 MEAN, RANGE)，也就是将一段数据变成一个值，所以在给定产品、站点、腔室、参数+STEP 下，每一片 WAFER 只有一个值。

```
In [1]: import warnings
warnings.filterwarnings('ignore')

import datetime
import pymysql
import pandas as pd
import numpy as np
import datetime
import matplotlib.pyplot as plt

from tqdm import tqdm
from sqlalchemy import create_engine
from matplotlib.lines import Line2D

In [2]: # 获取uva数据
user = 'root'
host = '10.52.199.81'
password = 'Hexchip@40123'
db = 'etl'
port = 9030

engine = create_engine("mysql+pymysql://{}:{}@{}:{}/{}".format(user = user,
                                                                password = password,
                                                                host = host,
                                                                port = port,
                                                                db = db))

In [3]: df_uva_case1 = pd.read_sql_query("SELECT * FROM etl.DWD_FOC_CASE_FD_UVA_DATA_CASE2_PROCESSED", engine)

In [4]: df_uva_case1.shape
Out[4]: (1677834, 16)
```

```
In [5]: def uva_data_certain_params_plot(df, prod, oper, tool, param):

    # 查找同一个组合下对应的数据
    df_certain_param = (df.query("PRODC1 == '{}' & OPER_NO == '{}' & TOOL_NAME == '{}'".format(prod, oper, tool)))[['WAFER_ID',
                                                            'OPER_NO', 'TOOL_NAME', 'PRODC1', 'START_TIME', 'parametric_name', 'STATISTIC_RESULT', 'label']]
    .sort_values('START_TIME')
    .reset_index(drop=True)

    df_certain_param['START_TIME'] = pd.to_datetime(df_certain_param['START_TIME'], infer_datetime_format=True, format='%Y%m%d')
    df_certain_param['STATISTIC_RESULT'] = df_certain_param['STATISTIC_RESULT'].astype(float)

    para_unique_names = df_certain_param[df_certain_param['parametric_name'].str.contains(param)][['parametric_name']].unique()
    print(para_unique_names)
    n = len(para_unique_names)

    for i in range(n):
        para_unique = df_certain_param[df_certain_param['parametric_name'].str.contains(para_unique_names[i])]
        legend_elements = [
            Line2D([0], [0], color='deeppink', label=f'Bad Wafer: {para_unique[para_unique['label'] == 1]['WAFER_ID'].nunique()}'),
            Line2D([0], [0], color='royalblue', label=f'Good Wafer: {para_unique[para_unique['label'] == 0]['WAFER_ID'].nunique()}')]

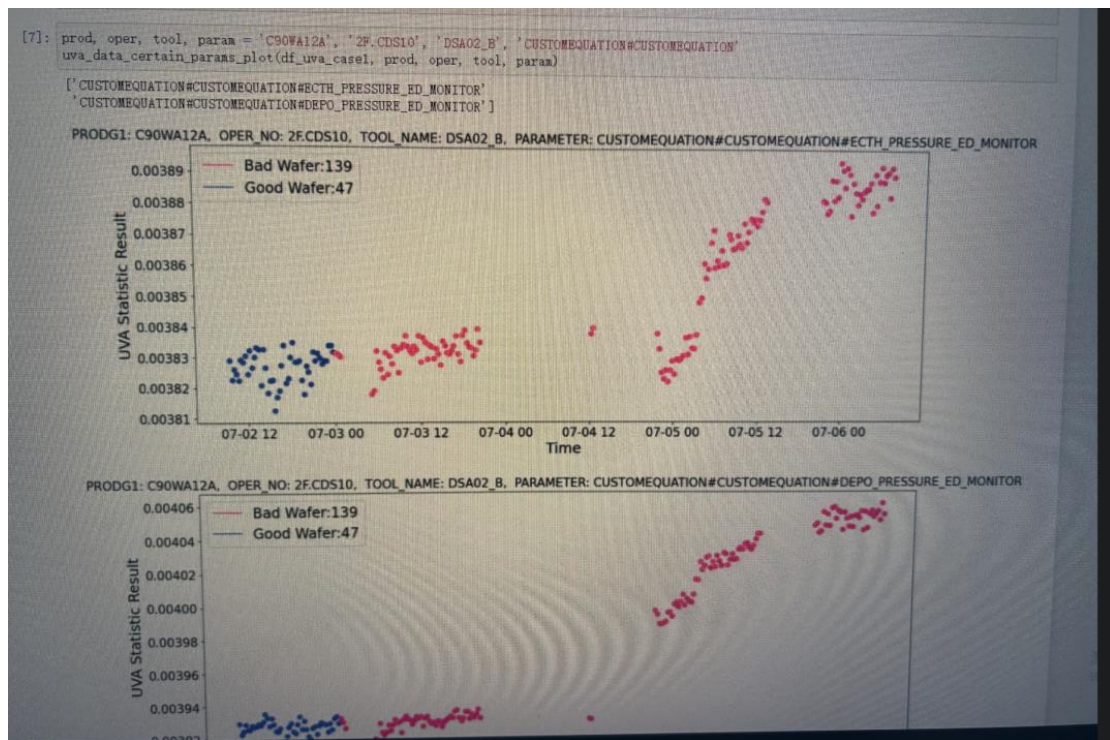
        # 分组，进行画图
        plt.figure(figsize=(16, 6))
        grouped = para_unique.groupby('label')
        for name, group in grouped:
            grp = group.sort_values('START_TIME').reset_index(drop=False)

            color = 'deeppink' if grp['label'].iloc[0] == 1 else 'royalblue'
            plt.scatter(grp['START_TIME'], grp['STATISTIC_RESULT'], color=color)

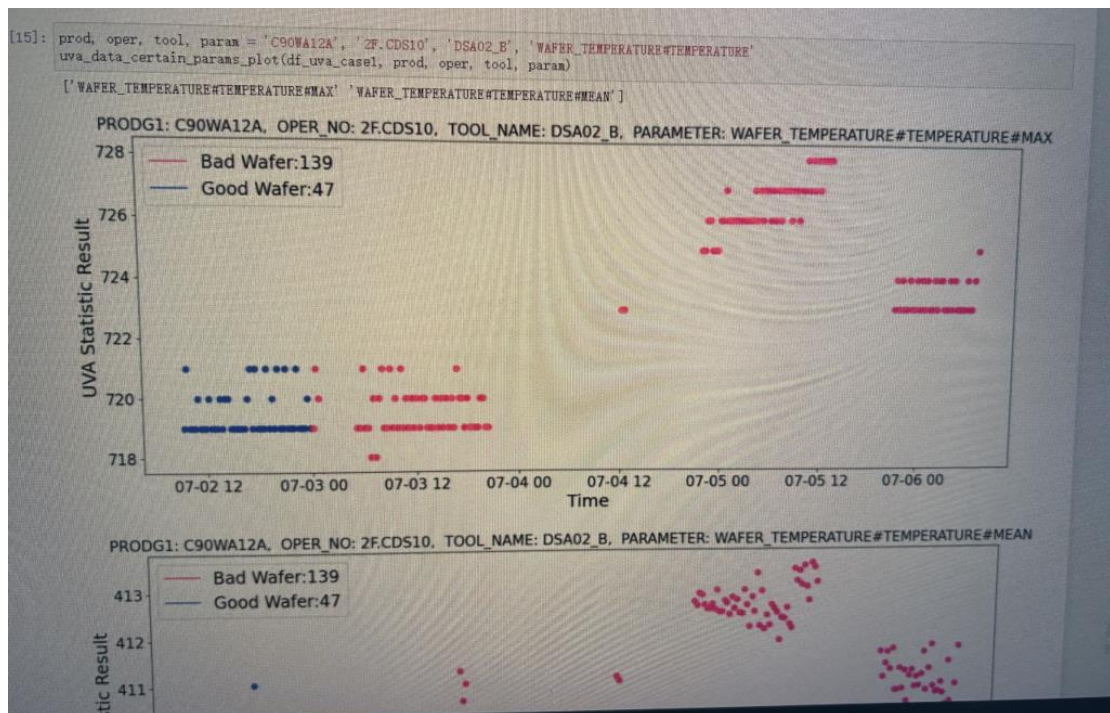
        # plt.yticks(1/n, arange(df_certain_param['sensor_value'].min(), df_certain_param['sensor_value'].max(), step = 10))
        plt.xlabel('Time', size=18)
        plt.xticks(size=15)
        plt.ylabel('UVA Statistic Result', size=18)
        plt.yticks(size=15)
        plt.legend(handles=legend_elements, fontsize='xx-large')

    plt.title('PRODC1: {}, OPER_NO: {}, TOOL_NAME: {}, PARAMETER: {}'.format(prod, oper, tool, para_unique_names[i]), size=15)
    plt.show()
```

### 例子 1: customequation



例子 2: 在产品为 C90WA12A, 站点为 2F.CDS10, 腔室为 DSA02\_B, 参数为 WAFER\_TEMPERATURE, STEP 信息为 TEMPERATURE 中, 分别提取了 MEAN 和 MAX 两种统计值, 所以在 7 月 2 日——7 月 6 日之间的 UVA 对应两幅图:

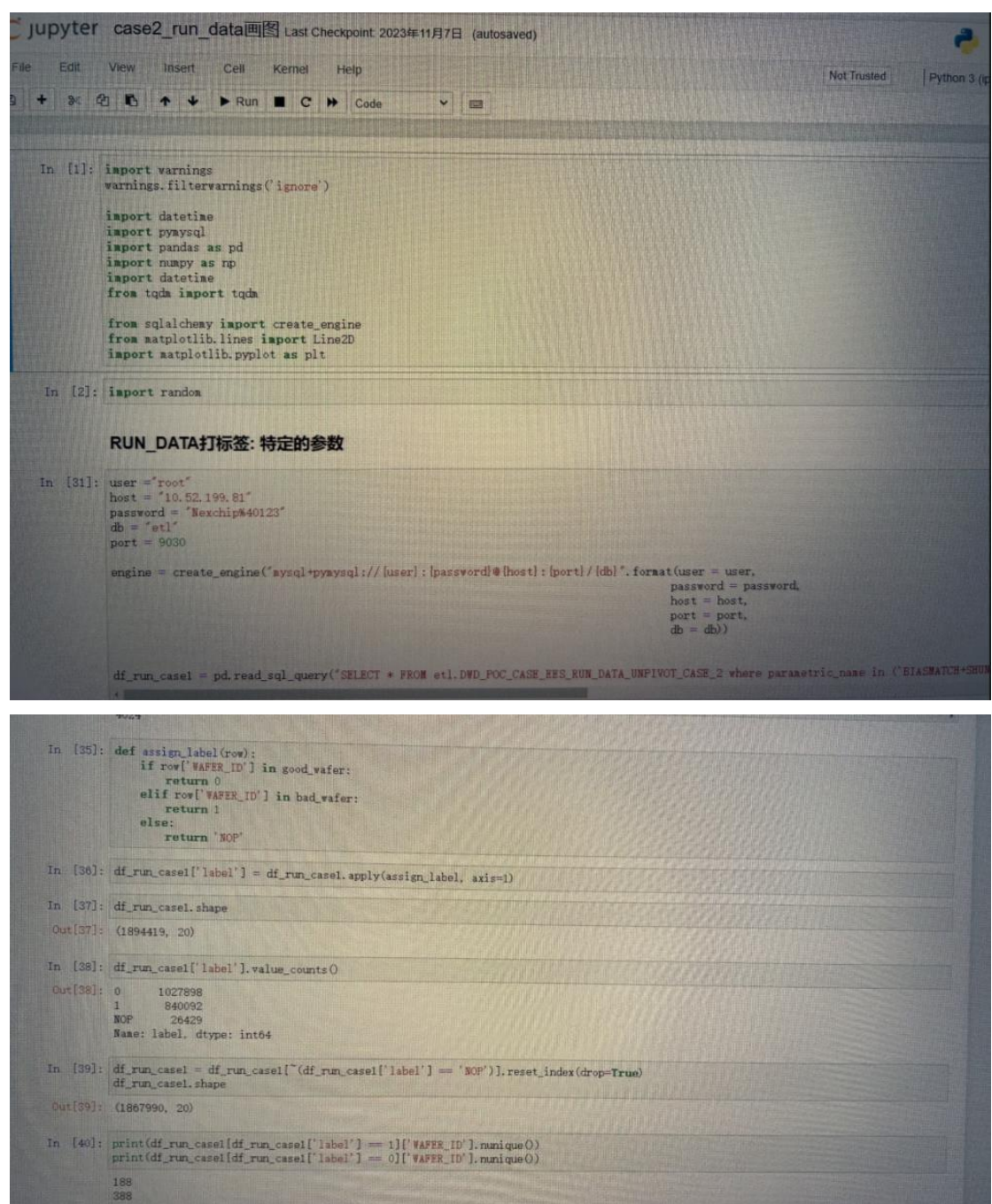




## 2. RUN DATA 下钻画图：时间趋势图

支持选取特定数量的 g/b，支持选取特定参数取值范围内进行画图，要灵活

RUN DATA: 对每片 WAFER 的 sensor\_value，对同一个 parametric\_name 按照时间先后排序，每片 WAFER 在该时间段内的此参数是一条随着时间变化的曲线。所以在给定产品、站点、腔室、参数下，每一片 WAFER 是一条曲线。横轴是 Time Index(时间从小到大进行排序后，再从 1 开始计数)，因为 good wafer 和 bad wafer 可能会在不同时间段内有参数值，按照真实时间去画图会导致曲线“压缩”，导致看不出真实的 good 和 bad 的 RUN DATA 差异。



```
jupyter case2_run_data画图 Last Checkpoint: 2023年11月7日 (autosaved)
File Edit View Insert Cell Kernel Help Not Trusted Python 3 (p

In [1]: import warnings
warnings.filterwarnings('ignore')

import datetime
import pyaysql
import pandas as pd
import numpy as np
import datetime
from tqdm import tqdm

from sqlalchemy import create_engine
from matplotlib.lines import Line2D
import matplotlib.pyplot as plt

In [2]: import random

RUN_DATA打标签: 特定的参数

In [31]: user = "root"
host = "10.52.199.81"
password = "Hexchip40123"
db = "etl"
port = 9030

engine = create_engine('mysql+pyaysql://[user]:[password]@[host]:[port]/[db]'.format(user = user,
password = password,
host = host,
port = port,
db = db))

df_run_case1 = pd.read_sql_query('SELECT * FROM etl.DWD_POC_CASE_RES_RUN_DATA_UNPIVOT_CASE_2 where parametric_name in ('BIASMATCH+GRUN

In [35]: def assign_label(row):
if row['WAFER_ID'] in good_wafer:
return 0
elif row['WAFER_ID'] in bad_wafer:
return 1
else:
return 'NOP'

In [36]: df_run_case1['label'] = df_run_case1.apply(assign_label, axis=1)

In [37]: df_run_case1.shape
Out[37]: (1894419, 20)

In [38]: df_run_case1['label'].value_counts()
Out[38]:
0      1027898
1       840092
NOP     26429
Name: label, dtype: int64

In [39]: df_run_case1 = df_run_case1[~(df_run_case1['label'] == 'NOP')].reset_index(drop=True)
df_run_case1.shape
Out[39]: (1867990, 20)

In [40]: print(df_run_case1[df_run_case1['label'] == 1]['WAFER_ID'].nunique())
print(df_run_case1[df_run_case1['label'] == 0]['WAFER_ID'].nunique())
188
388
```

```
In [23]: # df_run_case_uva = df_run_case1[df_run_case1['WAFER_ID'].isin(uva_wafer)].reset_index(drop=True)
# df_run_case_uva
```

```
In [43]: df_run_case_uva = df_run_case1
```

```
In [44]: print('数据维度是:', df_run_case_uva.shape)
print('PRODC1是:', df_run_case_uva['PRODC1'].unique())
print('TOOL_NAME是:', df_run_case_uva['TOOL_NAME'].unique())
print('OPER_NO是:', df_run_case_uva['OPER_NO'].unique())
print('parametric_name是:', df_run_case_uva['parametric_name'].unique())

数据维度是: (1867990, 20)
PRODC1是: ['C90WA14A' 'C90WA11A' 'C90WA12A' 'C90WA20A' 'C90WA07P' 'C90WA15A']
TOOL_NAME是: ['DSA02_C' 'DSA02_B' 'DSA02_A']
OPER_NO是: ['ZF.CDS10']
parametric_name是: ['BUFFER+FORELINE+PRESSURE' 'NF3+SIDE+PRESSURE' 'BIASMATCH+SERIES+SET'
'BIASMATCH+SHUNT+SET' 'BUFFER+PRESSURE' 'NF3+CLN+PRESSURE']
```

```
In [26]: def run_data_trace_plot(df, prod, oper, tool, para, value_range='all', wafer_num='all'):

# 查找同一个条件下对应的数据
df_certain_para = (df.query('PRODC1 == {} & OPER_NO == {} & TOOL_NAME == {} & parametric_name == {}'.format(prod, oper, tool, para))['WAFER_ID', 'OPER_NO', 'TOOL_NAME', 'PRODC1',
'TIME_STAMP', 'parametric_name', 'sensor_value', 'label'])

.sort_values('TIME_STAMP')
.reset_index(drop=True)
df_certain_para['TIME_STAMP'] = pd.to_datetime(df_certain_para['TIME_STAMP'], infer_datetime_format=True, format='%Y%m%d')
df_certain_para['sensor_value'] = df_certain_para['sensor_value'].astype(float)

# 根据给定的value_range抽取对应的数据
if value_range == 'all':
    df_certain_para = df_certain_para
else:
    df_certain_para = df_certain_para.query(value_range)

# 根据wafer_num, 在good和bad中分别抽取对应数量的数据
if wafer_num == 'all':
    df_certain_para = df_certain_para
else:
    label1_data = df_certain_para[df_certain_para['label'] == 1]
    label0_data = df_certain_para[df_certain_para['label'] == 0]

    random.seed(123)
    selected_wafer_label1 = random.sample(label1_data['WAFER_ID'].unique().tolist(), wafer_num)
    selected_wafer_label0 = random.sample(label0_data['WAFER_ID'].unique().tolist(), wafer_num)
    selected_data = pd.concat([label1_data[label1_data['WAFER_ID'].isin(selected_wafer_label1)],
    label0_data[label0_data['WAFER_ID'].isin(selected_wafer_label0)]])
    df_certain_para = selected_data

legend_elements = [
    Line2D([0], [0], color='deeppink', label=f'Bad Wafer: {df_certain_para[df_certain_para['label'] == 1]['WAFER_ID'].nunique()}',
    Line2D([0], [0], color='royalblue', label=f'Good Wafer: {df_certain_para[df_certain_para['label'] == 0]['WAFER_ID'].nunique()}'
```

```
# 按照分组, 进行画图
plt.figure(figsize=(16, 6))
grouped = df_certain_para.groupby('WAFER_ID')
for name, group in grouped:
    grp = group.sort_values('TIME_STAMP').reset_index(drop=True)
    color = 'deeppink' if grp['label'].iloc[0] == 1 else 'royalblue'
    linewidth = 1 if grp['label'].iloc[0] == 1 else 3
    alpha = 0.5 if grp['label'].iloc[0] == 1 else 1

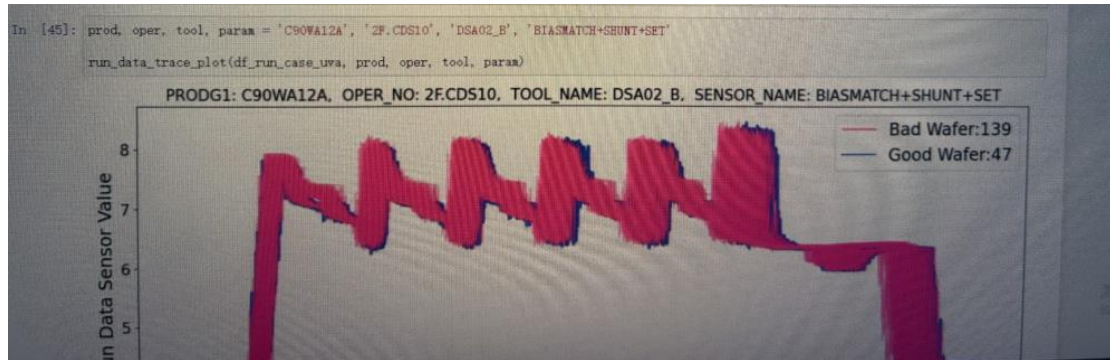
    plt.plot(grp['sensor_value'], color=color, linewidth=linewidth, alpha=alpha)

plt.xlabel('Time Index', size=18)
plt.xticks(size=15)
plt.ylabel('Run Data Sensor Value', size=18)
plt.yticks(size=15)
plt.legend(handles=legend_elements, fontsize='xx-large')

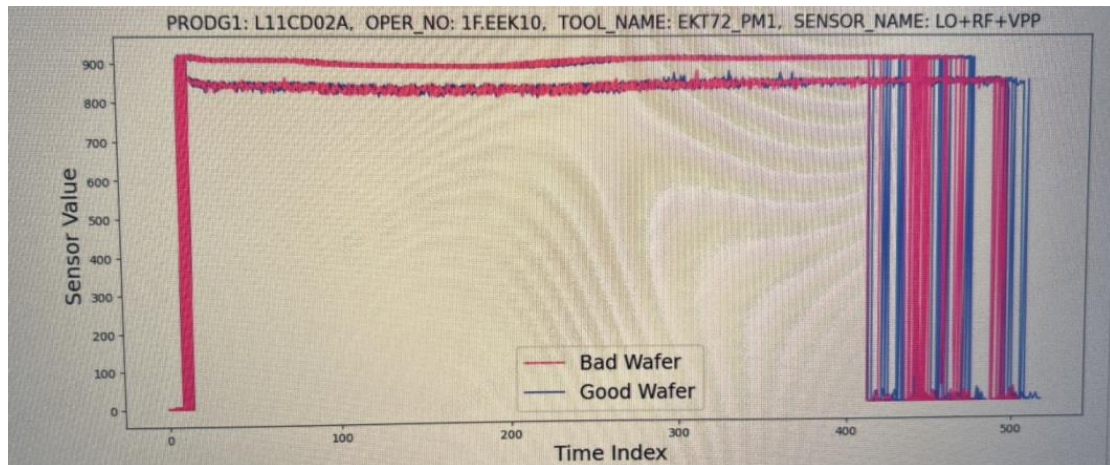
plt.legend(handles=legend_elements, fontsize='xx-large')
plt.title('PRODC1: {}, OPER_NO: {}, TOOL_NAME: {}, SENSOR_NAME: {}'.format(prod, oper, tool, para), size=15)
plt.show()
```



例：在产品为 C90WA12A，站点为 2F.CDS10，腔室为 DSA02\_B，参数为 BIASMATCH+SHUNT+SET，所有 WAFER 在特定时间段内的曲线图



例：在产品为 L11CD02A，站点为 1F.EEK10，腔室为 EKT72\_PM1，参数为 LO+RF+VPP，所有 WAFER 在特定时间段内的曲线图



可以观察到有很多“毛刺”的地方聚焦在  $Sensor\ Value \geq 800$ ，按照相同的条件，只画出  $Sensor\ Value \geq 800$  的图：

