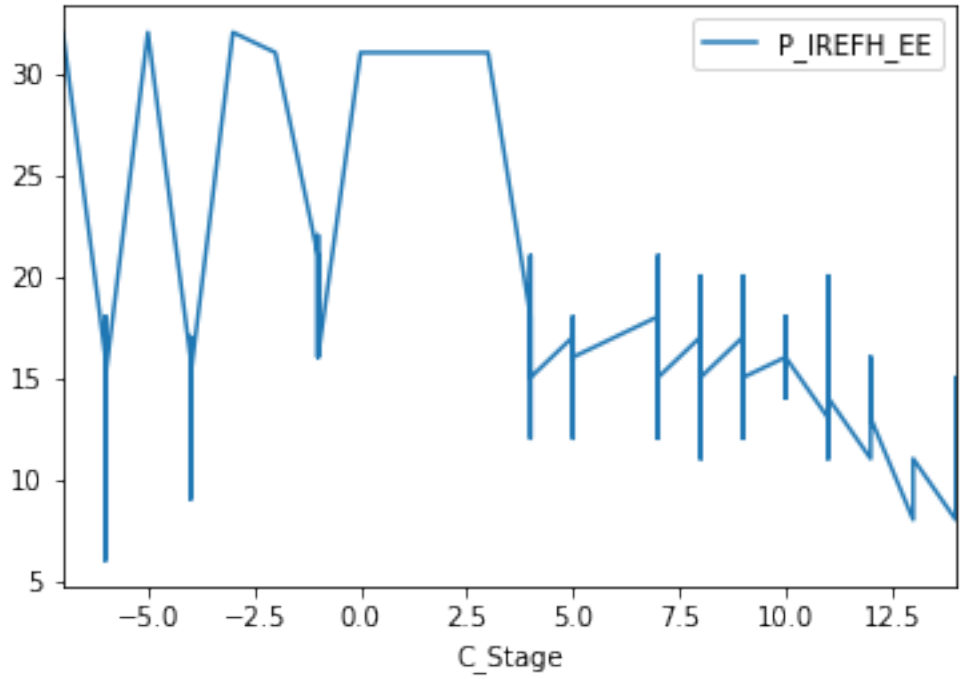
MelexisTask

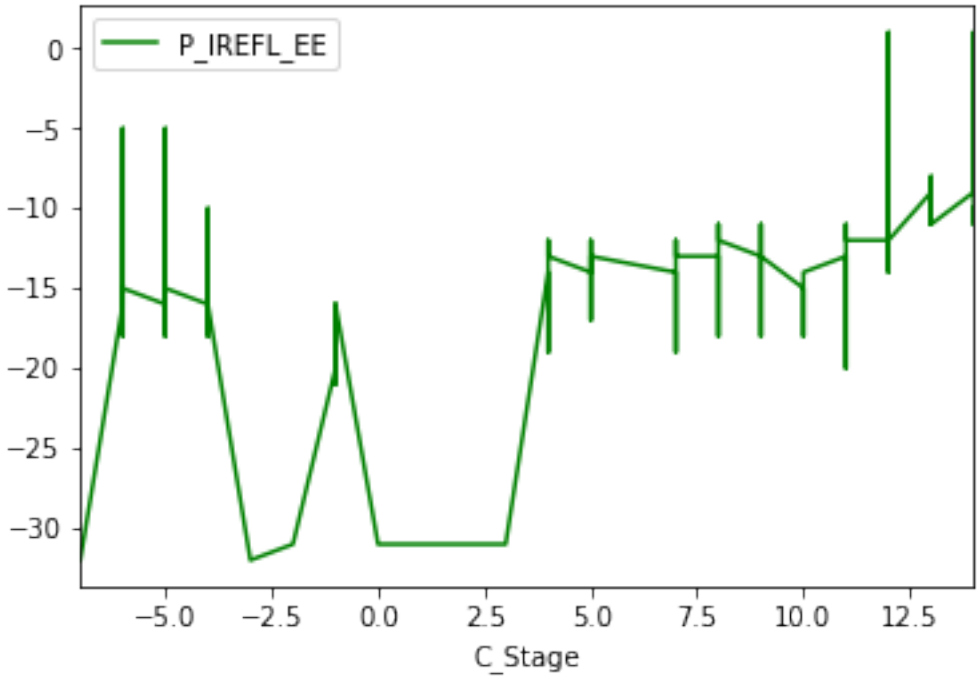
May 15, 2019

0.1 Python3 script that process and visualize data given in "Combined_data.csv"

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import configparser
```

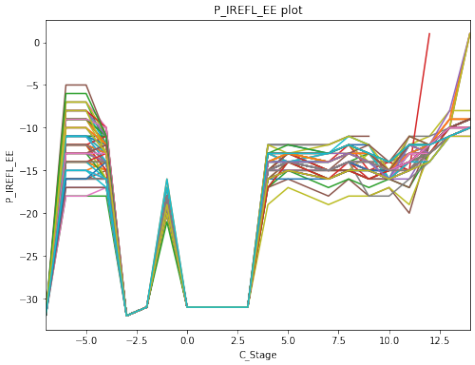
[2]:
```
datFrame = pd.read_csv("Combined_data.csv")
#datFrame.replace({"No_Device": np.nan})
datFrame[datFrame.P_IREFL_EE == "No_Device"] = np.nan
#datFrame = datFrame.dropna()

conf = configparser.ConfigParser()
conf.read('config.ini',encoding='utf-8')

datFrame = datFrame[pd.notnull(datFrame[conf['SimpleValues']['FirstGraph']])]
datFrame = datFrame[pd.notnull(datFrame[conf['SimpleValues']['SecGraph']])]

datFrame[conf['SimpleValues']['FirstGraph']] = pd.
 ↪to_numeric(datFrame[conf['SimpleValues']['FirstGraph']])
datFrame[conf['SimpleValues']['SecGraph']] = pd.
 ↪to_numeric(datFrame[conf['SimpleValues']['SecGraph']])

#datFrame = datFrame['P_IREFL_EE'].notnull()

P_IREFL_EE = datFrame.plot.line(x = "C_Stage", y =␣
 ↪conf['SimpleValues']['FirstGraph'])
P_IREFH_EE = datFrame.plot.line(x = "C_Stage", y =␣
 ↪conf['SimpleValues']['SecGraph'], color="green")
```

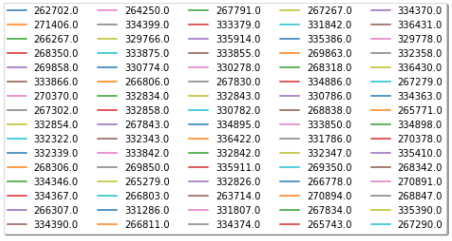

P_IREFH_EE
30
25
20
15
10
5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
C_Stage


P_IREFL_EE
0
−5
−10
−15
−20
−25
−30
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
C_Stage

0.2 Plot of P_IREFL versus a production stage number

[3]:
```
#datFrame
graph = plt.gca()
plt.figure(1)
temp_set = set() #in order to not draw repeatings chips more than one time
temp_set = set(datFrame.ChipID)
for col in datFrame.ChipID:
    if col in temp_set:
        temp1 = datFrame[datFrame.ChipID == col]
        temp1.plot.line(x = "C_Stage", y = conf['SimpleValues']['SecGraph'],␣
↪title =conf['SimpleValues']['SecGraph'] + " plot", figsize={8,6},label = col,␣
↪ax = graph)
        temp_set.remove(col)

plt.ylabel(conf['SimpleValues']['SecGraph'])
chartBox = graph.get_position()
graph.set_position([chartBox.x0, chartBox.y0, chartBox.width, chartBox.height])
graph.legend(loc='upper center', bbox_to_anchor=(2, 1), shadow=True, ncol=5)
plt.show()
```

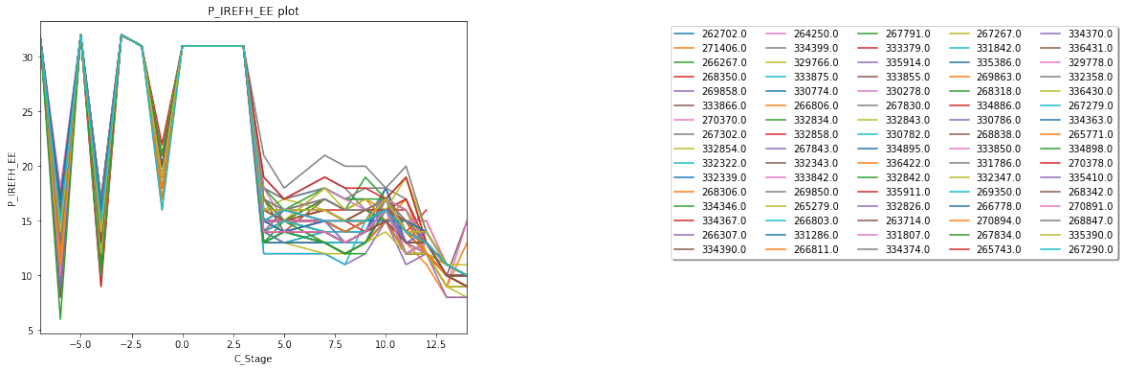




0.3 Plot of P_IREFH versus a production stage number

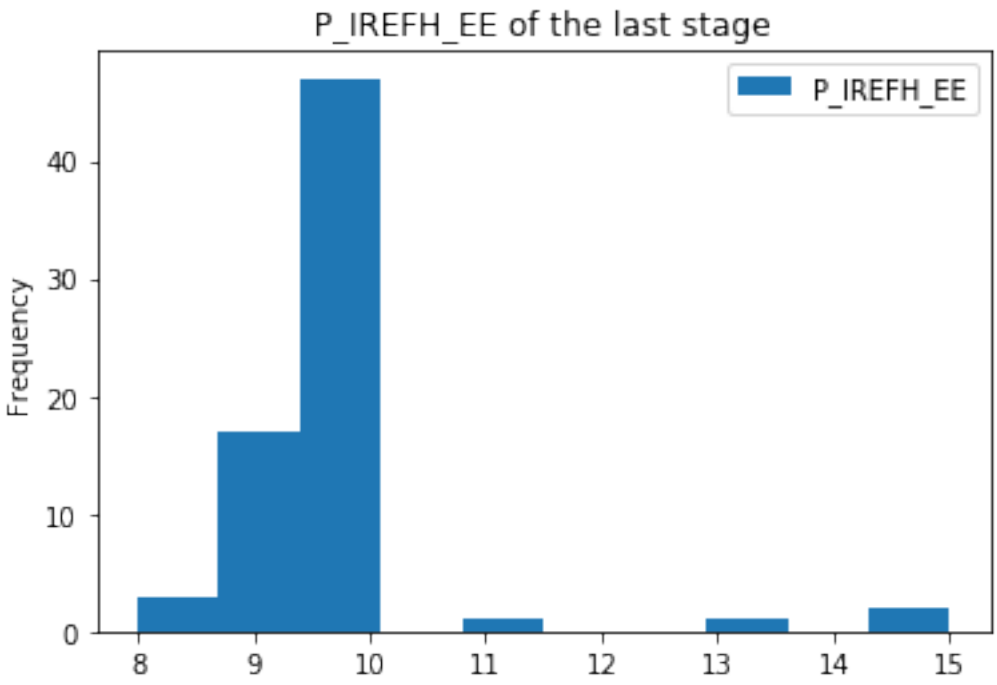
[12]:
```
plt.figure(2)
graph2 = plt.gca()
temp_set = set(datFrame.ChipID)
for col in datFrame.ChipID:
    if col in temp_set:
        temp2 = datFrame[datFrame.ChipID == col]
        temp2.plot.line(x = "C_Stage", y = conf['SimpleValues']['FirstGraph'],␣
↪title =conf['SimpleValues']['FirstGraph']+ " plot", figsize={8,6}, label =␣
↪col, ax = graph2)
        temp_set.remove(col)
```

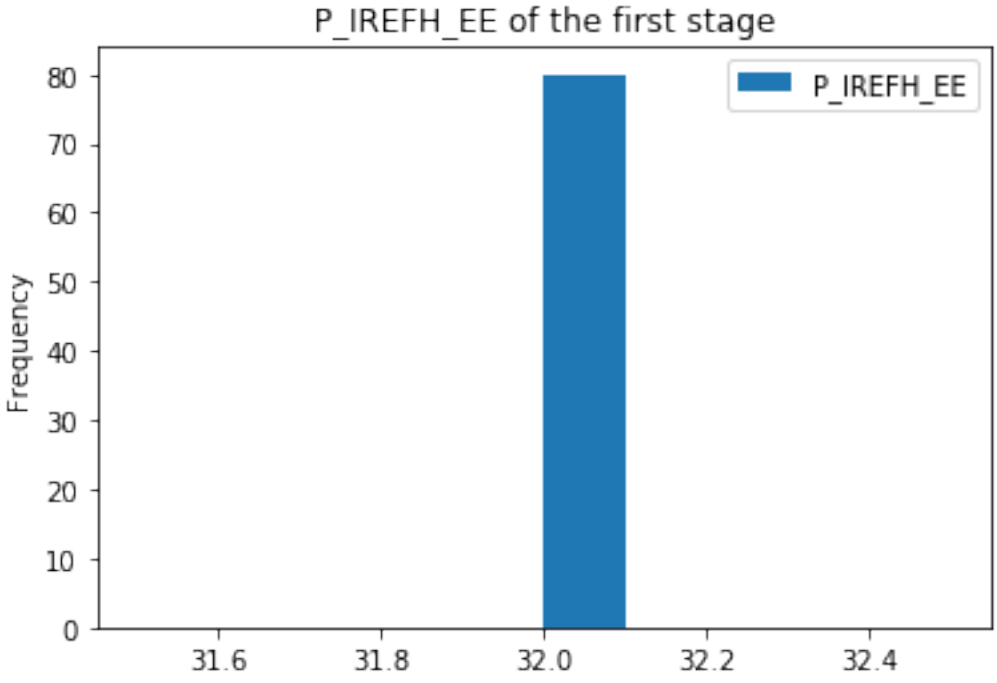
```
plt.ylabel(conf['SimpleValues']['FirstGraph'])
chartBox = graph2.get_position()
graph2.set_position([chartBox.x0, chartBox.y0, chartBox.width, chartBox.height])
graph2.legend(loc='upper center', bbox_to_anchor=(2, 1), shadow=True, ncol=5)
plt.show()
```



0.3.1 Histogram of the P_IREFL_EE P_IREFH_EE values across all the chips on the first and the last stage of production

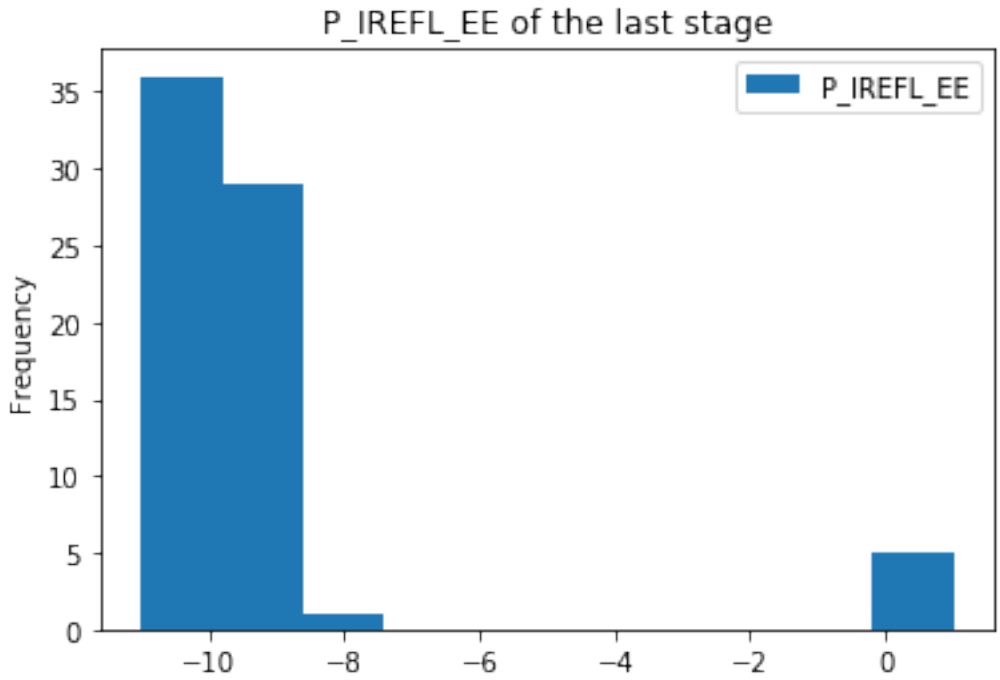
[5]:
```
df_temp = datFrame[datFrame.C_Stage == datFrame.C_Stage.max()]
df_temp.plot.hist(y=conf['SimpleValues']['FirstGraph'], x="C_Stage", title =␣
↪conf['SimpleValues']['FirstGraph'] + " of the last stage")
df_temp1 = datFrame[datFrame.C_Stage == datFrame.C_Stage.min()]
df_temp1.plot.hist(y=conf['SimpleValues']['FirstGraph'], x="C_Stage", title =␣
↪conf['SimpleValues']['FirstGraph'] + " of the first stage")
plt.show()
```

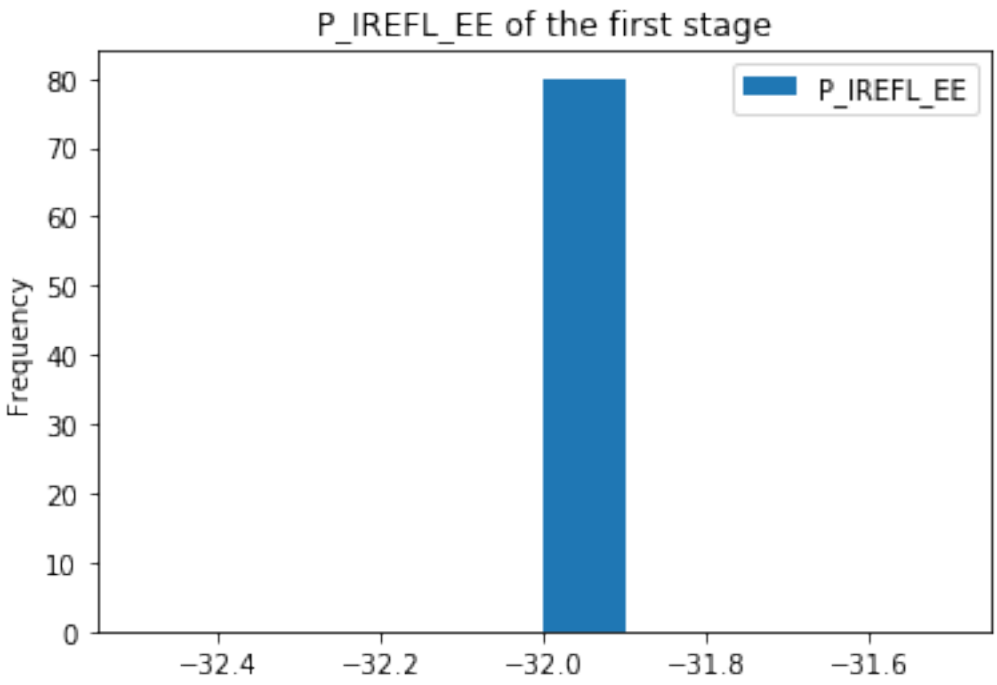




[6]:
```
df_temp2 = datFrame[datFrame.C_Stage == datFrame.C_Stage.max()]
```

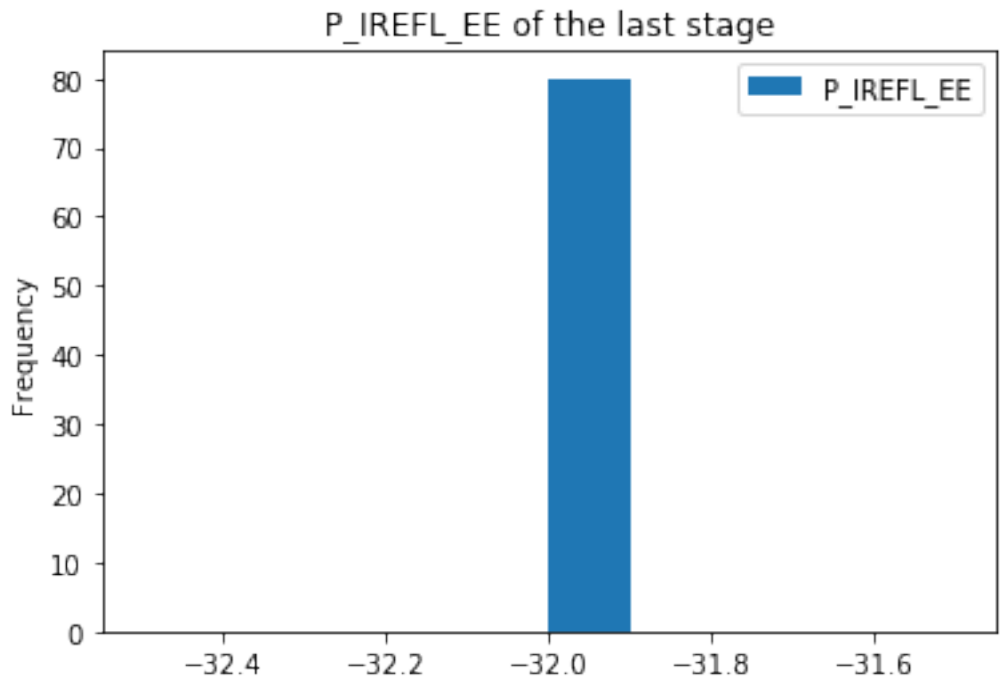
```
df_temp2.plot.hist( y=conf['SimpleValues']['SecGraph'], x="C_Stage", title =␣
 ↪conf['SimpleValues']['SecGraph'] + " of the last stage")
df_temp3 = datFrame[datFrame.C_Stage == datFrame.C_Stage.min()]
df_temp3.plot.hist(y=conf['SimpleValues']['SecGraph'], x="C_Stage", title =␣
 ↪conf['SimpleValues']['SecGraph'] + " of the first stage")
plt.show()
```





0.4 Here you can alter number of stages, to build histogram

[11]:
```
df_temp3 = datFrame[datFrame.C_Stage == float(conf['Hist']['stage_num'])]
df_temp3.plot.hist( y=conf['SimpleValues']['SecGraph'], x="C_Stage", title =␣
 ↪conf['SimpleValues']['SecGraph'] + " of the last stage")
plt.show()
```



0.5 Use set to define ChipID that u want to work with

[10]:
```
temp_set = {331842.0,335386.0,269863.0,268318.0,334886.0,330786.0,268838.
 ↪0,333850.0,331786.0,332347.0,269350.0,266778.0,270894.0,267834.0,265743.
 ↪0,334370.0}
df = pd.DataFrame()
csv_df = pd.DataFrame()
for numb in temp_set:
    temp = datFrame.loc[datFrame.ChipID == numb]
    df = df.append(temp, ignore_index = True)
# data to work with is stored i csv-file
#uncomment if you need csv
# csv_df = df
# csv_df.to_csv("File.csv")
df
```

[10]:

	ChipID	C_Bake[h]	C_Stage	C_Temp	C_Cond	P_DeviceName	\
0	331842.0	NaN	-7.0	35.0	PR1	54	
1	331842.0	NaN	-6.0	150.0	PR2	54	
2	331842.0	NaN	-5.0	150.0	PR2	54	
3	331842.0	NaN	-4.0	150.0	PR3	54	
4	331842.0	NaN	-3.0	-40.0	FT1	54	
5	331842.0	NaN	-2.0	-40.0	FT1	54	
6	331842.0	NaN	-1.0	125.0	FT2	54	

7	331842.0	NaN	0.0	35.0	INIT	NaN
8	331842.0	NaN	1.0	35.0	RELIABILITY35	54
9	331842.0	NaN	2.0	150.0	RELIABILITY150	54
10	331842.0	NaN	3.0	-40.0	RELIABILITYm40	54
11	331842.0	NaN	4.0	35.0	AFTER_SOLDER	54
12	331842.0	NaN	5.0	85.0	FL_ENDURANCE85	54
13	331842.0	NaN	7.0	35.0	AFTER_FL_EDR	54
14	331842.0	NaN	8.0	35.0	PRG_HTOL	54
15	331842.0	0.0	9.0	35.0	AFTER_EE_EDR	54
16	331842.0	0.0	10.0	150.0	AFTER_EE_EDR	54
17	331842.0	0.0	11.0	-40.0	AFTER_EE_EDR	54
18	331842.0	20.0	12.0	35.0	HTOL 20h	54
19	331842.0	500.0	13.0	35.0	HTOL 500h	54
20	331842.0	500.0	14.0	-40.0	HTOL 500h	54
21	330786.0	NaN	-7.0	35.0	PR1	59
22	330786.0	NaN	-6.0	150.0	PR2	59
23	330786.0	NaN	-5.0	150.0	PR2	59
24	330786.0	NaN	-4.0	150.0	PR3	59
25	330786.0	NaN	-3.0	-40.0	FT1	59
26	330786.0	NaN	-2.0	-40.0	FT1	59
27	330786.0	NaN	-1.0	125.0	FT2	59
28	330786.0	NaN	0.0	35.0	INIT	NaN
29	330786.0	NaN	1.0	35.0	RELIABILITY35	59
..	...	...	...	...	...	...
306	332347.0	NaN	5.0	85.0	FL_ENDURANCE85	64
307	332347.0	NaN	7.0	35.0	AFTER_FL_EDR	64
308	332347.0	NaN	8.0	35.0	PRG_HTOL	64
309	332347.0	0.0	9.0	35.0	AFTER_EE_EDR	64
310	332347.0	0.0	10.0	150.0	AFTER_EE_EDR	64
311	332347.0	0.0	11.0	-40.0	AFTER_EE_EDR	64
312	332347.0	20.0	12.0	35.0	HTOL 20h	64
313	332347.0	500.0	13.0	35.0	HTOL 500h	64
314	332347.0	500.0	14.0	-40.0	HTOL 500h	64
315	268318.0	NaN	-7.0	35.0	PR1	57
316	268318.0	NaN	-6.0	150.0	PR2	57
317	268318.0	NaN	-5.0	150.0	PR2	57
318	268318.0	NaN	-4.0	150.0	PR3	57
319	268318.0	NaN	-3.0	-40.0	FT1	57
320	268318.0	NaN	-2.0	-40.0	FT1	57
321	268318.0	NaN	-1.0	125.0	FT2	57
322	268318.0	NaN	0.0	35.0	INIT	NaN
323	268318.0	NaN	1.0	35.0	RELIABILITY35	57
324	268318.0	NaN	2.0	150.0	RELIABILITY150	57
325	268318.0	NaN	3.0	-40.0	RELIABILITYm40	57
326	268318.0	NaN	4.0	35.0	AFTER_SOLDER	57
327	268318.0	NaN	5.0	85.0	FL_ENDURANCE85	57
328	268318.0	NaN	7.0	35.0	AFTER_FL_EDR	57

329	268318.0	NaN	8.0	35.0	PRG_HTOL	57
330	268318.0	0.0	9.0	35.0	AFTER_EE_EDR	57
331	268318.0	0.0	10.0	150.0	AFTER_EE_EDR	57
332	268318.0	0.0	11.0	-40.0	AFTER_EE_EDR	57
333	268318.0	20.0	12.0	35.0	HTOL 20h	57
334	268318.0	500.0	13.0	35.0	HTOL 500h	57
335	268318.0	500.0	14.0	-40.0	HTOL 500h	57

	P_IREFL_EE	P_IREFH_EE	P_IREFL_FLASH	P_IREFH_FLASH
0	-32	32	-26	9
1	-11	11	-28	13
2	-11	32	-14	16
3	-12	13	-14	12
4	-32	32	-24	8
5	-31	31	NaN	8
6	-19	19	NaN	NaN
7	-31	31	NaN	NaN
8	-31	31	NaN	NaN
9	-31	31	NaN	NaN
10	-31	31	NaN	NaN
11	-13	13	NaN	NaN
12	-14	13	-32	17
13	-13	13	-35	10
14	-12	13	-38	15
15	-13	13	-31	17
16	-16	16	-25	18
17	-12	12	-33	14
18	-12	13	-29	17
19	-10	10	-21	17
20	-9	10	-22	15
21	-32	32	-22	11
22	-8	9	-24	13
23	-8	32	-12	17
24	-11	10	-13	11
25	-32	32	-20	9
26	-31	31	NaN	12
27	-19	20	NaN	NaN
28	-31	31	NaN	NaN
29	-31	31	NaN	NaN
..	...	...	...	...
306	-13	13	-31	20
307	-12	12	-35	14
308	-11	12	-37	18
309	-12	13	-30	19
310	-15	14	-24	21
311	-11	12	-32	15
312	-12	12	-27	20

313	-10	11	-18	19
314	-9	10	-19	15
315	-32	32	-24	9
316	-9	10	-26	13
317	-9	32	-15	15
318	-11	12	-15	11
319	-32	32	-24	8
320	-31	31	NaN	12
321	-18	21	NaN	NaN
322	-31	31	NaN	NaN
323	-31	31	NaN	NaN
324	-31	31	NaN	NaN
325	-31	31	NaN	NaN
326	-17	17	NaN	NaN
327	-14	15	-26	14
328	-16	17	-31	9
329	-15	16	-33	13
330	-16	16	-29	16
331	-15	15	-24	17
332	-14	15	-31	12
333	-13	14	-27	16
334	-11	11	-20	15
335	-10	10	-23	12

[336 rows x 10 columns]