

```
In [73]: # imports
import pandas as pd
import agrilution_aws
import logging
import boto3
from datetime import datetime
import sys
from boto3.dynamodb.conditions import Key, Attr
import time
from agrilution_aws import DynamoDbApi
from matplotlib.pyplot import figure
from matplotlib import pyplot as plt
import seaborn as sns
import plotly.express as px
import dask.dataframe as dd
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from plotly import tools
import warnings
warnings.filterwarnings('ignore')
pd.options.mode.chained_assignment = None
import dask.dataframe as dd
import datetime
```

```
In [2]: # globals

# dynamoDB API
dynamo = DynamoDbApi(logging.getLogger(), table_name = 'archive')
# timestamp in ms marking 1st of feb
timestamp1 = 1643673600000
# timestamp in ms marking 12th of may
timestamp2 = 1652313600000
```

In [3]: *# List of all Lab cubes*

```
plantcubes = {
    'A1_lower': 'd6472f5d-94f9-4a31-9a8e-ddc6744023d6',
    'A1_upper': 'bf6b3065-a5ad-49f0-96e3-f1ed22e55e18',
    'A2_lower': '07b17561-3b04-4094-a8ab-2f67315adfd',
    'A2_upper': '2ba34bbe-1611-4c9b-8a5e-1c802ff77768',
    'A3_lower': '26b03d30-3a9d-4460-a0cb-7ef5c1d5dec8',
    'A3_upper': '955605fe-8666-449b-96b4-e973b1e197da',
    'A4_lower': '09ef2ce0-2f99-45cf-8cb5-99550fca494f',
    'A4_upper': 'b637f6a6-b6e2-486c-86db-cc431d0b2a58',
    'B1_lower': '5a9039ae-957b-42b2-9d09-3baf73cf0020',
    'B1_upper': '0b66fd54-465b-409f-838f-ca5e494e68fb',
    'B2_lower': 'd9dd3086-fe92-4cab-b235-be2b283c4999',
    'B2_upper': '2853d150-f30a-4f35-a4fc-5985b35876dc',
    'B3_lower': 'a27588d5-bc01-44ab-b96d-cad7f86402b0',
    'B3_upper': 'd22ff6af-211b-4743-a5ae-5fd89ffbe446',
    'B4_lower': '11c45cd6-8d1f-4140-a545-0db886918e3b',
    'B4_upper': '510d7df1-234c-46f8-a153-ec792edc93b1',
    'C1_lower': '0427a2fa-8a50-4d00-ad56-6246c03ef9d0',
    'C1_upper': 'eac52b39-02c0-4a7a-a9e5-010709ee15c8',
    'C2_lower': 'ab713fff-4bd2-4a72-afdd-603e31b57689',
    'C2_upper': '09aefdec-f638-4e2d-91d2-375094a3d881',
    'C3_lower': '8cb8a481-a70d-4988-b419-d905d06ca65d',
    'C3_upper': '7d53b428-7777-47f0-9605-01ac8bda96f4',
    'C4_lower': '1acd7d04-fb3b-4983-abbf-24053e3a1499',
    'C4_upper': '5b23e086-1365-48a2-af39-defa77768aa5',
    'D1_lower': '5ae3a1b3-5354-4b23-ab83-aa9f3029098d',
    'D1_upper': '820b0870-b586-45b8-9a1e-fdd41a842f5d',
    'D2_lower': 'd183f2bd-d1df-4f83-a34d-6c72601b97f2',
    'D2_upper': '69a5e2a3-624c-4522-b0ee-ee28846fc700',
    'D3_lower': 'f598f96e-b0f4-4009-85e1-e621e8306c36',
    'D3_upper': '9788f724-0b7a-47ae-8e95-2c35152e20b8',
    'E1' : '2933af4a-51d4-4894-aa60-753219ca1918',
    'E2' : 'f29ffb36-be56-46e1-9e9d-d05e44e9a1a0',
    'E3' : 'b2d1811e-dbff-4fcb-a219-468adfb045ea',
    'E4' : '422d6453-a501-4ed9-bd4d-02b510a6e6d7',
    'E5' : '2b9c5df5-e286-4f0a-ab87-2271535677b6',
    'E6' : '12652341-6356-4c7f-9a60-eb5d82b16a57',
    'E7' : '52fdc759-32a3-43da-8207-3e4b89bafaae',
    'E8' : '424b5b0a-724f-4ab6-9688-f3fb2fab1cef',
    'E9' : 'dfde8871-522f-4ee5-a572-82049fe112cd',
    'E10' : 'c9299fd6-a636-4487-a252-837399139e8e',
```

```
}
```

```
In [4]: # List of data frames for each single cube between feb and may
frames = []
# iterate through plantcubes
for cube in plantcubes.values():
    # query_table of agrutils package already queries until no next token is given anymore
    resp = dynamo.query_table(
        KeyConditionExpression = (
            Key('plantcube').eq(cube) &
            Key('timestamp').between(timestamp1,timestamp2)
        )
    )
    # convert to pandas data frame
    df = pd.DataFrame(resp)
    # attach to list of all data frames
    frames.append(df)
# create one big data frame for all cubes
all_cubes= pd.concat(frames)
```

```
In [5]: all_cubes.head()
```

Out[5]:

	temp_led_a	temp_b	plantcube	timestamp	fan_b	fan_a	fan_a_tacho	fan_b_tacho	cooling	humid_b	...	verbose_reporting	recipe_id	mode	led_a_board_state	stage	lev
0	30	24.33	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673600867	10	1	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	
1	31	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673602071	15	6	1020	1290	True	NaN	...	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605873	NaN	NaN	810	1230	NaN	90	...	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605957	NaN	NaN	NaN	NaN	NaN	91	...	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673606028	NaN	NaN	NaN	NaN	NaN	92	...	NaN	NaN	NaN	NaN	NaN	

5 rows × 46 columns



```
In [6]: all_cubes.to_csv('feb-may-dn')
```

```
In [ ]: all_cubes = pd.read_csv('feb-may-dn')
```

```
In [ ]: df2 = all_cubes.copy()
```

In [10]: `all_cubes.columns`

Out[10]: Index(['temp_led_a', 'temp_b', 'plantcube', 'timestamp', 'fan_b', 'fan_a', 'fan_a_tacho', 'fan_b_tacho', 'cooling', 'humid_b', 'temp_a', 'rssi', 'temp_led_b', 'led_a', 'led_b', 'humid_a', 'light_b', 'light_a', 'fan_led_a_tacho', 'fan_led_b', 'fan_led_b_tacho', 'fan_led_a', 'temp_tank', 'pump', 'valve', 'tank_level_raw', 'ec', 'wifi_level', 'connected', 'firmware_ncu', 'door', 'total_offset', 'user_button', 'signal_led', 'tank_level', 'firmware_mcu', 'verbose_reporting', 'recipe_id', 'mode', 'led_a_board_state', 'stage', 'led_b_board_state', 'owner', 'user_offset', 'plants', 'seedbundle_variant'], dtype='object')

In [11]: *#extracting the required columns*
`cols = [1,2,3,10,28,37,38]`
`all_cubes = all_cubes[all_cubes.columns[cols]]`

In [12]: `all_cubes.head()`

Out[12]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
0	24.33	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673600867	NaN	NaN	NaN	NaN
1	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673602071	NaN	NaN	NaN	NaN
2	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605873	NaN	NaN	NaN	NaN
3	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605957	NaN	NaN	NaN	NaN
4	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673606028	NaN	NaN	NaN	NaN

In [13]: `all_cubes.dropna(subset=['connected','recipe_id','temp_b','temp_a','mode'], how='all', inplace=True)`

In [14]: *#converting timestamp to datetime format*
`all_cubes['timestamp'] = all_cubes['timestamp'].astype('int64')`
`all_cubes['timestamp'] = pd.to_datetime(all_cubes['timestamp'], unit='ms')`

```
In [15]: #replacing the plantcube name with their alias names
dict1 = {v : k for k, v in plantcubes.items()}
all_cubes.plantcube = all_cubes.plantcube.replace(dict1)
all_cubes.head()
```

Out[15]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
0	24.33	A1_lower	2022-02-01 00:00:00.867	NaN	NaN	NaN	NaN
7	NaN	A1_lower	2022-02-01 00:00:12.864	22.74	NaN	NaN	NaN
15	24.43	A1_lower	2022-02-01 00:00:38.146	NaN	NaN	NaN	NaN
24	24.53	A1_lower	2022-02-01 00:01:31.859	22.84	NaN	NaN	NaN
85	NaN	A1_lower	2022-02-01 00:06:27.499	22.74	NaN	NaN	NaN

```
In [16]: all_cubes = all_cubes.reset_index()
```

```
In [17]: #applying ffill for the columns connected and recipe id
all_cubes['connected'] = all_cubes.groupby('plantcube')['connected'].apply(lambda x:x.fillna(method='ffill'))
all_cubes['recipe_id'] = all_cubes.groupby('plantcube')['recipe_id'].apply(lambda x:x.fillna(method='ffill'))
all_cubes['mode'] = all_cubes.groupby('plantcube')['mode'].apply(lambda x:x.fillna(method='ffill'))
```

```
In [18]: #after forward filling the columns 'connected' and 'recipe id' in the above step. Remove the rows if any of these column
#values are null.
all_cubes.dropna(subset=['connected','recipe_id'], how='any', inplace=True)
```

```
In [20]: all_cubes = all_cubes.drop('index', axis=1)
```

```
In [21]: #changing the datatype of the temperature columns
all_cubes['temp_a'] = all_cubes['temp_a'].astype(float)
all_cubes['temp_b'] = all_cubes['temp_b'].astype(float)
```

```
In [22]: idata = all_cubes.copy()
```

```
In [23]: #dropping the rows which contains the mode(debug)  
t1 = idata[idata['mode'] == 1]
```

```
In [24]: #total records in debug mode  
t1.shape[0]
```

Out[24]: 3940

```
In [25]: #remove the records in debug mode  
idata = idata[idata['mode'] != 1]
```

```
In [26]: idata.head()
```

Out[26]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
9158	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1	NaN
9159	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1	0
9160	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1640171855	0
9161	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1640171855	0
9162	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1640171855	0

```
In [27]: #adding recipe along with it.  
rdf = pd.read_csv('Recipe_table_sJan')
```

```
In [28]: rdf.drop('Unnamed: 0', axis=1, inplace=True)
```

In [29]:

rdf.head(30)

Out[29]:

		layers	plantcube	recipe_id
0	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1640171700	
1	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1640171806	
2	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1640171855	
3	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1644837004	
4	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1644931116	
5	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1649666148	
6	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1649854144	
7	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1649934487	
8	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1650446765	
9	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1650462922	
10	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1653983191	
11	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1653983554	
12	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1653985982	
13	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1654078825	
14	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1657093174	
15	[[{'periods': [{'duration': Decimal('86400'), ...	A1_lower	1657281884	
16	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1640074948	
17	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1640171708	
18	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1643099474	
19	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1643292942	
20	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1645094620	
21	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1646732433	
22	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1652711767	
23	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1652711832	
24	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1652711870	

		layers	plantcube	recipe_id
25	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1653900414	
26	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1654078815	
27	[[{'periods': [{'duration': Decimal('86400'), ...	A1_upper	1659961675	
28	[[{'periods': [{'duration': Decimal('86400'), ...	A2_lower	1638954537	
29	[[{'periods': [{'duration': Decimal('86400'), ...	A2_lower	1640171729	

```
In [30]: #joining the recipe and archive table based on the attributes plantcube and recipe_id
jdf = pd.merge(idata, rdf, on=['plantcube','recipe_id'], how="left",indicator=True)
```

```
In [31]: jdf.head()
```

Out[31]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	_merge
0	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1	NaN	NaN	left_only
1	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1	0	NaN	left_only
2	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both
3	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both
4	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both

```
In [32]: #recipe_id - 1 is the default recipe. It is just used to reset the cube. It doesn't mean anything.
jdf.loc[jdf.recipe_id == 1, 'layers'] = "default recipe"
jdf.head()
```

Out[32]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	_merge
0	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1	NaN	default recipe	left_only
1	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1	0	default recipe	left_only
2	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both
3	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both
4	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...	both

```
In [33]: jdf.drop('_merge', axis=1, inplace=True)
```

```
In [34]: jdf.head()
```

Out[34]:

	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers
0	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1	NaN	default recipe
1	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1	0	default recipe
2	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...
3	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...
4	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1640171855	0	[[{'periods': [{'duration': Decimal('86400'), ...

```
In [35]: jdf['layers'] = jdf.groupby('plantcube')['layers'].apply(lambda x:x.fillna(method='ffill'))
```

```
In [36]: jdf['recipe']= jdf['layers'].map(str)
#changing the datatype of column 'recipe' to category
jdf['recipe']= jdf['recipe'].astype('category')
#converting the values in the column 'recipe' to numerical codes
jdf['recipe'] = jdf['recipe'].cat.codes
```

```
In [37]: idata1 = jdf.copy()
```

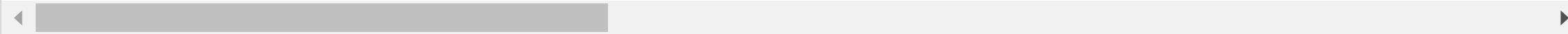
```
In [38]: idata1.recipe.unique()
```

```
Out[38]: array([14,  9, 12, 13,  7,  5,  3,  4,  2,  6, 11, 15,  1,  0, 10,  8],  
             dtype=int8)
```

```
In [39]: idata1.to_csv("febMay12.05-preprocessed")
```

```
In [40]: idata1 = pd.read_csv('febMay12.05-preprocessed')
```

```
In [41]: dataframes = ['A1_lower', 'A1_upper', 'A2_lower', 'A2_upper', 'A3_lower', 'A3_upper', 'A4_lower', 'A4_upper', 'B1_lower', 'B1_upper', 'B2_lower', 'B2_upper', 'B3_
```



```
In [42]: val = ""
def my_func(ndf):
    val = ndf
    # creating a dataframe to store the plantcube
    df = idata1[idata1.plantcube == val]
    df.head(10)

    #set the timestamp as index
    df['timestamp'] = pd.to_datetime(df['timestamp'])
    df = df.set_index('timestamp')

    #interpolating the temperature values based on linear interpolation method after resampling it by date.
    df['temp_a'] = df.resample('D')['temp_a'].apply(lambda x:x.interpolate(method="time",limit_direction = "forward"))
    df['temp_b'] = df.resample('D')['temp_b'].apply(lambda x:x.interpolate(method="time",limit_direction = "forward"))

    #converting connected as category type
    df['connected'] = df['connected'].astype('category')

    #instead of true and false, converting it into 0's and 1's
    df['connected'] = df['connected'].cat.codes

    return df

res = []
for dataframe in dataframes:
    data = my_func(dataframe)
    res.append(data)
res1= pd.concat(res)
```

```
In [43]: res1
```

Out[43]:

	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe
timestamp									
2022-02-07 07:54:22.697	0	NaN	A1_lower	NaN	1	1	NaN	default recipe	14
2022-02-07 07:54:22.724	1	NaN	A1_lower	NaN	1	1	0.0	default recipe	14
2022-02-07 07:54:22.772	2	NaN	A1_lower	NaN	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
2022-02-07 07:54:24.010	3	21.740000	A1_lower	20.03	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
2022-02-07 07:58:58.207	4	21.681092	A1_lower	19.93	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
...
2022-05-11 23:56:20.499	3755587	21.590000	E10	18.21	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
2022-05-11 23:56:51.497	3755588	21.490000	E10	18.21	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
2022-05-11 23:57:27.497	3755589	21.390000	E10	18.21	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
2022-05-11 23:58:12.506	3755590	21.290000	E10	18.21	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
2022-05-11 23:59:23.494	3755591	21.190000	E10	18.21	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10

3755592 rows × 9 columns

```
In [44]: res1.recipe.unique()
```

Out[44]: array([14, 9, 12, 13, 7, 5, 3, 4, 2, 6, 11, 15, 1, 0, 10, 8],
dtype=int64)

```
In [66]: recipe8 = res1[res1.recipe == 8]
recipe8.layers.values[0]
```

```
Out[66]: "[[{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('0'), Decimal('57600')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]], [{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('0'), Decimal('57600')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]"]
```

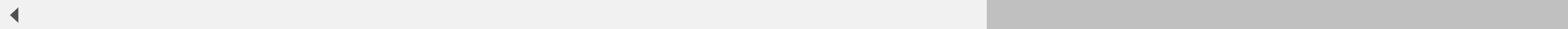
```
In [67]: recipe9 = res1[res1.recipe == 9]
recipe9.layers.values[0]
```

```
Out[67]: "[[{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('0'), Decimal('57600')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]], [{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('0'), Decimal('57600')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]"]
```

```
In [68]: recipe10 = res1[res1.recipe == 10]
recipe10.layers.values[0]
```

```
Out[68]: "[[{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('70'), Decimal('57530')], 'light': [Decimal('61'), Decimal('39'), Decimal('33'), Decimal('10')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]], [{'periods': [{'duration': Decimal('86400'), 'temp': Decimal('23'), 'watering': [Decimal('60'), Decimal('17940')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('1')}, {'periods': [{'duration': Decimal('57600'), 'temp': Decimal('23'), 'watering': [Decimal('70'), Decimal('57530')], 'light': [Decimal('61'), Decimal('39'), Decimal('33'), Decimal('10')]}], {'duration': Decimal('28800'), 'temp': Decimal('21'), 'watering': [Decimal('0'), Decimal('28800')], 'light': [Decimal('0'), Decimal('0'), Decimal('0'), Decimal('0')]}], 'cycles': Decimal('60')}]"]
```

In [69]: 3_upper', 'C4_lower', 'C4_upper', 'D1_lower', 'D1_upper', 'D2_lower', 'D2_upper', 'D3_lower', 'D3_upper', 'E1', 'E2', 'E3', 'E4', 'E5', 'E6', 'E7', 'E8', 'E9', 'E10']



```
In [70]: #function to find out when the plantcube starts and ends the recipe
def my_func(dfn):
    val = dfn
    dfn1 = idata1[idata1.plantcube == val]
    current_recipe = None
    start_plantcube = None
    start_time = None
    time = None
    result = []
    for recipe, time ,plantcube in zip(dfn1['recipe'], dfn1['timestamp'],dfn1['plantcube']):
        if recipe != current_recipe:
            if current_recipe is not None and start_plantcube is not None and start_time is not None and time is not None:
                result.append([start_plantcube, current_recipe, start_time, time])
                current_recipe, start_time,start_plantcube = recipe, time,plantcube

    result.append([start_plantcube, current_recipe, start_time, time])
    ddata1 = pd.DataFrame(result, columns=['plantcube','recipe','RecipeStartTime','RecipeEndTime'])
    return(ddata1)

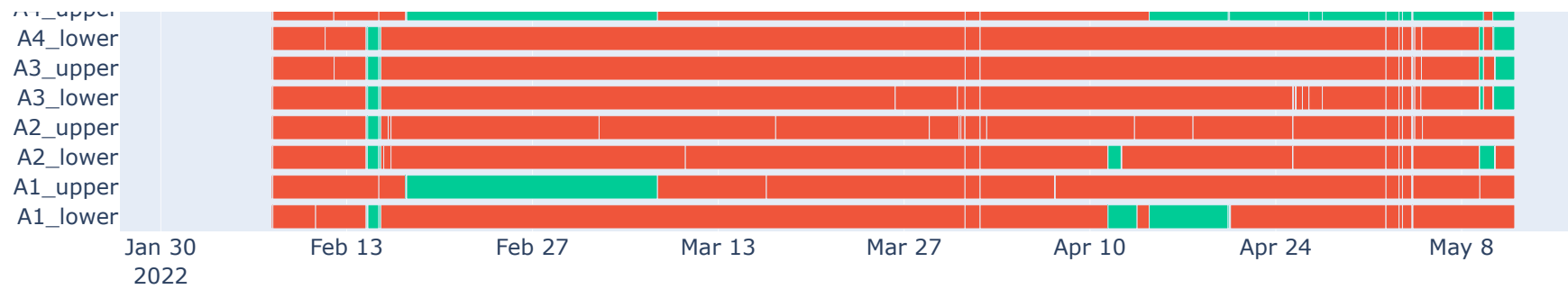
resn = []
for dataframe in dataframes:
    dataframe = my_func(dataframe)
    resn.append(dataframe)
resn1= pd.concat(resn)
print(resn1)
```

	plantcube	recipe	RecipeStartTime		RecipeEndTime	
0	A1_lower	14	2022-02-07	07:54:22.697	2022-02-07	07:54:22.772
1	A1_lower	9	2022-02-07	07:54:22.772	2022-02-07	10:32:03.169
2	A1_lower	14	2022-02-07	10:32:03.169	2022-02-07	10:32:03.395
3	A1_lower	9	2022-02-07	10:32:03.395	2022-02-07	10:33:30.587
4	A1_lower	14	2022-02-07	10:33:30.587	2022-02-07	10:33:30.660
..	
17	E10	14	2022-03-31	15:12:18.594	2022-03-31	15:12:18.675
18	E10	8	2022-03-31	15:12:18.675	2022-04-01	17:18:10.927
19	E10	14	2022-04-01	17:18:10.927	2022-04-01	17:18:11.013
20	E10	8	2022-04-01	17:18:11.013	2022-05-10	12:05:06.905
21	E10	10	2022-05-10	12:05:06.905	2022-05-11	23:59:23.494

[1563 rows x 4 columns]


```
In [74]: resn1['recipe'] = resn1['recipe'].astype(str)
fig = px.timeline(
    resn1, x_start="RecipeStartTime", x_end="RecipeEndTime", y="plantcube",
    color='recipe', height=800, width=1000
)
fig.show()
```





```
In [ ]: #recipe 8,9,10 - runs day for 16 hrs(23 degrees),night for 8hrs(21 degrees). Rest of the recipes doesn't have proper day and
#night cycle.
```

```
In [79]: res1.recipe.unique()
```

```
Out[79]: array([14,  9, 12, 13,  7,  5,  3,  4,  2,  6, 11, 15,  1,  0, 10,  8],
              dtype=int64)
```

```
In [86]: res2 = res1[(res1.recipe == 8)|(res1.recipe == 9)|(res1.recipe == 10)]
```

```
In [88]: res2.plantcube.unique()
```

```
Out[88]: array(['A1_lower', 'A1_upper', 'A2_lower', 'A2_upper', 'A3_lower',
               'A3_upper', 'A4_lower', 'A4_upper', 'B1_lower', 'B1_upper',
               'B2_lower', 'B2_upper', 'B3_upper', 'B3_lower', 'B4_lower',
               'B4_upper', 'C1_lower', 'C1_upper', 'C2_lower', 'C2_upper',
               'C3_lower', 'C3_upper', 'C4_lower', 'C4_upper', 'D1_lower',
               'D1_upper', 'D2_lower', 'D2_upper', 'D3_lower', 'D3_upper', 'E1',
               'E2', 'E3', 'E4', 'E6', 'E9', 'E10'], dtype=object)
```

```
In [ ]: #During the duration, all the plantcubes from february to 12-05 running the same user_offset except c3_lower,c4_lower,d3_lower
```

```
In [89]: res2 = res2.reset_index()
```

```
In [92]: r1 = res2[(res2['plantcube'] != 'C3_lower')&(res2['plantcube'] != 'C4_lower')&(res2['plantcube'] != 'D3_lower')]
```

```
In [93]: r1
```

Out[93]:

	timestamp	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe
0	2022-02-07 07:54:22.772	2	NaN	A1_lower	NaN	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
1	2022-02-07 07:54:24.010	3	21.740000	A1_lower	20.030000	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
2	2022-02-07 07:58:58.207	4	21.681092	A1_lower	19.930000	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
3	2022-02-07 08:00:38.209	5	21.659608	A1_lower	19.830000	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
4	2022-02-07 08:02:09.480	6	21.640000	A1_lower	19.769666	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
...
3083047	2022-05-11 23:56:20.499	3755587	21.590000	E10	18.210000	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
3083048	2022-05-11 23:56:51.497	3755588	21.490000	E10	18.210000	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
3083049	2022-05-11 23:57:27.497	3755589	21.390000	E10	18.210000	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
3083050	2022-05-11 23:58:12.506	3755590	21.290000	E10	18.210000	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10
3083051	2022-05-11 23:59:23.494	3755591	21.190000	E10	18.210000	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10

2990511 rows × 10 columns

```
In [94]: #plantcubes with user offset 32400(9:00)
# 9a.m to 1a.m - day(16 hr)
# 1a.m to 9a.m - night(8 hr)
#If the timestamp is from 10 a.m to 12 a.m, then the dayflag will be set to True
r1['dayflag'] = ((r1.timestamp.dt.hour >= 10 ) & (r1.timestamp.dt.hour <= 23))
#If the timestamp is from 2 a.m to 8 a.m, then the nightflag will be set to True
r1['nightflag'] = ((r1.timestamp.dt.hour >= 2 ) & (r1.timestamp.dt.hour <= 8))
```

```
In [96]: final_r = r1.copy()
```

```
In [97]: daynight = final_r[(final_r.nightflag == True)|(final_r.dayflag == True)]
```

```
In [98]: #day temperature
daydf = daynight[daynight.dayflag == True]
nightdf = daynight[daynight.nightflag == True]
```

```
In [99]: daydf = daydf.round(3)
daydf
```

Out[99]:

	timestamp	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode		layers	recipe	dayflag	nightflag
132	2022-02-07 10:00:08.689	134	23.940	A1_lower	21.667	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		9	True	False
133	2022-02-07 10:00:44.681	135	23.932	A1_lower	21.630	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		9	True	False
134	2022-02-07 10:02:17.754	136	23.912	A1_lower	21.530	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		9	True	False
135	2022-02-07 10:04:30.665	137	23.883	A1_lower	21.430	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		9	True	False
136	2022-02-07 10:06:44.670	138	23.854	A1_lower	21.330	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		9	True	False
...
3083047	2022-05-11 23:56:20.499	3755587	21.590	E10	18.210	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		10	True	False
3083048	2022-05-11 23:56:51.497	3755588	21.490	E10	18.210	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		10	True	False
3083049	2022-05-11 23:57:27.497	3755589	21.390	E10	18.210	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		10	True	False
3083050	2022-05-11 23:58:12.506	3755590	21.290	E10	18.210	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		10	True	False
3083051	2022-05-11 23:59:23.494	3755591	21.190	E10	18.210	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...		10	True	False

2073577 rows × 12 columns

```
In [100]: #for night dataframe- target should be 21
nightdf = nightdf.round(3)
nightdf
```

Out[100]:

	timestamp	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe	dayflag	nightflag
0	2022-02-07 07:54:22.772	2	NaN	A1_lower	NaN	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	False	True
1	2022-02-07 07:54:24.010	3	21.740	A1_lower	20.030	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	False	True
2	2022-02-07 07:58:58.207	4	21.681	A1_lower	19.930	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	False	True
3	2022-02-07 08:00:38.209	5	21.660	A1_lower	19.830	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	False	True
4	2022-02-07 08:02:09.480	6	21.640	A1_lower	19.770	1	1640171855	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	False	True
...
3082348	2022-05-11 08:52:27.745	3754888	23.690	E10	21.578	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10	False	True
3082349	2022-05-11 08:53:03.643	3754889	23.590	E10	21.558	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10	False	True
3082350	2022-05-11 08:53:47.651	3754890	23.490	E10	21.533	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10	False	True
3082351	2022-05-11 08:54:28.420	3754891	23.423	E10	21.510	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10	False	True
3082352	2022-05-11 08:54:48.159	3754892	23.390	E10	21.513	1	1652184304	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	10	False	True

621586 rows × 12 columns

```
In [103]: #Target temperature
#Daythreshold = 23
#Nightthreshold = 21

#day temperature average
dayavg = daydf.groupby('plantcube')['plantcube','temp_a','temp_b'].mean(numeric_only = True)
dayavg['deviation_temp_a'] = abs(dayavg['temp_a'] - 23)
dayavg['deviation_temp_b'] = abs(dayavg['temp_b'] - 23)
dayavg['diff - a&b'] = abs(dayavg['temp_a'] - dayavg['temp_b'])
dayavg = dayavg.round(3)
dayavg
```

Out[103]:

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					
A1_lower	22.652	23.863	0.348	0.863	1.211
A1_upper	23.014	23.484	0.014	0.484	0.470
A2_lower	22.744	23.702	0.256	0.702	0.959
A2_upper	22.787	23.849	0.213	0.849	1.062
A3_lower	22.848	23.245	0.152	0.245	0.397
A3_upper	23.049	23.964	0.049	0.964	0.916
A4_lower	22.945	23.686	0.055	0.686	0.741
A4_upper	22.930	23.661	0.070	0.661	0.731
B1_lower	23.001	23.741	0.001	0.741	0.741
B1_upper	22.906	23.874	0.094	0.874	0.968
B2_lower	22.778	23.730	0.222	0.730	0.952
B2_upper	22.708	23.981	0.292	0.981	1.274
B3_lower	23.061	23.507	0.061	0.507	0.446
B3_upper	22.321	23.879	0.679	0.879	1.558
B4_lower	22.990	23.280	0.010	0.280	0.290
B4_upper	23.344	21.450	0.344	1.550	1.895
C1_lower	22.778	22.971	0.222	0.029	0.194

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					
C1_upper	23.041	23.615	0.041	0.615	0.574
C2_lower	23.003	22.239	0.003	0.761	0.764
C2_upper	22.504	23.793	0.496	0.793	1.289
C3_upper	22.940	23.073	0.060	0.073	0.133
C4_upper	22.782	23.436	0.218	0.436	0.654
D1_lower	22.592	23.526	0.408	0.526	0.933
D1_upper	22.955	23.504	0.045	0.504	0.549
D2_lower	22.906	23.264	0.094	0.264	0.358
D2_upper	22.890	23.755	0.110	0.755	0.865
D3_upper	22.928	23.500	0.072	0.500	0.572
E1	23.008	23.876	0.008	0.876	0.868
E10	22.443	23.516	0.557	0.516	1.074
E2	22.765	23.821	0.235	0.821	1.056
E3	22.996	23.969	0.004	0.969	0.972
E4	23.335	23.385	0.335	0.385	0.050
E6	23.165	23.989	0.165	0.989	0.824
E9	23.134	21.686	0.134	1.314	1.449

```
In [102]: #night temperature average
nightavg = nightdf.groupby('plantcube')['plantcube','temp_a','temp_b'].mean(numeric_only = True)
nightavg['deviation_temp_a'] = abs(nightavg['temp_a'] - 21)
nightavg['deviation_temp_b'] = abs(nightavg['temp_b'] - 21)
nightavg['diff - a&b'] = abs(nightavg['temp_a'] - nightavg['temp_b'])
nightavg = nightavg.round(3)
nightavg
```

Out[102]:

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					
A1_lower	20.891	22.223	0.109	1.223	1.332
A1_upper	21.306	21.959	0.306	0.959	0.653
A2_lower	21.396	20.196	0.396	0.804	1.200
A2_upper	21.093	22.016	0.093	1.016	0.924
A3_lower	20.860	21.639	0.140	0.639	0.779
A3_upper	21.191	22.024	0.191	1.024	0.833
A4_lower	21.273	21.955	0.273	0.955	0.682
A4_upper	21.189	21.997	0.189	0.997	0.808
B1_lower	21.030	22.042	0.030	1.042	1.012
B1_upper	20.987	22.085	0.013	1.085	1.098
B2_lower	20.787	21.954	0.213	0.954	1.166
B2_upper	21.206	22.135	0.206	1.135	0.929
B3_lower	21.281	21.405	0.281	0.405	0.124
B3_upper	20.716	22.116	0.284	1.116	1.400
B4_lower	21.343	21.554	0.343	0.554	0.211
B4_upper	21.704	19.590	0.704	1.410	2.113
C1_lower	21.462	21.779	0.462	0.779	0.317
C1_upper	21.287	22.030	0.287	1.030	0.743
C2_lower	21.341	21.501	0.341	0.501	0.160
C2_upper	21.272	22.271	0.272	1.271	0.998

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					
C3_upper	21.156	19.479	0.156	1.521	1.677
C4_upper	21.154	22.039	0.154	1.039	0.886
D1_lower	20.900	21.991	0.100	0.991	1.091
D1_upper	21.203	21.793	0.203	0.793	0.590
D2_lower	19.526	21.441	1.474	0.441	1.915
D2_upper	21.281	22.064	0.281	1.064	0.783
D3_upper	21.357	22.013	0.357	1.013	0.656
E1	21.325	22.144	0.325	1.144	0.819
E10	20.798	21.855	0.202	0.855	1.057
E2	21.359	22.058	0.359	1.058	0.698
E3	21.491	22.100	0.491	1.100	0.609
E4	21.406	21.719	0.406	0.719	0.313
E6	21.510	22.170	0.510	1.170	0.661
E9	21.256	21.412	0.256	0.412	0.156

In [104]:

```
#day target - both sensors - above 0.8
higher_deviation_day = dayavg[(dayavg['deviation_temp_a']> 0.8) & (dayavg['deviation_temp_b']> 0.8)]
higher_deviation_day
```

Out[104]:

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					

```
In [105]: #night temperature - both sensors - above 0.8
higher_deviation_night = nightavg[(nightavg['deviation_temp_a']> 0.8) & (nightavg['deviation_temp_b']> 0.8)]
higher_deviation_night
```

Out[105]:

	temp_a	temp_b	deviation_temp_a	deviation_temp_b	diff - a&b
plantcube					

```
In [106]: #standard deviation
s1 = daydf.groupby('plantcube')['temp_a','temp_b'].std()
s1
```

Out[106]:

	temp_a	temp_b
plantcube		
A1_lower	0.539921	0.446205
A1_upper	0.437000	0.496939
A2_lower	0.602971	0.522254
A2_upper	0.514202	0.525382
A3_lower	0.452810	0.604986
A3_upper	0.434043	0.523331
A4_lower	0.434230	0.579249
A4_upper	0.487528	0.634397
B1_lower	0.369922	0.450801
B1_upper	0.446871	0.340694
B2_lower	0.504629	0.492981
B2_upper	0.486253	0.449122
B3_lower	0.433579	0.630927
B3_upper	0.607649	0.454784
B4_lower	0.410680	0.713000
B4_upper	0.988462	1.315448
C1_lower	0.543479	0.647379
C1_upper	0.452360	0.510089
C2_lower	0.448307	0.715250
C2_upper	0.608469	0.537058
C3_upper	0.391736	0.997818
C4_upper	0.616304	0.813882

	temp_a	temp_b
plantcube		
D1_lower	0.543360	0.643844
D1_upper	0.452681	0.547050
D2_lower	0.508494	0.484643
D2_upper	0.511869	0.500541
D3_upper	0.486980	0.730351
E1	0.421667	0.452898
E10	0.819512	0.616213
E2	0.491956	0.499402
E3	0.454149	0.518684
E4	0.457770	0.619187
E6	0.544421	0.688890
E9	0.289854	0.698365

```
In [107]: s2 = nightdf.groupby('plantcube')['temp_a','temp_b'].std()  
s2
```

Out[107]:

	temp_a	temp_b
plantcube		
A1_lower	1.003246	0.774404
A1_upper	0.979892	0.885315
A2_lower	1.134615	2.614526
A2_upper	0.936024	0.891650
A3_lower	1.670237	0.854271
A3_upper	0.980982	0.924380
A4_lower	0.818841	0.878324
A4_upper	0.868784	0.976329
B1_lower	0.839749	0.751424
B1_upper	0.855083	0.627632
B2_lower	1.479474	0.800261
B2_upper	0.869173	0.913697
B3_lower	0.869701	0.939022
B3_upper	1.028581	0.828151
B4_lower	0.905939	0.976785
B4_upper	1.256117	1.335685
C1_lower	0.996670	1.069036
C1_upper	1.001044	0.920542
C2_lower	1.067776	0.719354
C2_upper	1.059429	1.042680
C3_upper	1.151515	3.128848
C4_upper	1.201150	1.150596
D1_lower	1.106159	0.916028

	temp_a	temp_b
plantcube		
D1_upper	0.889556	0.986454
D2_lower	2.987034	0.838401
D2_upper	1.010068	0.977206
D3_upper	0.880936	0.950179
E1	1.054307	0.947139
E10	0.808557	0.650996
E2	0.995084	1.084294
E3	1.003593	0.989444
E4	0.961084	0.894690
E6	0.980688	1.074105
E9	1.001673	0.710542

In []: