In [1]:
```
# 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
from plotly.offline import init_notebook_mode, iplot
from plotly.graph_objs import *
init_notebook_mode(connected=True)
from tabulate import tabulate
from IPython.display import display
import plotly.io as pio
```

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 1st of may
timestamp2 = 1651363200000
```

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-2f67315adfdd',
    '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	...	firmware_mcu	verbose_reporting	recipe_id	mode	led_a_board_s
0	30	24.33	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673600867	10	1	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	
2	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605873	NaN	NaN	810	1230	NaN	90	...	NaN	NaN	NaN	NaN	
3	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673605957	NaN	NaN	NaN	NaN	NaN	91	...	NaN	NaN	NaN	NaN	
4	NaN	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673606028	NaN	NaN	NaN	NaN	NaN	92	...	NaN	NaN	NaN	NaN	

5 rows × 45 columns

In [6]:
```
all_cubes.columns
```

Out[6]:
```
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'],
      dtype='object')
```

In [8]:
```
#extracting the required columns
cols = [1,2,3,10,28,37,38]
df1 = all_cubes[all_cubes.columns[cols]]
```

In [9]:
```
df1.head()
```

Out[9]:

	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 [10]:
```
#storing the dataframe into csv
df1.to_csv('Feb-May.csv')
```

In [11]:
```
#reading the file as dask dataframe
#df2 = dd.read_csv('Feb-May.csv',dtype = 'unicode')
df2 = pd.read_csv('Feb-May.csv')
```

In [ ]:
```
#converting it into pandas dataframe
#df2 = df2.compute()
```

In [12]:
```
df2.head()
```

Out[12]:

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

In [13]:
```
#Number of Null values in each column
df2.isnull().sum()
```

Out[13]:
```
Unnamed: 0          0
temp_b        23585212
plantcube            0
timestamp            0
temp_a        23562565
connected     25416855
recipe_id     25419391
mode          25420126
dtype: int64
```

In [14]:
```
#dropping the row if all the values in the given columns are NA
df2.dropna(subset=['connected','recipe_id','temp_b','temp_a','mode'], how='all', inplace=True)
```

In [15]:
```
df2.head()
```

Out[15]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
0	0	24.33	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673600867	NaN	NaN	NaN	NaN
7	7	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673612864	22.74	NaN	NaN	NaN
15	15	24.43	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673638146	NaN	NaN	NaN	NaN
24	24	24.53	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673691859	22.84	NaN	NaN	NaN
85	85	NaN	d6472f5d-94f9-4a31-9a8e-ddc6744023d6	1643673987499	22.74	NaN	NaN	NaN

In [16]:
```
#converting timestamp to datetime format
df2['timestamp'] = df2['timestamp'].astype('int64')
df2['timestamp'] = pd.to_datetime(df2['timestamp'], unit='ms')
```

In [17]:
```
#replacing the plantcube name with their alias names
dict1 = {v : k for k, v in plantcubes.items()}
df2.plantcube = df2.plantcube.replace(dict1)
df2.head()
```

Out[17]:

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

In [18]:
```
df2 = df2.reset_index()
```

In [19]:
```
#applying ffill for the columns connected and recipe id
df2['connected'] = df2.groupby('plantcube')['connected'].apply(lambda x:x.fillna(method='ffill'))
df2['recipe_id'] = df2.groupby('plantcube')['recipe_id'].apply(lambda x:x.fillna(method='ffill'))
df2['mode'] = df2.groupby('plantcube')['mode'].apply(lambda x:x.fillna(method='ffill'))
```

In [20]:
```
df2.isnull().sum()
```

Out[20]:
```
index              0
Unnamed: 0         0
temp_b       1711013
plantcube          0
timestamp          0
temp_a       1688366
connected      16496
recipe_id     214126
mode          250205
dtype: int64
```

In [21]:
```
#after forward filling the columns 'connected' and 'recipe id' in the above step. Remove the rows if any of these column
#values are null.
df2.dropna(subset=['connected','recipe_id'], how='any', inplace=True)
```

In [22]:
```
df2 = df2.drop('index', axis=1)
```

In [23]:
```
#changing the datatype of the temperature columns
df2['temp_a'] = df2['temp_a'].astype(float)
df2['temp_b'] = df2['temp_b'].astype(float)
```

In [24]:
```
#copying dataframe df2 to idata
idata = df2.copy()
```

In [19]:
```
idata.drop('Unnamed: 0', axis=1, inplace=True)
```

In [25]:
```
#dropping the rows which contains the mode(debug)
t1 = idata[idata['mode'] == 1]
```

In [26]:
```
#total records in debug mode
t1.shape[0]
```

Out[26]: 2094

In [27]:
```
#remove the records in debug mode
idata = idata[idata['mode'] != 1]
```

In [28]:
```
idata.head()
```

Out[28]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
9158	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN
9159	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0
9160	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0
9161	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0
9162	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0

In [29]:
```
#idata = idata.reset_index()
```

In [30]:
```
#idata.head()
```

In [31]:
```
#idata.drop('index', axis=1, inplace=True)
```

In [32]:
```
idata.head()
```

Out[32]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode
9158	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN
9159	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0
9160	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0
9161	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0
9162	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0

In [33]:
```
#adding recipe along with it.
rdf = pd.read_csv('Recipe_table_sJan')
```

In [34]:
```
rdf.drop('Unnamed: 0', axis=1, inplace=True)
```

In [35]:
```
rdf.head(30)
```

Out[35]:

	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 [36]:
```
#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 [37]:
```
jdf.head()
```

Out[37]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	_merge
0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	NaN	left_only
1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	NaN	left_only
2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both
3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both
4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both

In [38]:
```
jdf.loc[jdf.recipe_id == 1, 'layers'] = "default recipe"
jdf.head()
```

Out[38]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	_merge
0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	default recipe	left_only
1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	default recipe	left_only
2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both
3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both
4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	both

In [39]:
```
jdf.drop('_merge', axis=1, inplace=True)
```

In [40]:
```
jdf.head()
```

Out[40]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers
0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	default recipe
1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	default recipe
2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...
3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...
4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...

In [41]:
```
jdf.isnull().sum()
```

Out[41]:
```
Unnamed: 0          0
temp_b        1599795
plantcube           0
timestamp           0
temp_a        1592075
connected           0
recipe_id           0
mode            35802
layers           9365
dtype: int64
```

In [42]:
```
jdf['layers'] = jdf.groupby('plantcube')['layers'].apply(lambda x:x.fillna(method='ffill'))
```

In [43]:
```
jdf.head()
```

Out[43]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers
0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	default recipe
1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	default recipe
2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...
3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...
4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...

In [44]:
```
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 [45]:
```
jdf.isnull().sum()
```

Out[45]:
```
Unnamed: 0          0
temp_b        1599795
plantcube           0
timestamp           0
temp_a        1592075
connected           0
recipe_id           0
mode            35802
layers              7
recipe              0
dtype: int64
```

In [46]:
```
idata1 = jdf.copy()
```

In [47]:
```
idata1.head()
```

Out[47]:

	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	recipe
0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	default recipe	14
1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	default recipe	14
2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9

In [48]:
```
idata1.recipe.unique()
```

Out[48]:
```
array([14,  9, 12, 13,  7,  5,  3,  4,  2,  6, 11, 15,  1,  0, 10,  8],
      dtype=int8)
```

In [49]:
```
idata1.to_csv("preprocessed-Feb_May")
```

In [50]:
```
idata1 = pd.read_csv("preprocessed-Feb_May")
```

In [51]:
```
idata1[idata1.plantcube == 'A1_lower']
```

Out[51]:

	Unnamed: 0.1	Unnamed: 0	temp_b	plantcube	timestamp	temp_a	connected	recipe_id	mode	layers	recipe
0	0	58675	NaN	A1_lower	2022-02-07 07:54:22.697	NaN	True	1.000000e+00	NaN	default recipe	14
1	1	58676	NaN	A1_lower	2022-02-07 07:54:22.724	NaN	True	1.000000e+00	0.0	default recipe	14
2	2	58677	NaN	A1_lower	2022-02-07 07:54:22.772	NaN	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
3	3	58680	21.74	A1_lower	2022-02-07 07:54:24.010	20.03	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
4	4	58701	NaN	A1_lower	2022-02-07 07:58:58.207	19.93	True	1.640172e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
...	...	...	...	...	...	...	...	...	...	...	...
103372	103372	690933	NaN	A1_lower	2022-04-29 15:59:56.445	22.58	True	1.650463e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
103373	103373	690936	NaN	A1_lower	2022-04-29 16:01:18.445	22.48	True	1.650463e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
103374	103374	690937	23.97	A1_lower	2022-04-29 16:01:48.439	NaN	True	1.650463e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
103375	103375	690943	NaN	A1_lower	2022-04-29 16:02:47.430	22.38	True	1.650463e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9
103376	103376	690944	NaN	A1_lower	2022-04-29 16:03:32.594	NaN	False	1.650463e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9

103377 rows × 11 columns

Plantcubes where sensors not working

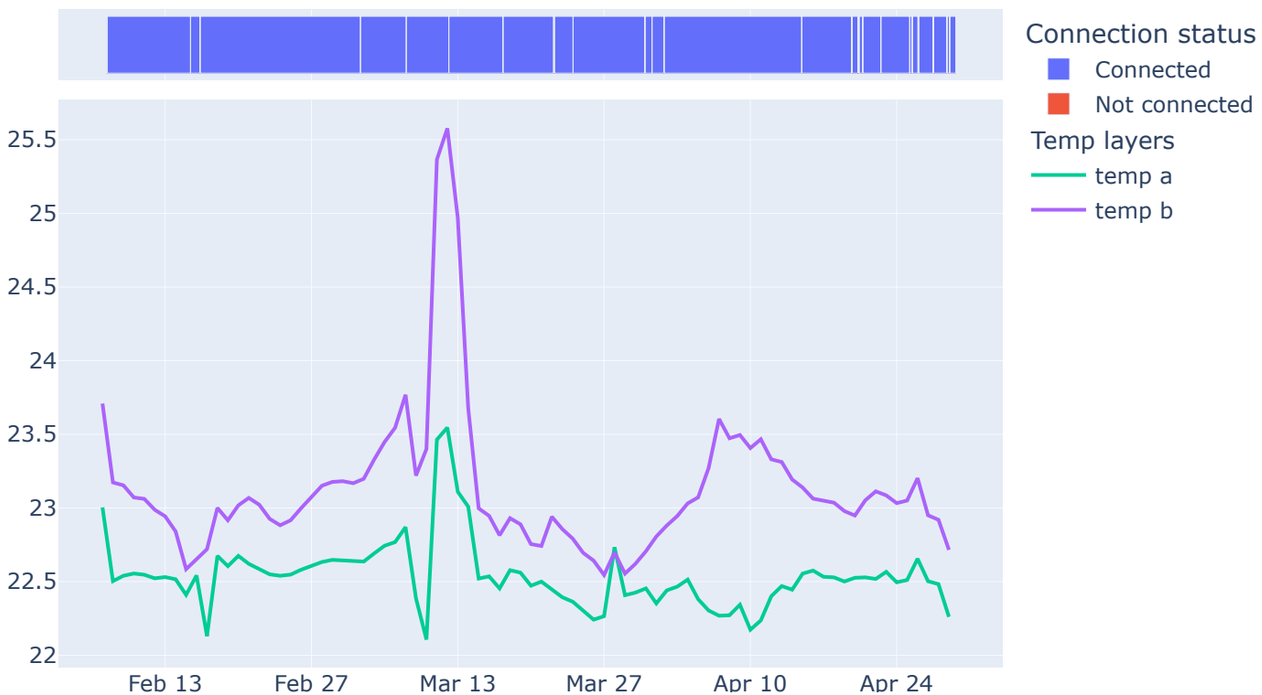
In [52]:
```
dataframes =['A1_lower','A1_upper','A2_lower','A2_upper','A3_lower','A3_upper','A4_lower','A4_upper','B1_lower','B1_upper','B2_lower','B2_upper','B3_
```

In [53]:
```
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="linear",limit_direction = "forward"))
    df['temp_b'] = df.resample('D')['temp_b'].apply(lambda x:x.interpolate(method="linear",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

    #function to find out when the connection starts and ends
    current_event = None
    start_plantcube = None
    start_time = None
    time = None
    result = []
    for event, time ,plantcube in zip(df['connected'], df.index,df['plantcube']):
        if event != current_event:
            if current_event 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_event, start_time, time])
            current_event, start_time,start_plantcube = event, time,plantcube

    result.append([start_plantcube, current_event, start_time, time])
    ddata = pd.DataFrame(result, columns=['plantcube','connected','EventStartTime','EventEndTime'])

    #converting connected attribute to string values
    ddata['connected'] = ddata['connected'].astype(str)
    ddata['connected'].replace('0','Not connected',inplace=True)
    ddata['connected'].replace('1','Connected',inplace=True)

    #visualization of A1 lower plantcube for two layers (temp a and temp b)
```

```
    a1l_a = df[['temp_a']].resample('D').mean()
    a1l_b = df[['temp_b']].resample('D').mean()
    fig = go.Figure()
    trace1 = go.Scatter(x=a1l_a.index, y=a1l_a.temp_a, name="temp a", mode="lines",legendgroup="group2",legendgrouptitle_text="Temp layers")
    trace2 = go.Scatter(x=a1l_b.index, y=a1l_b.temp_b, name="temp b", mode="lines",legendgroup="group2",legendgrouptitle_text="Temp layers")

    #visulaization of connection status
    fig1 = px.timeline(
     ddata, x_start="EventStartTime", x_end="EventEndTime", y="plantcube",
     color='connected', height=200, width=1000
    )

    fig = tools.make_subplots(rows=2, cols=1,
                              figure=fig1,
                              shared_xaxes=True,
                              vertical_spacing=0.03,
                              row_width=[0.4, 0.05]
                             )
    fig.add_trace(trace1, row=2, col=1)
    fig.add_trace(trace2, row=2, col=1)
    fig.update_layout(xaxis2_showticklabels=True,height=500, width=750,showlegend=True,yaxis1={'visible': False, 'showticklabels': False},legend={"ti
    fig.show()
    return df

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

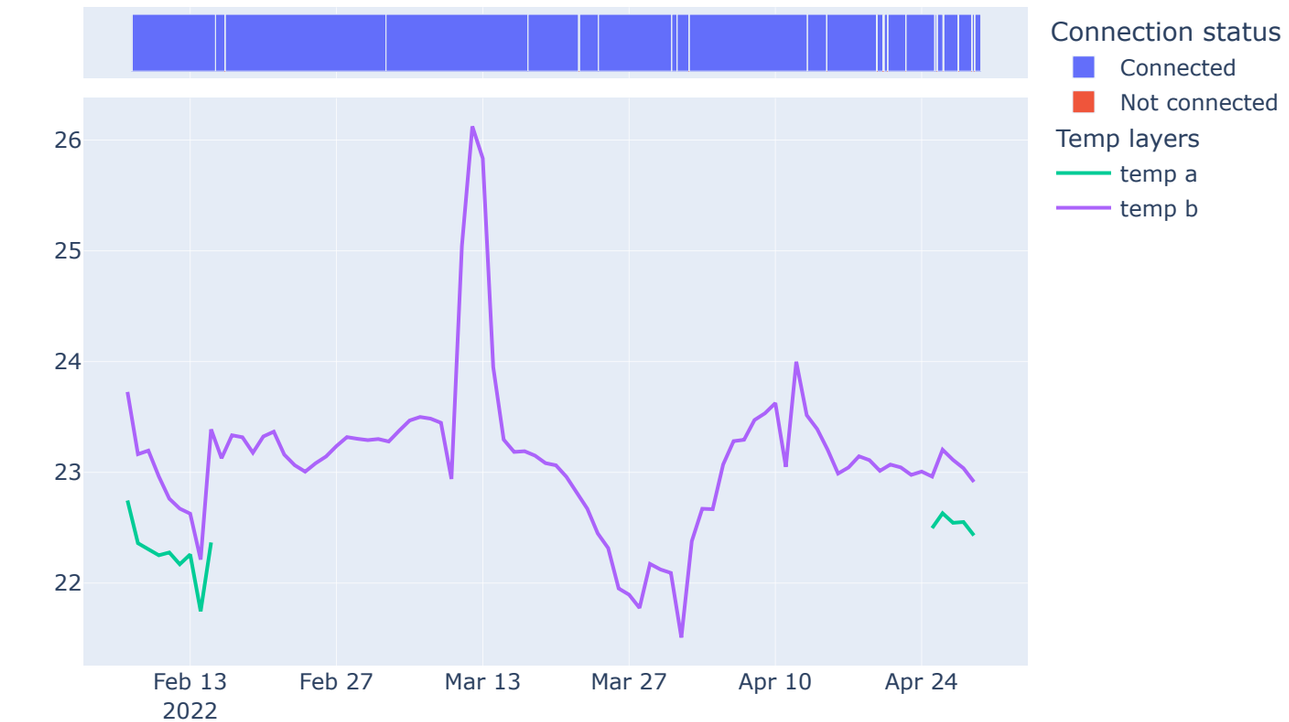

23.5
23
22.5
22
21.5
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

A1_upper

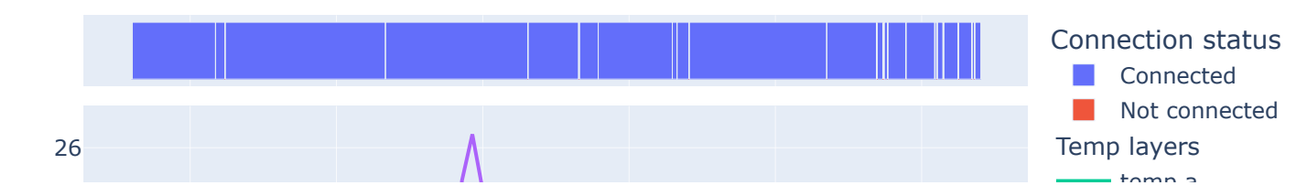

Connection status
Connected
Not connected
Temp layers
temp a
temp b
25.5
25
24.5
24
23.5
23
22.5
22
Feb 13
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

2022

A2_lower



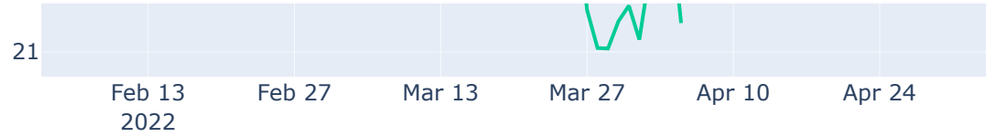
A2_upper



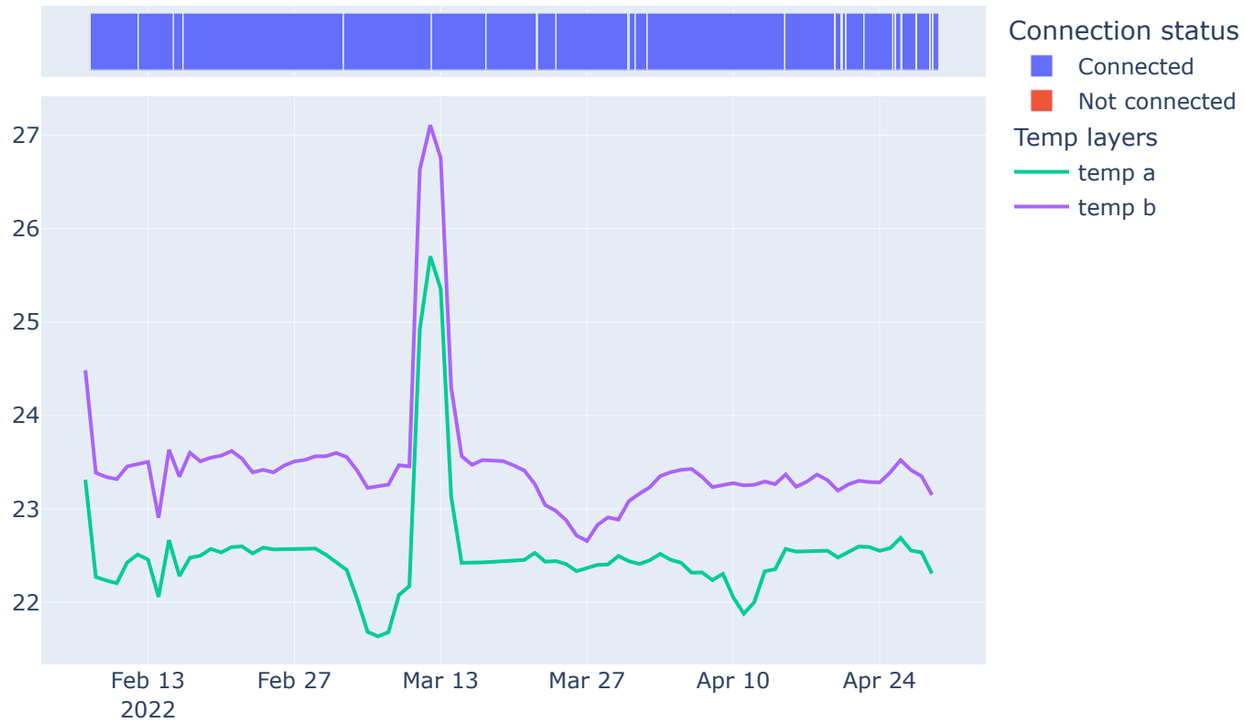

temp a
temp b
25
24
23
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

A3_lower

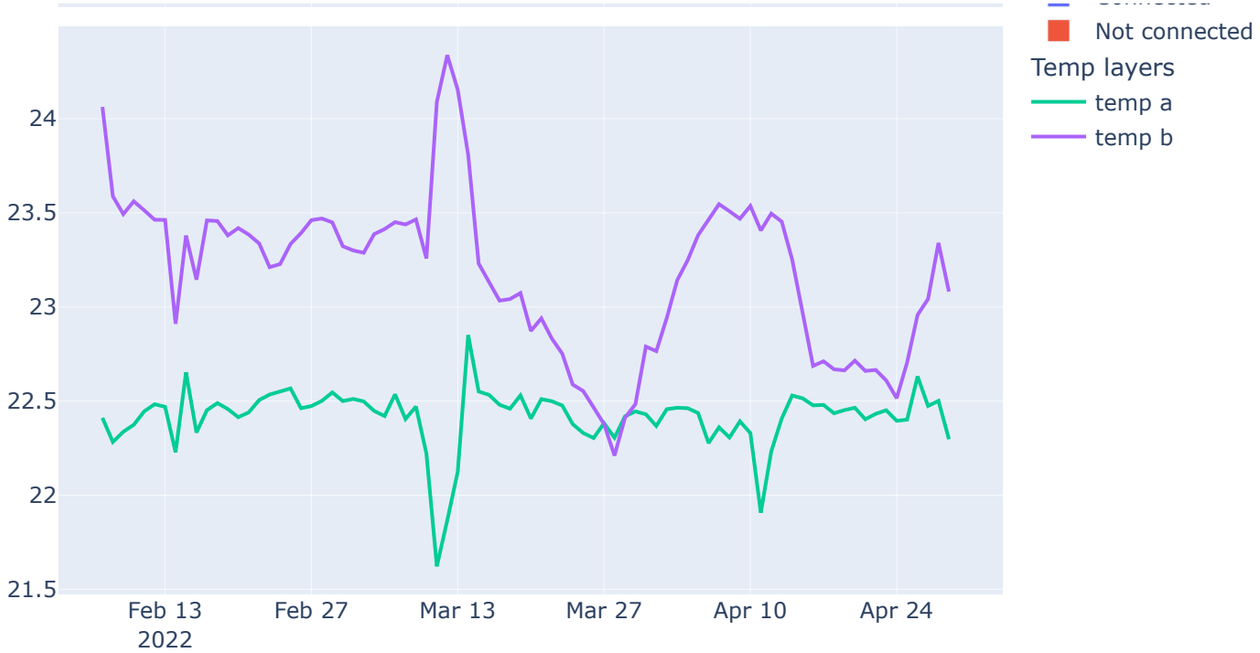

Connection status
Connected
Not connected
Temp layers
temp a
temp b
24
23.5
23
22.5
22
21.5



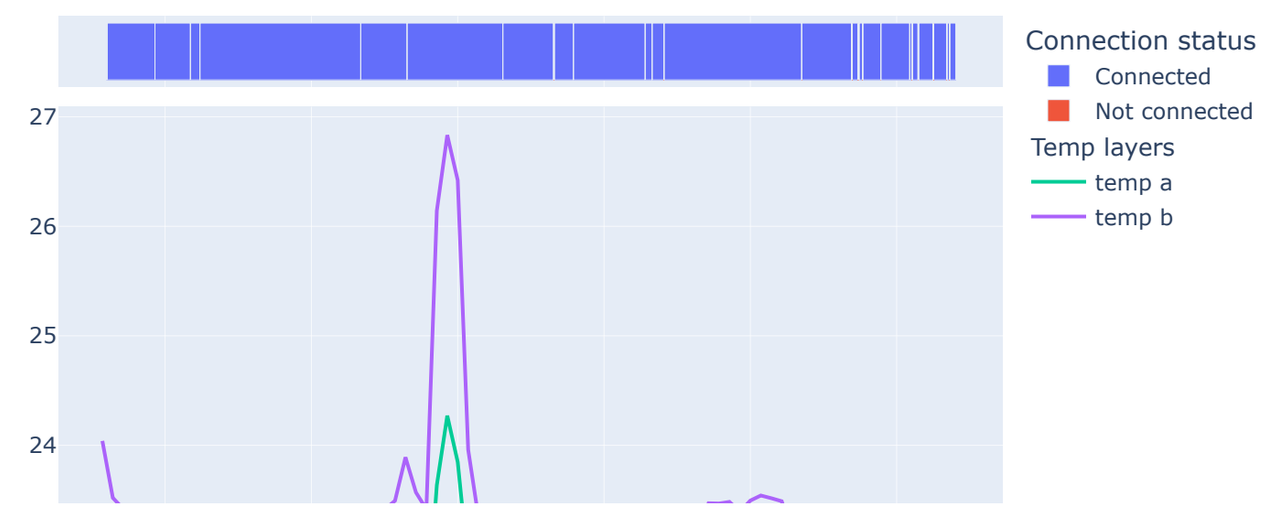
A3_upper

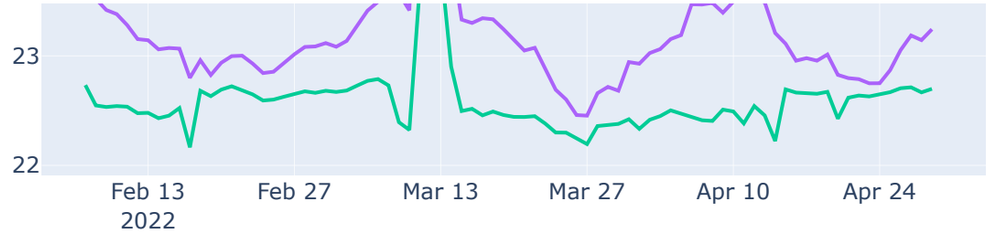


A4_lower

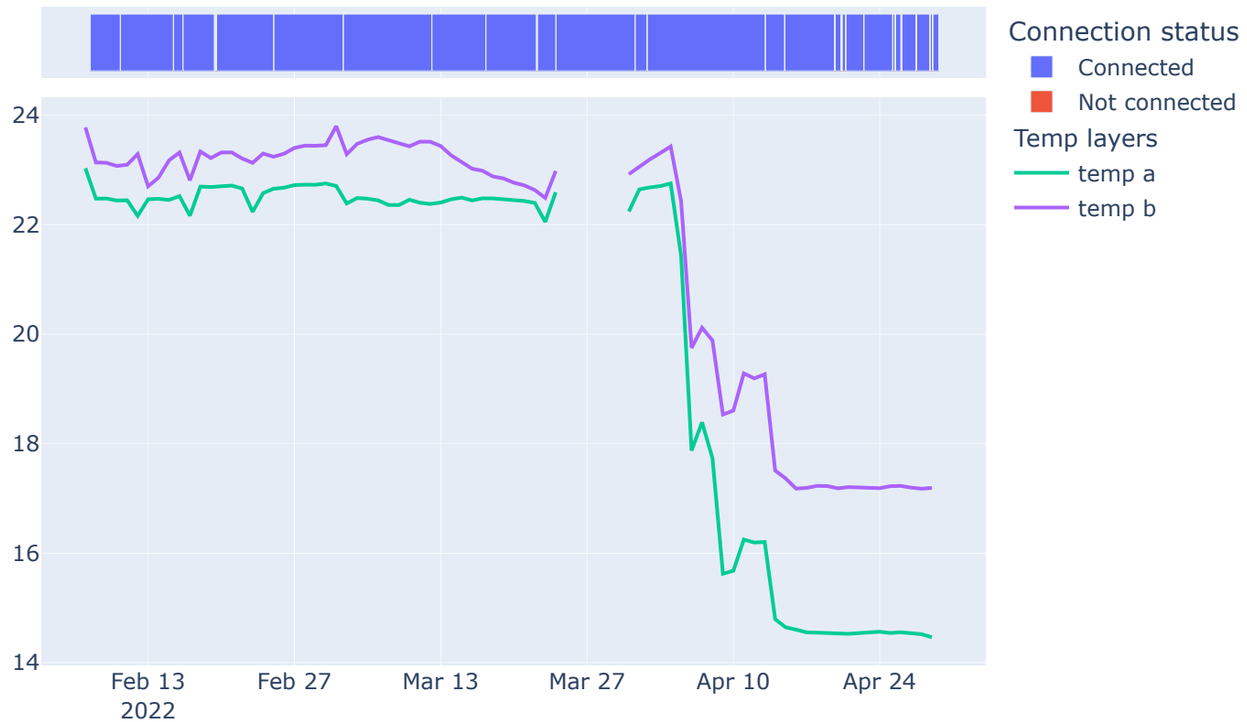

Not connected
Temp layers
temp a
temp b
24
23.5
23
22.5
22
21.5
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

A4_upper

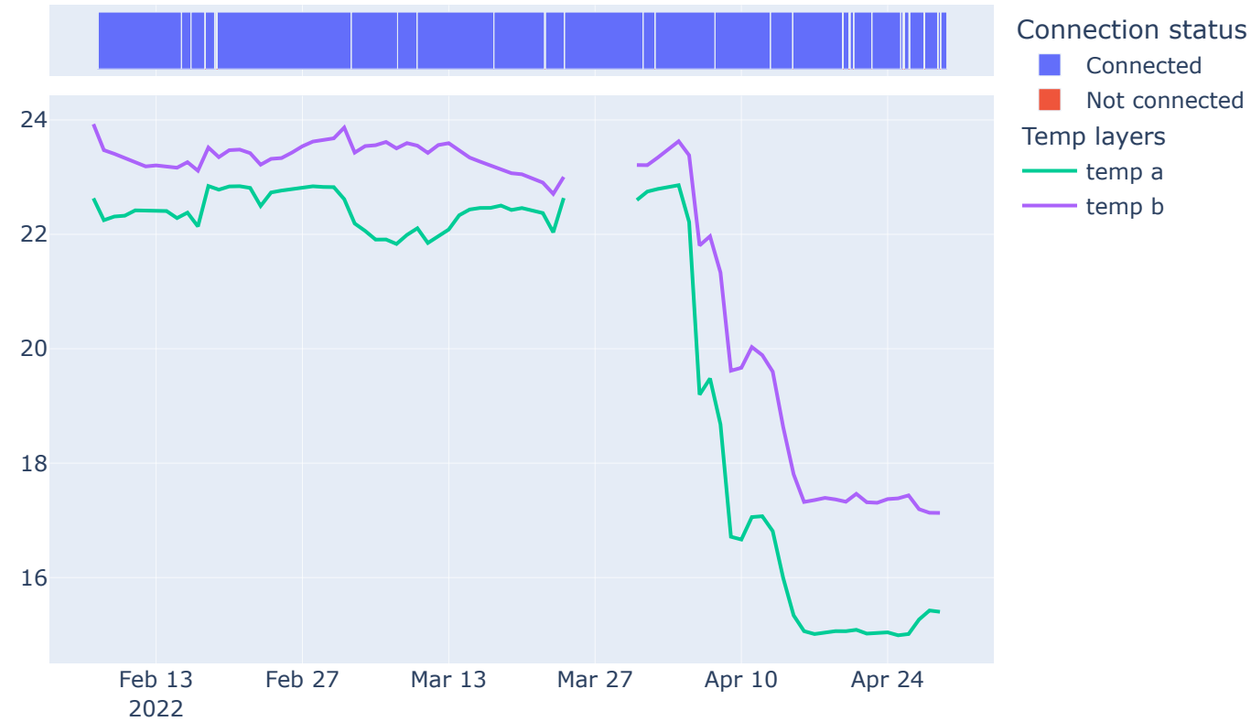

Connection status
Connected
Not connected
Temp layers
temp a
temp b
27
26
25
24



B1_lower



B1_upper

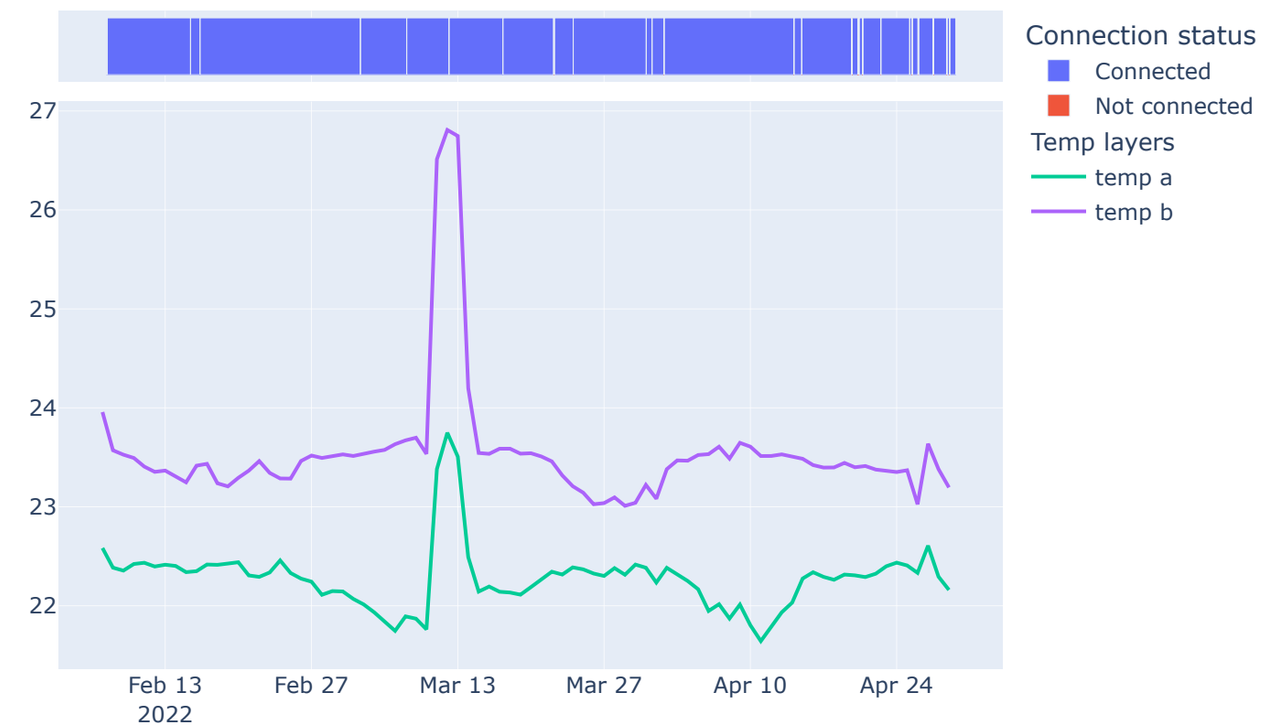

Connection status
Connected
Not connected
Temp layers
temp a
temp b
24
22
20
18
16
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

B2_lower

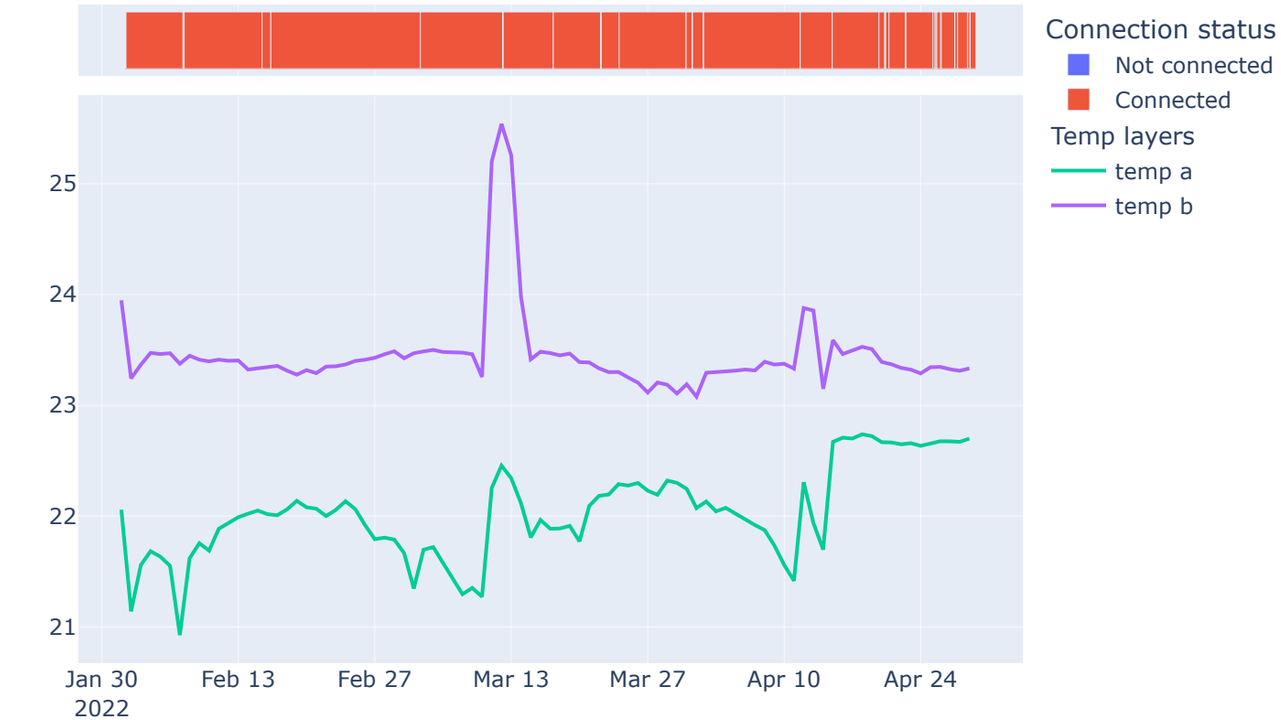

Connection status
Connected
Not connected
Temp layers
temp a
temp b
24.5
24
23.5
23


22.5
22
21.5
21
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

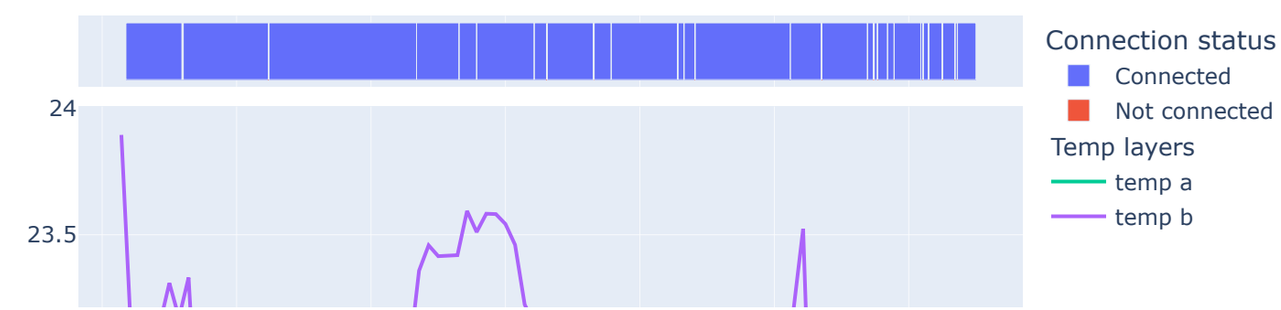
B2_upper


27
26
25
24
23
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24
Connection status
Connected
Not connected
Temp layers
temp a
temp b

B3_upper

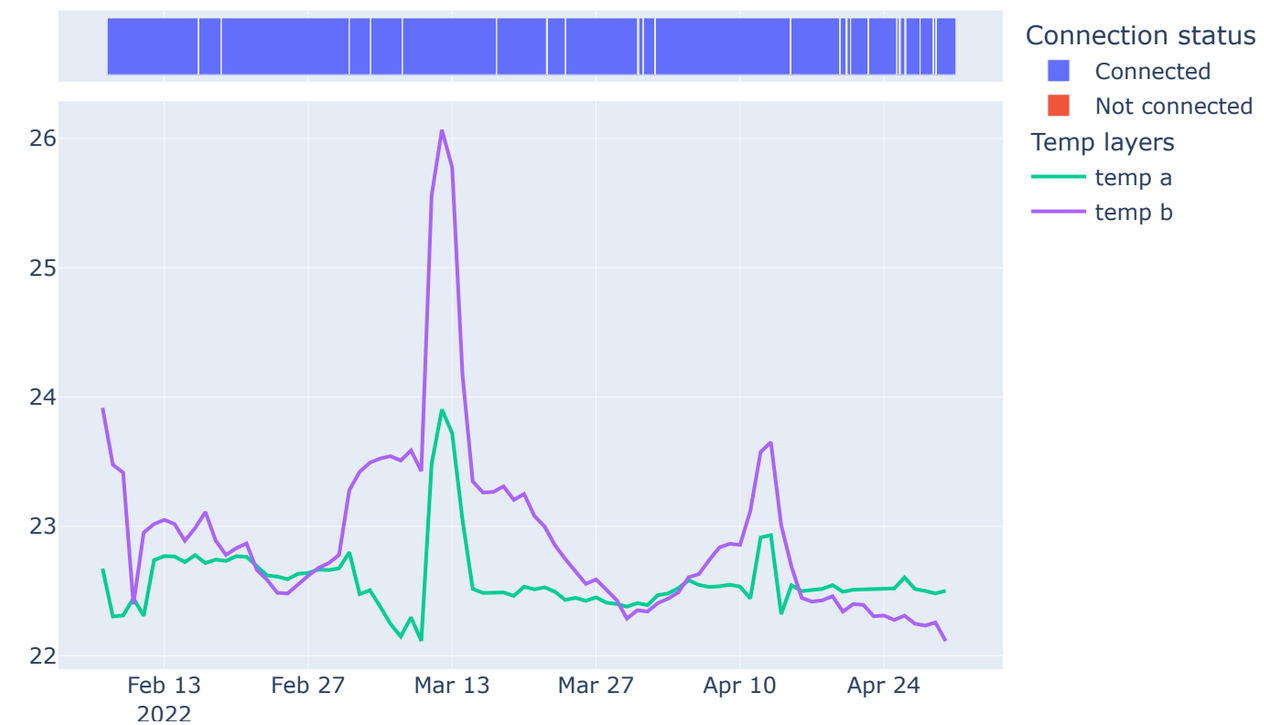


B3_lower

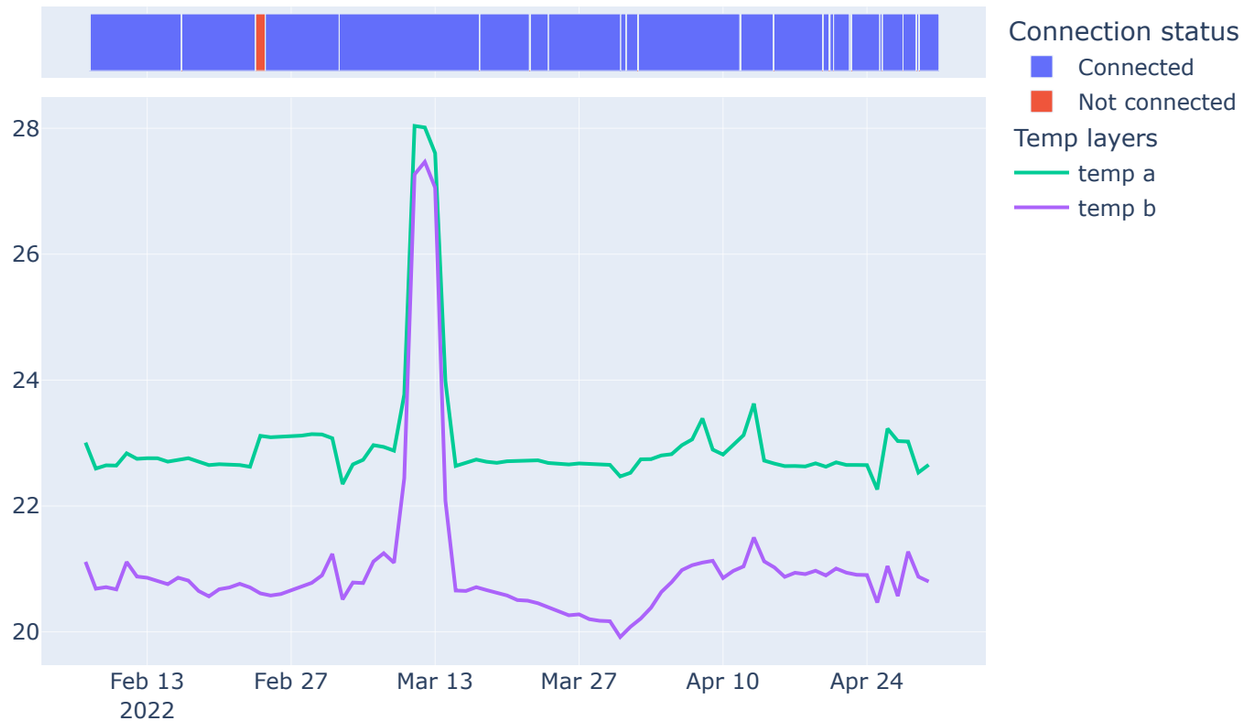



23
22.5
22
Jan 30
2022
Feb 13
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

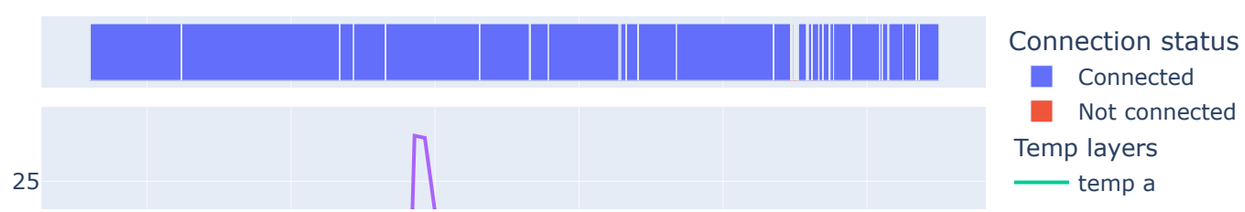
B4_lower

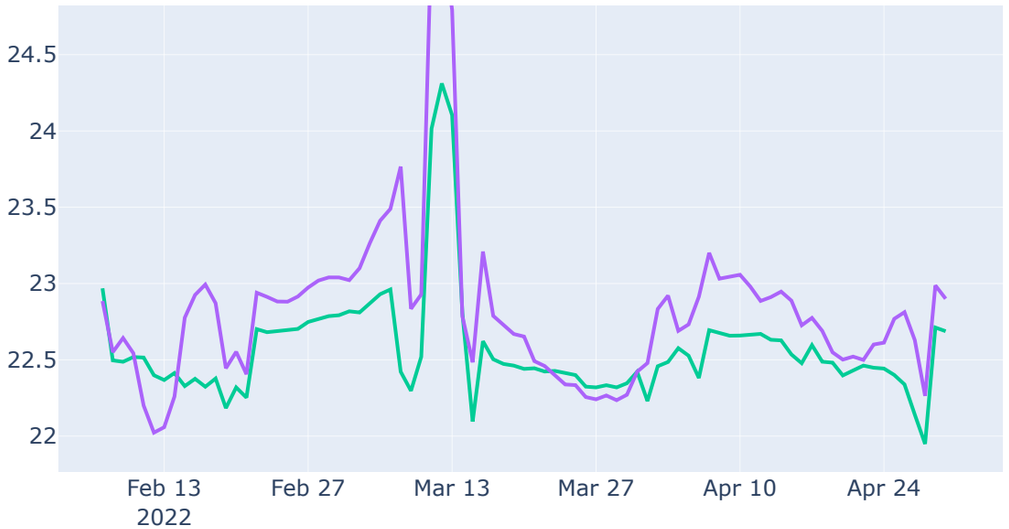

Connection status
Connected
Not connected
Temp layers
temp a
temp b
26
25
24
23
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

B4_upper

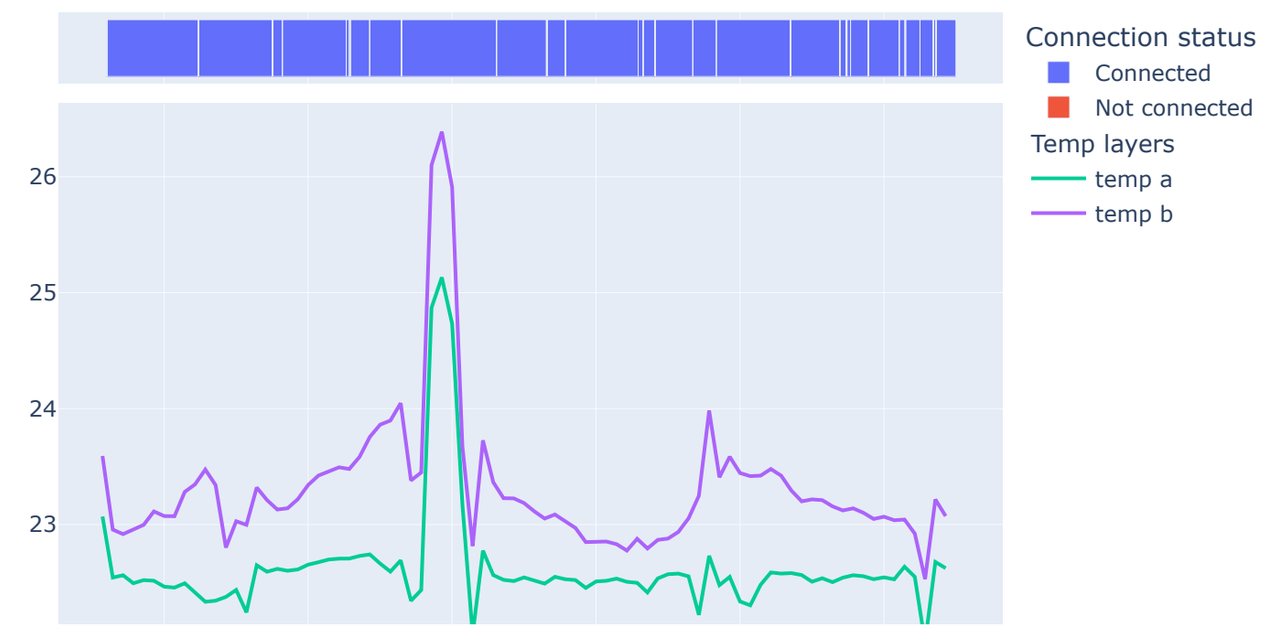


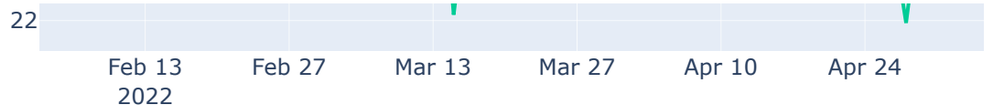
C1_lower



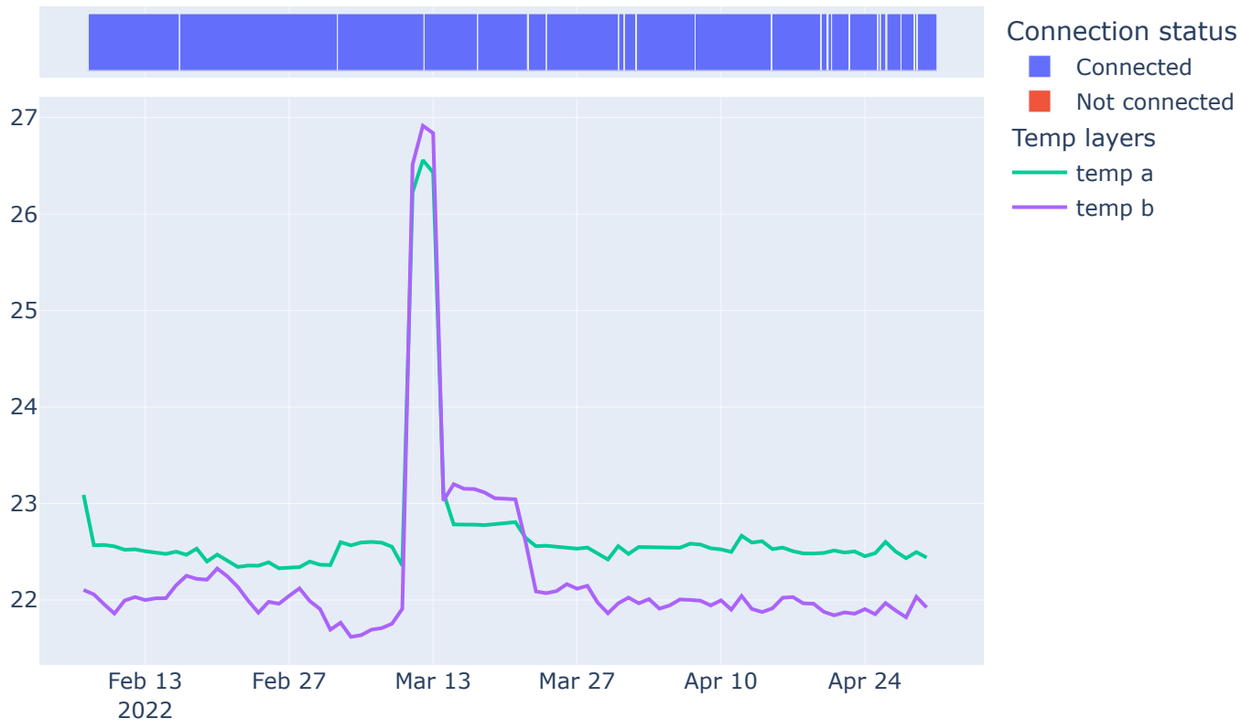

temp b
24.5
24
23.5
23
22.5
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

C1_upper

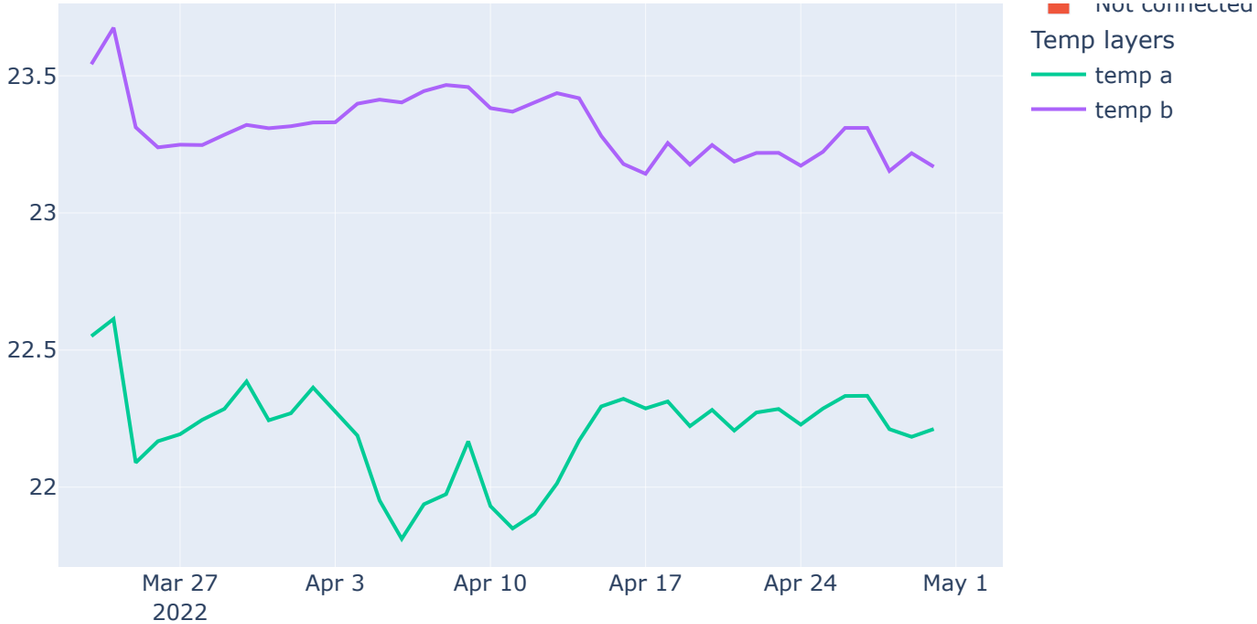

Connection status
Connected
Not connected
Temp layers
temp a
temp b
26
25
24
23



C2_lower



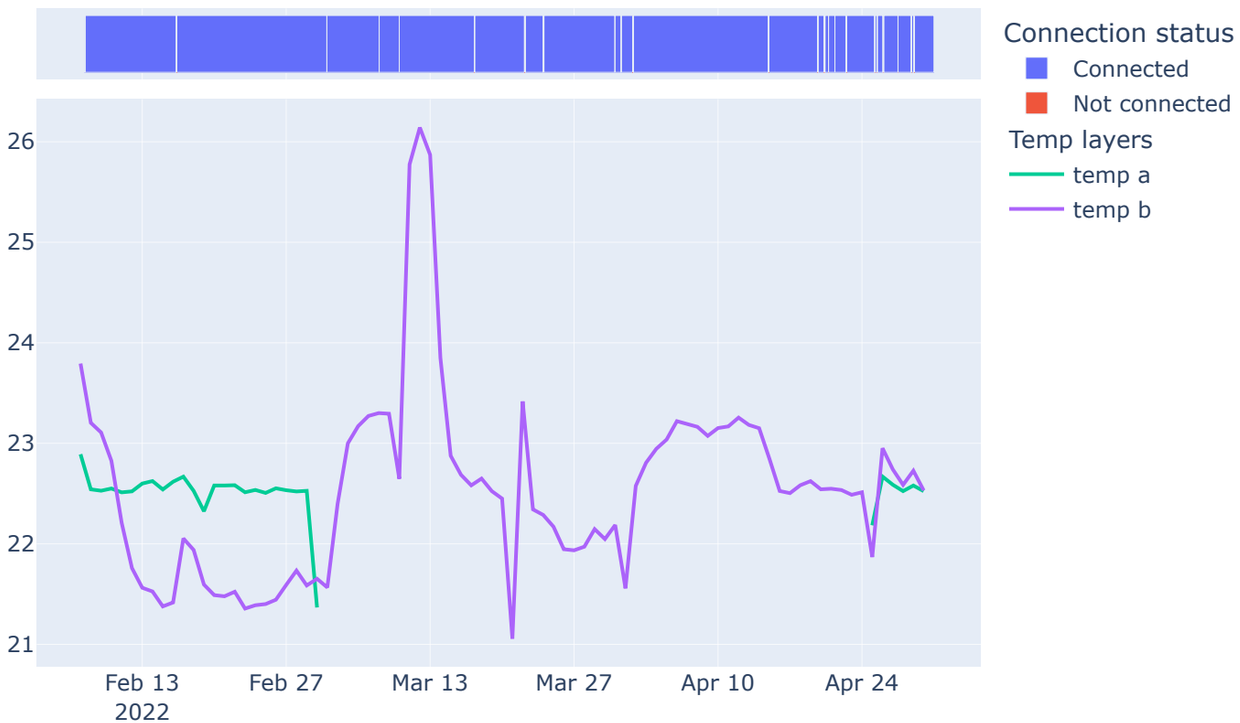
C2_upper


Not connected
Temp layers
temp a
temp b
23.5
23
22.5
22
Mar 27
2022
Apr 3
Apr 10
Apr 17
Apr 24
May 1

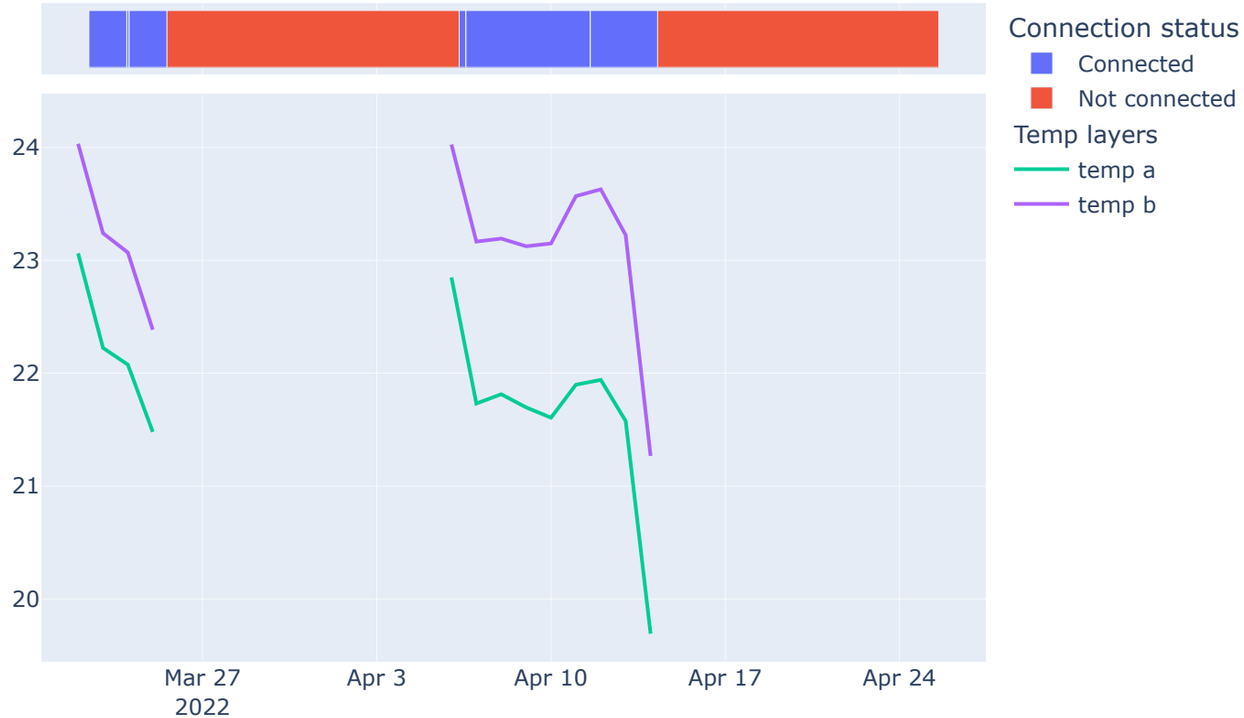
C3_lower


Connection status
Connected
Not connected
Temp layers
temp a
temp b
24
23.5
23
22.5

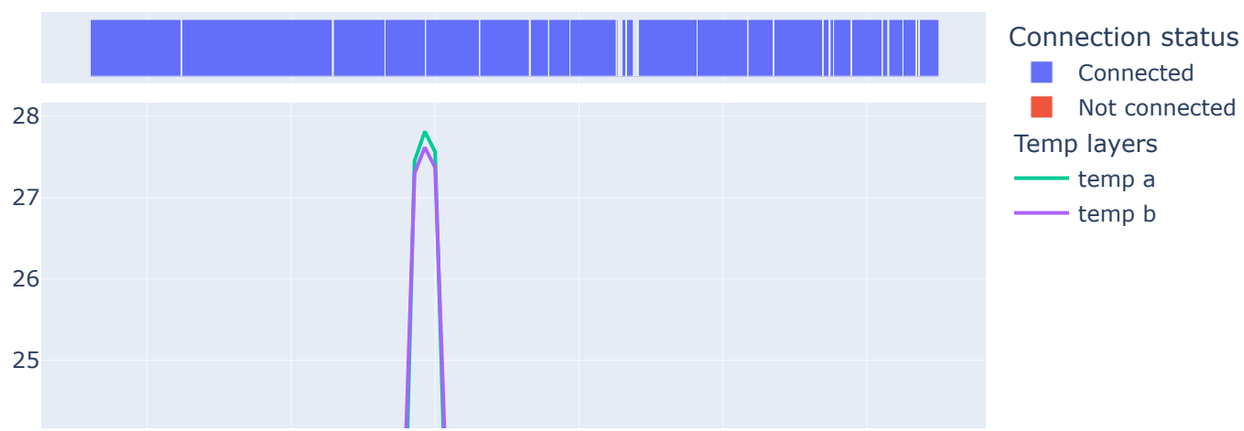
C3_upper

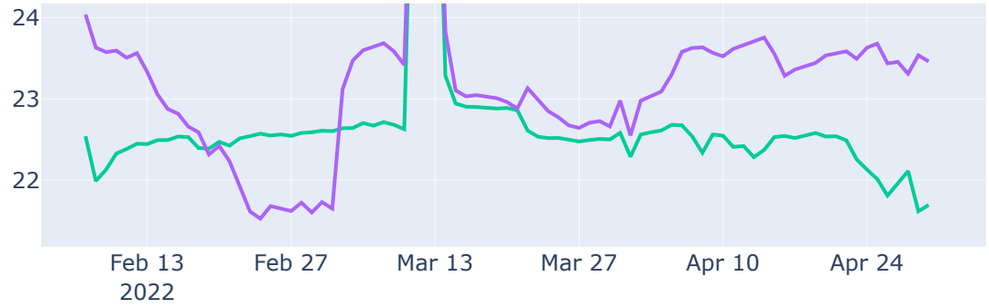


C4_lower

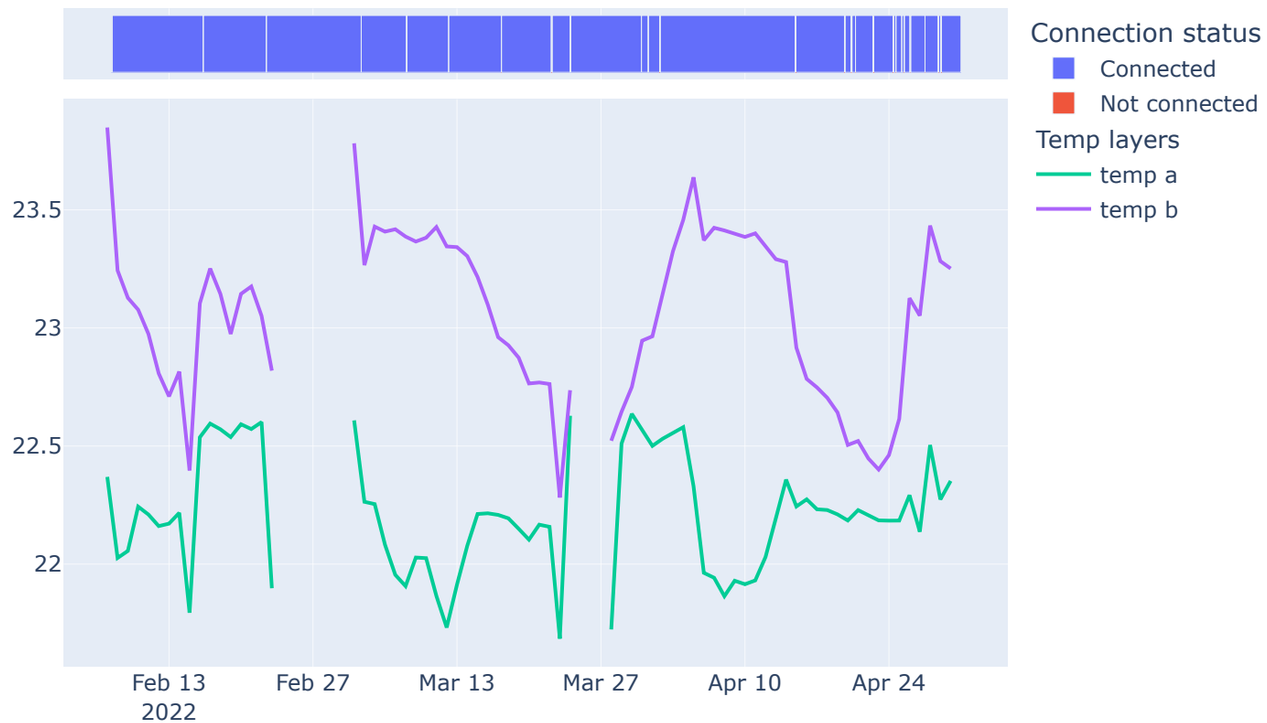

Connection status
Connected
Not connected
Temp layers
temp a
temp b
24
23
22
21
20
Mar 27
2022
Apr 3
Apr 10
Apr 17
Apr 24

C4_upper

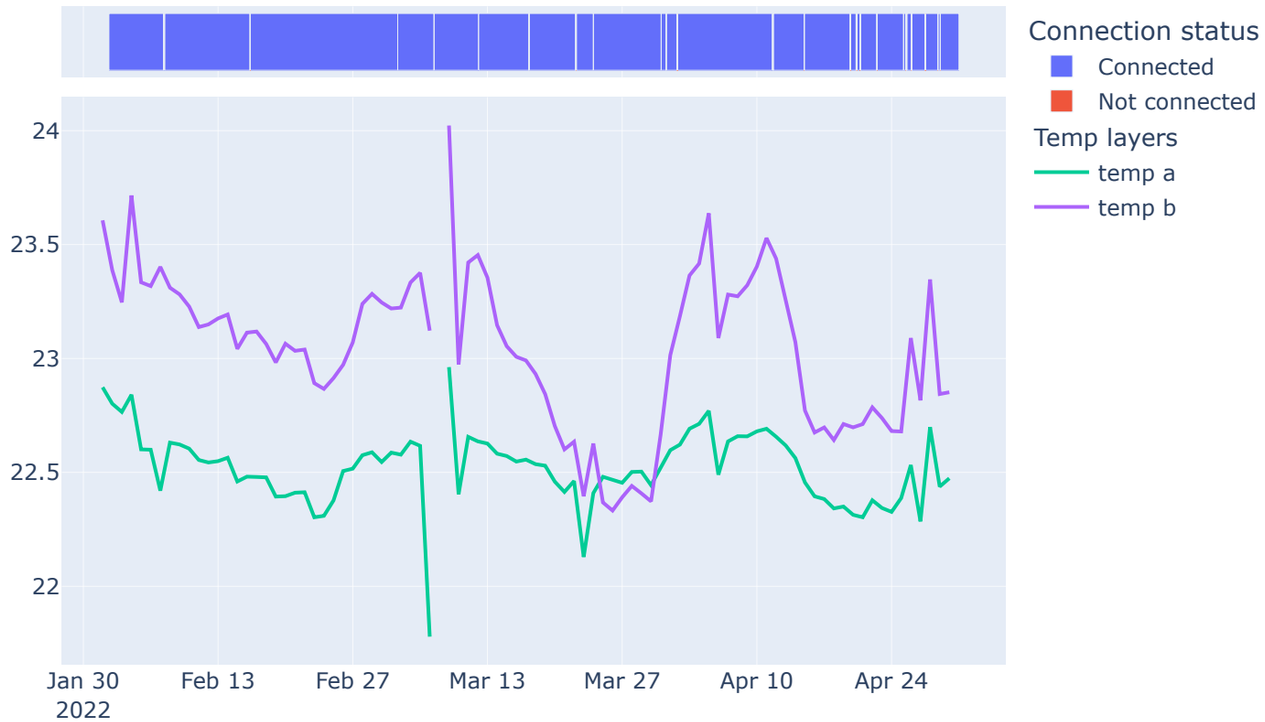

Connection status
Connected
Not connected
Temp layers
temp a
temp b
28
27
26
25


24
23
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

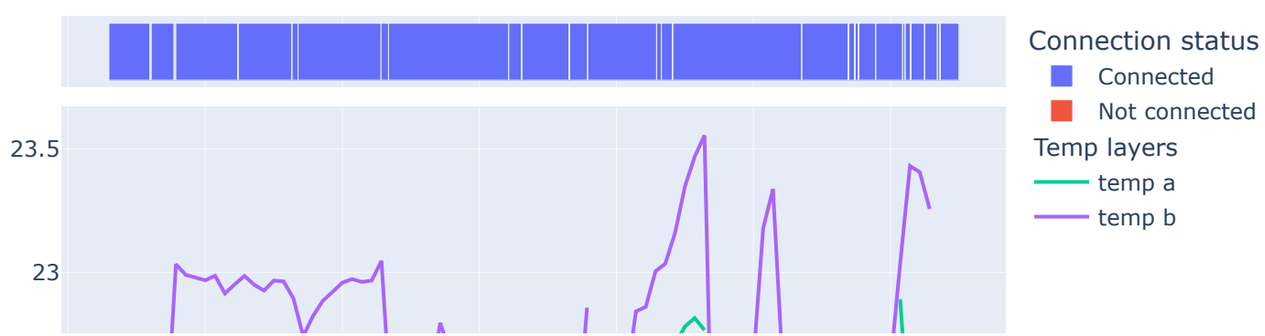
D1_lower

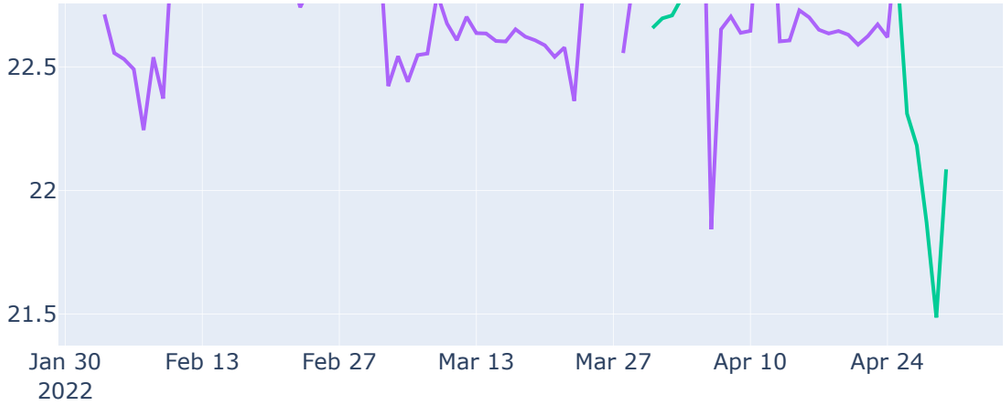

Connection status
Connected
Not connected
Temp layers
temp a
temp b
23.5
23
22.5
22
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

D1_upper

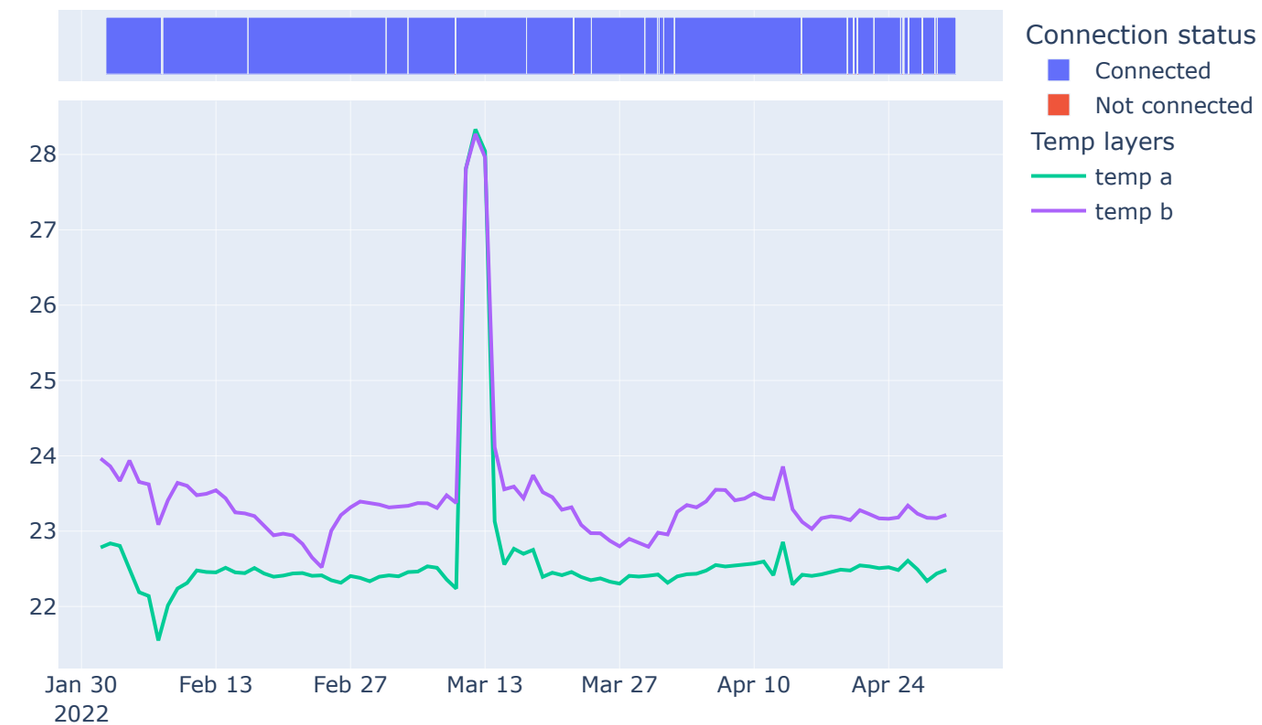


D2_lower

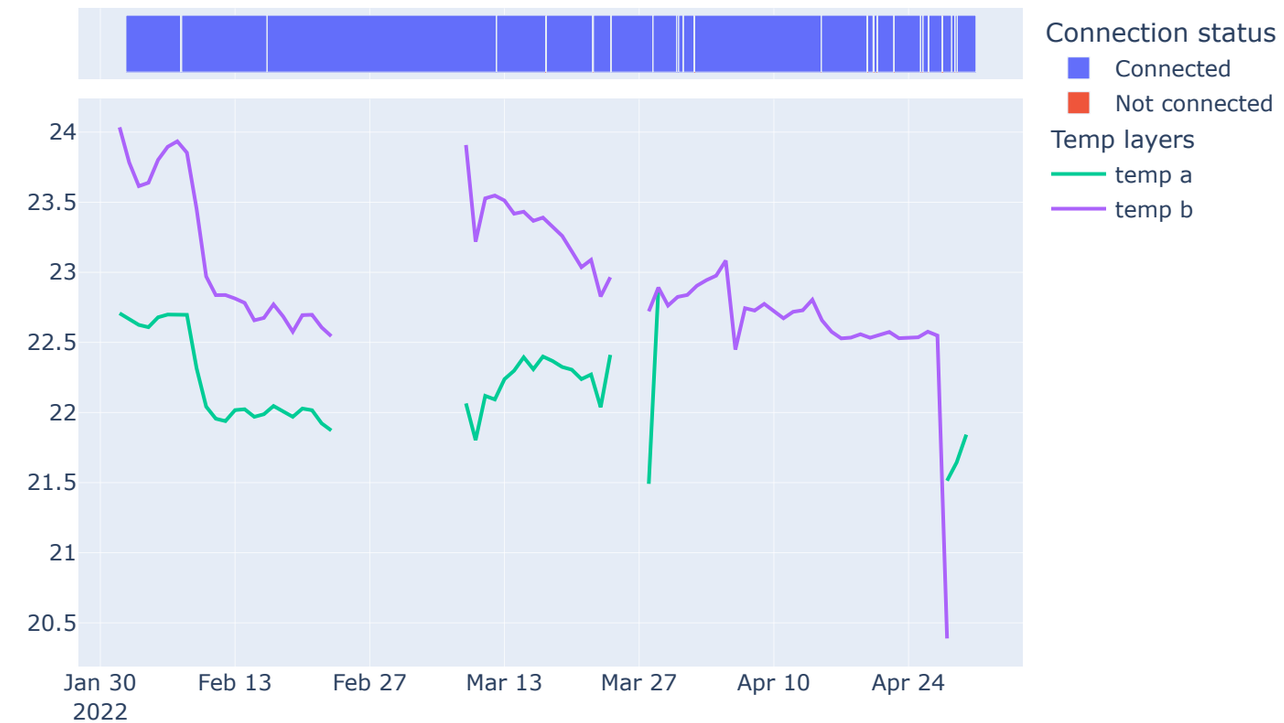



22.5
22
21.5
Jan 30
2022
Feb 13
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

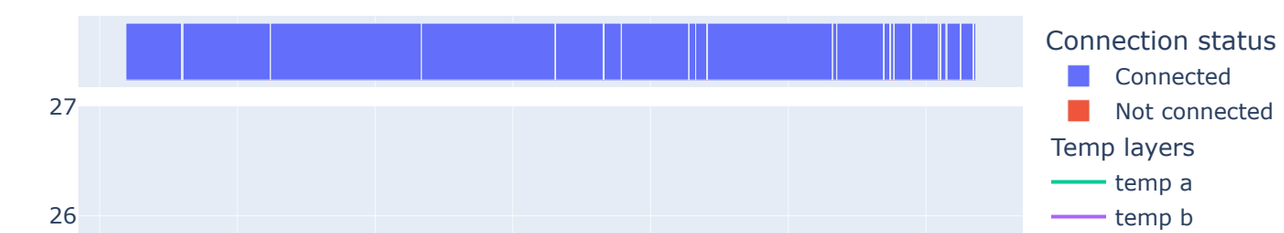
D2_upper

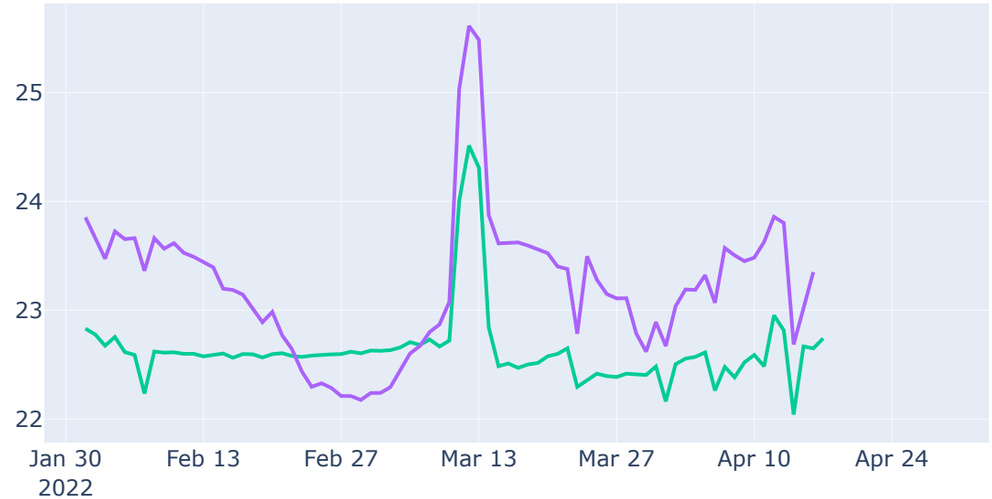

Connection status
Connected
Not connected
Temp layers
temp a
temp b
28
27
26
25
24
23
22
Jan 30
2022
Feb 13
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

D3_lower



D3_upper



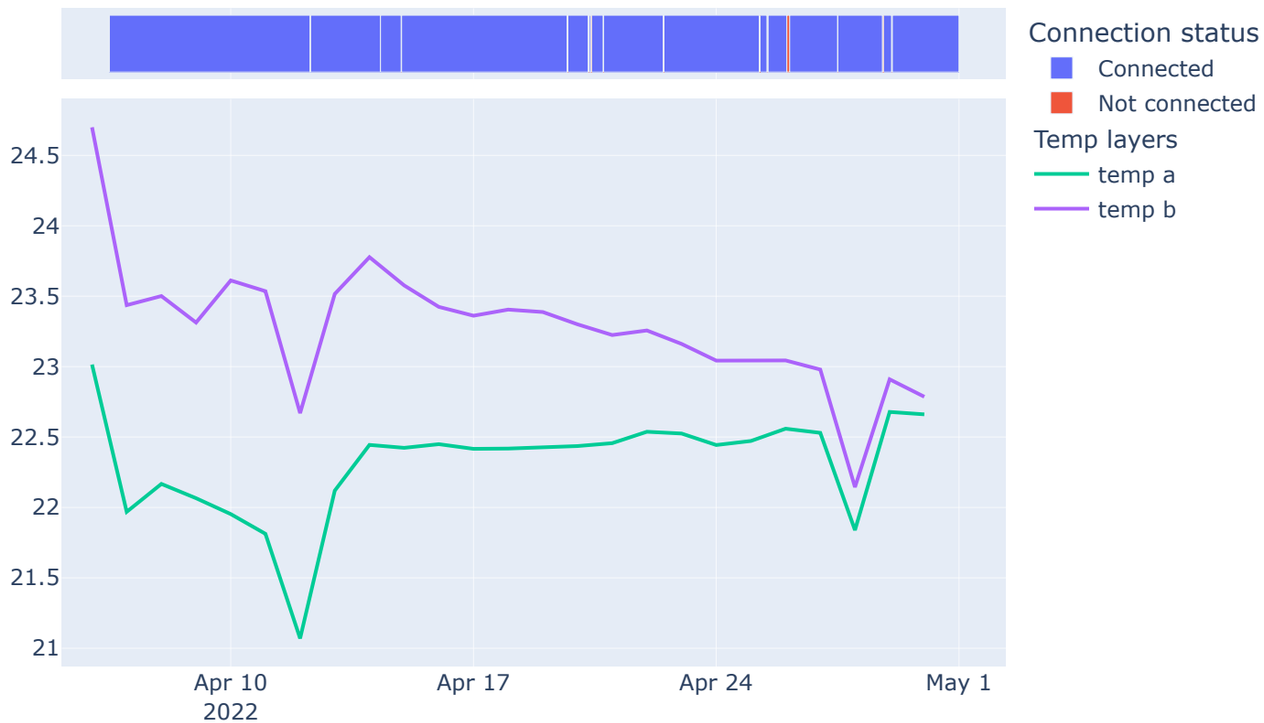

25
24
23
22
Jan 30
2022
Feb 13
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

E1

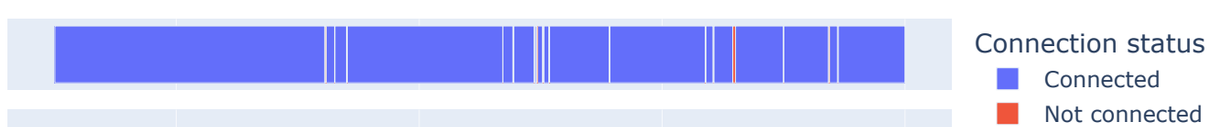

Connection status
Connected
Not connected
Temp layers
temp a
temp b
25.5
25
24.5
24
23.5
23
22.5
22

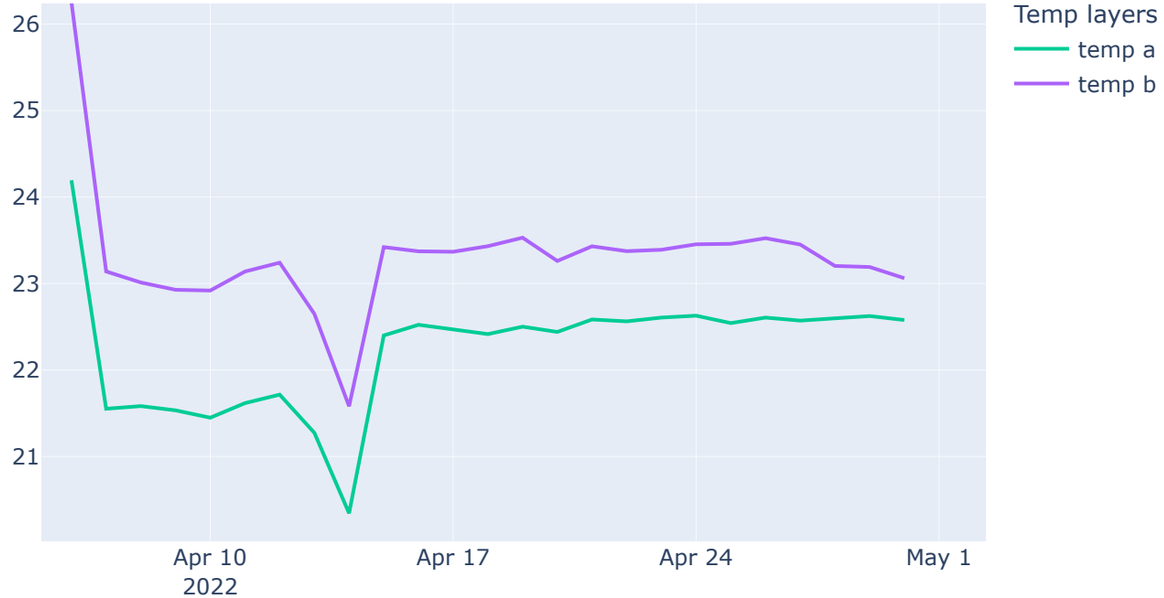

Apr 10
2022
Apr 17
Apr 24
May 1

E2

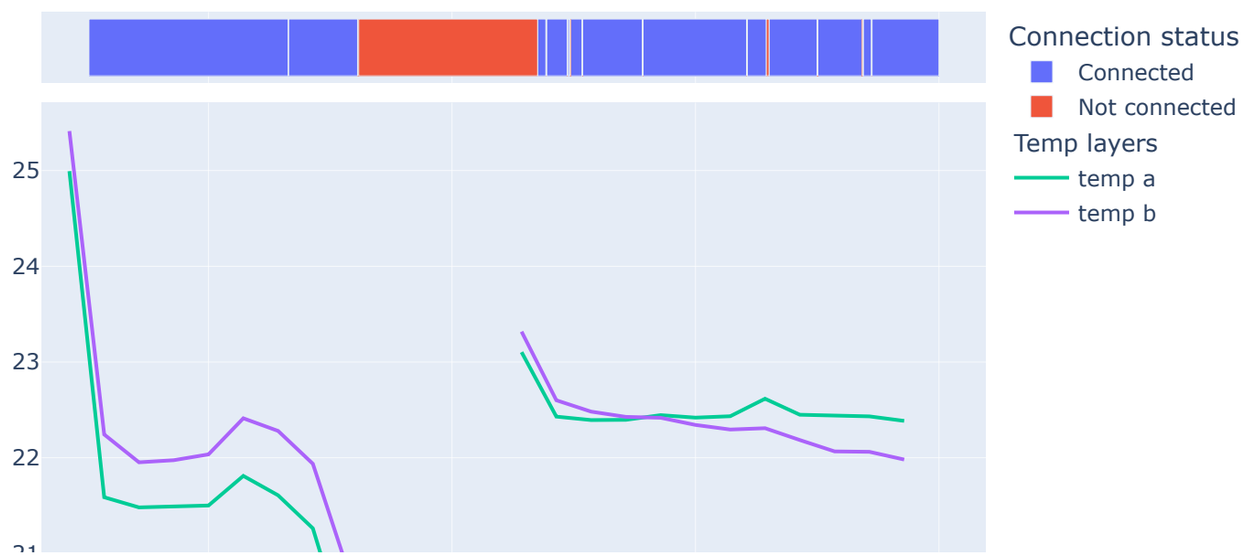

Connection status
Connected
Not connected
Temp layers
temp a
temp b
24.5
24
23.5
23
22.5
22
21.5
21
Apr 10
2022
Apr 17
Apr 24
May 1

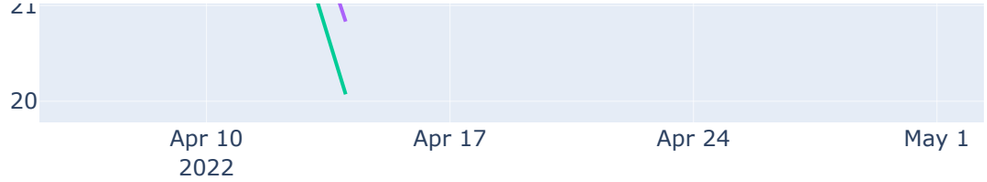
E3


Connection status
Connected
Not connected

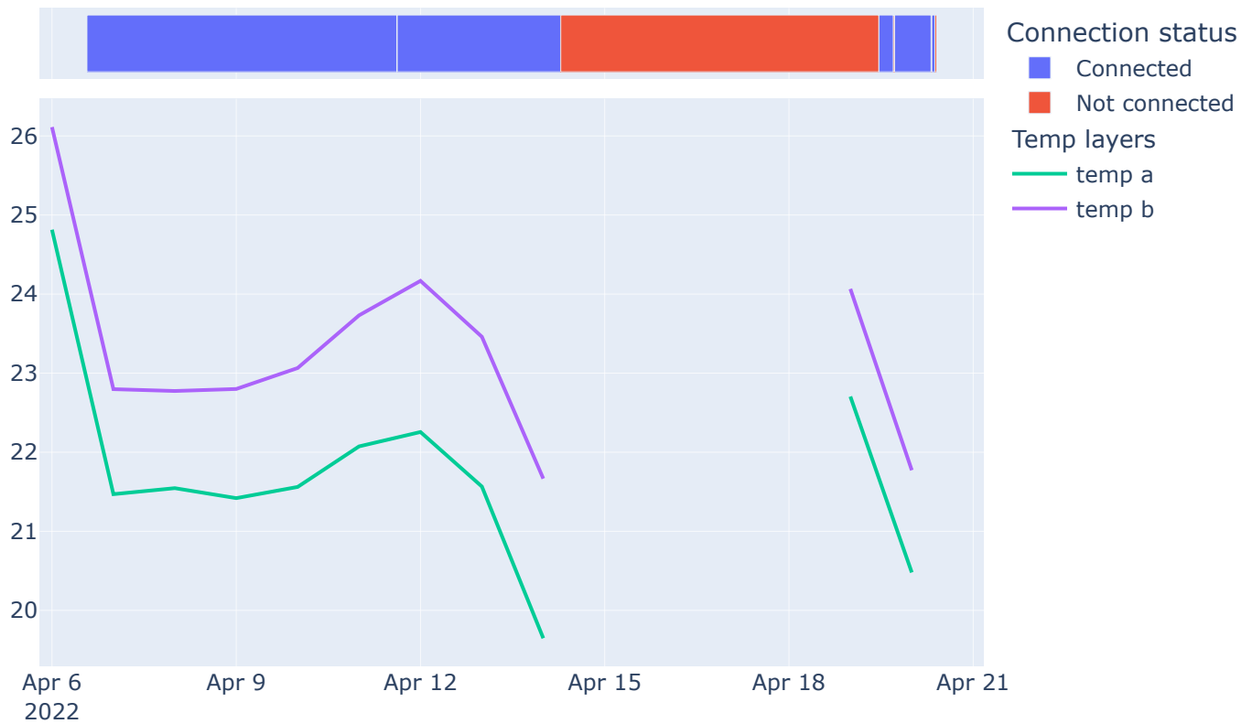

26
25
24
23
22
21
Apr 10
2022
Apr 17
Apr 24
May 1
Temp layers
temp a
temp b

E4

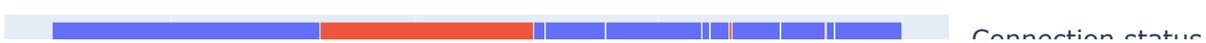

Connection status
Connected
Not connected
Temp layers
temp a
temp b
25
24
23
22

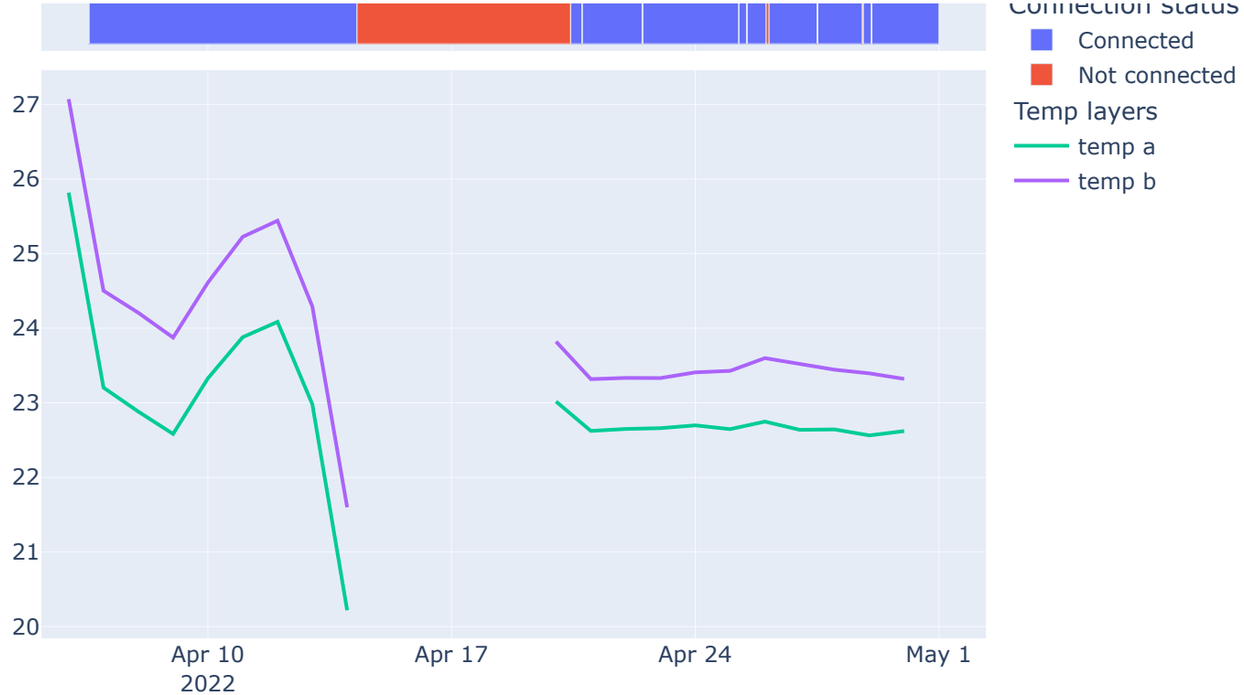

21
20
Apr 10
2022
Apr 17
Apr 24
May 1

E5

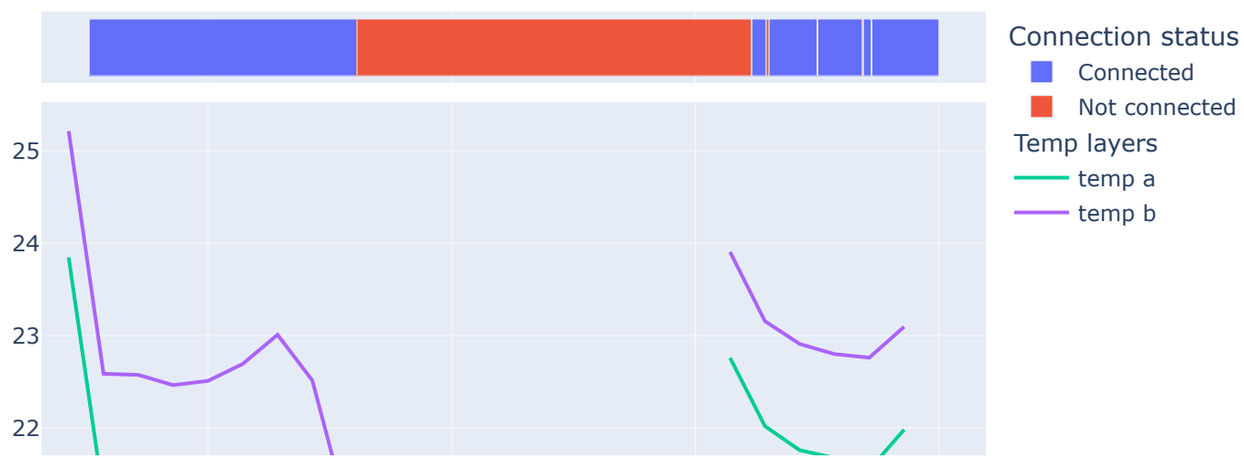

Connection status
Connected
Not connected
Temp layers
temp a
temp b
26
25
24
23
22
21
20
Apr 6
2022
Apr 9
Apr 12
Apr 15
Apr 18
Apr 21

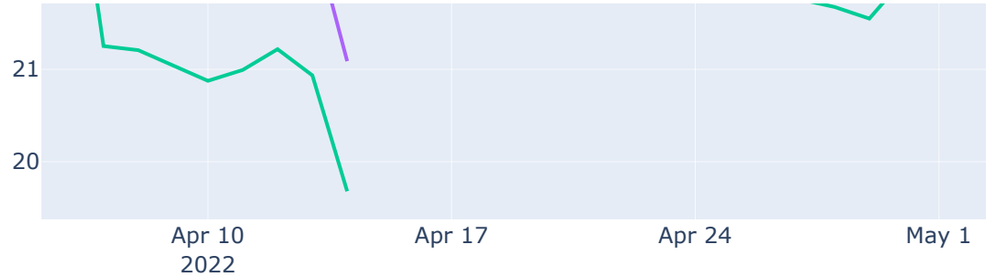
E6



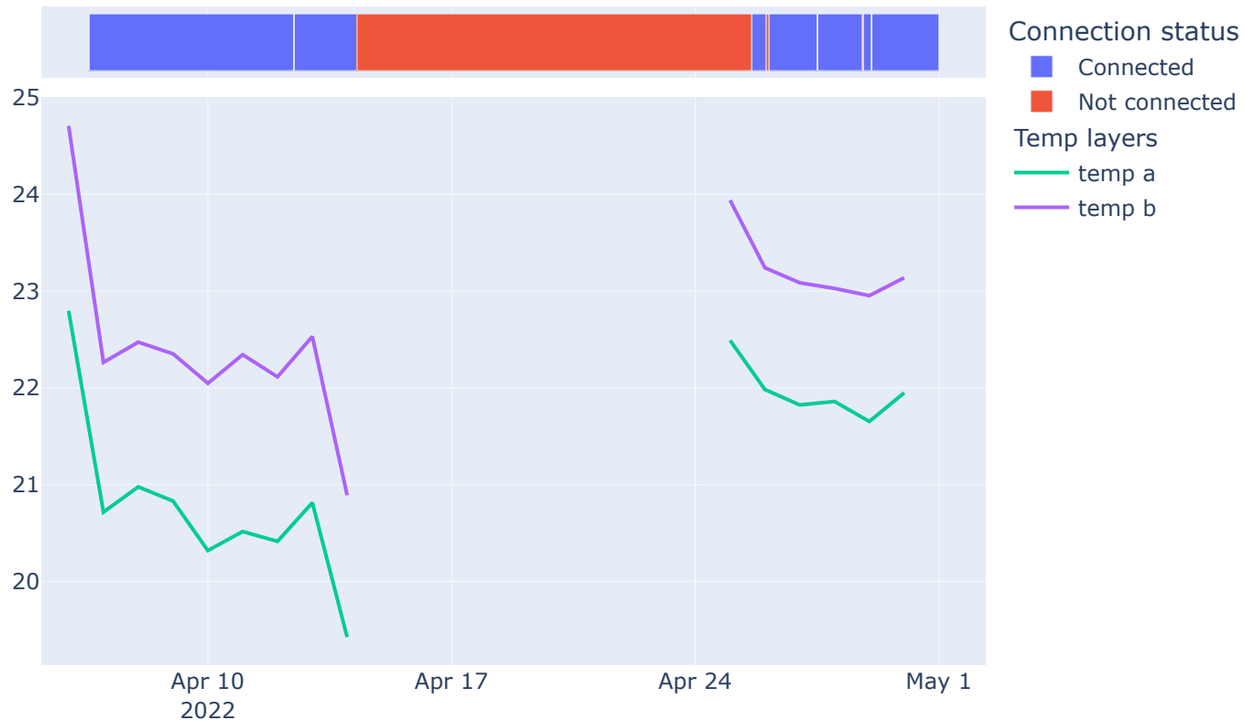

Connection status
Connected
Not connected
Temp layers
temp a
temp b
27
26
25
24
23
22
21
20
Apr 10
2022
Apr 17
Apr 24
May 1

E7

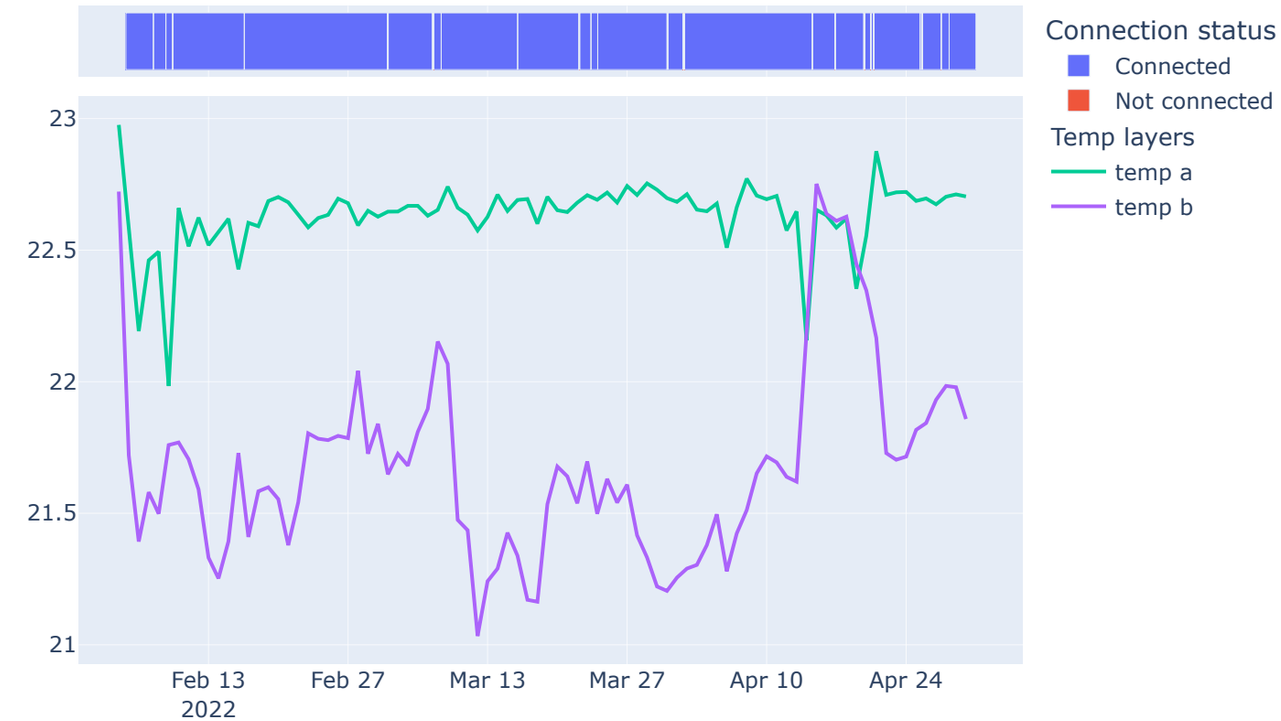

Connection status
Connected
Not connected
Temp layers
temp a
temp b
25
24
23
22


21
20
Apr 10
2022
Apr 17
Apr 24
May 1

E8

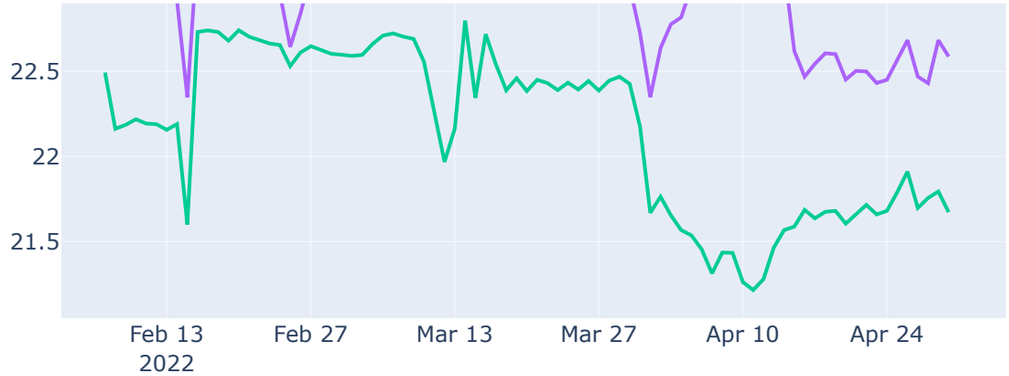

25
24
23
22
21
20
Apr 10
2022
Apr 17
Apr 24
May 1
Connection status
Connected
Not connected
Temp layers
temp a
temp b

E9


Connection status
Connected
Not connected
Temp layers
temp a
temp b
23
22.5
22
21.5
21
Feb 13
2022
Feb 27
Mar 13
Mar 27
Apr 10
Apr 24

E10


Connection status
Connected
Not connected
Temp layers
temp a
temp b
24
23.5
23



In [54]:
```
#list of plantcubes where sensors not working
#A2_lower,A3_lower,C3_lower,C3_upper,D2_lower,D3_lower,D3_upper
```

In [55]:
```
a5 = res1[(res1.temp_b > 25)& (res1.temp_a > 25)]
a5.plantcube.unique()
```

Out[55]:
```
array(['A1_upper', 'A2_upper', 'A3_upper', 'A4_upper', 'B4_lower',
       'B4_upper', 'C1_lower', 'C1_upper', 'C2_lower', 'C4_lower',
       'C4_upper', 'D1_upper', 'D2_upper', 'D3_lower', 'D3_upper', 'E1',
       'E3', 'E4', 'E5', 'E6', 'E7', 'E10'], dtype=object)
```

In [56]:
```
a6 = res1[(res1.temp_b < 15)& (res1.temp_a < 15)]
a6.plantcube.unique()
```

Out[56]:
```
array(['C3_upper'], dtype=object)
```

In [57]:
```
#automation

#minimum and maximum temperature for recipe1 to recipe13
#Night target
mintemp = [20,16,16,16,16,16,16,21,21,21,16,23,23]
#day target
maxtemp = [23,23,23,23,23,23,23,23,23,23,23,23,23]
rha = []
rla = []

for i in range(0,13):
    j = i+1
    print("Recipe ",j)
    print('\n')
    recipe = res1[res1.recipe == j]

    print("Max temperature:",maxtemp[i])
    rh= recipe[(recipe.temp_a > (maxtemp[i]+1)) & (recipe.temp_b > (maxtemp[i]+1))]
    if rh.empty:
        print("")
    else:
        print("Greater than max:")
        print(rh.plantcube.unique())
    rh['dev_temp_a'] = abs(rh['temp_a'] - maxtemp[i])
    rh['dev_temp_b'] = abs(rh['temp_b'] - maxtemp[i])
    rh = rh.reset_index()
    #appending the greater than max values for all recipes
    rha.append(rh)
    g1 = rh.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
    if g1.empty:
        print("")
    else:
        print("Deviation:")
        #print(tabulate(g1,headers='keys', tablefmt='psql'))
        display(g1)
    print('\n')

    print("Min temperature:",mintemp[i])
    rl= recipe[(recipe.temp_a < (mintemp[i]-1)) & (recipe.temp_b < (mintemp[i]-1))]
    if rl.empty:
        print("")
    else:
```

```
        print("Lesser than min:")
        print(rl.plantcube.unique())
    rl['dev_temp_a'] = abs(rl['temp_a'] - mintemp[i])
    rl['dev_temp_b'] = abs(rl['temp_b'] - mintemp[i])
    rl = rl.reset_index()
    #appending the greater than max values for all recipes
    rla.append(rl)
    g2 = rl.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
    if g2.empty:
        print("")
    else:
        print("Deviation:")
        #print(tabulate(g2,headers='keys', tablefmt='psql'))
        display(g2)
    print('\n')
    print('**************************************************')
final_rh = pd.concat(rha, ignore_index=True)
final_rl = pd.concat(rla, ignore_index=True)
```

```
Recipe  1


Max temperature: 23
Greater than max:
['C3_lower']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
C3_lower	4.182776	1.727496

```
Min temperature: 20



**************************************************
```

```
Recipe  2

Max temperature: 23



Min temperature: 16



**************************************************
Recipe  3

Max temperature: 23



Min temperature: 16



**************************************************
Recipe  4

Max temperature: 23



Min temperature: 16



**************************************************
Recipe  5
```

```
Max temperature: 23


Min temperature: 16


**************************************************
Recipe  6

Max temperature: 23


Min temperature: 16


**************************************************
Recipe  7

Max temperature: 23


Min temperature: 16


**************************************************
Recipe  8
```

```
Max temperature: 23



Min temperature: 21



**********************************************
Recipe  9


Max temperature: 23
Greater than max:
['A1_upper' 'A2_upper' 'A3_lower' 'A3_upper' 'A4_upper' 'B1_upper'
 'B2_upper' 'B3_upper' 'B3_lower' 'B4_lower' 'B4_upper' 'C1_lower'
 'C1_upper' 'C2_lower' 'C4_lower' 'C4_upper' 'D1_lower' 'D1_upper'
 'D2_lower' 'D2_upper' 'D3_upper' 'E1' 'E3' 'E6']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
A1_upper	1.642957	3.305345
A2_upper	1.483596	3.780427
A3_lower	1.307945	1.951138
A3_upper	2.238570	3.693784
A4_upper	1.856389	4.314499
B1_upper	1.088333	1.622500
B2_upper	1.356270	4.434819
B3_lower	1.156875	2.273125
B3_upper	1.269565	3.054136
B4_lower	1.546740	3.426937
B4_upper	4.611672	3.882140

plantcube	dev_temp_a	dev_temp_b
C1_lower	1.850903	2.670573
C1_upper	2.359871	3.423191
C2_lower	2.956326	3.178682
C4_lower	1.569829	3.225584
C4_upper	4.099415	4.096589
D1_lower	1.323611	1.435000
D1_upper	2.489792	3.076597
D2_lower	1.305625	2.003906
D2_upper	4.327096	4.513477
D3_upper	2.459336	3.008893
E1	1.018889	3.040000
E3	1.630357	2.487321
E6	1.050000	2.666667

```
Min temperature: 21
Lesser than min:
['A2_lower' 'A3_lower' 'A4_upper' 'B3_lower' 'B4_lower' 'B4_upper'
 'C1_lower' 'C2_lower' 'C3_upper' 'C4_upper' 'D1_lower' 'D1_upper'
 'D3_lower' 'D3_upper']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
A2_lower	3.735542	2.882530
A3_lower	1.165000	1.050000
A4_upper	1.067500	1.156667
B3_lower	1.080244	1.429634

plantcube	dev_temp_a	dev_temp_b
B4_lower	1.020556	1.420000
B4_upper	1.038056	2.536667
C1_lower	1.410765	1.190055
C2_lower	1.247460	1.200668
C3_upper	5.298073	3.954981
C4_upper	1.113972	1.257021
D1_lower	1.497083	1.086250
D1_upper	1.185802	1.156848
D3_lower	1.948851	1.150000
D3_upper	1.612729	1.187859

```
**************************************************
Recipe  10


Max temperature: 23



Min temperature: 21
Lesser than min:
['E9']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
E9	1.156756	1.151509

```
**************************************************
Recipe  11


Max temperature: 23




Min temperature: 16




**************************************************
Recipe  12


Max temperature: 23
Greater than max:
['B4_upper' 'C1_lower' 'D3_lower']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
B4_upper	3.008667	1.427929
C1_lower	1.223514	2.278198
D3_lower	2.339583	3.368519

```
Min temperature: 23
Lesser than min:
['A1_lower' 'A2_lower' 'A3_lower' 'A4_lower' 'A4_upper' 'B2_lower'
 'B3_lower' 'B4_lower' 'B4_upper' 'C1_lower' 'D1_lower' 'D1_upper'
 'D3_lower' 'E1']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b

plantcube	dev_temp_a	dev_temp_b
plantcube		
A1_lower	4.279503	3.294572
A2_lower	3.455119	1.741548
A3_lower	1.966449	1.495483
A4_lower	2.208889	1.367937
A4_upper	1.244442	1.050732
B2_lower	2.265110	2.057636
B3_lower	1.079911	1.485627
B4_lower	1.134844	1.139962
B4_upper	1.099135	3.065476
C1_lower	1.571181	1.129250
D1_lower	2.315625	1.179286
D1_upper	1.384067	1.472639
D3_lower	2.172077	1.166520
E1	1.361111	1.091111

```
**************************************************
Recipe  13


Max temperature: 23
Greater than max:
['E10']
Deviation:
```

	dev_temp_a	dev_temp_b
plantcube		

plantcube	dev_temp_a	dev_temp_b
E10	2.286838	2.883476

```
Min temperature: 23
Lesser than min:
['C2_lower' 'C3_lower' 'C4_upper' 'D1_lower' 'D1_upper' 'D3_lower' 'E9'
 'E10']
Deviation:
```

plantcube	dev_temp_a	dev_temp_b
C2_lower	2.596403	1.630396
C3_lower	1.177778	1.730145
C4_upper	2.352312	1.306959
D1_lower	4.205000	1.473000
D1_upper	1.478333	1.064861
D3_lower	3.223228	3.101313
E10	1.238376	1.240919
E9	1.394090	1.423172

****

In [58]:
```
#above the max target
final_rh
```

Out[58]:

	timestamp	Unnamed: 0.1	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe	dev_temp_a	dev_temp_b
0	2022-02-10 15:15:46.846	1986830	59267	24.506667	C3_lower	24.03	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.03	1.506667
1	2022-02-10 15:15:46.921	1986831	59268	24.510000	C3_lower	24.18	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.18	1.510000
2	2022-02-10 15:15:47.000	1986832	59269	24.513333	C3_lower	24.34	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.34	1.513333
3	2022-02-10 15:15:50.773	1986833	59270	24.516667	C3_lower	24.49	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.49	1.516667
4	2022-02-10 15:15:50.850	1986834	59271	24.520000	C3_lower	24.62	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.62	1.520000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11012	2022-03-14 08:22:24.560	3288364	305087	24.900000	E10	24.23	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.23	1.900000
11013	2022-03-14 08:22:29.428	3288365	305088	24.850000	E10	24.18	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.18	1.850000
11014	2022-03-14 08:22:47.489	3288366	305089	24.800000	E10	24.13	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.13	1.800000
11015	2022-03-14 08:22:58.496	3288367	305090	24.750000	E10	24.08	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.08	1.750000
11016	2022-03-14 08:23:09.419	3288368	305091	24.700000	E10	24.03	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.03	1.700000

11017 rows × 13 columns

In [59]:
```
final_rh
```

Out[59]:

	timestamp	Unnamed: 0.1	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe	dev_temp_a	dev_temp_b
0	2022-02-10 15:15:46.846	1986830	59267	24.506667	C3_lower	24.03	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.03	1.506667
1	2022-02-10 15:15:46.921	1986831	59268	24.510000	C3_lower	24.18	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.18	1.510000
2	2022-02-10 15:15:47.000	1986832	59269	24.513333	C3_lower	24.34	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.34	1.513333
3	2022-02-10 15:15:50.773	1986833	59270	24.516667	C3_lower	24.49	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.49	1.516667
4	2022-02-10 15:15:50.850	1986834	59271	24.520000	C3_lower	24.62	1	1.644501e+09	3.0	[[{'periods': [{'duration': Decimal('55800'), ...	1	1.62	1.520000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11012	2022-03-14 08:22:24.560	3288364	305087	24.900000	E10	24.23	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.23	1.900000
11013	2022-03-14 08:22:29.428	3288365	305088	24.850000	E10	24.18	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.18	1.850000
11014	2022-03-14 08:22:47.489	3288366	305089	24.800000	E10	24.13	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.13	1.800000
11015	2022-03-14 08:22:58.496	3288367	305090	24.750000	E10	24.08	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.08	1.750000
11016	2022-03-14 08:23:09.419	3288368	305091	24.700000	E10	24.03	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.03	1.700000

11017 rows × 13 columns

In [60]:
```
final_rh['date_time1'] = final_rh['timestamp'].dt.date
```

In [61]:
```
#below the min target
final_rl
```

Out[61]:

	timestamp	Unnamed: 0.1	Unnamed: 0	temp_b	plantcube	temp_a	connected	recipe_id	mode	layers	recipe	dev_temp_a	dev_temp_b
0	2022-04-25 07:01:45.559	282306	521494	17.975000	A2_lower	17.225	1	1.649755e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	3.775	3.025000
1	2022-04-25 07:01:46.310	282308	521497	17.925000	A2_lower	17.175	1	1.649755e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	3.825	3.075000
2	2022-04-25 07:02:00.457	282309	521503	17.900000	A2_lower	17.150	1	1.649755e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	3.850	3.100000
3	2022-04-25 07:02:05.454	282310	521504	17.790000	A2_lower	17.040	1	1.649755e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	3.960	3.210000
4	2022-04-25 07:02:10.457	282311	521505	17.690000	A2_lower	16.940	1	1.649755e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	9	4.060	3.310000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11286	2022-03-15 06:58:58.708	3289364	310968	21.950000	E10	21.680	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.320	1.050000
11287	2022-03-15 06:59:11.706	3289365	310969	21.983333	E10	21.630	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.370	1.016667
11288	2022-03-15 09:00:34.303	3289432	311481	21.983333	E10	21.730	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.270	1.016667
11289	2022-03-15 09:00:59.873	3289433	311482	21.950000	E10	21.680	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.320	1.050000
11290	2022-03-15 09:02:29.303	3289434	311484	21.983333	E10	21.630	1	1.644931e+09	0.0	[[{'periods': [{'duration': Decimal('86400'), ...	13	1.370	1.016667

11291 rows × 13 columns

In [62]:
```
final_rl['date_time1'] = final_rl['timestamp'].dt.date
```

In [63]:
```
#above the max target - deviation
frh_g1 = final_rh.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
frh_g2 = final_rh.groupby('plantcube')['plantcube'].count()
frh_g3 = final_rh.groupby([final_rh.plantcube]).date_time1.nunique()
frh = pd.concat([frh_g1, frh_g2,frh_g3],axis=1)
frh.rename(columns = {'plantcube':'No of records considered', 'date_time1':'No of days included'}, inplace = True)
frh
```

Out[63]:

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
A1_upper	1.642957	3.305345	345	6
A2_upper	1.483596	3.780427	203	4
A3_lower	1.307945	1.951138	253	3
A3_upper	2.238570	3.693784	732	15
A4_upper	1.856389	4.314499	396	5
B1_upper	1.088333	1.622500	6	1
B2_upper	1.356270	4.434819	244	3
B3_lower	1.156875	2.273125	16	1
B3_upper	1.269565	3.054136	69	4
B4_lower	1.546740	3.426937	271	6
B4_upper	4.597218	3.860010	3327	15
C1_lower	1.832273	2.658922	1246	7
C1_upper	2.359871	3.423191	619	10
C2_lower	2.956326	3.178682	425	7
C3_lower	4.182776	1.727496	196	2
C4_lower	1.569829	3.225584	234	6
C4_upper	4.099415	4.096589	513	5
D1_lower	1.323611	1.435000	12	1
D1_upper	2.489792	3.076597	48	2

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
D2_lower	1.305625	2.003906	16	1
D2_upper	4.327096	4.513477	496	6
D3_lower	2.339583	3.368519	36	1
D3_upper	2.459336	3.008893	1165	7
E1	1.018889	3.040000	3	1
E10	2.286838	2.883476	117	2
E3	1.630357	2.487321	28	2
E6	1.050000	2.666667	1	1

In [64]:
```
#below the min target - deviation
frl_g1 = final_rl.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
frl_g2 = final_rl.groupby('plantcube')['plantcube'].count()
frl_g3 = final_rl.groupby([final_rl.plantcube]).date_time1.nunique()
frl = pd.concat([frl_g1, frl_g2,frl_g3],axis=1)
frl.rename(columns = {'plantcube':'No of records considered', 'date_time1':'No of days included'}, inplace = True)
frl
```

Out[64]:

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
A1_lower	4.279503	3.294572	181	2
A2_lower	3.622566	2.422854	139	2
A3_lower	1.956174	1.489772	234	3
A4_lower	2.208889	1.367937	21	1
A4_upper	1.233243	1.057437	158	10
B2_lower	2.265110	2.057636	228	1
B3_lower	1.079922	1.483715	1201	41
B4_lower	1.134358	1.141152	706	12
B4_upper	1.098504	3.060010	1161	10
C1_lower	1.517118	1.149742	181	4
C2_lower	1.908835	1.411360	206	18
C3_lower	1.177778	1.730145	9	1
C3_upper	5.298073	3.954981	173	2
C4_upper	1.879125	1.287877	123	2
D1_lower	2.359124	1.197845	116	10
D1_upper	1.364242	1.433304	863	19
D3_lower	2.198548	1.222854	4508	16
D3_upper	1.612729	1.187859	102	3
E1	1.361111	1.091111	9	1
E10	1.238376	1.240919	78	3

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
E9	1.377630	1.404332	894	17

In [65]:
```
#merge two dataframes
merge = pd.concat([final_rh, final_rl], axis=0)
```

In [66]:
```
merge.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
```

Out[66]:

plantcube	dev_temp_a	dev_temp_b
A1_lower	4.279503	3.294572
A1_upper	1.642957	3.305345
A2_lower	3.622566	2.422854
A2_upper	1.483596	3.780427
A3_lower	1.619414	1.729455
A3_upper	2.238570	3.693784
A4_lower	2.208889	1.367937
A4_upper	1.678669	3.385590
B1_upper	1.088333	1.622500
B2_lower	2.265110	2.057636
B2_upper	1.356270	4.434819
B3_lower	1.080934	1.494094
B3_upper	1.269565	3.054136
B4_lower	1.248744	1.775183
B4_upper	3.692136	3.653058
C1_lower	1.792299	2.467498
C1_upper	2.359871	3.423191
C2_lower	2.614356	2.601712
C3_lower	4.050849	1.727612
C3_upper	5.298073	3.954981
C4_lower	1.569829	3.225584
C4_upper	3.670019	3.553394
D1_lower	2.262044	1.220078
D1_upper	1.423546	1.519888

plantcube	dev_temp_a	dev_temp_b
D2_lower	1.305625	2.003906
D2_upper	4.327096	4.513477
D3_lower	2.199666	1.239853
D3_upper	2.391180	2.862290
E1	1.275556	1.578333
E10	1.867453	2.226453
E3	1.630357	2.487321
E6	1.050000	2.666667
E9	1.377630	1.404332

In [67]:
```
m_g1 = merge.groupby('plantcube')['plantcube','dev_temp_a','dev_temp_b'].mean(numeric_only = True)
m_g2 = merge.groupby('plantcube')['plantcube'].count()
m_g3 = merge.groupby([merge.plantcube]).date_time1.nunique()
m = pd.concat([m_g1, m_g2,m_g3],axis=1)
m.rename(columns = {'plantcube':'No of records considered', 'date_time1':'No of days included'}, inplace = True)
m
```

Out[67]:

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
A1_lower	4.279503	3.294572	181	2
A1_upper	1.642957	3.305345	345	6
A2_lower	3.622566	2.422854	139	2
A2_upper	1.483596	3.780427	203	4
A3_lower	1.619414	1.729455	487	6
A3_upper	2.238570	3.693784	732	15
A4_lower	2.208889	1.367937	21	1
A4_upper	1.678669	3.385590	554	15
B1_upper	1.088333	1.622500	6	1
B2_lower	2.265110	2.057636	228	1
B2_upper	1.356270	4.434819	244	3
B3_lower	1.080934	1.494094	1217	42
B3_upper	1.269565	3.054136	69	4
B4_lower	1.248744	1.775183	977	18
B4_upper	3.692136	3.653058	4488	25
C1_lower	1.792299	2.467498	1427	10
C1_upper	2.359871	3.423191	619	10
C2_lower	2.614356	2.601712	631	25
C3_lower	4.050849	1.727612	205	3
C3_upper	5.298073	3.954981	173	2

plantcube	dev_temp_a	dev_temp_b	No of records considered	No of days included
C4_lower	1.569829	3.225584	234	6
C4_upper	3.670019	3.553394	636	7
D1_lower	2.262044	1.220078	128	11
D1_upper	1.423546	1.519888	911	21
D2_lower	1.305625	2.003906	16	1
D2_upper	4.327096	4.513477	496	6
D3_lower	2.199666	1.239853	4544	17
D3_upper	2.391180	2.862290	1267	10
E1	1.275556	1.578333	12	2
E10	1.867453	2.226453	195	5
E3	1.630357	2.487321	28	2
E6	1.050000	2.666667	1	1
E9	1.377630	1.404332	894	17

In [ ]:

In [ ]: