

## Import Libraries

```
In [1]: import numpy as np
import pandas as pd
import datetime
from datetime import datetime
import math

import sys
import os

pd.options.mode.chained_assignment = None

pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option('display.max_colwidth', None)

import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

```
In [2]: np.random.seed(311)
```

```
In [3]: print(sys.version)

3.11.5 (main, Sep 11 2023, 08:19:27) [Clang 14.0.6 ]
```

## Load Data

```
In [4]: current_file_path = os.path.dirname(os.path.realpath('__file__'))
current_folder_name = 'code'

project_base_path = current_file_path.removesuffix(current_folder_name)
temperature_filepath = project_base_path + '/data/input/temperature data.xls'
```

```
In [5]: temperature_df = pd.read_excel(
    temperature_filepath,
    header = 1
)

temperature_df.head()
```

Out[5]:

	Date	Time	Date/time	Room A	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU
0	2023-02-01	22:00:00	2023-02-01 22:00:00	20.5	15.2	13.3	13.7	15.3	15.6
1	2023-02-01	23:00:00	2023-02-01 23:00:00	20.8	15.2	13.4	13.8	15.4	15.7
2	2023-02-02	00:00:00	2023-02-02 00:00:00	21.0	15.2	13.5	14.0	15.5	15.8
3	2023-02-02	01:00:00	2023-02-02 01:00:00	21.1	15.3	13.5	14.1	15.6	15.8
4	2023-02-02	02:00:00	2023-02-02 02:00:00	21.3	15.3	13.7	14.2	15.7	15.9

In [6]: temperature\_df.shape

Out[6]: (2154, 36)

In [7]: print('Columns:\n')  
print(\*temperature\_df.columns, sep = ' ')

Columns:

Date Time Date/time Room A GSW+FM AFLN+MU AFLN+FB AFLH+FM AFLH+MU AFLH GSW+  
FB AFLN+FM GSW+MU AFLH+FB Room B GSW+MU.1 AFLH.1 AFLH+FB.1 GSW+FM.1 AFLH+M  
U.1 AFLN+FM.1 GSW+FB.1 AFLN+FB.1 AFLH+FM.1 AFLN+MU.1 Room C AFLH+FM.2 GSW+F  
B.2 GSW+MU.2 AFLN+FM.2 GSW+FM.2 AFLH.2 AFLN+MU.2 AFLN+FB.2 AFLH+FB.2 AFLH+M  
U.2

# Transform Dataframe for organizing blocks(bottles) better

Split dataset into subsets per each block(bottle)

In [8]: temperature\_df\_subset\_A = temperature\_df[  
[  
    'Date', 'Time', 'Date/time', 'Room A',  
    'GSW+FM', 'AFLN+MU', 'AFLN+FB',  
    'AFLH+FM', 'AFLH+MU', 'AFLH',  
    'GSW+FB', 'AFLN+FM', 'GSW+MU', 'AFLH+FB'  
]]  
  
temperature\_df\_subset\_B = temperature\_df[  
[  
    'Date', 'Time', 'Date/time', 'Room B',  
    'GSW+MU.1', 'AFLH.1', 'AFLH+FB.1',  
    'GSW+FM.1', 'AFLH+MU.1', 'AFLN+FM.1',  
    'GSW+FB.1', 'AFLN+FB.1', 'AFLH+FM.1', 'AFLN+MU.1'  
]]  
]

```
temperature_df_subset_C = temperature_df[
    [
        'Date', 'Time', 'Date/time', 'Room C',
        'AFLH+FM.2', 'GSW+FB.2', 'GSW+MU.2',
        'AFLN+FM.2', 'GSW+FM.2', 'AFLH.2',
        'AFLN+MU.2', 'AFLN+FB.2', 'AFLH+FB.2', 'AFLH+MU.2'
    ]
]
```

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

Out[9]:

	Date	Time	Date/time	Room A	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU
0	2023-02-01	22:00:00	2023-02-01 22:00:00	20.5	15.2	13.3	13.7	15.3	15.6
1	2023-02-01	23:00:00	2023-02-01 23:00:00	20.8	15.2	13.4	13.8	15.4	15.7
2	2023-02-02	00:00:00	2023-02-02 00:00:00	21.0	15.2	13.5	14.0	15.5	15.8
3	2023-02-02	01:00:00	2023-02-02 01:00:00	21.1	15.3	13.5	14.1	15.6	15.8
4	2023-02-02	02:00:00	2023-02-02 02:00:00	21.3	15.3	13.7	14.2	15.7	15.9

Transform columns into a common style:

Add a column indicating a block

In [10]: `temperature_df_subset_A['Block'] = 'A'`  
`temperature_df_subset_B['Block'] = 'B'`  
`temperature_df_subset_C['Block'] = 'C'`

Remove block indicator

In [11]: `temperature_df_subset_A = temperature_df_subset_A.rename(columns = {'Room A`  
`temperature_df_subset_B = temperature_df_subset_B.rename(columns = {'Room B`  
`temperature_df_subset_C = temperature_df_subset_C.rename(columns = {'Room C`

Remove digits from column names for treatments

In [12]: `def remove_digits_from_column_names(df):`  
`df.columns = df.columns.str.replace('[\.\d]', '', regex = True)`  
`remove_digits_from_column_names(temperature_df_subset_B)`  
`remove_digits_from_column_names(temperature_df_subset_C)`

Combine datasets with each block into one

In [13]: `temperature_df_transformed = pd.concat([`  
`temperature_df_subset_A,`

```
temperature_df_subset_B,
temperature_df_subset_C
],
ignore_index = True
)
```

In [14]: temperature\_df\_transformed.head(10)

Out[14]:

	Date	Time	Date/time	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU
0	2023-02-01	22:00:00	2023-02-01 22:00:00	20.5	15.2	13.3	13.7	15.3	15.6
1	2023-02-01	23:00:00	2023-02-01 23:00:00	20.8	15.2	13.4	13.8	15.4	15.7
2	2023-02-02	00:00:00	2023-02-02 00:00:00	21.0	15.2	13.5	14.0	15.5	15.8
3	2023-02-02	01:00:00	2023-02-02 01:00:00	21.1	15.3	13.5	14.1	15.6	15.8
4	2023-02-02	02:00:00	2023-02-02 02:00:00	21.3	15.3	13.7	14.2	15.7	15.9
5	2023-02-02	03:00:00	2023-02-02 03:00:00	21.4	15.4	13.7	14.3	15.8	16.0
6	2023-02-02	04:00:00	2023-02-02 04:00:00	21.5	15.4	13.8	14.5	15.9	16.1
7	2023-02-02	05:00:00	2023-02-02 05:00:00	21.6	15.5	14.0	14.6	16.0	16.2
8	2023-02-02	06:00:00	2023-02-02 06:00:00	21.7	15.5	14.1	14.7	16.1	16.2
9	2023-02-02	07:00:00	2023-02-02 07:00:00	21.9	15.6	14.2	14.9	16.2	16.3

## Data Cleaning

Convert Date to a pandas datetime object

```
In [15]: temperature_df_transformed['Date'] = pd.to_datetime(
           temperature_df_transformed['Date']
         )

temperature_df_transformed['Date/time'] = pd.to_datetime(
           temperature_df_transformed['Date/time']
         )
```

Some records have unrealistic values for Date/time

```
In [16]: temperature_df_transformed['Date/time'].min()
```

```
Out[16]: Timestamp('1970-01-01 00:00:00.000000045')
```

```
In [17]: temperature_df_transformed[
    temperature_df_transformed['Date/time'] == temperature_df_transformed['Date/time'].head()
```

```
Out[17]:
```

	Date	Time	Date/time	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH+FB
2150	NaT	NaN	1970-01-01 00:00:00.000000045	0.0	68.0	209.0	410.0	435.0	410.0	435.0
4304	NaT	NaN	1970-01-01 00:00:00.000000045	0.0	0.0	317.0	529.0	497.0	317.0	529.0
6458	NaT	NaN	1970-01-01 00:00:00.000000045	0.0	0.0	210.0	617.0	465.0	210.0	617.0

Remove the records with unrealistic or missing values for Date and time

```
In [18]: n_records_before_cleaning_dates = temperature_df_transformed.shape[0]
```

```
In [19]: temperature_df_transformed.drop(
    temperature_df_transformed[
        temperature_df_transformed['Date/time'] <= np.datetime64('2000-01-01')
    ].index,
    inplace = True
)
```

```
In [20]: temperature_df_transformed['Date/time'].min()
```

```
Out[20]: Timestamp('2023-02-01 22:00:00')
```

```
In [21]: n_records_after_cleaning_dates = temperature_df_transformed.shape[0]
```

```
In [22]: difference = n_records_before_cleaning_dates - n_records_after_cleaning_dates
print(
    f'{difference} rows have been removed'
)
```

12 rows have been removed

```
In [23]: print('Columns:\n')
print(*temperature_df_transformed.columns.tolist(), sep = ' ')
```

Columns:

Date Time Date/time Room GSW+FM AFLN+MU AFLN+FB AFLH+FM AFLH+MU AFLH GSW+FB  
AFLN+FM GSW+MU AFLH+FB Block

Column 'Time' is redundant, we can drop it

```
In [24]: temperature_df_transformed.drop('Time', axis = 1, inplace = True)
```

# Data Transformation

We want to have a column counting number of hours as a part of a day from the start of the experiment for visualization

```
In [25]: starting_time = temperature_df_transformed['Date/time'].min()

temperature_df_transformed['Relative_time'] = temperature_df_transformed[
    'Date/time'
] - starting_time

temperature_df_transformed['Hours'] = (
    (
        temperature_df_transformed[
            'Relative_time'
        ].dt.days
    ) * 24 + (
        temperature_df_transformed[
            'Relative_time'
        ].dt.components['hours']
    ) + 1
) / 24
```

And we want to have a column counting number of days from the start of the experiment for taking an average per day

```
In [26]: starting_day = temperature_df_transformed['Date'].min()

temperature_df_transformed['Day'] = (
    temperature_df_transformed['Date'] - starting_day
).dt.days.astype(int)

temperature_df_transformed.drop('Date', axis = 1, inplace = True)

temperature_df_transformed.head()
```

```
Out[26]:
```

	Date/time	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	GSW+FB
0	2023-02-01 22:00:00	20.5	15.2	13.3	13.7	15.3	15.6	17.7	15.1
1	2023-02-01 23:00:00	20.8	15.2	13.4	13.8	15.4	15.7	17.8	15.1
2	2023-02-02 00:00:00	21.0	15.2	13.5	14.0	15.5	15.8	17.9	15.2
3	2023-02-02 01:00:00	21.1	15.3	13.5	14.1	15.6	15.8	18.0	15.2
4	2023-02-02 02:00:00	21.3	15.3	13.7	14.2	15.7	15.9	18.0	15.2

We want to get averages per block for each experiment

```
In [27]: columns = temperature_df_transformed.columns.tolist()
columns.remove('Block')

temperature_df_transformed_average = temperature_df_transformed[
    columns
].groupby(['Date/time']).mean()
temperature_df_transformed_average['Block'] = 'Avr'

In [28]: temperature_df_transformed.drop('Date/time', axis = 1, inplace = True)

temperature_df_transformed = pd.concat([
    temperature_df_transformed,
    temperature_df_transformed_average
],
    ignore_index = True)

In [29]: temperature_df_transformed.head(48)
```

Out [29] :

	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	GSW+FB	AFLN+FM
0	20.5	15.2	13.3	13.7	15.3	15.6	17.7	15.1	13.8
1	20.8	15.2	13.4	13.8	15.4	15.7	17.8	15.1	13.9
2	21.0	15.2	13.5	14.0	15.5	15.8	17.9	15.2	14.1
3	21.1	15.3	13.5	14.1	15.6	15.8	18.0	15.2	14.2
4	21.3	15.3	13.7	14.2	15.7	15.9	18.0	15.2	14.4
5	21.4	15.4	13.7	14.3	15.8	16.0	18.1	15.3	14.5
6	21.5	15.4	13.8	14.5	15.9	16.1	18.2	15.3	14.6
7	21.6	15.5	14.0	14.6	16.0	16.2	18.3	15.4	14.8
8	21.7	15.5	14.1	14.7	16.1	16.2	18.4	15.4	14.9
9	21.9	15.6	14.2	14.9	16.2	16.3	18.4	15.5	15.0
10	21.9	15.7	14.3	15.0	16.3	16.4	18.5	15.6	15.2
11	21.5	15.7	14.4	15.1	16.4	16.5	18.6	15.6	15.3
12	21.6	15.8	14.5	15.3	16.4	16.6	18.6	15.7	15.4
13	21.8	15.9	14.6	15.4	16.6	16.7	18.7	15.8	15.6
14	22.0	15.9	14.7	15.5	16.7	16.7	18.7	15.8	15.7
15	22.0	16.0	14.8	15.7	16.7	16.8	18.8	15.9	15.8
16	22.1	16.1	14.9	15.8	16.8	16.9	18.8	16.0	16.0
17	22.2	16.2	15.0	15.9	16.9	17.0	18.9	16.1	16.1
18	22.4	16.3	15.1	16.1	17.0	17.0	19.0	16.1	16.2
19	22.5	16.3	15.2	16.2	17.1	17.1	19.0	16.2	16.4
20	22.6	16.4	15.4	16.3	17.2	17.2	19.1	16.3	16.5
21	22.7	16.5	15.5	16.5	17.3	17.3	19.2	16.4	16.6
22	22.6	16.6	15.6	16.6	17.4	17.4	19.2	16.4	16.8
23	22.6	16.7	15.7	16.8	17.5	17.5	19.3	16.5	16.9



	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	GSW+FB	AFLN+FM	
	24	22.6	16.7	15.8	16.9	17.6	17.6	19.4	16.6	17.0
	25	22.5	16.8	15.9	17.1	17.6	17.6	19.4	16.7	17.2
	26	22.5	16.9	16.0	17.2	17.7	17.7	19.5	16.8	17.3
	27	22.5	17.0	16.1	17.4	17.8	17.8	19.6	16.8	17.5
	28	22.3	17.0	16.2	17.5	17.9	17.9	19.7	16.9	17.6
	29	22.2	17.1	16.3	17.7	18.0	18.0	19.7	17.0	17.7
	30	22.1	17.2	16.4	17.9	18.1	18.1	19.8	17.1	17.9
	31	22.0	17.3	16.5	18.0	18.2	18.1	19.9	17.1	18.0
	32	21.9	17.3	16.5	18.2	18.2	18.2	19.9	17.2	18.2
	33	21.8	17.4	16.6	18.4	18.3	18.3	20.0	17.3	18.4
	34	21.5	17.4	16.7	18.7	18.4	18.4	20.1	17.3	18.6
	35	21.4	17.5	16.8	18.9	18.5	18.5	20.2	17.4	18.8
	36	21.4	17.6	16.9	19.2	18.6	18.6	20.3	17.4	19.0
	37	21.3	17.6	17.0	19.5	18.7	18.7	20.4	17.5	19.2
	38	21.3	17.7	17.0	19.8	18.8	18.7	20.5	17.6	19.5
	39	20.9	17.7	17.1	20.2	18.8	18.8	20.6	17.6	19.8
	40	20.9	17.8	17.2	20.6	19.0	19.0	20.8	17.6	20.2
	41	20.9	17.8	17.3	21.0	19.1	19.1	20.9	17.7	20.7
	42	21.1	17.8	17.4	21.4	19.1	19.2	21.1	17.7	21.2
	43	21.3	17.9	17.5	21.9	19.2	19.3	21.2	17.7	21.8
	44	21.4	17.9	17.6	22.4	19.4	19.5	21.5	17.8	22.5
	45	21.6	17.9	17.7	22.8	19.5	19.7	21.7	17.8	23.3
	46	21.6	18.0	17.8	23.3	19.7	19.8	21.9	17.9	24.1
	47	21.6	18.0	17.8	23.3	19.7	19.8	21.9	17.9	24.1

And here we get an average per day

```
In [30]: # we go through each block
# select subset with that block
# group by per day to calculate mean
# then we combine subsets with each block vertically

blocks = set(temperature_df_transformed['Block'].tolist())
print(blocks)

subsets_per_block = []

for block in blocks:
    subset = temperature_df_transformed.loc[
        temperature_df_transformed['Block'] == block
    ]

    columns = subset.columns.tolist()
    columns.remove('Block')

    subset = subset[columns]

    subset = subset.groupby(['Day']).mean()

    subset['Block'] = block
    subsets_per_block.append(subset)

temperature_df_transformed = pd.concat(subsets_per_block)

{'A', 'C', 'Avr', 'B'}
```

```
In [31]: temperature_df_transformed.head(48)
```

Out[31]:

	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	G
Day								
0.0	20.650000	15.200000	13.350000	13.750000	15.350000	15.650000	17.750000	15.1
1.0	21.962500	15.950000	14.662500	15.479167	16.595833	16.691667	18.687500	15.8
2.0	21.616667	17.579167	17.041667	20.204167	18.791667	18.850000	20.700000	17.4
3.0	20.841667	18.425000	21.454167	31.091667	25.054167	25.254167	28.825000	18.3
4.0	22.195833	18.950000	27.762500	40.779167	35.320833	32.725000	36.979167	18.8
5.0	20.879167	20.120833	31.812500	49.304167	41.737500	37.270833	42.220833	19.5
6.0	19.258333	21.400000	33.412500	55.037500	45.366667	39.508333	44.912500	19.6
7.0	19.058333	23.641667	33.604167	57.608333	47.687500	40.645833	45.966667	19.3
8.0	19.291667	27.037500	33.204167	57.537500	49.666667	40.979167	46.000000	19.4
9.0	18.958333	30.975000	32.291667	56.808333	51.504167	40.958333	45.941667	20.1
10.0	19.029167	34.200000	31.170833	56.091667	52.683333	40.816667	45.879167	21.5
11.0	19.458333	36.387500	30.337500	55.650000	52.820833	40.800000	45.991667	23.5
12.0	19.891667	38.683333	30.020833	55.345833	52.245833	40.987500	46.445833	26.0
13.0	20.170833	41.583333	29.850000	55.004167	51.433333	41.254167	46.945833	28.1
14.0	20.745833	44.541667	29.866667	54.662500	50.754167	41.562500	47.404167	30.9
15.0	19.891667	46.183333	29.487500	53.816667	50.029167	41.470833	47.379167	33.1
16.0	19.516667	46.258333	28.912500	52.916667	49.487500	41.133333	46.858333	35.2
17.0	19.483333	45.012500	28.320833	52.137500	49.308333	40.687500	45.883333	37.7
18.0	19.345833	42.879167	27.854167	51.479167	49.370833	40.137500	44.525000	41.4
19.0	19.241667	39.879167	27.362500	50.316667	49.133333	39.291667	42.616667	45.1
20.0	19.554167	36.441667	27.083333	48.766667	48.941667	38.529167	40.541667	46.8
21.0	19.366667	33.033333	26.900000	46.612500	48.512500	37.858333	39.254167	46.9
22.0	19.545833	30.029167	26.795833	43.829167	47.708333	37.075000	37.675000	46.4

	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	G
Day								
23.0	19.733333	27.760870	26.786957	40.886957	46.617391	36.413043	35.500000	45.5
24.0	19.591667	25.995833	26.862500	37.725000	45.129167	35.729167	33.450000	43.4
25.0	19.700000	24.679167	26.950000	34.708333	43.366667	35.000000	31.683333	40.2
26.0	20.025000	23.820833	27.245833	32.204167	41.545833	34.387500	30.329167	36.4
27.0	20.300000	23.375000	27.695833	30.483333	39.650000	33.941667	29.387500	32.8
28.0	20.233333	23.095833	28.070833	29.220833	37.687500	33.483333	28.658333	29.5
29.0	19.133333	20.393750	27.212500	26.343750	35.525000	26.681250	27.875000	27.5
30.0	19.775000	18.805263	37.231579	26.736842	30.778947	21.705263	25.547368	24.1
31.0	19.654167	21.341667	49.366667	30.441667	30.475000	25.950000	24.808333	24.2
32.0	19.570833	23.504167	53.575000	33.608333	32.279167	28.812500	25.366667	26.1
33.0	19.479167	25.066667	55.058333	36.083333	33.266667	30.691667	25.591667	26.8
34.0	19.333333	25.758333	55.537500	37.337500	33.766667	31.870833	25.570833	26.4
35.0	19.291667	25.754167	54.704167	38.041667	33.783333	32.545833	25.475000	25.6
36.0	19.062500	25.441667	52.912500	38.637500	33.429167	32.862500	25.291667	24.7
37.0	18.858333	24.870833	50.550000	38.608333	32.495833	32.791667	24.970833	23.6
38.0	19.404167	24.579167	48.116667	39.920833	31.508333	32.866667	24.862500	22.8
39.0	19.391667	24.654167	45.695833	40.270833	30.733333	33.287500	25.037500	22.4
40.0	19.454167	24.866667	43.258333	39.458333	29.987500	33.725000	25.220833	22.1
41.0	19.625000	25.183333	40.908333	38.325000	29.175000	34.170833	25.404167	21.9
42.0	19.733333	25.670833	38.933333	37.241667	28.479167	34.604167	25.604167	21.8
43.0	19.762500	26.258333	37.316667	36.270833	27.858333	34.925000	25.800000	21.7
44.0	20.125000	26.945833	35.750000	35.208333	27.237500	34.937500	25.933333	21.1
45.0	20.108333	27.929167	34.429167	34.341667	26.854167	34.900000	26.195833	21.8

	Room	GSW+FM	AFLN+MU	AFLN+FB	AFLH+FM	AFLH+MU	AFLH	GSW
Day								
46.0	19.879167	29.033333	33.195833	33.408333	26.550000	34.770833	26.395833	22.0
47.0	20.000000	28.100000	31.041667	32.000000	26.170167	34.000000	26.100000	22.0

## Change of Requirements

We don't need the treatments below anymore due to change of requirements. Therefore we can drop them

```
In [32]: not_needed_treatments = ['AFLH+FM', 'AFLH+FB', 'AFLH+MU', 'AFLH']
```

```
In [33]: temperature_df_transformed.drop(
          columns = not_needed_treatments,
          inplace = True
        )
```

```
In [34]: temperature_df_transformed.columns
```

```
Out[34]: Index(['Room', 'GSW+FM', 'AFLN+MU', 'AFLN+FB', 'GSW+FB', 'AFLN+FM', 'GSW+MU',
              'Relative_time', 'Hours', 'Block'],
              dtype='object')
```

We need to change the treatments that begin with "AFLN" to "AF" to correspond to the text in the article

```
In [35]: temperature_df_transformed = temperature_df_transformed.rename(
          columns = {
              'AFLN+MU': 'AF+MU',
              'AFLN+FB': 'AF+FB',
              'AFLN+FM': 'AF+FM',
          }
        )
```

We also need to change the treatments that begin with "GSW" to "GS"

```
In [36]: temperature_df_transformed = temperature_df_transformed.rename(
          columns = {
              'GSW+FM': 'GS+FM',
              'GSW+FB': 'GS+FB',
              'GSW+MU': 'GS+MU',
          }
        )
```

## Visualize Temperature

Select treatments to visualize

```
In [37]: temperature_columns = [
          'AF+FB', 'GS+FB',
          'AF+FM', 'GS+FM',
        ]
```

```

    'AF+MU', 'GS+MU'
]

```

Add colors indicating each block(bottle)

```

In [38]: blocks = {
    'A': 'tab:olive', 'B': 'tab:olive', 'C': 'tab:olive', 'Avr': 'tab:orange'
}

```

Add markers for legend for blocks

```

In [39]: def plot_temperature(df, date_column, column, ax, plot_room_temperature):
    is_room_temperature_plotted = False

    if plot_room_temperature:
        columns = [column, date_column, 'Room']
    else:
        columns = [column, date_column]

    for block_name, color in blocks.items():

        data = df.loc[
            df['Block'] == block_name, columns
        ]

        block_alpha = 1 if block_name == 'Avr' else 0.8
        block_ls = 'solid' if block_name == 'Avr' else 'dotted'

        ax.plot(
            date_column,
            column,
            data = data,
            color = color,
            alpha = block_alpha,
            ls = block_ls
        )

        if plot_room_temperature and not is_room_temperature_plotted:
            ax.plot(date_column, 'Room', data = data, color = 'tab:gray')

        ax.set_title(column, loc = 'center', fontsize = 'medium')
        ax.set_ylabel('')
        ax.set_xlabel('Days')
        ax.grid(True)

        ax.set_ylim([10, 65])

        for label in ax.get_xticklabels():
            label.set_ha('right')
            label.set_rotation(30.)
            label.set_color('tab:gray')
            label.set_size(8)

```

```

In [40]: def plot_temperature_for_multiple_columns(
    df,
    date_column,
    title = 'Temperature per treatment',
    plot_room_temperature = True,
    columns = temperature_columns
):

```

```

n = math.ceil(len(temperature_columns) / 2)

fig = plt.figure(figsize = (11, 4 * n))
fig.subplots_adjust(hspace = 0.6, wspace = 0.5)

fig.suptitle(
    title,
    color = 'royalblue',
    fontsize = 16,
    y = 0.93
)

for i, col in enumerate(temperature_columns):

    ax = fig.add_subplot(n, 2, i + 1)
    plot_temperature(df, date_column, col, ax, plot_room_temperature)

```

In [41]: plot\_temperature\_for\_multiple\_columns(temperature\_df\_transformed, 'Hours')

### Temperature per treatment

