Exploraotry Data Analysis

Felipe Buchbinder, Young Kyung Kim and Shota Takeshima

Introduction

Our project's goal is to use reinforcement learning to build a model that can assist actors managing a power grid system to maintain the stability of power generation and distribution in the event of a perturbation caused by a random failure, targeted attack or unexpected increase (or decrease) in the demand for energy.

To this end, we'll use a simulator called City Learn.

Simulator description

City Learn is a simulation environment for energy demand response. In its simplest form, it considers a generator (that supplies energy) and a building (that consumes energy). Demand for energy varies according to time zone and time of day. The building can stock enery during periods of low-demand to use during periods of high-demand, but there's a limit to how much energy the building can stock. The building seeks to minimize a cost function which considers a series of factors: amount spent on energy, risk of shortage, risk of rampage, and others.

In a not-so-simple scenario, there are multiple generators and multiple buildings. An agent decides how much energy should be used by or stocked at each building, so as to minimize the cost function for the *network* (not any individual building). There are various different types of energy that can be considered, such as thermal energy, batteries, and air-to-water heat pumps. While they can all be stocked, they do have different prices.

A more technical description

City Learn simulates an environment where a set of N buildings are, at any given time t, in a state that is completely defined by 28 variables. These variables are of the following types:

- Temporal variables as “month”, “day”, or “hour”
- Temperature variables
- Humidity variables
- Diffuse solar radiation variables
- Storage variables(How much energy is stored)

We find it important to highlight that all but the last type of variable are related to the demand of energy. Indeed, the demand for energy varies through time, with temperature , humidity or with the speed with which energy dissipates. The last kind of variable, however, refers to storage. And storage is ultimately a "managerial" decision: it's how the agent choses to manage the network. Such choice has implications that deserve our attention: If a building stocks energy today, it is betting that energy today will be cheaper than in the future. Moreover, if it choses to use energy it has previously stocked, it is limited by the choice of how much it decided to stock in the past. Hence, the storage variables insert a path dependency into our problem.

As for the actions, there are only 2 actions possible for the agent to take

- Increasing or decreasing the energy in cooling storage
- Increasing or decreasing the energy in domestic hot water storage

Since each building has its own storage, such 2 actions are, in reality, 2 actions *per building*. In other words, the agent must decide how much energy will de put in storage (or taken out of storage) for each building.

Such agent may be a single agent, managing the entire network, or a set of agents, one responsible for each building, which act independently but can comunicate with each other.

We have considered a setting of one agent per building acting independently. For simplicity, however, we often refer to this agents in singular form.

Data description

Four the data generation, we gethered the dara we hereby describe. It consistes of 36 buildings distributed through 4 different climate zones (so we have 4 networks with 9 buildings each). For each building, we look at 6 variables:

1. The indoor temperature, in degrees Celsius
2. The average unmet cooling setpoing difference, in degrees Celsius. This is the average difference between the indoor temperatures and the cooling temperature setpoints in the different zones of the building.
3. The indoor relative humidity
4. The electric power consumed by equipments, in kWh
5. The DWH energy being used for heating
6. The cooling load

Each of this variables is simulated at every hour of a 12 month period. During this period, some days have daylight saving status and some have not, and this is captured in a variable that's also on our dataset.

This gives us a set of 36 time series. This series operate under the optimization policy of City Learn's default. We did not change anything on the optimizaiton policy, since this is merely an initial exploratory research. So its not like there's no network optimization going on, but it's not like we have designed the optimizer either. Ideally, by the time our project is over, we aim to be doing a better optimization than this.

Exploratory Data Analysis

In [154]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [155]:
```
#Load generated dataset
df = pd.read_csv("../00_data/simulation_data_for_EDA")
print(f'Our dataset has {df.shape[0]} rows')
```

```
Our dataset has 315360 rows
```

For each of the four climate zones considered, there are 78,840 observations.

In [156]:
```
for climate_zone in set(df.climate_zone):
    print(f'Observations for climate zone {climate_zone}: {df[df.climate_zone
== climate_zone].shape[0]}')
```

```
Observations for climate zone 1: 78840
Observations for climate zone 2: 78840
Observations for climate zone 3: 78840
Observations for climate zone 4: 78840
```

Means and standard deviation of variables of interest

In [145]:
```
#by time zone
(df.
 iloc[:,5:].
 drop(columns='building').
 groupby('climate_zone',as_index=False).
 agg(['mean','std']).
 round(2)
)
```

Out[145]:

	Indoor Temperature [C]		Average Unmet Cooling Setpoint Difference [C]		Indoor Relative Humidity [%]		Equipment Electric Power [kWh]		DHW Heating [kWh]		Cooling Load [kWh]	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
climate_zone												
1	23.42	1.20	0.05	0.20	50.76	11.24	11.06	8.65	2.69	3.85	28.50	40.92
2	22.52	1.38	0.06	0.25	42.55	13.43	11.18	8.71	2.74	3.96	19.60	31.50
3	22.27	1.45	0.04	0.18	41.26	14.35	11.04	8.49	2.76	4.05	20.79	35.23
4	21.89	1.41	0.07	0.29	34.88	14.49	11.34	8.58	2.84	4.00	14.38	27.26

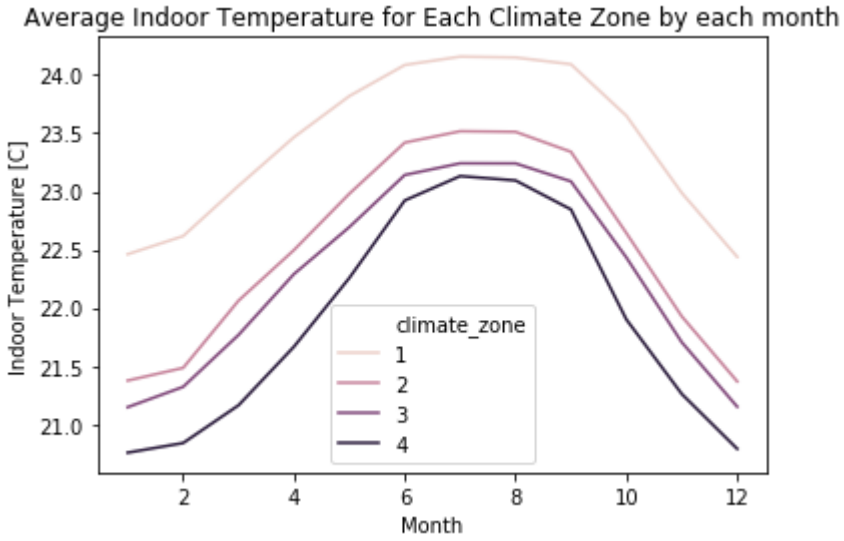
The table above shows us the mean and standard deviation for the variables considered in each climate zone. Sample sizes are so large that the sample errors for the mean are negligible, and we can take sample means as being very good proxies for population mean. However, we are hesitant to make any conclusions for this table because the standard deviation captures not only differences between buildings, but also differences accross time. As we'll show next, these differences can be quite considerable.

Time evolution

In [146]:
```
df_climate_season_mean = df.groupby(['climate_zone','Month']).mean().reset_index()
df_climate_season_mean = df_climate_season_mean[['climate_zone',"Month",'Indoor Temperature [C]', 'Average Unmet Cooling Setpoint Difference [C]',\
                                    'Indoor Relative Humidity [%]',"Equipment Electric Power [kWh]",\
                                     "DHW Heating [kWh]", "Cooling Load [kWh]"]]
```

Average indoor temperature

In [147]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="Indoor Temperature [C]", hue="climate_zone")
ax = ax.set_title('Average Indoor Temperature for Each Climate Zone by each month')
plt.savefig("../graphs/Average Indoor Temperature for Each Climate Zone by each month.png")
```

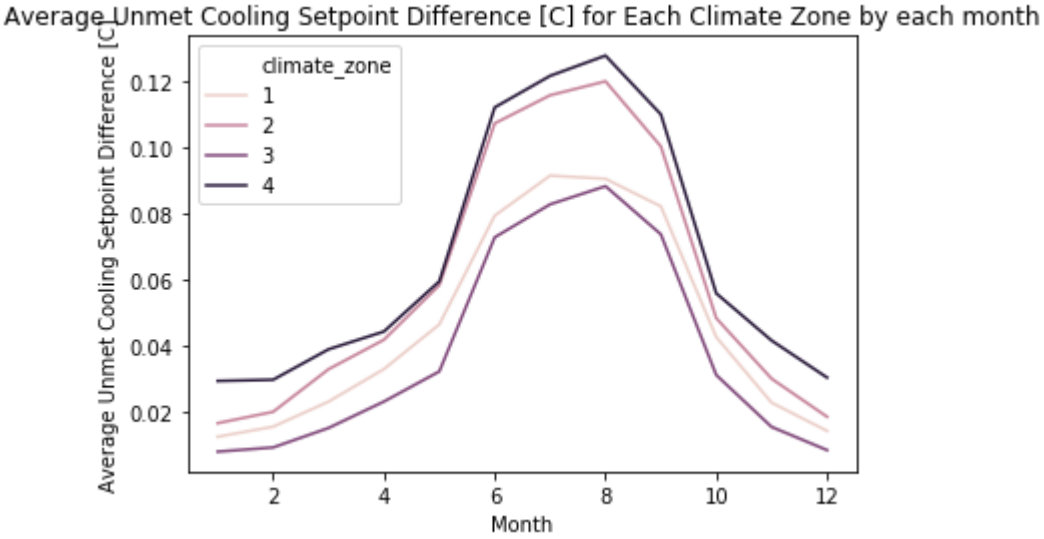


Indoor temperature varies considerably between months. This shows a substantial portion of the variance of the previous table may be due to variability throughout the year, rather than differences among buildings. Indeed, subsequent graphs will confirm these findings for other variables.

The above graph, however, also shows tha June-September is the warmer period in all time zones. This suggests the simulator is only considering time zones in the Northern Hemisphere, where it is Summertime during this period. We shall further need to read the documentation to see if this is indeed the case. This is not a problem, however, since it is only the sequence of months (but not their actual labels) that have any practical consequences for designing an optimal policy. However, it is something to be aware, that climate zones differ on their highest temperatures, but not on *when* their temperatures is the highest

Average Unmet Cooling Setpoint Difference

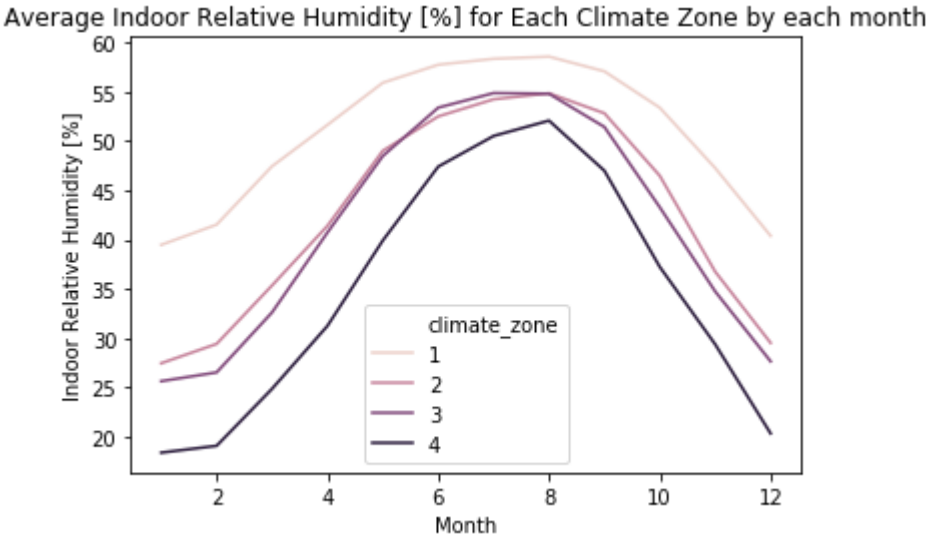
In [148]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="Average Unmet Cooling Setpoint Difference [C]", hue="climate_zone")
ax = ax.set_title('Average Unmet Cooling Setpoint Difference [C] for Each Climate Zone by each month')
plt.savefig("../graphs/Average Unmet Cooling Setpoint Difference [C] for Each Climate Zone by each month.png")
```



The same pattern observed for indoor temperature is also observed for the average unmet cooling setpoint difference. This is indeed expected, so such a match is welcome.

Average Indoor Relative Humidity

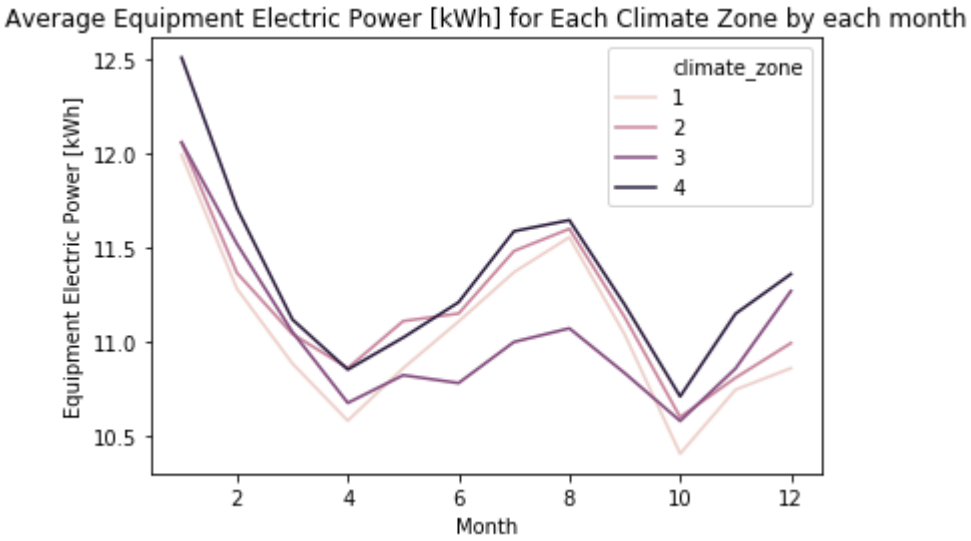
In [149]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="Indoor Relative Humidity [%]", hue="climate_zone")
ax = ax.set_title('Average Indoor Relative Humidity [%] for Each Climate Zone by each month')
plt.savefig("../graphs/Average Indoor Relative Humidity [%] for Each Climate Zone by each month.png")
```



Warmer months are also the most humid months for all climate zones. Note, however, that climate zones 2 and 3 are very similar in terms of humidity, but not in terms of temperature. Further in our project, it will be interesting to see how a difference in humidity alone affects the optimal policy for managing energy storage.

Average equipment electric power

In [150]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="Equipment Electri
c Power [kWh]", hue="climate_zone")
ax = ax.set_title('Average Equipment Electric Power [kWh] for Each Climate Zon
e by each month')
plt.savefig("../graphs/Average Equipment Electric Power [kWh] for Each Climate
Zone by each month.png")
```

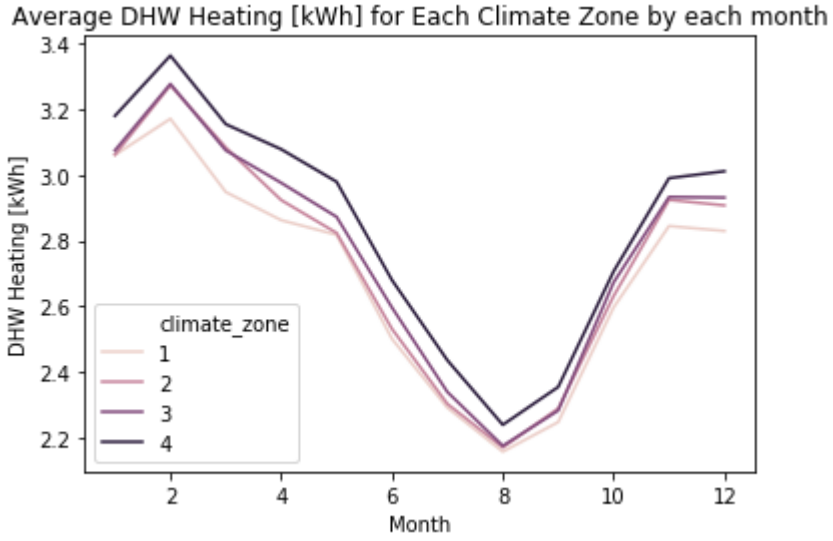


We here see a very different pattern than in previous graphs. Albeit different, this pattern is totally consistent with what we've seen before. When temperatures are lowest, during the start and end of the year, there's need for heating, so the average equipment electric power use is higher. We see this in months 1-3 and 11-12 in the graph above. Likewise, when the temperature is highest, during months 6-8, there's need for cooling, so we also have an increase on average equipment electric power use. This is also observed in the graph above. Between the harsh times of winter and summer, temperatures are mild, and there's no need for either cooling or heating. Thus, the average equipment electric power use has the two "valleys" we see in the graph above.

The two following graphs will confirm our reasoning.

Average DWH Heating

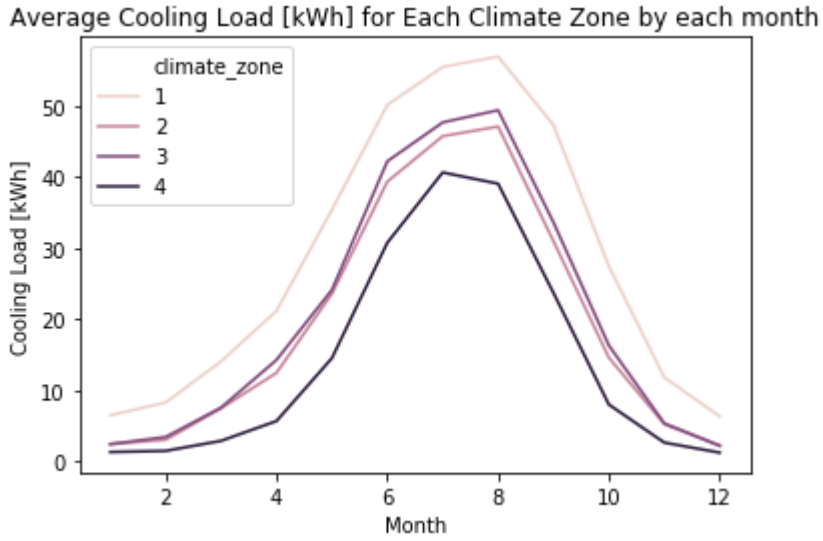
In [151]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="DHW Heating [kW
h]", hue="climate_zone")
ax = ax.set_title('Average DHW Heating [kWh] for Each Climate Zone by each mon
th')
plt.savefig("../graphs/Average DHW Heating [kWh] for Each Climate Zone by each
month.png")
```



Heating is most used when it is coldest, during months 1-3, and 11-12, and less when it is warmer, during months 6-8. This is in unisson with previous graphs and the story we have learned from them so far.

Cooling load

In [152]:
```
ax = sns.lineplot(data=df_climate_season_mean, x="Month", y="Cooling Load [kW
h]", hue="climate_zone")
ax = ax.set_title('Average Cooling Load [kWh] for Each Climate Zone by each mo
nth')
plt.savefig("../graphs/Average Cooling Load [kWh] for Each Climate Zone by eac
h month.png")
```

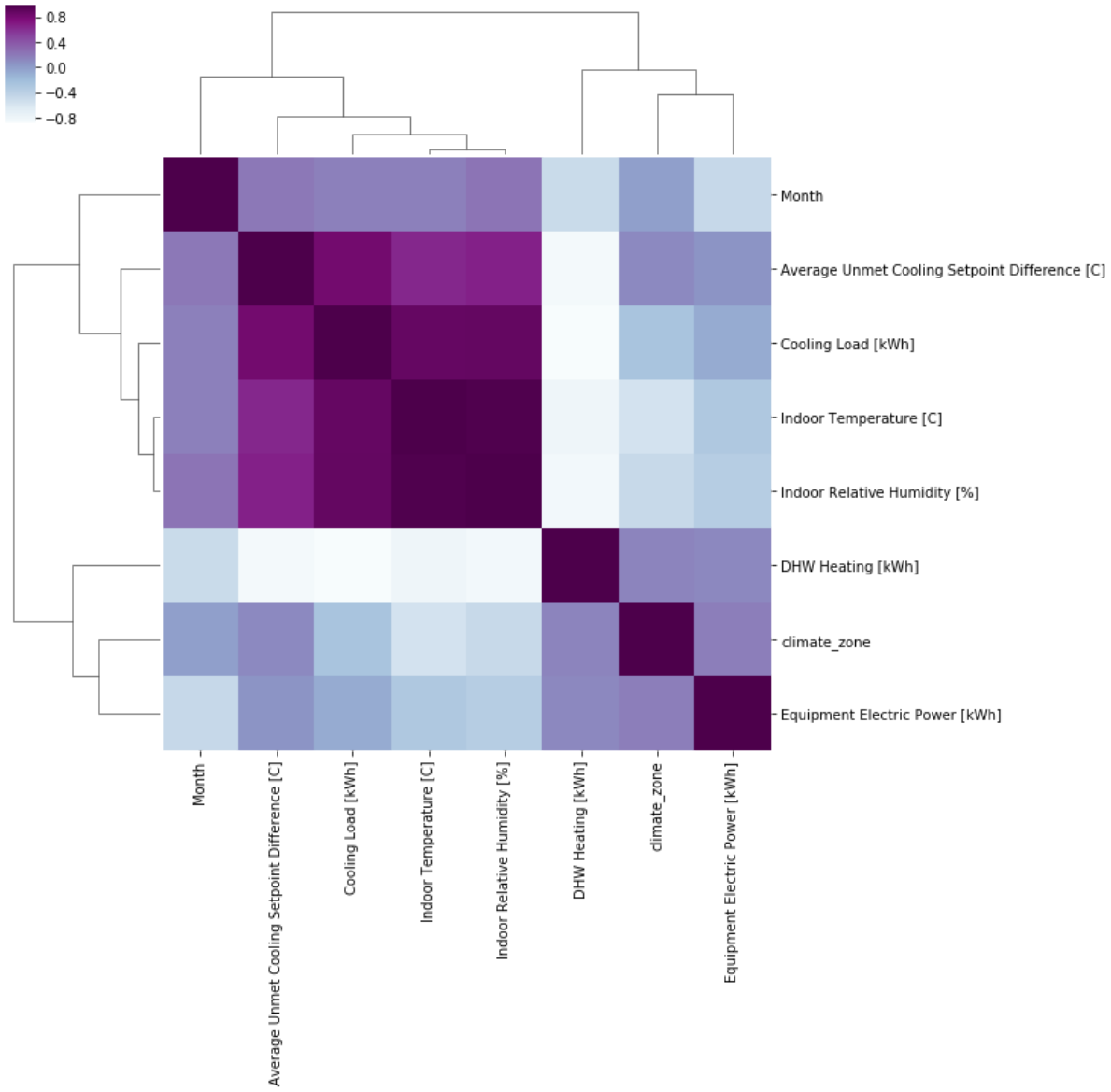


Cooling exhibits an opposite, but complementary pattern, to DWH heating. Cooling load rises during the warmer months (6-8) and is low during the rest of the year.

Correlations between variables

So far, we have only been analysing one variable at a time. In practice, the management of the power grid must consider all variables simultaneously. It thus helps to consider which variables are independent from each other, and which are not. We will not dive deep into exploring correlations between individual variables, since reinforcement learning operates on a model free environment. Yet, for our own, human, understanding, it's useful to take a look at a correlation plot. This is what we'll do in this section.

In [153]:
```
sns.clustermap(df_climate_season_mean.corr(), cmap='BuPu')
pass
```



Average unmet cooling setpoint difference, cooling load, indoor temperature and relative humidity seem all to be positively correlated with each other, and negatively correlated with DWH heating. While it comes as no surprise that cooling and heating are negatively correlated, the existence of a positive relationship between humidity and cooling (and a negative relationship between humidity and heating) is quite interesting. It is something to bear in mind throughout the development of the project.