

Sådan kører du programmet

Læs readme.md (i editor eller på github, <https://github.com/CholstTheDev/RoomCoolingSimulation>)

Plan for at bygge programmet

Først analyseres de nødvendige funktioner af programmet. De er:

- Simulere diskret tid i et kølerum
- Vurder om termostat skal tænde inden for hvert interval
- Bestem med tilfældighed om døren skal være åben
- Opdater temperaturen af rummet
- Udfør dette for en hel måned.
- Plot den data - og vurder

Ud fra dette, bestemmer jeg at lave programmet bestående af 4 moduler

- simulation.py (for at binde det sammen)
- cooler_instance.py (for at simulere kølerrummet)
- thermostat.py (for at indeholde logik for termostater)
- visualisation.py (alt med at plotte)

Implementation af termostater

Termostater er lavet ud fra en abstract class:

```
from abc import ABC, abstractmethod
```

```
class Thermostat(ABC):
    """
    Thermostat base class.
    """
    def __init__(self, config: dict):
        self.config = config

    @abstractmethod
    def evaluate_cooler_state(self, room: "cooler_instance.CoolerInstance") -> bool:
        """
        Returns whether the thermostat should be on or off
        """
        # dette er tomt, fordi det er en abstrakt metode - den skal implementeres i en klasse der
        # inheritor fra denne
```

Dette betyder at der nemt kan laves andre termostater og bruge dem, uden at skulle ændre i koden. Det forventes blot, at de alle bruger den samme struktur til `evaluate_cooler_state` når det implementeres. Her er implementationen af det simple termostat:

```
class SimpleThermostat(Thermostat):
    """
    Control mode: if the room temperature is above 5 degrees, turn on the compressor
    """
    def evaluate_cooler_state(self, room: "cooler_instance.CoolerInstance") -> bool:
        return room.current_temperature > 5
```

Simpelt termostat forbrug

Gennemsnitligt brugte det simple termostat **13954 kroner** om måneden på strøm, hvoraf 0 var til madspild. Det er ca. 2000 kroner over budget. Not good.

Intelligent termostater

Jeg lavede 6 intelligente termostater, som beskrives herunder. Alle modeller indeholder threshold, så de køber strøm hvis temperaturen når omkring 6.3 grader, og lader være med at købe omkring 3.5 grader.

Termostat	Funktion	avg. over 100 sim
OPPORTUNIST	Køber strøm når det er billigere end en hvis værdi	11.101 kr
DESPERATION	Er villig til at købe dyrere energi jo varmere der er i rummet	11.590 kr
DESPERATION_OPPORTUNIST	Blanding af de to forrige	10.918 kr
PEERREVIEW	Køber energi hvis det er billigere end det gennemsnitlige omkring den.	10.790 kr
PARTITION	Køber alt den billigste energi, inden for forudbestemte udsnit.	10.711 kr
DESPERATION_EXPONENTIAL	Er villig til at købe dyrere energi jo varmere det er i rummet. Exponentielt.	11.609 kr.

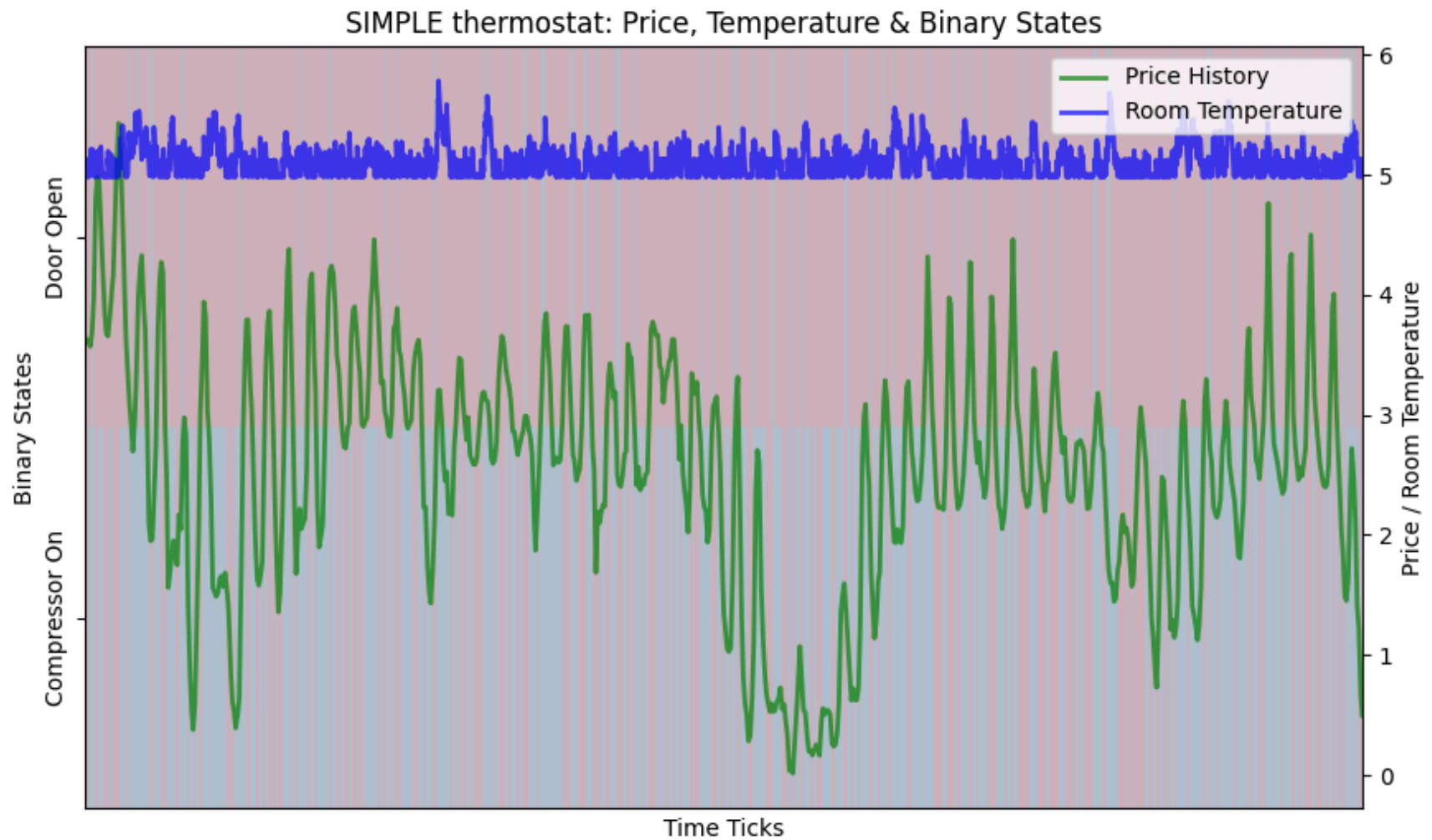
Den model der klarede sig bedst var `PARTITION`, men dens funktion er ikke særlig realistisk, da den kender hele månedens udgifter på "forhånd", derfor vil jeg hellere fokusere på `PEERREVIEW` modellen, der næsten klarede sig lige så godt.

Modellen fungerer ved at sammenligne intervallets (`n` i opgavesættet) pris, med priserne `look_around` bredde omkring den. Så køber den strømmen, såfremt:

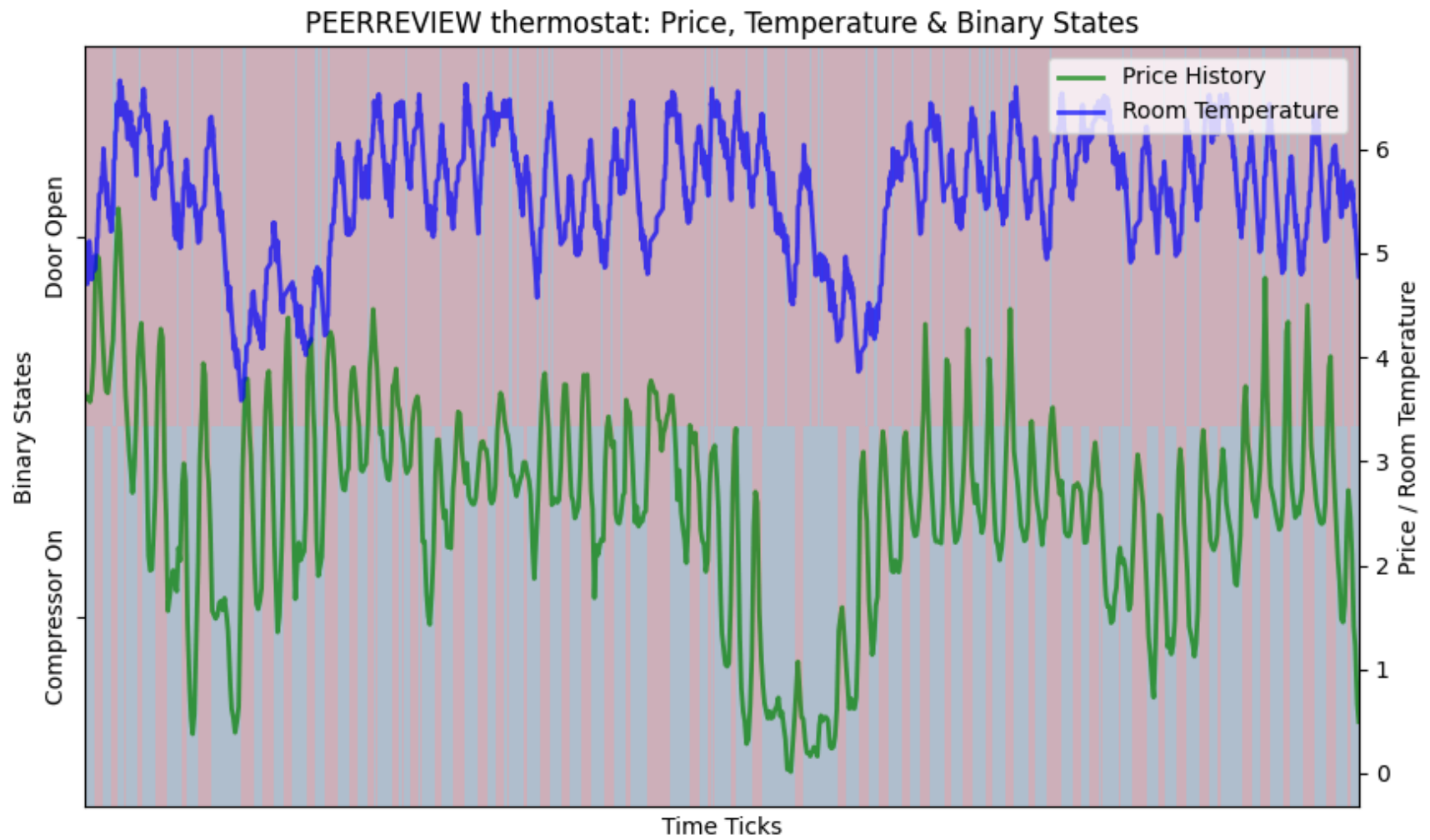
- Prisen er under gennemsnittet af de priser omkring den
- Temperaturen ikke er under 3.5
- Temperaturen er over 6.3

Plots

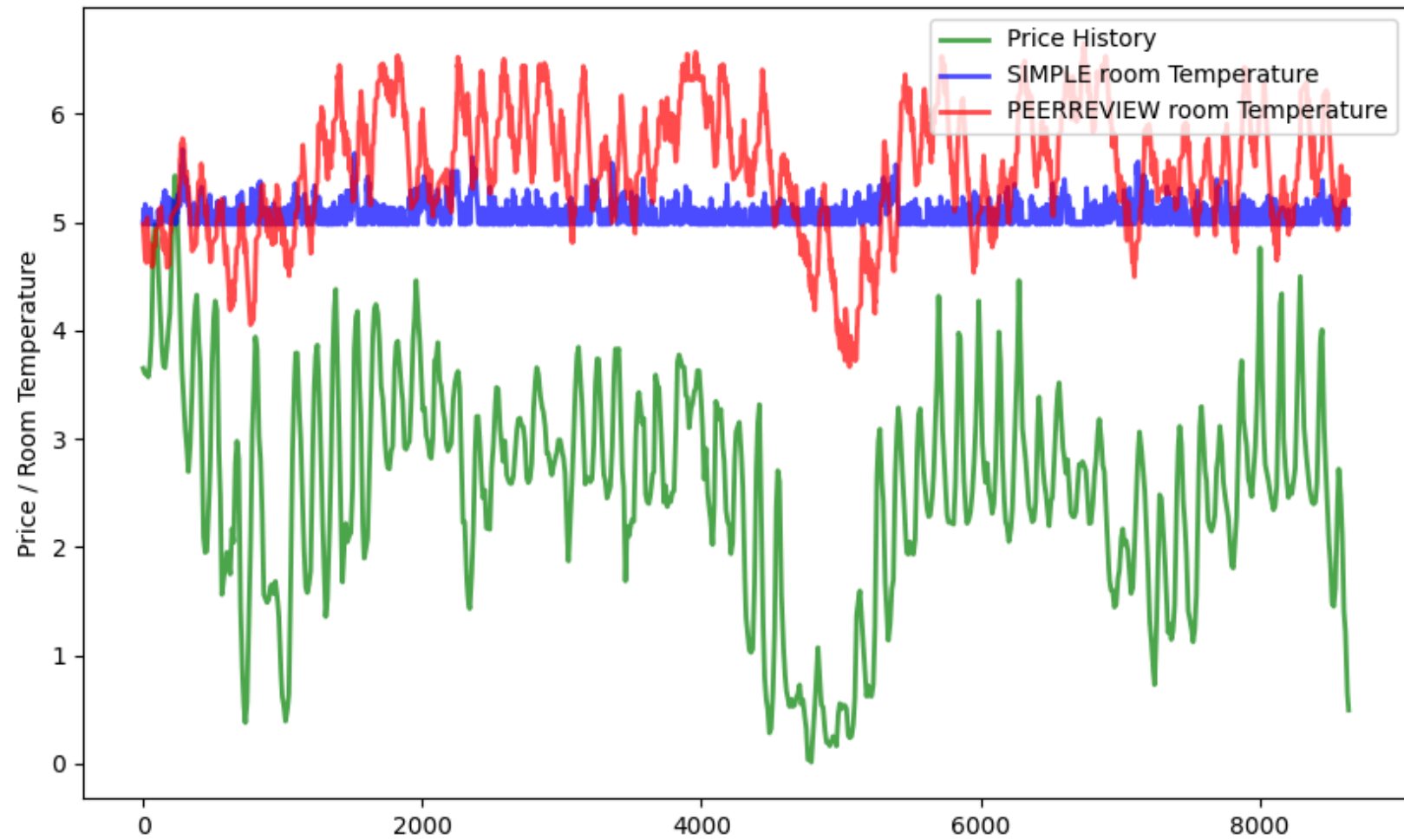
Jeg har brugt plots til at visualisere mine forsøg igennem projektet, for at kunne identificere når et termostats opførsel ikke er som forventet. Nedenunder ses et plot over temperaturen i rummet lagt sammen med prisen for `SIMPLE` termostadet.



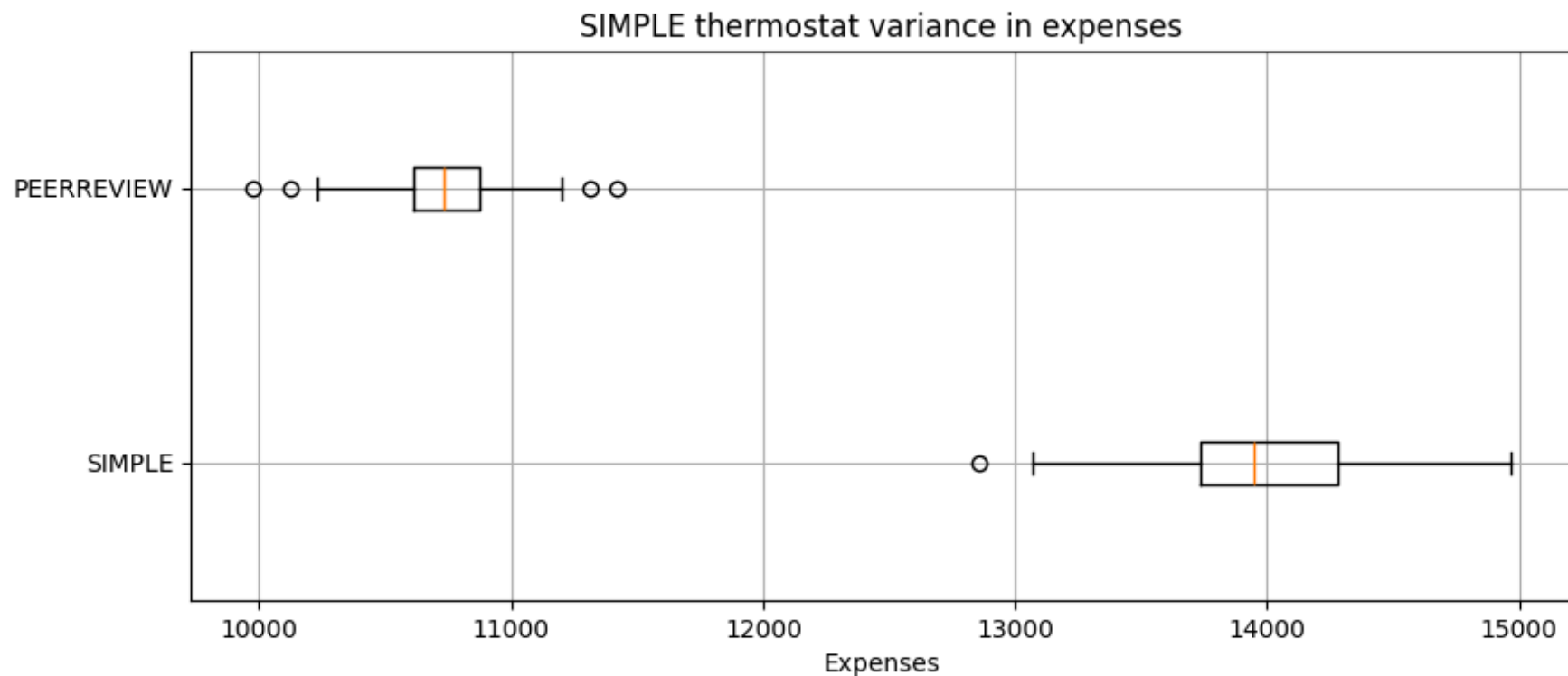
Det kan ses at der ikke rigtig bliver taget forbehold for om prisen er dyr eller ej, men blot at temperaturen aldrig rigtig løber ned under 5. Her ses det samme plot for `PEERREVEIW` modellen:



Det er tydeligt at se at den meget bedre tager prisfald i betragtning. Her er de sammenlagt:

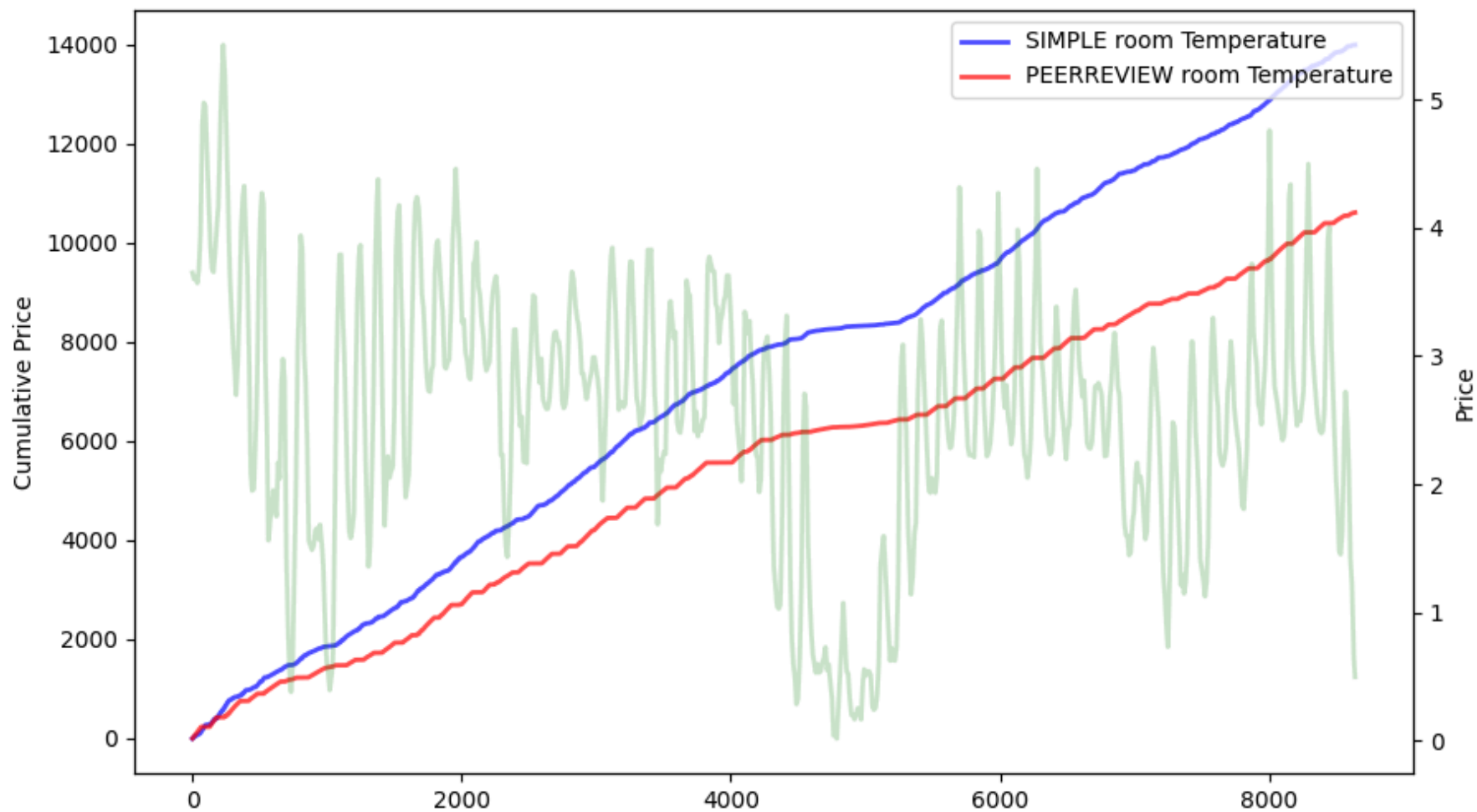


Det er dog også interessant at undersøge variansen i dataen. Det gør jeg med et boxplot.



Det kan ses at selv det simple termostat har virkelig dårlige måneder, hvor de koster op imod 14.750 kr. På PEERREVIEW modellen overstiges budgettet for måneden på intet tidspunkt, hvilket jeg ser som en succes.

For også at medtage et tredje plot, viser jeg en kumulativ sum, der sammenligner de to:



Her synes jeg det er spændende at se hvordan de to ligner hinanden, da de begge forholdsvis konsekvent køber strøm, men PEERREVIEW er bedre til at vælge de gange hvor det er billigt.