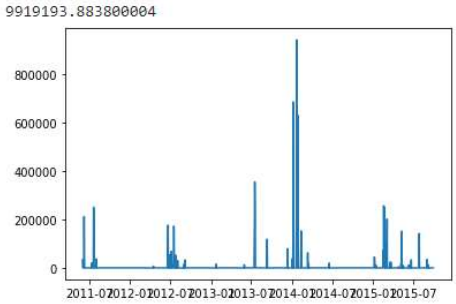
EN.553.753
Junzhou Lin

Project 3[Detail is in the coding provided]

After sorting out the data and creating an hourly headroom data frame. I had three stages of thinking processes.

Stage 1 (just to get a sense of feeling): I decided to take off the minhour=16 constraint, not consider turn on/off by hour, instead of by day. Then filter the revenue, if the next day's profit is positive, then I will stay on. If the next day's profit is negative, I will turn it off. [The 10k turn on cost is considered when calculating the profit]

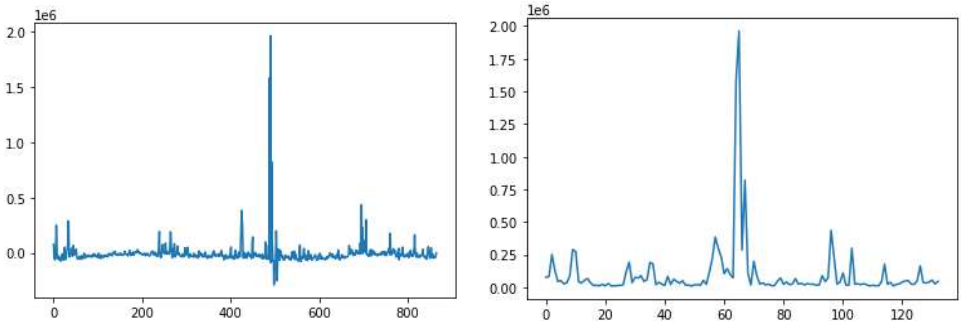


Therefore, if I robustly turn on and off by daily profit, I can at least secure $10m profit.

Now, I can add in the 16 minimum hour constraint and the profit should improve upon $9.92 mil.

Stage 2: Instead of using the headroom data frame, I create a headroom list with a length of 1583*24, since I don't really care about exactly when to turn on and off, as long as the time periods are connected.

After taking out all the possible periods with a positive hourly headroom and setting up a long and tedious algorithm [detail is in the .ipynb file], I have successfully increased the profit by $2m, which adds to $11.94m in total.



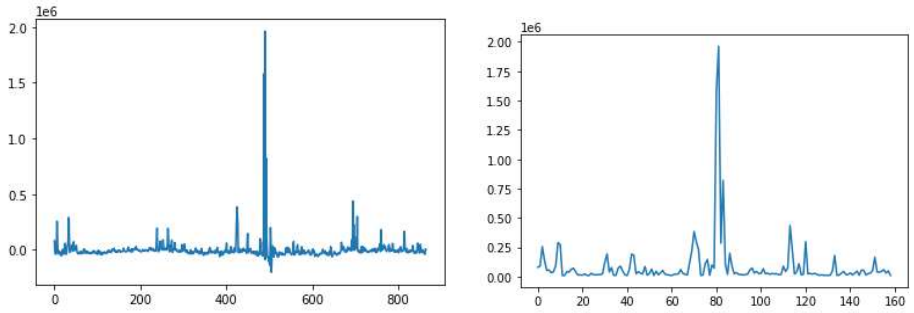
The left graph represents the return of all possible periods with at least one positive hourly headroom. The right graph represents all profitable periods.

Stage 3: Stage 2 finds the profit in a very straightforward way, but it's not the most optimal way.

```
sum(hrlist[-16:]),sum(hrlist[-17:-1]),sum(hrlist[-18:-2]),sum(hrlist[-19:-3])
```

```
(-16.163966, -2.3774539999999966, 6.824017000000005, 3.2723260000000067)
```

As I can see in the screenshot above, there is still room to improve our profit by shifting or extending the "ON" period to escape from a heavier loss and use a lighter loss instead. Moreover, I created an algorithm to have our code check whether it's more optimal to connect multiple periods or to split them to avoid some losses.



By doing so, I can increase the profit by $385k to $12.33m.

In the end, I think there is still room for development. Because in our code, if I encounter something like [-2,-20,[-2,10,20,30,-9,-6,-100,-10]], our code can only shift 1 hour at a time and would not try to escape from -100, because -10>-20. But it would take a tremendous amount of time and effort to write code like this, it wouldn't be worth it since I think the return afterward would only be increased by a bit.