1팀 중간 회의록

1차 목표

1. 각자 수립한 가설에 대해 간단한 분석 및 실험 진행
2. 어떤 컬럼(feature)이 모델 성능에 긍정적인 영향을 줄 수 있을지 개인적으로 탐색 결과를 가볍게 정리해서 공유 (간단한 수치나 시각화도 OK)

컬럼 정보

컬럼 정보

변수명	변수 설명	내용	단위
tm	시간	날짜 및 시각	00~23시
branch_ID	지사명	비식별화 처리 (위치 정보 제공 불가)	-
ta	기온	정시 기온	°C
wd	풍향	정시 10분 평균 풍향	degree
ws	풍속	정시 10분 평균 풍속	m/s
rn_day	일강수량	해당시간까지의 일강수량	mm
rn_hr1	시간 강수량	1시간 강수량	mm
hm	상대 습도	정시 상대 습도	%
si	일사량	ASOS 일사량	MJ/m^2
ta_chi	체감온도	500m 객관분석 자료	°C
heat_demand	열수요	시간당 지사별 열공급량	Gcal/h

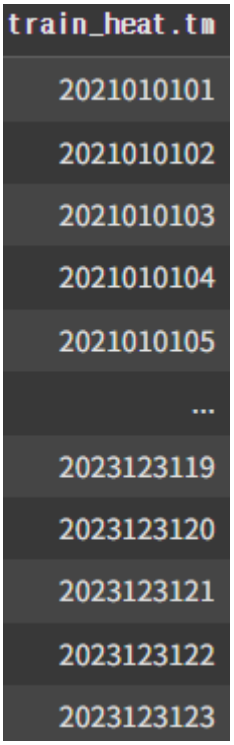
※ 기상 관측 장비의 오류나 고장 등의 이유로 관측이 진행되지 않은 경우 결측(미관측)으로 -99.0 표시

컬럼 개수 : 11개

컬럼별 샘플 수 : 499300개

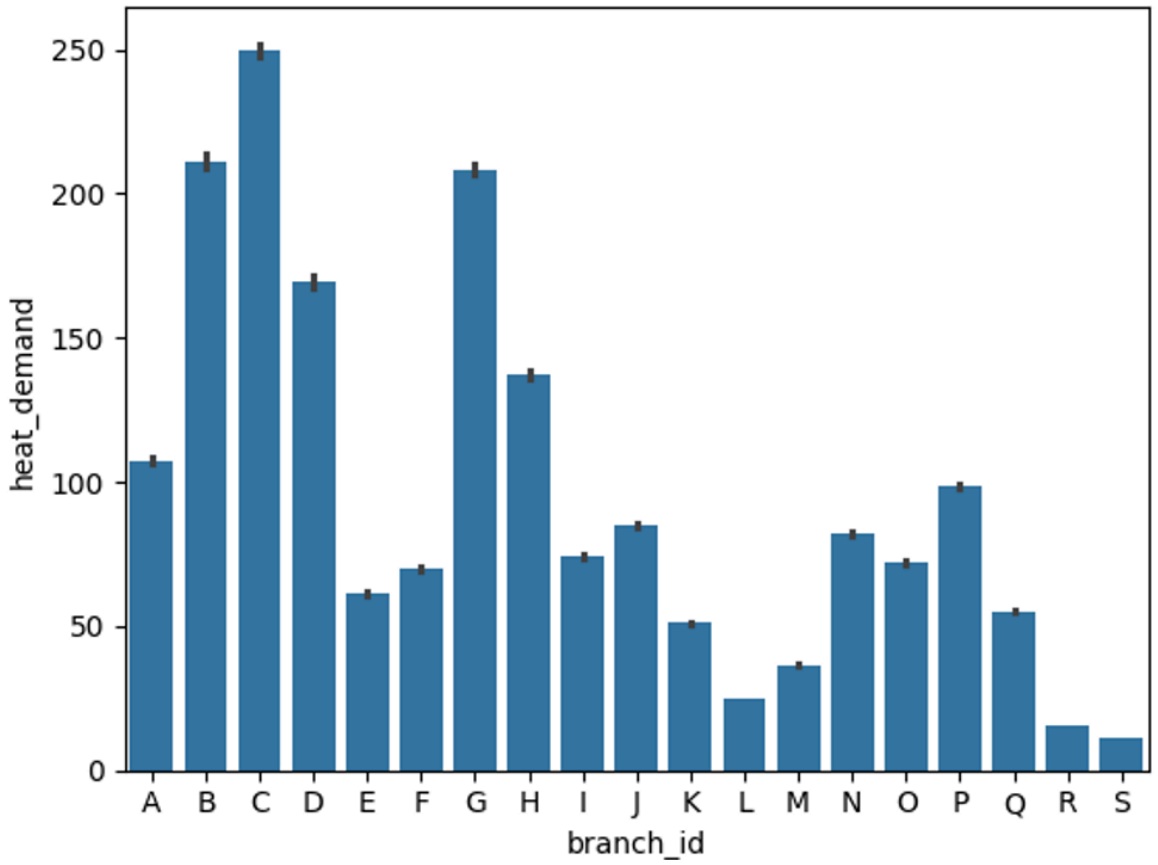
특이사항

- tm
  → int형. 년, 월, 일, 시간으로 구성. 파생 변수 생성 필요해보임.



train_heat.tm
2021010101
2021010102
2021010103
2021010104
2021010105
...
2023123119
2023123120
2023123121
2023123122
2023123123

- branch_ID
  → 지사명. object형. 컬럼별 순서쌍이 중요하지 않으니 원핫인코딩 필요



train_heat.branch_id
A
A
A
A
A

- si
  → 일사량. null값(-99.0)이 상당히 많음. 결측치 처리시 주의

-99.0 비율: 46.65%

- wd
  → 풍향. 음수값이 존재할 수 없으나 -9.9값 들어있음

1. 데이터 전처리

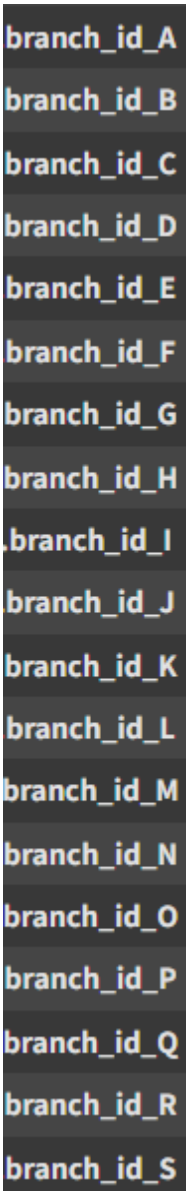
(1) 결측치 처리

- 결측치를 제거, 최빈값, 평균값으로 대체했을 때의 차이가 유의미하지 않음.
  단, Target값은 분리하여 결측치
  제거
  또한, 결측치가 너무 많은 일사량(si) 컬럼은 컬럼 자체를 제거.

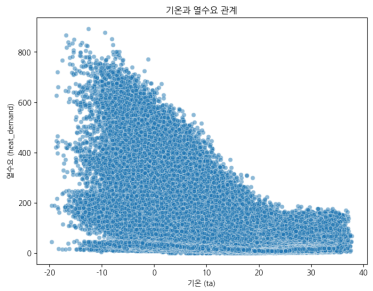
(2) 파생 변수 생성

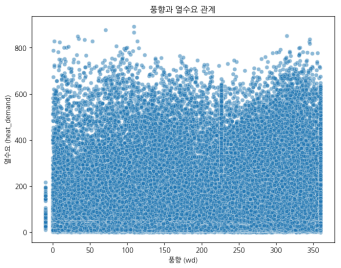
- tm(날짜)컬럼은 year, month, day, hour으로 분리. 전략에 따라 season, week 컬럼 생성 가능

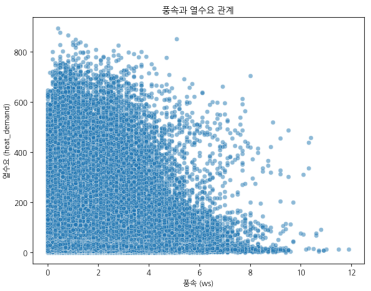
- branch_id(지사명) 컬럼은 **원핫인코딩** 이용해서 19개의 지점 컬럼으로 분리

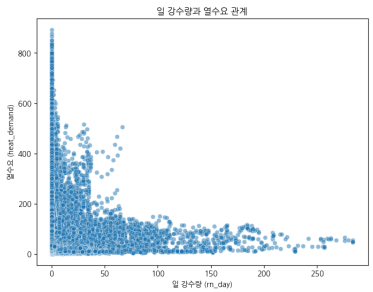

branch_id_A
branch_id_B
branch_id_C
branch_id_D
branch_id_E
branch_id_F
branch_id_G
branch_id_H
branch_id_I
branch_id_J
branch_id_K
branch_id_L
branch_id_M
branch_id_N
branch_id_O
branch_id_P
branch_id_Q
branch_id_R
branch_id_S

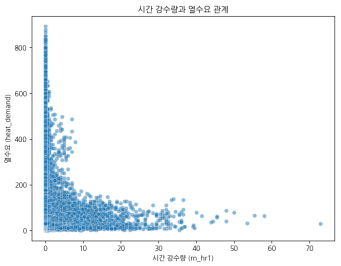
2. 시각화

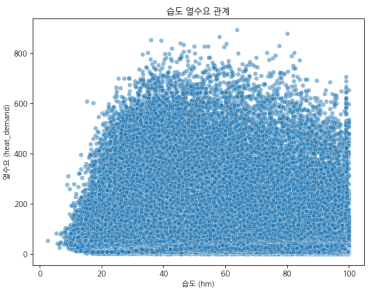


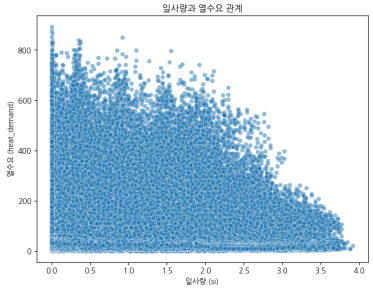


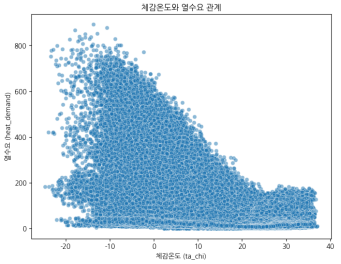


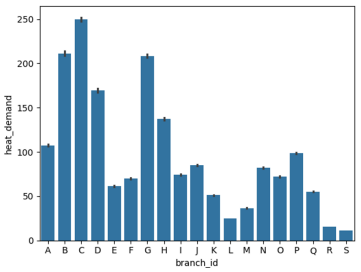


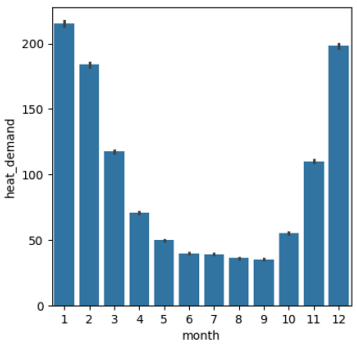


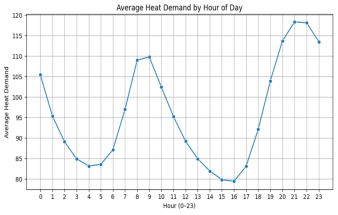


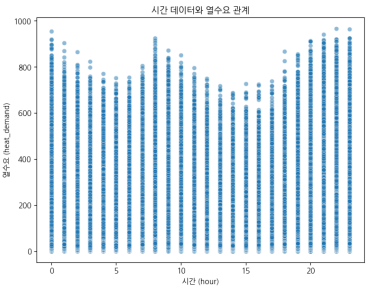






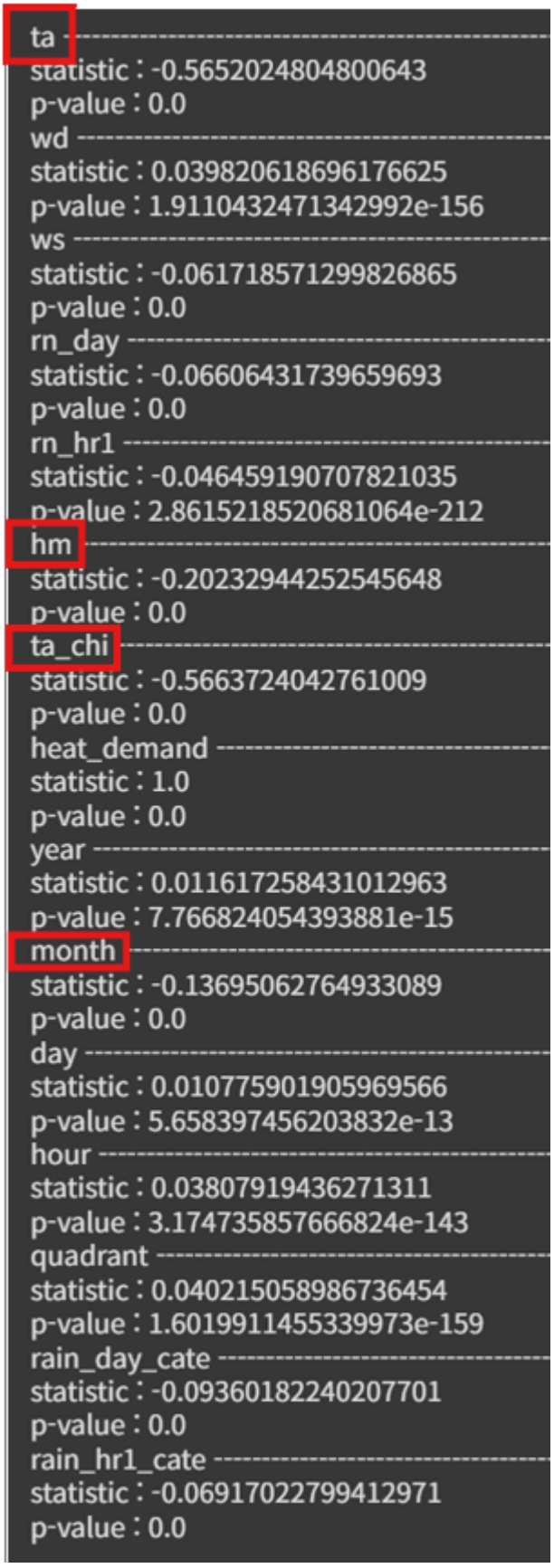






3. 변수 중요도

- 상관분석



- 분산분석(branch_id)

```
* f-statistic: 14671.508504394998
* p-value: 0.0
```

Tier	변수명	유형	분석 결과	설명
1	`season`	ANOVA	F=66,466, p=0.0	계절은 열 수요 패턴에 가장 큰 영향을 미침
	`train_heat.ta_chi`	상관분석	r=-0.582, p=0.0	체감온도는 열 수요와 강한 음의 선형 관계
	`train_heat.ta`	상관분석	r=-0.580, p=0.0	실제 기온도 체감온도만큼 예측력 있음 (다중공선성 주의)
	`heat_index_like`	상관분석	r=-0.543, p=0.0	온도+습도 기반 파생 피처로 높은 상관성
	`month_cos`	상관분석	r=+0.451, p=0.0	월 주기성 cos 변환, 강한 양의 관계

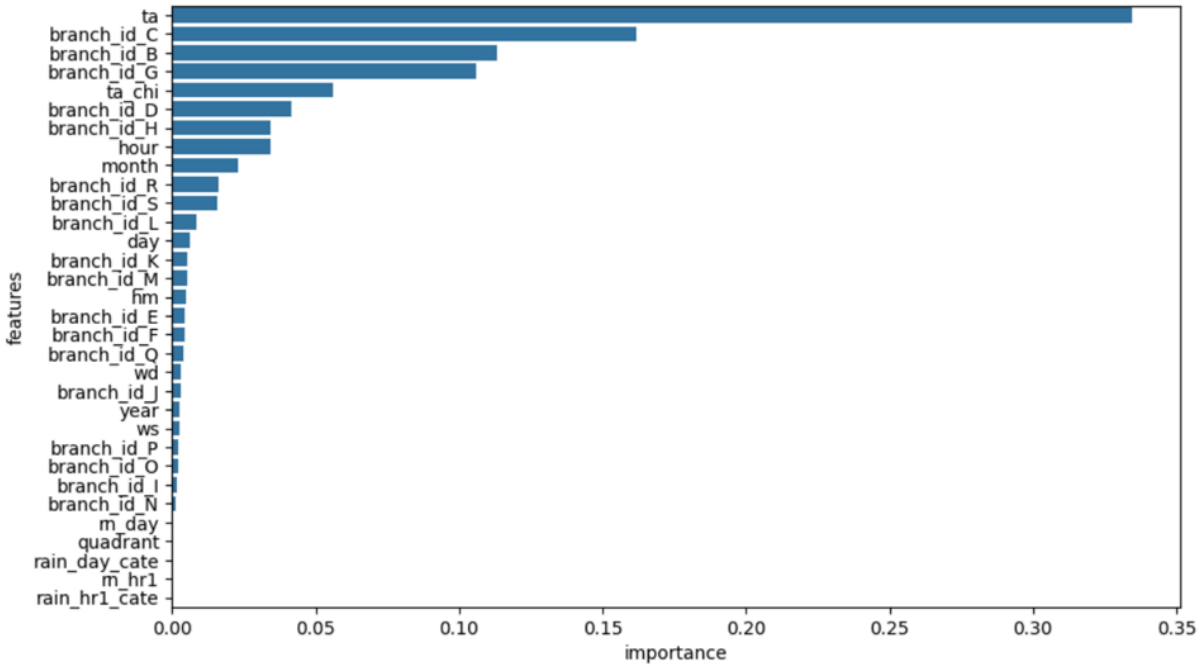
Tier	변수명	유형	분석 결과	설명
2	`heat_demand (지연값)`	ANOVA	F=14,671, p=0.0	이전 열 수요는 현재 예측에 중요한 피처
	`month_sin`	상관분석	r=+0.290, p=0.0	월 주기성 sin 변환, 유의미한 예측력
	`interaction_month2`	상관분석	r=+0.196, p=0.0	상호작용 파생 변수, 양의 관계
	`interaction_month1`	상관분석	r=+0.184, p=0.0	다른 월 상호작용 파생 변수
	`train_heat.hm`	상관분석	r=-0.161, p=0.0	상대습도도 예측에 유의미한 음의 상관성

Tier	변수명	유형	분석 결과	설명
3	`month` (숫자형)	상관분석	r=-0.160, p=0.0	sin/cos 변환 피처가 더 유리, 원본은 제거 가능
	`train_heat.si`	상관분석	r=-0.110, p=0.0	일사량도 일정 영향 있음

Tier	변수명	유형	분석 결과	설명
4	`train_heat.rn_day`	상관분석	r=-0.066, p=0.0	일일 강수량, 약한 음의 상관성
	`train_heat.wd`	상관분석	r=+0.057, p=0.0	풍향 원본, sin/cos 변환 추천
	`train_heat.rn_hr1`	상관분석	r=-0.046, p<0.001	시간 강수량
	`train_heat.ws`	상관분석	r=-0.045, p<0.001	풍속
	`hour_cos`	상관분석	r=+0.042, p<0.001	시간 주기성 cos 변환
	`hour`	상관분석	r=+0.037, p<0.001	원본 시간, sin/cos 추천
	`interaction_hour1`	상관분석	r=-0.031, p<0.001	시간 관련 상호작용 피처
	`train_heat.wd_cos`	상관분석	r=+0.030, p=N/A	풍향 cos 변환
	`train_heat.wd_sin`	상관분석	r=-0.028, p=N/A	풍향 sin 변환

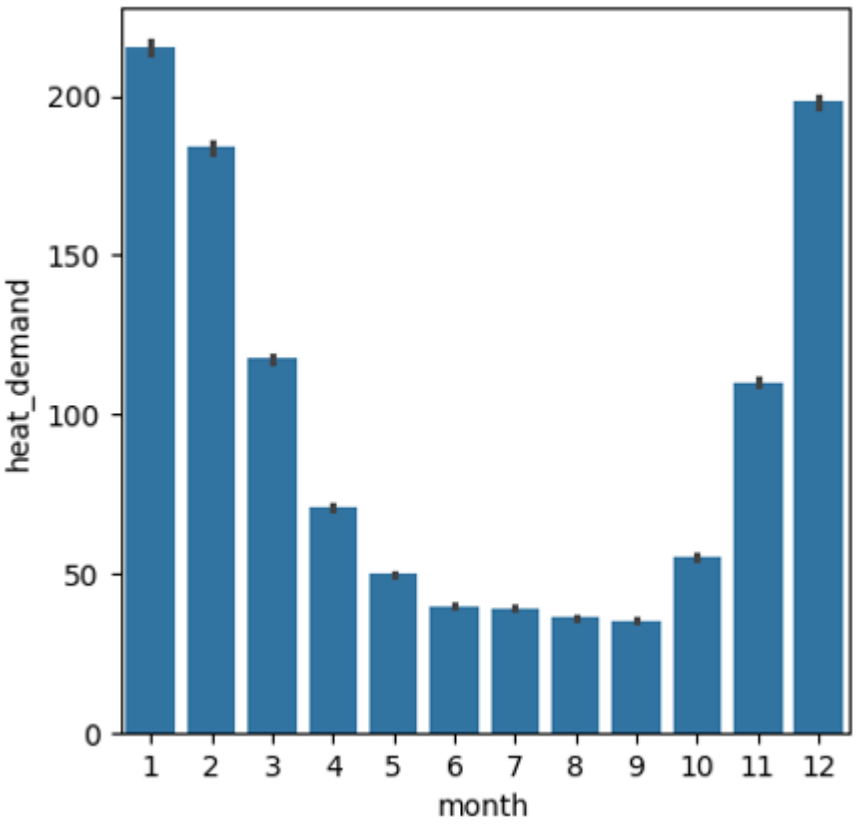
Tier	변수명	유형	분석 결과	설명
5	`interaction_hour2`	상관분석	r=-0.015, p<0.001	매우 낮은 관계지만 유의성 있음
	`hour_sin`	상관분석	r≈-0.007, p<0.001	시간 sin 변환, 영향도 매우 작음

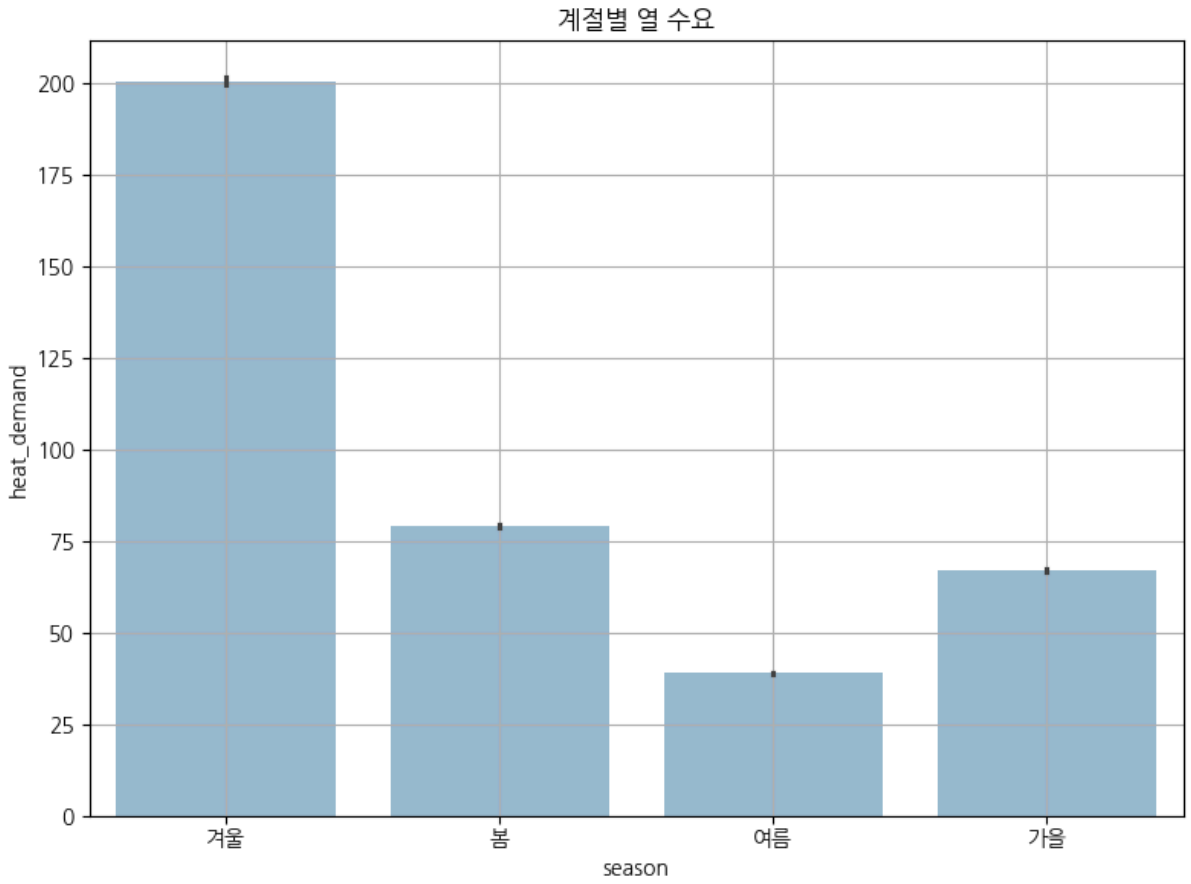
- Feature Importance 분석 (RandomForest)



4. 가설 검증

(1) 월 데이터(month)는 열 수요와 관련이 없을 것이다.



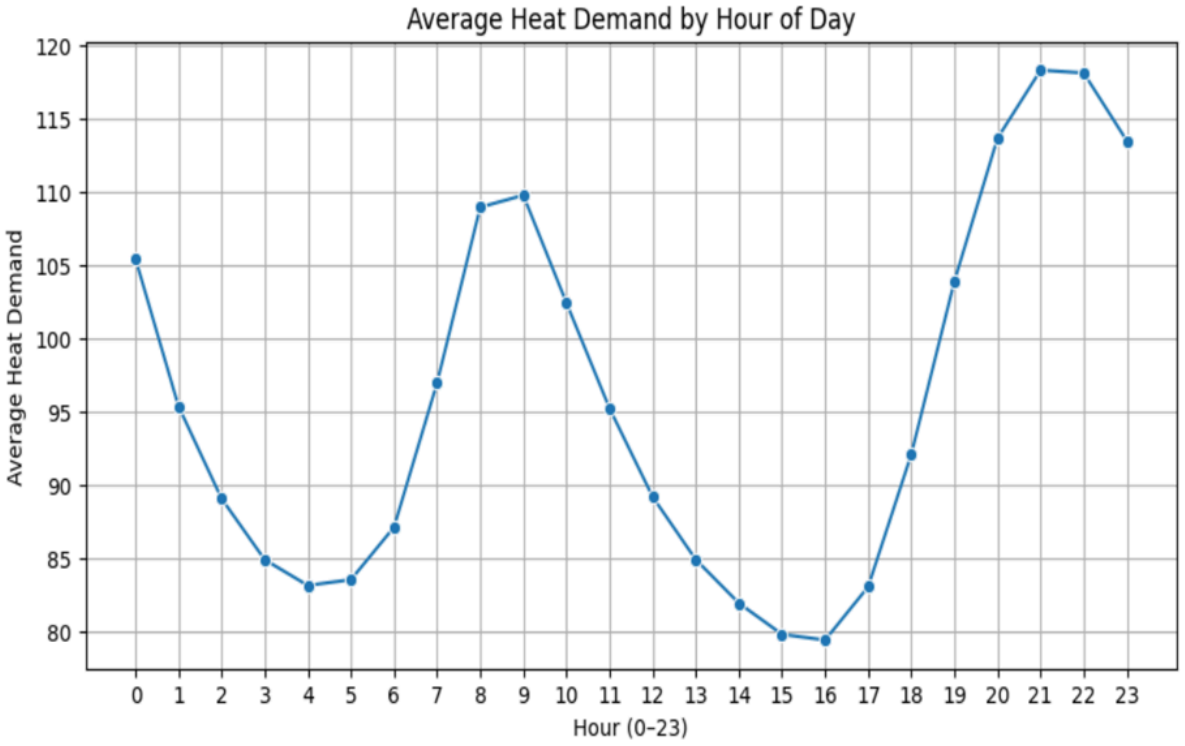



```
ANOVA = spst.f_oneway(df_date[df_date['month']==1]['heat_demand'], df_date[df_date['month']==2]['heat_demand'],
            df_date[df_date['month']==3]['heat_demand'], df_date[df_date['month']==4]['heat_demand'],
            df_date[df_date['month']==5]['heat_demand'], df_date[df_date['month']==6]['heat_demand'],
            df_date[df_date['month']==7]['heat_demand'], df_date[df_date['month']==8]['heat_demand'],
            df_date[df_date['month']==9]['heat_demand'], df_date[df_date['month']==10]['heat_demand'],
            df_date[df_date['month']==11]['heat_demand'], df_date[df_date['month']==12]['heat_demand'])
print(f'통계량 : {ANOVA[0]}')
print(f'p-value : {ANOVA[1]}')
```

```
통계량 : 19216.16150999749
p-value : 0.0
```

→ 대립가설 채택. 즉, **열 수요는 월 데이터(month)와 관련이 있다.**

(2) 시간(hour) 데이터는 열 수요와 관련이 없을 것이다.




```
ANOVA = spst.f_oneway(df_date[df_date['hour']==0]['heat_demand'], df_date[df_date['hour']==1]['heat_demand'],
                      df_date[df_date['hour']==2]['heat_demand'], df_date[df_date['hour']==3]['heat_demand'],
                      df_date[df_date['hour']==4]['heat_demand'], df_data[df_date['hour']==5]['heat_demand'],
                      df_date[df_date['hour']==6]['heat_demand'], df_date[df_date['hour']==7]['heat_demand'],
                      df_date[df_date['hour']==8]['heat_demand'], df_date[df_date['hour']==9]['heat_demand'],
                      df_date[df_date['hour']==10]['heat_demand'], df_date[df_date['hour']==11]['heat_demand'],
                      df_date[df_date['hour']==12]['heat_demand'], df_date[df_date['hour']==13]['heat_demand'],
                      df_date[df_date['hour']==14]['heat_demand'], df_date[df_date['hour']==15]['heat_demand'],
                      df_date[df_date['hour']==16]['heat_demand'], df_date[df_date['hour']==17]['heat_demand'],
                      df_date[df_date['hour']==18]['heat_demand'], df_date[df_date['hour']==19]['heat_demand'],
                      df_data[df_date['hour']==20]['heat_demand'], df_date[df_date['hour']==21]['heat_demand'],
                      df_date[df_date['hour']==22]['heat_demand'], df_date[df_date['hour']==23]['heat_demand'])
print(f'통계량 : {ANOVA[0]}')
print(f'p-value : {ANOVA[1]}')
```

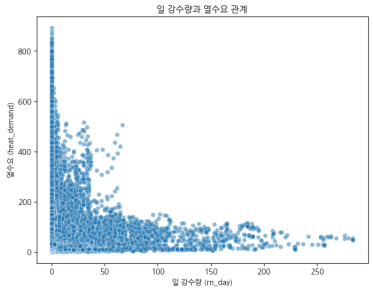
```
통계량 : 234.30559996105023
p-value : 0.0
```

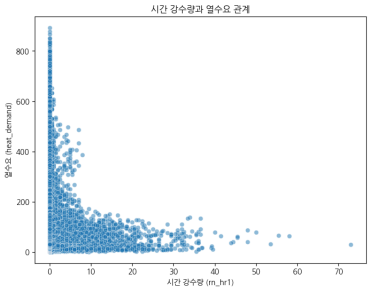
- 시간대별로 열수요에 명확하고 반복적인 패턴이 존재
  - 두 번의 높은 구간: 출근 시간대(7-9시), 저녁 시간대(19-22시)
  - 가장 낮은 구간: 새벽이른 아침 (2-5시) & 오후 (15-16시)

→ 대립 가설 채택. 즉, **열 수요는 시간 데이터(hour)와 관련이 있다.**

(3) 강수량 관련 데이터(rn_day, rn_hr1)은 열 수요와 관련 없을 것이다.

* 대부분의 데이터는 0(비가 안옴)에 치중되어 있음.






```
[66]  1 T_TEST = spst.ttest_ind(df_rain[df_rain['rain_day_cate']==1]['heat_demand'],
      2                         df_rain[df_rain['rain_day_cate']==0]['heat_demand'])
      3 print(f"통계량 : {T_TEST[0]}")
      4 print(f"p-value : {T_TEST[1]}")

      통계량 : -62.88865246941221
      p-value : 0.0
```

→ 대립 가설 채택. 즉, **강수 유무는 열 수요평균에 유의미하다.**

5. 모델 예측

```
12 rf_model.score(x_test,y_test)
0.9858534269933912
```

모델 $R^2-Score$ 0.98

```
1 sk.metrics.root_mean_squared_error(y_test,rf_model.predict(x_test))
13.693735459632249
```

모델$RMSE$ 13.69

```
[70]  1 sk.metrics.mean_absolute_error(y_test,rf_model.predict(x_test))
7.908438090129955
```

모델MAE 7.9

다음 목표

1. 중요도가 높았던 변수들 외에도 추가적인 엔지니어링 후, 최종적으로 영향력 높은 변수 선정하기
2. 다중공선성 고려하여 필요없는 변수 제거(ex 기온과 체감온도, 시간 강수량과 일 강수량 등)
3. 총 정리 후 모델 학습 해보기