제목	여름철 가정용 전력 하루 증감량 예상 데이터 셋 구축 성과 보고서		
날짜	08/06	팀원	6조 - 임희진, 우동주, 배송이, 최혜정

0. 개요

2010 ~ 2020년 여름철 기상 상태 및 여러 요인에 따른 전력 증감량 예상 데이터 셋을 구축함. 여름철의 경우, 냉방 기기 이용률이 대폭 증가하여, 하루 전력량 소모가 급격히 증가함. 이에 따라 전력 수요를 예측하기 힘들어 비용적인면의 증가뿐 아니라 전력 수급에 있어 매년 예측에 어려움이 있음. 이에 따라 기상 요인 및 그 외 다른 변수에 따른 하루 전력 증감량을 예상할 수 있도록 하려 함.

1. 활용 데이터

1) 활용 데이터

① 기상청 기상 관측 데이터

- 2010 ~ 2020년도 기상 관측 데이터 open api를 통해 수집
- 전국 88개 관측소의 관측치 수집 및 전국 단위 데이터로 변경(일별 평균)

[출처]: 공공데이터포털(https://www.data.go.kr/index.do)_ 종관기상관측 장비로 관측한 일 기상자료를 조회하는 서비스

② 한국 전력 통계 데이터

- 전체 전력량 중 월별 가정용 전력 데이터 비율 수집

[출처]: 한국 전력 공사(https://home.kepco.co.kr/kepco/main.do)_지식센터_ 전기자료

③ 일별 전력 사용량 데이터

- 전국 일별 사용 전력량 데이터에 월별로 가정용 전력 사용 비율을 대입하여 가정용 하루 사용량 수집

[출처]: smart kpx 전력 거래소(https://www.kpx.or.kr/)_부가메뉴_전력관련정보_전일 전력수급실적

2) 데이터 Value

① 전력 수요 예측을 하지 않을 시, 잉여 공급 예비력 양이 증가하게 됨. 이에 따라 **발전비용 증가 및 소모하지 못 하고 소실되는 전력량이 증가**하게 됨. 따라서 전력량을 예측함에 따라 발전비용 및 소실되는 전력량을 감소할 수 있도록 함.

② 발전 비용을 감소시키기 위해 적정 예비력 보다 적게 생산 시, 갑작스러운 전력수요의 변동, 수요예측 오차, 발전기 고장등의 상황에 대처가 불가능해짐. **대처가 불가능할 시, 전력 공급이 중단되며, 전국적인 대정전으로 이어질 수 있어 큰 혼란**이 일어날 수 있어 전력 예측을 통해 이를 방지할 수 있도록 함.

③ 전력은 저장이 불가능하여 생산과 동시에 소비가 이루어져야 함. 따라서 출력량을 미리 예측하여 발전량을 조절 해야하며, 이를 위해 기상상태 및 다른 변수에 의해 당일 발전량 예측 데이터 셋을 구축함. 이에 따라 발전 비용 감소 및 전력수요의 변동을 예측할 수 있도록 함.

2. 데이터 셋

1) 데이터 셋

① 종속 변수

- **전년 동기 평균 대비 하루 소모 전력 증감량(pct)**
- (일 소모 전력(10,000kw단위) - 전년 동기 동월 평균량) / 전년 동기 동월 평균량

② 독립변수

- 기상 데이터의 경우, 88개 관측소의 관측치를 평균값으로 산출해, 전국 평균 관측치로 데이터 셋을 구축함.

Columns name	설명
Avgrhm	전년 동기 평균 대비 **평균 습도 차이**
Avgts	전년 동기 평균 대비 **평균 지면 온도 차이**
Maxta	전년 동기 평균 대비 **최고 온도 차이**
Sumrn	전년 동기 평균 대비 **하루 강수량 차이**
Day	**주중: 1 / 주말, 공휴일: 0 구분**
Olympic	**올림픽 진행 여부** **진행: 1 / 미진행: 0**

-> 각 데이터 별 비교 분석을 위해 Day, Olympic 제외 Standard Scaler로 정규화 진행

3. 데이터 탐색

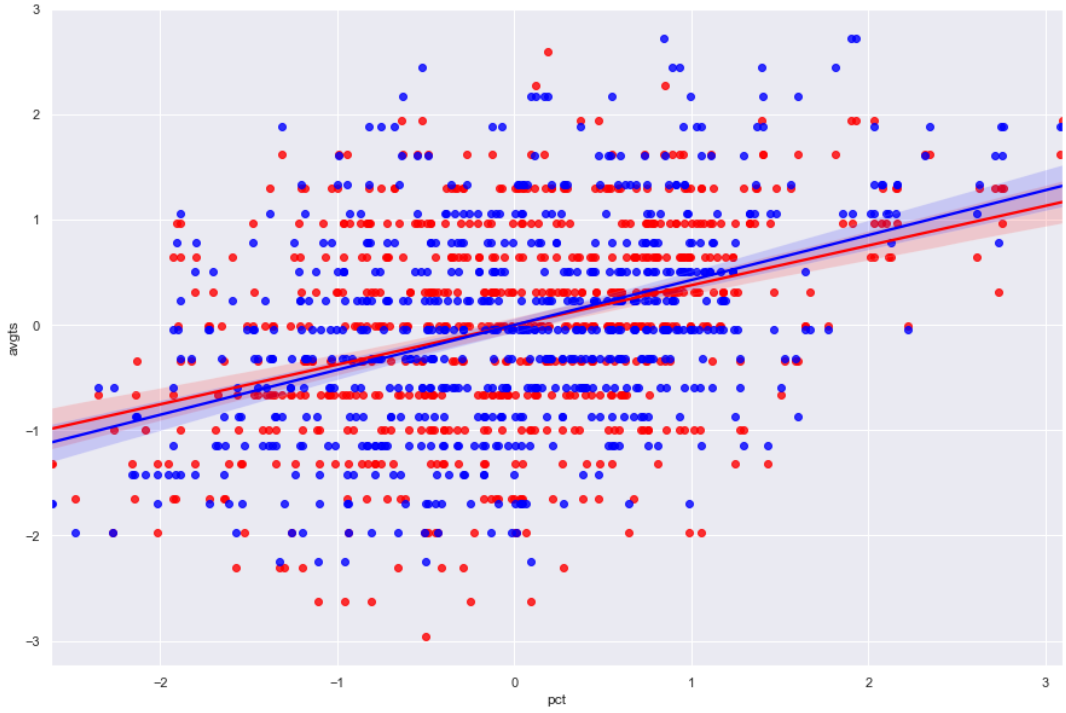
1) 데이터 상관관계

	avgrhm	avgts	maxta	sumrn	pct	day	olympic
avgrhm	1.000000	-0.552979	-0.541246	0.651349	**-0.204373**	-0.000370	-0.237291
avgts	-0.552979	1.000000	0.945074	-0.433738	**0.426574**	-0.001947	0.362062
maxta	-0.541246	0.945074	1.000000	-0.455402	**0.377025**	0.004194	0.291320
sumrn	0.651349	-0.433738	-0.455402	1.000000	-0.055489	-0.026570	-0.139657
pct	-0.204373	0.426574	0.377025	-0.055489	1.000000	0.522925	0.187690
day	-0.000370	-0.001947	0.004194	-0.026570	**0.522925**	1.000000	-0.024430
olympic	-0.237291	0.362062	0.291320	-0.139657	**0.187690**	-0.024430	1.000000

- avgts, maxta, day의 상관계수가 다른 요소들의 비해 크게 나타남.
- Avgrhm, Olympic의 상관계수는 약하지만 어느정도 영향이 있는 것으로 판단됨.

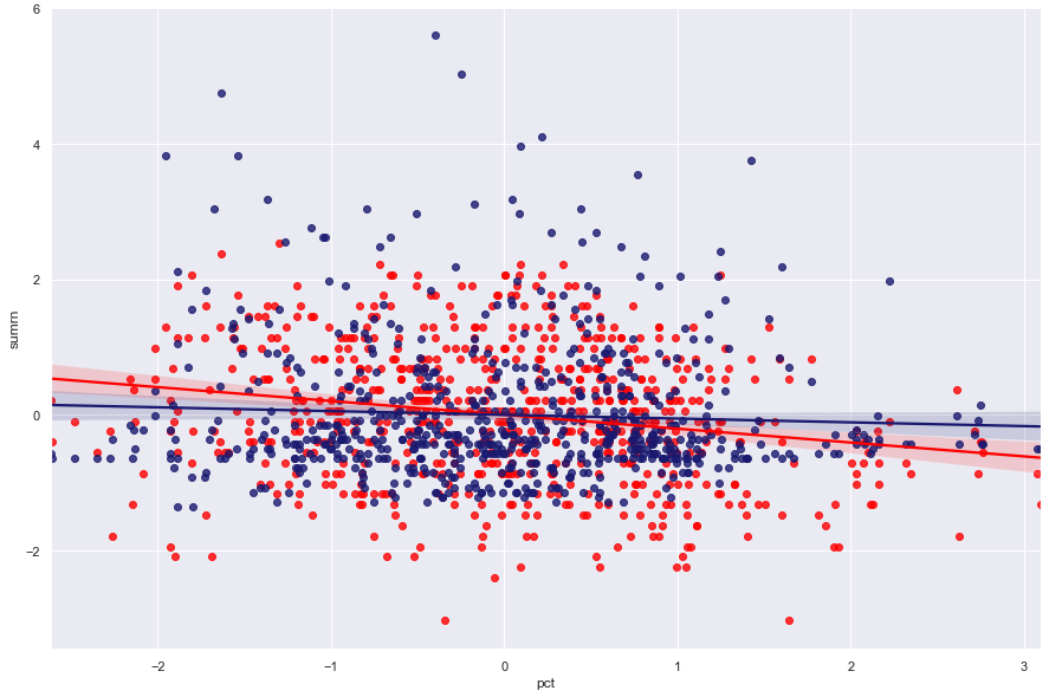
① Avgts & Maxta

- 두 요소 모두 강하지는 않지만 일정한 양의 상관관계를 보이고 있음.
- **sns.regplot(x='pct',y='maxta',data=df,color='red')**
- **sns.regplot(x='pct',y='avgts',data=df,color='blue')**

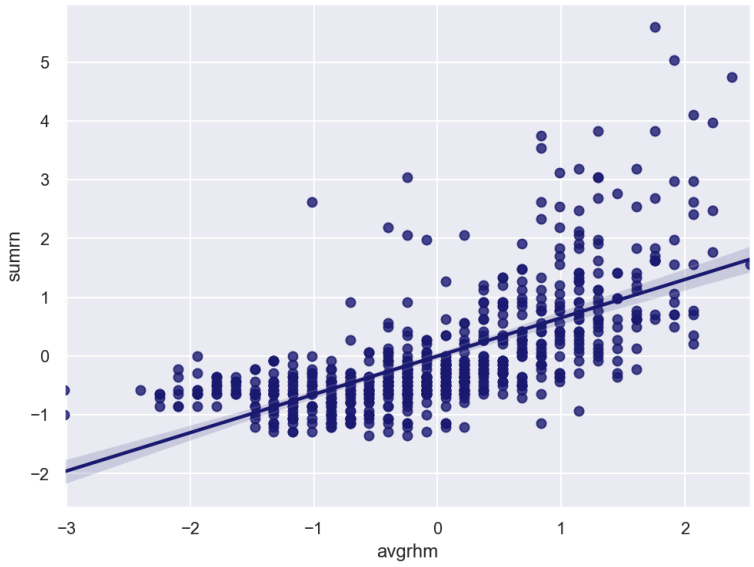


② Avgrhm & Sumrn

- Avgthm의 경우 약하지만 음의 상관관계를 보이고 있음.
- Sumrn은 다른 요소들과 비교하여 회귀선이 직선으로 나오는 것으로 보아, 전력량 증감에 영향을 미치지 않는 것으로 보임.
- **sns.regplot(x='pct',y='avgrhm',data=df,color='red')**
- **sns.regplot(x='pct',y='sumrn',data=df,color='midnightblue')**

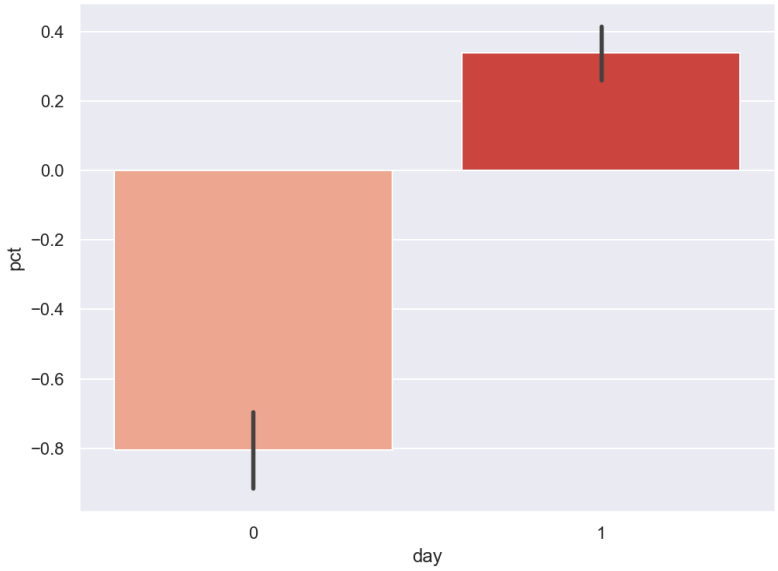


- Avgrhm과 sumrn은 완만한 양의 상관관계를 보임. 이는 강수량이 습도에 어느정도 영향을 미치는 것으로 판단함.
- **sns.regplot(x='avgrhm',y='sumrn',data=df,color='midnightblue')**

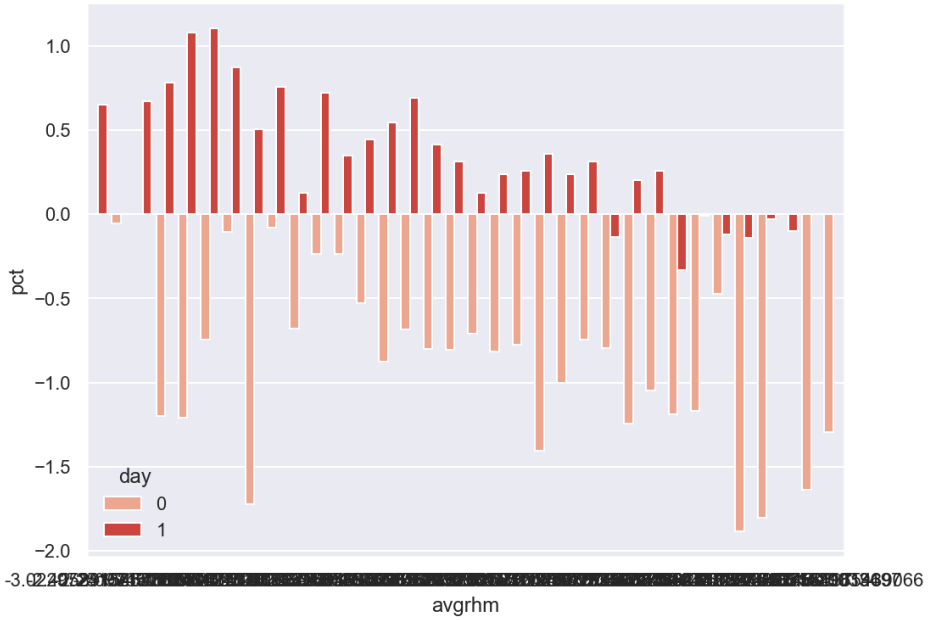


③ Day

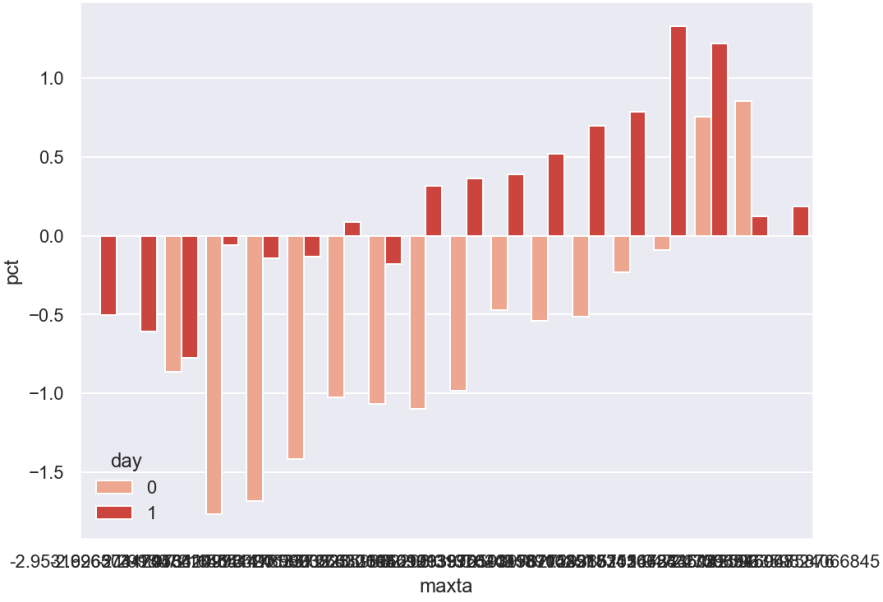
- 주중: 1 / 주말 및 공휴일: 0
- 주중 전기 사용량이 증가하고, 주말 및 공휴일은 사용량이 감소함.



- 단, 주중 상관없이 습도가 높아지면 전력 사용량이 감소함.
- **sns.barplot(x='avgrhm',y='pct',hue='day',data=df,palette='Reds',ci=None)**

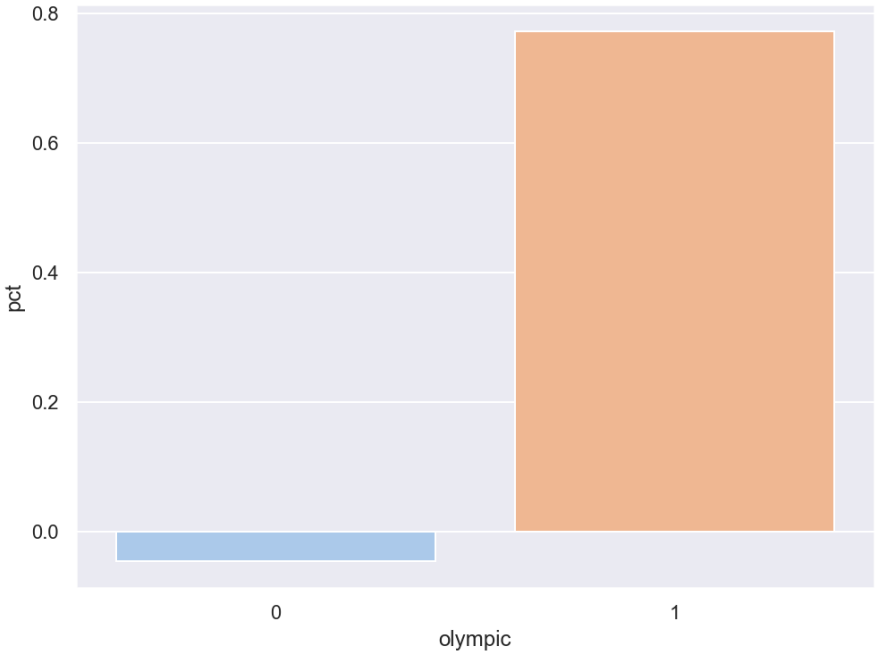


- 최고기온이 높아질수록 주중 관계 없이 전력 사용량이 증가하게 됨.
- **sns.barplot(x='maxta',y='pct',hue='day',palette='Reds',data=df,ci=None)**



④ Olympic

- 올림픽의 경우, 682일(10~20년 7~8월) 중 개최일이 약 38일 정도로 약 5%에 해당함.
- 5%의 비중인 것에 비해 0.18769라는 상관계수는 의미가 있다고 판단되며, 실로 올림픽이 개최할 시기에 전력량이 증가하는 것을 아래 baplot을 통해 볼 수 있음.
- **sns.barplot(x='olympic',y='pct',palette='pastel',data=df,ci=None)**



4. 데이터 예측 모델

예측 모델은 연속형 데이터로, 회귀 분석 모델을 사용함. 종속변수 pct를 기준으로 측정하였으며, 예측 정확도는 약 56%로 판정됨. 이는 전력 사용량에 더 많은 요인이 있는 것으로 판단되며, 이에 따라 독립변수 보완이 필요할 것으로 보임.

```
import numpy as np
X = sample2.drop(['pct'],axis=1)
y = sample2['pct']

# train data 와 test data로 구분(7:3 비율)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,                #독립 변수
                                                    y,                #종속 변수
                                                    test_size=0.3,    #검증 30%
                                                    random_state=11) #랜덤 추출 값

# 단순회귀분석 모형 생성 및 평가
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

lr = LinearRegression()
lr.fit(X_train,y_train)
y_preds = lr.predict(X_test)
mse = mean_squared_error(y_test,y_preds)
rmse = np.sqrt(mse)
r2 = r2_score(y_test,y_preds)
print('MSE:{0:.4f}, RMSE:{1:.4f}, R2:{2:.4f}'.format(mse,rmse,r2))
```

```
MSE:0.4958, RMSE:0.7041, R2:0.5601
```