

# 기상 현상에 따른 지역별 태양광 발전 효율

1 조 : 김영원 , 장용원 , 변형근

# 목차

---

**1** [ 주제 선정 근거 ]

**2** [ 데이터 수집 및 분석 ]

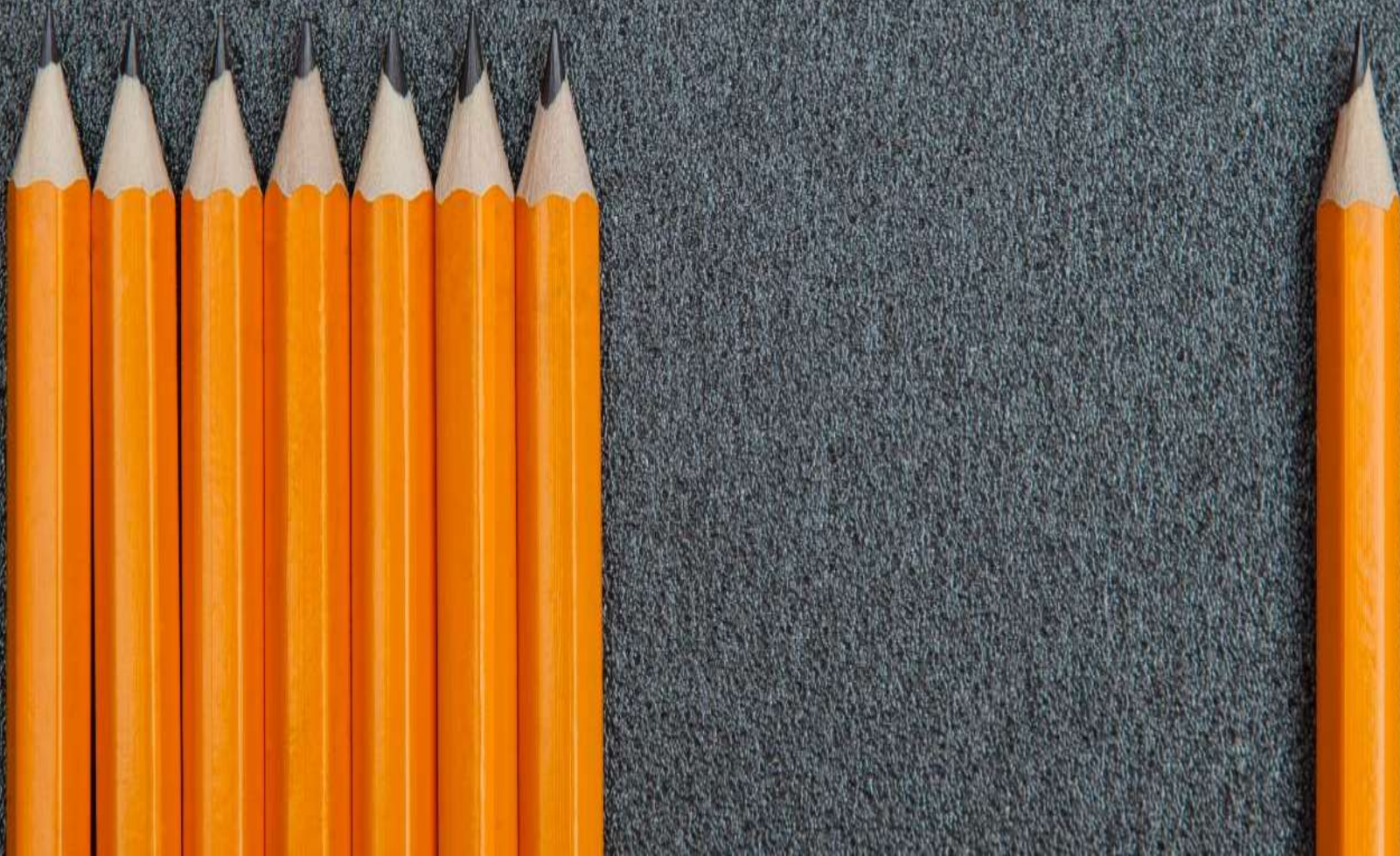
**3** [ 결론 ]





Part 1,

주제 선정 근거





# 주제 선정 근거

## 1 인플레이 감축법



인플레이션 감축법은 내년부터 10년간 에너지 안보와 기후변화 대응, 의료비 지원 등을 위해 4300억달러(약 560조원)를 투입하는 것이 골자다. 이 중에서 3690억달러(480조원)는 친환경 에너지 공급망을 강화하는 데 쓰인다.

## 2 태양광 발전 국내 동향

우리나라는 지난해 2030 국가온실가스 감축 목표(NDC)에 국제감축 수단을 활용해 3350만톤CO<sub>2</sub>eq(전체 감축목표의 11.5%)의 온실가스를 감축한다는 목표를 제시했다. 이와 관련, 심의회는 국제감축사업 통합지원 플랫폼 구축 및 운영계획을 보고했다.

충북도는 '기후위기 대응을 위한 탄소중립·녹색성장 기본조례안'을 입법예고하고 다음 달 2일까지 도민 의견을 수렴한다고 24일 밝혔다. 조례안은 온실가스 감축 목표와 계획, 충북 2050 탄소중립녹색성장위원회 설치·운영, 온실가스 감축과 적응 시책, 탄소중립 지역사회 이행과 확산 등에 관한 사항을 담았다.

## 3 태양광 발전 관련 정책

[ 2050년까지 건물·교통·도시 등서 탄소중립 실현 ]

[ 탄소중립 실현을 위한 '제로에너지 특화도시' 조성 추진 ]

[ 도심내 건물 외벽까지...태양광 발전량 정확한 예측 기술 개발 ]

[ 환경부, 소속·산하기관에 건물일체형 태양광 도입한다 ]

5개 대학 '탄소중립 그린캠퍼스' 조성...3년간 18억 지원

정부, 올해 신재생에너지 설비 설치에 3192억원 지원

# 사용 기술 및 데이터

## [ 사용 기술 ]

- 크롤링 (BeautifulSoup, selenium)
- 워드 클라우드 (wordcloud)
- 네트워크 그래프 (networkx)
- 데이터 차트화 (pandas)
- 지도 시각화 (folium)

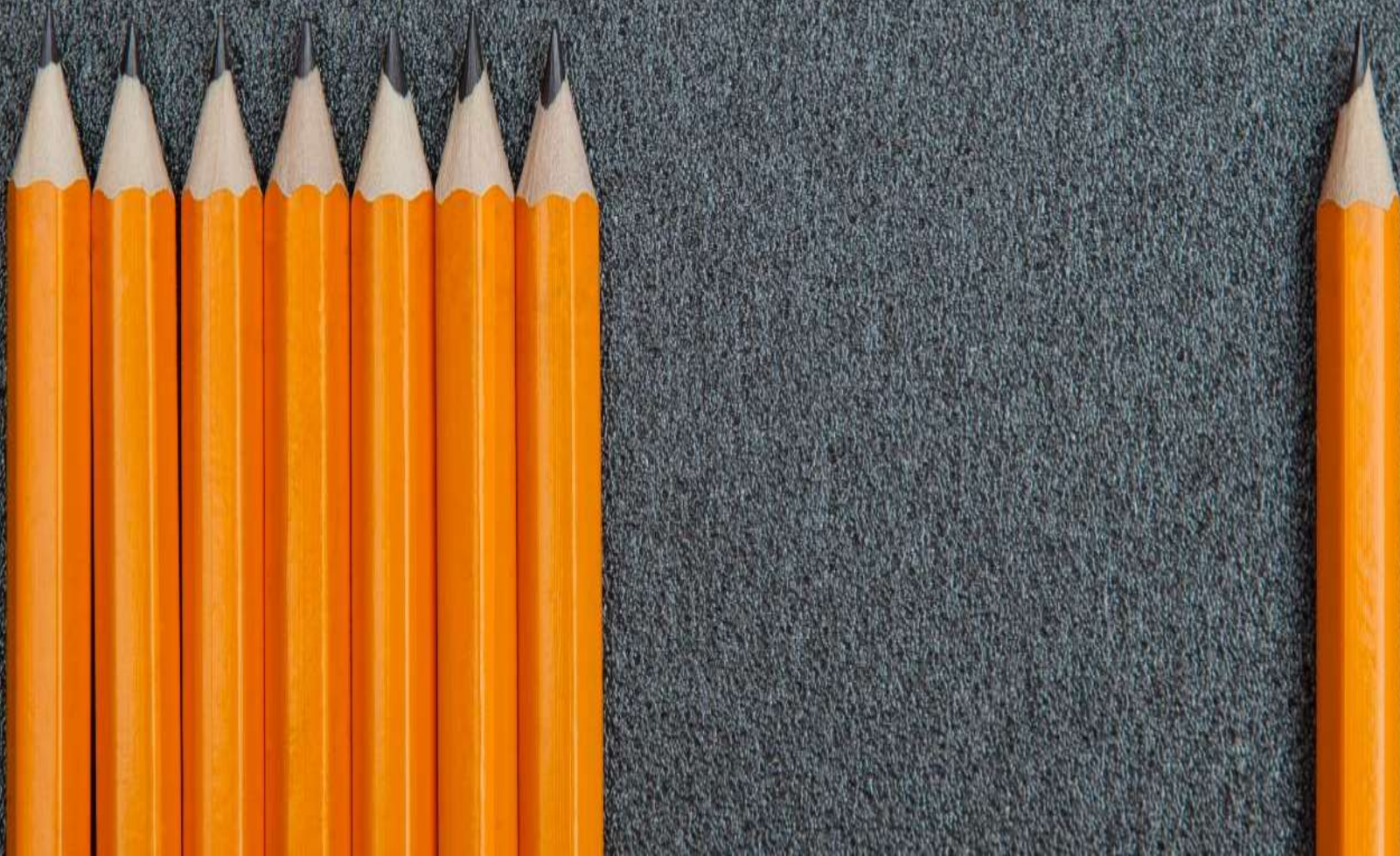
## [ 사용 데이터 ]

- 기상청\_지상(종관, ASOS) 기상 관측자료
- 한국전력거래소\_지역별 시간별 태양광 발전량
- 빅카인즈 키워드 분석
- 정책브리핑 기사 본문



Part 2,

데이터 수집 및 분석





## 키워드 분석

수집대상: 빅카인즈

수집기간: 2022. 08. 22 ~ 2022. 08. 23

수집주제: 태양광

키워드 수집량: 993,476개



빅카인즈 워드클라우드



## 빅카인즈 네트워크 그래프

# 키워드 분석

수집대상: 정책브리핑

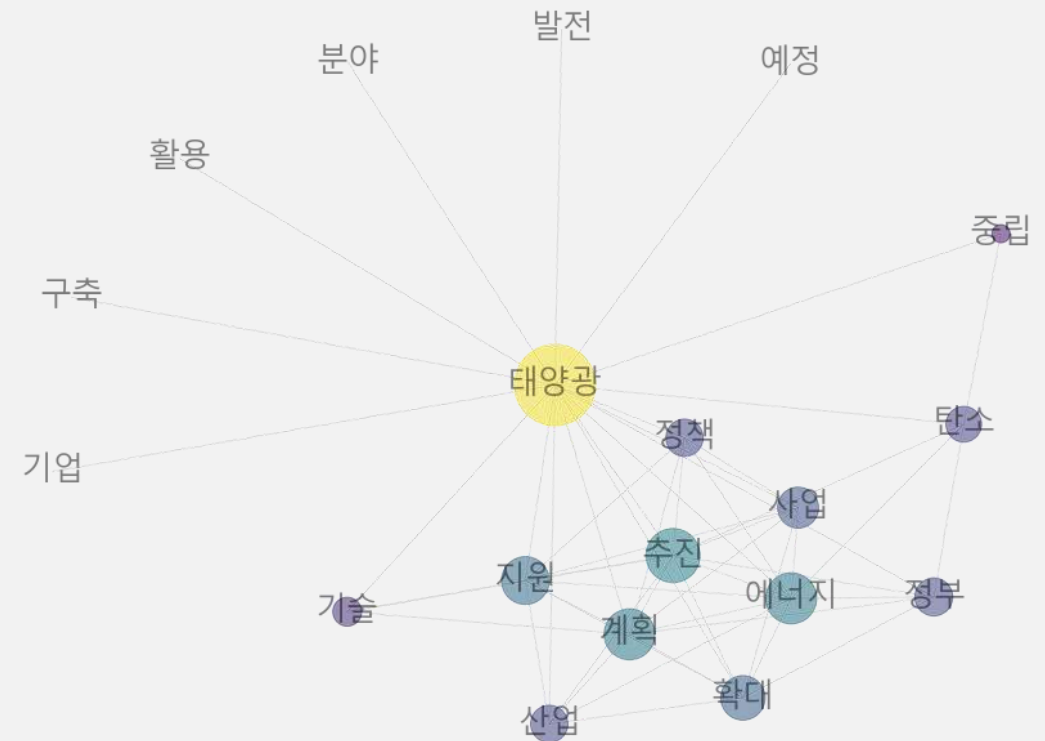
수집기간: 2022. 08. 22 ~ 2022. 08. 23

수집주제: 태양광

키워드 수집량: 82,397개



정책브리핑 워드클라우드



정책브리핑 네트워크 그래프



# 키워드 분석

수집대상: 해외 논문

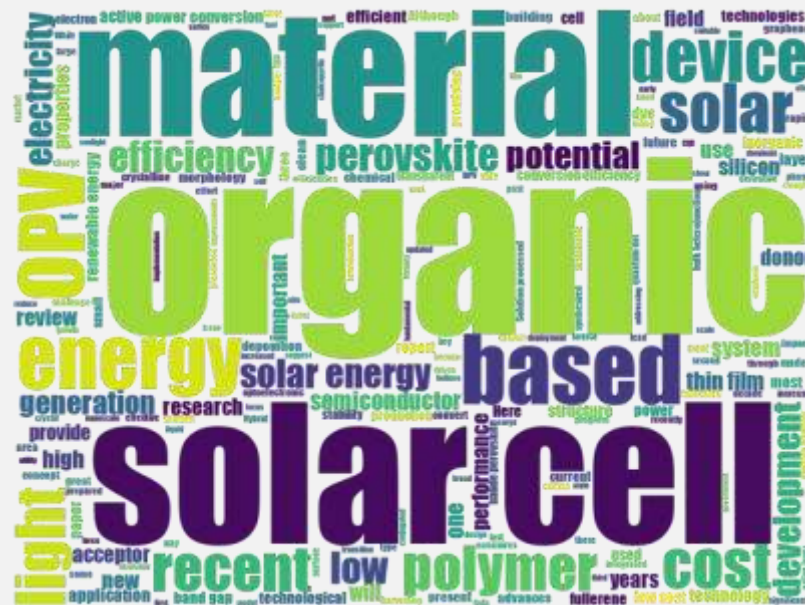
수집 키워드: photovoltaics



논문 제목 워드클라우드

수집 사이트: 구글 스콜라

논문 수집량: 200개



논문 본문 워드클라우드

# 지도 시각화

```
# 시 도 별 지도 그리기 코드
csvFile = 'Si_Do_Ele_students_Teacher_ratio.csv'
SiDodf = pd.read_csv(csvFile, encoding = 'euc-kr')

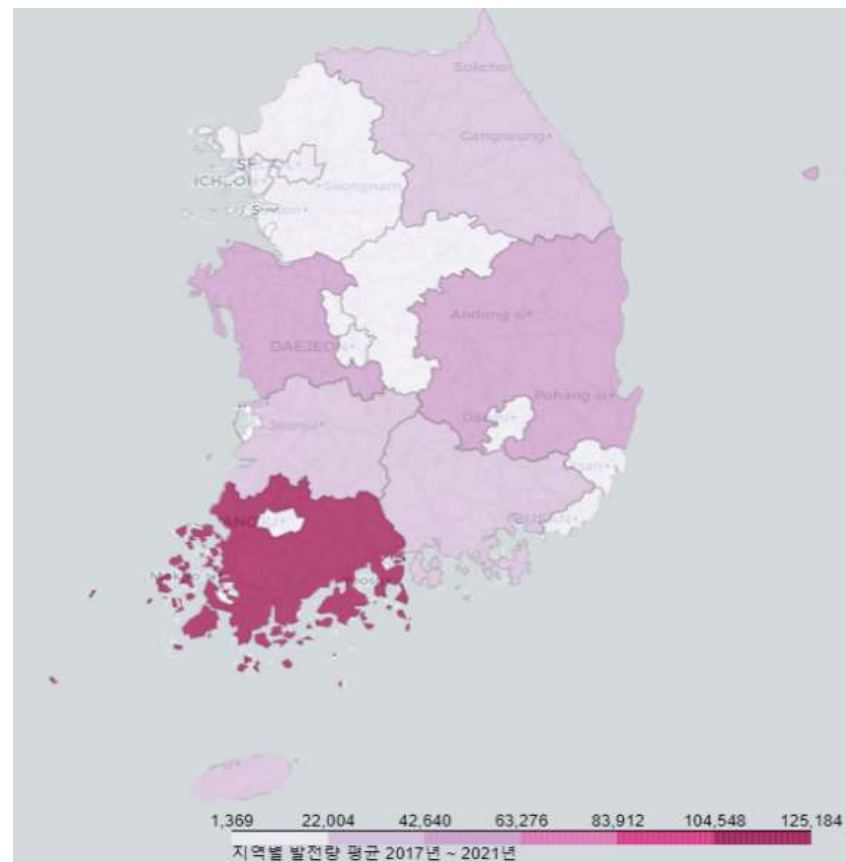
SiDo_geo = 'Si_Do_map_utf8.json'

# Initialize the map at Daejeon
m = folium.Map(location=[36.45, 127.42], #
                tiles="cartodbpositron", zoom_start=7)

# Json map polygon 경계좌표값으로 구역별 색칠하기
folium.Choropleth(
    geo_data = SiDo_geo,
    name='지역별 발전량 총합 2017년 ~ 2021년',
    data=sol_mean,
    columns=['지역', '발전량'],
    key_on='feature.properties.CTP_KOR_NM',
    fill_color='PuRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='지역별 발전량 총합 2017년 ~ 2021년').add_to(m)

# 색칠한 layer 켜고 끄기 버튼 추가
folium.LayerControl().add_to(m)
m
```

## 지역별 발전량 평균





# 데이터 전처리

# Null값 시각화

```
img = sns.heatmap(weather_power.isna(), cbar=False)
os.makedirs('./이미지자료', exist_ok=True)
plt.savefig('./이미지자료/태양광_null.png')
plt.show()
```

# Null값 1000개 이상 목록

# Null값이 너무 많아 data 제거하기로 함

```
nulllist = weather_power.isna().sum()[weather_power.isna().sum() > 1000]
plt.bar(nulllist.sort_values(ascending=False).index, nulllist.sort_values(ascending=False))
plt.xticks(rotation=90)
plt.savefig('./이미지자료/Null값_barplot.png')
plt.show()
```

# 삭제 리스트 제작

```
delidx = weather_power.isna().sum()[weather_power.isna().sum() > 1000].index
```

# 미삭제 목록 제작

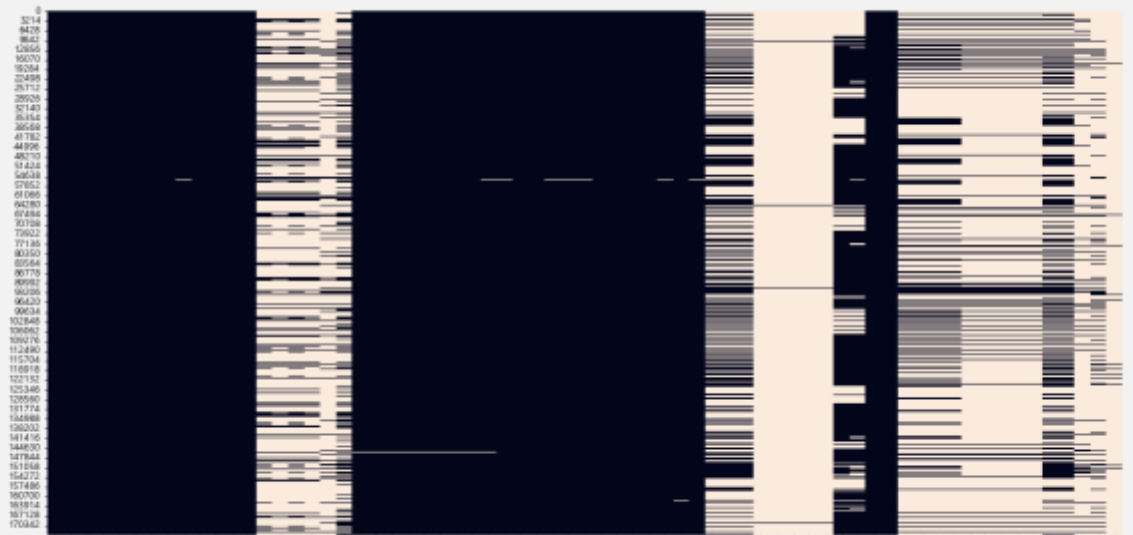
```
colnames = [i for i in weather_power.columns if i not in delidx]
```

# 일강수량 Null 은 비가 안온 날로 간주 -> fillna(0) 적용 할 예정

```
colnames.append('일강수량')
```

# 삭제 후 DataFrame 재직

```
weather_power_refined = weather_power[colnames]
```



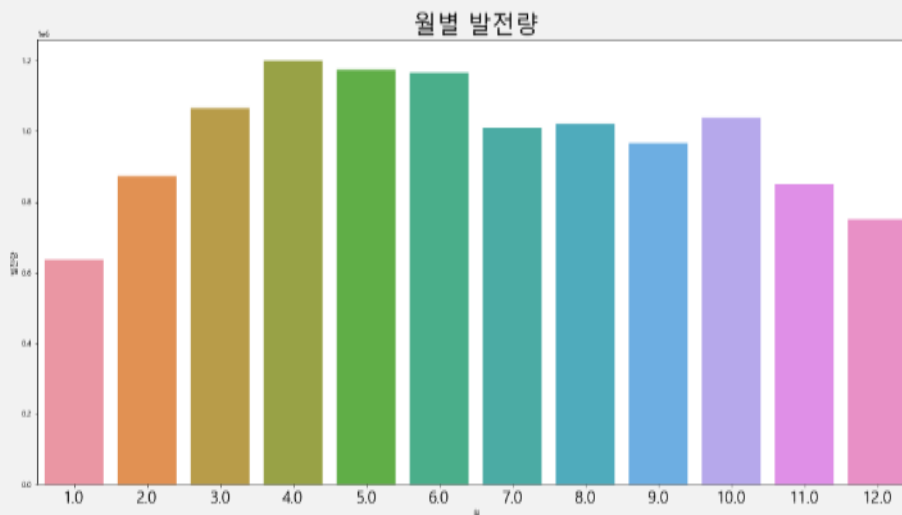
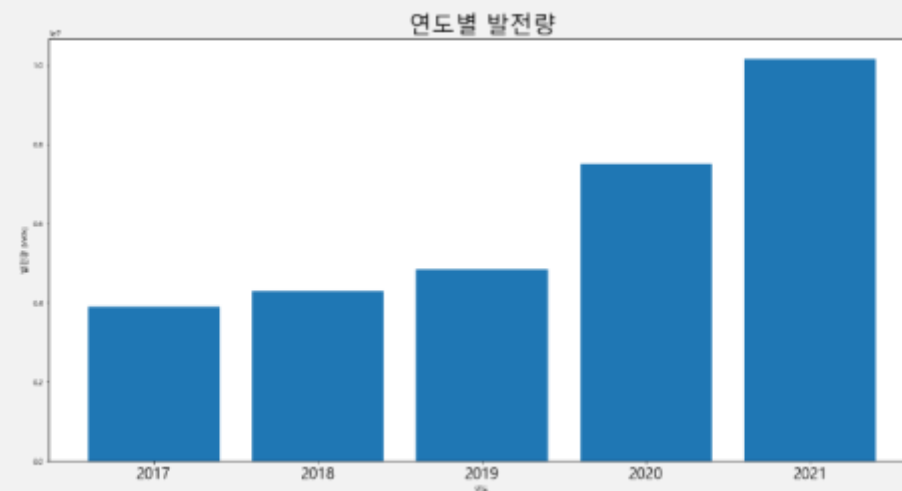
NULL값 시각화

# 데이터 차트화

```
plt.bar(x='년', height='발전량', data=weather_power_groupby)
plt.title('연도별 발전량', fontdict={'fontsize' : 30})
plt.xticks(fontsize=20)
plt.xlabel('연도')
plt.ylabel('발전량 (kWh)')
plt.savefig('./이미지자료/연도별_발전량.png')
plt.show()
```

#월별 발전량

```
plt.title('월별 발전량', fontdict={'fontsize' : 30})
sns.barplot(x='월' , y='발전량', data=weather_power_groupby, ci=None)
plt.savefig('./이미지자료/월별_발전량.png')
plt.show()
```

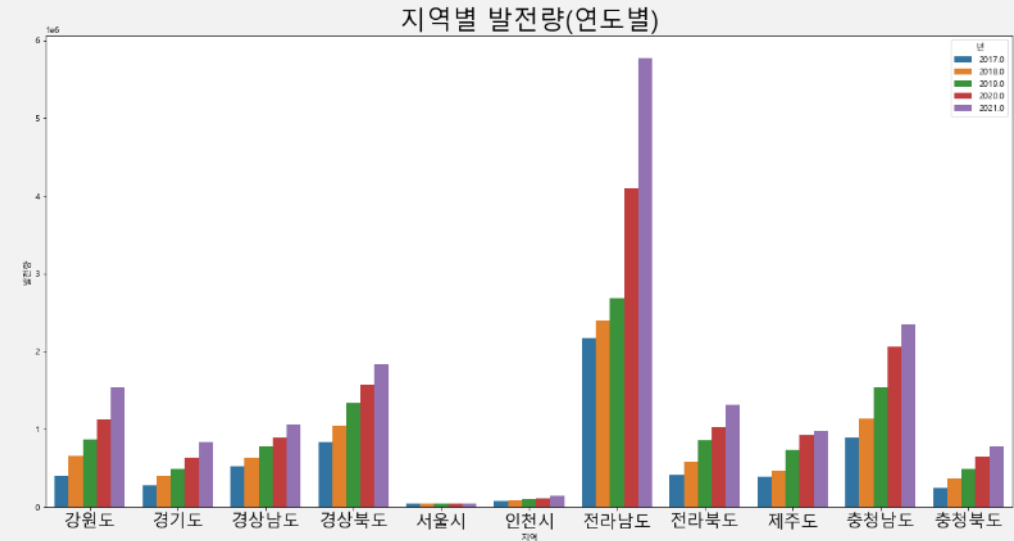




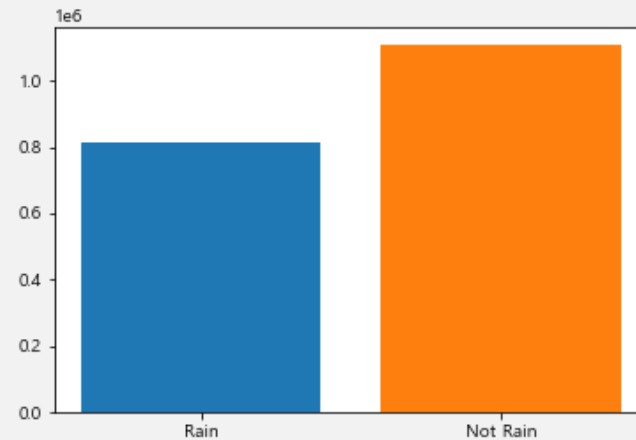
# 데이터 차트화

#지역별 발전량 연도별 구분

```
plt.title('지역별 발전량(연도별)', fontdict={'fontsize' : 30})
sns.barplot(x='지역', y='발전량', data=weather_power_groupby, hue='년', ci=None)
plt.savefig('./이미지자료/지역별_발전량(연도별).png')
plt.show()
```



```
plt.bar('Rain', df_rain['발전량'].sum() / len(df_rain))
plt.bar('Not Rain', df_not_rain['발전량'].sum() / len(df_not_rain))
plt.savefig('강수에 따른 일 평균 발전량 효율.png')
plt.show()
```



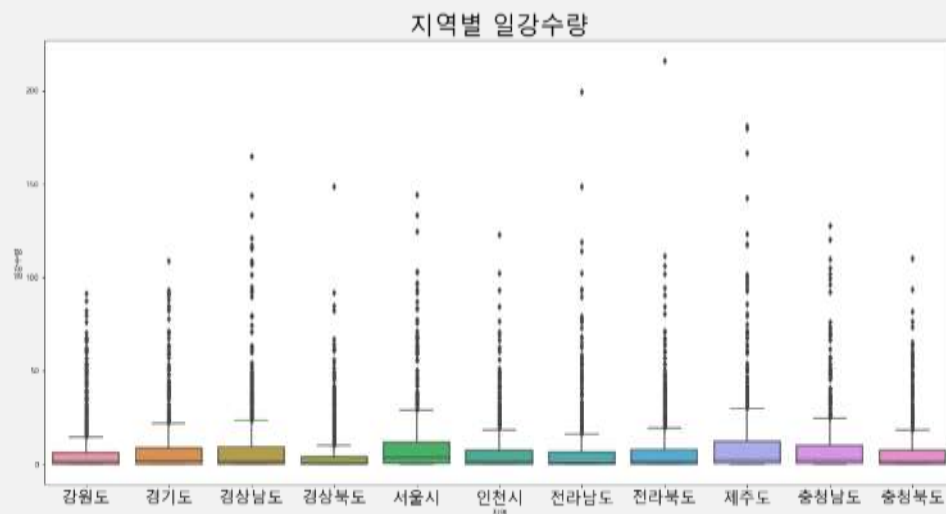
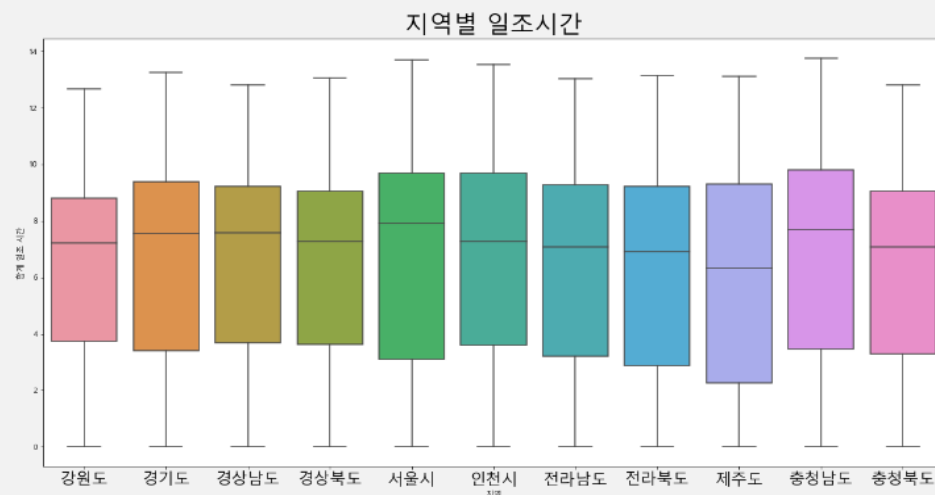
# 데이터 차트화

일조 시간 : 태양의 직사광이 지표면에 비친 시간

```
plt.title('지역별 일조시간', fontdict={'fontsize' : 30})
sns.boxplot(x='지역', y='합계 일조 시간', data=weather_power_groupby)

plt.savefig('./이미지자료/지역별_일조시간.png')
plt.show()
```

```
plt.title('지역별 일강수량', fontdict={'fontsize' : 30})
sns.boxplot(x='지역', y='일강수량', data=w_p_rain)
plt.savefig('./이미지자료/지역별_일강수량.png')
```

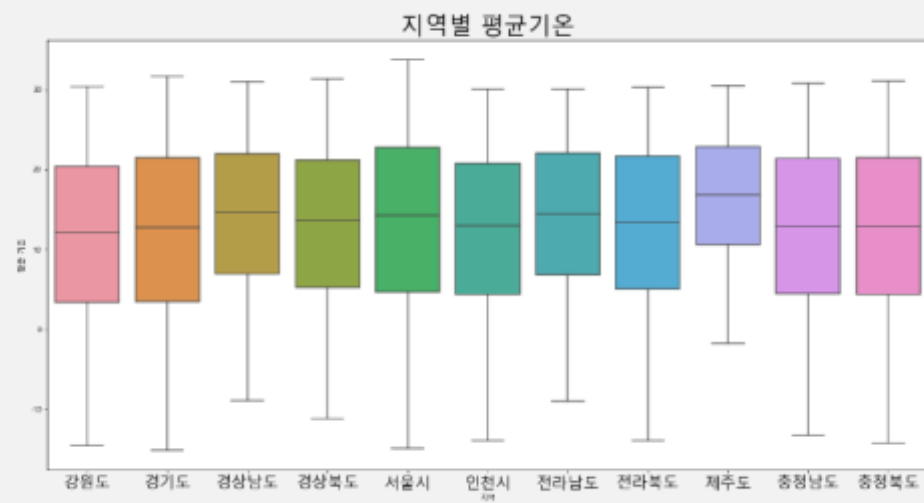
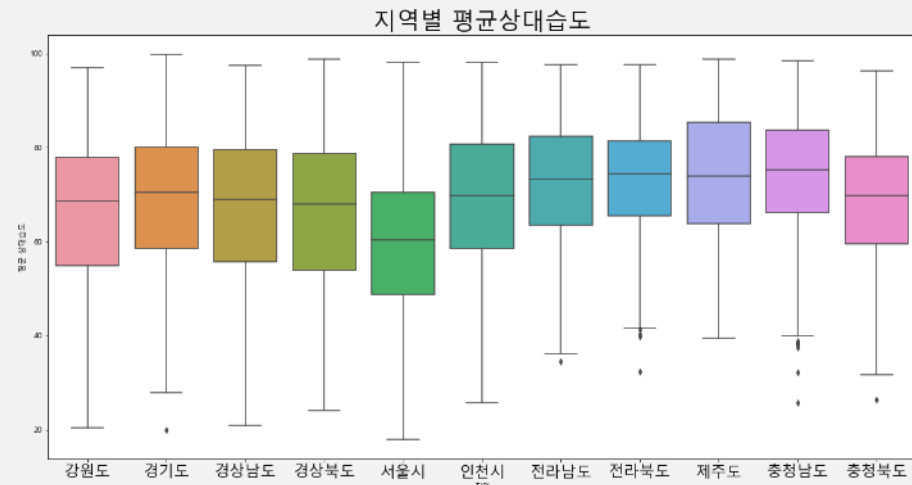




# 데이터 차트화

```
plt.title('지역별 평균상대습도', fontdict={'fontsize' : 30})
sns.boxplot(x='지역', y='평균 상대습도', data=weather_power_groupby)
plt.savefig('./이미지자료/지역별_상대습도.png')
```

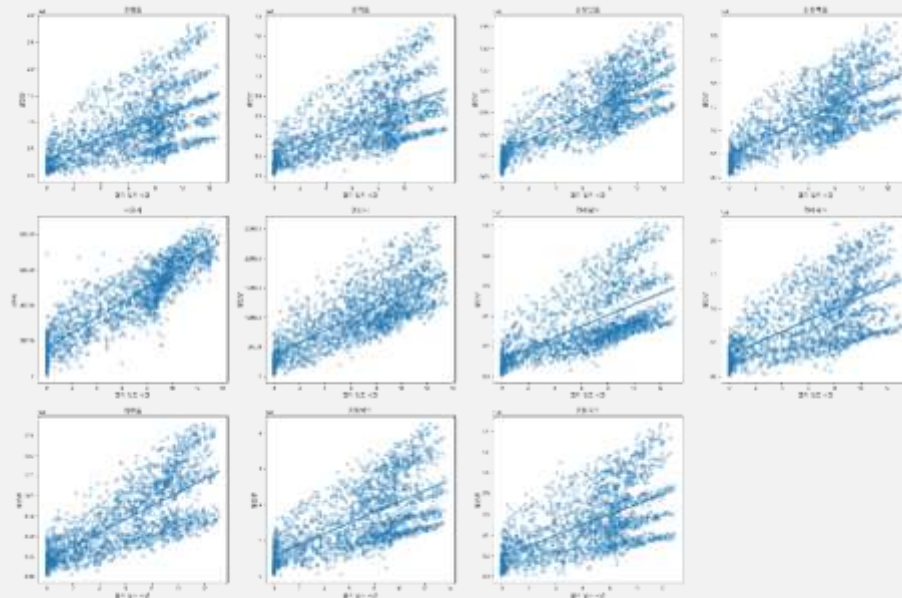
```
plt.title('지역별 평균기온', fontdict={'fontsize' : 30})
sns.boxplot(x='지역', y='평균 기온', data=weather_power_groupby)
plt.savefig('./이미지자료/지역별_평균기온.png')
```



# 데이터 차트화

```
plt.figure(figsize=(30,20))

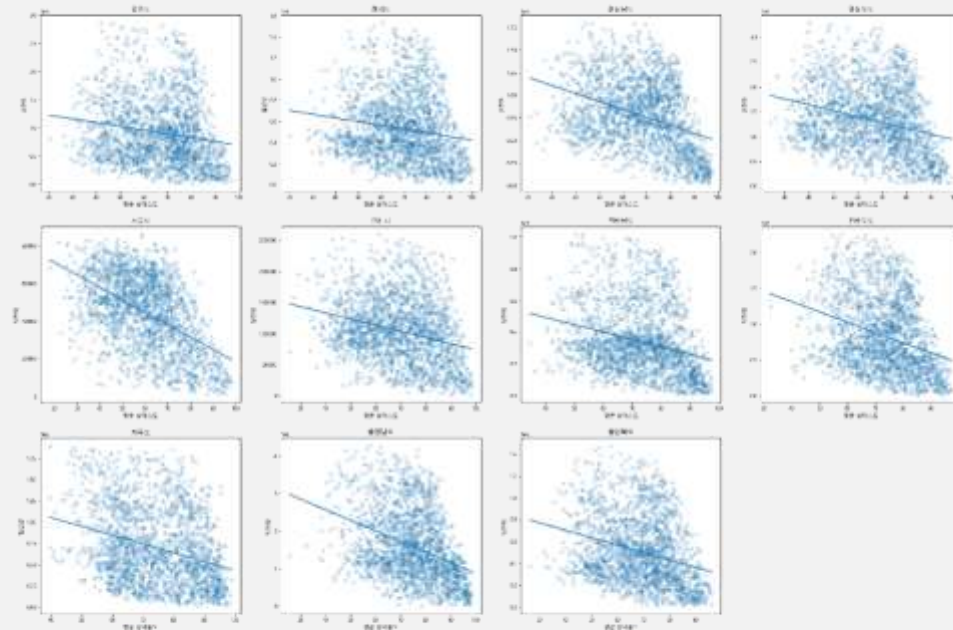
for idx,i in enumerate(weather_power_groupby['지역'].unique()):
    plt.subplot(3,4,idx+1)
    plt.title(f'{i}')
    sns.regplot(x='합계 일조 시간',y='발전량',data=weather_power_groupby[weather_power_groupby['지역']== i],scatter_kws={'alpha':0.3})
plt.savefig('./이미지자료/지역별_일조시간별_발전량_회귀선.png')
plt.show()
```



# 데이터 차트화

```
plt.figure(figsize=(30,20))

for idx, i in enumerate(weather_power_groupby['지역'].unique()):
    plt.subplot(3,4,idx+1)
    plt.title(f'{i}')
    sns.regplot(x='평균 상대습도',y='발전량',data=weather_power_groupby[weather_power_groupby['지역']== i],ci=None,scatter_kws={'alpha':0.2})
plt.savefig('./이미지자료/지역별_상대습도_발전량_회귀선.png')
plt.show()
```



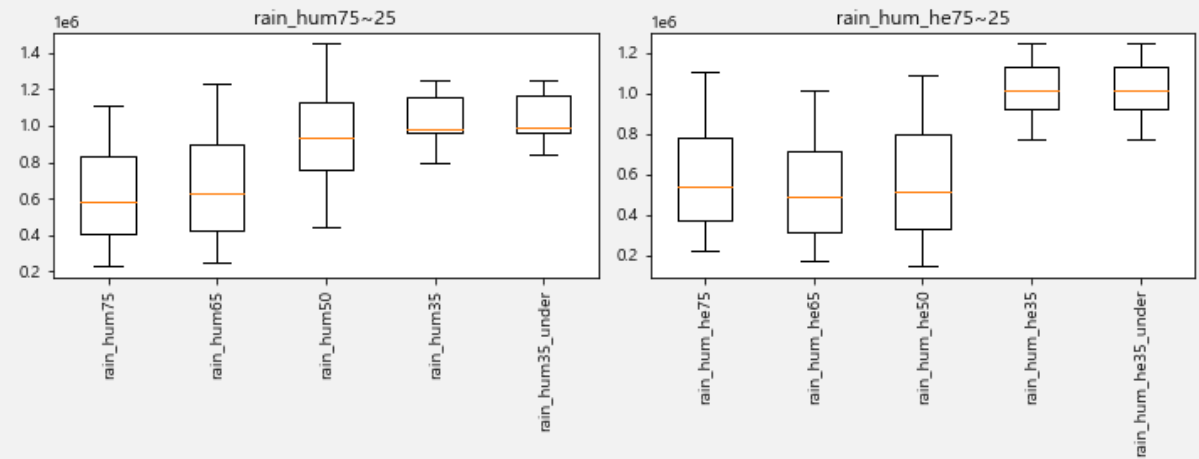


# 데이터 차트화

```
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
plt.boxplot([rain_hum75, rain_hum65, rain_hum50, rain_hum35, rain_hum35_under])
plt.title('rain_hum75~25')
plt.xticks([1, 2, 3, 4, 5], ['rain_hum75', 'rain_hum65', 'rain_hum50', 'rain_hum35', 'rain_hum35_under'])
plt.xticks(rotation=90)

plt.subplot(1,2,2)
plt.boxplot([rain_hum_he75, rain_hum_he65, rain_hum_he50, rain_hum_he35, rain_hum_he35_under])
plt.title('rain_hum_he75~25')
plt.xticks([1, 2, 3, 4, 5], ['rain_hum_he75', 'rain_hum_he65', 'rain_hum_he50', 'rain_hum_he35', 'rain_hum_he35_under'])
plt.xticks(rotation=90)

plt.tight_layout()
plt.savefig('이상치 없앤 Rain_Hum_box.png')
plt.show()
```

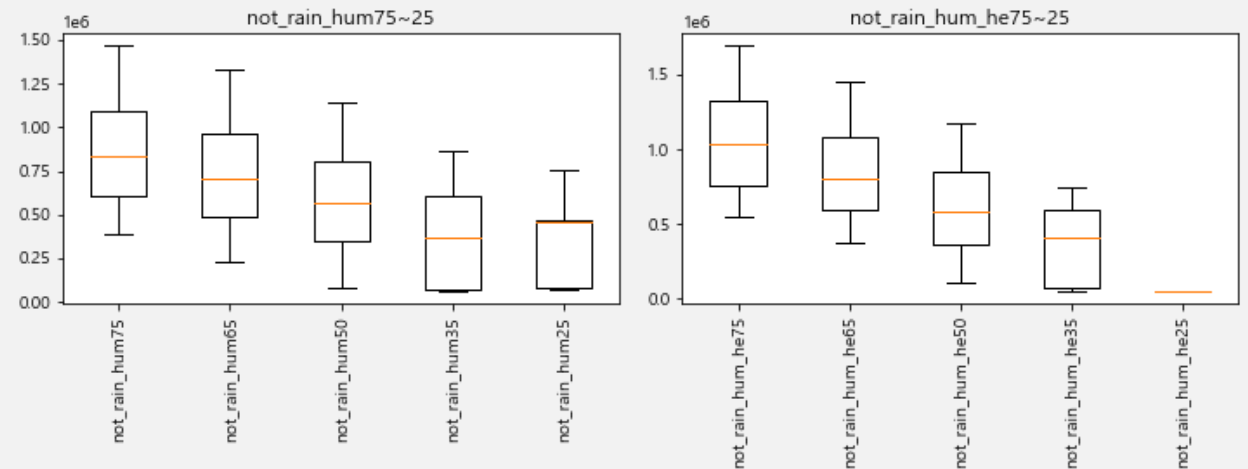


# 데이터 차트화

```
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
plt.boxplot([not_rain_hum75, not_rain_hum65, not_rain_hum50, not_rain_hum35, not_rain_hum25])
plt.title('not_rain_hum75~25')
plt.xticks([1, 2, 3, 4, 5], ['not_rain_hum75', 'not_rain_hum65', 'not_rain_hum50', 'not_rain_hum35', 'not_rain_hum25'])
plt.xticks(rotation=90)

plt.subplot(1,2,2)
plt.boxplot([not_rain_hum_he75, not_rain_hum_he65, not_rain_hum_he50, not_rain_hum_he35, not_rain_hum_he25])
plt.title('not_rain_hum_he75~25')
plt.xticks([1, 2, 3, 4, 5], ['not_rain_hum_he75', 'not_rain_hum_he65', 'not_rain_hum_he50', 'not_rain_hum_he35', 'not_rain_hum_he25'])
plt.xticks(rotation=90)

plt.tight_layout()
plt.savefig('이상치 없앤 Not_Rain_Hum_box.png')
plt.show()
```





Part 3,

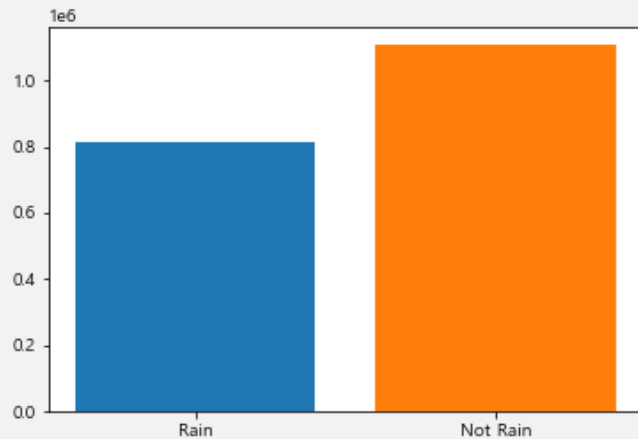
결론



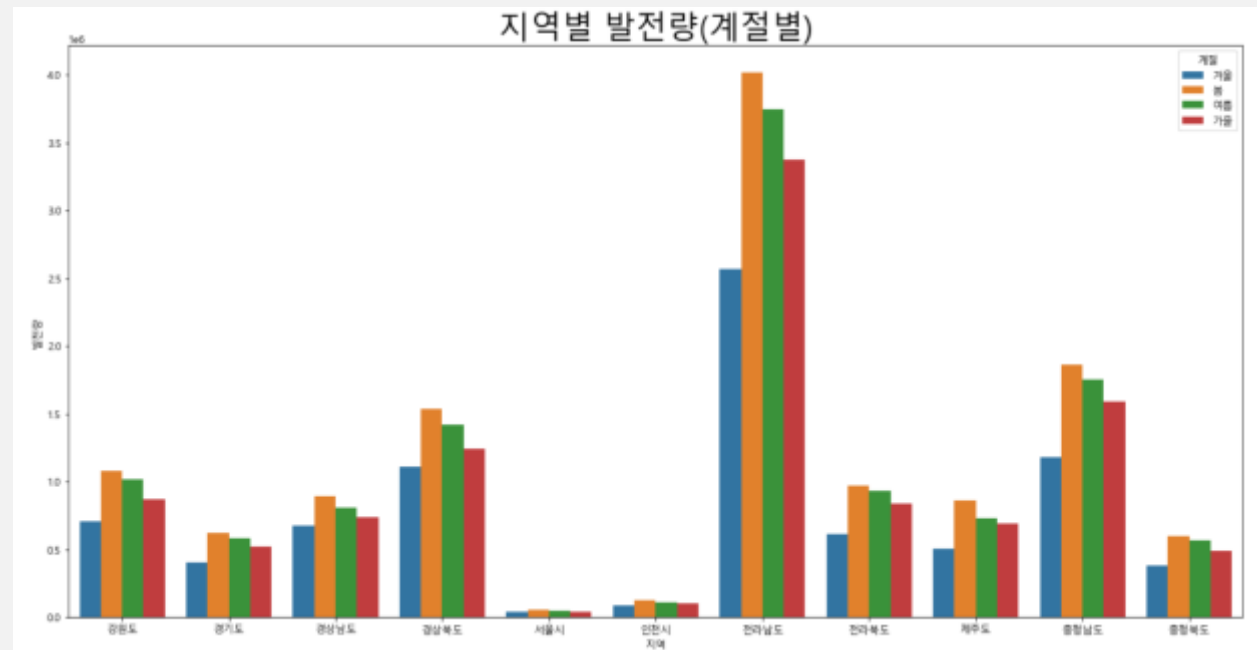


# 결론

비가오면 효율이 떨어진다.

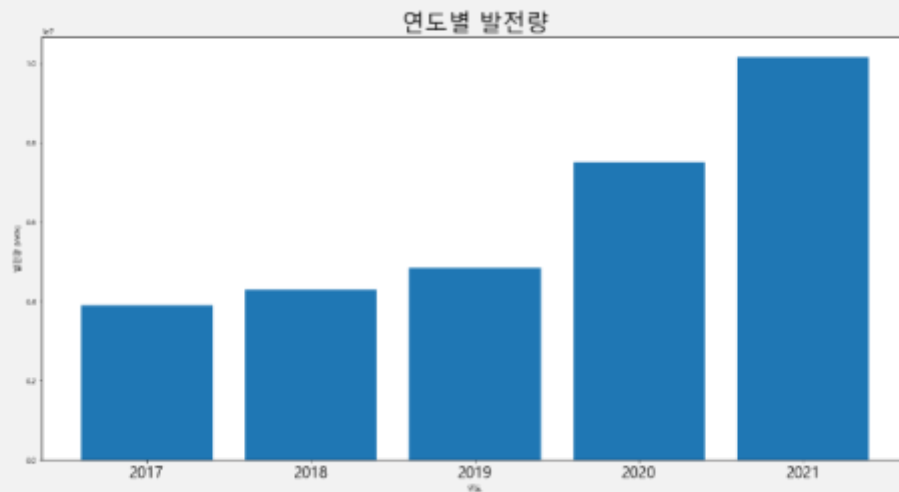


봄, 여름, 가을, 겨울순으로 발전량이 많다.

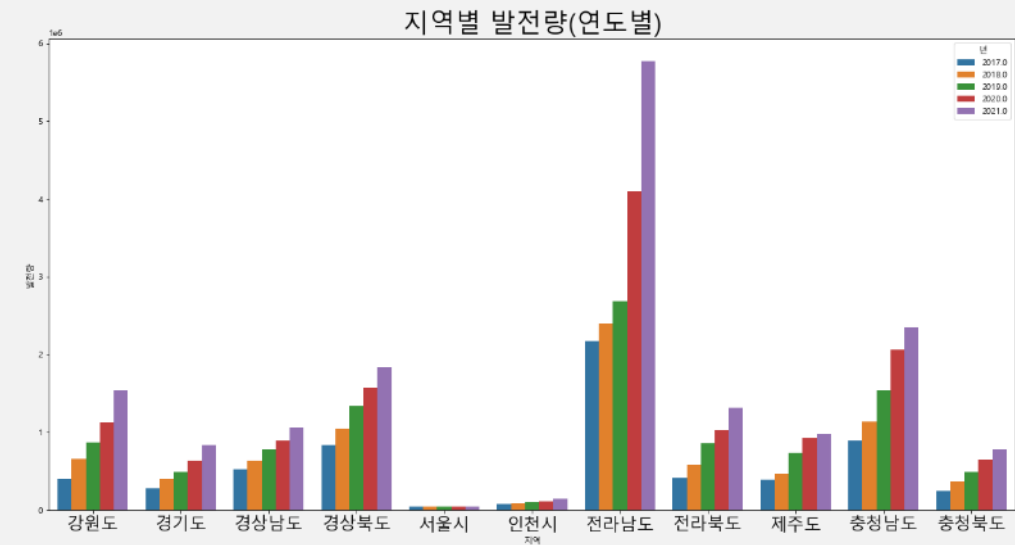


# 결론

연도가 지날수록 발전량이 많아진다.

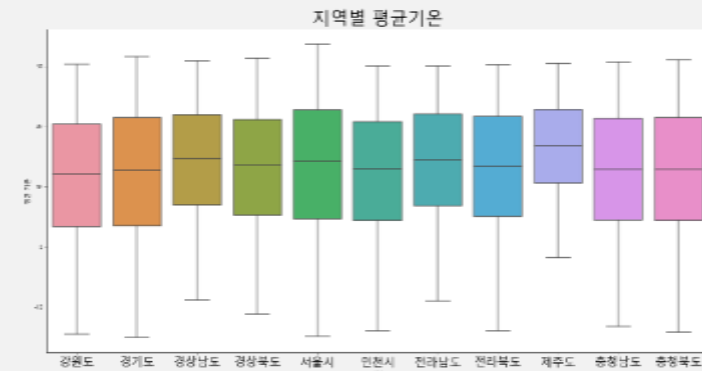
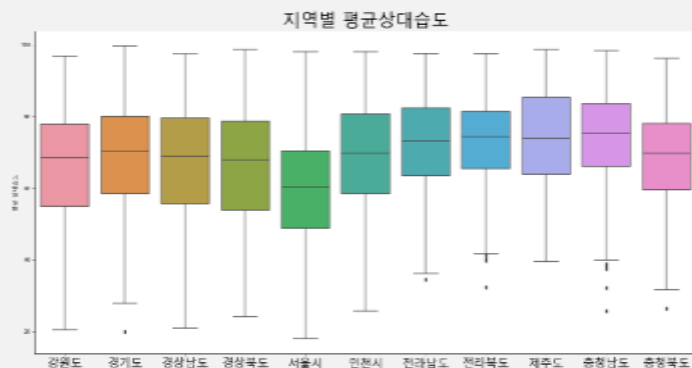
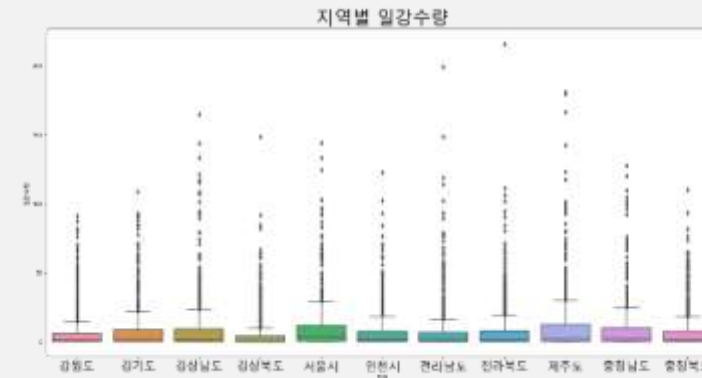
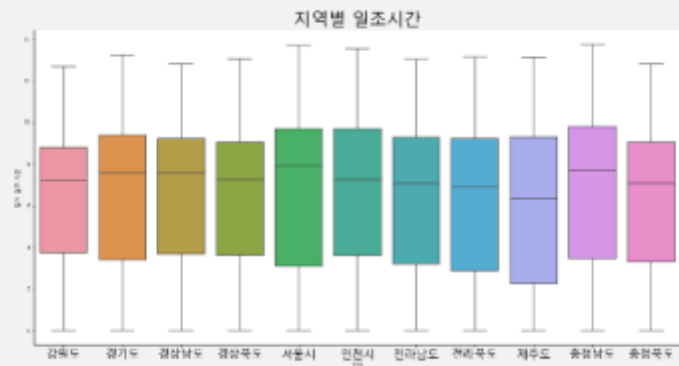


전라남도의 발전량이 독보적으로 많다.



# 결론

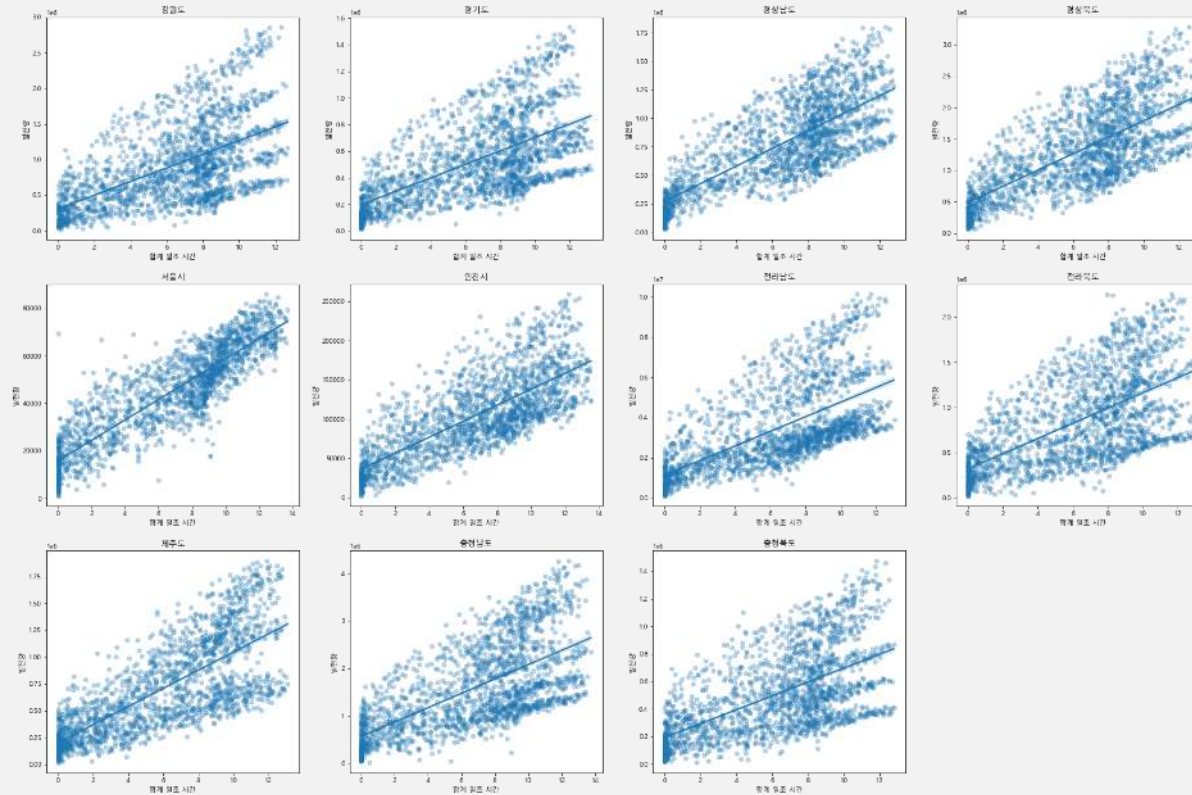
지역별로 일조시간, 일강수량, 평균상대습도, 평균기온은 큰 차이가 없다.





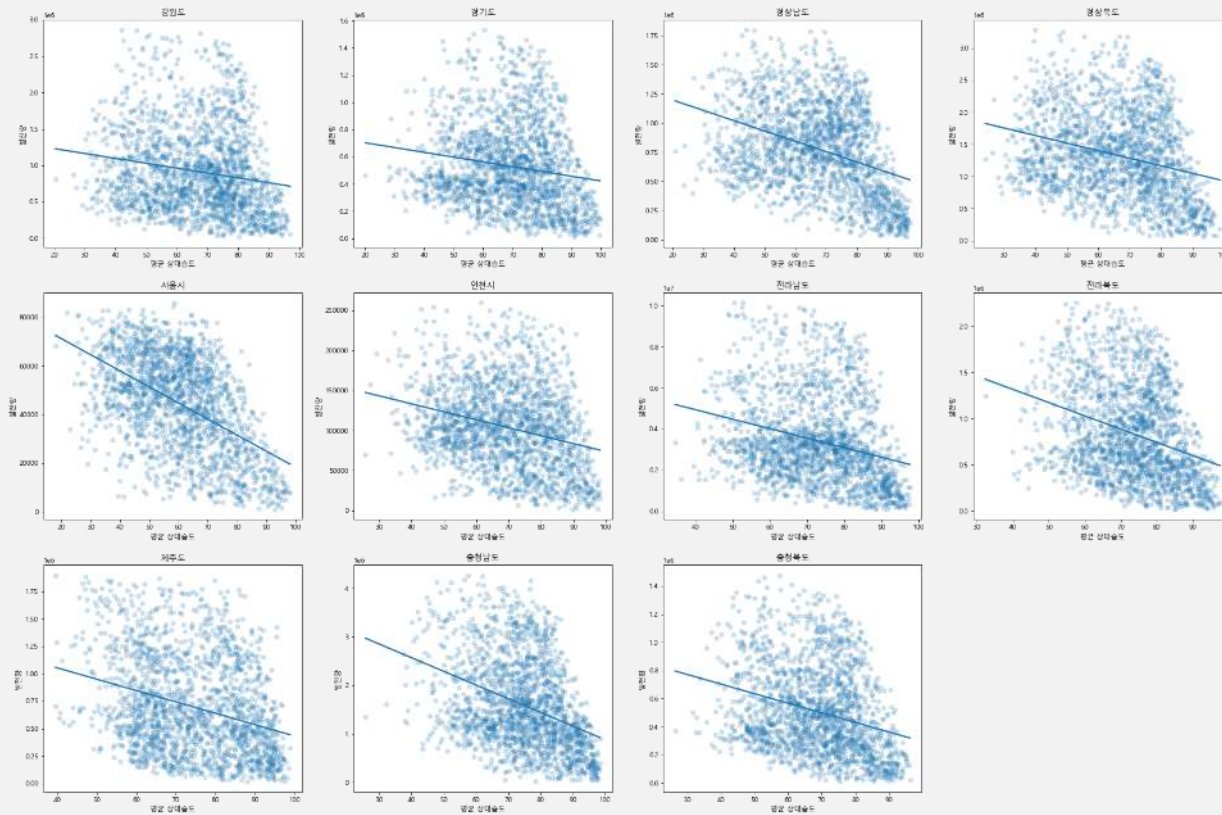
# 결론

일조시간이 많을수록 발전량이 늘어난다.



# 결론

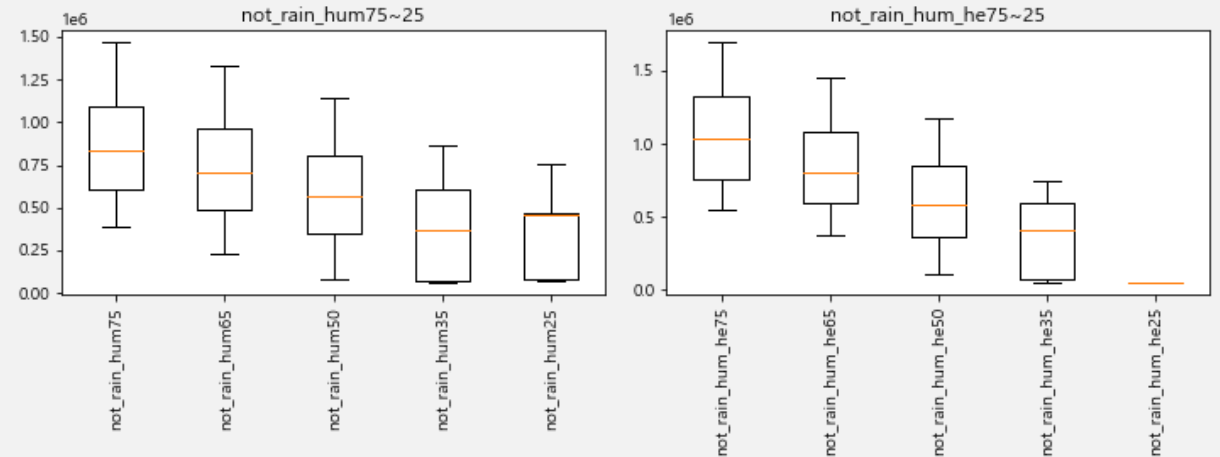
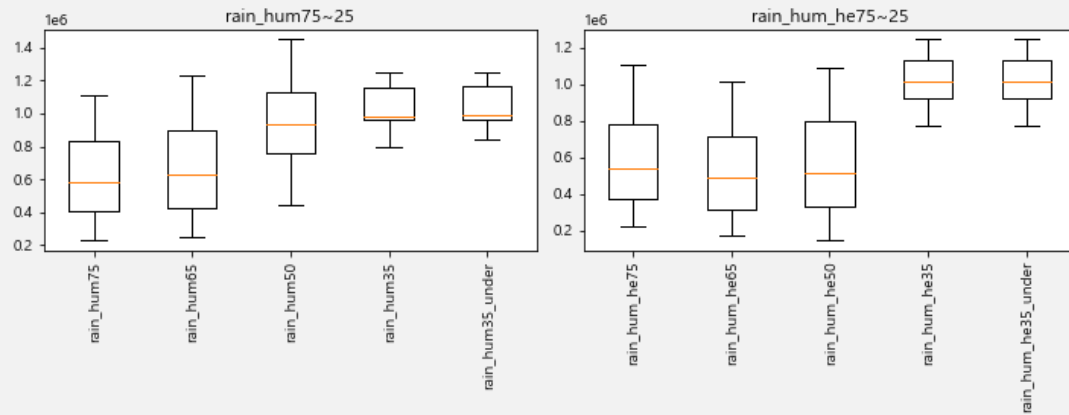
평균상대습도가 낮을수록 발전량이 늘어난다.



# 결론

비가 온 날은 평균상대습도가 낮을수록  
발전량이 늘어난다.

비가 안 온 날은 평균상대습도가 높을수록  
발전량이 늘어난다.







감사합니다