

재생에너지 분석

_태양에너지와 풍력ener지를 중심으로

Like_lion_13회 _정수빈.py

목차

01 에너지 전환 필요성 및 현황

02 태양열, 풍력 발전 분석

03 결론 및 한계

PART 1.

에너지 전환 필요성 및 현황



에너지 전환 및 에너지

001 >> 1차 에너지

: 오랜 세월 자연적으로 형성된 천연 상태의 가공되지 않은 에너지.

화석연료 : 석탄, 석유, 천연가스(LNG)

신재생에너지 : 수력, 원자력, 태양열, 태양광, 풍력, 바이오매스, 지열에너지, 등

002 >> 2차 에너지

: 1차에너지를 전환 가공해 얻을 수 있는 에너지

ex – 전력, 도시가스, 석유제품 등

003 >> 에너지전환 (Energy Transition)

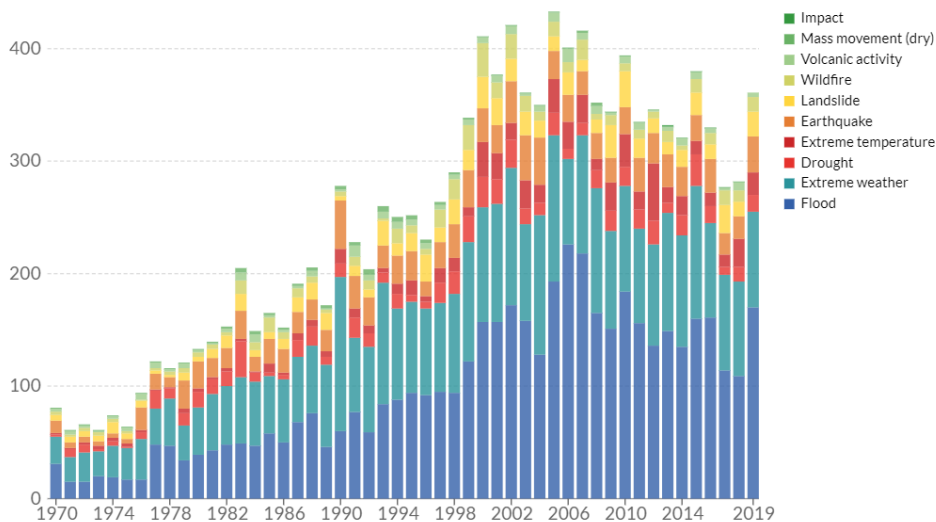
: 현 지속 불가능한 에너지 공급체계를 재생에너지 이용한 지속가능한 방법으로 바꾸는 것

: 에너지 효율 높이거나 기기 바꾸는 것 포함, 산업 및 사회 전체를 에너지의 관점에서 새롭게 재구성

Part 1 Global reported natural disasters by type, 1970 to 2019

The annual reported number of natural disasters, categorised by type. This includes both weather and non-weather related disasters.

Our World
in Data



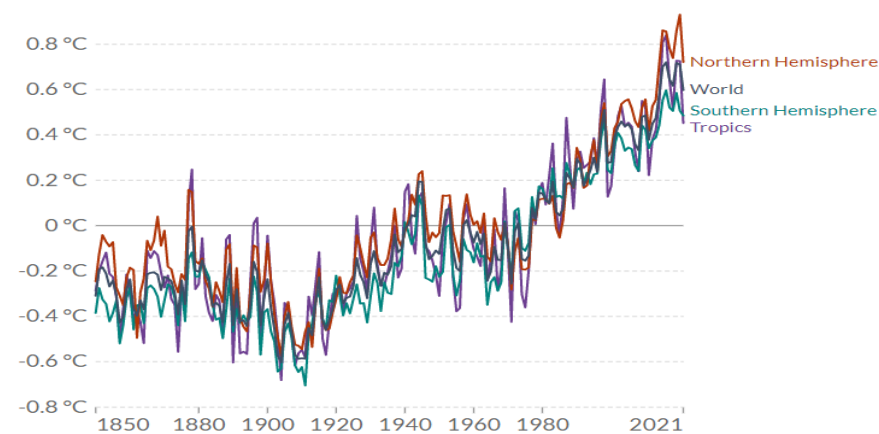
Source: EMDAT (2020): OFDA/CRED International Disaster Database, Université catholique de Louvain - Brussels - Belgium
OurWorldInData.org/natural-disasters • CC BY

Annual sea surface temperature anomaly

This is measured at a nominal depth of 20cm, and given relative to the average temperature from the period of 1961 - 1990.

Our World
in Data

+ Add region



Source: Met Office Hadley Center

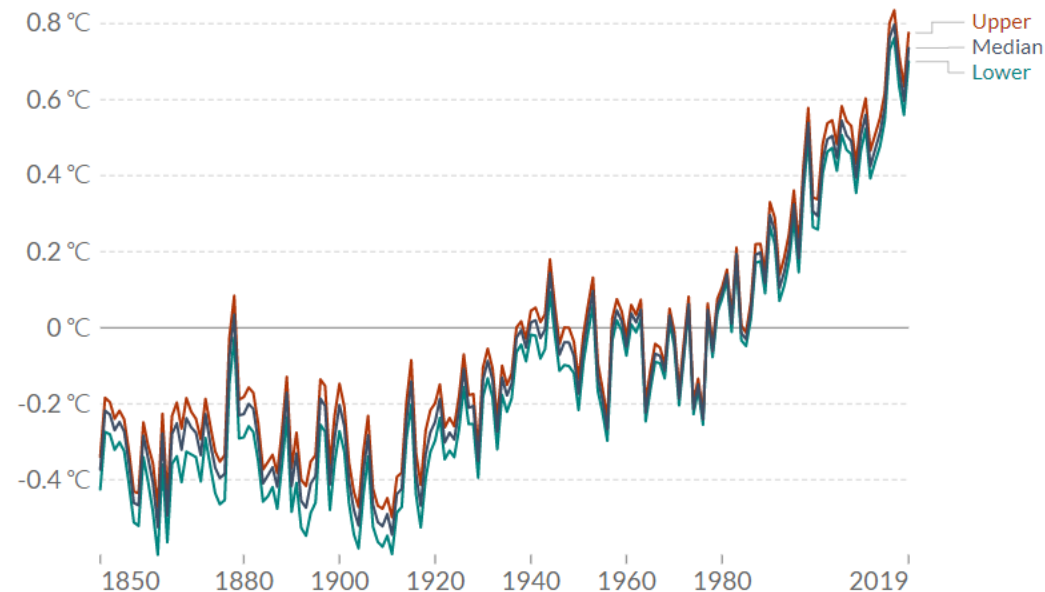
OurWorldInData.org/climate-change • CC BY

Average temperature anomaly, Global

Global average land-sea temperature anomaly relative to the 1961-1990 average temperature.

Our World
in Data

↔ Change region



Source: Hadley Centre (HadCRUT4) OurWorldInData.org/co2-and-other-greenhouse-gas-emissions • CC BY
Note: The red line represents the median average temperature change, and grey lines represent the upper and lower 95% confidence intervals.

기후변화에 대한 국제적 대응

1992 유엔기후변화협약

Annex별로 감축목표 설정
Annex1 42개국
Annex2 24개국
non-Annex

2009 post-2012체제 좌초

선진국과개도국의의견차

2015 파리협정

선진국에만 온실가스 감축 의무 부과하던
교토의정서 체제를 넘어 모든 국가가 참여

1997 교토의정서

6가지 온실가스 정의
배출권거래제, 공동이행제도 등 도입

2011~ 더반플랫폼 합의

기온 상승을 산업화 이전 대비
2도 이내로 억제하기 위하여
2020년 이후의 국가별 기여
방안 결정

“

기후변화에 대한 노력은 잘 이루어지고 있는가?.

”

PART 2.

태양열, 풍력 발전 1차 분석



>> 목적

1. 신재생에너지 발전규모가 큰 국가와 에너지전환에 적극적인 국가를 알아본다
2. 신재생에너지 발전량과 연관된 요인을 분석한다

.

>> 데이터 수집 및 분석 방법

1. Our World in Data / BP (British Petroleum) /
2. 통계시각화 및 회귀분석

01_태양열, 풍력에너지 현황

사용 라이브러리

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas import Series, DataFrame
import matplotlib
```

사용 데이터

출처 : BP 에너지원별 보고서

solar.head()

	ISO	Megawatts	1996	1997	1998	1999	2000	2001	2002	2003	...	2014	2015	2016	2017	2018	2019	2020	growth_rate_2020
0	CAN	Canada	3	3	5	6	7	9	10	12	...	1843	2517	2661	2913	3100	3310	3325	0.20%
1	MEX	Mexico	10	11	12	13	14	15	16	16	...	116	173	389	674	2541	4426	5630	26.90%
2	USA	US	14	15	15	17	19	22	28	73	...	15984	21684	32958	41357	51426	58924	73814	24.90%
3	ARG	Argentina	0	0	0	0	0	0	0	0	...	8	9	9	9	191	442	764	72.40%
4	BRA	Brazil	0	0	0	0	0	0	0	0	...	20	41	148	1296	2470	4615	7881	70.30%

5 rows × 30 columns

wind.head()

	ISO	Megawatts	1995	1996	1997	1998	1999	2000	2001	2002	...	2014	2015	2016	2017	2018	2019	2020	grc
0	CAN	Canada	0.0	0.0	26.0	83.0	126.0	139.0	214.0	270.0	...	9694.0	11214.0	11973.0	12403.0	12816.0	13413.0	13577.0	
1	MEX	Mexico	0.0	0.0	2.0	2.0	2.0	17.0	17.0	18.0	...	2569.0	3271.0	4051.0	4199.0	4875.0	6591.0	8128.0	
2	USA	US	0.0	0.0	1611.0	2141.0	2445.0	2377.0	3864.0	4417.0	...	64232.0	72573.0	81286.0	87597.0	94417.0	103571.0	117744.0	
3	ARG	Argentina	0.0	0.0	9.0	14.0	14.0	14.0	25.0	26.0	...	215.0	187.0	187.0	257.0	750.0	1609.0	2624.0	
4	BRA	Brazil	0.0	0.0	4.0	19.0	22.0	22.0	24.0	22.0	...	4888.0	7633.0	10129.0	12304.0	14843.0	15438.0	17198.0	

5 rows × 31 columns

2017-2019 데이터 변형 및 추가

```
solar_3yrs = solar.iloc[:, [0, 23, 24, 25]]
solar_3yrs.head()
```

```
wind_3yrs = wind.iloc[:, [0, 24, 25, 26]]
wind_3yrs.head()
```

```
solar_3yrs_sum = solar_3yrs.iloc[:, 1:4].sum(axis=1)
solar['3yrs_sum'] = solar_3yrs.iloc[:, 1:4].sum(axis=1)
solar.head()
```

	2016	2017	2018	2019	2020	growth_rate_2020	growth_rate_9020	share_2020	3yrs_sum
0	361	2913	3100	3310	3325	0.20%	42.60%	0.50%	9323
1	389	674	2541	4426	5630	26.90%	67.80%	0.80%	7641
2	358	41357	51426	58924	73814	24.90%	47.80%	10.40%	151707
3	9	9	191	442	764	72.40%	na	0.10%	642
4	148	1296	2470	4615	7881	70.30%	na	1.10%	8381

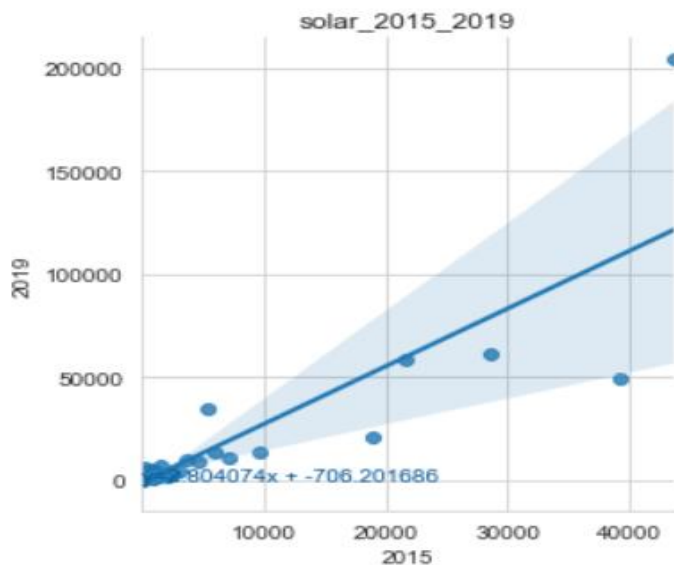
```
wind_3yrs_sum = wind_3yrs.iloc[:, 1:4].sum(axis=1)
wind['3yrs_sum'] = wind_3yrs.iloc[:, 1:4].sum(axis=1)
wind.head()
```

	2017	2018	2019	2020	growth_rate_2020	growth_rate_9020	share_2020	3yrs_sum
0	361.0	2913.0	3100.0	3325.0	0.90%	15.10%	1.90%	38632.0
1	389.0	4875.0	6591.0	8128.0	23.00%	31.50%	1.10%	15665.0
2	357.0	94417.0	103571.0	117744.0	13.40%	11.70%	16.10%	285585.0
3	57.0	750.0	1609.0	2624.0	62.60%	50.10%	0.40%	2616.0
4	148.0	14843.0	15438.0	17198.0	11.10%	38.30%	2.30%	42585.0

01_태양에너지 현황

```
from statsmodels.formula.api import ols
import statsmodels.api as sm
```

```
z = np.polyfit(solar['2015'], solar['2019'], 1)
f = np.poly1d(z)
print("기울기 :", z[0], ', ', "y절편 :", z[1])
print("f(1):", f(1))
###
sns.set_style("whitegrid")
sns.lmplot(x='2015', y='2019', data=solar, size=4)
plt.title("solar_2015_2019")
plt.text(4,10, "y=%fx + %f" % (z[0], z[1]), color = '#005599')
plt.show()
```



기울기 : 2.804074212834738 , y절편 : -706.2016858120044
f(1): -703.3976115991696

1_ 대부분의 국가가 태양에너지를

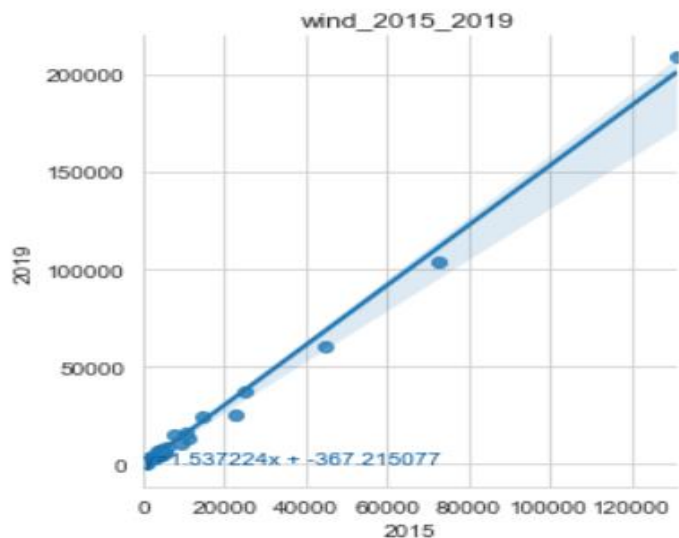
확대하고자 하는 세계적 흐름에 따라가고 있음

2_ Solar는 회귀선에서 동떨어진 국가가 꽤 있음.

01_풍력에너지 현황

```
from statsmodels.formula.api import ols
import statsmodels.api as sm
```

```
z = np.polyfit(wind['2015'], wind['2019'], 1)
f = np.polyval(z)
print("기울기 :", z[0], ', ', "y절편 :", z[1])
print("f(1):", f(1))
###
sns.set_style("whitegrid")
sns.lmplot(x='2015', y='2019', data=wind, size=3)
plt.title("wind_2015_2019")
plt.text(4,10, "y=%fx + %f" % (z[0], z[1]), color = '#005599')
plt.show()
```



기울기 : 1.5372244135482653 , y절편 : -367.21507666814915
f(1): -365.6778522546009

1_ solar와 wind 모두 대부분의 국가가 재생에너지를 확대하고자 하는 세계적 흐름에 따라가고 있음

2_ wind는 회귀선에 근접함,

많은 국가가 2015년에 비해 2019년에 약 1.5배 증가

3_ Solar는 회귀선에서 동떨어진 국가가 꽤 있음.

wind에 비해 solar는 국가의 개별적 요인에 의해 영향을 많이 받는 재생에너지원일까?

주의 : solar보다 wind가 가파르게 증가한 듯 보이나,

2015년의 x축값의 범위가 달라 실제로는 solar가 더 가파르게 증가함.

보완점 1) x축과 y축의 범위 설정할 수 있다면 증가추세 시각적으로도 정확히 표현할 수 있을 것

보완점 2) 회귀식 위치 조정

01_재생에너지 현황_# 발생한 문제 및 해결과정

```
wind.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 50 entries, 0 to 49
```

```
Data columns (total 32 columns):
```

#	Column	Non-Null Count	Dtype
0	ISO	45 non-null	object
1	Megawatts	49 non-null	object
2	1995	45 non-null	float64
3	1996	45 non-null	float64
4	1997	45 non-null	float64
5	1998	45 non-null	float64
6	1999	45 non-null	float64
7	2000	45 non-null	float64
8	2001	45 non-null	float64
9	2002	45 non-null	float64
10	2003	45 non-null	float64
11	2004	45 non-null	float64
12	2005	45 non-null	float64
13	2006	45 non-null	float64
14	2007	45 non-null	float64
15	2008	45 non-null	float64
16	2009	45 non-null	float64
17	2010	45 non-null	float64
18	2011	45 non-null	float64
19	2012	45 non-null	float64
20	2013	45 non-null	float64
21	2014	45 non-null	float64
22	2015	45 non-null	float64
23	2016	45 non-null	float64
24	2017	45 non-null	float64
25	2018	45 non-null	float64
26	2019	45 non-null	float64
27	2020	45 non-null	float64

발생한 문제 : float64형태의 데이터로는 산점도 그릴 수 없었음

LinAlgError: SVD did not converge in Linear Least Squares

```
wind['2019'] = wind['2019'].fillna(-1)
```

```
wind['2019'] = wind['2019'].astype('int')
```

```
wind.info()
```

15	2008	45 non-null	float64
16	2009	45 non-null	float64
17	2010	45 non-null	float64
18	2011	45 non-null	float64
19	2012	45 non-null	float64
20	2013	45 non-null	float64
21	2014	45 non-null	float64
22	2015	50 non-null	int32
23	2016	45 non-null	float64
24	2017	45 non-null	float64
25	2018	45 non-null	float64
26	2019	50 non-null	int32
27	2020	45 non-null	float64

02_GDP 및 인구와의 상관관계

>> 사용 라이브러리

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib
```

```
from pandas import Series, DataFrame
from matplotlib import font_manager
```

```
from statsmodels.formula.api import ols
import statsmodels.api as sm
```

>> 데이터 전처리

```
# population_m = population/1,000,000
# population_b = population/1,000,000,000
# gdp_b = gdp/1,000,000,000
# gdp_t = gdp/1,000,000,000,000
```

```
solar_2019['gdp_t'] = solar_2019['gdp_b']/1000
wind_2019['gdp_t'] = wind_2019['gdp_b']/1000
```

```
solar_2019['population_m'] = solar_2019['population']/1000000
wind_2019['population_m'] = wind_2019['population']/1000000
```

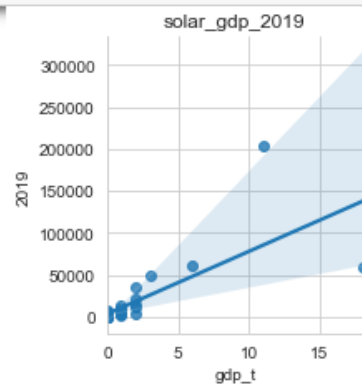
```
solar_2019['population_b'] = solar_2019['population']/1000000000
wind_2019['population_b'] = wind_2019['population']/1000000000
```

```
solar_2019['gdp_b'] = solar_2019['gdp_b'].astype('int')
wind_2019['gdp_b'] = wind_2019['gdp_b'].astype('int')
solar_2019['gdp_t'] = solar_2019['gdp_t'].astype('int')
wind_2019['gdp_t'] = wind_2019['gdp_t'].astype('int')
solar_2019['population_m'] = solar_2019['population_m'].astype('int')
wind_2019['population_m'] = wind_2019['population_m'].astype('int')
```

02_GDP 및 인구와의 상관관계

>> gdp_태양열

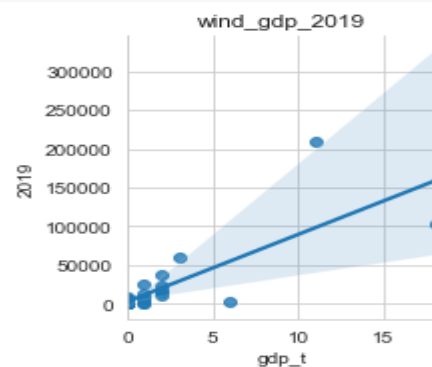
```
sns.set_style("whitegrid")
sns.lmplot(x="gdp_t", y='2019', data=solar_2019, size=3)
plt.title("solar_gdp_2019")
plt.show()
solar_2019[['gdp_t', '2019']].corr()
```



	gdp_t	2019
gdp_t	1.000000	0.729093
2019	0.729093	1.000000

>> gdp_풍력

```
sns.set_style("whitegrid")
sns.lmplot(x='gdp_t', y='2019', data=wind_2019, size=3)
plt.title("wind_gdp_2019")
plt.show()
wind_2019[['gdp_t', '2019']].corr()
```



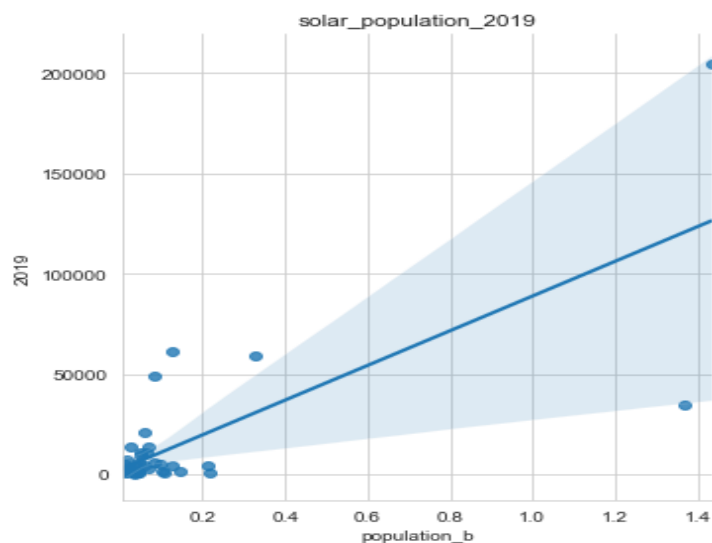
	gdp_t	2019
gdp_t	1.000000	0.790502
2019	0.790502	1.000000

1. gdp가 증가함에 따라 태양열, 풍력 둘 다 증가하는 경향
2. 이상치 (solar _ 중국/미국 , wind_ 중국/미국)

02_GDP 및 인구와의 상관관계

>> 인구_태양열

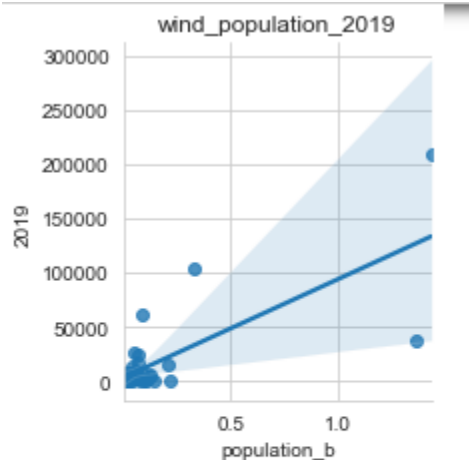
```
sns.set_style("whitegrid")
sns.lmplot(x="population_b", y='2019', data=solar_2019, size=5)
plt.title("solar_population_2019")
plt.show()
solar_2019[['population_b', '2019']].corr()
```



	population_b	2019
population_b	1.000000	0.759915
2019	0.759915	1.000000

>> 인구_풍력

```
sns.set_style("whitegrid")
sns.lmplot(x="population_b", y='2019', data=wind_2019, size=5)
plt.title("wind_population_2019")
plt.show()
wind_2019[['population_b', '2019']].corr()
```



	population_b	2019
population_b	1.000000	0.747377
2019	0.747377	1.000000

1. 인구가 증가함에 따라 태양열, 풍력 둘 다 증가하는 경향
2. 이상치 (solar _ 중국/인도 , wind _ 중국/인도)

03_ 자연재해와의 상관관계

자연재해 선정 이유 : 자연재해의 피해를 몸소 겪은 국가는 기후변화에 더 민감하게 반응하여 에너지 전환을 중요하게 여길 것이다

>> 데이터 파일 병합

```
disaster.head()
```

	Entity	Code	Year	total_disaster	disaster/person
0	Afghanistan	AFG	2019	117897320	1178.97320
1	Albania	ALB	2019	4052785	40.52785
2	Algeria	DZA	2019	160459033	1604.59033
3	Armenia	ARM	2019	47050238	470.50238
4	Australia	AUS	2019	5808438	58.08438

```
solar_2019_ = solar_2019.merge(disaster, left_on = 'Code', right_on = 'Code')
solar_2019_
```

```
wind_2019_ = wind_2019.merge(disaster, left_on = 'Code', right_on = 'Code')
wind_2019_
```

```
solar_2019_.columns
```

```
Index(['Code', 'country', 'year', 'co2', 'co2_growth_prct', 'co2_growth_abs',
      'co2_per_capita', 'share_global_co2', 'cumulative_co2',
      'share_global_cumulative_co2', 'population', 'Year_x', 'gdp_b', 'gdp',
      'ISO', '2017', '2018', '2019', '3yrs_sum', 'gdp_t', 'population_m',
      'population_b', 'Entity', 'Year_y', 'total_disaster',
      'disaster/person'],
      dtype='object')
```

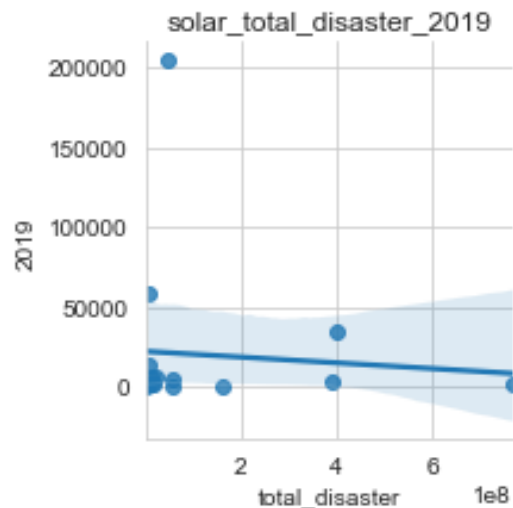
```
wind_2019_.columns
```

```
Index(['Code', 'country', 'year', 'co2', 'co2_growth_prct', 'co2_growth_abs',
      'co2_per_capita', 'share_global_co2', 'cumulative_co2',
      'share_global_cumulative_co2', 'population', 'Year_x', 'gdp_b', 'gdp',
      'ISO', '2017', '2018', '2019', '3yrs_sum', 'gdp_t', 'population_m',
      'population_b', 'Entity', 'Year_y', 'total_disaster',
      'disaster/person'],
      dtype='object')
```

03_ 자연재해와의 상관관계

자연재해와 태양열

```
sns.set_style("whitegrid")
sns.lmplot(x="total_disaster", y='2019', data=solar_f, size=5)
plt.title("solar_total_disaster_2019")
plt.show()
```

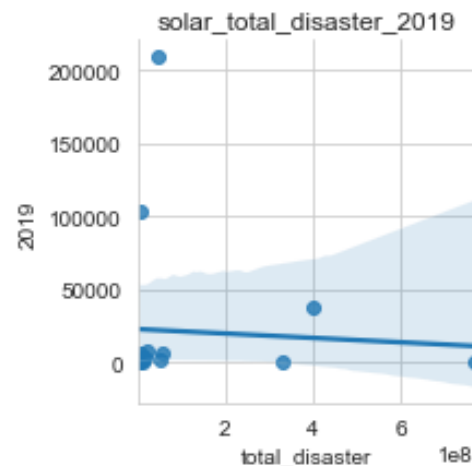


```
solar_f[['total_disaster', '2019']].corr()
```

	total_disaster	2019
total_disaster	1.000000	-0.075862
2019	-0.075862	1.000000

자연재해와 풍력

```
sns.set_style("whitegrid")
sns.lmplot(x="total_disaster", y='2019', data=wind_f, size=5)
plt.title("solar_total_disaster_2019")
plt.show()
```



```
wind_f[['total_disaster', '2019']].corr()
```

	total_disaster	2019
total_disaster	1.000000	-0.057496
2019	-0.057496	1.000000

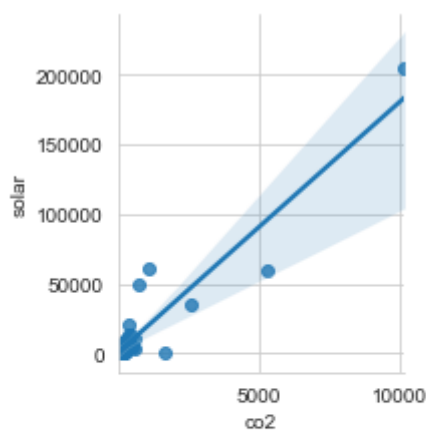
자연재해와 재생에너지 발전량은 상관관계가 없다.

1인당 자연재해와 재생에너지 발전량 관계도 거의 유사한 결과를 얻었다.

04_ 태양열-co2와의 상관관계

< Co2발생량 >

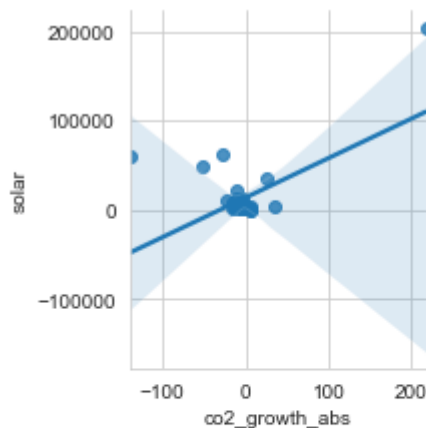
```
sns.set_style("whitegrid")
sns.lmplot(x="co2", y='solar', data=solar, size=5)
plt.show()
solar[['co2', 'solar']].corr()
```



	co2	solar
co2	1.000000	0.929917
solar	0.929917	1.000000

< Co2 증가 절대량 >

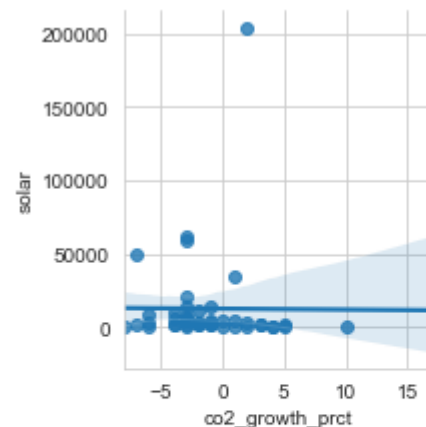
```
sns.set_style("whitegrid")
sns.lmplot(x="co2_growth_abs", y='solar', data=solar, size=3)
plt.show()
solar[['co2_growth_abs', 'solar']].corr()
```



	co2_growth_abs	solar
co2_growth_abs	1.000000	0.556678
solar	0.556678	1.000000

< Co2증가 비율 >

```
sns.set_style("whitegrid")
sns.lmplot(x="co2_growth_prct", y='solar', data=solar, size=3)
plt.show()
solar[['co2_growth_prct', 'solar']].corr()
```



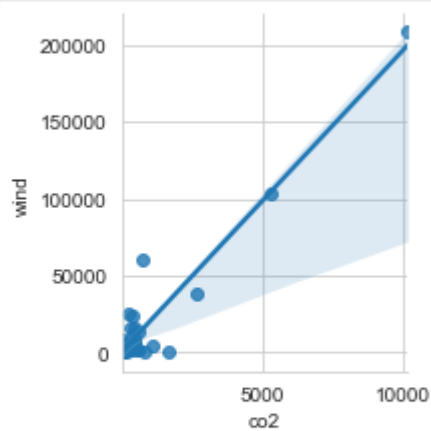
	co2_growth_prct	solar
co2_growth_prct	1.000000	-0.008401
solar	-0.008401	1.000000

CO2와 태양열 발전량은 높은 상관관계가 있으며 1인당 CO2발생량 또한 거의 유사한 결과를 얻었다.
 CO2의 절대적 증가량은 태양열 발전량과의 상관관계가 높지 않다
 CO2의 증가비율은 태양열 발전량과의 상관관계가 거의 없다

04_ 풍력-co2와의 상관관계

< Co2발생량 >

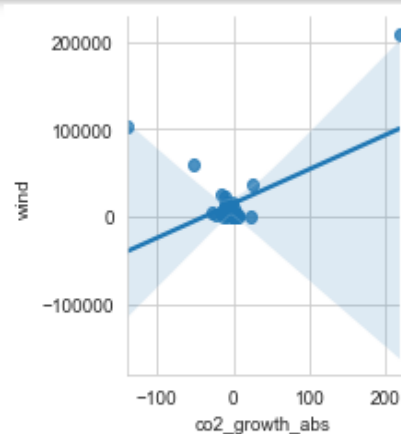
```
sns.set_style("whitegrid")
sns.lmplot(x="co2", y="wind", data=wind, size=3)
plt.show()
wind[['co2', 'wind']].corr()
```



	co2	wind
co2	1.000000	0.948925
wind	0.948925	1.000000

< Co2 증가 절대량 >

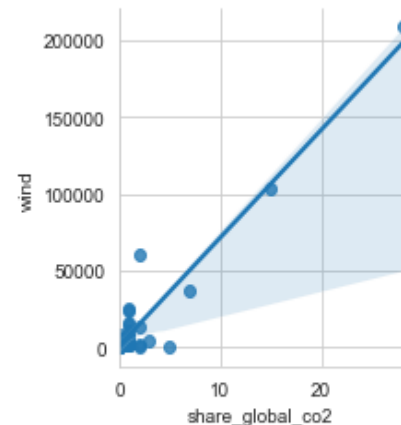
```
sns.set_style("whitegrid")
sns.lmplot(x="co2_growth_abs", y="wind", data=wind, size=3)
plt.show()
wind[['co2_growth_abs', 'wind']].corr()
```



	co2_growth_abs	wind
co2_growth_abs	1.000000	0.459631
wind	0.459631	1.000000

< 세계Co2발생 비중 >

```
sns.set_style("whitegrid")
sns.lmplot(x="share_global_co2", y="wind", data=wind, size=3)
plt.show()
wind[['share_global_co2', 'wind']].corr()
```



	share_global_co2	wind
share_global_co2	1.000000	0.947309
wind	0.947309	1.000000

CO2와 풍력 발전량, 세계 CO2발생 비중과 태양열 발전량은 상관관계가 높다.
CO2의 증가절대량은 풍력 발전량과의 상관관계가 높지 않다

05_ 회귀분석

>>상관관계가 0.7이상인 변수

<solar>

0	Code	44 non-null	object
1	country	44 non-null	object
2	year	44 non-null	int64
3	co2	44 non-null	int64
4	co2_growth_prct	44 non-null	int64
5	co2_growth_abs	44 non-null	int64
6	co2_per_capita	44 non-null	int64
7	share_global_co2	44 non-null	int64
8	cumulative_co2	44 non-null	int64
9	share_global_cumulative_co2	44 non-null	int64
10	population	44 non-null	int64
11	Year	44 non-null	int64
12	gdp_b	44 non-null	float64
13	gdp	44 non-null	int64
14	ISO	44 non-null	object
15	2017	44 non-null	int64
16	2018	44 non-null	int64
17	2019	44 non-null	int64
18	3yrs_sum	44 non-null	int64

<wind>

0	Code	44 non-null	object
1	country	44 non-null	object
2	year	44 non-null	int64
3	co2	44 non-null	int64
4	co2_growth_prct	44 non-null	int64
5	co2_growth_abs	44 non-null	int64
6	co2_per_capita	44 non-null	int64
7	share_global_co2	44 non-null	int64
8	cumulative_co2	44 non-null	int64
9	share_global_cumulative_co2	44 non-null	int64
10	population	44 non-null	int64
11	Year	44 non-null	int64
12	gdp_b	44 non-null	float64
13	gdp	44 non-null	int64
14	ISO	44 non-null	object
15	2017	44 non-null	int64
16	2018	44 non-null	int64
17	2019	44 non-null	int64
18	3yrs_sum	44 non-null	int64

05_ 회귀분석

>> 모형: $\text{solar} = \text{co2} + \text{share_global_co2} + \text{share_global_cumulative_co2} + \text{population} + \text{gdp_b}$

```
from sklearn.linear_model import LinearRegression
import statsmodels.formula.api as sm
```

```
result = sm.ols(formula = 'solar ~ co2 + share_global_co2 + share_global_cumulative_co2 + population + gdp_b', data = solar).fit()
result.summary()
```

	coef	std err	t	P> t	[0.025	0.975]	Dep. Variable:	solar	R-squared:	0.912	Omnibus:	45.074	Durbin-Watson:	2.032
Intercept	407.4948	1848.344	0.220	0.827	-3331.133	4146.123	Model:	OLS	Adj. R-squared:	0.901	Prob(Omnibus):	0.000	Jarque-Bera (JB):	202.195
co2	29.3817	18.291	1.606	0.116	-7.615	66.379	Method:	Least Squares	F-statistic:	81.20	Skew:	2.453	Prob(JB):	1.24e-44
share_global_co2	-2861.2711	6567.998	-0.436	0.666	-1.61e+04	1.04e+04	Date:	Fri, 08 Oct 2021	Prob (F-statistic):	1.51e-19	Kurtosis:	12.152	Cond. No.	1.32e+09
share_global_cumulative_co2	-5698.5263	1315.057	-4.333	0.000	-8358.481	-3038.572	Time:	10:16:16	Log-Likelihood:	-476.35				
population	-1.393e-05	1.06e-05	-1.319	0.195	-3.53e-05	7.43e-06	No. Observations:	45	AIC:	964.7				
gdp_b	6.1998	1.795	3.453	0.001	2.569	9.831	Df Residuals:	39	BIC:	975.5				
							Df Model:	5						
							Covariance Type:	nonrobust						

```
print(result.params)
```

```
Intercept      407.494841
co2             29.381660
share_global_co2 -2861.271082
share_global_cumulative_co2 -5698.526270
population      -0.000014
gdp_b           6.199780
dtype: float64
```

P>[t] : p value
: 0.05보다 작고 0에 가까울수록 매우 유의미한 데이터, 0.05보다 크면 불필요한 데이터

05_회귀분석

>> 모형: $\text{solar} = \text{co2} + \text{population} + \text{gdp_b}$

```
# co2 다중공선성 문제 보완 |
result = sm.ols(formula = 'solar ~ co2 + population + gdp_b', data = solar).fit()
result.summary()
```

P>[t] : p value
: 0.05보다 작고 0에 가까울수록 매우 유의미한 데이터, 0.05보다 크면 불필요한 데이터

OLS Regression Results

Dep. Variable:	solar	R-squared:	0.866	Omnibus:	22.237	Durbin-Watson:	1.767
Model:	OLS	Adj. R-squared:	0.856	Prob(Omnibus):	0.000	Jarque-Bera (JB):	66.224
Method:	Least Squares	F-statistic:	88.41	Skew:	1.083	Prob(JB):	4.17e-15
Date:	Fri, 08 Oct 2021	Prob (F-statistic):	6.09e-18	Kurtosis:	8.534	Cond. No.	3.61e+08
Time:	10:40:50	Log-Likelihood:	-485.88				
No. Observations:	45	AIC:	979.8				
Df Residuals:	41	BIC:	987.0				
Df Model:	3						
Covariance Type:	nonrobust						

	coef	std err	t	P> t	[0.025	0.975]
Intercept	1195.5745	2168.961	0.551	0.584	-3184.732	5575.881
co2	19.3150	3.059	6.315	0.000	13.138	25.492
population	-1.998e-06	1.23e-05	-0.162	0.872	-2.69e-05	2.29e-05
gdp_b	-0.6948	1.091	-0.637	0.528	-2.898	1.509

05_ 회귀분석

>> 모형: solar = share_global_cumulative_co2 + population + gdp_b

co2 다중공선성 문제 보완 4

```
result = sm.ols(formula = 'solar ~ share_global_cumulative_co2 + population + gdp_b', data = solar).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	solar	R-squared:	0.756	Omnibus:	18.792	Durbin-Watson:	2.016
Model:	OLS	Adj. R-squared:	0.738	Prob(Omnibus):	0.000	Jarque-Bera (JB):	115.364
Method:	Least Squares	F-statistic:	42.42	Skew:	0.358	Prob(JB):	8.89e-26
Date:	Fri, 08 Oct 2021	Prob (F-statistic):	1.23e-12	Kurtosis:	10.811	Cond. No.	3.56e+08
Time:	10:43:14	Log-Likelihood:	-499.36				
No. Observations:	45	AIC:	1007.				
Df Residuals:	41	BIC:	1014.				
Df Model:	3						
Covariance Type:	nonrobust						

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-2821.7950	2858.137	-0.987	0.329	-8593.921	2950.331
share_global_cumulative_co2	-3813.0170	2057.451	-1.853	0.071	-7968.123	342.089
population	5.61e-05	1.04e-05	5.403	0.000	3.51e-05	7.71e-05
gdp_b	9.6777	2.831	3.419	0.001	3.961	15.395

P>[t] : p value

: 0.05보다 작고 0에 가까울수록 매우 유의미한 데이터, 0.05보다 크면 불필요한 데이터

05_회귀분석

>>모형: wind=co2+share_global_co2+cumulative_co2+share_global_cumulative_co2+population+gdp_b

```
result = sm.ols(formula = 'wind ~ co2 + share_global_co2 + cumulative_co2 + share_global_cumulative_co2 + population + gdp_b',
                data = wind).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	wind	R-squared:	0.911	Omnibus:	20.588	Durbin-Watson:	1.812
Model:	OLS	Adj. R-squared:	0.897	Prob(Omnibus):	0.000	Jarque-Bera (JB):	79.847
Method:	Least Squares	F-statistic:	63.30	Skew:	0.831	Prob(JB):	4.59e-18
Date:	Fri, 08 Oct 2021	Prob (F-statistic):	5.86e-18	Kurtosis:	9.387	Cond. No.	1.65e+09
Time:	10:51:54	Log-Likelihood:	-469.44				
No. Observations:	44	AIC:	952.9				
Df Residuals:	37	BIC:	965.4				
Df Model:	6						
Covariance Type:	nonrobust						

	coef	std err	t	P> t	[0.025	0.975]
Intercept	734.8144	2149.772	0.342	0.734	-3621.037	5090.666
co2	19.5724	23.510	0.833	0.410	-28.063	67.208
share_global_co2	303.5916	8543.186	0.036	0.972	-1.7e+04	1.76e+04
cumulative_co2	-0.6552	0.496	-1.320	0.195	-1.661	0.350
share_global_cumulative_co2	8354.8101	7556.004	1.106	0.276	-6955.109	2.37e+04
population	-8.836e-06	1.16e-05	-0.759	0.452	-3.24e-05	1.47e-05
gdp_b	3.4354	2.085	1.648	0.108	-0.788	7.659

05_ 회귀분석

>> 모형: wind = co2 + cumulative_co2 + population + gdp_b

co2의 다중공선성 문제

```
result = sm.ols(formula = 'wind ~ cumulative_co2 + population + gdp_b', data = wind).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	wind	R-squared:	0.793	Omnibus:	17.266	Durbin-Watson:	1.985
Model:	OLS	Adj. R-squared:	0.777	Prob(Omnibus):	0.000	Jarque-Bera (JB):	84.892
Method:	Least Squares	F-statistic:	50.93	Skew:	0.401	Prob(JB):	3.68e-19
Date:	Fri, 08 Oct 2021	Prob (F-statistic):	1.01e-13	Kurtosis:	9.757	Cond. No.	3.55e+08
Time:	10:50:58	Log-Likelihood:	-488.11				
No. Observations:	44	AIC:	984.2				
Df Residuals:	40	BIC:	991.4				
Df Model:	3						
Covariance Type:	nonrobust						

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-2783.8822	2874.959	-0.968	0.339	-8594.392	3026.628
cumulative_co2	0.0243	0.133	0.184	0.855	-0.243	0.292
population	5.552e-05	1.03e-05	5.387	0.000	3.47e-05	7.63e-05
gdp_b	5.6113	2.954	1.900	0.065	-0.359	11.581

05_회귀분석 1차_결론 및 한계

>> 중간 결론

$\text{solar} = \text{co2} + \text{share_global_cumulative_co2} + \text{population} + \text{gdp_b}$

$\text{wind} = \text{co2} + \text{cumulative_co2} + \text{population} + \text{gdp_b}$

>> 한계

__ 채택된 변수가 적고 모형이 두루뭉술하다.

__ **population** 변수에 대해 일관성을 지키지 못했다.

__ 통계적 해석 미비하였다.

05_ 회귀분석 1차_보완점 및 향후 계획

>> 향후 계획

1. 과제 개요

- 신재생에너지 발전규모가 큰 국가와 에너지 전환에 적극적인 국가를 알아본다.
- 국가별 신재생에너지 발전에 유리한 요인을 분석한다.
- 한국의 신재생에너지 발전에 대한 유불리 요인을 확인하고 어떤 특성을 고려하여 에너지 전환을 도모해야 할지 전략을 제시한다.

2. 모형 보완을 위해 추가할 변수

1. 기후협약 개수 : 국가적 지위에 따른 국제적 약속 (Annex 1/ Annex2 / non-Annex)
2. 국토면적 및 구성 : 산림비율
3. 기후 (ex : 태양열 – 연평균기온, 연교차 / 풍력 – 풍속)
4. 환경정책 선진성 : (ex – 배출권거래제 도입한 년도, 배출권거래 규모)
5. 지정학적 특성
 - _ 직접 : 이웃국가와 신재생에너지 거래 가능한지
 - _ 간접 : 주변 국가와의 우호적 관계 (공동협약 개수)



PART 3.

태양열 발전 2차 분석

변수 추가

UN : un기후협약 당시 국제적 협력도
_annex 2 / annex1 / non-annex :

무역량 (수출 및 수입)

면적

country	un_0	un_1	un_2
Algeria	1	0	0
Argentina	1	0	0
Australia	0	0	1
Austria	0	0	1
Belgium	0	0	1
Brazil	1	0	0
Bulgaria	0	1	0
Canada	0	0	1
Chile	1	0	0
China	1	0	0
Czechia	0	1	0
Denmark	0	0	1
Egypt	1	0	0
France	0	0	1
Germany	0	0	1

export_mil	import_mi	ln_trade	trade
36300	44600	4.907949	80900
65950	57230	5.09054	123180
184300	203100	5.58816	387400
142900	138600	5.449478	281500
250800	257100	5.705778	507900
189700	143900	5.523226	333600
24620	28470	4.725013	53090
402400	419000	5.914555	821400
62230	56860	5.075875	119090
2011000	1437000	6.537567	3448000
131000	132400	5.420616	263400
95970	82290	5.251054	178260
20880	50070	4.850952	70950
505400	525400	6.013174	1030800
1283000	987600	6.356141	2270600
26700	42730	4.841547	69430

ln_area	area_total	area_con	area_con_a
5.376894	238174	238174	8517
5.444107	278040	273669	33701
5.888809	774122	769202	30920
3.923658	8388	8252	1392
3.484727	3053	3028	881
5.930224	851577	835814	63518
4.045323	11100	10856	3629
5.994746	987975	896559	38815
4.878924	75670	74353	1656
5.982272	960001	942470	135675
3.896912	7887	7721	2532
3.63266	4292	4000	2419
4.480108	30207	29572	9329
4.739644	54909	54756	19075
4.553373	35758	34939	11913
4.120442	13196	12890	3222

변수 추가

기온

ln_temp_a	temp_av	temp_cold	temp_hot	ln_temp_v	temp_var
1.365862	23.22	10.9	34.1	1.365488	23.2
1.162863	14.55	5.2	22.5	1.238046	17.3
1.341237	21.94	13.2	29.9	1.222716	16.7
0.860338	7.25	-6	19.2	1.401401	25.2
1.028571	10.68	-1.1	21.6	1.356026	22.7
1.40671	25.51	22.7	27.2	0.653213	4.5
1.053463	11.31	-4	23.8	1.444045	27.8
-999	-6.36	-27.6	12.8	1.606381	40.4
0.940018	8.71	2.7	14.1	1.056905	11.4
0.851258	7.1	-10.5	21	1.498311	31.5
0.941014	8.73	-5.8	21.2	1.431364	27
0.942504	8.76	-3.9	19.7	1.372912	23.6
1.361728	23	11.9	31.6	1.294466	19.7
1.067815	11.69	1	22.8	1.338456	21.8
0.983626	9.63	-3.6	21.7	1.403121	25.3
1.15351	14.24	2.2	25.6	1.369216	23.4

1인당gdp

ln_gdpper	gdpper1
4	4692
4	9781
5	57532
5	50139
5	47403
4	11182
4	9027
5	52124
4	15038
4	8021
4	24230
5	66356
3	3008
5	45601
5	47176
4	24061

< 변수 >

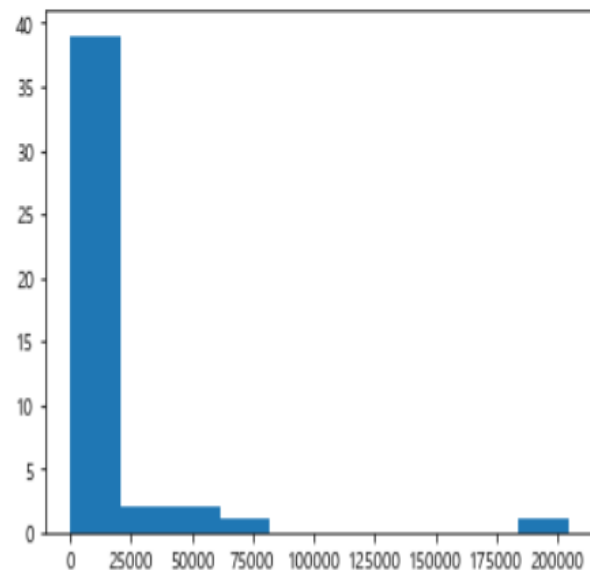
- 태양에너지
- GDP
- 무역량
- CO2
- 인구
- 기온 (평균기온, 연교차)
- 국토면적
- 국제적 협력정도

모델 선택_1: 정수vs로그

```
X = solar[['trade', 'gdp', 'co2', 'population', 'un_0', 'un_1', 'un_2', 'area_total', 'temp_var']]
y = solar['solar']
```

```
plt.hist(y)
```

```
(array([39., 2., 2., 1., 0., 0., 0., 0., 0., 1.]),
 array([2.040000e+02, 2.064110e+04, 4.107820e+04, 6.151530e+04,
        8.195240e+04, 1.023895e+05, 1.228266e+05, 1.432637e+05,
        1.637008e+05, 1.841379e+05, 2.045750e+05]),
 <BarContainer object of 10 artists>)
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
model = LinearRegression().fit(X_train, y_train)
pred = model.predict(X_test)
pred
```

```
array([ 5934.49539137, -420.19069022,  9167.54414941, 186884.90268033,
        20899.87023132, -654.79563427,  4128.42542275,  2669.68855505,
         860.67251342, 1760.41661322, -2593.02764359, 12225.85952303])
```

```
print("훈련 데이터 세트 점수 : {:.2f}".format(model.score(X_train, y_train)))
print("테스트 데이터 세트 점수 : {:.2f}".format(model.score(X_test, y_test)))
```

훈련 데이터 세트 점수 : 0.98

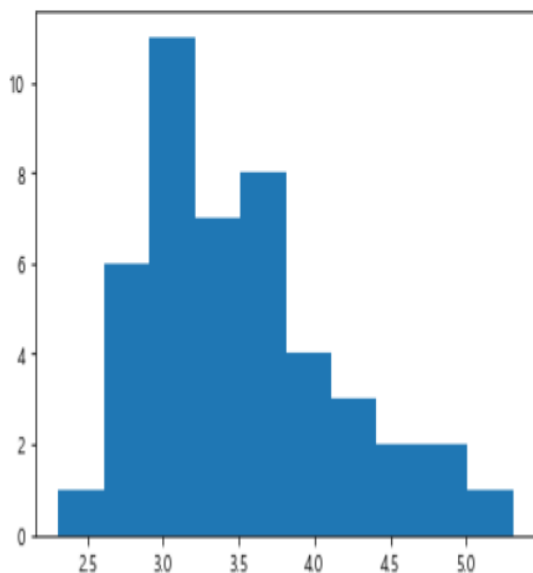
테스트 데이터 세트 점수 : -4.79

모델 선택_1: 정수vs로그

```
X = solar[['ln_trade', 'ln_gdp', 'ln_co2', 'ln_population', 'un_0', 'un_1', 'un_2', 'ln_area', 'ln_temp_av']]
y = solar['ln_solar']
```

```
plt.hist(y)
```

```
(array([ 1.,  6., 11.,  7.,  8.,  4.,  3.,  2.,  2.,  1.]),
 array([2.30963017, 2.60975241, 2.90987465, 3.20999688, 3.51011912,
        3.81024136, 4.1103636 , 4.41048584, 4.71060808, 5.01073032,
        5.31085256]),
 <BarContainer object of 10 artists>)
```



```
from sklearn.linear_model import Ridge
```

```
ridge_p = [10, 5, 1, 0.1, 0.01]
for i in ridge_p:
    ridge = Ridge(alpha=i).fit(X_train, y_train)

    print("alpha : {}".format(i))
    print("훈련 데이터 세트 점수 : {:.2f}".format(ridge.score(X_train, y_train)))
    print("테스트 데이터 세트 점수 : {:.2f}".format(ridge.score(X_test, y_test)))
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
model = LinearRegression().fit(X_train, y_train)
pred = model.predict(X_test)
pred
```

```
array([3.63381901, 2.82441196, 4.07542307, 5.0322449 , 4.0653877 ,
        3.23791584, 3.87620598, 3.29824237, 2.85454354, 3.51779675,
        2.67686462, 3.73132805])
```

```
print("훈련 데이터 세트 점수 : {:.2f}".format(model.score(X_train, y_train)))
print("테스트 데이터 세트 점수 : {:.2f}".format(model.score(X_test, y_test)))
```

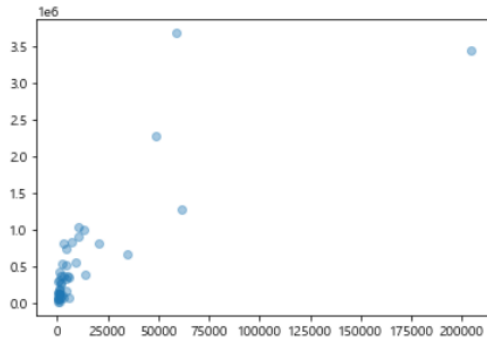
```
훈련 데이터 세트 점수 : 0.83
테스트 데이터 세트 점수 : 0.68
```

```
alpha : 10
훈련 데이터 세트 점수 : 0.74
테스트 데이터 세트 점수 : 0.72
alpha : 5
훈련 데이터 세트 점수 : 0.78
테스트 데이터 세트 점수 : 0.72
alpha : 1
훈련 데이터 세트 점수 : 0.81
테스트 데이터 세트 점수 : 0.69
alpha : 0.1
훈련 데이터 세트 점수 : 0.83
테스트 데이터 세트 점수 : 0.68
alpha : 0.01
훈련 데이터 세트 점수 : 0.83
테스트 데이터 세트 점수 : 0.68
```

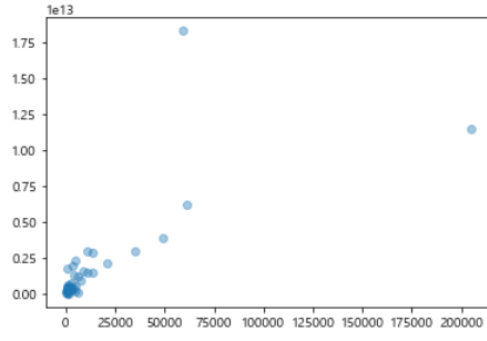
>> 일반모델보다 로그로 변환한 모델이 더 효과적이다

모델선택1_정수vs로그 by산점도

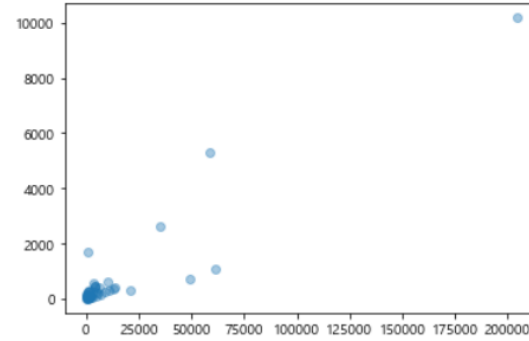
```
plt.scatter(solar[['solar']], solar[['trade']],alpha=0.4)
plt.show()
```



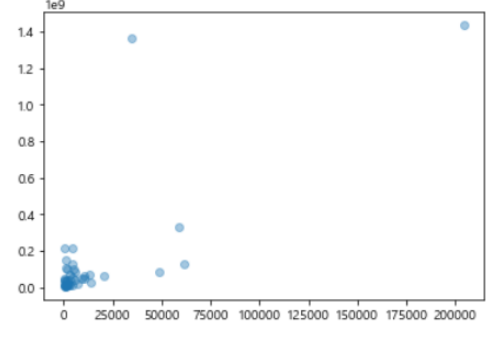
```
plt.scatter(solar[['solar']], solar[['gdp']],alpha=0.4)
plt.show()
```



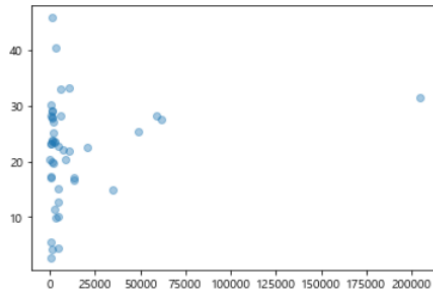
```
plt.scatter(solar[['solar']], solar[['co2']],alpha=0.4)
plt.show()
```



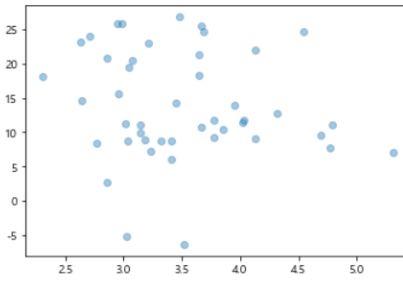
```
plt.scatter(solar[['solar']], solar[['population']],alpha=0.4)
plt.show()
```



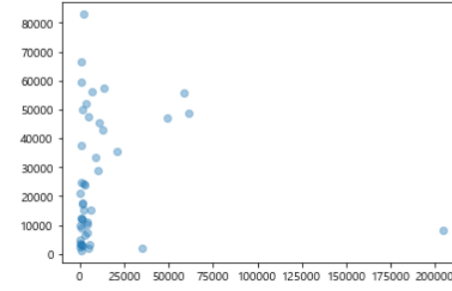
```
plt.scatter(solar[['solar']], solar[['temp_var']],alpha=0.4)
plt.show()
```



```
plt.scatter(solar[['ln_solar']], solar[['temp_av']],alpha=0.4)
plt.show()
```

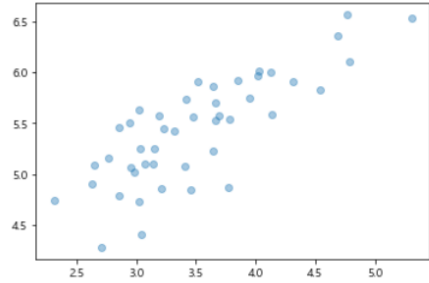


```
plt.scatter(solar[['solar']], solar[['gdpper1']],alpha=0.4)
plt.show()
```

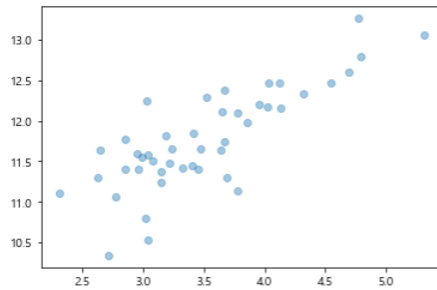


모델선택1_정수vs로그 by산점도

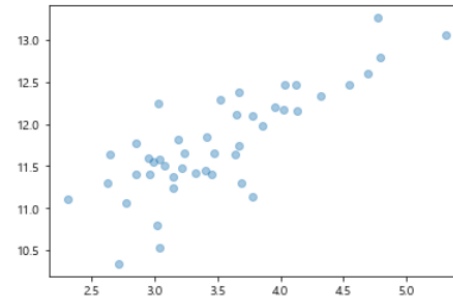
```
plt.scatter(solar[['ln_solar']], solar[['ln_trade']], alpha=0.4)
plt.show()
```



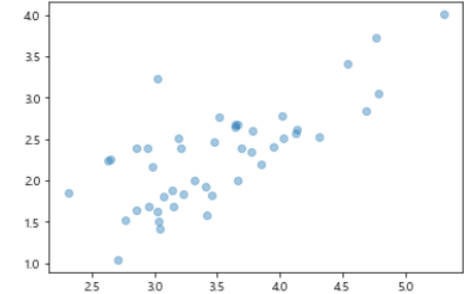
```
plt.scatter(solar[['ln_solar']], solar[['ln_gdp']], alpha=0.4)
plt.show()
```



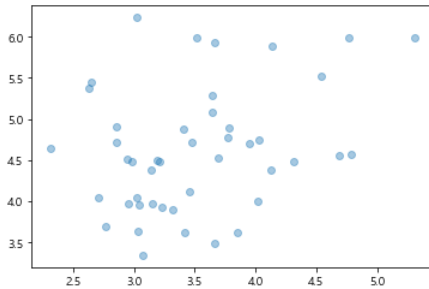
```
plt.scatter(solar[['ln_solar']], solar[['ln_gdp']], alpha=0.4)
plt.show()
```



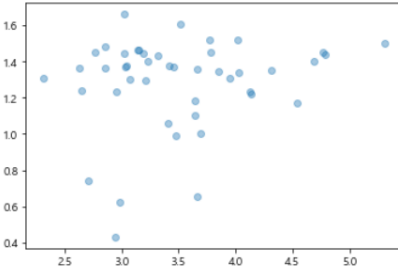
```
plt.scatter(solar[['ln_solar']], solar[['ln_co2']], alpha=0.4)
plt.show()
```



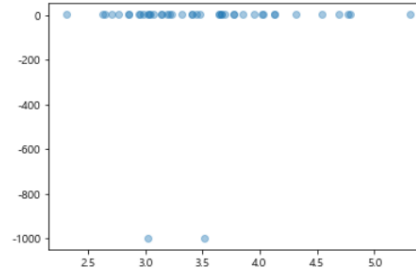
```
plt.scatter(solar[['ln_solar']], solar[['ln_area']], alpha=0.4)
plt.show()
```



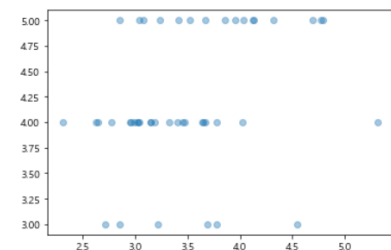
```
plt.scatter(solar[['ln_solar']], solar[['ln_temp_var']], alpha=0.4)
plt.show()
```



```
plt.scatter(solar[['ln_solar']], solar[['ln_temp_av']], alpha=0.4)
plt.show()
```



```
plt.scatter(solar[['ln_solar']], solar[['ln_gdpper1']], alpha=0.4)
plt.show()
```



로그로 변환한 데이터에서 변수간의 관계가 보다 뚜렷이 드러난다.

모델선택2_로그선형vs앙상블

<선형>

```
sol = LinearRegression()  
sol.fit(x_train, y_train)
```

```
LinearRegression()
```

```
sol.score(x_test, y_test)
```

```
0.7348623084646657
```

<random forest>

```
model = RandomForestRegressor()  
model.fit(X_train, y_train)
```

```
print("model score(결정 계수) : ", model.score(X_test, y_test) )
```

```
model score(결정 계수) : 0.41487425437634673
```

<gradient boosting>

```
model = GradientBoostingRegressor()  
model.fit(X_train, y_train)
```

```
print("model score(결정 계수) : ", model.score(X_test, y_test) )
```

```
model score(결정 계수) : 0.5654168600461784
```

앙상블_random forest

```
model = RandomForestRegressor()
model.fit(X_train, y_train)

print("model score(결정 계수) : ", model.score(X_test, y_test) )
```

model score(결정 계수) : 0.41487425437634673

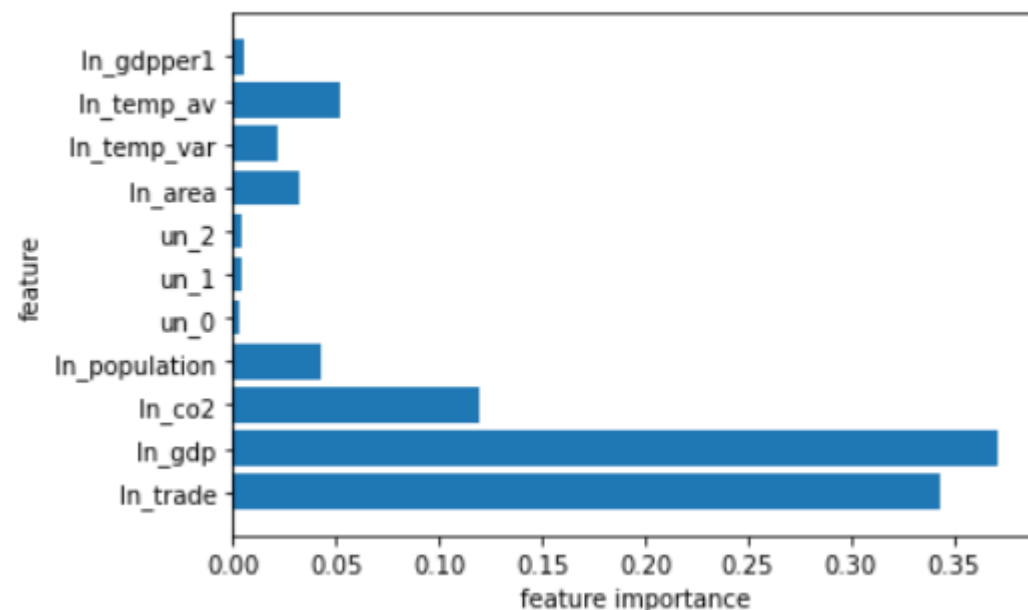
```
print("피쳐 이름 : ", sel)
```

피쳐 이름 : ['ln_trade', 'ln_gdp', 'ln_co2', 'ln_population', 'un_0', 'un_1', 'un_2', 'ln_area', 'ln_temp_var', 'ln_temp_av', 'ln_gdpper1']

```
model.feature_importances_
```

```
array([0.34346309, 0.37147435, 0.11979227, 0.04279478, 0.00297895,
       0.00357071, 0.00462847, 0.0325149 , 0.02123677, 0.05177702,
       0.0057687 ])
```

```
plot_feature_important_common(model, X_train, sel)
```



앙상블_gradient boosting

```
model = GradientBoostingRegressor()
model.fit(X_train, y_train)

print("model score(결정 계수) : ", model.score(X_test, y_test) )

model.score(결정 계수) : 0.5654168600461784
```

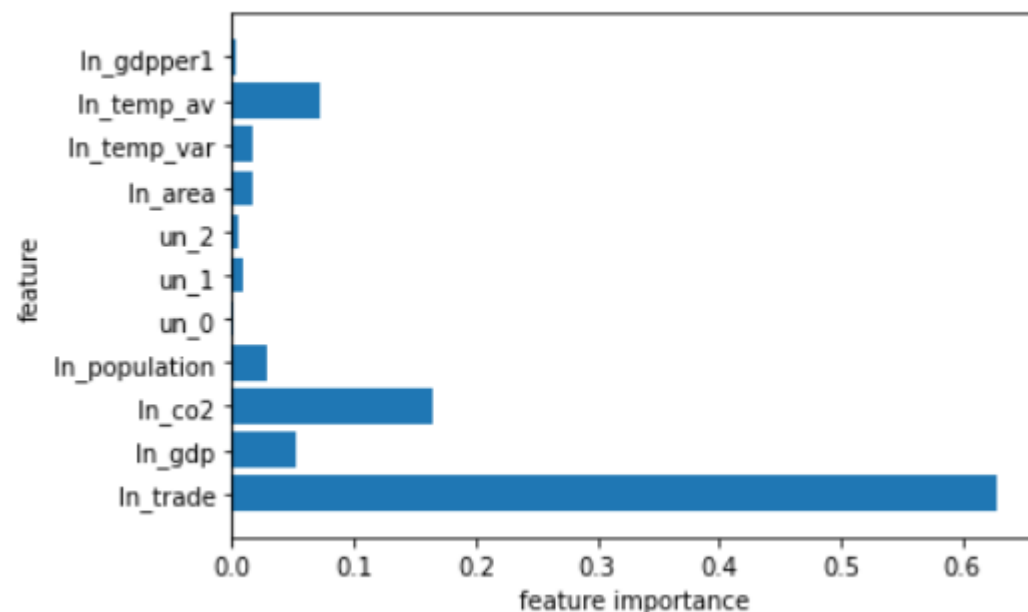
```
print("피쳐 이름 : ", sel)
```

```
피쳐 이름 : ['ln_trade', 'ln_gdp', 'ln_co2', 'ln_population', 'un_0', 'un_1', 'un_2', 'ln_area', 'ln_temp_var', 'ln_temp_av', 'ln_gdp_per1']
```

```
model.feature_importances_
```

```
array([6.28685053e-01, 5.16039148e-02, 1.65425891e-01, 2.94817580e-02,
       1.14523240e-04, 8.55099072e-03, 4.96539728e-03, 1.76377202e-02,
       1.76704604e-02, 7.22179427e-02, 3.64634831e-03])
```

```
plot_feature_important_common(model, X_train, sel)
```



최종모델

```
result = sm.ols(formula= 'ln_solar~ ln_trade + ln_gdp + ln_co2+ ln_population +ln_temp_av + ln_area +
                        un_0 + un_1 + un_2', data = solar).fit()
```

```
result.summary()
```

	coef	std err	t	P> t	[0.025	0.975]	Dep. Variable:	ln_solar	R-squared:	0.809
Intercept	1.9824	2.109	0.940	0.354	-2.296	6.260	Model:	OLS	Adj. R-squared:	0.766
ln_trade	0.4708	0.291	1.619	0.114	-0.119	1.061	Method:	Least Squares	F-statistic:	19.05
ln_gdp	-0.3260	0.335	-0.974	0.336	-1.005	0.353	Date:	Tue, 09 Nov 2021	Prob (F-statistic):	8.43e-11
ln_co2	0.8810	0.315	2.797	0.008	0.242	1.520	Time:	14:10:18	Log-Likelihood:	-7.3024
ln_population	0.1001	0.250	0.401	0.691	-0.407	0.607	No. Observations:	45	AIC:	32.60
ln_temp_av	0.0008	0.000	2.703	0.010	0.000	0.001	Df Residuals:	36	BIC:	48.86
ln_area	-0.1364	0.120	-1.137	0.263	-0.380	0.107	Df Model:	8		
un_0	0.4735	0.699	0.678	0.502	-0.944	1.891	Covariance Type:	nonrobust		
un_1	0.5698	0.659	0.864	0.393	-0.767	1.907				
un_2	0.9391	0.769	1.222	0.230	-0.620	2.498				

$$\ln_solar = 1.9824 + 0.4708 \ln_trade - 0.326 \ln_gdp + 0.8810 \ln_co2 + 0.1001 \ln_population + 0.008 \ln_temp_av - 0.1364 \ln_area + 0.4735 un_0 + 0.5698 un_1 + 0.9391 un_2 + e$$

최종모델 해석

```
result = sm.ols(formula= 'ln_solar~ ln_trade + ln_gdp + ln_co2+ ln_population +ln_temp_av + ln_area +
                        un_0 + un_1 + un_2', data = solar).fit()
```

```
result.summary()
```

	coef	
Intercept	1.9824	해석(로그 기준)
ln_trade	0.4708	무역량은 0.4708만큼 비례한다
ln_gdp	-0.3260	Gdp는 0.3260만큼 반비례한다
ln_co2	0.8810	Co2는 0.8810만큼 비례하여 가장 큰 영향을 미치는 요인이다.
ln_population	0.1001	인구는 0.1001만큼 비례하며 영향이 작다
ln_temp_av	0.0008	연평균기온은 0.0008만큼 비례하며 영향이 작다
ln_area	-0.1364	국토면적은 0.1364만큼 반비례하며 영향이 작다
un_0	0.4735	국제협력이 낮은 국가는 태양열발전이 적다
un_1	0.5698	중간 수준의 국제협력을 하는 국가는 태양열발전 또한 중간 수준이다
un_2	0.9391	선진국은 태양열발전에 더욱 힘쓴다.

국가별 태양열 발전 예측값과 비교

```
south_korea = [[5.960994, 12.17026, 2.786041, 7.709485, 1,0,0, 4.001734, 1.056905]]
my_predict_log = sol.predict(south_korea)
my_predict_log
```

	예측값(로그)	실제값(로그/실제)	발전량 과대/과소
한국	3.77071589	4.021396 (10505)	과대
미국	4.99287721	4.770292 (58924)	과소
독일	4.49658147	4.690595 (49045)	과대
프랑스	4.00252587	4.033223 (10795)	과대

결론 및 한계

< 결론 >

$$\ln_solar = 1.9824 + 0.4708 \ln_trade - 0.326 \ln_gdp + 0.8810 \ln_co2 + 0.1001 \ln_population \\ + 0.008 \ln_temp_av - 0.1364 \ln_area + 0.4735 un_0 + 0.5698 un_1 + 0.9391 un_2 + e$$

한국, 프랑스, 독일은 더 많은 양의 태양열 발전을 했고, 미국은 적은 양의 발전을 했다.

< 한계 >

통계적 해석이 미비하여 모형이 충분히 정교하지 못하다.

변수를 추가하면서 변수에 해당하는 국가를 제외하다 보니 자료의 크기가 줄어들었다.

감사합니다