Information Visualization
Final Project

Solar Power Visualization

Seonghyeon Cho

Data Abstraction

- This data has been gathered at two solar power plants in India over a 34 day period. It has two pairs of files each pair has one power generation dataset and one sensor readings dataset. The power generation datasets are gathered at the inverter level each inverter has multiple lines of solar panels attached to it. The sensor data is gathered at a plant level single array of sensors optimally placed at the plant.
- https://www.kaggle.com/datasets/anikannal/solar-power-generation-data?select=Plant_1_Weather_Sensor_Data.csv&sort=votes

Data Abstraction

- 데이터 크기
  - 68778 rows, 10 attributes 5MB file
- 관심있는 attribute
  - DATE_TIME
  - DC_POWER
  - AC_POWER
  - DAILY_YIELD
  - AMBIENT_TEMPERATURE
  - MODULE_TEMPERATURE

Attribute	Attribute Type	Description
DATE_TIME	Time	Date and time for each observation. Observations recorded at 15 minute intervals.
PLANT_ID	String	Plant ID - this will be common for the entire file.
SOURCE_KEY	String	Source key in this file stands for the inverter id.
DC_POWER	Number	Amount of DC power generated by the inverter (source_key) in this 15 minute interval. Units - kW.
AC_POWER	Number	Amount of AC power generated by the inverter (source_key) in this 15 minute interval. Units - kW.
DAILY_YIELD	Number	Daily yield is a cumulative sum of power generated on that day, till that point in time.
TOTAL_YIELD	Number	This is the total yield for the inverter till that point in time.
AMBIENT_TEMPERATURE	Number	This is the ambient temperature at the plant.
MODULE_TEMPERATURE	Number	There's a module (solar panel) attached to the sensor panel. This is the temperature reading for that module.
IRRADIATION	Number	Amount of irradiation for the 15 minute interval.

Data Cleaning, Data Preprocessing

- 데이터가 inverter_data.csv와 sensor_data.csv로 나누어져 있어서 merge
- Inverter개수가 많아서 같은 시간대(15분 간격)끼리 합해서 하나의 row로 요약
- 가끔씩 데이터의 특정 cell이 비어있거나 특정 시간대에 값이 없는 경우가 있어 주변 값을 사용해 interpolate
- 이상치나 date format이 다른 문제가 있어 수정
- Pandas 사용

information-visualization-final-project > data.py > ...

You, yesterday | 1 author (You)

```python
import pandas as pd
data1 = pd.read_csv('data1.csv')
data2 = pd.read_csv('data2.csv')

data1 = data1[['DATE_TIME', 'DC_POWER', 'AC_POWER', 'DAILY_YIELD', 'TOTAL_YIELD']]
data2['DATE_TIME'] = data2['DATE_TIME'].apply(lambda x: x[8:10] + '-' + x[5:7] + '-' + x[0:4] + ' ' + x[11:16])

data = {}

for i in range(len(data1)):
  data[data1.iloc[i]['DATE_TIME']] = {'DC_POWER': 0, 'AC_POWER': 0, 'DAILY_YIELD': 0}

dc, ac = 0, 0
for i in range(len(data1)):
  if i > 0 and data1.iloc[i - 1]['DATE_TIME'][:10] != data1.iloc[i]['DATE_TIME'][:10]:
    dc, ac = 0, 0
  data[data1.iloc[i]['DATE_TIME']]['DC_POWER'] += data1.iloc[i]['DC_POWER']
  data[data1.iloc[i]['DATE_TIME']]['AC_POWER'] += data1.iloc[i]['AC_POWER']
  dc += data1.iloc[i]['DC_POWER']
  ac += data1.iloc[i]['AC_POWER']
  data[data1.iloc[i]['DATE_TIME']]['ACCU_DC_POWER'] = dc
  data[data1.iloc[i]['DATE_TIME']]['ACCU_AC_POWER'] = ac
  data[data1.iloc[i]['DATE_TIME']]['DAILY_YIELD'] += data1.iloc[i]['DAILY_YIELD']
fo (variable) data: dict ):
                    TIME'] not in data: continue
  data[data2.iloc[i]['DATE_TIME']]['AMBIENT_TEMPERATURE'] = data2.iloc[i]['AMBIENT_TEMPERATURE']
  data[data2.iloc[i]['DATE_TIME']]['MODULE_TEMPERATURE'] = data2.iloc[i]['MODULE_TEMPERATURE']
  data[data2.iloc[i]['DATE_TIME']]['IRRADIATION'] = data2.iloc[i]['IRRADIATION']

df = pd.DataFrame(data).transpose()

df.to_csv('data.csv')

print(df)      You, yesterday • initial commit
```

2. Task Abstraction

- 1. DATE_TIME(day) - DC_POWER, AC_POWER
  - **Why**
    - They can know relation between date_time and generated power. So they can predict expected electricity production, especially in terms of date.
  - **What**
    - The relationship between date and electricity production.
  - **action + target**
    - Discover correlation between date time and electricity production.

2. Task Abstraction

- 2. DATE_TIME(hour) – DC_POWER, AC_POWER
  - **Why**
    - They can know relation between date_time and generated power. So they can predict expected electricity production , especially in terms of hour.
  - **What**
    - The relationship between hour and electricity production.
  - **action + target**
    - Discover correlation between hour time and electricity production.

2. Task Abstraction

- 3. DC_POWER, AC_POWER – AMBIENT_TEMPERATURE
  - **Why**
    - They can predict electricity production using temperature. So they can predict electricity production more precisely.
  - **What**
    - The relationship between temperature and electricity production.
  - **action + target**
    - Discover correlation between temperature and electricity production.

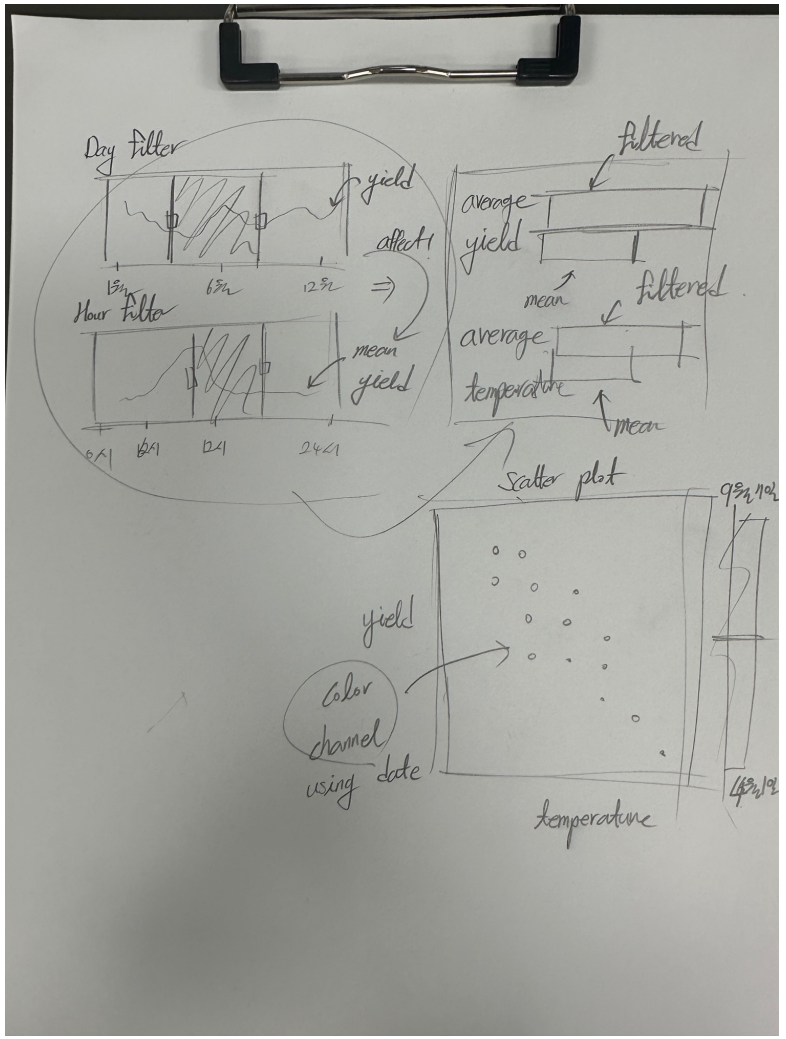
3. Creating Personas

- rahul, solar panel management, 42 years old, India

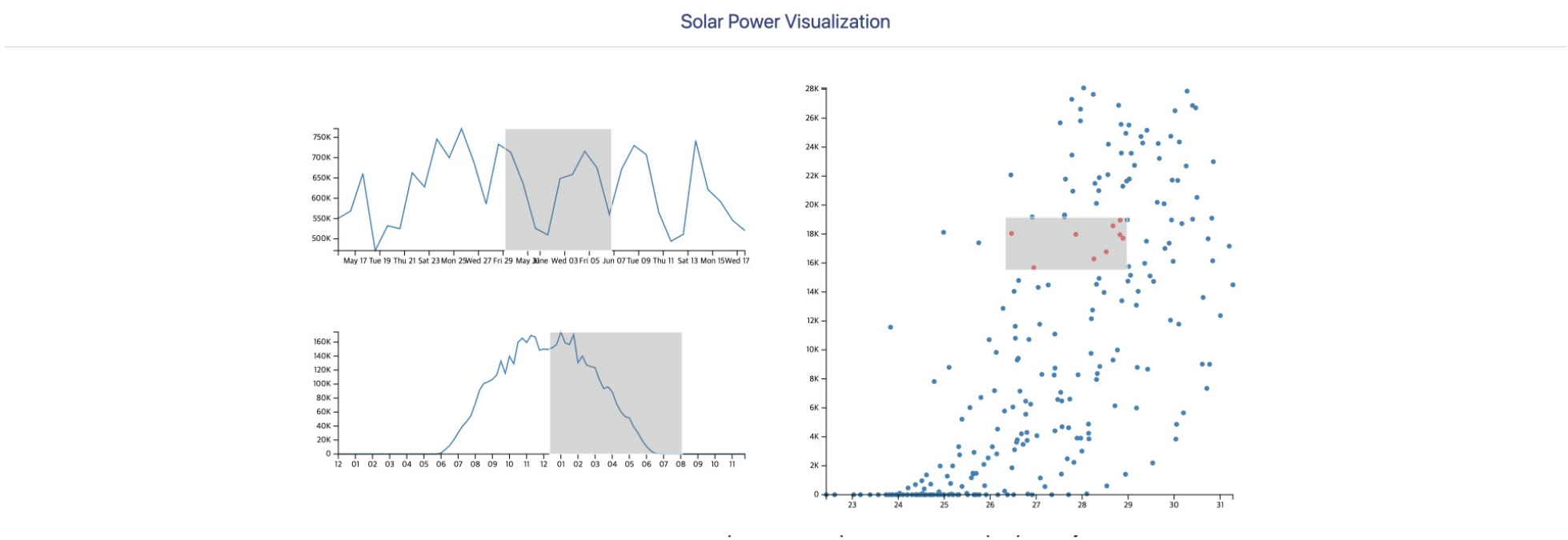
- 성현, elementary school student, 11 years old, Korea

4. Vis Idiom Design

- Day filter using brush
- Hour filter using brush
- Temperature, yield linking



5. Implementation using D3.js



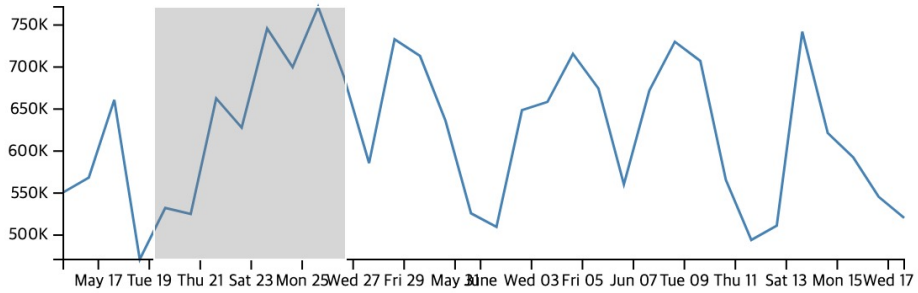
Sat May 30 2020 13:00:00	28.83	53.63	18959.49	80019.05
Tue Jun 02 2020 14:15:00	26.95	39.81	15683.84	123298.91
Wed Jun 03 2020 13:45:00	27.86	46.63	17974.77	109848.67
Thu Jun 04 2020 15:15:00	28.82	47.46	17951.40	146938.09
Thu Jun 04 2020 15:30:00	28.67	44.88	18567.24	151423.96
Thu Jun 04 2020 15:45:00	28.89	46.51	17704.25	156033.34
Thu Jun 04 2020 16:00:00	28.52	44.30	16769.15	160275.29
Sat Jun 06 2020 13:30:00	26.46	30.74	18039.32	103280.23
Sat Jun 06 2020 14:00:00	28.26	43.64	16278.77	113311.21

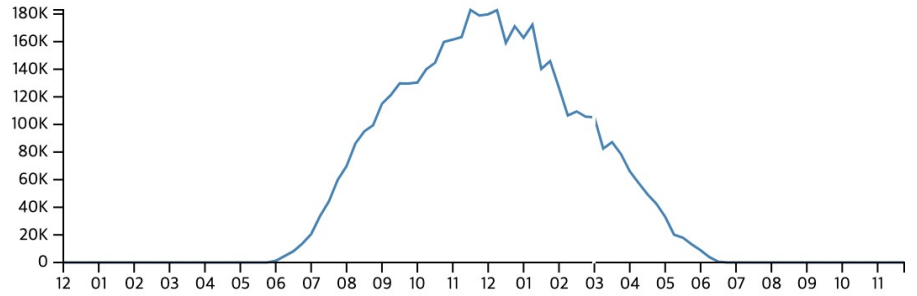
5. Implementation using D3.js

- 깃허브 페이지를 사용하여 배포
- Repo: https://github.com/st42597/information-visualization-final-project
- Deploy: https://st42597.github.io/information-visualization-final-project/

6. Case Study

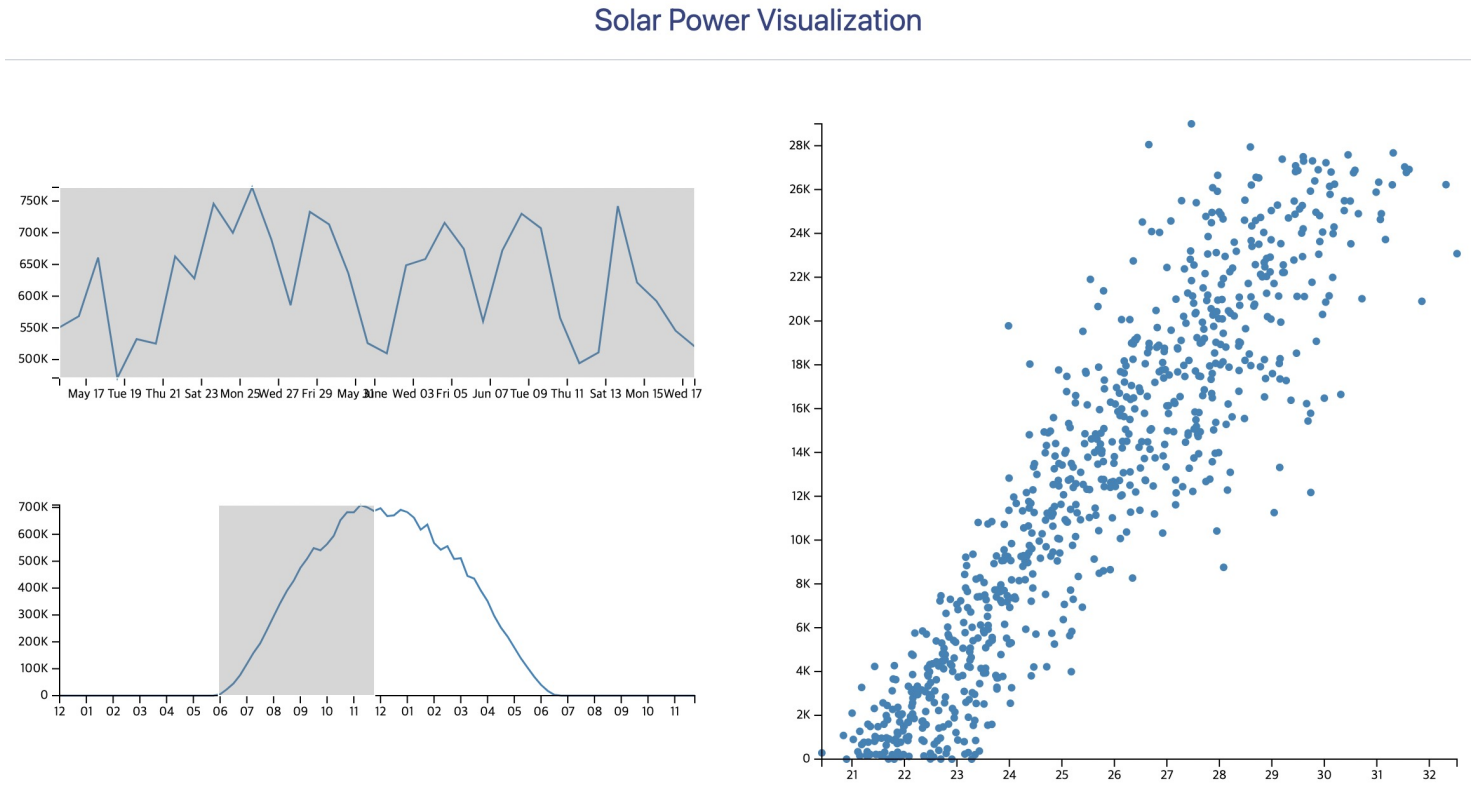
- Day filter를 조금씩 움직이면서 밑의 hour-yield 차트의 모양 변화를 확인해보면 데이터의 34일 동안은 대체로 오전 6시부터 전기가 생산되기 시작하여 오후 6시 반까지 전기가 생산되며 12시 부근에서 평균적으로 가장 높은 전기 생산량을 보인다.





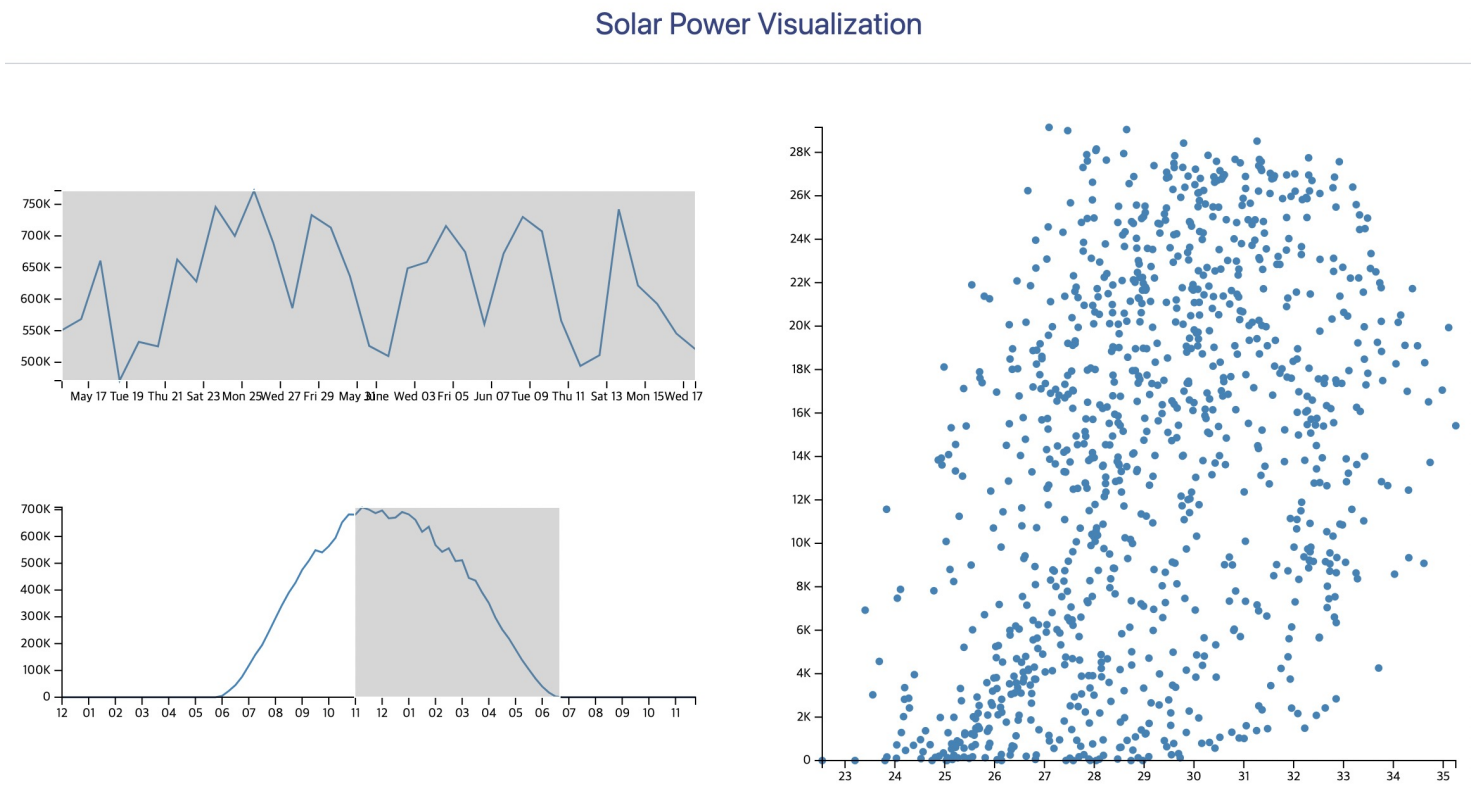
6. Case Study

- 해가 뜨는 시점의 산점도를 확인해보면 온도와 전기 생산량의 상관관계를 확인할 수 있다.



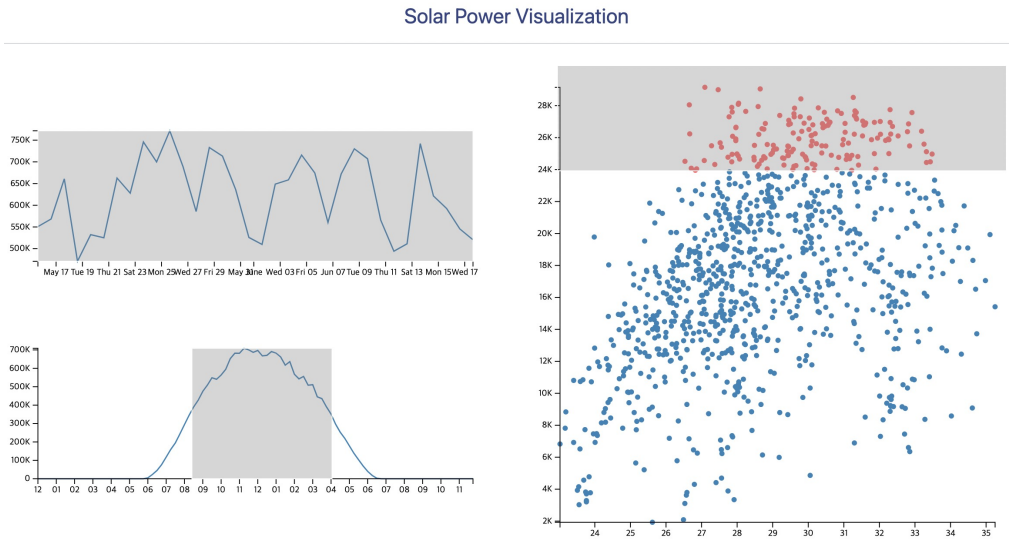
6. Case Study

- 해가 높이 뜨고 해가 지는 시점에서는 온도와 전기 생산량간의 큰 상관관계는 확인할 수 없다.



6. Case Study

- 전체 날짜 중에서 전기 생산이 가장 많이 되는 15분간의 인스턴스를 scatterplot으로 그리고 상위에 있는 인스턴스를 table을 사용하여 분석해봤다. 그 결과 DAILY_YIELD가 적은 날에도 15분간의 순간 전기 생산량은 많을 수 있다는 것을 봐서 전기 생산량은 하루 동안 일정한 경향을 보이지 않고 매우 빠르게 변화한다는 사실을 알 수 있다.



Sat May 23 2020 12:15:00	31.27	63.34	28516.90	94647.71
Sat May 23 2020 12:30:00	31.13	65.55	25656.23	101514.07
Sat May 23 2020 13:15:00	32.27	58.09	24999.86	118450.09
Sat May 23 2020 13:30:00	32.17	59.10	25824.09	124645.52
Sat May 23 2020 13:45:00	32.82	62.58	26860.41	131243.41
Sun May 24 2020 10:45:00	29.53	52.17	25115.52	51363.09
Sun May 24 2020 11:15:00	30.17	54.68	23987.65	63187.57
Sun May 24 2020 11:30:00	30.65	58.19	24898.63	69591.18
Sun May 24 2020 11:45:00	31.09	57.58	24897.83	76075.66