



Busan science high school

2023 Ocean ICT Festival

2023 BOIF

C
09

QR 코드 영역
QR 삽입 후
테두리 삭제

Youtube 영상 QR

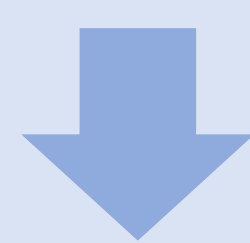
최적의 염분차 발전 장치 위치 선정

염파동: 2301 서정인, 2306 조수민

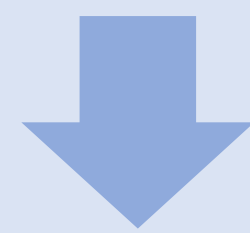
작품 개요

작품 목적

염분차 발전 방식은 지역적 제한이 없는 해안 에너지로 유망한 신재생 에너지 방식이지만 이론을 실제 설비로 발전해내는 데에는 아직 생산량에 한계가 있어 상용화되지 못함



바닷물과 민물의 염도 차를 분석하여 최적의 염분차 발전 방식을 선정하는 프로그램 제작



염분차 발전 방식의 상용화에 도움

이론적 배경

염분차 발전 방식: 바닷물과 민물 사이의 염도 차이를 이용한 발전 방식

- 역삼투 기술을 이용하여 바닷물과 민물 사이에 있는 얇은 막을 통과하는 염이 이온화되면서 전기적인 에너지가 생기게 되고, 이것이 전기발전을 작동시켜 전기 생산
- 해안 지역에서 유망한 신재생 에너지 방식 중 하나

각 지역의 염분농도, 해류, 수심, 지형 등의 요소들을 파악 + 최적의 염분차 발전 장치 위치 결정

코드 소개

1. 연안별 수온, 염분, 유속을 고려한 최적의 염분차 발전 장소 선정 코드

```
import pandas as pd  
filename='2012~2022연안데이터.csv'  
data=pd.read_csv(filename)  
data.head(49)
```

```
filename='2012~2022수온데이터.csv'  
data=pd.read_csv(filename)  
data.head(49)
```

```
filename='2020월별유속데이터.csv'  
data=pd.read_csv(filename)  
data.head(28)
```

49개 연안의 2012~2022년별 수온, 염분 데이터 수집(KOSIS사용) + 27개 연안의 2022년 월별 유속 데이터 수집(KHOA사용)

```
total_values_per_row = data.iloc[:, 2:].sum(axis=1) / 12  
sorted_indices = total_values_per_row.sort_values(ascending=False).index  
for idx in sorted_indices:  
    print(f"지역: {data.iloc[idx, 0]}, 염분Sal: {total_values_per_row[idx]}")
```

```
total_values_per_row = data.iloc[:, 2:].sum(axis=1) / 12  
sorted_indices = total_values_per_row.sort_values(ascending=False).index  
for idx in sorted_indices:  
    print(f"지역: {data.iloc[idx, 0]}, 수온Temp. (°C): {total_values_per_row[idx]}")
```

각 연안별 데이터들의 평균을 구하여 염분차 발전 방식의 효율이 높은 순으로 나열

```
row_sums = data.iloc[:, 2:].sum(axis=1)  
valid_columns = data.iloc[:, 1:].count(axis=1)  
row_averages = row_sums / valid_columns  
sorted_indices = row_averages.sort_values(ascending=False).index  
for idx in sorted_indices:  
    print(f"지역: {data.iloc[idx, 0]}, 유속: {row_averages[idx]}")
```

2. 우리나라 지도에 연안별 수온, 염분, 유속 데이터를 표시하는 코드

76개 연안의 위도, 경도 데이터 수집

```
import pandas as pd  
data = pd.read_csv("/content/drive/MyDrive/OCEAN ICT/수온염분위도경도.csv")  
ocean = data[["연안구분명", "위도", "경도", "염분Sal", "수온Temp. (°C)"]]
```

```
import pandas as pd  
data2 = pd.read_csv("/content/drive/MyDrive/OCEAN ICT/유속위도경도.csv")  
ocean2 = data2[["연안구분명", "위도", "경도", "유속"]]
```

```
import folium  
map1 = folium.Map(location=[36, 127], zoom_start=7)  
for i in ocean.index:  
    name = ocean.loc[i, '연안구분명']  
    lat = ocean.loc[i, '위도']  
    lon = ocean.loc[i, '경도']  
    if ocean['염분Sal'][i] < 25.5:  
        code_color = 'gray'  
    elif ocean['염분Sal'][i] >= 25.5 and ocean['염분Sal'][i] < 26:  
        code_color = 'orange'  
    elif ocean['염분Sal'][i] >= 26 and ocean['염분Sal'][i] < 26.5:  
        code_color = 'pink'  
    elif ocean['염분Sal'][i] >= 26.5 and ocean['염분Sal'][i] < 27:  
        code_color = 'green'  
    elif ocean['염분Sal'][i] >= 27 and ocean['염분Sal'][i] < 27.5:  
        code_color = 'blue'  
    elif ocean['염분Sal'][i] >= 27.5 and ocean['염분Sal'][i] < 28:  
        code_color = 'purple'  
    else: code_color = 'red'  
    marker = folium.Marker([lat, lon], popup=name, icon=folium.Icon(color=code_color)).add_to(map1)
```

염분<25.5: 회색
25.5<염분<26: 주황색
26<염분<26.5: 분홍색
26.5<염분<27: 초록색
27<염분<27.5: 파란색
27.5<염분<28: 보라색
그 외: 빨간색 으로 표시

```
map2 = folium.Map(location=[36, 127], zoom_start=7)  
for i in ocean.index:  
    name = ocean.loc[i, '연안구분명']  
    lat = ocean.loc[i, '위도']  
    lon = ocean.loc[i, '경도']  
    if ocean['수온Temp. (°C)'][i] < 12:  
        code_color = 'orange'  
    elif ocean['수온Temp. (°C)'][i] >= 12 and ocean['수온Temp. (°C)'][i] < 13:  
        code_color = 'pink'  
    elif ocean['수온Temp. (°C)'][i] >= 13 and ocean['수온Temp. (°C)'][i] < 14:  
        code_color = 'green'  
    elif ocean['수온Temp. (°C)'][i] >= 14 and ocean['수온Temp. (°C)'][i] < 15:  
        code_color = 'blue'  
    elif ocean['수온Temp. (°C)'][i] >= 15 and ocean['수온Temp. (°C)'][i] < 16:  
        code_color = 'purple'  
    else: code_color = 'red'  
    marker = folium.Marker([lat, lon], popup=name, icon=folium.Icon(color=code_color)).add_to(map2)
```

수온<12: 주황색
12<수온<13: 분홍색
13<수온<14: 초록색
14<수온<15: 파란색
15<수온<16: 보라색
그 외: 빨간색 으로 표시

```
import folium  
map3 = folium.Map(location=[36, 127], zoom_start=7)  
for i in ocean2.index:  
    name = ocean2.loc[i, '연안구분명']  
    lat = ocean2.loc[i, '위도']  
    lon = ocean2.loc[i, '경도']  
    if ocean2['유속'][i] < 20:  
        code_color = 'gray'  
    elif ocean2['유속'][i] >= 20 and ocean2['유속'][i] < 30:  
        code_color = 'orange'  
    elif ocean2['유속'][i] >= 30 and ocean2['유속'][i] < 40:  
        code_color = 'pink'  
    elif ocean2['유속'][i] >= 40 and ocean2['유속'][i] < 50:  
        code_color = 'green'  
    elif ocean2['유속'][i] >= 50 and ocean2['유속'][i] < 60:  
        code_color = 'blue'  
    elif ocean2['유속'][i] >= 60 and ocean2['유속'][i] < 70:  
        code_color = 'purple'  
    else: code_color = 'red'  
    marker = folium.Marker([lat, lon], popup=name, icon=folium.Icon(color=code_color)).add_to(map3)
```

유속<20: 회색
20<유속<30: 주황색
30<유속<40: 분홍색
40<유속<50: 초록색
50<유속<60: 파란색
60<유속<70: 보라색
그 외: 빨간색 으로 표시

결론 도출

수온을 고려한 최적의 발전 장소 상위 5개

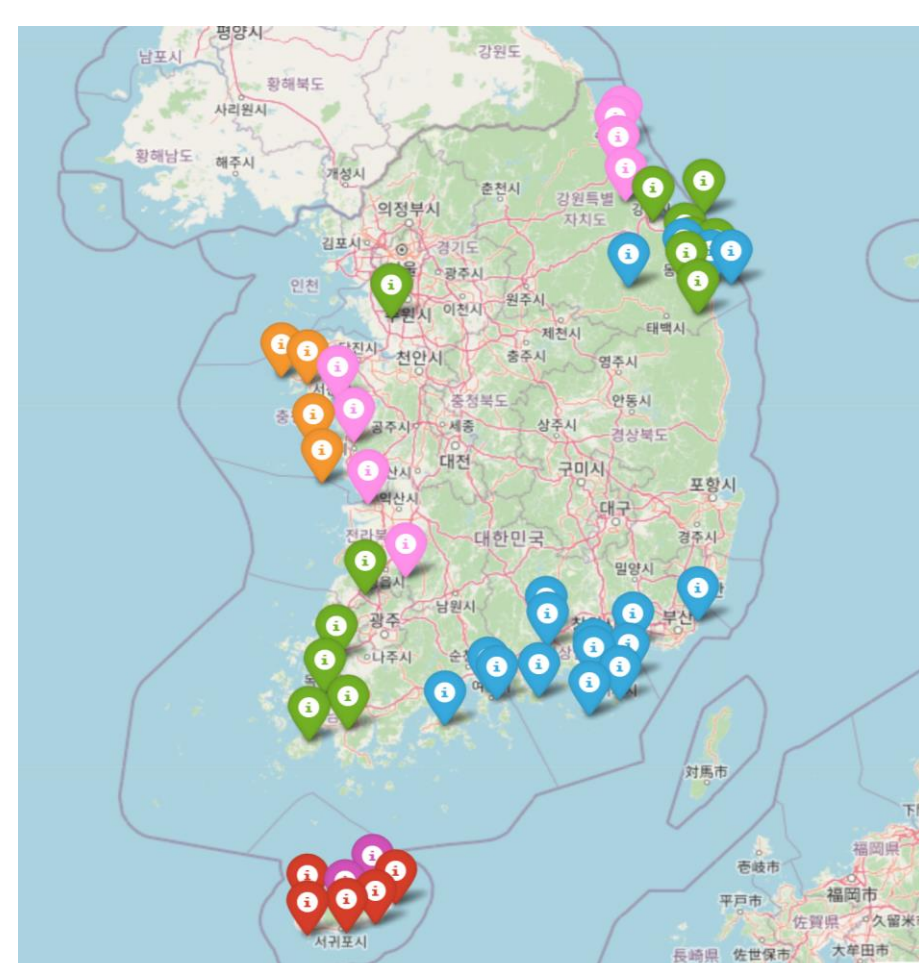
지역: 서귀포연안, 수온Temp. (°C): 16.822499999999998
지역: 표선연안, 수온Temp. (°C): 16.740000000000002
지역: 대정연안, 수온Temp. (°C): 16.414166666666667
지역: 설산연안, 수온Temp. (°C): 16.253333333333334
지역: 한림연안, 수온Temp. (°C): 16.059166666666666

염분을 고려한 최적의 발전 장소 상위 5개

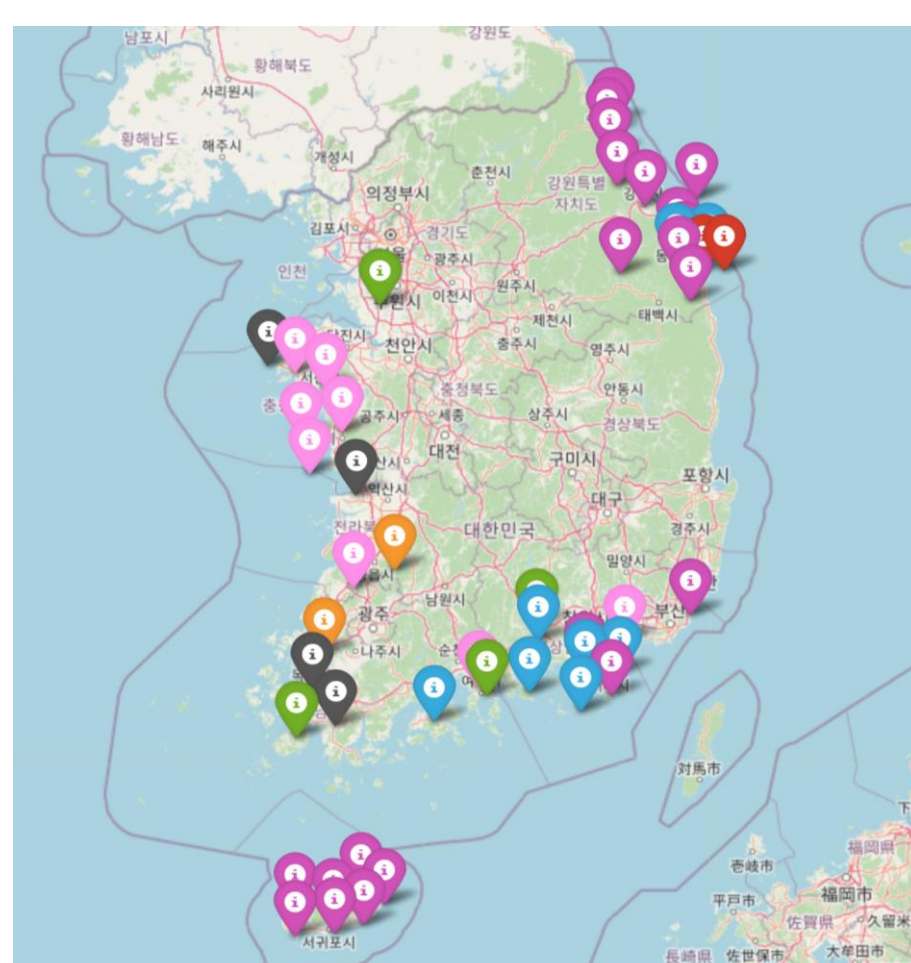
지역: 후포연안, 염분Sal: 28.06
지역: 축산연안, 염분Sal: 28.015833333333337
지역: 죽변연안, 염분Sal: 27.991666666666664
지역: 기장연안, 염분Sal: 27.944999999999997
지역: 동해연안, 염분Sal: 27.915000000000003

유속을 고려한 최적의 발전 장소 상위 5개

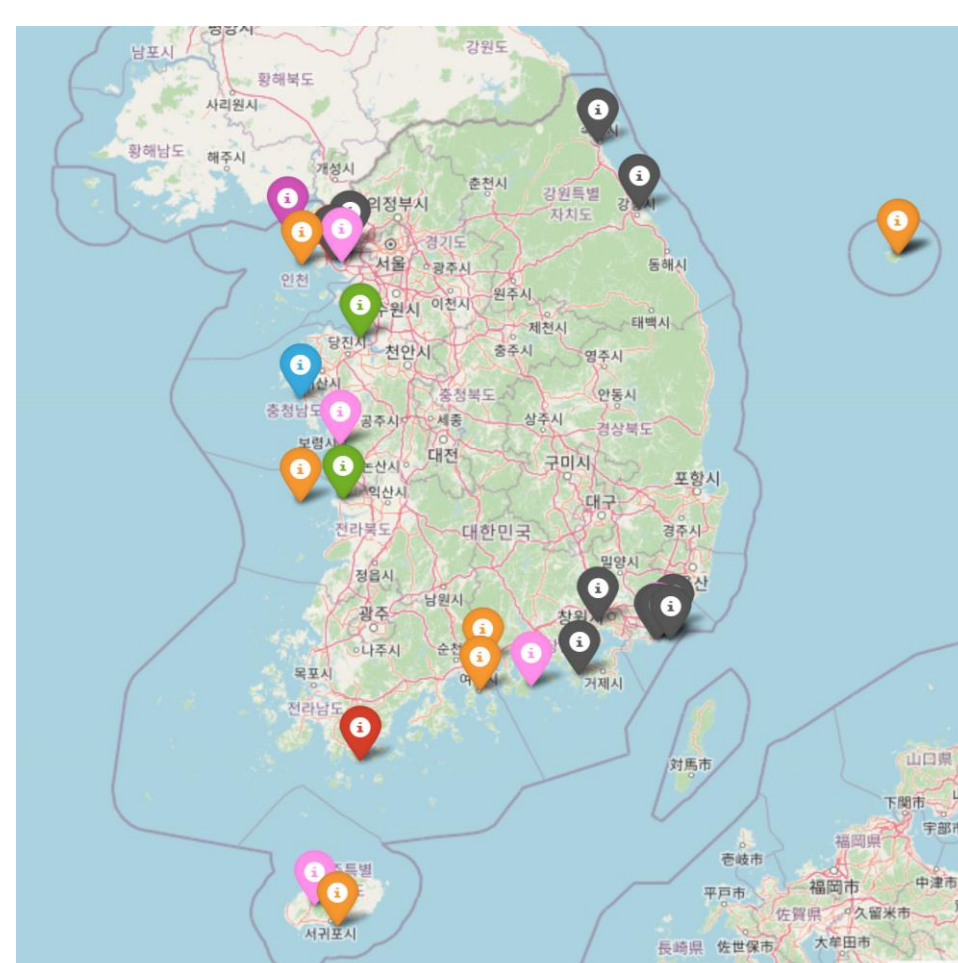
지역: 완도항, 유속: 77.44166666666666
지역: 경인항, 유속: 70.38333333333333
지역: 우이도, 유속: 60.76666666666667
지역: 태안항, 유속: 58.26363636363636
지역: 평택당진항, 유속: 43.824999999999996



수온 표시 지도



염분 표시 지도



유속 표시 지도