



Busan science high school

2023 Ocean ICT Festival

2023 BOIF

B

20

QR 코드 영역
QR 삽입 후
테두리 삭제

Youtube 영상 QR

해양 확산 시뮬레이션 프로그램 제작

2201 김가현 2203 장서현

동기

최근 일본에서 방사능 오염수를 유출함에 따라 인근 해역이 오염되고 있다. 확산 모델을 이용한 해양 확산 시뮬레이션을 통해 인근 어장에 대한 영향과 수산물 섭취 가능 여부를 보고자 한다.

화학

오염수 속 방사성 물질의 반감기와 화학적 특성을 이용하여 오염수의 확산에 따른 잔류 위험 물질의 분포를 예측한다.

생명

인근 해역에 유입된 방사능 오염물질로 인한 생물의 유전적 특성 변화 및 생태계 파괴를 고려해 수산물 섭취 가능 여부를 나타낸다.

과정

```
import math
import numpy as np
import pandas as pd
import geopandas as gpd
from fiona.crs import from_epsg # 좌표변환
import contextily as ctx
from geopy.distance import great_circle # 직선거리를 계산
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
mpl.rcParams['font', family='NanumGothic'] #한글 폰트 적용시
# 누출지점 정보
df_pt = pd.DataFrame(
    {'id': [1],
     'name': ['누출지점1'],
     'lat': [37.76],
     'lon': [140.47]})
# 누출지점 포인트 공간데이터 생성
gdf_pt_geom = gpd.points_from_xy(df_pt.lon, df_pt.lat)
gdf_pt4326 = gpd.GeoDataFrame(df_pt, geometry=gdf_pt_geom, crs=from_epsg(4326))
gdf_pt3857 = gdf_pt4326.to_crs(epsg=3857)
# 포인트 범위에 해당하는 격자 폴리곤 생성
from shapely.geometry import Polygon

# 범위
xmin = gdf_pt3857.geometry.x - 3000
xmax = gdf_pt3857.geometry.x + 3000
ymin = gdf_pt3857.geometry.y - 3000
ymax = gdf_pt3857.geometry.y + 3000
# 격자 크기
wide = 50
length = 50
cols = list(range(int(np.floor(xmin)), int(np.ceil(xmax)), wide))
rows = list(range(int(np.floor(ymin)), int(np.ceil(ymax)), length))
rows.reverse()
polygons = []
for x in cols:
    for y in rows:
        polygons.append(Polygon([(x,y),(x+wide,y),(x+wide,y-length),(x,y-length)]))
grid3857 = gpd.GeoDataFrame({'geometry': polygons})
grid3857.crs="epsg:3857"
# 격자 순번
grid3857['grid_id'] = grid3857.index + 1
grid4326 = grid3857.to_crs(epsg=4326)
grid4326['grid_lon'] = grid4326.centroid.geometry.x
grid4326['grid_lat'] = grid4326.centroid.geometry.y
ax = grid3857.plot(facecolor='none', edgecolor='gray', linewidth=1, figsize=(12, 12), alpha=0.5)
gdf_pt3857.plot(ax=ax, markersize=100, color='Red')
ctx.add_basemap(ax, source=ctx.providers.Stamen.TonerLite)
ax.set_axis_off()
plt.show()
grid4326['key'] = 1
gdf_pt4326['key'] = 1
merged = grid4326.merge(gdf_pt4326, on='key')

# 해류방향
direction = -30
direction = direction * math.pi / 180
a = 1000 #누출지점의 오염도
b = 200 #거리 척도
k = 2.5 #이심률(해류의 강도)
def calculate_distance(row):
    point1 = (row['lat'], row['lon'])
    point2 = (row['grid_lat'], row['grid_lon'])
    return great_circle(point1, point2).meters

def calculate_angle(row):
    dx = row['grid_lon'] - row['lon']
    dy = row['grid_lat'] - row['lat']
    return math.atan2(dy, dx)

merged['dist'] = merged.apply(calculate_distance, axis=1)
merged['angle'] = merged.apply(calculate_angle, axis=1)
merged['level'] = merged.apply(lambda x: a * math.exp(-(math.pow(x['dist'] / b, 2) * math.pow(math.exp(-k * math.cos(x['angle'])-dir

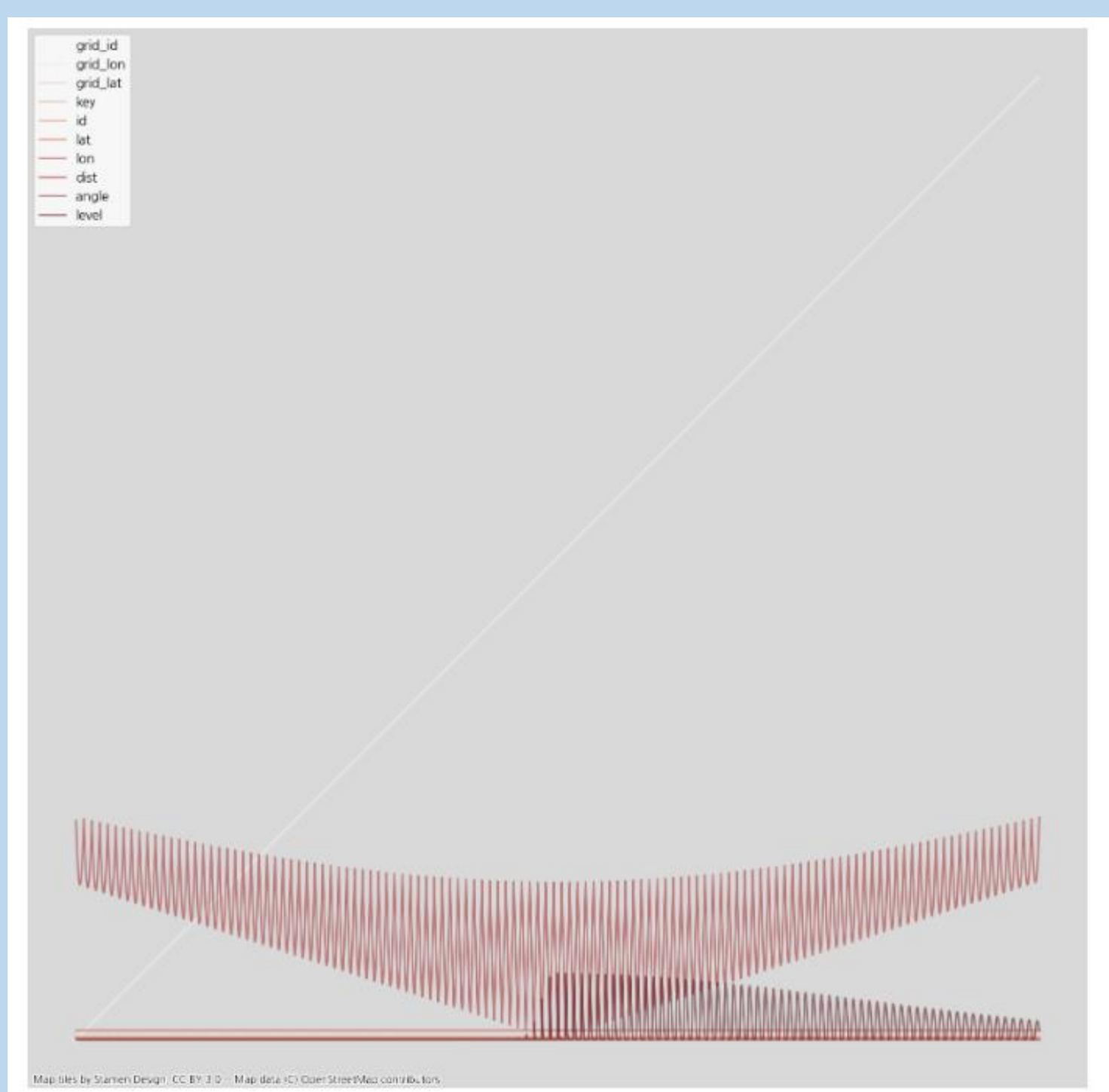
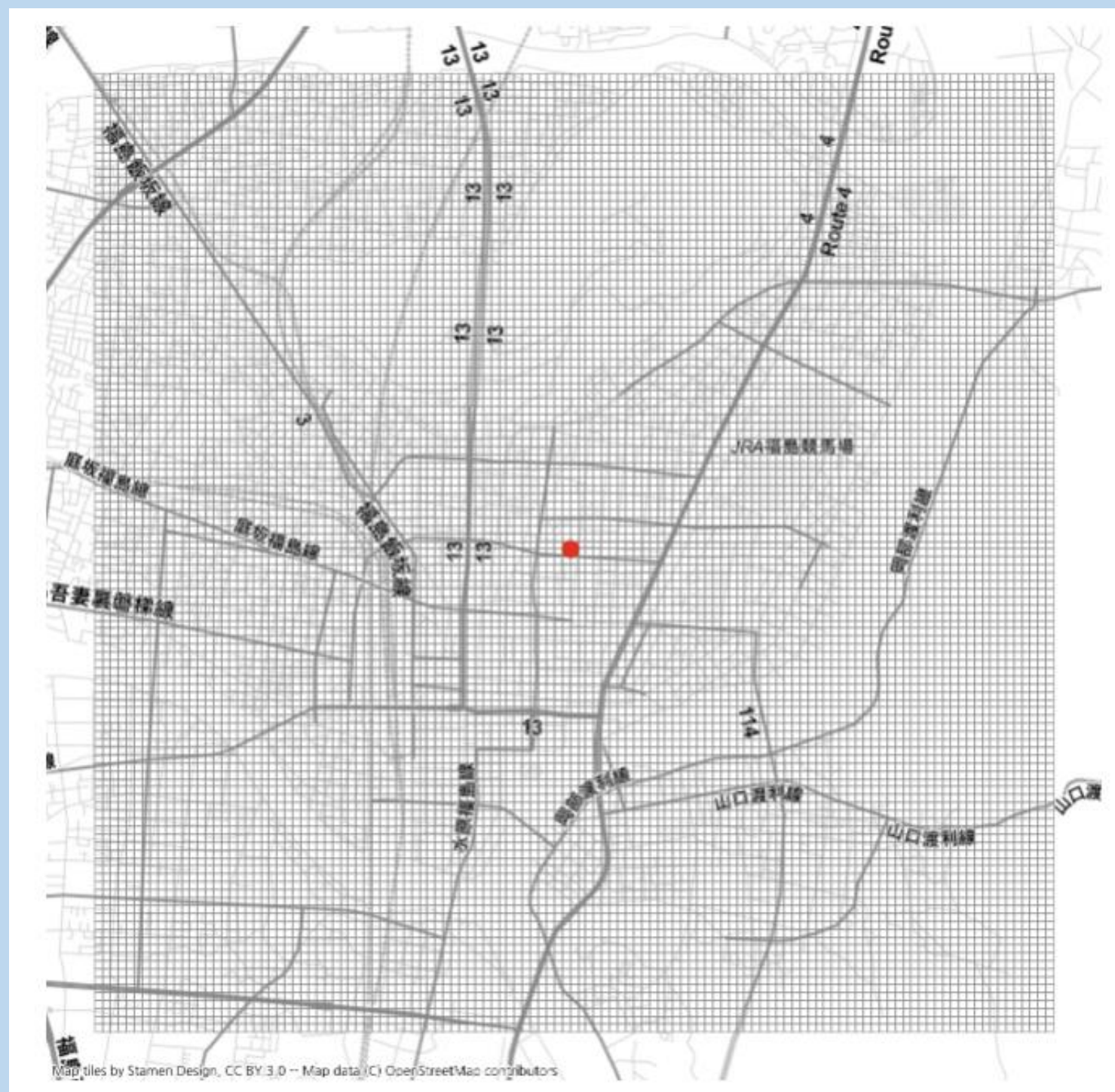
ax = merged.plot(column='level', cmap='Reds', figsize=(15, 15), alpha=0.5)
ctx.add_basemap(ax, source=ctx.providers.Stamen.TonerLite)
ax.set_axis_off()
plt.show()
```

결과

C:\Users\samsung\AppData\Local\Temp\ipykernel_67380\4023996943.py:45: UserWarning: Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this operation.

```
grid4326["grid_lon"] = grid4326.centroid.geometry.x
C:\Users\samsung\AppData\Local\Temp\ipykernel_67380\4023996943.py:45: UserWarning: Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this operation.
```

```
grid4326["grid_lat"] = grid4326.centroid.geometry.y
```



결론: 확산 시작 지점에 해당하는 위도와 경도 주변 지도가 생성되었고, 후쿠시마로 지정하였다. 지도의 정확한 지점에 누출지점이 빨간색 점으로 찍혀 있다. 지도 아래에 오염 정도, 해류방향, 유속, 거리 및 이심률에 따른 오염 정도가 빨간색 그래프로 표시되어 있음을 확인할 수 있다.

제언: 지도에 더 넓은 범위를 나타내어 바다의 영역을 더 확장한 프로그램을 코딩하면 좋을 것 같다.

기대효과

- 인근 해역에 시간이 지남에 따라 방사능 오염수가 확산되는 정도를 알 수 있다.
- 인근 어장이 방사능 오염수에 오염되는 시기 및 영향을 확인하여 시간대별 섭취가능한 정도를 나타냄으로써 안전한 식재료를 구입할 수 있다.