12am_grind

July 31, 2024

[1]:
```
import geopandas as gpd
import pandas as pd
import contextily as ctx
import pathlib
import numpy as np
import networkx
from matplotlib import pyplot as plt
from shapely.geometry import Point, Polygon, LineString
```

[3]:
```
hospitals_gdf = gpd.read_file('Hospitals_2024.geojson')
shelters_gdf = gpd.read_file('Shelters_2024.geojson')
```

[5]:
```
shelters_gdf
```

[5]:
```
      SHELTER_ID                               SHELTER_NAME  \
0         218372                        Chazy Central School
1         183805                        Ansonia High School
2         184071                        GRANBY MIDDLE SCHOOL
3         119033                     Worcester Senior Center
4          71719                     Jefferson Village School
…          …                                             …
4580      365208                          Conard High School
4581      365254                        Lewiston High School
4582      365358        Weymouth High School (Regional Center)
4583      365363                          Holy Trinity Church
4584      365382  John A. Millar Civic Center - Community Room

                ADDRESS_1           CITY COUNTY_PARISH FIPS_CODE STATE   ZIP  \
0     609 Miner Farm Road          CHAZY       CLINTON                  NY  12921
1          20 Pulaski Hwy        Ansonia     NEW HAVEN                  CT   6401
2      321 SALMON BROOK ST        GRANBY      HARTFORD                  CT  06035
3        128 PROVIDENCE ST      WORCESTER     WORCESTER                  MA  01604
4         48 Washington Rd      JEFFERSON       LINCOLN                  ME  04348
…                      …            …             …       …     …     …
4580        100 Beecwood Rd  West Hartford      Hartford      None    CT   6106
4581           156 East Ave       Lewiston  Androscoggin      None    ME   4240
4582          1 Wildcat Way       Weymouth       Norfolk      None    MA   2190
```

```
4583         1409 Park Ave      Woonsocket    Providence      None    RI    2895
4584     94 Randall Avenue         Houlton     Aroostook      None    ME    4730

     MAIL_ADDR_SAME_AS_PHYS_ADDR MAILING_ADDRESS_1  …  SCORE STATUS  \
0                            YES                    …  100.0      M
1                            YES                    …   81.0      M
2                             NO                    …   81.0      M
3                             NO                    …   81.0      M
4                             NO           Box 260  …   81.0      M
…                            …               …   …  …    …
4580                        None              None  …    NaN   None
4581                        None              None  …    NaN   None
4582                        None              None  …    NaN   None
4583                        None              None  …    NaN   None
4584                        None              None  …    NaN   None

     MATCH_TYPE LOC_NAME       GEOX       GEOY  FACILITY_TYPE  \
0             A   Street -73.433769  44.887701        SHELTER
1             A   Street -73.064238  41.329884        SHELTER
2             A   Street -72.790043  41.956001        SHELTER
3             A   Street -71.792237  42.247570        SHELTER
4             A   Street -69.432074  44.222706        SHELTER
…          …      …        …          …             …
4580       None   Street -72.752085  41.735502        SHELTER
4581       None   Street -70.202282  44.093143        SHELTER
4582       None   Street -70.942783  42.182527        SHELTER
4583       None   Street -71.516473  41.984584        SHELTER
4584       None   Street -67.829881  46.133704        SHELTER

      SUBFACILITY_CODE DATA_SOURCE_ID                    geometry
0           GENPOPSHEL            0.0   POINT (-73.43377 44.8877)
1           GENPOPSHEL            0.0  POINT (-73.06424 41.32988)
2           GENPOPSHEL            0.0    POINT (-72.79004 41.956)
3                OTHER            0.0  POINT (-71.79224 42.24757)
4           GENPOPSHEL            0.0  POINT (-69.43207 44.22271)
…                  …            …                         …
4580          EMEREVAC          101.0   POINT (-72.75208 41.7355)
4581        GENPOPSHEL          101.0  POINT (-70.20228 44.09314)
4582        GENPOPSHEL          101.0  POINT (-70.94278 42.18253)
4583        GENPOPSHEL          101.0  POINT (-71.51647 41.98458)
4584        GENPOPSHEL          101.0   POINT (-67.82988 46.1337)

[4585 rows x 73 columns]
```

[6]:
```
game_grid = gpd.read_file('https://files.bwsi-remote-sensing.net/data/
↪final_2024/game_grid_2024.geojson')
```

```
[10]: import osmnx as ox
west, south, east, north = game_grid.total_bounds
airfields = ox.geometries_from_bbox(north, south, east, west, tags={'aeroway':␣
↪'aerodrome'})
airfields = airfields.to_crs(game_grid.crs)
random_airfields = airfields.sample(5)
samp_hos = hospitals_gdf.sample(20)
```

```
/tmp/ipykernel_1946/1837452284.py:3: FutureWarning: The `geometries` module and
`geometries_from_X` functions have been renamed the `features` module and
`features_from_X` functions. Use these instead. The `geometries` module and
function names are deprecated and will be removed in the v2.0.0 release. See the
OSMnx v2 migration guide: https://github.com/gboeing/osmnx/issues/1123
  airfields = ox.geometries_from_bbox(north, south, east, west, tags={'aeroway':
'aerodrome'})
/opt/conda/lib/python3.11/site-packages/osmnx/geometries.py:48: FutureWarning:
The `north`, `south`, `east`, and `west` parameters are deprecated and will be
removed in the v2.0.0 release. Use the `bbox` parameter instead. See the OSMnx
v2 migration guide: https://github.com/gboeing/osmnx/issues/1123
  return features.features_from_bbox(north, south, east, west, tags=tags)
/opt/conda/lib/python3.11/site-packages/osmnx/_overpass.py:254: UserWarning:
This area is 162 times your configured Overpass max query area size. It will
automatically be divided up into multiple sub-queries accordingly. This may take
a long time.
  multi_poly_proj = utils_geo._consolidate_subdivide_geometry(poly_proj)
```

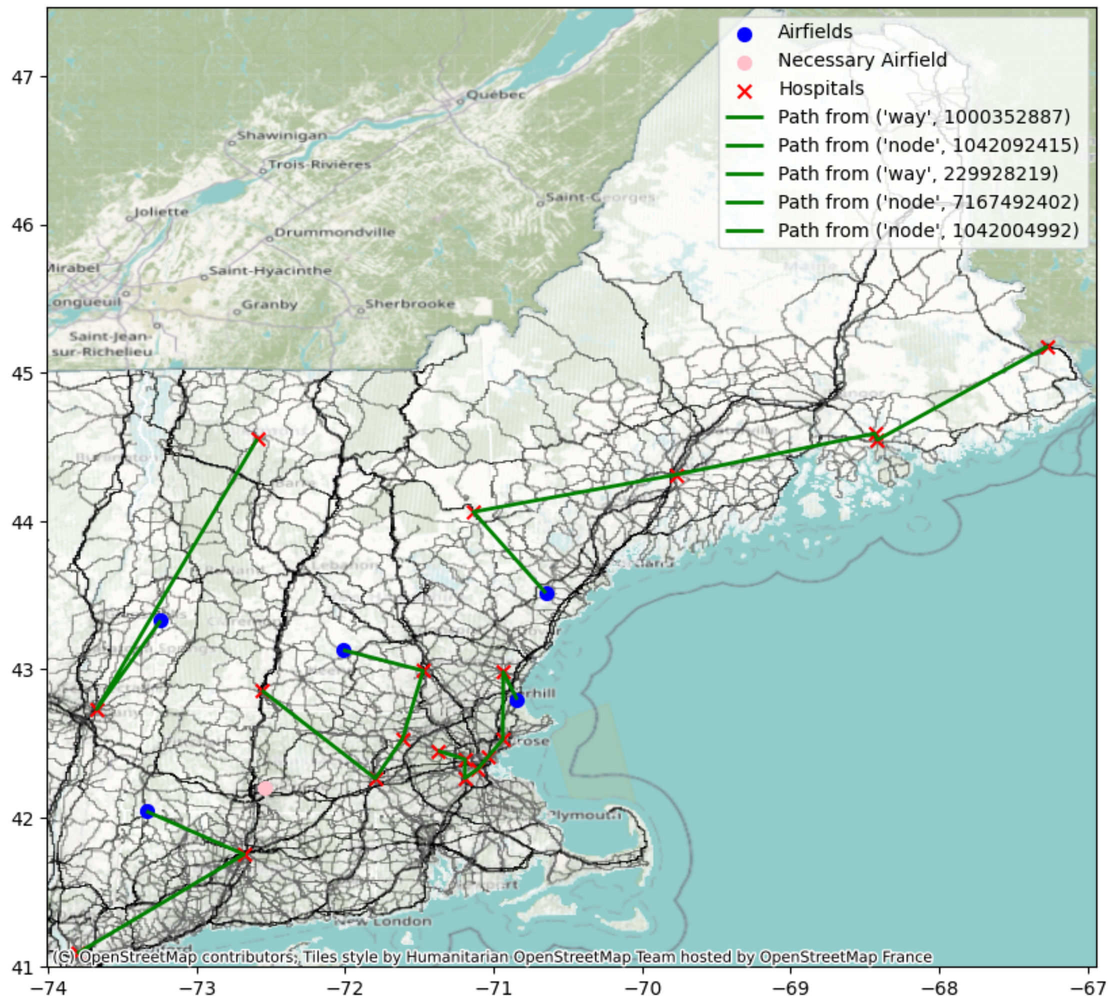
```
[12]: import osmnx as ox
import matplotlib.pyplot as plt
import geopandas as gpd
import numpy as np
from shapely.geometry import Point, LineString
import contextily as ctx

# airbase + hospital classes
class Airbase:
    def __init__(self, name, location):
        self.name = name #name right now is just the index
        self.location = location.centroid  # only works if its a centroid for␣
↪wtvr reason
        self.hospitals = []

    def add_hospital(self, hospital):
        self.hospitals.append(hospital) #for each airbase, I want to know what␣
↪hospitals it should go to

class Hospital:
    def __init__(self, name, location):
        self.name = name #the name right now is just the index
        self.location = location

    def distance_to_airbase(self, airbase):
        x1, y1 = self.location.x, self.location.y
        x2, y2 = airbase.location.x, airbase.location.y
        return np.sqrt((x2 - x1)**2 + (y2 - y1)**2)

    def distance_to(self, other_location):
        x1, y1 = self.location.x, self.location.y
        x2, y2 = other_location.x, other_location.y
        return np.sqrt((x2 - x1)**2 + (y2 - y1)**2)

def assign_hospitals_to_airbases(airbases, hospitals):
    for hospital in hospitals:
        closest_airbase = min(airbases, key=lambda airbase: hospital.
 ↪distance_to_airbase(airbase))
        closest_airbase.add_hospital(hospital)

def greedy_path(airbase):
    hospitals = airbase.hospitals[:]
    path = [airbase.location] #the brackets help it during the linestring
    current_location = airbase.location

    while hospitals:
        closest_hospital = min(hospitals, key=lambda hospital: hospital.
 ↪distance_to(current_location))
        path.append(closest_hospital.location)
        current_location = closest_hospital.location
        hospitals.remove(closest_hospital)

    return LineString(path)
# creating plot
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(1, 1, 1)

# adding transport score (HAS NOT BEEN CONSIDERED YET)
game_grid.plot(column='transport_score', cmap='Greys', alpha=0.8, ax=ax)

# just for reference (not important currently)
'''plot_route(game_grid,
           transport_network,
           [np.random.choice(game_grid['MGRS']),
            np.random.choice(game_grid['MGRS'])],
           ax=ax,
           buffer=0.01)
'''
```

```
# bounding coord boc
west, south, east, north = game_grid.total_bounds

# airfields w/in bounding box
airfields = ox.geometries_from_bbox(north, south, east, west, tags={'aeroway':␣
 ↪'aerodrome'})
airfields = airfields.to_crs(game_grid.crs)

# sample 5 random airfields and 5 random hospitals
#random_airfields = airfields.sample(5)
#samp_hos = hospitals_gdf.sample(5)

# convert airfields and sample hospitals to airbase and hospital objects [right␣
 ↪now i can only figure out how to use the index (but ideally i'd want to use␣
 ↪id)]
airbases = [Airbase(idx, row.geometry) for idx, row in random_airfields.
 ↪iterrows()]
hospitals = [Hospital(idx, row.geometry) for idx, row in samp_hos.iterrows()]

# assign hospitals to the closest airbase
assign_hospitals_to_airbases(airbases, hospitals)

# gdf for a specific point (this is meant to be the west air base, but has not␣
 ↪been implemented yet)
point_coord = Point(-72.5436, 42.1991)  # longitude, latitude
point_gdf = gpd.GeoDataFrame([1], geometry=[point_coord], crs="EPSG:4326")
point_gdf = point_gdf.to_crs(game_grid.crs)
# basemap
ctx.add_basemap(ax, crs=game_grid.crs)

# plot airfields, hospitals, and the specific point
random_airfields.centroid.plot(ax=ax, color='blue', marker='o', markersize=50,␣
 ↪label='Airfields')
point_gdf.plot(ax=ax, color='pink', marker='o', markersize=50, label='Necessary␣
 ↪Airfield')
samp_hos.plot(ax=ax, marker='x', color='red', markersize=50, label='Hospitals')

# trace paths from each airbase to its assigned hospitals
for airbase in airbases:
    if airbase.hospitals:  # check if there are any assigned hospitals (this␣
 ↪likely won't happen in the true simulation, but it fixed the error for now)
        path = greedy_path(airbase)
        gpd.GeoSeries([path]).plot(ax=ax, color='green', linewidth=2,␣
 ↪label=f'Path from {airbase.name}')

# set the axis limits to the overall bounds
```

```python
ax.set_xlim([west, east])
ax.set_ylim([south, north])

# print out which hospitals are assigned to which airbase (only index rn)
for airbase in airbases:
    print(f"{airbase.name} covers the following hospitals:")
    for hospital in airbase.hospitals:
        print(f" - {hospital.name}")

# plotting
plt.legend()
plt.show()
```

```
/tmp/ipykernel_1946/781073228.py:69: FutureWarning: The `geometries` module and
`geometries_from_X` functions have been renamed the `features` module and
`features_from_X` functions. Use these instead. The `geometries` module and
function names are deprecated and will be removed in the v2.0.0 release. See the
OSMnx v2 migration guide: https://github.com/gboeing/osmnx/issues/1123
  airfields = ox.geometries_from_bbox(north, south, east, west, tags={'aeroway':
'aerodrome'})
/opt/conda/lib/python3.11/site-packages/osmnx/geometries.py:48: FutureWarning:
The `north`, `south`, `east`, and `west` parameters are deprecated and will be
removed in the v2.0.0 release. Use the `bbox` parameter instead. See the OSMnx
v2 migration guide: https://github.com/gboeing/osmnx/issues/1123
  return features.features_from_bbox(north, south, east, west, tags=tags)
/opt/conda/lib/python3.11/site-packages/osmnx/_overpass.py:254: UserWarning:
This area is 162 times your configured Overpass max query area size. It will
automatically be divided up into multiple sub-queries accordingly. This may take
a long time.
  multi_poly_proj = utils_geo._consolidate_subdivide_geometry(poly_proj)
/tmp/ipykernel_1946/781073228.py:91: 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.

  random_airfields.centroid.plot(ax=ax, color='blue', marker='o', markersize=50,
label='Airfields')
```

```
('way', 1000352887) covers the following hospitals:
 - 140
 - 49
('node', 1042092415) covers the following hospitals:
 - 32
 - 90
('way', 229928219) covers the following hospitals:
 - 291
 - 98
 - 86
 - 226
 - 159
 - 144
 - 65
('node', 7167492402) covers the following hospitals:
 - 132
 - 301
 - 26
 - 55
 - 210
('node', 1042004992) covers the following hospitals:
 - 346
 - 330
 - 125
 - 252
```



Airfields
Necessary Airfield
Hospitals
Path from ('way', 1000352887)
Path from ('node', 1042092415)
Path from ('way', 229928219)
Path from ('node', 7167492402)
Path from ('node', 1042004992)
47
46
45
44
43
42
41
−74
−73
−72
−71
−70
−69
−68
−67
(C) OpenStreetMap contributors, Tiles style by Humanitarian OpenStreetMap Team hosted by OpenStreetMap France

```
[ ]:
```