Retrieving OpenStreetMap data for the MIKSERI project

Custom workflow for downloading POI data form OpenStreetMap from the MIKSERI-project case areas. This notebook was originally modified from automating GIS processes 2020 course materials.

Sources / References:

- © OpenStreetMap contributors 2021. (see OSM copyright and license)
- osmnx (version 1.0.1) Python package + Boeing 2017 (link to article available on research gate)
- AutoGIS 2020 materials autogis.github.io and/or Tenkanen, Heikinheimo & Whipp 2021 (soon to be) available at pythongis.org

Introduction

MIKSERI project

The MIKSERI-project (*Lähiympäristöjen kehittäminen kaupunkikudosten ja toiminnallisen sekoittuneisuuden näkökulmasta*) focuses on urban development from the perspective of functional mixes across urban fabrics. MIKSERI project description in Finnish.

OpenStreetMap

OpenStreetMap (OSM) is a free and editable worldmap available at www.openstreetmap.org. OpenStreetMap is a community effort - anyone is free to edit the map as long as they follow the commonly agreed principles (there are currently more than 4 million contributors) (stats from OSM wiki, November 2019). OSM contains data about streets, buildings, services and landuse. You can also sign up as a contributor if you want to edit the map. More details about OpenStreetMap and its contents are available in the OpenStreetMap Wiki.

There are different ways of accessing OSM data for further analysis. This notebook introduces the use of osmnx Python package that accesses the data via the OverPass API.

OSMnx

OSMnx (Boeing 2017) is a Python module that can be used to retrieve, construct, analyze, and visualize street networks from OpenStreetMap, and to retrieve data about Points of Interest (POIs) such as restaurants, schools, and lots of different kind of services. It is also easy to conduct network routing based on walking, cycling or driving by combining OSMnx functionalities with a package called NetworkX.

- Boeing, G. 2017. "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks." Computers, Environment and Urban Systems 65, 126-139. doi:10.1016/j.compenvurbsys.2017.05.004

- Interview: "Meet the developer: Introduction to OSMnx package by Geoff Boeing".

This notebook has been tested to work with osmnx version 1.0.1 and Python 3.8 in February 2021.

Python environment

The Python environment was set up following these steps:

1. Install Anaconda (or Miniconda) on your computer
2. Instructions for creating a conda environment including basic GIS tools is available on the AutoGIS course page. You can use the provided yaml file (comes with some extra packages not needed in here..) for creating the environment.

In [1]:

```
import os
import glob
import osmnx as ox
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
```

Settings / custom functions

Proxy server

In [2]:

```
# Organization-spesicif proxy settings in separate file my_proxy.py - file is ignored b
# Might be needed for fetching OSM or background maps over the network.. (Comment out i
from my_proxy import http_proxy

os.environ['http_proxy'] = http_proxy
os.environ['https_proxy'] = http_proxy
```

Output folder

In [3]:

```
# Output folder (note, contents of this folder are ignored by git!)
out_folder = r"..\results\downloaded_pois"

if not os.path.exists(out_folder):
    os.makedirs(out_folder)
    print("created", out_folder)

else:
    print(out_folder, "exists.")
```

```
..\results\downloaded_pois exists.
```

Functions for repeating the process

Function for fetching the data. Check Example section for a simple run-through of the process.

In [4]:

```
def get_pois(row, tags, cols, renamings):
    """Function for fetching OSM data for each row in a polygon geodataframe"""

    # GET THE DATA
    print("Fetchin POIs from", row["name"])
    pois = ox.geometries_from_polygon(row.geometry, tags)
    print("Got", len(pois), "objects")

    # Select columns for output file
    pois = pois[cols]

    # Rename columns
    pois = pois.rename(columns=renamings)

    return pois
```

Function for saving results per each region as a separate geopackage (easier to have separate files for further analysis steps.. would also be possible to save all in one file).

Tags are combined in the process based on the tag lists. Polygon geometries are converted to point centroids in the process.

In [5]:

```
def save_selected(pois, gpkg, tag_dict, category="amenity"):
    """Select features based on tag dictionary, save separate layer to geopackage for e

    for key in tag_dict.keys():

        # Select rows based on list of keys
        selected_pois = pois[pois[category].isin(tag_dict.get(key))].copy()

        selected_pois = selected_pois.reset_index(drop=True)

        # Add info of category to new column for potential further use..
        selected_pois["{}_reclass".format(category)] = key

        # Save selected rows to geopackage as new layer
        print("Saving", len(selected_pois), f" objects ({key})..")
        selected_pois.to_file(gpkg, layer=key, driver="GPKG")
```

In [6]:

```
def save_pois_to_gpkg(pois, gpkg, amenity_tag_dict={"":""}, shop_tag_dict={"":""}, tour
    """MIKSERI-spesific custom function for saving objects per category to geodataframe

    Converts polygon geom to point before saving, check for potential duplicates afterw

    ## PREPARE DATA FOR SAVING:

    # Save info of original geom type
    pois.loc[pois["geometry"].type=="Point", "source"] = "OSM_point"
    pois.loc[pois["geometry"].type=="Polygon", "source"] = "OSM_polygon"

    # Approximate polygon location using a point centroid
    # Calculate centroid on-the-fly using a local projection (EPSG:3067)
    poly_centroids = pois.loc[pois["geometry"].type=="Polygon"].to_crs(3067).centroid.t
    pois.loc[pois["geometry"].type=="Polygon", "geometry"] = poly_centroids

    # CUSTOM: REMOVE MATONPESUPAIKKA IF EXISTS
    if any(pois.loc[pois["name"].isin(["matonpesupaikka", "Matonpesupaikka"])]):
        pois = pois.drop(pois.loc[pois["name"].isin(["matonpesupaikka", "Matonpesupaikk
    # CUSTOM: REMOVE shop=no
    if any(pois.loc[pois["shop"].isin(["no"])]):
        pois = pois.drop(pois.loc[pois["shop"].isin(["no"])].index)

    # SAVE LAYERS PER SELECTED CATEGORIES (CUSTOM APPROACH FOR MIKSERI PROJECT PURPOSES

    # ALL SHOPS
    print("Saving", len(pois[pois["shop"].notnull()]), "objects (all shops)..")
    pois[pois["shop"].notnull()].to_file(gpkg, layer="all_shops", driver="GPKG")

    #SHOP CATEGORIES
    save_selected(pois, gpkg, shop_tag_dict, category="shop")

    # ALL AMENITIES
    print("Saving", len(pois[pois["amenity"].notnull()]), "objects (all amenities)..")
    pois[pois["amenity"].notnull()].to_file(gpkg, layer="all_amenities", driver="GPKG")

    # AMENITY CATEGORIES
    save_selected(pois, gpkg, amenity_tag_dict, category="amenity")
    #could add saving all others.

    # ALL TOURISM
    print("Saving", len(pois[pois["tourism"].notnull()]), "objects (all tourism)..")
    pois[pois["tourism"].notnull()].to_file(gpkg, layer="all_tourism", driver="GPKG")

    # TOURISM CATEGORIES
    save_selected(pois, gpkg, tourism_tag_dict, category="tourism")
    #could add saving all others.

    print("DONE saving data to", gpkg, "\n")
```

MIKSERI Case areas

Included regions:

- Joensuu
- Tampere
- Kuopio
- Lappeenranta

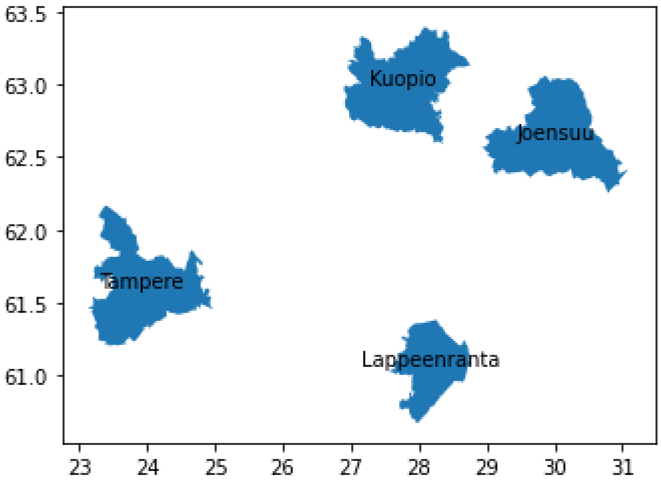
Read in case area polygons from this repository

In [7]:

```
case_areas_fp = r"..\case_areas\MIKSERI_case_areas.geojson"
case_areas = gpd.read_file(r"..\case_areas\MIKSERI_case_areas.geojson")
```

In [8]:

```
ax = case_areas.plot()

#Add names to plot to verify they went into correct polygons :D
case_areas.apply(lambda x: ax.annotate(text=x["name"],
                                       xy=x.geometry.centroid.coords[0],
                                       ha='center'),axis=1);
```



Projection

The polygons need to be in WGS84 (EPSG:4326) for osmnx.

In [9]:
```
case_areas.crs.name
```

Out[9]:
```
'WGS 84'
```

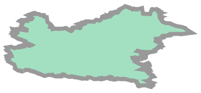
In [10]:
```
# Ensure that epsg:4326 (in the default case, this is not needed, but no harm to reproj
case_areas = case_areas.to_crs(epsg=4326)
```

In [11]:
```
case_areas.crs.name
```

Out[11]:
```
'WGS 84'
```

In [12]:
```
# Demo with one case; 0 = Lappeenranta; 3=Kuopio
case = case_areas.loc[3]
case.geometry
```

Out[12]:



In [13]:
```
case
```

Out[13]:
```
Id                                                          0
name                                                   Kuopio
geometry    POLYGON ((28.13315142426847 63.38697654903667,...
Name: 3, dtype: object
```

Example - fetch data for one case area

Point-of-interest (POI) is a generic concept that describes point locations that represent places of interest. In OpenStreetMap, many POIs are described using the amenity-tags. We can, for excample, retrieve all restaurat locations by referring to the tag `amenity=restaurant` . See all available amenity categories from OSM wiki. Various kinds of shops are tagged using shop-tags.

Here, we will retrieve all shops and amenities for further inspection.

In [14]:

```python
# List key-value pairs for tags
#tags = {'amenity': 'restaurant'}
tags = {"amenity": True, 'shop':True, 'tourism':True}

# Retrieve restaurants
pois = ox.geometries_from_polygon(case.geometry, tags)

# Check the result
len(pois)
```

Out[14]:

```
2862
```

Let's explore what kind of attributes we have in our GeoDataFrame:

In [15]:

```python
pois.head(2)
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[15]:

	unique_id	osmid	element_type	amenity	geometry	addr:city	addr:country	name	religion
0	node/34980260	34980260	node	parking	POINT (27.62741 62.88878)	NaN	NaN	NaN	NaN
1	node/34980285	34980285	node	parking	POINT (27.62989 62.88945)	NaN	NaN	NaN	NaN

2 rows × 240 columns

In [16]:

```python
# Available columns
#print(pois.columns.values)
len(pois.columns.values)
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[16]:

```
240
```

As you can see, there is quite a lot of (potential) information related to the amenities. Let's subset the columns and inspect the data further. Useful columns include at least `name` , `address` information and opening_hours information:

In [17]:

```
# Select some useful cols and print
cols = ['name', 'name:fi', 'amenity','shop','tourism','opening_hours',
        'addr:city', 'addr:country', 'addr:housenumber', 'addr:postcode', 'addr:street'
        'leisure', 'website', 'geometry']

# Print only selected cols
pois[cols].head()
```

Out[17]:

	name	name:fi	amenity	shop	tourism	opening_hours	addr:city	addr:country	add
0	NaN	NaN	parking	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	parking	NaN	NaN	NaN	NaN	NaN	
2	Männistön Pyhän Johanneksen Kirkko	NaN	place_of_worship	NaN	NaN	NaN	Kuopio	FI	
3	Puijo	NaN	NaN	NaN	attraction	NaN	Kuopio	FI	
4	Savon Pizza-Kebab	NaN	restaurant	NaN	NaN	NaN	NaN	NaN	

In [18]:

```
print(pois.amenity.value_counts().head(15))
```

```
parking             1232
restaurant           106
school                77
bench                 74
shelter               61
social_facility       53
fuel                  51
fast_food             51
kindergarten          46
place_of_worship      39
bicycle_rental        37
cafe                  37
pub                   28
post_box              27
waste_basket          26
Name: amenity, dtype: int64
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

In [19]:

```
print(pois.tourism.value_counts().head(15))
```

```
information        42
picnic_site        42
artwork            30
museum             20
viewpoint          17
hotel              17
attraction         12
chalet             11
hostel              5
guest_house         4
camp_site           4
caravan_site        3
apartment hotel     3
gallery             2
wilderness_hut      1
Name: tourism, dtype: int64
```

In [20]:

```python
print("shops:", len(pois[pois.shop.notnull()]))
print("amenities:", len(pois[pois.amenity.notnull()]))
print("tourism:", len(pois[pois.tourism.notnull()]))
```

```
shops: 402
amenities: 2261
tourism: 214
```

In [21]:

```python
pois.head()
```

Out[21]:

	unique_id	osmid	element_type	amenity	geometry	addr:city	addr:country	
0	node/34980260	34980260	node	parking	POINT (27.62741 62.88878)	NaN	NaN	
1	node/34980285	34980285	node	parking	POINT (27.62989 62.88945)	NaN	NaN	
2	node/162657407	162657407	node	place_of_worship	POINT (27.69370 62.90690)	Kuopio	FI	Mä Johar
3	node/197391072	197391072	node	NaN	POINT (27.65611 62.90950)	Kuopio	FI	
4	node/197420228	197420228	node	restaurant	POINT (27.70601 62.91659)	NaN	NaN	Savo

5 rows × 240 columns

In [22]:

```python
# Result includes also polygons, we can restrict the result based on the geometry type
#pois[pois["geometry"].type=="Point"]
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
```

```
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Handle polygons (note, this might introduce duplicates into the data!)

In [23]:

```
pois.loc[pois["geometry"].type=="Point", "source"] = "OSM_point"
pois.loc[pois["geometry"].type=="Polygon", "source"] = "OSM_polygon"

# Approximate polygon location using a point centroid
# Calculate centroid on-the-fly using a local projection (EPSG:3067)
poly_centroids = pois.loc[pois["geometry"].type=="Polygon"].to_crs(3067).centroid.to_cr
pois.loc[pois["geometry"].type=="Polygon", "geometry"] = poly_centroids
```

As we can see, there is a lot of information available. Also, if some of the information need updating, you can go over to www.openstreetmap.org and edit the source data! :)

Fetch POIs for all regions

Selected tags and columns

In [24]:

```
# Tags to fetch
tags = {'amenity': True,
        'shop':True,
        'tourism':True
       }

# List of columns to select from the result
cols = ['name', 'name:fi', 'amenity','shop','tourism','opening_hours',
        'addr:city', 'addr:country', 'addr:postcode', 'addr:street',
        'geometry','website']

# Dictionary for renaming columns. Note! for shapefile output, max 8 characters will be
renamings = {'name:fi': 'name_fi',
             'addr:city' : 'addr_city',
             'addr:country' : 'addr_country',
             'addr:postcode' : 'addr_postcode',
             'addr:street': 'addr_street'
                        }
```

In [25]:

```
#  layername Tag(s) dict for amenity classes that should get stored as separate layers
amenity_tag_dict = {"restaurants": ["restaurant", "fast_food"],
                     "cafes" : ["cafe"],
                     "pubs_clubs" : ["pub", "nightclub", "bar"],
                     "pharmacies": ["pharmacy"],
                     "libraries": ["library"],
                     "banks": ["bank"],
                     "theatres":["theatre"],
                     "cinemas": ["cinema"],
                     "kindergartens" : ["kindergarten", "childcare"]
                     # other amenities are omitted
                    }
```

In [26]:

```
#  layername Tag(s) dict for amenity classes that should get stored as separate layers
shop_tag_dict = { "alcohol_shops" : ["alcohol"],
                 "kiosks" : ["kiosk"],
                 "grocery_stores" : ["supermarket", "grocery", "convenience"],
                 "department_stores" : ["department_store", "variety_store"],
                 "bakery" : ["pastry", "bakery"],
                 "personal_services" : ["beauty", "tattoo", "massage", "hairdresser"] ,
                 "clothing" : ["clothes", "shoes"],
                 "sports_outdoors" : ["sports", "outdoors"],
                 "opticians" :["optician"],
                 "florists" : ["florist"],
                }
```

In [27]:

```
tourism_tag_dict = {"hotels":["hotel", "motel", "hostel"],
                    "museums":["museum"]
                    }
```

In [28]:

```
# Fetch amenities for all case areas (results come stacked into one Pandas Series..)
results = case_areas.apply(get_pois, tags=tags, cols=cols, renamings=renamings, axis=1)
```

```
Fetchin POIs from Lappeenranta
Got 2414 objects
Fetchin POIs from Tampere
Got 7488 objects
Fetchin POIs from Joensuu
Got 2559 objects
Fetchin POIs from Kuopio
Got 2862 objects
```

In [29]:

```
#selected_shop_tags = [value for values in shop_tag_dict.values() for value in values]
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

In [30]:

```
results[0]
```

Out[30]:

	name	name_fi	amenity	shop	tourism	opening_hours	addr_city	addr_country	a
0	NaN	Vihtolan laavu	shelter	NaN	NaN	NaN	NaN	NaN	
1	NaN	Melkkolan laavu	shelter	NaN	NaN	NaN	NaN	NaN	
2	Simolan Alakosken laavu	NaN	shelter	NaN	NaN	NaN	NaN	NaN	
3	NaN	Simolan urh.kentta laavu	shelter	NaN	NaN	NaN	NaN	NaN	

	name	name_fi	amenity	shop	tourism	opening_hours	addr_city	addr_country	a
4	NaN	Multamäen laavu	shelter	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
2409	IsoKristiina	NaN	NaN	mall	NaN	Mo-Sa 07:00-21:00; Su 10:00-21:00	Lappeenranta	NaN	
2410	NaN	NaN	fire_station	NaN	NaN	NaN	NaN	NaN	
2411	NaN	NaN	fountain	NaN	NaN	NaN	NaN	NaN	
2412	NaN	NaN	parking	NaN	NaN	NaN	NaN	NaN	
2413	NaN	NaN	parking	NaN	NaN	NaN	NaN	NaN	

2414 rows × 12 columns

In [31]:

```
# FOR EACH AREA, CREATE SEPARATE GPKG
for i in range(len(case_areas)):

    # INITIATE OUTPUT FILE AND SAVE STUDY AREA EXTENT
    area = case_areas[case_areas.index == i]
    area_name = area.at[i, "name"]
    print(area_name)

    # output file name
    gpkg = os.path.join(out_folder, "OSM_{}.gpkg".format(area_name))

    # Save study area extent
    area.to_file(gpkg, layer="{}_region".format(area_name), driver="GPKG")

    # Save points
    save_pois_to_gpkg(results[i],
                      gpkg=gpkg,
                      amenity_tag_dict=amenity_tag_dict,
                      shop_tag_dict=shop_tag_dict,
                      tourism_tag_dict=tourism_tag_dict
                     )
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
Lappeenranta
Saving 356 objects (all shops)..
Saving 3  objects (alcohol_shops)..
Saving 10  objects (kiosks)..
Saving 41  objects (grocery_stores)..
Saving 4  objects (department_stores)..
Saving 5  objects (bakery)..
Saving 56  objects (personal_services)..
Saving 33  objects (clothing)..
Saving 11  objects (sports_outdoors)..
Saving 8  objects (opticians)..
Saving 3  objects (florists)..
Saving 1887 objects (all amenities)..
Saving 103  objects (restaurants)..
Saving 34  objects (cafes)..
Saving 23  objects (pubs_clubs)..
Saving 12  objects (pharmacies)..
Saving 13  objects (libraries)..
Saving 6  objects (banks)..
Saving 2  objects (theatres)..
Saving 1  objects (cinemas)..
Saving 33  objects (kindergartens)..
Saving 184 objects (all tourism)..
Saving 12  objects (hotels)..
Saving 14  objects (museums)..
DONE saving data to ..\results\downloaded_pois\OSM_Lappeenranta.gpkg

Tampere
Saving 1072 objects (all shops)..
Saving 19  objects (alcohol_shops)..
Saving 46  objects (kiosks)..
Saving 202  objects (grocery_stores)..
Saving 26  objects (department_stores)..
Saving 19  objects (bakery)..
Saving 133  objects (personal_services)..
Saving 123  objects (clothing)..
Saving 21  objects (sports_outdoors)..
Saving 21  objects (opticians)..
Saving 16  objects (florists)..
Saving 6024 objects (all amenities)..
Saving 446  objects (restaurants)..
Saving 136  objects (cafes)..
Saving 152  objects (pubs_clubs)..
Saving 41  objects (pharmacies)..
Saving 41  objects (libraries)..
Saving 33  objects (banks)..
Saving 16  objects (theatres)..
Saving 3  objects (cinemas)..
Saving 119  objects (kindergartens)..
Saving 393 objects (all tourism)..
Saving 30  objects (hotels)..
Saving 36  objects (museums)..
DONE saving data to ..\results\downloaded_pois\OSM_Tampere.gpkg

Joensuu
Saving 356 objects (all shops)..
Saving 4  objects (alcohol_shops)..
Saving 9  objects (kiosks)..
Saving 56  objects (grocery_stores)..
Saving 12  objects (department_stores)..
Saving 10  objects (bakery)..
Saving 32  objects (personal_services)..
Saving 18  objects (clothing)..
Saving 5  objects (sports_outdoors)..
Saving 6  objects (opticians)..
Saving 6  objects (florists)..
Saving 1980 objects (all amenities)..
Saving 118  objects (restaurants)..
Saving 53  objects (cafes)..
Saving 30  objects (pubs_clubs)..
Saving 13  objects (pharmacies)..
Saving 14  objects (libraries)..
Saving 11  objects (banks)..
Saving 7  objects (theatres)..
Saving 1  objects (cinemas)..
Saving 60  objects (kindergartens)..
Saving 227 objects (all tourism)..
Saving 15  objects (hotels)..
Saving 8  objects (museums)..
DONE saving data to ..\results\downloaded_pois\OSM_Joensuu.gpkg

Kuopio
Saving 400 objects (all shops)..
Saving 10  objects (alcohol_shops)..
Saving 21  objects (kiosks)..
Saving 63  objects (grocery_stores)..
Saving 8  objects (department_stores)..
Saving 7  objects (bakery)..
Saving 33  objects (personal_services)..
Saving 23  objects (clothing)..
Saving 9  objects (sports_outdoors)..
Saving 9  objects (opticians)..
Saving 13  objects (florists)..
Saving 2258 objects (all amenities)..
Saving 157  objects (restaurants)..
Saving 37  objects (cafes)..
Saving 32  objects (pubs_clubs)..
Saving 17  objects (pharmacies)..
Saving 17  objects (libraries)..
Saving 11  objects (banks)..
Saving 7  objects (theatres)..
Saving 3  objects (cinemas)..
Saving 52  objects (kindergartens)..
Saving 214 objects (all tourism)..
Saving 22  objects (hotels)..
Saving 20  objects (museums)..
DONE saving data to ..\results\downloaded_pois\OSM_Kuopio.gpkg
```

Explore results

In [32]:

```
files = glob.glob(os.path.join(out_folder, "*gpkg"))
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

In [33]:

```
files
```

Out[33]:

```
['..\\results\\downloaded_pois\\OSM_Joensuu.gpkg',
 '..\\results\\downloaded_pois\\OSM_Kuopio.gpkg',
 '..\\results\\downloaded_pois\\OSM_Lappeenranta.gpkg',
 '..\\results\\downloaded_pois\\OSM_Tampere.gpkg']
```

In [34]:

```
results[1][results[1]["amenity"]=="childcare"]
```

Out[34]:

	name	name_fi	amenity	shop	tourism	opening_hours	addr_city	addr_country	addr_p
1222	Kurikankulman päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
1223	Kurikansiiven päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
1224	Kurikan Helmen päiväkoti	NaN	childcare	NaN	NaN	NaN	Pirkkala	NaN	
1857	Pereen päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
1963	International Early Education Center	NaN	childcare	NaN	NaN	7:15-17:00	Tampere	NaN	
2535	Linnainmaan päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
2755	Touhula	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
3808	Sammon päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
3977	Iidesrannan päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
4487	Kurikanpirtin päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5028	Kurikan Helmen päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5047	Luhtaa Day Care Center	NaN	childcare	NaN	NaN	NaN	NaN	NaN	

	name	name_fi	amenity	shop	tourism	opening_hours	addr_city	addr_country	addr_p
5087	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5088	Menkalan päiväkoti	NaN	childcare	NaN	NaN	NaN	Nokia	NaN	
5092	Viholan päiväkoti	NaN	childcare	NaN	NaN	NaN	Nokia	NaN	
5093	Kartanon päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5099	Peppi & Bobby	NaN	childcare	NaN	NaN	NaN	Nokia	NaN	
5104	Harjuniityn päiväkoti	NaN	childcare	NaN	NaN	NaN	Nokia	NaN	
5115	Touhula	NaN	childcare	NaN	NaN	Mo-Fr 06:30-17:00	Nokia	NaN	
5191	Suoraman päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5192	Pajupillin päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5308	Pellervon koulu ja päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
5376	Amurin päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
5512	Kissanmaan päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
5522	Kyöstin päiväkoti	NaN	childcare	NaN	NaN	NaN	Pirkkala	NaN	
5525	Pilke Tiedepäiväkoti Hokaus	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5764	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	

	name	name_fi	amenity	shop	tourism	opening_hours	addr_city	addr_country	addr_p
5770	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
5771	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
6087	Hippoksen päiväkoti	NaN	childcare	NaN	NaN	NaN	Tampere	NaN	
6376	Piriniityn päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
6689	Ruutanan päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
6732	Salorinteen päiväkoti	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
6734	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
6867	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
7351	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
7353	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
7391	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	
7453	NaN	NaN	childcare	NaN	NaN	NaN	NaN	NaN	

Potential duplicates

First round of quality assurance: check for duplicated POI names. Requires further visual inspection (e.g. in QGIS). For example, it is likely that there are several Hesburgers or Subways in one region, but less likely that there are several "Ninan Keittiö" in Tampere --> check manually and remove potential duplicates..

In [35]:

```
def explore_duplicates(gpkg, layer="restaurants"):
    print("\n-------", gpkg.split("_")[1][:-5], layer, "- potential duplicates---------

    data = gpd.read_file(gpkg, layer=layer)

    #counts = pd.DataFrame(data.name.value_counts(), columns=["count"])
    #print(data.name.value_counts())

    counts = pd.DataFrame(data.name.value_counts())
    if len(counts[counts["name"] > 1])>0:
        print(counts[counts["name"] > 1])

for gpkg in files:

    print("\n-----------")
    print(os.path.basename(gpkg))
    for layer in ["restaurants", "cafes", "pubs_clubs", "kindergartens"]:
        explore_duplicates(gpkg, layer)
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
-----------
OSM_Joensuu.gpkg

------- poi restaurants - potential duplicates------------
                  name
Subway               4
Hesburger            2
Kotipizza            2
Sulo                 2
Burger King          2
Pizzapojat           2
Mihailin Taverna     2

------- poi cafes - potential duplicates------------

------- poi pubs_clubs - potential duplicates------------
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
------- poi kindergartens - potential duplicates------------
                        name
Gävlenlinnan päiväkoti     2
Marjalan päiväkoti         2
Linnunlahden päiväkoti     2

-----------
OSM_Kuopio.gpkg
```

```
------- poi restaurants - potential duplicates------------
           name
Hesburger     4
Kotipizza     2
Subway        2

------- poi cafes - potential duplicates------------
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
------- poi pubs_clubs - potential duplicates------------

------- poi kindergartens - potential duplicates------------

-----------
OSM_Lappeenranta.gpkg

------- poi restaurants - potential duplicates------------
           name
Hesburger     3
Don Papa      2
Kotipizza     2
Pamukkale     2

------- poi cafes - potential duplicates------------
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
------- poi pubs_clubs - potential duplicates------------

------- poi kindergartens - potential duplicates------------

-----------
OSM_Tampere.gpkg

------- poi restaurants - potential duplicates------------
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
```

```
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
                       name
Subway                   14
Kotipizza                11
Hesburger                 7
Pancho Villa              5
McDonald's                4
Burger King               3
Zarillo                   3
Pepper                    2
Hanko Sushi               2
Pizza Service             2
Captain Hook              2
Ninan keittiö             2
Kasvisravintola Gopal     2

------- poi cafes - potential duplicates------------
                 name
Brander             2
kahvila             2
Robert's Coffee     2
Ståhlberg           2
Arnold's            2
Café Linkosuo       2

------- poi pubs_clubs - potential duplicates------------

------- poi kindergartens - potential duplicates------------
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
                          name
Nekalan päiväkoti            3
Touhula                      3
Harjunsalon päiväkoti        2
Menkalan päiväkoti           2
Tursolan päiväkoti           2
Linnainmaan päiväkoti        2
Omenatarhan päiväkoti        2
Kurikan Helmen päiväkoti     2
Viinikan päiväkoti           2
```

Visualize

Simple maps for quick visual check

In [37]:

```
import fiona
import contextily as ctx

#package = "OpenStreetMap_cafes_and_restaurants_FEB2021.gpkg"
```

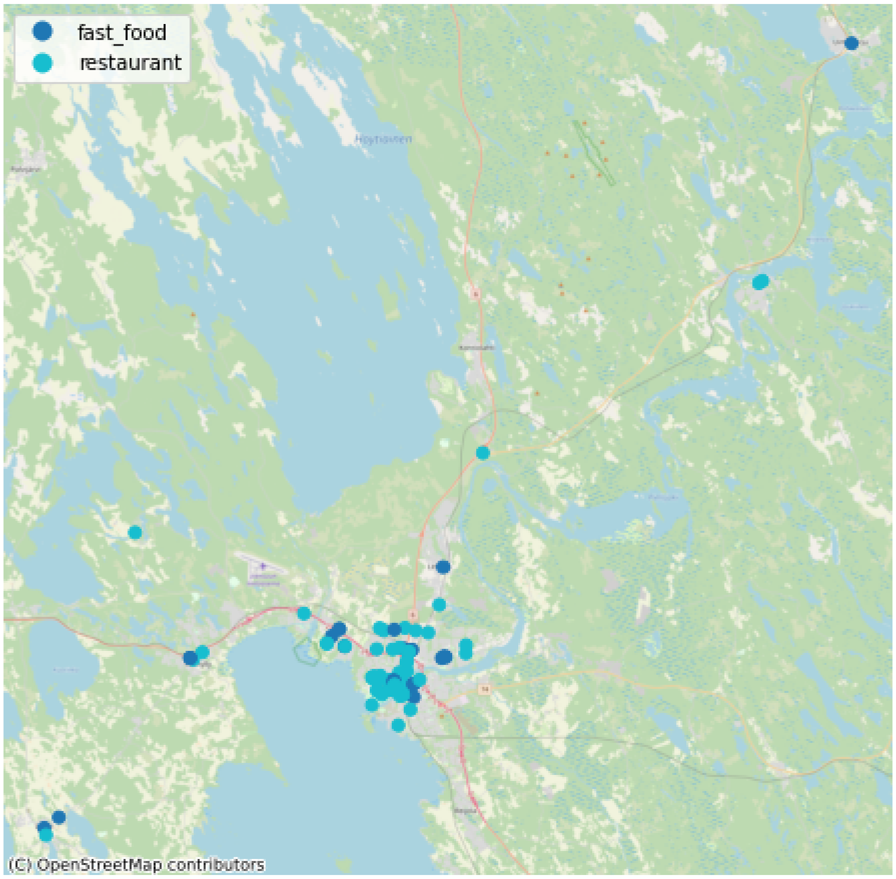
```python
for gpkg in files:

    fig, ax = plt.subplots(figsize=(12,8))

    data = gpd.read_file(gpkg, layer="restaurants")
    layername="{}_restaurants".format(os.path.basename(gpkg).split("_")[1][:-5])

    print(layername)

    #Re-project for plotting with basemap
    data = data.to_crs(epsg=3857)

    # Fetch amenities for all case areas
    data.plot(ax=ax, column="amenity", legend=True)

    ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)

    plt.axis("off")

    # Uncomment for saving img file and edit filepath if needed..
    #plt.savefig(os.path.join(r"..\img", layername + ".png"), dpi=300)
```

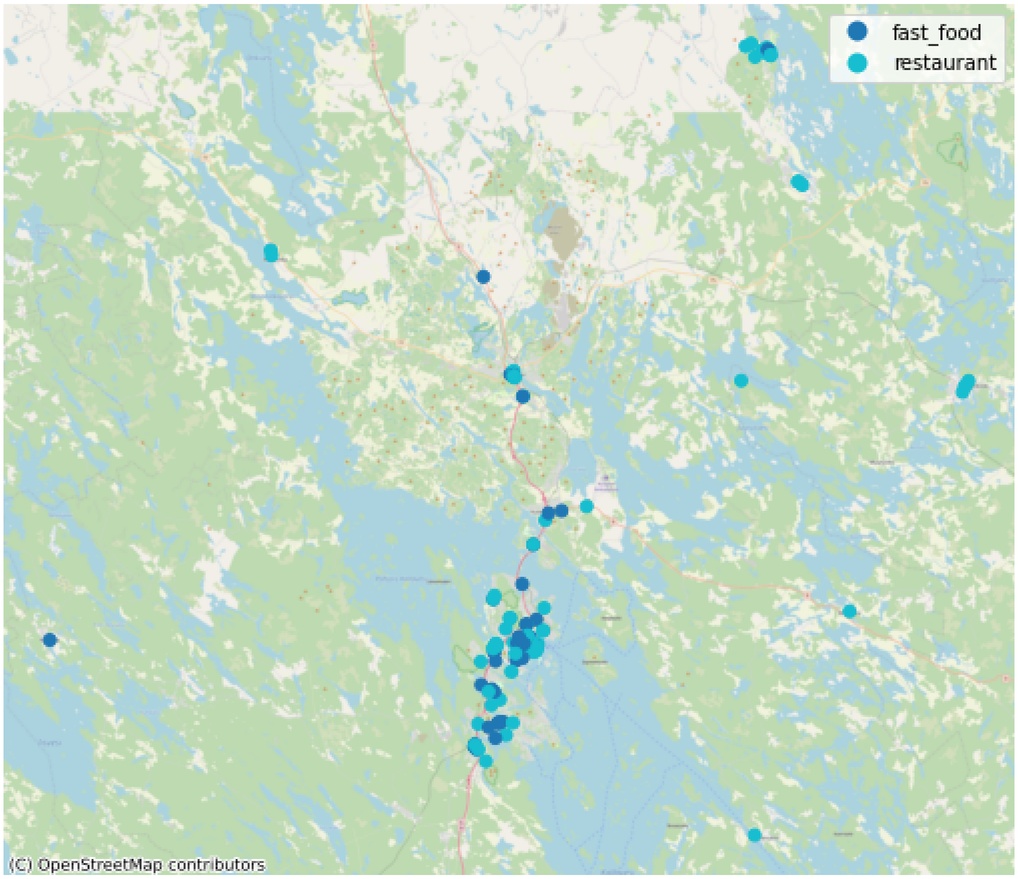
```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\ipykernel\ipkernel.py:283: D
eprecationWarning: `should_run_async` will not call `transform_cell` automatically in th
e future. Please pass the result to `transformed_cell` argument and any exception that h
appen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
Joensuu_restaurants
```

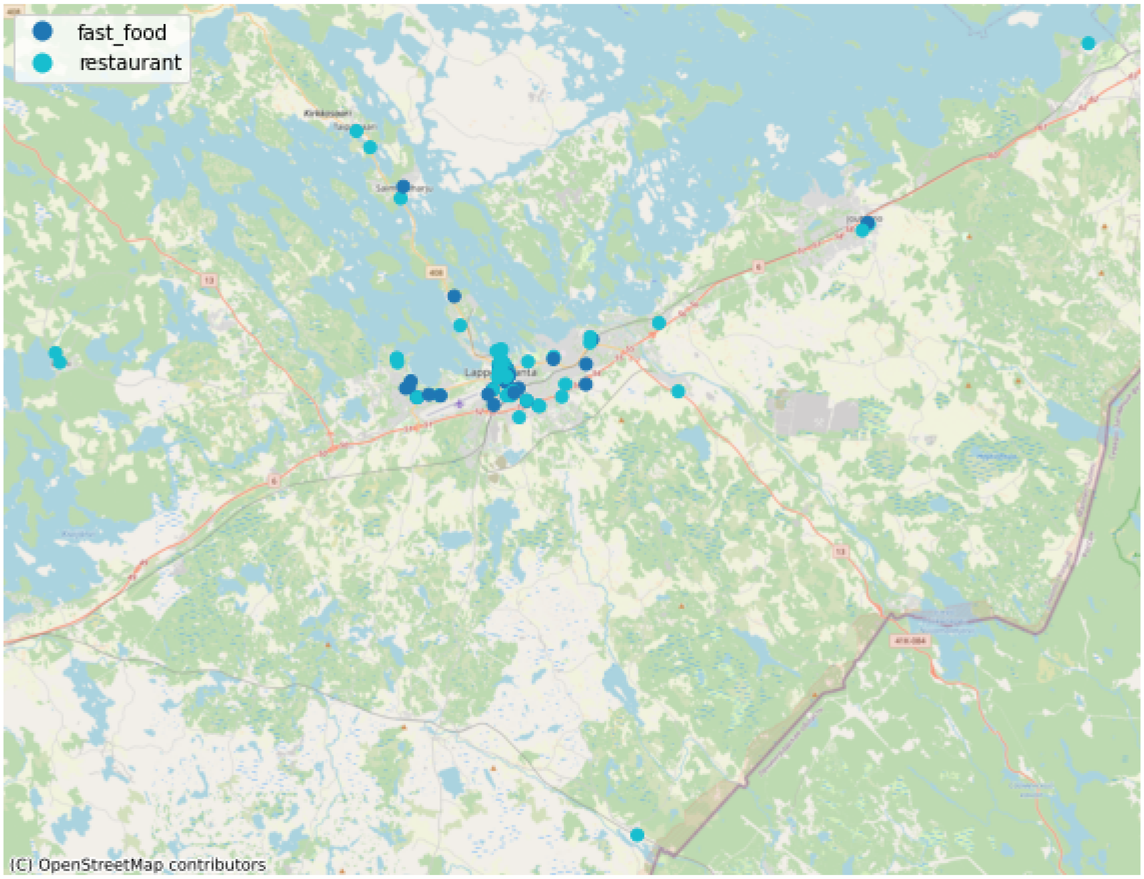
```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
Kuopio_restaurants
```

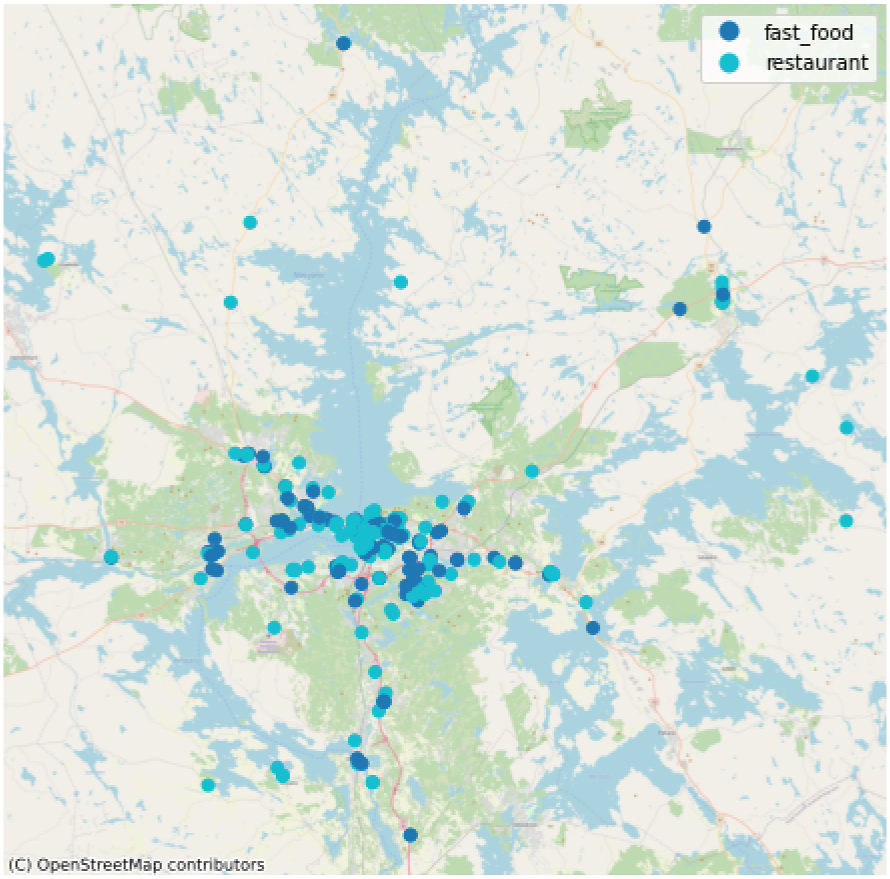
```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
Lappeenranta_restaurants
```

```
D:\ProgramFiles\Anaconda3\envs\python-gis\lib\site-packages\geopandas\geodataframe.py:42
2: RuntimeWarning: Sequential read of iterator was interrupted. Resetting iterator. This
can negatively impact the performance.
  for feature in features_lst:
Tampere_restaurants
```


● fast_food
● restaurant
Höytiäinen
(C) OpenStreetMap contributors


● fast_food
● restaurant
(C) OpenStreetMap contributors


fast_food
restaurant
(C) OpenStreetMap contributors


fast_food
restaurant
(C) OpenStreetMap contributors

Next steps

- Manual check of data quality (start with potential duplicates)
- Accessibility analysis

In [ ]: