catchment_area_analysis_supermarkets

June 17, 2024

1 Catchment area analysis of supermarkets

1.1 Libraries and settings

[ ]:
```
# Libraries
import os
import time
import glob
import json
import folium
import requests
import platform
import numpy as np
import pandas as pd
import geopandas as gdp
import matplotlib.pyplot as plt
from openrouteservice import client
from IPython.display import clear_output
from IPython.display import display, HTML

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

1.2 Import supermarket data

[ ]:
```
# Read OpenRouteService API key
with open(file='ors_token.txt', mode='r') as file:
    api_key = file.read()
# print(api_key)

# Import supermarket data
df_orig = pd.read_csv('supermarkets_data_enriched.csv',
                      sep=',',
                      encoding='utf-8')[['id',
                                         'lat',
                                         'lon',
                                         'brand',
```

```
                                        'bfs_number',
                                        'bfs_name',
                                        'addr:housenumber',
                                        'addr:postcode',
                                        'geometry']]

# Subset, only supermarkets with complete address
df = df_orig.dropna()
print(df.shape)
df.head()
```

```
(967, 9)
```

[ ]:

	id	lat	lon	brand	bfs_number	bfs_name	addr:housenumber	addr:postcode	geometry
1	280130028	47.155492	9.039666	ALDI	3315	Schänis	3	8718.0	POINT (9.0396663 47.1554921)
2	6122906632	47.158959	9.044477	Denner	3315	Schänis	14	8718.0	POINT (9.0444769 47.1589589)
4	36726161	47.226191	8.980329	Migros	3339	Uznach	25	8730.0	POINT (8.9803292 47.2261912)
6	7389830915	47.225056	8.970650	Denner	3339	Uznach	15	8730.0	POINT (8.9706499 47.2250563)
7	39947904	47.376732	8.542161	Coop	261	Zürich	1	8001.0	POINT (8.5421608 47.3767316)

1.3 Create isochrone for a single supermarket

[ ]:
```
# Settings
municip = 'Winterthur'
transport = 'driving-car' # alternativle 'foot-walking'
traveltime = 20 # in minutes

# Client settings
ors = client.Client(key=api_key)

# Set up supermarket dictionary with a single supermarket
df_sub = df.loc[df['bfs_name'] == municip].iloc[0]
supermarkets = {df_sub['brand']: {'location': [df_sub['lon'],
                                               df_sub['lat']]}}
print(supermarkets)

# Set up folium map
map = folium.Map(tiles='openstreetmap',
                 location=([df_sub['lat'],
                            df_sub['lon']]),
                 zoom_start=11)
```

```
# Parameters for server-request
params_iso = {'profile': transport,
              'range': [traveltime*60], # in seconds
              'attributes': ['total_pop']}

# Server request
for name, apt in supermarkets.items():

    # Add coords to request parameters
    params_iso['locations'] = [apt['location']]

    # Perform isochrone request
    apt['iso'] = ors.isochrones(**params_iso)

    # Add GeoJson to map
    folium.features.GeoJson(apt['iso']).add_to(map)

    # Save GeoJson as file
    # with open(f'{name}.json', 'w') as f:
        # f.write(json.dumps(apt['iso']))

    # Reverse coords due to weird folium lat/lon syntax
    folium.map.Marker(list(reversed(apt['location'])),
                      icon = folium.Icon(color='green',
                                         icon_color='#cc0000',
                                         icon='home',
                                         prefix="fa"),
                      popup = name
                      ).add_to(map)

# Plot map
map
```

```
{'Migros': {'location': [8.7064484, 47.4918744]}}
```

[ ]:
```
<folium.folium.Map at 0x10f1ad42e40>
```

1.4 Getting the number of residents in the isochrone area

[ ]:
```
# Save the data as pandas data frame
data = pd.DataFrame(apt['iso']["features"])

# Create data frame from column 'properties'
df_pop = pd.DataFrame(data.loc[0, 'properties'])
val = df_pop['total_pop'][0]
print('Number of residents in isochrone area:', f'{val:,.0f}')
```

{'type': 'FeatureCollection', 'bbox': [8.457399, 47.346561, 8.9652, 47.602965],
'features': [{'type': 'Feature', 'properties': {'group_index': 0, 'value':
1200.0, 'center': [8.706409857044083, 47.49185026202366], 'total_pop':
468446.0}, 'geometry': {'coordinates': [[[8.457809, 47.419494], [8.460263,
47.419775], [8.462716, 47.420057], [8.464777, 47.420257], [8.466869, 47.420461],
[8.469713, 47.420713], [8.472572, 47.420967], [8.475432, 47.421221], [8.478291,
47.421474], [8.47907, 47.421556], [8.479837, 47.421674], [8.480883, 47.421903],
[8.483201, 47.42254], [8.48582, 47.423321], [8.48667, 47.423489], [8.495001,
47.423048], [8.495674, 47.421864], [8.49659, 47.421086], [8.497017, 47.420879],
[8.497416, 47.420744], [8.500482, 47.419832], [8.502217, 47.419942], [8.502349,
47.420044], [8.507536, 47.426736], [8.510512, 47.427478], [8.510653, 47.427525],
[8.511669, 47.427868], [8.513011, 47.428538], [8.515898, 47.429624], [8.519236,
47.430244], [8.526708, 47.429305], [8.528477, 47.429228], [8.529223, 47.429206],
[8.534734, 47.42544], [8.535344, 47.422058], [8.53662, 47.420477], [8.537334,
47.419058], [8.536265, 47.410635], [8.531004, 47.409684], [8.530107, 47.409322],
[8.529949, 47.409203], [8.528573, 47.407029], [8.525216, 47.39973], [8.524469,
47.398469], [8.524197, 47.397716], [8.522039, 47.392917], [8.523246, 47.391108],
[8.523258, 47.391102], [8.524657, 47.390446], [8.531596, 47.38637], [8.534159,
47.384209], [8.539375, 47.380468], [8.542432, 47.381362], [8.542467, 47.381412],
[8.545604, 47.386574], [8.548742, 47.388337], [8.550812, 47.392016], [8.550833,
47.392175], [8.552098, 47.397276], [8.557484, 47.400965], [8.559131, 47.40133],
[8.559711, 47.401473], [8.560085, 47.401617], [8.565175, 47.402694], [8.569295,
47.400452], [8.570928, 47.399127], [8.571147, 47.398935], [8.573142, 47.397168],
[8.58157, 47.395512], [8.587474, 47.392433], [8.588919, 47.392276], [8.589531,
47.392354], [8.594842, 47.388278], [8.592617, 47.383636], [8.592562, 47.383593],
[8.59253, 47.381756], [8.592437, 47.380912], [8.592675, 47.380743], [8.594861,
47.380003], [8.595237, 47.379886], [8.597561, 47.379547], [8.5993, 47.379591],
[8.6014, 47.379758], [8.601472, 47.379826], [8.601476, 47.381053], [8.601884,
47.383105], [8.601645, 47.386597], [8.610434, 47.387801], [8.611806, 47.385929],
[8.61295, 47.384504], [8.621105, 47.382325], [8.621897, 47.381566], [8.622329,
47.381109], [8.624945, 47.383582], [8.628961, 47.387428], [8.636585, 47.391858],
[8.637387, 47.391597], [8.637807, 47.391496], [8.638299, 47.391434], [8.645547,
47.389977], [8.649072, 47.390711], [8.651489, 47.391391], [8.658596, 47.390842],
[8.661503, 47.385127], [8.662085, 47.383977], [8.662199, 47.383893], [8.666438,
47.380293], [8.667215, 47.380141], [8.667957, 47.380012], [8.67247, 47.378181],
[8.675831, 47.375042], [8.676299, 47.374477], [8.678074, 47.373325], [8.679435,
47.372697], [8.687281, 47.370037], [8.688564, 47.368145], [8.693979, 47.367483],
[8.696902, 47.364399], [8.69698, 47.364253], [8.699226, 47.364286], [8.70691,
47.359943], [8.706629, 47.358424], [8.707207, 47.357574], [8.707503, 47.357256],
[8.710422, 47.355655], [8.712352, 47.355795], [8.712467, 47.355897], [8.714747,
47.35763], [8.714892, 47.357814], [8.723355, 47.356305], [8.72346, 47.353671],
[8.723483, 47.353651], [8.725457, 47.353383], [8.727615, 47.353864], [8.736121,
47.351869], [8.739495, 47.348246], [8.741653, 47.346561], [8.744956, 47.347993],
[8.744834, 47.348273], [8.744651, 47.348614], [8.744425, 47.348951], [8.744068,
47.349376], [8.743612, 47.349802], [8.743226, 47.350106], [8.738572, 47.355321],
[8.738355, 47.355734], [8.738047, 47.356242], [8.737672, 47.35678], [8.737387,
47.357147], [8.736797, 47.357822], [8.736057, 47.358555], [8.733316, 47.360858],
[8.731056, 47.363661], [8.72646, 47.363518], [8.723598, 47.364556], [8.722767,

47.364876], [8.71938, 47.371934], [8.721672, 47.374711], [8.724854, 47.377577],
[8.726041, 47.380976], [8.733761, 47.385114], [8.736131, 47.377013], [8.735293,
47.375266], [8.73525, 47.375155], [8.736374, 47.372924], [8.737182, 47.372639],
[8.740724, 47.374486], [8.74124, 47.374762], [8.741609, 47.375018], [8.750334,
47.37489], [8.751531, 47.374807], [8.753905, 47.375346], [8.760465, 47.377195],
[8.764211, 47.377928], [8.764229, 47.377932], [8.766727, 47.382843], [8.767425,
47.384413], [8.768135, 47.388171], [8.766912, 47.389498], [8.765902, 47.390994],
[8.766807, 47.399677], [8.768302, 47.402951], [8.770898, 47.41139], [8.77549,
47.418863], [8.782485, 47.424034], [8.783059, 47.424313], [8.783169, 47.424391],
[8.783121, 47.426401], [8.78298, 47.426621], [8.780429, 47.431526], [8.781433,
47.435543], [8.783051, 47.439531], [8.78364, 47.440437], [8.784853, 47.448015],
[8.792753, 47.451572], [8.793973, 47.454959], [8.796115, 47.462639], [8.803008,
47.467459], [8.805173, 47.469139], [8.813355, 47.471444], [8.814122, 47.471535],
[8.814792, 47.471606], [8.822702, 47.473238], [8.825969, 47.473281], [8.830068,
47.478405], [8.832363, 47.480143], [8.832417, 47.481555], [8.83234, 47.48174],
[8.832036, 47.482272], [8.830959, 47.483458], [8.83102, 47.491983], [8.832764,
47.4931], [8.835665, 47.496145], [8.841486, 47.501003], [8.842391, 47.502423],
[8.842919, 47.504305], [8.841689, 47.504932], [8.83631, 47.509087], [8.834226,
47.516671], [8.838501, 47.520907], [8.839551, 47.521584], [8.841575, 47.523013],
[8.847639, 47.528467], [8.849371, 47.528529], [8.850446, 47.52854], [8.851911,
47.528527], [8.854452, 47.528445], [8.857016, 47.528362], [8.858699, 47.528331],
[8.860569, 47.528343], [8.862665, 47.528406], [8.864727, 47.528467], [8.865665,
47.528478], [8.866572, 47.52847], [8.867914, 47.528411], [8.869036, 47.528313],
[8.870357, 47.528135], [8.871259, 47.527974], [8.872127, 47.527787], [8.87359,
47.527399], [8.874608, 47.527073], [8.876066, 47.526548], [8.877827, 47.525862],
[8.879623, 47.525165], [8.880565, 47.52482], [8.881631, 47.52446], [8.882581,
47.524164], [8.88394, 47.523781], [8.887124, 47.522979], [8.88862, 47.522574],
[8.894492, 47.519452], [8.896503, 47.517577], [8.89729, 47.516875], [8.898137,
47.516185], [8.898709, 47.515765], [8.899698, 47.515119], [8.900483, 47.51467],
[8.901347, 47.514231], [8.902546, 47.513696], [8.903335, 47.513397], [8.904002,
47.513169], [8.904652, 47.512968], [8.905678, 47.512686], [8.906708, 47.512442],
[8.908114, 47.512143], [8.913329, 47.506907], [8.912956, 47.503326], [8.915612,
47.499276], [8.922798, 47.500104], [8.926429, 47.501047], [8.934616, 47.504027],
[8.935622, 47.503308], [8.936219, 47.50285], [8.93692, 47.502274], [8.937811,
47.50148], [8.938245, 47.50106], [8.939238, 47.500014], [8.940054, 47.499077],
[8.941385, 47.497485], [8.942811, 47.495783], [8.943674, 47.494835], [8.944266,
47.494229], [8.944856, 47.493661], [8.945961, 47.492696], [8.946777, 47.492055],
[8.947594, 47.491468], [8.94833, 47.490976], [8.949169, 47.490458], [8.950167,
47.489892], [8.954495, 47.48721], [8.961738, 47.484543], [8.963215, 47.487826],
[8.961002, 47.488822], [8.95879, 47.489817], [8.956752, 47.490732], [8.954715,
47.491648], [8.953459, 47.492247], [8.952506, 47.492726], [8.951856, 47.493071],
[8.950945, 47.493589], [8.950227, 47.494036], [8.949597, 47.494459], [8.94888,
47.494976], [8.948189, 47.495523], [8.947266, 47.496335], [8.946767, 47.496818],
[8.946252, 47.497348], [8.945476, 47.498204], [8.944147, 47.499793], [8.942817,
47.501386], [8.941953, 47.502378], [8.940858, 47.503536], [8.941576, 47.512393],
[8.941029, 47.513635], [8.938624, 47.514583], [8.935791, 47.513017], [8.93157,
47.512021], [8.925737, 47.514337], [8.92504, 47.51468], [8.924382, 47.514859],
[8.922799, 47.515221], [8.914983, 47.514641], [8.912658, 47.51506], [8.910349,

47.515476], [8.908827, 47.515765], [8.907533, 47.516042], [8.906574, 47.516276],
[8.898904, 47.52026], [8.896946, 47.522085], [8.896262, 47.522674], [8.895739,
47.523092], [8.895006, 47.523623], [8.894214, 47.524129], [8.893566, 47.524493],
[8.892694, 47.524921], [8.891981, 47.525226], [8.891296, 47.525489], [8.890365,
47.525803], [8.889677, 47.526015], [8.888068, 47.526453], [8.884819, 47.527272],
[8.88356, 47.527629], [8.882702, 47.527897], [8.88172, 47.52823], [8.880859,
47.528545], [8.879132, 47.529218], [8.877371, 47.529903], [8.87583, 47.530459],
[8.874691, 47.530827], [8.873966, 47.531036], [8.873005, 47.531279], [8.872021,
47.531493], [8.870994, 47.531679], [8.86952, 47.53188], [8.868233, 47.531997],
[8.866735, 47.532066], [8.865702, 47.532078], [8.864688, 47.532067], [8.862558,
47.532004], [8.860462, 47.531942], [8.858676, 47.531931], [8.857084, 47.531962],
[8.854568, 47.532043], [8.852027, 47.532125], [8.846508, 47.539143], [8.846576,
47.545855], [8.851065, 47.547865], [8.8525, 47.548087], [8.854347, 47.547941],
[8.85777, 47.546131], [8.860179, 47.544552], [8.862561, 47.54334], [8.864225,
47.542489], [8.864412, 47.542355], [8.865931, 47.542242], [8.866213, 47.542416],
[8.86622, 47.54242], [8.8669, 47.542835], [8.870483, 47.544026], [8.87081,
47.544083], [8.873311, 47.544553], [8.874157, 47.545045], [8.875758, 47.545813],
[8.87665, 47.546242], [8.876978, 47.546439], [8.878221, 47.547239], [8.8835,
47.54932], [8.8845, 47.549681], [8.885635, 47.549981], [8.887706, 47.550835],
[8.888735, 47.551442], [8.888775, 47.551495], [8.889208, 47.552876], [8.889202,
47.552918], [8.88887, 47.55493], [8.888866, 47.554935], [8.88713, 47.558425],
[8.891166, 47.563896], [8.893131, 47.563994], [8.898237, 47.562305], [8.90433,
47.560289], [8.905441, 47.560029], [8.906485, 47.559982], [8.906773, 47.559995],
[8.907425, 47.56016], [8.910407, 47.56095], [8.915013, 47.562726], [8.91533,
47.562848], [8.917731, 47.56564], [8.918988, 47.567736], [8.918803, 47.570359],
[8.923305, 47.576256], [8.92571, 47.577673], [8.927736, 47.578844], [8.929965,
47.580064], [8.931361, 47.580777], [8.932663, 47.581397], [8.933618, 47.581826],
[8.934578, 47.582229], [8.935436, 47.582568], [8.936345, 47.582902], [8.937629,
47.583335], [8.938202, 47.583516], [8.939442, 47.583877], [8.94131, 47.58435],
[8.942491, 47.584606], [8.943597, 47.58482], [8.944527, 47.58498], [8.945945,
47.58519], [8.947831, 47.585414], [8.949517, 47.585571], [8.951515, 47.585708],
[8.953885, 47.585817], [8.956551, 47.58589], [8.958932, 47.585926], [8.962066,
47.585952], [8.9652, 47.585977], [8.965171, 47.589577], [8.962037, 47.589551],
[8.958903, 47.589526], [8.956497, 47.58949], [8.953787, 47.589416], [8.95135,
47.589304], [8.949271, 47.589162], [8.947499, 47.588999], [8.945522, 47.588765],
[8.944002, 47.588542], [8.942987, 47.588368], [8.94181, 47.588141], [8.940547,
47.587868], [8.93856, 47.587367], [8.9372, 47.586974], [8.936545, 47.586768],
[8.935196, 47.586314], [8.934196, 47.585948], [8.933258, 47.585579], [8.932224,
47.585145], [8.931191, 47.584683], [8.929814, 47.584027], [8.92833, 47.583271],
[8.926008, 47.582002], [8.92391, 47.580791], [8.921477, 47.579357], [8.919059,
47.577931], [8.917016, 47.576771], [8.909843, 47.576756], [8.905068, 47.576881],
[8.904605, 47.576913], [8.904124, 47.576902], [8.90369, 47.576846], [8.903164,
47.576738], [8.902815, 47.576645], [8.902824, 47.573451], [8.897848, 47.568357],
[8.896935, 47.568169], [8.896077, 47.568019], [8.894977, 47.567871], [8.894321,
47.567802], [8.892891, 47.567692], [8.8852, 47.571677], [8.884424, 47.57308],
[8.882417, 47.574915], [8.880714, 47.575053], [8.872234, 47.57363], [8.870775,
47.574887], [8.870474, 47.574884], [8.868667, 47.573144], [8.861851, 47.570193],
[8.855849, 47.568981], [8.852896, 47.564237], [8.852077, 47.562629], [8.847319,

47.556318], [8.845236, 47.556278], [8.843122, 47.556238], [8.841463, 47.556177],
[8.840148, 47.556095], [8.838988, 47.555989], [8.837775, 47.555837], [8.836295,
47.555586], [8.835028, 47.555314], [8.833831, 47.555003], [8.832697, 47.55454],
[8.831657, 47.554176], [8.829818, 47.553541], [8.828698, 47.552994], [8.827628,
47.552405], [8.826764, 47.551871], [8.825547, 47.551032], [8.824544, 47.550263],
[8.819729, 47.549552], [8.819051, 47.558224], [8.820889, 47.559301], [8.81907,
47.562407], [8.813056, 47.560748], [8.80409, 47.560826], [8.800874, 47.561697],
[8.796742, 47.562044], [8.789481, 47.564352], [8.787911, 47.566139], [8.786205,
47.567554], [8.785978, 47.567531], [8.77713, 47.5676], [8.772114, 47.571502],
[8.769641, 47.574118], [8.765737, 47.574261], [8.757786, 47.576974], [8.750314,
47.576637], [8.742717, 47.574984], [8.740787, 47.575362], [8.740216, 47.575506],
[8.735578, 47.575501], [8.729711, 47.57962], [8.729224, 47.580369], [8.722228,
47.584784], [8.721638, 47.585349], [8.721567, 47.585393], [8.72017, 47.584748],
[8.71403, 47.584572], [8.711121, 47.586591], [8.708861, 47.588464], [8.708441,
47.588677], [8.707412, 47.588545], [8.699263, 47.59148], [8.697744, 47.592372],
[8.693506, 47.599802], [8.693572, 47.600526], [8.693556, 47.60111], [8.693447,
47.601807], [8.693223, 47.602564], [8.69316, 47.602711], [8.692682, 47.602965],
[8.689567, 47.601161], [8.688318, 47.596856], [8.687465, 47.595288], [8.679465,
47.594465], [8.67922, 47.594488], [8.677233, 47.592677], [8.674359, 47.589014],
[8.668846, 47.585253], [8.667203, 47.583373], [8.666448, 47.581484], [8.658189,
47.578124], [8.65796, 47.578], [8.650312, 47.575236], [8.646275, 47.575221],
[8.641382, 47.575629], [8.640419, 47.575582], [8.631745, 47.575264], [8.630966,
47.57529], [8.627803, 47.574514], [8.625481, 47.573813], [8.619584, 47.567975],
[8.61538, 47.567264], [8.610135, 47.565597], [8.603629, 47.565863], [8.600636,
47.568476], [8.600029, 47.568648], [8.599238, 47.56886], [8.598918, 47.568918],
[8.596655, 47.567609], [8.595119, 47.566207], [8.594824, 47.565881], [8.594722,
47.56551], [8.596603, 47.557178], [8.597549, 47.548518], [8.597726, 47.548438],
[8.598706, 47.548054], [8.599963, 47.54773], [8.603397, 47.545376], [8.603136,
47.538861], [8.604025, 47.538102], [8.604849, 47.537902], [8.605575, 47.537744],
[8.608862, 47.535962], [8.610874, 47.534293], [8.612883, 47.525812], [8.608887,
47.522944], [8.601053, 47.522642], [8.600231, 47.522982], [8.599751, 47.523138],
[8.597365, 47.524267], [8.595083, 47.525328], [8.593895, 47.525936], [8.592461,
47.525092], [8.591846, 47.521545], [8.590335, 47.515099], [8.589708, 47.508296],
[8.589343, 47.506989], [8.582444, 47.502012], [8.582281, 47.501972], [8.58172,
47.499242], [8.581834, 47.499056], [8.582227, 47.498479], [8.582664, 47.498143],
[8.583595, 47.49759], [8.58627, 47.491486], [8.583403, 47.488583], [8.583034,
47.488404], [8.580643, 47.487544], [8.579927, 47.487247], [8.579641, 47.486846],
[8.579563, 47.486239], [8.579892, 47.484946], [8.580947, 47.483794], [8.588428,
47.479688], [8.588699, 47.478737], [8.586223, 47.471057], [8.578355, 47.472153],
[8.57736, 47.47444], [8.575802, 47.474518], [8.571722, 47.474302], [8.570405,
47.473861], [8.567061, 47.473368], [8.559696, 47.478068], [8.5577, 47.48042],
[8.557672, 47.480453], [8.555417, 47.480837], [8.548881, 47.484601], [8.547446,
47.486367], [8.546246, 47.487875], [8.545057, 47.489369], [8.542575, 47.492551],
[8.540085, 47.495743], [8.536373, 47.503687], [8.53617, 47.504526], [8.535468,
47.506201], [8.535042, 47.50691], [8.534642, 47.507468], [8.534172, 47.508032],
[8.531408, 47.505725], [8.533321, 47.499623], [8.533485, 47.4991], [8.533709,
47.498524], [8.534188, 47.497589], [8.534746, 47.496727], [8.5351, 47.49624],
[8.537262, 47.493508], [8.539736, 47.490337], [8.542218, 47.487155], [8.543429,

47.485633], [8.544629, 47.484125], [8.546087, 47.482331], [8.547534, 47.48055],
[8.54898, 47.47877], [8.550426, 47.476989], [8.551172, 47.476109], [8.552,
47.475194], [8.553287, 47.473885], [8.554037, 47.473177], [8.555156, 47.472181],
[8.556354, 47.471191], [8.557492, 47.470314], [8.558669, 47.469465], [8.560282,
47.468388], [8.561937, 47.467372], [8.56362, 47.466424], [8.564708, 47.466018],
[8.567297, 47.464436], [8.569545, 47.459214], [8.564021, 47.453009], [8.563893,
47.452965], [8.563763, 47.452885], [8.563735, 47.452847], [8.563678, 47.452741],
[8.563159, 47.450735], [8.563108, 47.450466], [8.563105, 47.4503], [8.56314,
47.449937], [8.563189, 47.449722], [8.559144, 47.441848], [8.557861, 47.441249],
[8.551485, 47.443274], [8.55079, 47.447195], [8.550709, 47.447318], [8.542256,
47.450269], [8.5403, 47.452149], [8.538714, 47.453717], [8.538479, 47.453932],
[8.536705, 47.454006], [8.536604, 47.453948], [8.53012, 47.449422], [8.52836,
47.44784], [8.52809, 47.444928], [8.528182, 47.44471], [8.52828, 47.438068],
[8.523345, 47.435115], [8.522381, 47.434955], [8.520738, 47.434641], [8.51933,
47.434334], [8.518429, 47.43412], [8.517513, 47.433875], [8.515651, 47.43333],
[8.514882, 47.433078], [8.512622, 47.432329], [8.511819, 47.432075], [8.509516,
47.4313], [8.505373, 47.430071], [8.496865, 47.431523], [8.496202, 47.43198],
[8.49572, 47.432289], [8.494963, 47.432635], [8.494098, 47.432787], [8.493464,
47.432722], [8.492334, 47.432185], [8.483593, 47.434265], [8.48052, 47.436232],
[8.478337, 47.43337], [8.477973, 47.42506], [8.475114, 47.424807], [8.472254,
47.424553], [8.469395, 47.424299], [8.46652, 47.424044], [8.464428, 47.42384],
[8.462306, 47.423633], [8.459852, 47.423352], [8.457399, 47.42307], [8.457809,
47.419494]]], 'type': 'Polygon'}}], 'metadata': {'attribution':
'openrouteservice.org | OpenStreetMap contributors', 'service': 'isochrones',
'timestamp': 1718634955892, 'query': {'profile': 'driving-car', 'locations':
[[8.7064484, 47.4918744]], 'range': [1200.0], 'range_type': 'time',
'attributes': ['total_pop']}, 'engine': {'version': '8.1.0', 'build_date':
'2024-06-05T10:07:23Z', 'graph_date': '2024-06-10T14:45:17Z'}}}
Number of residents in isochrone area: 468,446

1.5 Estimating the purchasing power in the isochrone area

[ ]:
```
# Simple assumtions:
# 500 CHF per resident and month
# Residents / employee ratio = 2:1
# 20 CHF per employee and working day

# Result
val = ((df_pop['total_pop'][0] * 500) + (df_pop['total_pop'][0] * 0.5 * 20 *␣
 ↪20)) / 10**6
print('Purchasing power per month:',
      f'{val:.1f}',
     'Mio. CHF')
```

Purchasing power per month: 327.9 Mio. CHF

1.6 Automization of the analysis for multiple of supermarkets

[ ]:
```
# Client settings
ors = client.Client(key=api_key)

# Parameters for server-request
params_iso = {'profile': 'driving-car',
              'range': [15*60],
              'attributes': ['total_pop']}

# Df to store the data
pop_out = []

try:
    for i in range(0, 10):

        print('Preparing isochrone for supermarkt', df[['id',
                                                        'brand',
                                                        'bfs_name',
                                                        'addr:housenumber',
                                                        'addr:postcode']].
 ↪iloc[i])

        supermarkets = {df['brand'].iloc[i]: {'location': [df['lon'].iloc[i],
                                                           df['lat'].iloc[i]]}}
        # Server request
        for apt in supermarkets.values():
            # Add coords to request parameters
            params_iso['locations'] = [apt['location']]

            # Perform isochrone request
            apt['iso'] = ors.isochrones(**params_iso)
            time.sleep(1)

            # Save the data as pandas data frame
            data = pd.DataFrame(apt['iso']["features"])

            # Create data frame from column 'properties'
            df_pop = pd.DataFrame(data.loc[0, 'properties'])

            # Write values
            pop_out.append([df['id'].iloc[i],
                            df['brand'].iloc[i],
                            df['bfs_number'].iloc[i],
                            df['bfs_name'].iloc[i],
                            df['addr:housenumber'].iloc[i],
                            df['addr:postcode'].iloc[i],
```

```
                            df_pop['total_pop'][0],
                            ((df_pop['total_pop'][0] * 500) +␣
 ↪(df_pop['total_pop'][0] * 0.5 * 20 * 20)) / 10**6])
            clear_output(wait=True)

except:
    pop_out.append([df['id'].iloc[i],
                    df['brand'].iloc[i],
                    df['bfs_number'].iloc[i],
                    df['bfs_name'].iloc[i],
                    df['addr:housenumber'].iloc[i],
                    df['addr:postcode'].iloc[i],
                    np.NaN,
                    np.NaN])
    clear_output(wait=True)
```

```
Preparing isochrone for supermarket Migros:
ID: 119249170
Name: Zürich
Address: 31-35, 8001.0
```

[ ]:
```
# Print list
df_pop_out = pd.DataFrame(pop_out, columns=['id', 'brand', 'bfs_number',␣
 ↪'bfs_name', 'addr:housenumber', 'addr:postcode', 'total_pop',␣
 ↪'purchasing_power'])
display(HTML(df_pop_out.to_html(index=False)))
```

```
<IPython.core.display.HTML object>
```