QuackOSM & OvertureMaestro

Open geospatial data at your fingertips

Kamil Raczycki, EuroPython, July 17, 2025

About me

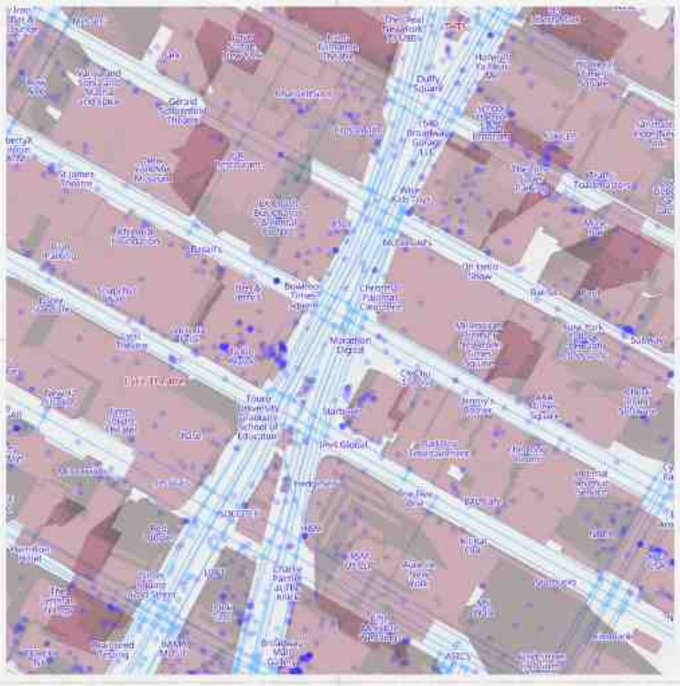
Kamil Raczycki

- MSc in Data Science 🎓
- Senior Data Scientist working with 🌍 data
- Open-source hobbyist contributor 👨‍💻
- Co-developer of **SRAI** library, maintainer of **QuackOSM** and **OvertureMaestro** libraries

kamilraczycki.com
/raczyckikamil
/raczeq

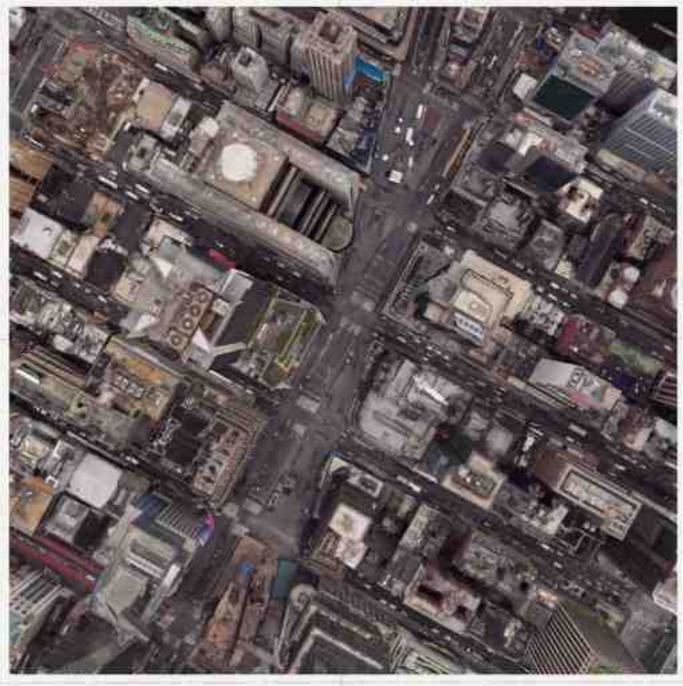
Intro to geo data

Vector vs raster (satellite)



Vector
detailed geometries 🏙️🛣️

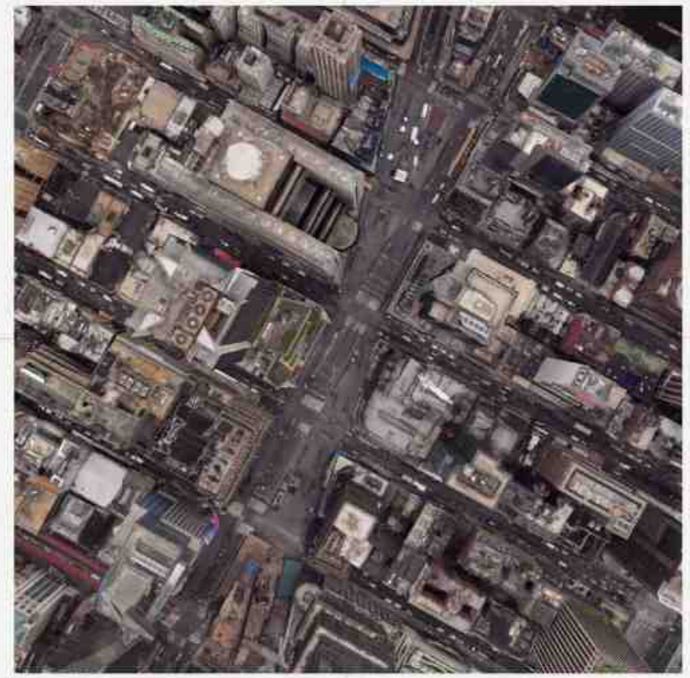
Raster
pixels 🛰️🌧️



Vector vs raster (satellite)

Vector
detailed geometries

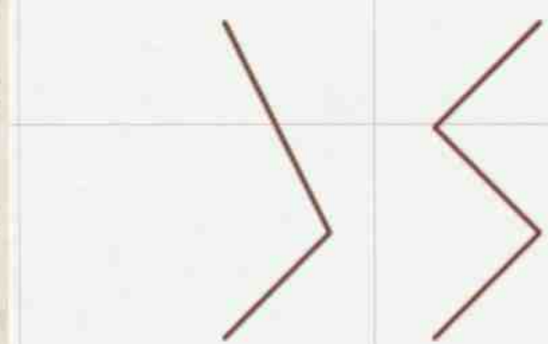
Raster
pixels



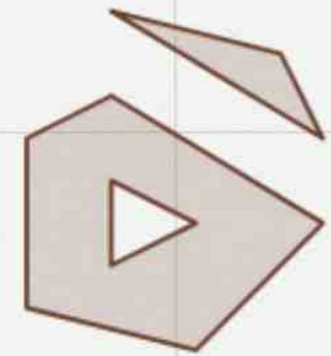
OGC Simple Features

Point

LineString



Polygon



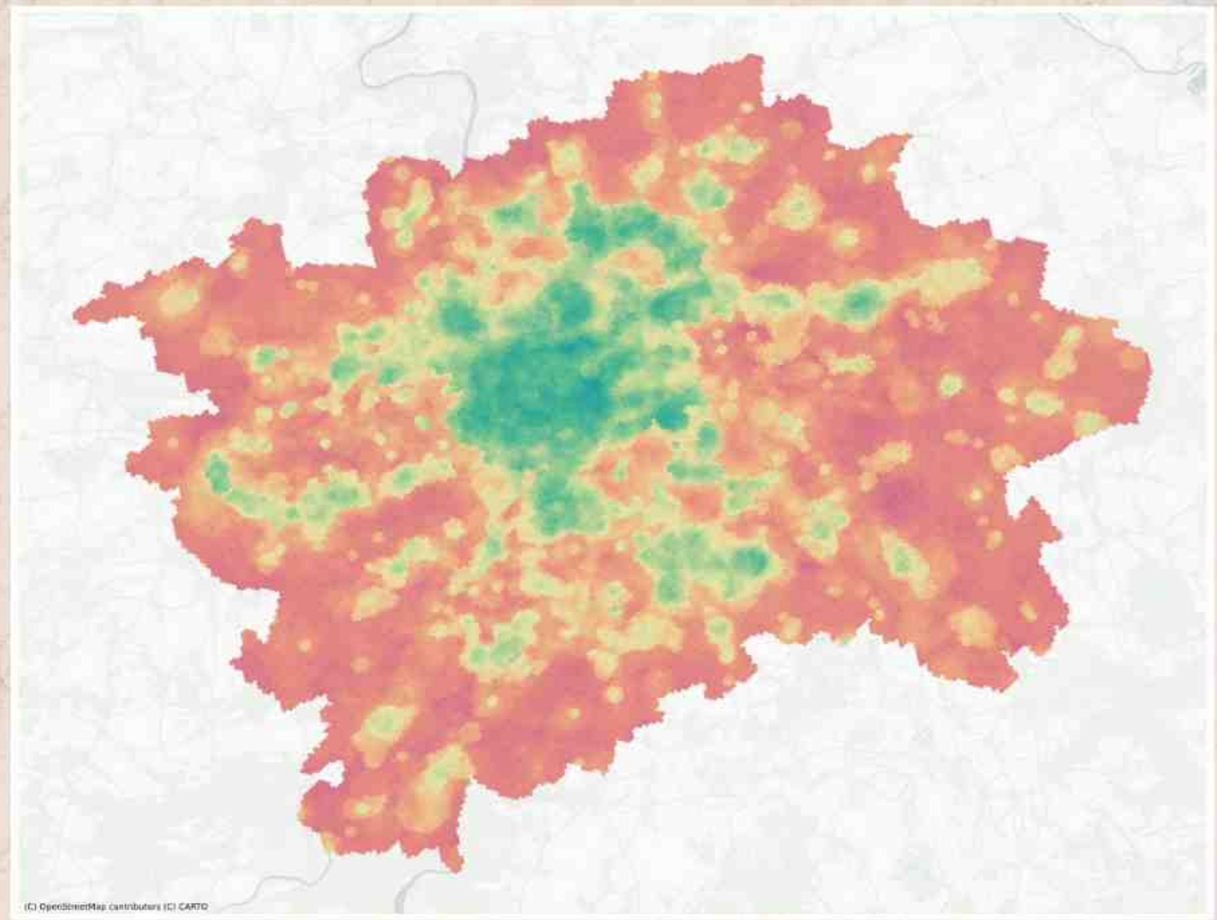
... and MultiPoint, MultiLineString, MultiPolygon, GeometryCollection

Where is geo data used?

Where is geo data used?

Research

🧍‍♂️➡️🧍‍♀️ | 📍🏙️ | 🧠🗺️

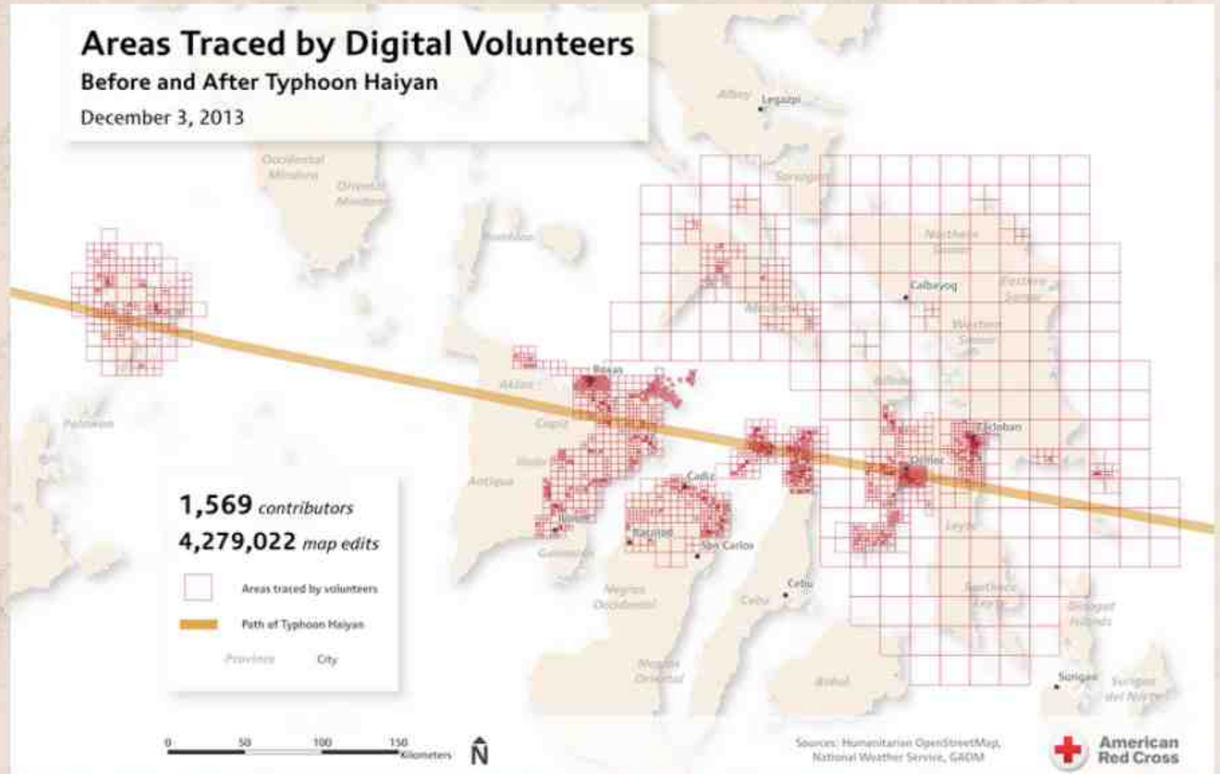


🔌⚡🚗 EV Station heatmap for Prague
Trained with data 📊 from 📍Amsterdam

Where is geo data used?

Research

Public Interest



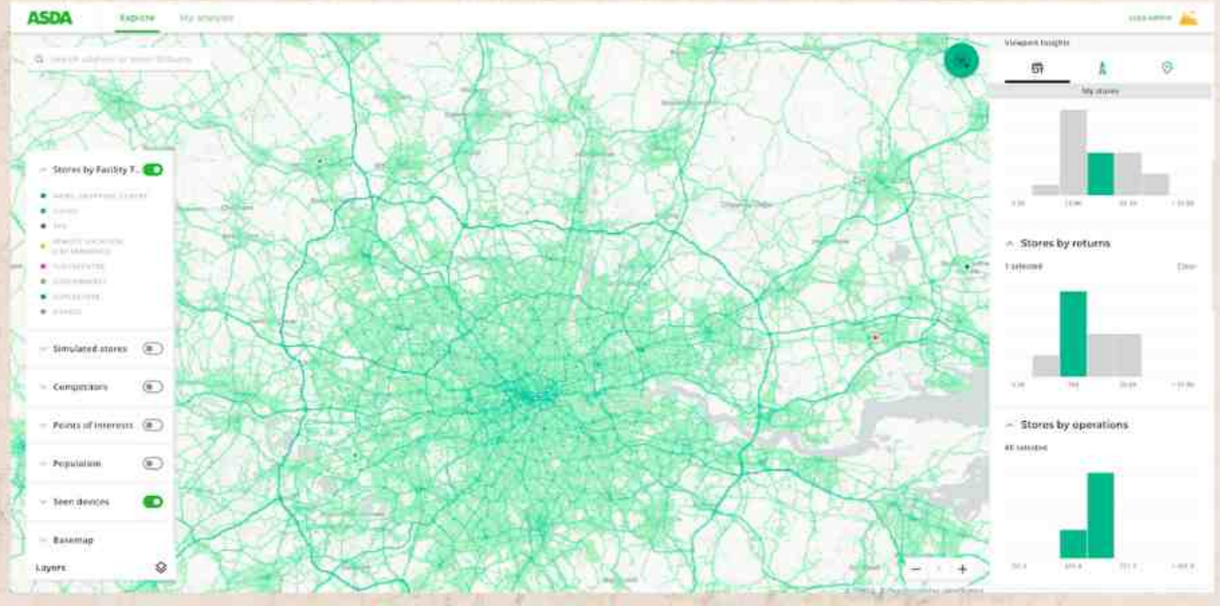
https://preparecenter.org/story/openstreetmap-in-humanitarian-response/

Where is geo data used?

Research

Public Interest

Business



https://carto.com/blog/how-asda-uses-location-intelligence-site-selection

Where is geo data used?

Research

Public Interest

Business

Gaming

Image from City Bus Manager Steam page

Data sources

OpenStreetMap

Overture Maps Foundation

Data sources

OpenStreetMap

- Buildings
- Roads
- Water
- Land Use
- POIs

& more

Overture Maps Foundation

Data sources

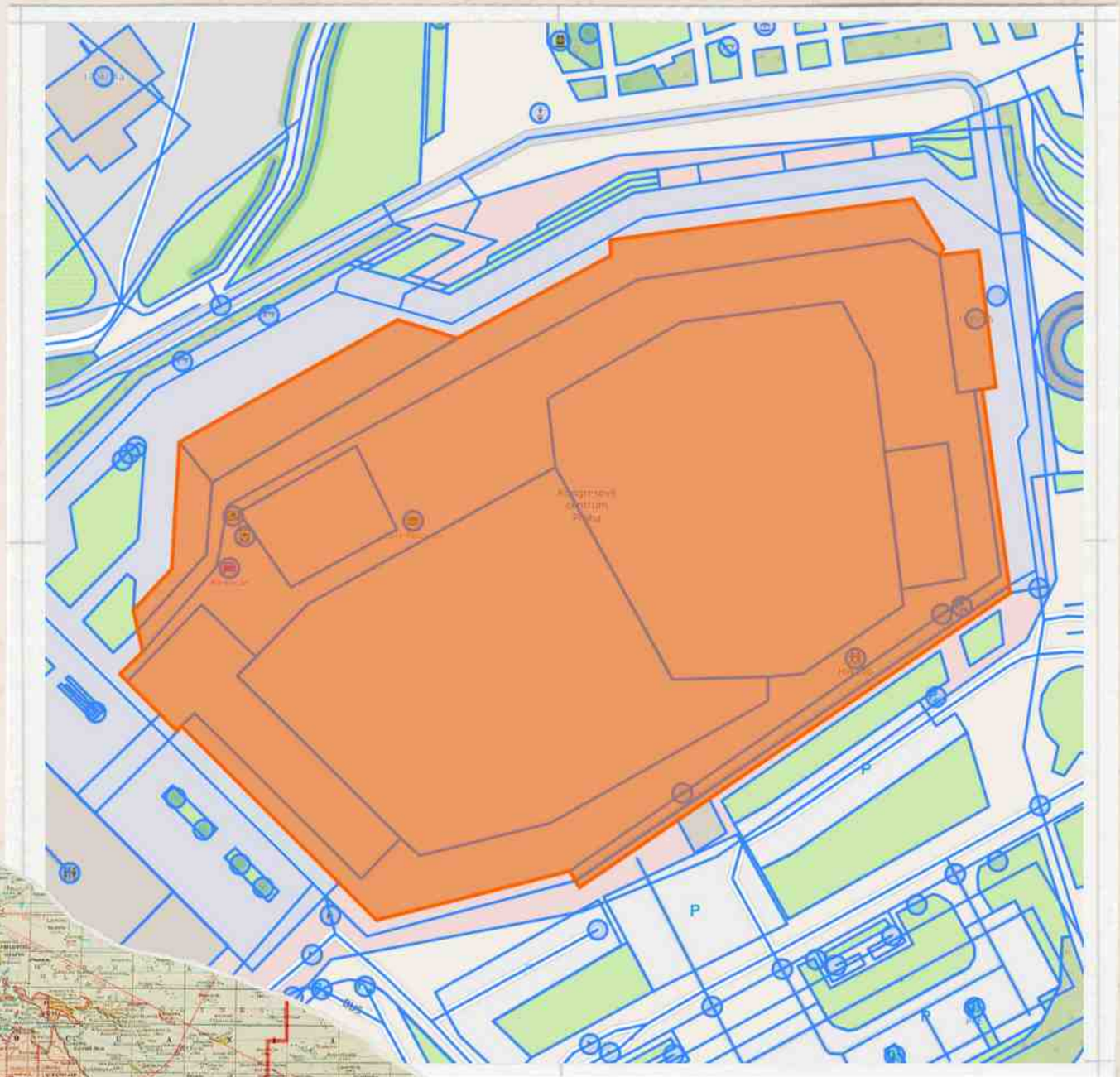
OpenStreetMap

Overture Maps Foundation

Governmental & civic open data

Contributors

https://docs.overturemaps.org/attribution/



OpenStreetMap

Tags

building	yes
building:colour	lightgray
building:levels	7
height	26
name	Kongresové centrum Praha
name:de	Kongresszentrum Prag
name:en	Prague Congress Centre

Sub Regions

Click on the region name to see the overview page for that region, or select one of the file extension links for quick access.

Sub Region	Quick Links		
	.osm.pbf		.shp.zip
Albania	[.osm.pbf]	(47.5 MB)	[.shp.zip]
Andorra	[.osm.pbf]	(3.1 MB)	[.shp.zip]
Austria	[.osm.pbf]	(739 MB)	[.shp.zip]
Azores	[.osm.pbf]	(16.1 MB)	[.shp.zip]
Belarus	[.osm.pbf]	(312 MB)	[.shp.zip]
Belgium	[.osm.pbf]	(618 MB)	[.shp.zip]
Bosnia-Herzegovina	[.osm.pbf]	(143 MB)	[.shp.zip]
Bulgaria	[.osm.pbf]	(145 MB)	[.shp.zip]
Croatia	[.osm.pbf]	(176 MB)	[.shp.zip]
Cyprus	[.osm.pbf]	(32.3 MB)	[.shp.zip]
Czech Republic	[.osm.pbf]	(852 MB)	[.shp.zip]
Denmark	[.osm.pbf]	(444 MB)	[.shp.zip]
Estonia	[.osm.pbf]	(110 MB)	[.shp.zip]
Faroe Islands	[.osm.pbf]	(7.2 MB)	[.shp.zip]
Finland	[.osm.pbf]	(644 MB)	[.shp.zip]
France	[.osm.pbf]	(4.5 GB)	✖
Georgia	[.osm.pbf]	(89 MB)	[.shp.zip]
Germany	[.osm.pbf]	(4.2 GB)	✖

Geofabrik extracts

Not what you were looking for? Geofabrik is a consulting and software development firm based in Karlsruhe, Germany specializing in OpenStreetMap

OSM BY THE SLICE
made to order • baked fresh minutely

• Data updated about 15 hours ago

Rectangle | Angled Rectangle | Polygon | Circle

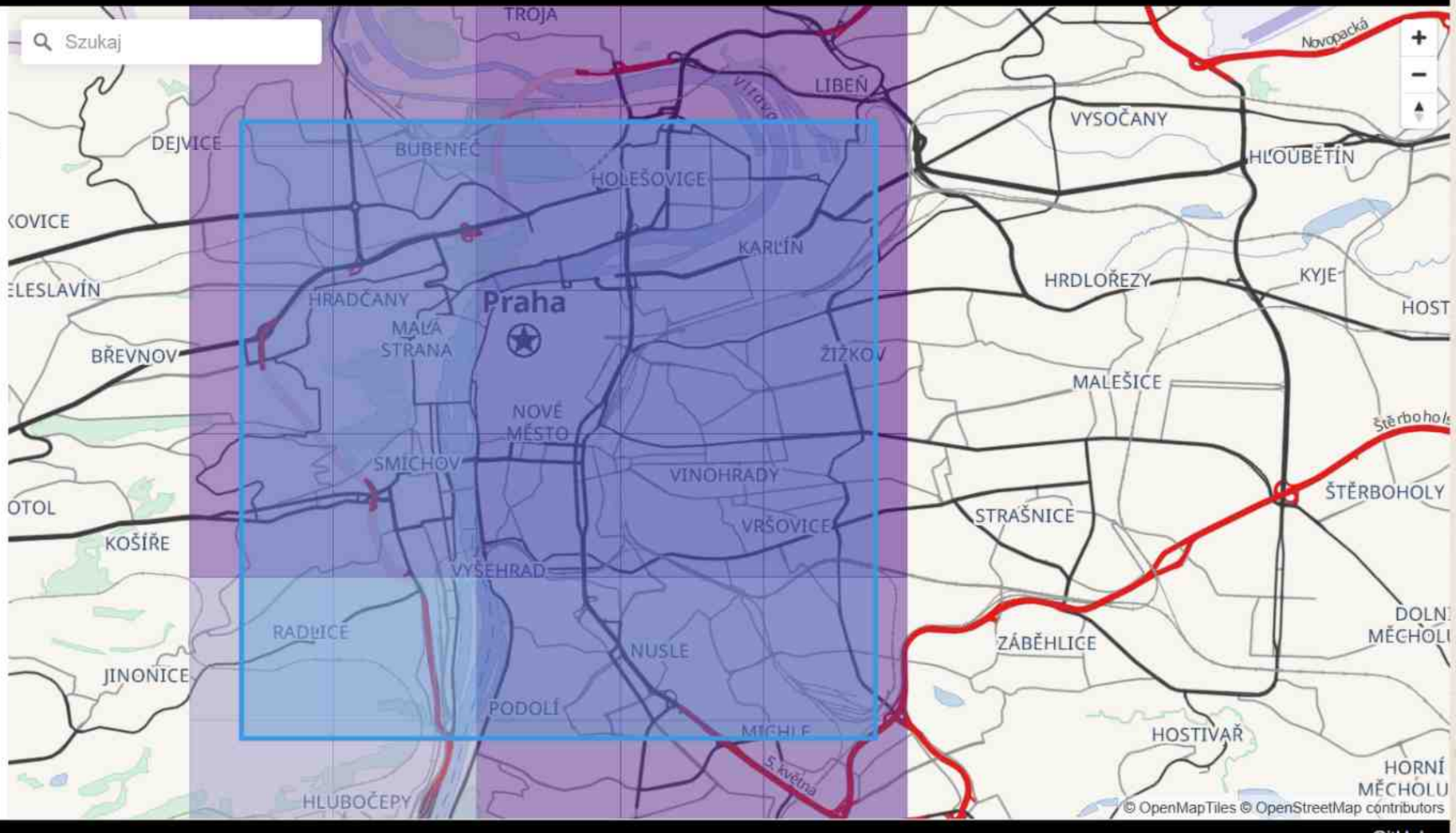
Paste bbox or GeoJSON:

Load

name this area...

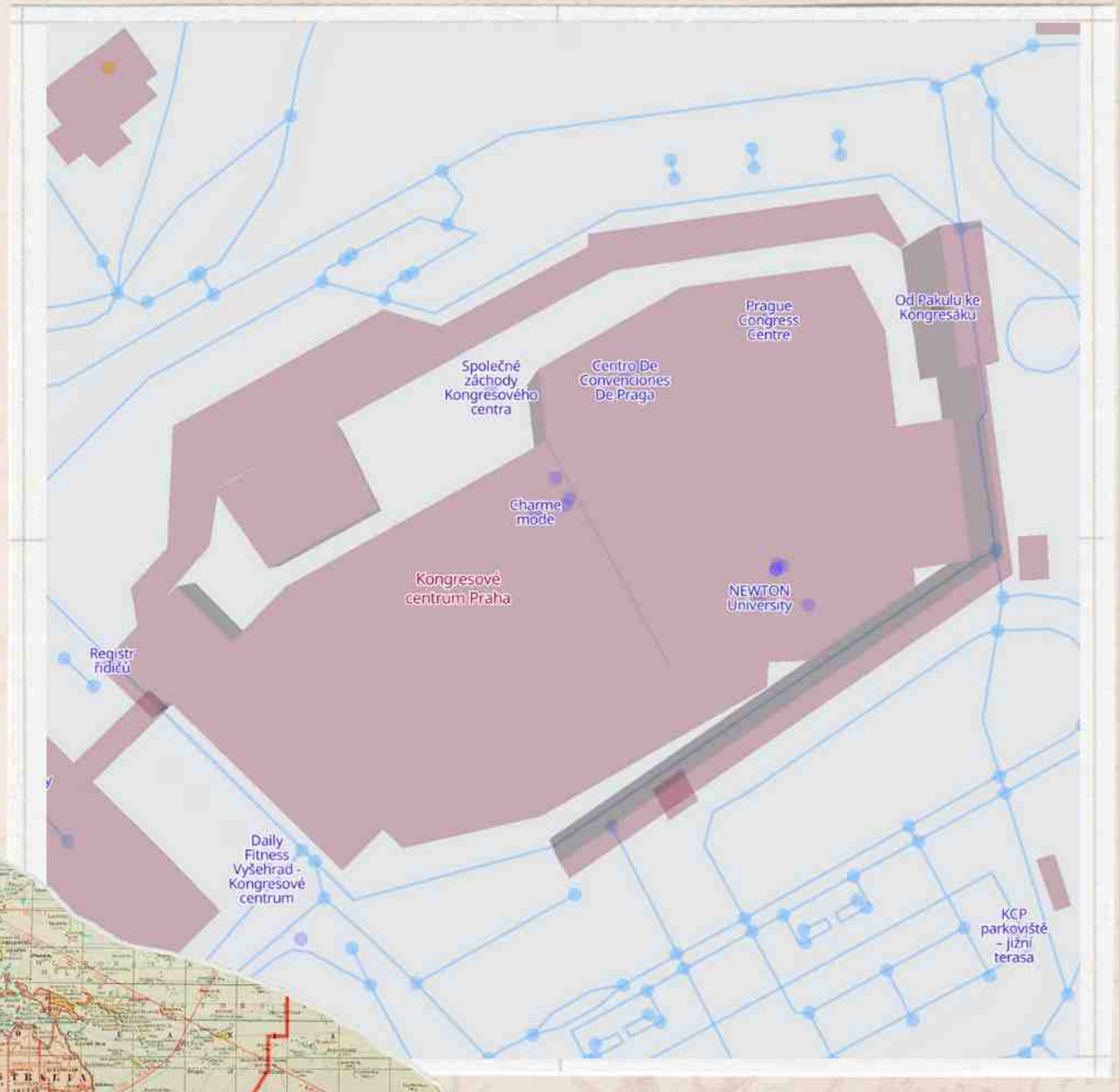
Generate Slice

SliceOSM



an OpenStreetMap US Community Project

GitHub



Overture Maps Foundation

id: 021ae5dd-1300-4a93-a2c5-7ae37be6318d

theme: places

type: place

sources:

dataset: meta

record_id: 341332855920781

update_time: 2025-06-02T07:00:00.000Z

version: 1

names + {"primary":"Prague Congress Centre","comm

confidence + 0.9459263554653523

categories + {"primary":"event_planning","alternate":

websites + ["http://www.kcp.cz/"]

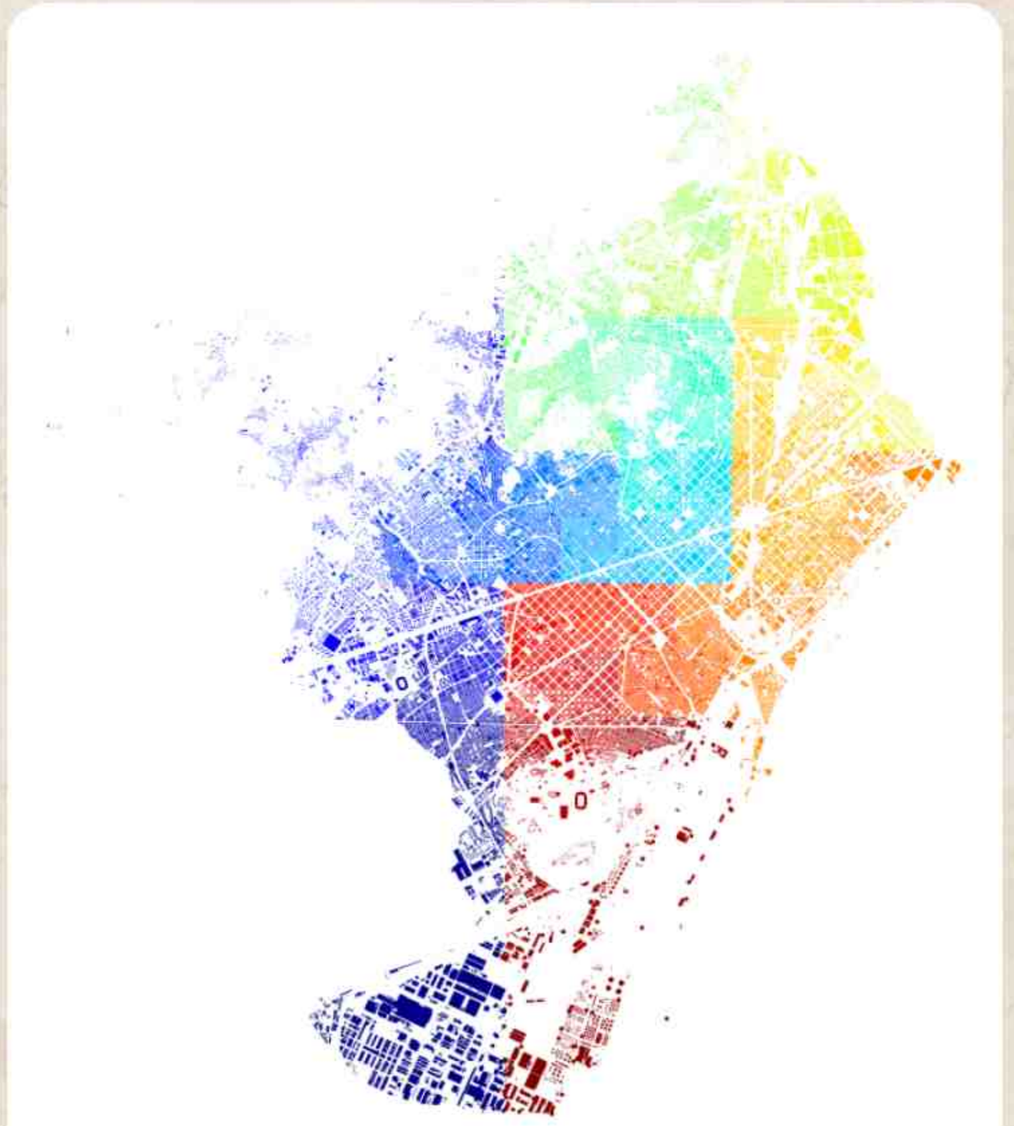
socials + ["https://www.facebook.com/341332855920

Overture Maps data

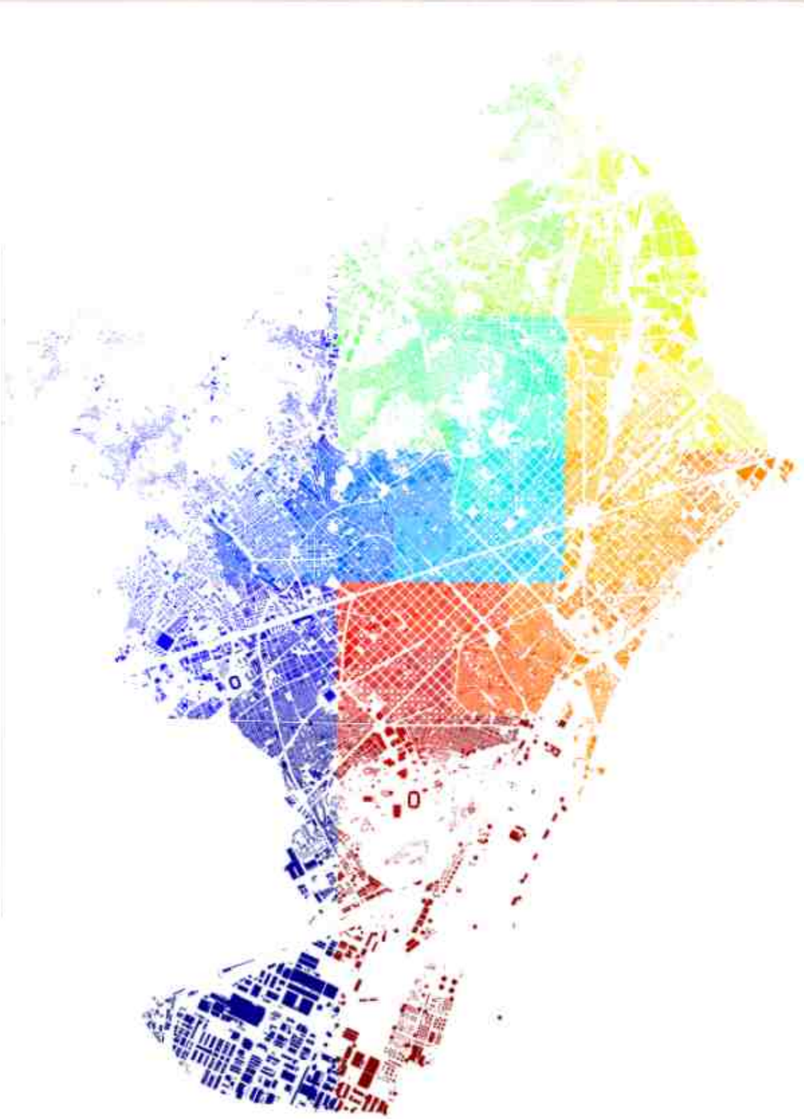
```
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00000-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00001-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00002-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00003-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00004-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00005-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00006-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00007-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00008-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00009-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00010-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00011-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00012-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00013-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00014-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00015-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00016-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00017-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00018-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
overturemaps-us-west-2/release/2025-06-25.0/theme=buildings/type=building/part-00019-8a741876-e04d-4e66-bc96-0171910fa1b1-c000.zstd.parquet
...
```


Data
Spatial Data
Corporate needs you to find the differences between this picture and this picture.
They're the same picture.
imgflip.com

Motivation

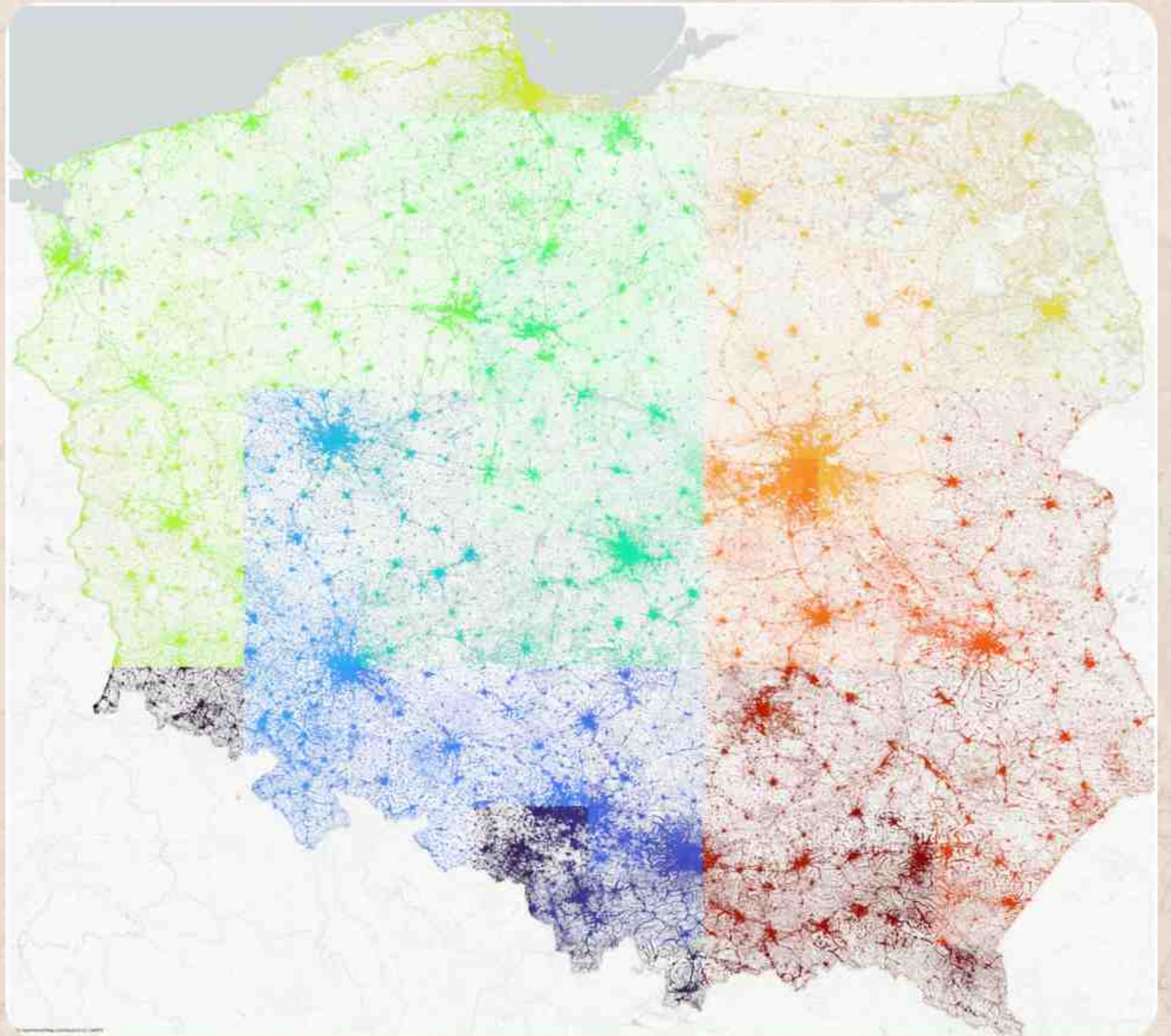


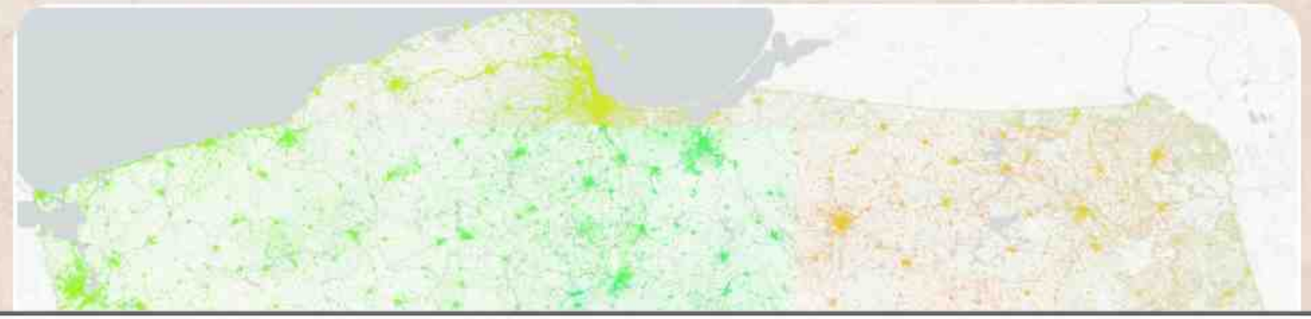

Brust




Brent


WAITING FOR OSMNX TO
FINISH DOWNLOADING DATA
MthruF.com

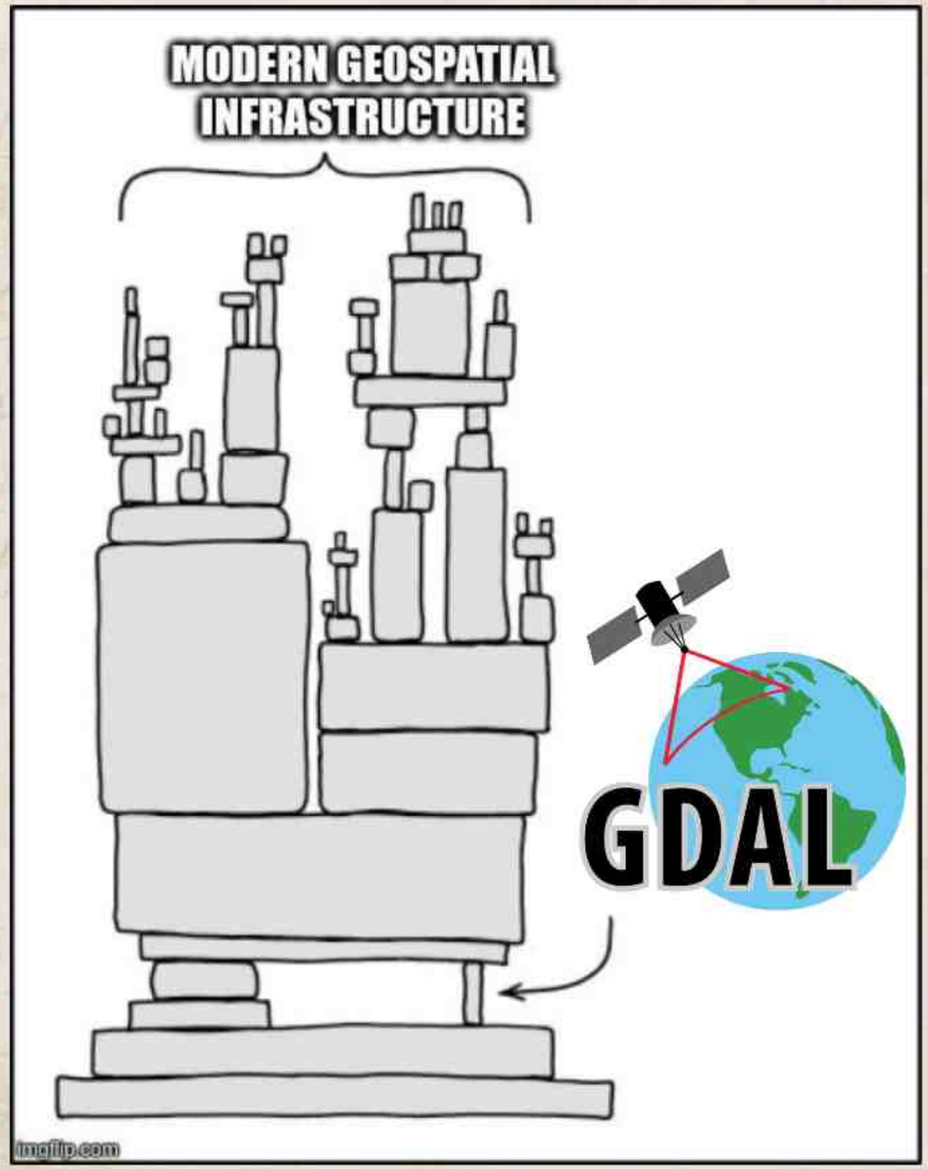




Kernel Restarting

The kernel appears to have died. It will restart automatically.

OK


MODERN GEOSPATIAL
INFRASTRUCTURE
GDAL
imgflip.com


ONE DOES NOT SIMPLY
PIP INSTALL GDAL
imgflip.com


PYOSMIUM
WAITING FOR OSMNX TO
FINISH DOWNLOADING DATA
MthruF.com

PARSING

Problem

OpenStreetMap data is hard to read at scale*

*in Python ecosystem

Ideal solution

Ideal solution

❌🧱 No GDAL / Spark

Ideal solution

❌🧱 **No GDAL / Spark**

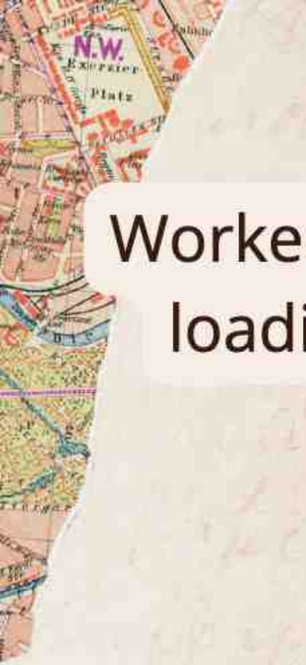
🖥️ **Runs on a single machine**

Ideal solution

❌🧱 No GDAL / Spark

🖥️ Runs on a single machine

⚡ Fast



Worked on improving loading OSM data

Mar 2023

Worked on improving loading OSM data

DuckDB spatial release

Mar 2023

Apr

Worked on improving loading OSM data

DuckDB spatial release

Added ST_ReadOSM

Mar 2023

Apr

Jun

Worked on improving loading OSM data

DuckDB spatial release

Added ST_ReadOSM

QuackOSM release

Mar 2023

Apr

Jun

Jul

Dec

Development time

Libraries

QuackOSM

OpenStreetMap

OvertureMaestro

Overture Maps

QuackOSM & OvertureMaestro

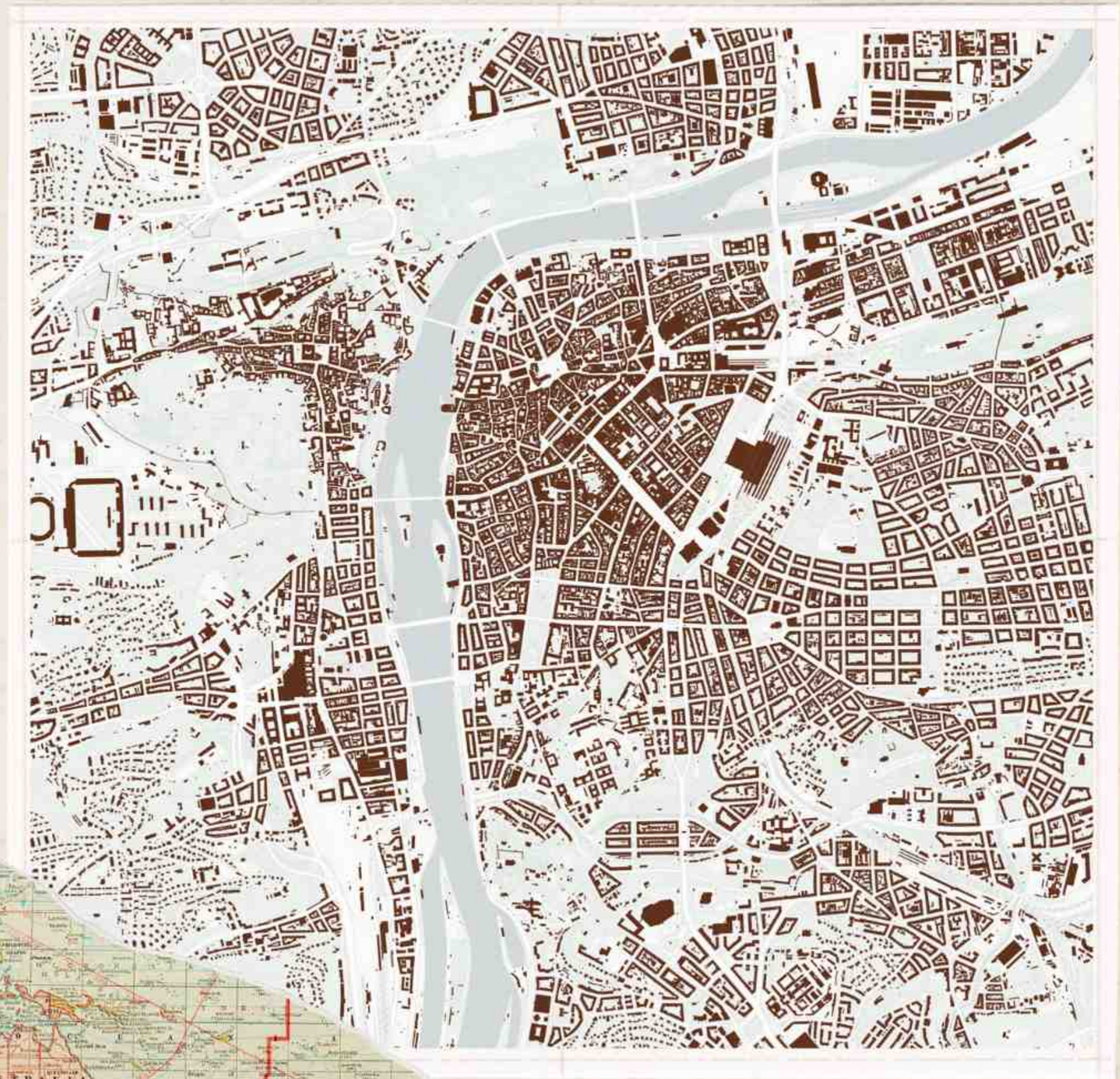
Get buildings from Prague

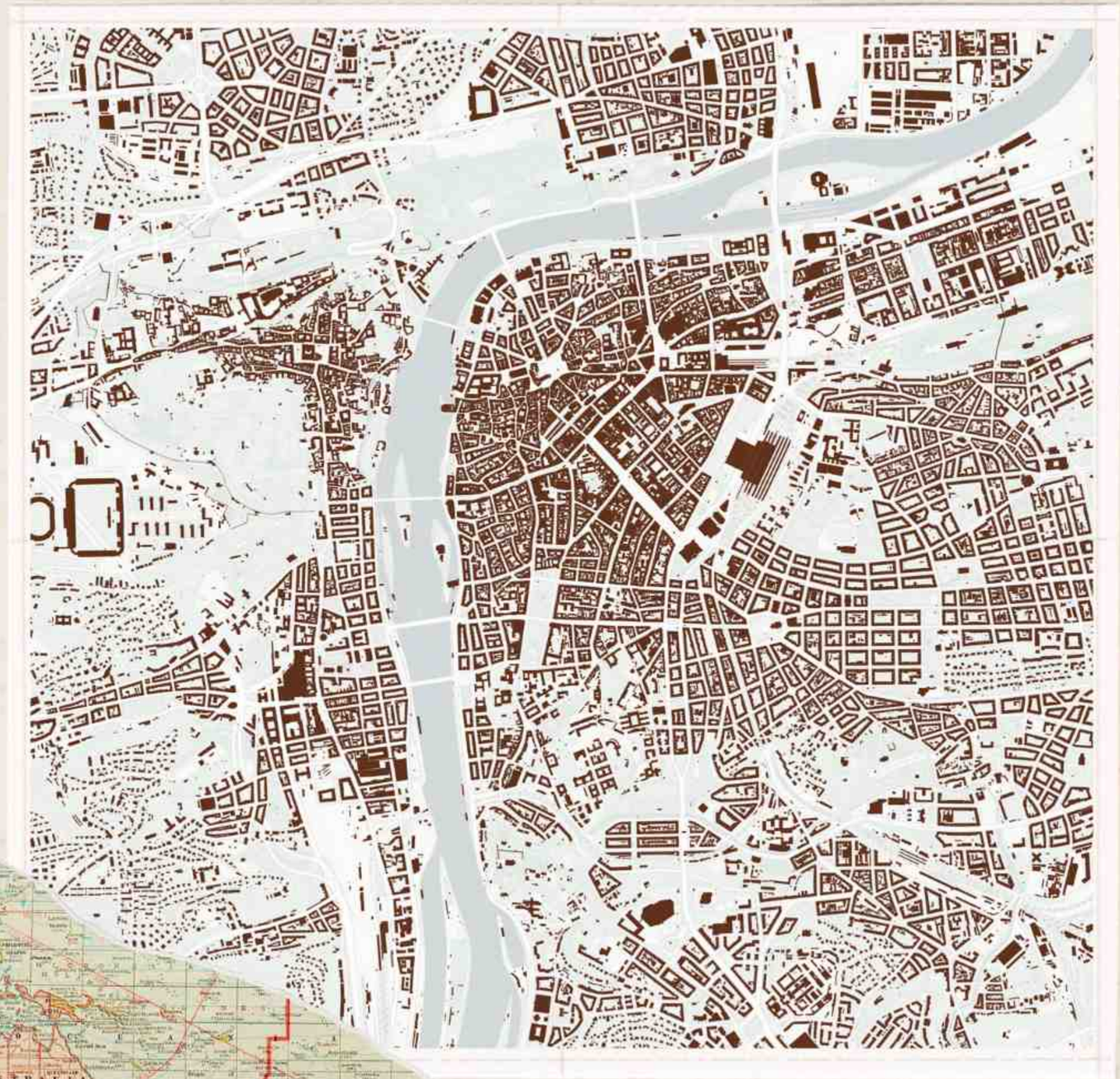
```python
import quackosm as qosm

qosm.convert_geometry_to_parquet(
    geometry_filter=qosm.geocode_to_geometry("Prague"),
    tags_filter={"building": True},
)
```

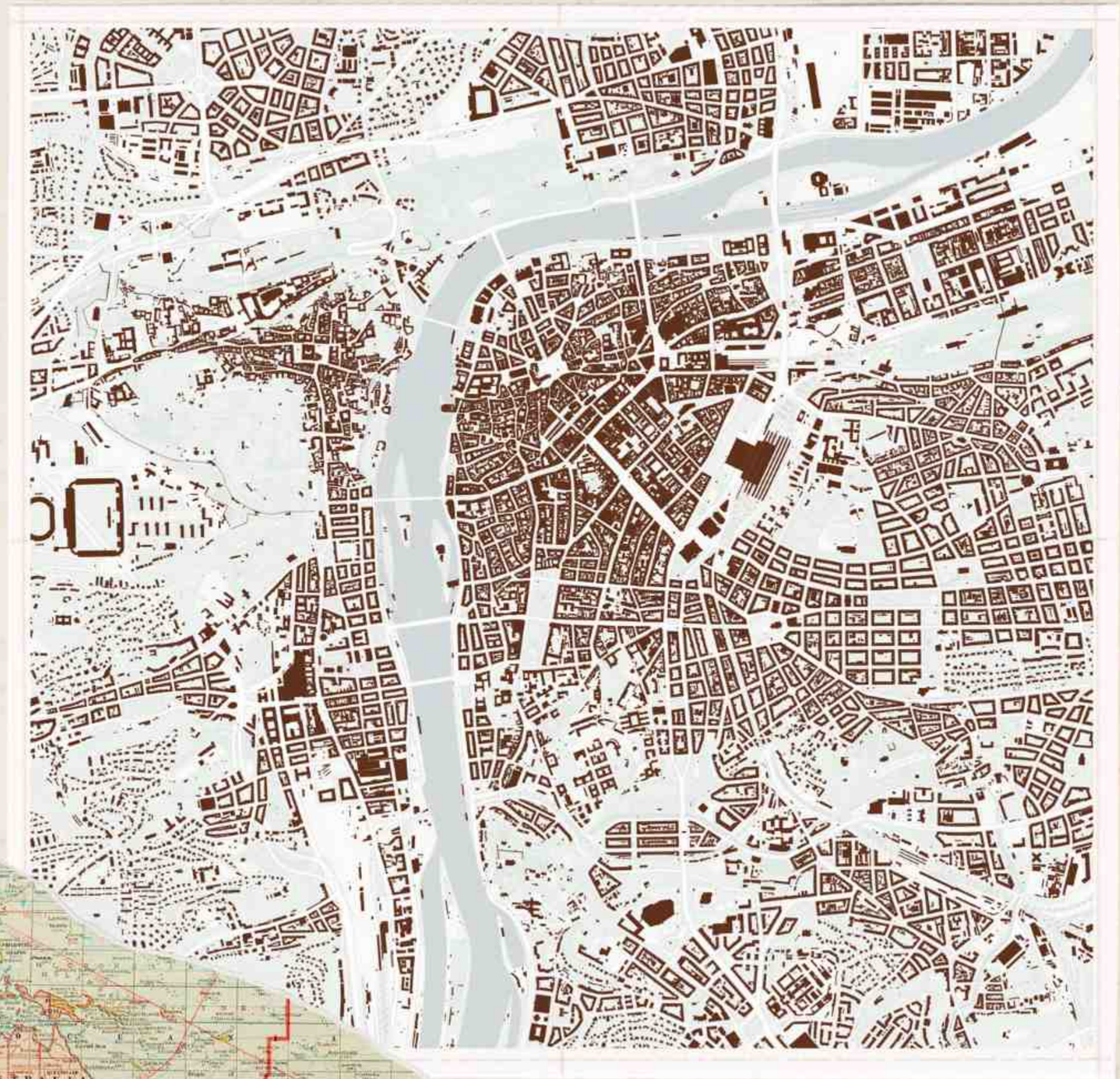
```python
import overturemaestro as om

om.convert_geometry_to_parquet(
    theme="buildings", type="building",
    geometry_filter=om.geocode_to_geometry("Prague"),
)
```



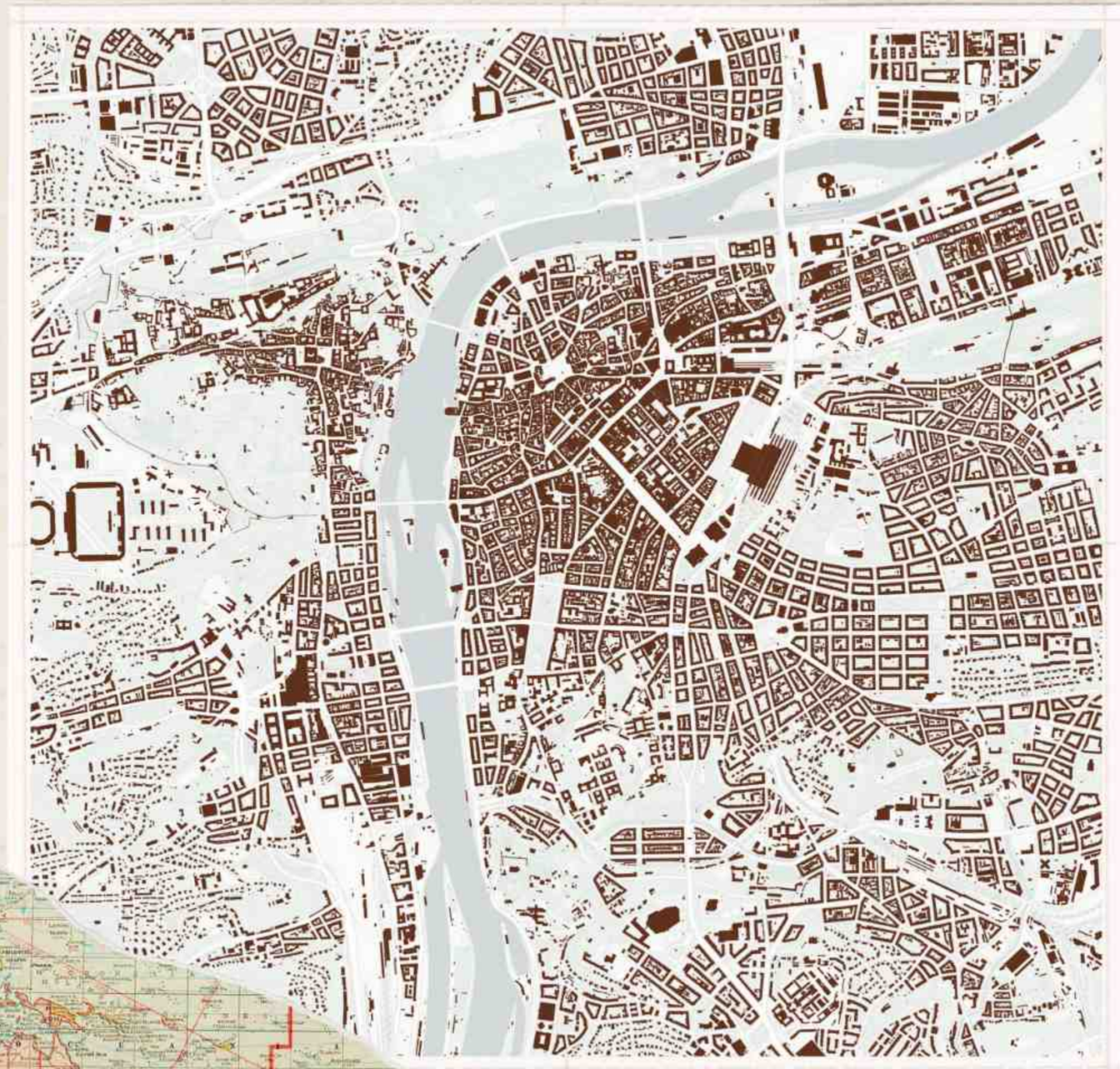


Key libraries features



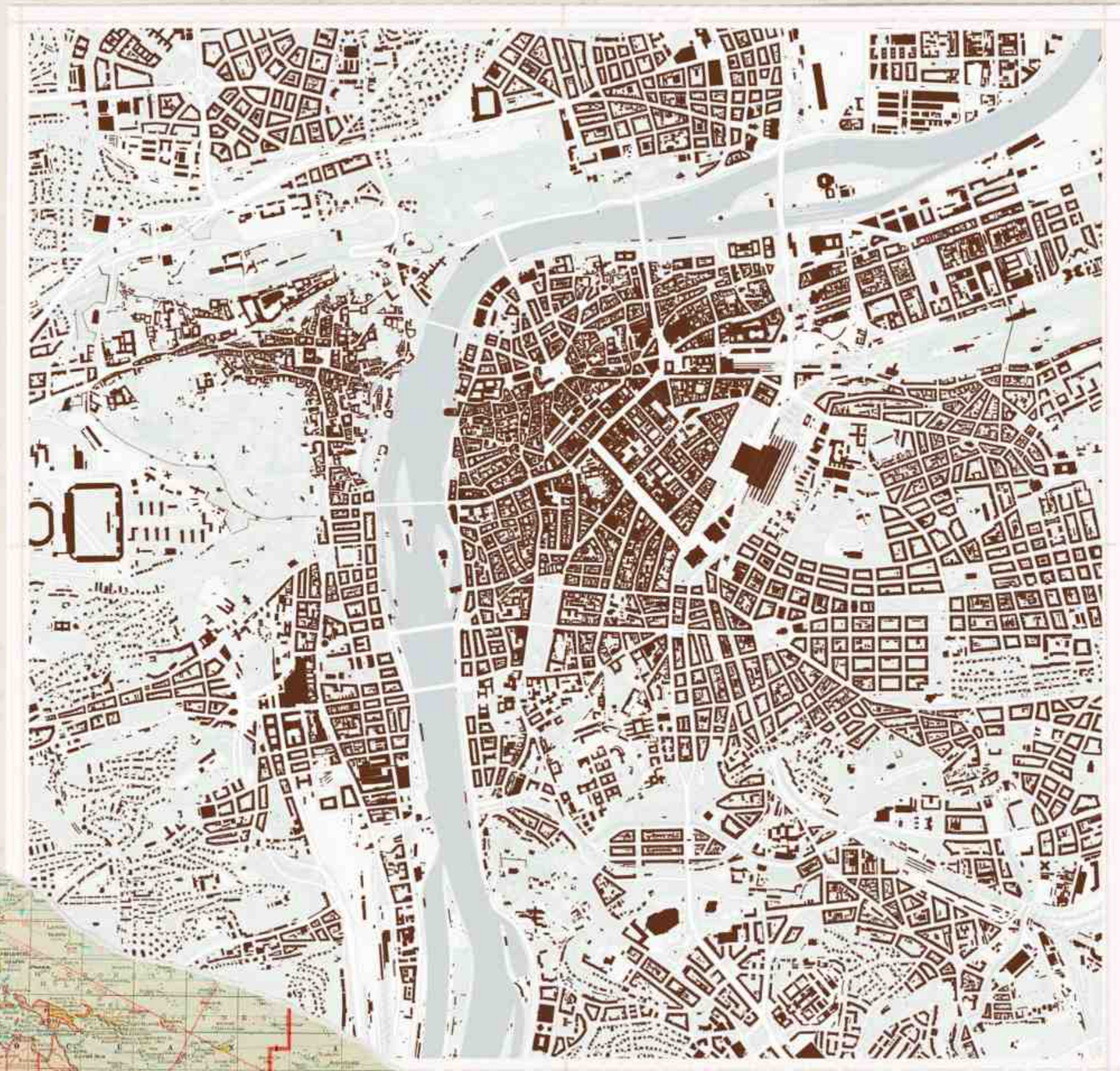
Key libraries features

- ⚙ Simple API & CLI



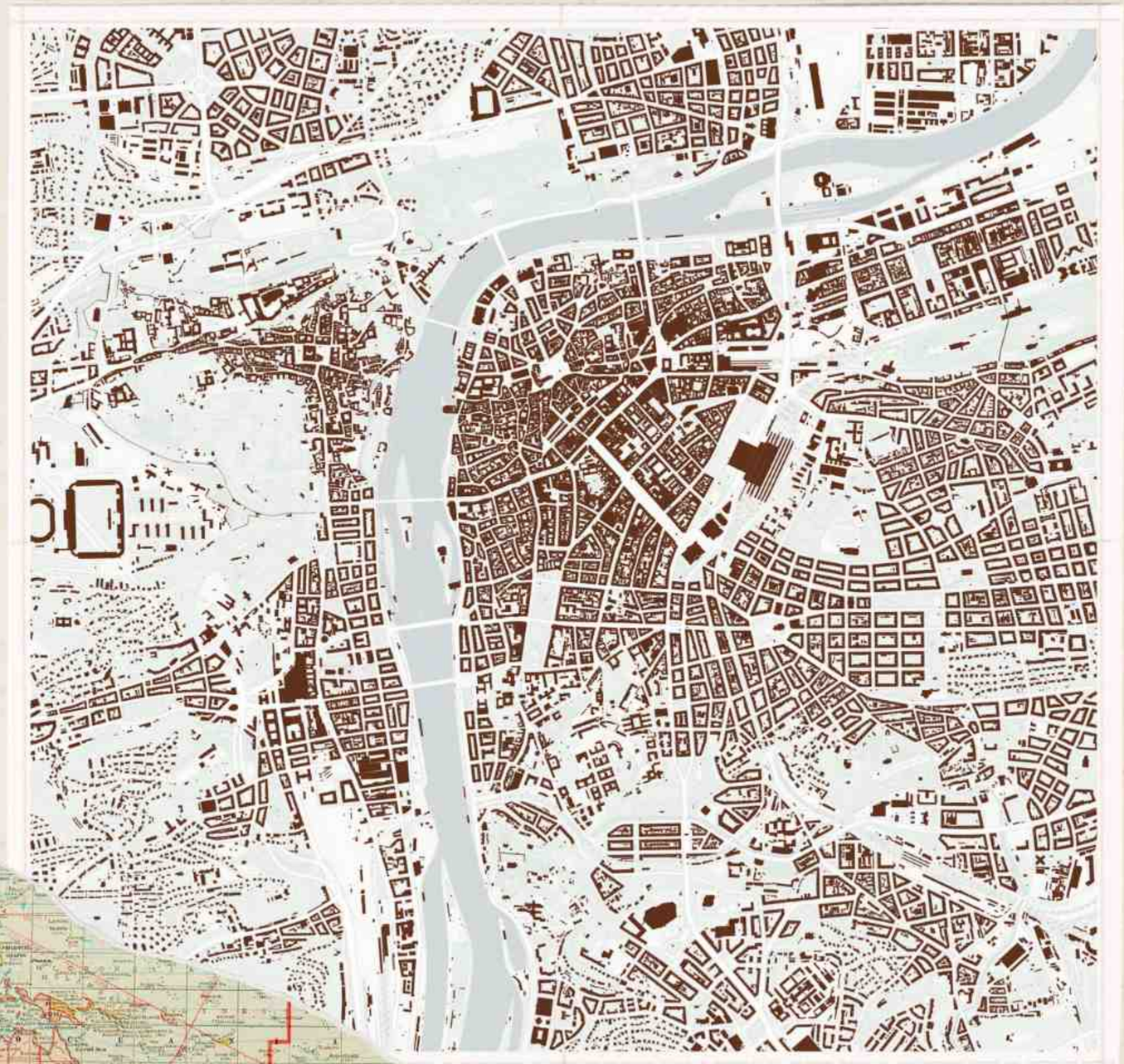
Key libraries features

- ⚙️ Simple API & CLI
- 💾 Saves as GeoParquet



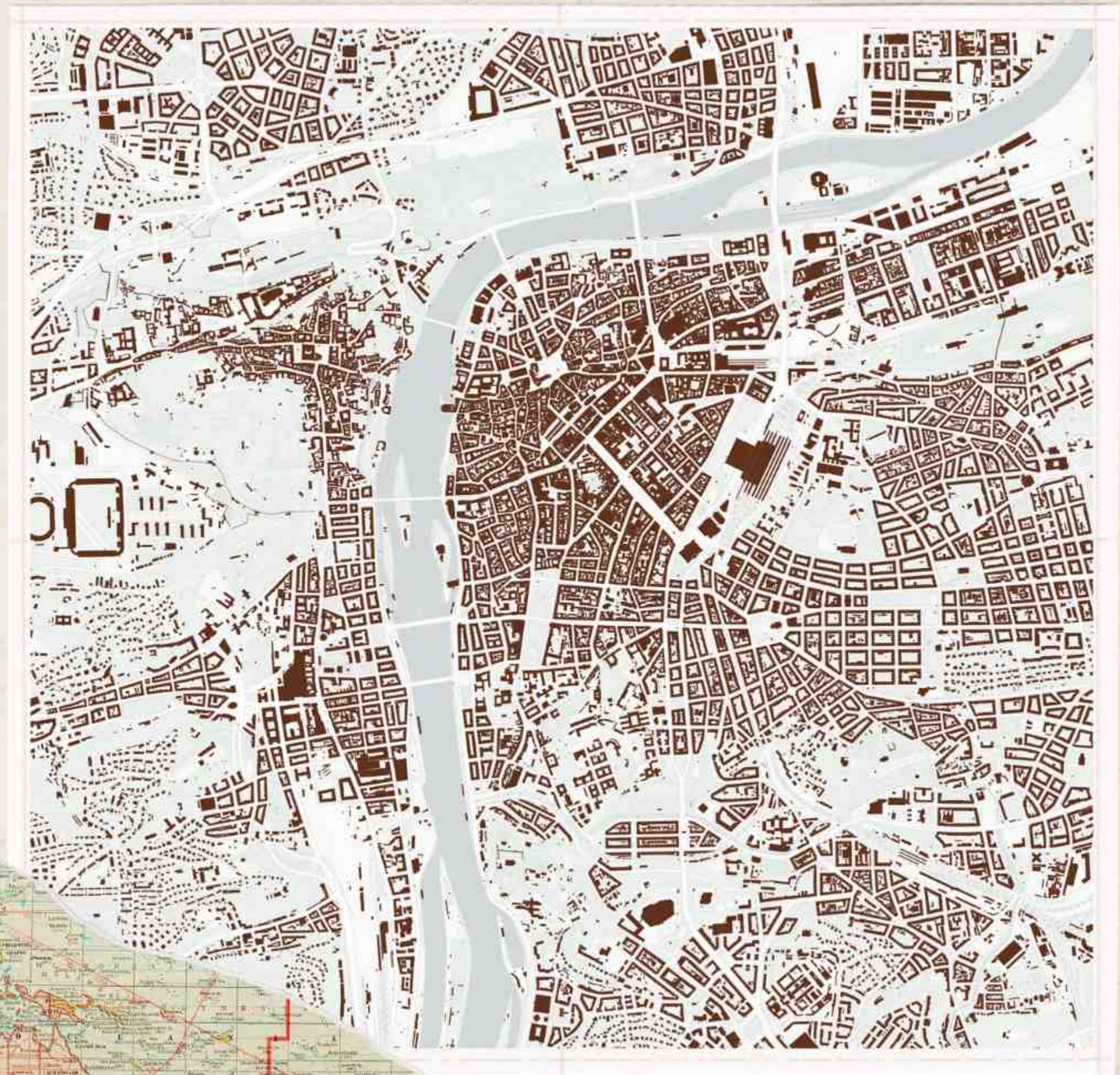
Key libraries features

- ⚙️ Simple API & CLI
- 💾 Saves as GeoParquet
- 🔗 Integrates with GeoPandas



Key libraries features

- ⚙️ Simple API & CLI
- 💾 Saves as GeoParquet
- 🔗 Integrates with GeoPandas
- 📐 Geometry-based queries



Key libraries features

- ⚙️ Simple API & CLI
- 💾 Saves as GeoParquet
- 🔗 Integrates with GeoPandas
- 📐 Geometry-based queries
- 🚀 Handles large regions

DuckDB

```
D INSTALL spatial;
D LOAD spatial;
D SELECT
      feature_id, building, geometry
  FROM
      read_parquet("prague_buildings.parquet")
  WHERE
      ST_DWithin_Spheroid(
          ST_Centroid(geometry), ST_Point(14.428745, 50.062123), 500
      )
LIMIT 5;
```

feature_id varchar	building varchar	geometry geometry
way/49852558	residential	POLYGON ((14.4287696 50.0574797, 14.4286896 50.0574962, 14.428758…
way/49852557	residential	POLYGON ((14.4290254 50.05769, 14.4289774 50.0575936, 14.4289737 …
way/49852511	residential	POLYGON ((14.4288186 50.0577467, 14.4288839 50.0578779, 14.428956…
way/49852512	residential	POLYGON ((14.4293305 50.0577261, 14.4290668 50.0577294, 14.429067…
way/49852514	residential	POLYGON ((14.4293305 50.0577275, 14.42933 50.0578726, 14.4295824 …

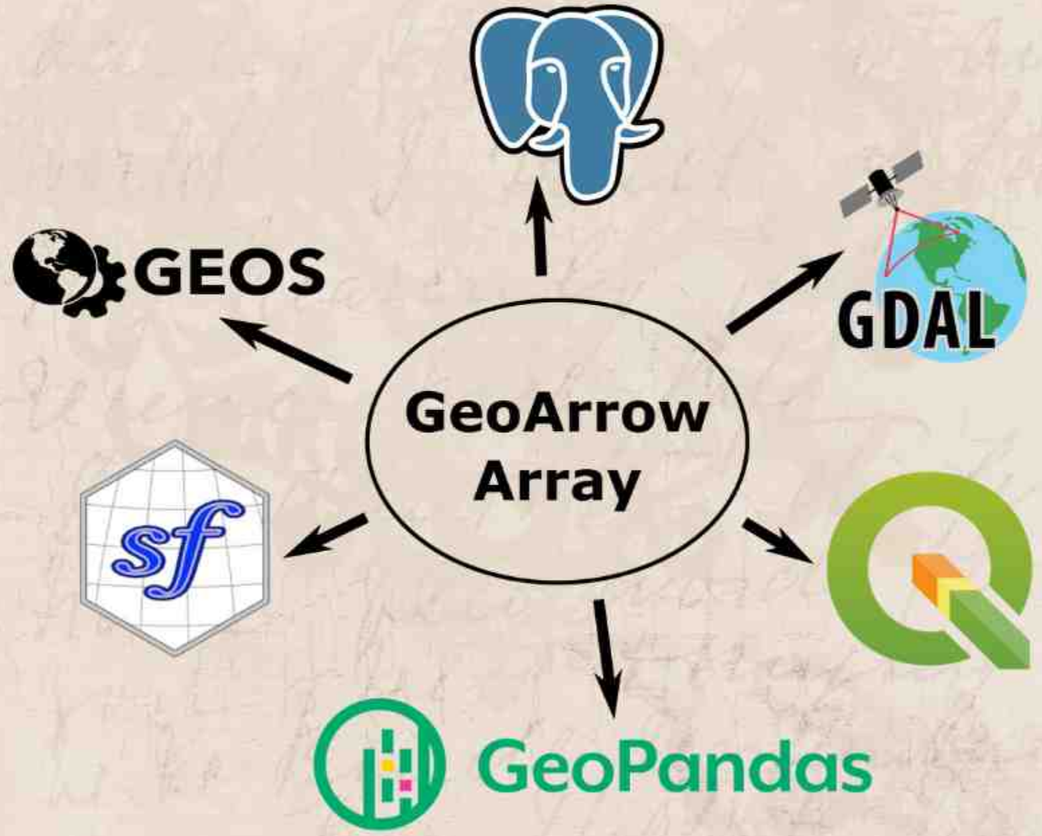
DuckDB

```
D SELECT *
  FROM ST_READOSM('praha.osm.pbf')
  WHERE kind = 'node'
  AND tags IS NOT NULL
  LIMIT 1;
     kind = node
       id = 12596614452
     tags = {highway=street_lamp}
     refs = NULL
      lat = 50.0342533
      lon = 14.5807804
ref_roles = NULL
ref_types = NULL
```

How to read OSM data with DuckDB | Towards Data Science

towardsdatascience.com

GeoArrow



Graphic from Joris Van den Bossche presentation

GeoArrow

```
>>> import pyarrow.parquet as pq
>>> tbl = pq.read_table('prague_buildings.parquet')

pyarrow.Table
feature_id: string
geometry: binary
  -- field metadata --
  ARROW:extension:name: 'geoarrow.wkb'
  ARROW:extension:metadata: '{"crs": "{\"$schema\":\"https://proj.org/sch' + 1479
building: string
-- schema metadata --
geo: '{"version":"1.1.0","primary_column":"geometry","columns":{"geometry' + 127
----
feature_id: [["way/166135774","way/465649732","way/104625751","way/55798658","way/207291670",...
geometry: [[0103000000010000000500000677BF486FB982C4074B7EBA52900494079B29B19FD982C408C135FE...
building: [["yes","residential","yes","farm_auxiliary","yes",...,"residential","residential",...
```

This is optional

This is required

GeoArrow

```
>>> from geoarrow.pyarrow import io
>>> tbl = io.read_geoparquet_table('prague_buildings.parquet')

pyarrow.Table
feature_id: string
geometry: extension<geoarrow.wkb<WkbType>>
building: string
----
feature_id: [["way/166135774","way/465649732","way/104625751","way/55798658","way/207291670",...
geometry: [[0103000000010000000500000077BF486FB982C4074B7EBA52900494079B29B19FD982C408C135FE...
building: [["yes","residential","yes","farm_auxiliary","yes",...,"residential","residential",...
```

Dependency stack

DuckDB

Pooch

Polars

GeoPandas

Typer

APACHE ARROW

GeoPy

QuackOSM

QuackOSM

Installs without any external dependencies

Just run any of these commands - say goodbye to installing GDAL

```
pip install quackosm[cli]
```

```
uv add quackosm[cli]
```

```
conda install conda-forge::quackosm
```

QuackOSM

Transforms PBF files into GeoParquet format without memory limitations

```
$ QuackOSM --osm-extract-query Poland --osm-extract-source Geofabrik
Finished operation in 0:21:37
files/geofabrik_europe_poland_nofilter_noclip_compact_sorted.parquet
```

Library automatically offloads bigger-than-memory transformations to disk, using DuckDB

QuackOSM

Transforms PBF files into GeoParquet format without memory limitations

Existing file → `convert_pbf_to_parquet()`

Find by geometry → `convert_geometry_to_parquet()`

Find by string query → `convert_osm_extract_to_parquet()`

Available base functions

QuackOSM

Integrates with GeoPandas and DuckDB

```
convert_pbf_to_geodataframe()
convert_geometry_to_geodataframe()
convert_osm_extract_to_geodataframe()
```

```
convert_pbf_to_duckdb()
convert_geometry_to_duckdb()
convert_osm_extract_to_duckdb()
```

QuackOSM

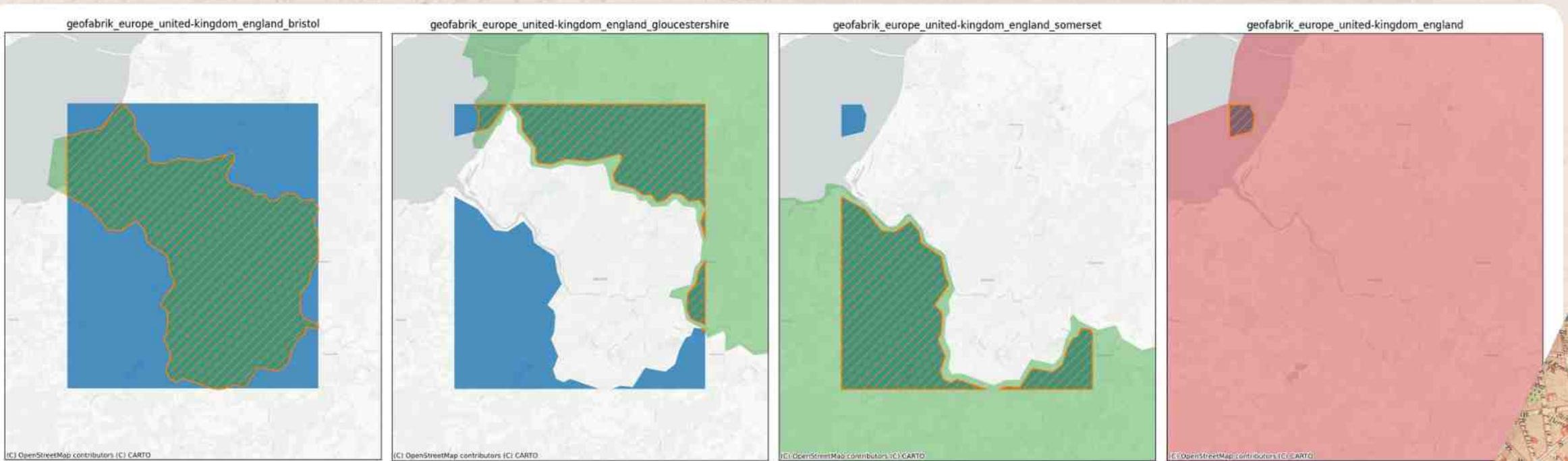
Can be run as CLI

```
$ QuackOSM -h
Usage: QuackOSM [OPTIONS] PBF file path
```

```
$ QuackOSM --geom-filter-geocode Prague
  --osm-tags-filter '{"building": true}'
```

QuackOSM

Automatically matches PBF extracts for a given geometry from multiple sources



QuackOSM

Can filter data based on geometry and tags

```python
import quackosm as qosm

qosm.convert_pbf_to_parquet( # filter features from file
    pbf_path="czech-republic-latest.osm.pbf",
    geometry_filter=qosm.geocode_to_geometry("Prague")
)

qosm.convert_geometry_to_parquet( # automatically filters
    geometry_filter=qosm.geocode_to_geometry("Prague")
)
```

QuackOSM

Can filter data based on geometry and tags

```python
from quackosm import convert_pbf_to_parquet

convert_pbf_to_parquet( # bars and cafes 🍻 & ☕
    pbf_path="czech-republic-latest.osm.pbf",
    tags_filter={"amenity": ["bar", "cafe"]},
)

convert_pbf_to_parquet( # local ice cream shops 🍦
    pbf_path="czech-republic-latest.osm.pbf",
    tags_filter={"amenity": "ice_cream", "brand": False},
)
```

QuackOSM

Can filter data based on geometry and tags

```python
import quackosm as qosm

qosm.convert_pbf_to_geodataframe(
    pbf_path="czech-republic-latest.osm.pbf",
    tags_filter={"name:*": "*Astronom*"},
)
```

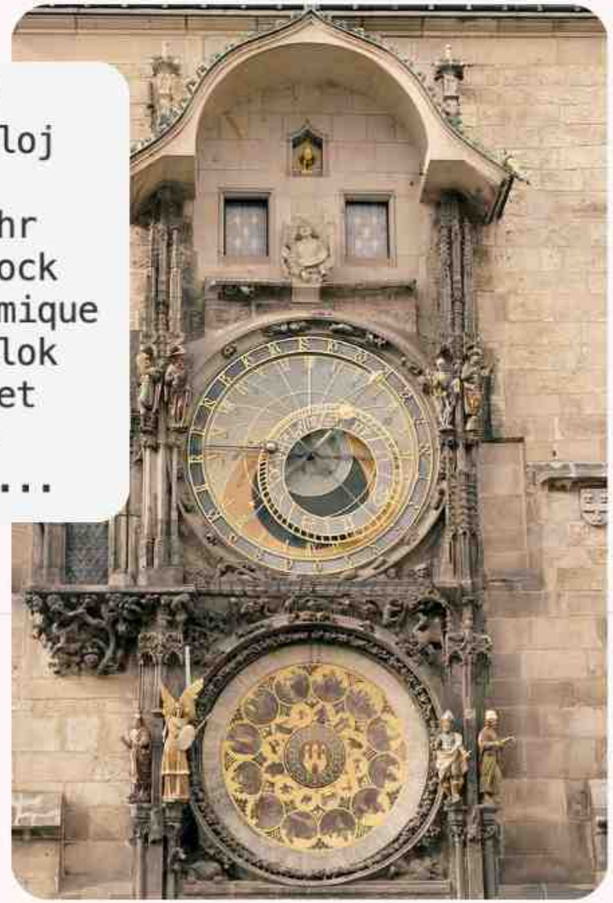
feature_id	name:cs	name:de	...	name:tr	geometry
way/382636013	Astronomická věž	Astronomischer Turm	...	None	POLYGON ((14.416...
node/4748580936	None	Astronomische Uhr	...	Astronomik Saat	POINT (14.42071 ...

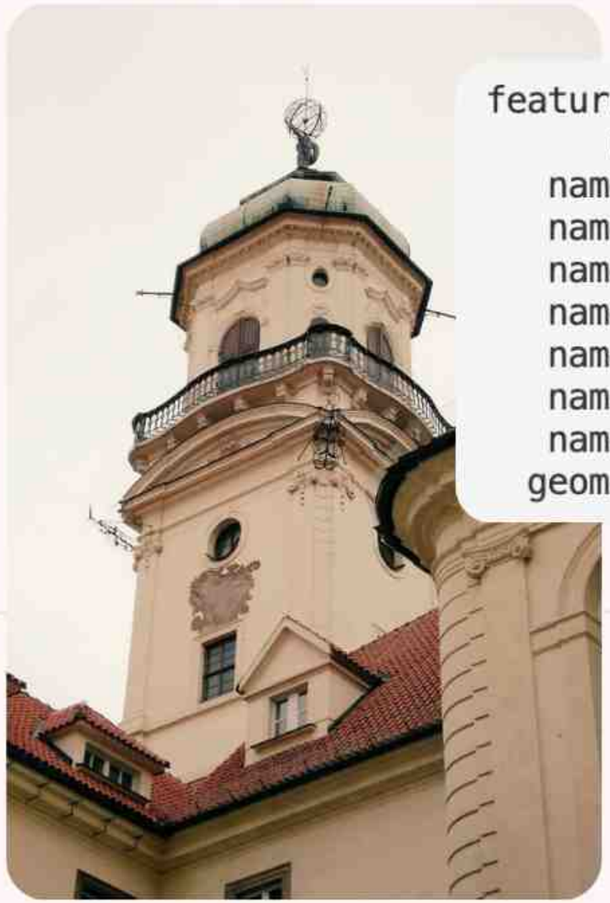
2 rows × 8 columns

QuackOSM

Can filter data based on geometry and tags

```
feature_id = node/4748580936
      name = Staroměstský orloj
   name:cs = NULL
   name:de = Astronomische Uhr
   name:en = Astronomical Clock
   name:fr = Horloge Astronomique
   name:nl = Astronomische Klok
   name:sv = Astronomiska uret
   name:tr = Astronomik Saat
geometry = POINT (14.42071...
```





```
feature_id = way/382636013
      name = Astronomická věž
   name:cs = Astronomická věž
   name:de = Astronomischer Turm
   name:en = Astronomical Tower
   name:fr = NULL
   name:nl = NULL
   name:sv = NULL
   name:tr = NULL
geometry = POLYGON ((14.41...
```

QuackOSM

Alternatives 1/2

esy.osm.shape	PBF	Protobuf reader
OSMnx	Overpass	Most popular library
PyDriosm	PBF	GDAL-based
PyOsmium	PBF	C++ reader
Pyrosm	PBF	Cython reader

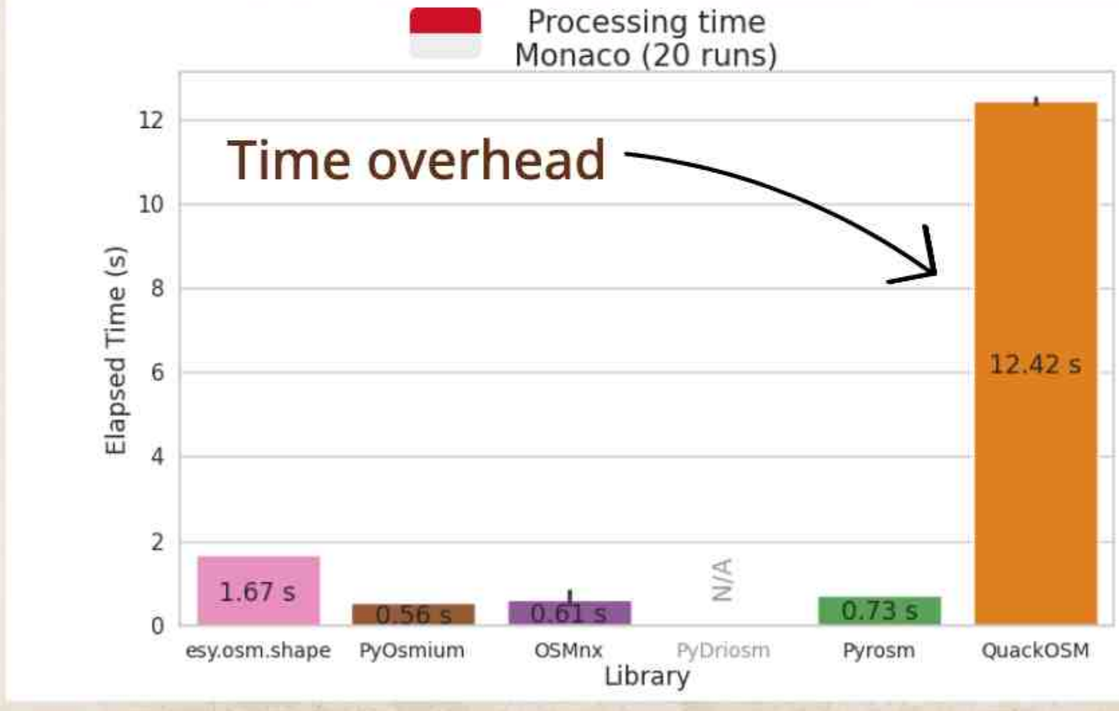
QuackOSM

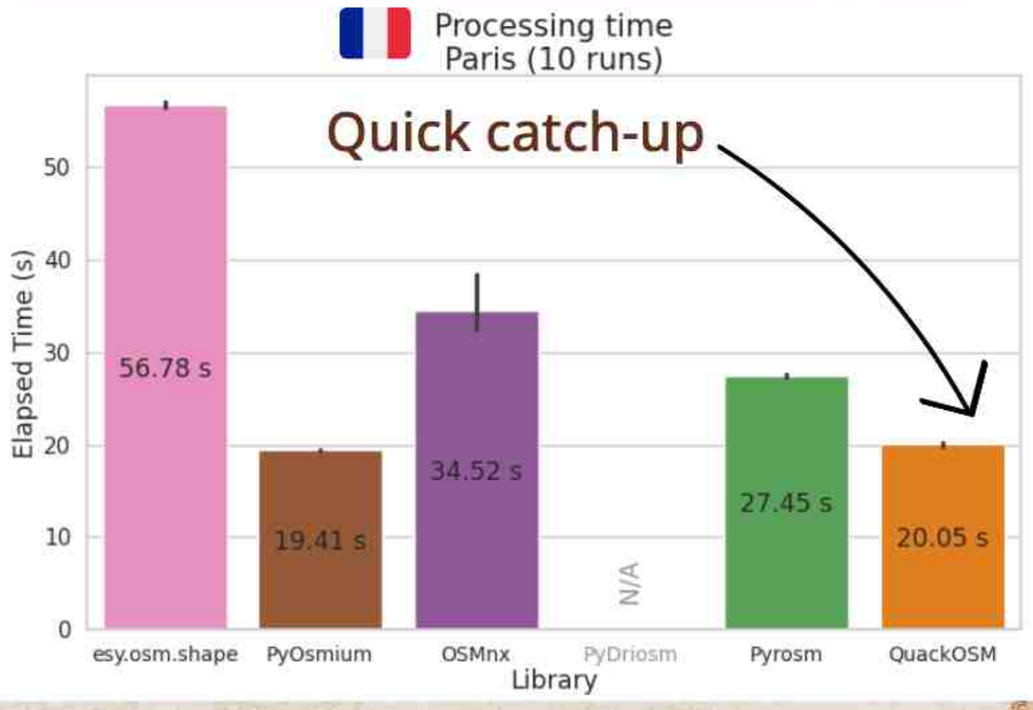
Alternatives 2/2

Apache Sedona	PBF	Spark-based PBF file reader (v1.7.1)
LayerCake	GeoParquet	Buildings & highways only (for now)

QuackOSM

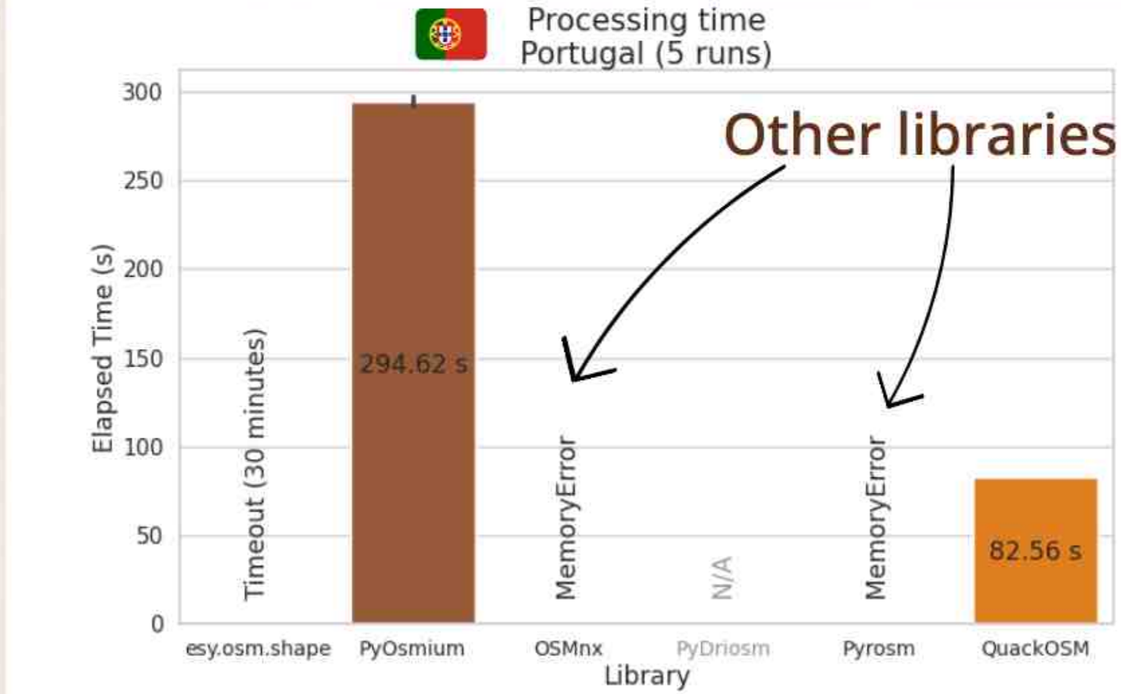
Benchmark - read buildings (processing time) 1/2

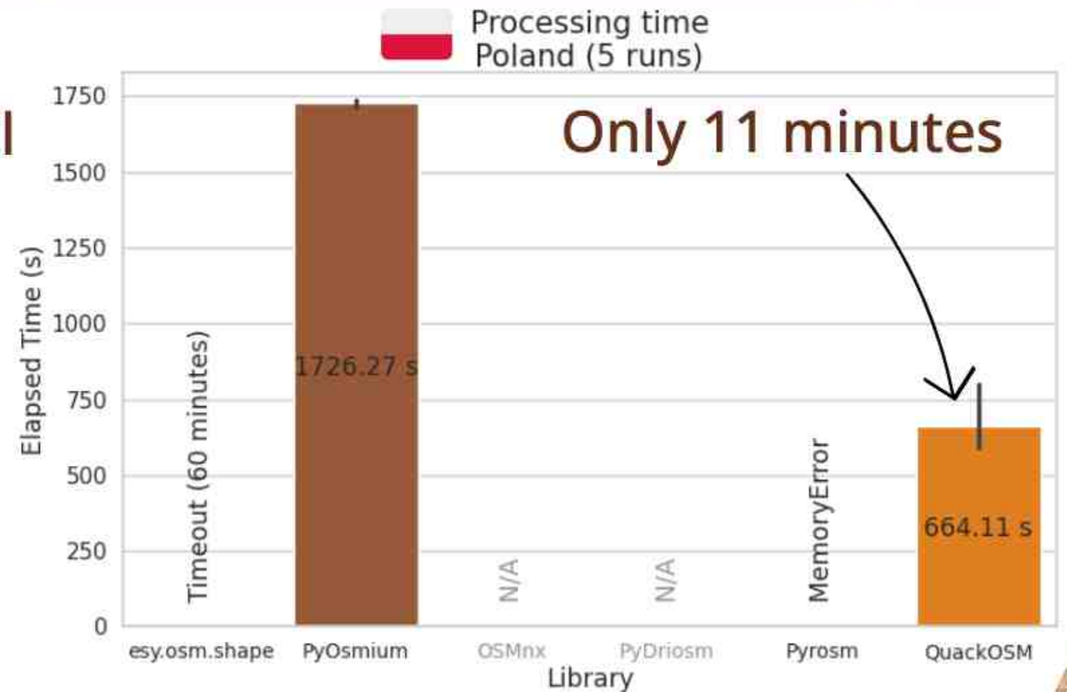




QuackOSM

Benchmark - read buildings (processing time) 2/2





RaczeQ/**osm-python-readers-benchmark**

1 Contributor

0 Issues

1 Star

0 Forks

GitHub - RaczeQ/osm-python-readers-benchmark

github.com

OvertureMaestro

OvertureMaestro

Installs without any external dependencies

Library is available on PyPI

```
pip install overturemaestro[cli]
```

```
uv add overturemaestro[cli]
```

OvertureMaestro

Returns data as GeoParquet or GeoDataFrame

```
convert_bounding_box_to_geodataframe()
convert_bounding_box_to_parquet()
convert_geometry_to_geodataframe()
convert_geometry_to_parquet()
convert_bounding_box_to_geodataframe_for_multiple_types()
convert_bounding_box_to_parquet_for_multiple_types()
convert_geometry_to_geodataframe_for_multiple_types()
convert_geometry_to_parquet_for_multiple_types()
```

OvertureMaestro

Returns data as GeoParquet or GeoDataFrame

```python
import overturemaestro as om

om.convert_geometry_to_geodataframe(
    theme="buildings", type="building", geometry_filter=om.geocode_to_geometry("Prague")
)

om.convert_bounding_box_to_parquet_for_multiple_types(
    theme_type_pairs=[("buildings", "building"), ("transportation", "segment")],
    bbox=(14.2244355, 49.9419006, 14.7067867, 50.1774301)
)
```

🏘️ & 🛣️

OvertureMaestro

Returns data as GeoParquet or GeoDataFrame

	geometry	bbox	subtype	class	...	version	names	height	has_parts
id									
b5ad109d-...	POLYGON ((14.368...	{'xmin': 14.3677...	residential	apartments	...	1	None	25.0	False
e033f837-...	POLYGON ((14.428...	{'xmin': 14.4288...	commercial	commercial	...	1	{'primary': 'Lid...	6.0	False
825e040f-...	POLYGON ((14.555...	{'xmin': 14.5552...	outbuilding	roof	...	1	None	2.8	False
ae617863-...	POLYGON ((14.372...	{'xmin': 14.3709...	commercial	warehouse	...	1	{'primary': 'Kon...	12.0	False
2a6233db-...	POLYGON ((14.403...	{'xmin': 14.4028...	religious	synagogue	...	1	{'primary': 'Smí...	8.0	True
...	...	...	...	...	...	...	...	...	...
46f17b8d-...	POLYGON ((14.440...	{'xmin': 14.4407...	service	service	...	1	None	4.0	False
8061aba6-...	POLYGON ((14.472...	{'xmin': 14.4724...	residential	garage	...	1	None	2.5	False
42936214-...	POLYGON ((14.407...	{'xmin': 14.4074...	religious	chapel	...	1	{'primary': 'Mod...	8.0	False
40cfb70e-...	POLYGON ((14.407...	{'xmin': 14.4067...	religious	church	...	1	{'primary': 'Nan...	12.0	True
2879f6d4-...	POLYGON ((14.487...	{'xmin': 14.4872...	civic	civic	...	1	None	7.0	False

839 rows × 24 columns

OvertureMaestro

Can be run as CLI

```
$ OvertureMaestro buildings building --geom-filter-geocode Prague
  Downloading parquet files (buildings/building) 100% ━━━━━━━━ 5/5 • 0:00:11 < 0:00:00 • 3.36 it/s
⠸ Preparing multiprocessing environment • 0:00:00
  Reading all parquet files row groups 100% ━━━━━━━━━━━━━━━━ 5/5 • 0:00:00 < 0:00:00 • 0 it/s
  Filtering data by geometry (buildings/building) 100% ━━━━━━━━ 5/5 • 0:00:01 < 0:00:00 • 5.92 it/s
⠸ Saving final geoparquet file • 0:00:00
⠼ Sorting result file by geometry • 0:00:04
Finished operation in 0:00:19
files/2025-06-25.0/theme=buildings/type=building/36ddb61c_nofilter_sorted.parquet
```

OvertureMaestro

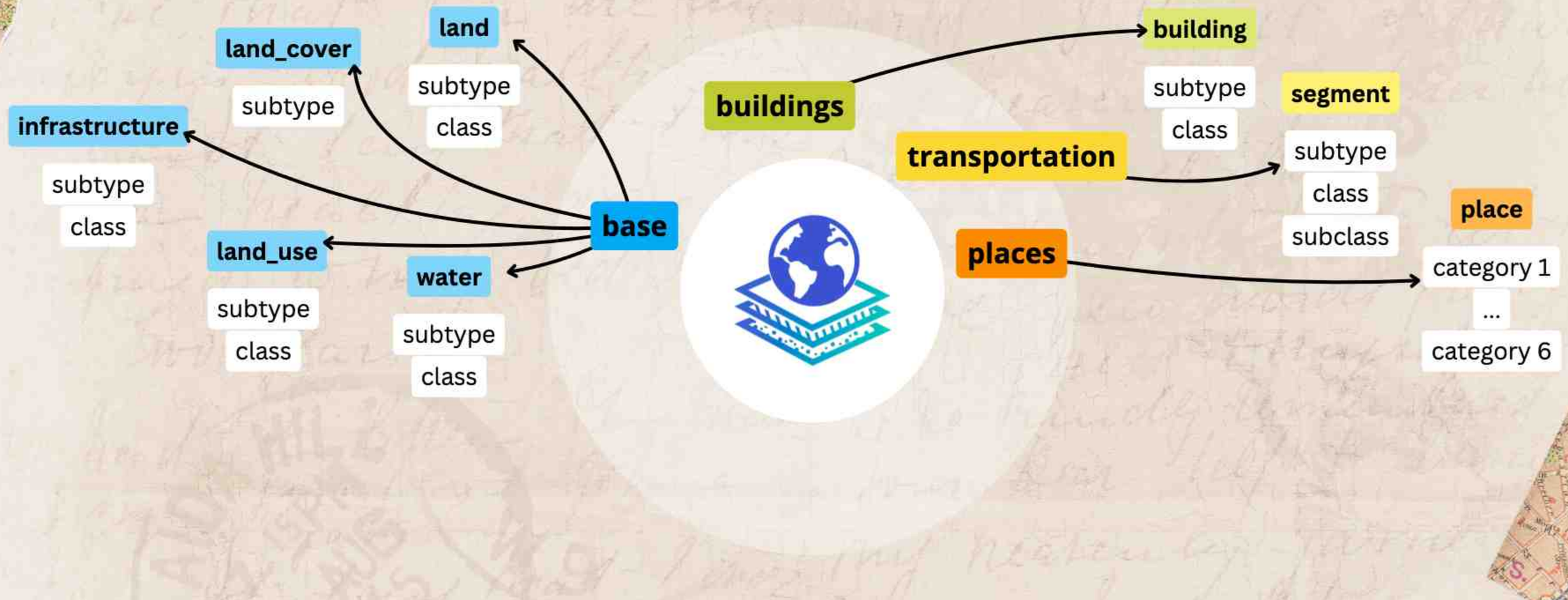
Can filter data using pyarrow expressions and select columns


```python
import overturemaestro as om

om.convert_bounding_box_to_geodataframe(  # select rivers
    theme="base",
    type="water",
    bbox=(14.2244355, 49.9419006, 14.7067867, 50.1774301),
    pyarrow_filter=[("subtype", "=", "river")],
    # id and geometry columns are automatically added
    columns_to_download=["names", "sources"],
)
```

OvertureMaestro

Includes dedicated functions for a unified combined file format



OvertureMaestro

Includes dedicated functions for a unified combined file format

```
{
    "id": "08b3969a40291fff0004b100d54c0d4d",
    "geometry": "<POLYGON ((7.418 43.726, 7.418 43.725, 7.418 43....>",
    "base|water|human_made|swimming_pool": true,
    "base|water|ocean|ocean": false,
    "base|water|physical|bay": false,
    "base|water|pond|pond": false,
    "base|water|reservoir|basin": false,
    "base|water|stream|stream": false,
    "base|water|water|water": false
}
```

ID

theme

type

subtype

class

```
from overturemaestro import geocode_to_geometry
from overturemaestro.advanced_functions import (
    convert_geometry_to_wide_form_geodataframe,
)

convert_geometry_to_wide_form_geodataframe(
    "base", "water", geocode_to_geometry("Monaco")
)
```

OvertureMaestro

Includes dedicated functions for a unified combined file format

	geometry	base|infrastructure	base|land	...	buildings|building	places|place	transportation|segment
id							
ae663fa4-...	POINT (14.49107 ...	True	False	...	False	False	False
db4cef76-...	POLYGON ((14.507...	False	False	...	True	False	False
49c86206-...	POLYGON ((14.407...	False	False	...	True	False	False
e6e48b74-...	POLYGON ((14.486...	False	False	...	True	False	False
db5ba18f-...	LINESTRING (14.3...	False	False	...	False	False	True
...	...	...	...	...	...	...	...
d054db8b-...	LINESTRING (14.5...	False	False	...	False	False	True
fdf65ad9-...	LINESTRING (14.4...	False	False	...	False	False	True
1ff764df-...	POINT (14.46998 ...	True	False	...	False	False	False
dff008bb-...	POINT (14.26968 ...	False	False	...	False	True	False
be7b7417-...	LINESTRING (14.3...	False	False	...	False	False	True

603109 rows × 9 columns

OvertureMaestro

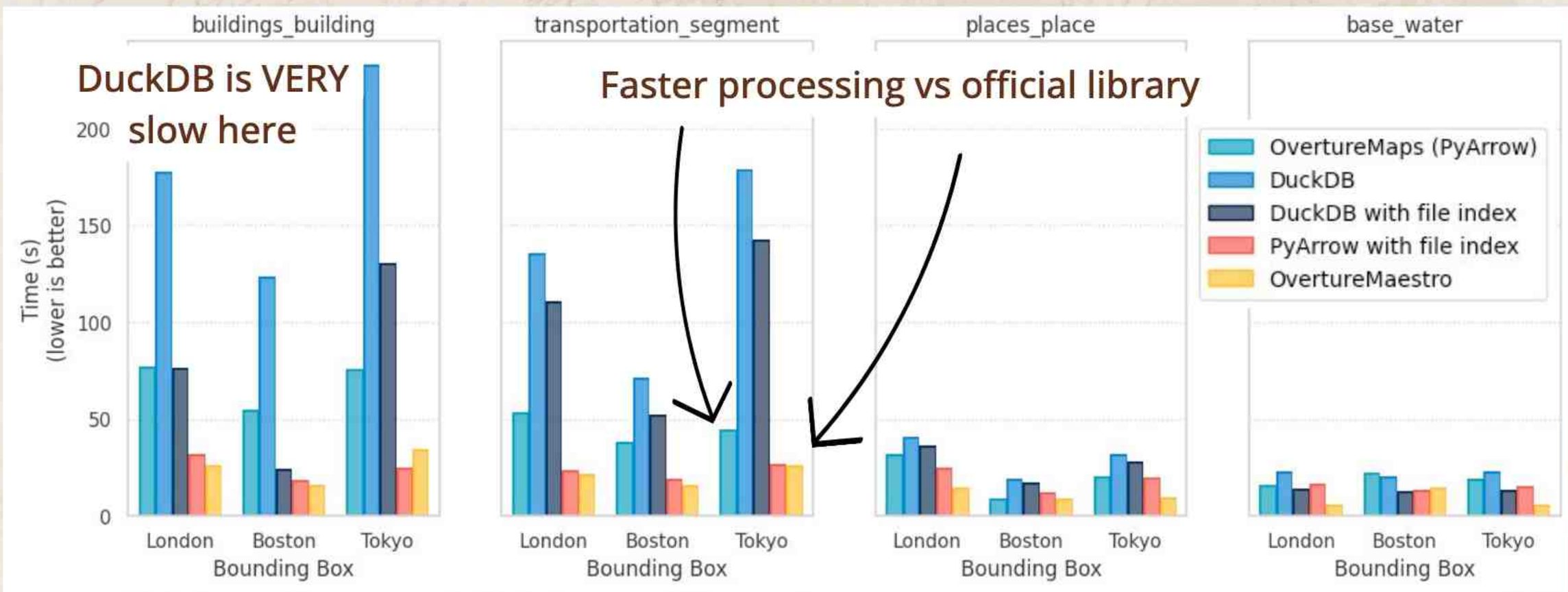
Alternative ways to get Overture Maps data

overturemaps-py	Download data for a bounding box
DuckDB	Query data directly from the public S3 bucket
Pure PyArrow dataset	Query data directly from the public S3 bucket
BigQuery / Snowflake / Databricks	Datasets are available directly on those platforms

https://docs.overturemaps.org/getting-data/

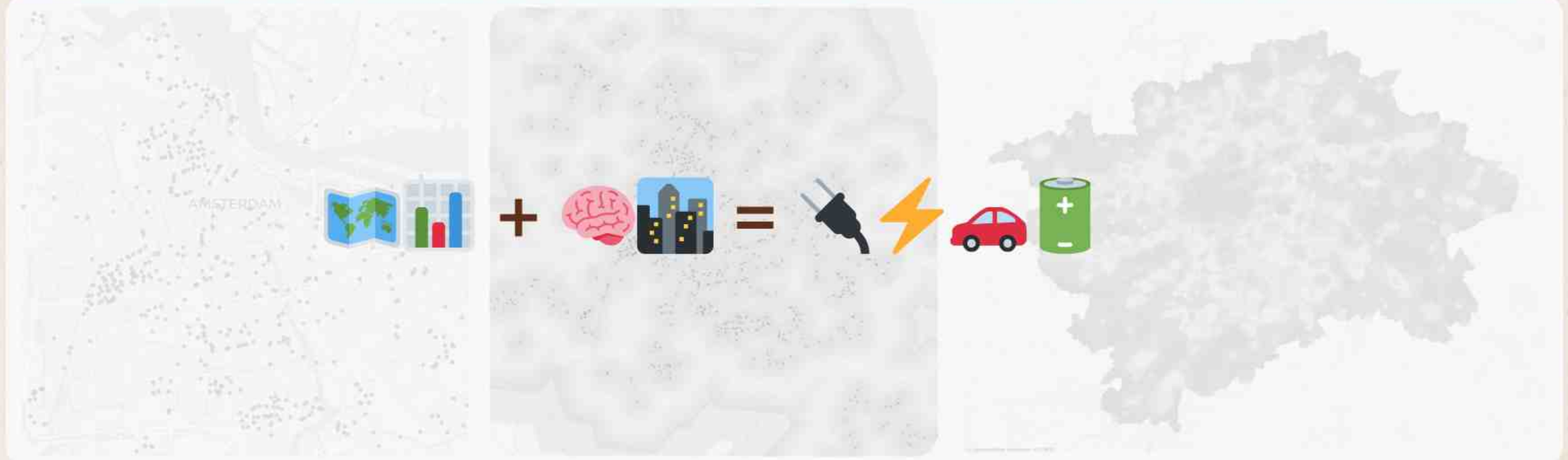
OvertureMaestro

Benchmark - processing time



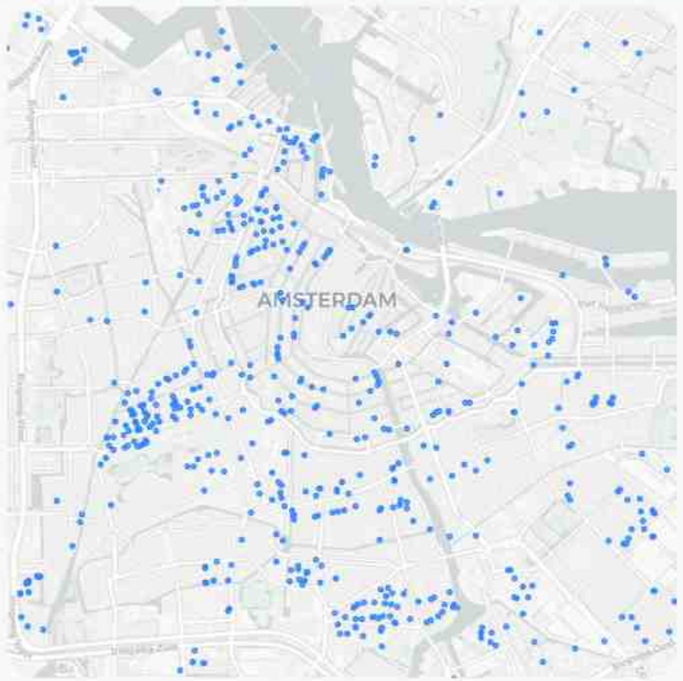
Example use case

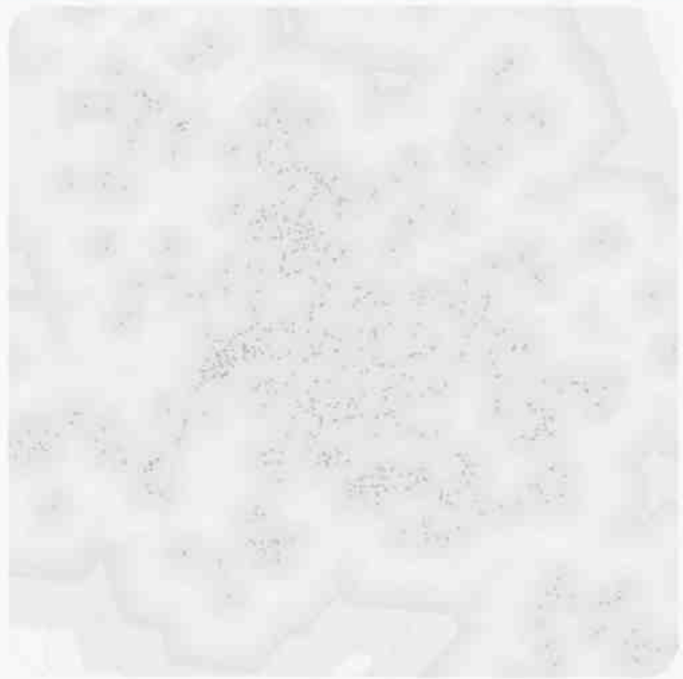
Prague EV charging stations ML model



https://kraina-ai.github.io/srai/latest/examples/use_cases/simple_machine_learning_with_overture_maps_data/

Prague EV charging stations ML model

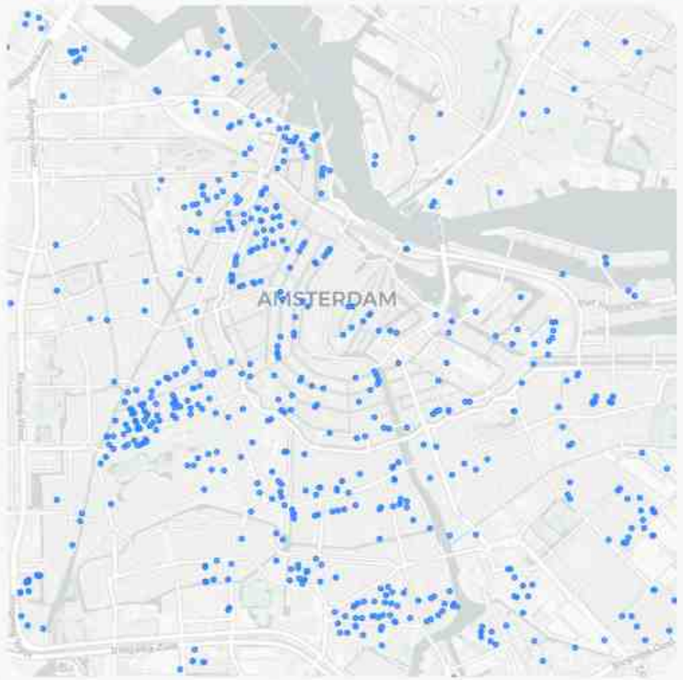




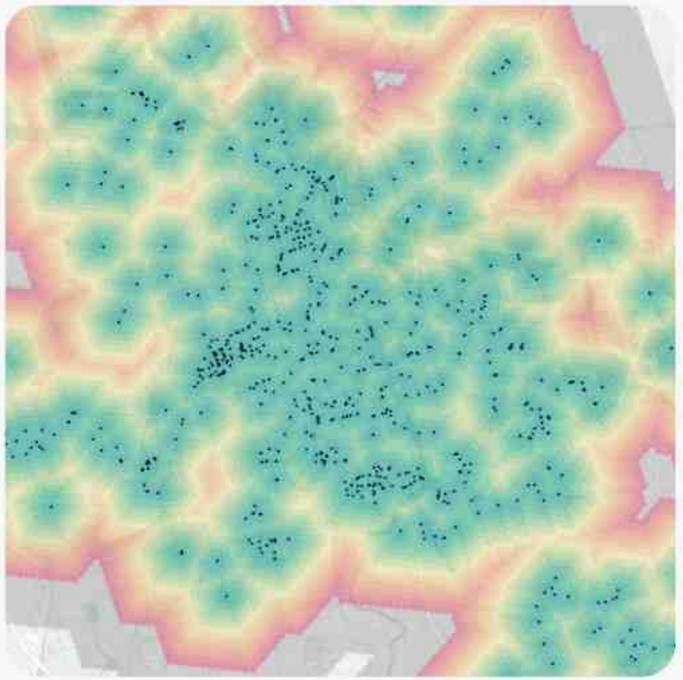
Download data for Amsterdam

https://kraina-ai.github.io/srai/latest/examples/use_cases/simple_machine_learning_with_overture_maps_data/

Prague EV charging stations ML model



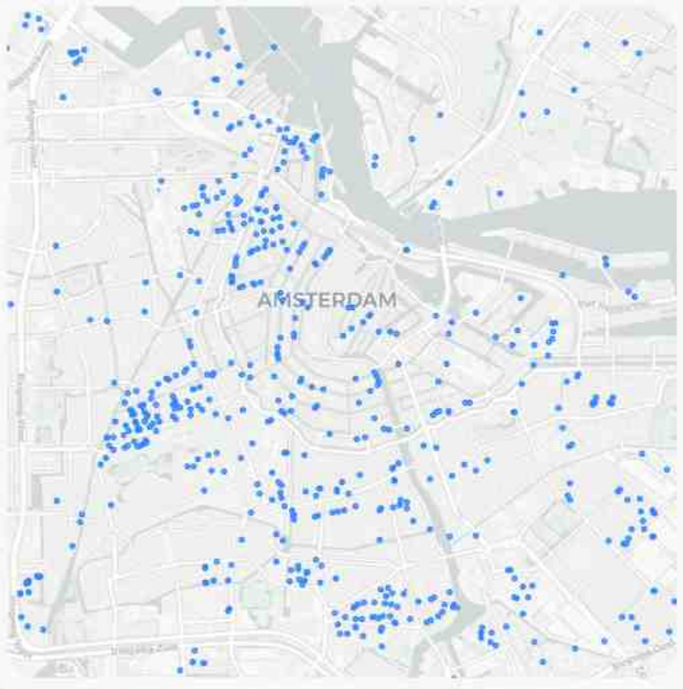
Download data for Amsterdam



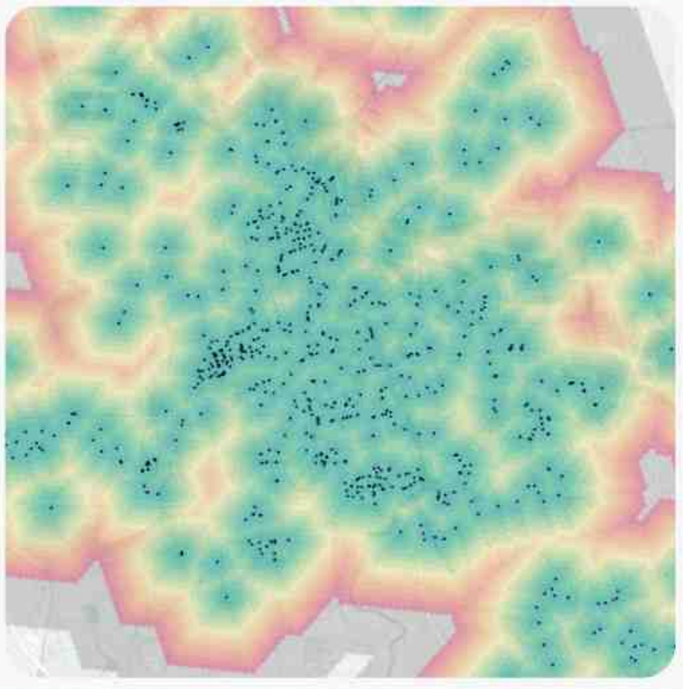
Assign distance from station outwards

https://kraina-ai.github.io/srai/latest/examples/use_cases/simple_machine_learning_with_overture_maps_data/

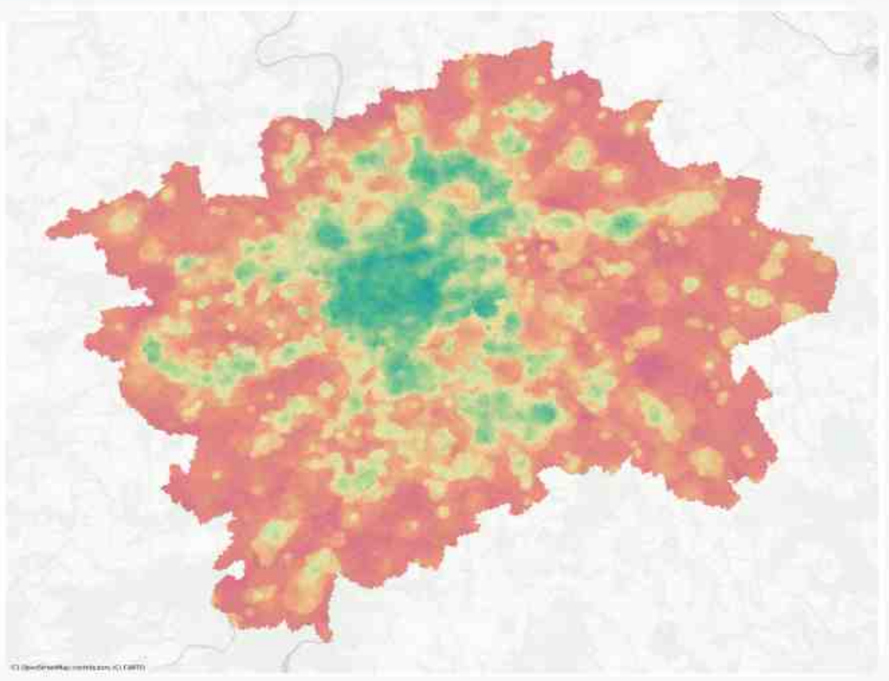
Prague EV charging stations ML model



Download data for Amsterdam



Assign distance from station outwards



Train the model with spatial features and apply to Prague

https://kraina-ai.github.io/srai/latest/examples/use_cases/simple_machine_learning_with_overture_maps_data/

Thanks to

DuckDB Spatial extension author:

Max Gabrielsson

PyArrow, GeoArrow & GeoPandas maintainer:

Joris Van den Bossche

GeoArrow maintainers:

Dewey Dunnington, Kyle Barron

Fruitful conversations about **GeoParquet** and **Overture**:

Chris Holmes, Jacob Wasserman

Thank you for the attention! Questions?

```
pip install quackosm[cli] overturemaestro[cli]
```

```
conda install conda-forge::quackosm
```

https://github.com/kraina-ai/quackosm/ 289★

https://github.com/kraina-ai/overturemaestro/ 28★

kamilraczycki.com
/raczyckikamil
/raczeq

Slides available on
EuroPython pretalx