Overture Data Processing

Uses DuckDB through Python to query the Overture Maps Data for the United Kingdom. Data processed into both `gpkg` and `geoparquet` formats.

`src/queries.py` queries the Overture AWS hosted files using a DuckDB, `src/clean.py` then processes the `.gpkg` to normalise the `json` columns and add/remove lists to save as `.parquet` and `.gpkg` formats.

Different geographic extents may be specified to retrieve data for different regions.

Reproduce results

It is highly recommended to use a virtual environment to reproduce these results locally (e.g. using `python -m venv` or Anaconda, which is likely the easiest approach on a Windows system).

1. Unzip the main file `overture-uk.zip` and enter the directory `cd overture-uk`. (ANON: Following publication these scripts will be hosted on a Public GitHub repository)
2. Ensure the local python version is `>=3.11,<=3.12` (I recommend using `pyenv` or `conda`).
3. `pip install -r requirements.txt` or equivalent; e.g. This projects uses `pdm` so `pdm install` will work and instead uses the `pyproject.toml` file to identify dependencies.

WARNING: Typical `conda` installation may fail due to the way `pdm` generated the `requirements.txt` file. It is recommended to only use `conda` to manage the python version if preferred, but installation of dependencies should be attempted though `pip` (which is included in a new `conda` environment by default).

3. Run `git init` to generate a `.git` directory, and `dvc init` to generate a `.dvc` directory, which allow `dvc` commands to function correctly. (ANON: These directories were removed for anonymity).

NOTE: *Please ensure that there are no residual files within the `raw/` directory (if it exists) before running analysis; files ending in `*.gpkg.tmp_rtree_uk_places.db` will not be overwritten and will cause errors. This will only be the case if previous runs have been attempted and cancelled before completing.*

Replicate UK results

First note that full replication of this dataset requires the following files that cannot be redistributed. They must be named as shown and in the correct directory (`$HOME/data`):

- `"~/data/OA_2021_BGC.gpkg": Output Areas for 2021 (BGC)`

  https://geoportal.statistics.gov.uk/maps/6beafcfd9b9c4c9993a06b6b199d7e6d

- `"~/data/OA_lookup-2021.csv": Output Areas lookup for 2021`

  https://geoportal.statistics.gov.uk/maps/4d6cf4e41ec845f6bdf5056499c37578_0

- `"~/data/SG_DataZoneBdry_2011.zip": Scotland DataZones for 2011`

  https://spatialdata.gov.scot/geonetwork/srv/api/records/7d3e8709-98fa-4d71-867c-d5c8293823f2

- `"~/data/NI_DZ21.zip": Northern Ireland DataZones for 2021`

  https://www.nisra.gov.uk/support/geography/data-zones-census-2021

- `"~/data/LAD_BUC_2022.gpkg": Local Authority Districts for 2022`

  https://geoportal.statistics.gov.uk/maps/42af123c4663466496dafb4c8fcb0c82_0

NOTE: If any of the above links no longer work, please query the Open Geography Portal to find them.

Without these files, the processing will retrieve and clean the UK Overture data, but the final stage of post-processing to attach census information and remove non-UK points will **fail**. Additionally, the process of retrieving and cleaning all UK POIs takes a very long time. If you are interested only in reproducing the download and cleaning stage for another area *please see the next section.*

To replicate our UK results, first set up the project as instructed above, then:

- Run `dvc repro pipelines/uk_full/dvc.yaml`

NOTE: *Downloading the Overture data will take a long time, and it will appear like nothing is happening. To verify that the data is still being downloaded you can look inside the* `data/raw` *directory; the output file size will be increasing.*

Reproduce results for other bounding boxes

Contained in `pipelines/custom/` is a `params.yaml` that may be used to specify the bounding box for another location. This file contains demos for either Nepal, or Seattle.

To retrieve the Overture data for these bounding boxes, first initialise the project as above, then:

1. Edit `pipelines/custom/params.yaml`: Adjust the `filename` and `bounds` as necessary. Smaller areas will be much faster.

Inside `pipelines/custom/params.yaml`:

```
# filename: nepal_places
# bounds:
#     minx: 80.0601
#     maxx: 88.2040
#     miny: 26.3475
#     maxy: 30.4470

filename: seattle_places
bounds:
  minx: -122.4447744
  maxx: -122.2477071
  miny: 47.5621587
  maxy: 47.7120663
```

2. Run: `dvc repro pipelines/custom/dvc.yaml`

NOTE: *Downloading the Overture data will take a long time, and it will appear like nothing is happening. To verify that the data is still being downloaded you can look inside the* ***data/raw*** *directory; the output file size will be increasing.*

Common Issues

Leftover rtree files

When running any pipeline using `dev repro`, the following error may occur.

```
> dvc repro pipelines/uk_full/dvc.yaml

Running stage 'pipelines/uk_full/dvc.yaml:query':
> python -m src.query --minx -9.0 --maxx 2.01 --miny 49.75 --maxy 61.01 --filename uk_places

Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "{..}/src/query.py", line 54, in <module>
    duckdb.query(query)
duckdb.duckdb.IOException: IO Error: GDAL Error (1): sqlite3_exec(PRAGMA journal_mode = OFF;
PRAGMA synchronous = OFF;
CREATE VIRTUAL TABLE my_rtree USING rtree(id, minx, maxx, miny, maxy)) failed: table my_rtre
ERROR: failed to reproduce 'pipelines/uk_full/dvc.yaml:query': failed to run: python -m src.
```

This error means that a residual file is left over from a previous run. To solve this error, please remove the file `uk_places.gpkg.tmp_rtree_uk_places.db` (or equivalent) from `./data/raw/`.

Bug fixes

1. There was a bug with the previous version where the following code in `clean.py` was failing:

```
lambda x: x[0] if not isinstance(x, float) else {}
```

Due to a difference in how `NoneType` is handled, we have replaced each occurrence with:

```
lambda x: x[0] if x isinstance(x, Iterable) else {}
```

2. Query leaves residual files, these are now removed when a query completes, in `query.py`:

```
rtree_file = Path(f"data/raw/{filename}.gpkg.tmp_rtree_{filename}.db")
if rtree_file.exists():
    rtree_file.unlink()
```

3. DuckDB does not include `httpfs` or `spatial` by default, these have been added to the `sql` query:

```
INSTALL httpfs;
INSTALL spatial;
```