Reproducibility review of: Analysis of cycling network evolution in OpenStreetMap through a data quality prism

Alexander Kmoch

2023-06-07

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://osf.io/9kp7u/. To cite the report use

> Kmoch, Alexander (2023, March). Reproducibility review of: Analysis of cycling network evolution in OpenStreetMap through a data quality prism. https://doi.org/10.17605/osf.io/9kp7u

Reviewed paper

Bres, R., Peralta, V., Le-Guilcher, A., Devogele, T., Olteanu Raimond, A.-M., and de Runz, C.: Analysis of cycling network evolution in OpenStreetMap through a data quality prism, AGILE GIScience Ser., 4, 3, https://doi.org/10.5194/agile-giss-4-3-2023, 2023

Summary

The authors execute their study fully within a framework of open source geospatial software. The study is reproducible and the authors provide provide 2 scripts that query and subsequently visualise aspects of the data. Furthermore, they provide the scripts and a How-to / Readme file that made it easy to reproduce. The repository is located at https://github.com/raphael-bres/AGILE2023-reproduction

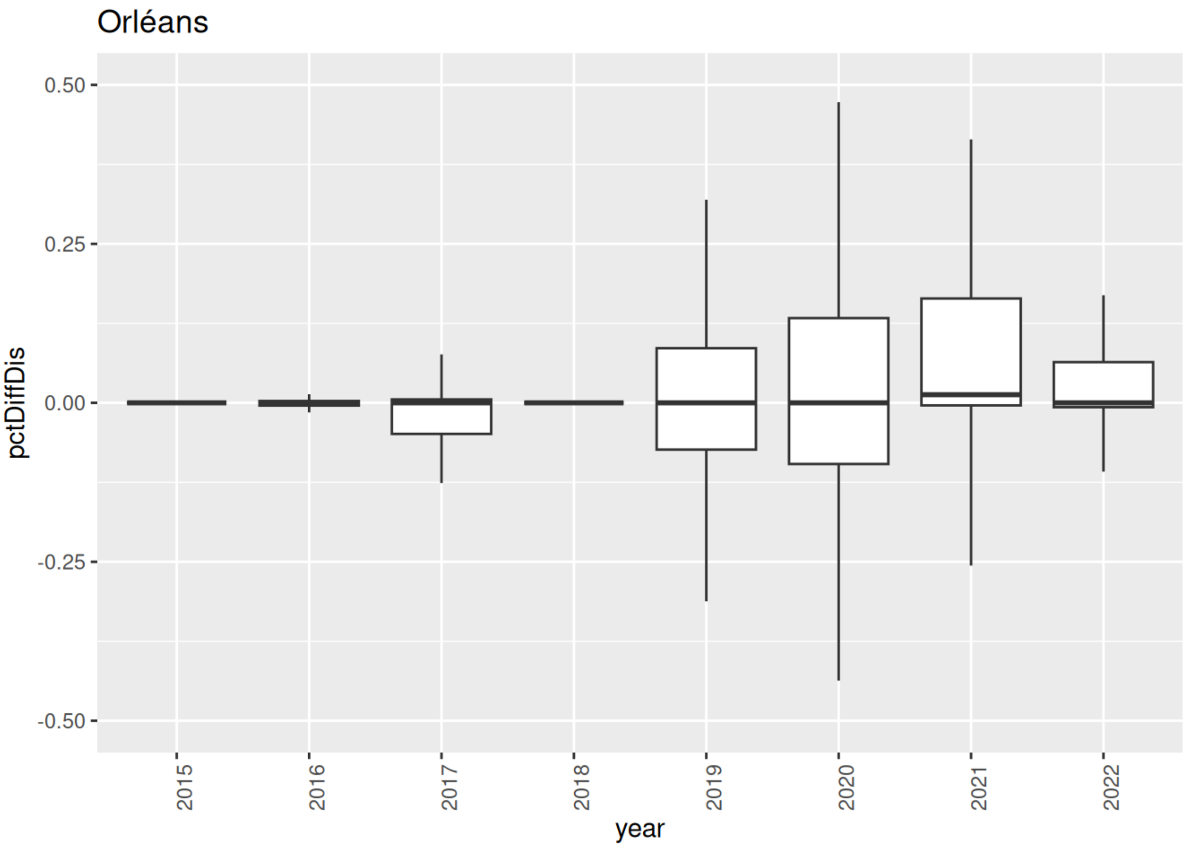
Reproduction was *successful*.

The main challenges in reproduction were related to acquiring the correct data and the handling of the OSRM software. This was overcome jointly through authors communication and fortunate experience with the Docker ecosystem of the reviewer.

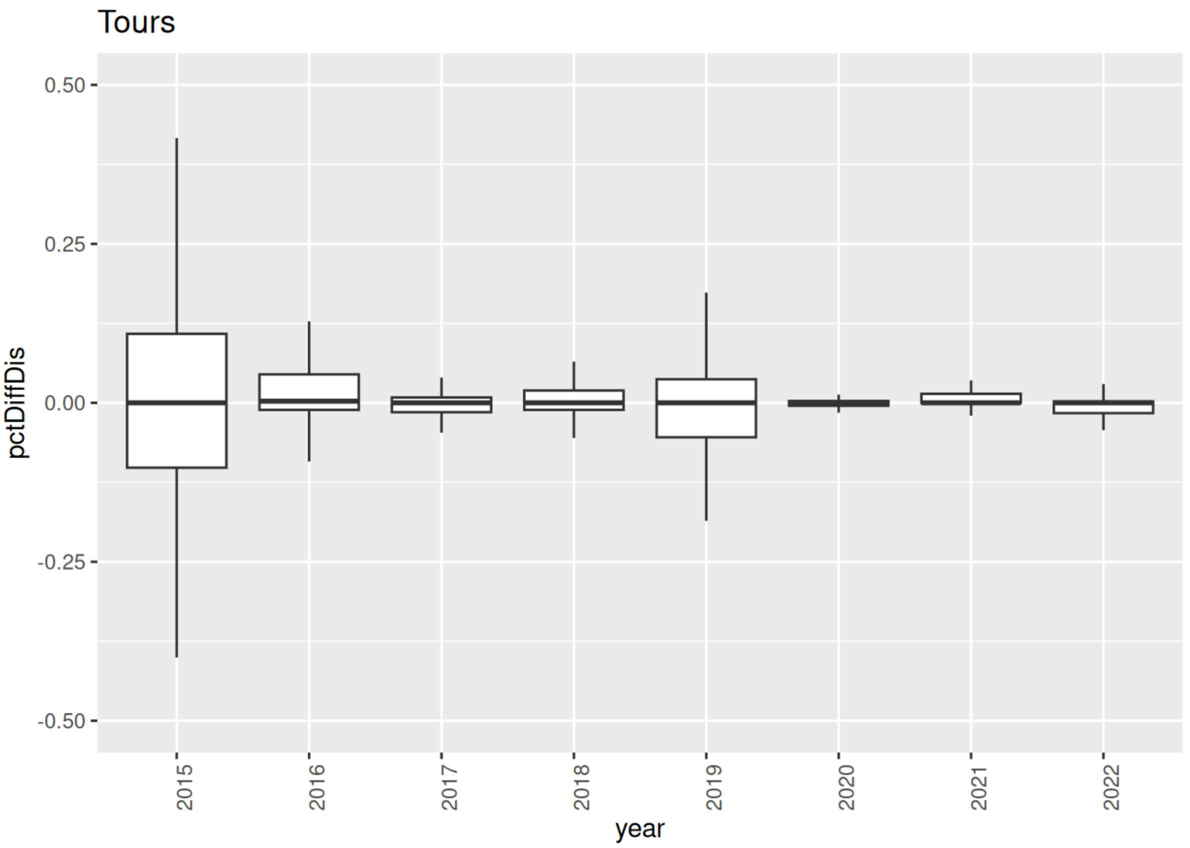
The data analysis and visualisation was straightforward and the results were consistent with the original paper.

- alldataZones.geojson
- dataZones.csv

Figures 4a and 4b are reproduced below:


Orléans
0.50
0.25
0.00
-0.25
-0.50
pctDiffDis
2015
2016
2017
2018
2019
2020
2021
2022
year

[ ]


Tours
0.50
0.25
0.00
-0.25
-0.50
pctDiffDis
2015
2016
2017
2018
2019
2020
2021
2022
year

[ ]

Reproducibility reviewer notes

Data acquisition:

The authors used Open Street Map data January snapshots from 2014 to 2022 for the region France, centre. After initial misunderstnadings, as the authors used the OSM internal download, we agreed that the data was downloaded from https://download.geofabrik.de/europe/france/centre.html, files centre-140101.osm.pbf to centre-220101.osm.pbf

The authors used `Open Source Routing Machine: The OpenStreetMap Data Routing Engine` (OSRM) as the routing engine. The description of the whole installation and setup being based on a system with Ubuntu 20 was maybe a bit narrow, especially as they actually used the Docker container procedure instead of a system installation.

The authors provided a link to the OSRM website, but the installation instructions were not considering the latest change in license terms of the Docker registry, and OSRM had switched their container registry. The reviewer used the OSRM backend Docker container from `ghcr.io/project-osrm/osrm-backend:v5.27.1`

```
docker pull ghcr.io/project-osrm/osrm-backend:v5.27.1
```

Initially 3 preparatory commands need to be run. I adapted the command based on the newer container naming, in the folder with the downloaded osm…pbf data files:

```
for year in $(ls -1 *0101.osm.pbf); do
  echo $year;
  fibase=$(basename $year .osm.pbf);
  echo $fibase;
  docker run -t -v "${PWD}:/data" ghcr.io/project-osrm/osrm-backend:v5.27.1 \
    osrm-extract -p /opt/bicycle.lua /data/$fibase.osm.pbf;
  docker run -t -v "${PWD}:/data" ghcr.io/project-osrm/osrm-backend:v5.27.1 \
    osrm-partition /data/$fibase.osrm;
  docker run -t -v "${PWD}:/data" ghcr.io/project-osrm/osrm-backend:v5.27.1 \
    osrm-customize /data/$fibase.osrm;
done
```

Finally, the routing engine needed to be running and accessible for all years. I used the following command to start the container:

```
docker run -t -i -p 5014:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-140101.osrm
docker run -t -i -p 5015:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-150101.osrm
docker run -t -i -p 5016:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-160101.osrm
docker run -t -i -p 5017:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-170101.osrm
docker run -t -i -p 5018:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-180101.osrm
docker run -t -i -p 5019:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-190101.osrm
docker run -t -i -p 5020:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-200101.osrm
docker run -t -i -p 5021:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-210101.osrm
docker run -t -i -p 5022:5000 -v ./repro_data:/data ghcr.io/project-osrm/osrm-backend \
  osrm-routed --algorithm mld /data/centre-220101.osrm
```

Running the Python script was straightforward.

Then I had to run the R script, but I got an error:

```
Error in is.data.frame(x) : object 'd20dist' not found

d20dist
```

In line 190 rowMeans(d20dist) likely meant to refer to d20Dist. I fixed the typo and the updated script is attached in the materials folder of this report.

The R markdown generated the figures from the data that was processed from the Python script.