Reproducibility Review of: Beyond Walking and Biking: Expanding the 15-minute City Area through Public Transport

Item	Value
Title	Beyond Walking and Biking: Expanding the 15-minute City Area through Public Transport
Authors	Manuel Canestrini, Ioannis Giannopoulos]
Ref. paper	AGILE GIScience Ser., 6, 2, https://doi.org/10.5194/agile-giss-6-2-2025
Codechecker(s)	Carlos Granell, 0000-0003-1004-9695
Date of Check	2025-04-03
Summary	Full reproduction
Repository	https://doi.org/10.48436/0cffc-hax39
Ref. certificate	https://doi.org/10.17605/osf.io/st98a

Table 1: Reproduction metadata

Output	Comment
~/results/analysis/average_*.csv (3 files)	Table 1
~/results/analysis/median_*.csv (3 files)	Table 2
~/results/analysis/15min*.csv (4 files)	Table 3
~/results/plots/heatmap_walk.png	Figure 1
~/results/plots/heatmap_walk_PT.png	Figure 2
~/results/plots/heatmap_differences.png	Figure 3
~/results/plots/lineplot_walk.png	Figure 4
~/results/plots/lineplot_walk_PT.png	Figure 5

Table 2: Summary of output files generated

Summary

The authors included a DASA section where they briefly but accurately describe key details of the datasets used, software version and scripts generated, and how to access these resources. While the link to the code and data repository was anonymized, the author provided the DOI to the data/code repository for reproduction which will be included in the final version of the paper: https://doi.org/10.48436/0cffc-hax39.

Tables 1-3 and Figures 1-5 were eligible for reproduction. All of them were successfully reproduced. In general, **the paper is fully reproducible** because I was able to verify the claims and results presented in the article by re-executing the scripts provided with the accompanying datasets.

The full citation of the paper is:

Manuela Canestrini and Ioannis Giannopoulos. Beyond Walking and Biking: Expanding the 15-Minute City Area through Public Transport. *AGILE GIScience Ser.*, 6, 2, https://doi.org/10.5194/agile-giss-6-2-2025, 2025

Reproducibility reviewer notes

The data/code repository is hosted by the TU Wien. Access to the https://doi.org/10.48436/0cffc-hax39 shows basic metadata of the repo, including authors, abstract, description of contained components (code, data, results), selected licenses, and link to a downloadable zip file. Permissive licenses: “All data files are licensed under CC BY 4.0, all software files are licensed under MIT License”.

Once downloaded and unzipped, the folder contains a README.txt file where the authors provide clear and detailed instructions for reproducing the article's results. No further communication with the authors was necessary to fully reproduce the article's results, indicating that the repository included all the necessary resources for reproduction and the accuracy of the instructions provided.

Two small caveats that I describe in detail below:

- Versions of some python package used were old compared with the current versions at the time of submission. This had some implications in the reproduction process.
- The data/code repository was well organized. Execution instruction includes how to create a computational environment with Conda. Whenever possible, it's advisable to provide additional mechanisms for reproduction by including descriptive files such as Dockerfile and compose.yaml to detail the computational environment.

Installation prerequisites and computational environment

My local machine runs a Windows 11 Pro, 16 GB, and has installed Ubuntu 20.04.6 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64). I used WSL environment for reproduction.

The downloaded project folder contains:

- **code** folder, which includes python script files for analysis
- **data** folder, which includes required datasets in csv format
- **results** folder, which includes generated outputs (data, figures,) by authors and used in the paper
- **README.txt** with detailed instruction for reproduction
- **requeriments.txt** with pinned versions of required python packages

In general, I strictly followed the instructions in the README.txt to exec the /code/scripts in the specified order to recreate Table and Figures. I, however, created a couple of files in the root folder to recreate the computational environment using Docker Compose.

Dockerfile

```
# Use an official Python image

# https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html#jupyter-minimal-notebook

FROM quay.io/jupyter/minimal-notebook:python-3.11

# Install system dependencies

ARG NB_USER

ARG NB_UID

ENV USER=${NB_USER}

ENV NB_UID=${NB_UID}

ENV HOME=/home/${NB_USER}

USER root

COPY . ${HOME}

RUN chown -R ${NB_UID} ${HOME}

RUN apt-get update && apt-get install -y build-essential pkg-config


# Copy requirements and install Python dependencies

RUN pip install --upgrade pip

RUN pip install --no-cache-dir -r requirements.txt


RUN rmdir work
```

```
USER ${NB_USER}
```

Compose.yml

```
services:
 jupyter:
  build:
   context: .
   dockerfile: Dockerfile
  environment:
   JUPYTER_ALLOW_INSECURE_WRITES: true
  volumes:
   - ./:/home/jovyan
  ports:
   - 8888:8888
  command: "start-notebook.py --NotebookApp.token=''"
```

I opened a terminal window to start the container in the root folder

```
$ docker compose up -d –build
```

I opened a browser tab with the URL localhost:8080/lab to launch a web-based JupyterLab session (https://jupyter.org/). For reproduction, I only needed to run a series of Python scripts in the right order, so I simply opened a terminal session within JupyterLab.

As explained in the README.txt, scripts located in the ~/**code** folder had to be executed in a particular order.

- **Step A: a_extract_pois.py** (1x)
- **Step B: b_accessibility_mp.py** (1x) --> only showcasing the accessibility computation, actual results precomputed and provided in folder **~/results/accessibility**
- **Step C: c_analysis.py** (2x - different user input)
- **Step D: d_plots.py** (1x)

I skipped step B above because it has no influence on the reproduction of the figures and tables in the paper (which otherwise would so for replication purposes). Data files located in the **~/data/accessibility** folder are the correct input datasets to run subsequent Steps C and D.

Step A was related to data preparation. It generates 9 data files (CSV), one per service/POI category, in the folder **~/results/pois/**.

Step C generates a series of data files in CSV format which serve to recreate Tables 1, 2, and 3 in Section 3 (Results) of the paper. Connections between generated data files and Tables are:

- Table 1 requires 3 files in **~/results/analysis/average_*.csv**
- Table 2 requires 3 files in **~/results/analysis/median_*.csv**
- Table 3 requires 4 files in **~/results/analysis/15min*.csv**

Step D generates 5 plots (PNG) which correspond to Figures 1 to 5 in Section 3 (Results) of the paper:

- Figure 1 corresponds to **~/results/plots/heatmap_walk.png**
- Figure 2 corresponds to **~/results/plots/heatmap_walk_PT.png**
- Figure 3 corresponds to **~/results/plots/heatmap_differences.png**
- Figure 4 corresponds to **~/results/plots/lineplot_walk.png**
- Figure 5 corresponds to **~/results/plots/lineplot_walk_PT.png**

Data preparation

No specific steps were required to collect additional data. All input datasets were located within the **~/data** folder, divided further into specific subfolders. Data files are ready to be used by the analysis scripts. For strict reproduction purposes, authors provided all the necessary datasets in good shape and format. Good Job!

As an additional note, and considering replicability, future researchers could use the same analytical methods (scripts, etc.) but apply them to a city other than Vienna to verify the internal validity of the study. This would require recreating the same number of datasets for another city. For doing so, data collection and preparation are crucial to recreating similar datasets for, say, Madrid. In this sense, the article is less precise in specifying the data collection and preparation steps, although some aspects are included in the DASA section (e.g., OSM snapshot date, etc.).

Step A was then aimed at preprocessing some input data files for subsequent analysis in Step C and D. Note that I created the folder ~/**repro** to store script outputs as text files.

```
jovyan@ee1cb4d0c34a:~$ mkdir repro
```

For Step A, I ran the script **~/code/a_extract_pois.py**.

```
jovyan@ee1cb4d0c34a:~$ python ~/code/a_extract_pois.py > ~/repro/a_output.txt

Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),

(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries) but was not found to be installed on your system. If this would cause problems for you, please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd

jovyan@ee1cb4d0c34a:~$ cat ~/repro/a_output.txt
```

```
healthcare

services

transport

outdoor

supplies

restaurant

culture

education

physical

done
```

Deprecation warning messages were related to using old versions of some python packages. These do not seem to be critical for reproduction.

Running the code

Steps C and D compute the results and then are connected to Section 3 (Results) on the paper.

Step C required executing script **~/code/c_analysis.py**. It needs to be executed twice, to consider or not the terrain slope. This is implemented with an input statement (use command line arguments instead?). Here, I changed the source code to capture the script output to a file by commenting out this input statement.

```
#terrain = input("Include slope-dependent walking and biking speed? Please enter either 'yes' or 'no': ")

terrain = 'yes'
```

```
jovyan@ee1cb4d0c34a:~$ python ~/code/c_analysis.py > ~/repro/c_terrain_yes_output.txt

Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),

(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries) but was not found to be installed on your system. If this would cause problems for you,

please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd

jovyan@ee1cb4d0c34a:~$ python ~/code/c_analysis.py > ~/repro/c_terrain_no_output.txt

 Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),

(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries) but was not found to be installed on your system. If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

With a little extra effort, the authors could have created a file called, say, Table1.csv, which would merge the required CSV data files. This way, the resulting data in Table1.csv would have a similar number of columns and rows as the corresponding table in the article.

Step D required executing script **~/code/d_plots.py**.

```
jovyan@ee1cb4d0c34a:~$ python ~/code/d_plots.py > ~/repro/d_output.txt

/home/jovyan/code/d_plots.py:3: DeprecationWarning:

Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),

(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries) but was not found to be installed on your system. If this would cause problems for you, please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd

Traceback (most recent call last):

  File "/home/jovyan/code/d_plots.py", line 134, in <module>

    borders = gpd.read_file(path_roi)

              ^^^^^^^^^^^^^^^^^^^^^^^^

  File "/opt/conda/lib/python3.11/site-packages/geopandas/io/file.py", line 297, in _read_file

    return _read_file_fiona(

           ^^^^^^^^^^^^^^^^^

  File "/opt/conda/lib/python3.11/site-packages/geopandas/io/file.py", line 315, in _read_file_fiona

    if _is_zip(str(path_or_bytes)):

       ^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/opt/conda/lib/python3.11/site-packages/geopandas/io/file.py", line 173, in _is_zip

    parsed = fiona.path.ParsedPath.from_uri(path)

             ^^^^^^^^^^

AttributeError: module 'fiona' has no attribute 'path'. Did you mean: 'Path'?
```

The script ran an exception because the “fiona” module installed was 1.10.1 and the script **~/code/d_plots.py** required version 1.9.6 (see https://stackoverflow.com/questions/78949093/how-to-resolve-attributeerror-module-fiona-has-no-attribute-path).

```
jovyan@ee1cb4d0c34a:~$ pip show fiona geopandas

Name: fiona

Version: 1.10.1

Summary: Fiona reads and writes spatial data files

---

Name: geopandas

Version: 0.14.0

Summary: Geographic pandas extensions
```

To fix this, I recommend that authors pin the "fiona" module to the correct version in the **requirements.txt** file. Otherwise, "geopandas" manages dependencies automatically and may install the wrong version of fiona, as in my computational environment. I reinstalled "fiona" to the 1.9.6 version using pip with the -I argument (stands for --ignore-installed which will ignore the installed packages, overwriting them). This resolved the dependency issue, and the script ran successfully.

Authors' response: Authors added the "fiona" version specification to the requirements.txt as suggested (fiona==1.9).

```
jovyan@ee1cb4d0c34a:~$ pip install -Iv --user fiona==1.9.6

jovyan@ee1cb4d0c34a:~$ pip show fiona geopandas

WARNING: Ignoring invalid distribution ~iona (/opt/conda/lib/python3.11/site-packages)

Name: fiona

Version: 1.9.6

Summary: Fiona reads and writes spatial data files

---

Name: geopandas

Version: 0.14.0

Summary: Geographic pandas extensions
```

Outputs and results

Fig 1 below shows the original Table 1 and Figs 2, 3 and 4 show the reproduced data files after executing Step C needed for the original Table 1. Table 2 and 3 in the paper are omitted here because the process is the same. Yet, I verify that the generated data files correspond to the associated columns in these two tables.

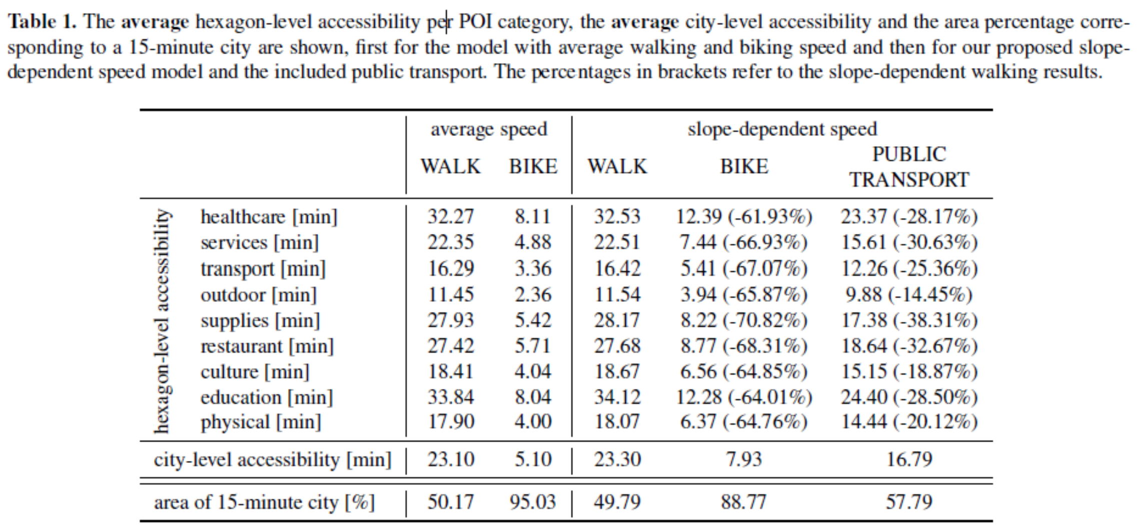
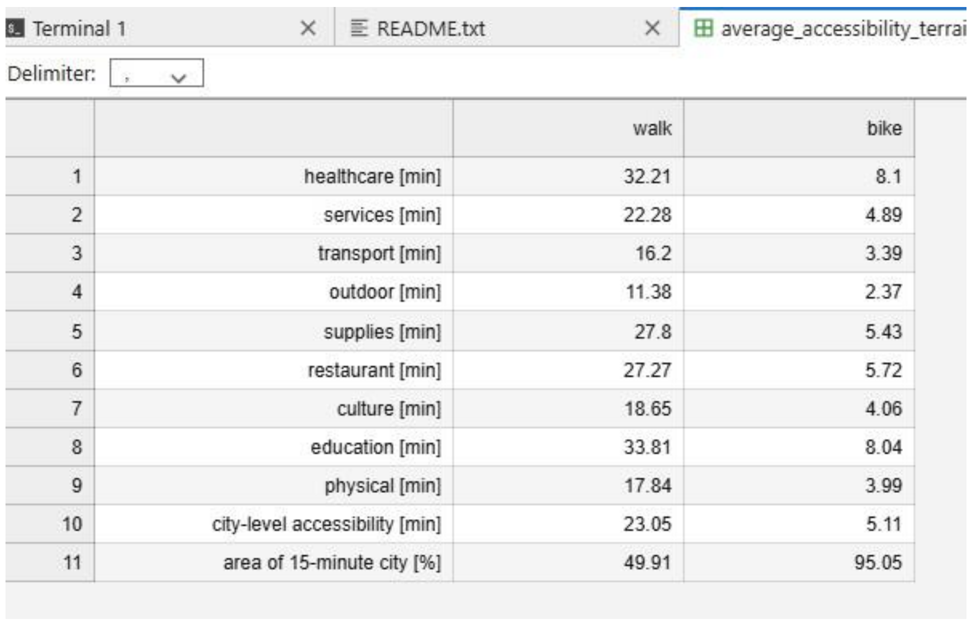


Table 1. The **average** hexagon-level accessibility per POI category, the **average** city-level accessibility and the area percentage corresponding to a 15-minute city are shown, first for the model with average walking and biking speed and then for our proposed slope-dependent speed model and the included public transport. The percentages in brackets refer to the slope-dependent walking results.

		average speed		slope-dependent speed		
		WALK	BIKE	WALK	BIKE	PUBLIC TRANSPORT
hexagon-level accessibility	healthcare [min]	32.27	8.11	32.53	12.39 (-61.93%)	23.37 (-28.17%)
	services [min]	22.35	4.88	22.51	7.44 (-66.93%)	15.61 (-30.63%)
	transport [min]	16.29	3.36	16.42	5.41 (-67.07%)	12.26 (-25.36%)
	outdoor [min]	11.45	2.36	11.54	3.94 (-65.87%)	9.88 (-14.45%)
	supplies [min]	27.93	5.42	28.17	8.22 (-70.82%)	17.38 (-38.31%)
	restaurant [min]	27.42	5.71	27.68	8.77 (-68.31%)	18.64 (-32.67%)
	culture [min]	18.41	4.04	18.67	6.56 (-64.85%)	15.15 (-18.87%)
	education [min]	33.84	8.04	34.12	12.28 (-64.01%)	24.40 (-28.50%)
	physical [min]	17.90	4.00	18.07	6.37 (-64.76%)	14.44 (-20.12%)
city-level accessibility [min]		23.10	5.10	23.30	7.93	16.79
area of 15-minute city [%]		50.17	95.03	49.79	88.77	57.79

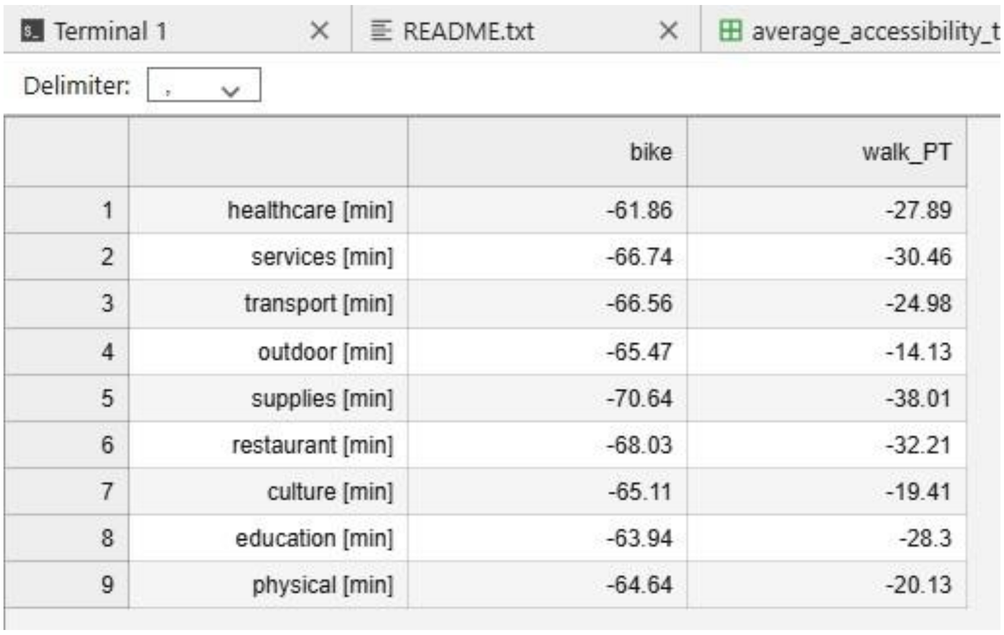
Fig 1. Original Table 1



Delimiter: ,

		walk	bike
1	healthcare [min]	32.21	8.1
2	services [min]	22.28	4.89
3	transport [min]	16.2	3.39
4	outdoor [min]	11.38	2.37
5	supplies [min]	27.8	5.43
6	restaurant [min]	27.27	5.72
7	culture [min]	18.65	4.06
8	education [min]	33.81	8.04
9	physical [min]	17.84	3.99
10	city-level accessibility [min]	23.05	5.11
11	area of 15-minute city [%]	49.91	95.05

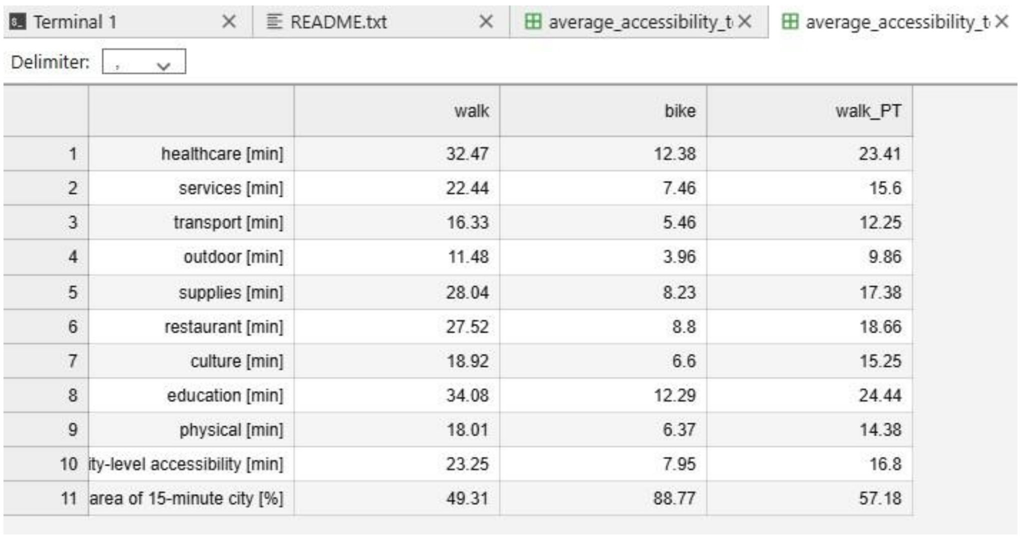
Fig 2. Generated **~/results/analysis/average_accessibility_terrain_no.csv** after executing script **~/code/c_analysis.py**.


Terminal 1 | README.txt | average_accessibility_t

Delimiter: ,

		bike	walk_PT
1	healthcare [min]	-61.86	-27.89
2	services [min]	-66.74	-30.46
3	transport [min]	-66.56	-24.98
4	outdoor [min]	-65.47	-14.13
5	supplies [min]	-70.64	-38.01
6	restaurant [min]	-68.03	-32.21
7	culture [min]	-65.11	-19.41
8	education [min]	-63.94	-28.3
9	physical [min]	-64.64	-20.13

Fig 3. Generated **~/results/analysis/average_accessibility_terrain_yes.csv** after executing script **~/code/c_analysis.py**.


Terminal 1 | README.txt | average_accessibility_t | average_accessibility_t

Delimiter: ,

		walk	bike	walk_PT
1	healthcare [min]	32.47	12.38	23.41
2	services [min]	22.44	7.46	15.6
3	transport [min]	16.33	5.46	12.25
4	outdoor [min]	11.48	3.96	9.86
5	supplies [min]	28.04	8.23	17.38
6	restaurant [min]	27.52	8.8	18.66
7	culture [min]	18.92	6.6	15.25
8	education [min]	34.08	12.29	24.44
9	physical [min]	18.01	6.37	14.38
10	ity-level accessibility [min]	23.25	7.95	16.8
11	area of 15-minute city [%]	49.31	88.77	57.18

Fig 4. Generated **~/results/analysis/average_decrease_terrain_yes.csv** after executing script **~/code/c_analysis.py**.

In general, the values of the three data files match the columns of Table 1 in the paper (Fig 1). Yet, there are small discrepancies between values, especially if we look at the decimal digits. This problem applies to Tables 1, 2 and 3 in the paper. For instance, compare these values:

- In Fig 1, value of row "healthcare [min]" in column "Average speed – Walk" is **32.27**
- In Fig 2, value of row "healthcare [min]" in column "walk" is **32.21**

If the input data located in the **~/data/accessibility** folder is the same for the original paper and for my reproduction, my suggestion to explain these small differences may be the inherent floating-point representation error in python and, in general, in any programming language (see https://dev.to/somacdivad/the-right-way-to-compare-floats-in-python-2fml). Another explanation might be type coercion (pandas.to_numeric)?

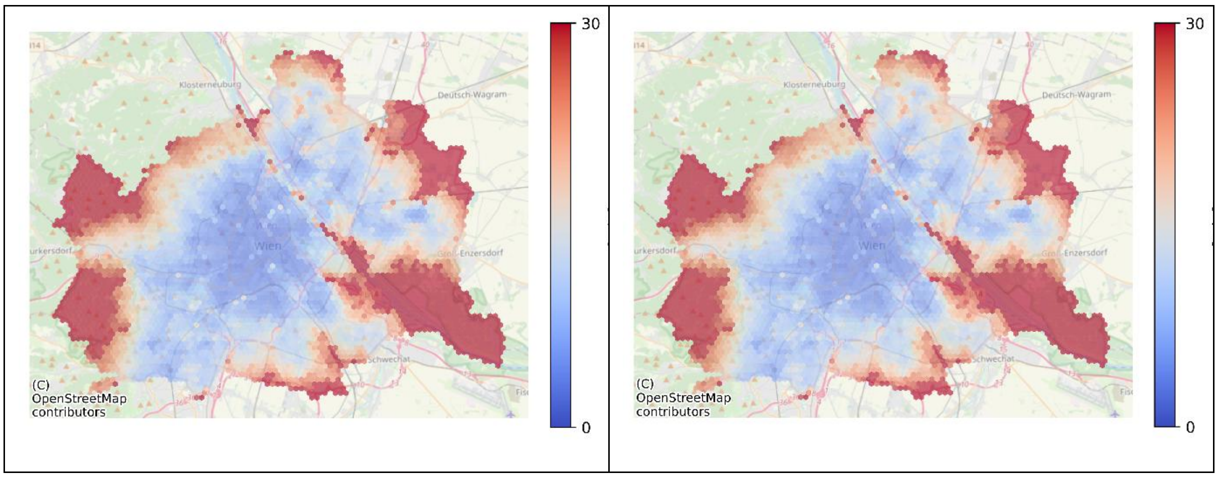
Authors' response: The discrepancies in Tables 1-3 are the result of using originally two different sources for the city boundary (GADM data from https://gadm.org/download_country.html and "Bezirksgrenzen Wien" from the Open Data Platform of Austria https://www.data.gv.at). When authors were preparing the data/code repository for the reproducibility review, they realized that inconsistency and changed data/code repo to use only the shapefile from the Open Data Platform. The input data located in the **~/data/accessibility** folder currently is the correct one, but it was different for the originally submitted paper. The values in Tables 1-3 have been updated accordingly in the camera-ready version of the paper.

To briefly explain why the authors think these discrepancies occurred: The two boundary shapes seem to result in slight differences in the placement of the hexagons. These different placements affect the locations of the hexagon centroids, which in turn might lead to different starting nodes being picked in the graph. As a result, the authors suspect that the values of the accessibility analysis have these slight variations.

If the above explanation is correct (floating-point error representation), I would suggest to authors update Tables 1, 2 and 3 in the paper to use only integer numbers or numbers with only one decimal digital. I am not sure if reporting 34.47 min or 34.4 min or 34 min is equivalent in this study. My guess is that 34 min is fine.

The same rationale may be applied to the percentages in Tables 1, 2 and 3 in the paper; I would remove decimal digits to report, for example, 61% instead of 61.93%.

Figs 5 to 9 show the original vs reproduced plots after executing Step D. Good job!



Original plot (Fig 1 in the paper)	Reproduced plot

Fig 5. Generated **~/results/plots/heatmap_walk.png** after executing script **~/code/d_plots.py**.

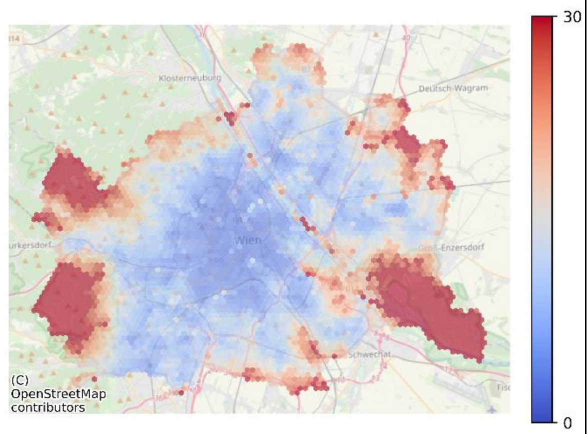
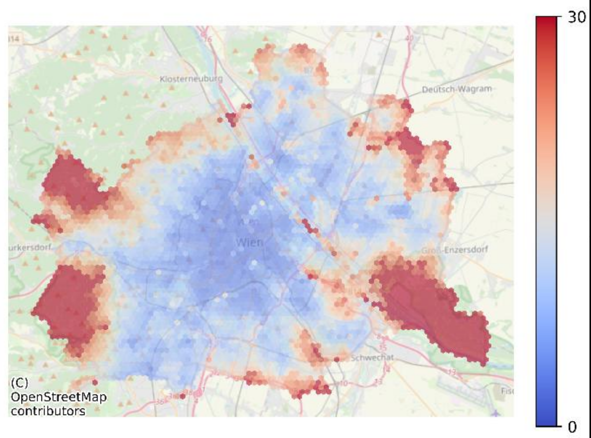
 Klosterneuburg Deutsch-Wagram Wien urkersdorf Groß-Enzersdorf Schwechat (C) OpenStreetMap contributors 30 0	 Klosterneuburg Deutsch-Wagram Wien urkersdorf Groß-Enzersdorf Schwechat (C) OpenStreetMap contributors 30 0
Original plot (Fig 2 in the paper)	Reproduced plot

Fig 6. Generated **~/results/plots/heatmap_walk_PT.png** after executing script **~/code/d_plots.py**.

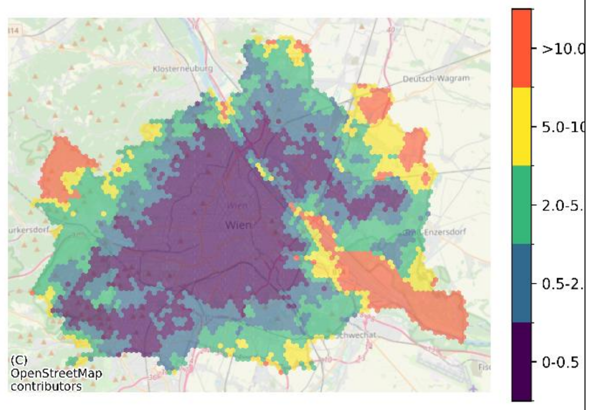
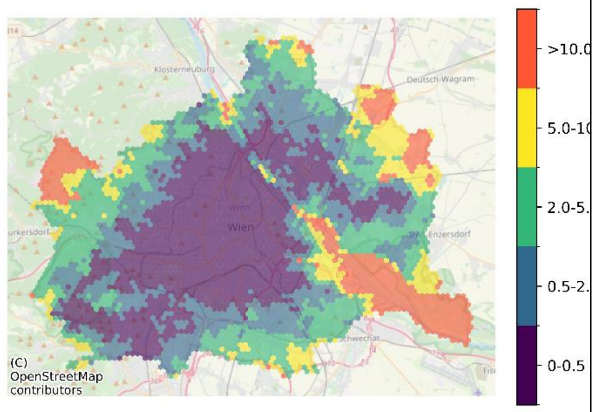
 Klosterneuburg Deutsch-Wagram Wien urkersdorf Enzersdorf Schwechat (C) OpenStreetMap contributors >10.0 5.0-10 2.0-5. 0.5-2. 0-0.5	 Klosterneuburg Deutsch-Wagram Wien urkersdorf Enzersdorf Schwechat (C) OpenStreetMap contributors >10.0 5.0-10 2.0-5. 0.5-2. 0-0.5
Original plot (Fig 3 in the paper)	Reproduced plot

Fig 7. Generated **~/results/plots/heatmap_differences.png** after executing script **~/code/d_plots.py**.

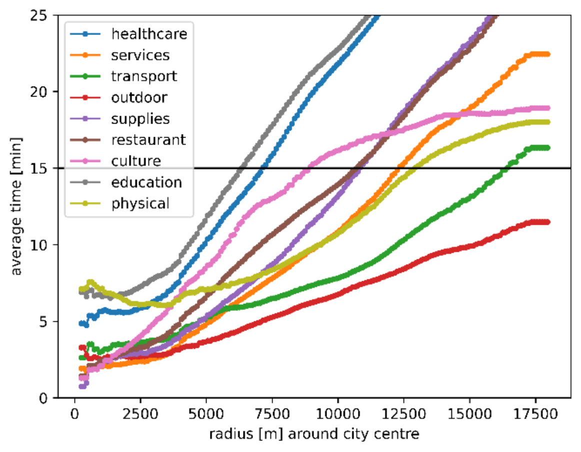
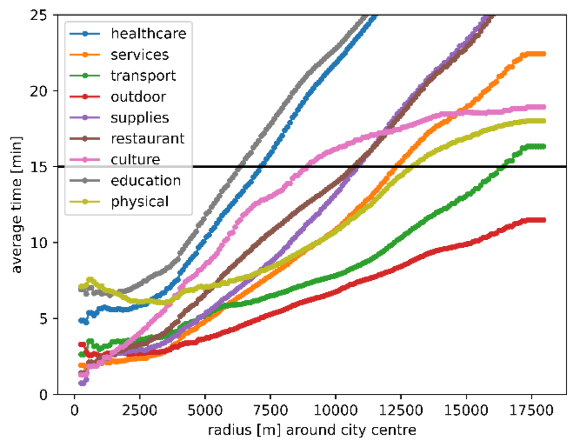
	
Original plot (Fig 4 in the paper)	Reproduced plot

Fig 8. Generated **~/results/plots/lineplot_walk.png** after executing script **~/code/d_plots.py**.

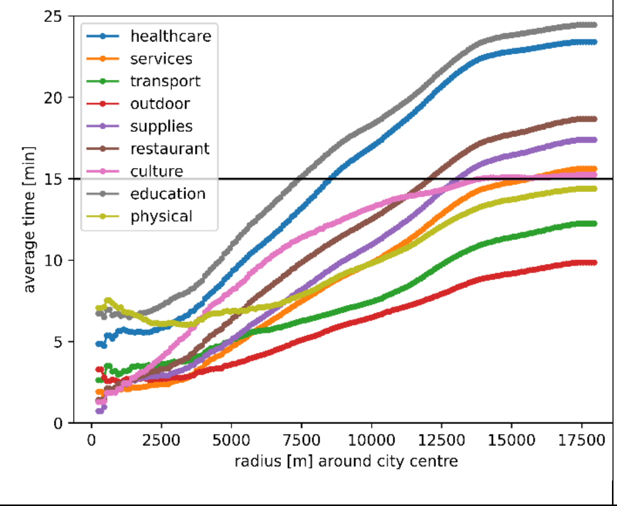
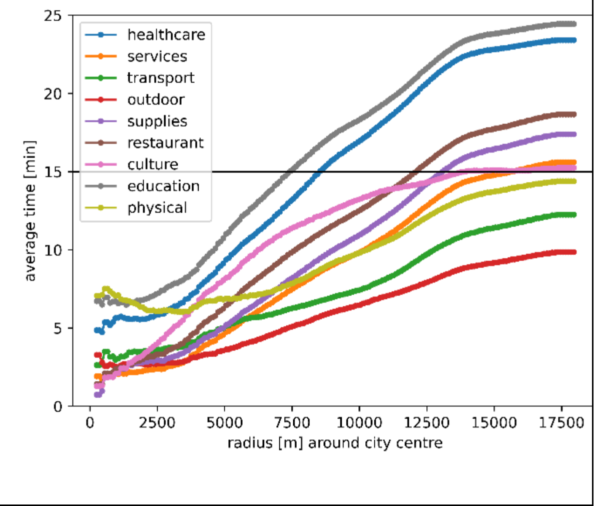
	
Original plot (Fig 5 in the paper)	Reproduced plot

Fig 9. Generated **~/results/plots/lineplot_walk_PT.png** after executing script **~/code/d_plots.py**.

Recommendations

- The authors archived data/code in a permanent repository (DOI). For completeness, it is good practice to add such a resource as regular reference to the published paper. Therefore, please properly cite your data/code repository and not just link to it from the DASA section.
- Good job on providing intermediate/input data sets to reproduce the study without downloading and preparing data from original sources.
- The READme.txt file provides detailed instructions on how to run the script series. It would be helpful to indicate next to each generated figure the number used in the manuscript (e.g., Fig 1, Fig 2, etc.).

Citing this document

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://doi.org/10.17605/osf.io/st98a

To cite the report use

Granell, C. (2025, April 4). Reproducibility review of: Beyond Walking and Biking: Expanding the 15-minute City Area through Public Transport. https://doi.org/10.17605/osf.io/st98a