Reproducibility review of: Accessibility for pedestrians under heat stress - the example of Heidelberg, Germany

Lorena Abad

2025-03-20

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 OSF LINK HERE. To cite the report use

FULL REPORT CITATION HERE

Reviewed paper

ADD FULL CITATION

Summary

The authors included a DASA section on the paper with an anonymous GitHub link to the repository with the code for the reproduction. After initial contact with the authors, the unanonymized link to the GitHub repository was obtained and forked into the reproducible AGILE organization.

The repository prepared by the authors has clear instructions on reproducibility for the Jupyter notebooks and points to the included QGIS project to recreate figures 2-4. Figure 5 is the result of one of the notebooks (`lcz_analysis.pynb`). A table of the reproduction of each Jupyter notebook with notes and comments can be found on the next section. One cell was added at the end of each notebook to document the session info of the reproduction environment.

Initially, the code limited the AOI to a subset for testing purposes. This was not initially clear from the documentation, but contacting the authors clarified the discrepancies. For this report, the code was successfully run for the whole AOI. Figures 2 to 5 could also be successfully reproduced besides manual edits in Figure 5.

See next section for technical details and recommendations.

Reproducibility reviewer notes

Re-running scripts

Notebook	Comment
Get_POIs.ipynb	Easy reproduction and resulted in no errors or issues. Note on the last output, there are 4 more amenities than the original run, 1 more supermarket and 3 more doctors
Isocalors_POI.ipynb	Updated backslashes to forward slashes, changed `append` to `_append` due to differences in geopandas versions, pyogrio save to JSON also needed to specify `driver = "GeoJSON"`, generated files saved to data directory.
lcz_analysis.ipynb	Updated backslashes to forward slashes, changed `append` to `_append` due to differences in geopandas versions, pyogrio save to JSON also needed to specify `driver = "GeoJSON"`, generated files saved to data directory. Figure 5 generated.

Generated files and figures

Figures 2 to 5 were successfully generated with the provided code and data in the GitHub repository.

- **Figure 3 and 4:** Files `vul_pop_grid_hd_fct5_noon.tif` and `vul_pop_grid_hd_fct3_noon.tif` are not generated by code but manually uploaded to GitHub.
- From communication with the authors:

  The layers `vul_pop_grid_hd_fct5_noon.tif` and `vul_pop_grid_hd_fct3_noon.tif` were created using QGIS. For this, the vulnerable population was extracted from the population raster file `zensus_2022_hd_LAEA.tif` and masked using `vul_area_transport_all.json` via the "Clip Raster by Mask Layer" function in QGIS. Specifically, the corresponding polygons were selected: - For `vul_pop_grid_hd_fct5_noon.tif`: shadow_factor = 5 AND timeofday = 'noon' AND value = 900 - For `vul_pop_grid_hd_fct3_noon.tif`: shadow_factor = 3 AND timeofday = 'noon' AND value = 900 The function was applied separately for shadow_factor = 5 and shadow_factor = 3, using `zensus_2022_hd_LAEA.tif` as the input layer and `vul_areas_transport_all` with the "Selected features only" option enabled. The results were saved as `vul_pop_grid_hd_fct5_noon.tif` and `vul_pop_grid_hd_fct3_noon.tif.`

- Figure 5 matches the paper submission. The legend seems to be manually edited afterwards.

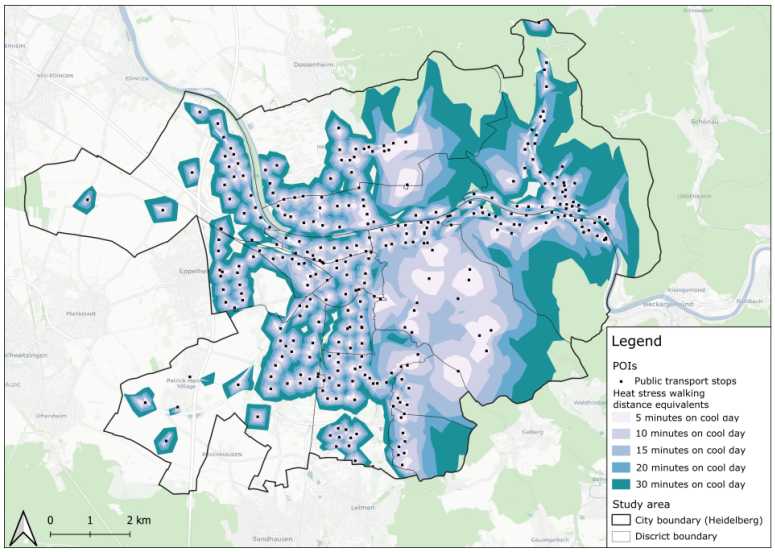
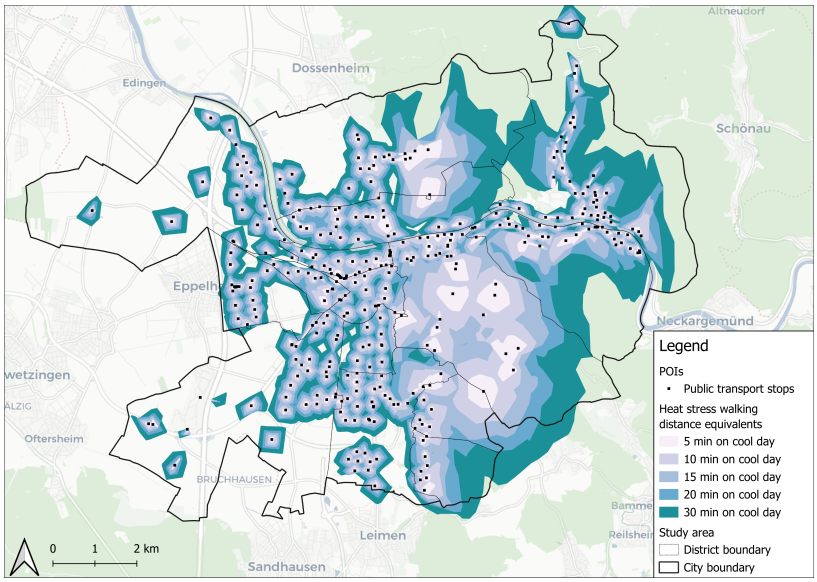


Figure 2. Areas accessible for different *heat stress walking distance equivalent*. Parameter: *heat sensitivity factor* = 5, time of day: afternoon (16:00 h CEST), Day 170 (June 18th/19th). Data source: OSM Contributors, Map Style by CartoDB.

(a) From submitted manuscript



(a) From reproduction attempt

Figure 2

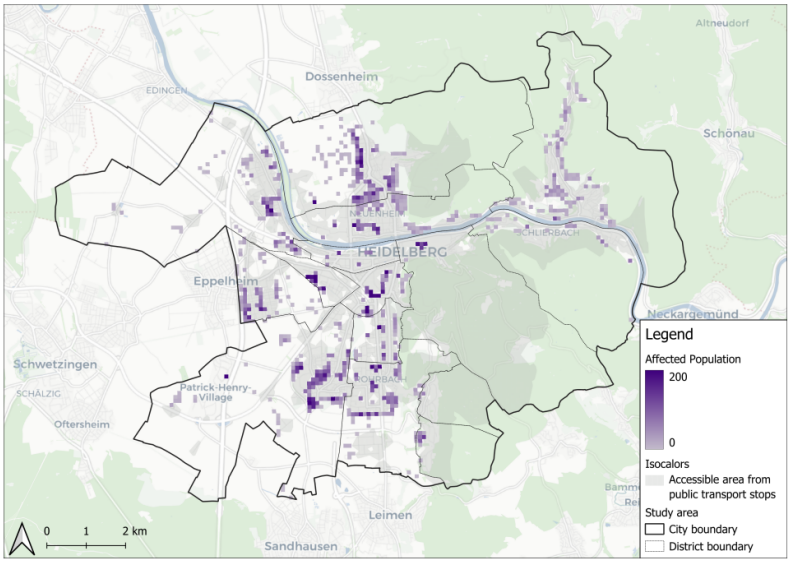
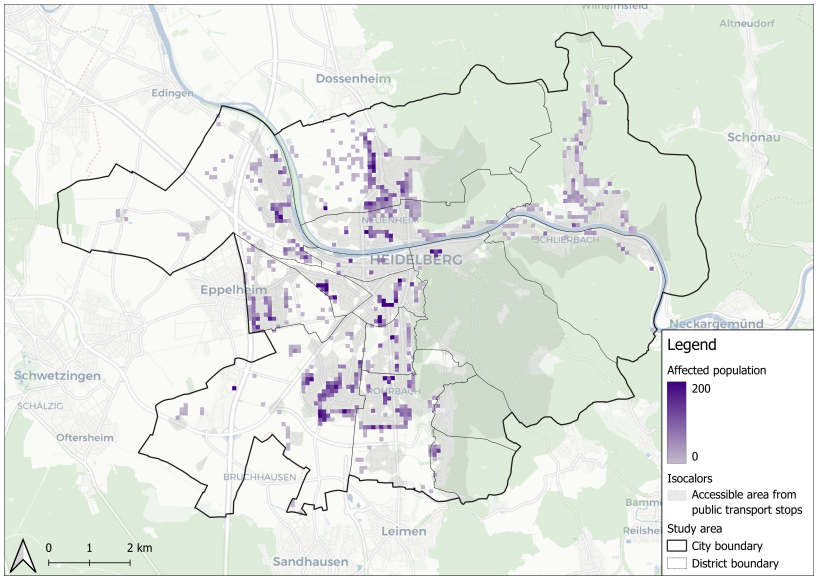


Figure 3. Population without access to at least one public transport stop under heat stress conditions with *heat sensitivity factor* 5 (very high heat sensitivity). The residential population is shown by registration address in 100 x 100 meter grids of the 2022 census. Heat stress conditions: very high heat sensitivity (Hsf 5), noon (13:00, CEST), *heat stress walking distance equivalent* 15 min. Data source: Census 2022, OSM contributors, map style by CartoDB.

(a) From submitted manuscript



(a) From reproduction attempt

Figure 3

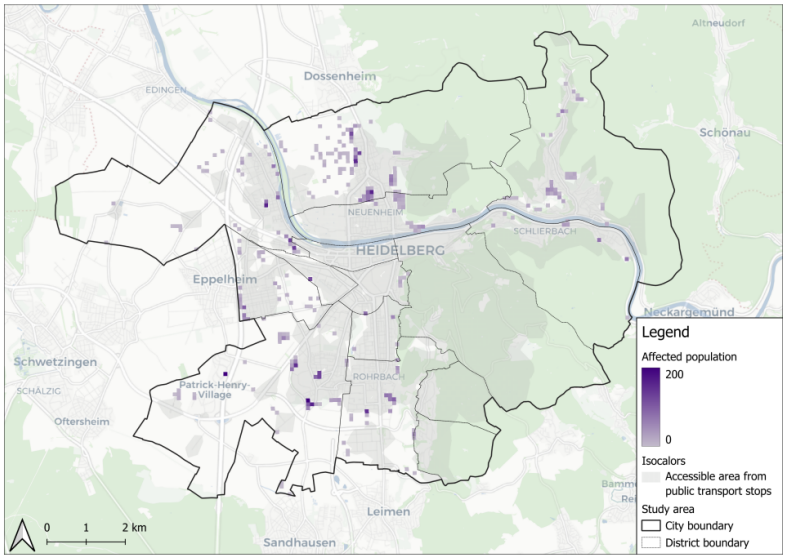
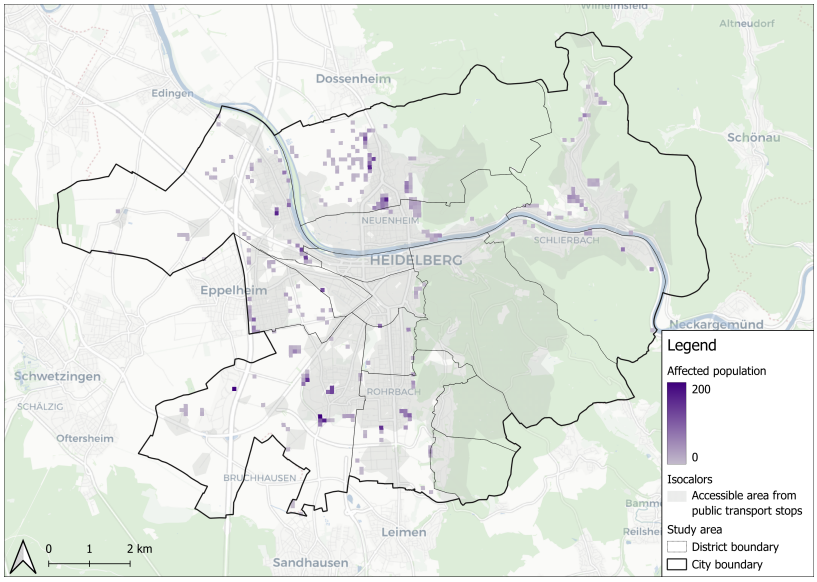


Figure 4. Population without access to at least one public transport stop under heat stress conditions with *heat sensitivity factor* 3 (moderate heat sensitivity). The residential population is shown by registration address in 100 x 100m grids of the 2022 census. Heat stress conditions: moderate heat sensitivity (Hsf 3), noon (13:00, CEST), *heat stress walking distance equivalent* 15 min. Data source: Census 2022, OSM contributors, map style by CartoDB.

(a) From submitted manuscript



(a) From reproduction attempt

Figure 4

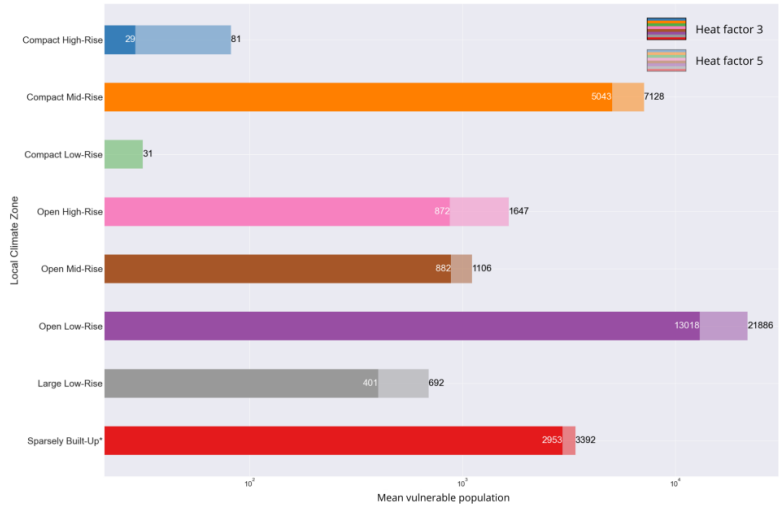
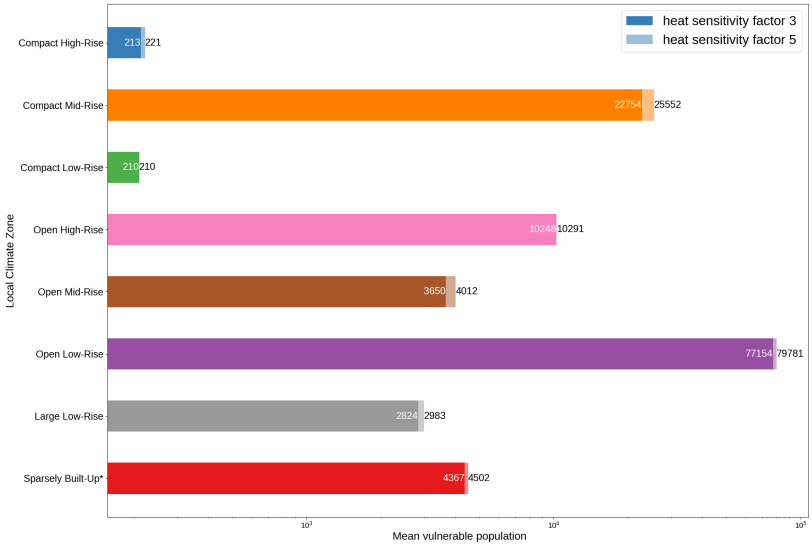


Figure 5. Absolute number of people affected for *heat sensitivity factors* 3 and 5 averaged by local climate zone. The bars show values averaged across the four times of day (10:00, 13:00, 16:00, 19:00 CEST) and the *heat stress walking distance equivalents* for 5, 10, 15, 20, 30 minutes. LCZ 9 (Sparsley Built-Up) is of limited significance due to limitations in the data base. Representation of *heat sensitivity factor* 3: completely opaque, *heat sensitivity factor* 5: semitransparent. Absolute numbers in white font for Hsf 3, in black font for Hsf 5. Please note the logarithmic scale used for the x-axis.

(a) From submitted manuscript



(a) From reproduction attempt

Figure 5

Recommendations

- Including the library versions and OS environments for the analysis can help identify discrepancies between code and pinpoint the cause of errors easily. I recommend using packages such as `session_info` to do so.
- It is smart to provided code that runs on a subset when the analysis takes resources and time to run. However, this should be properly documented in the reproduction steps, in this case, in the repository README.
- If there is data generated without code, an explanation of the steps is of value, to be able to understand and replicate the workflow.

Other than this, the reproduction code is very well documented, had minimal issues and was fairly straightforward to run, supporting in a very good way the analysis.