# Reproducibility review of: OpenStreetMap Suitability Analysis for Wheelchair Routing

Egor Kotov

2026-04-25

## Table of contents

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/8c3vf.

To cite the report use:

Kotov, E. (2026). Reproducibility review of: OpenStreetMap Suitability Analysis for Wheelchair Routing. https://doi.org/10.17605/OSF.IO/8c3vf

**Reviewed paper:**

Majic, I., Gegenleithner, T., and Scholz, J.: OpenStreetMap Suitability Analysis for Wheelchair Routing, AGILE GIScience Ser., 7, 12, https://doi.org/10.5194/agile-giss-7-12-2026, 2026.

| | |
|---|---|
| **Submission Title** | OpenStreetMap Suitability Analysis for Wheelchair Routing |
| **Authors** | Majic, I., Gegenleithner, T., and Scholz, J. |
| **Ref. Paper** | https://doi.org/10.5194/agile-giss-7-12-2026 |
| **Reproducibility Reviewer** | Egor Kotov (iD) |
| **Review Repository** | https://github.com/e-kotov/agile-2026-review-paper-033 |

## Table of contents

## 1 Summary

Following the release of an updated dataset (v2), the reproduction study now covers the full pipeline, including database setup and all analysis notebooks (01 to 04). While the v2 reproducibility package significantly improved the structure and documentation, several surgical fixes were still required to address environment-specific dependency conflicts, undocumented tool behaviors, and logic bugs in the provided scripts.

- **Result of the reproduction**: All notebooks were executed successfully.
- **Main outcome**: Successful database initialization and execution of the complete analysis pipeline for Graz, Linz, and Salzburg.

# 2 Reproducibility reviewer notes

## 2.1 Scope

The review covers the full recomputation of the pipeline as described in the paper's `README.md`, including the cross-city comparison.

## 2.2 Technical Details

Reproduction was performed using a single Apptainer/Docker container containing: - PostgreSQL 16 + PostGIS + pgRouting - osm2pgsql + osm2pgrouting - Python 3.14.3 (Explicitly managed via `uv`)

The environment is based on the **`ghcr.io/osgeo/gdal:ubuntu-small-3.11.5`** Docker image.

### 2.2.1 Host Environment Requirements

The host-side orchestration scripts (e.g., data acquisition, table extraction) require a standard **Python 3** installation. These scripts rely exclusively on the Python standard library; no external Python packages are required on the host system. The project metadata and Python version requirements are formally defined in the included **`pyproject.toml`** file.

Execution of the reproduction pipeline requires either **Docker** or **Apptainer** (formerly Singularity) to run the containerized environment. For report generation, **Quarto** must be installed on the host.

For efficient management of Python environments and tools, the use of `uv` is recommended, though not strictly required. For example, host scripts can be executed seamlessly using `uv run <script_name>.py`.

### 2.2.2 Container Image Build and Distribution

The Docker image was built for the `linux/amd64` platform and pushed to Docker Hub for public access.

```
# Build the image locally for amd64 platform (from repo root)
docker build --platform linux/amd64 -t paper-033 -f containers/Dockerfile .

# Tag and push to Docker Hub
docker tag paper-033 \
  egorkotovdhub/agile-2026-paper-033:latest
docker push egorkotovdhub/agile-2026-paper-033:latest
```

### 2.2.3 HPC/Apptainer Execution

Due to `fakeroot` limitations on standard HPC clusters, local Apptainer builds with `%post` modifications are unsupported. Instead, reviewers should pull the pre-built image directly from Docker Hub:

```
# Pull the pre-built image
module load apptainer
apptainer pull --force \
  repro-reviews/paper-033/containers/paper_033.sif \
  docker://egorkotovdhub/agile-2026-paper-033:latest
```

The container's `entrypoint.sh` is specifically configured to allow local database connections without a password (`trust` authentication) and uses `/tmp` for the PostgreSQL socket to avoid "read-only file system" errors when running within restrictive Apptainer bind mounts. Ensure `PGDATA` is exported to a local directory (e.g., `/work/repro-reviews/paper-033/pgdata`) when executing the container to avoid HPC temporary space limits.

## 2.3 Execution Steps

The reproduction process is automated through a series of scripts. The chronological steps to replicate the results are as follows:

### 2.3.1 Data Acquisition

The dataset must be downloaded from Figshare before running the pipeline. This is automated via a helper script:

```
./scripts/download_data.sh 31333180 repro/DATA
```

### 2.3.2 Build or Pull the Container Image

The environment is defined in `containers/Dockerfile` and `containers/apptainer.def` (which are kept in sync). To build the image locally using Docker or pull the pre-built image using Apptainer:

```
./scripts/build.sh
```

Alternatively, to build the Apptainer `.sif` image locally from the definition file (requires `fakeroot` or root privileges):

```
./scripts/build.sh --local
```

On HPC systems where local builds are restricted, the pre-built image can be pulled directly from Docker Hub as described in the Technical Details section.

#### 2.3.3 Execute the Reproduction Pipeline

The pipeline initializes the database, imports the data, and executes the analysis notebooks. This is handled by an internal script (`scripts/internal_repro.sh`) which is executed automatically inside the container by the following host-side wrapper scripts:

**Via SLURM:**

```
sbatch slurm/run_repro.slurm
```

**Locally (Docker/Apptainer):**

```
./scripts/run.sh
```

#### 2.3.4 Extract Tables, Generate Patch Set, and Render Report

Once the pipeline completes, the results are extracted, all surgical changes are consolidated into a patch file, and the final report is rendered.

```
# Extract LaTeX tables from executed notebooks
python3 scripts/extract_tables.py

# Generate the reproducibility patch set
./scripts/generate_repro_patchset.sh

# Render the final Quarto report
cd report && quarto render report.qmd
```

### 2.4 Resource Utilization and Runtime

The reproduction study was conducted on a Scientific Compute Cluster (SCC) using SLURM for resource management. The following resources were utilized for the full v2 pipeline:

- **CPUs**: 2 allocated.
- **Memory**: 8 GB requested, 8.00 GB peak resident set size (MaxRSS) utilized.
- **Wall Clock Time**: 1 hour 19 minutes and 23 seconds.
- **Total Consumption**: ~2.6 core-hours.

# 3 Reproduction Efforts

During the reproduction study of the v2 package, the following technical observations and adjustments were made to ensure the pipeline executed successfully in the reproduction environment:

## 3.1 1. Dependency and Format Alignment

The authors' updated `requirements.txt` included strict version pins (e.g., `geopandas==1.1.2`, `pandas==2.3.3`) that were initially incompatible with older base images. To achieve full alignment, the container was built using a modern **GDAL 3.11.5 / Ubuntu** base and explicitly installs **Python 3.14.3** via `uv`. All required dependencies, including those omitted by the authors (like `setuptools`), are now pre-installed in the container's virtual environment.

A significant reproducibility hurdle was discovered in the data import phase: both `osm2pgsql` 1.11.0 and 2.x default to a "new" middle database format that uses `jsonb` for the tags column in slim mode. This broke the authors' SQL enrichment code, which depends on the non-default `legacy` format (producing `text[]` columns) to perform `hstore` conversions. To resolve this, the import scripts were surgically patched to include the `--middle-database-format legacy` flag.

## 3.2 2. Elevation Subquery Fragility

In `setup.sql`, the queries used to extract elevation values for street segments (lines 72-81) failed with the error: `more than one row returned by a subquery used as an expression`. This occurred when a segment endpoint intersected multiple polygons in the grid generated from the high-resolution DEM (e.g., at cell boundaries). The queries were patched with `LIMIT 1` to ensure a deterministic single-value return.

## 3.3 3. Database Setup Mismatch

The authors' `setup.sh` script contained a redundant call to `comparison.sql` targeting the `fire_horse` database. Because `comparison.sql` expects prefixed tables (e.g., `graz_ways`) that only exist in the separate `osm_comparison` database, this caused the script to fail. The redundant call was removed from `setup.sh`, as the comparison database is already correctly handled by `import_comparison.sh`.

## 3.4 4. Path and Directory Structure Alignment

The Figshare v2 download provided a flat file structure, whereas the authors' scripts and notebooks expected data to be organized in `DATA/`, `fonts/`, and `Comparison/` subdirectories. The workspace was restructured to match these expectations before execution.

## 3.5 5. Numerical Stability (Notebook 02)

Surgical fixes were applied to `02_route_profiles.ipynb` to handle edge cases in the elevation profiling logic. These included adding checks for empty DataFrames (when no path is found) and ensuring a minimum of 5 data points before applying the Savitzky-Golay smoothing filter, preventing runtime crashes on short or non-existent route segments.

## 3.6 6. Stochasticity and Small Discrepancies

As noted in the v1 review, the set of 1,000 routes in Experiment 3 (`03_routing_analysis.ipynb`) is generated without an explicit random seed. Consequently, while the general trends remain consistent, the specific values in Tables 10-12 differ slightly from the paper. Similarly, Figure 9b (Gradient Distribution) shows minor mean variations (3.60% reproduced vs 3.33% in paper) likely due to stochasticity in the OSM data import and routing topology generation.

### 3.6.1 Summary of Surgical Edits (Patches)

To achieve successful execution and full results extraction of the v2 package, surgical edits were applied to the following files:

- **`setup.sh` & `import_comparison.sh`**: Added `--middle-database-format legacy` to all `osm2pgsql` commands to ensure data type compatibility with the authors' SQL.
- **`setup.sql`**: Added `LIMIT 1` to elevation subqueries to prevent execution crashes on DEM cell boundaries.
- **`01_data_exploration.ipynb`**: Restored the Wordcloud generation cell and fixed x-axis legibility for Figure 9b.
- **`02_route_profiles.ipynb`**: Implemented empty dataframe validation and numerical stability checks for the Savitzky-Golay filter.
- **`03_routing_analysis.ipynb`**: Fixed redundant slope multiplication in statistics.
- **`requirements.txt`**: Added version pin for `geoalchemy2==0.17.1` to ensure dependency availability.

All surgical changes made to the author's original files are documented and consolidated in the **`repro_patches.diff`** file located in the root of the reproduction repository.

## 3.7 Recommendations for Future Reproducibility

Based on the efforts required for the v2 package, the following additional recommendations are offered:

1. **Full Toolchain Specification**: When relying on non-default behaviors of system tools (like `osm2pgsql` legacy format), explicitly document the required flags in the `README.md` or provide a pre-configured setup script.

2. **Defensive SQL**: When using subqueries to fetch values for table updates or creation, always include `LIMIT 1` or appropriate aggregation (e.g., `AVG`, `MAX`) if there is any possibility of multiple spatial intersections.
3. **Defensive Analysis Logic**: Explicitly handle edge cases in notebooks, such as empty results from spatial queries or insufficient data for signal processing filters.

# 4 Acknowledgements

This work used the Scientific Compute Cluster at GWDG, the joint data center of Max Planck Society for the Advancement of Science (MPG) and University of Göttingen. In part funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 405797229

# 5 Reproduced Figures

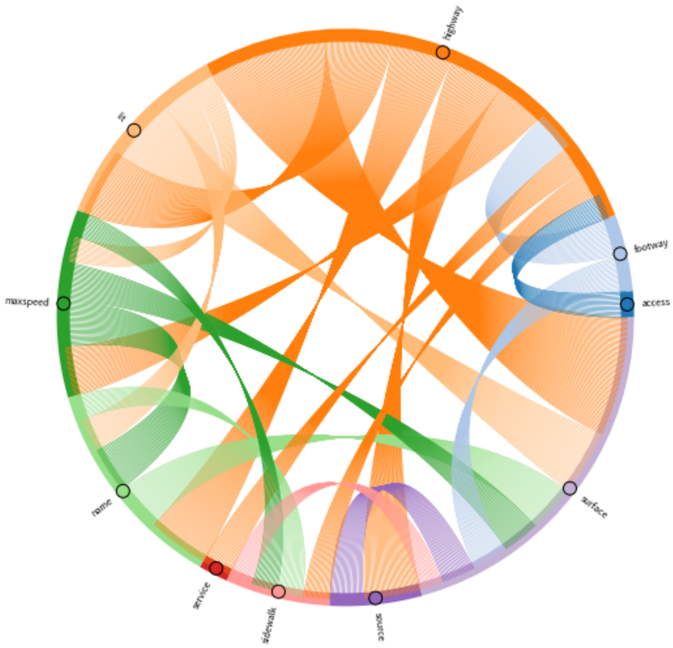
This section presents the results of the reproduction for each figure in the original paper. Reproduced images are set to a medium size (approximately half an A4 page). Figures with discrepancies are cross-referenced to the "Reproduction Efforts" section.

## 5.1 Data Exploration (Notebook 01)

### 5.1.1 Figure 1

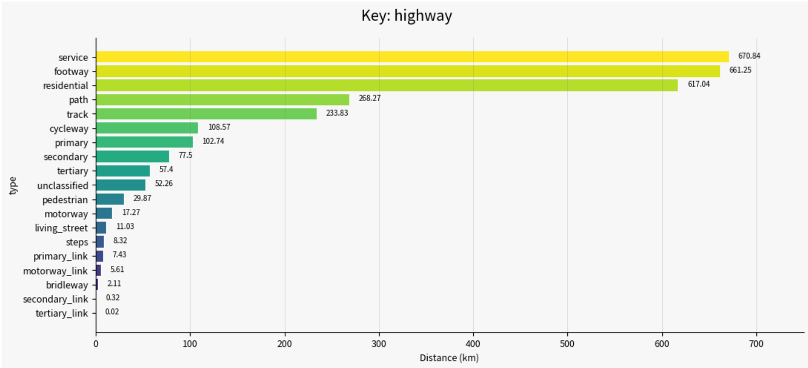
*Not reproducible (Conceptual schematic of wheelchair requirements).*

### 5.1.2 Figure 2




**Status**: Success.

### 5.1.3 Figure 3: Types of roads (key highway)

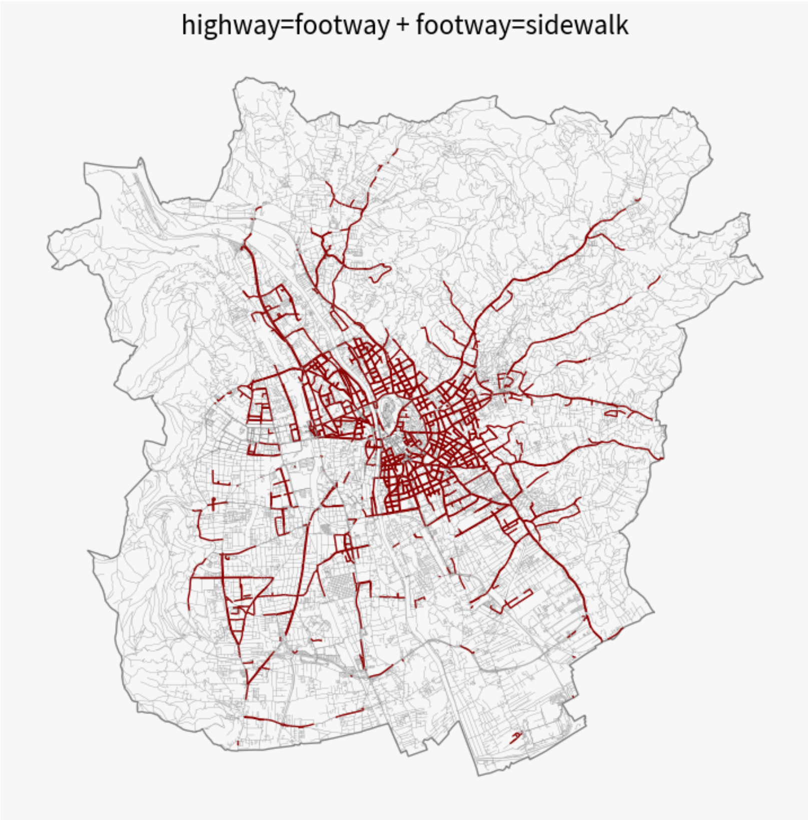



**Status**: Success.

### 5.1.4 Figure 4a: Sidewalk mapping (sidewalk=*)


**Status**: Success.

### 5.1.5 Figure 4b: Sidewalk mapping (footway=sidewalk)




**Status**: Success.

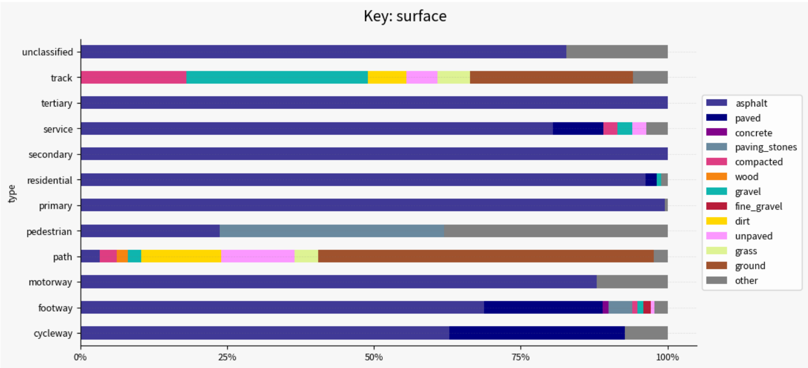
### 5.1.6 Figure 5a: Sidewalk Mapping (Crossings)


**Status**: Success.

### 5.1.7 Figure 5b: Sidewalk Mapping (Footway)

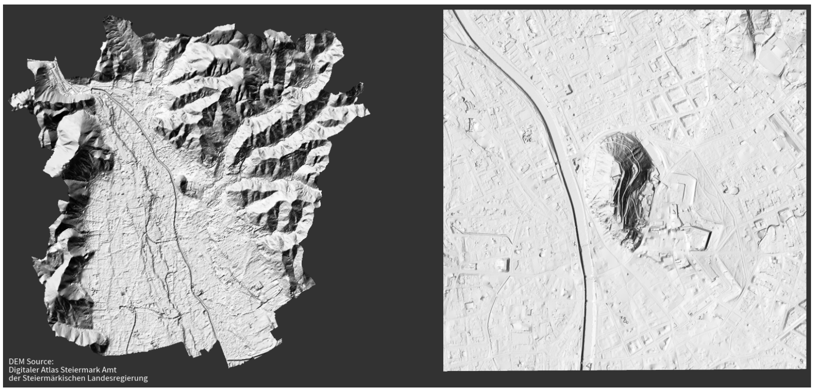

**Status**: Success.

### 5.1.8 Figure 6: Surface per road type



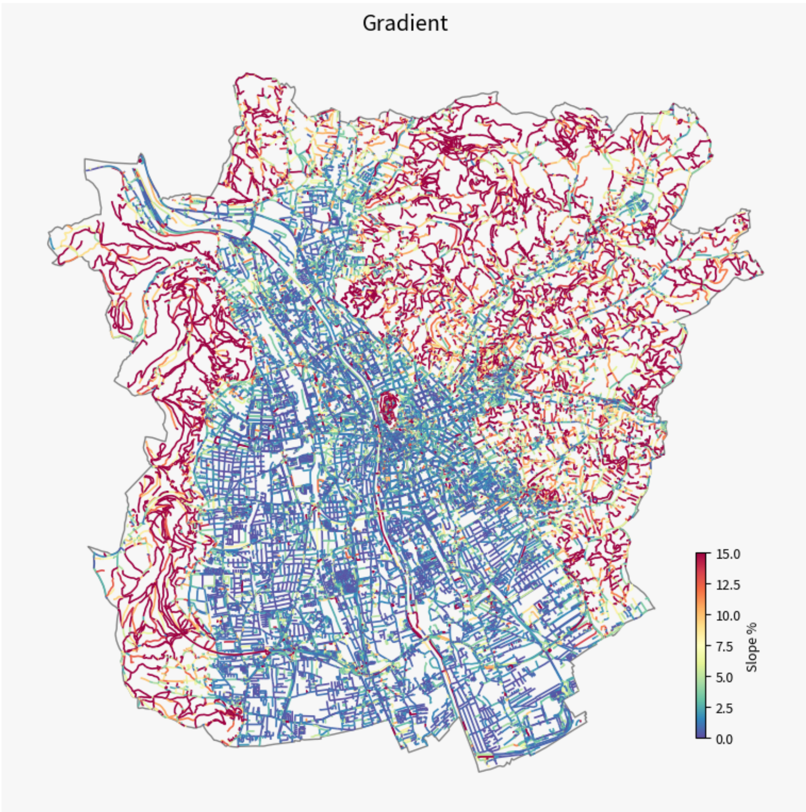

**Status**: Success.

### 5.1.9 Figure 7: DEM Graz



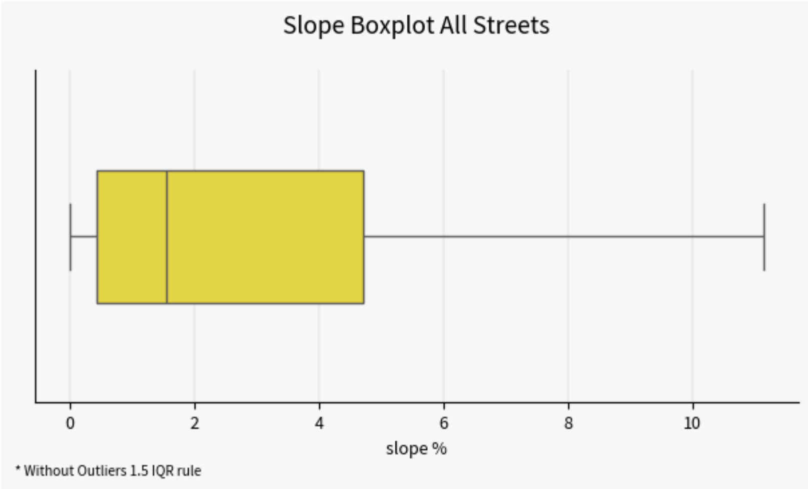

**Status**: Success.

### 5.1.10 Figure 8: Gradient Visualization




**Status**: Success.

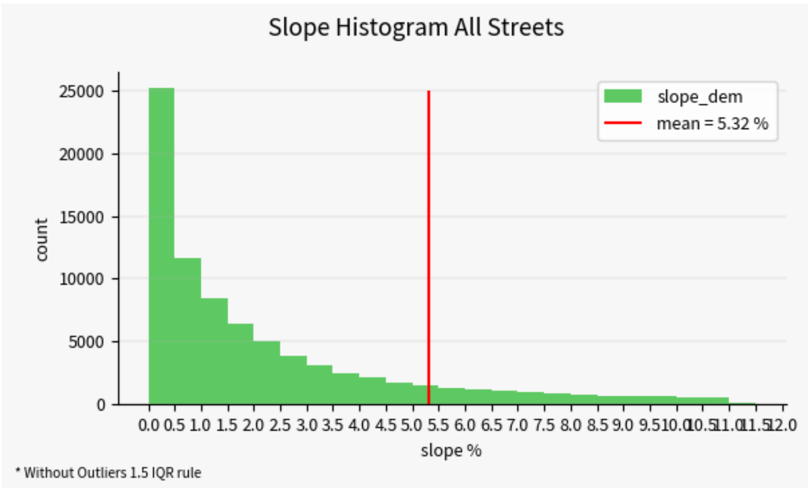
### 5.1.11 Figure 9a: Gradient Distribution (Boxplot)




**Status**: **Issues**.

**Reproduction Note**: Figure shows slightly different range of values compared to the figure in the original paper. Specifically, the upper whisker in the reproduced plot is a bit over 8, while in the original plot it is at 8 or just below it.

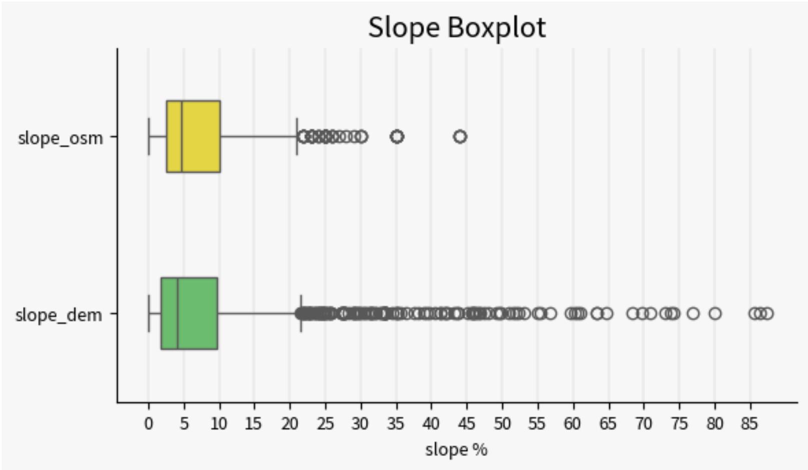
### 5.1.12 Figure 9b: Gradient Distribution (Histogram)




**Status**: **Minor Issues**.

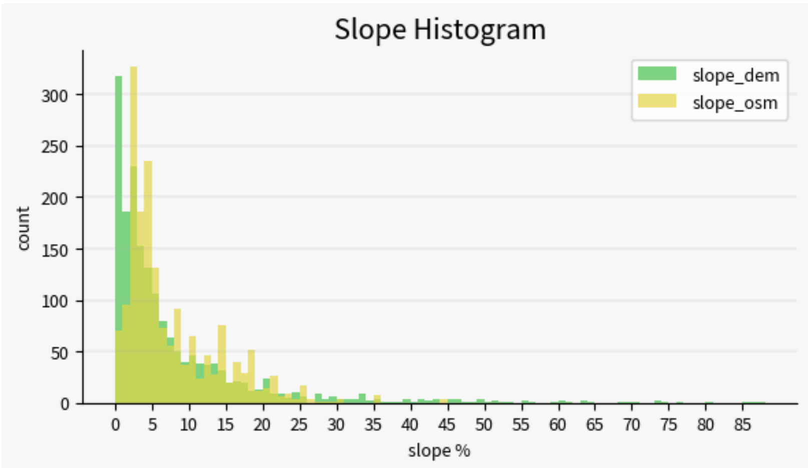
**Reproduction Note**: The reproduced mean (5.32%) differs significantly from the reported value in the paper (3.33%), reflecting an internal inconsistency in the authors' code logic (see point #12). This study only updated the x-axis tick interval to ensure legibility (see point #11); otherwise, the diverging result is produced exactly by the authors' original computational logic.

#### 5.1.13 Figure 10a: OSM Incline vs. DEM (Boxplot)



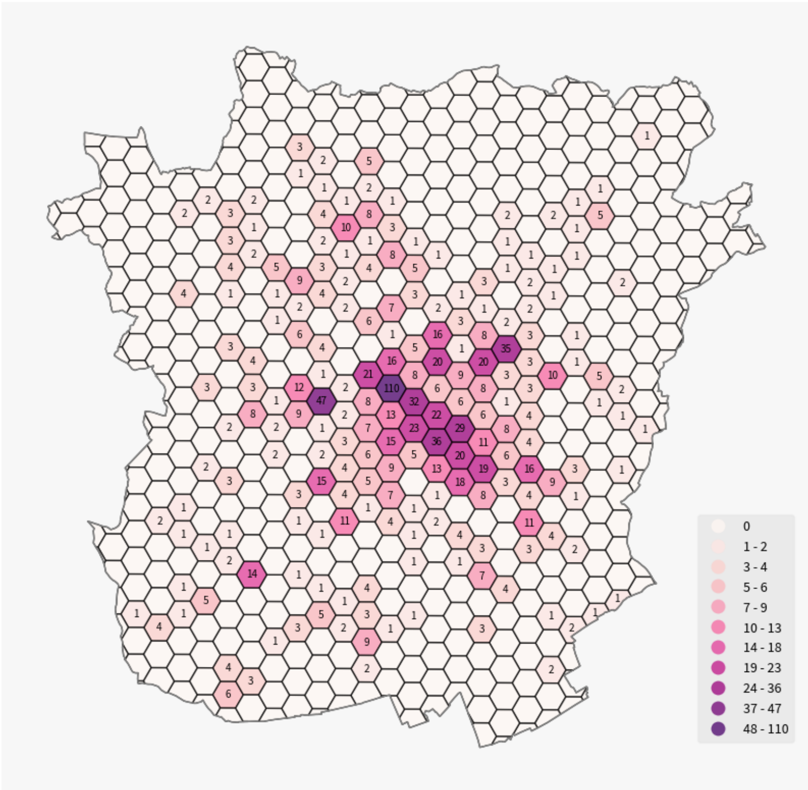

**Status**: Success.

#### 5.1.14 Figure 10b: OSM Incline vs. DEM (Histogram)



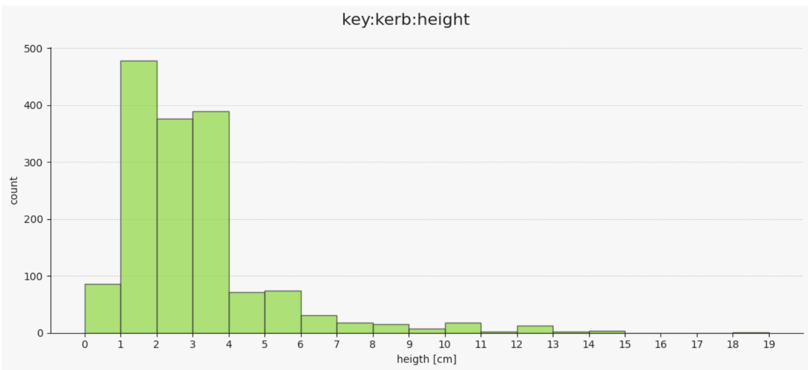

**Status**: **Issues**. **Reproduction Note**: The histogram distribution differs from the figure in the original paper. Figure 10 does not appear to be explicitly referenced or described in the paper text.

### 5.1.15 Figure 11: Hexagon map of steps



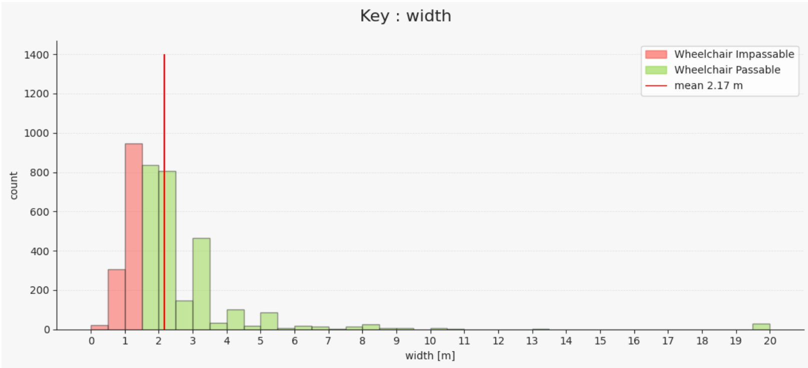

**Status**: Success.

### 5.1.16 Figure 12: Heights of kerbs



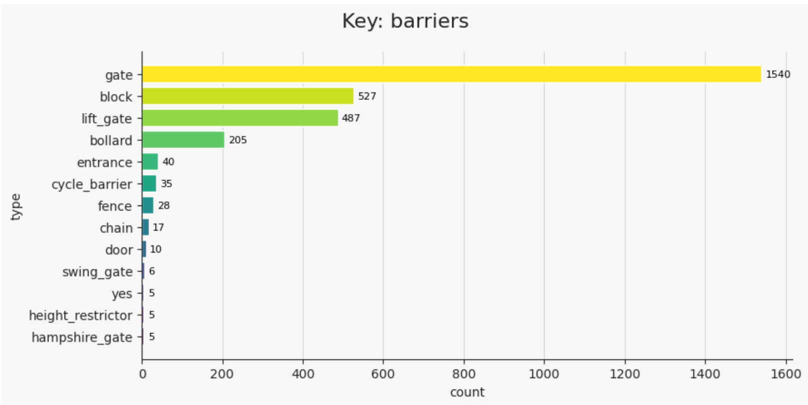

**Status**: Success.

### 5.1.17 Figure 13: Width information




**Status**: Success.

### 5.1.18 Figure 14: Barriers in Graz



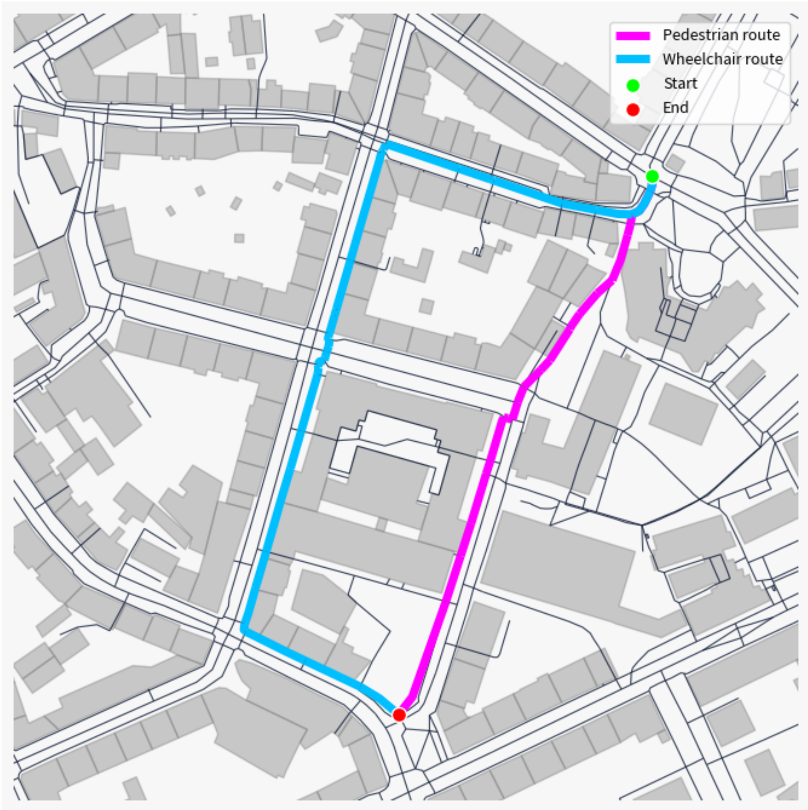

**Status**: Success.

### 5.1.19 Figure 15

*Not reproducible (Photograph of critical location).*

## 5.2 Route Profiles & Case Studies (Notebook 02)

### 5.2.1 Figure 16: Experiment 1 Routes



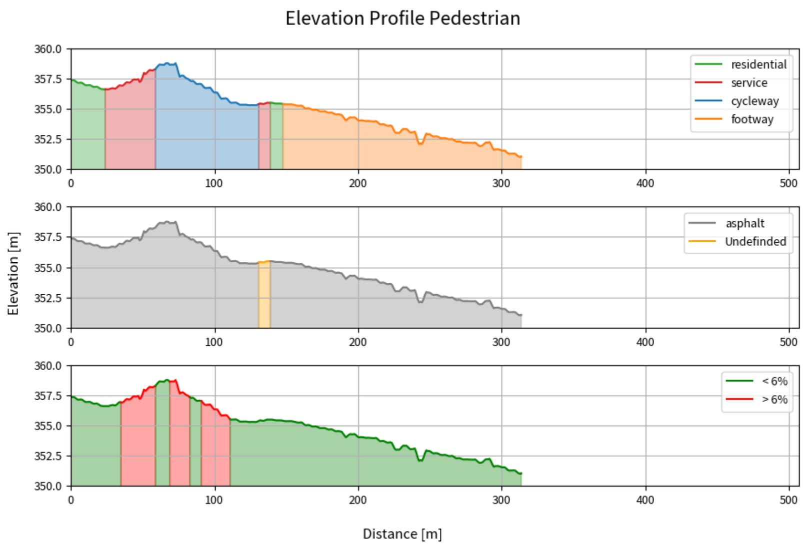

**Status**: Success.

### 5.2.2 Figure 17: Experiment 1; Elevation Profile Wheelchair

*(Skipped: The resulting route array was empty as* `pgr_dijkstra` *failed to find a valid path between the specified nodes in the strictly constrained wheelchair network).*

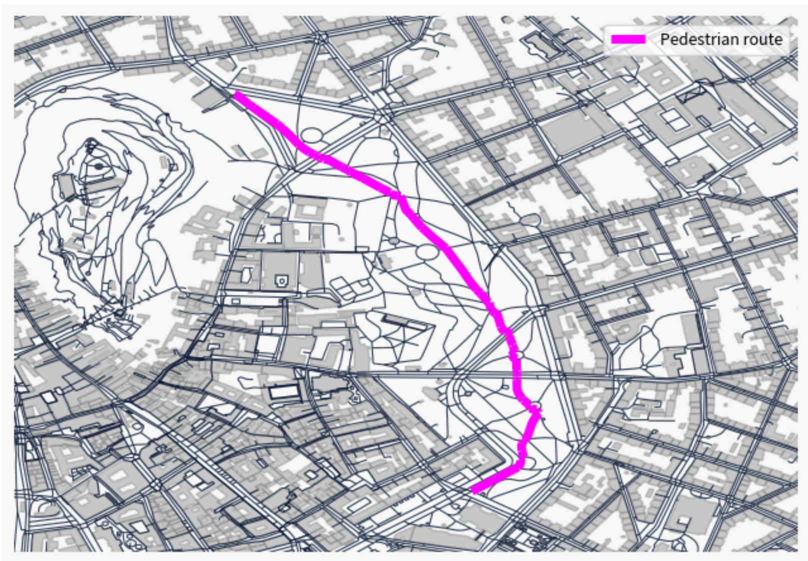
**Status**: **Issues**.

### 5.2.3 Figure 18: Experiment 1; Elevation Profile Pedestrian



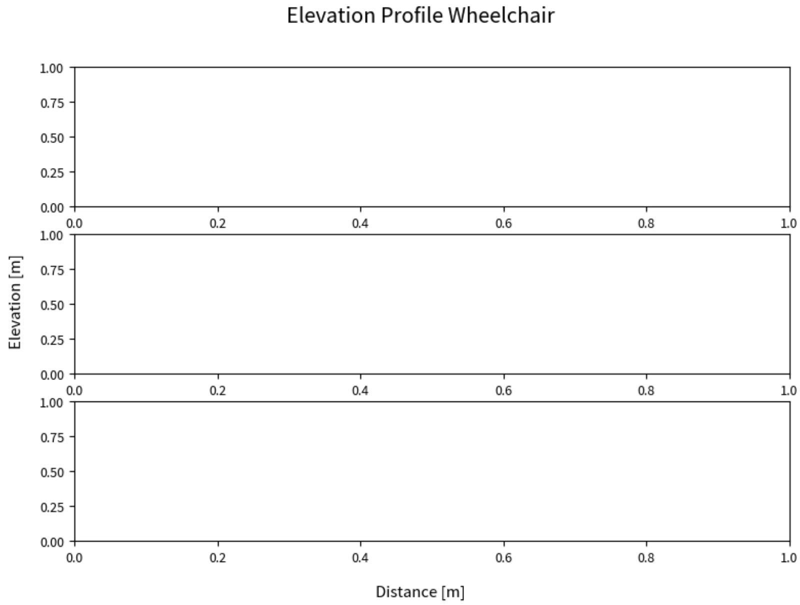

**Status**: **Issues**.

### 5.2.4 Figure 19: Experiment 2 Routes



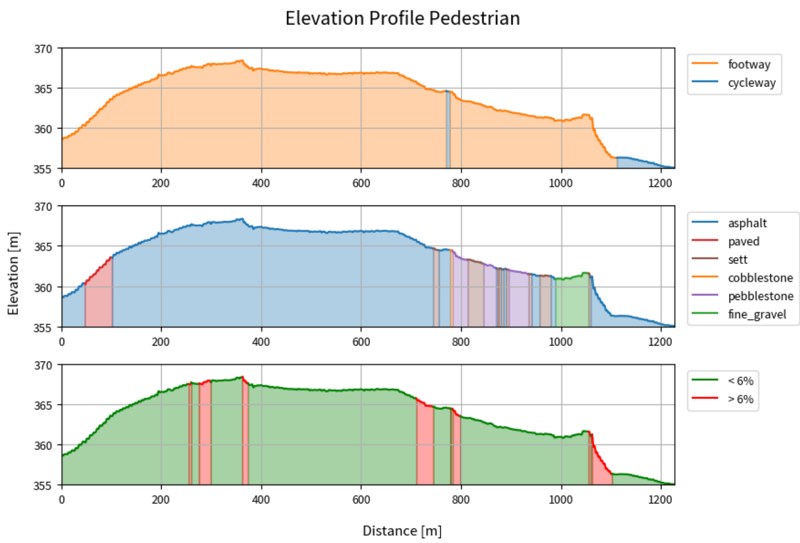

**Status**: **Issues**. **Reproduction Note**: Severe map distortion caused by a CRS mismatch in the plotting code (buildings in 4326 vs. routes in 3857). Furthermore, logic errors in the notebook led to missing routes and points (see point #7 in Reproduction Efforts).

### 5.2.5 Figure 20: Experiment 2; Elevation Profile Wheelchair




**Status**: **Issues**. **Reproduction Note**: The plot is empty as the resulting route array contained no data during this execution (see point #5).

### 5.2.6 Figure 21: Experiment 2; Elevation Profile Pedestrian



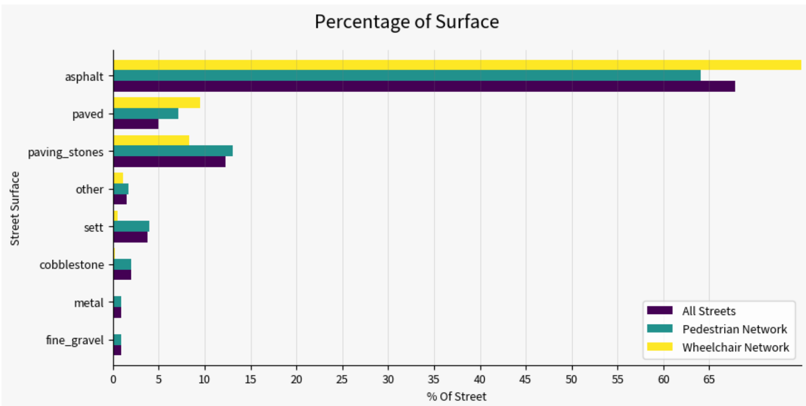

**Status**: Success.

## 5.3 Routing Analysis (Notebook 03)

### 5.3.1 Figure 22: Overview Map Experiment 3


**Status**: **Issues**. **Reproduction Note**: Shows a different set of rendered lines compared to the paper because the 1000 route combinations were generated dynamically without a fixed random seed (see point #8).

### 5.3.2 Figure 23: Detailed Surface Conditions

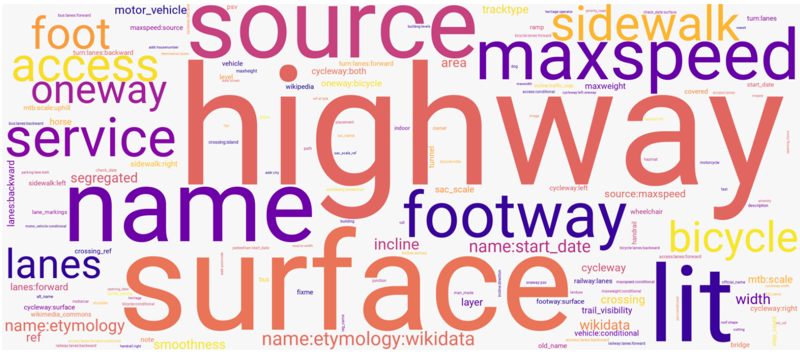


**Status**: **Issues**. **Reproduction Note**: Different order of horizontal bars, values don't match with the figure in the paper.

## 5.4 Additional Figures (Not in Paper)

The following figures were generated by the notebooks but do not have a direct corresponding figure number in the original paper.

### 5.4.1 Tag Wordcloud




**Status**: Success.

## 5.5 Cross-City Comparison

The following figures correspond to the city comparison section of the paper, contrasting the network characteristics of Graz, Linz, and Salzburg within a 1500m buffer from the city centers.

### 5.5.1 Figure 24: City Comparison - All streets


**Status**: Success.

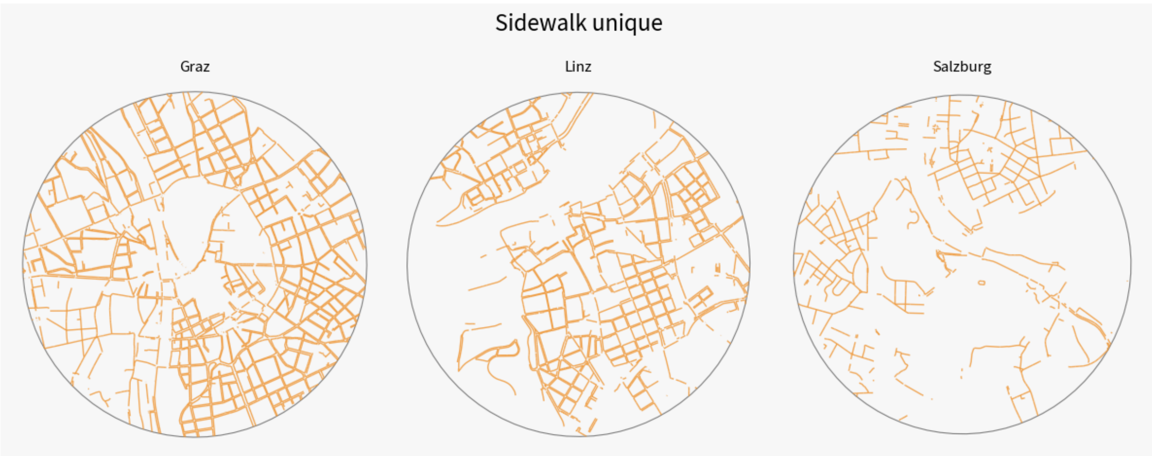
### 5.5.2 Figure 25: City Comparison - Sidewalk=*


**Status**: Success.

### 5.5.3 Figure 26: City Comparison - Highway = Footway and Footway = Highway


**Status**: Success.

### 5.5.4 Figure 27: City Comparison - Sidewalk unique




**Status**: Success.

### 5.5.5 Figure 28: City Comparison - Hexagon Map Sidewalk


**Status**: Success.

## 6 Reproduced Tables

The following tables were generated by the reproduction pipeline in `03_routing_analysis.ipynb` and `04_city_comparison.ipynb` and correspond to the statistical summaries in the original paper. To update these tables from the executed notebook, run `python3 scripts/extract_tables.py` from the paper root directory.

**Note on Table Reproducibility**: - **Tables 1, 2, and 5**: Conceptual or summary tables not intended for computational reproduction. - **Tables 3, 4, 6, 7, 8, and 9**: Statistical or route summaries for which no reproduction code was provided in the authors' notebooks. - **Tables 10, 11, 12, and 13**: Successfully reproduced using the Experiment 3 benchmark set and city comparison, with variations noted below.

**Note on Table Values**: The percentages and counts in these tables are derived strictly from the set of 1000 routes generated during the Experiment 3 benchmark (or the full buffer for Table 13). Because the route selection is non-deterministic (see point #8 in Reproduction Efforts), these values differ from the original paper, although they reflect the same general trends in network suitability.

### Table 10: Analysis of Surface Types for Three Distinct Network Categories

| surface | All Streets | Pedestrian Network | Wheelchair Network |
|---|---|---|---|
| asphalt | 67.8% | 64.0% | 77.3% |
| paving_stones | 12.3% | 13.0% | 8.3% |
| paved | 5.0% | 7.1% | 9.5% |
| sett | 3.8% | 3.9% | 0.5% |
| cobblestone | 2.0% | 2.0% | 0.2% |
| other | 1.5% | 1.7% | 1.1% |
| fine_gravel | 0.9% | 0.9% | 0.0% |
| metal | 0.9% | 0.9% | 0.0% |

**Table 11: Analysis Regarding Highway Type for 3 Different Network Types**

| highway | All Streets | Pedestrian Network | Wheelchair Network |
|---|---|---|---|
| footway | 34.9% | 45.1% | 55.2% |
| residential | 20.7% | 17.2% | 17.1% |
| pedestrian | 16.1% | 16.4% | 4.9% |
| primary | 11.9% | 3.1% | 0.0% |
| cycleway | 4.8% | 9.2% | 20.7% |
| secondary | 4.6% | 2.5% | 0.0% |
| service | 4.5% | 4.3% | 1.7% |
| other | 2.0% | 1.5% | 0.1% |

**Table 12: Detailed Kerb Type Analysis for the Three Networks**

| kerb | All Streets | Pedestrian Network | Wheelchair Network |
|---|---|---|---|
| flush | 4265 | 6278 | 8637 |
| lowered | 6948 | 9920 | 10826 |
| no | 253 | 370 | 845 |
| raised | 1064 | 1370 | 0 |
| rolled | 73 | 95 | 0 |
| yes | 61 | 55 | 0 |

**Table 13: Comparison of the Three Cities (Graz, Linz, Salzburg)**

| | Graz | Linz | Salzb. | graz_point_1500m | linz_point_1500m | salzburg_point_1500m |
|---|---|---|---|---|---|---|
| 0 | 24094.00 | 15649.00 | 9171.00 | NaN | NaN | NaN |
| 0 | 387457.50 | 357774.17 | 234498.04 | NaN | NaN | NaN |
| 0 | 119547.55 | 87959.87 | 6843.91 | NaN | NaN | NaN |
| 0 | 65183.95 | 14913.98 | 42287.29 | NaN | NaN | NaN |
| 0 | 16.82 | 4.17 | 18.03 | NaN | NaN | NaN |
| 0 | 30.85 | 24.59 | 2.92 | NaN | NaN | NaN |
| 0 | NaN | NaN | NaN | 0.00 | 0.00 | 0.00 |