PUBLIC TRANSPORTATION AND ANALYSIS

Phase 2: Innovation

Intoduction:

Transportation analysis provides the basis for transportation planning. Transportation planning is a process of finding feasible alternatives and components of a transportation system to support human activities in a community.

[Core features:](https://github.com/CxAalto/gtfspy#core-features)

* Import one or multiple GTFS feeds into one [SQLite](https://www.sqlite.org/) database for efficient querying of the data.
* Augment the sqlite with real walking distances between PT stops using Open Street Map (OSM) data.
* Compute simple statistics for the public transport networks such as number of stops, routes, network length.
* Filter databases spatially and temporally to match your area and time region of interest.
* Perform accessibility analyses using a routing/profiling engine
  + Adapted from the [Connection Scan Algorithm](http://i11www.iti.uni-karlsruhe.de/extra/publications/dpsw-isftr-13.pdf) (CSA).
  + Compute all Pareto-optimal journey alternatives between an origin-destination pair, and summarize connectivity with measures on travel time and number of transfers.
* Produce data extracts in various formats (network edge lists, geojson).

[**Prerequisites**](https://github.com/CxAalto/gtfspy#prerequisites)

* [Python 3.8](https://www.python.org/)
* Supported platforms: Linux, OSX & Windows
* Optional: [git](https://git-scm.com/) is used for development.

[**Install**](https://github.com/CxAalto/gtfspy#install)

[**Linux and Mac OS**](https://github.com/CxAalto/gtfspy#linux-and-mac-os)

pip install gtfspy

[**Windows**](https://github.com/CxAalto/gtfspy#windows)

Windows should work, but has not been tested or and may not be supported as much. Please report problems.

Windows users may need to install Shapely library first. [Download Shapely wheel](https://www.lfd.uci.edu/~gohlke/pythonlibs/#shapely) and then run:

pip install wheel

pip install {path to the Shapely wheel file on your PC}

If you come across the Microsoft Visual C++ 14.0 is required error, you may need to download the latest Microsoft Visual C++ Build Tools. You can download it [from here](https://visualstudio.microsoft.com/cs/downloads/).

After that, continue with:

pip install gtfspy

[**Development quickstart**](https://github.com/CxAalto/gtfspy#development-quickstart)

Use this if you want to be able to edit gtfspy's source code.

git clone git@github.com:CxAalto/gtfspy.git

cd gtfspy/

pip install -r requirements.txt # install any requirements

nosetests . # run tests

Remember to also add the gtfspy directory to your PYTHONPATH environment variable.

[**Examples**](https://github.com/CxAalto/gtfspy#examples)

* [Importing a GTFS feed into a sqlite database](https://github.com/CxAalto/gtfspy/blob/master/examples/example_import.py)
* [TODO: Validate an imported feed](https://github.com/CxAalto/gtfspy/blob/master/examples/example_validation.py)
* [Compute and plot temporal distance profiles between an origin--destination pair](https://github.com/CxAalto/gtfspy/blob/master/examples/example_temporal_distance_profile.py)
* [Visualizing the public transport network on map](https://github.com/CxAalto/gtfspy/blob/master/examples/example_map_visualization.py)
* [Filter GTFS feed spatially and temporally](https://github.com/CxAalto/gtfspy/blob/master/examples/example_filter.py)
* [Extract a network / a temporal network from the GTFS database](https://github.com/CxAalto/gtfspy/blob/master/examples/example_export.py)
* [TODO! Run a simple accessibility analysis pipeline!](https://github.com/CxAalto/gtfspy/blob/master/examples/example_accessibility_analysis.py)

[**Contributing**](https://github.com/CxAalto/gtfspy#contributing)

We welcome contributions as GitHub pull requests. In your pull request, please also add yourself as a contributor in the list below.

[**Versioning**](https://github.com/CxAalto/gtfspy#versioning)

This library is not yet stabilised, and new features are being developed. Thus code organization and interfaces may change at a fast pace. More precise versioning scheme will be decided upon later.

[**Changelog**](https://github.com/CxAalto/gtfspy#changelog)

View the [changelog](https://github.com/CxAalto/gtfspy/blob/master/CHANGELOG.md).

[**Authors**](https://github.com/CxAalto/gtfspy#authors)

[**Package maintainers**](https://github.com/CxAalto/gtfspy#package-maintainers)

* Rainer Kujala
* Richard Darst
* Christoffer Weckström

[**Other contributors**](https://github.com/CxAalto/gtfspy#other-contributors)

* Manuel Rios ([marz7002](https://github.com/marz7002))
* Nils Haglund
* Michaela Ockova ([evelyn9191](https://github.com/evelyn9191))
* You?

[**Licensing**](https://github.com/CxAalto/gtfspy#licensing)

[**Code**](https://github.com/CxAalto/gtfspy#code)

This source code of this project licensed under the MIT License - see the [LICENSE.txt](https://github.com/CxAalto/gtfspy/blob/master/LICENSE.txt) file for details.

[**Example data**](https://github.com/CxAalto/gtfspy#example-data)

The OpenStreetMap data (.osm.pbf file(s) under examples/data) is licenced under the [Open Data Commons Open Database License](https://opendatacommons.org/licenses/odbl/) (ODbL) by the [OpenStreetMap Foundation](http://osmfoundation.org/) (OSMF).

The GTFS data used for the examples is provided by the City of Kuopio (Finland), and have been downloaded from <http://bussit.kuopio.fi/gtfs/gtfs.zip> [data licensed [under CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/deed)].

[**Usage for scientific purposes**](https://github.com/CxAalto/gtfspy#usage-for-scientific-purposes)

If you use this Python package for scientific purposes, please cite our paper

Rainer Kujala, Christoffer Weckström, Miloš N. Mladenović, Jari Saramäki, Travel times and transfers in public transport: Comprehensive accessibility analysis based on Pareto-optimal journeys, In Computers, Environment and Urban Systems, Volume 67, 2018, Pages 41-54, ISSN 0198-9715, <https://doi.org/10.1016/j.compenvurbsys.2017.08.012>.

[**Acknowledgments**](https://github.com/CxAalto/gtfspy#acknowledgments)

* The development of this Python package has benefited from the support by Academy of Finland through the DeCoNet project.
* For running the Java routing, we use the [Graphhopper routing library](https://github.com/graphhopper/graphhopper).

**NOTE:**

File Naming Convention: TechnologyName\_Phase2 After completion upload your file to your private GitHub account. Please give access to your faculty evaluators[ facultyevaluator@gmail.com ] and industry evaluator [ IndustryEvaluator@skillup.online ] to your private GitHub repository for evaluation process