Package: uaengine (via r-universe)

November 4, 2024

Title Routing and aggregation engine for 'Urban Analyst'

Version 0.3.2.009

Description Routing and aggregation engine for 'Urban Transport Analyst'.

License GPL-3

URL https://github.com/UrbanAnalyst/uaengine

BugReports https://github.com/UrbanAnalyst/uaengine/issues

Imports checkmate, cli, geodist, hms, m4ra, methods, parallel, raster, sf, sfheaders

Suggests colourvalues, dodgr, dplyr, fs, geojsonio, gtfsrouter, knitr, mapdeck, memoise, osmdata, rappdirs, RcppParallel, rmarkdown, testthat (>= 3.0.0), tidycensus, withr

LinkingTo cpp11

Remotes urbananalyst/m4ra, hypertidy/geodist

SystemRequirements osmium-tool

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

Repository https://urbananalyst.r-universe.dev

RemoteUrl https://github.com/UrbanAnalyst/uaengine

RemoteRef HEAD

RemoteSha 4ff21d9f62a59e7d42e5607281418be55a2fd13e

Contents

`pop2point` *Assign raster cell population density values to points*

Description

Assign raster cell population density values to points

Usage

```
pop2point(verts, geotiff, normalise = TRUE)
```

Arguments

`verts`	A table of vertices with longitude and latitude coordinates, such as the "verts" component of a **silicate**-class object representing a local street network, generated with **dodgr** function, `dodgr_streetnet_sc()`.
`geotiff`	Path to 'geotiff' file containing population density estimates, and including the area defined by 'net_sc'.
`normalise`	If `TRUE`, normalise results so such of densities at all network vertices is equal to sum of original densities.

Value

A `data.frame` containing all vertices of 'net_sc', and corresponding point estimates of population density.

ua_binary_compare *Statistically compare UA values between two spatial groups*

Description

Spatial groups must be distinguished by a named variable in the "soc" data.table.

Usage

```
ua_binary_compare(city, results_path, soc, bin_var, d = 10)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
results_path	Local path to directory where results are to be written.
soc	Socio-demographic data with an **sf**-format column of polygons for each observed value of target variable.
bin_var	The same of the variable in "soc" used to distinguish binary groups. This must be a logical variable (values as TRUE or FALSE only).
d	One value of 'dlim' parameters used in ua_index or ua_index_batch call to generate UA data. Resultant plot will be based on data for this value.

ua_export *Export results to the form required for the front-end app*

Description

Export results to the form required for the front-end app

Usage

```
ua_export(city, results_path, soc = NULL, dlim = 10, pairwise = FALSE)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
results_path	Local path to directory where results are to be written.
soc	An **sf** `data.frame` object with polygons defining areas in which socio-demographic variables were collated, and a column called "social_index" containing the socio-demographic variable of interest. If not provided, results are not aggregated and simply returned for each original point.
dlim	Return results for specified value(s) of dlim only. If only one value specified, names of results will not be labelled with actual value.
pairwise	If TRUE, results are aggregated for each pair of variables instead of individually.

Value

The input table, soc, with additional columns of UA index values attached. This table can then be saved and used directly in the UA front-end.

Note

All exported variables are transformed such that lower/negative values are good, while higher/positive values are bad. This requires inverting these variables:

- bike_index
- natural

Those are both on unit scales, so require simple '1 - x' transforms.

`ua_extract_osm` *Extract network data with osmium, convert to* **sc** *format, and collate all results into a single* **osmdata** *object.*

Description

Extract network data with osmium, convert to **sc** format, and collate all results into a single **osmdata** object.

Usage

```
ua_extract_osm(city, path, bbox = NULL, bbox_expand = 0.05, osm_id = NULL)
```

Arguments

city	Name of city (used to name resultant files).
path	Path to a local '.pbf' or '.bz2' file with OpenStreetMap data for nominated city.
bbox	Optional bounding box within which data are to be extracted. If not given, result includes the entire network within the nominated OSM file. bbox can be either a matrix obtained from the **osmdata** function, getbb (or equivalent), or an object from which a bounding box can be extracted. Objects currently recognised are matrices or arrays, which should have two columns (x and y / longitude and latitude), or an **sf** object from which a bounding box can be extracted. Alternatively, bbox can be a local path to a 'geojson' file containing a single polygonal outline to be used to trim the OSM data.
bbox_expand	A proportional amount by which to extend the limits of the bounding box defined by the bbox argument, defaulting to 5%.
osm_id	In lieu of a bounding box, the ID of an Open Street Map object (generally a relation) can be used to provide the boundary to trim the OSM file.

ua_index *Calculate "UA_index" for a specified set of origin locations.*

Description

Calculate "UA_index" for a specified set of origin locations.

Usage

```
ua_index(
  city,
  gtfs_path,
  osm_path = NULL,
  popdens_geotif = NULL,
  from = NULL,
  initial_mode = "foot",
  final_mode = "foot",
  soc = NULL,
  soc_var = NULL,
  dlims = c(5, 10),
  duration_max = 15,
  quiet = FALSE
)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
gtfs_path	Path to `.Rds` version of GTFS feed for specified city.
osm_path	Path to OSM data processed by ua_extract_osm function (only needed first time to call internal `ua_prepare_data()` function).
popdens_geotif	Optional path to local 'geotiff' file with population density estimates. If provided, all transport indices are adjusted to account for effects of local population density.
from	Character vector of Open Street Map IDs of vertices from which to calculate travel times.
initial_mode	Initial mode of transport from each 'from' point to public transport system (or to destination points, where single-model transport is faster).
final_mode	Final mode of transport from each public transport destination point to all other network nodes.
soc	Socio-demographic data with an **sf**-format column of polygons for each observed value of target variable.
soc_var	Name of target variable in `soc` data set.

dlims	Vector of distance limits in kilometres over which average values of transport index should be calculated. One value of "ua_index" is then derived for each value of dlims.
duration_max	Parameter to control maximal duration examined in multi-modal routing queries, specified in minutes. This value must be strictly greater than the maximal value of dlims, and is used to speed up multi-modal queries. May be set to NULL to return full results from all possible durations / distances.
quiet	If FALSE, display progress information on screen.

Value

An **sf** data.frame, with geometries of points defined by from, and values for transport indices at specified distances, both raw values and values adjusted for local population densities, along with corresponding values of the UA index at each value of 'dlim'.

---

ua_index_batch *Batch calculation of 'ua_index' for a single city*

Description

Repeatedly calculates values from the ua_index function for a sub-set of all origin vertices within nominated city, and saves results to a local file. Finally results can generally be reliably interpolated, and it is thus generally not necessarily to calculate ua_index for every origin point. Instead, it is recommended to only calculate values for a fixed proportion specified by the parameter, coverage.

Usage

```
ua_index_batch(
  city,
  gtfs_path,
  popdens_geotif = NULL,
  results_path = NULL,
  initial_mode = "foot",
  final_mode = "foot",
  soc = NULL,
  soc_var = NULL,
  dlims = c(5, 10),
  batch_size = 1000,
  coverage = 1/4,
  nthreads = NULL
)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
gtfs_path	Path to .Rds version of GTFS feed for specified city.

popdens_geotif	Optional path to local 'geotiff' file with population density estimates. If provided, all transport indices are adjusted to account for effects of local population density.
results_path	Local path to directory where results are to be written.
initial_mode	Initial mode of transport from each 'from' point to public transport system (or to destination points, where single-model transport is faster).
final_mode	Final mode of transport from each public transport destination point to all other network nodes.
soc	Socio-demographic data with an **sf**-format column of polygons for each observed value of target variable.
soc_var	Name of target variable in soc data set.
dlims	Vector of distance limits in kilometres over which average values of transport index should be calculated. One value of "ua_index" is then derived for each value of dlims.
batch_size	Number of vertices to calculate in each batch before saving to local file. Increases in this parameter can greatly increase the sizes of intermediate results, leading to memory problems. Note that batch runs will only be able to stopped and restarted if this parameter is kept constant. Changes to this parameter will cause re-started runs to revert to the beginning, and will generate results which overlap with any previous results still retained in results_path.
coverage	Proportion of all points for which values are to be calculated. As described above, coverage can generally safely be set to values less than 1, with complete coverage then generated through the more computationally efficient ua_interpolate function.
nthreads	Optional parameter to specify number of threads to be used in multi-threaded calculations. Default is maximum number of available threads (from RcppParallel::defaultNumThrea

Note

Importantly, this function can safely be interrupted at any time, and will simply restart at the point where it was stopped.

`ua_interpolate` *Interpolate "UA_index" between vertices on network, project those on to the original network edges, and return the full network graph.*

Description

Interpolate "UA_index" between vertices on network, project those on to the original network edges, and return the full network graph.

Usage

```
ua_interpolate(city, initial_mode = "foot", ua_dat = NULL, soc, npts = 3L)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
initial_mode	Initial mode of transport from each 'from' point to public transport system (or to destination points, where single-model transport is faster).
ua_dat	Result of ua_index function.
soc	Socio-demographic data with an **sf**-format column of polygons for each observed value of target variable, as passed to ua_index function.
npts	Number of nearest neighbours to use to interpolate values at each point.

Value

The full network graph with additional variables quantifying for each edges the values defined in ua_dat as projected on to those edges.

---

ua_osm_to_sc *Convert a series of '.osm' files to* **silicate** *format, and combine all into a single object.*

Description

This function saves a new '.Rds' file in the 'path' directory.

Usage

```
ua_osm_to_sc(path, city)
```

Arguments

path	Path to the directory containing various '.osm' files produced by ua_extract_osm.
city	Name of city (used to name resultant files).

Value

A single **silicate** 'SC' object containing combined data from all individual '.osm' files. This file is also saved as a '.Rds' object in the 'path' directory.

ua_plot_network — *Generate an interactive plot of results from 'ua_interpolate' projected on to street network.*

Description

Generate an interactive plot of results from 'ua_interpolate' projected on to street network.

Usage

```
ua_plot_network(graph, var = "ua_index_d10")
```

Arguments

graph	A street network with values of UA index calculated from ua_index and interpolated back onto network with ua_interpolate.
var	Which variable to plot.

Value

Nothing; function called for its side-effect of opening an interactive visualisation in local default browser.

ua_plot_polygons — *Generate an interactive plot of results in same polygons as original socio-demographic data.*

Description

Generate an interactive plot of results in same polygons as original socio-demographic data.

Usage

```
ua_plot_polygons(
  city,
  results_path,
  soc,
  d = 10,
  what = c("transport_rel", "transport_abs", "social", "ua_rel", "ua_abs"),
  zoom = 10,
  alpha = 0.5
)
```

Arguments

city	Name of city, used to name and define local path to pre-calculated street networks and transport times with **m4ra** package.
results_path	Local path to directory where results are to be written.
soc	Socio-demographic data with an **sf**-format column of polygons for each observed value of target variable.
d	One value of 'dlim' parameters used in ua_index or ua_index_batch call to generate UA data. Resultant plot will be based on data for this value.
what	The value which is to be plotted, one of "transport", "social", "ua_rel" or "ua_abs". The relative UA index ("ua_rel") combines socio-demographic variable, "soc", with the transport index relative to equivalent automobile travel times. The absolute UA index ("ua_abs") combines uses as a transport index the absolute multi-modal travel times for all modes excluding private automobile.
zoom	Initial zoom level to use in resultant map.
alpha	Transparency of polygons (0 = completely; 1 = not transparent).

Value

Nothing; function called for its side-effect of opening an interactive visualisation in local default browser.

---

ua_prepare_data *Additional data preparation beyond standard* m4ra *routines*

Description

Additional data preparation beyond standard m4ra routines

Usage

```
ua_prepare_data(osm_path, water_dist = 20, quiet = FALSE)
```

Arguments

osm_path	Path to directory containing data processed by the ua_extract_osm function for desired city.
water_dist	Distance below which any edges are considered adjacent to water, and thus categorised as "blue space" edges.
quiet	If FALSE, display progress information on screen.

Value

Name of file holding integer index for nominated city identifying which vertices of the foot-weighted network are within or adjacent to natural spaces.

ua_to_geojson *Write UA index data to local 'geojson' file*

Description

Writing large graphs to 'geojson' output can take some time - typically on the order of minutes per 100,000 graph edges.

Usage

```
ua_to_geojson(ua_dat, ndigits = 2L, filename = NULL)
```

Arguments

ua_dat	Full network graph including UA data returned from ua_index and ua_interpolate functions.
ndigits	Number of digits for rounding UA data in 'geojson' representation.
filename	Name and/or full path to local file where 'geojson' data are to be stored.

ua_vertices *Extract vertices within perimiter of external data set*

Description

Extract vertices within perimiter of external data set

Usage

```
ua_vertices(x, city, mode)
```

Arguments

x	External data set, presumed to represent some socio-demographic variable, containing values of defined variable and **sf**-format polygons defining each observation of that variable.
city	Name of city for which vertices are extracted; passed to **m4ra** functions which store pre-processed networks in locations defined by name of 'city'
mode	Mode of transport used to generate network; this should be the same as the 'initial_mode' parameter passed to **m4ra** functions.

Value

A vertex table of only those vertices within the bounding polygon defined by x.

Index