rnet

Release 0.2.0

Kota Sakazaki

May 12, 2022

CONTENTS:

1 Data **1**

1.1 Classes 1

1.2 Containers 4

CHAPTER

ONE

DATA

1.1 Classes

1.1.1 The MapData Class

class `rnet.`**`MapData`**(*vertices*, *links*, ***, *crs*, *name*)

Bases: `rnet.data.classes.Data`

Class for representing map data.

Parameters

- **`vertices`** (`pandas.DataFrame`) – Frame containing vertex data.
- **`links`** (`pandas.DataFrame`) – Frame containing link data.

Keyword Arguments

- **`crs`** (`int`) – EPSG code of the CRS in which vertex coordinates are represented.
- **`name`** (`str`) – Data source name.

Variables

- **`vertices`** (`pandas.DataFrame`) – Frame containing vertex data.
- **`links`** (`pandas.DataFrame`) – Frame containing link data.
- **`vertex_count`** (`int`) – Number of vertices.
- **`link_count`** (`int`) – Number of links.

`bounds`()

Return the coordinates that define the bounding box for the set of vertices.

Returns 4-tuple of the form (`xmin`, `ymin`, `xmax`, `ymax`).

Return type Tuple[float]

`dump`()

Print information about the instance.

classmethod from_osm(*path_to_osm*, ***kwargs*)

Instantiate class from an OSM data source.

Parameters

- **`path_to_osm`** (`str`) – Path to OSM file.

- **name** (str, optional) – Data source name. If unspecified, then the OSM file name is used.

Keyword Arguments

- **include** (List[str], optional) – List of tags to include. All tags are included by default.
- **exclude** (List[str], optional) – List of tags to exclude. No tags are excluded by default.

Note: If required, either the *include* or *exclude* keyword should be given, not both. In the case that both are given, *include* takes precedence and *exclude* is ignored.

out(**, crs=None, include='all', exclude=None*)

Export vertex and link data frames.

Keyword Arguments

- **crs** (int, optional) – EPSG code of CRS for vertex coordinates. If different from `.crs`, coordinates are transformed to *crs*. If None, coordinates are not transformed. Default: None.
- **include** (*'all'* or List[str], optional) – List of tags to include. If 'all', all tags are included. Default: 'all'.
- **exclude** (List[str], optional) – List of tags to exclude. If None, no tags are excluded. Default: None.

Returns 2-tuple containing `.vertices` and `.links` frames with links filtered and vertices transformed.

Return type Tuple[pandas.DataFrame, pandas.DataFrame]

Note: The keyword *include* takes precedence over *exclude*.

1.1.2 The ElevationData Class

class rnet.**ElevationData**(*x, y, z, *, crs, name*)

Bases: `rnet.data.classes.Data`

Class for representing a grid of elevation data points.

Parameters

- **x** (numpy.ndarray, shape $(nx,)$) – x-coordinates of grid.
- **y** (numpy.ndarray, shape $(ny,)$) – y-coordinates of grid.
- **z** (numpy.ndarray, shape (ny, nx)) – Array of z-coordinates.

Keyword Arguments

- **crs** (int) – EPSG code of CRS in which (x, y) coordinates are represented.
- **name** (str) – Data source name.

Variables

- **x** (numpy.ndarray, shape (*nx*,)) – *x*-coordinates of grid.
- **y** (numpy.ndarray, shape (*ny*,)) – *y*-coordinates of grid.
- **z** (numpy.ndarray, shape (*ny*, *nx*)) – Array of *z*-coordinates.
- **nx** (int) – Grid width.
- **ny** (int) – Grid height.
- **point_count** (int) – Number of data points.
- **xmin** (float) – Minimum *x*-coordinate.
- **xmax** (float) – Maximum *x*-coordinate.
- **ymin** (float) – Minimum *y*-coordinate.
- **ymax** (float) – Maximum *y*-coordinate.
- **zmin** (float) – Minimum *z*-coordinate.
- **zmax** (float) – Maximum *z*-coordinate.

bounds()

Return the coordinates that define the three-dimensional bounding box for the set of data points.

Returns 6-tuple of the form (xmin, ymin, zmin, xmax, ymax, zmax).

Return type Tuple[float]

dump()

Print information about the instance.

classmethod **from_tif**(*path_to_tif*, ***kwargs*)

Instantiate class from a TIF data source.

Parameters **path_to_tif** (str) – Path to TIF file.

Keyword Arguments **name** (str, optional) – Data source name. If unspecified, TIF file name is used.

get_elev(*x*, *y*, ***, *r=0.001*, *p=2*)

Return elevation at a single point.

Parameters

- **x** (float) – *x*-coordinate.
- **y** (float) – *y*-coordinate.

Keyword Arguments

- **r** (float, optional) – Radius for neighboring point search. Default: 0.001.
- **p** (int, optional) – Power setting for IDW interpolation. Default: 2.

Returns Elevation at point (x, y).

Return type float

See also:

get_elevs() Returns elevations at multiple points.

get_elevs(*points*, ***, *r=0.001*, *p=2*)

Return elevations at multiple points.

Parameters **points** (numpy.ndarray, shape (*N*, 2)) – The *N* points at which to compute elevations.

Keyword Arguments

- **r** (float, optional) – Radius for neighbor search. Default: 50.
- **p** (int, optional) – Power setting. Default: 2.

Returns

elevations `elevations[i]` is the elevation at `points[i]`.

Return type List[float]

Warning: Points that are outside of the data bounds will return a corresponding elevation value of `numpy.nan`.

See also:

`get_elev()` Returns elevation at a single point.

out()

Export data frame with index `.y`, columns `.x`, and values `.z`.

Return type pandas.DataFrame

query(*i*, *j*)

Return elevation at (`x[i]`, `y[j]`).

Parameters

- **i** (int) – Row number.
- **j** (int) – Column number.

Return type float

Raises **IndexError** – If *i* or *j* is out of bounds.

1.2 Containers

1.2.1 The MapDataContainer Class

class rnet.**MapDataContainer**(*name=None*)

Bases: `rnet.data.containers.DataContainer`

Container for map data.

Parameters **name** (str, optional) – Container name. If None, a name is generated automatically. Default: None.

add(*source*, *crs=None*)

Add map data to the container.

Parameters

- **source** (str or MapData) – Either (1) path to OSM file, (2) path to directory containing `vertices.csv` and `links.csv` pair, or (3) MapData instance.
- **crs** (int, optional) – EPSG code of the CRS in which vertex coordinates are represented. Required only if *source* is of type (2).

out(**, assume_unique=False, crs=4326, include='all', exclude=None*)

Creates a MapData instance containing concatenated frames.

Keyword Arguments

- **assume_unique** (bool, optional) – If True, vertices and links in all data sources are assumed to be unique. If False, data sources are checked for uniqueness and only unique features are retained. Default: False.
- **crs** (int, optional) – EPSG code of CRS for vertex coordinates. Default: 4326.
- **include** (*'all'* or List[str], optional) – List of tags to include. If 'all', all tags are included. Default: 'all'.
- **exclude** (List[str], optional) – List of tags to exclude. If None, no tags are excluded. Default: None.

Returns MapData instance.

Return type *MapData*

See also:

MapData Class for representing map data.

1.2.2 The ElevationDataContainer Class

class rnet.**ElevationDataContainer**(*name=None*)

Bases: `rnet.data.containers.DataContainer`

Container for elevation data.

Parameters **name** (str, optional) – Container name. If None, a name is generated automatically. Default: None.

add(*source*, *crs=None*)

Adds elevation data to the container.

Parameters

- **source** (str or ElevationData) – Either (1) path to TIF file, (2) path to CSV file, or (3) ElevationData instance.
- **crs** (int, optional) – EPSG code of the CRS in which point coordinates are represented. Required only if *source* is of type (2).

out(**, assume_unique=False, crs=4326*)

Creates an ElevationData instance containing concatenated frames.

Keyword Arguments

- **assume_unique** (bool, optional) – If True, points in all data sources are assumed to be unique. If False, data sources are checked for uniqueness and only unique features are retained. Default: False.
- **crs** (int, optional) – EPSG code of CRS for (x, y) coordinates. Default: 4326.

Returns ElevationData instance.

Return type *ElevationData*

See also:

ElevationData Class for representing elevation data.