Package ‘flexpolyline’

August 7, 2020

Type Package
Title Flexible Polyline Encoding
Version 0.1.1
Description Binding to the C++ implementation of the flexible polyline encoding by HERE <https://github.com/heremaps/flexible-polyline>. The flexible polyline encoding is a lossy compressed representation of a list of coordinate pairs or coordinate triples. The encoding is achieved by:
(1) Reducing the decimal digits of each value;
(2) encoding only the offset from the previous point;
(3) using variable length for each coordinate delta; and
(4) using 64 URL-safe characters to display the result.
License GPL-3
URL https://munterfinger.github.io/flexpolyline,
    https://github.com/munterfinger/flexpolyline
BugReports https://github.com/munterfinger/flexpolyline/issues
LinkingTo Rcpp
Imports Rcpp, sf (>= 0.9-3)
Suggests testthat (>= 2.3.2), stringr (>= 1.4.0), knitr (>= 1.28),
    markdown (>= 2.1), covr (>= 3.5.0)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
VignetteBuilder knitr
NeedsCompilation yes
Author Merlin Unterfinger [aut, cre] (<https://orcid.org/0000-0003-2020-2366>),
    HERE Europe B.V. [aut, cph] (Flexible polyline encoding C++ implementation)
Maintainer Merlin Unterfinger <info@munterfinger.ch>
Repository CRAN
Date/Publication 2020-08-07 09:10:10 UTC
R topics documented:

- `decode` .................................................. 2
- `decode_sf` ........................................... 3
- `encode` .................................................. 3
- `encode_sf` ............................................. 4

Description

This function calls `hf::polyline_decode` and `hf::get_third_dimension` of the C++ implementation of the flexible polyline encoding by HERE. Depending on the dimensions of the encoded line, a two or three dimensional line is decoded.

Usage

decode(encoded)

Arguments

- `encoded` character, encoded flexible polyline string.

Value

A matrix containing the coordinates of the decoded line.

Examples

```r
# 2d line
decode("BFoz5xJ67i1B1B7PzIhaxL7Y")

# 3d line
decode("BlBoz5xJ67i1BU1B7PUzIhaUxL7YU")
```
**decode_sf**

*Wrapper function for decoding to simple features*

**Description**

A wrapper function for `decode` that converts the input polylines, encoded in the flexible polyline encoding, to simple feature geometries of the sf package.

**Usage**

```r
decode_sf(encoded, crs = sf::NA_crs_)
```

**Arguments**

- `encoded` character, encoded flexible polyline string.
- `crs` integer or character, coordinate reference system to assign to the sf object (default = `sf::NA_crs_`).

**Value**

An sf object, containing the geometries of the decoded lines (Geometry type: "LINESTRING").

**Note**

The function returns an sf object, therefore the input set of encoded polylines must be of consistent dimension (e.g. "XY", "XYM" or "XYZ") to meet the requirements of the constructor of sf objects. For mixed dimensions use the `decode` function directly.

**Examples**

```r
# 2d line
decode_sf("BFoz5xJ67i1B1B7PzIhaxL7Y")

# 3d line
decode_sf("B1Boz5xJ67i1BU1B7PUzIhaUxL7YU")
```

**encode**

*Encode a line in the flexible polyline encoding format*

**Description**

This function calls hf::polyline_encode of the C++ implementation of the flexible polyline encoding by HERE. Depending on the dimensions of the input coordinates, a two or three dimensional line is encoded.
Usage

encode(line, precision = 5L, third_dim = 3L, third_dim_precision = 5L)

Arguments

  line          matrix, coordinates of the line in 2d or 3d (column order: LNG, LAT, DIM3).
  precision     integer, precision to use in encoding (between 0 and 15, default=5).
  third_dim     integer, type of the third dimension (0: ABSENT, 1: LEVEL, 2: ALTITUDE, 3: ELEVATION, 4, 6: CUSTOM1, 7: CUSTOM2, default=3).
  third_dim_precision
                    integer, precision to use in encoding for the third dimension (between 1 and 15, default=5).

Value

The line as string in the flexible polyline encoding format.

Examples

# 2D
line2d <- matrix(
  c(8.69821, 50.10228,
    8.69567, 50.10201,
    8.69150, 50.10063,
    8.68752, 50.09878),
  ncol = 2, byrow = TRUE
)
encode(line2d)

# 3D
line3d <- matrix(
  c(8.69821, 50.10228, 10,
    8.69567, 50.10201, 20,
    8.69150, 50.10063, 30,
    8.68752, 50.09878, 40),
  ncol = 3, byrow = TRUE
)
encode(line3d)

---

**encode_sf**  
Wrapper function for encoding simple features

Description

A wrapper function for `encode` that converts simple feature geometries of the sf package to flexible polyline encoded strings.
**encode_sf**

Usage

```r
encode_sf(
  line,
  precision = 5,
  third_dim = NULL,
  third_dim_precision = precision
)
```

Arguments

- **line**
  - simple feature, sf, sfc or sfg object with geometry type "LINESTRING".
- **precision**
  - integer, precision to use in encoding (between 0 and 15, default=5).
- **third_dim**
  - integer, type of the third dimension (0: ABSENT, 1: LEVEL, 2: ALTITUDE, 3: ELEVATION, 4, 6: CUSTOM1, 7: CUSTOM2, default=NULL).
- **third_dim_precision**
  - integer, precision to use in encoding for the third dimension (between 1 and 15, default=precision).

Value

The line as string in the flexible polyline encoding format.

Examples

```r
# 2D
line2d <- sf::st_linestring(
  matrix(
    c(8.69821, 50.10228,
      8.69567, 50.10201,
      8.69150, 50.10063,
      8.68752, 50.09878),
    ncol = 2, byrow = TRUE
  )
)
encode_sf(line2d)

# 3D
line3d <- sf::st_linestring(
  matrix(
    c(8.69821, 50.10228, 10,
      8.69567, 50.10201, 20,
      8.69150, 50.10063, 30,
      8.68752, 50.09878, 40),
    ncol = 3, byrow = TRUE
  )
)
encode_sf(line3d)
```
Index

decode, 2, 3
decode_sf, 3

encode, 3, 4
encode_sf, 4