Package 'plusCode2'

April 2, 2024

Type Package

Title Coordinates to 'Plus Code' Conversion Tool

Version 0.1.0

Date 2024-04-02

Description Generates 'Plus Code' of geometric objects or data frames that contain them, giving the possibility to specify the precision of the area. The main feature of the package comes from the open-source code developed by 'Google Inc.' present in the repository <https://github.com/google/open-location-code/blob/main/java/src/main/java/com/google/openlocationcode/OpenLocationCode.java>. For details about 'Plus Code', visit <https://maps.google.com/pluscodes/> or <https://github.com/google/open-location-code>.

License MIT + file LICENSE

URL https://github.com/Armando-d/plusCode2

Encoding UTF-8

Depends R (>= 4.3)

Imports sf

RoxygenNote 7.3.1

NeedsCompilation no

Author Armando d'Aniello [aut, cre, cph],
Google Inc. [cph] (© 2014 Google Inc. All rights reserved.)

Maintainer Armando d'Aniello <armando.daniello@istat.it>

Repository CRAN

Date/Publication 2024-04-02 13:12:03 UTC

R topics documented:

---

checkAndConvert   *Check and convert input object to sfc_POINT object*

Description

Checks on the input object and any conversions:

- Check that the input object is of length 1, if it is not an error appears.

- Verifies that the input object is of the sfc (Simple Features Collection) class, if it is not, it returns NA and reports a warning.

- If the input object is of sfc class, checks that it is of type POINT, if it is not, the centroid of the geometric object is calculated.

- If the input object is of the sfc_POINT class, check that it has a reference system and that this is the WGS84 for the decimal coordinates; if it is not, it is set or converted.

Usage

```
checkAndConvert(x)
```

Arguments

x   A generic object.

Value

An object of class sfc_POINT.

Examples

```
library(sf)
checkAndConvert(0)
###
polygon <- st_sfc(st_polygon(list(cbind(c(0, 1, 1, 0, 0), c(0, 0, 1, 1, 0)))))
checkAndConvert(polygon)
###
point <- st_sfc(st_point(c(0, 0)))
checkAndConvert(point)
```

---

computeLatitudePrecision

Longitude normalization

Description

Calculates the latitude precision value for a given code length. Lengths <= 10 have the same precision for latitude and longitude, but lengths > 10 have different precisions due to the grid method having fewer columns than rows.

Usage

```
computeLatitudePrecision(codeLength)
```

Arguments

codeLength   A number.

Value

A number.

Examples

```
computeLatitudePrecision(10)
computeLatitudePrecision(12)
```

---

dealWithDf   *Deal with data frame*

Description

Adapt the genPluscode function to a class object (sf, data.frame), it returns a vector containing the pluscode of the coordinates in the records of the input data frame.

Usage

```
dealWithDf(x, codeLength)
```

Arguments

x   An object of class (sf, data.frame).

codeLength   A number.

Value

A character vector of length nrow(x).

Examples

```
library(sf)
coordinates <- data.frame(lon = -6.56718, lat = 52.50145)
geom_df <- st_as_sf(coordinates, coords = c("lon", "lat"), crs = 4326)
dealWithDf(geom_df, 12)
###
coordinates_df <- data.frame(lon = c(-6.56718, -6), lat = c(52.50145, 52))
geom_df <- st_as_sf(coordinates_df, coords = c("lon", "lat"), crs = 4326)
dealWithDf(geom_df, 12)
```

---

genPluscode   *Generate Plus Code*

Description

Main function of the package. It generates a plus code giving as input a sfc object or a sf data.frame and a code length. Code length is related to the precision as dimension of the area. The precision of a plus code is indicated by the number of digits after the "+" sign. For example:

10 (2 digits after "+") -> area of 13.7 by 13.7 meters.

11 (3 digits after "+") -> area of 2.7 by 3.5 meters.

12 (4 digits after "+") -> area of 0.5 by 0.8 meters.

Usage

```
genPluscode(geom, codeLength = 10)
```

Arguments

geom	An object of class sfc or (sf, data.frame).
codeLength	An optional number between 4 and 15 exluding 5, 7, 9; default is 10.

Value

Depending on input it returns a character or an object of class (sf, data.frame).

Examples

```
library(sf)
point <- st_sfc(st_point(c(0, 0)))
genPluscode(point,10)
###
coordinates_df <- data.frame(lon = c(-6.56718, -6), lat = c(52.50145, 52))
geom_df <- st_as_sf(coordinates_df, coords = c("lon", "lat"), crs = 4326)
genPluscode(geom_df)
```

---

normalizeLatitude *Latitude normalization*

Description

Limit latitude within coherence range (-90, 90)

Usage

```
normalizeLatitude(latitude)
```

Arguments

latitude   A number.

Value

A number.

Examples

```
normalizeLatitude(80)
normalizeLatitude(100)
```

---

normalizeLongitude *Longitude normalization*

Description

Limit longitude within coherence range (-180, 180)

Usage

```
normalizeLongitude(longitude)
```

Arguments

longitude   A number.

Value

A number.

Examples

```
normalizeLongitude(100)
normalizeLongitude(181)
```

Index