Package ‘tidygeocoder’
March 22, 2020

Type Package
Title Tidyverse-Style Interface for Geocoding
Version 0.2.5
Author Jesse Cambon
Maintainer Jesse Cambon <jesse.cambon@gmail.com>
Description An intuitive tidyverse-style interface for geocoding. Obtains latitude and longitude coordinates in tibble format from addresses. The currently supported services are the US Census geocoder and Nominatim (OSM).

URL https://github.com/jessecambon/tidygeocoder
BugReports https://github.com/jessecambon/tidygeocoder/issues
Depends R (>= 2.10)
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Imports rlang, tibble, purrr, tidyr, tmaptools, httr, jsonlite, stringr
RoxygenNote 7.1.0
Suggests knitr, rmarkdown, ggplot2, ggrepel, maps, testthat (>= 2.1.0)
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2020-03-22 17:40:02 UTC

R topics documented:

geocode ................................................................. 2
geo_cascade .......................................................... 3
geo_census ........................................................... 3
geo_osm ............................................................... 4
sample_addresses ...................................................... 5
geocode

Geocode street addresses in a dataframe

Description

Takes a dataframe containing addresses as an input. Returns the dataframe with latitude and longitude coordinate columns using a user specified geocoder function.

Usage

geocode(.tbl, address, method = "census", lat = lat, long = long, ...)

Arguments

- `.tbl`: dataframe
- `address`: name of column containing addresses in `.tbl`
- `method`: the geocoder function you want to use
  - "census": `geo_census` - can only handle US street level addresses
  - "osm": `geo_osm` - more versatile than Census but has a usage limit
  - "cascade": `geo_cascade` - first tries to use census then tries osm
- `lat`: name of latitude field
- `long`: name of longitude field
- `...`: arguments supplied to the relevant geocoder function

Details

See example usage in vignette("tidygeocoder")

Value

input dataframe (.tbl) with latitude and longitude fields appended

Examples

```r
library(dplyr)
sample_addresses %>% geocode(addr)
sample_addresses %>% geocode(addr, method='cascade', lat=latitude, long=longitude)
```
**geo_cascade**

*Geocode addresses*

**Description**

First attempts to use the US Census Geocoder (geo_census) method and then uses the Nomina-tim/OSM (geo_osm) method if the census method failed. Returns latitude and longitude coordinates and the method used to return results (OSM or Census)

**Usage**

```r
geo_cascade(address, lat = lat, long = long, verbose = FALSE)
```

**Arguments**

- **address**: single line address.
- **lat**: name of latitude field
- **long**: name of longitude field
- **verbose**: logical. If TRUE outputs logs.

**Value**

latitude and longitude coordinates and the geocoder method used (geo_method) in tibble format (3 columns)

**Examples**

```r
geo_cascade("1600 Pennsylvania Ave Washington, DC")
geo_cascade("Paris, France")
```

---

**geo_census**

*Geocode street addresses*

**Description**

Obtain latitude and longitude coordinates from an address using the US Census geocoder. Only works for addresses within the US. Addresses must also be at the street level (ie. 60 Main St. Pawnee, IN not Pawnee, IN).
Usage

```r
geo_census(
  address,
  lat = lat,
  long = long,
  verbose = FALSE,
  benchmark = 4,
  API_URL = "https://geocoding.geo.census.gov/geocoder/locations/onelineaddress?"
)
```

Arguments

- `address`: single line address. Street must be included.
- `lat`: name of latitude field
- `long`: name of longitude field
- `verbose`: logical. If TRUE outputs logs.
- `benchmark`: parameter for the US Census Geocoder
- `API_URL`: URL of Census API

Value

latitude and longitude coordinates in tibble format

Examples

```r
geo_census("1600 Pennsylvania Ave Washington, DC")
```

---

### `geo_osm`

**Geocode addresses**

**Description**

Obtains latitude and longitude coordinates from an address using the Nominatim (OSM) geocoder service. Can be used with non-US or non-street level addresses unlike the Census geocoder. This function calls the geocode_OSM function from the tmaptools package.

**Usage**

```r
geo_osm(address, lat = lat, long = long, verbose = FALSE)
```
Arguments

<table>
<thead>
<tr>
<th>name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>single line address</td>
</tr>
<tr>
<td>lat</td>
<td>name of latitude field</td>
</tr>
<tr>
<td>long</td>
<td>name of longitude field</td>
</tr>
<tr>
<td>verbose</td>
<td>logical. If TRUE outputs logs.</td>
</tr>
</tbody>
</table>

Details

WARNING - This service has a usage limit and it will return missing coordinates once the usage limit is reached.

Value

latitude and longitude coordinates in tibble format

Examples

```r
geo_osm("1600 Pennsylvania Ave Washington, DC")
geo_osm("Paris, France", verbose=TRUE)
```

---

### sample_addresses

**Some sample addresses for testing**

<table>
<thead>
<tr>
<th>sample_addresses</th>
<th>Some sample addresses for testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Description of the address</td>
</tr>
<tr>
<td>addr</td>
<td>Single line address</td>
</tr>
</tbody>
</table>

Source

NA
Index

* Topic datasets
  sample_addresses, 5
  geo_cascade, 2, 3
  geo_census, 2, 3
  geo_osm, 2, 4
  geocode, 2
  sample_addresses, 5