

# 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/jesse cambon/tidygeocoder>

**BugReports** <https://github.com/jesse cambon/tidygeocoder/issues>

**Depends** R (>= 2.10)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** rlang, tibble, purrr, tidyr, dplyr, 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	<i>Geocode street addresses in a dataframe</i>
---------	--

---

### Description

Takes a dataframe containing addresses as a 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 <ul style="list-style-type: none"><li>• "census": <a href="#">geo_census</a> - can only handle US street level addresses</li><li>• "osm": <a href="#">geo_osm</a> - more versatile than Census but has a usage limit</li><li>• "cascade": <a href="#">geo_cascade</a> - first tries to use census then tries osm</li></ul>
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

```
library(dplyr)
sample_addresses %>% geocode(addr)

sample_addresses %>% geocode(addr, method='cascade', lat=latitude, long=longitude)
```

---

geo_cascade	<i>Geocode addresses</i>
-------------	--------------------------

---

**Description**

First attempts to use the US Census Geocoder (geo\_census) method and then uses the Nominatim/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**

```
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**

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

---

geo_census	<i>Geocode street addresses</i>
------------	---------------------------------

---

**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**

```

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

```

**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**

```

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 tmtools package.

**Usage**

```

geo_osm(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.

**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**

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

---

sample_addresses	<i>Some sample addresses for testing</i>
------------------	--

---

**Description**

Some sample addresses for testing

**Usage**

```
sample_addresses
```

**Format**

A tibble dataframe with single line addresses

**name** Description of the address

**addr** Single line address

**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