

Package ‘lingtypology’

May 13, 2017

Type Package

Title Linguistic Typology and Mapping

Version 1.0.4

Depends R (>= 2.10)

Imports leaflet,

stats,
stringdist,
magrittr,
grDevices,
rowr,
MASS,
sp

Author George Moroz

Maintainer George Moroz <agricolamz@gmail.com>

Description Provides R with the Glottolog database <<http://glottolog.org>> and some more abilities for purposes of linguistic mapping. The Glottolog database contains the catalogue of languages of the world. This package helps researchers to make a linguistic maps, using philosophy of the Cross-Linguistic Linked Data project <<http://clld.org/>>, which allows for while at the same time facilitating uniform access to the data across publications. A tutorial for this package is available on GitHub pages <<https://ropensci.github.io/lingtypology/>> and package vignette. Maps created by this package can be used both for the investigation and linguistic teaching.

License GPL (>= 2)

URL <https://CRAN.R-project.org/package=lingtypology>, <https://github.com/ropensci/lingtypology/>

BugReports <https://github.com/ropensci/lingtypology/issues>

LazyData TRUE

RoxygenNote 6.0.1

Suggests knitr,
rmarkdown,
testthat,
covr

VignetteBuilder knitr

R topics documented:

aff.lang	2
area.lang	3
circassian	4
countries	4
country.lang	5
ejective_and_n_consonants	5
glottolog.modified	6
glottolog.original	7
gltc.iso	8
gltc.lang	9
is.glottolog	9
iso.gltc	10
iso.lang	11
lang.aff	12
lang.country	12
lang.gltc	13
lang.iso	14
lat.lang	15
long.lang	15
map.feature	16
polygon.points	20
url.lang	20
Index	22

aff.lang	<i>Get affiliation by language</i>
----------	------------------------------------

Description

Takes any vector of languages and return affiliation.

Usage

```
aff.lang(x, glottolog.source = "modified")
```

Arguments

- x A character vector of the languages (can be written in lower case)
- glottolog.source A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[area.lang](#), [country.lang](#), [iso.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
aff.lang('Korean')
aff.lang(c('Korean', 'Polish'))
```

area.lang	<i>Get macro area by language</i>
-----------	-----------------------------------

Description

Takes any vector of languages and return macro area.

Usage

```
area.lang(x, glottolog.source = "modified")
```

Arguments

x	character vector of the languages (can be written in lower case)
glottolog.source	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [country.lang](#), [iso.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
area.lang('Adyghe')
area.lang(c('Adyghe', 'Aduge'))
```

circassian	<i>Circassian villages in Russia</i>
------------	--------------------------------------

Description

A dataset contains the list of the Circassian villages in Russia with genealogical affiliation, coordinates and district names. Most data collected during the fieldworks (2011–2016).

Usage

circassian

Format

A data frame with 157 rows and 6 variables:

longitude longitude

latitude latitude

village name of the village

district names of the subjects of the Russian Federation: kbr — Kabardino–Balkar Republic, kch — Karachay–Cherkess Republic, kk — Krasnodar Krai, ra — Republic of Adygea, stv — Stavropol Krai

dialect names of the Circassian dialects

language according standard Circassian devision there are Adyghe and Kabardian languages

countries	<i>Catalogue of countries names.</i>
-----------	--------------------------------------

Description

Catalogue of countries names.

Usage

countries

Format

A data frame with 86 rows and 3 variables:

common common name

official official name

abbreviation abbreviated name

official_languages official languages from the given country

country.lang	<i>Get country by language</i>
--------------	--------------------------------

Description

Takes any vector of languages and return affiliation.

Usage

```
country.lang(x, intersection = FALSE, glottolog.source = "modified")
```

Arguments

x	character vector of the languages (can be written in lower case)
intersection	logical. If TRUE, function returns vector of countries, where all languages from x argument are spoken.
glottolog.source	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [iso.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
country.lang('Udi')
country.lang(c('Udi', 'Laz'))
country.lang(c('Udi', 'Laz'), intersection = TRUE)
```

ejective_and_n_consonants

Number of consonants and presence of ejectives

Description

Number of consonants and presence of ejectives

Usage

```
ejective_and_n_consonants
```

Format

A data frame with 27 rows and 3 variables:

language language name

n.cons.lapsyd number of consonants. Based on **LAPSyD** database.

ejectives presence of ejective sounds

glottolog.modified	<i>Catalogue of languages of the world</i>
--------------------	--

Description

A dataset contains the modified catalogue of languages of the world involving genealogical affiliation, macro-area, country, iso code, and coordinates.

Usage

glottolog.modified

Format

A data frame with 8566 rows and 7 variables:

iso code based on ISO 639-3 <http://www-01.sil.org/iso639-3/>

language name of the language

affiliation genealogical affiliation

area have six values Africa, Australia, Eurasia, North America, Papunesia, South America

country list of countries, where the language is spoken

latitude latitude

longitude longitude

glottocode languoid code from Glottolog 2.7

alternate_names alternative language names

affiliation-HH some additional source for affiliation

dialects dialects of language

language_development language development

language_status language status. In glottolog.modified comments are removed. In glottolog.original they are reserved. Have 14 categories: 1 (Nationl); 2 (Provincial); 3 (Wider communication); 4 (Educational); 5 (Developing); 6a (Vigorous); 6b (Threatened); 7 (Shifting); 8a (Moribund); 8b (Nearly extinct); 8b (Reintroduced); 9 (Dormant); 9 (Second language only); 10 (Extinct)

language_use language use

location location

other_comments other_comments

population population and its source
population_numeric pure population info
timespan some historical information
typology some information form WALS
writing information about writing system

Details

Glottolog 2.7. Hammarstrom, Harald & Forkel, Robert & Haspelmath, Martin & Bank, Sebastian. 2016. Max Planck Institute for the Science of Human History. Accessed on 2016-06-15.

glottolog.original	<i>Catalogue of languages of the world</i>
--------------------	--

Description

A dataset contains the original catalogue of languages of the world involving genealogical affiliation, macro-area, country, iso code, and coordinates.

Usage

glottolog.original

Format

A data frame with 8566 rows and 7 variables:

iso code based on ISO 639-3 <http://www-01.sil.org/iso639-3/>

language name of the language

affiliation genealogical affiliation

area have six values Africa, Australia, Eurasia, North America, Papunesia, South America

country list of countries, where the language is spoken

latitude latitude

longitude longitude

glottocode languoid code from Glottolog 2.7

alternate_names alternative language names

affiliation-HH some additional source for affiliation

dialects dialects of language

language_development language development

language_status language status. In glottolog.modified comments are removed. In glottolog.original they are reserved. Have 14 categories: 1 (Natioanl); 2 (Provincial); 3 (Wider communication); 4 (Educational); 5 (Developing); 6a (Vigorous); 6b (Threatened); 7 (Shifting); 8a (Moribund); 8b (Nearly extinct); 8b (Reintroduced); 9 (Dormant); 9 (Second language only); 10 (Extinct)

language_use language use
location location
other_comments other_comments
population population and its source
population_numeric pure population info
timespan some historical information
typology some information form WALS
writing information about writing system

Details

Glottolog 2.7. Hammarstrom, Harald & Forkel, Robert & Haspelmath, Martin & Bank, Sebastian. 2016. Max Planck Institute for the Science of Human History. Accessed on 2016-06-15.

Source

<http://glottolog.org/>

gltc.iso

Get Glottocode by ISO 639–3 code

Description

Takes any vector of ISO 639–3 codes and returns Glottocodes.

Usage

```
gltc.iso(x, glottolog.source = "modified")
```

Arguments

x A character vector of the Glottocodes.
glottolog.source A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
gltc.iso('ady')
gltc.iso(c('ady', 'rus'))
```

`gltc.lang`*Get Glottocode by language*

Description

Takes any vector of languages and returns Glottocode.

Usage

```
gltc.lang(x, glottolog.source = "modified")
```

Arguments

`x` A character vector of the languages (can be written in lower case)

`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
gltc.lang('Adyghe')
gltc.lang(c('Adyghe', 'Udi'))
```

`is.glottolog`*Are these languages in glottolog?*

Description

Takes any vector of languages or ISO codes and return a logical vector.

Usage

```
is.glottolog(x, response = FALSE, glottolog.source = "modified")
```

Arguments

`x` A character vector of languages (can be written in lower case) or ISO codes

`response` logical. If TRUE, when language is absent, return warnings with a possible candidates.

`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

Examples

```
is.glottolog(c('Adyghe', 'Russian'))
is.glottolog('Buyaka')

# Add warning message with suggestions
is.glottolog(c('Adygey', 'Russian'), response = TRUE)
# > FALSE TRUE
# Warning message:
# In is.glottolog(c('Adyge', 'Russian'), response = TRUE) :
# Language Adyge is absent in our version of the Glottolog database. Did you mean Aduge, Adyghe?
```

iso.gltc

Get ISO 639-3 code by Glottocode

Description

Takes any vector of Glotocodes and returns ISO code.

Usage

```
iso.gltc(x, glottolog.source = "modified")
```

Arguments

`x` A character vector of Glottocodes.

`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
iso.gltc('adyg1241')
iso.gltc(c('adyg1241', 'udii1243'))
```

iso.lang

Get ISO 639–3 code by language

Description

Takes any vector of languages and returns ISO code.

Usage

```
iso.lang(x, glottolog.source = "modified")
```

Arguments

x A character vector of the languages (can be written in lower case)

glottolog.source A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [lat.lang](#), [long.lang](#)

Examples

```
iso.lang('Adyghe')
iso.lang(c('Adyghe', 'Udi'))
```

lang.aff	<i>Get languages by affiliation</i>
----------	-------------------------------------

Description

Takes any vector of affiliations and return languages.

Usage

```
lang.aff(x, list = FALSE, glottolog.source = "modified")
```

Arguments

x	A character vector of the affiliations (can be written in lower case)
list	logical. If TRUE, returns a list of languages, if FALSE return a named vector.
glottolog.source	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[lang.country](#), [lang.iso](#)

Examples

```
lang.aff('Slavic')
lang.aff(c('Slavic', 'Celtic'))
lang.aff(c('Slavic', 'Celtic'), list = TRUE)
```

lang.country	<i>Get languages by country</i>
--------------	---------------------------------

Description

Takes any vector of countries and return languages.

Usage

```
lang.country(x, list = FALSE, glottolog.source = "modified")
```

Arguments

`x` character vector of the countries (can be written in lower case)
`list` logical. If TRUE, returns a list of languages, if FALSE return a vector.
`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[lang.aff](#), [lang.iso](#)

Examples

```
lang.country('North Korea')  
lang.country(c('North Korea', 'Luxembourg'))  
lang.country(c('North Korea', 'Luxembourg'), list = TRUE)
```

lang.gltc	<i>Get language by Glottocode</i>
-----------	-----------------------------------

Description

Takes any vector of Glottocodes and return languages.

Usage

```
lang.gltc(x, glottolog.source = "modified")
```

Arguments

`x` A character vector of the Glottocodes.
`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[lang.aff](#), [lang.country](#)

Examples

```
lang.gltc('adyg1241')  
lang.gltc(c('adyg1241', 'udii1243'))
```

`lang.iso`*Get language by ISO 639–3 code*

Description

Takes any vector of ISO codes and return languages.

Usage

```
lang.iso(x, glottolog.source = "modified")
```

Arguments

<code>x</code>	A character vector of the ISO codes.
<code>glottolog.source</code>	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[lang.aff](#), [lang.country](#)

Examples

```
lang.iso('ady')  
lang.iso(c('ady', 'rus'))
```

lat.lang	<i>Get latitude by language</i>
----------	---------------------------------

Description

Takes any vector of languages and return latitude.

Usage

```
lat.lang(x, glottolog.source = "modified")
```

Arguments

x	A character vector of the languages (can be written in lower case)
glottolog.source	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [iso.lang](#), [long.lang](#)

Examples

```
lat.lang('Adyghe')
long.lang('Adyghe')
lat.lang(c('Adyghe', 'Russian'))
long.lang(c('Adyghe', 'Russian'))
```

long.lang	<i>Get longitude by language</i>
-----------	----------------------------------

Description

Takes any vector of languages and return longitude.

Usage

```
long.lang(x, map.orientation = "Pacific", glottolog.source = "modified")
```

Arguments

`x` A character vector of the languages (can be written in lower case)

`map.orientation` A character vector with values "Pacific" and "Atlantic". It distinguishes Pacific-centered and Atlantic-centered maps. By default is "Pacific".

`glottolog.source` A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

See Also

[aff.lang](#), [area.lang](#), [country.lang](#), [iso.lang](#), [lat.lang](#)

Examples

```
lat.lang('Adyghe')
long.lang('Adyghe')
lat.lang(c('Adyghe', 'Russian'))
long.lang(c('Adyghe', 'Russian'))
long.lang(c('Adyghe', 'Aleut'), map.orientation = "Pacific")
```

map.feature

Create a map

Description

Map a set of languages and color them by feature or two sets of features.

Usage

```
map.feature(languages, features = "", popup = "", label = "",
  label.hide = TRUE, label.fsize = 14, label.position = "right",
  stroke.features = NULL, latitude = NULL, longitude = NULL,
  density. estimation = FALSE, density. estimation.opacity = 0.2,
  density.longitude.width = NULL, density.latitude.width = NULL,
  color = NULL, stroke.color = NULL, image.url = NULL,
  image.width = 100, image.height = 100, image.X.shift = 0,
  image.Y.shift = 0, title = NULL, stroke.title = NULL, control = FALSE,
  legend = TRUE, legend.opacity = 1, legend.position = "topright",
  stroke.legend = TRUE, stroke.legend.opacity = 1,
  stroke.legend.position = "bottomleft", radius = 5, stroke.radius = 9.5,
  opacity = 1, stroke.opacity = 1, scale.bar = TRUE,
  scale.bar.position = "bottomleft", minimap = FALSE,
```



```

minimap.position = "bottomright", minimap.width = 150,
minimap.height = 150, tile = "OpenStreetMap.Mapnik", tile.name = NULL,
zoom.control = FALSE, map.orientation = "Pacific",
glottolog.source = "modified")

```

Arguments

languages	character vector of languages (can be written in lower case)
features	character vector of features
popup	character vector of strings that will appear in pop-up window
label	character vector of strings that will appear near points
label.hide	logical. If FALSE, labels are displayed allways. If TRUE, labels are displayed on mouse over. By default is TRUE.
label.fsize	numeric value of the label font size. By default is 14.
label.position	the position of labels: "left", "right", "top", "bottom"
stroke.features	additional independent stroke features
latitude	numeric vector of latitudes
longitude	numeric vector of longitudes
density.estimation	mostly logical. If TRUE, it creates a density estimation polygon. If write here "blank", it creates density estimation polygon without points.
density.estimation.opacity	a numeric vector of density estimation opacity.
density.longitude.width	bandwidths for longitude values. Defaults to normal reference bandwidth (see bandwidth.nrd).
density.latitude.width	bandwidths for latitude values. Defaults to normal reference bandwidth (see bandwidth.nrd).
color	vector of colors or palette. The color argument can be (1) a character vector of RGM or named colors; (2) the name of an RColorBrewer palette; (3) the full name of a viridis palette; (4) a function that receives a single value between 0 and 1 and returns a color. For more examples see colorNumeric
stroke.color	vector of stroke colors
image.url	character vector of URLs with an images
image.width	numeric vector of image widths
image.height	numeric vector of image heights
image.X.shift	numeric vector of image's X axis shift relative to the latitude-longitude point
image.Y.shift	numeric vector of image's Y axis shift relative to the latitude-longitude point
title	title of a legend
stroke.title	title of a stroke-feature legend

control	logical. If TRUE, function show layer control buttons. By default is TRUE.
legend	logical. If TRUE, function show legend. By default is FALSE.
legend.opacity	a numeric vector of legend opacity.
legend.position	the position of the legend: "topright", "bottomright", "bottomleft", "topleft"
stroke.legend	logical. If TRUE, function show stroke.legend. By default is FALSE.
stroke.legend.opacity	a numeric vector of stroke.legend opacity.
stroke.legend.position	the position of the stroke.legend: "topright", "bottomright", "bottomleft", "topleft"
radius	a numeric vector of radii for the circles.
stroke.radius	a numeric vector of stroke radii for the circles.
opacity	a numeric vector of marker opacity.
stroke.opacity	a numeric vector of stroke opacity.
scale.bar	logical. If TRUE, function shows scale-bar. By default is TRUE.
scale.bar.position	the position of the scale-bar: "topright", "bottomright", "bottomleft", "topleft"
minimap	logical. If TRUE, function shows mini map. By default is FALSE.
minimap.position	the position of the minimap: "topright", "bottomright", "bottomleft", "topleft"
minimap.width	The width of the minimap in pixels.
minimap.height	The height of the minimap in pixels.
tile	a character vector with a map tiles, popularized by Google Maps. See here for the complete set.
tile.name	a character vector with a user's map tiles' names
zoom.control	logical. If TRUE, function shows zoom controls. By default is FALSE.
map.orientation	a character vector with values "Pacific" and "Atlantic". It distinguishes Pacific-centered and Atlantic-centered maps. By default is "Pacific".
glottolog.source	A character vector that define which glottolog database is used: "original" or "modified" (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

Examples

```
map.feature(c("Adyghe", "Russian"))

## Map all Slavic languages
map.feature(lang.aff(c("Slavic")))
```

```

## Color languages by feature
df <- data.frame(lang = c("Adyghe", "Kabardian", "Polish", "Russian", "Bulgarian"),
  feature = c("polysynthetic", "polysynthetic", "fusion", "fusion", "fusion"))
map.feature(df$lang, df$feature)
## ... or add a control buttons for features
map.feature(df$lang, df$feature, control = TRUE)

## Adding pop-up
df <- data.frame(lang = c("Adyghe", "Kabardian", "Polish", "Russian", "Bulgarian"),
  feature = c("polysynthetic", "polysynthetic", "fusion", "fusion", "fusion"),
  popup = c("Circassian", "Circassian", "Slavic", "Slavic", "Slavic"))
map.feature(df$lang, df$feature, df$popup)

## Adding labels
df <- data.frame(lang = c("Adyghe", "Kabardian", "Polish", "Russian", "Bulgarian"),
  feature = c("polysynthetic", "polysynthetic", "fusion", "fusion", "fusion"),
  popup = c("Circassian", "Circassian", "Slavic", "Slavic", "Slavic"))
map.feature(df$lang, df$feature, label = df$lang)

## Add your own coordinates
map.feature("Adyghe", latitude = 43, longitude = 57)

## Change map tile
map.feature("Adyghe", tile = "Thunderforest.OpenCycleMap")

## Add you own colors
df <- data.frame(lang = c("Adyghe", "Kabardian", "Polish", "Russian", "Bulgarian"),
  feature = c("polysynthetic", "polysynthetic", "fusion", "fusion", "fusion"),
  popup = c("Circassian", "Circassian", "Slavic", "Slavic", "Slavic"))
map.feature(df$lang, df$feature, df$popup, color = c("green", "navy"))

## Map two sets of features
df <- data.frame(lang = c("Adyghe", "Kabardian", "Polish", "Russian", "Bulgarian"),
  feature = c("polysynthetic", "polysynthetic", "fusion", "fusion", "fusion"),
  popup = c("Circassian", "Circassian", "Slavic", "Slavic", "Slavic"))
map.feature(df$lang, df$feature, df$popup,
  stroke.features = df$popup)

## Add a pictures to plot
df <- data.frame(lang = c("Russian", "Russian"),
  lat = c(55.75, 59.95),
  long = c(37.616667, 30.3),
  urls = c("https://goo.gl/5OUv1E", "https://goo.gl/UWmvDw"))
map.feature(languages = df$lang,
  latitude = df$lat,
  longitude = df$long,
  image.url = df$urls)

## Add a minimap to plot
map.feature(c("Adyghe", "Russian"), minimap = TRUE)

## Remove scale bar
map.feature(c("Adyghe", "Russian"), scale.bar = FALSE)

```

polygon.points	<i>Get kernel density estimation poligon from coordinates</i>
----------------	---

Description

This function is based on this answer: <https://gis.stackexchange.com/a/203623>

Usage

```
polygon.points(latitude, longitude, latitude_width, longitude_width)
```

Arguments

latitude	numeric vector of latitudes
longitude	numeric vector of longitudes
latitude_width	bandwidths for latitude values. Defaults to normal reference bandwidth (see bandwidth.nrd).
longitude_width	bandwidths for longitude values. Defaults to normal reference bandwidth (see bandwidth.nrd).

url.lang	<i>Make a url-link to glottolog page for a language</i>
----------	---

Description

Takes any vector of languages and return links to glottolog pages.

Usage

```
url.lang(x, popup = "", glottolog.source = "modified")
```

Arguments

x	A character vector of languages (can be written in lower case)
popup	character vector of strings that will appear in pop-up window of the function map.feature
glottolog.source	A character vector that define which glottolog database is used: 'original' or 'modified' (by default)

Author(s)

George Moroz <agricolamz@gmail.com>

Examples

```
url.lang('Korean')  
url.lang(c('Gangou', 'Hachijo', 'Adyghe', 'Ganai'))
```

Index

*Topic **datasets**

- circassian, [4](#)
- countries, [4](#)
- ejective_and_n_consonants, [5](#)
- glottolog.modified, [6](#)
- glottolog.original, [7](#)

aff.lang, [2](#), [3](#), [5](#), [8](#), [9](#), [11](#), [15](#), [16](#)
area.lang, [3](#), [3](#), [5](#), [8](#), [9](#), [11](#), [15](#), [16](#)

bandwidth.nrd, [17](#), [20](#)

circassian, [4](#)
colorNumeric, [17](#)
countries, [4](#)
country.lang, [3](#), [5](#), [8](#), [9](#), [11](#), [15](#), [16](#)

ejective_and_n_consonants, [5](#)

glottolog.modified, [6](#)
glottolog.original, [7](#)
gltc.iso, [8](#)
gltc.lang, [9](#)

is.glottolog, [9](#)
iso.gltc, [10](#)
iso.lang, [3](#), [5](#), [11](#), [15](#), [16](#)

lang.aff, [12](#), [13](#), [14](#)
lang.country, [12](#), [12](#), [13](#), [14](#)
lang.gltc, [13](#)
lang.iso, [12](#), [13](#), [14](#)
lat.lang, [3](#), [5](#), [8](#), [9](#), [11](#), [15](#), [16](#)
long.lang, [3](#), [5](#), [8](#), [9](#), [11](#), [15](#), [15](#)

map.feature, [16](#)

polygon.points, [20](#)

url.lang, [20](#)