Package 'lutz'

June 24, 2018

Type Package

Title Look Up Time Zones of Point Coordinates

Version 0.2.0

Description Input latitude and longitude values or an 'sf/sfc' POINT object and get back the timezone in which they exist. Two methods are implemented. One is very fast and uses the 'V8' package to access the 'tz-lookup.js' 'Javascript' library (<https://github.com/darkskyapp/tz-lookup/>). This method also works outside of countries' borders and in international waters, however speed comes at the cost of accuracy - near time zone borders away from populated centres there is a chance that it will return the incorrect time zone. The other method is slower but more accurate - it uses the sf package to intersect points with a detailed map of time zones from here: <https://github.com/evansiroky/timezone-boundary-builder/>.

License MIT + file LICENSE

URL https://github.com/ateucher/lutz

BugReports https://github.com/ateucher/lutz/issues

Depends R (>= 2.10)

Imports V8 (>= 1.5), stats

Suggests testthat, sf (>= 0.5), sp, rgdal, datasets, covr

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Andy Teucher [aut, cre]

Maintainer Andy Teucher <andy.teucher@gmail.com>

Repository CRAN

Date/Publication 2018-06-24 17:04:40 UTC

R topics documented:

`tz_lookup` *Lookup time zones of sf or sp points*

Description

There are two methods - `"fast"`, and `"accurate"`. The `"fast"` version can look up many thousands of points very quickly, however when a point is near a time zone boundary and not near a populated centre, it may return the incorrect timezone. If accuracy is more important than speed, use `method = "accurate"`.

Usage

```
tz_lookup(x, crs = NULL, method = "fast", warn = TRUE)
```

Arguments

`x`	either an `sfc` or `sf` points or `SpatialPoints(DataFrame)` object
`crs`	the coordinate reference system: integer with the EPSG code, or character with proj4string. If not specified (i.e., `NULL`) and `x` has no existing `crs`, EPSG: 4326 is assumed (lat/long).
`method`	method by which to do the lookup. Either `"fast"` (default) or `"accurate"`.
`warn`	By default, if `method = "fast"` a warning is issued about the potential for inaccurate results. Set `warn` to `FALSE` to turn this off.

Value

character vector the same length as `x` specifying the time zone of the points.

Examples

```
if (require("sf")) {

state_pts <- lapply(seq_along(state.center$x), function(i) {
  st_point(c(state.center$x[i], state.center$y[i]))
})

state_centers_sf <- st_sf(st_sfc(state_pts))

state_centers_sf$tz <- tz_lookup(state_centers_sf)

plot(state_centers_sf[, "tz"])
}
```

`tz_lookup_coords` *Lookup time zones of lat/long pairs*

Description

There are two methods - `"fast"`, and `"accurate"`. The `"fast"` version can look up many thousands of points very quickly, however when a point is near a time zone boundary and not near a populated centre, it may return the incorrect timezone. If accuracy is more important than speed, use `method = "accurate"`.

Usage

```
tz_lookup_coords(lat, lon, method = "fast", warn = TRUE)
```

Arguments

`lat`	numeric vector of latitudes
`lon`	numeric vector of longitudes the same length as `x`
`method`	method by which to do the lookup. Either `"fast"` (default) or `"accurate"`.
`warn`	By default, if `method = "fast"` a warning is issued about the potential for inaccurate results. Set `warn` to `FALSE` to turn this off.

Value

character vector the same length as x and y specifying the time zone of the points.

Examples

```
tz_lookup_coords(42, -123)
tz_lookup_coords(lat = c(48.9, 38.5, 63.1, -25), lon = c(-123.5, -110.2, -95.0, 130))
```

Index