Package: smartmap (via r-universe)

September 2, 2024

Type Package

Title Smartly Create Maps from R Objects

Version 0.1.1.9002

Maintainer Stefan Fleck <stefan.b.fleck@gmail.com>

Description Preview spatial data as 'leaflet' maps with minimal
effort. smartmap is optimized for interactive use and
distinguishes itself from similar packages because it does not
need real spatial ('sp' or 'sf') objects an input; instead, it
tries to automatically coerce everything that looks like
spatial data to sf objects or leaflet maps. It - for example -
supports direct mapping of: a vector containing a single
coordinate pair, a two column matrix, a data.frame with
longitude and latitude columns, or the path or URL to a
(possibly compressed) 'shapefile'.

License MIT + file LICENSE

Imports leaflet, magrittr, sf, utils

Suggests knitr, covr, testthat, leafsync, rmarkdown

Depends R (>= 3.6.0)

URL https://github.com/s-fleck/smartmap

BugReports https://github.com/s-fleck/smartmap/issues

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1.9000

VignetteBuilder knitr

Repository https://s-fleck.r-universe.dev

RemoteUrl https://github.com/s-fleck/smartmap

RemoteRef HEAD

RemoteSha 5846c920bf2e00519b9494b26bce9bd321f3e7c6

Contents

---

as_coord_matrix   *Coerce an R object to a matrix of coordinates*

Description

A `coord_matrix` is a `matrix` with two columns named ″lon″ and ″lat″ to represent spatial point data. They are used as an intermediary when converting some R objects to `sf::sf()` objects.

`as_coord_matrix()` can smartly convert a range of R objects to `coord_matrix`. If you are a package developer and want to add support for smartmap to your package without having to depend on the heavy **sf** package, it is enough to provide an `as_coord_matrix()` method.

Usage

```
as_coord_matrix(x, ...)

## Default S3 method:
as_coord_matrix(x, ...)

## S3 method for class 'numeric'
as_coord_matrix(x, ...)

## S3 method for class 'sf'
as_coord_matrix(x, ...)

## S3 method for class 'sfc_POINT'
as_coord_matrix(x, ...)

## S3 method for class 'matrix'
as_coord_matrix(x, ..., loncol = guess_loncol(x), latcol = guess_latcol(x))

## S3 method for class 'data.frame'
as_coord_matrix(x, ..., loncol = guess_loncol(x), latcol = guess_latcol(x))
```

Arguments

x   any of the following:

  - a `matrix`: Either a matrix with named `longitude` and `latitude` columns
    or an unnamed two column matrix containing longitude and latitude (in that
    order)
  - a `data.frame` with named `longitude` and `latitude` columns
  - an [sf::sfc_POINT](#) object
  - a named or unnamed `numeric` vector of length 2 containing a single longitude-
    latitude coordinate pair
  - a `character` scalar path or URL to a shapefile or zipped shapefile

...   passed on to methods

loncol, latcol   character scalars. Names of the columns of x containing longitude and lati-
                 tude. The default trying guessing the columns.

Value

`as_coord_matrix()` returns a `coord_matrix` object: A numeric `matrix` with the columns `"lon"`and
`"lat"` (in that order)

See Also

<https://stackoverflow.com/questions/7309121/preferred-order-of-writing-latitude-longitude-tuples>

---

smap   *View spatial objects as interactive leaflet maps*

Description

Can be used to preview spatial R objects

Usage

```
smap(
  x,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)

## S3 method for class 'leaflet'
smap(
  x,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
```

```
)

## S3 method for class 'sf'
smap(
  x,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)

## Default S3 method:
smap(
  x,
  labels = NULL,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)

## S3 method for class 'sfc'
smap(
  x,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)

## S3 method for class 'sfg'
smap(
  x,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)

## S3 method for class 'matrix'
smap(
  x,
  labels = NULL,
  ...,
  tools = TRUE,
  provider = getOption("smap.providers", "OpenStreetMap")
)
```

Arguments

x   any input supported by [smart_as_sf()](#) or a [leaflet](#) map

- a matrix: Either a matrix with named longitude and latitude columns

or an unnamed two column matrix containing longitude and latitude (in that order)

- a `data.frame` with named `longitude` and `latitude` columns
- an [sf::sfc_POINT](#) object
- a named or unnamed `numeric` vector of length 2 containing a single longitude-latitude coordinate pair
- a `character` scalar path or URL to a shapefile or zipped shapefile
- a `leaflet` map

`...`	passed on to methods.
`tools`	`logical` scalar. If `TRUE` show additional tools on the resulting map (such as a ruler and the ability to switch between several background tiles)
`provider`	`character` vector. Name of one or several valid providers for [leaflet::addProviderTiles()](#). If `tools == TRUE` you will be able to switch interactively between all supplied providers on the returned leaflet map, if `tools == FALSE` only the first provider will be used.
`labels`	an optional `character` vector of popup labels

Value

a [leaflet::leaflet](#) object

Examples

```
wp <- matrix(
  c(16.419684, 48.186065,
    16.373894, 48.207853,
    16.285887, 48.083053),
  byrow = TRUE,
  ncol = 2
)


smap(wp)
smap(c(16.419684, 48.186065))
```

---

smart_as_sf	*Smartly convert an object to a simple features data frame*

Description

Converts R objects to [sf::sf](#) objects, but supports a wider range of input data than [sf::st_as_sf](#).

Usage

```
smart_as_sf(x, ...)

## Default S3 method:
smart_as_sf(x, ...)

## S3 method for class 'data.frame'
smart_as_sf(x, ...)

## S3 method for class 'character'
smart_as_sf(x, ...)
```

Arguments

x   any of the following:

- a `matrix`: Either a matrix with named `longitude` and `latitude` columns or an unnamed two column matrix containing longitude and latitude (in that order)
- a `data.frame` with named `longitude` and `latitude` columns
- an [sf::sfc_POINT](#) object
- a named or unnamed `numeric` vector of length 2 containing a single longitude-latitude coordinate pair
- a `character` scalar path or URL to a shapefile or zipped shapefile

...   ignored

Value

an [sf::sf](#) data.frame

Note

smart_as_sf.default() looks if an [`sf::st_as_sf()`](#), [`sf::st_as_sfc()`](#) or [`as_coord_matrix()`](#) method exists for x (in that order). If you are a package developer and want to support smartmap for a custom S3 class in your package, it is enough to provide one of these methods.

Examples

```
smart_as_sf(data.frame(lat = c(1,2,3), longitude = c(3,4,5)))
smart_as_sf(c(1, 2))
```

---

st_as_sf *Convert coordinate matrices to sf objects*

Description

Convert coordinate matrices to sf objects

Usage

```
## S3 method for class 'coord_matrix'
st_as_sf(x, ...)
```

Arguments

x   a coord_matrix

...   ignored

Value

an [sf::sf()]() object with an sfc_POINT-geometry column

See Also

[sf::st_as_sf()]()

---

st_as_sfc *Convert coordinate matrices to sfc objects*

Description

Convert coordinate matrices to sfc objects

Usage

```
## S3 method for class 'coord_matrix'
st_as_sfc(x, ...)
```

Arguments

x   a coord_matrix

...   ignored

Value

an [sf::sfc()]() object of subclass sfc_POINT

See Also

[sf::st_as_sfc()](sf::st_as_sfc())

---

unsf	*Convert sf objects to normal data.frames with longitude and latitude colu,ns*

Description

Convert sf objects to normal data.frames with longitude and latitude colu,ns

Usage

```
unsf(x, coord_names = c("lon", "lat"))
```

Arguments

x	an `sf::sf` object
coord_names	`character` vector of length 2. The names of the new that should contain the coordinates from the geometry column of x.

Value

a `data.frame`

Index