

# Package ‘smartmap’

July 2, 2020

**Type** Package

**Title** Smartly Create Maps from R Objects

**Version** 0.1.1

**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 as 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** covr, testthat

**Depends** R (>= 3.6.0)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1.9000

**NeedsCompilation** no

**Author** Stefan Fleck [aut, cre] (<<https://orcid.org/0000-0003-3344-9851>>)

**Repository** CRAN

**Date/Publication** 2020-07-02 11:30:09 UTC

## R topics documented:

as_coord_matrix . . . . .	2
smap . . . . .	3
smart_as_sf . . . . .	5
st_as_sf . . . . .	6
st_as_sf_c . . . . .	7

---

as_coord_matrix	<i>Coerce an R object to a matrix of coordinates</i>
-----------------	--

---

### 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 latitude. 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

<code>x</code>	any input supported by <code>smart_as_sf()</code> or a <code>leaflet</code> map <ul style="list-style-type: none"> <li>• a <code>matrix</code>: Either a matrix with named longitude and latitude columns or an unnamed two column matrix containing longitude and latitude (in that order)</li> <li>• a <code>data.frame</code> with named longitude and latitude columns</li> <li>• an <code>sf::sfc_POINT</code> object</li> <li>• a named or unnamed numeric vector of length 2 containing a single longitude-latitude coordinate pair</li> <li>• a character scalar path or URL to a shapefile or zipped shapefile</li> <li>• a <code>leaflet</code> map</li> </ul>
<code>...</code>	passed on to methods.
<code>tools</code>	logical scalar. If <code>TRUE</code> 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 <code>leaflet::addProviderTiles()</code> . If <code>tools == TRUE</code> you will be able to switch interactively between all supplied providers on the returned leaflet map, if <code>tools == FALSE</code> 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**

<code>x</code>	a <a href="#">coord_matrix</a>
<code>...</code>	ignored

**Value**

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

**See Also**[sf::st\\_as\\_sfc\(\)](#)

# Index

as\_coord\_matrix, 2  
as\_coord\_matrix(), 6

coord\_matrix, 6, 7  
coord\_matrix(as\_coord\_matrix), 2

leaflet, 4  
leaflet::addProviderTiles(), 5  
leaflet::leaflet, 5

sf::sf, 5, 6  
sf::sf(), 2, 6  
sf::sfc(), 7  
sf::sfc\_POINT, 2, 4, 6  
sf::st\_as\_sf, 5  
sf::st\_as\_sf(), 6, 7  
sf::st\_as\_sfc(), 6, 7  
smap, 3  
smart\_as\_sf, 5  
smart\_as\_sf(), 4  
st\_as\_sf, 6  
st\_as\_sfc, 7