

Package ‘sfheaders’

November 7, 2019

Type Package

Title Constructs Simple Feature Objects

Date 2019-11-07

Version 0.0.2

Description Converts R and 'Rcpp' objects to Simple Features 'sf', without depending on the Simple Feature library. Conversion functions are available at both the R level, and through 'Rcpp'.

License GPL-3

URL <https://dcooley.github.io/sfheaders/>

BugReports <https://github.com/dcooley/sfheaders/issues>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

SystemRequirements C++11

LinkingTo Rcpp

Imports Rcpp

Suggests testthat, covr

NeedsCompilation yes

Author David Cooley [aut, cre]

Maintainer David Cooley <david.cooley.au@gmail.com>

Repository CRAN

Date/Publication 2019-11-07 06:10:09 UTC

R topics documented:

<code>sfc_linestring</code>	2
<code>sfc_multilinestring</code>	3
<code>sfc_multipoint</code>	4
<code>sfc_multipolygon</code>	5

sfc_point	7
sfc_polygon	8
sfg_linestring	10
sfg_multilinestring	11
sfg_multipoint	12
sfg_multipolygon	13
sf_point	14
sf_polygon	15
sf_linestring	16
sf_multilinestring	17
sf_multipoint	18
sf_multipolygon	20
sf_point	22
sf_polygon	23

Index 25

sfc_linestring	<i>sfc LINESTRING</i>
----------------	-----------------------

Description

constructs sfc of MULTIPOINT objects

Usage

```
sfc_linestring(obj = NULL, x = NULL, y = NULL, z = NULL,
              m = NULL, linestring_id = NULL)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
linestring_id	column of ids for linestrings

Value

sfc object of LINESTRING geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- matrix( c(1:4), ncol = 2 )
sfc_linestring( x )

x <- data.frame( id = 1:2, x = 1:2, y = 2:1 )
sfc_linestring( x )
sfc_linestring( x, x = "x", y = "y" )
sfc_linestring( x, x = "y", y = "x" )
sfc_linestring( x, linestring_id = "id", x = "x", y = "y")
```

sfc_multilinestring *sfc MULTILINESTRING*

Description

constructs an sfc of MULTILINESTRING objects

Usage

```
sfc_multilinestring(obj = NULL, x = NULL, y = NULL, z = NULL,
  m = NULL, multilinestring_id = NULL, linestring_id = NULL)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multilinestring_id	column of ids for multilinestrings
linestring_id	column of ids for linestrings (within multilinestrings)

Value

sfc object of MULTILINESTRING geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```

m <- matrix(c(0,0,0,0,1,1), ncol = 3 )
sfc_multilinestring( m )

m <- matrix(c(0,0,0,0,0,1,0,1,1,1,2,2,1,2,3), ncol = 3, byrow = TRUE)
sfc_multilinestring( obj = m )
sfc_multilinestring( obj = m, multilinestring_id = 1 )
sfc_multilinestring( obj = m, linestring_id = 1 )

sfc_multilinestring( obj = m, linestring_id = 1, multilinestring_id = 1 )

sfc_multilinestring( obj = m, x = 2, y = 3 )
sfc_multilinestring( obj = m, x = 1, y = 2, z = 3 )
sfc_multilinestring( obj = m, x = 2, y = 3, linestring_id = 1, multilinestring_id = 1 )

df <- data.frame(
  ml_id = c(1,1,1,1,1,1,1,1,1,2,2,2,2,2)
  , l_id = c(1,1,1,2,2,3,3,3,1,1,1,2,2)
  , x = rnorm(13)
  , y = rnorm(13)
  , z = rnorm(13)
  , m = rnorm(13)
)

sfc_multilinestring( obj = df, x = "x", y = "y")
sfc_multilinestring( obj = df, x = "x", y = "y", z = "z")
sfc_multilinestring( obj = df, x = "x", y = "y", z = "z", m = "m")

sfc_multilinestring( obj = df, x = 2, y = 3)
sfc_multilinestring( obj = df, x = 2, y = 3, z = 4)
sfc_multilinestring( obj = df, x = 2, y = 3, z = 4, m = 5)

sfc_multilinestring( obj = df, multilinestring_id = "ml_id", linestring_id = "l_id" )
sfc_multilinestring( obj = df, multilinestring_id = 1, linestring_id = 2 )

```

sfc_multipoint

sfc MULTIPOINT

Description

constructs sfc of MULTIPOINT objects

Usage

```

sfc_multipoint(obj, x = NULL, y = NULL, z = NULL, m = NULL,
  multipoint_id = NULL)

```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multipoint_id	column of ids for multipoints

Value

sfc object of MULTIPOINT geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- matrix( c(1:4), ncol = 2 )
sfc_multipoint( x )

x <- data.frame( id = 1:2, x = 1:2, y = 2:1 )
sfc_multipoint( x )
sfc_multipoint( x, x = "x", y = "y" )
sfc_multipoint( x, x = "y", y = "x" )
sfc_multipoint( x, multipoint_id = "id", x = "x", y = "y")
```

sfc_multipolygon *sfc MULTIPOLYGON*

Description

constructs an sfc of MULTIPOLYGON objects

Usage

```
sfc_multipolygon(obj = NULL, x = NULL, y = NULL, z = NULL,
  m = NULL, multipolygon_id = NULL, polygon_id = NULL,
  linestring_id = NULL, close = TRUE)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multipolygon_id	column of ids for multipolygons
polygon_id	column of ids for polygons
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sfc object of MULTIPOLYGON geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
m <- matrix(c(0,0,0,0,1,0,0,1,1,0,0,1,0,0,0), ncol = 3, byrow = TRUE )
sfc_multipolygon( m )

df <- data.frame(
  id = c(1,1,1,1,1)
  , x = c(0,0,1,1,0)
  , y = c(0,1,1,0,0)
)

sfc_multipolygon( df, x = "x", y = "y" )

df <- data.frame(
  id = c(1,1,1,1,1,2,2,2,2)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sfc_multipolygon( df, multipolygon_id = "id", polygon_id = "id", linestring_id = "id")

df <- data.frame(
  id1 = c(1,1,1,1,1,1,1,1,1)
  , id2 = c(1,1,1,1,1,2,2,2,2)
)
```

```

    , x = c(0,0,1,1,0,1,1,2,2,1)
    , y = c(0,1,1,0,0,1,2,2,1,1)
  )

sfc_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2")

df <- data.frame(
  id1 = c(1,1,1,1,1,1,1,1,1,2,2,2,2,2)
  , id2 = c(1,1,1,1,1,2,2,2,2,2,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1,3,3,4,4,3)
  , y = c(0,1,1,0,0,1,2,2,1,1,3,4,4,3,3)
)

sfc_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2")

df <- data.frame(
  id1 = c(1,1,1,1,1,2,2,2,2,2)
  , id2 = c(1,1,1,1,1,1,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sfc_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2" )
sfc_multipolygon( df, polygon_id = "id1", linestring_id = "id2" )
sfc_multipolygon( df, x = "x", y = "y", polygon_id = "id1")
sfc_multipolygon( df, x = "x", y = "y", polygon_id = "id1", linestring_id =)
sfc_multipolygon( df, x = "x", y = "y", linestring_id = "id1")
sfc_multipolygon( df, x = "x", y = "y", linestring_id = "id2")

df <- data.frame(
  id1 = c('a','a','a','a','a','b','b','b','b','b')
  , id2 = c(1,1,1,1,1,1,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sfc_multipolygon( df, x = "x", y = "y", polygon_id = "id1")

```

sfc_point

sfc POINT

Description

constructs sfc of POINT objects

Usage

```
sfc_point(obj, x = NULL, y = NULL, z = NULL, m = NULL)
```

Arguments

obj	sorted vector, matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column

Value

sfc object of POINT geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- c(1:3)
sfc_point( x )

x <- matrix( c(1:10) , ncol = 2 )
sfc_point( x )

x <- setNames( as.data.frame( x ), c("x","y") )
sfc_point( x )
sfc_point( obj = x, x = "x", y = "y" )
sfc_point( obj = x, x = "y", y = "x" )
```

sfc_polygon

sfc POLYGON

Description

constructs an sfc of POLYGON objects

Usage

```
sfc_polygon(obj = NULL, x = NULL, y = NULL, z = NULL, m = NULL,
  polygon_id = NULL, linestring_id = NULL, close = TRUE)
```


Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
polygon_id	column of ids for polygons
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sfc object of POLYGON geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```

m <- matrix(c(0,0,0,0,1,1), ncol = 2 )
sfc_polygon( m )

m <- matrix(c(0,0,0,0,0,1,0,1,1,1,2,2,1,2,3,1,3,2), ncol = 3, byrow = TRUE)
sfc_polygon( obj = m )
sfc_polygon( obj = m, polygon_id = 1 )
sfc_polygon( obj = m, linestring_id = 1 )

sfc_polygon( obj = m, linestring_id = 1, polygon_id = 1 )

sfc_polygon( obj = m, x = 2, y = 3 )
sfc_polygon( obj = m, x = 1, y = 2, z = 3 )
sfc_polygon( obj = m, x = 2, y = 3, linestring_id = 1, polygon_id = 1 )

df <- data.frame(
  ml_id = c(1,1,1,1,1,1,1,1,2,2,2,2,2,2)
  , l_id = c(1,1,1,2,2,2,3,3,3,1,1,1,2,2,2)
  , x = rnorm(15)
  , y = rnorm(15)
  , z = rnorm(15)
  , m = rnorm(15)
)

sfc_polygon( obj = df, x = "x", y = "y")

```

```

sfc_polygon( obj = df, x = "x", y = "y", z = "z" )
sfc_polygon( obj = df, x = "x", y = "y", z = "z", m = "m" )

sfc_polygon( obj = df, x = 2, y = 3 )
sfc_polygon( obj = df, x = 2, y = 3, z = 4 )
sfc_polygon( obj = df, x = 2, y = 3, z = 4, m = 5 )

sfc_polygon( obj = df, polygon_id = "ml_id", linestring_id = "l_id" )
sfc_polygon( obj = df, polygon_id = 1, linestring_id = 2 )

```

sfg_linestring	<i>sfg linestring</i>
----------------	-----------------------

Description

constructs sfg LINESTRING object

Usage

```
sfg_linestring(obj, x = NULL, y = NULL, z = NULL, m = NULL)
```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column

Value

sfg object of LINESTRING geometry

Examples

```

sfg_linestring( matrix( 1:24, ncol = 2 ) )
sfg_linestring( matrix( 1:24, ncol = 3 ) )
sfg_linestring( matrix( 1:24, ncol = 4 ) )

sfg_linestring( matrix( 1:24, ncol = 4 ), x = 3, y = 2, z = 3 )

sfg_linestring( data.frame( x = 1:10, y = 11:20 ) )
sfg_linestring( data.frame( x = 1:10, y = 11:20, z = 21:30 ) )
sfg_linestring( data.frame( x = 1:10, y = 11:20, z = 21:30 ), x = "x", y = "z" )

```

sfg_multilinestring *sfg_multilinestring*

Description

constructs sfg MULTILINESTRING object

Usage

```
sfg_multilinestring(obj, x = NULL, y = NULL, z = NULL, m = NULL,  
  linestring_id = NULL)
```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
linestring_id	column of ids for lines

Value

sfg object of MULTILINESTRING geometry

Examples

```
sfg_multilinestring( matrix( 1:24, ncol = 2 ) )  
sfg_multilinestring( matrix( 1:24, ncol = 3 ) )  
sfg_multilinestring( matrix( 1:24, ncol = 4 ) )  
  
## different lines  
m <- cbind( matrix( 1:24, ncol = 2 ), c(rep(1, 6), rep(2, 6) ) )  
sfg_multilinestring( obj = m, x = 1, y = 2, linestring_id = 3 )  
  
## just specifying linestring_id will use all others as the geometries  
sfg_multilinestring( obj = m, linestring_id = 3 )  
  
df <- data.frame( x = 1:12, y = 1:12, z = 13:24, id = c(rep(1,6), rep(2,6)))  
sfg_multilinestring( df, x = "x", y = "y" )  
sfg_multilinestring( df, x = "x", y = "y", linestring_id = "id" )  
  
sfg_multilinestring( df, linestring_id = "id" )
```

sfg_multipoint *sfg multipoint*

Description

constructs sfg MULTIPOINT object

Usage

```
sfg_multipoint(obj, x = NULL, y = NULL, z = NULL, m = NULL)
```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column

Value

sfg object of MULTIPOINT geometry

Examples

```
sfg_multipoint( 1:2 )
sfg_multipoint( 1:3 )
sfg_multipoint( 1:4 )

sfg_multipoint( matrix( 1:3, ncol = 3 ) )
sfg_multipoint( data.frame( x = 1, y = 2, z = 3 ) )

sfg_multipoint( matrix( 1:4, ncol = 2 ) )
sfg_multipoint( matrix( 1:24, ncol = 2, byrow = TRUE ) )
sfg_multipoint( matrix( 1:24, ncol = 3, byrow = TRUE ) )
sfg_multipoint( matrix( 1:24, ncol = 4, byrow = TRUE ) )

sfg_multipoint( data.frame( x = 1:5, y = 1:5 ) )

## using columns

sfg_multipoint( matrix( 1:24, ncol = 4, byrow = TRUE ), x = 1, y = 2 )
sfg_multipoint( matrix( 1:24, ncol = 4, byrow = TRUE ), x = 1, y = 2, z = 3 )
sfg_multipoint( matrix( 1:24, ncol = 4, byrow = TRUE ), x = 3, y = 4 )

df <- data.frame( x = 1:5, y = 1:5, z = 11:15, m = 11:15 )
sfg_multipoint( df, x = "x", y = "y" )
```

```
sfg_multipoint( df, x = "x", y = "y", z = "z" )
sfg_multipoint( df, x = "x", y = "y", z = "z", m = "m" )
```

```
sfg_multipolygon      sfg multipolygon
```

Description

constructs sfg MULTIPOLYGON object

Usage

```
sfg_multipolygon(obj, x = NULL, y = NULL, z = NULL, m = NULL,
  polygon_id = NULL, linestring_id = NULL, close = TRUE)
```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
polygon_id	column of ids for polygons (within the multipolygon)
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sfg object of MULTIPOLYGON geometry

Examples

```
df <- data.frame(
  polygon_id = c(rep(1, 5), rep(2, 10))
  , line_id = c(rep(1, 10), rep(2, 5))
  , x = c(0,0,1,1,0,2,2,5,5,2,3,3,4,4,3)
  , y = c(0,1,1,0,0,2,5,5,2,2,3,4,4,3,3)
  , z = c(1)
  , m = c(1)
)

m <- as.matrix( df )

sfg_multipolygon( df[, c("x","y") ] )
```

```

sfg_multipolygon(
  df, x = "x", y = "y", polygon_id = "polygon_id", linestring_id = "line_id"
)
sfg_multipolygon(
  df, x = "x", y = "y", z = "z", polygon_id = "polygon_id", linestring_id = "line_id"
)
sfg_multipolygon(
  df, x = "x", y = "y", z = "z", m = "m", polygon_id = "polygon_id", linestring_id = "line_id"
)

sfg_multipolygon( m[, c("x","y") ] )

sfg_multipolygon(
  m, x = "x", y = "y", polygon_id = "polygon_id", linestring_id = "line_id"
)
sfg_multipolygon(
  m, x = "x", y = "y", z = "z", polygon_id = "polygon_id", linestring_id = "line_id"
)
sfg_multipolygon(
  m, x = "x", y = "y", z = "z", m = "m", polygon_id = "polygon_id", linestring_id = "line_id"
)

```

sfg_point

sfg point

Description

constructs sfg POINT object

Usage

```
sfg_point(obj, x = NULL, y = NULL, z = NULL, m = NULL)
```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column

Value

sfg object of POINT geometry

Examples

```

sfg_point( 1:2 )
sfg_point( 1:3 )
sfg_point( 1:4 )

sfg_point( matrix( 1:3, ncol = 3 ) )
sfg_point( data.frame( x = 1, y = 2, z = 3 ) )

sfg_point( data.frame( x = 1, y = 2, z = 3 ), x = "x", y = "y" )
sfg_point( data.frame( x = 1, y = 2, z = 3 ), x = 1, y = 3 )

```

sfg_polygon

sfg polygon

Description

constructs sfg POLYGON object

Usage

```

sfg_polygon(obj, x = NULL, y = NULL, z = NULL, m = NULL,
  linestring_id = NULL, close = TRUE)

```

Arguments

obj	matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sfg object of POLYGON geometry

Examples

```

sfg_polygon( matrix( 1:24, ncol = 2 ) )
sfg_polygon( matrix( 1:24, ncol = 3 ) )
sfg_polygon( matrix( 1:24, ncol = 4 ) )

## different lines
m <- cbind( matrix( 1:24, ncol = 2 ), c(rep(1, 6), rep(2, 6)) )
sfg_polygon( obj = m, x = 1, y = 2, linestring_id = 3 )

## just specifying linestring_id will use all others as the geometries
sfg_polygon( obj = m, linestring_id = 3 )

df <- data.frame( x = 1:12, y = 1:12, z = 13:24, id = c(rep(1,6), rep(2,6)))
sfg_polygon( df, x = "x", y = "y" )
sfg_polygon( df, x = "x", y = "y", linestring_id = "id" )

sfg_polygon( df, linestring_id = "id" )

```

sf_linestring

sf **LINestring**

Description

constructs sf of MULTIPOINT objects

Usage

```

sf_linestring(obj = NULL, x = NULL, y = NULL, z = NULL, m = NULL,
  linestring_id = NULL)

```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
linestring_id	column of ids for linestrings

Value

sf object of LINestring geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- matrix( c(1:4), ncol = 2 )
sf_linestring( x )

x <- data.frame( id = 1:2, x = 1:2, y = 2:1 )
sf_linestring( x )
sf_linestring( x, x = "x", y = "y" )
sf_linestring( x, x = "y", y = "x" )
sf_linestring( x, linestring_id = "id", x = "x", y = "y")
```

sf_multilinestring *sf* MULTILINESTRING

Description

constructs an sf of MULTILINESTRING objects

Usage

```
sf_multilinestring(obj = NULL, x = NULL, y = NULL, z = NULL,
  m = NULL, multilinestring_id = NULL, linestring_id = NULL)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multilinestring_id	column of ids for multilinestrings
linestring_id	column of ids for linestrings (within multilinestrings)

Value

sf object of MULTILINESTRING geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```

m <- matrix(c(0,0,0,0,1,1), ncol = 3 )
sf_multilinestring( m )

m <- matrix(c(0,0,0,0,0,1,0,1,1,1,2,2,1,2,3), ncol = 3, byrow = TRUE)
sf_multilinestring( obj = m )
sf_multilinestring( obj = m, multilinestring_id = 1 )
sf_multilinestring( obj = m, linestring_id = 1 )

sf_multilinestring( obj = m, linestring_id = 1, multilinestring_id = 1 )

sf_multilinestring( obj = m, x = 2, y = 3 )
sf_multilinestring( obj = m, x = 1, y = 2, z = 3 )
sf_multilinestring( obj = m, x = 2, y = 3, linestring_id = 1, multilinestring_id = 1 )

df <- data.frame(
  ml_id = c(1,1,1,1,1,1,1,1,2,2,2,2,2)
  , l_id = c(1,1,1,2,2,3,3,3,1,1,1,2,2)
  , x = rnorm(13)
  , y = rnorm(13)
  , z = rnorm(13)
  , m = rnorm(13)
)

sf_multilinestring( obj = df, x = "x", y = "y")
sf_multilinestring( obj = df, x = "x", y = "y", z = "z")
sf_multilinestring( obj = df, x = "x", y = "y", z = "z", m = "m")

sf_multilinestring( obj = df, x = 2, y = 3)
sf_multilinestring( obj = df, x = 2, y = 3, z = 4)
sf_multilinestring( obj = df, x = 2, y = 3, z = 4, m = 5)

sf_multilinestring( obj = df, multilinestring_id = "ml_id", linestring_id = "l_id" )
sf_multilinestring( obj = df, multilinestring_id = 1, linestring_id = 2 )

```

Description

constructs sf of MULTIPOINT objects

Usage

```
sf_multipoint(obj, x = NULL, y = NULL, z = NULL, m = NULL,  
multipoint_id = NULL)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multipoint_id	column of ids for multipoints

Value

sf object of MULTIPOINT geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- matrix( c(1:4), ncol = 2 )  
sf_multipoint( x )  
  
x <- data.frame( id = 1:2, x = 1:2, y = 2:1 )  
sf_multipoint( x )  
sf_multipoint( x, x = "x", y = "y" )  
sf_multipoint( x, x = "y", y = "x" )  
sf_multipoint( x, multipoint_id = "id", x = "x", y = "y")
```

sf_multipolygon	<i>sf MULTIPOLYGON</i>
-----------------	------------------------

Description

constructs an sf of MULTIPOLYGON objects

Usage

```
sf_multipolygon(obj = NULL, x = NULL, y = NULL, z = NULL,
  m = NULL, multipolygon_id = NULL, polygon_id = NULL,
  linestring_id = NULL, close = TRUE)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
multipolygon_id	column of ids for multipolygons
polygon_id	column of ids for polygons
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sf object of MULTIPOLYGON geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
m <- matrix(c(0,0,0,0,1,0,0,1,1,0,0,1,0,0,0), ncol = 3, byrow = TRUE )
sf_multipolygon( m )

df <- data.frame(
  id = c(1,1,1,1,1)
```

```

    , x = c(0,0,1,1,0)
    , y = c(0,1,1,0,0)
  )

sf_multipolygon( df, x = "x", y = "y" )

df <- data.frame(
  id = c(1,1,1,1,1,2,2,2,2,2)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sf_multipolygon( df, multipolygon_id = "id", polygon_id = "id", linestring_id = "id")

df <- data.frame(
  id1 = c(1,1,1,1,1,1,1,1,1,1)
  , id2 = c(1,1,1,1,1,2,2,2,2,2)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sf_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2")

df <- data.frame(
  id1 = c(1,1,1,1,1,1,1,1,1,2,2,2,2,2)
  , id2 = c(1,1,1,1,1,2,2,2,2,2,1,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1,3,3,4,4,3)
  , y = c(0,1,1,0,0,1,2,2,1,1,3,4,4,3,3)
)

sf_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2")

df <- data.frame(
  id1 = c(1,1,1,1,1,2,2,2,2,2)
  , id2 = c(1,1,1,1,1,1,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

sf_multipolygon( df, multipolygon_id = "id1", polygon_id = "id2" )
sf_multipolygon( df, polygon_id = "id1", linestring_id = "id2" )
sf_multipolygon( df, x = "x", y = "y", polygon_id = "id1")
sf_multipolygon( df, x = "x", y = "y", polygon_id = "id1", linestring_id =)
sf_multipolygon( df, x = "x", y = "y", linestring_id = "id1")
sf_multipolygon( df, x = "x", y = "y", linestring_id = "id2")

df <- data.frame(
  id1 = c('a','a','a','a','a','b','b','b','b','b')
  , id2 = c(1,1,1,1,1,1,1,1,1,1)
  , x = c(0,0,1,1,0,1,1,2,2,1)
  , y = c(0,1,1,0,0,1,2,2,1,1)
)

```

```
sf_multipolygon( df, x = "x", y = "y", polygon_id = "id1")
```

sf_point	<i>sf POINT</i>
----------	-----------------

Description

constructs sf of POINT objects

Usage

```
sf_point(obj, x = NULL, y = NULL, z = NULL, m = NULL)
```

Arguments

obj	sorted vector, matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column

Value

sf object of POINT geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
x <- c(1:3)
sf_point( x )

x <- matrix( c(1:10) , ncol = 2 )
sf_point( x )

x <- setNames( as.data.frame( x ), c("x","y") )
sf_point( x )
sf_point( obj = x, x = "x", y = "y" )
sf_point( obj = x, x = "y", y = "x" )
```

sf_polygon	<i>sf POLYGON</i>
------------	-------------------

Description

constructs an sf of POLYGON objects

Usage

```
sf_polygon(obj = NULL, x = NULL, y = NULL, z = NULL, m = NULL,
           polygon_id = NULL, linestring_id = NULL, close = TRUE)
```

Arguments

obj	sorted matrix or data.frame
x	x geometry column
y	y geometry column
z	z geometry column
m	m geometry column
polygon_id	column of ids for polygons
linestring_id	column of ids for lines (within polygons)
close	logical indicating whether polygons should be closed. If TRUE, all polygons will be checked and force closed if possible

Value

sf object of POLYGON geometries

notes

sfheaders functions do not perform any validity checks on the geometries. Nor do they set Coordinate Reference Systems, EPSG, PROJ4 or precision attributes.

The data.frame and matrices you send into the sfheader functions must be ordered.

Examples

```
m <- matrix(c(0,0,0,0,1,1), ncol = 2 )
sf_polygon( m )

m <- matrix(c(0,0,0,0,0,1,0,1,1,1,2,2,1,2,3,1,3,4), ncol = 3, byrow = TRUE)
sf_polygon( obj = m )
sf_polygon( obj = m, polygon_id = 1 )
sf_polygon( obj = m, linestring_id = 1 )

sf_polygon( obj = m, linestring_id = 1, polygon_id = 1 )
```

```
sf_polygon( obj = m, x = 2, y = 3 )
sf_polygon( obj = m, x = 1, y = 2, z = 3 )
sf_polygon( obj = m, x = 2, y = 3, linestring_id = 1, polygon_id = 1 )

df <- data.frame(
  ml_id = c(1,1,1,1,1,1,1,1,1,2,2,2,2,2,2)
  , l_id = c(1,1,1,2,2,2,3,3,3,1,1,1,2,2,2)
  , x = rnorm(15)
  , y = rnorm(15)
  , z = rnorm(15)
  , m = rnorm(15)
)

sf_polygon( obj = df, x = "x", y = "y")
sf_polygon( obj = df, x = "x", y = "y", z = "z")
sf_polygon( obj = df, x = "x", y = "y", z = "z", m = "m")

sf_polygon( obj = df, x = 2, y = 3)
sf_polygon( obj = df, x = 2, y = 3, z = 4)
sf_polygon( obj = df, x = 2, y = 3, z = 4, m = 5)

sf_polygon( obj = df, polygon_id = "ml_id", linestring_id = "l_id" )
sf_polygon( obj = df, polygon_id = 1, linestring_id = 2 )
```


Index

sf_linestring, 16
sf_multilinestring, 17
sf_multipoint, 18
sf_multipolygon, 20
sf_point, 22
sf_polygon, 23
sfc_linestring, 2
sfc_multilinestring, 3
sfc_multipoint, 4
sfc_multipolygon, 5
sfc_point, 7
sfc_polygon, 8
sfg_linestring, 10
sfg_multilinestring, 11
sfg_multipoint, 12
sfg_multipolygon, 13
sfg_point, 14
sfg_polygon, 15