Package: sf2 (via r-universe)

May 5, 2026

Type Package

Title Spatial Analysis tools

Version 0.1.6

Description sf2 extends the function of sf package.

License GPL-3

LazyData TRUE

import RoxygenNote: 6.1.0

Imports magrittr, stringr, purrr, terra, sf, tidyselect, rlang, data.table, dplyr, graphics, methods

Suggests testthat (>= 2.1.0), igraph, exactextractr, sp, knitr, rmarkdown, covr

VignetteBuilder knitr

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Encoding UTF-8

Depends R (>= 3.6)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://rpkgs.r-universe.dev

Date/Publication 2025-09-07 12:13:20 UTC

RemoteUrl https://github.com/rpkgs/sf2

RemoteRef HEAD

RemoteSha 2088240ee34486b52ad8fd5506a525b26e76a16a

Contents

`get_RegionalMean` *get_RegionalMean*

Description

get_RegionalMean

Usage

```
get_RegionalMean(f, shp = sf2::bou1_4p)

get_RegionalMeans(fs, shp = sf2::bou1_4p)
```

`guess_network` *guess_network*

Description

guess_network

Usage

```
guess_network(info)
```

Arguments

`info:`	the first two columns should be `[i, contains]`

References

https://stackoverflow.com/questions/25130462/get-disjoint-sets-from-a-list-in-r

Examples

```
## Not run:
info = shp_contains(shp)
l_net = guess_networ(info)

## End(Not run)
```

`load_data` *load_data*

Description

load_data

Usage

```
load_china_river()

load_china()

load_continent()
```

Details

- load_china_river
- `river$i$`: river polylines
- `river_p$i$`: river polygons
-
-

`make_grid` *make_grid*

Description

construct SpatialGridDataFrame

Usage

```
make_grid(
  range = c(-180, 180, -90, 90),
  cellsize = 1,
  midgrid = c(TRUE, TRUE),
  prj = NULL,
  type = "base",
  fix_lon360 = FALSE
)

get_grid.lonlat(lon, lat, fix_lon360 = FALSE)

fix_lon(x)
```

Arguments

range	A numeric vector, [lon_min, lon_max, lat_min, lat_max]
cellsize	Numeric vector, grid cell size [cellsize_lon, cellsize_lat].
midgrid	A vector, [midgrid_lon, midgrid_lat]. If midgrid = false, then begin point and end point locate on grid lines; If true, then begin point and end point in the middle of grid.
prj	sp::CRS-class() Projection.
type	Character, one of mat or vec. • base: If image(x) looks correct, x can be put into grid directly: grid@data <- data.frame(x = as.numeric(x)) • gdal: If image(x) looks correct, x needs to flipud: grid@data <- data.frame(x = as.numeric(Ipaper::flipud(x))) In version >= 0.1.3, c("mat", "vec") was renamed as c("base", "gdal").
fix_lon360	boolean

Details

The input matrix suits for image, can be directly input to make_grid.

Examples

```
range <- c(73, 105, 25, 40) # Tibetan Plateau
grid <- make_grid(range, cellsize = 1 / 2, midgrid = TRUE)
```

make_rast *make_rast*

Description

make_rast

Usage

```
make_rast(
  range = c(-180, 180, -90, 90),
  cellsize = 1,
  nlyrs = 1,
  vals = NULL,
  ...
)
```

Arguments

range	A numeric vector, [lon_min, lon_max, lat_min, lat_max]
cellsize	Numeric vector, grid cell size [cellsize_lon, cellsize_lat].
nlyrs	positive integer. Number of layers
...	other parameters to terra::rast(), e.g., names, vals.

See Also

terra::rast()

overlap_fraction *overlap_fraction*

Description

overlap_fraction

Usage

```
overlap_fraction(shp, cellsize = 0.1, range = NULL, outfile = NULL, ...)
```

Value

- I : the order of base, [lon, lat], image works directly on it.
- cell: the order of gdal, [lon, rev(lat)], returned by values(r),

Examples

```
## Not run:
library(nctools)

f = "//kong-nas/CMIP6/DATA/China/CN0.5.1_ChinaDaily_025x025/yearly/CN05.1_Tmax_1961_2021_yearly_025x025.nc"

data = ncread(f, "tmax")
image(data[, , 1]) # should works

r = rast(f)
range = as.vector(ext(r))
cellsize = res(r)

## End(Not run)

# shp = basin_Baihe
load_china()
shp = bou1_4p
cellsize <- 0.5
range <- st_range(shp, cellsize)

dat = overlap_fraction(shp, cellsize = cellsize, range, outfile = NULL)
print(dat)
```

---

rast_df *rast_coord*

Description

rast_coord

Usage

```
rast_df(r, .area = TRUE, na.rm = FALSE)
```

Value

- I : the order of base, [lon, lat], image works directly on it.
- cell: the order of gdal, [lon, rev(lat)], returned by values(r),

`rdist.earth` *rdist.earth*

Description

rdist.earth

Usage

```
rdist.earth(x1, x2 = NULL)
```

Arguments

`x1`	A position matrix `[lat, lon]`, degree unit.
`x2`	same as x1

`select.SpatRaster` *Select layers from a SpatRaster object*

Description

This function allows you to select specific layers from a SpatRaster object using tidyverse-style syntax similar to dplyr's select. Supports selection by name, position, and tidyselect helpers.

Usage

```
## S3 method for class 'SpatRaster'
select(.data, ...)
```

Arguments

`.data`	A SpatRaster object containing multiple layers
`...`	Expressions for selecting layers. You can use: • Layer names: `select(r, layer1, layer2)` • Numeric indices: `select(r, 1, 3)` • Negative selection: `select(r, -layer1)` (exclude layer1) • tidyselect helpers: `select(r, starts_with("temp"))`

Value

A SpatRaster object containing only the selected layers

Examples

```
## Not run:
library(terra)
r <- rast(c("elevation.tif", "slope.tif", "aspect.tif"))

# Select by name
r_elev <- select(r, elevation)

# Select multiple layers
r_sub <- select(r, elevation, slope)

# Exclude layers with negative selection
r_no_aspect <- select(r, -aspect)

# Use tidyselect helpers
r_s <- select(r, starts_with("s"))

## End(Not run)
```

`st_clip` *clip shapefile*

Description

clip shapefile

Usage

```
st_clip(x, y)
```

`st_dissolve` *st_dissolve*

Description

st_dissolve

Usage

```
st_dissolve(x, by = NULL, ...)

## S3 method for class 'sf'
st_dissolve(x, by = NULL, ...)

## S3 method for class 'character'
st_dissolve(x, by = NULL, ...)
```

st_extractId *st_extractId*

Description

st_extractId

Usage

```
st_extractId(x, y, plot = TRUE, ...)
```

Arguments

x	A sf point, or file path
y	A sf polygon, or file path
plot	Boolean. Whether to visualize extracted points?
...	others passed to sf::st_within()

st_range *get range of* sf *object*

Description

get range of sf object

Usage

```
st_range(x, cellsize = 0.1)
```

Arguments

x	sf object
cellsize	double

st_rect *st_rect*

Description

st_rect

Usage

```
st_rect(range, crs = st_crs(4326))
```

Arguments

range	[lon_min, lon_max, lat_min, lat_max]

Examples

```
## Not run:
xlim <- c(112, 115)
ylim <- c(21, 23)
range <- c(xlim, ylim)
poly <- st_rect(range)
write_sf(poly, "poly.shp")

## End(Not run)
```

write_shp *write and read ESRI Shapefile*

Description

write and read ESRI Shapefile

Usage

```
write_shp(x, file, ...)
```

Arguments

x	Spatial* Object from sp package
file	The path of ESRI shapefile
...	other parameters to sf::write_sf()

Index