

Package ‘ggspatial’

January 4, 2021

Type Package

Title Spatial Data Framework for ggplot2

Version 1.1.5

Maintainer Dewey Dunnington <dewey@fishandwhistle.net>

Description Spatial data plus the power of the ggplot2 framework means easier mapping when input data are already in the form of spatial objects.

License GPL-3

Depends R (>= 2.10)

Imports sf, ggplot2 (>= 3.0.0), rosm (>= 0.2), abind, methods, tibble, scales, tidyr, rlang, grid, glue

Suggests prettymapr, knitr, rmarkdown, sp, rgdal, raster, testthat (>= 2.1.0), dplyr, withr, ggrepel, stars, covr, vdiff, lwgeom

URL <https://paleolimbot.github.io/ggspatial/>,
<https://github.com/paleolimbot/ggspatial>

BugReports <https://github.com/paleolimbot/ggspatial/issues>

LazyData TRUE

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Dewey Dunnington [aut, cre] (<<https://orcid.org/0000-0002-9415-4582>>),
Brent Thorne [ctb] (<<https://orcid.org/0000-0002-1099-3857>>)

Repository CRAN

Date/Publication 2021-01-04 17:30:15 UTC

R topics documented:

annotation_map_tile	2
annotation_north_arrow	3
annotation_scale	5

annotation_spatial_hline	7
df_spatial	8
fixed_plot_aspect	9
geom_polypath	10
geom_spatial_rect	11
geom_spatial_segment	12
layer_spatial	14
layer_spatial.bbox	16
layer_spatial.Raster	17
layer_spatial.stars	18
load_longlake_data	20
north_arrow_orientteering	21
stat_spatial_identity	22
xy_transform	24

Index	25
--------------	-----------

annotation_map_tile *Add background OSM tiles*

Description

Uses `rosm::osm.image()` to add background tiles. If you are publishing a map using these tiles, make sure to use the proper attribution (e.g., "Copyright OpenStreetMap contributors" when using an OpenStreetMap-based tile set).

Usage

```

annotation_map_tile(
  type = "osm",
  zoom = NULL,
  zoomin = -2,
  forcedownload = FALSE,
  cachedir = NULL,
  progress = c("text", "none"),
  quiet = TRUE,
  interpolate = TRUE,
  data = NULL,
  mapping = NULL,
  alpha = 1
)

GeomMapTile

```

Arguments

type	The map type (one of that returned by <code>rosm::osm.types</code>)
zoom	The zoom level (overrides zoomin)
zoomin	Delta on default zoom. The default value is designed to download fewer tiles than you probably want. Use -1 or 0 to increase the resolution.
forcedownload	Re-download cached tiles?
cachedir	Specify cache directory
progress	Use <code>progress = "none"</code> to suppress progress and zoom output
quiet	Use <code>quiet = FALSE</code> to see which URLs are downloaded
interpolate	Passed to <code>grid::rasterGrob()</code>
data, mapping	Specify data and mapping to use this geom with facets
alpha	Use to make this layer semi-transparent

Format

An object of class `GeomMapTile` (inherits from `Geom`, `ggproto`, `gg`) of length 5.

Value

A `ggplot2` layer

Examples

```
library(ggplot2)
load_longlake_data()

ggplot() +
  annotation_map_tile(zoom = 13, cachedir = system.file("rosm.cache", package = "ggspatial")) +
  geom_sf(data = longlake_waterdf, fill = NA, col = "grey50")
```

annotation_north_arrow

Spatial-aware north arrow

Description

Spatial-aware north arrow

Usage

```

annotation_north_arrow(
  mapping = NULL,
  data = NULL,
  ...,
  height = unit(1.5, "cm"),
  width = unit(1.5, "cm"),
  pad_x = unit(0.25, "cm"),
  pad_y = unit(0.25, "cm"),
  rotation = NULL,
  style = north_arrow_orienteing
)

```

GeomNorthArrow

Arguments

mapping, data, ...	See Aesthetics
height, width	Height and width of north arrow
pad_x, pad_y	Padding between north arrow and edge of frame
rotation	Override the rotation of the north arrow (degrees counterclockwise)
style	A grob or callable that produces a grob that will be drawn as the north arrow. See north_arrow_orienteing for options.

Format

An object of class GeomNorthArrow (inherits from Geom, ggproto, gg) of length 5.

Value

A ggplot2 layer

Aesthetics

The following can be used as parameters or aesthetics. Using them as aesthetics is useful when facets are used to display multiple panels, and a different (or missing) scale bar is required in different panels. Otherwise, just pass them as arguments to `annotation_north_arrow()`.

- `which_north`: "grid" results in a north arrow always pointing up; "true" always points to the north pole from whichever corner of the map the north arrow is in.
- `location`: Where to put the scale bar ("tl" for top left, etc.)

Examples

```

cities <- data.frame(
  x = c(-63.58595, 116.41214),

```

```
y = c(44.64862, 40.19063),
city = c("Halifax", "Beijing")
)

ggplot(cities) +
  geom_spatial_point(aes(x, y), crs = 4326) +
  annotation_north_arrow(which_north = "true") +
  coord_sf(crs = 3995)

ggplot(cities) +
  geom_spatial_point(aes(x, y), crs = 4326) +
  annotation_north_arrow(which_north = "grid") +
  coord_sf(crs = 3995)
```

annotation_scale

Spatial-aware scalebar annotation

Description

Spatial-aware scalebar annotation

Usage

```
annotation_scale(
  mapping = NULL,
  data = NULL,
  ...,
  plot_unit = NULL,
  bar_cols = c("black", "white"),
  line_width = 1,
  height = unit(0.25, "cm"),
  pad_x = unit(0.25, "cm"),
  pad_y = unit(0.25, "cm"),
  text_pad = unit(0.15, "cm"),
  text_cex = 0.7,
  text_face = NULL,
  text_family = "",
  tick_height = 0.6
)
```

GeomScaleBar

Arguments

mapping, data, ...

See Aesthetics

plot_unit	For non-coord_sf applications, specify the unit for x and y coordinates. Must be one of km, m, cm, mi, ft, or in.
bar_cols	Colours to use for the bars
line_width	Line width for scale bar
height	Height of scale bar
pad_x, pad_y	Distance between scale bar and edge of panel
text_pad, text_cex, text_face, text_family	Parameters for label
tick_height	Height of ticks relative to height of scale bar

Format

An object of class `GeomScaleBar` (inherits from `Geom`, `ggproto`, `gg`) of length 5.

Value

A `ggplot2` layer.

Aesthetics

The following can be used as parameters or aesthetics. Using them as aesthetics is useful when facets are used to display multiple panels, and a different (or missing) scale bar is required in different panels. Otherwise, just pass them as arguments to `annotation_scale`.

- `width_hint`: The (suggested) proportion of the plot area which the scalebar should occupy.
- `unit_category`: Use "metric" or "imperial" units.
- `style`: One of "bar" or "ticks"
- `location`: Where to put the scale bar ("tl" for top left, etc.)
- `line_col` and `text_col`: Line and text colour, respectively

Examples

```
cities <- data.frame(
  x = c(-63.58595, 116.41214),
  y = c(44.64862, 40.19063),
  city = c("Halifax", "Beijing")
)

ggplot(cities) +
  geom_spatial_point(aes(x, y), crs = 4326) +
  annotation_scale() +
  coord_sf(crs = 3995)
```

`annotation_spatial_hline`*Projected horizontal and vertical lines*

Description

Projected horizontal and vertical lines

Usage

```
annotation_spatial_hline(  
  mapping = NULL,  
  data = NULL,  
  stat = "identity",  
  ...,  
  intercept = waiver(),  
  limits = NULL,  
  detail = 100,  
  crs = NULL,  
  na.rm = FALSE,  
  show.legend = NA  
)
```

```
annotation_spatial_vline(  
  mapping = NULL,  
  data = NULL,  
  stat = "identity",  
  ...,  
  intercept = waiver(),  
  limits = NULL,  
  detail = 100,  
  crs = NULL,  
  na.rm = FALSE,  
  show.legend = NA  
)
```

GeomSpatialXline

Arguments

<code>mapping</code>	An aesthetic mapping created with <code>ggplot2::aes()</code> .
<code>data</code>	A data frame or other object, coerced to a data.frame by <code>ggplot2::fortify()</code> .
<code>stat</code>	Statistical transformation to use on this layer. See <code>ggplot2::layer()</code> .
<code>...</code>	Passed to the combined stat/geom as parameters or fixed aesthetics.
<code>intercept</code>	The x or y value that should be constant in the given crs. Can also be passed as an aesthetic through data and mapping.

limits	Use NULL to guess the minimum and maximum x or y value in the non-constant dimension, or specify a vector of length 2 to specify manually.
detail	The number of points that should be used when converting the line into segments.
crs	The crs of the x and y aesthetics, or NULL to use default lon/lat crs (with a message).
na.rm	Should missing aesthetic values be removed?
show.legend	See <code>ggplot2::layer()</code> .

Format

An object of class `GeomSpatialXline` (inherits from `GeomHline`, `Geom`, `ggproto`, `gg`) of length 4.

Examples

```
cities <- data.frame(
  x = c(-63.58595, 116.41214, 0),
  y = c(44.64862, 40.19063, 89.9),
  city = c("Halifax", "Beijing", "North Pole")
)

p <- ggplot(cities, aes(x, y, label = city)) +
  geom_spatial_point(crs = 4326) +
  # view of the north pole
  coord_sf(crs = 3995)

p +
  # longitude lines
  annotation_spatial_vline(
    intercept = seq(-180, 180, by = 10),
    crs = 4326
  ) +
  # latitude lines
  annotation_spatial_hline(
    intercept = seq(0, 90, by = 10),
    crs = 4326
  )
```

df_spatial

Create a ggplot-friendly data frame from a spatial object

Description

Create a ggplot-friendly data frame from a spatial object

Usage

```
df_spatial(x, ...)
```


Arguments

x A spatial object
 ... Passed to specific methods

Value

A tibble with coordinates as x and y, features as feature_id, and parts as part_id.

Examples

```
load_longlake_data()
df_spatial(longlake_osm)
df_spatial(longlake_depthdf)
df_spatial(as(longlake_depthdf, "Spatial"))
```

fixed_plot_aspect *Enforce a plot aspect ratio*

Description

When using a fixed-aspect coordinate system, `fixed_plot_aspect()` expands either the width or height of the plot to ensure that the output has dimensions that make sense. This is a useful workaround for getting reasonable-shaped plots when using `ggplot2::coord_sf()` or `ggplot2::coord_fixed()` when the data happen to be aligned vertically or horizontally.

Usage

```
fixed_plot_aspect(ratio = 1)
```

Arguments

ratio The desired aspect ratio (width / height)

Value

A `ggplot2::layer()` that can be added to a `ggplot2::ggplot()`.

Examples

```
library(ggplot2)
df <- data.frame(x = 0:5, y = seq(0, 10, length.out = 6))
ggplot(df, aes(x, y)) +
  geom_point() +
  fixed_plot_aspect(ratio = 1) +
  coord_fixed()
```

geom_polypath

*Polygons with holes in ggplot2***Description**

This geometry correctly plots polygons with holes in ggplot2 at the expense of doing so (slightly) more slowly than [geom_polygon](#). This implementation fixes a bug in the ggpolypath package, which provides similar functionality.

Usage

```
geom_polypath(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  rule = "winding",
  ...
)
```

Arguments

mapping	An aesthetic mapping, created with aes . The aesthetic will mostly likely need to contain a group mapping.
data	A data.frame containing the coordinates to plot.
stat	A statistic to apply (most likely "identity")
position	A position to apply (most likely "identity")
na.rm	Should missing coordinate be removed?
show.legend	Should a legend be shown for mapped aesthetics?
inherit.aes	Should aesthetics be inherited?
rule	A fill rule to apply. One of "winding" or "evenodd".
...	Passed to the geom and/or stat.

Value

A ggplot2 layer

Examples

```
library(ggplot2)
load_longlake_data()
ggplot(df_spatial(longlake_waterdf), aes(x, y, group = piece_id)) +
  geom_polypath()
```

geom_spatial_rect	<i>Projected rectangular regions</i>
-------------------	--------------------------------------

Description

If you need to plot a `sf::st_bbox()`, use `layer_spatial()` instead. While the implementation is slightly different, these functions are intended to behave identically to `ggplot2::geom_rect()` and `ggplot2::geom_tile()`.

Usage

```
geom_spatial_rect(  
  mapping = NULL,  
  data = NULL,  
  ...,  
  crs = NULL,  
  detail = 30,  
  linejoin = "mitre",  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE  
)
```

```
geom_spatial_tile(  
  mapping = NULL,  
  data = NULL,  
  ...,  
  crs = NULL,  
  detail = 30,  
  linejoin = "mitre",  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE  
)
```

StatSpatialRect

StatSpatialTile

Arguments

mapping	An aesthetic mapping created with <code>ggplot2::aes()</code> .
data	A data frame or other object, coerced to a data.frame by <code>ggplot2::fortify()</code> .
...	Passed to the combined stat/geom as parameters or fixed aesthetics.
crs	The crs of the x and y aesthetics, or NULL to use default lon/lat crs (with a message).

detail	Passed to <code>sf::st_segmentize()</code> : the number of line segments per quadrant of the bounding box. Increase this number for a smoother projected bounding box.
linejoin	How corners should be joined
na.rm	Should missing aesthetic values be removed?
show.legend	See <code>ggplot2::layer()</code> .
inherit.aes	See <code>ggplot2::layer()</code> .

Format

An object of class `StatSpatialRect` (inherits from `Stat`, `ggproto`, `gg`) of length 4.

An object of class `StatSpatialTile` (inherits from `StatSpatialRect`, `Stat`, `ggproto`, `gg`) of length 4.

Examples

```
library(ggplot2)
tile_df <- expand.grid(
  x = seq(-140, -52, by = 20),
  y = seq(40, 70, by = 10)
)

ggplot(tile_df, aes(x, y)) +
  geom_spatial_tile(crs = 4326) +
  coord_sf(crs = 3979)

# the same plot using geom_spatial_rect()
ggplot(
  tile_df,
  aes(xmin = x - 10, xmax = x + 10, ymin = y - 5, ymax = y + 5)
) +
  geom_spatial_rect(crs = 4326) +
  coord_sf(crs = 3979)
```

geom_spatial_segment *Spatial line segments*

Description

While the implementation is slightly different, this function is intended to behave identically to `ggplot2::geom_segment()`. Use `great_circle = FALSE` and `detail = NULL` if you wish ignore the fact that the earth is round.

Usage

```
geom_spatial_segment(
  mapping = NULL,
  data = NULL,
  ...,
  crs = NULL,
  detail = waiver(),
  great_circle = TRUE,
  wrap_dateline = TRUE,
  arrow = NULL,
  lineend = "butt",
  linejoin = "round",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

StatSpatialSegment

Arguments

mapping	An aesthetic mapping created with <code>ggplot2::aes()</code> .
data	A data frame or other object, coerced to a data.frame by <code>ggplot2::fortify()</code> .
...	Passed to the combined stat/geom as parameters or fixed aesthetics.
crs	The crs of the x and y aesthetics, or NULL to use default lon/lat crs (with a message).
detail	Passed to <code>sf::st_segmentize()</code> : the number of line segments per quadrant of the bounding box. Increase this number for a smoother projected bounding box.
great_circle	If TRUE, use <code>lwgeom::st_geod_segmentize()</code> to connect the (x, y) and (xend, yend) with the shortest possible great circle along the earth.
wrap_dateline	When using <code>great_circle = TRUE</code> , using <code>wrap_dateline = TRUE</code> splits the great circle along the dateline. You may want to pass FALSE here if using <code>arrow</code> and a projection that wraps the dateline.
arrow	An arrow specification as a call to <code>grid::arrow()</code> .
lineend	See <code>ggplot2::geom_segment()</code> .
linejoin	How corners should be joined
na.rm	Should missing aesthetic values be removed?
show.legend	See <code>ggplot2::layer()</code> .
inherit.aes	See <code>ggplot2::layer()</code> .

Format

An object of class `StatSpatialSegment` (inherits from `StatSpatialRect`, `Stat`, `ggproto`, `gg`) of length 3.

Examples

```

library(ggplot2)

# visualize flights from
# Halifax -> Anchorage -> Berlin -> Halifax
cities <- data.frame(
  lon = c(-63.58595, 116.41214, 13.50, -149.75),
  lat = c(44.64862, 40.19063, 52.51, 61.20),
  city = c("Halifax", "Beijing", "Berlin", "Anchorage"),
  city_to = c("Anchorage", "Beijing", "Berlin", "Halifax")
)

cities$lon_end <- cities$lon[c(4, 3, 1, 2)]
cities$lat_end <- cities$lat[c(4, 3, 1, 2)]

p <- ggplot(cities, aes(lon, lat, xend = lon_end, yend = lat_end)) +
  geom_spatial_point(crs = 4326)

# by default, geom_spatial_segment() connects points
# using the shortest distance along the face of the earth
# wrapping at the date line
p +
  geom_spatial_segment(crs = 4326) +
  coord_sf(crs = 3857)

# to let the projection handle the dateline,
# use `wrap_dateline = FALSE` (most useful for
# when using `arrow`)
p +
  geom_spatial_segment(
    wrap_dateline = FALSE,
    arrow = grid::arrow(),
    crs = 4326
  ) +
  coord_sf(crs = 3995)

# to ignore the roundness of the earth, use
# `great_circle = FALSE`
p +
  geom_spatial_segment(
    great_circle = FALSE,
    arrow = grid::arrow(),
    crs = 4326
  ) +
  coord_sf(crs = 3995)

```

Description

See also [layer_spatial.Raster\(\)](#), [layer_spatial.stars\(\)](#), and [layer_spatial.bbox\(\)](#) for implementations for other types of spatial objects.

Usage

```
layer_spatial(data, mapping, ...)  
  
annotation_spatial(data, mapping, ...)  
  
## Default S3 method:  
layer_spatial(  
  data,  
  mapping = aes(),  
  inherit.aes = FALSE,  
  sf_params = list(),  
  ...  
)  
  
## Default S3 method:  
annotation_spatial(  
  data,  
  mapping = aes(),  
  inherit.aes = FALSE,  
  sf_params = list(),  
  ...  
)  
  
shadow_spatial(data, ...)  
  
## Default S3 method:  
shadow_spatial(data, ...)
```

Arguments

data	An object that can be coerced to an sf object using st_as_sf .
mapping	A mapping, created using aes .
...	Passed to geom_sf
inherit.aes	Inherit aesthetics from <code>ggplot()</code> ?
sf_params	Passed to st_as_sf .

Value

A `ggplot2` [layer](#).

Examples

```

library(ggplot2)
load_longlake_data()

ggplot() +

  # annotation_spatial() layers don't train the scales, so data stays central
  annotation_spatial(longlake_roadsdf, size = 2, col = "black") +
  annotation_spatial(longlake_roadsdf, size = 1.6, col = "white") +

  # raster layers train scales and get projected automatically
  layer_spatial(longlake_depth_raster, aes(alpha = stat(band1)), fill = "darkblue") +
  scale_alpha_continuous(na.value = 0) +

  # layer_spatial() layers train the scales
  layer_spatial(longlake_depthdf, aes(col = DEPTH_M)) +

  # spatial-aware automagic scale bar
  annotation_scale(location = "tl") +

  # spatial-aware automagic north arrow
  annotation_north_arrow(location = "br", which_north = "true")

```

layer_spatial.bbox *Add a bounding box to a map*

Description

To include a bounding box without drawing it, use `shadow_spatial()` on the original object.

Usage

```

## S3 method for class 'bbox'
layer_spatial(data, mapping = aes(), ..., detail = 30)

## S3 method for class 'bbox'
annotation_spatial(data, mapping = aes(), ..., detail = 30)

## S3 method for class 'bbox'
shadow_spatial(data, ..., detail = 30)

```

Arguments

data	A bounding box generated by <code>sf::st_bbox()</code>
mapping	A mapping, created using <code>aes</code> .
...	Passed to <code>geom_sf</code>
detail	Passed to <code>sf::st_segmentize()</code> : the number of line segments per quadrant of the bounding box. Increase this number for a smoother projected bounding box.

Examples

```

library(ggplot2)
load_longlake_data()
ggplot() +
  layer_spatial(sf::st_bbox(longlake_waterdf)) +
  layer_spatial(longlake_depthdf)

# use shadow_spatial() to include the geographic area of an object
# without drawing it
ggplot() +
  shadow_spatial(longlake_waterdf) +
  layer_spatial(longlake_depthdf)

```

layer_spatial.Raster *Spatial ggplot2 layer for raster objects*

Description

This is intended for use with RGB(A) rasters (e.g., georeferenced imagery or photos). To work with bands as if they were columns, use [df_spatial](#) and [geom_raster](#).

Usage

```

## S3 method for class 'Raster'
layer_spatial(
  data,
  mapping = NULL,
  interpolate = NULL,
  is_annotation = FALSE,
  lazy = FALSE,
  dpi = 150,
  ...
)

## S3 method for class 'Raster'
annotation_spatial(data, mapping = NULL, interpolate = NULL, ...)

```

StatSpatialRaster

StatSpatialRasterAnnotation

StatSpatialRasterDf

GeomSpatialRaster

Arguments

data	A Raster object
mapping	Currently, only RGB or RGBA rasters are supported. In the future, one may be able to map specific bands to the fill and alpha aesthetics.
interpolate	Interpolate resampling for rendered raster image
is_annotation	Lets raster exist without modifying scales
lazy	Delay projection and resample of raster until the plot is being rendered
dpi	if lazy = TRUE, the dpi to which the raster should be resampled
...	Passed to other methods

Format

An object of class `StatSpatialRaster` (inherits from `Stat`, `ggproto`, `gg`) of length 3.

An object of class `StatSpatialRaster` (inherits from `StatSpatialRaster`, `Stat`, `ggproto`, `gg`) of length 3.

An object of class `StatSpatialRasterDf` (inherits from `Stat`, `ggproto`, `gg`) of length 5.

An object of class `GeomSpatialRaster` (inherits from `Geom`, `ggproto`, `gg`) of length 5.

Value

A `ggplot2` layer

Examples

```
library(ggplot2)
load_longlake_data()
ggplot() + layer_spatial(longlake_osm)
ggplot() + layer_spatial(longlake_depth_raster) + scale_fill_continuous(na.value = NA)
```

layer_spatial.stars *Spatial ggplot2 layer for stars objects*

Description

This is intended for use with RGB(A) rasters (e.g., georeferenced imagery or photos). To work with bands as if they were columns, use [df_spatial](#) and [geom_raster](#).

Usage

```
## S3 method for class 'stars'
layer_spatial(
  data,
  mapping = NULL,
  interpolate = NULL,
  is_annotation = FALSE,
  lazy = FALSE,
  dpi = 150,
  options = character(),
  ...
)

## S3 method for class 'stars'
annotation_spatial(data, mapping = NULL, interpolate = NULL, ...)

StatSpatialStars

StatSpatialStarsAnnotation

StatSpatialStarsDf

GeomSpatialStars
```

Arguments

data	A stars object
mapping	Currently, only RGB or RGBA rasters are supported. In the future, one may be able to map specific bands to the fill and alpha aesthetics.
interpolate	Interpolate resampling for rendered raster image
is_annotation	Lets raster exist without modifying scales
lazy	Delay projection and resample of raster until the plot is being rendered
dpi	if lazy = TRUE, the dpi to which the raster should be resampled
options	GDAL options for warping/resampling (see st_warp)
...	Passed to other methods

Format

An object of class StatSpatialStars (inherits from Stat, ggproto, gg) of length 3.

An object of class StatSpatialStars (inherits from StatSpatialStars, Stat, ggproto, gg) of length 3.

An object of class StatSpatialStarsDf (inherits from Stat, ggproto, gg) of length 5.

An object of class GeomSpatialStars (inherits from Geom, ggproto, gg) of length 5.

Value

A ggplot2 layer

load_longlake_data *Load longlake test data*

Description

Load longlake test data

Usage

```
load_longlake_data(  
  env = parent.frame(),  
  vector_format = c("sf", "sp"),  
  raster_format = c("raster", "stars", "stars_proxy"),  
  which = NULL  
)
```

Arguments

env The environment in which to assign the objects

vector_format, raster_format
 The format in which objects should be loaded

which An optional subset of objects to be loaded

Source

The Nova Scotia Topographic Database (<https://geonova.novascotia.ca/>) and Open Street Map (<https://www.openstreetmap.org/>).

Examples

```
load_longlake_data()
```

north_arrow_ortienteering
North arrow styles

Description

North arrow styles

Usage

```
north_arrow_ortienteering(  
    line_width = 1,  
    line_col = "black",  
    fill = c("white", "black"),  
    text_col = "black",  
    text_family = "",  
    text_face = NULL,  
    text_size = 10,  
    text_angle = 0  
)  
  
north_arrow_fancy_ortienteering(  
    line_width = 1,  
    line_col = "black",  
    fill = c("white", "black"),  
    text_col = "black",  
    text_family = "",  
    text_face = NULL,  
    text_size = 10,  
    text_angle = 0  
)  
  
north_arrow_minimal(  
    line_width = 1,  
    line_col = "black",  
    fill = "black",  
    text_col = "black",  
    text_family = "",  
    text_face = NULL,  
    text_size = 10  
)  
  
north_arrow_nautical(  
    line_width = 1,  
    line_col = "black",  
    fill = c("black", "white"),  
    text_size = 10,  
)
```

```

    text_face = NULL,
    text_family = "",
    text_col = "black",
    text_angle = 0
  )

```

Arguments

```

line_width, line_col, fill
    Parameters customizing the appearance of the north arrow
text_col, text_family, text_face, text_size, text_angle
    Parameters customizing the text of the north arrow

```

Value

A Grob with npc coordinates (more or less) 0 to 1

Examples

```

grid::grid.newpage()
grid::grid.draw(north_arrow_orienteeing())

grid::grid.newpage()
grid::grid.draw(north_arrow_fancy_orienteeing())

grid::grid.newpage()
grid::grid.draw(north_arrow_minimal())

grid::grid.newpage()
grid::grid.draw(north_arrow_nautical())

```

stat_spatial_identity *Spatial-aware ggplot2 layers*

Description

These layers are much like their counterparts, [stat_identity](#), [geom_point](#), [geom_path](#), and [geom_polygon](#), except they have a `crs` argument that ensures they are projected when using `coord_sf`. Stats are applied to the x and y coordinates that have been transformed.

Usage

```

stat_spatial_identity(
  mapping = NULL,
  data = NULL,
  crs = NULL,
  geom = "point",

```

```

    position = "identity",
    ...,
    show.legend = NA,
    inherit.aes = TRUE
  )

geom_spatial_point(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_path(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_polygon(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_text(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_label(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_text_repel(mapping = NULL, data = NULL, crs = NULL, ...)

geom_spatial_label_repel(mapping = NULL, data = NULL, crs = NULL, ...)

```

Arguments

mapping	An aesthetic mapping created with <code>ggplot2::aes()</code> .
data	A data frame or other object, coerced to a <code>data.frame</code> by <code>ggplot2::fortify()</code> .
crs	The crs of the x and y aesthetics, or <code>NULL</code> to use default lon/lat crs (with a message).
geom	The geometry to use.
position	The position to use.
...	Passed to the combined stat/geom as parameters or fixed aesthetics.
show.legend, inherit.aes	See <code>ggplot2::layer()</code> .

Value

A `ggplot2::layer()`.

Examples

```

cities <- data.frame(
  x = c(-63.58595, 116.41214, 0),
  y = c(44.64862, 40.19063, 89.9),
  city = c("Halifax", "Beijing", "North Pole")
)

library(ggrepel)
ggplot(cities, aes(x, y)) +
  geom_spatial_point(crs = 4326) +
  stat_spatial_identity(aes(label = city), geom = "label_repel") +

```

```
coord_sf(crs = 3857)
```

xy_transform	<i>Coordinate transform</i>
--------------	-----------------------------

Description

Coordinate transform, propotating non-finite cases.

Usage

```
xy_transform(x, y, from = 4326, to = 4326, na.rm = FALSE)
```

Arguments

x	The x coordinate
y	The y coordinate
from	From CRS
to	To CRS
na.rm	Warn for non-finite cases?

Value

A data.frame with x and y components.

Examples

```
xy_transform(c(1, 2, 3), c(1, 2, 3), to = 3857)  
xy_transform(c(1, 2, 3), c(NA, NA, NA), to = 3857)  
xy_transform(c(1, 2, 3), c(NA, 2, 3), to = 3857)  
xy_transform(c(1, 2, 3), c(1, 2, NA), to = 3857)
```


Index

- * **datasets**
 - annotation_map_tile, 2
 - annotation_north_arrow, 3
 - annotation_scale, 5
 - annotation_spatial_hline, 7
 - geom_spatial_rect, 11
 - geom_spatial_segment, 12
 - layer_spatial.Raster, 17
 - layer_spatial.stars, 18
- aes, 10, 15, 16
- annotation_map_tile, 2
- annotation_north_arrow, 3
- annotation_scale, 5
- annotation_spatial (layer_spatial), 14
- annotation_spatial.bbox
 - (layer_spatial.bbox), 16
- annotation_spatial.Raster
 - (layer_spatial.Raster), 17
- annotation_spatial.stars
 - (layer_spatial.stars), 18
- annotation_spatial_hline, 7
- annotation_spatial_vline
 - (annotation_spatial_hline), 7
- coord_sf, 22
- df_spatial, 8, 17, 18
- fixed_plot_aspect, 9
- fixed_plot_aspect(), 9
- geom_path, 22
- geom_point, 22
- geom_polygon, 10, 22
- geom_polypath, 10
- geom_raster, 17, 18
- geom_sf, 15, 16
- geom_spatial_label
 - (stat_spatial_identity), 22
- geom_spatial_label_repel
 - (stat_spatial_identity), 22
- geom_spatial_path
 - (stat_spatial_identity), 22
- geom_spatial_point
 - (stat_spatial_identity), 22
- geom_spatial_polygon
 - (stat_spatial_identity), 22
- geom_spatial_rect, 11
- geom_spatial_segment, 12
- geom_spatial_text
 - (stat_spatial_identity), 22
- geom_spatial_text_repel
 - (stat_spatial_identity), 22
- geom_spatial_tile (geom_spatial_rect), 11
- GeomMapTile (annotation_map_tile), 2
- GeomNorthArrow
 - (annotation_north_arrow), 3
- GeomScaleBar (annotation_scale), 5
- GeomSpatialRaster
 - (layer_spatial.Raster), 17
- GeomSpatialStars (layer_spatial.stars), 18
- GeomSpatialXline
 - (annotation_spatial_hline), 7
- ggplot2::aes(), 7, 11, 13, 23
- ggplot2::coord_fixed(), 9
- ggplot2::coord_sf(), 9
- ggplot2::fortify(), 7, 11, 13, 23
- ggplot2::geom_rect(), 11
- ggplot2::geom_segment(), 12, 13
- ggplot2::geom_tile(), 11
- ggplot2::ggplot(), 9
- ggplot2::layer(), 7–9, 12, 13, 23
- grid::arrow(), 13
- grid::rasterGrob(), 3
- layer, 15
- layer_spatial, 14

`layer_spatial()`, 11
`layer_spatial.bbox`, 16
`layer_spatial.bbox()`, 15
`layer_spatial.Raster`, 17
`layer_spatial.Raster()`, 15
`layer_spatial.stars`, 18
`layer_spatial.stars()`, 15
`load_longlake_data`, 20
`lwgeom::st_geod_segmentize()`, 13

`north_arrow_fancy_orienteering`
 (`north_arrow_orienteering`), 21
`north_arrow_minimal`
 (`north_arrow_orienteering`), 21
`north_arrow_nautical`
 (`north_arrow_orienteering`), 21
`north_arrow_orienteering`, 4, 21

`rosm::osm.image()`, 2
`rosm::osm.types`, 3

`sf::st_bbox()`, 11, 16
`sf::st_segmentize()`, 12, 13, 16
`shadow_spatial(layer_spatial)`, 14
`shadow_spatial()`, 16
`shadow_spatial.bbox`
 (`layer_spatial.bbox`), 16
`st_as_sf`, 15
`st_warp`, 19
`stat_identity`, 22
`stat_spatial_identity`, 22
`StatSpatialRaster`
 (`layer_spatial.Raster`), 17
`StatSpatialRasterAnnotation`
 (`layer_spatial.Raster`), 17
`StatSpatialRasterDf`
 (`layer_spatial.Raster`), 17
`StatSpatialRect(geom_spatial_rect)`, 11
`StatSpatialSegment`
 (`geom_spatial_segment`), 12
`StatSpatialStars(layer_spatial.stars)`,
 18
`StatSpatialStarsAnnotation`
 (`layer_spatial.stars`), 18
`StatSpatialStarsDf`
 (`layer_spatial.stars`), 18
`StatSpatialTile(geom_spatial_rect)`, 11

`xy_transform`, 24