

# Package ‘ggspatial’

December 14, 2018

**Type** Package

**Title** Spatial Data Framework for ggplot2

**Version** 1.0.3

**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), ggplot2 (>= 2.2.1.9000)

**Imports** sf, rosm (>= 0.2), abind, reshape2, methods, plyr, raster, tibble, scales, tidyr, rlang, grid

**Suggests** prettypapr, mapproj, knitr, rmarkdown, sp, rgdal, testthat, dplyr, withr, ggrepel, stars, covr

**URL** <https://github.com/paleolimbot/ggspatial>

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

**LazyData** TRUE

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2018-12-14 21:10:04 UTC

## R topics documented:

annotation_map_tile . . . . .	2
annotation_north_arrow . . . . .	3
annotation_scale . . . . .	4
df_spatial . . . . .	6

geom_polypath . . . . .	6
layer_spatial . . . . .	7
layer_spatial.Raster . . . . .	8
load_longlake_data . . . . .	10
north_arrow_ortienteering . . . . .	10
stat_spatial_identity . . . . .	11
xy_transform . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

annotation\_map\_tile *Add background OSM tiles*

---

## Description

Uses [osm.image](#) to add background tiles.

## 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)
```

GeomMapTile

## Arguments

type	The map type
zoom	The zoom level (overrides zoomin)
zoomin	Delta on default zoom
forcedownload	Re-download cached tiles?
cachedir	Specify cache directory
progress	Use progress = "none" to suppress progress and zoom output
quiet	Use quiet = FALSE to see which URLs are downloaded
interpolate	Parameter for raster
data, mapping	Specify data and mapping to use this geom with facets

## Format

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

## Value

A ggplot2 layer

**Examples**

```
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 <a href="#">north_arrow_orienteing</a> 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)
```

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, and features as .feature

**Examples**

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

---

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 ggpoly path 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 <a href="#">aes</a> . 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**

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

---

layer_spatial	<i>Turn a spatial object into a ggplot2 layer</i>
---------------	---

---

**Description**

Turn a spatial object into a ggplot2 layer

**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(), ...)
```

**Arguments**

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

**Value**

A ggplot2 [layer](#).

**Examples**

```
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.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 = TRUE,
  is_annotation = FALSE, lazy = FALSE, dpi = 150, ...)

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

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

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

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.

**Value**

A ggplot2 layer

**Examples**

```
load_longlake_data()
ggplot() + layer_spatial(longlake_osm)
```

```
ggplot() + layer_spatial(longlake_depth_raster) + scale_fill_continuous(na.value = NA)
```

---

load\_longlake\_data      *Load longlake test data*

---

### Description

Load longlake test data

### Usage

```
load_longlake_data(env = parent.frame())
```

### Arguments

env                      The environment in which to assign the objects

### Source

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

### Examples

```
load_longlake_data()
```

---

north\_arrow\_orienteeing  
                          *North arrow styles*

---

### Description

North arrow styles

### Usage

```
north_arrow_orienteeing(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_orienteeing(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_orienteering())

grid::grid.newpage()
grid::grid.draw(north_arrow_fancy_orienteering())

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 <a href="#">aes</a> .
data	A data frame or other object, coerced to a data.frame by <a href="#">fortify</a> .
crs	The crs of the x and y aesthetics, or NULL to use default lon/lat crs.
geom	The geometry to use.
position	The position to use.
...	Passed to the base ggplot2 functions <a href="#">geom_point</a> , <a href="#">geom_path</a> , <a href="#">geom_polygon</a> , <a href="#">geom_text</a> , <a href="#">geom_label</a> , <a href="#">geom_text_repel</a> , and <a href="#">geom_label_repel</a> , respectively.
show.legend, inherit.aes	See <a href="#">layer</a> .

**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

## \*Topic **datasets**

- annotation\_map\_tile, 2
  - annotation\_north\_arrow, 3
  - annotation\_scale, 4
  - layer\_spatial.Raster, 8
- aes, 7, 8, 12
- annotation\_map\_tile, 2
- annotation\_north\_arrow, 3
- annotation\_scale, 4
- annotation\_spatial (layer\_spatial), 7
- annotation\_spatial.Raster  
(layer\_spatial.Raster), 8
- annotation\_spatial.stars  
(layer\_spatial.Raster), 8
- coord\_sf, 11
- df\_spatial, 6, 8
- fortify, 12
- geom\_label, 12
- geom\_label\_repel, 12
- geom\_path, 11, 12
- geom\_point, 11, 12
- geom\_polygon, 6, 11, 12
- geom\_polypath, 6
- geom\_raster, 8
- geom\_sf, 8
- geom\_spatial\_label  
(stat\_spatial\_identity), 11
- geom\_spatial\_label\_repel  
(stat\_spatial\_identity), 11
- geom\_spatial\_path  
(stat\_spatial\_identity), 11
- geom\_spatial\_point  
(stat\_spatial\_identity), 11
- geom\_spatial\_polygon  
(stat\_spatial\_identity), 11
- geom\_spatial\_text  
(stat\_spatial\_identity), 11
- geom\_spatial\_text\_repel  
(stat\_spatial\_identity), 11
- geom\_text, 12
- geom\_text\_repel, 12
- GeomMapTile (annotation\_map\_tile), 2
- GeomNorthArrow  
(annotation\_north\_arrow), 3
- GeomScaleBar (annotation\_scale), 4
- GeomSpatialRaster  
(layer\_spatial.Raster), 8
- layer, 8, 12
- layer\_spatial, 7
- layer\_spatial.Raster, 8
- layer\_spatial.stars  
(layer\_spatial.Raster), 8
- load\_longlake\_data, 10
- north\_arrow\_fancy\_orienteeing  
(north\_arrow\_orienteeing), 10
- north\_arrow\_minimal  
(north\_arrow\_orienteeing), 10
- north\_arrow\_nautical  
(north\_arrow\_orienteeing), 10
- north\_arrow\_orienteeing, 3, 10
- osm.image, 2
- st\_as\_sf, 8
- stat\_identity, 11
- stat\_spatial\_identity, 11
- StatSpatialRaster  
(layer\_spatial.Raster), 8
- StatSpatialRasterAnnotation  
(layer\_spatial.Raster), 8
- StatSpatialRasterDf  
(layer\_spatial.Raster), 8
- xy\_transform, 13