

# Package ‘ggspatial’

April 2, 2017

**Type** Package

**Title** Spatial Data Framework for ggplot2

**Version** 0.2.1

**Date** 2017-04-01

**Author** Dewey Dunnington <dewey@fishandwhistle.net>

**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 (>= 2.0), sp

**Imports** rosm (>= 0.2), abind, reshape2, methods, rgdal, maptools, plyr, raster, scales

**Suggests** prettymapr, mapproj

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

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

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-04-02 06:31:40 UTC

## R topics documented:

annotation_spraster . . . . .	2
as.CRS . . . . .	3
fortify.Raster . . . . .	3
geom_osm . . . . .	4
geom_spatial . . . . .	6
geom_spatial.Raster . . . . .	8

longlake_buildingsdf . . . . .	9
stat_project . . . . .	9
stat_rgba . . . . .	10
xyTransform . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

annotation\_spraster     *Annotation raster using Raster\* layers*

---

## Description

This is a thin wrapper around [annotation\\_raster](#), with the bounds arguments filled in using the bounds of the raster. Like [annotation\\_raster](#), this will not adjust the extents of the plot. The coordinates used are in the coordinate system of the raster, which are likely not lat/lon.

## Usage

```
annotation_spraster(raster, interpolate = FALSE, na.value = NA)
```

```
geom_spraster_rgb(raster, interpolate = FALSE, na.value = NA)
```

## Arguments

raster	A Raster* object
interpolate	TRUE to interpolate rendering
na.value	A value to represent NAs, since a transparency band may or may not exist for this raster. If na.value = NA, a transparency band will be created to remove missing values from display.

## Value

An annotation layer or list of layers (`geom_spraster_rgb`)

## Examples

```
ggplot() +
  annotation_spraster(longlake_osm) +
  geom_spatial(longlake_waterdf, toepsg = 26920) +
  coord_fixed()
```

---

as.CRS	<i>Extract a projection from an object</i>
--------	--

---

### Description

Many functions in this package require projection information. This function allows a CRS to be extracted from various objects, notably `Spatial*`, `Raster*`, and integers (which are assumed to be EPSG codes). This allows for less verbose syntax when dealing with projections in this package.

### Usage

```
as.CRS(x)
```

### Arguments

x                    An object

### Value

A CRS object, or NA if one cannot be extracted

### Examples

```
as.CRS(4326) # integer
as.CRS(longlake_osm) # raster
as.CRS(longlake_waterdf) # spatial
```

---

fortify.Raster	<i>Turn a Raster into a data.frame</i>
----------------	--

---

### Description

Like others in the `fortify` family, this method coerces its input into a `data.frame`. Each band in the raster is a column in the data frame, alongside x and y coordinate columns. This is the format required for input to `geom_raster`, such that a Raster object can be passed directly to `geom_raster` without a conversion function. Band columns are named `band1`, `band2`, `band3`, etc., for use in creating a mapping.

### Usage

```
## S3 method for class 'Raster'
fortify(model, data = NULL, format = c("wide", "long"),
  ...)
```

**Arguments**

model	A Raster object
data	Unused
format	Use 'long' to get values in a single column; otherwise values are in one column per band.
...	Not used by this method

**Value**

A data.frame with columns, x and y as coordinates in the projection of the Raster,

**Examples**

```
# use with ggplot()
df <- fortify(longlake_osm)
ggplot(df, aes(x, y, fill = band1)) + geom_raster() +
  coord_fixed()

# identical usage with ggraster()
ggraster(longlake_osm, aes(fill = band1))

# use long format to facet by band
dflong <- fortify(longlake_osm, format = "long")
ggplot(dflong, aes(x, y, fill = value)) +
  geom_raster() + facet_wrap(~band) +
  coord_fixed()

# can use on other raster types as well
x <- rosm::osm.raster("gatineau qc")
ggraster(x, aes(fill = band1))
df <- fortify(x, format = "long")
ggplot(df, aes(x, y, fill = value)) +
  geom_raster() + facet_wrap(~band) +
  coord_fixed()
```

**Description**

This geometry is a lazy version of its counterparts in the ggmap and rosm packages. All tile downloading/loading/drawing is done at plot draw, such that a complete backdrop of tiles can be calculated. The arguments are essentially a wrapper around [osm.image](#) and [annotation\\_raster](#) that can plot a specific bounding box or default to the extents of the plot. The `ggosm()` function is a shorthand for the common case of `ggplot() + geom_osm(...) + coord_map()`.

**Usage**

```
geom_osm(x = NULL, zoomin = 0, zoom = NULL, type = NULL,
         forcedownload = FALSE, cachedir = NULL, progress = c("text", "none"),
         quiet = TRUE)
```

```
ggosm(x = NULL, zoomin = 0, zoom = NULL, type = NULL,
      forcedownload = FALSE, cachedir = NULL, progress = c("text", "none"),
      quiet = TRUE)
```

**Arguments**

x	An object that can be coerced to a bounding box using <a href="#">extract_bbox</a> , or NULL to use the plot extents to fetch tiles (probably what you want).
zoomin	The amount by which to adjust the automatically calculated zoom (or manually specified if the zoom parameter is passed). Use +1 to zoom in, or -1 to zoom out.
zoom	Manually specify the zoom level (not recommended; adjust zoomin instead).
type	A map type; one of that returned by <a href="#">osm.types</a> . User defined types are possible by passing any object coercible to type <a href="#">tile_source</a> (see <a href="#">as.tile_source</a> ).
forcedownload	TRUE if cached tiles should be re-downloaded. Useful if some tiles are corrupted.
cachedir	The directory in which tiles should be cached. Defaults to <code>getwd()/rosm.cache</code> .
progress	A progress bar to use, or "none" to suppress progress updates
quiet	Pass FALSE to see more error messages, particularly if your tiles do not download/load properly.

**Value**

A ggplot2 layer object

**Examples**

```
library(prettypmapr)
# use as a backdrop for geographical data
cities <- geocode(c("Halifax, NS", "Moncton, NB", "Montreal QC"))
ggplot(cities, aes(lon, lat, shape = query)) + geom_osm() +
  geom_point() + coord_map()

# use ggosm() shorthand
ggosm() + geom_point(aes(lon, lat), cities)

# use on its own with a bounding box
ggosm(searchbbox("vermont, USA"))

# use alternative map types (see rosm::osm.types())
ggosm(type = "stamenwatercolor") + geom_point(aes(lon, lat), cities)
```

---

 geom\_spatial

 A ggplot2 geom for Spatial\* objects
 

---

### Description

A function returning a geom\_\* object based on the Spatial\* input. Also will happily project a regular data.frame provided x and y aesthetics are specified. The result is a geom\_\* for use with ggplot2, with aesthetics and other arguments passed on to that geom.

### Usage

```
geom_spatial(data, ...)

ggspatial(data, mapping = NULL, ...)

## Default S3 method:
geom_spatial(data, mapping = NULL, show.legend = TRUE,
  inherit.aes = NULL, position = "identity", crsfrom = NA, crsto = NA,
  geom = "point", ...)

## S3 method for class 'SpatialPoints'
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, attribute_table = NULL, ...)

## S3 method for class 'SpatialPointsDataFrame'
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, ...)

## S3 method for class 'SpatialLines'
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, attribute_table = NULL, ...)

## S3 method for class 'SpatialLinesDataFrame'
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, ...)

## S3 method for class 'SpatialPolygons'
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, rule = "winding", attribute_table = NULL,
  ...)

## S3 method for class 'SpatialPolygonsDataFrame'
```

```
geom_spatial(data, mapping = NULL,
  show.legend = TRUE, inherit.aes = NULL, position = "identity",
  crsfrom = NA, crsto = NA, rule = "winding", ...)
```

### Arguments

data	A Spatial* object or data.frame.
...	Arguments passed on to the geom_* (e.g. lwd, fill, etc.)
mapping	A mapping as created by aes() or aes_string()
show.legend	Logical describing the legend visibility.
inherit.aes	Logical describing if aesthetics are inherited
position	Passed on to geom_*
crsfrom	An object that can be coerced to a CRS using <a href="#">as.CRS</a> ; defaults to the CRS of the data or lat/lon if that does not exist
crsto	An object that can be coerced to a CRS using <a href="#">as.CRS</a> ; defaults to lat/lon so that the plot can be projected using coord_map()
geom	For data frames, the geometry to use
attribute_table	For SpatialPoints, SpatialLines, and SpatialPolygons, an attribute table that matches the input object.
rule	One of 'evenodd' or 'winding', if the Spatial object is a polygon layer.

### Value

A ggplot2 'layer' object.

### Examples

```
library(prettymapr)
ns <- searchbbox("Nova Scotia")
cities <- geocode(c("Wolfville, NS", "Windsor, NS", "Halifax, NS"))
ggplot(cities, aes(x=lon, y=lat)) + geom_spatial(crsto=26920)
# default projection is Spherical Mercator (EPSG:3857)
ggplot(cities, aes(x=lon, y=lat)) + geom_spatial() + coord_map()

# plot a number of spatial objects
ggplot() +
  geom_spatial(longlake_waterdf, fill="lightblue") +
  geom_spatial(longlake_marshdf, fill="grey", alpha=0.5) +
  geom_spatial(longlake_streamsdf, col="lightblue") +
  geom_spatial(longlake_road sdf, col="black") +
  geom_spatial(longlake_buildingsdf, pch=1, col="brown", size=0.25) +
  coord_map()
```

---

geom\_spatial.Raster     *Spatial Geometry for Raster\* Objects*

---

### Description

Similar to the [geom\\_spatial](#) family for Spatial\* objects, this method plots a spatial raster as a spatial raster. Note that projecting (or un-projecting) this layer will likely result in odd results (namely that geom\_raster will not function properly or at all). As a convenience, ggraster() is provided, which replaces the call to coord\_map() from [ggspatial](#) with a call to coord\_fixed().

### Usage

```
## S3 method for class 'Raster'
geom_spatial(data, mapping = NULL, show.legend = TRUE,
  inherit.aes = FALSE, position = "identity", crsfrom = crsfrom,
  crsto = crsto, geom = "raster", stat = "identity", ...)

ggraster(data, mapping = NULL, ...)
```

### Arguments

data	A Raster* object
mapping	A mapping
show.legend	Should the legend be shown for this layer?
inherit.aes	Should aesthetics be inherited from the base plot?
position	The position to apply (should probably always be 'identity')
crsfrom	Override the source projection
crsto	Override the source projection
geom	The geometry to use. Defaults to raster (obviously), but could also be another value if used with a different stat (e.g. contour)
stat	The stat to apply. Defaults to 'identity', but could be something else like 'contour', <a href="#">stat_rgba</a> ("rgba"), or <a href="#">stat_project</a> ("project").
...	Further arguments passed to the stat/geom

### Value

A ggplot2 layer

### Examples

```
# standard ggplot syntax
ggplot() + geom_spatial(longlake_osm, aes(fill = band1)) + coord_fixed()

# or use ggraster()
ggraster(longlake_osm, aes(fill = band1))
```

---

longlake\_buildingsdf    *Long Lake Spatial\* objects*

---

**Description**

A group of Spatial\* objects used to test [geom\\_spatial](#).

**Usage**

longlake\_buildingsdf

longlake\_buildingsdf

longlake\_depthdf

longlake\_marshdf

longlake\_osm

longlake\_road sdf

longlake\_stream sdf

longlake\_waterdf

**Format**

An object of class SpatialPointsDataFrame with 14 rows and 16 columns.

**Source**

The Nova Scotia Topographic Database (<http://www.mapsnovascotia.com/category.aspx?ic=24>) and Open Street Map (<http://www.openstreetmap.org>).

---

stat\_project                    *Statisitc to project coordinates*

---

**Description**

Projects coordinates using rgdal/sp. Takes params crsfrom and crsto, both wrapped in [as.CRS](#), such that you can pass an epsg code, a CRS object or NA to guess the input (will be either lat/lon or google mercator). If NA, crsto is assumed to be EPSG:4326 (Lat/Lon), so data can be used with [coord\\_map](#).

**Usage**

```
stat_project(mapping = NULL, data = NULL, crsfrom = NA, crsto = NA,
  position = "identity", show.legend = TRUE, inherit.aes = TRUE,
  geom = "point", ...)
```

**Arguments**

mapping	A mapping as created by <code>aes()</code> or <code>aes_string()</code>
data	A <code>Spatial*</code> object or <code>data.frame</code> .
crsfrom	An object that can be coerced to a CRS using <a href="#">as.CRS</a> ; defaults to the CRS of the data or lat/lon if that does not exist
crsto	An object that can be coerced to a CRS using <a href="#">as.CRS</a> ; defaults to lat/lon so that the plot can be projected using <code>coord_map()</code>
position	Passed on to <code>geom_*</code>
show.legend	Logical describing the legend visibility.
inherit.aes	Logical describing if aesthetics are inherited
geom	For data frames, the geometry to use
...	Passed to the geom

**Examples**

```
# longlake roads df is in UTM zone 20 (epsg:26920)
ggplot(longlake_road sdf, aes(long, lat)) +
  stat_project(geom = "path", crsfrom = 26920) +
  coord_map()
```

---

stat\_rgba

*Statistic to create RGB fill values*


---

**Description**

Usually used on conjunction with [geom\\_raster](#) or [geom\\_spatial.Raster](#). This function appears to work, however it is slow, and in general, only good for very small datasets.

**Usage**

```
stat_rgba(mapping = NULL, data = NULL, ..., limits_red = NA,
  limits_green = NA, limits_blue = NA, limits_alpha = NA)
```

**Arguments**

mapping	A mapping created with <a href="#">aes</a>
data	A data.frame
...	Passed to <a href="#">geom_raster</a>
limits_red	Data limits from which to scale red values. Use NULL to perform no transformation, or NA to use the data value without transformation.
limits_green	Data limits from which to scale green values
limits_blue	Data limits from which to scale blue values
limits_alpha	Data limits from which to scale alpha values

**Value**

A ggplot2 layer object

**Examples**

```
# using stat_rgba()
ggplot(longlake_osm, aes(x, y)) +
  stat_rgba(aes(red = band1, green = band2, blue = band3, alpha = 1),
            limits_red = NULL, limits_green = NULL, limits_blue = NULL,
            limits_alpha = NULL, interpolate = TRUE) +
  coord_fixed()

# using ggraster() with stat = 'rgba'
ggraster(longlake_osm, aes(red = band1, green = band2, blue = band3, alpha = 1),
         stat = "rgba")
```

---

xyTransform

*Project XY coordinates*


---

**Description**

The `sp` package provides a powerful interface with easy syntax for projection `Spatial*` objects, but raw coordinates are not as straightforward. Use this function to project raw coordinates, and [spTransform](#) to project `Spatial*` objects.

**Usage**

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

```
bboxTransform(bbox, from = 4326, to = 4326)
```

**Arguments**

x	The x values (or longitude)
y	The y values (or latitude)
from	The source projection, or an object that can be coerced to one using <a href="#">as.CRS</a>
to	The destination projection, or an object that can be coerced to one using <a href="#">as.CRS</a>
na.rm	Currently xyTransform does not work with non-finite values. Pass na.rm = TRUE to remove them, or else a (more helpful) error will be thrown if non-finite values exist.
bbox	The bounding box to transform. Note that bounding boxes are not truly transformed bounding boxes, but the bounding box of the transformed lower-left and upper-right coordinates. This is a perfect approximation in cylindrical systems but questionable in more complex systems.

**Value**

A matrix with 2 columns (x and y)

**Examples**

```
all_latlons <- expand.grid(x=-180:180, y=-70:70)
xyTransform(all_latlons$x, all_latlons$y, from = 4326, to = 3857)
bboxTransform(bbox(longlake_osm), from = 26920)
```

# Index

## \*Topic **datasets**

- longlake\_buildingsdf, 9
- osm.image, 4
- osm.types, 5
- spTransform, 11
- stat\_project, 8, 9
- stat\_rgba, 8, 10
- xyTransform, 11

aes, 11

annotation\_raster, 2, 4

annotation\_spraster, 2

as.CRS, 3, 7, 9, 10, 12

as.tile\_source, 5

bboxTransform (xyTransform), 11

coord\_map, 9

extract\_bbox, 5

fortify, 3

fortify.Raster, 3

geom\_osm, 4

geom\_raster, 3, 10, 11

geom\_spatial, 6, 8, 9

geom\_spatial.Raster, 8, 10

geom\_spraster\_rgb  
(annotation\_spraster), 2

ggosm (geom\_osm), 4

ggraster (geom\_spatial.Raster), 8

ggspatial, 8

ggspatial (geom\_spatial), 6

longlake\_buildingsdf, 9

longlake\_depthdf  
(longlake\_buildingsdf), 9

longlake\_marshdf  
(longlake\_buildingsdf), 9

longlake\_osm (longlake\_buildingsdf), 9

longlake\_road sdf  
(longlake\_buildingsdf), 9

longlake\_stream sdf  
(longlake\_buildingsdf), 9

longlake\_waterdf  
(longlake\_buildingsdf), 9