Package 'ggdibbler'

July 31, 2025

Type Package

Title Add Uncertainty to Data Visualisations

Version 0.1.0

Maintainer Harriet Mason <harriet.m.mason@gmail.com>

Description A 'ggplot2' extension for visualising uncertainty with the goal of signal suppression. Usually, uncertainty visualisation focuses on expressing uncertainty as a distribution or probability, whereas 'ggdibbler' differentiates itself by viewing an uncertainty visualisation as an adjustment to an existing graphic that incorporates the inherent uncertainty in the estimates. You provide the code for an existing plot, but replace one of the variables with a vector of distributions, and it will convert the visualisation into it's signal suppression counterpart.

License GPL-3

URL https://harriet-mason.github.io/ggdibbler/

Depends R (>= 4.1.0)

Imports distributional, dplyr, ggplot2, rlang, sf

Suggests knitr, rmarkdown, testthat (>= 3.0.0), vdiffr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Harriet Mason [aut, cre] (ORCID: <https://orcid.org/0009-0007-4568-8215>), Dianne Cook [aut] (ORCID: <https://orcid.org/0000-0002-3813-7155>), Sarah Goodwin [aut] (ORCID: <https://orcid.org/0000-0001-8894-8282>), Susan VanderPlas [aut] (ORCID: <https://orcid.org/0000-0002-3803-0972>)

Repository CRAN

Date/Publication 2025-07-31 10:00:31 UTC

Contents

`geom_sf_sample` *Visualise Sf Objectjects with Uncertainty*

Description

Identical to geom_sf, except that the fill for each area will be a distribution. This function will replace the fill area with a grid, where each cell is filled with an outcome from the fill distribution.

Usage

```
geom_sf_sample(
  mapping = aes(),
  data = NULL,
  stat = "sample",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  n = NULL,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

`stat` The statistical transformation to use on the data for this layer. When using a `geom_*()` function to construct a layer, the `stat` argument can be used the override the default coupling between geoms and stats. The `stat` argument accepts the following:

- A `Stat` ggproto subclass, for example `StatCount`.
- A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`.
- For more information and other ways to specify the stat, see the layer stat documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes.

You can also set this to one of "polygon", "line", and "point" to override the default legend.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.

`n` A parameter used to control the number of cells in each grid. Each area is broken up into an nxn grid

`...` Other arguments passed on to `layer()`'s `params` argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the `position` argument, or aesthetics that are required can *not* be passed through `...`. Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the `geom` part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.

- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `...`. This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

Value

A ggplot2 geom representing a sf_sample which can be added to a ggplot object

Examples

```
# In it's most basic form, the geom will make a subdivision
library(ggplot2)
library(dplyr)
basic_data <- toy_temp_dist |>
dplyr::filter(county_name %in% c("Pottawattamie County", "Mills County", "Cass County"))
basic_data |>
  ggplot() +
  geom_sf_sample(aes(geometry = county_geometry, fill=temp_dist))
# The original borders of the sf object can be hard to see,
 # so layering the original geometry on top can help to see the original boundaries
basic_data |>
  ggplot() +
  geom_sf_sample(aes(geometry = county_geometry, fill=temp_dist), linewidth=0.1, n=4) +
  geom_sf(aes(geometry=county_geometry), fill=NA, linewidth=1)
```

---

scale_type.distribution

Sets scale for distributions

Description

Generates a single value from the distribution and uses it to set the default ggplot scale. The scale can be changed later in the ggplot by using any scale_* function

Usage

```
## S3 method for class 'distribution'
scale_type(x)
```

Arguments

x	value being scaled

Value

A character vector of scale types. The scale type is the ggplot scale type of the outcome of the distribution.

`toy_temp` *A toy data set that has the ambient temperature as measured by a collection of citizen scientists for each Iowa county*

Description

There are several measurements for each county, with no location marker for individual scientists to preserve anonyminity. Counties can have different numbers of observations as well as a different levels of variance between the observations in the county.

Format

A tibble with 99 observations and 4 variables

county_name the name of each Iowa county

recorded_temp the ambient temperature recorded by the citizen scientist

scientistID the ID number for the scientist who made the recording

county_geometry the shape file for each county of Iowa

county_longitude the centroid longitude for each county of Iowa

county_latitude the centroid latitude for each county of Iowa

---

`toy_temp_dist` *A toy data set that provides data for a map with the temperature of each area represented by a random variable.*

Description

The map shows a wave pattern in temperature on the state of Iowa. Each estimate also has an uncertainty component added, and is represented as a distribution

Format

A tibble with 99 observations and 4 variables

county_name the name of each Iowa county

temp_dist the temperature of each county as a distribution

county_geometry the shape file for each county of Iowa

Index