Package ‘ggplot.multistats’

October 28, 2019

Title  Multiple Summary Statistics for Binned Stats/Geometries
Version  1.0.0
Description  Provides the ggplot binning layer stat_summaries_hex(),
            which functions similar to its singular form,
            but allows the use of multiple statistics per bin.
            Those statistics can be mapped to multiple bin aesthetics.

URL  https://github.com/flying-sheep/ggplot.multistats

BugReports  https://github.com/flying-sheep/ggplot.multistats/issues
License  GPL-3
Encoding  UTF-8
LazyData  true
Imports  methods, rlang, scales, hexbin, ggplot2
RoxygenNote  6.1.1
NeedsCompilation  no

Author  Philipp Angerer [aut, cre] (<https://orcid.org/0000-0002-0369-2888>)
Maintainer  Philipp Angerer <phil.angerer@gmail.com>
Repository  CRAN
Date/Publication  2019-10-28 13:50:05 UTC

R topics documented:

  draw_key_hexagon .................................................. 2
  normalize_function_list ........................................... 2
  stat_summaries_hex .............................................. 3

Index  5
draw_key_hexagon  

Draw a Hexagon

Description
The default legend key drawing function for `stat_summaries_hex`. This function can be used as key_glyph parameter by any layer.

Usage
```
draw_key_hexagon(data, params, size)
```

Arguments
- `data`: A single row data frame containing the scaled aesthetics to display in this key
- `params`: A list of additional parameters supplied to the geom.
- `size`: Width and height of key in mm.

Value
A hexagonal `polygonGrob`

See Also
The legend key drawing functions built into ggplot: `draw_key`.

Examples
```
library(ggplot2)
ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
  geom_hex(key_glyph = 'hexagon') +
  guides(fill = 'legend')
```

normalize_function_list  

Normalize a List of Functions

Description
Takes a list of functions and function names (or a vector of function names) and names it. Requires all entries with functions to be named and adds names to functions that were specified as names.

Usage
```
normalize_function_list(funs)
```
**stat_summaries_hex**

**Arguments**

- **funs**
  Valid list or vector of function names and/or functions.

**Value**

Named list or character vector of functions.

**Examples**

```r
normalize_function_list(c(value = 'mean'))
normalize_function_list(c('median', n = 'length'))
normalize_function_list(list('median', n = length))
normalize_function_list(list(Sum = sum, Custom = function(x) sum(nchar(as.character(x)))))
```

---

**stat_summaries_hex**   *Multi-Stat Binning Layer*

**Description**

Very similar to `stat_summary_hex`, but allows for multiple stats to be captured using the `funs` parameter.

**Usage**

```r
stat_summaries_hex(mapping = NULL, data = NULL, geom = "hex",
  position = "identity", ..., bins = 30, binwidth = NULL,
  drop = TRUE, funs = c(value = "mean"), na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, key_glyph = NULL)
```

StatSummariesHex

**Arguments**

- **mapping**
  Set of aesthetic mappings created by `aes()` or `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_summaries_hex

geom  The geometric object to use display the data
position Position adjustment, either as a string, or the result of a call to a position adjustment function.
... Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

bins numeric vector giving number of bins in both vertical and horizontal directions. Set to 30 by default.

binwidth Numeric vector giving bin width in both vertical and horizontal directions. Overrides bins if both set.
drop drop if the output of fun is NA.
funs A list or vector of functions and function names. See normalize_function_list for details.
n.a.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. It can also be a named logical vector to finely select the aesthetics to display.

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().

key_glyph A legend key drawing function or a string providing the function name minus the draw_key_ prefix. The default is draw_key_hexagon.

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

See Also
normalize_function_list for the funs parameter and draw_key_hexagon for the legend entry.

Examples
library(ggplot2)
# Define the variable used for the stats using z
ggplot_base <- ggplot(iris, aes(Sepal.Width, Sepal.Length, z = Petal.Width))
# The default is creating 'stat(value)' containing the mean
ggplot_base + stat_summaries_hex(aes(fill = stat(value)), bins = 5)
# but you can specify your own stats
ggplot_base + stat_summaries_hex(
  aes(fill = stat(median), alpha = stat(n)),
  funs = c('median', n = 'length'),
  bins = 5)
Index

*Topic `datasets`
  stat_summaries_hex, 3

aes(), 3
aes_(), 3
borders(), 4
draw_key, 2
draw_key_hexagon, 2, 4
fortify(), 3
ggplot(), 3
layer(), 4
normalize_function_list, 2, 4
polygonGrob, 2
stat_summaries_hex, 2, 3
stat_summary_hex, 3
StatSummariesHex(stat_summaries_hex), 3