

Package ‘ggquiver’

December 18, 2025

Version 0.4.0

Title Quiver Plots for 'ggplot2'

Description An extension of 'ggplot2' to provide quiver plots to visualise vector fields. This functionality is implemented using a geom to produce a new graphical layer, which allows aesthetic options. This layer can be overlaid on a map to improve visualisation of mapped data.

Depends R (>= 3.2.0)

Imports ggplot2, grid

Suggests maps, sf, pkgdown, vdiff, testthat

URL <https://github.com/mitchelloharawild/ggquiver>,
<https://pkg.mitchelloharawild.com/ggquiver/>

BugReports <https://github.com/mitchelloharawild/ggquiver/issues>

License GPL-3

Encoding UTF-8

ByteCompile true

RoxygenNote 7.3.3

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre]

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2025-12-18 12:20:11 UTC

Contents

geom_quiver	2
Index	6

`geom_quiver`*Quiver plots for ggplot2*

Description

Displays the direction and length of vectors on a graph.

Usage

```
geom_quiver(  
  mapping = NULL,  
  data = NULL,  
  stat = "quiver",  
  position = "identity",  
  center = FALSE,  
  rescale = FALSE,  
  vecsize = NULL,  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  ...  
)
```

GeomQuiver

```
stat_quiver(  
  mapping = NULL,  
  data = NULL,  
  geom = "quiver",  
  position = "identity",  
  center = FALSE,  
  rescale = FALSE,  
  vecsize = NULL,  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  ...  
)
```

StatQuiver

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	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>
stat	<p>The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following:</p> <ul style="list-style-type: none"> • A Stat ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	<p>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 <code>position</code> argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as <code>position_jitter()</code>. 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 <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
center	<p>If <code>FALSE</code> (the default), the vector lines will start at the specified <code>x</code> and <code>y</code>. If <code>TRUE</code>, the arrows will be centered about <code>x</code> and <code>y</code>.</p>
rescale	<p>If <code>FALSE</code> (the default), the vectors will not be rescaled. If <code>TRUE</code>, the vectors given by <code>(u, v)</code> will be rescaled using the <code>scale</code> function.</p>
vecsize	<p>By default (<code>NULL</code>), vectors sizing is automatically determined. If a grid can be identified, they will be scaled to the grid, if not, the vectors will not be scaled. By specifying a numeric input here, the length of all arrows can be adjusted. Setting <code>vecsize</code> to zero will prevent scaling the arrows.</p>
na.rm	<p>If <code>FALSE</code> (the default), removes missing values with a warning. If <code>TRUE</code> silently removes missing values.</p>
show.legend	<p>logical. Should this layer be included in the legends? <code>NA</code>, the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use <code>TRUE</code>. If <code>NA</code>, all levels are shown in legend, but unobserved levels are omitted.</p>

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. annotation_borders() .
...	Other arguments passed on to layer() 's <code>params</code> argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the <code>position</code> argument, or aesthetics that are required can <i>not</i> be passed through ... Unknown arguments that are not part of the 4 categories below are ignored. <ul style="list-style-type: none"> • 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, <code>colour = "red"</code> or <code>linewidth = 3</code>. The geom's documentation has an Aesthetics section that lists the available options. The 'required' aesthetics cannot be passed on to the <code>params</code>. 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 <code>stat_*()</code> function, the ... argument can be used to pass on parameters to the geom part of the layer. An example of this is <code>stat_density(geom = "area", outline.type = "both")</code>. The geom's documentation lists which parameters it can accept. • Inversely, when constructing a layer using a <code>geom_*()</code> function, the ... argument can be used to pass on parameters to the stat part of the layer. An example of this is <code>geom_area(stat = "density", adjust = 0.5)</code>. The stat's documentation lists which parameters it can accept. • The <code>key_glyph</code> 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.
geom	The geometric object to use to display the data for this layer. When using a <code>stat_*()</code> function to construct a layer, the <code>geom</code> argument can be used to override the default coupling between stats and geoms. The <code>geom</code> argument accepts the following: <ul style="list-style-type: none"> • A Geom ggproto subclass, for example <code>GeomPoint</code>. • A string naming the geom. To give the geom as a string, strip the function name of the <code>geom_</code> prefix. For example, to use <code>geom_point()</code>, give the geom as "point". • For more information and other ways to specify the geom, see the layer geom documentation.

Format

An object of class `GeomQuiver` (inherits from `GeomSegment`, `Geom`, `ggproto`, `gg`) of length 2.

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

Computed variables

x centered x start position for velocity arrow

y centered y start position for velocity arrow

xend centered x end position for velocity arrow

yend centered y end position for velocity arrow

Examples

```
library(ggplot2)
# Quiver plots of mathematical functions
field <- expand.grid(x=seq(0,pi,pi/12), y=seq(0,pi,pi/12))
ggplot(field, aes(x=x,y=y,u=cos(x),v=sin(y))) +
  geom_quiver()

# Removing automatic scaling
ggplot(seals, aes(x=long, y=lat, u=delta_long, v=delta_lat)) +
  geom_quiver(vecsize=NULL) +
  borders("state")
```

Index

* datasets

geom_quiver, 2

aes(), 2

annotation_borders(), 4

fortify(), 3

geom_quiver, 2

GeomQuiver (geom_quiver), 2

ggplot(), 3

key glyphs, 4

layer geom, 4

layer position, 3

layer stat, 3

layer(), 4

stat_quiver (geom_quiver), 2

StatQuiver (geom_quiver), 2