Package 'treemapify'

October 14, 2022

Title Draw Treemaps in 'ggplot2'

Version 2.5.5

URL https://wilkox.org/treemapify/

BugReports https://github.com/wilkox/treemapify/issues/

Description Provides 'ggplot2' geoms for drawing treemaps.

Depends R (>= 3.6)

Imports ggplot2 (>= 2.2.1), ggfittext (>= 0.5.0), grid (>= 3.1)

Suggests knitr, rmarkdown, testthat, vdiffr, spelling

License GPL (>= 3)

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

NeedsCompilation no

Author David Wilkins [aut, cre],
Bob Rudis [ctb] (<https://orcid.org/0000-0001-5670-2640>)

Maintainer David Wilkins <david@wilkox.org>

Repository CRAN

Date/Publication 2021-01-08 09:50:03 UTC

R topics documented:

`draw_key_rrect` *Round rect key glyph for legend*

Description

Round rect key glyph for legend

Usage

```
draw_key_rrect(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.

Author(s)

Bob Rudis (bob@rud.is)

`G20` *Statistics on the G-20 group of major world economies.*

Description

A dataset containing economic and demographic statistics about members of the G-20 group of major world economies.

Usage

```
G20
```

Format

A data frame with 20 rows and five variables:

region the country's region

country the country

gdp_mil_usd the country's GDP, in millions of US dollars

hdi the country's Human Development Index

econ_classification the country's economic classification

hemisphere the hemisphere in which the majority of the country's landmass lies

Source

https://en.wikipedia.org/wiki/G-20_major_economies

`geom_treemap` *A 'ggplot2' geom to draw a treemap.*

Description

A treemap is a rectangular plot divided into tiles, each of which represents a single observation. The relative area of each tile expresses a continuous variable.

Usage

```
geom_treemap(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  fixed = NULL,
  layout = "squarified",
  start = "bottomleft",
  radius = grid::unit(0, "pt"),
  ...
)
```

Arguments

`mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...`	Standard geom arguments as for `ggplot2::geom_rect()`.
`fixed`	Deprecated. Use `layout = "fixed"` instead. Will be removed in later versions.
`layout`	The layout algorithm, one of either 'squarified' (the default), 'scol', 'srow' or 'fixed'. See Details for full details on the different layout algorithms.
`start`	The corner in which to start placing the tiles. One of 'bottomleft' (the default), 'topleft', 'topright' or 'bottomright'.
`radius`	corner radius (default 0pt)

Details

`geom_treemap()` requires an `area` aesthetic. It will ignore any aesthetics relating to the x and y axes (e.g. `xmin` or `y`), as the x and y axes are not meaningful in a treemap. Several other standard 'ggplot2' aesthetics are supported (see Aesthetics). To add text labels to tiles, see `geom_treemap_text()`.

An optional `subgroup` aesthetic will cause the tiles to be clustered in subgroups within the treemap. See `geom_treemap_subgroup_border()` and `geom_treemap_subgroup_text()` to draw borders

around subgroups and label them, respectively. Up to three nested levels of subgrouping are supported, with `subgroup2` and `subgroup3` aesthetics and respective `geom_treemap_subgroup2_border()` etc. geoms.

Four layout algorithms are provided. With the default 'squarified' algorithm (`layout = "squarified"`), the priority is ensuring the tiles have an aesthetically pleasing aspect ratio; that is, they are not too narrow or too short. In this algorithm, tile placement proceeds from one corner, placing the tiles in either rows or columns until all the tiles are placed. See Bruls et al. (1999) for the full algorithm.

There are two variants on the 'squarified' algorithm. 'scol' forces tile placement to begin with a column, regardless of the effect on aspect ratio; 'srow' forces tile placement to been with a row. This will also apply to all subgroups. After the first row or column, the remaining tiles will be placed so as to optimise aspect ratios, as with the default algorithm.

With the 'fixed' layout algorithm (`layout = "fixed"`), the plot area is divided into vertical columns, which are each filled with an equal number of tiles beginning at the starting corner. Unlike the 'squarified' algorithm, with the 'fixed' algorithm the relative positions of the tiles are fixed by their order in the input data frame. This can result in aesthetically unpleasing layouts, but it allows side-by-side comparisons or animations to be created.

All 'treemapify' geoms added to a plot should have the same value for `layout` and `start`, or they will not share a common layout.

Aesthetics

- area (required)
- alpha
- colour
- fill
- linetype
- subgroup
- subgroup2
- subgroup3

Author(s)

David Wilkins (david@wilkox.org)

Bob Rudis (bob@rud.is)

References

Bruls, M., Huizing, K., & van Wijk, J. (1999). Squarified Treemaps (pp. 33-42). Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization. http://www.win.tue.nl/~vanwijk/stm.pdf

See Also

`geom_treemap_text()`, `geom_treemap_subgroup_border()`, `geom_treemap_subgroup_text()`

Examples

```
ggplot2::ggplot(G20, ggplot2::aes(area = gdp_mil_usd, fill = region)) +
 geom_treemap()
```

`geom_treemap_subgroup_border` *'ggplot2' geoms to draw a border around a subgroup of treemap tiles.*

Description

When `geom_treemap()` is used with a `subgroup`, `subgroup2` or `subgroup3` aesthetic to subgroup treemap tiles, `geom_treemap_subgroup_border`, `geom_treemap_subgroup2_border()` or `geom_treemap_subgroup3_bor` can be used to draw a border around each subgroup at the appropriate level.

Usage

```
geom_treemap_subgroup_border(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  fixed = NULL,
  layout = "squarified",
  start = "bottomleft",
  level = "subgroup",
  ...
)

geom_treemap_subgroup2_border(...)

geom_treemap_subgroup3_border(...)
```

Arguments

`mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...`	Standard geom arguments as for `ggplot2::geom_rect()`.
`fixed`	Deprecated. Use `layout = 'fixed'` instead. Will be removed in later versions.
`layout`	The layout algorithm, one of either 'squarified' (the default), 'scol', 'srow' or 'fixed'. See Details for full details on the different layout algorithms.
`start`	The corner in which to start placing the tiles. One of 'bottomleft' (the default), 'topleft', 'topright' or 'bottomright'.

`level` One of 'subgroup', 'subgroup2' or 'subgroup3', giving the subgrouping level for which to draw borders. It is recommended to use the aliases `geom_treemap_subgroup2_border()` and `geom_treemap_subgroup3_border()` instead of this argument.

Details

`geom_treemap_subgroup_border()` geoms require `area` and `subgroup` (or `subgroup2`, `subgroup3`) aesthetics. Several other standard 'ggplot2' aesthetics are supported (see Aesthetics).

Note that 'ggplot2' draws plot layers in the order they are added to the plot. This means that if you add a `geom_treemap_subgroup_border()` layer followed by a `geom_treemap_subgroup2_border()` layer, the second layer will be drawn on top of the first and may hide it.

The `layout` argument is used to set the treemap layout algorithm. All 'treemapify' geoms added to a plot should have the same value for `layout` and `start`, or they will not share a common layout (see `geom_treemap()` for details on the layout algorithms).

Aesthetics

- area (required)
- subgroup, subgroup2 or subgroup3 (required)
- colour
- size
- linetype
- alpha

See Also

`geom_treemap()`, `geom_treemap_subgroup_text()`

Examples

```
ggplot2::ggplot(G20, ggplot2::aes(area = gdp_mil_usd, fill = hdi,
                                  subgroup = hemisphere, subgroup2 = region)) +
  geom_treemap() +
  geom_treemap_subgroup2_border(colour = "white") +
  geom_treemap_subgroup_border()
```

---

geom_treemap_subgroup_text

'ggplot2' geoms to add text labels to treemap subgroups.

Description

When `geom_treemap()` is used with the `subgroup`, `subgroup2` or `subgroup3` aesthetic to subgroup treemap tiles, `geom_treemap_subgroup_text()`, `geom_treemap_subgroup2_text()` or `geom_treemap_subgroup3_text(` can be used to add a text label to each subgroup at the appropriate level.

Usage

```
geom_treemap_subgroup_text(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  padding.x = grid::unit(1, "mm"),
  padding.y = grid::unit(1, "mm"),
  place = "bottom",
  min.size = 4,
  grow = FALSE,
  reflow = FALSE,
  fixed = NULL,
  layout = "squarified",
  start = "bottomleft",
  level = "subgroup",
  ...
)

geom_treemap_subgroup2_text(...)

geom_treemap_subgroup3_text(...)
```

Arguments

`mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...`	Standard geom arguments as for `ggplot2::geom_text`.
`padding.x, padding.y`	`grid::unit()` object, giving horizontal or vertical padding between text and edge of tile. Defaults to 1 mm.
`place`	Where inside the box to place the text. Default is `bottom`; other options are `topleft`, `top`, `topright`, etc.
`min.size`	Minimum font size, in points. If provided, text that would need to be shrunk below this size to fit the box will not be drawn. Defaults to 4 pt.
`grow`	If `TRUE`, text will be grown as well as shrunk to fill the box.
`reflow`	If `TRUE`, text will be reflowed (wrapped) to better fit the box.
`fixed`	Deprecated. Use `layout = "fixed"` instead. Will be removed in later versions.
`layout`	The layout algorithm, one of either 'squarified' (the default), 'scol', 'srow' or 'fixed'. See Details for full details on the different layout algorithms.
`start`	The corner in which to start placing the tiles. One of 'bottomleft' (the default), 'topleft', 'topright' or 'bottomright'.
`level`	One of 'subgroup', 'subgroup2' or 'subgroup3', giving the subgrouping level for which to draw text labels. It is recommended to use the aliases `geom_treemap_subgroup2_text()` and `geom_treemap_subgroup3_text()` instead of this argument.

Details

geom_treemap_subgroup_text() geoms require area, label and subgroup (or subgroup2, subgroup3) aesthetics. Several other standard 'ggplot2' aesthetics are supported (see Aesthetics).

geom_treemap_subgroup_text() geoms use the 'ggfittext' package to fit text to the subgroup. All text drawing options available in ggfittext::geom_fit_text() (growing, reflowing, etc.) are also available here. For full details on how these options work, see the documentation for ggfittext::geom_fit_text().

The layout argument is used to set the treemap layout algorithm. All 'treemapify' geoms added to a plot should have the same value for layout and start, or they will not share a common layout (see geom_treemap() for details on the layout algorithms).

Aesthetics

- area (required)
- subgroup, subgroup2 or subgroup3 (required; the value of this variable will be the text label)
- colour
- size
- alpha
- family
- fontface
- angle

See Also

geom_treemap(), geom_treemap_subgroup_border()

Examples

```
ggplot2::ggplot(G20, ggplot2::aes(area = gdp_mil_usd, fill = hdi,
                                  subgroup = hemisphere, subgroup2 = region)) +
  geom_treemap() +
  geom_treemap_subgroup_text(place = "centre", grow = TRUE, alpha = 0.5) +
  geom_treemap_subgroup2_text()
```

---

geom_treemap_text *A 'ggplot2' geom to add text labels to treemap tiles.*

Description

geom_treemap_text() can be used to add a text label to each tile in a treemap created with geom_treemap().

Usage

```
geom_treemap_text(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  padding.x = grid::unit(1, "mm"),
  padding.y = grid::unit(1, "mm"),
  place = "topleft",
  min.size = 4,
  grow = FALSE,
  reflow = FALSE,
  fixed = NULL,
  layout = "squarified",
  start = "bottomleft",
  ...
)
```

Arguments

`mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...`	Standard geom arguments as for `ggplot2::geom_text()`.
`padding.x, padding.y`	`grid::unit()` object, giving horizontal or vertical padding between text and edge of tile. Defaults to 1 mm.
`place`	Where inside the box to place the text. Default is 'bottom'; other options are 'topleft', 'top', 'topright', etc.
`min.size`	Minimum font size, in points. If provided, text that would need to be shrunk below this size to fit the box will not be drawn. Defaults to 4 pt.
`grow`	If `TRUE`, text will be grown as well as shrunk to fill the box.
`reflow`	If `TRUE`, text will be reflowed (wrapped) to better fit the box.
`fixed`	Deprecated. Use `layout = "fixed"` instead. Will be removed in later versions.
`layout`	The layout algorithm, one of either 'squarified' (the default), 'scol', 'srow' or 'fixed'. See Details for full details on the different layout algorithms.
`start`	The corner in which to start placing the tiles. One of 'bottomleft' (the default), 'topleft', 'topright' or 'bottomright'.

Details

`geom_treemap_text()` requires `area` and `label` aesthetics. Several other standard 'ggplot2' aesthetics are supported (see Aesthetics).

`geom_treemap_text()` uses the 'ggfittext' package to fit text to tiles. All text drawing options available in `ggfittext::geom_fit_text()` (growing, reflowing, etc.) are also available here. For full details on how these options work, see the documentation for `ggfittext::geom_fit_text()`.

The `layout` argument is used to set the treemap layout algorithm. All 'treemapify' geoms added to a plot should have the same value for `layout` and `start`, or they will not share a common layout (see `geom_treemap()` for details on the layout algorithms).

Aesthetics

- area (required)
- label (required)
- subgroup, subgroup2 or subgroup3
- colour
- size
- alpha
- family
- fontface
- angle

See Also

`geom_treemap()`

Examples

```
ggplot2::ggplot(G20, ggplot2::aes(area = gdp_mil_usd,
                                  fill = econ_classification,
                                  label = country)) +
  geom_treemap() +
  geom_treemap_text()
```

`treemapify` *Generate a treemap layout.*

Description

`treemapify()` returns a data frame of tile coordinates for a treemap layout of a set of observations. This is only useful if you wish to draw the treemap without the help of the `ggplot2` geoms, or for some edge cases such as creating interactive treemaps with 'R Shiny' (see e.g. https://stackoverflow.com/q/45021775). The easiest way to draw a treemap with the 'treemapify' package is to use the provided 'ggplot2' geoms, such as `geom_treemap()`.

`data` must be a tidy data frame, i.e. each row must represent a single observation and each column a single variable. You must provide the name of the variable that will be represented by the area of each treemap tile with `area`. Optionally, you can also select up to three variables (with `subgroup`, `subgroup2` and `subgroup3`) to generate a layout in which the tiles are clustered into subgroups nested up to three levels deep.

Four layout algorithms are provided. With the default 'squarified' algorithm (`layout = "squarified"`), the priority is ensuring the tiles have an aesthetically pleasing aspect ratio; that is, they are not too narrow or too short. In this algorithm, tile placement proceeds from one corner, placing the tiles in either rows or columns until all the tiles are placed. See Bruls et al. (1999) for the full algorithm.

There are two variants on the 'squarified' algorithm. 'scol' forces tile placement to begin with a column, regardless of the effect on aspect ratio; 'srow' forces tile placement to been with a row. This will also apply to all subgroups. After the first row or column, the remaining tiles will be placed so as to optimise aspect ratios, as with the default algorithm.

With the 'fixed' layout algorithm (`layout = "fixed"`), the plot area is divided into vertical columns, which are each filled with an equal number of tiles beginning at the starting corner. Unlike the 'squarified' algorithm, with the 'fixed' algorithm the relative positions of the tiles are fixed by their order in the input data frame. This can result in aesthetically unpleasing layouts, but it allows side-by-side comparisons or animations to be created.

`treemapify_fixed` is an alias for `treemapify(layout = "fixed")`.

Usage

```
treemapify(
  data,
  area,
  subgroup,
  subgroup2,
  subgroup3,
  layout = "squarified",
  start = "bottomleft",
  fill = NULL,
  label = NULL,
  group = NULL,
  fixed = NULL,
  xlim = c(0, 1),
  ylim = c(0, 1)
)

treemapify_fixed(...)
```

Arguments

`data`	A tidy data frame.
`area`	Name of the variable (a column in `data`) to be mapped to the area of treemap tiles.
`subgroup, subgroup2, subgroup3`	Optionally, names of variables (columns in `data`) by which the tiles should be grouped, at up to three nested levels.
`layout`	The layout algorithm, one of either 'squarified' (the default), 'scol', 'srow' or 'fixed'. See Details for full details on the different layout algorithms.
`start`	The corner in which to start placing the tiles. One of 'bottomleft' (the default), 'topleft', 'topright' or 'bottomright'.

label, fill	Deprecated. Will be removed in later versions.
group	Deprecated. Use subgroup instead. Will be removed in later versions.
fixed	Deprecated. Use layout = "fixed" instead. Will be removed in later versions.
xlim, ylim	The boundaries of the treemap in the x and y dimensions. Must be a numeric vector of length two; both default to c(0, 1).
...	Additional arguments to be passed to treemapify().

Details

treemapify is for generating a data frame of raw treemap coordinates. If you want to draw a treemap with 'ggplot2', use geom_treemap() instead.

References

Bruls, M., Huizing, K., & van Wijk, J. (1999). Squarified Treemaps (pp. 33-42).Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization. http://www.win.tue.nl/~vanwijk/stm.pdf

See Also

geom_treemap()

Examples

```
treemapify(G20, area = "gdp_mil_usd")
```

Index