

Package ‘ggalignment’

May 8, 2026

Type Package

Title Plots 'D&D'-Style Alignment Charts

Version 1.0.2

Description 'D&D' alignment charts show 9 boxes with values for good through evil and values for chaotic through lawful. This package easily creates these alignment charts from user-provided image paths and alignment values.

License MIT + file LICENSE

Depends R (>= 3.4)

Imports dplyr (>= 1.0.0), ggimage (>= 0.2.0), ggplot2 (>= 3.3.0), magrittr (>= 1.0.0), rlang (>= 0.1.2)

Suggests rmarkdown (>= 2.0.0), knitr (>= 1.0), testthat (>= 3.0.0), vdiff (>= 1.0.0), roxygen2

VignetteBuilder knitr

URL <https://github.com/aftonsteps/ggalignment>

BugReports <https://github.com/aftonsteps/ggalignment/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Afton Coombs [aut, cre, cph],
Tan Ho [ctb] (ORCID: <<https://orcid.org/0000-0001-8388-5155>>)

Maintainer Afton Coombs <aftoncoombs@gmail.com>

Repository CRAN

Date/Publication 2025-02-24 05:50:02 UTC

Contents

alignment_vals	2
example_cats	2
ggalignment	3

Index	5
--------------	----------

alignment_vals	<i>Alignment Values</i>
----------------	-------------------------

Description

A vector of possible alignment values.

Usage

```
alignment_vals
```

Format

A data.frame vector containing 1 column of 9 elements, each one a possible alignment
alignment the nine possible alignments

Source

<https://dungeonsdragons.fandom.com/wiki/Alignment>

example_cats	<i>Example Cats</i>
--------------	---------------------

Description

Creates cat data with alignments for use in examples

Usage

```
example_cats()
```

Value

a data.frame containing example data for cats

Examples

```
example_cats()
```

ggalignment *Creates a D&D alignment chart*

Description

The primary function of the package, this function creates a D&D alignment chart from a dataframe with `img`, `x`, and `y` columns!

Usage

```
ggalignment(
  alignment,
  line_type = "dashed",
  line_color = "black",
  font_family = NULL,
  font_color = "black",
  font_size = NULL,
  background_color = "white",
  background_border = NA,
  max_images_per_dim = 2,
  max_image_dim = "width"
)
```

Arguments

<code>alignment</code>	a data.frame containing the data to be plotted, requiring columns <code>img</code> (for image path) and <code>alignment</code> , and optionally <code>x</code> and <code>y</code> specifying the coordinates for each image, where each box has coordinate limits from -1 to 1 in both axes.
<code>line_type</code>	the linetype for the box borders, which follows the ggplot2 allowable values for <code>linetype</code> for <code>geom_rect()</code> (e.g. <code>blank</code> , <code>solid</code> , <code>dashed</code> , <code>dotted</code> , <code>dotdash</code> , <code>longdash</code> , <code>twodash</code>)
<code>line_color</code>	the color for the bounding boxes of the alignments, defaults to <code>black</code> , and must be a named color such as <code>"black"</code>
<code>font_family</code>	the font family to be used on the alignment labels
<code>font_color</code>	the font color to be used on the alignment labels
<code>font_size</code>	the size of the font used on the alignment labels
<code>background_color</code>	the background color for the entire plot, defaults to <code>white</code> and must be a named color such as <code>"white"</code>
<code>background_border</code>	the color of the solid-line bounding box on the entire plot, defaults to <code>NA</code> and must be either <code>NA</code> or a named color such as <code>"black"</code>
<code>max_images_per_dim</code>	numeric representing the number of images that should fit in a single fact – for example, if you want an image to take up half the width of the fact, use <code>max_images_per_dim = 2</code>

`max_image_dim` one of "width" or "height", representing if the `max_images_per_dim` should count by width or height in the facet

Value

a ggplot containing the alignment chart

Examples

```
align_cats <- example_cats()
ggalignment(alignment = align_cats)
```

Index

* **datasets**

alignment_vals, [2](#)

alignment_vals, [2](#)

example_cats, [2](#)

ggalignment, [3](#)