Package ‘ggalignment’

November 4, 2022

Type Package
Title Plots ‘D&D’-Style Alignment Charts
Version 1.0.1
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), vdiffr (>= 1.0.0), roxygen2
VignetteBuilder knitr
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Config/testthat/edition 3
NeedsCompilation no
Author Afton Coombs [aut, cre, cph],
Tan Ho [ctb] (<https://orcid.org/0000-0001-8388-5155>)
Maintainer Afton Coombs <aftoncoombs@gmail.com>
Repository CRAN
Date/Publication 2022-11-04 10:20:02 UTC

R topics documented:

alignment_vals .......................................................... 2
table_1 .................................................................. 2
ggalignment ................................................................. 3

Index 5
alignment_vals  

### 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  

### 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

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

```r
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

- **alignment**: a data.frame containing the data to be plotted, requiring columns `img` (for image path) and `alignment`, and optionally `x` and `y` specifying the coordinates for each image, where each box has coordinate limits from -1 to 1 in both axes.
- **line_type**: the linetype for the box borders, which follows the ggplot2 allowable values for linetype for `geom_rect()` (e.g. blank, solid, dashed, dotted, dotdash, longdash, twodash)
- **line_color**: the color for the bounding boxes of the alignments, defaults to black, and must be a named color such as "black"
- **font_family**: the font family to be used on the alignment labels
- **font_color**: the font color to be used on the alignment labels
- **font_size**: the size of the font used on the alignment labels
- **background_color**: the background color for the entire plot, defaults to white and must be a named color such as "white"
- **background_border**: the color of the solid-line bounding box on the entire plot, defaults to NA and must be either NA or a named color such as "black"
- **max_images_per_dim**: 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 `max_images_per_dim = 2`
ggalignment

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