

# Package: ggtaichi (via r-universe)

June 24, 2026

**Type** Package

**Title** Taichi-Diagram Visualization for Two Data Sources

**Version** 0.1.0

**Description** A data visualization design that compares two (usually on a par with each other) data sources on one grid of taichi (yin-yang) diagrams, where the two interlocking fish of every symbol are filled by the two sources, while inheriting 'ggplot2' features.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/PursuitOfDataScience/ggtaichi>

**BugReports** <https://github.com/PursuitOfDataScience/ggtaichi/issues>

**Depends** R (>= 3.5.0)

**Imports** rlang, grid, ggplot2 (>= 3.0.0), ggnewscale (>= 0.4.5)

**RoxygenNote** 7.3.2

**Suggests** rmarkdown, knitr, dplyr, scales, tidyr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Youzhi Yu [aut, cre]

**Maintainer** Youzhi Yu <yuyouzhi666@icloud.com>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-24 08:37:02 UTC

**RemoteUrl** <https://github.com/cran/ggtaichi>

**RemoteRef** HEAD

**RemoteSha** 3cc778bc6b351a788e81f85af94f703978c24c92

## Contents

geom_taichi . . . . .	2
pitts_emojis . . . . .	3
pitts_tg . . . . .	4
remove_padding . . . . .	5
states_tg . . . . .	5
theme_taichi . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

geom_taichi	<i>Taichi</i>
-------------	---------------

---

### Description

The taichi geom turns each cell of a heatmap-like grid into a taichi (yin-yang) diagram. The two interlocking "fish" of the diagram use luminance to show the values from two data sources on the same plot, so four dimensions of data can be expressed at once: the x and y position of every taichi symbol plus the yin and yang values that fill its two halves.

### Usage

```
geom_taichi(
  yin,
  yin_name = NULL,
  yin_colors = c("gray100", "gray85", "gray50", "gray35", "gray0"),
  yang,
  yang_name = NULL,
  yang_colors = c("#FED7D8", "#FE8C91", "#F5636B", "#E72D3F", "#C20824"),
  ...
)
```

### Arguments

<code>yin</code>	The column name for the yin (dark) fish of the taichi symbol.
<code>yin_name</code>	The label name (in quotes) for the legend of the yin rendering. Default is NULL.
<code>yin_colors</code>	A color vector, usually as hex codes.
<code>yang</code>	The column name for the yang (light) fish of the taichi symbol.
<code>yang_name</code>	The label name (in quotes) for the legend of the yang rendering. Default is NULL.
<code>yang_colors</code>	A color vector, usually as hex codes.
<code>...</code>	<code>...</code> accepts any arguments <code>scale_fill_gradientn()</code> has.

### Value

A taichi diagram comparing two data sources.

## Examples

```
# taichi with categorical variables only

library(ggplot2)

data <- data.frame(x = rep(c("a", "b", "c"), 3),
  y = rep(c("d", "e", "f"), 3),
  yin_values = rep(c(1,5,7),3),
  yang_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
  geom_taichi(yin = yin_values,
    yang = yang_values)

# taichi with numeric variables only

data <- data.frame(x = rep(c(1, 2, 3), 3),
  y = rep(c(1, 2, 3), 3),
  yin_values = rep(c(1,5,7),3),
  yang_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
  geom_taichi(yin = yin_values,
    yang = yang_values)

# taichi with a mixture of numeric and categorical variables

data <- data.frame(x = rep(c("a", "b", "c"), 3),
  y = rep(c(1, 2, 3), 3),
  yin_values = rep(c(1,5,7),3),
  yang_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
  geom_taichi(yin = yin_values,
    yang = yang_values)
```

---

pitts\_emojis

*Popular Emojis*

---

## Description

The most popular Emoji of a given week in a given category from the Meltwater Tweet sample. They can be rendered by using "richtext" with `annotate()`.

## Usage

```
pitts_emojis
```

**Format**

An object of class character of length 270.

---

pitts_tg	<i>Pittsburgh COVID-related Google &amp; Twitter incidence rates</i>
----------	----------------------------------------------------------------------

---

**Description**

A data set containing the 30-week incidence rates of COVID related categories from week 1 starting from June 1, 2020 to week 30 that ended in the last Sunday of the year in Pittsburgh Metropolitan Statistical Area (MSA). The data columns are introduced below. One quick note about the columns of the data set: `week_start` as a column is present in the data set for illustration purposes, reminding users what week column is. In other words, it does not participate any visualization.

**Usage**

```
pitts_tg
```

**Format**

A data frame with 270 rows and 6 columns:

**msa** Metropolitan statistical area (Pittsburgh only).

**week** week 1 to week 30.

**week\_start** The Monday date of the week started.

**category** 9 Covid-related categories in total.

**Twitter** weekly tweets percentage (%) in the MSA falling into each category.

**Google** weekly Google search percentage (%) in the MSA falling into each category.

**Source**

Just like `states_tg`, Google is processed from Google Health API, and Twitter from Meltwater, a Twitter vendor. Both data sources are processed by the author of the package.

---

remove_padding	<i>Remove ggplot2 default padding</i>
----------------	---------------------------------------

---

### Description

The default ggplot2 plots give certain amount of padding for both continuous and discrete variables. Due to this padding, it makes the plots generated from 'geom\_taiichi()' look like there is something missing. Depends on users' preference, they can remove the "empty space" by using this function. The only thing users need to figure out is whether the 'x' and 'y' scales are continuous or discrete.

### Usage

```
remove_padding(x = "c", y = "d", ...)
```

### Arguments

x	x-axis scale, if it is continuous scale, input "c"; discrete, "d".
y	y-axis scale, if it is continuous scale, input "c"; discrete, "d".
...	...

### Value

remove\_padding

---

states_tg	<i>States' COVID-related Google &amp; Twitter incidence rates</i>
-----------	-------------------------------------------------------------------

---

### Description

A data set containing the 30-week incidence rates of COVID related categories from week 1 starting from June 1, 2020 to week 30 that ended in the last Sunday of the year in 4 states (Florida, Missouri, New York, and Texas). The data columns are introduced below. One quick note about the columns of the data set: week\_start as a column is present in the data set for illustration purposes, reminding users what week column is. In other words, it does not participate any visualization.

### Usage

```
states_tg
```

**Format**

A data frame with 1116 rows and 6 columns:

**state** state

**week** week 1 to week 30.

**week\_start** The Monday date of the week started.

**category** 9 Covid-related categories in total.

**Twitter** weekly tweets percentage (%) in state falling into each category.

**Google** weekly Google search percentage (%) in state falling into each category.

**Source**

Just like `pitts_tg`, Google is processed from Google Health API, and Twitter from Meltwater, a Twitter vendor. Both data sources are processed by the author of the package.

---

theme\_taichi

*Plot Themes*

---

**Description**

A light theme tuned for the taichi grid: it bottoms the legends, drops the panel grid and axis ticks, and gives the canvas a soft off-white background reminiscent of rice paper.

**Usage**

```
theme_taichi(  
  base_size = 11,  
  base_family = "",  
  base_line_size = base_size/22,  
  base_rect_size = base_size/22  
)
```

**Arguments**

`base_size` base font size  
`base_family` base font family  
`base_line_size` base size for line elements  
`base_rect_size` base size for rect elements

**Value**

Adding a taichi theme to all plots generated by using the `ggtaichi` package.

# Index

## \* datasets

pitts\_emojis, 3

pitts\_tg, 4

states\_tg, 5

geom\_taichi, 2

pitts\_emojis, 3

pitts\_tg, 4

remove\_padding, 5

states\_tg, 5

theme\_taichi, 6