# Package ‘ggVennDiagram’

February 20, 2024

**Type** Package  
**Title** A ‘ggplot2’ Implement of Venn Diagram  
**Version** 1.5.2  
**Maintainer** Chun-Hui Gao <gaospecial@gmail.com>

**Description** Easy-to-use functions to generate 2-7 sets Venn or upset plot in publication quality. `ggVennDiagram` plot Venn or upset using well-defined geometry dataset and `ggplot2`. The shapes of 2-4 sets Venn use circles and ellipses, while the shapes of 4-7 sets Venn use irregular polygons (4 has both forms), which are developed and imported from another package ‘venn’, authored by Adrian Dusa. We provided internal functions to integrate shape data with user provided sets data, and calculated the geometry of every regions/intersections of them, then separately plot Venn in four components, set edges/labels, and region edges/labels. From version 1.0, it is possible to customize these components as you demand in ordinary ‘ggplot2’ grammar. From version 1.4.4, it supports unlimited number of sets, as it can draw a plain upset plot automatically when number of sets is more than 7.

**Depends** R (>= 4.1.0)  
**Imports** ggplot2 (>= 3.4.0), dplyr, methods, tibble, aplot, venn (>= 1.12), yulab.utils, forcats

**URL**  
https://github.com/gaospecial/ggVennDiagram,  
https://gaospecial.github.io/ggVennDiagram/

**License** GPL-3  
**Encoding** UTF-8  
**RoxygenNote** 7.2.3  
**Suggests** testthat (>= 2.1.0), knitr, plotly, RColorBrewer, shiny, rmarkdown, tidyrr

**VignetteBuilder** knitr  
**NeedsCompilation** no


**Author** Chun-Hui Gao [aut, cre] (<https://orcid.org/0000-0002-1445-7939>), Guangchuang Yu [ctb] (<https://orcid.org/0000-0002-6485-8781>), Adrian Dusa [aut, cph] (<https://orcid.org/0000-0002-3525-9253>, Adrian Dusa is the author and copyright holder of venn, where ggVennDiagram imports the polygon coordinates enabling 5 - 7 sets Venn diagram.), Turgut Yigit Akyol [ctb] (<https://orcid.org/0000-0003-0897-7716>)

**Repository** CRAN

**Date/Publication** 2024-02-20 08:10:02 UTC

**R topics documented:**

- all_identical ......................................................... 3
- combinations .......................................................... 3
- discern ................................................................. 4
- discern_overlap ........................................................ 5
- get_shapes ............................................................. 6
- get_shape_by_id ...................................................... 6
- get_shape_data ....................................................... 7
- ggVennDiagram ....................................................... 7
- launch_app ............................................................ 9
- overlap ................................................................. 9
- plotData_add_venn .................................................... 10
- plot_shapes .......................................................... 11
- plot_shape_edge ..................................................... 11
- plot_venn ............................................................. 12
- print ................................................................. 13
- process_data ......................................................... 13
- process_upset_data ................................................ 14
- separate_longer_delim ............................................. 15
- shapes ............................................................... 15
- slice_idx ............................................................ 16
- unite ................................................................. 16
- upset-plot ........................................................... 17
- Venn-class ........................................................... 18
- VennPlotData ....................................................... 19
- venn_data ............................................................ 20
- venn_plot_data ..................................................... 21
- vensets ............................................................. 22

**Index** 23
**all_identical**

All members of a list have the same elements

**Description**

All members of a list have the same elements

**Usage**

```r
all_identical(list)
```

**Arguments**

- `list`: a list

**Value**

TRUE or FALSE

---

**combinations**

all possible combinations of n sets

**Description**

all possible combinations of n sets

**Usage**

```r
combinations(n)
```

**Arguments**

- `n`: dim
discern

Set difference.

Description

discern returns the difference between two group of sets selected from a Venn object. If multiple sets are chosen for the slices, union of those sets will be used.

Usage

discern(venn, slice1, slice2 = "all")

## S4 method for signature 'Venn'
discern(venn, slice1, slice2 = "all")

Arguments

venn (Required) A Venn object.
slice1 (Required) The name or the index of the set of interest. Multiple sets can be selected.
slice2 (Optional) The name or the index of the set of interest. Multiple sets can be selected. Default is all the sets except the sets of slice1.

Value

A vector showing the difference between slice1 and slice2.

Author(s)

tyakyol@gmail.com

Examples

venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
discern(venn, slice1 = 1)
discern(venn, slice1 = c(1, 2), slice2 = 3)
**discern_overlap**

*Calculate region of sets*

**Description**

calculate the unique region defined by `Venn` object and the parameter `slice`.

**Usage**

discern_overlap(venn, slice = "all")

```r
## S4 method for signature 'Venn'
discern_overlap(venn, slice = "all")
```

**Arguments**

- **venn**: a Venn object
- **slice**: index of Venn members, default is "all"

**Value**

region items

**Author(s)**

gaospecial@gmail.com

**Examples**

```r
library(ggVennDiagram)
venn <- Venn(list(A=1:3,B=2:5,C=c(1L,3L,5L)))

discern_overlap(venn, slice = "all")
# is equal to
overlap(venn, slice = "all")

# however, `discern_overlap()` only contains specific region
discern_overlap(venn, slice = 1:2)
# is different from
overlap(venn, slice = 1:2)
```
get_shapes

Description
Get all shapes

Usage
get_shapes()

Value
a tibble

Examples
get_shapes()

get_shape_by_id

Description
Specifying a shape

Usage
get_shape_by_id(id)

Arguments
id  shape id

Value
a shape

Examples
get_shape_by_id("401f")
get_shape_data

get applicable shape data for Venn object

Description

ggVennDiagram stores shapes as internal data. You may see all the shapes by using `plot_shapes()` or `get_shapes()`.

Usage

ggVennDiagram

get_shape_data(nsets, type = NULL, shape_id = NULL)

Arguments

- `nsets` number of sets
- `type` type of shape
- `shape_id` shape id

Value

- a tibble describing specific shape

Examples

get_shape_data(nsets = 4, type = "polygon")

ggVennDiagram

Description

ggVennDiagram main parser

Usage

ggVennDiagram(
  x,
  category.names = names(x),
  show_intersect = FALSE,
  set_color = "black",
  set_size = NA,
  label = c("both", "count", "percent", "none"),
  label_alpha = 0.5,
  label_geom = c("label", "text"),
  label_color = "black",
  
)
label_size = NA,
label_percent_digit = 0,
label_txtWidth = 40,
edge_lty = "solid",
edge_size = 1,
force_upset = FALSE,
nintersects = 20,
order.intersect.by = c("size", "name", "none"),
order.set.by = c("size", "name", "none"),
relative_height = 3,
relative_width = 0.3,
...

Arguments

x                     list of items
category.names        default is names(x)
show_intersect        if TRUE the text can be visualized by ‘plotly’
set_color             color of set labels ("black")
set_size              size of set labels (NA)
label                 format of region labels, select one from c("count","percent","both","none")
label_alpha           set 0 to remove the background of region labels
label_geom            layer of region labels, choose from c("label","text")
label_color           color of region labels ("black")
label_size            size of region labels (NA)
label_percent_digit   number of digits when formatting percent label (0)
label_txtWidth        width of text used in showing intersect members, will be ignored unless show_intersection is TRUE (40)
edge_lty              line type of set edges ("solid")
edge_size             line width of set edges (1)
force_upset           if TRUE, will always produce Upset plot no matter how many sets have (FALSE)
nintersects           number of intersects. If NULL, all intersections will show.
order.intersect.by    ‘size’, ‘name’, or "none"
order.set.by          ‘size’, ‘name’, or "none"
relative_height       the relative height of top panel in upset plot
relative_width        the relative width of left panel in upset plot
...                   useless
launch_app

Details
From version 1.4.4, 'ggVennDiagram' will plot a upset plot when the number of sets is more than 7. Besides, user can switch to a upset plot with 'upset_plot()' function. Please check the document of this function.

Value
A ggplot object

Examples
library(ggVennDiagram)
x = list(A=1:5,B=2:7,C=3:6,D=4:9)
ggVennDiagram(x) # 4d venn
ggVennDiagram(x[1:3]) # 3d venn
ggVennDiagram(x[1:2]) # 2d venn

launch_app

Launch Reactor Data Shiny App

Description
Launch Reactor Data Shiny App

Usage
launch_app()

Value
a shiny app

overlap

Intersection of many sets.

Description
overlap returns the same elements of the sets in a Venn object.

Usage
overlap(venn, slice = "all")

## S4 method for signature 'Venn'
overlap(venn, slice = "all")
### Arguments

- **venn**  
  (Required) A Venn object.

- **slice**  
  (Optional) The names or the indices of sets of interest. Default is "all", meaning the intersection will be calculated for all the sets.

### Value

A vector showing the intersection of the sets.

### Author(s)

tyakyol@gmail.com

### Examples

```r
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
overlap(venn)
overlap(venn, slice = c(1, 2))
```

---

### Description

join the shape data with set data

### Usage

```r
plotData_add_venn(plotData, venn)
```

### Arguments

- **plotData**  
  a VennPlot object that stores plot shapes

- **venn**  
  a Venn object that stores set values
**plot_shapes**

*plot all shapes provided by internal dataset*

**Description**

These shapes are mainly collected from the package venn, and VennDiagram. For Venn plot with more than 4 sets, it is usually impossible to plot with simple circle or ellipse. So we need to use a predefined coordinates in plot.

**Usage**

```r
plot_shapes()
```

**Details**

- Shape 101, 201, 301, 401, 402, 501, 502, 601 and 701 are from venn
- Shape 401f is from VennDiagram

See data-raw/shapes.R to find how we incorporate these data.

**Examples**

```r
plot_shapes()
```

---

**plot_shape_edge**

*Plot the set edge of a VennPlotData*

**Description**

This is for viewing the shape id and appearance of the shape.

**Usage**

```r
plot_shape_edge(x)
```

**Arguments**

- `x` a VennPlotData object

**Value**

a ggplot object

**Examples**

```r
shape = get_shape_by_id("301")
plot_shape_edge(shape)
```
plot_venn

Description

plot codes

Usage

plot_venn(
  data,
  show_intersect = FALSE,
  set_color = "black",
  set_size = NA,
  label = "both",
  label_geom = "label",
  label_alpha = 0.5,
  label_color = "black",
  label_size = NA,
  label_percent_digit = 0,
  label_txtWidth = 40,
  edge_lty = "solid",
  edge_size = 1,
  ...
)

Arguments

data plot data
show_intersect if TRUE the text can be visualized by 'plotly'
set_color color of set labels ("black")
set_size size of set labels (NA)
label format of region labels, select one from c("count", "percent", "both", "none")
label_geom layer of region labels, choose from c("label", "text")
label_alpha set 0 to remove the background of region labels
label_color color of region labels ("black")
label_size size of region labels (NA)
label_percent_digit number of digits when formatting percent label (0)
label_txtWidth width of text used in showing intersect members, will be ignored unless show_intersection is TRUE (40)
edge_lty line type of set edges ("solid")
edge_size line width of set edges (1)
...
**Value**

ggplot object, or plotly object if show_intersect is TRUE

**Description**

S3 method for upsetPlotData

S3 method for VennPlotData

**Usage**

```r
## S3 method for class 'upsetPlotData'
print(x, ...)

## S3 method for class 'VennPlotData'
print(x, ...)
```

**Arguments**

- `x`: a VennPlotData object
- `...`: useless

**process_data**

**get plot data**

**Description**

get plot data

**Usage**

```r
process_data(venn, nsets = NULL, shape_id = NULL, type = NULL)

## S4 method for signature 'Venn'
process_data(venn, nsets = length(venn@sets), shape_id = NULL, type = NULL)
```

**Arguments**

- `venn`: a Venn object
- `nsets`: This parameter will be set automatically.
- `shape_id`: apply filter to internal shapes. i.e. shape_id = "601"
- `type`: apply filter to internal shapes. i.e. type = "polygon"
Details

This function will conduct set operations and combine the outputs will stored shapes, thus produce a dataset for plot in next step.
Run `get_shapes()` to show all the characteristics of available shapes. Run `plot_shapes()` to view those shapes.

Examples

```r
## Not run:
venn = Venn(list(A=1:3,B=2:5,C=4:8))
data = process_data(venn)

## End(Not run)
```

Description

process upset data

Usage

```r
process_upset_data(
  venn,
  nintersects = 30,
  order.intersect.by = "size",
  order.set.by = "name",
  specific = TRUE
)
```

Arguments

- **venn**: a class Venn object
- **nintersects**: number of intersects. If NULL, all intersections will show.
- **order.intersect.by**: 'size', 'name', or "none"
- **order.set.by**: 'size', 'name', or "none"
- **specific**: whether return ONLY specific items for a subset, default is TRUE

Details

ggVennDiagram, by default, only return the specific subsets of a region. However, sometimes, we want to show all the overlapping items for two or more sets. For example: [https://github.com/gaospecial/ggVennDiagram/issues/64](https://github.com/gaospecial/ggVennDiagram/issues/64)
Therefore, we add a 'specific' switch to this function. While 'specific = FALSE', the separator will be changed from "/" to "~", and all the overlapping items will be returned. This feature is useful in plotting upset plot.
separate_longer_delim

Value

a upsetPlotData object

---

separate_longer_delim  *Implement of tidyr::separate_longer_delim*

---

Description

*Implement of tidyr::separate_longer_delim*

Usage

`separate_longer_delim(df, col, delim)`

Arguments

- `df`: a data.frame
- `col`: column
- `delim`: delimiter

Value

a data.frame

---

shapes  *shapes: shape data used to setup Venn plot*

---

Description

a collection of geometric shapes, which defined the edge and label of sets in a Venn plot. use `plot_shapes()` to see some of them.

Format

a list with several slots see `"?VennPlotData"`.

Source

- The venn datasets authored by Adrian Dusa ([https://CRAN.R-project.org/package=venn](https://CRAN.R-project.org/package=venn)).
- Parameters used to generate fancy four set ellipses are adopted from VennDiagram([https://CRAN.R-project.org/package=VennDiagram](https://CRAN.R-project.org/package=VennDiagram)).
- Wiki
slice_idx

**Description**

check and format slice name

**Usage**

slice_idx(venn, slice)

**Arguments**

- **venn**: a Venn object
- **slice**: a numeric or character vector

**Value**

the index of Venn (numeric vector) or "all"

---

unite

**Description**

unite returns the union of the sets in a Venn object.

**Usage**

unite(venn, slice = "all")

## S4 method for signature 'Venn'

unite(venn, slice = "all")

**Arguments**

- **venn**: (Required) A Venn object.
- **slice**: (Optional) The names or the indices of sets of interest. Default is "all", meaning the union will be calculated for all the sets.

**Value**

A vector showing the union of the sets.
**upset-plot**

*Author(s)*

tyakyol@gmail.com

**Examples**

```r
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
unite(venn)
unite(venn, slice = c(1, 2))
```

**Description**

This function generate a upset plot by creating a composite plot which contains subplots generated by ggplot2.

**Usage**

```r
plot_upset(
  venn,
  nintersects = NULL,
  order.intersect.by = c("size", "name", "none"),
  order.set.by = c("size", "name", "none"),
  relative_height = 3,
  relative_width = 0.3,
  top.bar.color = "grey30",
  top.bar.y.label = NULL,
  top.bar.show.numbers = TRUE,
  top.bar.numbers.size = 3,
  sets.bar.color = "grey30",
  sets.bar.show.numbers = FALSE,
  sets.bar.x.label = "Set Size",
  intersection.matrix.color = "grey30",
  specific = TRUE,
  ...
)
```

**Arguments**

- **venn**: a class Venn object
- **nintersects**: number of intersects. If NULL, all intersections will show.
- **order.intersect.by**: 'size', 'name', or "none"
- **order.set.by**: 'size', 'name', or "none"
relative_height
    the relative height of top panel in upset plot
relative_width
    the relative width of left panel in upset plot
top.bar.color
    default is "grey30"
top.bar.y.label
    default is NULL
top.bar.show.numbers
    default is TRUE
top.bar.numbers.size
    text size of numbers
sets.bar.color
    default is "grey30"
sets.bar.show.numbers
    default is FALSE
sets.bar.x.label
    default is "Set Size"
intersection.matrix.color
    default is "grey30"
specific
    whether only include specific items in subsets, default is TRUE.
... useless

Value
    an upset plot

Examples
    list = list(A = sample(LETTERS, 20),
                 B = sample(LETTERS, 22),
                 C = sample(LETTERS, 14),
                 D = sample(LETTERS, 30, replace = TRUE))
    venn = Venn(list)
    plot_upset(venn)
    plot_upset(venn, order.intersect.by = "name")
    plot_upset(venn, nintersects = 6)

Venn-class
    Venn is a S4 class to represent multiple sets.

Description
    Print user-friendly information of a Venn object
Usage

Venn(sets, names = NULL)

## S4 method for signature 'ANY'
Venn(sets, names = NULL)

## S4 method for signature 'Venn'
show(object)

Arguments

sets (Required) A list containing vectors in the same class. If a vector contains dupli-

cates they will be discarded. If the list doesn’t have names the sets will be

named as "Set_1", "Set_2", "Set_3" and so on.

names names of sets

object a Venn class object

Value

A Venn object.

Slots

sets A list object containing vectors in the same type.

names The names of the sets if it has names. If the list doesn’t have names, the sets will be

named as "Set_1", "Set_2", "Set_3" and so on.

Examples

venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))

print(venn)

VennPlotData

An S3 class constructor of representing Venn plot components.

Description

An S3 class constructor of representing Venn plot components.

Usage

VennPlotData(x)

Arguments

x data source of a VennPlotData object
Slots

shapeId  shape id

type  type of shape

nsets  number of sets

setEdge  a data.frame, the coordinates of set edges, can be retrieved by \texttt{venn\_setedge()}

setLabel  a data.frame, the coordinates of set labels, can be retrieved by \texttt{venn\_setlabel()}

regionEdge  a data.frame, the coordinates of different regions, can be retrieved by \texttt{venn\_regionedge()}

regionLabel  a data.frame, the centroid of the regions, where region labels anchored, can be retrieved by \texttt{venn\_regionlabel()}

setData  a data.frame, the set data provided by user, can be retrieved by \texttt{venn\_set()}

regionData  a data.frame, the region data that calculated by ggVennDiagram, can be retrieved by \texttt{venn\_region()}

---

\textbf{venn\_data} \hspace{1cm} \textit{Prepare Venn data}

\section*{Description}

Prepare Venn data

\section*{Usage}

\texttt{process\_set\_data(venn)}

\texttt{process\_region\_data(venn, sep = "/", specific = TRUE)}

\section*{Arguments}

\begin{itemize}
  \item \texttt{venn}  \hspace{1cm} a Venn object
  \item \texttt{sep}  \hspace{1cm} name and id separator for intersections
  \item \texttt{specific}  \hspace{1cm} whether return ONLY specific items for a subset, default is TRUE
\end{itemize}

\section*{Details}

ggVennDiagram, by default, only return the specific subsets of a region. However, sometimes, we want to show all the overlapping items for two or more sets. For example: https://github.com/gaospecial/ggVennDiagram/issues/64. Therefore, we add a 'specific' switch to this function. While 'specific = FALSE', the separator will be changed from "/" to "-", and all the overlapping items will be returned. This feature is useful in plotting upset plot.

\section*{Value}

\begin{itemize}
  \item a tibble
\end{itemize}
Examples

```r
x = list(
  A = sample(letters, 8),
  B = sample(letters, 8),
  C = sample(letters, 8),
  D = sample(letters, 8)
)

venn = Venn(x)
process_set_data(venn)
process_region_data(venn)
```

---

**Description**

Get VennPlotData slot

**Usage**

```r
venn_regionedge(obj)
venn_regionlabel(obj)
venn_setedge(obj)
venn_setlabel(obj)
venn_set(obj)
venn_region(obj)
```

**Arguments**

- `obj` a list that stores all the data from the S3 class `VennPlotData` object

**Value**

a tibble

**Examples**

```r
venn = Venn(list(A=1:5,B=2:7,C=3:6,D=4:9))
obj = process_data(venn)
venn_regionlabel(obj)  # return regionLabel data
venn_regionedge(obj)   # return regionEdge data
venn_setlabel(obj)     # return setLabel data
```
venn_setedge(obj)  # return setEdge data
venn_set(obj)      # set items
venn_region(obj)   # region items

vensets

Import venn shape coordinates

Description

Import venn shape coordinates

Usage

vensets()

Value

a data frame
Index

all_identical, 3
combinations, 3
discern, 4
discern,Venn-method (discern), 4
discern_overlap, 5
discern_overlap,Venn-method (discern_overlap), 5
get_shape_by_id, 6
get_shape_data, 7
get_shapes, 6
ggVennDiagram, 7
launch_app, 9
overlap, 9
overlap,Venn-method (overlap), 9
plot_shape_edge, 11
plot_shapes, 11
plot_upset (upset-plot), 17
plot_venn, 12
plotData_add_venn, 10
print, 13
process_data, 13
process_data,Venn-method (process_data), 13
process_region_data (venn_data), 20
process_set_data (venn_data), 20
process_upset_data, 14
separate_longer_delim, 15
shapes, 15
show,Venn-method (Venn-class), 18
slice_idx, 16
unite, 16
unite,Venn-method (unite), 16
upset-plot, 17
Venn (Venn-class), 18
Venn,ANY-method (Venn-class), 18
Venn-class, 18
venn_data, 20
venn_plot_data, 21
venn_region (venn_plot_data), 21
venn_regionedge (venn_plot_data), 21
venn_regionlabel (venn_plot_data), 21
venn_set (venn_plot_data), 21
venn_setedge (venn_plot_data), 21
venn_setlabel (venn_plot_data), 21
VennPlotData, 19
vensets, 22