Package 'wheatmap'

May 30, 2026

Type Package

Title Incrementally Build Complex Plots using Natural Semantics

Version 0.2.4

URL https://github.com/zwdzwd/wheatmap

BugReports https://github.com/zwdzwd/wheatmap/issues

Description Builds complex plots, heatmaps in particular, using natural semantics. Bigger plots can be assembled using directives such as 'LeftOf', 'RightOf', 'TopOf', and 'Beneath' and more. Other features include clustering, dendrograms and integration with 'ggplot2' generated grid objects. This package is particularly designed for bioinformaticians to assemble complex plots for publication.

Encoding UTF-8

License GPL-3

RoxygenNote 7.3.3

Imports grid, gtable, stats, colorspace, RColorBrewer, pals

Suggests knitr, rmarkdown, png, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Wanding Zhou [aut, cre]

Maintainer Wanding Zhou <zhouwanding@gmail.com>

Repository CRAN

Date/Publication 2026-05-30 18:50:02 UTC

Contents

`+.WObject` *merge plotting objects*

Description

merge plotting objects

Usage

```
## S3 method for class 'WObject'
group + p
```

Arguments

`group`	a WGroup or a plotting object
`p`	a new plotting object

Value

a WGroup

`AddWGroup` *Add a plotting object to a group*

Description

The object to be added are in the same coordinate system as the group.

Usage

```
AddWGroup(group.obj, new.obj)
```

Arguments

`group.obj`	WGroup object to be added to
`new.obj`	plotting object to be added

Value

a WGroup object where new.obj is added.

Beneath *Beneath*

Description

Generate dimension beneath another object. Tags the result with ‘placement = "south"‘; see [TopOf()] for what that controls.

Usage

```
Beneath(
  x = NULL,
  height = NULL,
  pad = 0.01,
  min.ratio = 0.02,
  h.aln = NULL,
  v.scale = NULL,
  v.scale.proportional = FALSE
)
```

Arguments

x	an object with dimension
height	the height of the new object (when NULL set proportional to the data)
pad	padding between the target and current
min.ratio	minimum ratio of dimensions when auto-scale
h.aln	object for horizontal alignment (when NULL, set to x)
v.scale	object for vertical scaling (when NULL, set to x)
v.scale.proportional	when v.scale is provided, whether to make proportional to data

Value

a dimension generator beneath x (with ‘placement = "south"‘)

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WColorBarH(1:20, cmp=CMPar(), continuous=FALSE, Beneath())
```

both.cluster *row- and column-cluster a matrix*

Description

row- and column-cluster a matrix

Usage

```
both.cluster(
  mat,
  extra.row = NULL,
  extra.column = NULL,
  hc.method = "ward.D2",
  dist.method = "euclidean"
)
```

Arguments

mat	input matrix
extra.row	extra row reordering
extra.column	extra column reordering
hc.method	method to use in hclust
dist.method	method to use in dist

Value

a list of clustered row, column and matrix

Examples

```
WHeatmap(both.cluster(matrix(rnorm(100),nrow=10))$mat)
```

BottomLeftOf *Bottom left of*

Description

Place a new object to the bottom left corner of another.

Usage

```
BottomLeftOf(x = NULL, just = c("right", "bottom"), v.pad = 0, h.pad = 0)
```

Arguments

x	target object, either a name, a object or NULL which refers to the last plotting object
just	the part from the new object that should be attached to
v.pad	vertical translational padding [0.0]
h.pad	horizontal translational padding [0.0]

Value

a WDimGenerator

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomLeftOf(just=c('right','top')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomLeftOf(just=c('right','bottom')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomLeftOf(just=c('left','bottom')))
```

BottomRightOf *Bottom right of*

Description

Place a new object to the bottom right corner of another.

Usage

```
BottomRightOf(x = NULL, just = c("left", "bottom"), v.pad = 0, h.pad = 0)
```

Arguments

x	target object, either a name, a object or NULL which refers to the last plotting object
just	the part from the new object that should be attached to
v.pad	vertical translational padding [0.0]
h.pad	horizontal translational padding [0.0]

Value

a WDimGenerator

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomRightOf(just=c('left','top')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomRightOf(just=c('left','bottom')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), BottomRightOf(just=c('right','bottom')))
```

CalcTextBounding *Calculate Text Bounding*

Description

Calculate bounding box including texts.

Usage

```
CalcTextBounding(x, group)
```

Arguments

x	object
group	the top-level WGroup object

Details

W.R.T lower left corner of the view port in the unit of points

---

CalcTextBounding.WHeatmap *Calculate Texting Bounding for WHeatmap*

Description

Calculate Texting Bounding for WHeatmap

Usage

```
## S3 method for class 'WHeatmap'
CalcTextBounding(x, group)
```

Arguments

x	an object of class WHeatmap
group	an object of class WGroup

Value

an object of class WDim in coordinate points

CMPar *Color Map Parameters*

Description

Create color map parameters

Usage

```
CMPar(
  dmin = NULL,
  dmax = NULL,
  brewer.name = NULL,
  brewer.n = 3,
  colorspace.name = NULL,
  colorspace.n = 2,
  cmap = NULL,
  label2color = NULL,
  use.data = FALSE,
  stop.points = NULL,
  na.color = "#C0C0C0",
  rev = FALSE,
  grey.scale = FALSE
)
```

Arguments

dmin	minimum for continuous color map
dmax	maximum for continuous color map
brewer.name	palette name for RColorbrewer
brewer.n	number of stop points in RColorBrewer for continuous color map
colorspace.name	colorspace name
colorspace.n	number of stops in colorspace palettes
cmap	customized colormap name
label2color	a named vector or list that defines label to color mapping explicitly for discrete color mapping

use.data	use data as color, data must be either common color names or hexdecimal color names
stop.points	custome stop points
na.color	color for NA values. Use NA for transparent (no plotting).
rev	reverse stop points
grey.scale	whether to use grey scale

Value

an object of class CMPar

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WColorBarV(1:20, cmp=CMPar(brewer.name = 'RdBu'), RightOf())
```

ColorMap *Constructor for ColoMap object*

Description

Create color maps

Usage

```
ColorMap(
  continuous = TRUE,
  colors = NULL,
  dmin = NULL,
  dmax = NULL,
  scaler = NULL,
  mapper = NULL
)
```

Arguments

continuous	whether colormap is continuous
colors	colors for each data point
dmin	miminum in continuous color map
dmax	maximum in continuous color map
scaler	scaler function from data range to 0-1
mapper	function that maps data to color

Value

an object of class ColorMap

column.cluster *column cluster a matrix*

Description

column cluster a matrix

Usage

```
column.cluster(mat, ..., hc.method = "ward.D2", dist.method = "euclidean")
```

Arguments

mat	input matrix
...	extra color bars or matrix that needs column reordered
hc.method	method to use in hclust
dist.method	method to use in dist

Value

a list of clustered row, column and matrix

Examples

```
WHeatmap(column.cluster(matrix(rnorm(100),nrow=10))$mat)
```

FromAffine *Convert from affine coordinates to absolute coordinates*

Description

Convert from affine coordinates to absolute coordinates

Usage

```
FromAffine(dm.affine, dm.sys)
```

Arguments

dm.affine	dimension on affine coordinates (relative coordinates)
dm.sys	dimension of the affine system

Value

dimension on the same coordinate system

getdim *Get dimensions*

Description

Get dimensions

Usage

```
getdim(x)
```

Arguments

x	WDim object or a plotting object

Value

vector of dimensions

grid.dendrogram *Draw dendrogram under grid system*

Description

The dendrogram can be renderred. A viewport is created which contains the dendrogram.

Usage

```
grid.dendrogram(
  dend,
  facing = c("bottom", "top", "left", "right"),
  max_height = NULL,
  order = c("normal", "reverse"),
  ...
)
```

Arguments

dend	a stats::dendrogram object.
facing	facing of the dendrogram.
max_height	maximum height of the dendrogram.
order	order
...	additional options

Details

-order should leaves of dendrogram be put in the normal order (1, ..., n) or reverse order (n, ..., 1)? -... pass to ‘grid::viewport‘ which contains the dendrogram.

This function only plots the dendrogram without adding labels. The leaves of the dendrogram locates at unit(c(0.5, 1.5, ...(n-0.5))/n, "npc").

Value

view port that plots dendrogram

GroupCheckNameUnique *Check whether group names are unique*

Description

Check whether group names are unique

Usage

```
GroupCheckNameUnique(group.obj)
```

Arguments

group.obj	a WGroup

Value

TRUE or FALSE

GroupDeepGet *Get an plotting object from a group's descendants*

Description

Get an plotting object from a group's descendants

Usage

```
GroupDeepGet(x, nm, force.unique = TRUE)
```

Arguments

x	a WGroup object
nm	name
force.unique	assume the name is unique in the descendants and get one object instead of a list

Value

if 'force.unique==FALSE' return a list. Otherwise, one plotting object.

`LeftOf` *LeftOf*

Description

Generate dimension to the left of another object. Tags the result with 'placement = "west"'; see [TopOf()] for what that controls.

Usage

```
LeftOf(
  x = NULL,
  width = NULL,
  pad = 0.01,
  min.ratio = 0.02,
  v.aln = NULL,
  h.scale = NULL,
  h.scale.proportional = FALSE
)
```

Arguments

`x`	an object with dimension
`width`	the width of the new object (when NULL, set proportional to data)
`pad`	padding between the target and current
`min.ratio`	minimum ratio of dimensions when auto-scale
`v.aln`	object for vertical alignment (when NULL, set to x)
`h.scale`	object for horizontal scaling (when NULL, set to x)
`h.scale.proportional`	when h.scale is provided, whether to make proportional to data

Value

a dimension to the left of x (with 'placement = "west"')

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WColorBarV(1:20, cmp=CMPar(), continuous=FALSE, LeftOf())
```

ly *show layout*

Description

show layout

Usage

```
ly(x)
```

Arguments

x	plot

Examples

```
ly(
  WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(rnorm(2000),nrow=40), cmp=CMPar(brewer.name = 'RdBu'),
    BottomRightOf(just=c('left','top'))))
```

MapToContinuousColors *map data to continuous color*

Description

map data to continuous color

Usage

```
MapToContinuousColors(data, cmp = CMPar(), given.cm = NULL)
```

Arguments

data	numeric vector
cmp	an color map parameter object of class CMPar
given.cm	given colormap

Value

an object of ColorMap

Examples

```
barplot(1:10, col=MapToContinuousColors(1:10)$colors)
barplot(1:20, col=MapToContinuousColors(c(1:10,10:1))$colors)
```

MapToDiscreteColors *map data to discrete color*

Description

map data to discrete color

Usage

```
MapToDiscreteColors(data, cmp = CMPar(), given.cm = NULL)
```

Arguments

data	numeric vector
cmp	an color map parameter object of class CMPar
given.cm	given color map

Value

an object of ColorMap

Examples

```
pie(rep(1,6), col=MapToDiscreteColors(c(1:3,10:13))$colors)
```

print.WDendrogram *print a dendrogram*

Description

print a dendrogram

Usage

```
## S3 method for class 'WDendrogram'
print(x, stand.alone = TRUE, layout.only = FALSE, cex = 1, ...)
```

Arguments

x	a dendrogram
stand.alone	plot is stand alone
layout.only	plot layout only
cex	factor to scaling texts
...	additional options (ignored)

Value

view port that contains the plotted dendrogram

Examples

```
WDendrogram(column.cluster(matrix(1:24,nrow=4))$column.clust)
```

`print.WGenerator` *print a WGenerator*

Description

This calls WGenerator and creates a WGroup to enclose the produced object.

Usage

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

Arguments

`x`	a WGenerator object
`...`	additional options

Value

the WGroup containing the plotting object

`print.WGG` *plot WGG object*

Description

plot WGG object

Usage

```
## S3 method for class 'WGG'
print(x, cex = 1, layout.only = FALSE, stand.alone = TRUE, ...)
```

Arguments

x	WGG
cex	scaling factor for text
layout.only	plot layout
stand.alone	produce a stand.alone plot
...	extra options

Value

printed ggobj object

---

print.WGrob *plot WGrob object*

Description

plot WGrob object

Usage

```
## S3 method for class 'WGrob'
print(x, cex = 1, layout.only = FALSE, stand.alone = TRUE, ...)
```

Arguments

x	WGrob
cex	scaling factor for text
layout.only	plot layout
stand.alone	produce a stand.alone plot
...	extra options

---

print.WGroup *Draw WGroup*

Description

Draw WGroup

Usage

```
## S3 method for class 'WGroup'
print(x, stand.alone = TRUE, cex = 1, layout.only = FALSE, ...)
```

Arguments

x	a WGroup
stand.alone	to plot stand alone
cex	factor for scaling fonts
layout.only	to plot layout only
...	additional options

---

print.WHeatmap *plot WHeatmap*

Description

plot WHeatmap

Usage

```
## S3 method for class 'WHeatmap'
print(x, cex = 1, layout.only = FALSE, stand.alone = TRUE, ...)
```

Arguments

x	a WHeatmap
cex	factor to scaling texts
layout.only	plot layout only
stand.alone	plot is stand alone
...	additional options

Value

NULL

Examples

```
print(WHeatmap(matrix(1:12, nrow=2)))
```

print.WLabel *print WLabel*

Description

print WLabel

Usage

```
## S3 method for class 'WLabel'
print(x, cex = 1, layout.only = FALSE, stand.alone = TRUE, ...)
```

Arguments

x	a WLabel object
cex	factor to scale text
layout.only	plot layout only
stand.alone	plot label stand alone
...	additional options

Examples

```
print(WLabel("This is a label."))
```

print.WRect *print WRect*

Description

print WRect

Usage

```
## S3 method for class 'WRect'
print(x, cex = 1, layout.only = FALSE, stand.alone = TRUE, ...)
```

Arguments

x	a WRect object
cex	factor for scaling text
layout.only	print layout only
stand.alone	plot WRect standalone
...	additional options

Value

the WRect object

`Resolve` *Resolve name to object*

Description

Resolve name to object

Usage

```
Resolve(x, group, ...)
```

Arguments

`x`	the target
`group`	the parent WGroup object
`...`	extra options passed to methods

`RightOf` *RightOf*

Description

Generate dimension to the right of another object. Tags the result with ‘placement = "east"‘; see [TopOf()] for what that controls.

Usage

```
RightOf(
  x = NULL,
  width = NULL,
  pad = 0.01,
  min.ratio = 0.02,
  v.aln = NULL,
  h.scale = NULL,
  h.scale.proportional = FALSE
)
```

Arguments

x	an object with dimension
width	the width of the new object (when NULL, set proportional to data)
pad	padding between the target and current
min.ratio	minimum ratio of dimensions when auto-scale
v.aln	object for vertical alignment (when NULL, set to x)
h.scale	object for horizontal scaling (when NULL, set to x)
h.scale.proportional	when h.scale is provided, whether to make proportional to data

Value

a dimension to the right of x (with ‘placement = "east"‘)

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WColorBarV(1:20, cmp=CMPar(), continuous=FALSE, RightOf())
```

---

row.cluster *row cluster a matrix*

Description

row cluster a matrix

Usage

```
row.cluster(mat, ..., hc.method = "ward.D2", dist.method = "euclidean")
```

Arguments

mat	input matrix
...	extra color bars or matrix that needs row reordered.
hc.method	method to use in hclust
dist.method	method to use in dist

Value

a list of clustered row, column and matrix

Examples

```
WHeatmap(row.cluster(matrix(rnorm(100),nrow=10))$mat)
```

ScaleGroup *Scale group*

Description

Scale group to incorporate text on margins

Usage

```
ScaleGroup(group.obj)
```

Arguments

group.obj	group object that needs to be scaled

Value

scaled group obj

ToAffine *Convert from absolute coordinates to affine coordinates*

Description

Convert from absolute coordinates to affine coordinates

Usage

```
ToAffine(dm, dm.sys)
```

Arguments

dm	dimension on the same coordinate system as the affine system (absolute coordinates)
dm.sys	dimension of the affine system

Value

dimension on affine coordinates (relative coordinates)

TopLeftOf *Top left of*

Description

Place a new object to the top left corner of another.

Usage

```
TopLeftOf(x = NULL, just = c("right", "bottom"), v.pad = 0, h.pad = 0)
```

Arguments

x	target object, either a name, a object or NULL which refers to the last plotting object
just	the part from the new object that should be attached to
v.pad	vertical translational padding [0.0]
h.pad	horizontal translational padding [0.0]

Value

a WDimGenerator

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopLeftOf(just=c('right','bottom')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopLeftOf(just=c('right','top')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopLeftOf(just=c('left','top')))
```

TopOf *Top of*

Description

Generate dimension top of another object. Both the returned closure and the dimension it produces carry ‘placement = "north"‘, which: * lets ‘WColorBarH(..., label.use.data = TRUE)‘ default ‘xticklabel.side = ’t’‘ so in-situ labels point UPWARD (away from the heatmap body); and * at draw time, lets the sibling-aware relayout shift the new object upward by the predecessor’s northward label overhang.

Usage

```
TopOf(
  x = NULL,
  height = NULL,
  pad = 0.01,
  min.ratio = 0.02,
  h.aln = NULL,
  v.scale = NULL,
  v.scale.proportional = FALSE
)
```

Arguments

x	an object with dimension
height	the height of the new object (when NULL, set to proportional to data)
pad	padding between the target and current
min.ratio	minimum ratio of dimensions when auto-scale
h.aln	object for horizontal alignment (when NULL, set to x)
v.scale	object for vertical scaling (when NULL, set to x)
v.scale.proportional	when v.scale is provided, whether to make proportional to data

Value

a dimension generator on top of x (with ‘placement = "north"‘)

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WColorBarH(1:20, cmp=CMPar(), continuous=FALSE, TopOf())
```

TopRightOf *Top right of*

Description

Place a new object to the top right corner of another.

Usage

```
TopRightOf(x = NULL, just = c("left", "bottom"), v.pad = 0, h.pad = 0)
```

Arguments

x	target object, either a name, a object or NULL which refers to the last plotting object
just	the part from the new object that should be attached to
v.pad	vertical translational padding [0.0]
h.pad	horizontal translational padding [0.0]

Value

a WDimGenerator

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopRightOf(just=c('left','bottom')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopRightOf(just=c('right','top')))

WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'), TopRightOf(just=c('left','top')))
```

WColorBarH *WColorBarH*

Description

A horizontal color bar.

Usage

```
WColorBarH(
  data,
  ...,
  label = NULL,
  label.side = "r",
  label.fontsize = 12,
  label.pad = 0.005,
  label.space = 0.05,
  label.use.data = FALSE
)
```

Arguments

`data`	numeric vector
`...`	additional options to [WHeatmap()]; may include ‘dm‘ (a [TopOf()] / [Beneath()] generator or hard ‘WDim‘), and any tick- label overrides such as ‘xticklabel.side‘
`label`	colorbar header label (drawn at ‘label.side‘)
`label.side`	l (for left) or r (for right) – header position
`label.fontsize`	label font size
`label.pad`	label padding
`label.space`	when label.use.data, the space between labels
`label.use.data`	use data to show legend in situ

Details

Since 0.2.3, when ‘label.use.data = TRUE‘ and the bar is placed via a [TopOf()] / [Beneath()] generator, the in-situ tick-label direction defaults outward (‘TopOf‘ -> labels go up; ‘Beneath‘ -> labels go down). Pass an explicit ‘xticklabel.side‘ via ‘...‘ to override.

Value

an object of class WColorBarH

Examples

```
WColorBarH(matrix(1:50))
```

WColorBarV *WColorBarV*

Description

A vertical color bar.

Usage

```
WColorBarV(
  data,
  ...,
  label = NULL,
  label.side = "t",
  label.fontsize = 12,
  label.pad = 0.005,
  label.space = 0.05,
  label.use.data = FALSE
)
```

Arguments

data	numeric vector
...	additional options to [WHeatmap()]; may include ‘dm‘ (a [RightOf()] / [LeftOf()] generator or hard ‘WDim‘), and any tick-label overrides such as ‘yticklabel.side‘
label	colorbar header label (drawn at ‘label.side‘)
label.side	t (for top) or b (for bottom) – header position
label.fontsize	label font size
label.pad	label padding
label.space	when label.use.data, the space between labels
label.use.data	use data to show legend in situ

Details

Since 0.2.3, when ‘label.use.data = TRUE‘ and the bar is placed via a [RightOf()] / [LeftOf()] generator (anything carrying a ‘placement‘ attribute), the in-situ tick-label direction defaults outward (‘RightOf‘ -> labels go right; ‘LeftOf‘ -> labels go left). Pass an explicit ‘yticklabel.side‘ via ‘...‘ to override.

Value

an object of class WColorBarV

Examples

```
WColorBarV(matrix(50:1))
```

WColumnBind *column bind non-overlapping objects*

Description

column bind non-overlapping objects

Usage

```
WColumnBind(..., nr = NULL, nc = NULL)
```

Arguments

...	plotting objects
nr	number of rows
nc	number of columns

Value

an object of class WDim

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40),name='a') +
  WHeatmap(matrix(rnorm(30), nrow=3), RightOf(),name='b') +
  WColorBarH(1:10, TopOf(WColumnBind('a','b')))
```

`WCustomize` *Customize an existing plot*

Description

Set per-side outer margins as a fraction of the figure dimension. Since wheatmap 0.2.3 these are a **minimum floor**: at draw time ‘print.WGroup‘ wraps the figure in a gtable whose edge cells are auto-sized to fit any outward-facing tick / colorbar / legend labels, and the auto value wins when it exceeds the floor. Pre-0.2.3 call sites that passed e.g. ‘mar.right = 0.25‘ to make room for long labels still work; the override just becomes redundant when the auto sizing already provides enough room.

Usage

```
WCustomize(
  mar.left = NULL,
  mar.right = NULL,
  mar.top = NULL,
  mar.bottom = NULL,
  mar = NULL
)
```

Arguments

`mar.left`	left margin floor, fraction of device width [0.03]
`mar.right`	right margin floor, fraction of device width [0.03]
`mar.top`	top margin floor, fraction of device height [0.03]
`mar.bottom`	bottom margin floor, fraction of device height [0.03]
`mar`	same floor in all four directions [0.03]

Details

Margins are interpreted in NPC of the device width (‘mar.left‘, ‘mar.right‘) or height (‘mar.top‘, ‘mar.bottom‘). ‘mar = 0.05‘ is a shorthand for all four.

Value

an object of class WCustomize

Examples

```
## Old "make room for the legend" pattern still works
WHeatmap(matrix(c('fred','frank','brad',
                'frank','fred','frank'), ncol=2)) +
  WLegendV(NULL, RightOf(), label.fontsize = 20) +
  WCustomize(mar.right=0.1)

## Equivalent under 0.2.3+: the auto-margin already reserves room
WHeatmap(matrix(c('fred','frank','brad',
                'frank','fred','frank'), ncol=2)) +
  WLegendV(NULL, RightOf(), label.fontsize = 20) +
  WCustomize()
```

`WDendrogram` *WDendrogram class*

Description

WDendrogram class

Usage

```
WDendrogram(
  clust = NULL,
  dm = WDim(0, 0, 1, 1),
  name = "",
  facing = c("bottom", "top", "left", "right")
)
```

Arguments

`clust`	hclust object
`dm`	plotting dimension
`name`	name of the dendrogram plot
`facing`	direction of the dendrogram plot

Value

an object of class WDendrogram

Examples

```
WDendrogram(column.cluster(matrix(1:24,nrow=4))$column.clust)
```

`WDim` *class WDim*

Description

class WDim

Usage

```
WDim(
  left = 0,
  bottom = 0,
  width = 1,
  height = 1,
  nr = 1,
  nc = 1,
  text.x = 0,
  text.y = 0,
  text.just = c("center", "center"),
  column.split = NULL,
  row.split = NULL
)
```

Arguments

`left`	left coordinate
`bottom`	bottom coordinate
`width`	width
`height`	height
`nr`	number of row
`nc`	number of column
`text.x`	x anchor for text
`text.y`	y anchor for text
`text.just`	just for text
`column.split`	a list of WDim objects for column split
`row.split`	a list of WDim objects for row split

Value

a WDim object

WGG *WGG object form ggplot with coordinates*

Description

WGG object form ggplot with coordinates

Usage

```
WGG(ggobj, dm = NULL, name = "")
```

Arguments

ggobj	ggplot plotting object
dm	dimension
name	name

Value

WGG object

WGrob *WGrob object plot from a gList of grob objects*

Description

WGrob object plot from a gList of grob objects

Usage

```
WGrob(glist, dm = NULL, name = "")
```

Arguments

glist	gList object
dm	dimension
name	name

Value

WGrob object

WGroup *Construct a WGroup*

Description

Construct a WGroup

Usage

```
WGroup(
  ...,
  name = "",
  group.dm = NULL,
  group.from.member = FALSE,
  mar = WMar(),
  affine = FALSE,
  nr = NULL,
  nc = NULL
)
```

Arguments

...	plotting objects to be grouped
name	name of the group
group.dm	group dimension, by default use the dm of the merge of members
group.from.member	group merged from member coordinates (require affine == FALSE), the supplied group.dm is ignored
mar	a WMar object
affine	whether the group members are on affine coordinates already
nr	number of rows
nc	number of columns

Value

a WGroup object

`WHeatmap` *WHeatmap object*

Description

Create a heatmap

Usage

```
WHeatmap(
  data = NULL,
  dm = NULL,
  name = "",
  continuous = NULL,
  cmp = NULL,
  cm = NULL,
  xticklabels = NULL,
  xticklabels.n = NULL,
  xticklabel.side = "b",
  xticklabel.fontsize = 12,
  xticklabel.rotat = 90,
  xticklabel.pad = 0.005,
  xticklabel.space = 0.05,
  xticklabel.use.data = FALSE,
  yticklabels = NULL,
  yticklabels.n = NULL,
  yticklabel.side = "l",
  yticklabel.fontsize = 12,
  yticklabel.rotat = 0,
  yticklabel.pad = 0.005,
  yticklabel.space = 0.05,
  yticklabel.use.data = FALSE,
  sub.name = NULL,
  bbox = FALSE,
  use_raster = FALSE,
  raster_device = NULL,
  raster_quality = 2,
  gp = NULL
)
```

Arguments

`data`	data matrix
`dm`	plotting dimension (a WDim or a WDimGenerator object)
`name`	name of the plot
`continuous`	whether the data should be treated as continuous or discrete

cmp	a CMPar object, for tunning color mapping parameters
cm	a given color map
xticklabels	to plot xtick labels, one may supply characters to plot just a subset of xtick labels
xticklabels.n	number of xtick labels to plot (resample for aethetics by default)
xticklabel.side	xticklabel side (t or b)
xticklabel.fontsize	xticklabel font size
xticklabel.rotat	xticklabel rotation
xticklabel.pad	padding between xticklabel and x-axis
xticklabel.space	xticklabel space
xticklabel.use.data	use data to label x-axis (most likely used by colorbar)
yticklabels	to plot ytick labels, one may supply characters to plot just a subset of ytick labels
yticklabels.n	number of ytick labels to plot (resample for aethetics by default)
yticklabel.side	yticklabel side (l or r)
yticklabel.fontsize	yticklabel font size
yticklabel.rotat	yticklabel rotation
yticklabel.pad	padding between yticklabel and y-axis
yticklabel.space	yticklabel space
yticklabel.use.data	use data to label y-axis (most likely used by colorbar)
sub.name	subclass name
bbox	whether to plot the boundary box (useful with white matrix elements)
use_raster	whether to render the heatmap body as a raster image using grid.raster(). This reduces file size and improves performance for large matrices. Default is FALSE.
raster_device	the off-screen graphics device for rasterization. Use "png" to render via a temp PNG file (requires the png package), or NULL (default) to construct the raster directly from the color matrix.
raster_quality	a scaling factor for the raster resolution when raster_device is set. Values > 1 increase resolution. Default is 2.
gp	a list of graphical parameters

Value

one or a list of heatmaps (depends on whether dimension is split)

Examples

```
WHeatmap(matrix(1:10, nrow=2), cmp=CMPar(brewer.name='Greens'))

WHeatmap(matrix(1:12,nrow=2), cmp=CMPar(brewer.name='Greens'), name='a') +
   WHeatmap(matrix(1:6,nrow=1), Beneath(pad=0.05), cmp=CMPar(brewer.name='Set2'), name='b') +
   WHeatmap(matrix(c(1:30,30:1),nrow=5), Beneath(pad=0.05), 'c', cmp=CMPar(cmap='jet')) +
    WHeatmap(matrix(1:24,nrow=4), RightOf('c'), 'd', cmp=CMPar(brewer.name='Set1')) +
    WLegendV('c', LeftOf('c', pad=0.01), yticklabel.side='l') +
    WLegendV('b', RightOf('b', width=0.1)) +
    WLegendV('a', RightOf('a')) +
    WHeatmap(matrix(1:100, nrow=10), RightOf('d'), cmp=CMPar(brewer.name='RdYlGn')) +
    WColorBarH(matrix(5:1), TopOf(), cmp=CMPar(colorspace.name = 'diverge_hcl')) +
    WColorBarH(matrix(50:1), TopOf(), cmp=CMPar(colorspace.name = 'terrain_hcl')) +
    WColorBarH(matrix(1:8), TopOf(), cmp=CMPar(colorspace.name = 'sequential_hcl')) +
    WColorBarH(matrix(1:8), TopOf(), cmp=CMPar(brewer.name = 'YlOrRd'))

## One could use %>% too, in combination with magrittr's add function
## Not run:
library(magrittr)
WColorBarH(1:10) %>% add(WColorBarV(rep(c('black','red','blue'),3), RightOf()))

## End(Not run)
```

`WLabel` *construct a WLabel*

Description

construct a WLabel

Usage

```
WLabel(
  x = NULL,
  dm = WDim(),
  name = "",
  fontsize = 12,
  rot = 0,
  color = "black"
)
```

Arguments

`x`	text to be labeled
`dm`	position
`name`	name
`fontsize`	font size

rot	rotation
color	color of the label

Value

a WLabel object

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) + WLabel("This is a label.", RightOf(), rot=-90)
```

WLegendH *WLegendH*

Description

a horizontal legend

Usage

```
WLegendH(
  x = NULL,
  dm = NULL,
  name = "",
  n.stops = 20,
  n.text = 5,
  label.fontsize = 12,
  width = 0.02,
  height = 0.05,
  decreasing = TRUE,
  ...
)
```

Arguments

x	a name or a plotting object, if NULL use the last plotting object
dm	position
name	name of the plotted legend
n.stops	number of stops in computing continuous legend
n.text	number of text labels in continuous legend
label.fontsize	label font size
width	width of each unit in plotted legend
height	height of each unit in plotted legend
decreasing	reversed color map
...	additional options to WHeatmap

Value

an object of class WLegendH

Examples

```
WHeatmap(matrix(1:4,nrow=2))+WLegendH(NULL, Beneath())
```

`WLegendV` *WLegendV*

Description

a vertical legend

Usage

```
WLegendV(
  x = NULL,
  dm = NULL,
  name = "",
  n.stops = 20,
  n.text = 5,
  label.fontsize = 12,
  width = 0.05,
  height = 0.02,
  decreasing = FALSE,
  ...
)
```

Arguments

`x`	a name or a plotting object, if NULL use the last plotting object
`dm`	position
`name`	name of the plotted legend
`n.stops`	number of stops in computing continuous legend
`n.text`	number of text labels in continuous legend
`label.fontsize`	label font size
`width`	width of each unit in plotted legend
`height`	height of each unit in plotted legend
`decreasing`	reversed color map
`...`	additional options to WHeatmap

Value

an object of class WLegendV

Examples

```
WHeatmap(matrix(1:4,nrow=2))+WLegendV(NULL, RightOf())
```

`WMatrix` *plot multiple figures in a matrix*

Description

This function can take WObject, or gg (from ggplot) since the coordinates are not set, gg can be converted to WGG

Usage

```
WMatrix(objs, ncols = 1)
```

Arguments

`objs`	a list of plotting objects either WObject or gg
`ncols`	number of columns

Value

WGroup

`WObject` *Construct a WObject*

Description

Construct a WObject

Usage

```
WObject(dm = NULL, name = "")
```

Arguments

`dm`	position
`name`	name

Value

a WObject

`WPosition` *place an arbitrary position w.r.t a subplot*

Description

place an arbitrary position w.r.t a subplot

Usage

```
WPosition(
  anchor.x,
  anchor.y,
  x = NULL,
  just = c("left", "bottom"),
  data.coord = FALSE
)
```

Arguments

`anchor.x`	x coordinates
`anchor.y`	y coordinates
`x`	plotting object to anchor
`just`	adjustment of new plot
`data.coord`	whether the coordinates is in term of data

Value

a WDimGenerator object

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40)) +
  WHeatmap(matrix(c(rnorm(100)+1,rnorm(100)), nrow=10),
    cmp=CMPar(brewer.name = 'RdGy'),
    WPosition(0.1,0.1,just=c('left','top')))
```

WRect *construct a WRect*

Description

construct a WRect

Usage

```
WRect(
  obj = NULL,
  x.span = NULL,
  y.span = NULL,
  color = "black",
  lwd = 3,
  fill = NA,
  name = ""
)
```

Arguments

obj	a plotting object or its name
x.span	x-axis/horizontal span (e.g., c(2,4))
y.span	y-axis/vertical span (e.g., c(5,9))
color	edge color
lwd	edge width
fill	fill color
name	name

Value

a WRect object

WRowBind *row bind non-overlapping objects*

Description

row bind non-overlapping objects

Usage

```
WRowBind(..., nr = NULL, nc = NULL)
```

Arguments

...	plotting objects
nr	number of rows
nc	number of columns

Value

an object of class WDim

Examples

```
WHeatmap(matrix(rnorm(2000),nrow=40),name='a') +
  WHeatmap(matrix(rnorm(30), nrow=3), Beneath(),name='b') +
  WColorBarV(1:10, LeftOf(WRowBind('a','b')))
```

[.WGroup *subset WGroup*

Description

subset WGroup

Usage

```
## S3 method for class 'WGroup'
x[i]
```

Arguments

x	a WGroup object
i	integer indexing element

Value

a subset of WGroup or NULL

Index