

Package: wheatmap (via r-universe)

September 4, 2024

Type Package

Title Incrementally Build Complex Plots using Natural Semantics

Version 0.2.0

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.

License GPL-3

RoxygenNote 7.1.2

Imports grid, stats, colorspace, RColorBrewer

Suggests knitr, rmarkdown

VignetteBuilder knitr

Language en-US

Repository <https://zwdzwd.r-universe.dev>

RemoteUrl <https://github.com/zwdzwd/wheatmap>

RemoteRef HEAD

RemoteSha 5867026dc25c01faa360919883710871da2c4980

Contents

+.WObject	3
AddWGroup	3
Beneath	4
both.cluster	5
BottomLeftOf	5

BottomRightOf	6
CalcTextBounding	7
CalcTextBounding.WHeatmap	7
CMPar	8
ColorMap	9
column.cluster	10
darkjet.stops	10
FromAffine	11
getdim	11
grid.dendrogram	12
GroupCheckNameUnique	13
GroupDeepGet	13
jet.stops	14
LeftOf	14
ly	15
MapToContinuousColors	15
MapToDiscreteColors	16
print.WDendrogram	16
print.WGenerator	17
print.WGG	18
print.WGrob	18
print.WGroup	19
print.WHeatmap	19
print.WLabel	20
print.WRect	20
Resolve	21
RightOf	21
row.cluster	22
ScaleGroup	23
ToAffine	23
TopLeftOf	24
TopOf	24
TopRightOf	25
WColorBarH	26
WColorBarV	27
WColumnBind	28
WCustomize	29
WDendrogram	30
WDim	30
WGG	31
WGrob	32
WGroup	32
WHeatmap	33
WLabel	36
WLegendH	37
WLegendV	38
WMatrix	39
WObject	39

<code>+.WObject</code>	3
WPosition	40
WRect	41
WRowBind	41
[.WGroup	42
Index	43

<code>+.WObject</code>	<i>merge plotting objects</i>
------------------------	-------------------------------

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	<i>Add a plotting object to a group</i>
-----------	---

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

Usage

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

Arguments

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

Value

a dimension generator beneath x

Examples

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

both.cluster	<i>row- and column-cluster a matrix</i>
--------------	---

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	<i>Bottom left of</i>
--------------	-----------------------

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	<i>Calculate Text Bounding</i>
------------------	--------------------------------

Description

Calculate bounding box including texts.

Usage

```
CalcTextBounding(x, ...)
```

Arguments

x	object
...	extra options

Details

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

CalcTextBounding.WHeatmap	<i>Calculate Texting Bounding for WHeatmap</i>
---------------------------	--

Description

Calculate Texting Bounding for WHeatmap

Usage

```

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

```

Arguments

hm	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 hexadecimal color names
stop.points	custome stop points
na.color	color for NA
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	mimumum 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)
```

darkjet.stops *darker jet color stops*

Description

darker jet color stops

Usage

```
darkjet.stops
```

Format

An object of class character of length 6.

FromAffine	<i>Convert from affine coordinates to absolute coordinates</i>
------------	--

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	<i>Get dimensions</i>
--------	-----------------------

Description

Get dimensions

Usage

```
getdim(x)
```

Arguments

x	WDim object or a plotting object
---	----------------------------------

Value

vector of dimensions

grid.dendrogram	<i>Draw dendrogram under grid system</i>
-----------------	--

Description

The dendrogram can be rendered. 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.

jet.stops	<i>jet color stops</i>
-----------	------------------------

Description

jet color stops

Usage

jet.stops

Format

An object of class character of length 75.

LeftOf	<i>LeftOf</i>
--------	---------------

Description

Generate dimension to the left of another object

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

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

<code>x</code>	a dendrogram
<code>stand.alone</code>	plot is stand alone
<code>layout.only</code>	plot layout only
<code>cex</code>	factor to scaling texts
<code>...</code>	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

<code>x</code>	a <code>WGenerator</code> object
<code>...</code>	additional options

Value

the `WGroup` containing the plotting object

<code>print.WGG</code>	<i>plot WGG object</i>
------------------------	------------------------

Description

plot WGG object

Usage

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

Arguments

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

Value

printed ggobj object

<code>print.WGrob</code>	<i>plot WGrob object</i>
--------------------------	--------------------------

Description

plot WGrob object

Usage

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

Arguments

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

print.WGroup	<i>Draw WGroup</i>
--------------	--------------------

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	<i>plot WHeatmap</i>
----------------	----------------------

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	<i>Resolve name to object</i>
---------	-------------------------------

Description

Resolve name to object

Usage

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

Arguments

x	the target
...	extra options

RightOf	<i>RightOf</i>
---------	----------------

Description

Generate dimension to the right of another object

Usage

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

Arguments

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

Value

a dimension to the right of `x`

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

<code>mat</code>	input matrix
<code>...</code>	extra color bars or matrix that needs row reordered.
<code>hc.method</code>	method to use in hclust
<code>dist.method</code>	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	<i>Scale group</i>
------------	--------------------

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	<i>Convert from absolute coordinates to affine coordinates</i>
----------	--

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

Usage

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

Arguments

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

Value

a dimension generator on top of x

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

<code>data</code>	numeric vector
<code>...</code>	additional options to WHeatmap
<code>label</code>	colorbar label
<code>label.side</code>	l (for left) or r (for right)
<code>label.fontsize</code>	label font size
<code>label.pad</code>	label padding
<code>label.space</code>	when <code>label.use.data</code> , the space between labels
<code>label.use.data</code>	use data to show legend in situ

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

<code>data</code>	numeric vector
<code>...</code>	additional options to WHeatmap
<code>label</code>	colorbar label
<code>label.side</code>	t (for top) or b (for bottom)

`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

Value

an object of class `WColorBarV`

Examples

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

<code>WColumnBind</code>	<i>column bind non-overlapping objects</i>
--------------------------	--

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	<i>Customize an existing plot</i>
------------	-----------------------------------

Description

Customize an existing plot

Usage

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

Arguments

<code>mar.left</code>	left margin [0.03]
<code>mar.right</code>	right margin [0.03]
<code>mar.top</code>	top margin [0.03]
<code>mar.bottom</code>	bottom margin [0.03]
<code>mar</code>	margin in all directions [0.03]

Value

an object of class `WCustomize`

Examples

```
WHeatmap(matrix(c('fred','frank','brad',  
                  'frank','fred','frank'), ncol=2)) +  
WLegendV(NULL, RightOf(), label.fontsize = 20) +  
WCustomize(mar.right=0.1)
```

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	<i>WGrob object plot from a gList of grob objects</i>
-------	---

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	<i>Construct a WGroup</i>
--------	---------------------------

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,
gp = NULL
)

```

Arguments

<code>data</code>	data matrix
<code>dm</code>	plotting dimension (a <code>WDim</code> or a <code>WDimGenerator</code> object)
<code>name</code>	name of the plot
<code>continuous</code>	whether the data should be treated as continuous or discrete
<code>cmp</code>	a <code>CMP</code> object, for tuning color mapping parameters
<code>cm</code>	a given color map
<code>xticklabels</code>	to plot xtick labels, one may supply characters to plot just a subset of xtick labels
<code>xticklabels.n</code>	number of xtick labels to plot (resample for aesthetics by default)
<code>xticklabel.side</code>	xticklabel side (t or b)
<code>xticklabel.fontsize</code>	xticklabel font size
<code>xticklabel.rotat</code>	xticklabel rotation
<code>xticklabel.pad</code>	padding between xticklabel and x-axis
<code>xticklabel.space</code>	xticklabel space
<code>xticklabel.use.data</code>	use data to label x-axis (most likely used by colorbar)
<code>yticklabels</code>	to plot ytick labels, one may supply characters to plot just a subset of ytick labels
<code>yticklabels.n</code>	number of ytick labels to plot (resample for aesthetics by default)

<code>yticklabel.side</code>	yticklabel side (l or r)
<code>yticklabel.fontsize</code>	yticklabel font size
<code>yticklabel.rotat</code>	yticklabel rotation
<code>yticklabel.pad</code>	padding between yticklabel and y-axis
<code>yticklabel.space</code>	yticklabel space
<code>yticklabel.use.data</code>	use data to label y-axis (most likely used by colorbar)
<code>sub.name</code>	subclass name
<code>bbox</code>	whether to plot the boundary box (useful with white matrix elements)
<code>gp</code>	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	<i>plot multiple figures in a matrix</i>
---------	--

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	<i>Construct a WObject</i>
---------	----------------------------

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	<i>construct a WRect</i>
-------	--------------------------

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	<i>row bind non-overlapping objects</i>
----------	---

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

- * **datasets**
 - darkjet.stops, 10
 - jet.stops, 14
- +.WObject, 3
- [.WGroup, 42
- AddWGroup, 3

- Beneath, 4
- both.cluster, 5
- BottomLeftOf, 5
- BottomRightOf, 6

- CalcTextBounding, 7
- CalcTextBounding.WHeatmap, 7
- CMPar, 8
- ColorMap, 9
- column.cluster, 10

- darkjet.stops, 10

- FromAffine, 11

- getdim, 11
- grid.dendrogram, 12
- GroupCheckNameUnique, 13
- GroupDeepGet, 13

- jet.stops, 14

- LeftOf, 14
- ly, 15

- MapToContinuousColors, 15
- MapToDiscreteColors, 16

- print.WDendrogram, 16
- print.WGenerator, 17
- print.WGG, 18
- print.WGrob, 18
- print.WGroup, 19

- print.WHeatmap, 19
- print.WLabel, 20
- print.WRect, 20

- Resolve, 21
- RightOf, 21
- row.cluster, 22

- ScaleGroup, 23

- ToAffine, 23
- TopLeftOf, 24
- TopOf, 24
- TopRightOf, 25

- WColorBarH, 26
- WColorBarV, 27
- WColumnBind, 28
- WCustomize, 29
- WDendrogram, 30
- WDim, 30
- WGG, 31
- WGrob, 32
- WGroup, 32
- WHeatmap, 33
- WLabel, 36
- WLegendH, 37
- WLegendV, 38
- WMatrix, 39
- WObject, 39
- WPosition, 40
- WRect, 41
- WRowBind, 41