Short tutorial of using WHeatmap

Contents

There are many packages that can generate heatmaps in R. WHeatmap designs a set of languages and a layer system that allows arbitrary positioning of heatmaps programmatically. This is a tutorial on the usage of wheatmap for generating complex heatmaps in a procedure way.

Preparation

We start with some data

```r
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 3.2.5
```

```r
load_all('~/tools/wheatmap/wheatmap/')
```

```
## Loading wheatmap
```

```r
m <- cbind(matrix(rnorm(20),nrow=4), 5+matrix(rnorm(8),nrow=4))
m2 <- matrix(1:16,nrow=4)
dimnames(m) <- list(c('w','x','y','z'), c('a','b','c','d','e','f','g'))
row.data <- c(1,2,3,1)
col.data <- c(1:6,6)
m
```

```
##             a          b         c          d          e        f        g
## w  0.08924029 -0.3628584 0.1295817 -0.5544558 -1.2641714 3.293815 4.024872
## x -0.08517412  0.3566101 0.9719953 -0.7664717 -0.6822191 3.479587 5.688247
##  [ reached getOption("max.print") -- omitted 2 rows ]
```

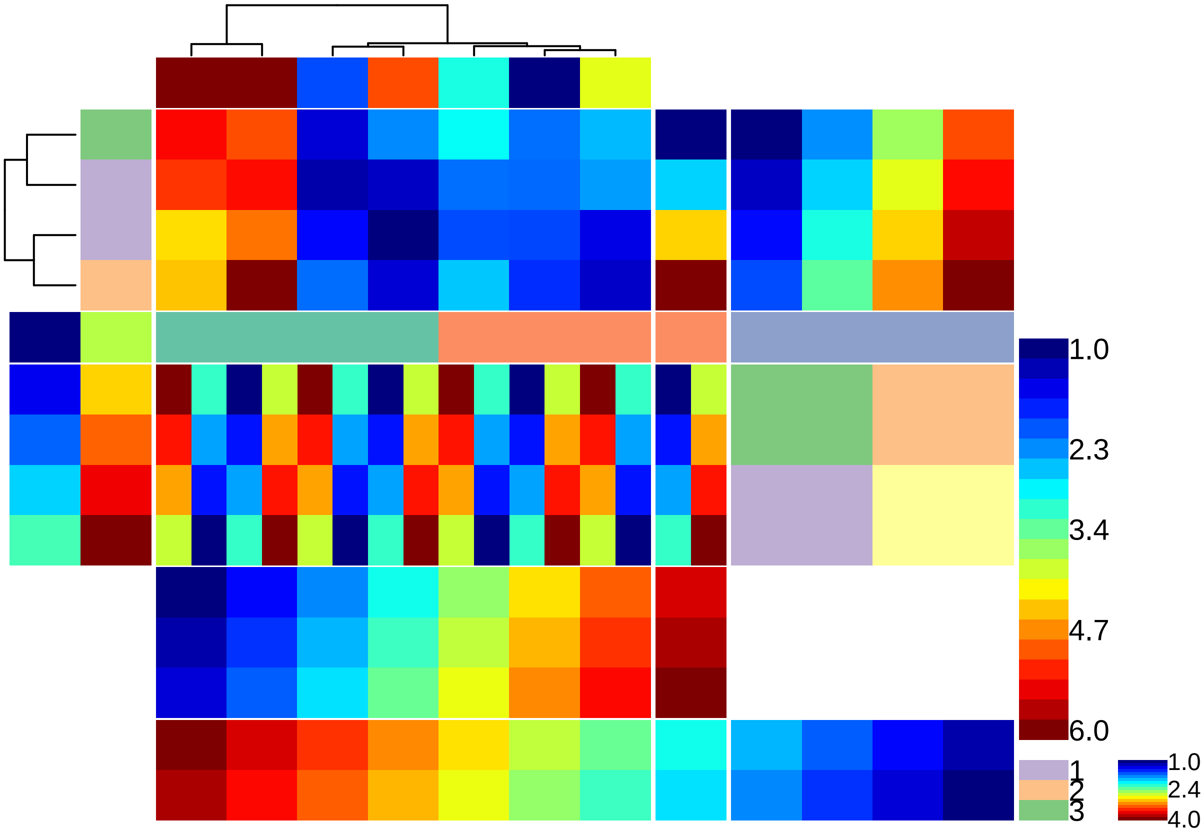
We perform some clustering

```r
cc <- both.cluster(m)
row.data <- row.data[cc$row.clust$order]
col.data <- col.data[cc$column.clust$order]
```

Final results

The end result of our tutorial can be done by the “one”-liner

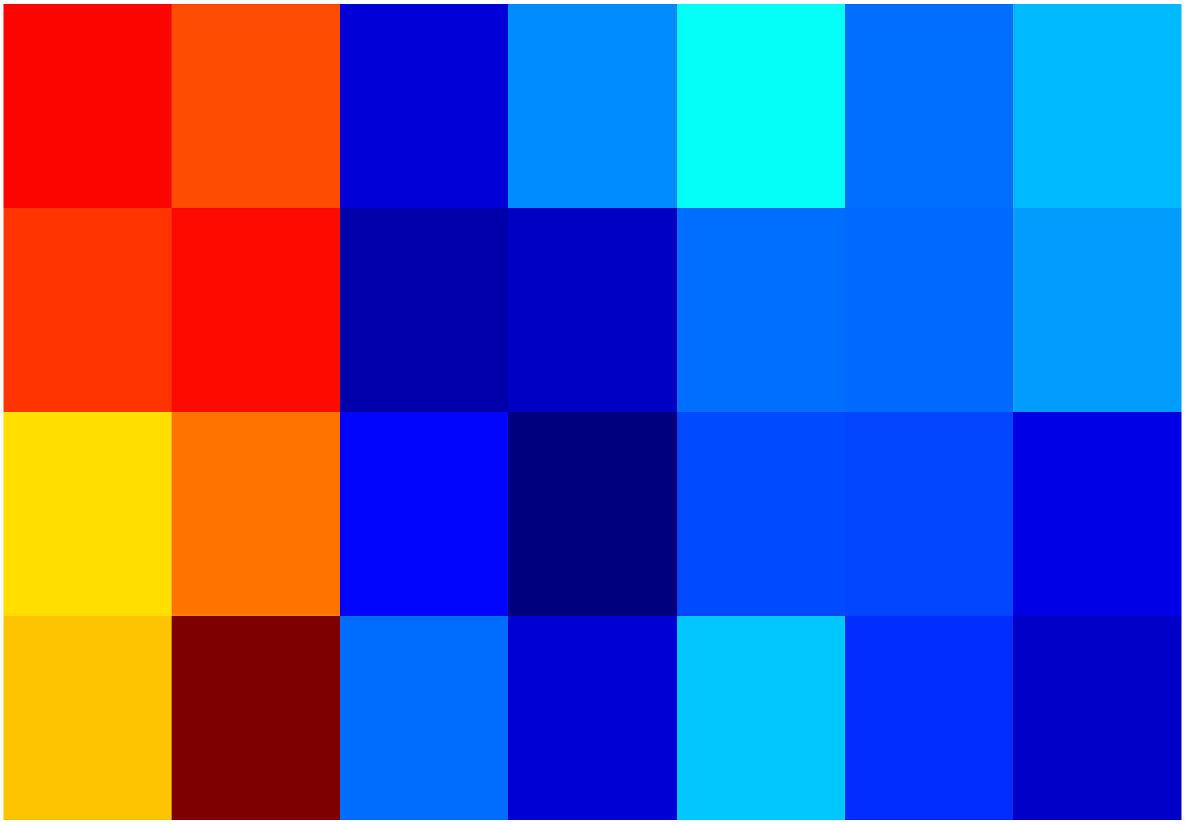
```
WHeatmap(cc$mat, name='h1') +
  WColorBarV(row.data, LeftOf('h1'), 'c1') +
  WColorBarH(col.data, TopOf('h1'), 'c2') +
  WDendrogram(cc$row.clust, LeftOf('c1'), facing='right') +
  WDendrogram(cc$column.clust, TopOf('c2'), facing='bottom') +
  WColorBarV(1:4, RightOf('h1'), 'c3', continuous=TRUE) +
  WHeatmap(m2, RightOf('c3'), 'h2') +
  WColorBarH(rep(c(1,2,3),each=4),
             Beneath(WColumnBind('h1', 'c3', 'h2')), 'c4',
             cmp=CMPar(brewer.name='Set2'), continuous=FALSE) +
  WHeatmap(matrix(rep(c(8:1,1:8),4),nrow=4),
           Beneath('c4', h.aln=WColumnBind('h1','c3')), 'h3') +
  WHeatmap(matrix(rep(1:10),ncol=2),LeftOf(WRowBind('c4.1.1','h3.1.1'))) +
  WHeatmap(matrix(1:4,nrow=2), RightOf('h3', h.scale='h2'), 'h4') +
  WHeatmap(matrix(1:24,nrow=3), Beneath('h3'), 'h5') +
  WHeatmap(matrix(24:1,nrow=2),
           Beneath('h5', h.aln=WColumnBind('h1','c3','h2')), 'h6') +
  WLegendV('c1', BottomRightOf('h6.1.3', h.pad=0.01), 'l1') +
  WLegendV('c2', TopOf('l1', pad = 0.1), 'l2') +
  WLegendV('c3', RightOf('l1', pad=0.1), n.text=3, label.fontsize = 10)
```



Step by step construction

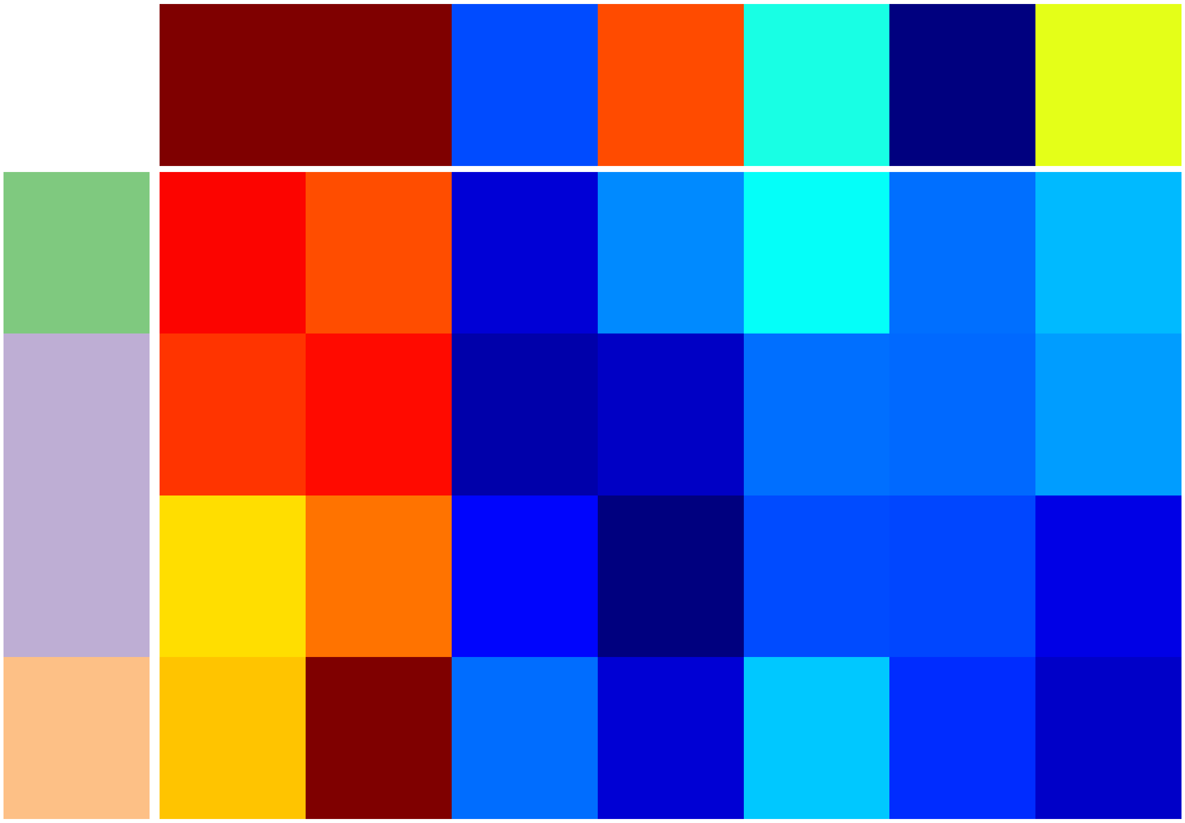
We plot one heatmap first

```
a <- WHeatmap(cc$mat, name='h1')
a
```



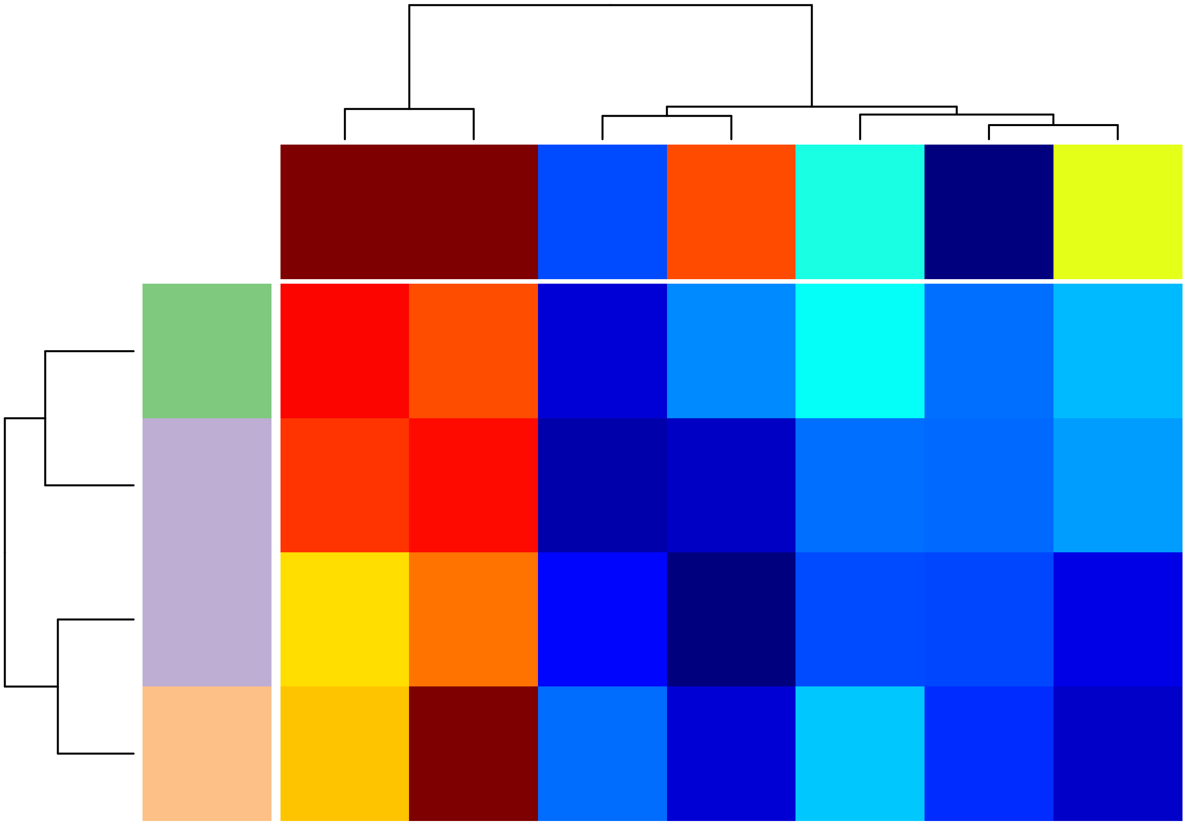
Then we add top and left color bars

```
a <- a + WColorBarV(row.data, LeftOf('h1'), 'c1')
a <- a + WColorBarH(col.data, TopOf('h1'), 'c2')
a
```



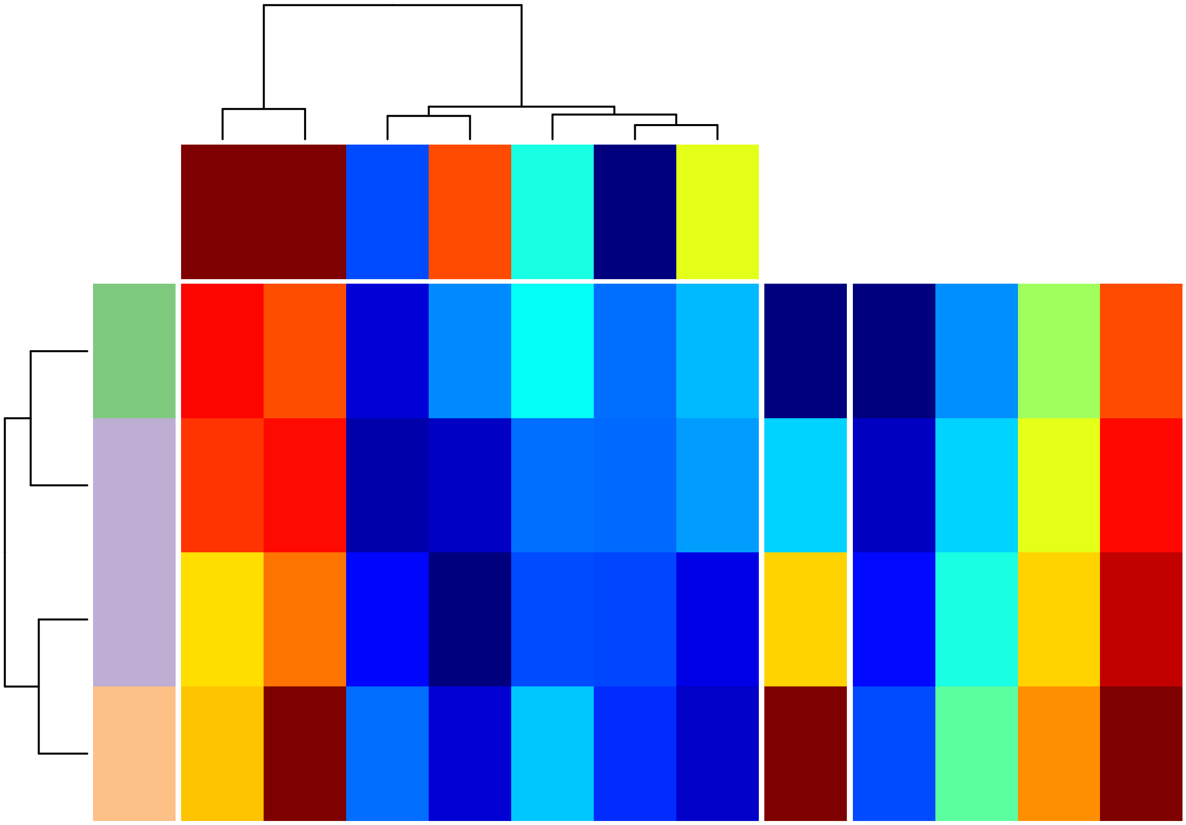
Then the dendrograms

```
a <- a + WDendrogram(cc$row.clust, LeftOf('c1'), facing='right')
a <- a + WDendrogram(cc$column.clust, TopOf('c2'), facing='bottom')
a
```



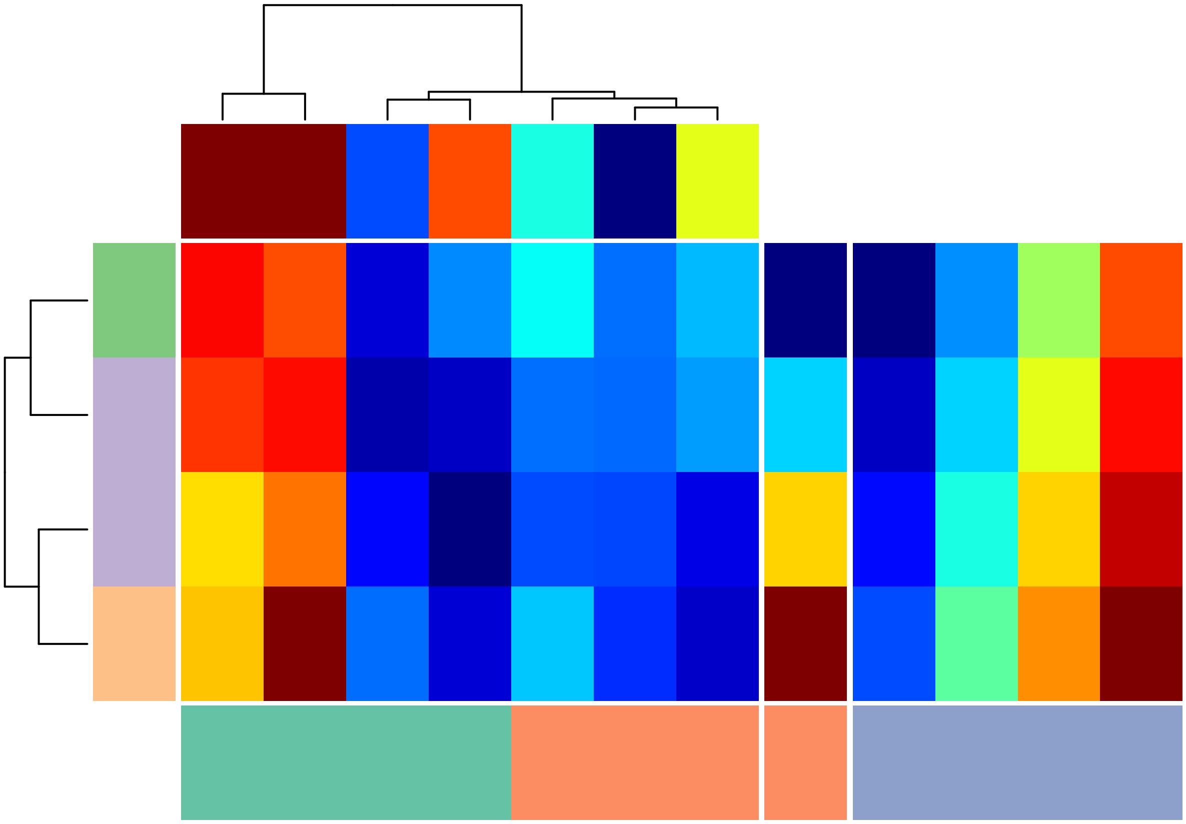
Then another vertical color bar on the right. This one we want to have a continuous scale. Then another heatmap on the further right.

```
a <- a + WColorBarV(1:4, RightOf('h1'), 'c3', continuous=TRUE)
a <- a + WHeatmap(m2, RightOf('c3'), 'h2')
a
```



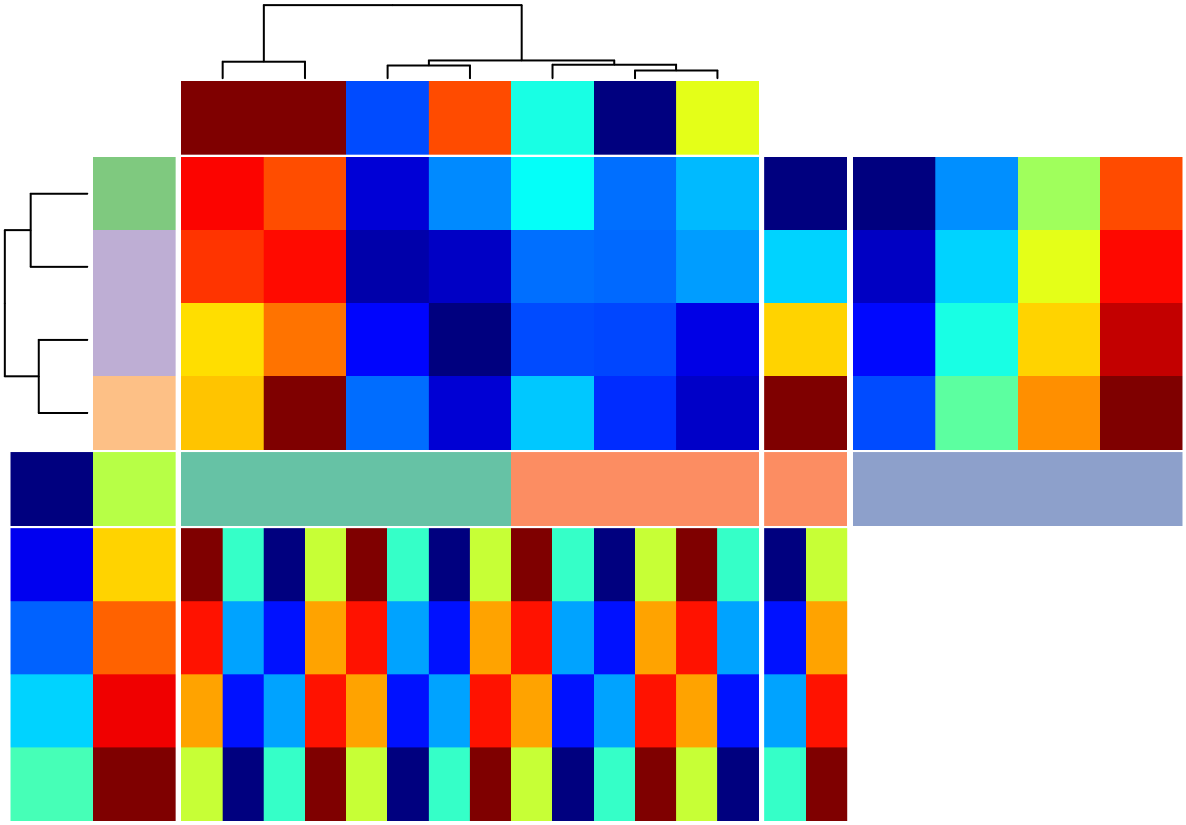
Now we can merge 3 items we plot and add a horizontal bar below. Note wheatmap automatically computes the split for you. It's the users' responsibility however, to make sure data are alignable.

```
a <- a + WColorBarH(rep(c(1,2,3),each=4),
                    Beneath(WColumnBind('h1', 'c3', 'h2')), 'c4',
                    cmp=CMPar(brewer.name='Set2'), continuous=FALSE)
a
```



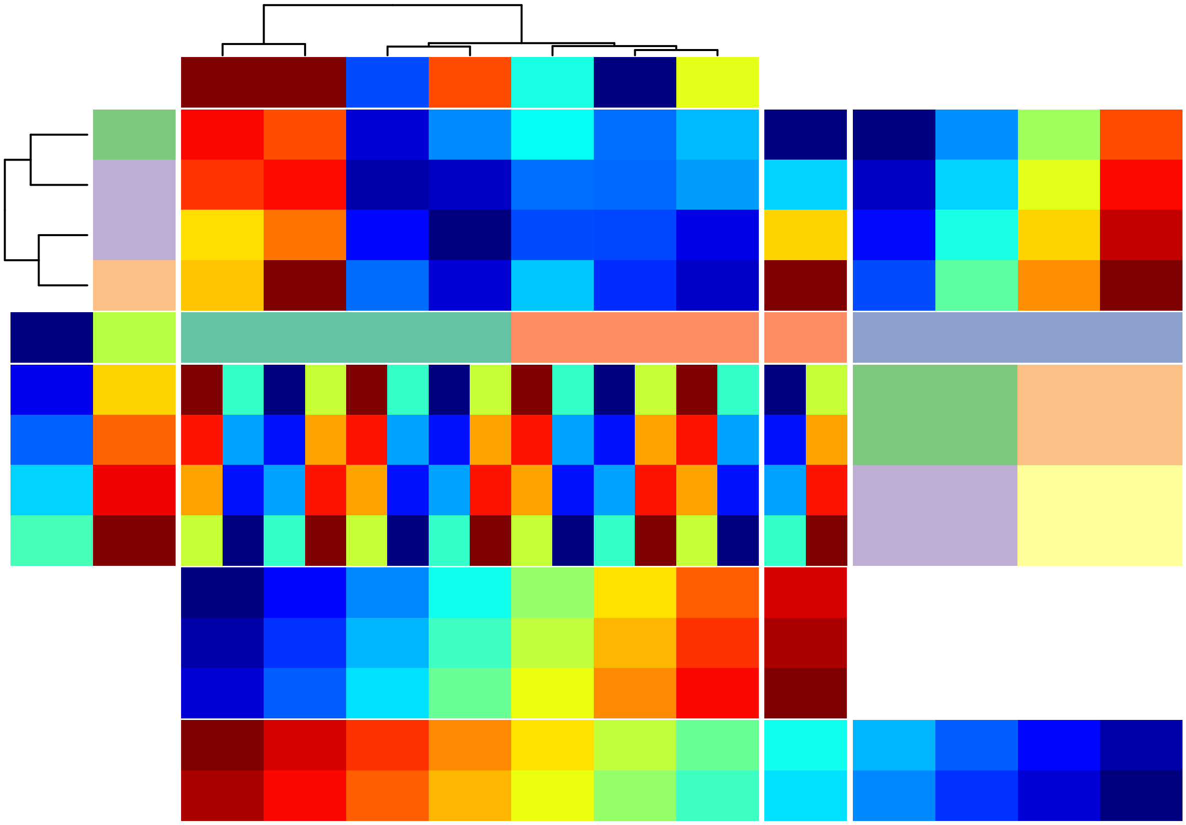
We then add another matrix that span two objects under c4. And a vertial 2-column heatmap on the left that span 2 elements.

```
a <- a + WHeatmap(
  matrix(rep(c(8:1,1:8),4),nrow=4),
  Beneath('c4', h.aln=WColumnBind('h1','c3')), 'h3')
a <- a + WHeatmap(matrix(rep(1:10),ncol=2),
                  LeftOf(WRowBind('c4.1.1','h3.1.1')))
a
```



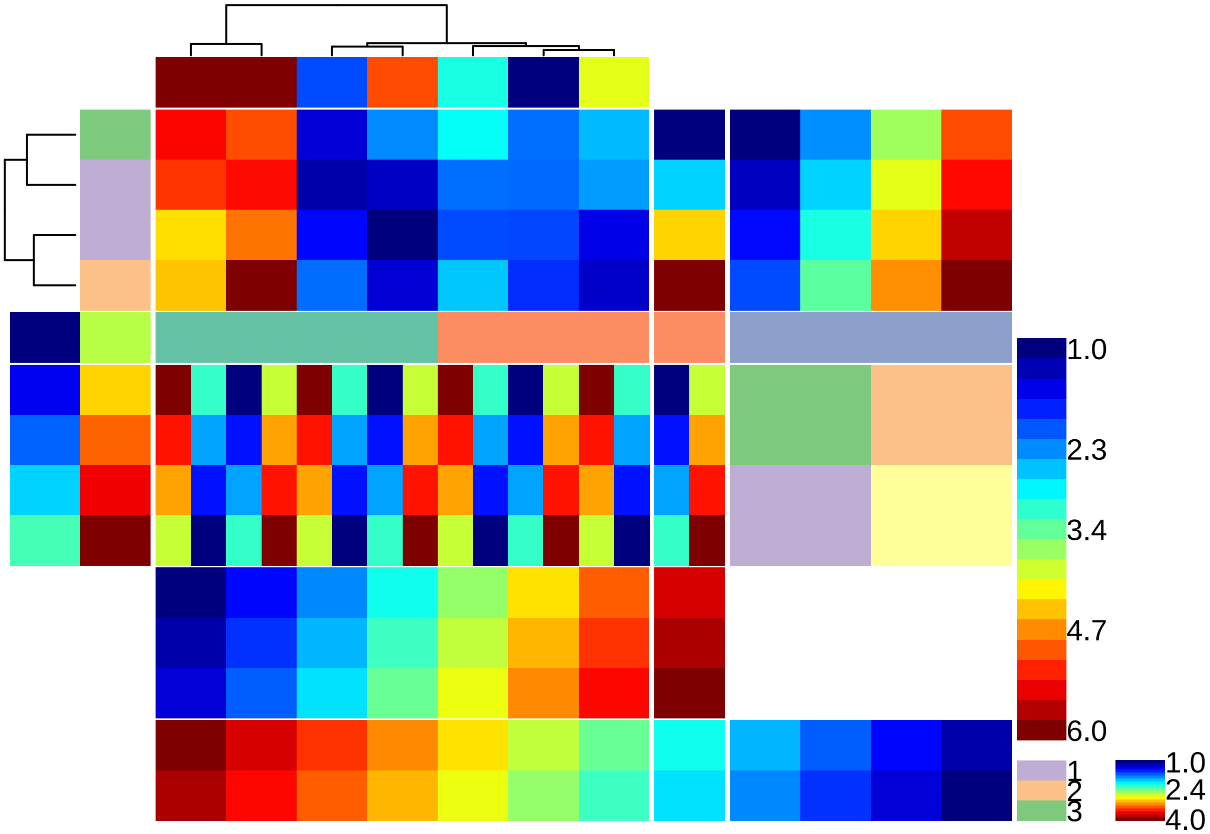
Another to the right of h3

```
a <- a + WHeatmap(
  matrix(1:4,nrow=2), RightOf('h3', h.scale='h2'), 'h4')
a <- a + WHeatmap(
  matrix(1:24,nrow=3), Beneath('h3'), 'h5')
a <- a + WHeatmap(
  matrix(24:1,nrow=2),
  Beneath('h5', h.aln=WColumnBind('h1','c3','h2')), 'h6')
a
```



Let’s add legend

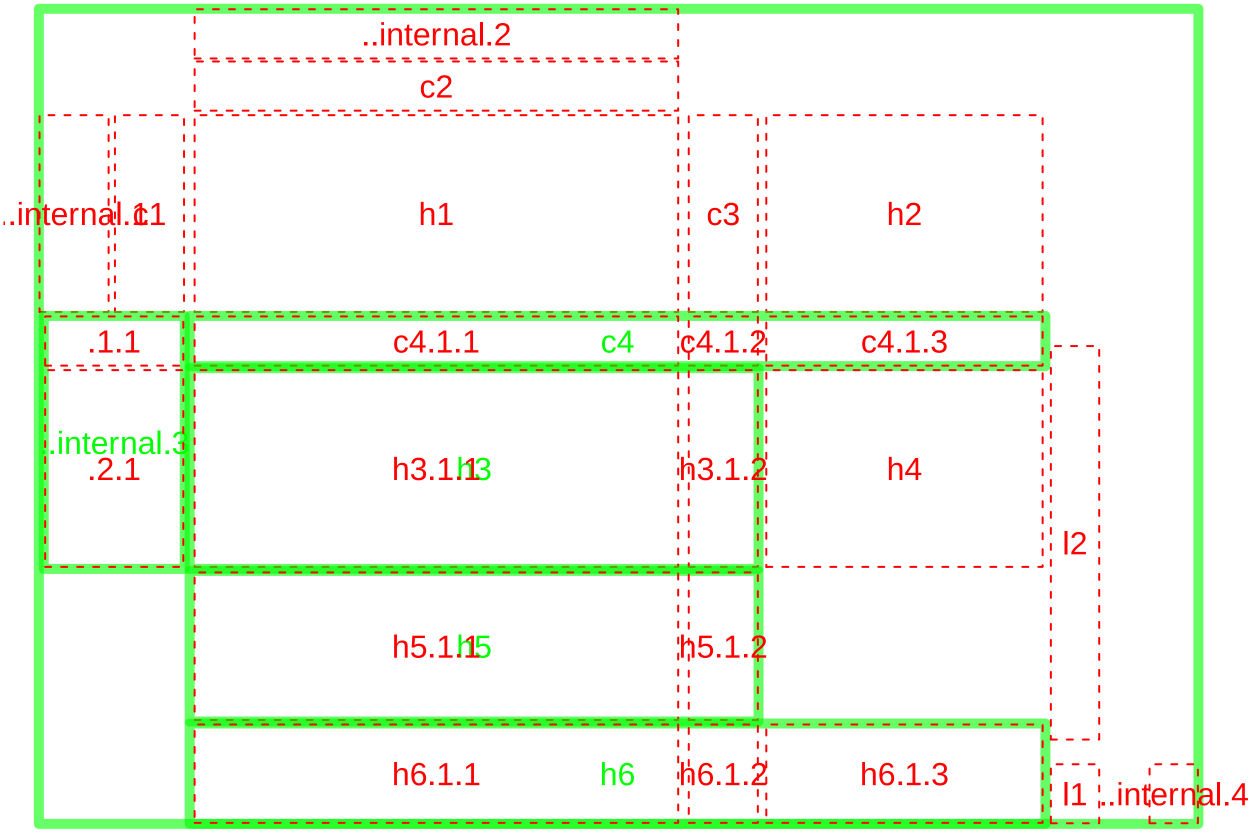
```
a <- a + WLegendV('c1', BottomRightOf('h6.1.3', h.pad=0.01), 'l1')
a <- a + WLegendV('c2', TopOf('l1', pad = 0.1), 'l2')
a <- a + WLegendV('c3', RightOf('l1', pad=0.1), n.text=3)
a
```



Show layout and refer to objects

We can view the internal layout by the providing the `layout.only=TRUE` option. This is useful to see the labeling visually.

```
print(a, layout.only=TRUE)
```

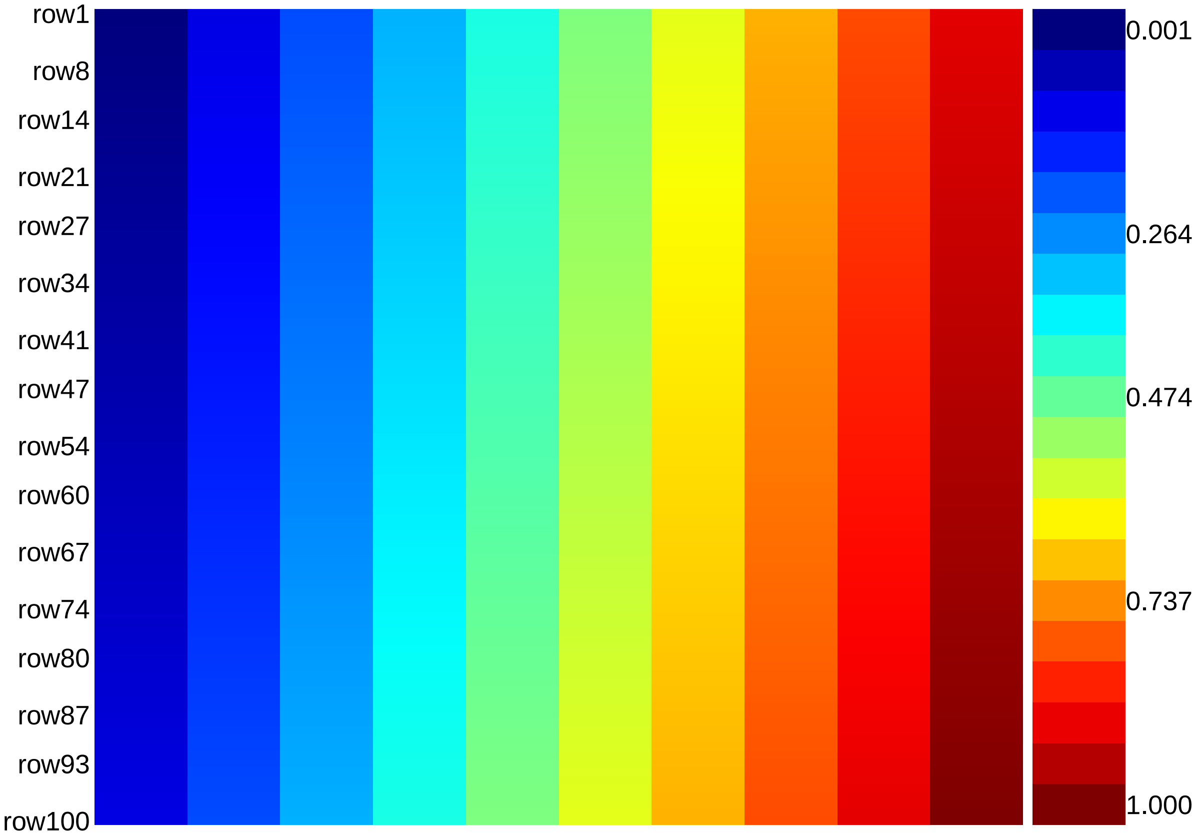


Each object has members with unique names. One can specify a name or have wheatmap generate a name. If an item is a group object by itself, it can also have members of its own. The names of members from different groups can be identical. When that's the case, one needs to use the full path to refer to the object.

Declutter text labels

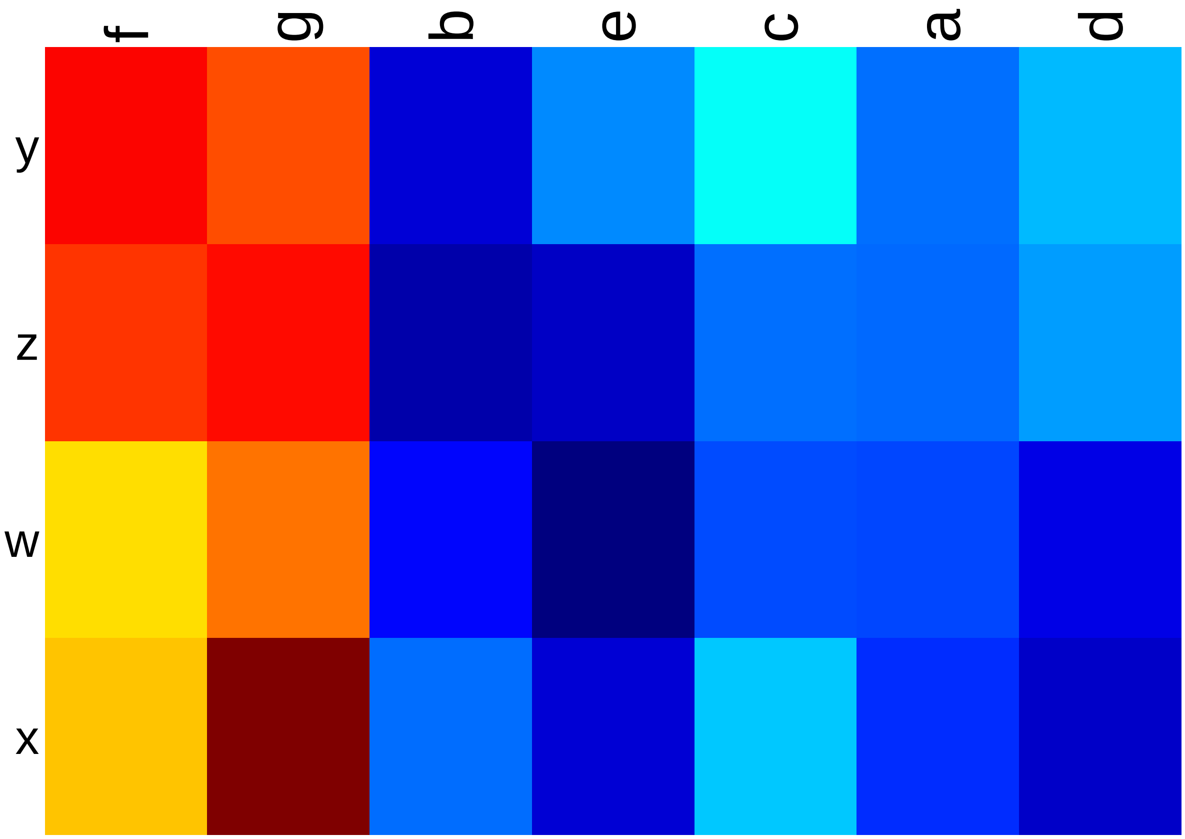
Wheatmap automatically de-cluttered the labels when there are too many. Below is an example of too many labels:

```
m <- matrix((1:1000)/1000, nrow=100)
rownames(m) <- paste0('row', 1:100)
WHeatmap(m, yticklabels = TRUE) + WLegendV(NULL, RightOf(), height=0.5)
```



Text and labels

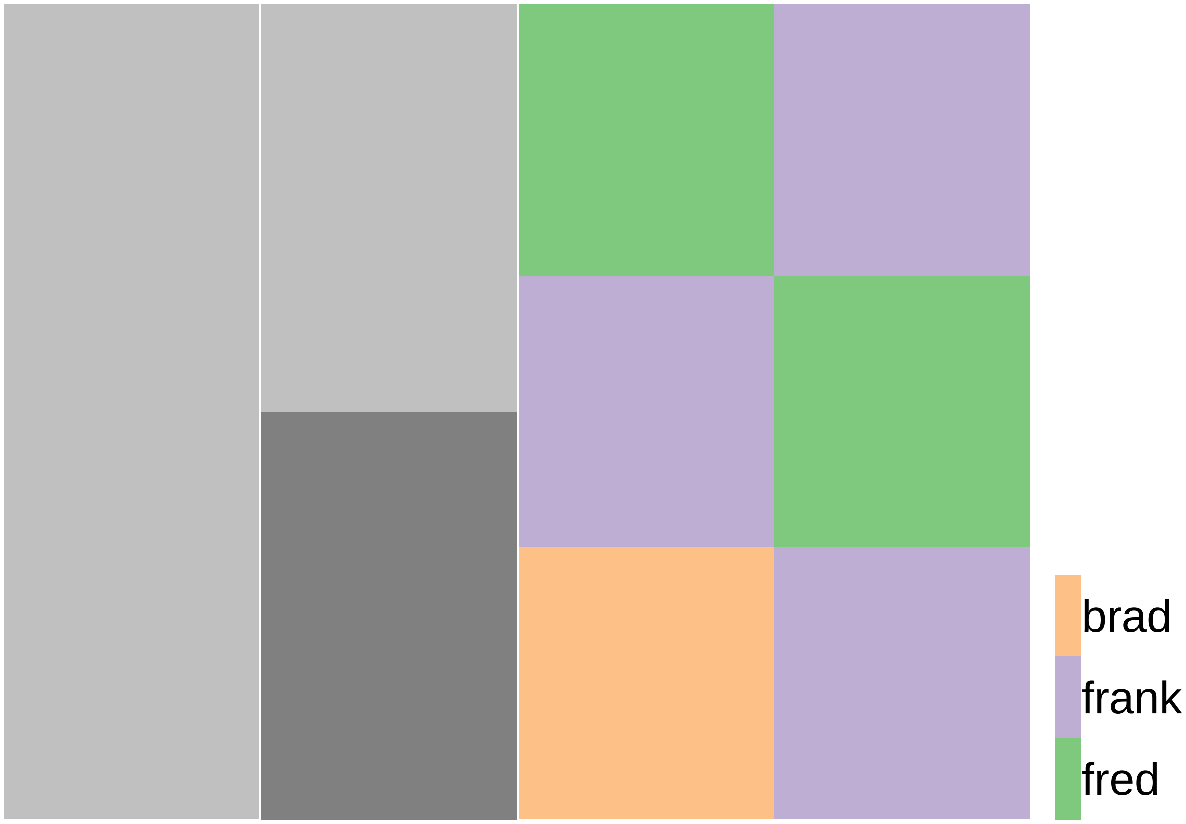
```
WHeatmap(cc$mat, name='h1',
         yticklabels = TRUE, yticklabel.fontsize=20,
         xticklabels = TRUE, xticklabel.side = 't', xticklabel.fontsize = 25)
```



Anchor by edge

The **LeftOf**, **TopOf**, **RightOf** and **Beneath** are for placing a new object by anchoring to the edge of an existing object.

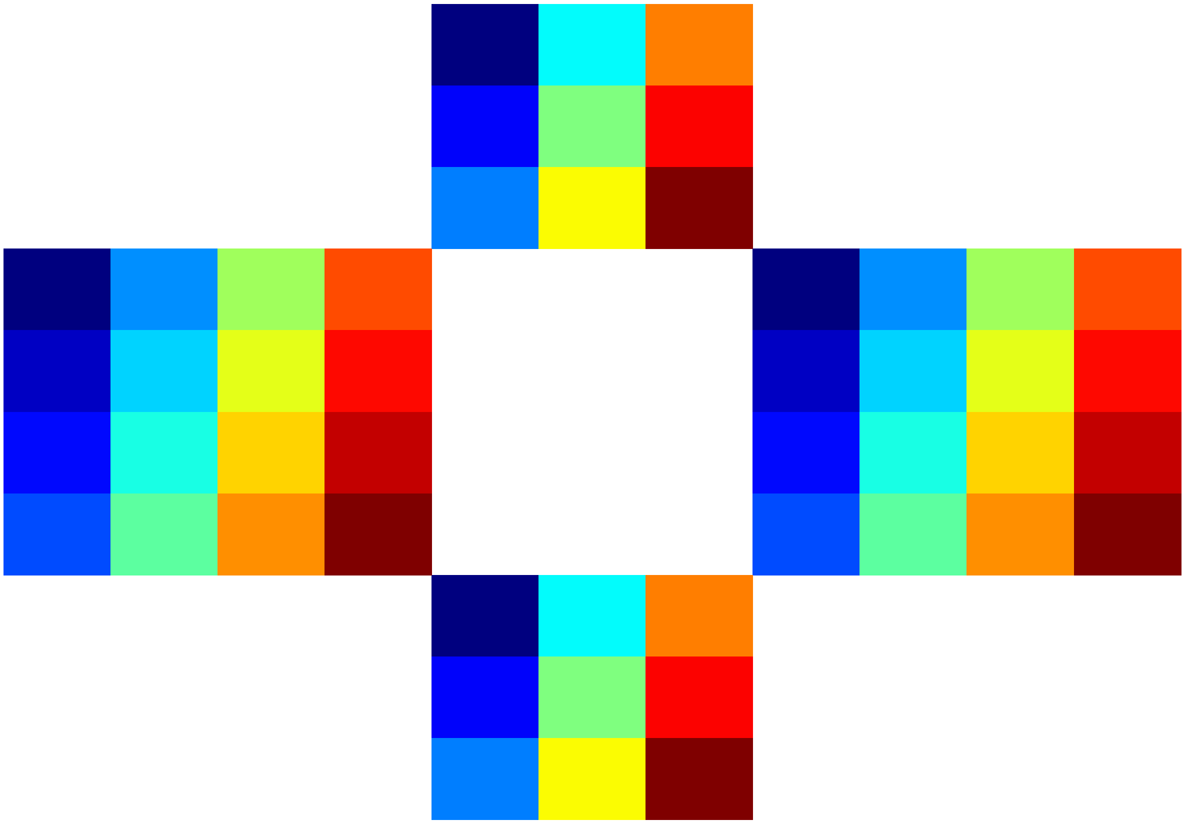
```r
WHeatmap(matrix(1:1)) +
  WHeatmap(matrix(1:2), RightOf()) +
  WHeatmap(matrix(c('fred','frank','brad','frank','fred','frank'), ncol=2), RightOf()) +
  WLegendV(NULL, BottomRightOf(h.pad=0.1), label.fontsize = 20, yticklabel.pad=0.05) +
  WCustomize(mar.right=0.1)
```



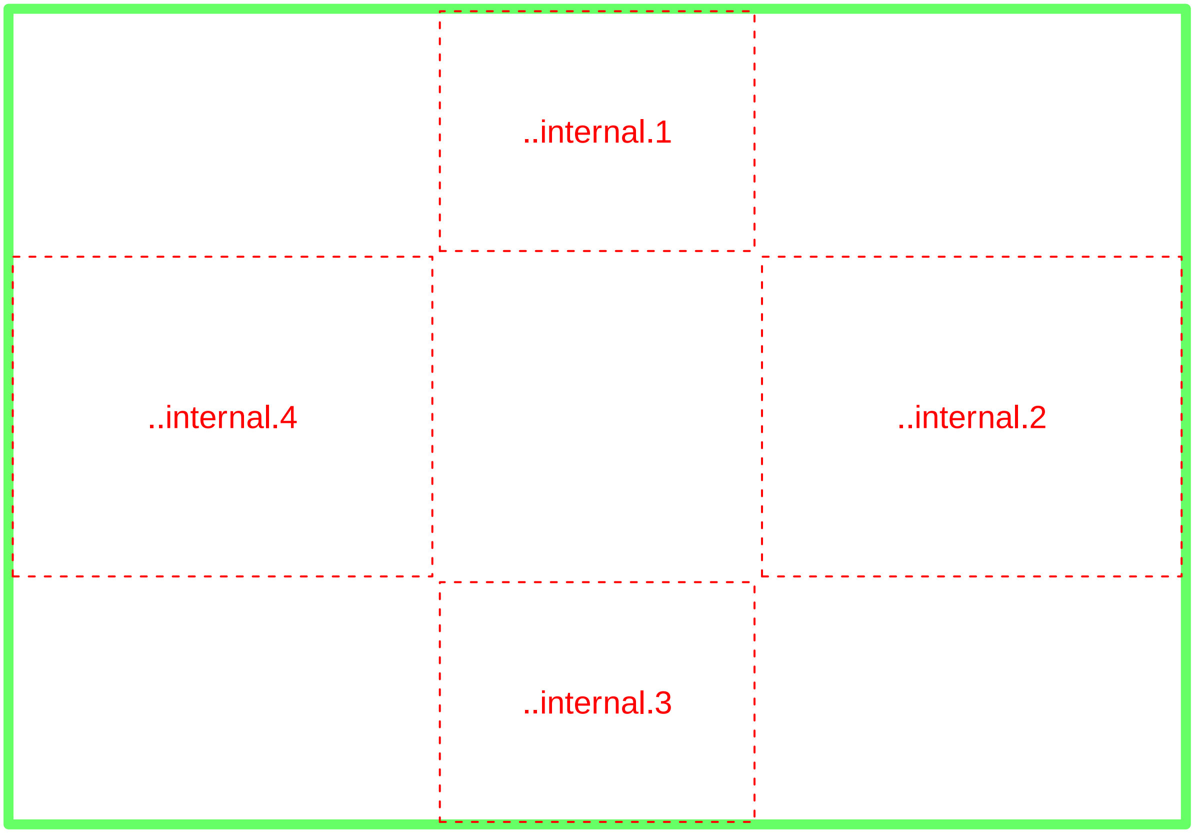
Anchor by corner

The **BottomRightOf**, **BottomLeftOf**, **TopLeftOf** and **TopRightOf** are for placing a new object by anchoring to the corner of an existing object.

```
a <- WHeatmap(matrix(1:9,nrow=3)) +
  WHeatmap(matrix(1:16,nrow=4), BottomRightOf(just='topleft')) +
  WHeatmap(matrix(1:9,nrow=3), BottomLeftOf(just='topright')) +
  WHeatmap(matrix(1:16,nrow=4), TopLeftOf(just='bottomright'))
a
```



```
ly(a)
```


..internal.1
..internal.4
..internal.2
..internal.3