Data Visualization with waffle: : **CHEAT SHEET**

Basics

Waffle works for creating square pie charts to visualize the percentage view of a categorical variable. To install it, use **install.packages("waffle",repos="http://cinc.rud.is")** to get the latest version.
Waffle package mainly contains two parts:
geom_waffle: use squares to construct simple Waffle charts
geom_pictogram: use Font Awesome 5 to make isotype pictogram

Install Font for R

Since the version of **Font Awesome 5** is always changing, to maintain consistency, Waffle package has embedded its installation document, use **install_fa_fonts** to see its location.

Install **Font Awesome 5** to your computer, then apply **extrafont::font_import()** to install it to R. For macOS, we might need **remotes::install_version("Rttf2pt1",version="1.3.8")** first to run the import function.

Search glyph

To preview the glyph in Font Awesome 5, Waffle also provide function **fa_list()** which will present a htmlwidget to preview all symbols.

	name	type	glyph
1	500px	brands	
2	accessible-icon	brands	
3	accusoft	brands	

To search for a certain glyph according to its name, also use **fa_grep()**. E.g., **fa_grep("apple")**

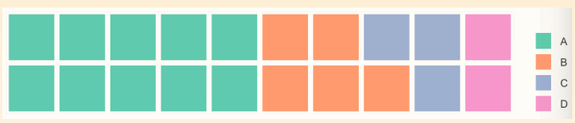
	name	type	glyph
45	apple	brands	
46	apple-alt	solid	

To get several name at the time, use "|". E.g., **fa_grep("apple|rocket")**

Basic Waffle Chart

`waffle()`

```
parts<-data.frame(names = LETTERS[1:4],vals =
c(10,5,3,2))
waffle(parts,rows=2)
```

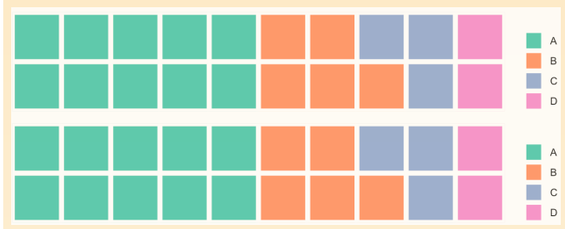


`use_glyph()` to change pictogram

```
waffle(parts,rows=2,use_glyph = 'rocket')
+expand_limits(y=c(0,3))
```

`iron()` for combining graphs

```
iron(waffle(parts,rows=2),waffle(parts,rows=2))
```



`geom_waffle()`

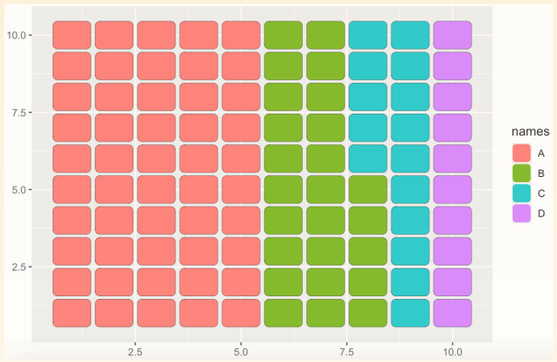
Main function for drawing waffle chart using ggplot2. It has several parameters:

make_proportional: If TRUE, no matter the value of data, it turns the data into 10*10 grids.

radius: use round squares for representing proportion

height, width: change the size of grids hence change the gaps

```
ggplot(parts, aes(fill=names, values=vals))
+geom_waffle(make_proportional =
TRUE,radius=unit(4,
"pt"),height=0.9,width=0.9)
```



`scale_color_manual()`

If we have already set color in **geom_waffle()**,**scale_color_manual()** will set color for the boundries;

If we haven't, **scale_color_manual()** will set color for main area.

It's a good way for highlighting certain category.

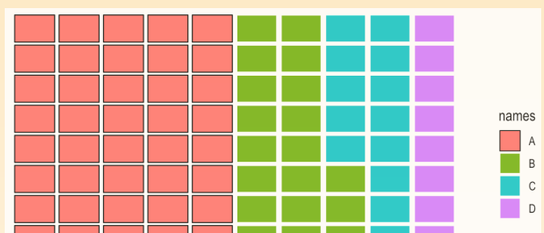
`theme_enhance_waffle()`

This function will remove panel grid, axis text ,axis ticks and axis titles, but it should be written at the last 1 or 2 lines.

`theme_ipsum()`

This function is in **package(hrbrthemes)**. It's a typography-centric package and provide themes such as Robots Condensed. We use the basic one with Arial Narrow.

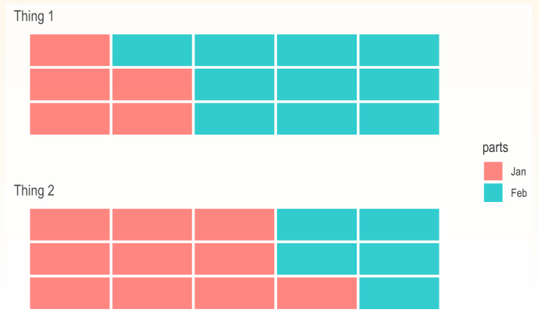
```
ggplot(parts, aes(fill=names, values=vals))
+theme_ipsum(grid='')
+geom_waffle(aes(colour=names),make_proportion
al =
TRUE,height=0.9,width=0.9,size=0.4)+scale_colo
ur_manual(values =
c("black","white","white","white"))
+theme_enhance_waffle()
```



`facet_wrap()`

```
tibble(
  parts = factor(rep(month.abb[1:2], 2),
levels=month.abb[1:3]),
  values = c(5, 10, 10, 5),
  fct = c(rep("Thing 1", 2), rep("Thing 2",
2))
) -> xdf

ggplot(xdf, aes(fill=parts, values=values))
+geom_waffle(color = "white", size=1, n_rows
=3)+facet_wrap(~fct,ncol=1)+theme_ipsum(grid='
')+theme_enhance_waffle()
```

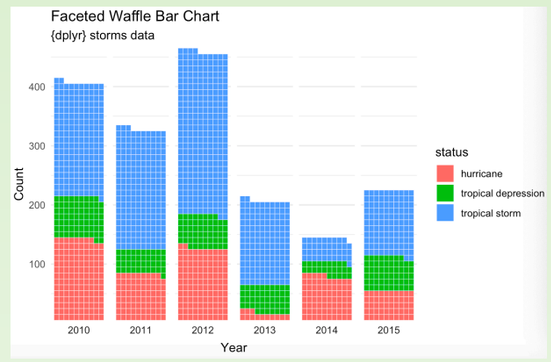


Waffle Bar Chart

To create waffle bar chart, we mainly use **flip** parameter in geom_waffle and **facet_wrap** to make bar chart vertical. Use **scale_x_discrete** for different facet/bar.

```
storms %>% filter(year >= 2010) %>%
count(year, status) -> storms_df

ggplot(storms_df, aes(fill = status, values =
n))+geom_waffle(color = "white", n_rows = 10,
flip = TRUE)+facet_wrap(~year, nrow = 1,
strip.position = "bottom")+theme_minimal()
+scale_x_discrete()+scale_y_continuous(labels
= function(x) x * 10,expand=c(0,0))+labs(
    title = "Faceted Waffle Bar Chart",
    subtitle = "{dplyr} storms data",
    x = "Year",y = "Count")
```



Pictograms

`geom_pictogram()`

geom_pictogram() and **scale_label_pictogram()** are combined use. **theme_void()** and **theme_minimal()** are recommended which make the graph clearer and brighter.

```
tibble(food_group = factor(c("Fruit",
"Sandwiches", "Pizza"),levels=c("Fruit",
"Sandwiches", "Pizza")),consumption = c(5, 20,
52)) -> xdf

ggplot(xdf, aes(label = food_group, values =
consumption, color = food_group)) +
    geom_pictogram(n_rows = 10,
make_proportional = TRUE) +
    scale_color_manual(
        name = NULL,
        values = c(
            Fruit = "#a40000",
            Sandwiches = "#c68958",
            Pizza = "#ae6056")) +
    scale_label_pictogram(
        name = NULL,
        values = c(
            Fruit = "apple-alt",
            Sandwiches = "bread-slice",
            Pizza = "pizza-slice")) +
    theme_void() +
    theme_enhance_waffle() +
    theme(legend.key.height = unit(2.25,
"line")) +
    theme(legend.text = element_text(size =
10, hjust = 0, vjust = 0.75))
```