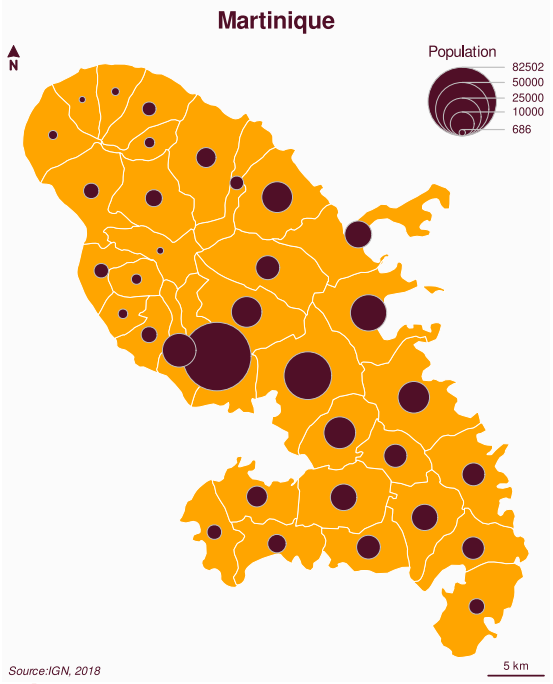
Thematic maps with mapsf :: **CHEAT SHEET**

Base map

```
library(mapsf)
# import the sample dataset
mtq ← mf_get_mtq()
# display a base map
mf_map(mtq, col = "orange", border = "white")
# display symbology
mf_map(mtq, var = "POP", type  = "prop",
       leg_title = "Population", add = TRUE)
# map layout
mf_layout(title = "Martinique",
          credits = "Source:IGN, 2018")
```



Colors

`mapsf` can use color palettes from `hcl.colors()`
`mf_get_pal()` is useful to create well-balanced asymetric diverging palettes

```
mf_get_pal(n = c(7, 2), pal = c("Burg", "Mint"))
```

Symbology

x should be an `sf` object of polygons, lines, or points.

Choropleth (ratios)
```
mf_map(x = mtq, var = "var", type = "choro",
       breaks = "quantile")
```

Typology (categories)
```
mf_map(x = mtq, var = "var", type = "typo")
```

Proportional Symbols (stocks)
```
mf_map(x = mtq, var = "var", type = "prop",
       inches = .3, symbol = "circle")
```

Graduated Symbols (stocks)
```
mf_map(x = mtq, var = "var", type = "grad",
       pch = 24)
```

Symbols (categories)
```
mf_map(x = mtq, var = "var", type = "symb",
       pch = 21:23)
```

Choropleth Proportional Symbols (stocks - ratios)
```
mf_map(x = mtq, var = c("var1", "var2"),
       type = "prop_choro")
```

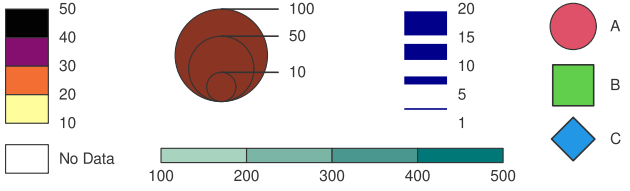
Colorized Proportional Symbols (stocks - categ.)
```
mf_map(x = mtq, var = c("var1", "var2"),
       type = "prop_typo")
```

Choropleth Symbols (stocks - categ.)
```
mf_map(x = mtq, var = c("var1", "var2"),
       type = "symb_choro")
```

Raster (single/multi band, continuous, classes, intervals)
```
mf_raster(x = raster, type = "continuous")
```

Legends

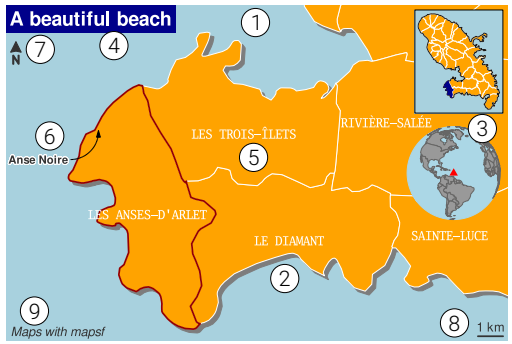
Default legends are plotted with each map.



One can use `mf_legend()` or the `maplegend` package for more customization options.

Map layout

Customization of tha layout design



① Set a map theme
```
mf_theme(background = "#aad3df",
         mar = c(0, 0, 0, 0))
```

② Create a shadow effect
```
mf_shadow( ... )
```

③ Create a map inset
```
mf_inset_on(mtq)
mf_map(mtq)
mf_inset_off()
```
Use `mf_inset_on(x = "worldmap")` and `mf_worldmap()` to display a globe

④ Display a title
```
mf_title( ... )
```

⑤ Display labels
```
mf_label( ... )
```

⑥ Display an annotation
```
mf_annotation( ... )
```

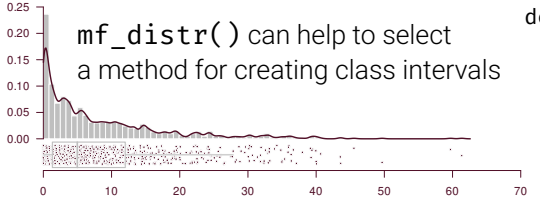
⑦ North arrow
```
mf_arrow( ... )
```

⑧ Scale bar
```
mf_scale( ... )
```

⑨ Credits
```
mf_credits( ... )
```

Class intervals

`mf_get_breaks()` uses `classInt` and additional methods (e.g., "Q6", "geom", "arith", "msd" or "ckmeans") to classify continuous variables.



`mf_distr()` can help to select a method for creating class intervals

Export maps

`mf_png()` and `mf_svg()` export maps in PNG and SVG formats. The width/height ratio of the map matches the one of a spatial object.

```
mf_svg(x = mtq, width = 3,
       filename = "map.svg")
mf_map(mtq)
dev.off()
```

width : 3 in.
(inferred) height : 3.83 in.

More ?

Vignettes on the website:

> How to use themes
> How to export maps
> How to create inset maps
> How to create faceted maps
> how to use a custom font family