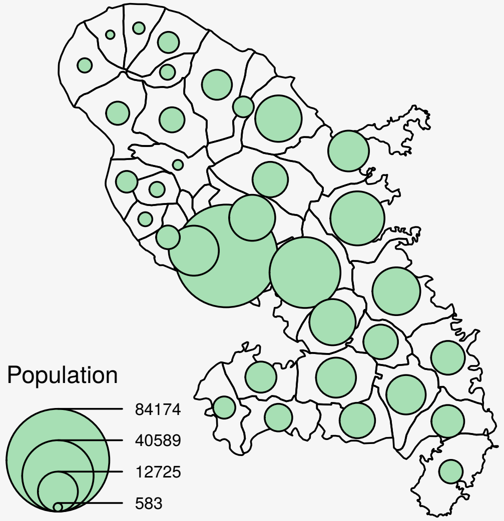
Thematic maps with cartography : : **CHEAT SHEET**

Use cartography with spatial objects from sf or sp packages to create thematic maps.

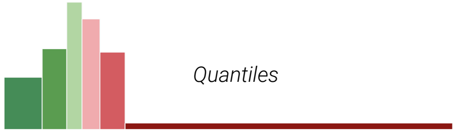
```
library(cartography)
library(sf)
mtq <- st_read("martinique.shp")
plot(st_geometry(mtq))
propSymbolsLayer(x = mtq, var = "P13_POP",
 legend.title.txt = "Population",
 col = "#a7dfb4")
```



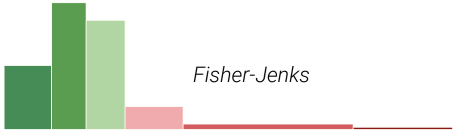
Classification

Available methods are: quantile, equal, q6, fisher-jenks, mean-sd, sd, geometric progression...

```
bks1 <- getBreaks(v = var, nclass = 6,
 method = "quantile")
bks2 <- getBreaks(v = var, nclass = 6,
 method = "fisher-jenks")
pal <- carto.pal("green.pal",3, "wine.pal", 3)
hist(var, breaks = bks1, col = pal)
```



```
hist(var, breaks = bks2, col = pal)
```



Symbology

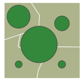
In most functions the x argument should be an sf object. sp objects are handled through spdf and df arguments.

Choropleth

```
choroLayer(x = mtq, var = "myvar",
 method = "quantile", nclass = 8)
```

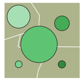
Typology

```
typoLayer(x = mtq, var = "myvar")
```



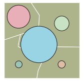
Proportional Symbols

```
propSymbolsLayer(x = mtq, var = "myvar",
 inches = 0.1, symbols = "circle")
```



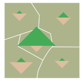
Colorized Proportional Symbols (relative data)

```
propSymbolsChoroLayer(x = mtq, var = "myvar",
 var2 = "myvar2")
```



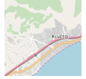
Colorized Proportional Symbols (qualitative data)

```
propSymbolsTypoLayer(x = mtq, var = "myvar",
 var2 = "myvar2")
```



Double Proportional Symbols

```
propTrianglesLayer(x = mtq, var1 = "myvar",
 var2 = "myvar2")
```

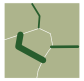


OpenStreetMap Basemap (see rosm package)

```
tiles <- getTiles(x = mtq, type = "osm")
tilesLayer(tiles)
```

Isopleth (see SpatialPosition package)

```
smoothLayer(x = mtq, var = "myvar",
 typefct = "exponential", span = 500,
 beta = 2)
```

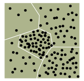


Discontinuities

```
discLayer(x = mtq.borders, df = mtq,
 var = "myvar", threshold = 0.5
```

Flows

```
propLinkLayer(x = mtq_link, df = mtq_df,
 var = "fij")
```



Dot Density

```
dotDensityLayer(x = mtq, var = "myvar")
```

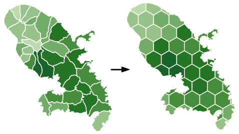
Labels

```
labelLayer(x = mtq, txt = "myvar",
 halo = TRUE, overlap = FALSE)
```

Transformations

Polygons to Grid

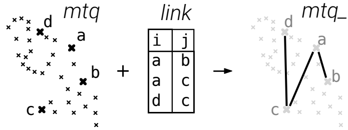
```
mtq_grid <- getGridLayer(x = mtq, cellsize = 3.6e+07,
  type = "hexagonal", var = "myvar")
```



Grids layers can be used by choroLayer() or propSymbolsLayer().

Points to Links

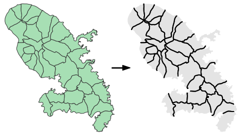
```
mtq_link <- getLinkLayer(x = mtq, df = link)
```



Links layers can be used by *LinkLayer().

Polygons to Borders

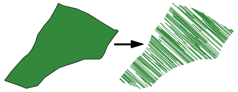
```
mtq_border <- getBorders(x = mtq)
```



Borders layers can be used by discLayer() function

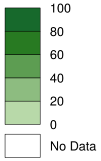
Polygons to Pencil Lines

```
mtq_pen <- getPencilLayer(x = mtq)
```



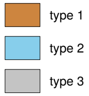
Legends

legendChoro()



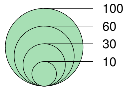
```
legendChoro(pos = "topleft",
 title.txt = "legendChoro()",
 breaks = c(0,20,40,60,80,100),
 col = carto.pal("green.pal", 5),
 nodata = TRUE, nodata.txt = "No Data")
```

legendTypo()



```
legendTypo(title.txt = "legendTypo()",
 col = c("peru", "skyblue", "gray77"),
 categ = c("type 1", "type 2", "type 3"),
 nodata = FALSE)
```

legendCirclesSymbols()



```
legendCirclesSymbols(var = c(10,100),
 title.txt = "legendCirclesSymbols()",
 col = "#a7dfb4ff", inches = 0.3)
```

See also legendSquaresSymbols(), legendBarsSymbols(), legendGradLines(), legendPropLines() and legendPropTriangles().

Map Layout

North Arrow:

```
north(pos = "topright")
```

Scale Bar:

```
barscale(size = 5)
```

Full Layout:

```
layoutLayer(
 title = "Martinique",
 tabtitle = TRUE,
 frame = TRUE,
 author = "Author",
 sources = "Sources",
 north = TRUE,
 scale = 5)
```

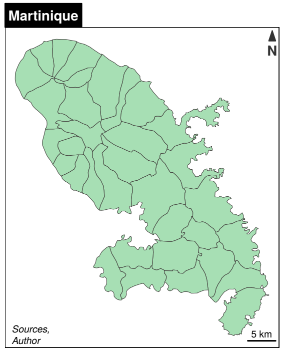
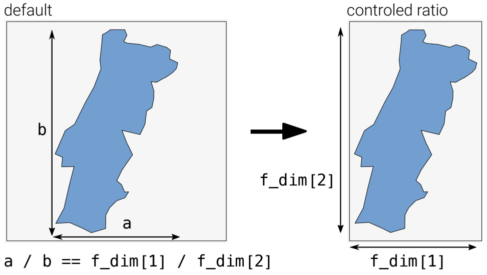


Figure Dimensions

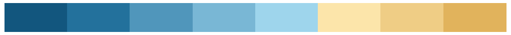
Get figure dimensions based on the dimension ratio of a spatial object, figure margins and output resolution.

```
f_dim <- getFigDim(x = sf_obj, width = 500,
  mar = c(0,0,0,0))
png("fig.png", width = 500, height = f_dim[2])
par(mar = c(0,0,0,0))
plot(sf_obj, col = "#729fcf")
dev.off()
```

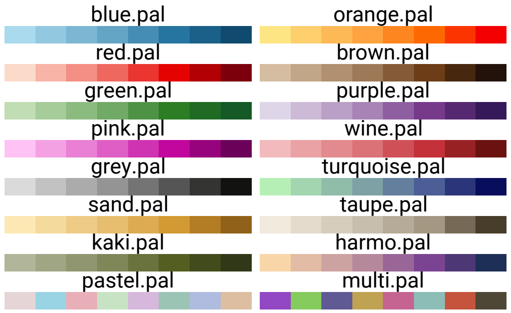


Color Palettes

```
carto.pal(pal1 = "blue.pal", n1 = 5,
 pal2 = sand.pal, n2 = 3)
```



```
display.carto.all(n = 8)
```



cartography *- https://github.com/riatelab/cartography - CC BY SA Timothée Giraud - 2018-07*