

Cartographie avec R

Timothée Giraud Hugues Pecout

11 mars 2024

Table des matières

Préambule	3
I Cartographie thématique	4
1 mapsf	5
1.1 Présentation	5
1.2 Documentation	7
Exercice	7
2 La sémiologie graphique	8
2.1 Principes généraux	8
2.2 Ressources	9
Exercice	9
3 Les types de cartes	11
3.1 Les données d'exemple	11
3.2 Afficher un fond de carte	11
3.3 Carte de symboles proportionnels	12
3.4 Carte choroplèthe	14
3.4.1 Les palettes de couleurs	16
3.4.2 Discrétisations	18
3.5 Carte de typologie	20
3.6 Carte de stocks et de ratios	23
3.7 Carte de stocks et de catégories	24
Exercice	26
4 La mise en page	27
4.1 Les données d'exemple	27
4.2 Les thèmes	27
4.2.1 Utiliser un thème prédéfini	27
4.2.2 Modifier un thème existant	28
4.2.3 Créer un thème	28
4.3 Les titres	31
4.4 Flèche d'orientation et échelle	33
4.5 Crédits	35

4.6	Habillage complet	36
4.7	Annotations	37
4.8	Étiquettes	38
4.9	Les légendes	40
4.10	Centrer la carte sur une région	41
4.11	Afficher plusieurs cartes sur la même figure	42
4.12	Export des cartes	44
4.13	Ajouter une image sur une carte	47
4.14	Placer précisément un élément sur la carte	48
4.15	Ajouter un ombrage à une couche	49
4.16	Création de cartons	49
	Exercice	52
II	Cartographie 3D et Anamorphoses	54
5	Les cartes en 3D	55
5.1	linemap	55
5.2	Relief Tanaka	57
6	Les cartogrammes	59
6.1	Les cartogrammes de Dorling	59
6.2	Les cartogrammes non continus	61
6.3	Les cartogrammes continus	62
6.4	Forces et faiblesses des cartogrammes	64
	References	65
	Appendices	66
A	Les données du projet	66

Préambule

L'objectif de ce document est de fournir un support à l'apprentissage de la cartographie thématique avec R.

Comment utiliser le manuel

Les données utilisées dans ce document sont stockées dans un projet RStudio. Vous devez le télécharger puis le décompresser sur votre machine. Il vous sera ensuite possible de tester l'ensemble des manipulations proposées dans ce document au sein du projet **geodata**.

[Télécharger le projet](#)

Contribution et feedback

Vous pouvez nous envoyer vos remarques et suggestions en [postant une issue](#) sur le [dépôt GitHub](#) de ce document.

Contexte

Ce manuel a été initialement conçu pour accompagner le cours “Cartographie avec R” du Master 2 Géomatique, géodécisionnel, géomarketing et multimédia (G2M) de l'Université Paris 8 Vincennes - Saint-Denis. Une version PDF est disponible ici : [📄](#).

Un deuxième manuel centré sur la géomatique est disponible ici : [Géomatique avec R](#).

Pour citer le document :

Giraud, T. et Pecout, H. (2024). Cartographie avec R. <https://doi.org/10.5281/zenodo.5905734>

partie I

Cartographie thématique

1 mapsf

1.1 Présentation

L'écosystème spatial du logiciel libre R est riche, dynamique et mature et plusieurs packages permettent d'importer, de traiter et de représenter les données spatiales. Le package `mapsf` (Giraud, 2023b) s'appuie sur cet écosystème pour intégrer la création de cartes thématiques de qualité dans les chaînes de traitements avec R.

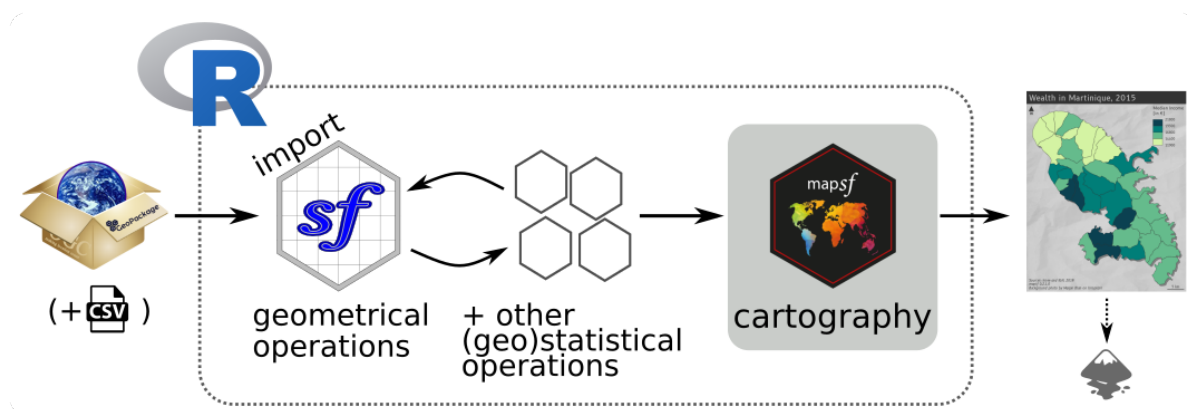


Figure 1.1: Une chaîne de traitements typique

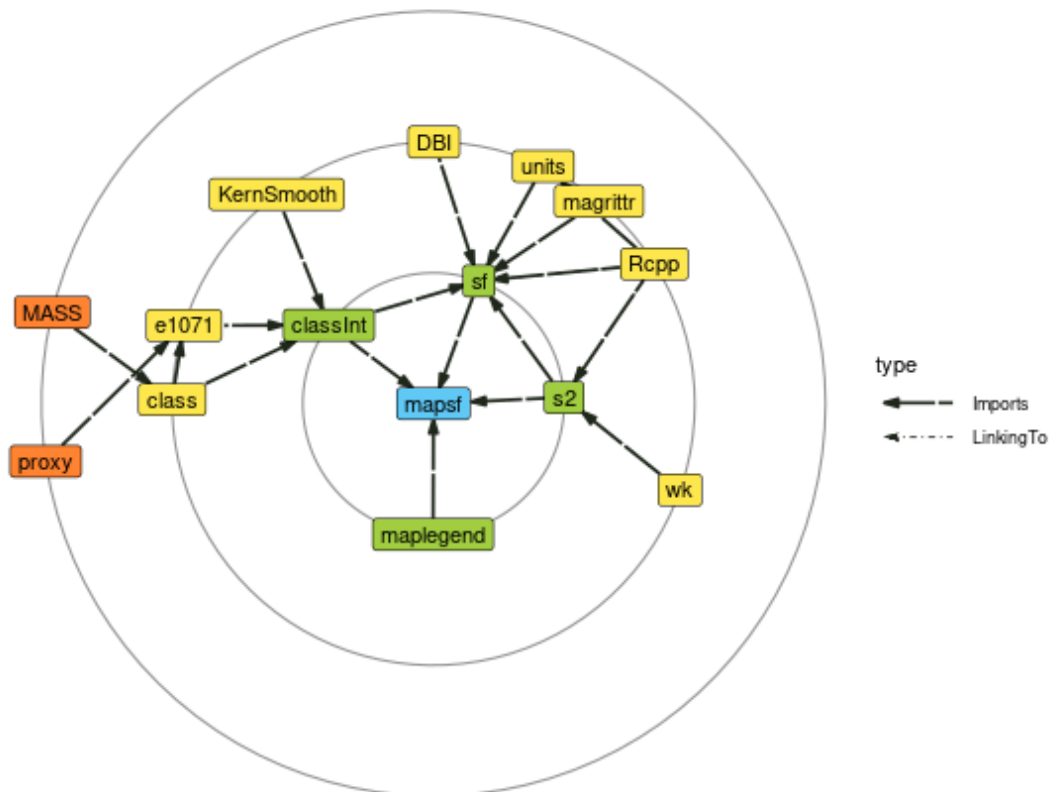
D'autres packages peuvent être utilisés pour réaliser des cartes thématiques.

Le package `ggplot2` (Wickham, 2016), en association avec le package `ggspatial` (Dunnington, 2023), permet par exemple d'afficher des objets spatiaux et de réaliser des cartes thématiques simples.

Le package `tmap` (Tennekes, 2018) est dédié à la création de cartes thématiques, il utilise une syntaxe proche de celle de `ggplot2` (enchaînement d'instructions combinées avec le signe `+`).

La documentation et les tutoriels pour utiliser ces deux packages sont facilement accessibles sur le web.

Dans ce document, nous utiliserons principalement le package `mapsf` dont les fonctionnalités sont assez complètes et la prise en main plutôt simple. De plus, le package est relativement léger.



Plot made with deepdep v0.4.2 on 2023-11-22 13:42:46

Figure 1.2: Un nombre minimal de dépendances bien connues

`mapsf` permet de créer la plupart des types de carte utilisés habituellement en cartographie statistique (cartes choroplèthes, typologies, symboles proportionnels ou gradués...). Pour chaque type de carte, plusieurs paramètres permettent de personnaliser la représentation cartographique. Ces paramètres sont les mêmes que ceux que l'on retrouve dans les logiciels de SIG ou de cartographie usuels (par exemple, le choix des discrétisations et des palettes de couleurs, la modification de de la taille des symboles ou la personnalisation des légendes). Associées aux fonctions de représentation des données d'autres fonctions sont dédiées à l'habillage cartographique (thèmes ou chartes graphiques, légendes, échelles, flèches d'orientation, titre, crédits, annotations...), à la création de cartons ou à l'export des cartes. `mapsf` est le successeur de `cartography` (Giraud et Lambert, 2017), il offre les mêmes fonctionnalités principales tout en étant plus léger et plus ergonomique.

mapsf 0.8.0 Get started Reference **Articles** Changelog

mapsf

Create and integrate thematic maps in your workflow. It allows you to design various cartographic representations (faceted maps, choropleth or typology maps). It also offers several functions to improve the graphic presentation of maps (e.g. scale bar, north arrow, title, labels). **mapsf** maps **sf** objects on base graphics.

Installation

You can install the released version of **mapsf** from [CRAN](#) with:

```
install.packages("mapsf")
```

Alternatively, you can install the development version of **mapsf** from GitHub (dev branch) with:

```
remotes::install_github("rriatelab/mapsf", ref = "dev")
```

Links

- [View on CRAN](#)
- [Browse source code](#)
- [Report a bug](#)

License

[GPL-3](#)

Citation

[Citing mapsf](#)

Developers

Timothée Giraud
Maintainer, author

[More about authors...](#)

Dev status

CRAN 0.7.1 3 months ago
R CMD check [passing](#)
downloads [125 times](#)

Site web : [Usage](#)

[cheat sheet](#) :

Thematic maps with mapsf : CHEAT SHEET

Base map

Create and integrate thematic maps in your workflow.

Symbology

This argument should be an sf object. Input geometries can be polygons, lines or points.

Map Layout

Along with cartographic functions, other functions are dedicated to customize the layout design.

Export Maps

mapsf::export() exports maps in PNG or SVG formats.

Colors

mapsf can use color palettes from RColorBrewer or ggthemes. You can also create your own color palette.

Legends

Default legends are plotted along maps. Customized parameters are available with the legend() function.

1.2 Documentation

Vignettes :

- [Get started](#)
- [How to create faceted maps](#)
- [How to create inset maps](#)
- [How to export maps](#)
- [How to use a custom font family](#)
- [How to use themes](#)

Exercice

1. Installez le package **mapsf**.
2. Quelle est la version du package?
3. Quelles sont les nouveautés par rapport à la version précédente?

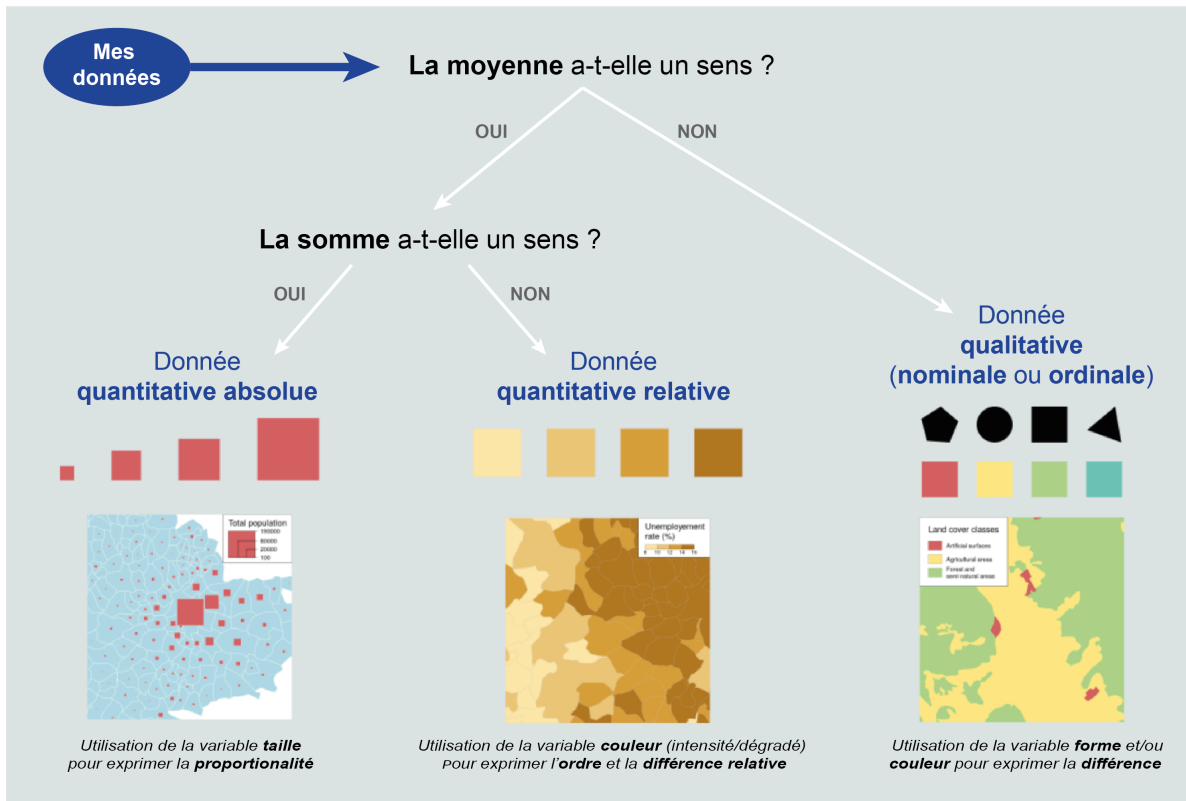
2 La sémiologie graphique

2.1 Principes généraux

La **sémiologie graphique** est, de façon générale, l'ensemble des règles qui régissent la construction d'un système de signes ou langage permettant la traduction graphique d'une information. Depuis la parution de la Sémiologie graphique de Jacques Bertin (1967), elle désigne surtout **un ensemble standardisé des règles et de pratiques inhérentes à la représentation cartographique** au-delà des travaux réalisés par Bertin.

[Wikipedia : Sémiologie graphique](#)

Pour une première approche de la cartographie thématique on peut se référer à la figure suivante qui résume de manière simplifiée la relation entre un type de variable et sa représentation graphique correcte.



Pourquoi on ne représente pas de stocks en aplat !

2.2 Ressources

Voici quelques ouvrages et ressources pour l'apprentissage de la cartographie thématique :

Exercice

Comment représenter les variables suivantes :

- Une population communale
- Un age médian par département
- Un taux de croissance
- Le statut administratif d'une commune (préfecture, sous-préfecture, commune simple...)
- L'espérance de vie par pays

3 Les types de cartes

La fonction `mf_map()` est la fonction centrale du package `mapsf`. Elle permet de réaliser la plupart des représentations usuelles en cartographie. Ces arguments principaux sont :

- `x`, un objet `sf` ;
- `var`, le nom de la variable à représenter ;
- `type`, le type de représentation.

3.1 Les données d'exemple

Les lignes suivantes importent les couches d'information spatiales situées dans le fichier [geopackage lot.gpkg](#).

```
library(sf)
# import des communes du Lot
com <- st_read("data/lot.gpkg", layer = "communes", quiet = TRUE)
# import des départements français
dep <- st_read("data/lot.gpkg", layer = "departements", quiet = TRUE)
# import des restaurants
resto <- st_read("data/lot.gpkg", layer = "restaurants", quiet = TRUE)
# import des routes autour de la commune de Gramat
route <- st_read("data/lot.gpkg", layer = "routes", quiet = TRUE)
```

3.2 Afficher un fond de carte

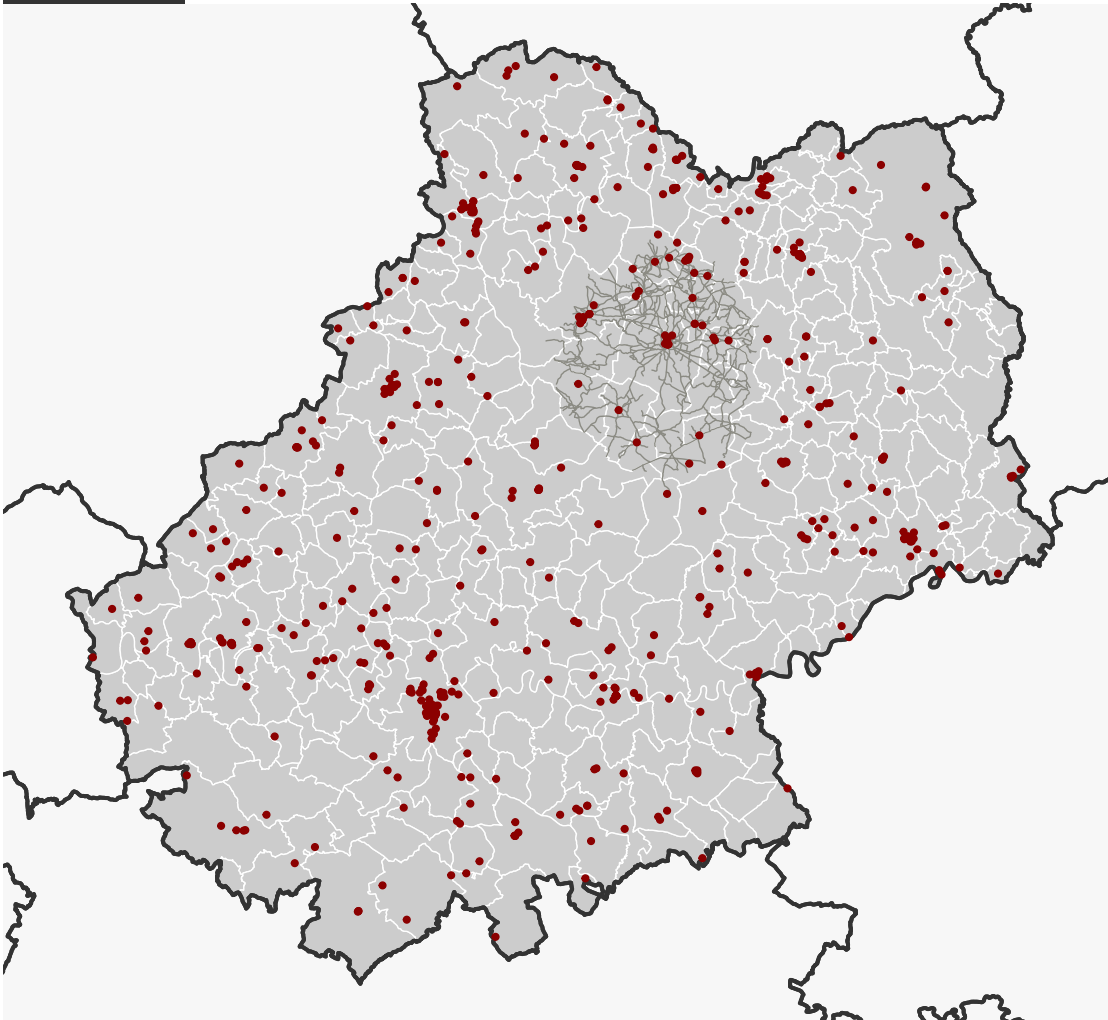
Utilisée sans précision de type, la fonction `mf_map()` affiche simplement les couches spatiales.

```
library(mapsf)
# Communes (polygones)
mf_map(x = com, border = "white")
# Départements (polygones, fond transparent)
mf_map(x = dep, lwd = 2, col = NA, add = TRUE)
# Routes (polylignes)
```



```
mf_map(x = route, lwd = .7, col = "ivory4", add = TRUE)
# Restaurants (points)
mf_map(x = resto, pch = 20, cex = .7, col = "darkred", add = TRUE)
# Ajout d'un titre
mf_title(txt = "Le Lot (46)")
```

Le Lot (46)



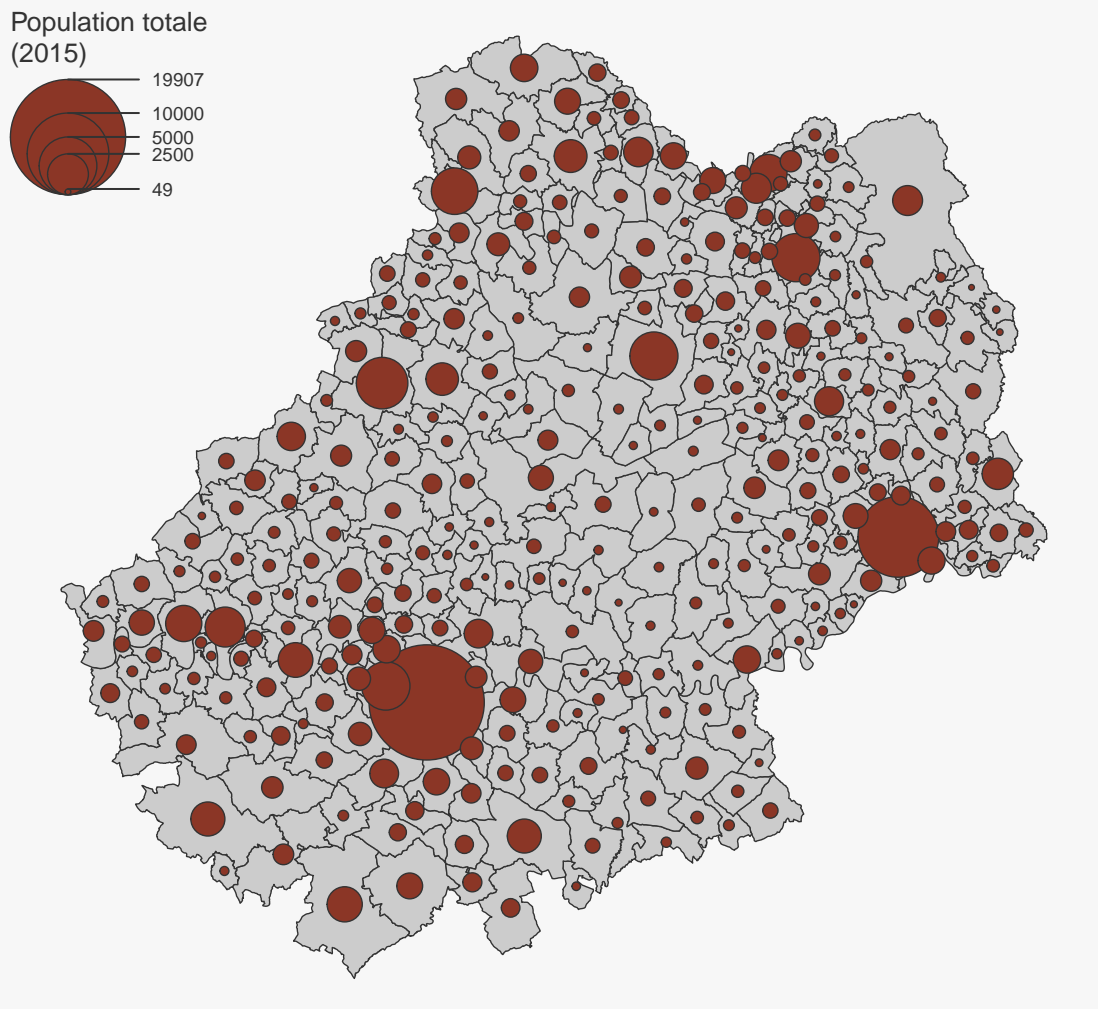
3.3 Carte de symboles proportionnels

Les cartes de symboles proportionnels sont utilisées pour représenter les variables de stocks (variables quantitatives absolues, la somme et la moyenne ont un sens). La fonction `mf_map(...,`

`type = "prop")` propose cette représentation.

```
# Communes
mf_map(x = com)
# Symboles proportionnels
mf_map(
  x = com,
  var = "POPULATION",
  type = "prop",
  leg_title = "Population totale\n(2015)"
)
# Titre
mf_title("Distribution de la population dans le Lot")
```

Distribution de la population dans le Lot



3.4 Carte choroplèthe

Les cartes choroplèthes sont utilisées pour représenter les variables de ratios (variables quantitatives relatives, la moyenne a un sens, la somme n'a pas de sens).

La fonction `mf_map(..., type = "choro")` permet de créer des cartes choroplètes. Les arguments `nbreaks` et `breaks` servent à paramétrer les discrétisations.

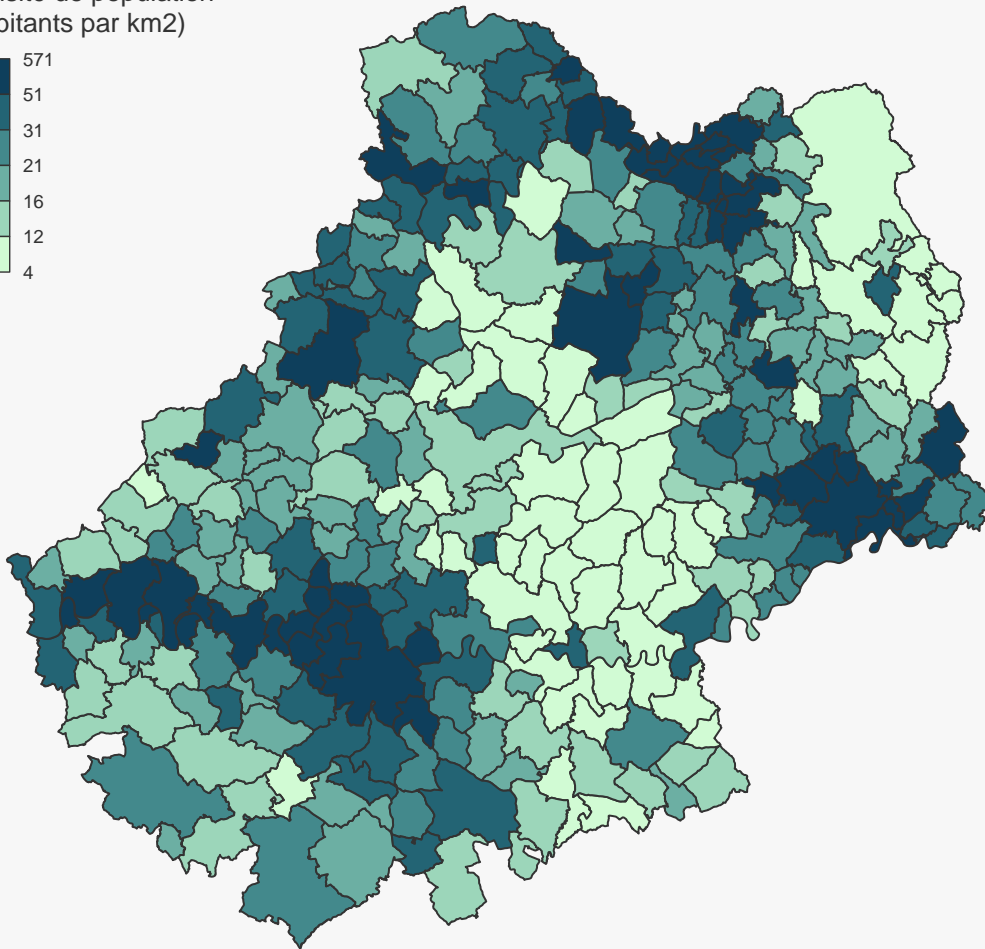
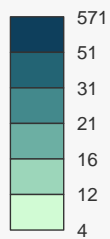
```
# Densité de population (hab./km2) en utilisant la fonction sf::st_area()  
com$DENS <- 1e6 * com$POP / as.numeric(st_area(com))  
mf_map(  

```

```
x = com,  
var = "DENS",  
type = "choro",  
breaks = "quantile",  
nbreaks = 6,  
pal = "Dark Mint",  
lwd = 1,  
leg_title = "Densité de population\n(habitants par km2)",  
leg_val_rnd = 0  
)  
mf_title("Distribution de la population dans le Lot (2018)")
```

Distribution de la population dans le Lot (2018)

Densité de population
(habitants par km²)

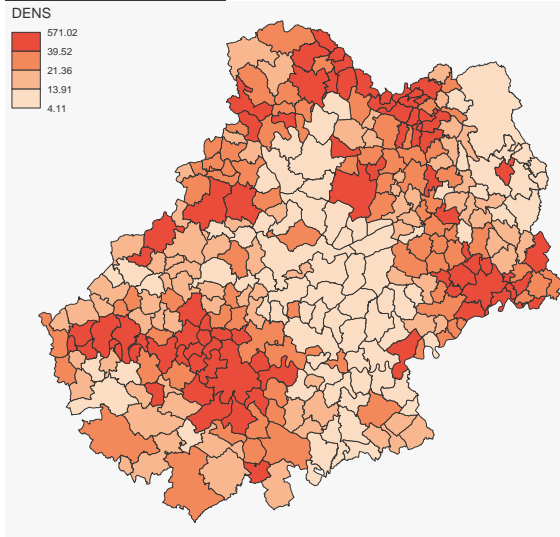


3.4.1 Les palettes de couleurs

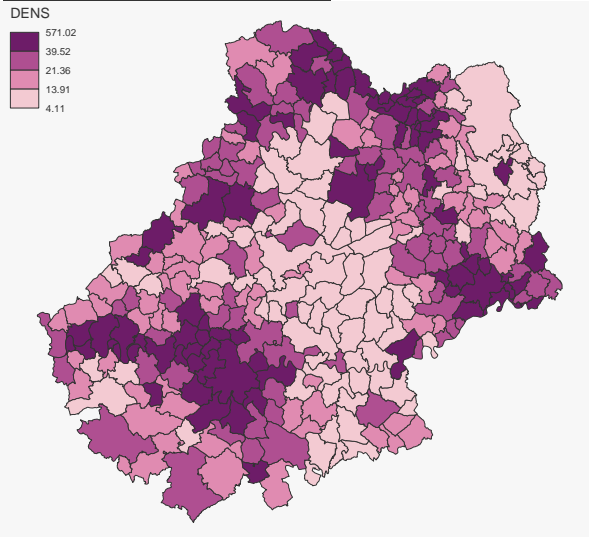
L'argument `pal` de `mf_map()` est dédié au choix d'une palette de couleur. `pal` peut être le nom d'une palette de couleur fournie par R ou alors un vecteur de couleurs définies par leurs codes hexadécimaux.

```
mf_map(x = com, var = "DENS", type = "choro",
       breaks = "quantile", nbreaks = 4,
       pal = "Peach")
mf_title("Palette définie par son nom")
ma_palette <- c("#F3CAD2", "#E08BB1", "#AF4F91", "#6D1C68")
mf_map(x = com, var = "DENS", type = "choro",
       breaks = "quantile", nbreaks = 4,
       pal = ma_palette)
mf_title("Palette définie par un vecteur de couleurs")
```

Palette définie par son nom



Palette définie par un vecteur de couleurs



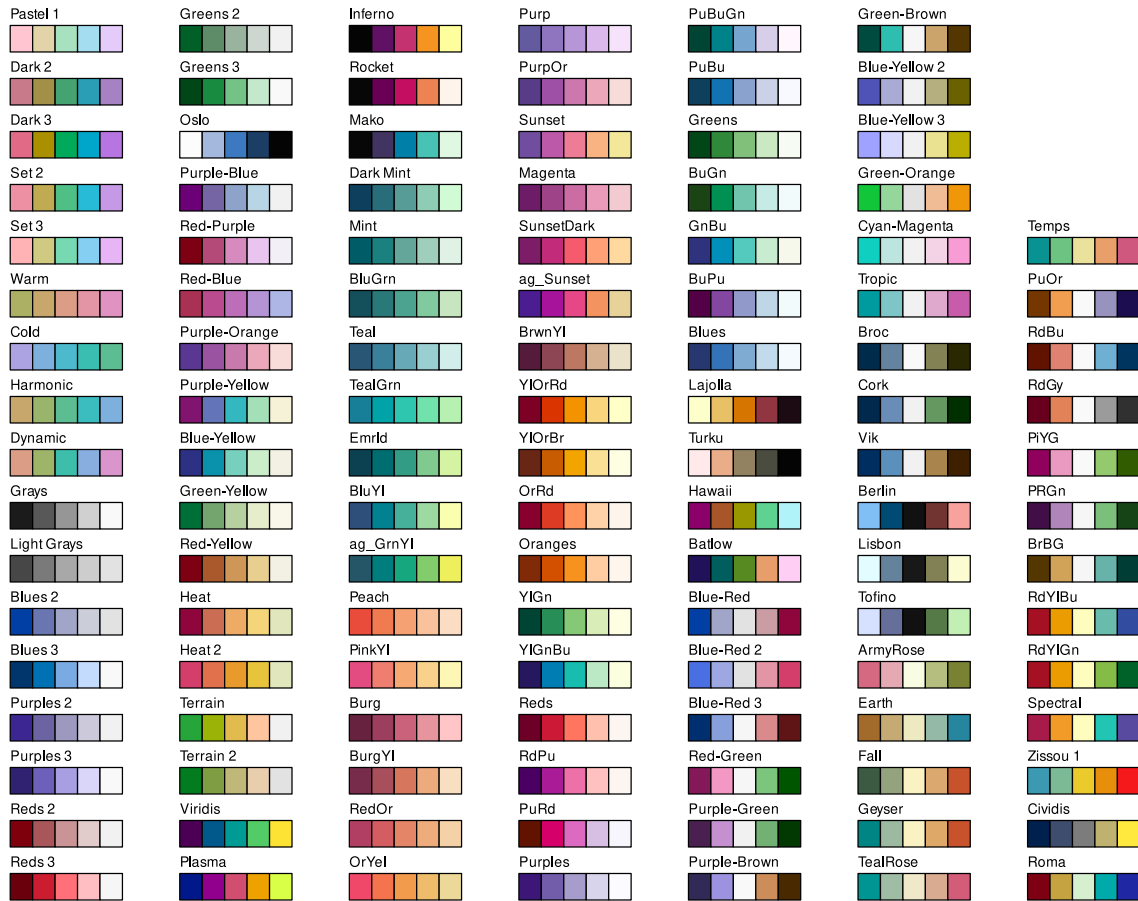


Figure 3.1: Les palettes de R

Les fonctions `hcl.colors()` et `mf_get_pal()` permettent de construire des palettes de couleurs. `mf_get_pal()` est surtout utile pour créer des palettes divergentes asymétriques équilibrées.

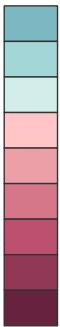
```
pal_equilibre <- mf_get_pal(n = c(6,3), palette = c("Burg", "Teal"))
pal_non_equi <- c(hcl.colors(6, "Burg"), hcl.colors(3,"Teal", rev = TRUE))
library(maplegend)
opar <- par(mar = c(0,0,0,0))
plot.new()
```

```

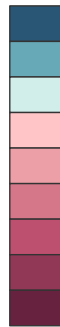
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(type = "choro", val = rep("", 10),
    pal = pal_equilibre, pos = "topleft",
    title = "Palette équilibrée")
leg(type = "choro", val = rep("", 10),
    pal = pal_non_equi, pos = "top",
    title = "Palette non équilibrée")
par(opar)

```

Palette équilibrée



Palette non équilibrée

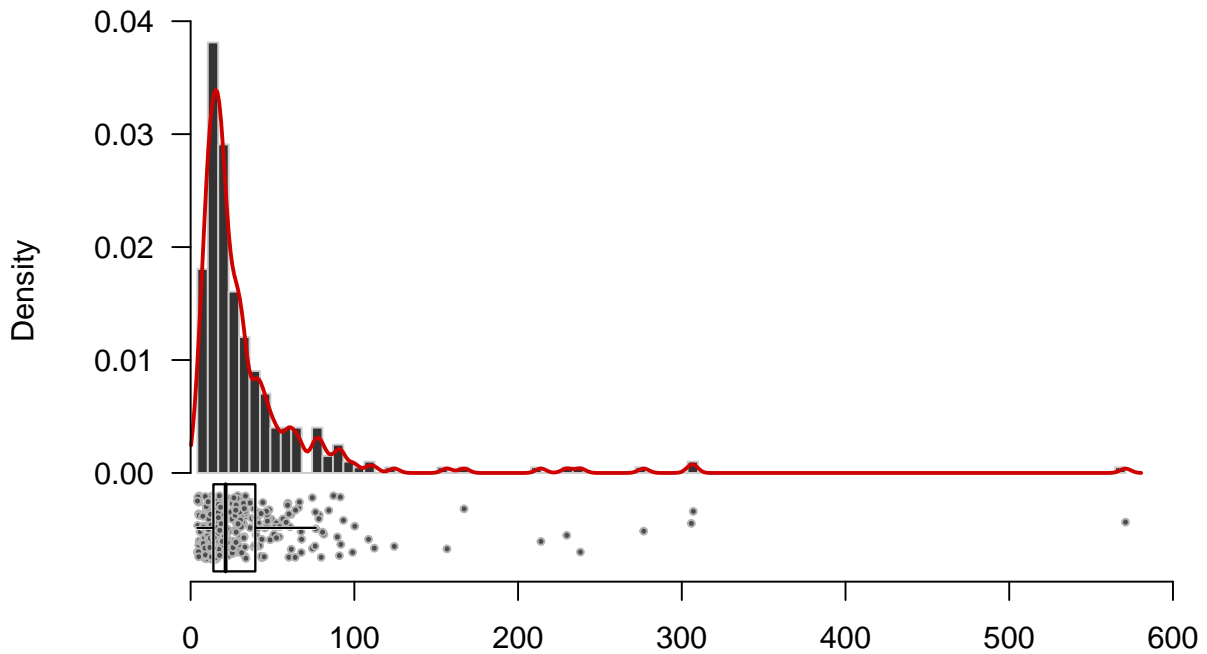


3.4.2 Discrétisations

Avant de réaliser une carte choroplèthe on doit d'abord étudier la distribution statistique de la variable que l'on souhaite cartographier. La fonction `mf_distr()` permet de visualiser les distributions.

```
mf_distr(com$DENS)
```

Distribution



Cette distribution est très dissymétrique à gauche.

La fonction `mf_get_breaks()` met à disposition les méthodes de discrétisations de variables classique : quantiles, moyenne/écart-type, amplitudes égales, moyennes emboîtées, Fisher-Jenks, géométrique...

Ici, l'utilisation de la méthode "geom" est assez appropriée.

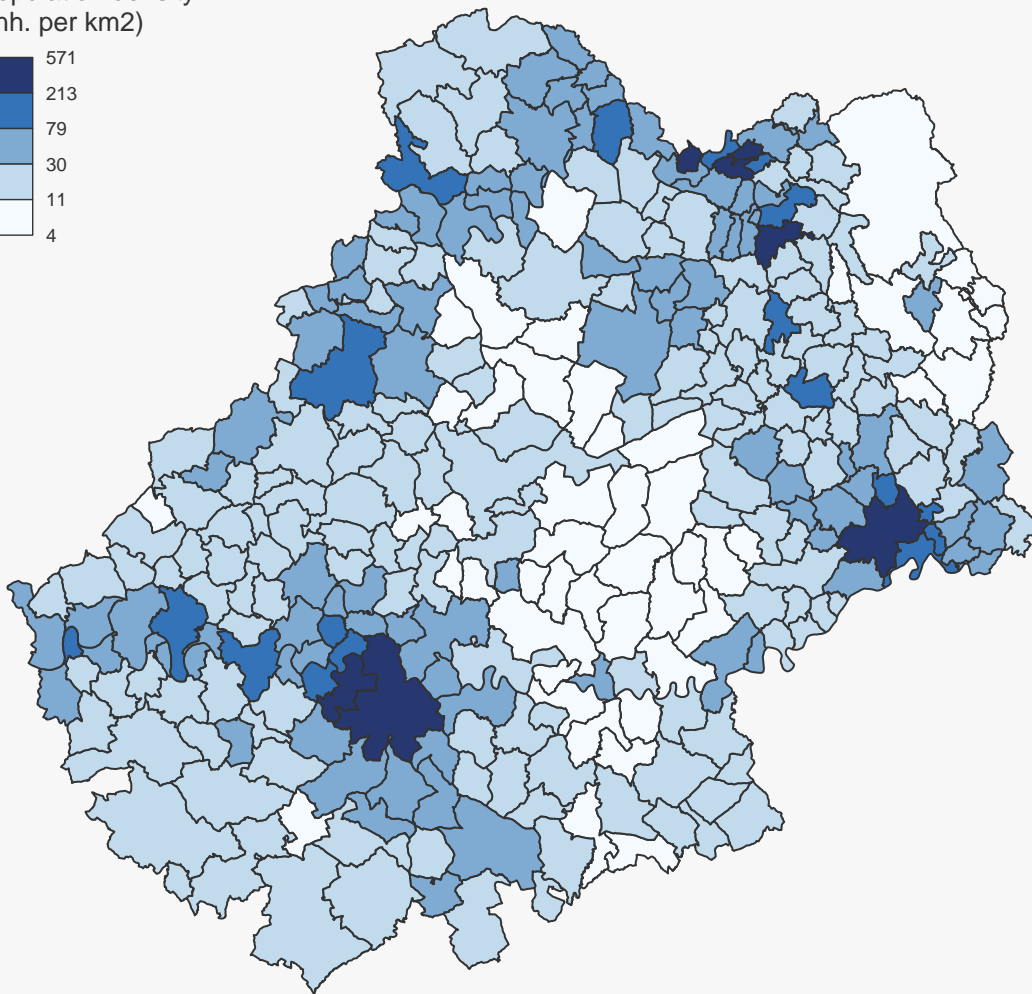
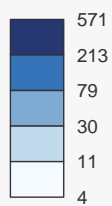
```
bks <- mf_get_breaks(x = com$DENS, nbreaks = 5, breaks = "geom")
bks
```

```
#> [1] 4.113775 11.033758 29.594181 79.376000 212.898252 571.024813
```

```
mf_map(
  x = com, var = "DENS", type = "choro",
  breaks = bks, pal = "Blues",
  lwd = 1,
  leg_title = "Population density\n(inh. per km2)",
  leg_val_rnd = 0
)
mf_title("Distribution de la population dans le Lot (2018)")
```


Distribution de la population dans le Lot (2018)

Population density
(inh. per km2)



3.5 Carte de typologie

Les cartes de typologies sont utilisées pour représenter les variables qualitatives. La fonction `mf_map(..., type = "typo")` propose cette représentation.

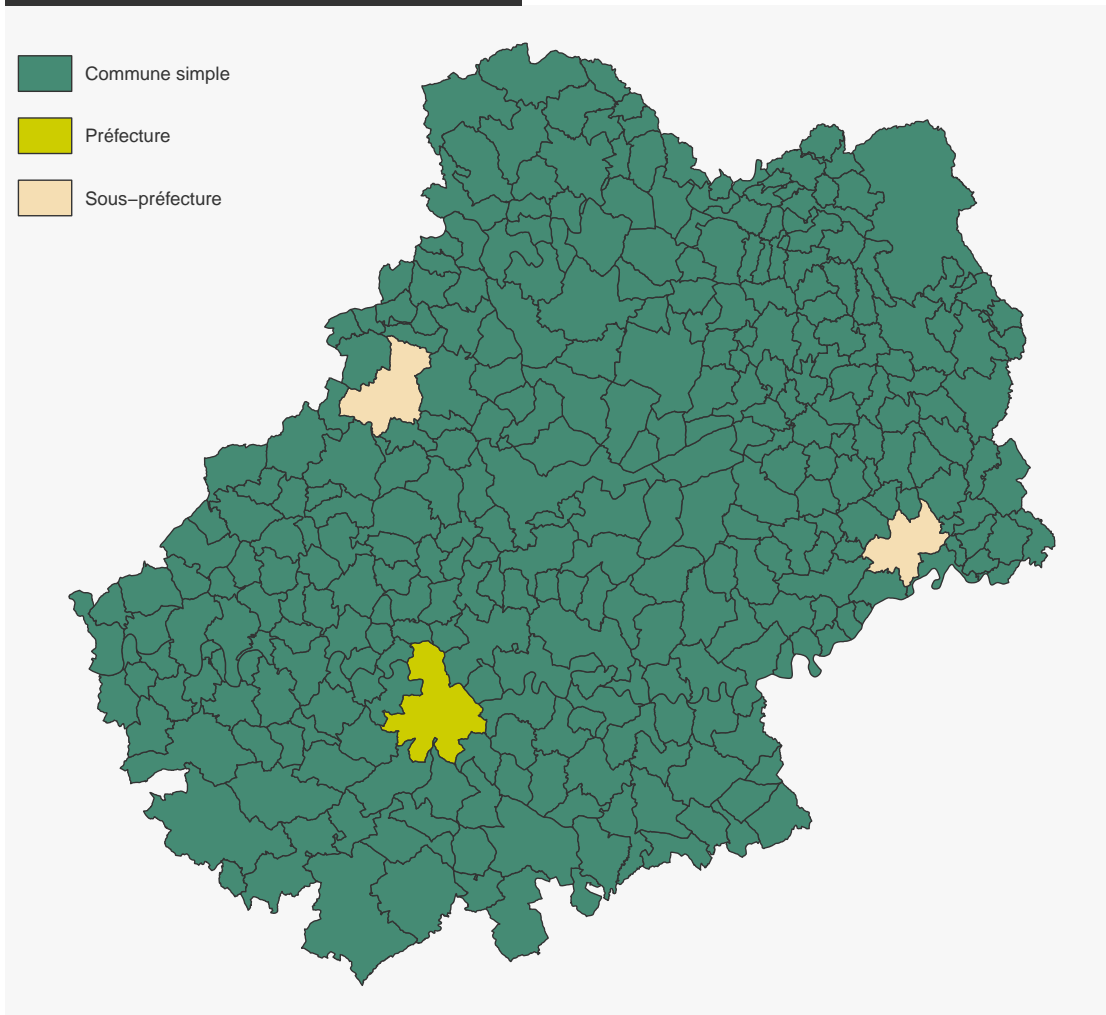
```
mf_map(  
  x = com,  
  var="STATUT",  
  type = "typo",
```

```

pal = c("aquamarine4", "yellow3","wheat"),
lwd = .7,
leg_title = ""
)
mf_title("Statut administratif des communes")

```

Statut administratif des communes



L'argument `val_order` sert à ordonner les modalités dans la légende.

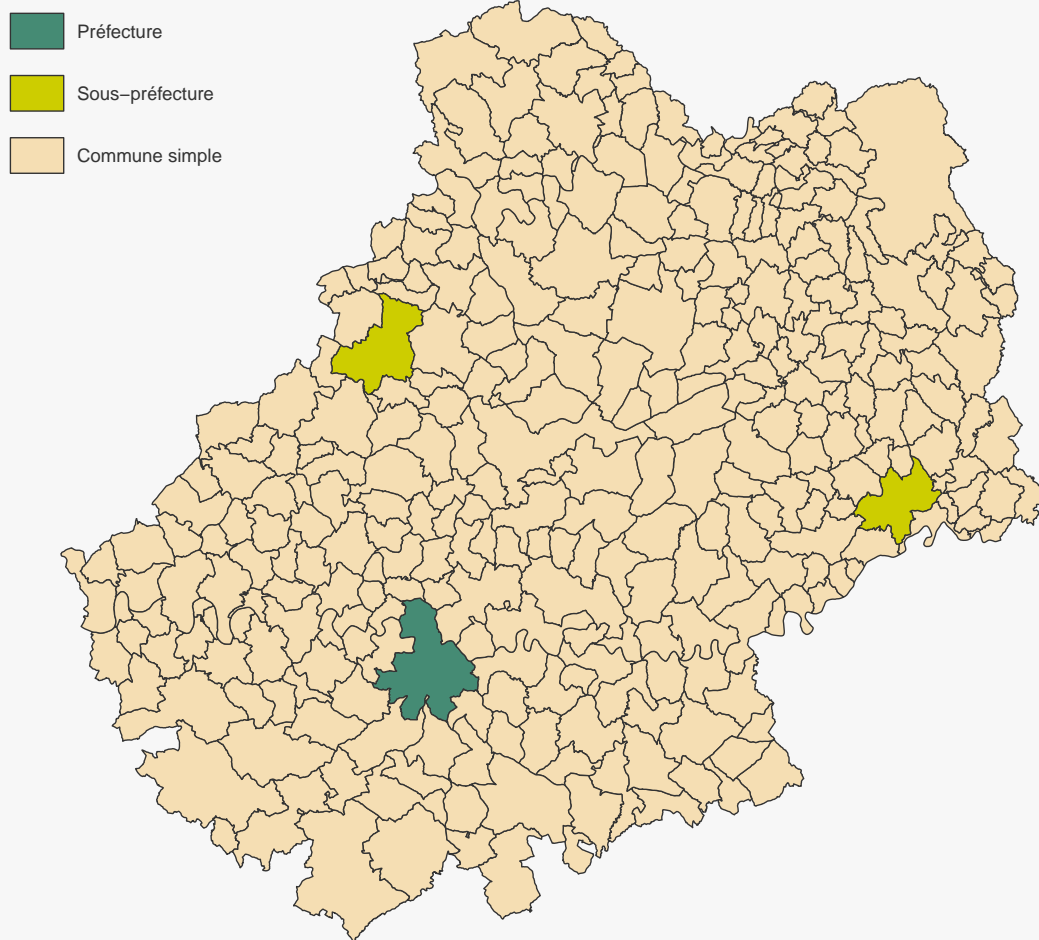
```

mf_map(
  x = com,
  var="STATUT",
  type = "typo",

```

```
pal = c("aquamarine4", "yellow3","wheat"),
val_order = c("Préfecture", "Sous-préfecture", "Commune simple"),
lwd = .7,
leg_title = ""
)
mf_title("Statut administratif des communes")
```

Statut administratif des communes

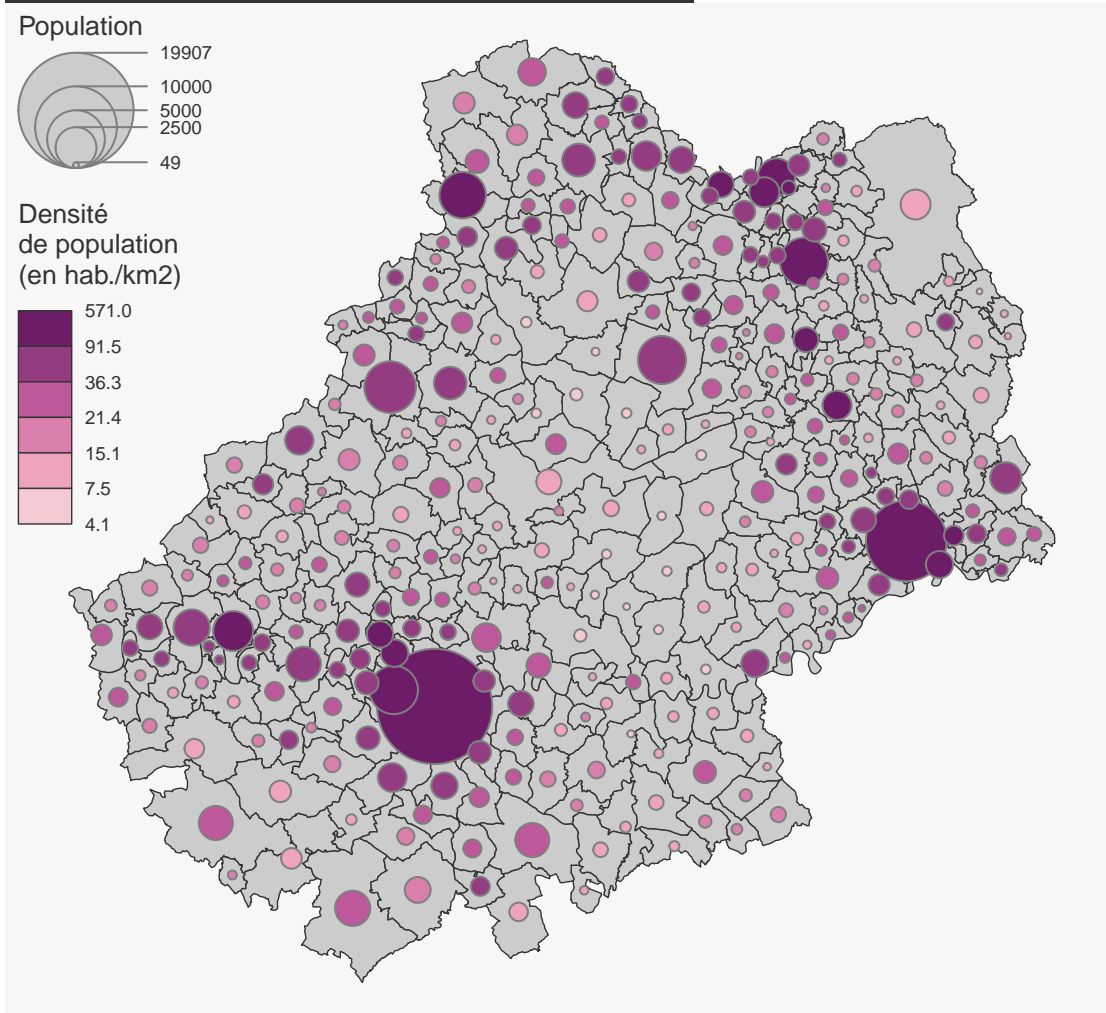


3.6 Carte de stocks et de ratios

La fonction `mf_map(..., var = c("var1", "var2"), type = "prop_choro")` représente des symboles proportionnels dont les surfaces sont proportionnelles aux valeurs d'une variable et dont la couleur repose sur la discrétisation d'une seconde variable. La fonction utilise les arguments des fonctions `mf_map(..., type = "prop")` et `mf_map(..., type = "choro")`.

```
mf_map(x = com)
mf_map(
  x = com,
  var = c("POPULATION", "DENS"),
  type = "prop_choro",
  border = "grey50",
  lwd = 1,
  leg_title = c("Population", "Densité\nde population\n(en hab./km2)"),
  breaks = "q6",
  pal = "Magenta",
  leg_val_rnd = c(0,1))
mf_title("Distribution de la population dans le Lot (2018)")
```

Distribution de la population dans le Lot (2018)



3.7 Carte de stocks et de catégories

La fonction `mf_map(..., var = c("var1", "var2"), type = "prop_typo")` représente des symboles proportionnels dont les surfaces sont proportionnelles aux valeurs d'une variable et dont la couleur représente les modalités d'une variable qualitative. La fonction utilise les arguments des fonctions `mf_map(..., type = "prop")` et `mf_map(..., type = "typo")`.

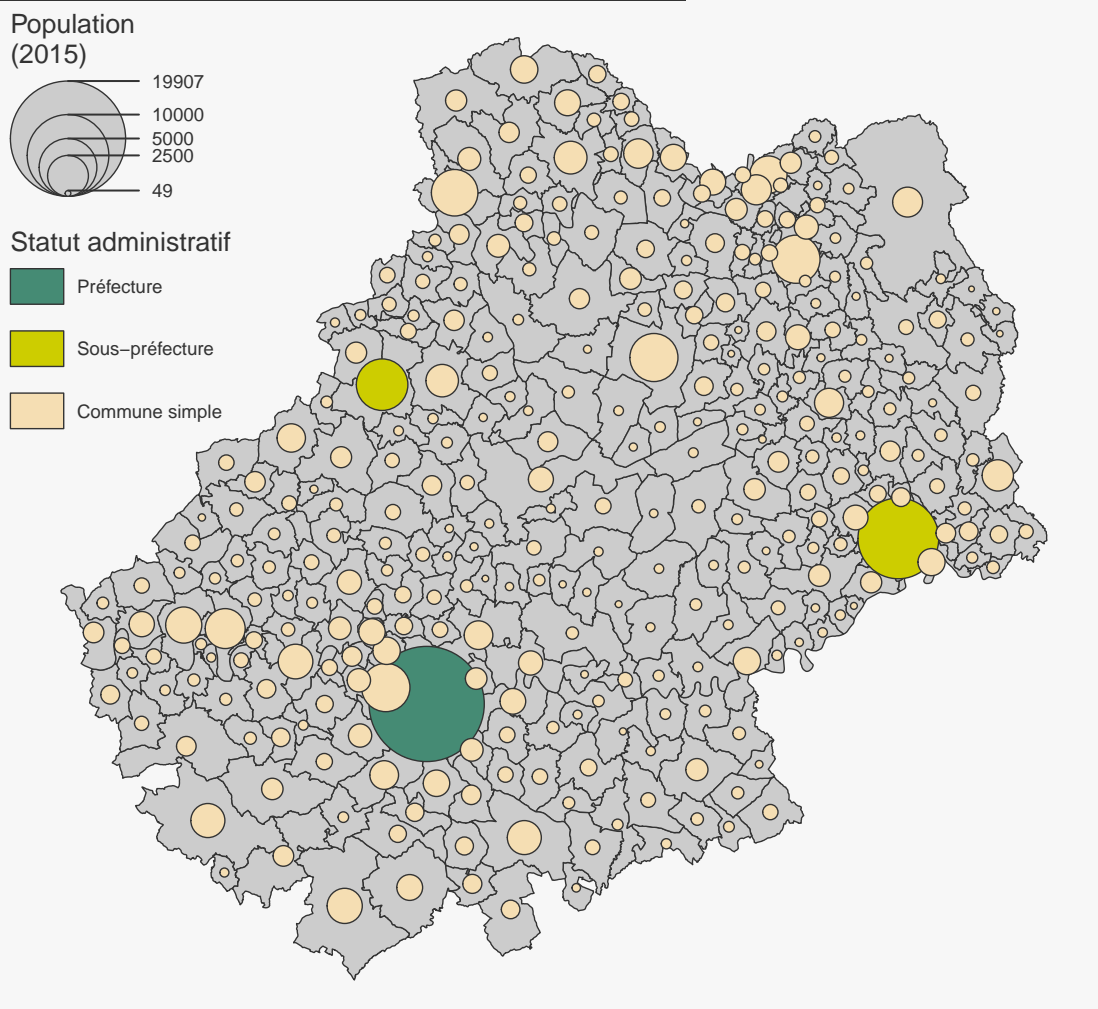
```
mf_map(x = com)
mf_map(
  x = com,
```

```

var = c("POPULATION", "STATUT"),
type = "prop_typo",
pal = c("aquamarine4", "yellow3", "wheat"),
val_order = c("Préfecture", "Sous-préfecture", "Commune simple"),
leg_pos = "topleft",
leg_title = c("Population\n(2015)",
              "Statut administratif"),
)
mf_title("Distribution de la population dans le Lot (2018)")

```

Distribution de la population dans le Lot (2018)



Exercice

1. Importer la couche des communes du département du Lot à partir du fichier geopackage **lot.gpkg**.
2. Importer le fichier **com.csv**.
Ce jeu de données porte sur les communes du Lot et contient plusieurs variables supplémentaires:
 - le nombre d'actifs (**ACT**).
 - le nombre d'actifs dans l'industrie (**IND**)
 - la part des actifs dans la population totale (**SACT**)
 - la part des actifs dans l'industrie dans le total des actifs (**SACT_IND**)
3. Joindre le jeu de données et la couche des communes.
4. Créer une carte de la population active.
Quel mode de représentation utiliser? Quels choix cela implique-t-il?
5. Créer une carte de la part de la population active dans la population totale.
Quel mode de représentation utiliser? Quels choix cela implique-t-il?

4 La mise en page

Pour être finalisée, une carte thématique doit contenir certains éléments additionnels tels que : le titre, l'auteur, la source, l'échelle, l'orientation...

4.1 Les données d'exemple

Les lignes suivantes importent les couches d'information spatiales situées dans le fichier [geopackage lot.gpkg](#).

```
library(sf)
# import des communes du Lot
com <- st_read("data/lot.gpkg", layer = "communes", quiet = TRUE)
# import des départements français
dep <- st_read("data/lot.gpkg", layer = "departements", quiet = TRUE)
# import des restaurants
resto <- st_read("data/lot.gpkg", layer = "restaurants", quiet = TRUE)
# import des routes autour de la commune de Gramat
route <- st_read("data/lot.gpkg", layer = "routes", quiet = TRUE)
```

4.2 Les thèmes

La fonction `mf_theme()` définit un thème cartographique. L'utilisation d'un thème permet de définir plusieurs paramètres graphiques qui sont ensuite appliqués aux cartes créées avec `mapsf`. Ces paramètres sont : les marges de la carte, la couleur principale, la couleur de fond, la position et l'aspect du titre.

4.2.1 Utiliser un thème prédéfini

Une série de thèmes prédéfinis est disponible par défaut (voir `?mf_theme`).


```

library(mapsf)
# Utilisation d'un thème prédéfini
mf_theme("default")
mf_map(com)
mf_title("Theme : 'default'")
mf_theme("darkula")
mf_map(com)
mf_title("Theme : 'darkula'")
mf_theme("candy")
mf_map(com)
mf_title("Theme : 'candy'")
mf_theme("nevermind")
mf_map(com)
mf_title("Theme : 'nevermind'")

```

4.2.2 Modifier un thème existant

Il est possible de modifier un thème existant. Dans cet exemple, nous utilisons le thème “default” et nous en modifions quelques paramètres.

```

library(mapsf)
mf_theme("default")
mf_map(com)
mf_title("default")
mf_theme("default", tab = FALSE, font = 4, bg = "grey60", pos = "center")
mf_map(com)
mf_title("modified default")

```

4.2.3 Créer un thème

Il est également possible de créer un thème.

```

mf_theme(
  bg = "lightblue",    # couleur de fond
  fg = "tomato1",     # couleur principale
  mar = c(0,0,1.5,0), # marges
  tab = FALSE,        # style "onglet" pour le titre
  inner = FALSE,      # titre à l'intérieur de la zone de carte ou à l'extérieur
  line = 1.5,         # espace dédié au titre

```

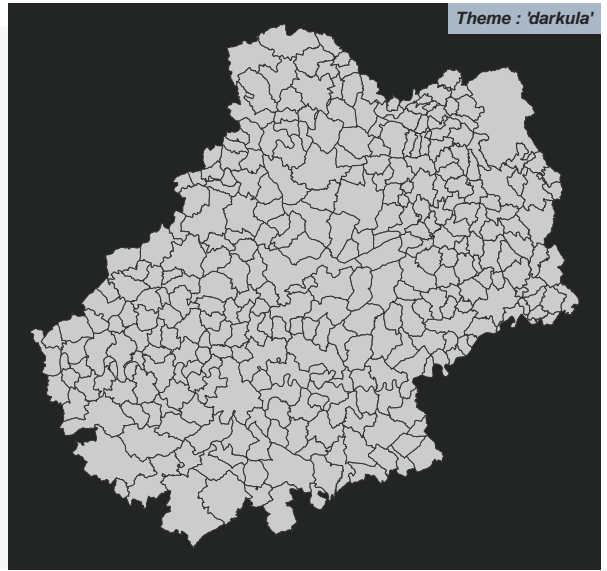
Theme : 'default'



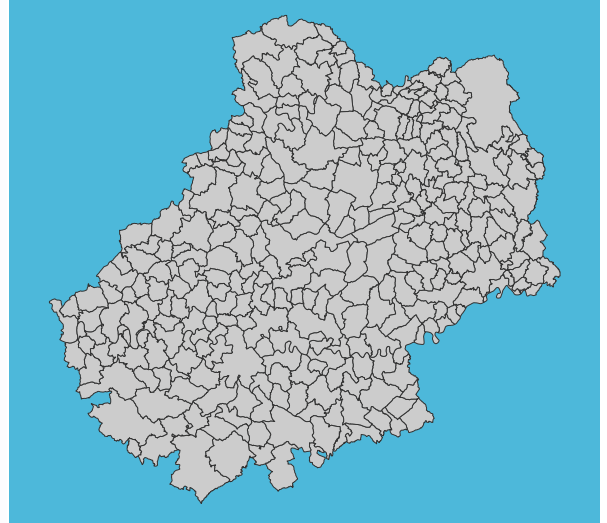
Theme : 'candy'



Theme : 'darkula'



Theme : 'nevermind'



default

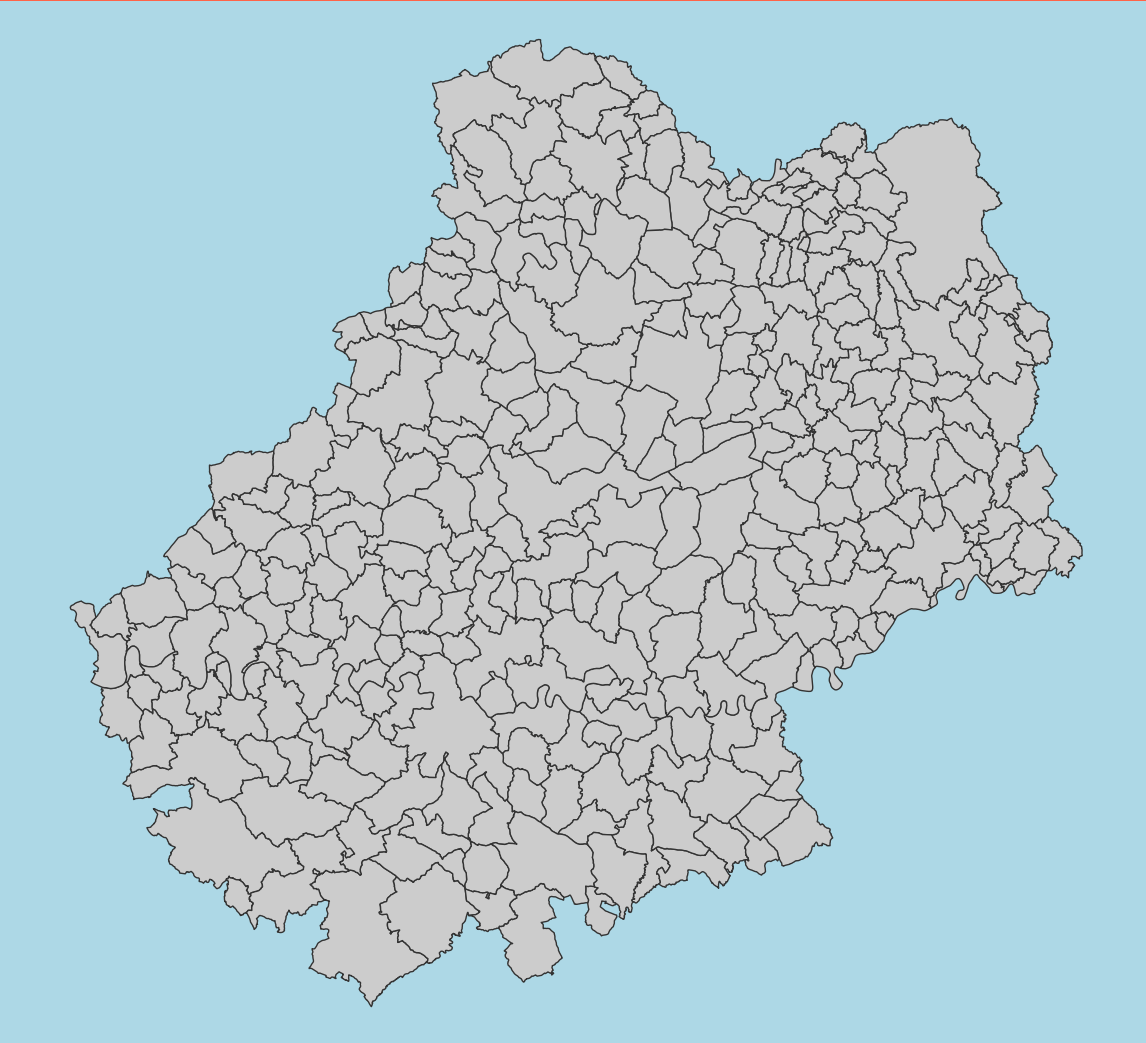


modified default



```
pos = "center", # position du titre
cex = 1.2,      # taille du titre
font = 2       # type de fonte pour le titre
)
mf_map(com)
mf_title("New theme")
```

New theme



Vignette

[How to Use Themes](#)

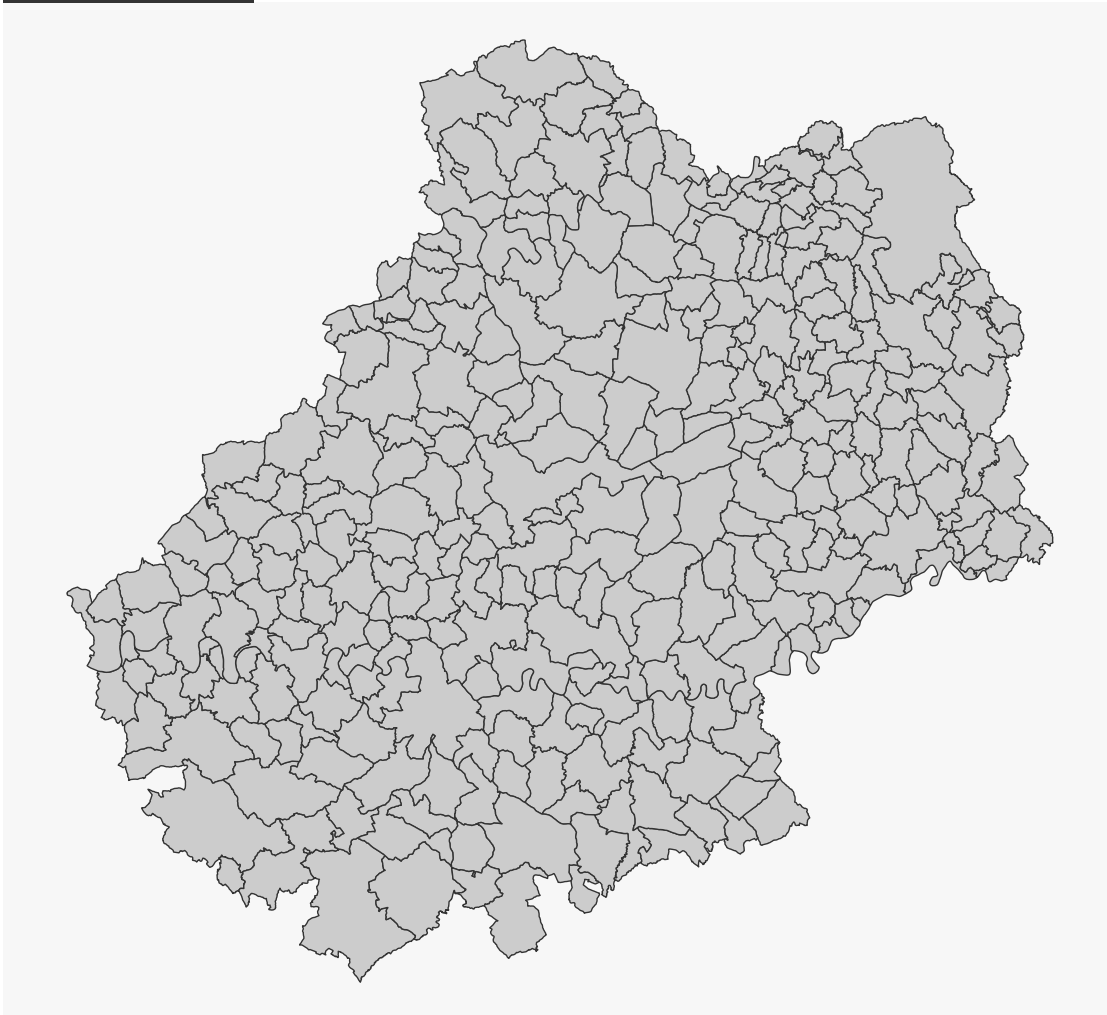
4.3 Les titres

La fonction `mf_title()` permet d'ajouter un titre à une carte.

```
mf_theme("default")  
mf_map(com)
```

```
mf_title("Titre de la carte")
```

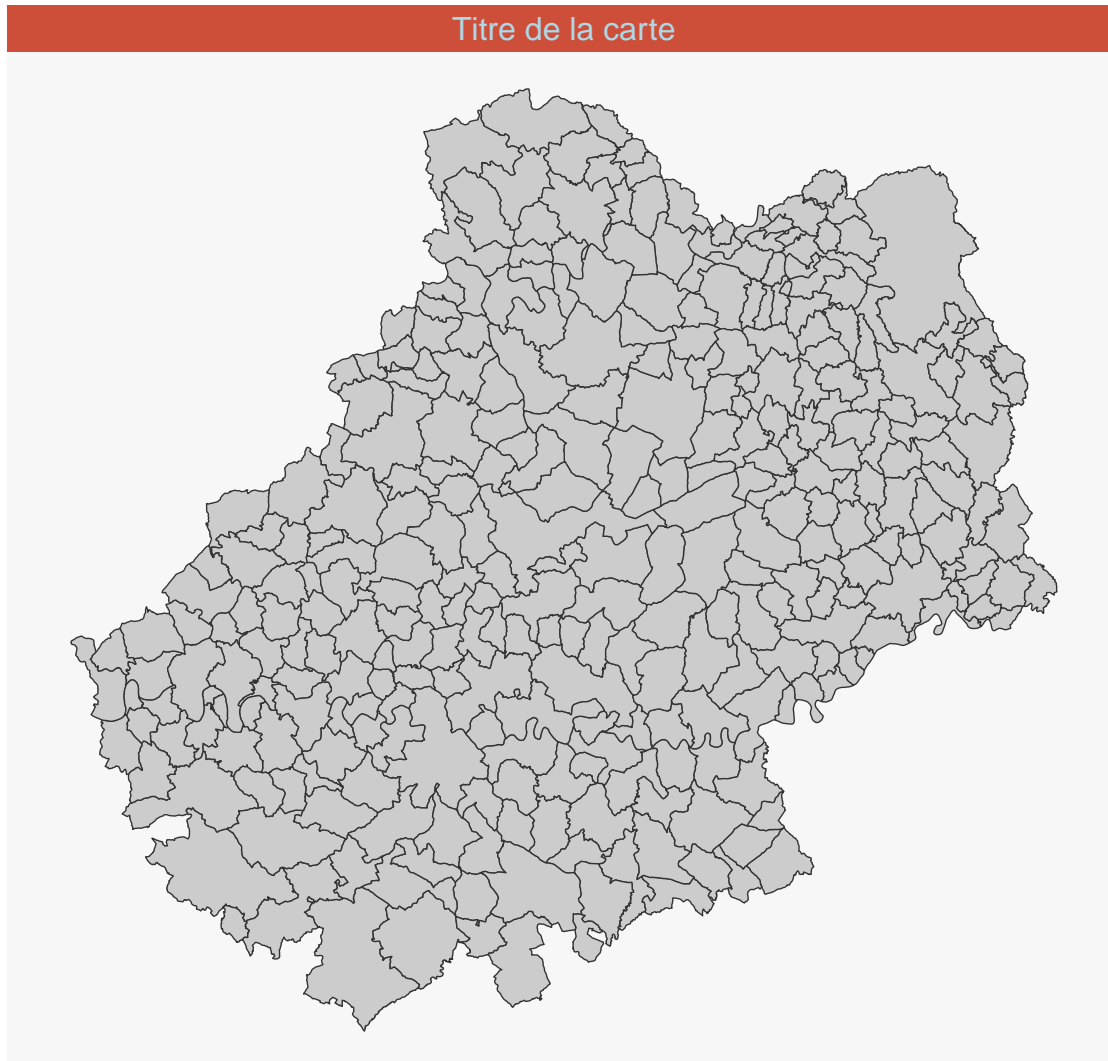
Titre de la carte



Il est possible de personnaliser l'aspect du titre.

```
mf_map(com)
mf_title(
  txt = "Titre de la carte",
  pos = "center",
  tab = FALSE,
  bg = "tomato3",
  fg = "lightblue",
  cex = 1,
```

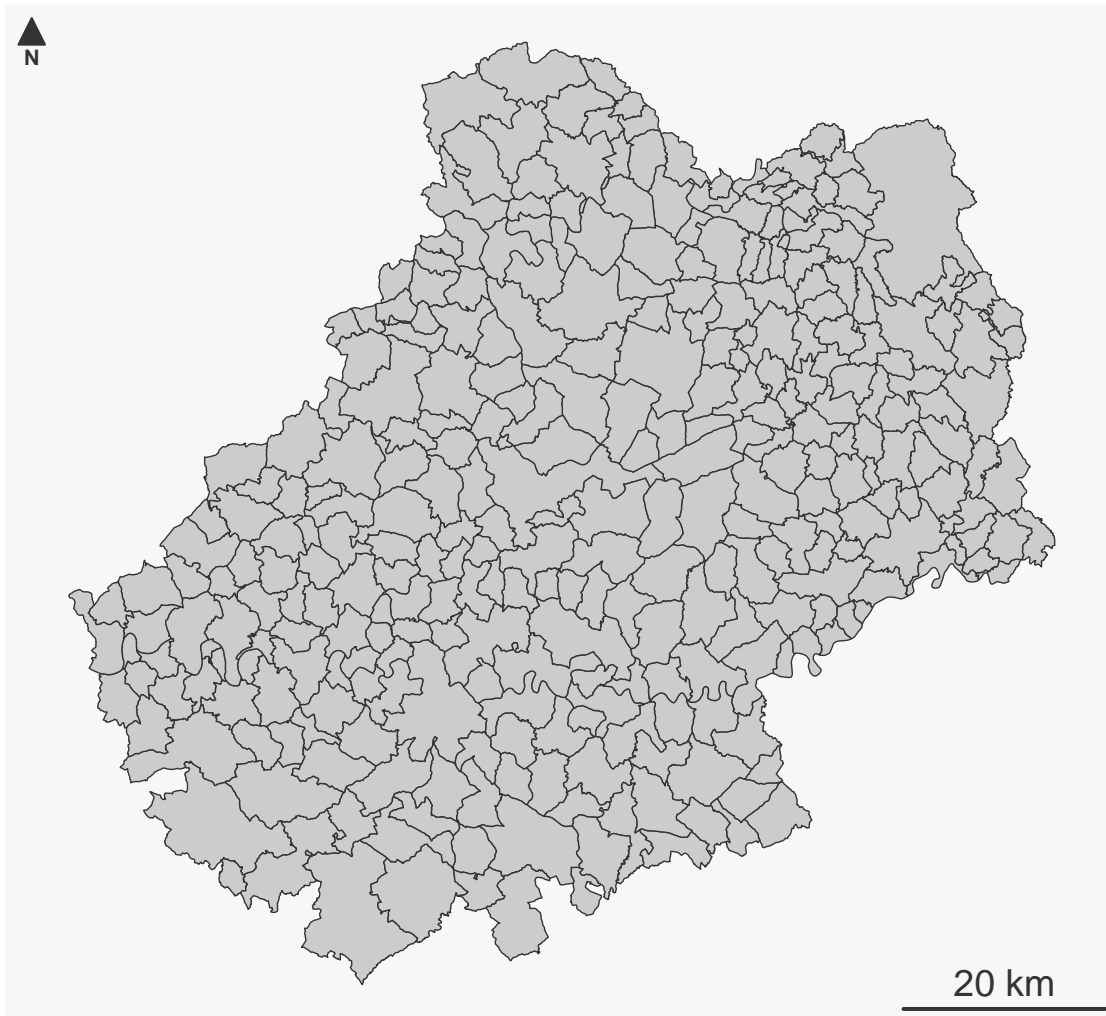
```
line = 1.2,  
font = 1,  
inner = FALSE  
)
```



4.4 Flèche d'orientation et échelle

La fonction `mf_arrow()` permet de choisir la position et l'aspect de la flèche d'orientation. La fonction `mf_scale()` permet de choisir la position et l'aspect de l'échelle.

```
mf_map(com)
mf_scale(size = 20, lwd = 2, cex = 1.2)
mf_arrow()
```



Toutes les cartes doivent avoir une échelle et une flèche d'orientation !?



[Carnet Observable interactif](#) de Nicolas Lambert

4.5 Crédits

La fonction `mf_credits()` permet d'afficher une ligne de crédits (sources, auteur...).

```
mf_map(com)  
mf_credits("Auteurs : TG & HP\nSources : IGN, 2018")
```




4.6 Habillage complet

La fonction `mf_layout()` permet d'afficher tous ces éléments (titre, c.

```
mf_map(com)
mf_layout(
  title = "Le Lot",
  credits = "Auteurs : TG & HP\nSources : IGN, 2018",
  arrow = TRUE,
  scale = TRUE
)
```

Le Lot



Auteurs : TG & HP
Sources : IGN, 2018

10 km

4.7 Annotations

```
# Sélection d'une commune
cahors <- com[com$NOM_COM == "Cahors", ]
mf_map(com)
mf_annotation(
  x = cahors,
  txt = "Cahors",
  halo = TRUE,
```

```
cex = 1.5  
)
```



4.8 Étiquettes

La fonction `mf_label()` est dédiée à l'affichage d'étiquettes.

```
# Selection des communes qui intersectent la  
# commune de Cahors  
com_sel <- st_filter(com, com[com$NOM_COM == "Cahors", ])  
mf_map(com_sel)
```

```
mf_label(  
  x = com_sel,  
  var = "NOM_COM",  
  col= "black",  
  halo = TRUE,  
  overlap = FALSE,  
  lines = FALSE  
)  
mf_scale()
```

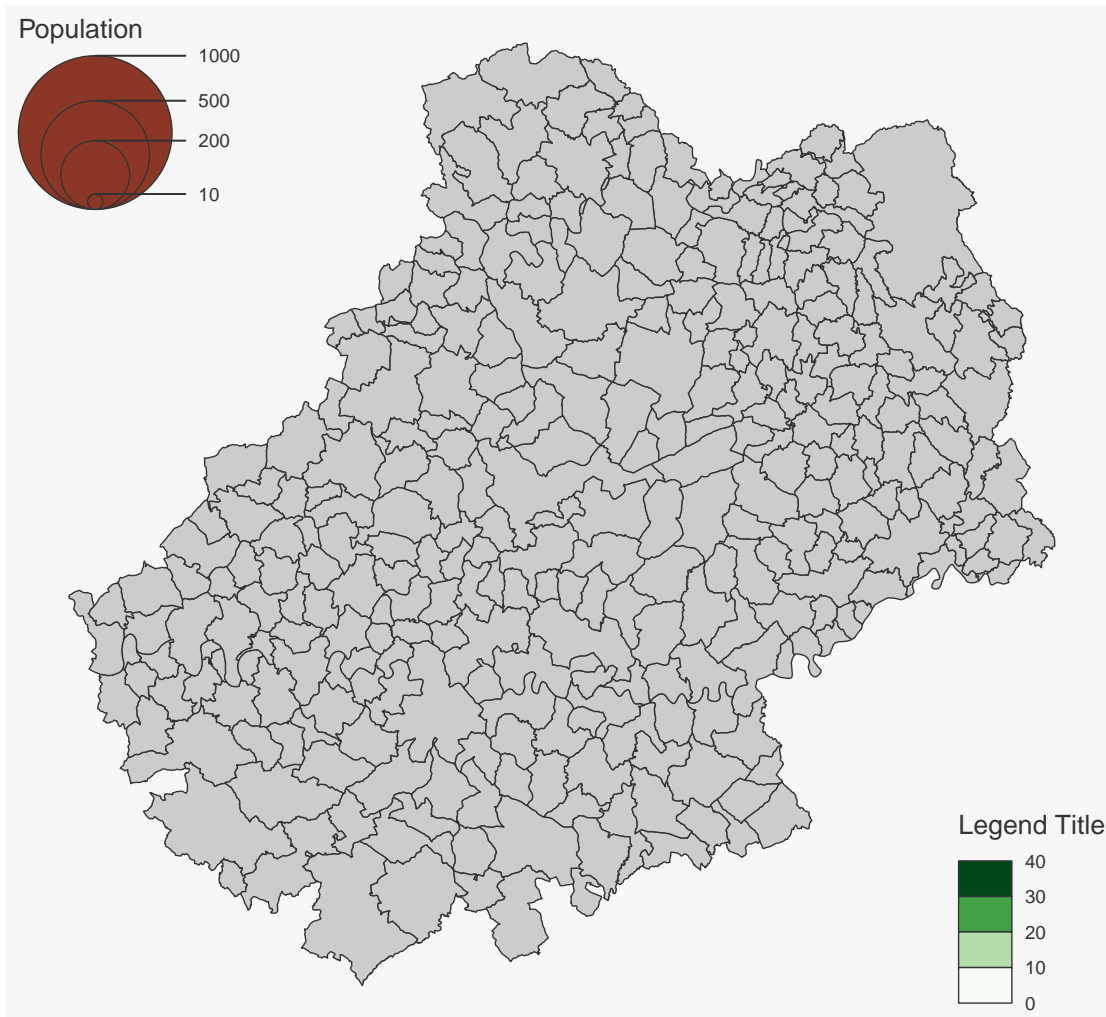


L'argument `halo = TRUE` permet d'afficher un léger halo autour des étiquettes et l'argument `overlap = FALSE` permet de créer des étiquettes ne se recouvrant pas.

4.9 Les légendes

La fonction `mf_legend()` permet d'afficher des légendes.

```
mf_map(com)
mf_legend(
  type = "prop",
  val = c(1000,500,200,10),
  inches = .4,
  title = "Population",
  pos = "topleft"
)
mf_legend(
  type = "choro",
  val = c(0,10,20,30,40),
  pal = "Greens",
  pos = "bottomright",
  val_rnd = 0
)
```



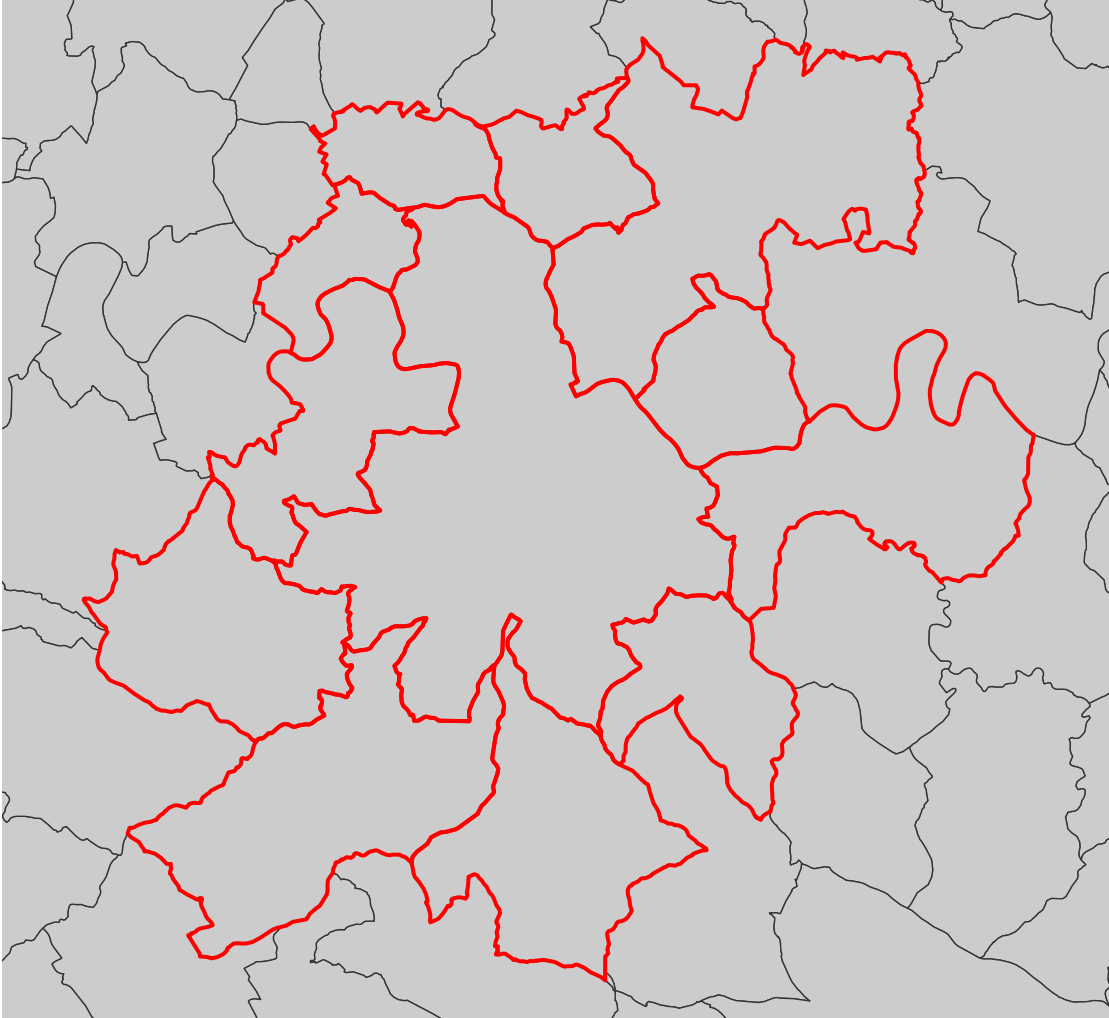
Le package maplegend

La fonction `mf_legend()` s'appuie sur le package [maplegend](#)

4.10 Centrer la carte sur une région

La fonction `mf_map()` permet d'initialiser une carte en la centrant sur un objet spatial en choisissant un fond et des bordures transparentes.

```
mf_map(x = com_sel, col = NA, border = NA)
mf_map(com, add = TRUE)
mf_map(com_sel, col = NA, border = "red", lwd = 2, add = TRUE)
```



4.11 Afficher plusieurs cartes sur la même figure

Il faut ici utiliser l'argument `mfrow` de la fonction `par()`. Le premier chiffre représente le nombre lignes et le deuxième le nombre de colonnes.

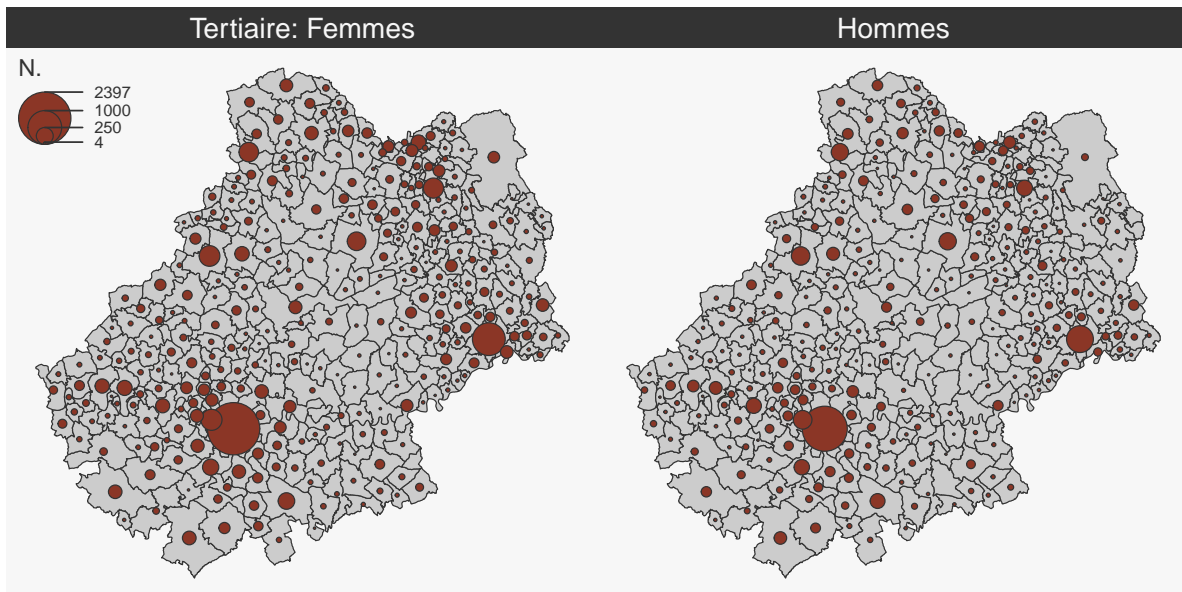
Par exemple `par(mfrow = c(1, 2))` (1 ligne et 2 colonnes) permet d'afficher deux cartes côte à côte.

```
# define the figure layout (1 row, 2 columns)
opar <- par(mfrow = c(1, 2))
# define theme
mf_theme("default", mar = c(0,0,1.2,0), tab = FALSE, pos = "center")
# first map
```

```

mf_map(com)
mf_map(com, "TER_F", "prop",
        inches = .2,
        val_max = 4000,
        leg_pos = "topleft",
        leg_title = "N.")
mf_title("Tertiaire: Femmes")
# second map
mf_map(com)
mf_map(com, "TER_H", "prop",
        inches = .2,
        val_max = 4000,
        leg_pos = NA)
mf_title("Hommes")
par(opar)

```



Comparer des cartes de cercles proportionnels

Notons ici l'utilisation des arguments `inches` et `val_max` avec des valeurs identiques pour les deux cartes. Cela rend possible les comparaisons entre les deux cartes.

Vignette

[How to Create Faceted Maps](#)

4.12 Export des cartes

`mf_export()` permet d'exporter des cartes en PNG ou en SVG.

Le ratio hauteur/largeur de la carte correspondra au ratio hauteur/largeur d'un objet spatial. Si `width` est spécifié alors `height` est déduit du ratio hauteur/largeur de `x`, de la taille des marges de la figure et des dimensions du titre (du thème utilisé en somme).

`mf_export()` peut aussi être utilisé pour ajouter de l'espace sur un ou plusieurs côté de la carte ou encore centrer la carte sur une zone particulière.

Toujours utiliser `add = TRUE` après un appel à `mf_export()` et ne pas oublier d'utiliser `dev.off()` pour finaliser l'export de la figure.

```
#création d'un thème personnel
mf_theme("candy", bg = "grey80")
mf_export(
  x = com,
  filename = "img/com_default.png",
  width = 600
)
mf_map(com, add = TRUE)
mf_title("Le Lot")
dev.off()
```



Figure 4.1: com_default.png

L'emprise de cette carte est exactement celle de la région affichée.

Lors de l'export suivant nous ajoutons de l'espace sur la partie droite de la carte (50% de la largeur de x).

```
mf_export(  
  x = com,  
  filename = "img/com_expand.png",  
  width = 600,  
  expandBB = c(0, 0, 0, .5)  
)  
mf_map(com, add = TRUE)  
mf_title("Le Lot")  
dev.off()
```

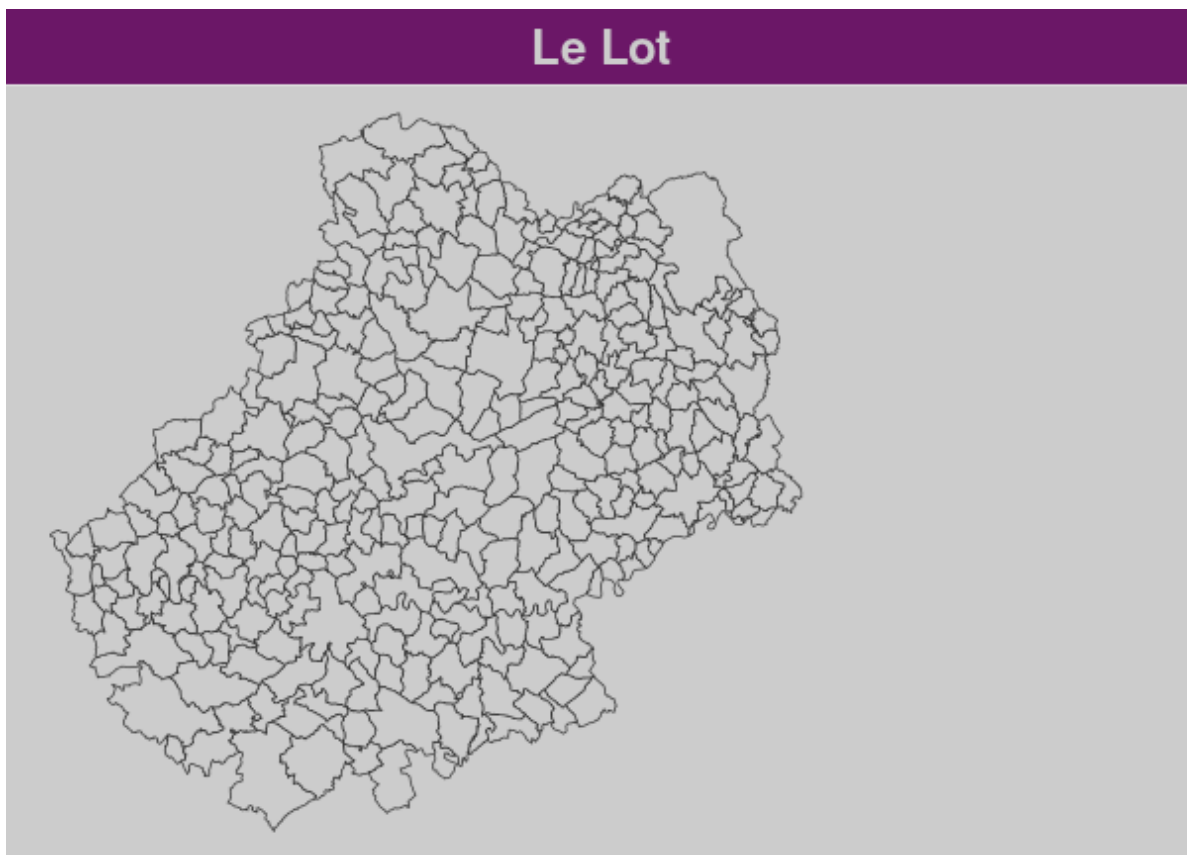


Figure 4.2: com_expand.png

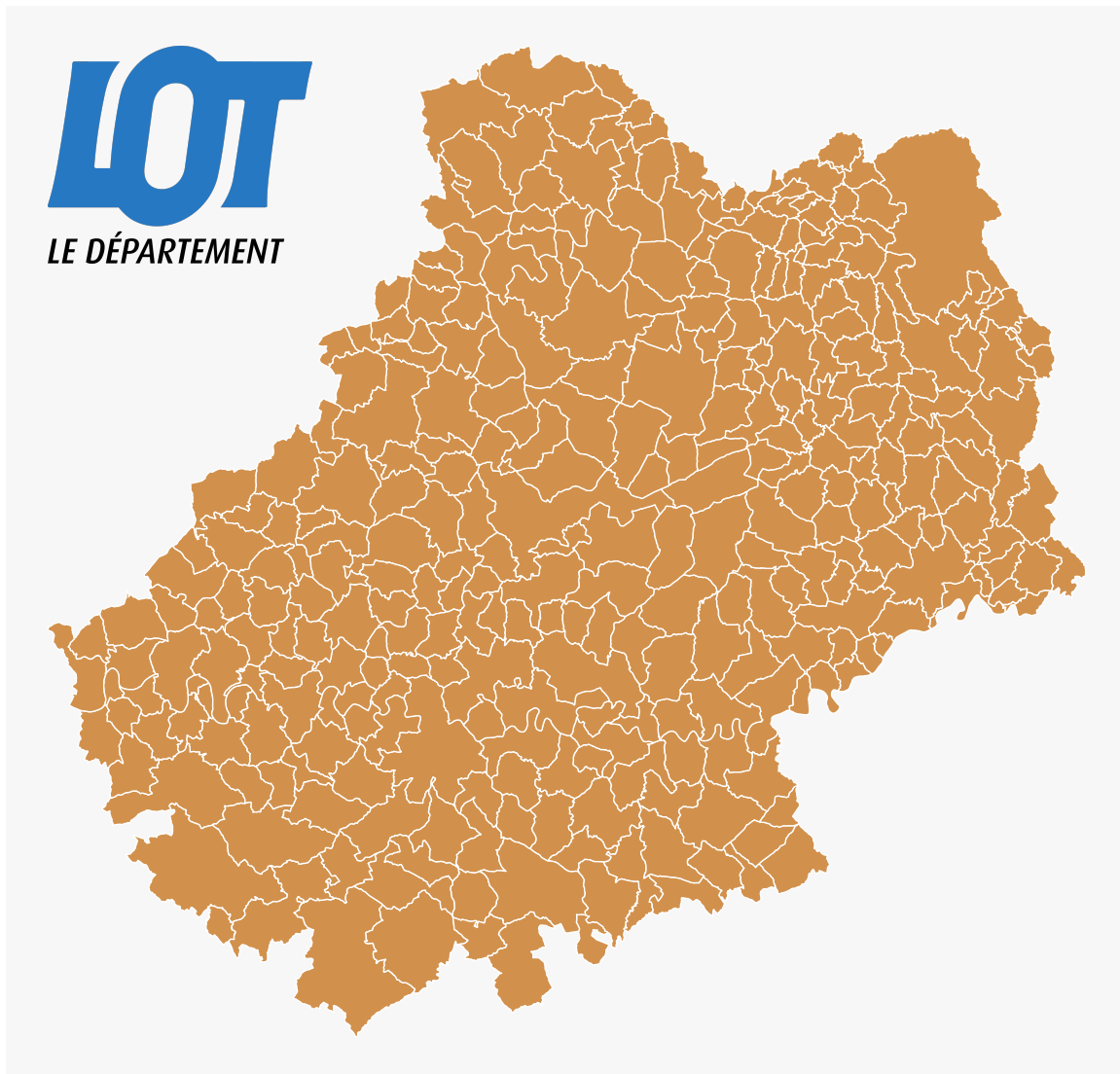
Vignette

[How to Export Maps](#)

4.13 Ajouter une image sur une carte

Cela peut être utile pour ajouter un logo, un pictogramme. La fonction `readPNG()` du package `png` permet l'ajout d'images sur une figure.

```
mf_theme("default", mar = c(0,0,0,0))
library(png)
# import de l'image
logo <- readPNG("img/Logo CG 46 - Bleu.png")
# dimension de l'image en unité de la carte
pp <- dim(logo)[2:1] * 20
# Définition d'un point d'encrage de l'image dans la figure, ici
# le coin supérieur gauche de la bounding box du département
xy <- st_bbox(com)[c(1,4)]
mf_map(com, col = "#D1914D", border = "white")
rasterImage(
  image = logo,
  xleft = xy[1],
  ybottom = xy[2] - pp[2],
  xright = xy[1] + pp[1],
  ytop = xy[2]
)
```



4.14 Placer précisément un élément sur la carte

Dans plusieurs fonctions de `mapsf` il est possible de placer des éléments d’habillage de manière interactive avec la position *“interactive”*.

Plus généralement la fonction `locator()` permet de cliquer sur une figure et d’obtenir les coordonnées d’un point dans le système de coordonnées de la figure (de la carte). `locator()` peut être utilisée sur la plupart des graphiques (pas ceux produits avec `ggplot2`).

4.15 Ajouter un ombrage à une couche

La fonction `mf_shadow()` permet de créer une ombre à une couche de polygones.

```
mf_theme("default")  
mf_shadow(com)  
mf_map(com, add=TRUE)
```



4.16 Création de cartons

La fonction `mf_inset_on()` permet de démarrer la création d'un carton. Il faut ensuite "refermer" le carton avec `mf_inset_off()`.

```

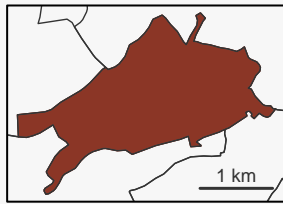
biars <- com[com$NOM_COM == "Biars-sur-Cère", ]
# affichage de toutes les communes
mf_map(com)
# affichage de la sélection de communes
mf_map(biars, col = "tomato4", border = "tomato1",
       lwd = 2, add = TRUE)

# Démarrer le carton
# en haut à gauche sur
# le ratio width/height de biars
# avec une largeur d'1/4 de la largeur de la figure
mf_inset_on(x = biars, pos = "topleft", cex = .25)
# couleur du fond
mf_theme(bg = "#f7f7f7")
# affichage de la commune cible
mf_map(biars, col = "tomato4", border = "tomato1")
# affichage des communes
mf_map(com, col = NA, add = TRUE)
# échelle
mf_scale(size = 1, pos = "bottomright")
# affichage d'un cadre
box()
# Fermer le carton
mf_inset_off()

# Ajouter titre et échelle
mf_title("Biars-sur-Cère")
mf_scale(10, pos = 'bottomright')

```

Biars-sur-Cère

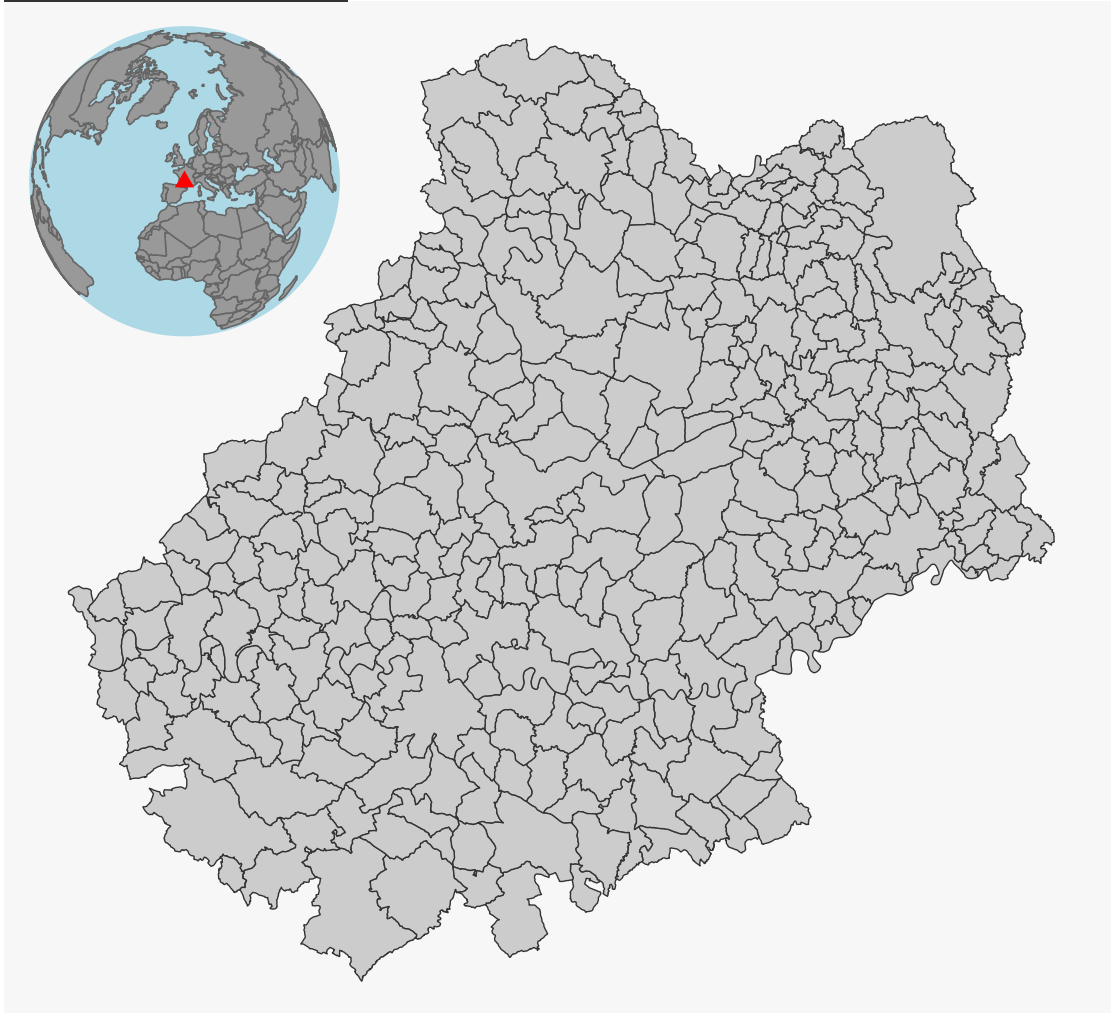


Carton “carte du monde”

On peut facilement créer un carton avec la position d'un point sur une carte du monde en combinant les fonctions `mf_worldmap()` et `mf_inset_on(x = "worldmap", ...)` :

```
mf_map(com)
mf_inset_on(x = "worldmap", cex = .3, pos = "topleft")
mf_worldmap(com)
mf_inset_off()
mf_title("Le Lot dans le Monde!")
```


Le Lot dans le Monde!



Vignette

[How to Create Inset Maps](#)

Exercice

1. Importer la couche des communes du département du Lot à partir du fichier geopackage **lot.gpkg**.
2. Importer le fichier **com.csv**.

Ce jeu de données porte sur les communes du lot et contient plusieurs variables supplémentaires:

- le nombre d'actifs (**ACT**).
- le nombre d'actifs dans l'industrie (**IND**)
- La part des actifs dans la population totale (**SACT**)
- La part des actifs dans l'industrie dans le total des actifs (**SACT_IND**)

3. Joindre le jeu de données et la couche des communes.
4. Créez une carte représentant la population active travaillant dans l'industrie.
5. Ajoutez les éléments d'habillage indispensables.
6. Utilisez un thème personnalisé.
7. Ajoutez un carton de localisation du Lot.
8. Exportez la carte au format PNG avec 800 pixels de large.
9. Comment rendre la carte plus intelligible ? Allez-y !

partie II

Cartographie 3D et Anamorphoses

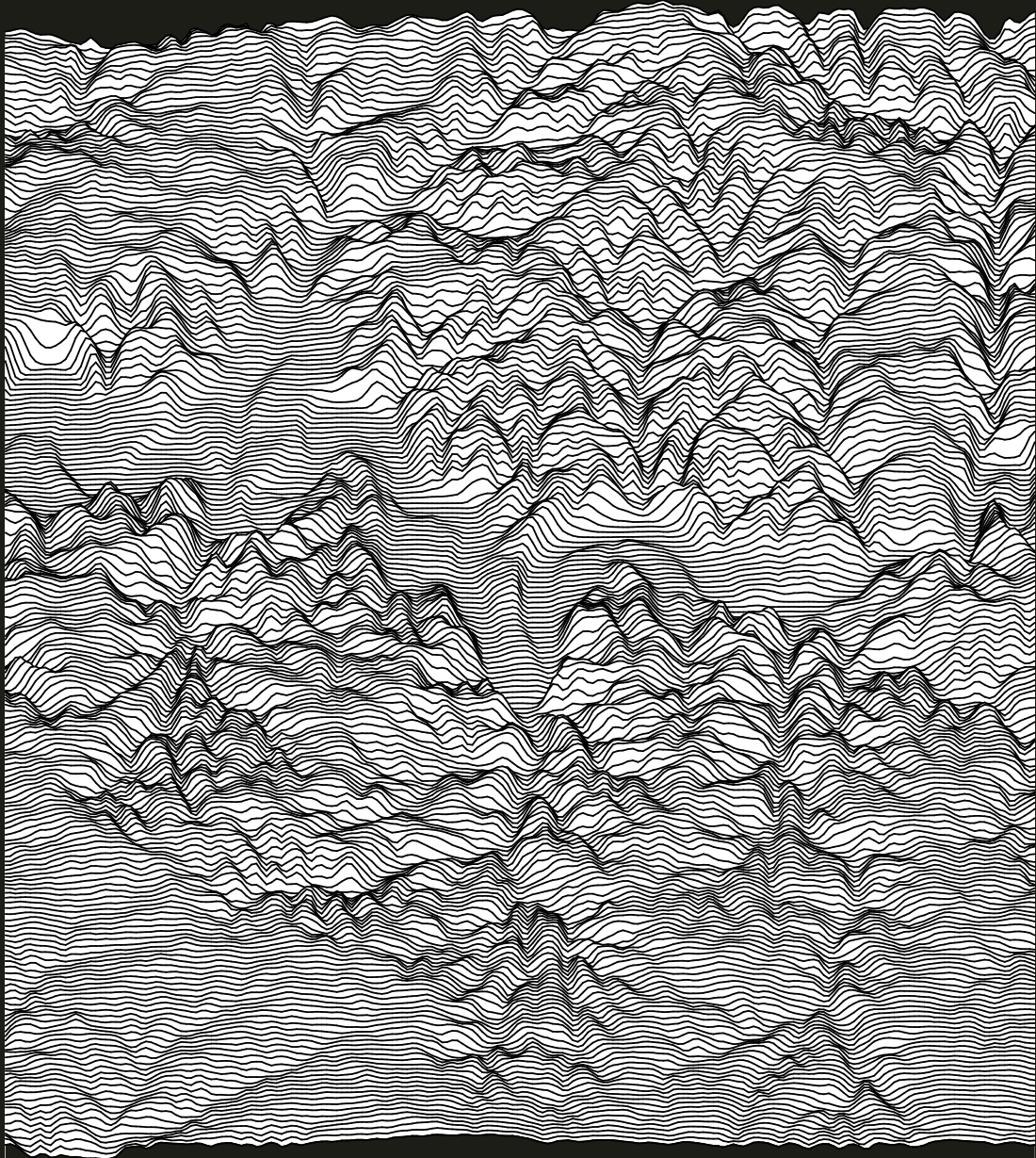
5 Les cartes en 3D

5.1 linemap

Le package `linemap` (Giraud, 2023a) permet de réaliser des cartes composées de lignes.

```
library(linemap)
library(sf)
library(terra)
library(mapsf)
com <- st_read("data/lot.gpkg", layer = "communes", quiet = TRUE)
elev <- rast("data/elev.tif")
cahors <- st_buffer(com[com$NOM_COM == "Cahors", ], 5000)
elev_cahors <- crop(elev, cahors)
mf_theme("green")
mf_map(cahors, col = NA, border = NA, expandBB = c(-.05,0,0.05,0))
linemap(x = elev_cahors,
        k = 5,
        bg = getOption("mapsf.bg"),
        add = TRUE, lwd = 1)
mf_title("Altitude vers Cahors")
mf_credits("Giraud & Pecout, 2023")
```

Altitude vers Cahors



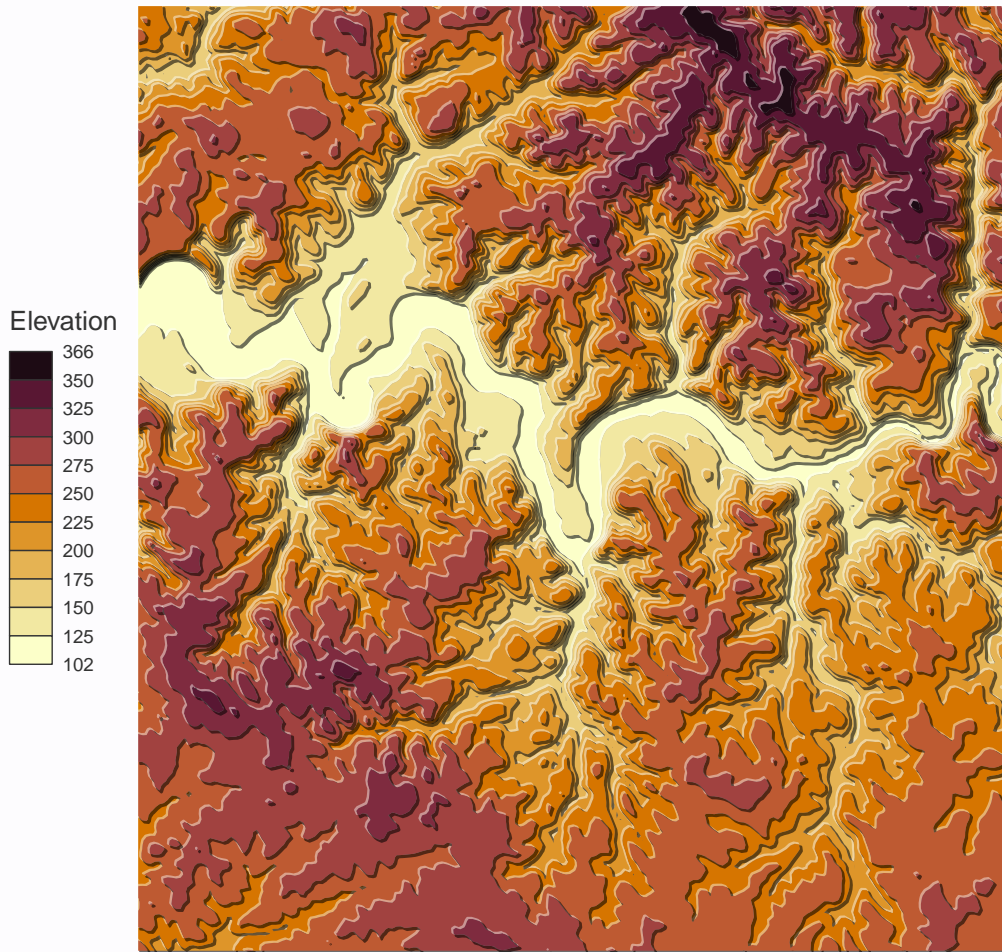
Giraud & Pecout, 2023

5.2 Relief Tanaka

Nous utilisons le package `tanaka` (Giraud, 2023c) qui met à disposition une méthode (Tanaka, 1950) utilisée pour améliorer la perception du relief.

```
library(tanaka)
library(terra)
mf_theme("candy")
mf_map(cahors, col = NA, border = NA)
tanaka(elev_cahors,
       breaks = seq(100,375,25),
       col = hcl.colors(11, "Lajolla"),
       add = TRUE)
mf_title("Altitude vers Cahors")
mf_credits("Giraud & Pecout, 2023")
```

Altitude vers Cahors



Giraud & Pecout, 2023

6 Les cartogrammes

“L’anamorphose classique est une représentation des États (ou de mailles quelconques) par **des rectangles ou des polygones quelconques** en fonction d’une **quantité** qui leur est rattaché. (...) On s’efforce de **garder l’arrangement général** des mailles ou la silhouette du continent.”

Brunet et al. (1993)

3 types d’anamorphoses ou cartogrammes sont ici présentés :

- Les cartogrammes de Dorling (Dorling, 1996)
- Les cartogrammes non contigus (Olson, 1976)
- Les cartogrammes contigus (Dougenik et al., 1985)

Ressources

- Un cours complet sur les anamorphoses : [Les anamorphoses cartographiques](#) (Lambert, 2015).
- Les cartogrammes avec R : [Faire des cartograms dans R](#)

Pour réaliser les cartogrammes nous utilisons le package `cartogram` (Jeworutzki, 2023).

6.1 Les cartogrammes de Dorling

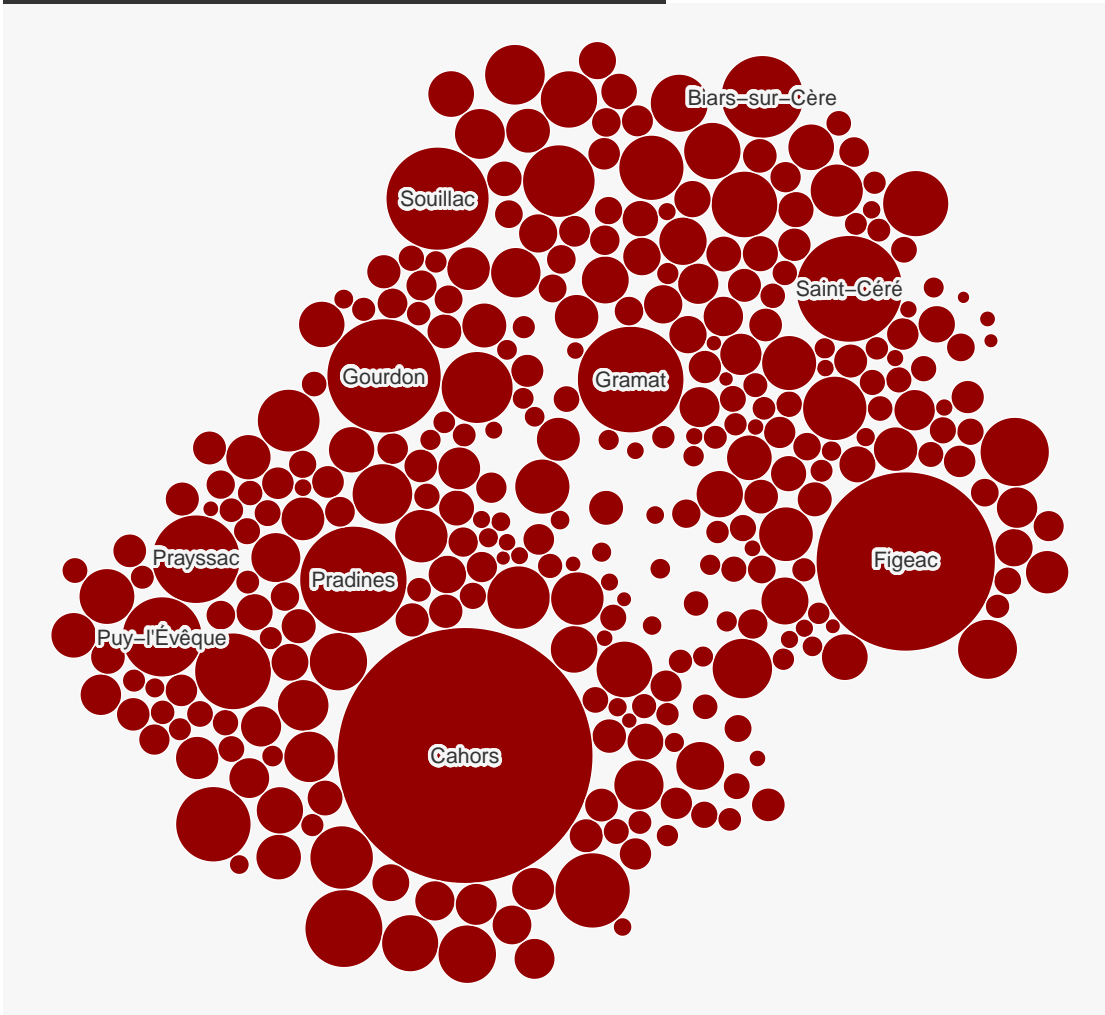
Les territoires sont représentés par des figurés (cercles, des carrés ou des rectangles) ne se recouvrant pas dont les surfaces sont proportionnelles à une variable. Les positions des figurés sont définies selon les positions de départ.

```
library(mapsf)
library(cartogram)
com <- st_read("data/lot.gpkg", layer = "communes", quiet = TRUE)
com_dorling <- cartogram_dorling(x = com, weight = "POPULATION", k = 7)
mf_map(com_dorling, col = "#940000", border = "white")
mf_label(
  x = com_dorling[order(com_dorling$POPULATION, decreasing = TRUE), ][1:10,],
  var = "NOM_COM",
```



```
overlap = FALSE, lines = FALSE,  
halo = TRUE,  
r = .15  
)  
mf_title("Population du Lot - Cartogramme de Dorling")
```

Population du Lot – Cartogramme de Dorling



Le paramètre k permet de faire varier le facteur d'expansion des cercles.

On identifie assez mal l'espace.

On peut nommer les cercles pour se repérer et/ou s'aider de la couleur pour faire apparaître des clusters et mieux identifier les blocs géographiques.

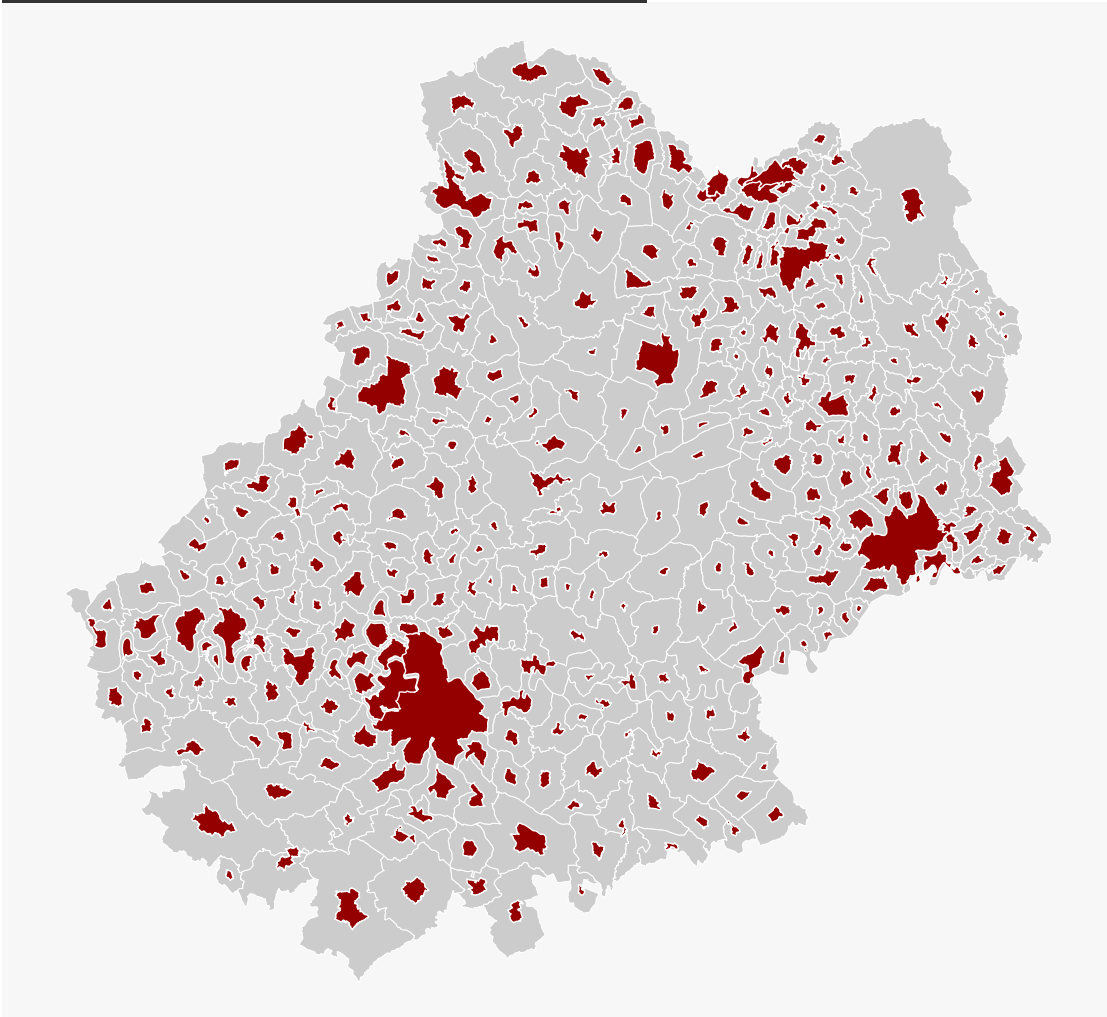
La perception des quantités est très bonne.
Les tailles de cercles sont vraiment comparables.

6.2 Les cartogrammes non continus

La taille des polygones est proportionnelle à une variable. L'agencement des polygones les uns par rapport aux autres est conservée. La forme des polygones est ressemblante.

```
com_ncont <- cartogram_ncont(x = com, weight = "POPULATION", k = 1.2)
mf_map(com, border = "white", lwd = 0.5,)
mf_map(com_ncont, col = "#940000", border="white", add = TRUE)
mf_title("Population du Lot - Cartogramme de Olson")
```

Population du Lot – Cartogramme de Olson



Le paramètre k permet de faire varier le facteur d'expansion des polygones.

La topologie des régions est perdue.

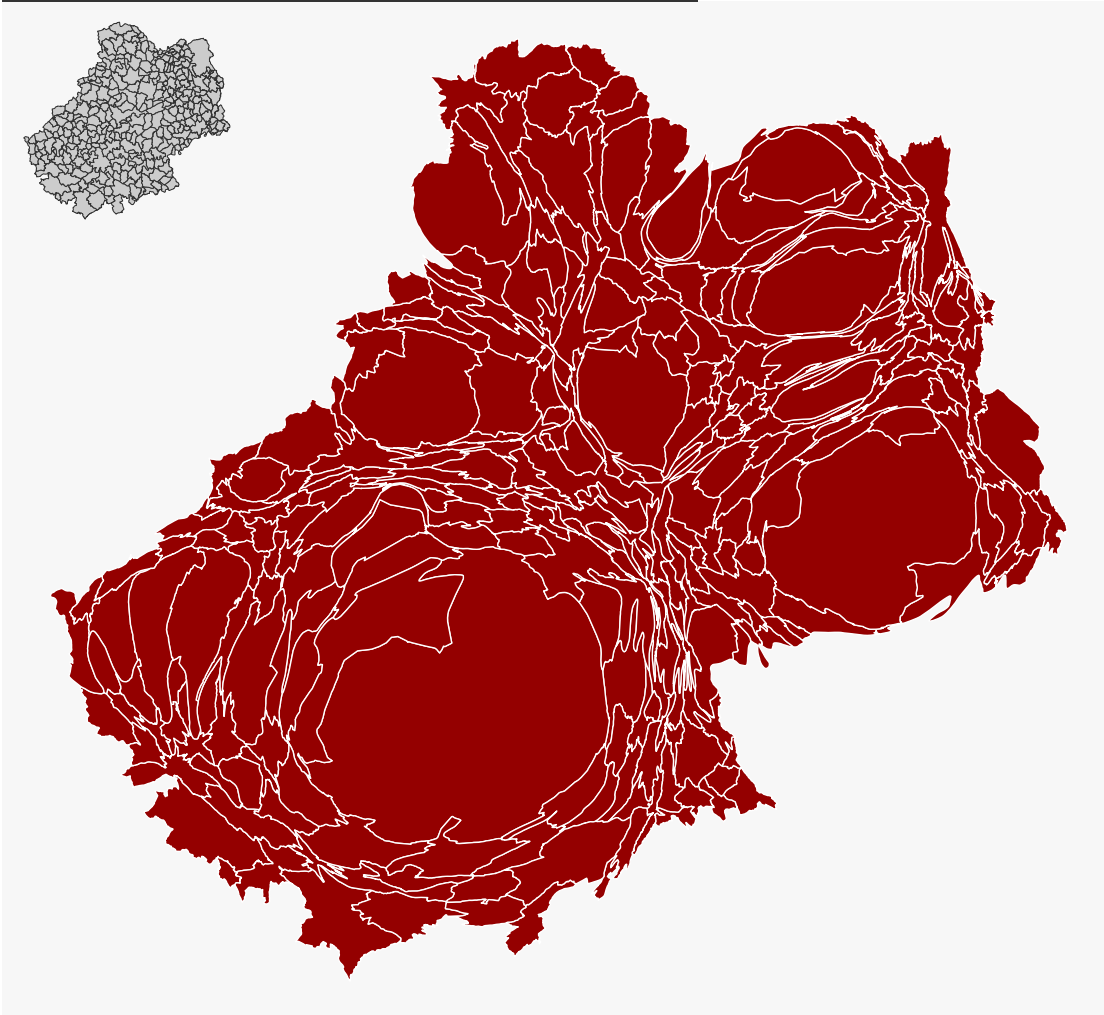
La conservation de la forme des polygones est optimale.

6.3 Les cartogrammes continus

La taille des polygones est proportionnelle à une variable. L'agencement des polygones les uns par rapport aux autres est conservée. Pour conserver la contiguïté, la forme des polygones est fortement transformée.

```
com_cont <- cartogram_cont(x = com, weight = "POPULATION",
                           prepare = "none",
                           itermax = 10)
mf_map(com_cont, col = "#940000", border= "white", add = FALSE)
mf_title("Population du Lot - Cartogramme de Dougenik")
mf_inset_on(com, cex = .2, pos = "topleft")
mf_map(com, lwd = .5)
mf_inset_off()
```

Population du Lot – Cartogramme de Dougenik



La forme des polygones est fortement distordue.
C'est une "vraie carte de géographie" : la topologie et la contiguïté sont conservées.

6.4 Forces et faiblesses des cartogrammes

Les cartogrammes sont des représentations cartographiques perçues comme **innovantes** (bien que la méthode date de 40 ans). Ces images très généralisées rendent bien compte des **quantités** et des **gradients**. Ce sont de vraies images de **communication** qui **provoquent**, suscitent **l'intérêt**, véhiculent un **message fort**, **interpellent**.

Mais les cartogrammes induisent une perte des **repères visuels** (difficile de retrouver son pays, ou sa région sur la carte), demandent un **effort de lecture** qui peut être important et ne permettent pas de **gérer des données manquantes**.

References

- Béguin, M. et Pumain, D. (2023). *La représentation des données géographiques: Statistique et cartographie*. Armand Colin.
- Bertin, J. (1967). *Sémiologie Graphique. Les Diagrammes - Les Réseaux - Les Cartes*. Paris: Mouton & Gauthier-Villars.
- Brunet, R., Ferras, R. et Théry, H. (1993). *Les mots de la géographie: dictionnaire critique*. La Documentation française.
- Dorling, D. (1996). *Area cartograms: their use and creation, concepts and techniques in modern geography* (vol. 59, p. 69). Institute of British Geographers.
- Dougenik, J. A., Chrisman, N. R. et Niemeyer, D. R. (1985). An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37(1), 75-81. <https://doi.org/10.1111/j.0033-0124.1985.00075.x>
- Dunnington, D. (2023). *ggspatial: Spatial Data Framework for ggplot2*. <https://CRAN.R-project.org/package=ggspatial>
- Giraud, T. (2023a). *linemap: Line Maps*. <https://CRAN.R-project.org/package=linemap>
- Giraud, T. (2023b). *mapsf: Thematic Cartography*. <https://CRAN.R-project.org/package=mapsf>
- Giraud, T. (2023c). *tanaka: Design Shaded Contour Lines (or Tanaka) Maps*. <https://CRAN.R-project.org/package=tanaka>
- Giraud, T. et Lambert, N. (2017). *Reproducible Cartography*. M. Peterson (dir.), Cham, Switzerland (p. 173-183). https://doi.org/10.1007/978-3-319-57336-6_13
- Jeworutzki, S. (2023). *cartogram: Create Cartograms with R*. <https://CRAN.R-project.org/package=cartogram>
- Lambert, N. (2015). Les anamorphoses cartographiques [Blog]. *Carnet néocartographique*. <https://neocarto.hypotheses.org/366>
- Lambert, N. et Zanin, C. (2016). *Manuel de cartographie: principes, méthodes, applications*. Armand Colin.
- Olson, J. M. (1976). Noncontiguous area cartograms. *The Professional Geographer*, 28(4), 371-380. <https://doi.org/10.1111/j.0033-0124.1976.00371.x>
- RIATE. (2023). Cartographie thématique avec Magrit. <https://magrit-formations.github.io/>
- Tanaka, K. (1950). The Relief Contour Method of Representing Topography on Maps. *The Geographical Review*, xl(3). <https://doi.org/doi:10.2307/211219>
- Tennekes, M. (2018). tmap: Thematic Maps in R. *Journal of Statistical Software*, 84(6), 1-39. <https://doi.org/10.18637/jss.v084.i06>
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>

A Les données du projet

Les données utilisées dans ce document sont stockées dans un projet RStudio. Vous devez le télécharger puis le décompresser sur votre machine. Il vous sera ensuite possible de tester l'ensemble des manipulations proposées dans ce document au sein du projet **geodata**.

[Télécharger le projet](#)

lot.gpkg

Ce fichier contient plusieurs couches d'informations.

- **departements** : les départements français métropolitains, [Admin Express COG Carto 3.0, IGN - 2021](#);
- **communes** : les communes du département du Lot (46) avec des données sur la population active occupée âgée de 25 à 54 ans, par secteur d'activité et sexe, au lieu de résidence, en 2017, [BD CARTO® 4.0, IGN - 2021 & Recensements harmonisés - Séries départementales et communales, INSEE - 2020](#);
- **routes** : les routes de la commune de Gramat et alentours (46128), [BD CARTO® 4.0, IGN - 2021](#);
- **restaurants** : les restaurants du Lot, [Base permanente des équipements \(BPE\), INSEE - 2021](#);
- **elevations** : une grille régulière de points d'altitude (pas d'1 km), [Jarvis A., H.I. Reuter, A. Nelson, E. Guevara, 2008, Hole-filled seamless SRTM data V4, International Centre for Tropical Agriculture \(CIAT\)](#).

com.csv

Ce fichier tabulaire contient des informations complémentaire sur la population active occupée âgée de 25 à 54 ans, par secteur d'activité et sexe, au lieu de résidence, en 2017, [Recensements harmonisés - Séries départementales et communales, INSEE - 2020](#).

- le nombre d'actifs (ACT);
- le nombre d'actifs dans l'industrie (IND);
- la part des actifs dans la population totale (SACT);
- la part des actifs dans l'industrie dans le total des actifs (SACT_IND).

elevation.tif

Une grille régulière de points d'altitude (pas de 30 mètres environ), [Jarvis A., H.I. Reuter, A. Nelson, E. Guevara, 2008, Hole-filled seamless SRTM data V4, International Centre for](#)

Tropical Agriculture (CIAT).

elev.tif est une version reprojétée en Lambert 93 de **elevation.tif**

✂ **clc_2018.tif**

Données CORINE Land Cover, [Corine Land Cover \(CLC\) 2018, Version 2020_20u1 - Copernicus Programme](#).

clc.tif est une version reprojétée en Lambert 93 de **clc_2018.tif**

✂ **Sentinel2A.tif**

Données Sentinel, [Sentinel, Sentinel-2A, S2A_OPER_MSI_L2A_DS_VGS2_20211012T140548_S20211012T140548_12 Octobre 2021 - Copernicus Programme](#), téléchargé le 28 décembre 2021.