

# Package ‘oceanis’

July 13, 2022

**Type** Package

**Title** Cartography for Statistical Analysis

**Version** 1.8.5

**Date** 2022-07-13

**Description** Creating maps for statistical analysis such as proportional circles, choropleth, typology and flows. Some functions use 'shiny' or 'leaflet' technologies for dynamism and interactivity.

The great features are :

- Create maps in a web environment where the parameters are modifiable on the fly ('shiny' and 'leaflet' technologies).
- Create interactive maps through zoom and pop-up ('leaflet' technology).
- Create frozen maps with the possibility to add labels.

**License** GPL (>= 2.0) | file LICENSE

**LazyLoad** yes

**URL** <https://github.com/insee-psar-at/oceanis-package/>

**BugReports** <https://github.com/insee-psar-at/oceanis-package/issues>

**Depends** R (>= 3.6.0),shiny (>= 1.4.0.2),shinyBS (>= 0.61)

**Imports** classInt (>= 0.4.2),sf (>= 0.9.0),shinyjs (>= 1.1),shinythemes (>= 1.1.2),DT (>= 0.12),stringr (>= 1.4.0),leaflet (>= 2.0.3),leaflet.extras (>= 1.0.0),ggplot2 (>= 3.2.0),dplyr (>= 0.8.4),lwgeom (>= 0.2.1),zip (>= 2.1.1),htmlwidgets (>= 1.5.4),webshot (>= 0.5.2)

**Suggests** knitr, rmarkdown, markdown, rio

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.2.0

**NeedsCompilation** no

**Author** Sébastien Novella [aut, cre],  
Sébastien Calvet [aut],  
Nicolas Chauvot [aut],  
PSAR-AT Unité CED - INSEE PACA [ctb],

SED - DR Hauts-de-France - INSEE [ctb],  
 PSAR-SL - DR Auvergne-Rhone-Alpes - INSEE [ctb],  
 PSAR-EER - DR Occitanie - INSEE [ctb],  
 SED - DR Pays-de-la-Loire - INSEE [ctb],  
 Institut national de la statistique et des études économiques [cph]

**Maintainer** Sébastien Novella <sebastien.novella@insee.fr>

**Repository** CRAN

**Date/Publication** 2022-07-13 13:10:02 UTC

## R topics documented:

oceanis-package . . . . .	3
add_legende_ . . . . .	4
add_source . . . . .	7
add_titre . . . . .	8
add_typo_symboles . . . . .	9
affiche_palette . . . . .	10
calcul_ . . . . .	11
coordonnees_etiquettes . . . . .	13
coord_legende . . . . .	14
distrib_variable . . . . .	15
donnees . . . . .	16
export_ . . . . .	19
export_qgis_ . . . . .	20
largeur_fleche . . . . .	22
leaflet_ . . . . .	23
plot_ . . . . .	30
rapport_ronds . . . . .	38
rayon_ronds . . . . .	39
recup_palette . . . . .	41
set_bordure_ronds . . . . .	42
set_couleur_ . . . . .	43
set_fonds_simples . . . . .	46
set_opacite_elargi . . . . .	47
set_pop_up . . . . .	48
set_style_oursins . . . . .	49
shiny_ . . . . .	50
zonage_a_facon . . . . .	55

**Index**

**57**

---

oceanis-package

Package : oceanis

---

## Description

Type : Package

Version : 1.0.3

Date : 2019-09-06

License : GPL(>=2.0)

## Details

Outil de cartographie permettant de réaliser des cartes d'études et d'analyses (ronds proportionnels, analyse en classes, typologie, oursins, fleches joignantes et fleches saphirs).

Le package oceanis répond a 3 types de besoins :

- Créer des cartes fixes avec la possibilité d'ajouter des étiquettes. Idéal pour les impressions de documents (.pdf ou autres).
- Créer des cartes interactives grâce au zoom et aux pop-up (technologie leaflet). Idéal pour l'intégration dans une application (en shiny par exemple) ou dans une présentation en direct (en HTML).
- Créer des cartes dans un environnement web où les paramètres sont modifiables à la volée (technologie R-Shiny + leaflet). Idéal pour une visualisation rapide des données ou pour une présentation en direct. Ici, la carte est externalisée vers une nouvelle fenêtre du navigateur.

Les fonctions de création de cartes sont classées en 3 catégories, correspondant aux 3 besoins précédents. Elles sont respectivement préfixées par `plot_`, `leaflet_` et `shiny_`.

D'autres fonctions existent pour compléter l'offre :

Des fonctions pour exporter une carte leaflet en projet Qgis, des fonctions pour modifier son apparence, des fonctions permettant de créer ses propres zonages (connus ou à façon) à partir de tableaux de données.

Un exemple possible d'enchaînement de fonctions :

- 1- Chargement du package oceanis (`library(oceanis)`)
- 2- Chargement des données avec la fonction `import()` du package rio
- 3- Chargement des fonds de carte avec la fonction `read_sf()` du package sf
- 4- Appel à une fonction de création de carte (`shiny_()`, `leaflet_()` ou `plot_()`)
- 5- Modification du paramétrage (legende, fonds supplémentaire, taille, couleur, habillage...)
- 6- Export de la carte en projet Qgis (uniquement pour les fonctions `shiny_()` et `leaflet_()`)
- ou 6-bis- Export de la carte en format image.

Pour résumer, voici un comparatif des fonctionnalités proposées :

PLOT

Interactivité : Non

Personnalisation / Parametrages : Fort  
 Integration dans les applications : Moyen  
 Export Qgis : Non  
 Etiquettes : Oui  
 Habillage : Fort  
 Zonage a facon : Oui  
 Representation elargie : Non  
**LEAFLET**  
 Interactivite : Oui  
 Personnalisation / Parametrages : Moyen  
 Integration dans les applications : Fort  
 Export Qgis : Oui  
 Etiquettes : Non  
 Habillage : Fort  
 Zonage a facon : Oui  
 Representation elargie : Oui  
**R-SHINY**  
 Interactivite : Oui  
 Personnalisation / Parametrages : Faible  
 Integration dans les applications : Moyen  
 Export Qgis : Oui  
 Etiquettes : Non  
 Habillage : Faible  
 Zonage a facon : Oui  
 Representation elargie : Oui

---

 add\_legende\_

*Add a legend on a 'leaflet' map*


---

### **Description**

Add a legend on a 'leaflet' map for proportional circles, choropleth, typology, proportional and sapphire arrows.

**Usage**

```
add_legende_fonds_simples(map, titre = NULL, lng = NULL, lat = NULL,
  labels = NULL, choixLeg = NULL, zoom = 8, map_leaflet = NULL)
```

```
add_legende_ronds(map, titre = NULL, lng = NULL, lat = NULL, precision = 0,
  zoom = 8, map_leaflet = NULL)
```

```
add_legende_classes(map, titre = NULL, lng = NULL, lat = NULL, typeLegende =
  1, zoom = 8, map_leaflet = NULL)
```

```
add_legende_typo(map, titre = NULL, lng = NULL, lat = NULL, labels = NULL,
  zoom = 8, map_leaflet = NULL)
```

```
add_legende_typo_symboles(map, titre = NULL, lng = NULL, lat = NULL, labels
  = NULL, zoom = 8, map_leaflet = NULL)
```

```
add_legende_joignantes(map, titre = NULL, lng = NULL, lat = NULL, precision
  = 0, zoom = 8, map_leaflet = NULL)
```

```
add_legende_saphirs(map, titre = NULL, lng = NULL, lat = NULL, precision =
  0, zoom = 8, map_leaflet = NULL)
```

**Arguments**

map	objet leaflet.
titre	chaîne de caractères (character). Titre de la légende des classes. Par défaut a NULL.
lng	valeur numérique (numeric). Longitude (x) de la légende des classes dans le système de projection WGS84 (code EPSG 4326). Par défaut a NULL.
lat	valeur numérique (numeric). Latitude (y) de la légende des classes dans le système de projection WGS84 (code EPSG 4326). Par défaut a NULL.
typeLegende	valeur numérique (numeric). 1 (par défaut) pour une légende littérale, 2 pour une légende en échelle.
labels	vecteur de caractères (character). Labels personnalisés pour la légende de la carte en typologie. Par défaut a NULL.
choixLeg	vecteur de numériques (numeric). Choix des couches à afficher dans la légende. Par défaut a NULL (toutes les couches présentes).
precision	valeur numérique (numeric). Arrondit les valeurs de la légende des ronds ou des flèches. 0 (par défaut) pour arrondir à l'unité, -1 pour arrondir à la dizaine, -2 à la centaine...
zoom	valeur numérique (numeric). Valeur entre 6 et 10 (8 par défaut). Définit le niveau de zoom de la carte pour un affichage optimal.
map_leaflet	objet leaflet. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut a NULL.

## Details

Par défaut, lng et lat sont NULL ; la légende est alors positionnée en haut à droite de la carte sauf pour l'analyse en classes où elle est positionnée en bas.

Les valeurs lng et lat peuvent être négatives. Une longitude à 0 correspond au méridien de Greenwich.

Pour déplacer la légende, re exécutez la fonction `add_legende_xxx` en spécifiant les paramètres lng et lat.

Pour supprimer la légende, re exécutez la fonction `add_legende_xxx` en laissant les paramètres lng et lat à NULL.

Pour supprimer le titre, re exécutez la fonction `add_legende_xxx` en laissant le paramètre titre à NULL.

Concernant la légende des ronds proportionnels, le grand cercle correspond à la valeur max en volume et le petit cercle au tiers de la valeur max.

Idem pour la largeur des flèches joignantes et des flèches saphirs.

Les valeurs de la légende des ronds et des flèches peuvent être arrondies grâce au paramètre `precision`.

Pour la légende des cartes en typologie, il est possible de spécifier ses propres labels sans avoir besoin de modifier les noms de variables dans la table de données.

Si le titre et les valeurs de légende sont trop décalés, il faut modifier la valeur du zoom : entre 6 (zoom maximal, niveau commune) et 10 (zoom minimal, niveau France metro)

## Value

Retourne un objet de type leaflet.

## See Also

[coord\\_legende](#)

[leaflet\\_fonds\\_simples](#)

[leaflet\\_ronds](#) [leaflet\\_classes](#) [leaflet\\_ronds\\_classes](#) [leaflet\\_classes\\_ronds](#) [leaflet\\_typo](#)

[leaflet\\_oursins](#) [leaflet\\_joignantes](#) [leaflet\\_saphirs](#)

## Examples

```
data("depm")
data("regm")

data("donnees_monoloc")

# Ronds proportionnels sur une analyse en classes
map <- leaflet_classes_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4)
map <- add_legende_classes(map = map, titre = "VAR_AN_MOY", lng = 9, lat = 48, zoom = 6)
map <- add_legende_ronds(map = map, titre = "POP_2015", lng = 9, lat = 50, zoom = 6)
```

```
map

data("donnees_biloc")

# Fleches joignantes
map <- leaflet_joignantes(data = donnees_biloc, fondMaille = regm, typeMaille = "REG",
  idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", filtreDist = 1000, filtreMajeurs = 3)
map <- add_legende_joignantes(map = map, titre = "MIGR")

map
```

---

add\_source                      *Add a source on a 'leaflet' map*

---

## Description

Add a source on a 'leaflet' map

## Usage

```
add_source(map, source)
```

## Arguments

map	objet leaflet.
source	chaîne de caractères (character). Source de la carte.

## Details

Pour supprimer la source, reexecutez la fonction add\_source en laissant le paramètre source a NULL.

## Value

Retourne un objet de type leaflet.

## See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#),  
[leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

## Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015")
map <- add_source(map = map, source = "Source : INSEE - RP2016")

map
```

---

add\_titre

*Add a title on a 'leaflet' map*

---

## Description

Add a title on a 'leaflet' map

## Usage

```
add_titre(map, titre, sousTitre = NULL)
```

## Arguments

map	objet leaflet.
titre	chaîne de caractères (character). Titre de la carte.
sousTitre	chaîne de caractères (character). Sous-titre de la carte. Par défaut a NULL.

## Details

Pour supprimer le titre ou le sous-titre, reexecutez la fonction `add_titre` en laissant le paramètre `titre` ou `sousTitre` a NULL.

## Value

Retourne un objet de type leaflet.

## See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#), [leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)



## Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015")
map <- add_titre(map = map,
titre = "Population des departements de France metropolitaine en 2015")

map
```

---

add\_typo\_symboles      *Add symbols on a 'leaflet' map*

---

## Description

Add a symbols on a 'leaflet' map for typology.

## Usage

```
add_typo_symboles(map, fondPoints, types = NULL, couleurs = NULL,
tailles = NULL, epaisseurs = NULL)
```

## Arguments

map	objet leaflet.
fondPoints	objets sf. Fond de points.
types	vecteur de numeriques (numeric). Par default a NULL. Voir details.
couleurs	vecteur de caracteres (character). Par default a NULL. Voir details.
tailles	vecteur de numeriques (numeric). Par default a NULL. Voir details.
epaisseurs	vecteur de numeriques (numeric). Par default a NULL. Voir details.

## Details

Pour ajouter des symboles a la carte leaflet, il faut fournir un fond de points et si besoin preciser le type des points (par default un carre plein - voir l'aide de `?graphics::points`), leurs couleurs (par default rouge), leurs tailles (par default 30) et l'epaisseur de la bordure du symbole si ce dernier n'est pas plein.

Attention a l'ordre des elements des vecteurs qui doivent correspondre a l'ordre du fond de points.

## Value

Retourne un objet de type leaflet.

**See Also**

[coord\\_legende](#)  
[leaflet\\_fonds\\_simples](#)  
[leaflet\\_ronds](#) [leaflet\\_classes](#) [leaflet\\_ronds\\_classes](#) [leaflet\\_classes\\_ronds](#) [leaflet\\_typo](#)  
[leaflet\\_oursins](#) [leaflet\\_joignantes](#) [leaflet\\_saphirs](#)

**Examples**

```

data("depm")
data("regm")

# Typologie en symboles
map <- leaflet_fonds_simples(listFonds = list(depm))
points <- suppressWarnings(sf::st_centroid(depm[depm$CODE=="75",]))
points <- rbind(points,suppressWarnings(sf::st_centroid(depm[depm$CODE=="13",])))
points <- rbind(points,suppressWarnings(sf::st_centroid(depm[depm$CODE=="69",])))
types <- c(16,15,15)
couleurs <- c("blue","red","red")
tailles <- c(25,20,20)
map <- add_typo_symboles(map, fondPoints=points, types = types,
couleurs = couleurs, tailles = tailles)

map

```

---

affiche\_palette

*Returns a palette colors of the graphic chart of INSEE*


---

**Description**

Returns a palette colors of the graphic chart of INSEE.

**Usage**

```
affiche_palette(nomPalette = "defaut")
```

**Arguments**

**nomPalette**      chaîne de caracteres (character). A choisir parmi "defaut", "Insee\_Rouge", "Insee\_Jaune", "Insee\_Bleu", "Insee\_Turquoise", "Insee\_Vert", "Insee\_Violet", "Insee\_Gris", "Turquoise\_Neg", "Vert\_Neg", "Violet\_Neg" ou "Gris\_Neg".

### Details

Les couleurs de palettes préfixées par "Insee\_" sont celles de la charte graphique INSEE. La fonction `affiche_palette(nomPalette)` permet de visualiser les couleurs de chaque palette disponible. Les modalités pour l'argument 'nomPalette' sont "Insee\_Rouge", "Insee\_Jaune", "Insee\_Bleu", "Insee\_Turquoise", "Insee\_Vert", "Insee\_Violet" et "Insee\_Gris" pour les palettes de la charte graphique INSEE, "defaut", "Turquoise\_Neg", "Vert\_Neg", "Violet\_Neg" et "Gris\_Neg" pour les palettes hors charte.

La palette "defaut" reprend les couleurs de la palette RdYIBu de ColorBrewer.

Seules les palettes "Insee\_Rouge" et "Insee\_Jaune" proposent des palettes pour des valeurs négatives et positives. Pour ces deux palettes, la palette "Insee\_Bleu" est utilisée pour représenter les valeurs négatives. Cette dernière peut également être utilisée seule pour des valeurs uniquement positives ou négatives.

La fonction permet de visualiser tous les dégradés possibles pour une palette donnée.

La liste des valeurs hexadécimales de toutes les palettes est disponible en chargeant les données suivantes : `data("palettes_insee")`

### Value

Affiche la palette dans l'onglet Plots.

### References

Un convertisseur de couleurs pour visualiser une couleur à partir de son nom, son code hexadécimal ou RGB : <http://www.proftnj.com/RGB3.htm>

ColorBrewer 2.0 : <https://colorbrewer2.org/#type=diverging&scheme=RdYIBu&n=10>

### See Also

[set\\_couleur\\_classes](#)

### Examples

```
affiche_palette(nomPalette = "defaut")
```

---

calcul\_

*Calculating a class variable*

---

### Description

Add to a data table a column whose values can be represented in a class analysis (ratio, part, evolution ...).

The calculation is based on volume data (from the population, for example).

**Usage**

```

calcul_ratio(data, var1, var2)

calcul_tx_evol_global(data, var1, var2)

calcul_tx_evol_ann_moy(data, var1, var2, nbAnnees)

calcul_part_ens(data, var)

calcullette(data, formule = NULL)

```

**Arguments**

data	tableau de donnees (data.frame).
var1	chaîne de caracteres (character). Variable en volume de la table.
var2	chaîne de caracteres (character). Variable en volume de la table.
nbAnnees	nombre (numeric). Nombre d'annees applicable au taux d'evolution annuel moyen.
var	chaîne de caracteres (character). Variable en volume de la table.
formule	chaîne de caracteres (character). Formule libre pour le calcul d'une variable de classes. Seuls les caracteres + - * / ^ ( ) sont acceptes.

**Details**

Les formules utilisees sont :

```

ratio (data[,var1]/data[,var2])*100
tx_evol_global ((data[,var2]-data[,var1])/data[,var1])*100
tx_evol_ann_moy ((data[,var2]/data[,var1])^(1/nbAnnees)-1)*100
part_ens (data[,var]/sum(data[,var],na.rm = TRUE))*100

```

**Value**

Retourne un objet data.frame.

**Examples**

```

data("donnees_monoloc")

ratio <- calcul_ratio(data = donnees_monoloc, var1 = "POP_2010", var2 = "POP_2015")
teg <- calcul_tx_evol_global(data = donnees_monoloc, var1 = "POP_2010", var2 = "POP_2015")
team <- calcul_tx_evol_ann_moy(data = donnees_monoloc, var1 = "POP_2010", var2 = "POP_2015",
nbAnnees = 5)
part <- calcul_part_ens(data = donnees_monoloc, var = "POP_2015")

```

---

`coordonnees_etiquettes`*Data table of labels*

---

## Description

Returns a data table for formatting labels and their position for a map in plot.

## Usage

```
coordonnees_etiquettes(fondMaille, listeCode)
```

## Arguments

<code>fondMaille</code>	objet sf. Fond de carte.
<code>listeCode</code>	vecteur de caracteres (character). Liste des codes de la maille a afficher sur la carte.

## Details

Le tableau des etiquettes contient le code, le libelle a afficher, les coordonnees x et y du centroid de la maille donc des etiquettes, la taille, le style et la couleur de la police. Le style de police (colonne FONT) est un entier : 1 (normal), 2 (gras, par default), 3 (italique) et 4 (gras italique).

Ce tableau peut etre modifie pour changer le style d'une ou plusieurs etiquettes mais aussi leur position (x et y) pour eviter le chevauchement.

Pour afficher la carte avec les etiquettes formatees, il faut passer ce tableau dans le parametre "etiquettes" d'une fonction plot.

## Value

Retourne un objet data.frame.

## See Also

[plot\\_ronds](#), [plot\\_classes](#), [plot\\_ronds\\_classes](#), [plot\\_classes\\_ronds](#), [plot\\_typo](#),  
[plot\\_oursins](#), [plot\\_joignantes](#), [plot\\_saphirs](#)

## Examples

```
data("donnees_monoloc")
data("depm")

etiquettes <- coordonnees_etiquettes(fondMaille = depm,
listeCode = c("06", "13", "31", "33", "44", "67", "69", "59", "75"))
etiquettes$LIBELLE <- c("Nice", "Marseille", "Toulouse", "Bordeaux", "Nantes",
"Lille", "Strasbourg", "Lyon", "Paris")
```

```
etiquettes[etiquettes$CODE=="75","TAILLE"] <- 1.3

# Ronds proportionnels sur une analyse en classes
fond_ronds <- plot_classes_ronds(data = donnees_monoloc, fondMaille = depm,
idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4,
titreLegRonds = "POP_2015", titreLegClasses = "VAR_AN_MOY",
xLegClasses = 1150000, yLegClasses = 6600000, etiquettes = etiquettes)
```

---

coord_legende	<i>Returns the lon / lat coordinates of a 'leaflet' map's legend</i>
---------------	--

---

### Description

Returns the lon / lat coordinates of a 'leaflet' map's legend for proportional circles, choropleth, typology, proportional and sapphire arrows.

### Usage

```
coord_legende(map)
```

### Arguments

map	objet leaflet.
-----	----------------

### Details

Au moins une legende doit exister sur la carte avant de pouvoir connaitre sa position. Les cartes en ronds proportionnels avec une analyse en classes ont 2 legendes.

### Value

Retourne un objet data.frame avec les coordonnees lon/lat de la (ou des) legende(s). Le type de legende est indique dans le row.names.

### See Also

[add\\_legende\\_ronds](#), [add\\_legende\\_classes](#),[add\\_legende\\_typo](#),  
[add\\_legende\\_joignantes](#), [add\\_legende\\_saphirs](#),  
[leaflet\\_ronds](#), [leaflet\\_classes](#),[leaflet\\_ronds\\_classes](#),  
[leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#),[leaflet\\_oursins](#), [leaflet\\_joignantes](#),  
[leaflet\\_saphirs](#)

## Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015")
map <- add_legende_ronds(map = map, titre = "POP_2015")

map

coord <- coord_legende(map)
map <- add_legende_ronds(map = map, titre = "POP_2015", lng = 8, lat = 50)

map
```

---

distrib\_variable      *Histogram of the distribution of the class variable*

---

## Description

Displays a histogram of the distribution of the class variable (ratio, part, evolution ...) based on the specified method, number of classes or breaks.

## Usage

```
distrib_variable(data, varRatio, methode = "kmeans", nbClasses = 3,
  bornes = NULL)
```

## Arguments

data	tableau de donnees (data.frame).
varRatio	chaîne de caracteres (character). Variable des classes de la table.
methode	chaîne de caracteres (character). A choisir parmi "kmeans" (par défaut), "fisher", "jenks" ou "quantile".
nbClasses	valeur numerique (numeric). Nombre de classes. Par défaut 3 classes.
bornes	vecteur de valeurs numeriques (numeric). Par défaut a NULL.

### Details

Si les bornes sont renseignées, aucune méthode est appliquée. Elle est considérée comme manuelle.

Si les données à représenter possèdent des valeurs négatives et positives, alors une borne de classe à zéro est gérée automatiquement pour les méthodes "kmeans", "fisher" et "jenks". La distribution des valeurs dans les classes, effectuée par la méthode spécifiée, est indépendante entre les valeurs négatives et les valeurs positives.

La méthode "quantile" ne gère pas de borne à zéro.

La fonction `ggplot2::ggplot` est utilisée pour la création de l'histogramme.

### Value

Retourne un objet de type `ggplot`

Affiche l'histogramme dans le plot.

### See Also

[ggplot](#)

### Examples

```
data("donnees_monoloc")  
  
distrib_variable(data = donnees_monoloc ,varRatio = "VAR_AN_MOY", nbClasses = 4)
```

---

donnees

*Examples of data*

---

### Description

Data tables for examples.

- `com_dep_13_30_83_84` for `zonage_a_facon`
- `donnees_a_facon` for `zonage_a_facon`
- `donnees_monoloc` for proportional circles, choropleth and typology
- `donnees_biloc` for proportionnal arrows
- `donnees_biloc_saphirs` for sapphire arrows
- `depm`, `regm`, `fram` for the meshes and contours of the maps
- `palettes_insee` for palette of the graphic chart of INSEE



**Usage**

```
data("com_dep_13_30_83_84")  
  
data("donnees_a_facon")  
  
data("donnees_monoloc")  
  
data("donnees_biloc")  
  
data("donnees_biloc_saphirs")  
  
data("dep")  
  
data("reg")  
  
data("fram")  
  
data("palettes_insee")
```

**Format**

com\_dep\_13\_30\_83\_84 Un data frame de 792 observations et 3 variables.

**"CODE"** vecteur caracteres  
**"LIBELLE"** vecteur caracteres  
**"geometry"** sfc\_GEOMETRY

Un data frame de 281 observations et 3 variables.

**"DEPCOM"** vecteur caracteres  
**"ZE2010"** vecteur caracteres  
**"LIB\_ZE2010"** vecteur caracteres

donnees\_monoloc Un data frame de 96 observations et 6 variables.

**"COD\_DEP"** vecteur caracteres  
**"LIB\_DEP"** vecteur caracteres  
**"POP\_2010"** vecteur numerique  
**"POP\_2015"** vecteur numerique  
**"VAR\_AN\_MOY"** vecteur numerique  
**"REG"** vecteur caracteres

donnees\_biloc Un data frame de 378 observations et 3 variables.

**"REG\_DEPART"** vecteur caracteres  
**"REG\_ARRIVEE"** vecteur caracteres  
**"MIGR"** vecteur numerique

donnees\_biloc\_saphirs Un data frame de 21 observations et 3 variables.

**"REG\_DEPART"** vecteur caracteres

**"REG\_ARRIVEE"** vecteur caracteres

**"MIGR"** vecteur numerique

depm Un data frame de 96 observations et 5 variables.

**"CODE"** vecteur caracteres

**"LIBELLE"** vecteur caracteres

**"REG"** vecteur caracteres

**"SURF"** vecteur numerique

**"geometry"** sfc\_GEOMETRY

regm Un data frame de 13 observations et 4 variables.

**"CODE"** vecteur caracteres

**"LIBELLE"** vecteur caracteres

**"SURF"** vecteur numerique

**"geometry"** sfc\_GEOMETRY

fram Un data frame de 1 observation et 4 variables.

**"CODE"** vecteur caracteres

**"LIBELLE"** vecteur caracteres

**"SURF"** vecteur numerique

**"geometry"** sfc\_GEOMETRY

palettes\_insee Une liste de vecteurs caracteres.

**"nomPalette"** vecteur caracteres de codes hexadecimaux

## Examples

```
data(com_dep_13_30_83_84)
data(donnees_a_facon)
data(donnees_monoloc)
data(donnees_biloc)
data(donnees_biloc_saphirs)
data(depm)
data(regm)
data(fram)
data(palettes_insee)
```

---

export\_                      *Export a 'leaflet' map in image format*

---

## Description

Export a 'leaflet' map in image format (.jpeg, .pdf or .png).

## Usage

```
export_jpeg(map, chemin, nomFichier)
```

```
export_pdf(map, chemin, nomFichier)
```

```
export_png(map, chemin, nomFichier)
```

## Arguments

map	objet leaflet.
chemin	chaîne de caractères (character). Chemin du dossier d'exportation de la carte.
nomFichier	chaîne de caractères (character). Nom du fichier en sortie sans l'extension.

## Details

Attention, l'export en format image peut durer quelques minutes.

## Value

Ne retourne aucun objet.

Exporte la carte dans le format .jpeg, .pdf ou .png à l'emplacement spécifique.

## See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#),  
[leaflet\\_typo](#), [leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

## Examples

```
data("depm")
data("regm")

data("donnees_monoloc")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015")
```

```
## Not run:  
export_jpeg(map = map, chemin = tempdir(), nomFichier = "map")  
export_pdf(map = map, chemin = tempdir(), nomFichier = "map")  
export_png(map = map, chemin = tempdir(), nomFichier = "map")  
## End(Not run)
```

---

export\_qgis\_

*Export a 'leaflet' map to Qgis project*

---

### Description

Export a 'leaflet' map to Qgis project.

### Usage

```
export_qgis_ronds(map, cheminDossier, nomFichier, titre1 = "", titre2  
= "", source = "")  
  
export_qgis_classes(map, cheminDossier, nomFichier, titre1 = "", titre2 =  
"", source = "")  
  
export_qgis_ronds_classes(map, cheminDossier, nomFichier, titre1 = "",  
titre2 = "", source = "")  
  
export_qgis_classes_ronds(map, cheminDossier, nomFichier, titre1 = "",  
titre2 = "", source = "")  
  
export_qgis_typo(map, cheminDossier, nomFichier, titre1 = "", titre2 = "",  
source = "")  
  
export_qgis_oursins(map, cheminDossier, nomFichier, titre1 = "", titre2 =  
"", source = "")  
  
export_qgis_joignantes(map, cheminDossier, nomFichier, titre1 = "", titre2 =  
"", source = "")  
  
export_qgis_saphirs(map, cheminDossier, nomFichier, titre1 = "", titre2 =  
"", source = "")
```

### Arguments

map	objet leaflet.
cheminDossier	chaîne de caractères (character). Chemin du dossier d'exportation du projet.
nomFichier	chaîne de caractères (character). Nom du projet en sortie sans l'extension.

titre1	chaîne de caractères (character). Titre principal de la carte, le plus souvent informatif.
titre2	chaîne de caractères (character). Titre secondaire de la carte, le plus souvent descriptif.
source	chaîne de caractères (character). Source de la carte.

### Details

Il faut obligatoirement ajouter la légende à la carte avant d'utiliser la fonction d'export Qgis.

Le projet .qgs est associé à tous les fonds ShapeFile qui composent la carte (.shp, .dbf, .prj, .shx).

Attention, l'export du projet Qgis peut durer quelques minutes.

### Value

Ne retourne aucun objet.

Exporte la carte en projet Qgis à l'emplacement spécifique.

### See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#),  
[leaflet\\_typo](#), [leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

### Examples

```
data("depm")
data("regm")

data("donnees_monoloc")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015")
map <- add_legende_ronds(map = map, titre = "POP_2015", lng = 9, lat = 50, zoom = 6)

## Not run:
export_qgis_ronds(map = map, cheminDossier = tempdir(), nomFichier = "projet_qgis",
  source = "Source : INSEE - RP2016")
## End(Not run)
```

---

largeur_fleche	<i>Width of the arrow for 'leaflet' maps</i>
----------------	--

---

### Description

Width of the largest arrow of the map in kilometers for 'leaflet' maps.

### Usage

```
largeur_fleche(map)
```

### Arguments

map                    objet leaflet.

### Details

La largeur de la fleche est disponible uniquement pour les representations de fleches joignantes et de fleches saphirs.

### Value

Retourne une valeur numerique.

### See Also

[leaflet\\_joignantes](#) [leaflet\\_saphirs](#)

### Examples

```
data("donnees_biloc")
data("regm")

# Fleches joignantes
map <- leaflet_joignantes(data = donnees_biloc, fondMaille = regm, typeMaille = "REG",
  idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", filtreDist = 1000, filtreMajeurs = 3)

map

largeur_fleche(map)
# [1] 100

# Fleches joignantes
map <- leaflet_joignantes(data = donnees_biloc, fondMaille = regm, typeMaille = "REG",
  idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", largeurFlecheMax = 200, filtreDist = 1000,
  filtreMajeurs = 3)
```

```
map

largeur_fleche(map)
# [1] 200
```

---

leaflet\_                      *Creation 'leaflet' maps*

---

## Description

Create interactive maps for statistical analysis by 'leaflet' (zoom + pop-up).

## Usage

```
leaflet_fonds_simples(listFonds, popup = NULL, init = TRUE, map =
NULL)
```

```
leaflet_ronds(data, fondMaille, fondMailleElargi = NULL, fondSuppl = NULL,
idData, varVolume, rayonRond = NULL, rapportRond = NULL, emprise = "FRM",
fondEtranger = NULL, fondChx = NULL, colPos = "#EB617F", colNeg = "#286AC7",
colBorderPos = "white", colBorderNeg = "white", epaisseurBorder = 1,
opacityElargi = 0.6, zoomMaille = NULL, map_proxy = NULL)
```

```
leaflet_classes(data, fondMaille, fondMailleElargi = NULL, fondSuppl = NULL,
idData, varRatio, methode = "kmeans", nbClasses = 3, bornes = NULL,
stylePalette = "defaut", opacityElargi = 0.6, colBorder = "white", precision
= 1, emprise = "FRM", fondEtranger = NULL, zoomMaille = NULL, map_proxy =
NULL)
```

```
leaflet_ronds_classes(data, fondMaille, fondMailleElargi = NULL, fondSuppl =
NULL, idData, varVolume, varRatio, rayonRond = NULL, rapportRond = NULL,
methode = "kmeans", nbClasses = 3, bornes = NULL, stylePalette = "defaut",
opacityElargi = 0.6, colBorderPos = "white", colBorderNeg = "white",
epaisseurBorder = 1, precision = 1, emprise = "FRM", fondEtranger = NULL,
fondChx = NULL, zoomMaille = NULL, map_proxy = NULL)
```

```
leaflet_classes_ronds(data, fondMaille, fondMailleElargi = NULL, fondSuppl =
NULL, idData, varVolume, varRatio, rayonRond = NULL, rapportRond = NULL,
methode = "kmeans", nbClasses = 3, bornes = NULL, stylePalette = "defaut",
opacityElargi = 0.6, colBorderClasses = "white", colBorderRondsPos =
"#303030", colBorderRondsNeg = "#303030", epaisseurBorder = 1.5, precision =
1, emprise = "FRM", fondEtranger = NULL, fondChx = NULL, zoomMaille = NULL,
map_proxy = NULL)
```

```
leaflet_typo(data, fondMaille, fondSuppl = NULL, idData, varTypo, emprise =
```

```
"FRM", fondEtranger = NULL, zoomMaille = NULL, map_proxy = NULL)
```

```
leaflet_oursins(data, fondMaille, fondSuppl = NULL, idDataDepart,
idDataArrivee, varFlux, filtreVol = 0, filtreDist = 100, filtreMajeurs = 10,
decalageAllerRetour = 0, decalageCentroid = 0, emprise = "FRM", fondEtranger
= NULL, zoomMaille = NULL, map_proxy = NULL)
```

```
leaflet_joignantes(data, fondMaille, typeMaille, fondSuppl = NULL,
idDataDepart, idDataArrivee, varFlux, largeurFlecheMax = NULL, filtreVol =
0, filtreDist = 100, filtreMajeurs = 10, decalageAllerRetour = 0,
decalageCentroid = 0, colFleche = "#286AC7", colBorder = "#303030", emprise
= "FRM", fondEtranger = NULL, zoomMaille = NULL, map_proxy = NULL)
```

```
leaflet_saphirs(data, fondMaille, typeMaille, fondSuppl = NULL,
idDataDepart, idDataArrivee, varFlux, largeurFlecheMax = NULL, direction =
"Ent", filtreVol = 0, colEntree = "#EB617F", colSortie = "#286AC7",
colBorder = "#303030", emprise = "FRM", fondEtranger = NULL, zoomMaille =
NULL, map_proxy = NULL)
```

## Arguments

<code>data</code>	tableau de donnees (data.frame).
<code>fondMaille</code>	objet sf. Fond de carte.
<code>listFonds</code>	list d'objets sf. Liste de fonds de carte.
<code>popup</code>	vecteur de numeriques (numeric). Choix des couches avec les popup actifs. Par default a NULL (toutes les couches ont les popup actifs).
<code>init</code>	booleen. TRUE pour creer une carte avec des fonds d'habillage. FALSE pour des fonds d'analyses. Par default a TRUE.
<code>map</code>	objet leaflet. Permet d'ajouter des couches a une carte existante. Par default a NULL.
<code>typeMaille</code>	chaîne de caracteres (character). Type de maille issus d'un zonage administratifs ou d'etudes ("REG", "DEP", "UU", "AU", "ZE", "BV", "EPCI" ou "COM").
<code>fondMailleElargi</code>	objet sf. Fond de carte. Par default a NULL.
<code>fondSuppl</code>	objet sf. Fond de carte. Par default a NULL.
<code>idData</code>	chaîne de caracteres (character). Variable identifiant la maille.
<code>idDataDepart</code>	chaîne de caracteres (character). Variable identifiant le depart du flux.
<code>idDataArrivee</code>	chaîne de caracteres (character). Variable identifiant l'arrivee du flux.
<code>varVolume</code>	chaîne de caracteres (character). Variable en volume pour des ronds proportionnels.
<code>varRatio</code>	chaîne de caracteres (character). Variable en ratio pour des analyses en classes.
<code>varTypo</code>	chaîne de caracteres (character). Variable de typologie.
<code>varFlux</code>	chaîne de caracteres (character). Variable de flux pour des oursins, fleches joignantes ou fleches saphirs.



rayonRond	valeur numerique (numeric). Rayon du rond le plus grand en metres. Par default a NULL.
rapportRond	valeur numerique (numeric). Rapport entre l'aire du rond le plus grand et la valeur maximale des donnees en volume. Par default a NULL.
methode	chaîne de caracteres (character). A choisir parmi "kmeans" (par default), "fisher", "jenks" ou "quantile".
nbClasses	valeur numerique (numeric). Nombre de classes. Par default 3 classes.
bornes	vecteur de valeurs numeriques (numeric). Le nombre de bornes correspond au nombre de classes -1. Par default a NULL.
precision	valeur numerique (numeric). Arrondi des valeurs en ratio pour le calcul des bornes de classes. Par default, 1 chiffre apres la virgule.
largeurFlecheMax	valeur numerique (numeric). Valeur correspondant a la largeur de la fleche la plus grande de la carte (km). Par default a NULL.
direction	chaîne de caracteres (character). Pour les fleches saphirs, type de flux a choisir parmi "Ent" pour entrees (par default), "Sor" pour sorties et "Sol" pour soldes (entrees - sorties).
filtreVol	valeur numerique (numeric). Filtre sur la valeur minimale du nombre de flux a afficher. Par default a 0 : affichage de tous les flux.
filtreDist	valeur numerique (numeric). Filtre sur la distance maximale des fleches pour les oursins et les fleches joignantes. Par default a 100 km : affichage des flux a moins de 100 km.
filtreMajeurs	valeur numerique (numeric). Filtre sur le nombre de flux entrants et sortants les plus importants d'un territoire. Par default a 10 : les 10 flux les plus importants sortant d'un territoire ou y entrant.
decalageAllerRetour	valeur numerique (numeric). Decalage entre les fleches aller-retour (km).
decalageCentroid	valeur numerique (numeric). Decalage des fleches depuis et vers les centroides des territoires (km).
colPos	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#EB617F".
colNeg	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#286AC7".
colBorder	chaîne de caracteres (character). Couleur nommee (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par default "white" pour les classes, "#303030" pour les fleches.
colBorderPos	chaîne de caracteres (character). Couleur nommee (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par default "white".
colBorderNeg	chaîne de caracteres (character). Couleur nommee (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par default "white".
colBorderRondsPos	chaîne de caracteres (character). Couleur nommee (par exemple "grey") ou hexadecimal (par exemple "#808080"). Par default "#303030".

colBorderRondsNeg	chaîne de caractères (character). Couleur nommée (par exemple "grey") ou hexadécimal (par exemple "#808080"). Par défaut "#303030".
colBorderClasses	chaîne de caractères (character). Couleur nommée (par exemple "grey") ou hexadécimal (par exemple "#808080"). Par défaut "white".
epaisseurBorder	chaîne de caractères (numeric). Épaisseur de la bordure des ronds. Par défaut a 1".
stylePalette	chaîne de caractères (character). Palette de la charte INSEE. A choisir entre "defaut", "Insee_Rouge", "Insee_Jaune", "Insee_Bleu", "Insee_Turquoise", "Insee_Vert", "Insee_Violet" ou "Insee_Gris" (voir 'affiche_palette(nomPalette)').
colFleche	chaîne de caractères (character). Couleur nommée (par exemple "orange") ou hexadécimal (par exemple "#FFA500"). Par défaut "#286AC7".
colEntree	chaîne de caractères (character). Couleur nommée (par exemple "orange") ou hexadécimal (par exemple "#FFA500"). Par défaut "#EB617F".
colSortie	chaîne de caractères (character). Couleur nommée (par exemple "blue") ou hexadécimal (par exemple "#0000FF"). Par défaut "#286AC7".
opacityElargi	valeur numérique (numeric). Opacité de la représentation élargie. Par défaut a 0.6 (valeur comprise entre 0-transparent et 1-opaque).
emprise	chaîne de caractères (character). "FRM" (par défaut pour la France métropolitaine), "971" (Guadeloupe), "972" (Martinique), "973" (Guyane), "974" (La Réunion), "976" (Mayotte) ou "999" (Étranger). Ce paramètre permet de définir le système de projection de la carte.
fondEtranger	objet sf. Fond de carte d'habillage personnalisable pour les fonds autres que la France (métropolitaine et DOM). Par défaut a NULL.
fondChx	objet sf. Fond des chefs-lieux. Pour les ronds proportionnels avec une maille communale, choix entre la position du centre des ronds sur les centroïdes des communes (NULL par défaut) ou sur les chefs-lieux (chx) des communes.
zoomMaille	vecteur de caractères ou numériques (character ou numeric). Identifiant(s) de la maille du paramètre fondMaille. Permet de zoomer sur une ou plusieurs entités de la maille. Par défaut a NULL (zoom sur l'emprise de la maille).
map_proxy	objet leaflet ou leaflet_proxy. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut a NULL.

## Details

Les cartes leaflet sont interactives, il est alors possible :

- de déplacer la carte avec un cliquer-glisser de la souris.
- de zoomer/dezoomer la carte avec la roulette de la souris ou les boutons +/- en haut à gauche de la carte. Le niveau de zoom peut varier de l'échelle de la France métropolitaine à l'échelle communale.
- de faire apparaître des informations en cliquant sur des territoires. En cliquant, par exemple, sur une maille d'une analyse en classes, un pop-up s'affiche indiquant le nom du territoire et sa valeur

en ratio. En cliquant ailleurs, le 1er pop-up est remplacé par un nouveau. On ferme le pop-up en cliquant sur sa croix.

- d'ajouter une légende, des éléments contextuels et de modifier son apparence via les fonctions `oceanis` associées.

Les fonctions `leaflet_()` retournent un objet de type `leaflet`. Si vous exécutez une fonction `leaflet_()` sans récupérer le résultat dans une variable, la carte s'affichera automatiquement dans la fenêtre viewer de RStudio mais ne pourra pas être réutilisée pour être modifiée. Si vous sauvegardez le résultat dans une variable, la carte ne s'affichera qu'à l'exécution de cette variable (voir exemples).

Il est possible de remplacer le fond d'habillage par défaut (France métropolitaine ou DOM) par un autre fond. Pour cela, il faut spécifier le paramètre `emprise="999"` ET ajouter un fond `sf` dans le paramètre `fondEtranger`, par exemple, une carte de l'Europe ou d'un pays particulier. Le système de coordonnées du fond doit être projeté (en unité de longueur) et non géographique (lng/lat). Si aucun code EPSG valide n'est trouvé pour ce fond, le code EPSG 3395 est choisi (projection Mercator). Attention car la légende des ronds sera d'autant plus déformée qu'elle se situera loin de l'équateur.

Les fonctions `leaflet` peuvent être intégrées dans les applications shiny. Le paramètre `map_proxy` permet d'actualiser uniquement les couches modifiées et non toute la carte entière (voir vignette).

**Fonds simples** Il s'agit d'une carte sans analyse avec uniquement des fonds. Le premier fond de la liste est positionné derrière la carte et le dernier fond de la liste devant. La personnalisation des fonds se fait avec la fonction `set_fonds_simples`.

**Ronds proportionnels** L'intérieur des cercles est de couleur orange (couleur par défaut) pour représenter les valeurs positives de la variable en volume et bleu pour les valeurs négatives.

Le paramètre `fondChx` est à renseigner UNIQUEMENT si la maille est communale.

Pour information, le chef-lieu (ou `chx`) est un point de la commune autour duquel la population est la plus dense. Le plus souvent, il s'agit de l'hôtel de ville de la commune.

**Analyse en classes** En semiologie, il est déconseillé de multiplier le nombre de classes pour des raisons de visibilité et de clarté de la carte.

Si les données ne comprennent que des valeurs positives ou que des valeurs négatives, il est conseillé 5 classes au maximum.

Si les données comprennent à la fois des valeurs positives et négatives, le nombre maximum conseillé est 9 classes.

Il est possible de spécifier manuellement les bornes des classes en implémentant le paramètre `bornes` d'un vecteur de valeurs numériques de type `c(bornes1, bornes2, ...)`. Le nombre de bornes correspond au nombre de classes -1 (`length(bornes) == nbClasses - 1`). Il n'est pas obligatoire de spécifier les valeurs min et max.

Les palettes de couleurs disponibles sont celles de la Charte Graphique INSEE. En exécutant la fonction `affiche_palette(nomPalette)`, il est possible de visualiser les couleurs de chaque palette disponible. À choisir entre "default", "Insee\_Rouge", "Insee\_Jaune", "Insee\_Bleu", "Insee\_Turquoise", "Insee\_Vert", "Insee\_Violet" ou "Insee\_Gris".

**Analyse en classes dans ronds proportionnels** L'analyse en classes colore l'intérieur des ronds (fonction `leaflet_ronds_classes()`)

**Ronds proportionnels sur analyse en classes** Les ronds proportionnels sont posés sur l'analyse en classes (fonction `leaflet_classes_ronds()`)

**Typologie** Même conseil que pour les analyses en classes. Le choix des couleurs ne doit pas être un dégradé si il n'existe pas de classement dans les modalités.

**Oursins** Les oursins representent uniquement l'existence d'un flux entre 2 territoires. Le sens du flux et le volume ne sont pas pris en compte.

Les filtres sur le volume, la distance et les flux majeurs sont independants. Chacun d'eux s'appliquent sur les donnees initiales. Autrement dit, le filtre sur les flux majeurs ne se fera pas apres un eventuel filtre sur la distance mais sur toutes les donnees en entree.

Le filtre sur le volume est plutot complementaire au filtre sur les flux majeurs. En effet, il est possible de vouloir representer des faibles flux significatifs. Le filtre sur les flux majeurs le permet contrairement au filtre sur le volume seul.

**Fleches joignantes** Les fleches joignantes representent un flux entre 2 territoires, son sens et son volume. Dans le cas de flux aller et retour, deux fleches de sens oppose sont superposees.

La regle des filtres est la meme que pour les oursins.

**Fleches saphirs** Les fleches saphirs convergent vers ou divergent d'un meme territoire. Elles permettent de representer les entrees, les sorties ou le solde (entrees - sorties) d'un territoire par rapport aux autres.

Pour un solde positif, les fleches sont representees en orange, pour un solde negatif en bleu.

Une attention particuliere est a apporter au choix de la variable idDataDepart et idDataArrivee selon si le flux est entrant, sortant ou en solde.

## Value

Retourne un objet leaflet.

## See Also

[zonage\\_a\\_facon](#),  
[rayon\\_ronds](#), [rapport\\_ronds](#), [largeur\\_fleche](#),  
[calcul\\_ratio](#), [calcul\\_tx\\_evol\\_global](#), [calcul\\_tx\\_evol\\_ann\\_moy](#), [calcul\\_part\\_ens](#), [calcullette](#),  
[distrib\\_variable](#),  
[add\\_legende\\_ronds](#), [add\\_legende\\_classes](#), [add\\_legende\\_typo](#), [add\\_legende\\_typo\\_symboles](#),  
[add\\_legende\\_joignantes](#), [add\\_legende\\_saphirs](#),  
[coord\\_legende](#),  
[add\\_source](#), [add\\_titre](#),  
[set\\_fonds\\_simples](#),  
[export\\_qgis\\_ronds](#), [export\\_qgis\\_classes](#), [export\\_qgis\\_ronds\\_classes](#), [export\\_qgis\\_classes\\_ronds](#),  
[export\\_qgis\\_typo](#), [export\\_qgis\\_oursins](#), [export\\_qgis\\_joignantes](#), [export\\_qgis\\_saphirs](#),  
[export\\_jpeg](#), [export\\_pdf](#), [export\\_png](#)

## Examples

```
data("depm")  
data("regm")  
  
data("donnees_monoloc")
```

```
# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015")

  map

# Analyse en classes
map <- leaflet_classes(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varRatio = "VAR_AN_MOY", nbClasses = 4)

  map

# Analyse en classes dans les ronds proportionnels
map <- leaflet_ronds_classes(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4)

  map

# Ronds proportionnels sur une analyse en classes
map <- leaflet_classes_ronds(data = donnees_monoloc, fondMaille = depm,
  idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4)

  map

# Typologie
map <- leaflet_typo(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
  varTypo = "REG")

  map

data("donnees_biloc")

# Oursins
map <- leaflet_oursins(data = donnees_biloc, fondMaille = regm,
  idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE", varFlux = "MIGR",
  filtreDist = 1000, filtreMajeurs = 3)

  map

# Fleches joignantes
map <- leaflet_joignantes(data = donnees_biloc, fondMaille = regm,
  typeMaille = "REG", idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", filtreDist = 1000, filtreMajeurs = 3)

  map

data("donnees_biloc_saphirs")
```

```
# Fleches saphirs
map <- leaflet_saphirs(data = donnees_biloc_saphirs, fondMaille = regm,
  typeMaille = "REG", idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", largeurFlecheMax = 80, direction = "Ent")

map
```

---

plot\_

*Creation maps in plot*


---

## Description

Displays the map in a plot output.

## Usage

```
plot_ronds(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse =
  NULL, idData, varVolume, rayonRond = NULL, rapportRond = NULL, emprise =
  "FRM", fondChx = NULL, precisionLegRonds = 0, titreLeg = "", xLeg = NULL,
  yLeg = NULL, cadreLeg = FALSE, xLimCadreLeg = NULL, yLimCadreLeg = NULL,
  titreCarte = "", sourceCarte = "", etiquettes = NULL, colPos =
  "#EB617F", colNeg = "#286AC7", colBorder = "white", colBorderMaille =
  "black", xlim = NULL, ylim = NULL)
```

```
plot_classes(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse =
  NULL, idData, varRatio, methode = "kmeans", nbClasses = 3, bornes = NULL,
  precisionLegClasses = 1, titreLeg = "", labels = NULL, xLeg = NULL,
  yLeg = NULL, cadreLeg = FALSE, xLimCadreLeg = NULL, yLimCadreLeg = NULL,
  titreCarte = "", sourceCarte = "", etiquettes = NULL, stylePalette = "defaut",
  palettePos = NULL, paletteNeg = NULL, colBorder = "white", xlim = NULL,
  ylim = NULL)
```

```
plot_ronds_classes(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse
  = NULL, idData, varVolume, varRatio, rayonRond = NULL, rapportRond = NULL,
  methode = "kmeans", nbClasses = 3, bornes = NULL, precisionLegRonds = 0,
  precisionLegClasses = 1, emprise = "FRM", fondChx = NULL, titreLegRonds =
  "", titreLegClasses = "", labels = NULL, xLegRonds = NULL, yLegRonds = NULL,
  xLegClasses = NULL, yLegClasses = NULL, cadreLeg = FALSE, xLimCadreLegRonds = NULL,
  yLimCadreLegRonds = NULL, xLimCadreLegClasses = NULL, yLimCadreLegClasses = NULL,
  titreCarte = "", sourceCarte = "", etiquettes = NULL, stylePalette = "defaut",
  palettePos = NULL, paletteNeg = NULL, colBorder = "white",
  colBorderMaille = "black", xlim = NULL, ylim = NULL)
```

```
plot_classes_ronds(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse
  = NULL, idData, varVolume, varRatio, rayonRond = NULL, rapportRond = NULL,
  methode = "kmeans", nbClasses = 3, bornes = NULL, precisionLegRonds = 0,
```

```
precisionLegClasses = 1, emprise = "FRM", fondChx = NULL, titreLegRonds = "",
titreLegClasses = "", labels = NULL, xLegRonds = NULL, yLegRonds = NULL,
xLegClasses = NULL, yLegClasses = NULL, cadreLeg = FALSE, xLimCadreLegRonds = NULL,
yLimCadreLegRonds = NULL, xLimCadreLegClasses = NULL, yLimCadreLegClasses = NULL,
titreCarte = "", sourceCarte = "", etiquettes = NULL, stylePalette = "defaut",
palettePos = NULL, paletteNeg = NULL, colBorder = "white",
colBorderRonds = "#303030", xlim = NULL, ylim = NULL)
```

```
plot_typo(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse = NULL,
idData, varTypo, titreLeg = "", xLeg = NULL, yLeg = NULL, titreCarte = "",
sourceCarte = "", etiquettes = NULL, paletteTypo = NULL, labels = NULL,
cadreLeg = FALSE, xLimCadreLeg = NULL, yLimCadreLeg = NULL, colBorder = "white",
xlim = NULL, ylim = NULL)
```

```
plot_typo_symboles(fondPoints, listFonds, emprise = "FRM", types = NULL,
couleurs = NULL, tailles = NULL, epaisseurs = NULL, titreLeg = "", xLeg =
NULL, yLeg = NULL, cadreLeg = FALSE, xLimCadreLeg = NULL, yLimCadreLeg = NULL,
titreCarte = "", sourceCarte = "", etiquettes = NULL, labels = NULL, xlim = NULL,
ylim = NULL)
```

```
plot_oursins(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse =
NULL, idDataDepart, idDataArrivee, varFlux, filtreVol = 0, filtreDist = 100,
filtreMajeurs = 10, decalageAllerRetour = 0, decalageCentroid = 0,
titreCarte = "", sourceCarte = "", etiquettes = NULL, epaisseur = 2,
colTrait = "black", colBorderMaille = "black", xlim = NULL, ylim = NULL)
```

```
plot_joignantes(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse =
NULL, typeMaille, idDataDepart, idDataArrivee, varFlux, largeurFlecheMax =
NULL, filtreVol = 0, filtreDist = 100, filtreMajeurs = 10,
decalageAllerRetour = 0, decalageCentroid = 0, emprise = "FRM",
precisionLegFleches = 0, titreLeg = "", xLeg = NULL, yLeg = NULL,
cadreLeg = FALSE, xLimCadreLeg = NULL, yLimCadreLeg = NULL, titreCarte = "",
sourceCarte = "", etiquettes = NULL, colFleche = "#286AC7", colBorder
= "white", colBorderMaille = "black", xlim = NULL, ylim = NULL)
```

```
plot_saphirs(data, fondMaille, fondSousAnalyse = NULL, fondSurAnalyse =
NULL, typeMaille, idDataDepart, idDataArrivee, varFlux, largeurFlecheMax =
NULL, direction = "Ent", filtreVol = 0, emprise = "FRM", precisionLegFleches
= 0, titreLeg = "", xLeg = NULL, yLeg = NULL, cadreLeg = FALSE,
xLimCadreLeg = NULL, yLimCadreLeg = NULL, titreCarte = "", sourceCarte = "",
etiquettes = NULL, colEntree = "#EB617F", colSortie = "#286AC7",
colBorder = "white", colBorderMaille = "black", xlim = NULL, ylim = NULL)
```

## Arguments

data	tableau de donnees (data.frame).
fondMaille	objet sf. Fond de carte.
typeMaille	chaîne de caracteres (character). Type de maille issus d'un zonage administratifs

	ou d'études ("REG", "DEP", "UU", "AU", "ZE", "BV", "EPCI" ou "COM").
fondSousAnalyse	list d'objets sf. Liste de fonds de carte a ajouter en-dessous de l'analyse. Par exemple, un fond de mer et des pays frontaliers. Par défaut a NULL.
fondPoints	objets sf. Fond de points.
fondSurAnalyse	list d'objets sf. Liste de fonds de carte a ajouter au-dessus de l'analyse. Par exemple, un fond de departement. Par défaut a NULL.
listFonds	list d'objets sf. Liste de fonds de carte a ajouter.
idData	chaîne de caracteres (character). Variable identifiant la maille.
idDataDepart	chaîne de caracteres (character). Variable identifiant le territoire de depart du flux.
idDataArrivee	chaîne de caracteres (character). Variable identifiant le territoire d'arrivee du flux.
varVolume	chaîne de caracteres (character). Variable en volume pour des ronds proportionnels.
varRatio	chaîne de caracteres (character). Variable en ratio pour des analyses en classes.
varTypo	chaîne de caracteres (character). Variable de typologie.
varFlux	chaîne de caracteres (character). Variable de flux pour des oursins, fleches joignantes ou fleches saphirs.
rayonRond	valeur numerique (numeric). Rayon du rond le plus grand en metres.
rapportRond	valeur numerique (numeric). Rapport entre l'aire du rond le plus grand et la valeur maximale des donnees en volume.
methode	chaîne de caracteres (character). A choisir parmi "kmeans" (par défaut), "fisher", "jenks" ou "quantile".
nbClasses	valeur numerique (numeric). Nombre de classes. Par défaut 3 classes.
bornes	vecteur de valeurs numeriques (numeric). Le nombre de bornes correspond au nombre de classes -1. Par défaut a NULL.
precisionLegRonds	valeur numerique (numeric). Arrondit les valeurs de la legende des ronds. 0 (par défaut) pour arrondir a l'unité, -1 pour arrondir a la dizaine, -2 a la centaine...
precisionLegClasses	valeur numerique (numeric). Arrondit les valeurs en ratio pour le calcul des bornes de classes. Par défaut, 1 chiffre apres la virgule.
precisionLegFleches	valeur numerique (numeric). Arrondit les valeurs de la legende des fleches. 0 (par défaut) pour arrondir a l'unité, -1 pour arrondir a la dizaine, -2 a la centaine...
largeurFlecheMax	valeur numerique (numeric). Valeur relative correspondant a la largeur de la fleche la plus grande de la carte. Par défaut a NULL.
direction	chaîne de caracteres (character). Pour les fleches saphirs, type de flux a choisir parmi "Ent" pour entrees (par défaut), "Sor" pour sorties et "Sol" pour soldes (entrees - sorties).



filtreVol	valeur numerique (numeric). Filtre des valeurs en volume de flux. Par default a 0 : affichage de tous les flux.
filtreDist	valeur numerique (numeric). Filtre sur la distance des fleches pour les oursins et les fleches joignantes. Par default a 100 km : affichage des flux a moins de 100 km.
filtreMajeurs	valeur numerique (numeric). Filtre sur les flux entrants et sortants les plus importants d'un territoire. Par default a 10 : 10 flux maximum sortent d'un territoire ou en entrent.
decalageAllerRetour	valeur numerique (numeric). Decalage entre les fleches aller-retour (km).
decalageCentroid	valeur numerique (numeric). Decalage des fleches depuis et vers les centroides des territoires (km).
emprise	chaîne de caracteres (character). "FRM" (par default pour la France metropolitaine), "971" (Guadeloupe), "972" (Martinique), "973" (Guyane), "974" (La Reunion) ou "976" (Mayotte). Ce parametre permet de definir le systeme de projection de la carte.
fondChx	objet sf. Fond des chefs-lieux. Pour les ronds proportionnels avec une maille communale uniquement, choix entre position du centre des ronds sur les centroides des communes (NULL par default) ou sur les chefs-lieux (chx) des communes.
types	vecteur de numeriques (numeric). Par default a NULL. Voir details.
couleurs	vecteur de caracteres (character). Par default a NULL. Voir details.
tailles	vecteur de numeriques (numeric). Par default a NULL. Voir details.
epaisseurs	vecteur de numeriques (numeric). Par default a NULL. Voir details.
titreLeg	chaîne de caracteres (character). Titre de la legende.
xLeg	valeur numerique (numeric). Coordonnees x de la legende dans le systeme de projection locale. Par default a NULL.
yLeg	valeur numerique (numeric). Coordonnees y de la legende dans le systeme de projection locale. Par default a NULL.
cadreLeg	booleen (logical). Affichage d'un cadre blanc autour de la legende. Par default FALSE.
xLimCadreLeg	vecteur numerique (numeric). Coordonnees xmin et xmax du cadre de la legende. Par default a NULL.
yLimCadreLeg	vecteur numerique (numeric). Coordonnees ymin et ymax du cadre de la legende. Par default a NULL.
xLimCadreLegRonds	vecteur numerique (numeric). Coordonnees xmin et xmax du cadre de la legende des ronds. Par default a NULL.
yLimCadreLegRonds	vecteur numerique (numeric). Coordonnees ymin et ymax du cadre de la legende des ronds. Par default a NULL.
xLimCadreLegClasses	vecteur numerique (numeric). Coordonnees xmin et xmax du cadre de la legende des classes. Par default a NULL.

yLimCadreLegClasses	vecteur numerique (numeric). Coordonnees ymin et ymax du cadre de la legende des classes. Par default a NULL.
titreLegRonds	chaîne de caracteres (character). Titre de la legende des ronds.
xLegRonds	valeur numerique (numeric). Coordonnees x de la legende des ronds dans le systeme de projection locale. Par default a NULL.
yLegRonds	valeur numerique (numeric). Coordonnees y de la legende des ronds dans le systeme de projection locale. Par default a NULL.
titreLegClasses	chaîne de caracteres (character). Titre de la legende des classes.
xLegClasses	valeur numerique (numeric). Coordonnees x de la legende des classes dans le systeme de projection locale. Par default a NULL.
yLegClasses	valeur numerique (numeric). Coordonnees y de la legende des classes dans le systeme de projection locale. Par default a NULL.
titreCarte	chaîne de caracteres (character). Titre de la carte.
sourceCarte	chaîne de caracteres (character). Source de la carte.
etiquettes	vecteur de caracteres (character) ou tableau de donnees (data.frame). Liste des codes de la maille dont on affiche le libelle sur la carte. Par default a NULL. Voir details.
colPos	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#EB617F".
colNeg	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#286AC7".
stylePalette	chaîne de caracteres (character). Palette de la charte INSEE. A choisir entre "default", "Insee_Rouge", "Insee_Jaune", "Insee_Bleu", "Insee_Turquoise", "Insee_Vert", "Insee_Violet" ou "Insee_Gris" (voir 'affiche_palette(nomPalette)').
palettePos	vecteur de caracteres (character). Couleurs nommees (par exemple c("red","orange","yellow")) ou hexadecimal (par exemple c("#FFA500")). Par default a NULL.
paletteNeg	vecteur de caracteres (character). Couleurs nommees (par exemple c("purple","blue","green")) ou hexadecimal (par exemple c("#800080","#0000FF","#008000")). Par default a NULL.
paletteTypo	vecteur de caracteres (character). Couleurs nommees (par exemple c("red","blue","green")) ou hexadecimal (par exemple c("#FFA500","#0000FF","#008000")). Par default a NULL.
labels	vecteur de caracteres (character). Labels personnalisés pour la legende de la carte en typologie. Par default a NULL.
epaisseur	valeur numerique (numeric). Epaisseur des traits des oursins. Par default 2.
colBorder	chaîne de caracteres (character). Couleur nommee (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par default "white".
colTrait	chaîne de caracteres (character). Couleur nommee (par exemple "black") ou hexadecimal (par exemple "#000000"). Par default "black".
colBorderMaille	chaîne de caracteres (character). Couleur nommee (par exemple "grey") ou hexadecimal (par exemple "#808080"). Par default "black".

colBorderRonds	chaîne de caractères (character). Couleur nommée (par exemple "grey") ou hexadécimal (par exemple "#808080"). Par défaut "#303030".
colFleche	chaîne de caractères (character). Couleur nommée (par exemple "orange") ou hexadécimal (par exemple "#FFA500"). Par défaut "#286AC7".
colEntree	chaîne de caractères (character). Couleur nommée (par exemple "orange") ou hexadécimal (par exemple "#FFA500"). Par défaut "#EB617F".
colSortie	chaîne de caractères (character). Couleur nommée (par exemple "blue") ou hexadécimal (par exemple "#0000FF"). Par défaut "#286AC7".
xlim	vecteur numérique (numeric). Coordonnées xmin et xmax de la carte. Par défaut a NULL.
ylim	vecteur numérique (numeric). Coordonnées ymin et ymax de la carte. Par défaut a NULL.

## Details

Les cartes en sortie "plot" ne sont pas interactives. Il faut éviter de zoomer sur la carte au risque de voir apparaître des décalages entre les couches.

Elles peuvent être personnalisées grâce aux paramètres de style des fonctions plot\_().

**Ronds proportionnels** L'intérieur des cercles est de couleur orange (couleur par défaut) pour représenter les valeurs positives de la variable en volume et bleu pour les valeurs négatives.

Le paramètre fondChx est à renseigner UNIQUEMENT si la maille est communale.

Pour information, le chef-lieu (ou chx) est un point de la commune autour duquel la population est la plus dense. Le plus souvent, il s'agit de l'hôtel de ville de la commune.

**Analyse en classes** En semiologie, il est déconseillé de multiplier le nombre de classes pour des raisons de visibilité et de clarté de la carte.

Si les données ne comprennent que des valeurs positives ou que des valeurs négatives, il est conseillé 5 classes au maximum.

Si les données comprennent à la fois des valeurs positives et négatives, le nombre maximum conseillé est 9 classes.

Il est possible de spécifier manuellement les bornes des classes en implémentant le paramètre bornes d'un vecteur de valeurs numériques de type c(bornes1, bornes2, ...). Le nombre de bornes correspond au nombre de classes -1 (length(bornes) == nbClasses - 1). Il n'est pas obligatoire de spécifier les valeurs min et max.

**Analyse en classes dans ronds proportionnels** L'analyse en classes colore l'intérieur des ronds : (fonction plot\_ronds\_classes())

**Ronds proportionnels sur analyse en classes** Les ronds proportionnels sont posés sur l'analyse en classes : (fonction plot\_classes\_ronds())

**Typologie** Meme conseil que pour les analyses en classes. Le choix des couleurs ne doit pas être un dégrade si il n'existe pas de classement dans les modalités.

Il est possible de spécifier ses propres labels sans avoir besoin de modifier les noms de variables dans la table de données.

**Typologie en symboles** Les fonds de carte doivent être passés en list et peuvent être personnalisés (voir ci-dessous modifier l'apparence des fonds).

Les parametres types, couleurs, tailles et epaisseurs sont des vecteurs de meme longueur. Chaque element correspond a un point du fond fourni. Verifier bien l'ordre des points pour modifier leur apparence ensuite.

Il existe differents types de points (voir l'aide de ?graphics::points).

**Oursins** Les oursins representent uniquement l'existence d'un flux entre 2 territoires. Le sens du flux et le volume ne sont pas pris en compte.

Les filtres sur le volume, la distance et les flux majeurs sont independants. Chacun d'eux s'appliquent sur les donnees initiales. Autrement dit, le filtre sur les flux majeurs ne se fera pas apres un eventuel filtre sur la distance mais sur toutes les donnees en entree.

Le filtre sur le volume est plutot complementaire au filtre sur les flux majeurs. En effet, il est possible de vouloir represente des faibles flux significatifs. Le filtre sur les flux majeurs le permet contrairement au filtre sur le volume seul.

**Fleches joignantes** Les fleches joignantes representent un flux entre 2 territoires, son sens et son volume. Dans le cas de flux aller et retour, deux fleches de sens oppose sont superposees.

La regle des filtres est la meme que pour les oursins.

**Fleches saphirs** Les fleches saphirs convergent vers ou divergent d'un meme territoire. Elles permettent de represente les entrees, les sorties ou le solde (entrees - sorties) d'un territoire par rapport aux autres.

Pour un solde positif, les fleches sont representees en orange, pour un solde negatif en bleu.

Une attention particuliere est a apporter au choix de la variable idDataDepart et idDataArrivee selon si le flux est entrant, sortant ou en solde.

La legende est positionnee par default sur la carte si les coordonnees x et y ne sont pas specifiees. Elle est positionnee en haut a droite de la carte sauf pour l'analyse en classes ou elle est positionnee en bas a droite.

Les coordonnees x et y doivent etre dans le systeme de projection locale :

- France metropolitaine : Lambert 93 (code epsg 2154)
- Guadeloupe : UTM 20 N (code epsg 32620)
- Martinique : UTM 20 N (code epsg 32620)
- Guyane : UTM 22 N (code epsg 2972)
- La Reunion : UTM 40 S (code epsg 2975)
- Mayotte : UTM 38 S (code epsg 4471)

Il est possible d'ajouter des fonds de carte pour l'habiller. Il y existe 2 parametres : fondSousAnalyse et fondSurAnalyse.

Le parametre fondSousAnalyse permet d'ajouter des fonds en-dessous de l'analyse et la parametre fondSurAnalyse au-dessus.

Chacun de ces parametres est une liste d'objets sf. Par default, la couleur de remplissage est transparente, la bordure est noire et l'epaisseur est de 1.

Pour modifier l'apparence de ces fonds, il faut ajouter des colonnes dans les objets sf correspondants :

- COL : une colonne COL pour modifier la couleur de remplissage.
- BORDER : une colonne BORDER pour modifier la couleur de la bordure.

- EPAISSEUR : une colonne EPAISSEUR pour modifier l'épaisseur de la bordure.

Pour ajouter des colonnes un objet sf, il faut utiliser la fonction `cbind`. Exemple : `paysf <- cbind(paysf, COL="grey", BORDER="#404040", EPAISSEUR=2)`

**etiquettes** Des etiquettes peuvent etre affichees sur les cartes en plot grace au parametre "etiquettes". Comment proceder ?

- Specifier un vecteur de codes des territoires a etiqueter appartenant a la maille. La fonction recupere les libelles a afficher. Les etiquettes se placent au centroide des territoires, reperes par leurs coordonnees x et y. Une mise en forme des etiquettes par default est proposee.

- Pour formater les etiquettes, il est utile de passer par la fonction `coordonnees_etiquettes`. A partir d'un fond de maille et d'un vecteur de codes, un data.frame est produit en sortie de la fonction.

Ce tableau contient le code du territoire, le libelle a afficher, les coordonnees x et y du centroide du territoire (position des etiquettes X et Y), la taille (TAILLE), le style (FONT) et la couleur de la police (COL). Le style de police est un entier : 1 (normal), 2 (gras, par default), 3 (italique) et 4 (gras italique).

Il est possible de modifier les valeurs du tableau pour changer le style d'une ou plusieurs etiquettes.

Il est egalement possible de modifier les coordonnees x et y pour deplacer les etiquettes et ainsi eviter leur chevauchement.

- Pour afficher la carte avec les etiquettes formatees, il faut passer ce tableau dans le parametre "etiquettes" de la fonction `plot`.

Il est tout de meme possible de zoomer sur la carte en specifiant les parametres `xlim` et `ylim`. Par default, la carte est centree sur le fond de maille. Pour connaitre les limites par default, executer la fonction `sf::st_bbox(fondMaille)`

## Value

Retourne un objet sf. Il s'agit du fond des ronds (ronds proportionnels, classes dans ronds proportionnels, ronds sur analyse en classes), de la maille (analyse en classes, typologie) ou des fleches (oursins, fleches joignantes, fleches saphirs).

Affiche dans le plot la carte demandee.

## See Also

[zonage\\_a\\_facon](#),

[coordonnees\\_etiquettes](#),

[rayon\\_ronds](#), [rapport\\_ronds](#), [largeur\\_fleche](#),

[calcul\\_ratio](#), [calcul\\_tx\\_evol\\_global](#), [calcul\\_tx\\_evol\\_ann\\_moy](#), [calcul\\_part\\_ens](#), [calcullette](#),

[distrib\\_variable](#),

[recup\\_palette](#)

## Examples

```

data("donnees_monoloc")
data("depm")

# Ronds proportionnels sur une analyse en classes
fond_ronds <- plot_classes_ronds(data = donnees_monoloc, fondMaille = depm,
idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4,
titreLegRonds = "POP_2015", titreLegClasses = "VAR_AN_MOY",
xLegClasses = 1150000, yLegClasses = 6600000)

etiquettes <- coordonnees_etiquettes(fondMaille = depm,
listeCode = c("06","13","31","33","44","67","69","59","75"))
etiquettes$LIBELLE <- c("Nice","Marseille","Toulouse","Bordeaux","Nantes","Lille",
"Strasbourg","Lyon","Paris")
etiquettes[etiquettes$CODE=="75","TAILLE"] <- 1.5

# Ronds proportionnels sur une analyse en classes
fond_ronds <- plot_classes_ronds(data = donnees_monoloc, fondMaille = depm,
idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4,
titreLegRonds = "POP_2015", titreLegClasses = "VAR_AN_MOY",
xLegClasses = 1150000, yLegClasses = 6600000,
etiquettes = etiquettes)

```

---

rapport\_ronds

*Ratio between radius and value for 'leaflet' maps*

---

## Description

Returns the ratio between the area of the largest circle and the maximum value of the volume data for 'leaflet' maps.

## Usage

```
rapport_ronds(map)
```

## Arguments

map                    objet leaflet.

## Details

Le rapport n'a pas d'unité. Il s'agit d'une valeur relative.

Le calcul du rapport sert à comparer plusieurs cartes en ronds proportionnels entre elles puisque la valeur du rapport est fonction du volume à représenter.

La formule utilisée pour calculer le rapport est :  $(\pi * (\text{rayonRond})^2) / \text{max\_var}$

Il peut y avoir une legere difference entre la valeur du rapport passee en parametre de la fonction `leaflet_ronds` et la valeur renvoyee par la fonction `rapport_ronds`. Cette difference s'explique par l'arrondi de pi a 6 chiffres apres la virgule.

### Value

Retourne un numerique.

### See Also

[leaflet\\_ronds](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#)

### Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
  varVolume = "POP_2015")

map

rapport_ronds(map)
# [1] 1924095

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
  varVolume = "POP_2015", rapportRond = 1900000)

map

rapport_ronds(map)
# [1] 1900022
```

---

rayon\_ronds

*Radius of the largest circle for 'leaflet' maps*

---

### Description

Returns the radius of the largest circle for 'leaflet' maps.

### Usage

```
rayon_ronds(map)
```

## Arguments

map                    objet leaflet.

## Details

L'unité du rayon est le metre.

A noter, que la taille du rayon est limitée afin d'éviter de trop masquer le territoire d'étude en arrière-plan.

En effet, la règle semiologique des 1/7ème s'applique à toutes les cartes affichant des ronds proportionnels. Cette règle spécifie que la somme des aires des ronds ne doit pas dépasser 1/7ème de l'aire du territoire d'étude.

## Value

Retourne un numérique.

## See Also

[leaflet\\_ronds](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#)

## Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
  varVolume = "POP_2015")

map

rayon_ronds(map)
# [1] 39944.67

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
  varVolume = "POP_2015", rayonRond = 30000)

map

rayon_ronds(map)
# [1] 30000
```



---

recup_palette	<i>Returns a palette of the graphic chart of INSEE</i>
---------------	--

---

### Description

Returns a palette of the graphic chart of INSEE.

### Usage

```
recup_palette(stylePalette, nbNeg = 0, nbPos = 0)
```

### Arguments

stylePalette	chaîne de caractères (character). À choisir parmi "default", "Insee_Rouge", "Insee_Jaune", "Insee_Bleu", "Insee_Turquoise", "Insee_Vert", "Insee_Violet", "Insee_Gris", "Turquoise_Neg", "Vert_Neg", "Violet_Neg" ou "Gris_Neg".
nbNeg	numeric. Nombre de classes négatives. De 0 (par défaut) à 6.
nbPos	numeric. Nombre de classes positives. De 0 (par défaut) à 6.

### Details

Les couleurs de palettes préfixées par "Insee\_" sont celles de la charte graphique INSEE. En exécutant la fonction `affiche_palette(nomPalette)`, il est possible de visualiser les couleurs de chaque palette disponible. Les modalités pour l'argument 'nomPalette' sont "Insee\_Rouge", "Insee\_Jaune", "Insee\_Bleu", "Insee\_Turquoise", "Insee\_Vert", "Insee\_Violet" et "Insee\_Gris" pour les palettes de la charte graphique INSEE, "default", "Turquoise\_Neg", "Vert\_Neg", "Violet\_Neg" et "Gris\_Neg" pour les palettes hors charte.

La palette "default" reprend les couleurs de la palette RdYlBu de ColorBrewer.

Seules les palettes "Insee\_Rouge" et "Insee\_Jaune" proposent des palettes pour des valeurs négatives et positives. Pour ces deux palettes, la palette "Insee\_Bleu" est utilisée pour représenter les valeurs négatives. Cette dernière peut également être utilisée seule pour des valeurs uniquement positives ou négatives.

La fonction renvoie une liste d'un vecteur composé du nom de la palette au format `nomPalette_xN_yP` et des valeurs hexadécimales de la palette.

La liste des valeurs hexadécimales de toutes les palettes est disponible en chargeant les données suivantes : `data("palettes_insee")`.

Il est obligatoire de spécifier un nombre de classes négatives ou positives supérieur à 0 et jusqu'à 6 maximum. Un message d'erreur apparaîtra si les choix du nombre de couleurs négatives et positives ne sont pas adaptés à la palette spécifiée.

### Value

Retourne une liste d'un vecteur caractères.

**References**

Un convertisseur de couleurs pour visualiser une couleur a partir de son nom, son code hexadecimal ou RGB : <http://www.proftnj.com/RGB3.htm>

ColorBrewer 2.0 : <https://colorbrewer2.org/#type=diverging&scheme=RdYlBu&n=10>

**See Also**

[set\\_couleur\\_classes](#)

**Examples**

```
recup_palette(stylePalette = "defaut", nbNeg = 3, nbPos = 3)

# $defaut_3N3P
# [1] "#003269" "#005289" "#95BAE2" "#E4A75A" "#D47130" "#B24B1D"
```

---

set\_bordure\_ronds      *Modify the circles' borders of 'leaflet' map's analysis*

---

**Description**

Modify the circles' borders of 'leaflet' map's analysis.

**Usage**

```
set_bordure_ronds(map, colBorderPos = "white", colBorderNeg =
"white", epaisseurBorder = 1, map_leaflet = NULL)
```

**Arguments**

map	objet leaflet.
colBorderPos	chaîne de caractères (character). Couleur nommée (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par défaut "white".
colBorderNeg	chaîne de caractères (character). Couleur nommée (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par défaut "white".
epaisseurBorder	chaîne de caractères (numeric). Épaisseur de la bordure des ronds. Par défaut a 1".
map_leaflet	objet leaflet. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut a NULL.

**Details**

Permet de modifier l'apparence des bordures des ronds, c'est-à-dire leur couleur et leur épaisseur.

Il est possible de distinguer les valeurs en volume positives des valeurs en volume négatives.

Applicable pour les analyses en ronds proportionnels, en classes dans les ronds et en ronds sur les classes.

**Value**

Retourne un objet de type leaflet.

**References**

Un convertisseur de couleurs pour visualiser une couleur à partir de son nom, son code hexadécimal ou RGB : <http://www.proftnj.com/RGB3.htm>

**See Also**

[recup\\_palette](#),  
[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#),  
[leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

**Examples**

```
data("depm")

data("donnees_monoloc")

# Ronds proportionnels sur une analyse en classes
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015")
map <- set_bordure_ronds(map = map, colBorderPos = "grey", epaisseurBorder = 3)

map
```

---

set\_couleur\_

*Modify the colors of 'leaflet' map's analysis*

---

**Description**

Modify the colors of 'leaflet' map's analysis.

**Usage**

```
set_couleur_ronds(map, colorPos = "#EB617F", colorNeg = "#286AC7",
  map_leaflet = NULL)
```

```
set_couleur_classes(map, stylePalette = "defaut", palettePos = NULL,
  paletteNeg = NULL, colBorder = "white", map_leaflet = NULL)
```

```
set_couleur_typo(map, paletteTypo = NULL, colBorder = "white", map_leaflet =
  NULL)
```

```
set_couleur_joignantes(map, colFleche = "#286AC7", colBorder = "black",
  map_leaflet = NULL)
```

```
set_couleur_saphirs(map, colEntree = "#EB617F", colSortie = "#286AC7",
  colBorder = "black", map_leaflet = NULL)
```

**Arguments**

map	objet leaflet.
colorPos	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#EB617F".
colorNeg	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#286AC7".
stylePalette	chaîne de caractères (character). À choisir parmi "defaut", "Insee_Rouge", "Insee_Jaune", "Insee_Bleu", "Insee_Turquoise", "Insee_Vert", "Insee_Violet" ou "Insee_Gris" (voir 'affiche_palette(nomPalette)').
palettePos	vecteur de caractères (character). Spécifier le vecteur de couleurs (nommées ou hexadécimales) des valeurs positives (au choix). Par défaut à NULL.
paletteNeg	vecteur de caractères (character). Spécifier le vecteur de couleurs (nommées ou hexadécimales) des valeurs négatives (au choix). Par défaut à NULL.
paletteTypo	vecteur de caractères (character). Spécifier le vecteur de couleurs (nommées ou hexadécimales) de la typologie (au choix). Par défaut à NULL.
colBorder	chaîne de caractères (character). Couleur nommée (par exemple "white") ou hexadécimale (par exemple "#FFFFFF"). Par défaut "white" pour classes et typo, "black" pour joignantes et saphirs.
colFleche	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#286AC7".
colEntree	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#EB617F".
colSortie	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#286AC7".
map_leaflet	objet leaflet. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut à NULL.

**Details**

Les palettes proposées pour l'analyse en classes sont celles utilisées par la Charte Graphique IN-SEE. Utilisez la fonction `affiche_palette(nomPalette)` pour visualiser les couleurs d'une palette. L'argument `'nomPalette'` peut prendre comme valeur : "defaut", "Insee\_Rouge", "Insee\_Jaune", "Insee\_Bleu", "Insee\_Turquoise", "Insee\_Vert", "Insee\_Violet" ou "Insee\_Gris".

Le code des couleurs est le code hexadécimal. Pour récupérer les codes hexadécimaux des différentes palettes proposées : `data("palettes_insee")`

**Value**

Retourne un objet de type `leaflet`.

**References**

Un convertisseur de couleurs pour visualiser une couleur à partir de son nom, son code hexadécimal ou RGB : <http://www.proftnj.com/RGB3.htm>

**See Also**

[recup\\_palette](#),  
[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#),  
[leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

**Examples**

```
data("depm")

data("donnees_monoloc")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015")
map <- set_couleur_ronds(map = map, colorPos = "orange")

map

# Ronds proportionnels sur une analyse en classes
map <- leaflet_classes_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4)
map <- set_couleur_classes(map = map, stylePalette = "Insee_Rouge", colBorder = "black")

map

data("donnees_biloc_saphirs")
data("regm")

# Fleches saphirs
```

```
map <- leaflet_saphirs(data = donnees_biloc_saphirs, fondMaille = regm, typeMaille = "REG",
  idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
  varFlux = "MIGR", largeurFlecheMax = 500, direction = "Ent")
map <- set_couleur_saphirs(map = map, colEntree = "#4BB375", colBorder = "grey")

map
```

---

set\_fonds\_simples      *Modify the style of 'leaflet' map's.*

---

## Description

Modify the style of 'leaflet' map's.

## Usage

```
set_fonds_simples(map, colRemplissageFonds = NULL, colBordureFonds =
  NULL, transparenceFonds = NULL, epaisseurFonds = NULL, map_leaflet = NULL)
```

## Arguments

map	objet leaflet.
colRemplissageFonds	vecteur de caracteres (character). Le vecteur comporte des couleurs nommées (par exemple "black") ou hexadecimal (par exemple "#000000"). Colore l'intérieur du fond. Par défaut a NULL.
colBordureFonds	vecteur de caracteres (character). Le vecteur comporte des couleurs nommées (par exemple "black") ou hexadecimal (par exemple "#000000"). Colore la bordure du fond. Par défaut a NULL.
transparenceFonds	valeur numerique (numeric). Le vecteur comporte les valeurs numeriques pour la transparence (valeur comprise entre 0 - transparent - et 1 - opaque). Par défaut a NULL.
epaisseurFonds	valeur numerique (numeric). Le vecteur comporte les valeurs numeriques pour l'épaisseur de la bordure des fonds. Par défaut a NULL.
map_leaflet	objet leaflet. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut a NULL.

## Value

Retourne un objet leaflet.

**See Also**

[leaflet\\_fonds\\_simples](#)  
[leaflet\\_ronds](#) [leaflet\\_classes](#) [leaflet\\_ronds\\_classes](#) [leaflet\\_classes\\_ronds](#)  
[leaflet\\_typo](#) [leaflet\\_oursins](#) [leaflet\\_joignantes](#) [leaflet\\_saphirs](#)

**Examples**

```
data("regm")
data("fram")

# Fonds simples
map <- leaflet_fonds_simples(listFonds = list(fram, regm[regm$CODE == '93',]))
map <- set_fonds_simples(map, colRemplissageFonds = c("white","grey"),
  colBordureFonds = c("grey","black"), transparenceFonds = c(1,0.6), epaisseurFonds = c(1,3))

map
```

---

set\_opacite\_elargi      *Modify the opacity of the expanded representation of a 'leaflet' map*

---

**Description**

Modify the opacity of the expanded representation of a 'leaflet' map for proportional circles and choropleth.

**Usage**

```
set_opacite_elargi(map, opacite = 0.6, map_leaflet = NULL)
```

**Arguments**

map	objet leaflet.
opacite	valeur numerique (numeric). Chiffre entre 0 (transparent) et 1 (opaque). Par default a 0.6 (60 pour cent d'opacite ou 40 pour cent de transparence).
map_leaflet	objet leaflet. Pour l'integration des fonctions leaflet dans les applications shiny (cf vignette). Par default a NULL.

**Details**

Seule la representation elargie est concerne par la modification de l'opacite. La representation principale reste 100 pour cent opaque.

**Value**

Retourne un objet leaflet.

### See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#), [leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#)

### Examples

```
data("donnees_monoloc")
data("depm")

# Ronds proportionnels
map <- leaflet_ronds(data = donnees_monoloc, fondMaille = depm[depm$REG=="93",],
fondMailleElargi = depm, fondSuppl = depm, idData = "COD_DEP",
varVolume = "POP_2015")
map <- set_opacite_elargi(map = map, opacite = 0.8)

map
```

---

set\_pop\_up

*Modify the pop\_up of 'leaflet' map's analysis*

---

### Description

Modify the pop\_up of 'leaflet' map's analysis.

### Usage

```
set_pop_up(map, popup = NULL, popupRonds = NULL)
```

### Arguments

map	objet leaflet.
popup	vecteur de caracteres (character). Peut etre du HTML pour mettre en forme le texte. Par default a NULL.
popupRonds	vecteur de caracteres (character). Peut etre du HTML pour mettre en forme le texte. Par default a NULL.

### Details

Le nouveau texte pour le pop-up peut etre du contenu HTML. Par exemple : "**<font color=#2B3E50>Greater City of Paris</font> </b><br><b><font color=#2B3E50>PART : </font></b><b> <font color=#2B3E50>21,2</font></b></b>**"  
Attention : conserver le meme ordre des pop-up dans le vecteur que celui des donnees pour que les valeurs correspondent bien a l'analyse.

Ne fonctionne que pour les ronds proportionnels, les analyses en classes et les typologies.

Pour supprimer les pop-up, laisser le parametre a NULL.



**Value**

Retourne un objet de type leaflet.

**See Also**

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#)

**Examples**

```
data("depm")

data("donnees_monoloc")

# Ronds proportionnels sur une analyse en classes
map <- leaflet_ronds_classes(data = donnees_monoloc[donnees_monoloc$REG=="93",],
  fondMaille = depm[depm$REG=="93",], idData = "COD_DEP", varVolume = "POP_2015",
  varRatio = "VAR_AN_MOY", nbClasses = 4)

new_popup <- c(paste0("<b><font color=#2B3E50>Bouches-du-Rhone</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50>2 016,6</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,4</font>"),
  paste0("<b><font color=#2B3E50>Alpes-Maritimes</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50>1 082,4</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,1</font>"),
  paste0("<b><font color=#2B3E50>Var</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50>1 048,7</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,8</font>"),
  paste0("<b><font color=#2B3E50>Vaucluse</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50> 557,5</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,6</font>"),
  paste0("<b><font color=#2B3E50>Alpes-de-Haute-Provence</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50> 161,8</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,2</font>"),
  paste0("<b><font color=#2B3E50>Hautes-Alpes</font></b><br>",
  "<font color=#2B3E50>Population 2015 : </font><font color=#2B3E50> 140,9</font><br>",
  "<font color=#2B3E50>Variation annuelle moyenne : </font><font color=#2B3E50>0,6</font>"))

map <- set_pop_up(map = map, popupRonds = new_popup)

map
```

---

set\_style\_oursins

*Modify the style of 'leaflet' map's urchins*

---

**Description**

Modify the style of 'leaflet' map's urchins.

## Usage

```
set_style_oursins(map, epaisseur = 2, colTrait = "black", map_leaflet  
= NULL)
```

## Arguments

map	objet leaflet.
epaisseur	valeur numerique (numeric). Par default a 2.
colTrait	chaîne de caracteres (character). Couleur nommée (par exemple "black") ou hexadecimal (par exemple "#000000"). Par default "black".
map_leaflet	objet leaflet. Pour l'integration des fonctions leaflet dans les applications shiny (cf vignette). Par default a NULL.

## Value

Retourne un objet leaflet.

## See Also

[leaflet\\_fonds\\_simples](#)  
[leaflet\\_ronds](#) [leaflet\\_classes](#) [leaflet\\_ronds\\_classes](#) [leaflet\\_classes\\_ronds](#)  
[leaflet\\_typo](#) [leaflet\\_oursins](#) [leaflet\\_joignantes](#) [leaflet\\_saphirs](#)

## Examples

```
data("donnees_biloc")  
data("regm")  
  
# Oursins  
map <- leaflet_oursins(data = donnees_biloc, fondMaille = regm, idDataDepart = "REG_DEPART",  
idDataArrivee = "REG_ARRIVEE", varFlux = "MIGR", filtreDist = 1000,  
filtreMajeurs = 3)  
map <- set_style_oursins(map = map, epaisseur = 3, colTrait = "grey")  
  
map
```

---

shiny\_

*Creation 'leaflet' maps in a 'shiny' web environment*

---

## Description

Creation 'leaflet' maps in a 'shiny' web environment where the parameters are modifiable on the fly.

**Usage**

```
shiny_ronds(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varVolume, emprise = "FRM", fondEtranger = NULL,
fondChx = NULL)
```

```
shiny_classes(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varRatio, emprise = "FRM", fondEtranger = NULL)
```

```
shiny_ronds_classes(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varVolume, varRatio, emprise = "FRM", fondEtranger
= NULL, fondChx = NULL)
```

```
shiny_classes_ronds(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varVolume, varRatio, emprise = "FRM", fondEtranger
= NULL, fondChx = NULL)
```

```
shiny_typo(data, fondMaille, fondContour, fondSuppl = NULL, idData, varTypo,
emprise = "FRM", fondEtranger = NULL)
```

```
shiny_oursins(data, fondMaille, fondContour, fondSuppl = NULL, idDataDepart,
idDataArrivee, varFlux, decalageAllerRetour = 0, decalageCentroid = 0,
emprise = "FRM", fondEtranger = NULL)
```

```
shiny_joignantes(data, fondMaille, typeMaille, fondContour, fondSuppl =
NULL, idDataDepart, idDataArrivee, varFlux, decalageAllerRetour = 0,
decalageCentroid = 0, emprise = "FRM", fondEtranger = NULL)
```

```
shiny_saphirs(data, fondMaille, typeMaille, fondContour, fondSuppl = NULL,
idDataDepart, idDataArrivee, varFlux, direction = "Ent", emprise = "FRM",
fondEtranger = NULL)
```

**Arguments**

<code>data</code>	tableau de donnees (data.frame).
<code>fondMaille</code>	objet sf. Fond de carte.
<code>typeMaille</code>	chaîne de caracteres (character). Type de maille issu d'un zonage administratif ou d'études ("REG", "DEP", "UU", "AU", "ZE", "BV", "EPCI" ou "COM").
<code>fondMailleElargi</code>	objet sf. Fond de carte. Par défaut a NULL.
<code>fondContour</code>	objet sf. Fond de carte.
<code>fondSuppl</code>	objet sf. Fond de carte. Par défaut a NULL.
<code>idData</code>	chaîne de caracteres (character). Variable identifiant la maille.
<code>idDataDepart</code>	chaîne de caracteres (character). Variable identifiant le depart du flux.
<code>idDataArrivee</code>	chaîne de caracteres (character). Variable identifiant l'arrivée du flux.
<code>varVolume</code>	chaîne de caracteres (character). Variable en volume pour les ronds proportionnels.

varRatio	chaîne de caractères (character). Variable en ratio pour l'analyse en classes.
varTypo	chaîne de caractères (character). Variable de typologie.
varFlux	chaîne de caractères (character). Variable de flux pour les oursins, fleches joignantes ou fleches saphirs.
direction	chaîne de caractères (character). Type de flux. A choisir parmi "Ent" pour entrees (par default), "Sor" pour sorties et "Sol" pour soldes (entrees - sorties).
decalageAllerRetour	valeur numerique (numeric). Decalage entre les fleches aller-retour (km).
decalageCentroid	valeur numerique (numeric). Decalage des fleches depuis et vers les centroides des territoires (km).
emprise	chaîne de caractères (character). "FRM" (par default pour la France metropolitaine), "971" (Guadeloupe), "972" (Martinique), "973" (Guyane), "974" (La Reunion), "976" (Mayotte) ou "999" (Etranger). Ce parametre permet de definir le systeme de projection de la carte.
fondEtranger	objet sf. Fond de carte d'habillage personnalisable pour les fonds autres que la France (metropolitaine et DOM). Par default a NULL.
fondChx	objet sf. Fond des chefs-lieux. Pour les ronds proportionnels avec une maille communale uniquement, choix entre position du centre des ronds sur les centroides des communes (NULL par default) ou sur les chefs-lieux (chx) des communes.

## Details

Les cartes sont externalisees de l'environnement R contrairement aux cartes plot et leaflet. Elles s'affichent dans le navigateur.

Les cartes produites possedent les memes avantages que les cartes leaflet (deplacement de la carte, zoom/dezoom, pop-up d'informations, placement libre de la legende).

L'environnement R-Shiny propose en plus la possibilite de modifier a la volee les parametres de la carte (ordre des fonds, ajout de fonds, representation elargie, taille et rapport des ronds, methode des classes, nombre de classes, bornes, filtre de flux, legende).

Les cartes peuvent etre exportees en projet Qgis via un bouton de l'interface R-Shiny. La fonctionnalite cree un .zip a dezipper au meme emplacement.

L'apparence des cartes (choix de la palette et des couleurs, etiquettes) n'est cependant pas modifiable. Le but des fonctions shiny est de proposer des cartes realisables et modifiables rapidement avec un minimum de parametrage. En effet, une mise en forme specifique pourra s'effectuer dans Qgis pour la diffusion par exemple.

Il est obligatoire de renseigner le contour du territoire d'études dans le paramètre fondContour.

Il est possible de remplacer le fond d'habillage par default (France metropolitaine ou DOM) par un autre fond. Pour cela, il faut specifier le parametre emprise="999" ET ajouter un fond sf dans le parametre fondEtranger, par exemple, une carte de l'Europe ou d'un pays particulier. Le systeme de coordonnees du fond doit etre projete (en unite de longueur) et non geographique (lng/lat). Si aucun code EPSG valide n'est trouve pour ce fond, le code EPSG 3395 est choisi (projection Mercator). Attention car la legende des ronds sera d'autant plus deformee qu'elle se situera loin de l'equateur.

**Ronds proportionnels** L'intérieur des cercles est de couleur orange (couleur par défaut) pour représenter les valeurs positives de la variable en volume et bleu pour les valeurs négatives.

Le paramètre `fondChx` est à renseigner UNIQUEMENT si la maille est communale.

Pour information, le chef-lieu (ou `chx`) est un point de la commune autour duquel la population est la plus dense. Le plus souvent, il s'agit de l'hôtel de ville de la commune.

**Analyse en classes** En semiologie, il est déconseillé de multiplier le nombre de classes pour des raisons de visibilité et de clarté de la carte.

Si les données ne comprennent que des valeurs positives ou que des valeurs négatives, il est conseillé 5 classes au maximum.

Si les données comprennent à la fois des valeurs positives et négatives, le nombre maximum conseillé est 9 classes.

**Analyse en classes dans ronds proportionnels** L'analyse en classes colore l'intérieur des ronds (fonction `shiny_ronds_classes()`)

**Ronds proportionnels sur analyse en classes** Les ronds proportionnels sont posés sur l'analyse en classes (fonction `shiny_classes_ronds()`)

**Typologie** Même conseil que pour les analyses en classes. Le choix des couleurs ne doit pas être un dégradé si il n'existe pas de classement dans les modalités.

**Oursins** Les oursins représentent uniquement l'existence d'un flux entre 2 territoires. Le sens du flux et le volume ne sont pas pris en compte.

Les filtres sur le volume, la distance et les flux majeurs sont indépendants. Chacun d'eux s'applique sur les données initiales. Autrement dit, le filtre sur les flux majeurs ne se fera pas après un éventuel filtre sur la distance mais sur toutes les données en entrée.

Le filtre sur le volume est plutôt complémentaire au filtre sur les flux majeurs. En effet, il est possible de vouloir représenter des faibles flux significatifs. Le filtre sur les flux majeurs le permet contrairement au filtre sur le volume seul.

**Fleches joignantes** Les fleches joignantes représentent un flux entre 2 territoires, son sens et son volume. Dans le cas de flux aller et retour, deux fleches de sens opposés sont superposées.

La règle des filtres est la même que pour les oursins.

**Fleches saphirs** Les fleches saphirs convergent vers ou divergent d'un même territoire. Elles permettent de représenter les entrées, les sorties ou le solde (entrées - sorties) d'un territoire par rapport aux autres.

Pour un solde positif, les fleches sont représentées en orange, pour un solde négatif en bleu.

Une attention particulière est à apporter au choix de la variable `idDataDepart` et `idDataArrivee` selon si le flux est entrant, sortant ou en solde.

## Value

Ne retourne aucun objet.

Ouvre une fenêtre avec un environnement R-Shiny dans le navigateur.

## See Also

[leaflet\\_ronds](#), [leaflet\\_classes](#), [leaflet\\_ronds\\_classes](#), [leaflet\\_classes\\_ronds](#), [leaflet\\_typo](#), [leaflet\\_oursins](#), [leaflet\\_joignantes](#), [leaflet\\_saphirs](#), [export\\_qgis\\_ronds](#), [export\\_qgis\\_classes](#), [export\\_qgis\\_ronds\\_classes](#), [export\\_qgis\\_classes\\_ronds](#),

```
export_qgis_typo,
export_qgis_oursins, export_qgis_joignantes, export_qgis_saphirs
```

## Examples

```
data("depm")
data("regm")
data("fram")

data("donnees_monoloc")

if(interactive()){
# Ronds proportionnels
shiny_ronds(data = donnees_monoloc, fondMaille = depm, fondContour = fram,
idData = "COD_DEP", varVolume = "POP_2015")

# Analyse en classes
shiny_classes(data = donnees_monoloc, fondMaille = depm, fondContour = fram,
idData = "COD_DEP", varRatio = "VAR_AN_MOY")

# Analyse en classes dans les ronds proportionnels
shiny_ronds_classes(data = donnees_monoloc, fondMaille = depm, fondContour = fram,
idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY")

# Ronds proportionnels sur une analyse en classes
shiny_classes_ronds(data = donnees_monoloc, fondMaille = depm, fondContour = fram,
idData = "COD_DEP", varVolume = "POP_2015", varRatio = "VAR_AN_MOY")

# Typologie
shiny_typo(data = donnees_monoloc, fondMaille = depm, fondContour = fram,
idData = "COD_DEP", varTypo = "REG")
}

data("donnees_biloc")

if(interactive()){
# Oursins
shiny_oursins(data = donnees_biloc, fondMaille = regm, fondContour = fram,
idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE", varFlux = "MIGR")

# Fleches joignantes
shiny_joignantes(data = donnees_biloc, fondMaille = regm, typeMaille = "REG",
fondContour = fram, idDataDepart = "REG_DEPART",
idDataArrivee = "REG_ARRIVEE", varFlux = "MIGR")
}

data("donnees_biloc_saphirs")

if(interactive()){
# Fleches saphirs
```

```
shiny_saphirs(data = donnees_biloc_saphirs, fondMaille = regm, typeMaille = "REG",
fondContour = fram, idDataDepart = "REG_DEPART", idDataArrivee = "REG_ARRIVEE",
varFlux = "MIGR", direction = "Ent")
}
```

---

zonage\_a\_facon

*Creating a custom zoning*


---

## Description

Creating a custom zoning from a group of entities.

## Usage

```
zonage_a_facon(fondMaille, groupe, idMaille, idGroupe, libGroupe,
fondContour = NULL)
```

## Arguments

fondMaille	objet sf. Fond de carte.
groupe	tableau de donnees (data.frame) contenant un identifiant de maille, un identifiant de groupes et un libelle de groupes.
idMaille	chaîne de caractere (character). Variable identifiant la maille.
idGroupe	chaîne de caractere (character). Variable identifiant les groupes. Le choix de l'identifiant de groupes est libre ("A", "B" et "C" par exemple).
libGroupe	chaîne de caractere (character). Variable des libelles de groupes.
fondContour	objet sf. Fond de carte. Par default a NULL.

## Details

La fonction `zonage_a_facon` permet de creer son propre zonage.

A partir d'un fond de maille, la fonction va regrouper certaines entites pour en former des plus grandes. Par exemple, partir d'une maille communale pour former une maille de zones d'emploi (voir exemples).

Il faut s'assurer que l'identifiant de maille de la table corresponde bien a l'identifiant du fond de maille.

Le zonage ainsi genere peut correspondre a un zonage connu (administratif ou d'etudes) ou inconnu (personnalise).

La fonction peut prendre en parametre un contour de territoire. Il s'agit du contour qui delimite l'ensemble de la maille. Il peut etre utile de le specifier si vous voulez tronquer la maille sur ce contour. Par exemple, pour représenter uniquement la partie regionale des zones d'emploi.

## Value

Retourne un objet de type sf (fond de carte)

**Examples**

```
data("donnees_a_facon")
data("com_dep_13_30_83_84")
data("depm")

ze13etplus <- zonage_a_facon(fondMaille = com_dep_13_30_83_84, groupe = donnees_a_facon,
idMaille = "DEPCOM", idGroupe = "ZE2010", libGroupe = "LIB_ZE2010",
fondContour = NULL) # sans contour
ze13 <- zonage_a_facon(fondMaille = com_dep_13_30_83_84, groupe = donnees_a_facon,
idMaille = "DEPCOM", idGroupe = "ZE2010", libGroupe = "LIB_ZE2010",
fondContour = depm[depm$CODE=="13",]) # avec contour

# affiche les ZE2010 ayant au moins une partie dans les Bouches-du-Rhone
plot(sf::st_geometry(ze13etplus), col = "transparent", border = "grey")
# affiche uniquement la partie des ZE2010 contenue dans les Bouches-du-Rhone
plot(sf::st_geometry(ze13), col = "transparent", add = TRUE)
# affiche le contour des Bouches-du-Rhone
plot(sf::st_geometry(depm[depm$CODE=="13",]), border = "red", add = TRUE)
```



# Index

- \* **datasets**
  - donnees, 16
- \* **documentation**
  - add\_legende\_, 4
  - add\_source, 7
  - add\_titre, 8
  - add\_typo\_symboles, 9
  - affiche\_paLETTE, 10
  - calcul\_, 11
  - coord\_legende, 14
  - coordonnees\_etiquettes, 13
  - distrib\_variable, 15
  - export\_, 19
  - export\_qgis\_, 20
  - largeur\_fleche, 22
  - leaflet\_, 23
  - plot\_, 30
  - rapport\_ronds, 38
  - rayon\_ronds, 39
  - recup\_paLETTE, 41
  - set\_bordure\_ronds, 42
  - set\_couleur\_, 43
  - set\_fonds\_simples, 46
  - set\_opacite\_elargi, 47
  - set\_pop\_up, 48
  - set\_style\_oursins, 49
  - shiny\_, 50
  - zonage\_a\_facon, 55
- \* **package**
  - oceanis-package, 3
- add\_legende\_, 4
- add\_legende\_classes, 14, 28
- add\_legende\_classes (add\_legende\_), 4
- add\_legende\_fonds\_simples (add\_legende\_), 4
- add\_legende\_joignantes, 14, 28
- add\_legende\_joignantes (add\_legende\_), 4
- add\_legende\_ronds, 14, 28
- add\_legende\_ronds (add\_legende\_), 4
- add\_legende\_saphirs, 14, 28
- add\_legende\_saphirs (add\_legende\_), 4
- add\_legende\_typo, 14, 28
- add\_legende\_typo (add\_legende\_), 4
- add\_legende\_typo\_symboles, 28
- add\_legende\_typo\_symboles (add\_legende\_), 4
- add\_source, 7, 28
- add\_titre, 8, 28
- add\_typo\_symboles, 9
- affiche\_paLETTE, 10
- calcul\_, 11
- calcul\_part\_ens, 28, 37
- calcul\_part\_ens (calcul\_), 11
- calcul\_ratio, 28, 37
- calcul\_ratio (calcul\_), 11
- calcul\_tx\_evol\_ann\_moy, 28, 37
- calcul\_tx\_evol\_ann\_moy (calcul\_), 11
- calcul\_tx\_evol\_global, 28, 37
- calcul\_tx\_evol\_global (calcul\_), 11
- calcullette, 28, 37
- calcullette (calcul\_), 11
- com\_dep\_13\_30\_83\_84 (donnees), 16
- coord\_legende, 6, 10, 14, 28
- coordonnees\_etiquettes, 13, 37
- depm (donnees), 16
- distrib\_variable, 15, 28, 37
- donnees, 16
- donnees\_a\_facon (donnees), 16
- donnees\_biloc (donnees), 16
- donnees\_biloc\_saphirs (donnees), 16
- donnees\_monoloc (donnees), 16
- export\_, 19
- export\_jpeg, 28
- export\_jpeg (export\_), 19
- export\_pdf, 28
- export\_pdf (export\_), 19

- export\_png, 28
- export\_png (export\_), 19
- export\_qgis\_, 20
- export\_qgis\_classes, 28, 53
- export\_qgis\_classes (export\_qgis\_), 20
- export\_qgis\_classes\_ronds, 28, 53
- export\_qgis\_classes\_ronds (export\_qgis\_), 20
- export\_qgis\_joignantes, 28, 54
- export\_qgis\_joignantes (export\_qgis\_), 20
- export\_qgis\_oursins, 28, 54
- export\_qgis\_oursins (export\_qgis\_), 20
- export\_qgis\_ronds, 28, 53
- export\_qgis\_ronds (export\_qgis\_), 20
- export\_qgis\_ronds\_classes, 28, 53
- export\_qgis\_ronds\_classes (export\_qgis\_), 20
- export\_qgis\_saphirs, 28, 54
- export\_qgis\_saphirs (export\_qgis\_), 20
- export\_qgis\_typo, 28, 54
- export\_qgis\_typo (export\_qgis\_), 20
  
- fram (donnees), 16
  
- ggplot, 16
  
- largeur\_fleche, 22, 28, 37
- leaflet\_, 23
- leaflet\_classes, 6–8, 10, 14, 19, 21, 43, 45, 47–50, 53
- leaflet\_classes (leaflet\_), 23
- leaflet\_classes\_ronds, 6–8, 10, 14, 19, 21, 39, 40, 43, 45, 47–50, 53
- leaflet\_classes\_ronds (leaflet\_), 23
- leaflet\_fonds\_simples, 6, 10, 47, 50
- leaflet\_fonds\_simples (leaflet\_), 23
- leaflet\_joignantes, 6–8, 10, 14, 19, 21, 22, 43, 45, 47, 48, 50, 53
- leaflet\_joignantes (leaflet\_), 23
- leaflet\_oursins, 6–8, 10, 14, 19, 21, 43, 45, 47, 48, 50, 53
- leaflet\_oursins (leaflet\_), 23
- leaflet\_ronds, 6–8, 10, 14, 19, 21, 39, 40, 43, 45, 47–50, 53
- leaflet\_ronds (leaflet\_), 23
- leaflet\_ronds\_classes, 6–8, 10, 14, 19, 21, 39, 40, 43, 45, 47–50, 53
- leaflet\_ronds\_classes (leaflet\_), 23
- leaflet\_saphirs, 6–8, 10, 14, 19, 21, 22, 43, 45, 47, 48, 50, 53
- leaflet\_saphirs (leaflet\_), 23
- leaflet\_typo, 6–8, 10, 14, 19, 21, 43, 45, 47–50, 53
- leaflet\_typo (leaflet\_), 23
  
- oceanis-package, 3
  
- palettes\_insee (donnees), 16
- plot\_, 30
- plot\_classes, 13
- plot\_classes (plot\_), 30
- plot\_classes\_ronds, 13
- plot\_classes\_ronds (plot\_), 30
- plot\_joignantes, 13
- plot\_joignantes (plot\_), 30
- plot\_oursins, 13
- plot\_oursins (plot\_), 30
- plot\_ronds, 13
- plot\_ronds (plot\_), 30
- plot\_ronds\_classes, 13
- plot\_ronds\_classes (plot\_), 30
- plot\_saphirs, 13
- plot\_saphirs (plot\_), 30
- plot\_typo, 13
- plot\_typo (plot\_), 30
- plot\_typo\_symboles (plot\_), 30
  
- rapport\_ronds, 28, 37, 38
- rayon\_ronds, 28, 37, 39
- recup\_palette, 37, 41, 43, 45
- regm (donnees), 16
  
- set\_bordure\_ronds, 42
- set\_couleur\_, 43
- set\_couleur\_classes, 11, 42
- set\_couleur\_classes (set\_couleur\_), 43
- set\_couleur\_joignantes (set\_couleur\_), 43
- set\_couleur\_ronds (set\_couleur\_), 43
- set\_couleur\_saphirs (set\_couleur\_), 43
- set\_couleur\_typo (set\_couleur\_), 43
- set\_fonds\_simples, 28, 46
- set\_opacite\_elargi, 47
- set\_pop\_up, 48
- set\_style\_oursins, 49
- shiny\_, 50
- shiny\_classes (shiny\_), 50

shiny\_classes\_ronds (shiny\_), [50](#)  
shiny\_joignantes (shiny\_), [50](#)  
shiny\_oursins (shiny\_), [50](#)  
shiny\_ronds (shiny\_), [50](#)  
shiny\_ronds\_classes (shiny\_), [50](#)  
shiny\_saphirs (shiny\_), [50](#)  
shiny\_typo (shiny\_), [50](#)  
  
zonage\_a\_facon, [28](#), [37](#), [55](#)