

Package ‘oceanis’

August 29, 2019

Type Package

Title Cartography for Statistical Analysis

Version 1.0.1

Date 2019-08-23

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' technology).
- 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 (>= 3.4.4),shiny (>= 1.0.5),shinyBS (>= 0.61)

Imports utils (>= 3.3.3),shinyjs (>= 1.0),shinythemes (>= 1.1.1),DT (>= 0.2),foreign (>= 0.8.67),sf (>= 0.6.3),sfc (>= 0.1.0),stringr (>= 1.2.0),leaflet (>= 2.0.2),leaflet.extras (>= 1.0.0),classInt (>= 0.2.3),units (>= 0.6.0),grDevices (>= 3.3.3),readODS (>= 1.6.7),xlsx (>= 0.5.7),ggplot2 (>= 2.2.1),dplyr (>= 0.7.2),graphics (>= 3.3.3),mapview (>= 2.5.0),tidyr (>= 0.7.1),lwgeom (>= 0.1.4),munsell (>= 0.5.0)

Suggests knitr, rmarkdown, rio

VignetteBuilder knitr

Encoding UTF-8

Author Sébastien CALVET - PSAR-AT - DR Provence-Alpes-Cote d'Azur - INSEE [cre, aut],
Sophie AUDRIC - PSAR-AT - DR Provence-Alpes-Cote d'Azur - INSEE [aut],
SED - DR Hauts-de-France - INSEE [ctb],
PSAR-SL - DR Auvergne-Rhone-Alpes - INSEE [ctb],
PSAR-EEP - DR Occitanie - INSEE [ctb],
SED - DR Pays-de-la-Loire - INSEE [ctb]

Maintainer Sébastien CALVET <sebastien.calvet@insee.fr>

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-28 23:50:02 UTC

R topics documented:

oceanis-package	2
add_fond_osm	4
add_legende	5
add_source_titre	8
add_typo_symboles	9
calcul_variable_classes	10
coordonnees_etiquettes	11
coord_legende	12
distrib_variable	13
donnees	15
export_image	17
export_qgis	18
largeur_fleche	20
leaflet	21
plot	27
rapport_ronds	35
rayon_ronds	36
recup_palette	37
set_couleur	39
set_fonds_simples	41
set_opacite_elargi	42
set_style_oursins	43
shiny	44
zonage_a_facon	48
Index	51

oceanis-package

Cartography for Statistical Analysis

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' technology). - Create interactive maps through zoom and pop-up ('leaflet' technology). - Create frozen maps with the possibility to add labels.

Details

Package : oceanis

Type : Package

Version : 0.8.6

Date : 2019-07-05

License : GPL(>=2.0)

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

Le package oceanis repond a 3 types de besoins :

- Creer des cartes fixes avec la possibilite d'ajouter des etiquettes. Ideal pour les impressions de documents (.pdf ou autres). Peut etre integre dans les applications.
- Creer des cartes interactives grace au zoom et aux pop-up (technologie leaflet). Utile pour l'integration dans une application ou dans une presentation en direct.
- Creer des cartes dans un environnement web ou les parametres sont modifiables a la volee (technologie R-Shiny + leaflet). Utile pour l'integration dans une application ou dans une presentation en direct. Ici, la carte est externalisee vers une nouvelle fenetre du navigateur.

Les fonctions de creation de cartes sont classees en 3 categories, correspondant aux 3 besoins precedents. Elles sont respectivement prefixees par `plot_`, `leaflet_` et `shiny_`.

D'autres fonctions existent pour completer l'offre :

Des fonctions pour exporter une carte leaflet en projet Qgis, des fonctions pour modifier son apparence, des fonctions permettant de creer ses propres zonages (connus ou a facon) a partir de tableaux de donnees.

Un exemple possible d'enchainement de fonctions :

- 1- Chargement du package oceanis (`library(oceanis)`)
- 2- Chargement des donnees avec la fonction `import()` du package `rio`
- 3- Chargement des fonds de carte avec la fonction `read_sf()` du package `sf`
- 4- Appel a une fonction de creation de carte (`shiny_()`, `leaflet_()` ou `plot_()`)
- 5- Modification du parametrage (legende, fonds supplementaire, 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 resumer, voici un comparatif des fonctionnalites proposees :

PLOT

Interactivite : 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

Author(s)

SÃ©bastien CALVET - PSAR-AT - DR Provence-Alpes-Cote d'Azur - INSEE [cre, aut], Sophie AUDRIC - PSAR-AT - DR Provence-Alpes-Cote d'Azur - INSEE [aut], SED - DR Hauts-de-France - INSEE [ctb], PSAR-SL - DR Auvergne-Rhone-Alpes - INSEE [ctb], PSAR-EEP - DR Occitanie - INSEE [ctb], SED - DR Pays-de-la-Loire - INSEE [ctb] Maintenir: SÃ©bastien CALVET <sebastien.calvet@insee.fr>

add_fond_osm

Add an OpenStreetMap background on a 'leaflet' map

Description

Add an OpenStreetMap background on a 'leaflet' map.

Usage

```
add_fond_osm(map)
```

Arguments

map objet leaflet.

Details

L'ajout du fond OSM est definitif pour l'objet leaflet mis en parametre.

Pour supprimer le fond OSM de la carte, l'objet leaflet doit etre regenere a partir d'une des fonctions leaflet_XXX.

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_fond_osm(map)

map
```

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 a NULL.

Pour supprimer le titre, reexecutez la fonction `add_legende_xxx` en laissant le parametre `titre` a `NULL`.

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

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

Les valeurs de la legende des ronds et des fleches peuvent etre arrondies grace au parametre `precision`.

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

Si le titre et les valeurs de legende sont trop decales, 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_titre *Add a source or title on a 'leaflet' map*

Description

Add a source or title on a 'leaflet' map.

Usage

```
add_source(map, source)

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

Arguments

map	objet leaflet.
source	chaîne de caractères (character). Source de la carte.
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 la source, reexecutez la fonction `add_source` en laissant le parametre `source` a NULL.

Pour supprimer le titre ou le sous-titre, reexecutez la fonction `add_titre` en laissant le parametre `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_source(map = map, source = "Source : INSEE - RP2016")
```


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

```

calcul_variable_classes

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 caractères (character). Variable en volume de la table.
formule	chaîne de caractères (character). Formule libre pour le calcul d'une variable de classes. Seuls les caractères + - * / ^ () sont acceptés.

Details

Les formules utilisées 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

fondMaille	objet sf. Fond de carte.
listeCode	vecteur de caractères (character). Liste des codes de la maille à afficher sur la carte.

Details

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

Ce tableau peut être modifié pour changer le style d'une ou plusieurs étiquettes mais aussi leur position (x et y) pour éviter le chevauchement.

Pour afficher la carte avec les étiquettes formatées, il faut passer ce tableau dans le paramètre "étiquettes" 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

Returns the lon / lat coordinates of a 'leaflet' map's legend

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

Usage

```
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")
```

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

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)
```

export_image	<i>Export a 'leaflet' map in image format</i>
--------------	---

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 spécifié.

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	<i>Creation 'leaflet' maps</i>
---------	--------------------------------

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, dom = "0", fondChx = NULL,
colPos = "#CD853F", colNeg = "#6495ED", colBorder = "white", opacityElargi = 0.6,
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, dom = "0", 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, colBorder = "white",
precision = 1, dom = "0", fondChx = 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", colBorderRonds = "#303030", precision = 1, dom = "0",
fondChx = NULL, map_proxy = NULL)
```

```
leaflet_typo(data, fondMaille, fondSuppl = NULL, idData, varTypo, dom = "0",
map_proxy = NULL)
```

```
leaflet_oursins(data, fondMaille, fondSuppl = NULL, idDataDepart, idDataArrivee,
varFlux, filtreVol = 0, filtreDist = 100, filtreMajeurs = 10,
decalageAllerRetour = 0, decalageCentroid = 0, dom = "0", 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 = "#CD853F", colBorder = "#303030", dom = "0",
map_proxy = NULL)

```

```

leaflet_saphirs(data, fondMaille, typeMaille, fondSuppl = NULL, idDataDepart,
idDataArrivee, varFlux, largeurFlecheMax = NULL, direction = "Ent",
filtreVol = 0, colEntree = "#CD853F", colSortie = "#6495ED",
colBorder = "#303030", dom = "0", 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.
<code>rayonRond</code>	valeur numerique (numeric). Rayon du rond le plus grand en metres. Par default a NULL.
<code>rapportRond</code>	valeur numerique (numeric). Rapport entre l'aire du rond le plus grand et la valeur maximale des donnees en volume. Par default a NULL.
<code>methode</code>	chaîne de caracteres (character). A choisir parmi "kmeans" (par default), "fisher", "jenks" ou "quantile".
<code>nbClasses</code>	valeur numerique (numeric). Nombre de classes. Par default 3 classes.
<code>bornes</code>	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 "#CD853F".
colNeg	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#6495ED".
colBorder	chaîne de caracteres (character). Couleur nommee (par exemple "white") ou hexadecimal (par exemple "#FFFFFF"). Par default "white" (ou "#303030" pour les fleches).
colBorderRonds	chaîne de caracteres (character). Couleur nommee (par exemple "grey") ou hexadecimal (par exemple "#808080"). Par default "#303030".
colBorderClasses	chaîne de caracteres (character). Couleur nommee (par exemple "grey") ou hexadecimal (par exemple "#808080"). Par default "white".
stylePalette	chaîne de caracteres (character). Palette de la charte INSEE. A choisir entre "InseeFlash", "InseeAnalyse", "InseeDossier", "InseePremiere" ou "default" (par default).
colFleche	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#CD853F".
colEntree	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#CD853F".
colSortie	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#6495ED".
opacityElargi	valeur numerique (numeric). Opacite de la representation elargie. Par default a 0.6 (valeur comprise entre 0-transparent et 1-opaque).

dom	chaîne de caractères (character). "0" (par défaut pour la France métropolitaine), "971" (Guadeloupe), "972" (Martinique), "973" (Guyane), "974" (La Réunion) ou "976" (Mayotte). Ce paramètre permet de définir le système de projection de la carte.
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.
map_proxy	objet leaflet ou leaflet_proxy. Pour l'intégration des fonctions leaflet dans les applications shiny (cf vignette). Par défaut à 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).

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 implementant le parametre bornes d'un vecteur de valeurs numeriques 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'interieur des ronds (fonction `leaflet_ronds_classes()`)

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

Typologie Meme conseil que pour les analyses en classes. Le choix des couleurs ne doit pas etre un degrade si il n'existe pas de classement dans les modalites.

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.

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](#),[add_fond_osm](#),
[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, dom = "0", fondChx = NULL,
  precisionLegRonds = 0, titreLeg = "", xLeg = NULL, yLeg = NULL,
  titreCarte = "", sourceCarte = "", etiquettes = NULL, colPos = "#CD853F",
  colNeg = "#6495ED", 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 = "", xLeg = NULL, yLeg = 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, dom = "0",
  fondChx = NULL, titreLegRonds = "", titreLegClasses = "",
  xLegRonds = NULL, yLegRonds = NULL, xLegClasses = NULL,

```

```
yLegClasses = 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, dom = "0",
fondChx = NULL, titreLegRonds = "", titreLegClasses = "",
xLegRonds = NULL, yLegRonds = NULL, xLegClasses = NULL,
yLegClasses = 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,
colBorder = "white", xlim = NULL, ylim = NULL)
```

```
plot_typo_symboles(fondPoints, listFonds, dom = "0", types = NULL, couleurs = NULL,
tailles = NULL, epaisseurs = NULL, titreLeg = "", xLeg = NULL, yLeg = 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,
dom = "0", precisionLegFleches = 0, titreLeg = "",
xLeg = NULL, yLeg = NULL, titreCarte = "", sourceCarte = "",
etiquettes = NULL, colFleche = "#CD853F",
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, dom = "0", precisionLegFleches = 0,
titreLeg = "", xLeg = NULL, yLeg = NULL, titreCarte = "", sourceCarte = "",
etiquettes = NULL, colEntree = "#CD853F", colSortie = "#6495ED",
colBorder = "white", colBorderMaille = "black", xlim = NULL, ylim = 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 issus d'un zonage administratifs ou d'études ("REG", "DEP", "UU", "AU", "ZE", "BV", "EPCI" ou "COM").
<code>fondSousAnalyse</code>	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 default a NULL.
<code>fondPoints</code>	objets sf. Fond de points.
<code>fondSurAnalyse</code>	list d'objets sf. Liste de fonds de carte a ajouter au-dessus de l'analyse. Par exemple, un fond de departement. Par default a NULL.
<code>listFonds</code>	list d'objets sf. Liste de fonds de carte a ajouter.
<code>idData</code>	chaîne de caracteres (character). Variable identifiant la maille.
<code>idDataDepart</code>	chaîne de caracteres (character). Variable identifiant le territoire de depart du flux.
<code>idDataArrivee</code>	chaîne de caracteres (character). Variable identifiant le territoire d'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.
<code>rayonRond</code>	valeur numerique (numeric). Rayon du rond le plus grand en metres.
<code>rapportRond</code>	valeur numerique (numeric). Rapport entre l'aire du rond le plus grand et la valeur maximale des donnees en volume.
<code>methode</code>	chaîne de caracteres (character). A choisir parmi "kmeans" (par default), "fisher", "jenks" ou "quantile".
<code>nbClasses</code>	valeur numerique (numeric). Nombre de classes. Par default 3 classes.
<code>bornes</code>	vecteur de valeurs numeriques (numeric). Le nombre de bornes correspond au nombre de classes -1. Par default a NULL.
<code>precisionLegRonds</code>	valeur numerique (numeric). Arrondit les valeurs de la legende des ronds. 0 (par default) pour arrondir a l' unite, -1 pour arrondir a la dizaine, -2 a la centaine...
<code>precisionLegClasses</code>	valeur numerique (numeric). Arrondit les valeurs en ratio pour le calcul des bornes de classes. Par default, 1 chiffre apres la virgule.
<code>precisionLegFleches</code>	valeur numerique (numeric). Arrondit les valeurs de la legende des fleches. 0 (par default) pour arrondir a l' unite, -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 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 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).
dom	chaîne de caracteres (character). "0" (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.
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 "#CD853F".
colNeg	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#6495ED".
stylePalette	chaîne de caracteres (character). Palette de la charte INSEE. A choisir entre "InseeFlash", "InseeAnalyse", "InseeDossier", "InseePremiere" ou "default" (par default).
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 caracteres (character). Couleur nommee (par exemple "grey") ou hexadecimal (par exemple "#808080"). Par default "#303030".
colFleche	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#CD853F".
colEntree	chaîne de caracteres (character). Couleur nommee (par exemple "orange") ou hexadecimal (par exemple "#FFA500"). Par default "#CD853F".
colSortie	chaîne de caracteres (character). Couleur nommee (par exemple "blue") ou hexadecimal (par exemple "#0000FF"). Par default "#6495ED".

xlim	vecteur de numeriques (numeric). Coordonnees xmin et xmax de la carte. Par default a NULL.
ylim	vecteur de numerique (numeric). Coordonnees ymin et ymax de la carte. Par default a NULL.

Details

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

Elles peuvent etre personnalisees grace aux parametres de style des fonctions plot_().

Ronds proportionnels L'interieur des cercles est de couleur orange (couleur par default) pour representer les valeurs positives de la variable en volume et bleu pour les valeurs negatives.

Le parametre fondChx est a 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'hotel de ville de la commune.

Analyse en classes En semiologie, il est deconseille de multiplier le nombre de classes pour des raisons de visibilite et de clarte de la carte.

Si les donnees ne comprennent que des valeurs positives ou que des valeurs negatives, il est conseille 5 classes au maximum.

Si les donnees comprennent a la fois des valeurs positives et negatives, le nombre maximum conseille est 9 classes.

Il est possible de specifier manuellement les bornes des classes en implementant le parametre bornes d'un vecteur de valeurs numeriques 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 specifier les valeurs min et max.

Analyse en classes dans ronds proportionnels L'analyse en classes colore l'interieur des ronds : (fonction plot_ronds_classes())

Ronds proportionnels sur analyse en classes Les ronds proportionnels sont poses 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 etre un degrade si il n'existe pas de classement dans les modalites.

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

Typologie en symboles Les fonds de carte doivent etre passes en list et peuvent etre personnalises (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 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 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 représenter 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'epaisseur 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 étiquettes, 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 à afficher, les coordonnées x et y du centre du territoire (position des étiquettes 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 défaut), 3 (italique) et 4 (gras italique).

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

Il est également possible de modifier les coordonnées x et y pour déplacer les étiquettes et ainsi éviter leur chevauchement.

- Pour afficher la carte avec les étiquettes formatées, il faut passer ce tableau dans le paramètre "étiquettes" de la fonction `plot`.

Il est tout de même possible de zoomer sur la carte en spécifiant les paramètres `xlim` et `ylim`. Par défaut, la carte est centrée sur le fond de maille. Pour connaître les limites par défaut, exécuter 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 flèches (oursins, flèches joignantes, flèches saphirs).

Affiche dans le `plot` la carte demandée.

See Also

`zonage_a_facon`,
`coordonnees_etiquettes`,
`rayon_ronds`, `rappport_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 légère différence entre la valeur du rapport passée en paramètre de la fonction `leaflet_ronds` et la valeur renvoyée par la fonction `rapport_ronds`. Cette différence s'explique par l'arrondi de π à 6 chiffres après la virgule.

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

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

Returns a palette of the graphic chart of INSEE

Description

Returns a palette of the graphic chart of INSEE.

Usage

```
recup_palette(stylePalette)
```

Arguments

`stylePalette` chaîne de caracteres (character). A choisir parmi "InseeFlash", "InseeAnalyse", "InseeDossier", "InseePremiere" ou "default".

Details

Les palettes proposees sont celles utilisees dans les publications Insee. Seule la palette par defaut propose des couleurs supplementaires par rapport a la palette utilisee dans la publication InseeFlash.

La fonction renvoie une liste de deux vecteurs. Le premier element de la liste correspond aux couleurs des valeurs positives et le deuxieme aux couleurs des valeurs negatives.

Les couleurs sont classées du plus foncé au plus clair pour les palettes des valeurs positives et du plus clair au plus foncé pour les palettes des valeurs négatives.

Le code des couleurs est le code hexadécimal.

- InseeFlash positives : "#9B231C", "#B24B1D", "#D47130", "#E4A75A", "#F2CE93"
- InseeFlash negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseeAnalyse positives : "#5E2057", "#853567", "#8E5981", "#BA97B2", "#D7C0CC"
- InseeAnalyse negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseeDossier positives : "#4F185E", "#65317B", "#9475A5", "#BFA5C6", "#E7D1E5"
- InseeDossier negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseePremiere positives : "#7F0029", "#CC1543", "#DE635B", "#F79C85", "#FDE3DE"
- InseePremiere negatives : "#ECF4D8", "#CDD78C", "#91B778", "#549534", "#005941"
- default positives : "#5A0A14", "#82141B", "#9B231C", "#B24B1D", "#D47130", "#E4A75A", "#F2CE93"
- default negatives : "#C9DAF0", "#95BAE2", "#5182B6", "#005289", "#003269", "#001E5A", "#000050"

Value

Retourne une liste de deux vecteurs caractères.

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

[set_couleur_classes](#)

Examples

```
recup_palette("InseeFlash")

#[[1]]
#[1] "#9B231C" "#B24B1D" "#D47130" "#E4A75A" "#F2CE93"
#[[2]]
#[1] "#ECF1FA" "#C9DAF0" "#95BAE2" "#5182B6" "#005289"
```

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 = "#CD853F", colorNeg = "#6495ED", colBorder = "white",  
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 = "#CD853F", colBorder = "black",  
map_leaflet = NULL)
```

```
set_couleur_saphirs(map, colEntree = "#CD853F", colSortie = "#6495ED",  
colBorder = "black", map_leaflet = NULL)
```

Arguments

map	objet leaflet.
colorPos	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#CD853F".
colorNeg	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#6495ED".
stylePalette	chaîne de caractères (character). À choisir parmi "InseeFlash", "InseeAnalyse", "InseeDossier", "InseePremiere" ou "defaut" (par défaut).
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".
colFleche	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#CD853F".
colEntree	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#CD853F".

colSortie	chaîne de caractères (character). Couleur nommée ou hexadécimale. Par défaut "#6495ED".
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 dans les publications Insee. Seule la palette par défaut propose des couleurs supplémentaires par rapport à la palette utilisée dans la publication InseeFlash.

Les couleurs sont classées du plus foncé au plus clair pour les palettes des valeurs positives et du plus clair au plus foncé pour les palettes des valeurs négatives.

Le code des couleurs est le code hexadécimale.

- InseeFlash positives : "#9B231C", "#B24B1D", "#D47130", "#E4A75A", "#F2CE93"
- InseeFlash negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseeAnalyse positives : "#5E2057", "#853567", "#8E5981", "#BA97B2", "#D7C0CC"
- InseeAnalyse negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseeDossier positives : "#4F185E", "#65317B", "#9475A5", "#BFA5C6", "#E7D1E5"
- InseeDossier negatives : "#ECF1FA", "#C9DAF0", "#95BAE2", "#5182B6", "#005289"
- InseePremiere positives : "#7F0029", "#CC1543", "#DE635B", "#F79C85", "#FDE3DE"
- InseePremiere negatives : "#ECF4D8", "#CDD78C", "#91B778", "#549534", "#005941"
- default positives : "#5A0A14", "#82141B", "#9B231C", "#B24B1D", "#D47130", "#E4A75A", "#F2CE93"
- default negatives : "#C9DAF0", "#95BAE2", "#5182B6", "#005289", "#003269", "#001E5A", "#000050"

Value

Retourne un objet de type leaflet.

References

Un convertisseur de couleurs pour visualiser une couleur à partir de son nom, son code hexadécimale 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_classes_ronds(data = donnees_monoloc, fondMaille = depm, idData = "COD_DEP",
varVolume = "POP_2015", varRatio = "VAR_AN_MOY", nbClasses = 4)
map <- set_couleur_ronds(map = map, colBorder = "grey")
map <- set_couleur_classes(map = map, stylePalette = "InseePremiere", 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 = "#91B778", 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 default a NULL.
epaisseurFonds	valeur numerique (numeric). Le vecteur comporte les valeurs numeriques pour l'epaisseur de la bordure des fonds. Par default a NULL.
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("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 concernee 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_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	chaine de caracteres (character). Couleur nommee (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, dom = "0", fondChx = NULL)
```

```
shiny_classes(data, fondMaille, fondMailleElargi = NULL, fondContour, fondSuppl = NULL,
idData, varRatio, dom = "0")
```

```
shiny_ronds_classes(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varVolume, varRatio, dom = "0",
fondChx = NULL)
```

```
shiny_classes_ronds(data, fondMaille, fondMailleElargi = NULL, fondContour,
fondSuppl = NULL, idData, varVolume, varRatio, dom = "0",
fondChx = NULL)
```

```
shiny_typo(data, fondMaille, fondContour, fondSuppl = NULL, idData, varTypo, dom = "0")
```

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

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

```
shiny_saphirs(data, fondMaille, typeMaille, fondContour, fondSuppl = NULL, idDataDepart,
idDataArrivee, varFlux, direction = "Ent", dom = "0")
```

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 départ 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.
<code>varRatio</code>	chaîne de caracteres (character). Variable en ratio pour l'analyse en classes.
<code>varTypo</code>	chaîne de caracteres (character). Variable de typologie.

varFlux	chaîne de caractères (character). Variable de flux pour les oursins, flèches joignantes ou flèches saphirs.
direction	chaîne de caractères (character). Type de flux. À choisir parmi "Ent" pour entrées (par défaut), "Sor" pour sorties et "Sol" pour soldes (entrées - sorties).
decalageAllerRetour	valeur numérique (numeric). Décalage entre les flèches aller-retour (km).
decalageCentroid	valeur numérique (numeric). Décalage des flèches depuis et vers les centroïdes des territoires (km).
dom	chaîne de caractères (character). "0" (par défaut pour la France métropolitaine), "971" (Guadeloupe), "972" (Martinique), "973" (Guyane), "974" (La Réunion) ou "976" (Mayotte). Ce paramètre permet de définir le système 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 centroïdes des communes (NULL par défaut) ou sur les chefs-lieux (chx) des communes.

Details

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

Les cartes produites possèdent les mêmes avantages que les cartes leaflet (déplacement de la carte, zoom/dezoom, pop-up d'informations, placement libre de la légende).

L'environnement R-Shiny propose en plus la possibilité de modifier à la volée les paramètres de la carte (ordre des fonds, ajout de fonds, représentation élargie, taille et rapport des ronds, méthode des classes, nombre de classes, bornes, filtre de flux, légende).

Les cartes peuvent être exportées en projet Qgis via un bouton de l'interface R-Shiny. La fonctionnalité crée un .zip à dézipper au même emplacement.

L'apparence des cartes (choix de la palette et des couleurs, étiquettes) n'est cependant pas modifiable. Le but des fonctions shiny est de proposer des cartes réalisables et modifiables rapidement avec un minimum de paramétrage. En effet, une mise en forme spécifique 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.

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 Meme 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")
}

```

Description

Creating a custom zoning from a group of entities.

Usage

```
zonage_a_facon(fondMaille, groupe, idMaille, idGroupe, libGroupe, fondContour = NULL,  
dom = "0")
```

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.
dom	chaîne de caracteres (character). "0" (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.

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)

See Also

[sf](#)

[read_sf](#)

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, dom = "0") # 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",], dom = "0") # 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

*Topic **datasets**

donnees, [15](#)

*Topic **documentation**

add_fond_osm, [4](#)

add_legende, [5](#)

add_source_titre, [8](#)

add_typo_symboles, [9](#)

calcul_variable_classes, [10](#)

coord_legende, [12](#)

coordonnees_etiquettes, [11](#)

distrib_variable, [13](#)

export_image, [17](#)

export_qgis, [18](#)

largeur_fleche, [20](#)

leaflet, [21](#)

plot, [27](#)

rapport_ronds, [35](#)

rayon_ronds, [36](#)

recup_palette, [37](#)

set_couleur, [39](#)

set_fonds_simples, [41](#)

set_opacite_elargi, [42](#)

set_style_oursins, [43](#)

shiny, [44](#)

zonage_a_facon, [48](#)

*Topic **package**

oceanis-package, [2](#)

add_fond_osm, [4](#), [25](#)

add_legende, [5](#)

add_legende_classes, [13](#), [25](#)

add_legende_classes (add_legende), [5](#)

add_legende_fonds_simples

(add_legende), [5](#)

add_legende_joignantes, [13](#), [25](#)

add_legende_joignantes (add_legende), [5](#)

add_legende_ronds, [13](#), [25](#)

add_legende_ronds (add_legende), [5](#)

add_legende_saphirs, [13](#), [25](#)

add_legende_saphirs (add_legende), [5](#)

add_legende_typo, [13](#), [25](#)

add_legende_typo (add_legende), [5](#)

add_legende_typo_symboles, [25](#)

add_legende_typo_symboles
(add_legende), [5](#)

add_source, [25](#)

add_source (add_source_titre), [8](#)

add_source_titre, [8](#)

add_titre, [25](#)

add_titre (add_source_titre), [8](#)

add_typo_symboles, [9](#)

calcul_part_ens, [25](#), [34](#)

calcul_part_ens
(calcul_variable_classes), [10](#)

calcul_ratio, [25](#), [34](#)

calcul_ratio (calcul_variable_classes),
[10](#)

calcul_tx_evol_ann_moy, [25](#), [34](#)

calcul_tx_evol_ann_moy
(calcul_variable_classes), [10](#)

calcul_tx_evol_global, [25](#), [34](#)

calcul_tx_evol_global
(calcul_variable_classes), [10](#)

calcul_variable_classes, [10](#)

calculette, [25](#), [34](#)

calculette (calcul_variable_classes), [10](#)

com_dep_13_30_83_84 (donnees), [15](#)

coord_legende, [7](#), [9](#), [12](#), [25](#)

coordonnees_etiquettes, [11](#), [34](#)

depm (donnees), [15](#)

distrib_variable, [13](#), [25](#), [34](#)

donnees, [15](#)

donnees_a_facon (donnees), [15](#)

donnees_biloc (donnees), [15](#)

donnees_biloc_saphirs (donnees), [15](#)

donnees_monoloc (donnees), [15](#)

export_image, [17](#)

- export_jpeg, 26
- export_pdf (export_image), 17
- export_pdf, 26
- export_pdf (export_image), 17
- export_png, 26
- export_png (export_image), 17
- export_qgis, 18
- export_qgis_classes, 25, 47
- export_qgis_classes (export_qgis), 18
- export_qgis_classes_ronds, 25, 47
- export_qgis_classes_ronds (export_qgis), 18
- export_qgis_joygnantes, 26, 47
- export_qgis_joygnantes (export_qgis), 18
- export_qgis_oursins, 26, 47
- export_qgis_oursins (export_qgis), 18
- export_qgis_ronds, 25, 47
- export_qgis_ronds (export_qgis), 18
- export_qgis_ronds_classes, 25, 47
- export_qgis_ronds_classes (export_qgis), 18
- export_qgis_saphirs, 26, 47
- export_qgis_saphirs (export_qgis), 18
- export_qgis_typo, 26, 47
- export_qgis_typo (export_qgis), 18

- fram (donnees), 15

- ggplot, 14

- largeur_fleche, 20, 25, 34
- leaflet, 21
- leaflet_classes, 5, 7–9, 13, 17, 19, 40, 42–44, 47
- leaflet_classes (leaflet), 21
- leaflet_classes_ronds, 5, 7–9, 13, 17, 19, 35, 37, 40, 42–44, 47
- leaflet_classes_ronds (leaflet), 21
- leaflet_fonds_simples, 7, 9, 42, 44
- leaflet_fonds_simples (leaflet), 21
- leaflet_joygnantes, 5, 7–9, 13, 17, 19, 20, 40, 42–44, 47
- leaflet_joygnantes (leaflet), 21
- leaflet_oursins, 5, 7–9, 13, 17, 19, 40, 42–44, 47
- leaflet_oursins (leaflet), 21
- leaflet_ronds, 5, 7–9, 13, 17, 19, 35, 37, 40, 42–44, 47
- leaflet_ronds (leaflet), 21
- leaflet_ronds_classes, 5, 7–9, 13, 17, 19, 35, 37, 40, 42–44, 47
- leaflet_ronds_classes (leaflet), 21
- leaflet_saphirs, 5, 7–9, 13, 17, 19, 20, 40, 42–44, 47
- leaflet_saphirs (leaflet), 21
- leaflet_typo, 5, 7–9, 13, 17, 19, 40, 42–44, 47
- leaflet_typo (leaflet), 21

- oceanis (oceanis-package), 2
- oceanis-package, 2

- plot, 27
- plot_classes, 12
- plot_classes (plot), 27
- plot_classes_ronds, 12
- plot_classes_ronds (plot), 27
- plot_joygnantes, 12
- plot_joygnantes (plot), 27
- plot_oursins, 12
- plot_oursins (plot), 27
- plot_ronds, 12
- plot_ronds (plot), 27
- plot_ronds_classes, 12
- plot_ronds_classes (plot), 27
- plot_saphirs, 12
- plot_saphirs (plot), 27
- plot_typo, 12
- plot_typo (plot), 27
- plot_typo_symboles (plot), 27

- rapport_ronds, 25, 34, 35
- rayon_ronds, 25, 34, 36
- read_sf, 49
- recup_palette, 34, 37, 40
- regm (donnees), 15

- set_couleur, 39
- set_couleur_classes, 38
- set_couleur_classes (set_couleur), 39
- set_couleur_joygnantes (set_couleur), 39
- set_couleur_ronds (set_couleur), 39
- set_couleur_saphirs (set_couleur), 39
- set_couleur_typo (set_couleur), 39
- set_fonds_simples, 25, 41
- set_opacite_elargi, 42
- set_style_oursins, 43
- sf, 49

shiny, [44](#)
shiny_classes (shiny), [44](#)
shiny_classes_ronds (shiny), [44](#)
shiny_joignantes (shiny), [44](#)
shiny_oursins (shiny), [44](#)
shiny_ronds (shiny), [44](#)
shiny_ronds_classes (shiny), [44](#)
shiny_saphirs (shiny), [44](#)
shiny_typo (shiny), [44](#)

zonage_a_facon, [25](#), [34](#), [48](#)