Graphiques

Auteur: Laura Tremblay-Boyer, contact: laura.t.boyer@gmail.com
Derniers changements: Nouméa, 27.11.2015

Fichier: `GS_Codes-graphiques.r`
Fonctions: `fig.2var()`

Les fonctionalités graphiques sont codées en utilisant la librairie `ggplot2` car elle donne plus de flexibilité au niveau du type de graphique et vous permet de customiser certains aspects visuels sans avoir à changer le code. Vous pouvez aussi sauvegarder l'image sous un object de classe ggplot (en utilisant la fonction `save`) qui contient le graphique ainsi que les données utilisées pour le bâtir. Finalement, il y a beaucoup de ressources d'aide pour ggplot en ligne qui vous permettront de créer des nouveaux types de graphique une fois la 'base' ggplot assemblée avec les données/l'aggrégation désirée.

Vous avez l'option de faire 3 types de graphiques principaux, tous accessibles avec la fonction `fig.2var()`: boxplot (typ.fig='boxplot'), barres (typ.fig='barre') et lignes/ séries temporelles (typ.fig='ligne'). Ces deux derniers incluent la barre d'erreur supérieure définie par la moyenne + l'écart type.

1 Spécification de la catégorie de données

La valeur de l'objet `bio.fig` détermine la catégorie de données utilisée dans le graphique. L'option la plus facile est d'utiliser les fonctions

`inv.fig()` `poissons.fig()` `LIT.fig()` `Quad.fig()`

qui sont des raccourcis qui accèdent directement à `fig.2var()` et modifient la valeur de l'object `bio.fig` automatiquement pour vous. Sinon, vous pouvez spécifier directement dans la console la valeur de l'objet `bio.fig` comme: 'inv', 'poissons', 'LIT', 'Quad' (valeur = 'inv' par défaut), et utiliser `fig.2var()` directement.

→ Lorsque `selection.projet()` est lancée, les tableaux nécéssaires à la création des graphiques pour le projet sont automatiquement créés dans l'environnement R. Vous pourrez les consulter directement au besoin. Le nom des objets contenant

ces tableaux est imprimé sur la console une première fois lors du lancement de `selection.projet()`, et ensuite à chaque fois que la fonction `fig.2var` est utilisée.

2 Options de la fonction `fig.2var()`

2.1 Options de base

1. `typ.fig`: le type de graph produit, 'barre', 'boxplot' ou 'ligne' (par défaut, 'barre')
2. `var.expl`: la variable de réponse qui sera représentée dans le graphique, si non-spécifiée, les valeurs par défaut sont la densité pour les invertébrés et les poissons, et la couverture (en %) de la catégorie Général/Coraux pour LIT et quadrats.
3. `var1`: la variable explicative en X (horizontal), par défaut la géomorphologie
4. `var2`: la variable explicative en Y (vertical), par défaut les campagnes (à noter que cette variable est aussi représentée par les couleurs du graphique)
5. `panneau` (optionnel): une variable explicative à inclure sous forme de fenêtre individuelle pour chaque niveau de la variable.

→ Les variables explicatives disponibles sont définies dans les objets `facteurs.spatio`, `facteurs.tempo` et `facteurs.taxo`. Pour LIT et Quadrats seulement, une variable additionelle, `LIT.lev`, peut être utilisée pour representer les niveaux définis dans la catégorie LIT spécifiée (voir argument `agLIT` ci-bas). Pour les variables de réponses acceptées par `var.expl`, voir `facteurs.var.expl`.

2.2 Options pour filtrer

1. `filtre.camp`: un filtre sur le type de campagne, 'A' pour les campagnes annuelles, 'S' pour les campagnes semestrielles ('A' par défaut)
2. `filtre, filtre2`: un (ou deux) filtre(s) optionel(s) sur n'importe quelle autre variable tant qu'elle soit présente sous forme de colonne dans le jeu de données. Vous pouvez utiliser deux fonctions raccourcis qui font le formattage du filtre pour vous selon si vous désirez inclure — `filtre.incl()` — ou exclure — `filtre.excl()` — la valeur spécifiée. Ces fonctions s'utilisent comme suit: le

premier argument est la variable/colonne sur lequel le filtre doit être appliqué, le deuxième argument est la (ou les) valeur(s) à inclure ou exclure:

Pour une valeur unique:
```
> fig.2var(..., filtre=filtre.incl("St", "ST1"))
```

Pour créer un vecteur avec de multiples valeurs à inclure ou exclure:
```
> fig.2var(..., filtre=filtre.excl("Campagne",c("A_2006","A_2007","A_2008")))
```

Alternativement, vous pouvez spécifier directement la valeur sous l'argument `filtre` ou `filtre2`, e.g. pour sélectionner les valeurs de la station "ST1", vous entreriez:
`fig.2var(..., filtre = "St == 'ST1'")`, ou, pour plusieurs valeurs,
`fig.2var(..., filtre = "Campagne %in% c('A_2013','A_2014')")` (attention à inclure des guillemets simples à l'intérieur des guillemets doubles... les fonctions `filtre.incl()` et `filtre.excl()` font ce travail de formattage de la ponctuation pour vous).

3. `tous.niveaux`: cette option, à mettre en TRUE (défaut) ou FALSE, détermine si les combinaisons de niveaux non-échantillonées sont incluses ou pas. C'est surtout important pour l'esthétique du graph, car normalement ggplot montre seulement les combinaisons de variables qui existent et peut changer la largeur ou répartition des barres/points pour remplir l'espace disponible (`tous.niveaux = FALSE`), alors que des fois on veut explicitement montrer qu'il n'y pas de valeurs pour une certaine variable (`tous.niveaux = TRUE`, e.g. garder un espace vide si une station n'a pas été échantillonée pendant plusieurs campagnes). Ça peut valoir la peine de comparer les deux options lorsque vous faites un graph.

4. Pour les invertébrés seulement, `agtaxo`: l'aggrégation taxonomique (Groupe, S_Groupe, Famille, Genre) et `typ.taxo`: la valeur à sélectionner dans l'aggrégation, eg. si agtaxo = "Groupe", typ.taxo pourrait être "Crustaces", "Mollusques", etc. (valeurs par défaut, Groupe et "Crustaces").

5. Pour LIT et Quadrats seulement, `agLIT`: la catégorie LIT à représenter (voir les colonnes de l'objet `index.LIT`: General, Acroporidae, Forme, Genre, All) (valeur par défaut: "General").

2.3 fig.etiq et definir des nouveaux niveaux

2.4 Ajustements des paramètres visuels

Plusieurs aspects visuels du graph peuvent être modifiés en rajoutant `+ commande` directement après la fonction qui lance le graph ou après un objet ggplot sauvegardé, soit:

```
> fig.2var() + commande
> obj <- fig.2var()
```

```
> obj + commande
```

Et plus concrètement, avec les vrais commandes:

```
 fig.2var(...  )  # plot produit...
obj = fig.2var() # mais si j'assigne à un object, pas de plot produit
obj + xlab("vavoum!!") # plot avec nouvelle étiquette en abscisse
fig.2var(...)  + ggtitle("je suis un titre") # ça fonctionne aussi
obj + verti.x.val # pour changer l'orientation des étiquettes
obj + no.x.val # ôter les étiquettes en X
```

Les commandes principales sont décrites ci-bas, suivi d'un astérisque lorsque ce sont des fonctions déjà incluses avec ggplot. Plusieurs commandes additionelles peuvent être trouvées en ligne en recherchant dans le manuel de ggplot (ou le forum de Stack Overflow avec l'option `[R]`). Notamment, la fonction `theme()` contrôle de nombreux paramètres visuels et est décrite ici: http://docs.ggplot2.org/current/theme.html, à voir aussi les fonctions utilisées par ggplot pour changer l'aspect du texte, `element_text()`.

inclure exemple modif texte, et rajouter boite autour de la legende, ... + theme vs. theme set?)

`+ ggtitle("mon titre")` : titre du graphique (*)

`+ coords_cartesian(xlim=c(xmin, xmax), ylim=c(ymin, ymax))` : limites du graphique (*)

`+ verti.x.val(angl=45, ...)` : change le format des valeurs sur l'ordonnée (voir `?element_text` pour les options de paramètres

`+ no.x.val` : ôte les valeurs de l'axe des abscisses

2.4.1 Palette de couleurs

Vous avez quelques options de palettes en addition de celle que j'ai spécifiée par défaut. Pour la changer, modifier la valeur de l'objet `palette.de.couleurs` dans `GS_Codes-graphiques.r`. Les options sont définies sous des objets s'appellant custom.colpal.a, custom.colpal.b, custom.colpal.c, custom.colpal.d. Vous pourrez aussi

faire votre propre palette en suivant le modèle défini dans les custom.colpal.X. Plus de détails sont inclus dans le commentaire de la section Couleurs de `GS_Codes-graphiques.r`.

2.4.2 Abbréviations pour géomorphologie

Les étiguettes pour la géomorphologie sont très longues et s'emboîtent souvent l'une sur l'autre. Pour utiliser un acronyme à la place du mot en entier, spécifiez "Geomorpho.abbrev" à la place de "Geomorpho" sous l'argument voulu.

2.5 Sauvegarde des graphiques

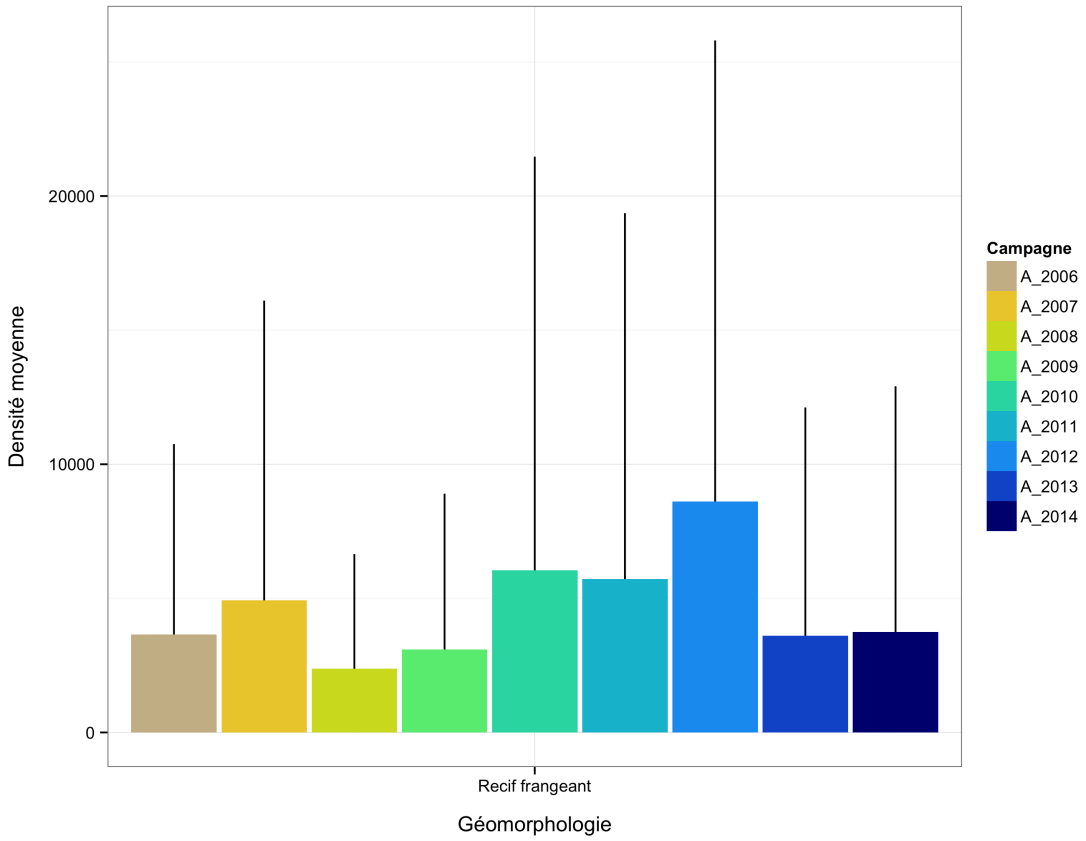
La fonction `ggsave()` permet de sauvegarder le graphique actif (voir la barre en haut de la fenêtre graphique) sous plusieurs formats (entre autres, .pdf et .png) ou un object formatté de class ggplot. Une fonction raccourci, `ggsave.proj()` , fait la sauvegarde automatique au dossier Graphiques du projet sélectionné, mais s'utilise exactement comme `ggsave()` autrement. Il suffit donc de spécifier l'argument `filename` en incluant l'extension graphique souhaitée, e.g.
ggsave.proj("nom-de-fichier-voulu.png")
ggsave.proj("nom-de-fichier-voulu.pdf")
ggsave.proj("nom-de-fichier-voulu.pdf", gg1) # gg1 est un objet ggplot

Voir ?ggsave pour les autres paramètres, entre autres `width` et `height` qui contrôlent les dimensions de l'image.

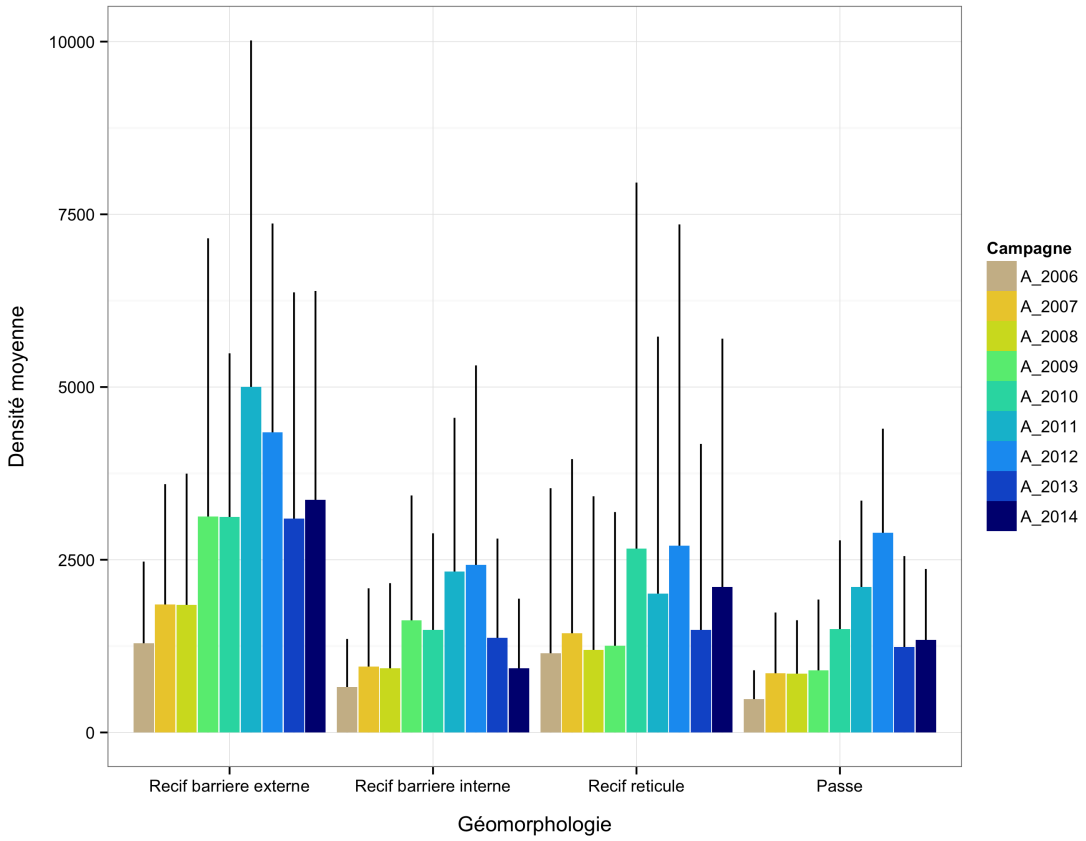
3 Exemples d'utilisation

Voici quelques exemples d'implémentation, c'est aussi important de jouer avec les diverses options, par e.g. en échangeant la position des variables (e.g. Campagne en var1, var2, panneau, etc.)

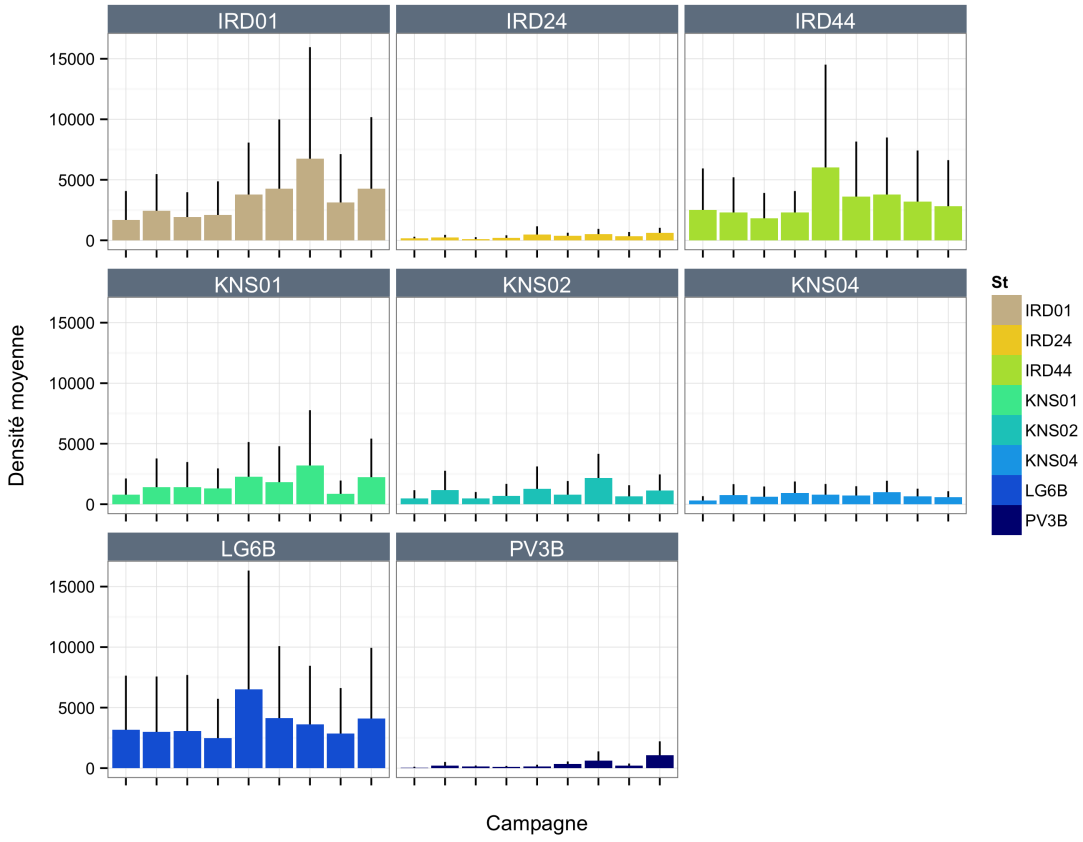
```
inv.fig("Campagne", filtre=filtre.incl("Geomorpho","Recif frangeant"))
```



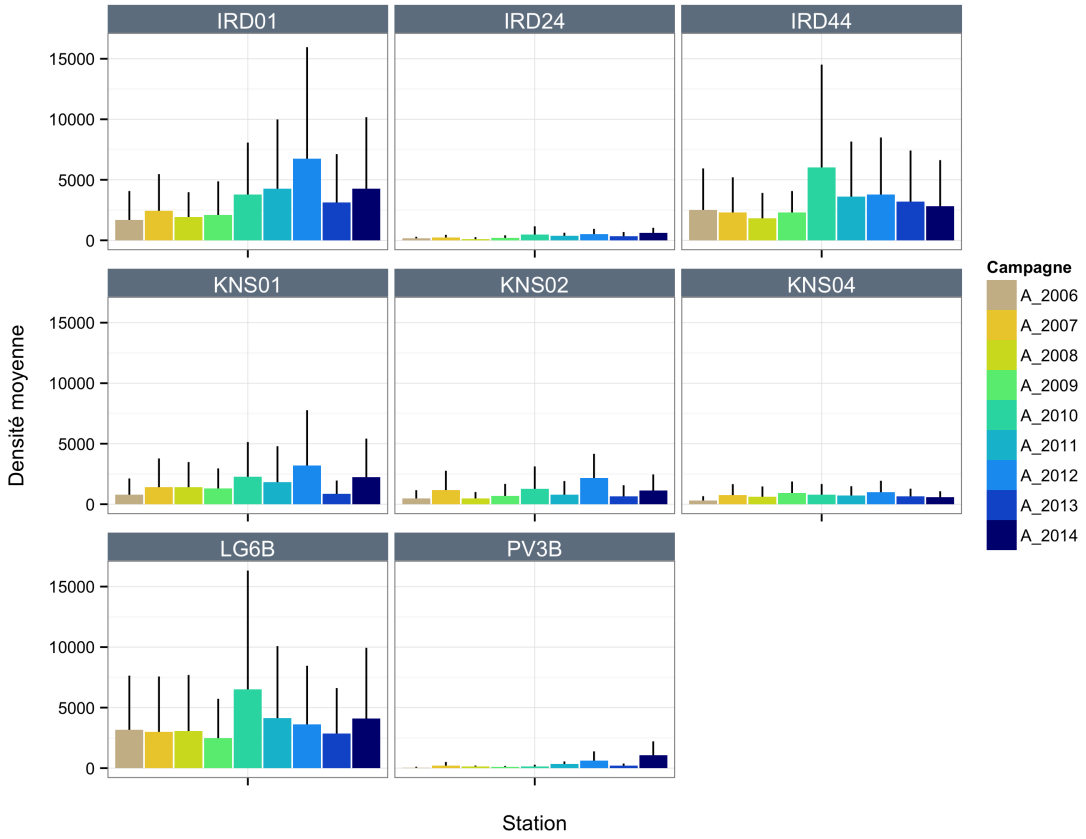
```
inv.fig("Campagne", filtre=filtre.excl("Geomorpho","Recif frangeant"))
```



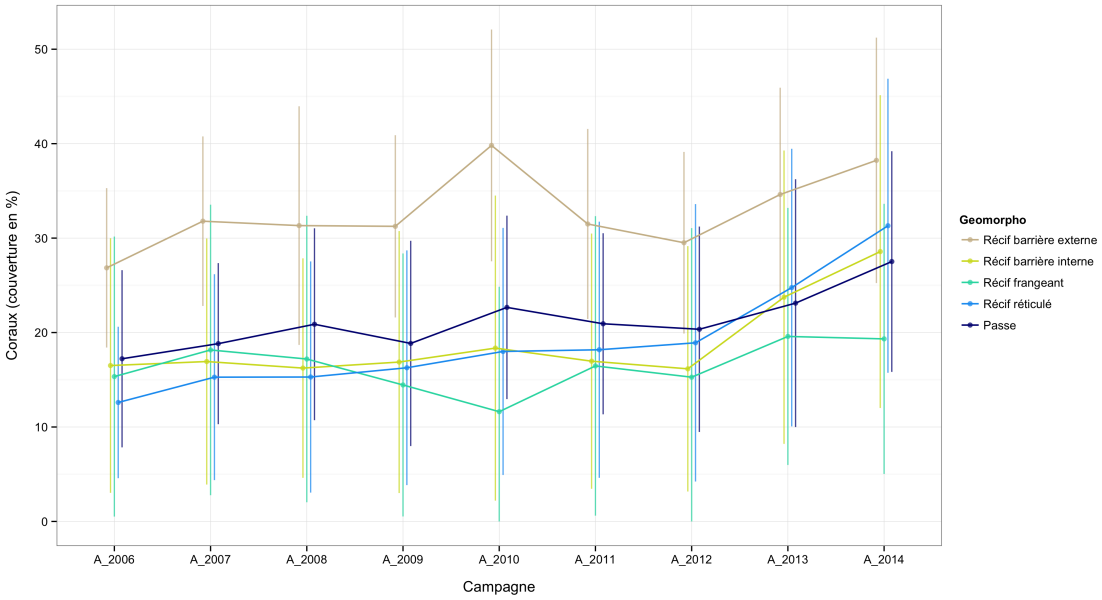
```
inv.fig(var1="Campagne", var2="St", filtre=filtre.incl("Geomorpho","Recif
reticule"), panneau="St") + no.x.val
```



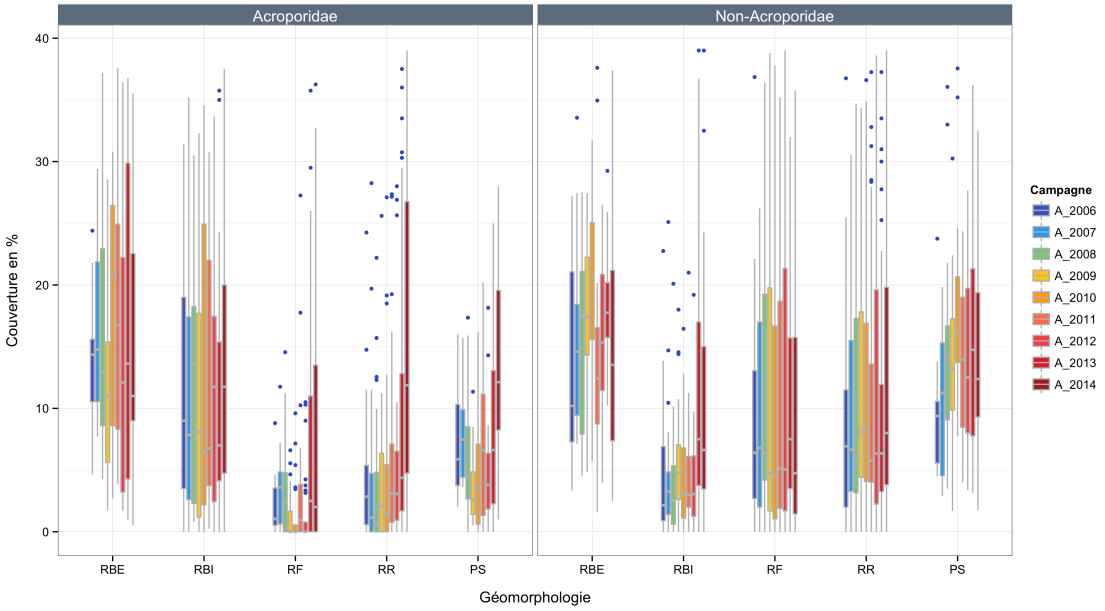
```
inv.fig(var1="Campagne", var2="St", filtre=filtre.incl("Geomorpho","Recif
reticule"), panneau="St") + No.x.val
```



```
LIT.fig(var1="Campagne", var2="Geomorpho", typ.fig="ligne", filtre=filtre.incl("LIT.l
"Coraux"))
```



```
LIT.fig(var1="Geomorpho.abbrev", agLIT="Acroporidae", tous.niveaux=FALSE,
panneau="LIT.lev", typ.fig="boxplot)
```



```
LIT.fig(agLIT="Forme", filtre=filtre.incl("LIT.lev", "Corail branchu"))
```