

Graphiques

Auteur: Laura Tremblay-Boyer, contact: lauratboyer@gmail.com

Nouméa le 8 septembre 2015

Fichier: `GS_Codes-graphiques.r`

Fonctions: `fig.2var()`

Les fonctionnalités graphiques sont codées en utilisant les fonctions de la librairie `ggplot` car elle donne plus de flexibilité au niveau du type de graphique et vous permettent de customiser certains aspects visuels sans avoir à changer le code. Vous pouvez aussi sauvegarder l'image sous un objet de classe `ggplot` (en utilisant la fonction `save`) qui contient le graphique mais aussi les données utilisées pour le bâtir. Finalement, il y a beaucoup de ressources d'aide pour `ggplot` en ligne qui pourront vous permettre 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 supérieur définie par la moyenne + l'écart type.

0.1 Spécification du type de données

Pour définir le type de données utilisées, vous spécifiez directement dans la console la valeur de l'objet `bio.fig` comme: `'inv'`, `'poissons'`, `'LIT'`, `'Quad'` (valeur = `'inv'` par défaut). Alternativement, les fonctions `inv.fig`, `poissons.fig`, `LIT.fig()` et `Quad.fig()` sont des raccourcis qui accèdent directement à `fig.2var()` en changeant la valeur de l'objet `bio.fig` automatiquement.

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

0.2.1 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 graph, si non-spécifiée, les valeurs par défaut sont la densité pour les invertébrés et les poissons, et la couverture (en %) de la catégorie 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

0.2.2 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 optionnel sur n’importe quelle autre variable tant qu’elle soit présente sous forme de colonne dans le jeu de données. Vous pouvez spécifier directement sous l’argument **filtre** ou **filtre2** e.g. si vous voulez juste sélectionner les valeurs de la station “ST1”, vous entreriez: `fig.2var(..., filtre = \ St == ‘ST1’)` (attention à inclure des guillemets simples à l’intérieur des guillemets doubles), ou bien utiliser deux fonctions raccourcis qui font le formatage 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:

```
> fig.2var(..., filtre=filtre.incl("St", "ST1"))
> fig.2var(..., filtre=filtre.excl("Campagne", c("A_2006", "A_2007", "A_2008")))
# c()
```

3. **tous.niveaux**: cette option, à mettre en TRUE (défaut) ou FALSE, détermine si les combinaisons de niveaux non-échantillonnées 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 échantillonnée pendant plusieurs campagne). 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”)

0.3 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
```

Les commandes principales sont décrites ci-bas, suivi d'un astérisque lorsque ce sont des fonctions déjà incluses avec ggplot. Plusieurs commandes additionnelles 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: `...`, à voir aussi les fonctions utilisées par ggplot pour changer l'aspect du texte, `element_text()`.

0.3.1 Titre du graphique

`ggtitle*()`

0.3.2 Limites du graphique en X et Y

`coords_cartesian*()`

0.3.3 Palette de couleurs

0.3.4 Etiquette en absisses

0.3.5 Abréviations pour géomorphologie

J'ai gardé les couleurs par défaut pour l'instant (arc-en-ciel = youpi!) mais on discutera du genre d'options que vous voulez.

Rajouter des options ggplot. Si vous voulez rajouter un titre (ou changer l'étiquette d'un des axes): `fig.2var(...) # plot produit... obj = fig.2var() # mais si j'assigne à un object, pas de plot produit obj + xlab("vavoum!!") # je tape le nom de l'objet dans la console avec "+" d'autres options ggplot, e.g. ici une nouvelle étiquette en X obj + ggtitle("je suis un titre") obj + verti.x.val # pour changer l'orientation des étiquettes, raccourci que j'ai écrit pour vous (donc pas une commande ggplot officielle) obj + no.x.val # ôter les étiquettes en X`

0.4 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 objet formaté de class `ggplot`. Il suffit de spécifier l'argument `filename` en incluant l'extension graphique souhaitée, e.g.

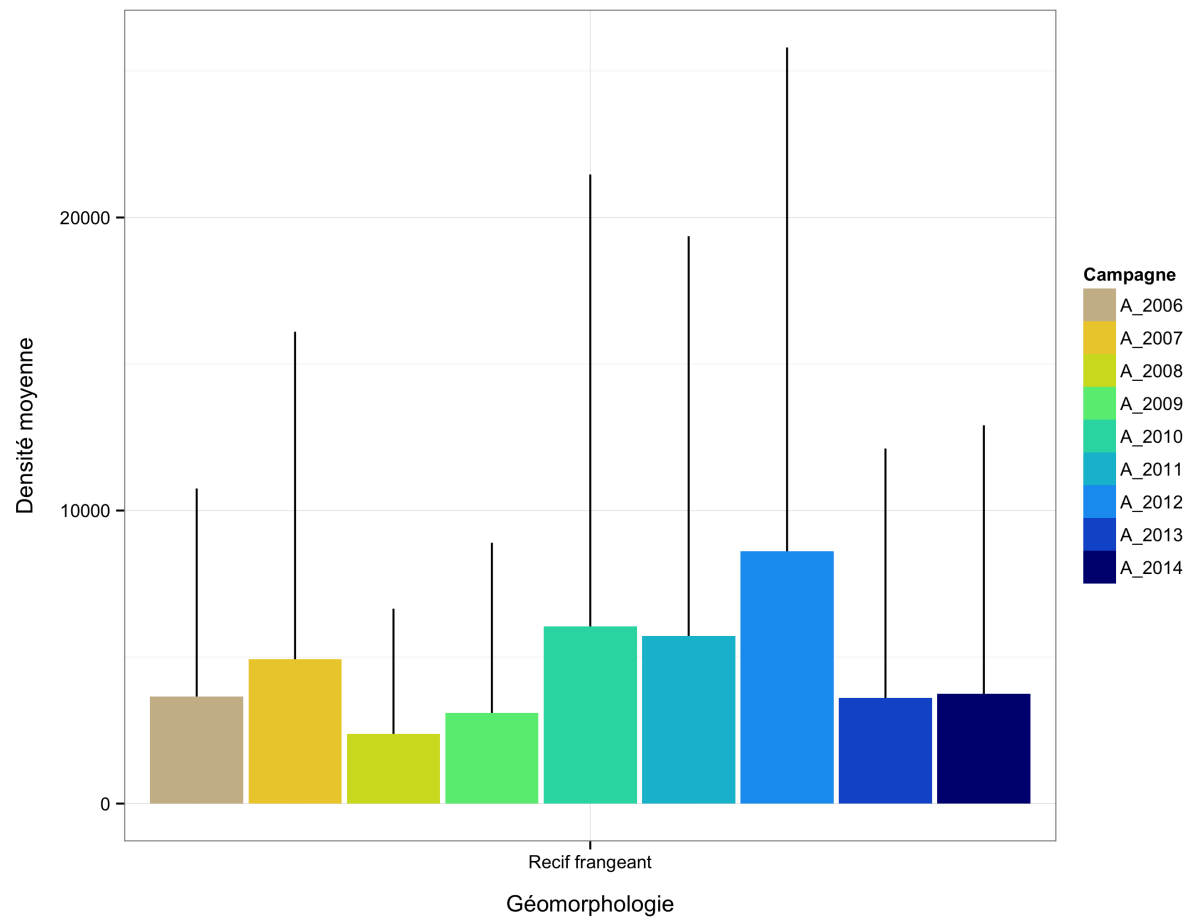
```
ggsave("nom-de-fichier-voulu.png")
ggsave("nom-de-fichier-voulu.pdf")
ggsave("nom-de-fichier-voulu.pdf", gg1) # gg1 est un objet ggplot
```

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

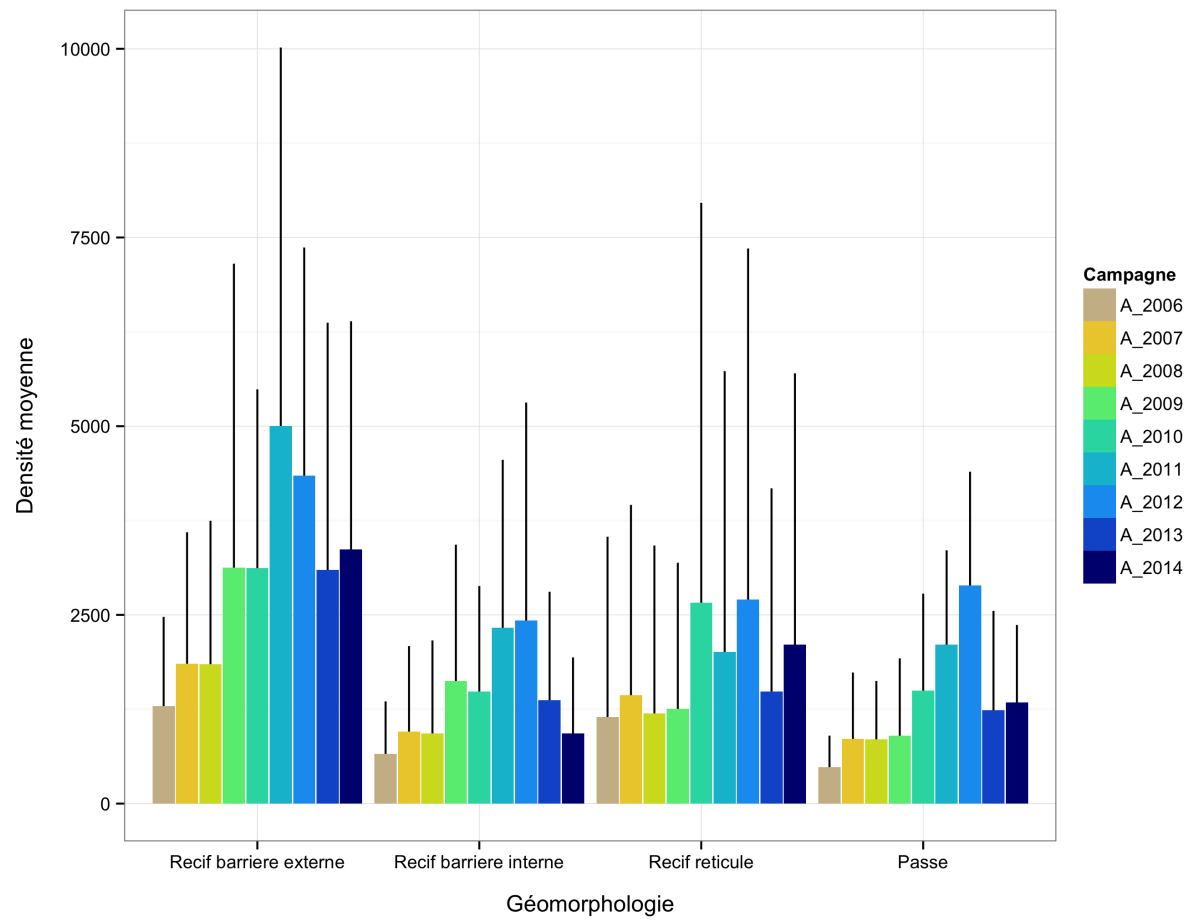
1 Exemples d'utilisation

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.)

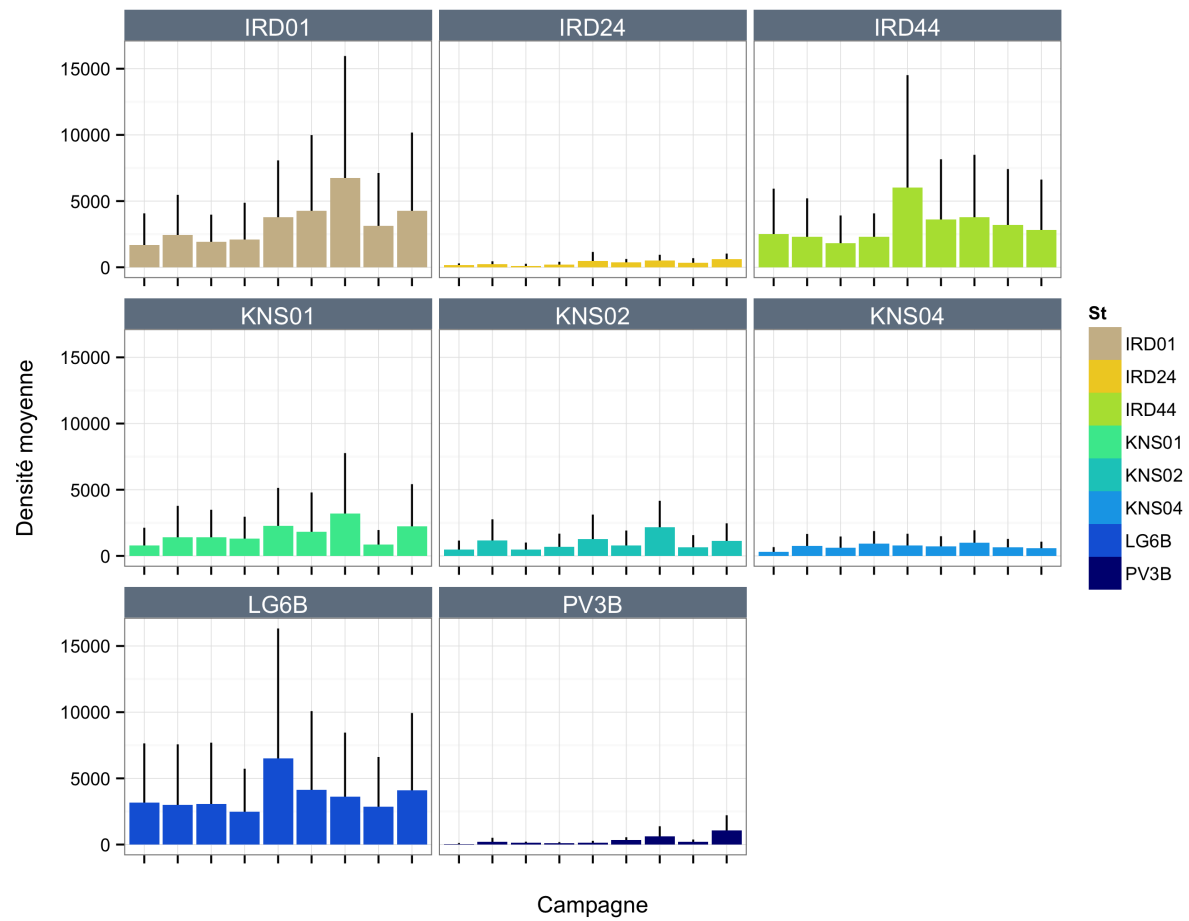
```
inv.fig("Campagne", filtre=filtre.incl("Geomorpho","Recif frangeant"))
```



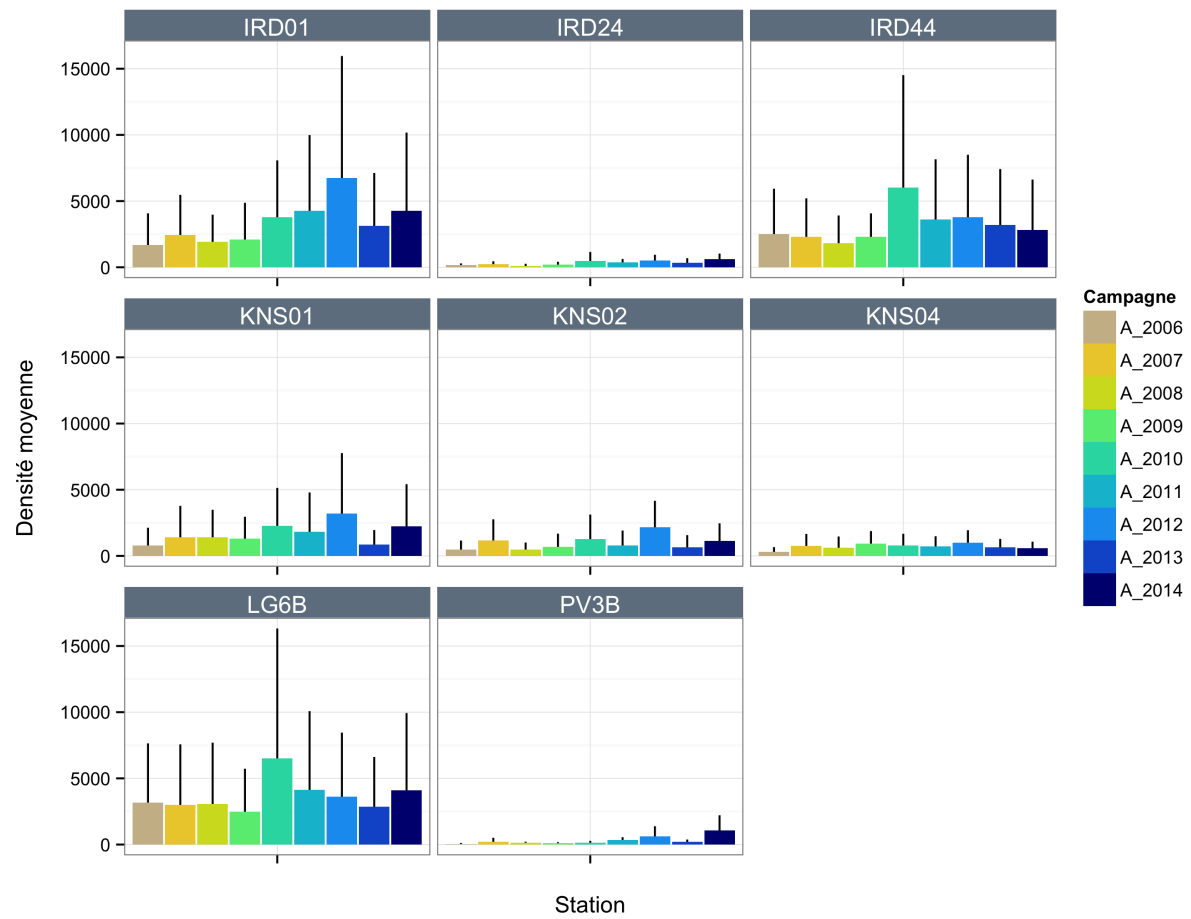
```
inv.fig("Campagne", filtre=filtre.excl("Geomorpho","Recif frangeant"))
```



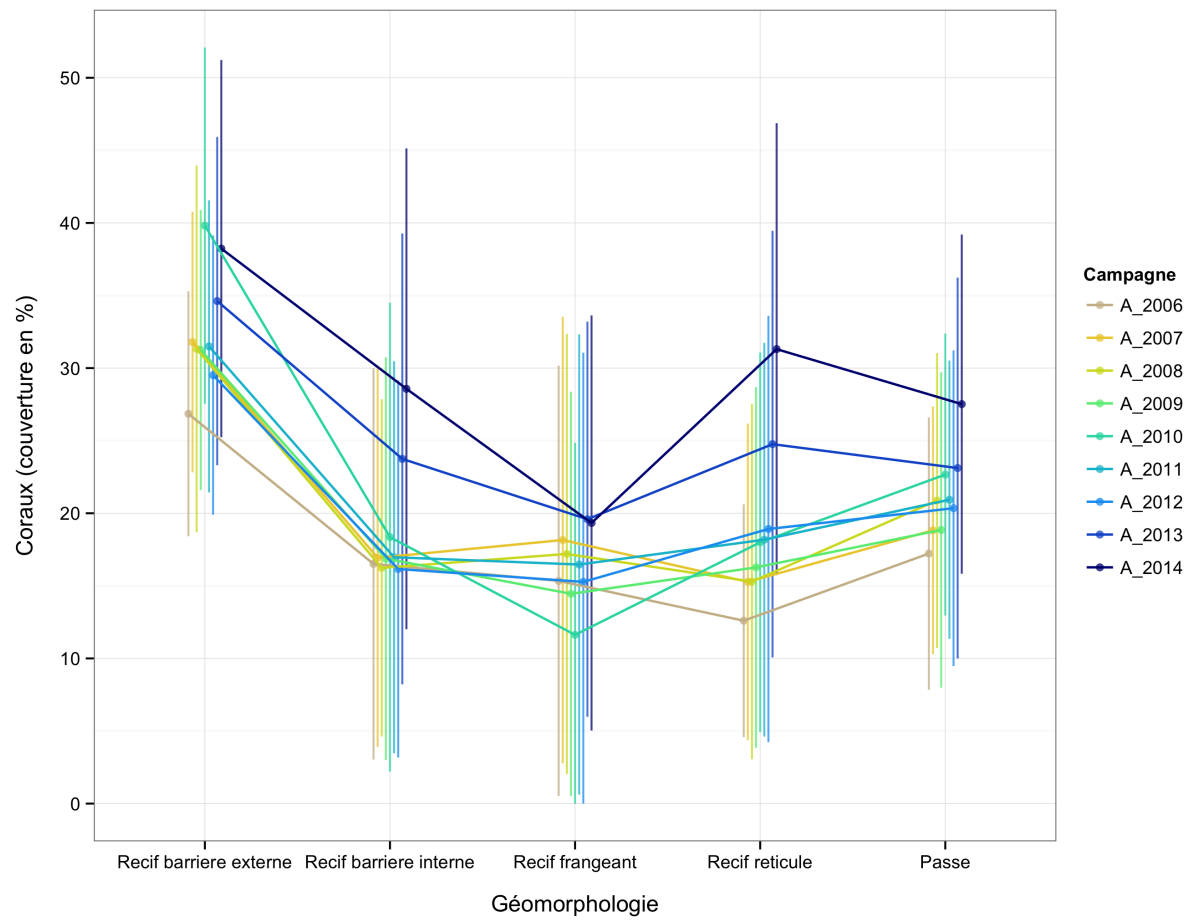
```
inv.fig(var1="Campagne", var2="St", filtre=filtre.incl("Geomorpho","Recif reticule"),
panneau="St") + no.x.val
```



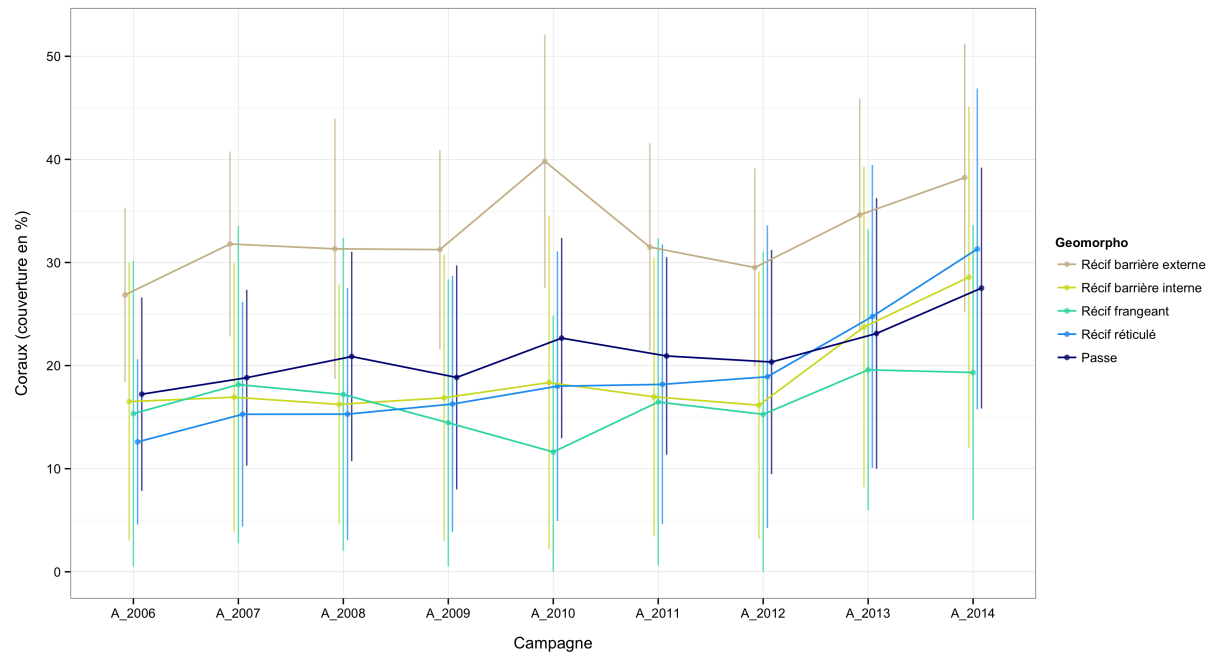
```
inv.fig(var1="Campagne", var2="St", filtre=filtre.incl("Geomorpho","Recif reticule"),
panneau="St") + No.x.val
```



LIT.fig(typ.fig="ligne")



```
LIT.fig(var1="Campagne", var2="Geomorpho", typ.fig="ligne")
```



```
LIT.fig(var.expl="Corail branchu")
```

