



Numéro [numéro 119](#) · [Géovisualisation](#) · Mise en ligne : 01/2017

Géovisualisation des discrétisations : une petite application pédagogique

par : [Laurent Jégou](#)

Mots-clés : [cartographie thématique](#), [géovisualisation](#), [pédagogie](#), [statistiques](#)

⇒ [Cliquer pour consulter et interagir avec la géovisualisation](#)

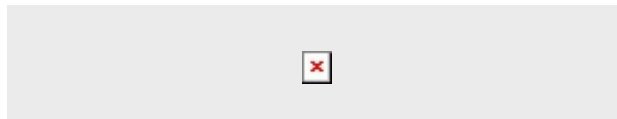


Figure 1. L'application.

Objectifs

L'objectif de cette petite application est de géovisualiser de manière simple les effets du choix d'une méthode de découpage en classes, appelée également discrétisation, pour produire une carte en plage de couleurs (ou choroplèthe). Pour cela, on affiche la carte sur une page HTML interactive, elle se redessine automatiquement en fonction des choix effectués par l'utilisateur.

Cette visualisation interactive, a visée purement pédagogique, cherche à montrer, d'une part, l'effet très important de la méthode choisie sur l'aspect de la carte produite. D'autre part, cette page permet de mieux comprendre le fonctionnement de ces méthodes aujourd'hui automatiques, que l'on retrouve dans la plupart des logiciels de cartographie ou de SIG, en les comparant entre elles et avec un histogramme des fréquences, ce dernier étant réglable dans sa finesse d'analyse.

Description de l'application

L'utilisation de cette petite application est simple : en bas et à droite de la fenêtre, une série de boutons permet de changer la méthode de discrétisation de la carte, et deux curseurs permettent d'agir sur le nombre de classes de la discrétisation et de barres du diagramme de fréquences.



Figure 2. Les commandes.

Par défaut, la carte est présentée selon la méthode de détermination des classes de Jenks, en cinq classes. L'utilisateur peut alors la redessiner selon une autre méthode et changer le nombre de classes, ou comparer l'histogramme résultant avec celui des fréquences. Pour visualiser efficacement les variations présentes dans la distribution des valeurs, la représentation cartographique se doit de regrouper dans une même classe colorée des valeurs proches, ce qui se traduit visuellement par une ressemblance entre les deux histogrammes : les classes doivent traduire assez fidèlement la distribution de la variable, les fréquences.



Figure 3. Comparaison des histogrammes.

La variable statistique choisie pour cette application est caractérisée par une répartition géographique originale et une distribution statistique intéressante à classifier, il s'agit de la part d'étrangers dans les nuitées d'hôtel, par département, en 2015, fournie par [l'INSEE](#).

Méthode et fonctionnement général

Bibliothèques Js utilisées

Comme pour [l'article précédent](#) de cette rubrique, nous avons utilisé les possibilités offertes par la bibliothèque de fonctions JavaScript [D3](#) pour dessiner les parties graphiques de cette page : la carte et sa légende ainsi que les histogrammes.

Deux autres bibliothèques de fonctions ont été mobilisées pour construire cette page :

- [Geostats JS](#), de Simon Georget, pour calculer les bornes de classes selon les méthodes habituelles (à laquelle nous avons ajouté celle des moyennes emboîtées).
- [Colorist JS](#) (composée ad-hoc pour d'autres projets web), pour les conversions de couleurs.

Données

La page utilise deux jeux de données :

- Le fond de carte des départements français, dont la source est le produit ["Géofla" de l'IGN](#), sous licence ouverte.
- Les données statistiques de [l'INSEE](#), par département pour 2015.

Le fond de carte a été généralisé puis converti dans le format [TopoJSON](#) pour une lecture plus rapide par D3. La conversion du format *shapefile* d'origine en TopoJSON a fait appel aux outils de la bibliothèque GDAL-OGR, comme expliqué dans [ce texte de Mike Bostock](#) (auteur principal de D3 et de TopoJSON). Cette étape est importante pour réduire le fond de carte à une taille de fichier adaptée (61kb) et une généralisation agréable de la carte affichée.

Fonctionnement de l'application

Le fonctionnement de l'application est relativement simple : au clic sur un élément actif de l'interface (bouton de méthode ou curseur de nombre de classes), des fonctions sont appelées pour régénérer les représentations graphiques (carte, histogramme) de la page. Le [code source](#) est commenté pour permettre une meilleure lecture ligne à ligne.

Comme toute application web, notre page HTML suit les principes de la *programmation événementielle*, c'est-à-dire que les algorithmes réagissent aux actions de l'utilisateur sur l'interface.

Ainsi, plusieurs fonctions sont définies, appelées avec des valeurs par défaut lors du premier lancement de l'application (l'ouverture de la page HTML dans un navigateur) :



- `discret()` pour le calcul des bornes de classes en fonction de la méthode et du nombre de classes choisies ;
- `color()` pour la détermination de la couleur de chacune des classes ;
- `dessinCarte()` pour le dessin de la carte et des histogrammes ;
- `legende()` pour le dessin de la légende ;
- `carte()`, qui regroupe les deux fonctions précédentes.

En outre, la page charge les données (fond et statistiques) et définit quelques paramètres de mise en page dans le fichier de styles (css).

À chaque clic sur un élément actif de l'interface (boutons, curseurs), la carte est redessinée en tenant compte des nouveaux paramètres. Sur la carte même, une interaction complémentaire est définie au survol du fond de carte par la souris, un petit cadre indique la valeur de la variable dans le département qui a été cliqué.

La fonction **discret()**, à qui on a fourni la variable statistique au moment de son chargement, prend en compte la méthode de discrétisation ainsi que le nombre de classes demandé pour produire un tableau des bornes de classes.

La fonction **color()** va produire un dégradé de couleurs en tenant compte de deux couleurs extrêmes et du nombre de classes voulu. Ce dégradé est calculé en utilisant une variation régulière des trois composantes teinte, saturation et luminosité (TSL).

DessinCarte(), quant à elle, se charge de redessiner la carte au format SVG dans un cadre rectangulaire de la page (une "div" html), en utilisant les informations du fond de carte et les couleurs définies par la fonction `color()` décrite précédemment. Elle prend soin de définir l'évènement de survol du fond par la souris pour déclencher l'affichage d'un petit panneau d'information comportant la valeur de la variable à cet endroit.

Cette fonction redessine ensuite les deux histogrammes, de manière quelque peu artisanale en procédant selon la méthode suivante :

- définition du rectangle dans lequel dessiner l'histogramme ;
- dimensionnement des axes en fonction de l'étendue des valeurs ;
- calcul des fréquences selon le nombre de barres demandé ;
- dessin des barres de fréquences ;
- dessin des axes et de leurs libellés.

D3 fournit des méthodes de calcul de fréquences et de positionnement des barres le long de l'axe des abscisses.

La fonction **legende()**, finalement, dessine les caissons de légende correspondant aux classes, ajoute les libellés de leurs valeurs ainsi que le titre.

Perspectives

Cette application a été utilisée dans des formations à la cartographie thématique de différents niveaux d'études (du L1 au M2), elle a aidé à illustrer visuellement les grandes différences de résultat induites par un simple changement de méthode de discrétisation (et, secondairement, de nombre de classes), comparaison facilitée par la souplesse et la simplicité de l'application. Même si, aujourd'hui, les logiciels de cartographie thématique assistée offrent ces méthodes, c'est au prix de plusieurs manipulations successives, la possibilité de voir les effets de la discrétisation de manière immédiate et sur la même mise en page est plus éloquente. Les étudiants perçoivent mieux l'importance de ce choix dans la construction de leurs cartes et sont plus



attentifs à cette partie, complexe, de l'application de la méthodologie.

Pour aller un peu plus loin, on pourrait envisager différents développements :

- permettre à l'utilisateur d'utiliser sa propre variable statistique ;
- son propre fond de carte ;
- d'entrer lui-même des bornes de classes ;
- d'ajouter de nouvelles méthodes de discrétisation (de *clustering*) ;
- de choisir ses propres couleurs.