

Traitement de données et visualisation

Série 05

Cours : Dr. Gérald Collaud et Dr. Jacques Monnard

Exercices : Elric Milet

- À rendre sur Moodle avant le **lundi 27 mars 2023 à 12h**
- Format : ZIP contenant
 1. un PDF avec vos réponses
 2. un dossier avec votre code



Exercice 1 – QCM

Recommandation : relisez les slides, puis remplissez ce QCM sans les regarder pour vérifier votre compréhension de la théorie.

Vous pouvez aussi justifier votre réponse en cas d'incertitude (facultatif).

1. La fonction D3js `style` (par exemple dans `d3.select("body").style("color", "red")`) va modifier l'attribut CSS de l'élément sélectionné.
 - a. Vrai
 - b. Faux

2. Dans le code JavaScript (utilisant D3js) suivant, sur quel élément définit-on l'attribut `height` ?

```
const svg = d3.select("body").append("svg").attr("width", 500).attr("height", 50);
```

- a. `body`
- b. `svg`
- c. `width`
- d. Aucun des 3

3. Dans le code suivant, que représente `i` ?

```
const dataset = [ 5, 10, 15, 20, 25 ];  
svg.selectAll("circle").data(dataset)  
    .join((enter) => enter.append("circle")  
        .attr("cx", (d, i) => (i * 50) + 25));
```

- a. Un pronom en anglais
- b. L'index de l'élément dans la liste `dataset`
- c. “.” en alphabet morse
- d. L'élément après celui qui est représenté par `d` (par exemple si `d=10`, `i=15`)
- e. Les valeurs de `d` modifiées d'une certaine manière

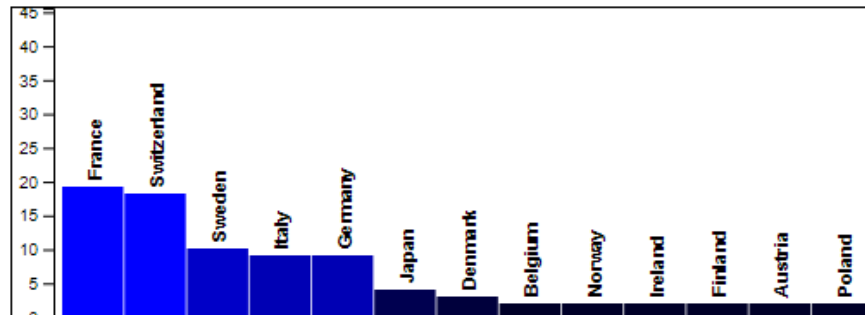
4. Dans le code de la question 3, quelles valeurs va prendre *i* ?
 - a. 0, 1, 2, 3, 4
 - b. 1, 2, 3, 4, 5
 - c. 10, 15, 20, 25, 5
 - d. 6, 11, 16, 21, 26
 - e. Aucune de ces réponses
5. Pour créer un *bar chart* avec la méthode vue en cours, on dessine des rectangles les uns à côté des autres.
 - a. Vrai
 - b. Faux
6. Dans une échelle D3js, *domain* concerne...
 - a. La portée de valeurs que l'on veut obtenir
 - b. La valeur maximum que l'on veut obtenir
 - c. La valeur minimum que l'on veut obtenir
 - d. Les coordonnées couvertes par l'échelle dans le graphique SVG
 - e. Une subdivision d'une zone climatique ou d'une grande aire biogéographique
 - f. Aucune de ces réponses
7. Pour ajouter du *padding* (rembourrage) à une échelle, on définit l'attribut *padding* directement dans la fonction D3js *range*.
 - a. Vrai
 - b. Faux
8. (Bonus) Un *commit* va directement envoyer les modifications effectuées au répertoire distant GitHub.
 - a. Vrai
 - b. Faux
9. (Bonus) Comment s'appelle, en général, la branche principale d'un répertoire Git ? (plusieurs réponses possibles)
 - a. primary
 - b. main
 - c. master
 - d. teacher
 - e. almighty
 - f. bowser

Exercice 2 – Diagramme en barres (bar chart)

Dans cet exercice, vous allez créer un bar chart représentant le nombre d'occurrences de chaque drapeau/pays dans les séries 4 des étudiant.e.s du cours.

Autrement dit, le set de données fourni sur Moodle comprend, pour chaque étudiant.e (dont le nom est anonymisé), les pays qu'ils et elles ont choisi pour le rendu de la série 4, et vous allez en faire une visualisation sous forme de bar chart.

Voici un exemple de résultat de cet exercice (le vôtre peut différer un peu) :



2.1 – Importation des données

D3js possède une fonction permettant d'importer directement des données JSON. Cependant, ici, vous allez devoir faire différemment, afin de pouvoir traiter les données au préalable comme expliqué à l'exercice 2.2.

Pour importer les données JSON, plusieurs méthodes s'offrent à vous. Celle présentée dans cette série est la suivante :

- ✓ 1. Téléchargez les données (students_flags.json) sur Moodle. Pour télécharger directement le fichier plutôt que d'afficher son contenu, faites "enregistrer le lien sous..." et choisissez l'emplacement que vous désirez. Alternativement, vous pouvez copier le texte directement dans un fichier .json vide que vous aurez créé au préalable.
- ✓ 2. Enregistrez ce fichier .json quelque part dans le dossier contenant votre projet, de préférence dans un dossier dédié (que vous pouvez par exemple appeler *data*).
- ✓ 3. Au début de votre fichier JavaScript, ajoutez la ligne suivante. N'oubliez pas de changer le chemin (*path*) en fonction d'où vous avez placé votre fichier .json.


```
import data from './data/students_flags.json' assert {type: 'json'};
```

Note importante : cette méthode risque de ne pas fonctionner sur Firefox. Utilisez plutôt Safari, Edge ou Chrome par exemple pour cet exercice.
- ✓ 4. Créez au moins un fichier HTML et un fichier JavaScript, de la même manière que pour les séries précédentes. N'oubliez pas de relier votre fichier JavaScript au fichier HTML via la balise `<script>`.
- ✓ 5. Définissez votre fichier JavaScript comme étant un module. Pour cela, spécifiez l'attribut *type* dans votre balise `<script>`, comme suit :


```
<script type="module" src="js/barchart_drapeaux.js"></script>
```






- 6 Essayez d'imprimer `data` sur la console (avec `console.log`), pour vérifier que son contenu est le bon et pour vous familiariser avec la structure de l'objet JSON.
- Indication : vous pouvez directement accéder au contenu de l'objet JSON sans avoir besoin de passer par `JSON.parse()`. Par exemple :
- ```
data["semester"] va retourner "Spring".
```





## 2.2 – Préparation des données

En pratique, les données que l'on a à disposition ne sont pas toujours organisées de la manière que l'on veut. Parfois, elles ne sont même pas stockées dans un format facilement manipulable (comme `.json`), ou ne permettent pas d'être enregistrées localement. D'autres fois, elles comportent trop d'erreurs et de problèmes pour pouvoir les utiliser directement. On dit que dans la plupart des projets impliquant du traitement de données, 50% à 80% du temps et de l'effort sont en fait aloués à la préparation des données plutôt qu'à leur traitement ou à leur visualisation.

Dans cet exercice, on vous fournit les données en `.json`, et elles sont prêtes à être utilisées. Cependant, vous allez tout de même devoir fournir un petit travail en amont, avant de pouvoir vous attaquer à la visualisation demandée plus loin.

-  1. Réfléchissez à ce dont vous aurez besoin quand vous réaliserez le diagramme en barres.  
Vraisemblablement, vous aurez besoin d'avoir le nombre d'occurrences de chaque drapeau associé à son nom. Par exemple, 42 occurrences pour "Switzerland".
-  2. Codez une solution intelligente. Pas de hard-coding ! Faites en sorte que votre fonction utilise `data`.  
Par exemple, votre fonction JavaScript pourrait retourner un `array` comprenant chaque drapeau avec son nombre d'occurrences, comme dans l'exemple du paragraphe 1.  
À l'inverse, n'écrivez **pas** en toute lettre dans votre code "Switzerland" : 42
-  3. Une fois que vos données sont prêtes à être utilisées, passez à la question suivante. Si vous ne voyez toujours pas trop quoi faire avec vos données, passez à la question suivante et revenez ici quand vous aurez une idée plus claire de ce dont vous avez besoin.

## 2.3 – Visualisation des données

-  - Il est maintenant temps de créer votre bar chart. Utilisez les données que vous avez préparées à l'exercice 2.2.  
Utilisez D3js dans un fichier externe `.js` plutôt que dans une balise `<script>`.
-  - N'hésitez pas à vous servir des codes d'exemple donnés en cours, savoir copier-coller et changer ce qu'il est nécessaire de changer est un atout.

- ✓ - N'oubliez pas d'importer D3js dans votre fichier HTML de la manière suivante :  

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```
- ✓ - Votre bar chart doit ressembler à quelque chose. N'hésitez pas à utiliser du style, des couleurs etc. si besoin.
- ✓ - Le nom des pays doit être affiché pour chaque barre.  
Vous avez le droit de ne pas afficher les pays les moins représentés par souci de place sur le graphique (par exemple, tous ceux qui n'ont qu'une seule occurrence).  
Optionnellement, vous pouvez aussi afficher la valeur pour chaque barre (donc le nombre d'occurrences).
- ✓ - Pour une meilleure lisibilité, il est recommandé de classer les pays par occurrence (les plus mentionnés à gauche, les moins mentionnés à droite).  
Vous avez probablement intérêt à effectuer cette étape en triant vos données directement à l'endroit où vous les préparez. Vous pouvez par exemple utiliser la fonction [sort\(\)](#) de JavaScript.  
Note : suivant comment vos données sont agencées, l'utilisation de *sort()* que vous devez employer dépasse le cadre de ce cours en termes de complexité. Vous ne serez donc pas pénalisé.e si vos données ne sont pas triées.
- ✓ - La création de votre bar chart doit se faire dans le même fichier JavaScript que votre préparation des données.

## 2.4 – Échelle et axe

- ✓ 1. Créez une échelle *yScale*. Vous pouvez optionnellement utiliser cette échelle dans la création des rectangles qui composent votre bar chart, mais elle vous sera surtout utile pour l'axe.
- ✓ 2. Créez un axe vertical, à afficher sur la gauche de votre diagramme et représentant les valeurs de vos barres. Utilisez l'échelle de y définie précédemment

## ✓ 2.5 – Ajout de la page au site Web

Ajoutez maintenant cette page au site Web que vous avez créé aux séries précédentes. Pensez à inclure la barre de navigation dans votre nouvelle page Web, et à l'actualiser (pour qu'elle comporte la nouvelle page) sur toutes les autres pages également.

Dans le cas où vous n'auriez pas rendu les séries précédentes, vous pouvez vous servir des solutions disponibles sur Moodle.

**Rendez le dossier entier, comprenant tout le site Web (et pas seulement la page des drapeaux).**