

TP 2 – Observable Plot

Visualisation Interactive de Données

Romain Vuillemot – romain.vuillemot@ec-lyon.fr

Objectifs du TP

- Charger un jeu de données et le manipuler avec Observable
- Réaliser des exemples de visualisation avec Observable Plot
- Créer votre propre visualisation à partir d'un prototype papier

Modalités de travail et de rendu :

- En cours, démarrage de l'activité
- Rendu individuel dans une (1) semaine en envoyant le lien vers le notebook Observable
- Possibilité de travail en binôme (mais rendu individuel tout de même)

1 Charger le jeu de données

Votre première tâche sera un jeu de données (disponible ici <https://gitlab.ec-lyon.fr/rvuillem/visu/-/blob/master/dataset/edt-inf-s6-22-23.xlsx>) dans un notebook. Ce jeu de données concerne les emplois du temps d'un semestre de cours d'une matière (divisée en 3 AF) dans une école d'ingénieur. Chaque ligne représente une séance pour un groupe de TD ou un cours :

Jour	Date	Heure	Groupe(s)	Code Cours	Num séance	Num séance
jeu	02/02/2023	08:00 - 10:00	B	22_ING_S06_MATIERE_2	1	1
ven	02/03/2023	08:00 - 10:00	B1a	22_ING_S06_MATIERE_1	1	1
ven	02/03/2023	08:00 - 10:00	B1b	22_ING_S06_MATIERE_1	1	1
ven	02/03/2023	10:15 - 12:15	B1a	22_ING_S06_MATIERE_1	2	2

Pour commencer à manipuler ce fichier, vous devez d'abord créer un notebook Observable et attacher le fichier à ce Notebook (sous forme de fichier Excel ou celui-ci converti en CSV). Ensuite réaliser les premières étapes d'exploration de données :

- Afficher les données sous forme de tableau en Observable
- Quels sont les types d'AF? Les types de séances? Les types de cours? Les types de groupes? La période couverte par le jeu de données?
- Typez les données (par exemple, `d3.timeParse` pour les dates)
- Calculer toute donnée supplémentaire que vous jugerez utile (ou fusion avec données externe), mais cela n'est pas obligatoire à ce stade

Pour rappel voici les étapes générales d'exploration et visualisation d'un jeu de données :

Checklist – Exploration de données

1. **Contexte de travail** problématique et visualisations existants ; utilisateurs ; tâches ; besoins.
2. **Stockage des données** origine des données ; fichier ou base de données ; CSV/JSON/TXT.
3. **Inspection des données brutes** nature des données ; filtrer ; identifier la qualité.
4. **Parser et typer** convertir en types ; formatage ; statistiques.
5. **Visualisation exploratoire** distribution selon l'espace ; temps ; dimensions.
6. **Transformer et dériver les données** somme cumulée ; temps écoulé ; aggregation.
7. **Visualiser** avec graphiques standards et avancés.
8. **Communiquer** tableaux de bord ; rapports ; diffusion aux utilisateurs.

FIGURE 1 – Checklist d'exploration (à adapter à votre jeu de données).

2 Line chart cumulatif

Nous allons désormais réaliser un graphique cumulatif. Ce type de graphique est très utile pour visualiser l'évolution d'une variable au cours du temps. Vous pourrez prendre le nombre de cours cumulé par jour, par semaine, par mois, etc. Voici un exemple de graphique cumulé :

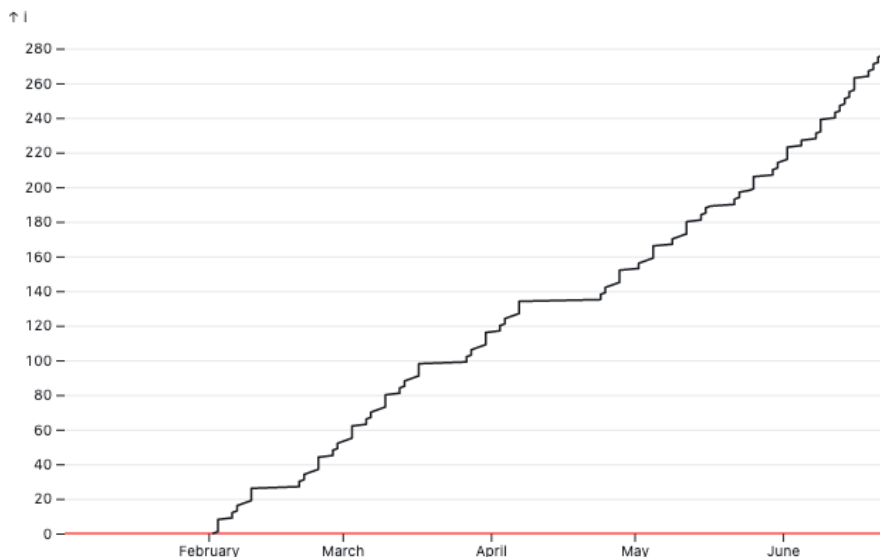


FIGURE 2 – Line chart cumulatif.

- Choisissez une variable à représenter de manière cumulée
- Si celle-ci n'existe pas déjà, calculez là ou créez la
- Rajouter une *facette* (<https://observablehq.com/@observablehq/plot-facets>) en fonction d'une variable (par exemple, le type de cours) comme ci-dessous :

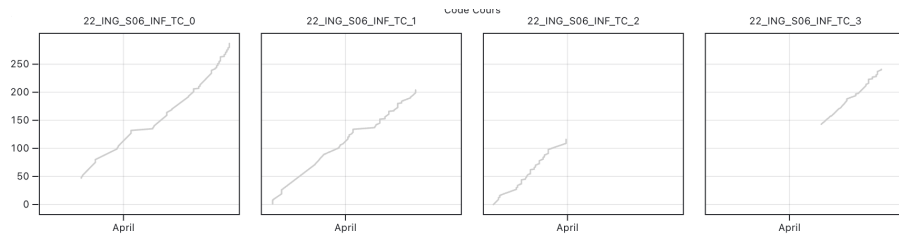


FIGURE 3 – Line chart cumulatif par facette.

3 Représentation séquentiel des événements

Désormais nous souhaitons représenter tous les cours sous forme de marques individuelles afin de réaliser une timeline de *séquence d'événements*. Pour cela nous allons utiliser un graphique de type *scatterplot* où les événements seront encodés sous forme de marques au fil du temps :

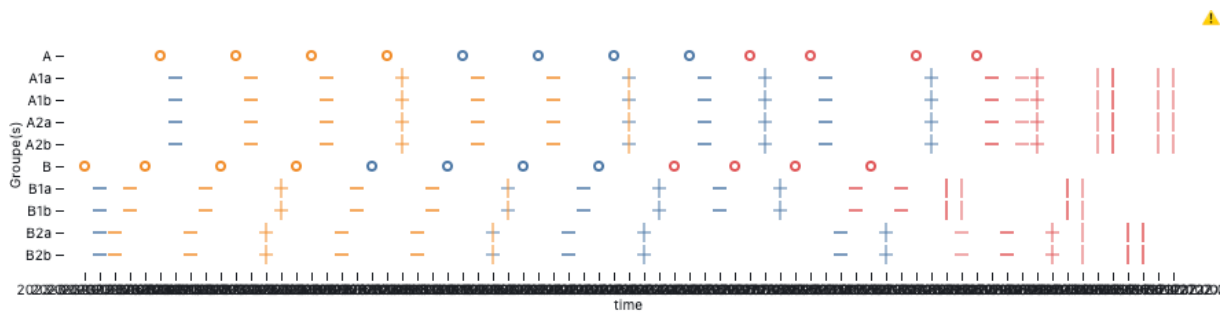


FIGURE 4 – Semestre global.

- Encodez chaque séance de travail sous forme de marque
- Adaptez la marque (forme, couleur) en fonction du type de séance
- Pensez à formater l'axe des axes afin qu'il soit plus lisible
- Rajouter des widgets (<https://observablehq.com/@observablehq/inputs>) afin de permettre le filtrage du graphique (liste de sélection par groupe de TP, check box pour filtrer par cours, etc.)

4 Représentation hebdomadaire des événements

Enfin vous allez représenter le même graphique mais pour une semaine donnée ou pour toutes les semaines et afficher avec une couleur le nombre total de cours par créneau. Pour cela vous réaliserez un graphique de type *heatmap* où chaque TD est représenté par un rectangle (vous pouvez utiliser la marque `tickX` et changer la `strokeWidth` pour obtenir un rectangle) :

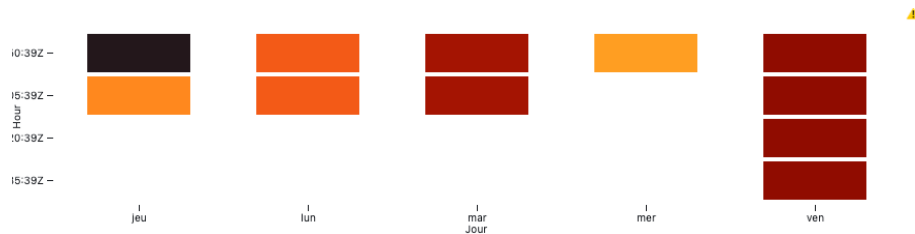


FIGURE 5 – Représentation par semaine.

- Chaque marque représente un créneau horaire par groupe de TD
- L'axe X représente les jours de la semaine
- L'axe Y représente les créneaux horaires
- Le remplissage du rectangle représente le nombre de cours ou le type de cours

5 Rendu : Prototype papier et conception dans Observable Plot

Vous devez rendre ce TP après avoir réalisé les visualisations ci-dessus. Vous incluez également une nouvelle visualisation de votre choix. Vous devez au préalable avoir proposé cette visualisation sous forme de papier un diagramme en précisant : quel est le problème à résoudre, quelles sont les données, quelles sont les variables, quelles sont les marques, quelles sont les propriétés visuelles (couleurs, forme, taille), etc. Vous devez également avoir réalisé cette visualisation dans Observable Plot. Vous devez inclure dans votre rendu :

1. nom, prénom
2. une image de votre prototype papier
3. l'implémentation de votre prototype papier