

Programmation de spécialité (python)

TD 8 : interface avec Dash

Julien Velcin

2023-2024

Partie 1 : Visualisation simple avec `dash_table`

Pour commencer, nous allons mettre en place un serveur Dash qui permet de visualiser un tableau de données. Plusieurs solutions existent, mais nous allons passer par la classe “native” `dash_table`. N’hésitez pas à aller jeter un oeil à la document en ligne : <https://dash.plotly.com/datatable>

1.1 Chargez les données d’utilisation du velib fournies par l’enseignant. Pour cela, vous pouvez utiliser la fonction `read_excel` fournie par la librairie `pandas`. Les différentes tables doivent être chargées dans des tableaux de données de type `DataFrame`.

1.2 Importez Dash et `dash_table` puis créez un objet Dash avec la variable de nom `__name__`. Créez ensuite un *layout* qui contient la table qui liste le nom des stations. La fonction `to_dict` permet de charger directement le tableau. Il est possible que vous soyez obligés de préciser le nom des deux paramètres nommés `data` et `column`.

1.3 Mettez un peu votre tableau en forme à l’aide des styles (<https://dash.plotly.com/datatable/style>). Par exemple, vous pouvez aligner le texte sur la gauche (paramètre `style_cell`), limiter le nombre de données par page (paramètre `page_size`), mettre dans une certaine couleur les cellules sélectionnées par l’utilisateur (en passant par le paramètre `css` cf. <https://community.plotly.com/t/remove-or-change-hotpink-selection-from-datatable/26168/3>), etc.

Partie 2 : Ajout de quelques interactions

L’un des principaux intérêts de Dash consiste à ajouter de l’interaction avec l’utilisateur.

2.1 Pour commencer, ajoutez la possibilité de récupérer l’information contenue dans l’une des cellules du tableau. Pour cela, vous devez a) ajouter un composant graphique (par ex. un “div”) au-dessus ou au-dessous de la table dans l’interface Dash, puis b) ajouter un `callback` prenant en entrée (Input) la cellule active (`active_cell`) et en sortie (Output) l’identifiant du composant que vous avez ajouté. La fonction doit retourner le contenu de la cellule, comme expliqué ici dans la section “Table with click callback” de <https://dash.plotly.com/datatable>

Il sera certainement nécessaire d’aller lire dans le tableau indiqué aux coordonnées récupérées grâce au `callback`.

2.2 Le problème avec l'exercice précédent est que l'interaction ne tient pas (encore) compte de la pagination réalisée pour afficher la table. Il est possible de récupérer les informations nécessaires (numéro de page courante et nombre d'objets par page) à l'aide des champs `page_current` et `page_size` de la table qui devront être ajoutés aux entrées de la fonction de `callback`. Il est certainement nécessaire aussi d'initialiser ces champs lors de la création de la table.

2.3 L'objectif consiste maintenant à afficher un histogramme d'utilisation d'une station lorsqu'on clique dessus. Pour ce faire, il faut commencer par rajouter une figure de type `px.bar` (fonction de `plotly-express`) comme un argument du composant graphique `Graph`, cf. <https://dash.plotly.com/layout>. Il faut ensuite enrichir la fonction de `callback` avec une nouvelle sortie (Output) qui permet de modifier la figure lorsqu'on clique sur une cellule.

2.4 Vous pouvez à présent mettre un peu en forme l'histogramme, par exemple en rajoutant des intitulés plus informatifs aux axes. Si ce n'est pas déjà fait, limitez l'échelle sur l'axe des ordonnées à la plage `[0, 1]`.

2.5 Pour finir, construisez un nouveau tableau qui calcule pour chaque station sa charge totale et si celle-ci est située sur une colline ou non. En utilisant les fonctions de la librairie `pandas` comme `groupeby`, vous pouvez calculer quelques statistiques (moyenne, min, max) d'utilisation des stations en fonction de leur position, puis les visualiser. Pour s'exercer à programmer des `callback`, vous pouvez donner la possibilité à l'utilisateur d'entrer le nom d'une station afin d'obtenir les différentes informations qui la concerne.

Attention, le numéro de la station dans la table des usages est indiqué en première colonne. Le nom de la station correspondante n'est donc pas donnée dans l'ordre dans la table des noms.

Il est possible de rajouter des figures variées avec une interaction, cf. par exemple <https://dash.plotly.com/basic-callbacks>.

Exemple de sortie :

14	EVANGILE
15	HÔPITAL BICHAT
16	PLACE DES TERNES - 5
17	BEAUBOURG
18	PYRÉNÉES-DAGORNO
19	BEAUBOURG PLACE MICHELET
20	MALON (GENTILLY)

