TP2 (version Python)

INF8808: Visualisation de données

Département de génie informatique et génie logiciel

Objectifs

L'objectif de ce travail pratique est de créer un diagramme à bandes empilées interactif à l'aide de données ouvertes en format CSV.

Avant de commencer, nous vous recommandons d'avoir effectué les lectures et exercices suivants:

Lectures :

- Bar Charts

  > https://plotly.com/python/bar-charts/

- Theming and Templates

  > https://plotly.com/python/templates/

- Hover Text and Formatting

  > https://plotly.com/python/hover-text-and-formatting/

- Basic Callbacks

  > https://dash.plotly.com/basic-callbacks

Exercices : Exercices TP2 : 1, 2, 3, 4

Introduction

Un diagramme à bandes est un type de visualisation de données qui représente des données à l'aide de rectangles dont les longueurs sont proportionnelles aux valeurs qu'ils représentent. Un diagramme à bandes empilées étend le diagramme à bandes de base en nous permettant de regrouper les données en une première variable catégorique, puis en utilisant des couleurs pour diviser chaque bande selon une variable

catégorique de deuxième niveau. Un diagramme à bandes empilées est utile lorsque nous sommes intéressés par la taille totale de chaque groupe ainsi que la taille de chaque sous-groupe.

Dans ce travail pratique, vous allez implémenter un diagramme à bandes empilées à l'aide des données de la pièce de Shakespeare, *Roméo et Juliette*. Les données ont été extraites de Kaggle, un site Web populaire au sujet de la science des données [1]. L'ensemble de données a été modifié pour obtenir la version à utiliser dans ce travail pratique. Il contient des données sur texte entier de *Roméo et Juliette*.

Description

Dans ce travail pratique, vous devrez compléter le code Python à l'aide de Plotly et Dash pour créer un diagramme à bandes empilées affichant des données sur le nombre lignes prononcées par chaque joueur dans chaque acte de la pièce.

Pour rendre le graphique interactif, on permettra de basculer entre deux modes d'affichage : « *Count* » et « *Percent* » (correspondant respectivement aux modes « Compte » et « Pourcentage » en français). En mode « *Count* », le graphique affiche le nombre de lignes prononcées par chaque joueur directement. En mode «Pourcentage», la valeur pour chaque joueur est relative et représente plutôt le pourcentage de lignes prononcées par un joueur donné dans un acte donné. Notons que chaque bande empilée contiendra un sous-groupe pour chacun des 5 joueurs qui prononcent le plus de lignes dans la pièce. Les autres joueurs seront regroupés dans leur propre catégorie « *Other* », contenant la somme des lignes prononcées par ces joueurs dans chaque acte et le pourcentage de lignes que cette valeur représente dans l'acte en question.

Pour personnaliser l'apparence du graphique, vous implémenterez également le code pour créer un nouveau thème, ainsi qu'un gabarit pour une info-bulle qui apparaitra lorsque chaque barre est survolée par le curseur.

Les sous-sections suivantes présentent les différentes parties que vous aurez à compléter pour ce travail pratique. Pendant que vous codez, nous vous recommandons de compléter le traitement des données d'abord, suivi de la mise en œuvre du diagramme à bandes lui-même. La création du thème et du gabarit pour l'info-bulle sont indépendant des autres parties. Le basculement entre les deux modes d'affichages peut être mis en oeuvre en dernier.

Structure des fichiers

Pour complèter ce travail, vous devrez remplir les différentes sections `TODO` dans les fichiers de l'archive fournie pour le travail pratique. Les commentaires dans le code expliquent en plus de détails les étapes à suivre. Les scripts à utiliser sont situé dans le répertoire `assets` de l'archive fournie pour le travail pratique.

Dans ce travail pratique, nous vous fournissons, dans l'archive qui vous est transmise, 7 fichiers Python utilisés pour accomplir la visualisation souhaitée :

- `app.py`: ce fichier génère la structure HTML de la page Web et orchestre les étapes requises pour créer la visualisation.
- `bar_chart.py`
- `hover_template.py`

- `modes.py`: ce fichier contient des constantes qui peuvent être utilisées pour aider à gérer les deux modes d'affichage. Vous n'avez pas besoin de modifier celui-ci.
- `preprocess.py`
- `server.py`: Ce fichier est utilisé pour lancer l'application. Vous n'avez pas besoin de modifier celui-ci.
- `template.py`

Données

L'ensemble de données se trouve dans le répertoire `src/assets/data/` dans l'archive fournie pour le travail pratique. L'ensemble de données contient les colonnes suivantes :

- `Act`: Cette colonne représente l'acte dans lequel la ligne est prononcée.
- `Scene`: Cette colonne représente la scène dans laquelle la ligne est prononcée.
- `Line`: Pour une *n*-ième ligne dans une scène donnée, cette colonne contiendra la valeur *n*.
- `Player`: Cette colonne contient le nom du joueur qui a prononcé la ligne.
- `PlayerLine`: Cette colonne contient le contenu prononcé par le joueur pour cette ligne.

Remarque: Comme rappel, le jeu de données représente les données d'une pièce de théâtre, qui sont généralement divisés en gros morceaux appelés « actes ». Dans cette pièce, il y a 5 actes. Puis, chaque acte est divisé en scènes qui contiennent des lignes présentées dans l'ordre dans lequel elles sont prononcées pendant la pièce. Notez également que par « joueur », nous faisons référence à un personnage joué par un acteur.

Prétraitement des données

Pour commencer, vous devrez prétraiter les données que nous vous fournissons. Les données contenues dans le fichier CSV sont brutes, il est donc nécessaire de réorganiser certaines parties de celles-ci afin qu'elles puissent être correctement utilisées par la bibliothèque Plotly. Pour ce faire, vous devez compléter le fichier `preprocess.py`.

Plus précisément, vous devrez effectuer ces étapes:

1. Pour chaque acte, regroupez les lignes prononcées par chaque joueur, en sommant le nombre de lignes prononcées par chaque joueur dans cet acte, en indiquant le pourcentage auquel ce nombre de lignes correspond dans l'acte en question (fonction `summarize_lines`)
2. Modifiez la structure pour inclure une catégorie « *Other* », qui contient la somme du nombre de lignes prononcées dans cet acte par chaque joueur n'appartenant pas aux 5 joueurs ayant le plus de lignes dans la pièce, puis en indiquant aussi le pourcentage de lignes que cette valeur représente dans l'acte en question (fonction `replace_others`)
3. Mettez à jour les noms des joueurs dans les données pour que chaque mot commence par une majuscule suivi de lettres minuscules (fonction `clean_names`)

La figure 1 illustre une partie des données résultantes pour aider à valider votre travail.

Act	Player	LineCount	LinePercent
1	Benvolio	34	14.406780
1	Juliet	16	6.779661
1	Mercutio	11	4.661017
1	Nurse	20	8.474576
1	Other	105	44.491525
1	Romeo	50	21.186441

Figure 1: Extrait des données prétraitées

Diagramme à bandes

Pour cette deuxième partie, vous devrez implémenter la partie principale de la visualisation de données. Tout d'abord, vous devrez afficher les données appropriées dans le diagramme à bandes, selon le mode d'affichage sélectionné. Deuxièmement, vous ajusterez l'étiquette de l'axe y en fonction du mode d'affichage sélectionné. Veuillez voir les commentaires dans le code pour plus de détails sur le texte à afficher. Pour compléter cette partie, vous devrez modifier le fichier `bar_chart.py`.

Voici les étapes que vous devrez accomplir pour cette partie:

1. Complétez la définition de la figure contenue dans le graphique pour inclure le thème et le titre à utiliser (fonction `init_figure`)
2. Affichez les données appropriées dans le diagramme à bandes en fonction du mode d'affichage actuel (fonction `draw`)
3. Affichez le texte approprié sur l'axe des y en fonction du mode d'affichage actuel (fonction `update_y_axis`)

Pour voir le résultat de cette partie, il pourrait être utile de partiellement remplir la fonction `radio_updated` dans `app.py`, qui sera aussi utilisée pour basculer entre les deux modes d'affichage.

Dans la figure 2 et dans la figure 3, nous pouvons voir ce à quoi le diagramme à bandes résultant devrait ressembler dans les deux modes d'affichage « *Count*» et « *Percent* », qui déterminent la portion des données à afficher.

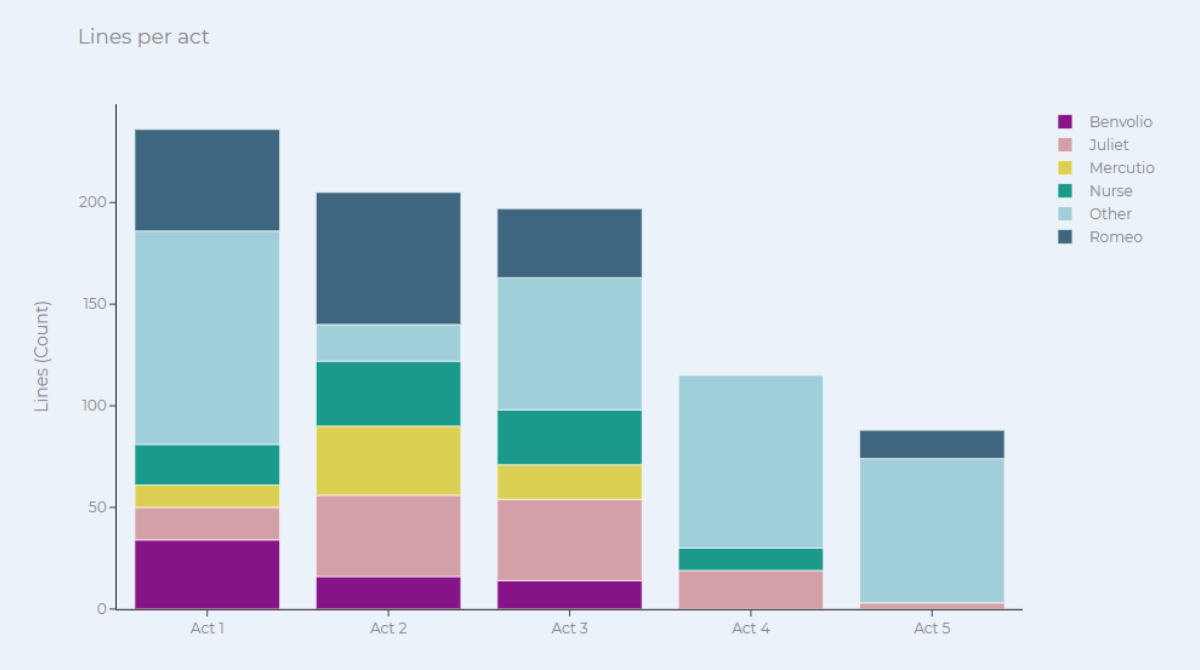

Figure 2: Le diagramme à bandes empilées en mode d'affichage « Count »

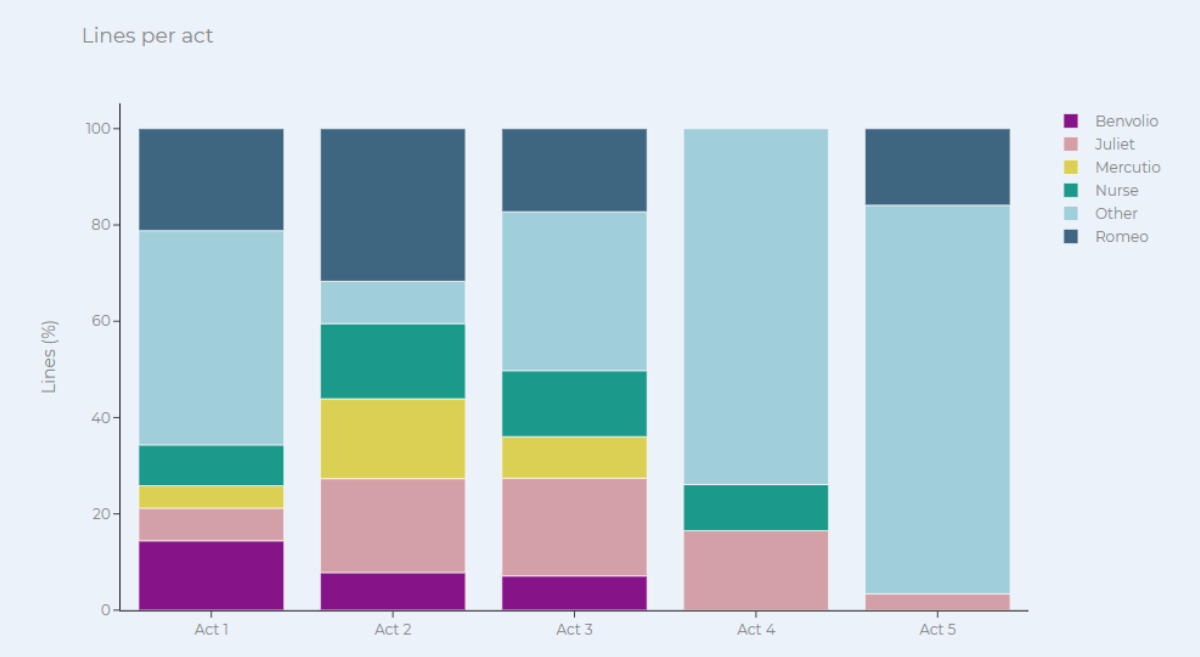

Figure 3: Le diagramme à bandes empilées en mode d'affichage « Percent »

Thème

Pour cette troisième partie, vous allez créer un gabarit personnalisé pour qui déterminera l'affichage visuel du diagramme à bandes. Le code pour cette partie se trouve dans le fichier `template.py`. Plus précisément, vous devrez compléter la fonction `create_template`. Assurez-vous de suivre attentivement les instructions dans les commentaires du code pour bien définir chaque élément du thème. Les valeurs à utiliser sont définies dans une variable au début du fichier. Assurez-vous que les couleurs dans le diagramme à bandes apparaissent dans le même ordre que dans la figure 2 et dans la figure 3, qui illustrent le résultat de l'application du thème.

Info-bulle

Pour cette quatrième partie, vous définirez un gabarit permettant d'afficher une info-bulle qui apparaît lorsque la souris survole une barre. L'info-bulle doit contenir le nom du joueur et le nombre ou le pourcentage de lignes qui y sont associé pendant un acte donné. Les commentaires dans le code donnent une description détaillée du contenu et l'apparence visuelle de l'info-bulle pour les deux modes d'affichage. Veuillez voir la figure 4 pour le résultat final attendu. Le code pour cette partie doit être écrit dans le fichier `hover_template.py` : il se trouve dans la fonction `get_hover_template`.

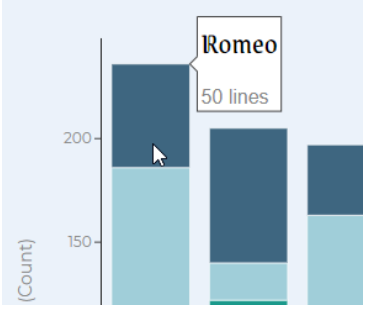

Figure 4: Le diagramme à bandes empilées avec info-bulle en mode d'affichage « Count »

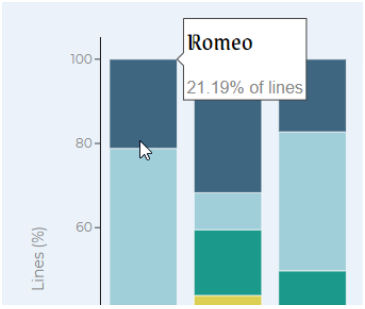

Figure 5: Le diagramme à bandes empilées avec info-bulle en mode d'affichage « Percent »

Basculage du mode d'affichage

Pour cette cinquième partie, vous allez implémenter une méthode pour basculer entre les deux modes d'affichage du graphique en fonction de la valeur courante du bouton radio en bas de la page. Chaque fois que la valeur sélectionnée dans le bouton radio est modifiée, le diagramme à bandes doit être modifié en conséquence. La sous-section des données affichée peut être sélectionnées à partir de la colonne appropriée dans les données, selon le mode d'affichage désiré. L'info-bulle et l'axe y doivent également afficher des textes correspondant au mode sélectionné, tel qu'indiqué dans les commentaires du code. De plus, le texte informatif en bas de la page à gauche doit afficher le mode d'affichage actuellement sélectionné. La figure 5 donne un exemple de l'apparence du pied de page lorsque le mode d'affichage sélectionné est « *Percent* ».

Pour compléter cette partie, vous devrez trouver les fonctions appropriées à appeler dans la fonction `radio_updated` dans `app.py`. Commencer par partiellement remplir cette fonction peut également être utile pour visualiser votre diagramme à bandes pendant que vous effectuez les étapes précédentes.

Figure 6: Les informations affichées dans le pied lorsque le mode d'affichage est « Percent »

Soumission

Les instructions pour la soumission sont:

1. Vous devez placer le code de votre projet dans un fichier ZIP compressé nommé matricule1_matricule2_matricule3.zip.

2. Le travail pratique doit être soumis avant le 6 février 23h59.

Évaluation

Dans l'ensemble, votre travail sera évalué selon la grille suivante. Chaque section sera évaluée sur l'exactitude et la qualité du travail.

Exigence	**Points**
Prétraitement des données	6
Diagramme à bandes	7
Thème	2
Info-bulle	2
Basculage du mode d'affichage	2
Qualité globale et clarté de la soumission	1
Total	**20**

Références

[1] L. Larsen, "Shakespeare plays," Kaggle, [Online]. Available: https://www.kaggle.com/kingburrito666/shakespeare-plays. [Accessed 30 07 2020].