



Créations de graphiques

CRÉATIONS DE GRAPHIQUES À PARTIR DE DONNÉES CSV

Réalisé par : Aiguier Maxime

SOMMAIRE

INTRODUCTION	3
Objectif	3
Langages	3
EX00	4
Création d'un graphique	4
EX01	5
Graphique en temps réel	5
EX02.A	6
Le problème de Monty Hall	6
EX02.B	8
Le problème de Monty Hall	8
EX03	9
Aller plus loin avec matplotlib	9

INTRODUCTION

Objectif

- Apprendre les bases de la data science avec des .csv en Python
- S'entraîner sur différents exercices

Languages

- Python

EX00

Création d'un graphique

Là, on rentre dans le vif du sujet !

Maintenant que vous êtes capable de parser un fichier .csv, il va falloir utiliser ces données !

Il existe plusieurs librairies en python permettant de générer un graphique en python (matplotlib, plotly, ...); vous êtes libres d'utiliser celle de votre choix.



Vous avez toutes les clés pour y arriver !

L'exercice:

Générez un graphique à partir des données contenues dans exemple.csv

EX01

Graphique en temps réel

On rentre dans la partie la plus complexe de ce cours. Maintenant que vous êtes capable d'afficher un graphique à partir de données, il va falloir mettre à jour ce graphique au fur et à mesure que celles-ci évoluent.

L'exercice:

Créez un programme qui affiche les éléments du fichier `test.csv` situé dans le répertoire `ressource`, en écrivant les chiffres en caractères numériques.

Je vous conseille d'utiliser la librairie Matplotlib pour cet exercice !

EX02.A

Le problème de Monty Hall

« Le problème de Monty Hall est connu pour sa solution contre-intuitive, même une fois qu'elle est expliquée.

Imaginez un jeu télévisé où il y a trois portes sur le plateau de jeu. Seule une de ces portes cache un trésor. Il n'y a rien derrière les deux autres portes. Rien ne permet de savoir quelle porte cache le trésor.

La tâche du joueur consiste à choisir parmi les trois portes celle qu'il veut ouvrir. Il aura droit au trésor s'il choisit la bonne porte, et rien sinon. Pour faire son choix, il n'a aucune information. Il doit donc simplement s'en remettre au hasard.

Une fois que le joueur a fait son choix, avant d'ouvrir la porte, le présentateur élimine, parmi les deux portes non choisies, une porte qui ne contient pas de trésor.

Le joueur doit alors faire un nouveau choix. Il peut soit choisir d'ouvrir la première porte qu'il avait choisie, soit changer pour la porte non éliminée par le présentateur. qu'a-t-il intérêt à faire ? »

L'exercice:

Récupérer le dossier src situé dans le répertoire ex02_MontyHall, et faites, à partir des données générées par celui-ci, un graphique d'analyse des résultats

EX02.B

Le problème de Monty Hall

Super, vous avez réussi à créer un graphique, mais est-il vraiment représentatif de la donnée que vous avez récupérée ?

Une compétence essentielle pour tout data scientist est de savoir présenter les résultats dans le bon format.

L'exercice:

Afficher les données selon le graphique le plus adapté possible.

Comment pourriez-vous suivre l'évolution des gains des joueurs en fonction du nombre de parties ?

EX03

Aller plus loin avec matplotlib

Matplotlib a vu le jour pour permettre de générer directement des graphiques à partir de Python. Au fil des années, Matplotlib est devenu une librairie puissante, compatible avec beaucoup de plateformes, et capable de générer des graphiques dans beaucoup de formats différents.

Cette librairie offre de nombreuses possibilités différentes, et c'est intéressant de connaître les différents aspects qui la compose.

Comme vous avez pu le constater, les graphiques générés par la librairie Matplotlib ne sont pas spécialement jolie.

Cela dit, le framework Seaborn fournit une interface qui permet de pallier à ces problèmes. Il utilise toujours Matplotlib, mais le fait en exposant des fonctions plus intuitives.

Réécrivez le code suivant dans votre éditeur de texte, puis exécutez le. Trouvez-vous cela plus beau ? ;)

```
1 import seaborn as sns
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 class Graph:
6
7     def handler(self):
8         sns.set()
9         x = np.linspace(0, 10, 500)
10        y = np.random.randn(500)
11        plt.plot(x,y)
12        plt.show()
13
```

Et ça ne s'arrête pas là !

Seaborn nous fournit aussi des fonctions pour des graphiques utiles pour l'analyse statistique. Par exemple, la fonction `seaborn.distplot()` permet non seulement de visualiser l'histogramme d'un échantillon, mais aussi d'estimer la distribution dont l'échantillon est issu.

Essayer là en remplaçant `plt.plot(x,y)` par `sns.distplot(y, kde=True)` !

Il n'y a pas d'exercice 03, félicitations, vous passez à la suite !