Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université de Carthage Institut National des Sciences Appliquées et de Technologie		Module : Machine Learning Audience : 4ème RT Enseignante : Hazar MLIKI

Atelier 1 : Gestion du graphique & importation des données

Objectifs

- Gestion du graphique avec la bibliothèque *MatPlotLib* pour générer des graphiques sous différentes formes (*pip install* ***matplotlib***).
- Introduire les techniques de base pour importer et accéder aux données sous différentes formes en utilisant la bibliothèque `pandas` (*pip install* ***pandas***).

Partie 1 : Gestion du graphique (Matplotlib)

1- Définition du plot

Les graphiques affichent ce qui a été défini numériquement. Pour générer un graphique (plot), il faut spécifier les valeurs à afficher et le module `matplotlib.pyplot`.

La fonction `plt.plot()` permet de créer un plot en spécifiant les valeurs d'axe *x* et les valeurs d'axe y telles qu'elles apparaissent dans « `values` ».

Exemple

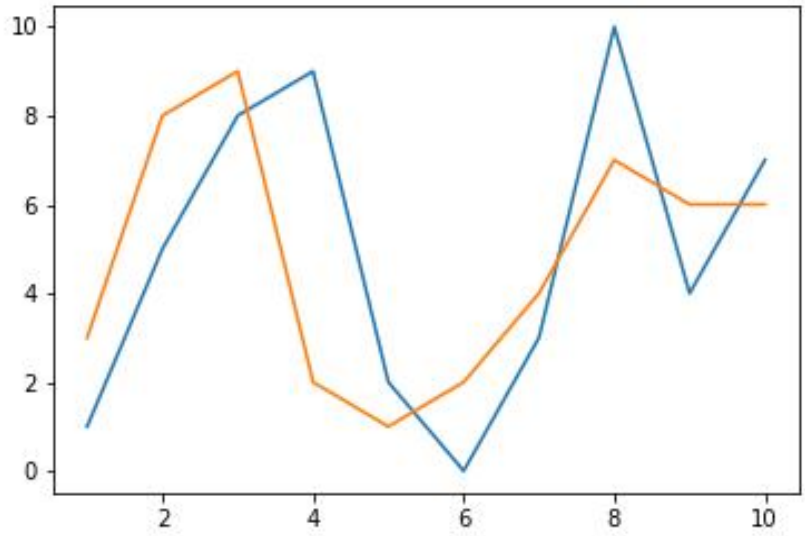
```
values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
import matplotlib.pyplot as plt
plt.plot(range(1,11), values)
plt.show()
```

Pour comparer deux ensembles de valeurs, il faut afficher les deux plots en faisant appel à `plt.plot()` plusieurs fois (pour chaque plot).

Pour sauvegarder le résultat du plot, on utilise la méthode `plt.savefig('nomFigure',format=jpg)` tout en spécifiant le nom et le format (png, pdf, ps, eps, jpg).

Question 1 : Afficher et enregistrer les deux plots suivants :

```
values1 = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
```



2- Définition de l'axe, des tiques et des grilles

- La méthode `plt.axes()` permet de récupérer les axes.
- Les méthodes `set_xlim()` et `set_ylim()` permettent de changer les limites (longueur) des axes des abscisses et des ordonnées.
- Les méthodes `set_xticks()` et `set_yticks()` permettent de changer les ticks d'affichage des données sur les axes des abscisses et des ordonnées.
- La méthode `grid()` permet d'afficher une grille sur le plot.

Exemple

```
values = [0, 5, 8, 9, 2, 0, 3, 10, 4, 7]
import matplotlib.pyplot as plt
ax = plt.axes()
ax.set_xlim([0, 11])
ax.set_ylim([-1, 11])
ax.set_xticks([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
ax.set_yticks([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
ax.grid()
plt.plot(range(1,11), values)
plt.savefig('MySamplePlot.png', format='jpg')
plt.show()
```

3- Modification de l'apparence des lignes

Il est possible de modifier le style et la couleur de la ligne sur le plot, ainsi qu'ajouter des marqueurs sur la ligne.

Styles de lignes en MatPlotLib

Character	*Line Style*
'-'	Solid line
'--'	Dashed line
'-.'	Dash-dot line
':'	Dotted line

Couleurs de lignes en MatPlotLib

Character	*Color*
'b'	Blue
'g'	Green
'r'	Red
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

Marqueurs de Lignes en MatPlotLib

Character	*Marker Type*
'.'	Point
','	Pixel
'o'	Circle
'v'	Triangle 1 down
'^'	Triangle 1 up
'<'	Triangle 1 left
'>'	Triangle 1 right
'1'	Triangle 2 down
'2'	Triangle 2 up
'3'	Triangle 2 left
'4'	Triangle 2 right
's'	Square
'p'	Pentagon
'*'	Star
'h'	Hexagon style 1
'H'	Hexagon style 2
'+'	Plus
'x'	X
'D'	Diamond
'd'	Thin diamond
'|'	Vertical line
'_'	Horizontal line

Exemple

```
values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
import matplotlib.pyplot as plt
plt.plot(range(1,11), values, 'r--o')
plt.plot(range(1,11), values2,'m:v')
plt.show()
```

4-Ajout de labels, annotations et légendes

- Label (étiquette) : il fournit une description de chacun des axes des données. On utilise les méthode plt.xlabel() et plt.ylabel().
- Annotation : elle permet d'attirer l'attention sur les points d'intérêt d'un graphique. Par exemple, pour indiquer qu'un point de données spécifique se trouve en dehors d'une plage particulière de valeurs, on utilise la méthode plt.annotate([x,y], text=« titre »).
- Légende : elle documente les différents éléments d'un plot. Chaque ligne est présentée par une description permettant de la différencier des autres lignes.

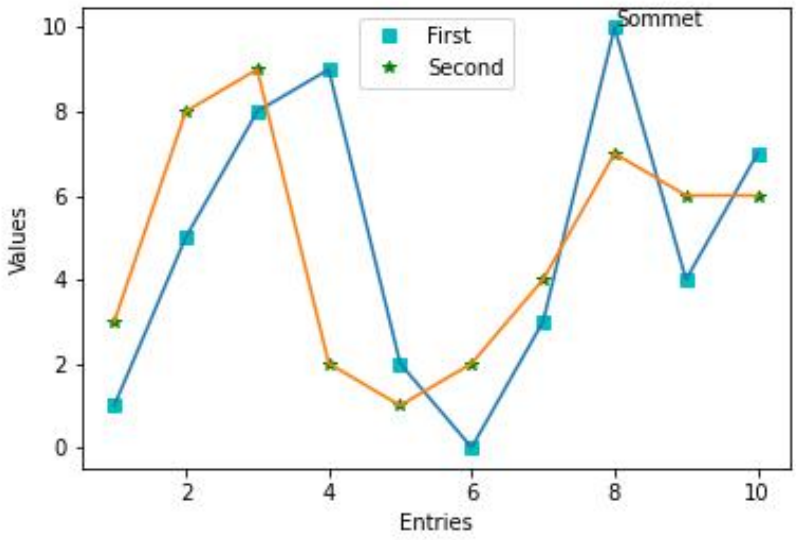
  On utilise la méthode plt.legend() avec le paramètre `loc` pour définir l'emplacement de la légende. plt.legend(['label1', 'label2'], loc=N)

  Le tableau suivant décrit les différentes valeurs N du paramètre « loc ».

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

Question 2 : Réalisez le plot des deux valeurs suivantes :

```
Values1 = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
```



5- Création d'un nuage de points (Scatter plot)

La fonction `scatter()` trace un point pour chaque observation.

L'observation de l'exemple ci-dessus est le résultat du passage de 13 voitures. L'axe X montre l'âge de la voiture et l'axe Y indique la vitesse de la voiture lors de son passage.

Exemple

```
import matplotlib.pyplot as plt
import numpy as np
age = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
vitesse = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(age, vitesse)
plt.show()
```

Question 3 : Comparer les deux observations suivantes. Y a-t-il des relations entre les observations ?

```
#day one, the age and speed of 13 cars:
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
#day two, the age and speed of 15 cars:
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
```

6- Création des graphiques à barres

La fonction `bar()` permet de dessiner des graphiques à barres. Cette fonction prend des arguments qui décrivent la disposition des barres. Les catégories et leurs valeurs sont représentées, respectivement, par le premier et le deuxième argument sous forme de tableaux.

Exemple

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array(["MPI", "RT", "IIA", "IMI"])
y = np.array([16.5, 15.8, 14.9, 13.7])
plt.bar(x,y)
plt.show()
```

7- Création d'un Pie Charts

Le graphique à secteurs (**Pie Charts**) dessine une section pour chaque valeur du tableau (dans l'exemple [35, 25, 25, 15]). Par défaut, le tracé de la première section commence à partir de l'axe des x et se déplace dans le sens inverse des aiguilles d'une montre.

Exemple

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["MPI", "RT", "IMI", "IIA"]
plt.pie(y, labels = mylabels)
plt.legend(title = "Four Fruits:", loc=2)
plt.show()
```

Partie 2 : Importation de données (Pandas)

Pandas est une bibliothèque Python open source. Elle permet de structurer et analyser les données volumineuses et complexes.

1- Création d'un DataFrame Pandas

Un Pandas DataFrame est d'habitude créé en chargeant les données à partir d'une base de données déjà existante (une base de données SQL, un fichier CSV ou un fichier Excel). Dans cet exemple, nous créons un Pandas DataFrame à partir d'une liste et un dictionnaire.

Exemple

```
import pandas as pd
lst = ['GL', 'RT', 'IMI', 'MPI', 'IIA', 'CBA', 'CH', 'BIO']
data = {'Nom':['Ali', 'Mohamed', 'Salah', 'Fatma'],'Age':[20, 21, 19,
18]}
df1 = pd.DataFrame(lst)
print(df1)
df2 = pd.DataFrame(data)
print(df2)
```

2- Gestion des lignes et des colonnes

Un DataFrame est une structure de données bidimensionnelle. Il est possible effectuer des opérations de base sur les lignes/colonnes comme : la sélection, la suppression, l'ajout et le renommage.

Exemple

```
import pandas as pd
data = {'Nom':['Ali', 'Salah', 'Fatma', 'Mohamed'],
        'Age':[27, 24, 22, 32],
        'Addresse':['Sfax', 'Tunis', 'Gabes', 'Sousse'],
        'Filière':['MPI', 'RT', 'GL', 'IMI']}
df = pd.DataFrame(data)
print(df[['Nom', 'Filière']])
```

La méthode `DataFrame.loc[]` permet de récupérer les lignes du DataFrame. Les lignes peuvent aussi être sélectionnées en passant un emplacement entier à la fonction `iloc[].`

Pour sélectionner une colonne particulière, il suffit de l'indexer entre deux corchets.

Exemple

```
import pandas as pd
data = pd.read_csv("Automobile.csv", index_col ="company")
first = data.loc["honda"]
second = data.loc["mazda"]
third = data["price"]
print(first, "\n\n\n", second, "\n\n\n", third)
```

Pour afficher les N premières et dernières lignes de la base de données, on utilise les fonctions `head(N)` et `tail(N)`

Question 4 : Afficher les 5 premières et dernières lignes de la base Automobile.

3- Description des données

La fonction `describe()` permet d'afficher des statistiques relatives à des données.

Exemple

```
import matplotlib.pyplot as plt
import pandas as pd
data = pd.read_csv("Automobile.csv", index_col ="company")
desc = data.describe()
print(desc)
```

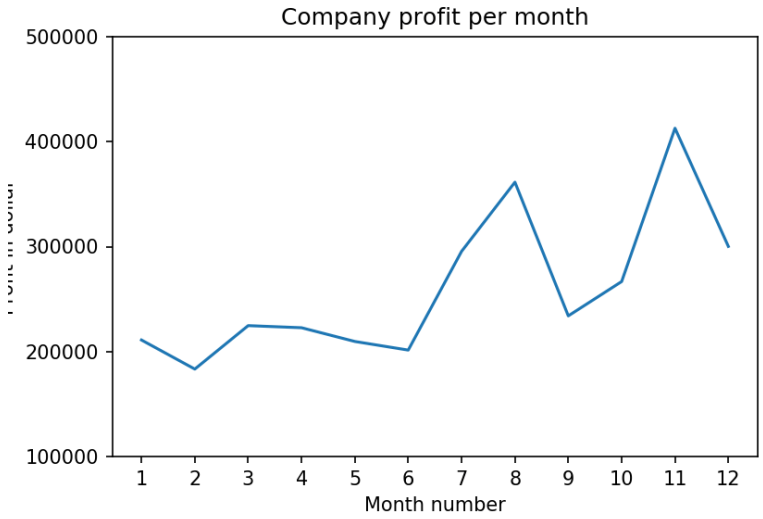
Question 5 : Décrire les statistiques affichées par la méthode describe().

Compte rendu : Utiliser les bibliothèques *Pandas* et *Matplotlib* pour visualiser les données de vente (***salesData.csv***) d'une entreprise.

month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
1	2500	1500	5200	9200	1200	1500	21100	211000
2	2630	1200	5100	6100	2100	1200	18330	183300
3	2140	1340	4550	9550	3550	1340	22470	224700
4	3400	1130	5870	8870	1870	1130	22270	222700
5	3600	1740	4560	7760	1560	1740	20960	209600
6	2760	1555	4890	7490	1890	1555	20140	201400
7	2980	1120	4780	8980	1780	1120	29550	295500
8	3700	1400	5860	9960	2860	1400	36140	361400
9	3540	1780	6100	8100	2100	1780	23400	234000
10	1990	1890	8300	10300	2300	1890	26670	266700
11	2340	2100	7300	13300	2400	2100	41280	412800
12	2900	1760	7400	14400	1800	1760	30020	300200

1- Lire le bénéfice total (total_profit) de tous les mois et afficher-le à l'aide d'un plot linéaire, avec :

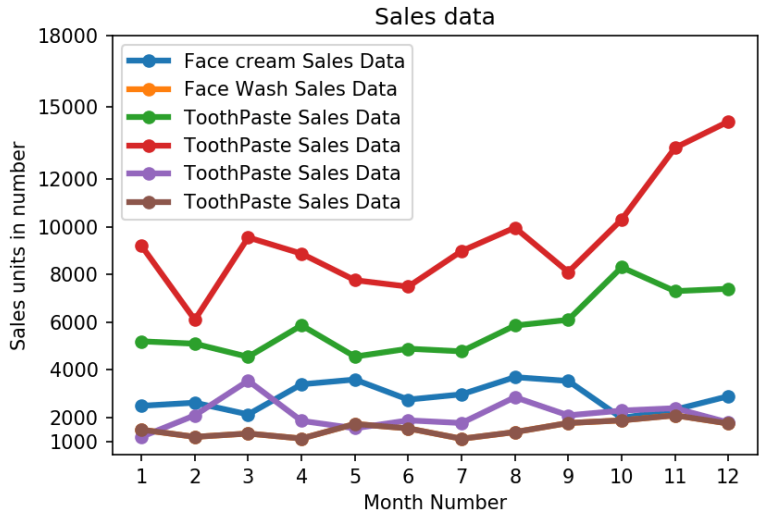
- `X label name = month_number`
- `Y label name = total_profit`



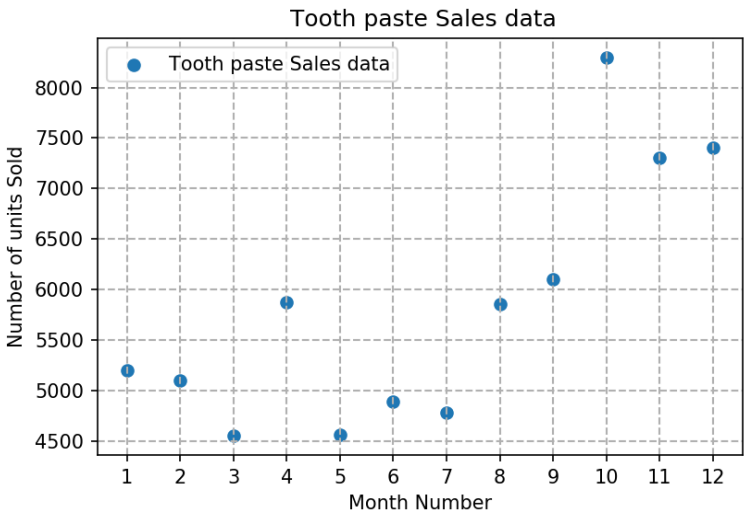
2- Récupérer le bénéfice total de tous les mois et afficher un plot linéaire avec les propriétés de style suivantes :

- Le style de la ligne est "dotted" et sa couleur est rouge.
- Afficher la légende « Company Sales data of last year » à droite en bas.
- X label name = Month Number
- Y label name = Sold units number
- Ajouter des marques circulaires.
- La couleur du marqueur de ligne est rouge
- La largeur de la ligne est 3

3- Affichez toutes les données de ventes de produits (sale data) en utilisant un plot multiligne.



4- Affichez les données de ventes de dentifrice (toothpaste sales data) de chaque mois à l'aide d'un nuage de points (scatter plot).



5- Calculez les données de vente totales de l'année dernière pour chaque produit et affichez-les à l'aide d'un graphique à secteurs (pit chart). Le Nombre d'unités vendues par an pour chaque produit est affiché en pourcentage.