

Auteur : M. Zouari Lazhar

Analyse des données



Importation de la bibliothèque pandas

```
In [1]: from pandas import *
# ou bien import pandas
# ou bien import pandas as alias
```

Création d'un dataframe

Importation des fichiers txt et csv

`df = pandas.read_csv('Chemin\Nom_Fichier.extension', sep = 'séparateur', index_col = N°_Colonne, header = 0 , encoding='Code_codage')`

Exemple :

```
In [2]: from pandas import *
df = read_csv('pays.csv', sep=';')
```

Importation des fichiers Excel

Syntaxe : `df = pandas.read_excel ("Chemin\Nom_Fichier.xls", "Nom_Feuille" , index_col = N°_Colonne , header = 0)`

Exemple :

```
In [3]: from pandas import *  
df = read_excel("pays.xlsx", "pays")
```

Manipulation d'un dataframe

Afficher le contenu d'un dataframe

Syntaxe : print(Nom_dataframe)

Exemple :

```
In [ ]: from pandas import *  
df = read_excel("pays.xlsx", "pays")  
print(df)
```

Activité :

Modifier le code suivant afin d'afficher le contenu de la deuxième feuille intitulé **Min_MAX** du fichier **pays.xlsx**

```
In [ ]: from pandas import *  
df = read_excel("Pays.xlsx", "Min_Max")  
print(df)
```

Afficher le contenu d'une colonne

Syntaxe : print(Nom_dataframe.Nom_colonne)

Exemple :

```
In [ ]: from pandas import *  
df = read_excel("pays.xlsx", "pays")  
print(df.Superficie) #print(df['Superficie'])
```

Afficher le contenu de plusieurs colonnes

Syntaxe : print(Nom_dataframe[[Nom_col1, Nom_col1, ...]])

Exemple :

```
In [4]: from pandas import *  
df = read_excel("pays.xlsx", "pays")  
print(df[['Pays', 'Habitant']])
```

	Pays	Habitant
0	Russ	146781095
1	Canada	37560207
2	USA	331883986
3	Chine	1394112547
4	Brésil	210301591
5	Australie	25105503

Activité :

Compléter le code suivant afin d'afficher le contenu d'une colonne donnée de la feuille **Pays** du fichier **pays.xlsx**

```
In [5]: from pandas import *
df = read_excel("pays.xlsx", "pays")
ch = input('nkjlk1k')
print(df[ch])
```

```
-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-5-fd2114ca9ba2> in <module>
      1 from pandas import *
      2 df = read_excel("pays.xlsx", "pays")
----> 3 ch = input('nkjlk1k')
      4 print(df[ch])

c:\users\kaout\appdata\local\programs\python\python37-32\lib\site-packages\ip
ykernel\kernelbase.py in raw_input(self, prompt)
    861         self._parent_ident,
    862         self._parent_header,
--> 863         password=False,
    864     )
    865

c:\users\kaout\appdata\local\programs\python\python37-32\lib\site-packages\ip
ykernel\kernelbase.py in _input_request(self, prompt, ident, parent, passwor
d)
    902         except KeyboardInterrupt:
    903             # re-raise KeyboardInterrupt, to truncate traceback
--> 904             raise KeyboardInterrupt("Interrupted by user") from N
one
    905         except Exception as e:
    906             self.log.warning("Invalid Message:", exc_info=True)

KeyboardInterrupt: Interrupted by user
```

Afficher les données des lignes de la position initiale et la position finale - 1

Syntaxe : `print(Nom_dataframe.iloc[pi:pf])`

Exemple :

```
In [7]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df.iloc[1:3])
```

	Pays	Superficie	Habitant
1	Canada	998467	37560207
2	USA	9826675	331883986

Activité 1 :

Compléter le code suivant afin d'afficher la première ligne de la feuille **Pays** du fichier **pays.xlsx**

```
In [11]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df.iloc[0])
```

```
Pays          Russ
Superficie    17098242
Habitant      146781095
Name: 0, dtype: object
```

Activité 2:

Compléter le code suivant afin d'afficher le contenu d'une colonne donnée de la feuille **Pays** du fichier **pays.xlsx**

```
In [11]: from pandas import *
df = read_excel("pays.xlsx", "pays")
```

Afficher les N premières lignes d'une dataframe

Syntaxe : print(Nom_dataframe.head(N))

Exemple :

```
In [13]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df.head(3))
```

```
      Pays  Superficie  Habitant
0    Russ    17098242  146781095
1  Canada     998467   37560207
2    USA     9826675  331883986
```

Afficher les N dernières lignes d'une dataframe

Syntaxe : dataframe[dataframe ["nom colonne"] opérateur_comparaison valeur]

Exemple :

```
In [3]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df.tail(2))
```

```
      Pays  Superficie  Habitant
4  Brésil    8514877  210301591
5  Australie    774122  25105503
```

Affichage des données d'un dataframe selon une condition

Syntaxe : dataframe[dataframe ["nom colonne"] opérateur_comparaison valeur]

Exemple 1 :

```
In [15]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print (df[df["Superficie"] > 9830000])
```

Empty DataFrame
Columns: [Pays, Superficie, Habitant]
Index: []

Exemple 2 :

```
In [15]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df[df['Superficie'].isin([9826675,17098242])])
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
2	USA	9826675	331883986

Activité :

Compléter le code suivant afin d'afficher la superficie et le nombre d'habitants d'un pays donnée a partir de la feuille **Pays** du fichier **pays.xlsx**

```
In [16]: from pandas import *
df = read_excel("pays.xlsx", "pays")
```

Affichage des données d'un dataframe selon plusieurs conditions

Syntaxe : dataframe[(dataframe ["Nom_Colonne"] Opérateur_Comparaison Valeur)

Opérateur_Logique (dataframe ["Nom_Colonne"] Opérateur_Comparaison Valeur)]

Exemple :

```
In [17]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print (df[(df["Superficie"] >=9826675) | (df["Superficie"] >=1000000)])
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
2	USA	9826675	331883986
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591

Trier les données d'un dataframe

Syntaxe : dataframe = dataframe.sort_values(by = [Liste des colonnes], ascending = [critère pour chaque colonne (True/False)])

Exemple :

```
In [18]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print(df.sort_values(by=['Pays'], ascending = False))
```

	Pays	Superficie	Habitant
2	USA	9826675	331883986
0	Russ	17098242	146781095
3	Chine	9596961	1394112547
1	Canada	998467	37560207
4	Brésil	8514877	210301591
5	Australie	774122	25105503

Activité :

Compléter le code suivant afin d'afficher le nom du pays le plus peuplé de la feuille **Pays** du fichier **pays.xlsx**

```
In [19]: from pandas import *
df = read_excel("pays.xlsx", "pays")
```

Renommer les libelles d'une colonne ou plusieurs colonnes

Syntaxe : df.rename(columns={"Colonne1": "Nouveau_Colonne1", "Colonne2": "Nouveau_Colonne2"}, inplace=True/False)

Exemple :

```
In [20]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df.rename(columns={"Pays": 'Nom_Pays'}, inplace=True)
print(df)
```

	Nom_Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
2	USA	9826675	331883986
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591
5	Australie	774122	25105503

Modifier le contenu d'une cellule dans un dataframe

Syntaxe : Dataframe.loc[N° ligne, "Nom_colonne"]=Valeur

Exemple :

```
In [21]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df.loc[0, "Pays"] = "Russie"
print (df)
```

	Pays	Superficie	Habitant
0	Russie	17098242	146781095
1	Canada	998467	37560207
2	USA	9826675	331883986
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591
5	Australie	774122	25105503

Ajouter des données dans un dataframe

Syntaxe : Nouveau_df= pandas.DataFrame([[La liste des valeurs]], columns=[La liste des colonnes])

dataframe= dataframe.append(Nouveau_dataframe ,sort=False, ignore_index=True/False)

Exemple :

```
In [22]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df1=DataFrame([["Tunisie",162155,11582075]],columns=["Pays","Superficie","Habitant"])
df=df.append(df1,sort=False,ignore_index=True)
print (df)
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
2	USA	9826675	331883986
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591
5	Australie	774122	25105503
6	Tunisie	162155	11582075

Activité :

Compléter le code suivant afin ajouter les informations concernant **la Tunisie** avec une superficie de **162155** et un nombre d'habitants **=11582075** à la **4ème position** dans la feuille **Pays** du fichier **pays.xlsx**

```
In [23]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df0=df.iloc[0:3]
print (df)
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
2	USA	9826675	331883986
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591
5	Australie	774122	25105503

Ajouter une colonne dans un dataframe

Syntaxe : Nom_Dataframe["Nom_Colonne"]=valeur

Exemple :

```
In [24]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df["Densité"]=df.Habitant/df.Superficie
print(df)
```

	Pays	Superficie	Habitant	Densité
0	Russ	17098242	146781095	8.584572
1	Canada	998467	37560207	37.617875
2	USA	9826675	331883986	33.773783
3	Chine	9596961	1394112547	145.266043
4	Brésil	8514877	210301591	24.698136
5	Australie	774122	25105503	32.430939

Supprimer une colonne

Syntaxe : del (Nom_dataframe["Nom_Colonne"])

Exemple :

```
In [25]: from pandas import *
df = read_excel("pays.xlsx", "pays")
del (df["Pays"])
print(df)
```

	Superficie	Habitant
0	17098242	146781095
1	998467	37560207
2	9826675	331883986
3	9596961	1394112547
4	8514877	210301591
5	774122	25105503

Supprimer une ligne

Syntaxe : dataframe = dataframe.drop([indice_ligne], inplace=(True / False))

Exemple :

```
In [26]: from pandas import *
df1 = read_excel("pays.xlsx", "pays")
df2 = df1.drop(2)
print(df2)
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
3	Chine	9596961	1394112547
4	Brésil	8514877	210301591
5	Australie	774122	25105503

Supprimer plusieurs lignes consécutives

Syntaxe : dataframe = dataframe.drop(dataframe.index[pi:pf], inplace=(True / False))

Exemple :

```
In [27]: from pandas import *
df = read_excel("pays.xlsx", "pays")
df.drop(df.index[2:5], inplace = True)
print(df)
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
5	Australie	774122	25105503

Supprimer les doublons

Syntaxe : dataframe = dataframe.drop_duplicates()

Exemple :

```
In [28]: from pandas import *
df = read_csv("pays03.csv", sep=";")
df=df.drop_duplicates()
print (df)
```

	Pays	Superficie	Habitant
0	Russ	17098242	146781095
1	Canada	998467	37560207
2	USA	9826675	331883986
3	Tunisie	162155	11582075
4	Chine	9596961	1394112547
5	Brésil	8514877	210301591
6	Australie	774122	25105503

Supprimer les lignes avec des données manquantes

Syntaxe : df.drop(df.index[df["nom_col"].isnull()], inplace=(True / False))

Exemple :

```
In [19]: from pandas import *
df = read_csv("pays02.csv", sep=";")
print("dataframe avant exécution du code : \n", df)
df.drop(df.index[df.isnull()], inplace=True)
print("Affichage après exécution du code : \n", df)
```

```
dataframe avant exécution du code :
      Pays  Superficie  Habitant
0  Nigéria      923768  212871345.0
1  Algérie      2381741  43003767.0
2  Angleterre      130395         NaN
```

IndexError Traceback (most recent call last)

```
<ipython-input-19-15153cc65c3d> in <module>
      2 df = read_csv("pays02.csv", sep=";")
      3 print("dataframe avant exécution du code : \n", df)
----> 4 df.drop(df.index[df.isnull()], inplace=True)
      5 print("Affichage après exécution du code : \n", df)
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\range.py in getitem__(self, key)

```
    684         )
    685         # fall back to Int64Index
--> 686         return super().__getitem__(key)
    687
    688     def __floordiv__(self, other):
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in getitem__(self, key)

```
    4289
    4290         key = com.values_from_object(key)
-> 4291         result = getitem(key)
    4292         if not is_scalar(result):
    4293             return promote(result)
```

IndexError: too many indices for array

Ajouter une colonne remplie par des données suite à une condition simple

Syntaxe : `df["Nom_Colonne_Resultat"] = numpy.where(df["Nom_Colonne"]
opérateur_comparaison valeur, Valeur1_Si_vrai, Valeur2_si_Faux)`

Exemple :

```
In [30]: from pandas import *
from numpy import where
df = read_excel("pays.xlsx", "pays")
df["Densité"]=(df.Habitant/df.Superficie)
df["Observation"] = where(df["Densité"]>60, "Très dense", "Moyenne")
print(df)
```

	Pays	Superficie	Habitant	Densité	Observation
0	Russ	17098242	146781095	8.584572	Moyenne
1	Canada	998467	37560207	37.617875	Moyenne
2	USA	9826675	331883986	33.773783	Moyenne
3	Chine	9596961	1394112547	145.266043	Très dense
4	Brésil	8514877	210301591	24.698136	Moyenne
5	Australie	774122	25105503	32.430939	Moyenne

Activité :

Compléter le code suivant afin d'ajouter et remplir la colonne **observation** dans la feuille **Pays** du fichier **pays.xlsx**, sachant que :

- Observation = Très dense dans le cas où densité ≥ 100
- Observation = Normale dans le cas où $30 \leq \text{densité} < 100$
- Observation = Faible dans le cas où densité < 30

```
In [11]: from pandas import *
from numpy import where
df = read_excel("pays.xlsx", "pays")
df["Densité"]=(df.Habitant/df.Superficie)
df["Observation"] = where(df["Densité"]>60, "Très dense", "Moyenne")
print(df)
```

	Pays	Superficie	Habitant	Densité	Observation
0	Russ	17098242	146781095	8.584572	Moyenne
1	Canada	998467	37560207	37.617875	Moyenne
2	USA	9826675	331883986	33.773783	Moyenne
3	Chine	9596961	1394112547	145.266043	Très dense
4	Brésil	8514877	210301591	24.698136	Moyenne
5	Australie	774122	25105503	32.430939	Moyenne

Les fonctions statistiques sur les dataframes



Exemple :

```
In [32]: from pandas import *
df = read_excel("pays.xlsx", "pays")
print('Nombre des pays :', len(df))
```

Nombre des pays : 6

Présentation des données à travers des graphiques

Importation de la bibliothèque pandas

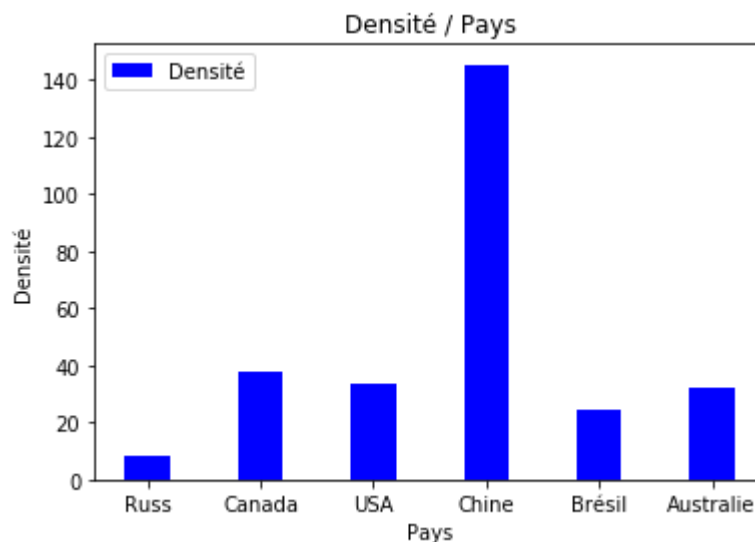
```
In [33]: from matplotlib.pyplot import *  
#Ou bien import matplotlib.pyplot as alias  
#ou bien import matplotlib.pyplot
```

Créer un graphique à barres

Syntaxe : `pyplot.bar ([Position1 , Position2 , PositionN] , [Valeur1 , Valeur2 , ValeurN] , align="Position" , width =Taille , color="Couleur/Code_Couleur")`

Exemple 1:

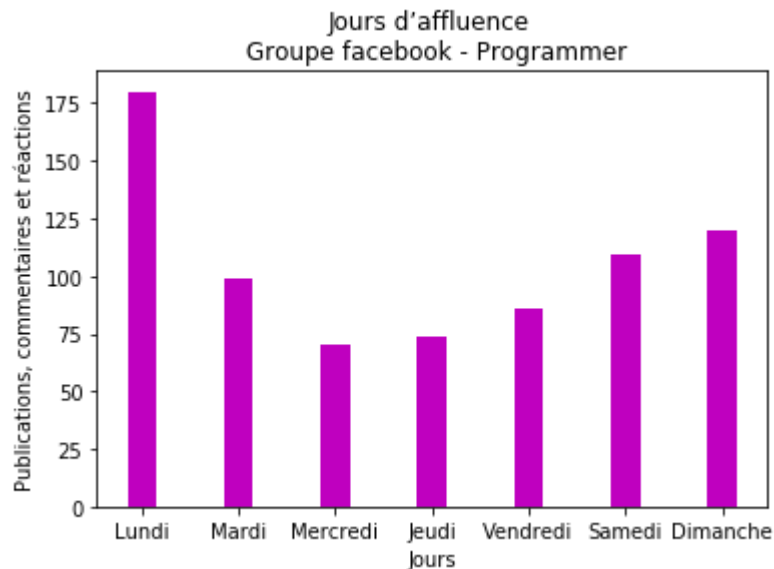
```
In [203]: from pandas import *  
from matplotlib.pyplot import *  
df = read_excel("pays.xlsx","pays")  
df["Densité"]=df.Habitant/df.Superficie  
bar(df.Pays,df["Densité"],width =0.4, color='b')  
xlabel('Pays')  
ylabel('Densité')  
title('Densité / Pays')  
legend(['Densité'], loc='upper left')  
show()
```



Exemple 2:

Statistiques du groupe Facebook Programmer :
Jours d'affluence

```
In [217]: from pandas import *
from matplotlib.pyplot import *
df = read_excel("c:\Analyse\Facebook_Group.xlsx", "Jours d'affluence")
bar(df['Jours d'affluence'], df["Publications, commentaires et réactions"], width=0.3, color='m')
xlabel('Jours')
ylabel('Publications, commentaires et réactions')
title('Jours d'affluence \n Groupe facebook - Programmer')
show()
```

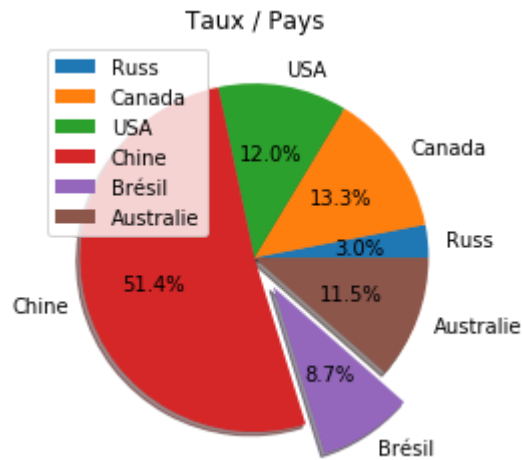


Créer un graphique de type camembert

Syntaxe : `pyplot.pie(liste, labels=[Libellé1, Libellé2, LibelléN], autopct = '%.1f%%', shadow = True/False, pctdistance = Nombre, labeldistance = Nombre)`

Exemple :

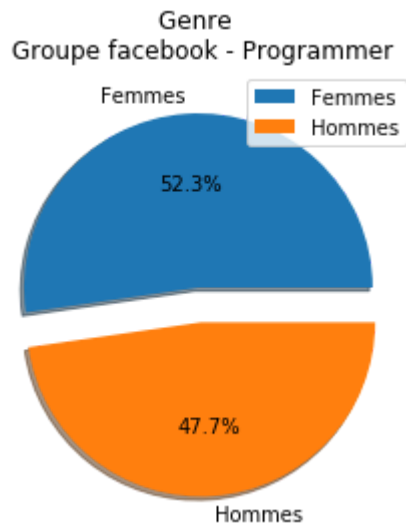
```
In [20]: from pandas import *
from matplotlib.pyplot import *
df1 = read_excel("pays.xlsx","pays")
df1["Densité"]=df1.Habitant/df1.Superficie
df2 = df1['Taux'] = df1['Densité'] / df1.Densité.sum()
explode = (0, 0, 0, 0, 0.2, 0)
pie(df2,autopct = '%.1f%%',shadow=True, labels=df1.Pays,explode=explode)
title('Taux / Pays')
legend(loc='upper left')
show()
```



Exemple 2:

Statistiques du groupe Facebook Programmer :
Genre (Femmes / Hommes)

```
In [6]: from pandas import *
from matplotlib.pyplot import *
df = read_excel("Facebook_Group.xlsx","Membres--Âge et genre")
f = df.Femmes.sum()
h = df.Hommes.sum()
explode = (0, 0.2)
pie([f,h],autopct = '%.1f%%',shadow=True, labels=['Femmes','Hommes'],explode=explode)
title('Genre \n Groupe facebook - Programmer')
legend(loc='upper right')
show()
```

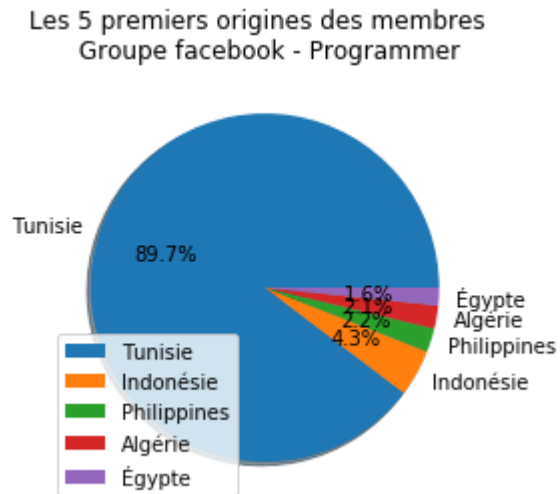


Exemple 3:

Statistiques du groupe Facebook Programmer :

Les 5 premiers origines des membres

```
In [9]: from pandas import *
from matplotlib.pyplot import *
df = read_excel("Facebook_Group.xlsx","Membres--Pays")
df.sort_values(by='Membres')
df = df.head()
pie(df.Membres,autopct = '%.1f%%',startangle=0,shadow=True, labels=df.Pays)
title('Les 5 premiers origines des membres \n Groupe facebook - Programmer')
legend(loc='lower left')
show()
```

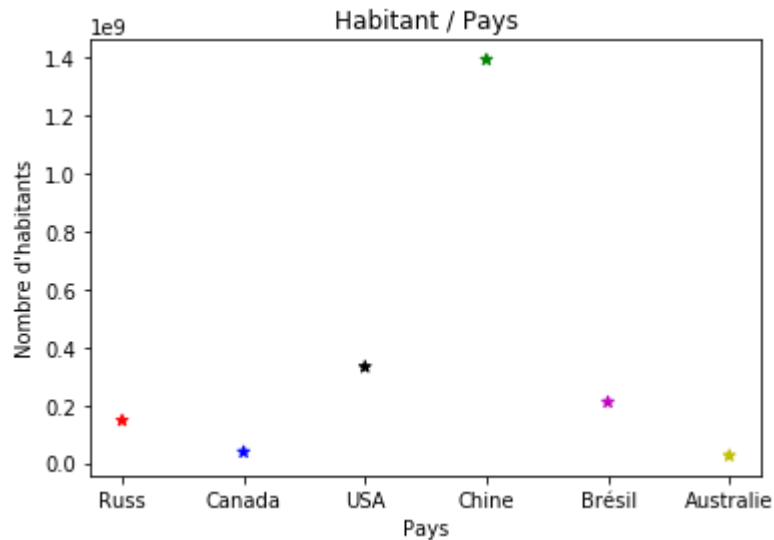


Créer un graphique de type scatter

Syntaxe : `pyplot.scatter(x, y,color=liste_couleurs,marker=type_marker)`

Exemple :

```
In [168]: from pandas import *
from matplotlib.pyplot import *
df = read_excel("pays.xlsx", "pays")
couleurs=["r", 'b', 'k', 'g', 'm', 'y']
scatter(df.Pays, df.Habitant, color=couleurs, marker='*')
xlabel('Pays')
ylabel("Nombre d'habitants")
title('Habitant / Pays')
show()
```

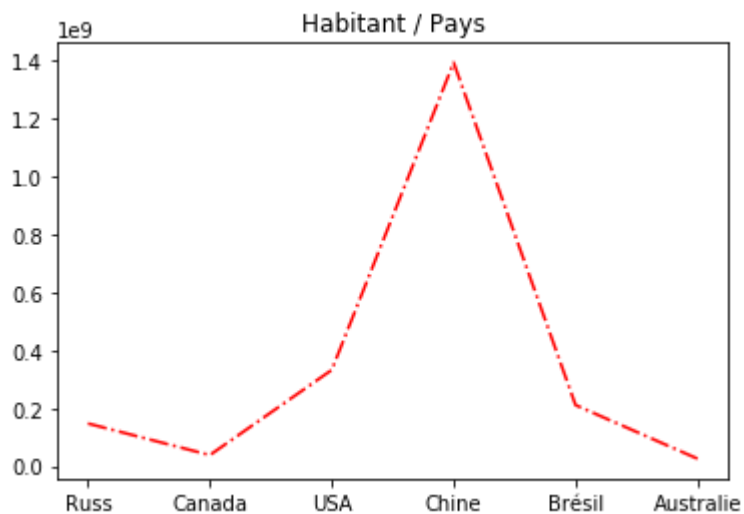


Créer un graphique de type scatter plot

Syntaxe : `pyplot.scatter(x, y, color=nom_couleur, linestyle=style)`

Exemple :

```
In [10]: from pandas import *
from matplotlib.pyplot import *
df = read_excel("pays.xlsx", "pays")
plot(df.Pays, df.Habitant, color = 'red', linestyle = 'dashdot')
title('Habitant / Pays')
show()
```



bonjour

In []: