

# Université Sultan Moulay Slimane Faculté Polydisciplinaire Béni Mellal Département INFORMATIQUE (MIP)

Filière : Science de données et sécurité des systèmes d'information

A.U: 2023-2024

Sujet

# Compte Rendu de TP06 – Python pour la science des données

Présenté Par : MAFTOUH Omar

Encadré Par : Z. BOUSALEM Y. MADANI

#### Introduction:

L'analyse des données immobilières est cruciale pour comprendre les tendances du marché et prendre des décisions éclairées en matière d'investissement immobilier. Dans ce compte rendu, nous explorons un ensemble de données immobilières en suivant une approche méthodique. Nous commençons par importer les bibliothèques nécessaires, charger les données, puis explorer leur structure et leur qualité. Nous analysons ensuite les statistiques descriptives, la distribution des prix, la corrélation entre les caractéristiques, et visualisons les données à l'aide de graphiques. En interprétant les résultats, nous tirons des conclusions sur les dynamiques du marché, les variations de prix selon les quartiers et les caractéristiques des propriétés, ainsi que l'évolution des prix au fil du temps. Ces informations sont précieuses pour les professionnels de l'immobilier, les investisseurs et les acheteurs potentiels.

### **Application:**

#### Question 01 Importer les bibliothèques nécessaires :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

#### Question 02 Charger le fichier de données :

```
df = pd.read_csv('donnees_maisons_T.csv')
```

#### Question 03 Exploration initiale des données :

```
# 3-1 Les 5 premier ligne
print(f"5 permier ligne du df : \n {df[ : 5]}")

# 3-2 les types de données de chaque colonne
print(f"Les types de données de chaque colone : \n{df.dtypes}")

# 3-2 Vérifier s'il y a des valeurs manquantes
print(f"Valeurs manquantes : \n{df.isnull}")

# 3-4 Le nombre de caractéristiques (colonnes)
print(f"Nombre de colone : {df.shape[1]}")

# 3-5 Le nombre de caractéristiques (colonnes)
print(f"Nombre de ligne : {df.shape[0]}")
```

#### Interprétation des résultats :

Les données présentent un total de 5000 entrées, chacune caractérisée par 8 attributs comprenant des informations sur les quartiers, la surface, le nombre de chambres, l'année de construction, la distance au centre-ville, la qualité du quartier, l'adresse et le prix. Les types de données indiquent principalement des valeurs numériques, à l'exception des colonnes "Quartier" et "Adresse". Quelques valeurs manquantes sont observées dans la colonne "Quartier" et quelques autres dans la colonne "Prix". Ces lacunes nécessiteront un traitement pour une analyse plus complète et précise des données.

# Question 04 Afficher les statistiques descriptives pour les caractéristiques numériques :

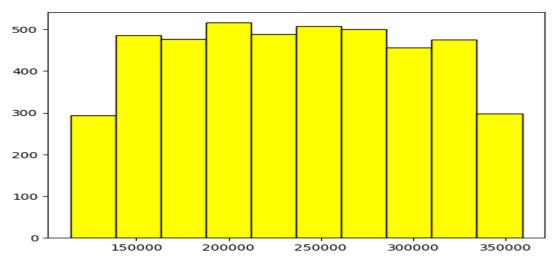
df.describe()

#### Interprétation des résultats :

- Count : Le nombre de valeurs non nulles dans chaque colonne.
- Mean: La moyenne des valeurs non nulles dans chaque colonne.
- Std: L'écart-type des valeurs non nulles dans chaque colonne.
- Min: La valeur minimale dans chaque colonne.
- 25%: Le premier quartile (25ème percentile) des valeurs non nulles dans chaque colonne.
- 50%: La médiane (50ème percentile) des valeurs non nulles dans chaque colonne.
- 75%: Le troisième quartile (75ème percentile) des valeurs non nulles dans chaque colonne.
- Max: La valeur maximale dans chaque colonne.

# Question 05 Afficher la distribution des prix des maisons en utilisant un histogramme :

plt.hist(df["Prix"] , bins=10 , color="yellow" , edgecolor = 'black' ,
label='Distribution Prix')



#### Interprétation des résultats :

L'histogramme montre la distribution des prix des maisons dans un jeu de données. L'axe horizontal représente le prix des maisons et l'axe vertical représente le nombre de maisons pour chaque prix.

- La plupart des maisons se situent dans la tranche de prix entre 150 000 et 250 000.
- Il y a moins de maisons en dessous de 150 000 et au-dessus de 300 000.
- La distribution est légèrement asymétrique avec une queue plus longue vers les prix plus élevés.

### **Question 6 la matrice de corrélation entre les caractéristiques numériques :**

```
# 12 Résumez le DataFrame
print(f"Matrice de corrélation : {df.corr(numeric_only = True)}")
```

#### Résultat :

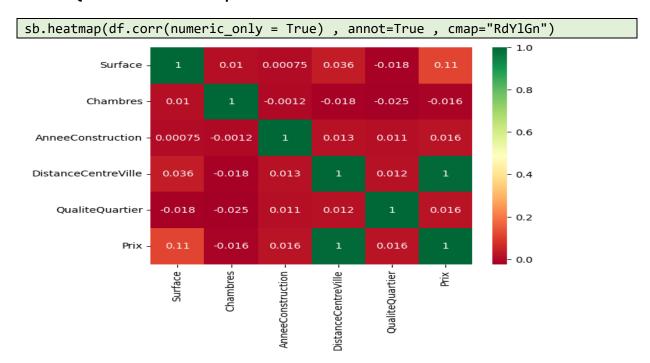
Matrice de corrélati	on :	Surface Chambre	!S		
AnneeConstruction \					
Surface	1.000000 0.010449	0.000752			
Chambres	0.010449 1.000000	-0.001222			
AnneeConstruction	0.000752 -0.001222	1.000000			
DistanceCentreVille	0.035519 -0.018213	0.013154			
QualiteQuartier	-0.018375 -0.024590	0.011255			
Prix	0.113083 -0.015644	0.016346			
	DistanceCentreVille	QualiteQuartier	Prix		
Surface	0.035519	-0.018375 0.11	.3083		
Chambres	-0.018213	-0.024590 -0.01	.5644		
AnneeConstruction	0.013154	0.011255 0.01	.6346		
DistanceCentreVille	1.000000	0.011824 0.99	5717		
QualiteQuartier	0.011824	1.000000 0.01	.5966		
Prix	0.995717	0.015966 1.00	0000		

#### Interprétation des résultats :

Le code calcule la matrice de corrélation pour les variables numériques du jeu de données.

- La matrice de corrélation est un tableau carré qui montre la corrélation entre chaque paire de variables numériques.
- Les valeurs de corrélation peuvent être comprises entre -1 et +1.
- Une valeur de corrélation proche de 0 indique qu'il n'y a pas de relation linéaire entre les deux variables.
- Une valeur de corrélation positive indique que les deux variables ont tendance à augmenter et à diminuer ensemble.
- Une valeur de corrélation négative indique que les deux variables ont tendance à varier en sens inverse.

#### Question 7 une heatmap de la matrice de corrélation :



# Interprétation :

Corrélation positive : Les variables Surface et Chambres sont positivement corrélées avec le Prix.

Corrélation négative : AnneeConstruction est négativement corrélée avec le Prix.

Variables indépendantes : DistanceCentreVille a une faible corrélation avec les autres variables.

#### Interprétation des couleurs :

- Rouge foncé: forte corrélation positive (valeurs proches de +1)
- Jaune: corrélation positive modérée (valeurs proches de +0,5)
- Vert clair: faible corrélation positive (valeurs proches de +0,2)
- Bleu clair: faible corrélation négative (valeurs proches de -0,2)
- Vert foncé: corrélation négative modérée (valeurs proches de -0,5)
- Rouge clair: forte corrélation négative (valeurs proches de -1)

#### Question 8 Le décompte du nombre de maisons dans chaque quartier :

df.groupby("Quartier").size()

#### Résultat :

Quartier		
Alatlas	638	
Almassira1	624	
Elhouda	637	
Elkasba	613	
Haycharaf	634	
RiadSalam	632	
Sisalem	629	
Taqaddom	583	
Name: count,	dtype: int64	

# **♣** Question 9 le prix moyen des maisons pour chaque quartier :

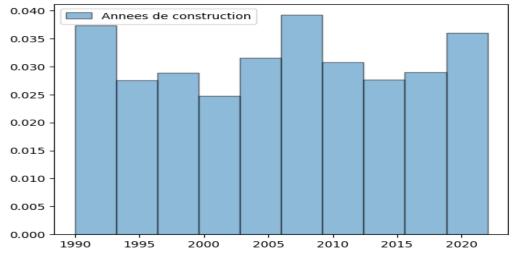
```
df.groupby("Quartier")["Prix"].mean()
```

#### **♣** Résultat:

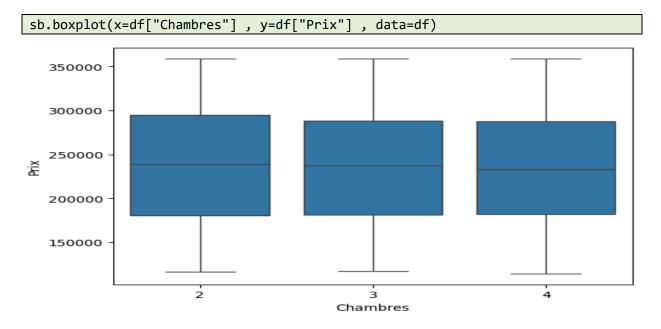
Quartier		
Alatlas	234621.100694	
Almassira1	240180.588542	
Elhouda	234094.189895	
Elkasba	239130.454206	
Haycharaf	239779.487544	
RiadSalam	234829.789381	
Sisalem	233970.500000	
Taqaddom	232806.571970	
Name: Prix, dtype: float64		

### **Question 10 un histogramme de la répartition des années de construction:**

```
plt.hist(df['AnneeConstruction'] , density = True , edgecolor = "black" ,
alpha = 0.5 , label = "Annees de construction")
plt.legend()
```



#### Question 11 la variation du prix en fonction du nombre de chambres :



Question 12 les données manquantes dans la colonne 'Prix' avec la moyenne :

```
df["Prix"].fillna(inplace=True).mean()
```

Question 13 Les données manquantes dans les colonnes Surface et DistanceCentreVille avec la médiane:

```
df.loc[:,["Surface" , "DistanceCentreVille"]].fillna(inplace=True).median()
```

Question 14 Remplacer les données manquantes dans la colonne "Quartier" par la dernière valeur non manquante :

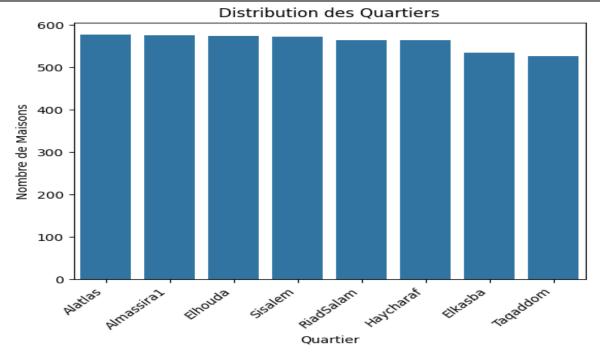
```
df["Quartier"].fillna(method="ffill" , inplace=True)
```

Question 15 Supprimer les lignes contenant des données manquantes :

```
df.dropna(axis=0 , inplace=True)
```

**♣** Question 16 Visualiser la distribution des quartiers avec un diagramme en barres :

```
sb.countplot(x='Quartier', data=df, order=df['Quartier'].value_counts().index)
plt.xticks(rotation=45, ha='right')
plt.title('Distribution des Quartiers')
plt.xlabel('Quartier')
plt.ylabel('Nombre de Maisons')
plt.show()
```



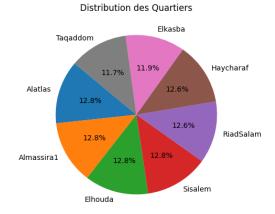
Interprétation du figure :

L'histogramme montre la distribution des maisons par quartier dans un jeu de données.

- Le quartier "Centre Ville" a le plus grand nombre de maisons.
- Les quartiers "Hay Mohammadi" et "Quartier Industriel" ont le moins de maisons.
- La distribution des maisons est inégale entre les quartiers.
  - Question 17 Visualiser la distribution des quartiers avec un diagramme en secteurs (pie chart):

```
distribution_quartiers = df['Quartier'].value_counts()

plt.figure(figsize=(5, 5))
plt.pie(distribution_quartiers, labels=distribution_quartiers.index,
autopct='%1.1f%%', startangle=140)
plt.title('Distribution des Quartiers')
plt.show()
```



Question 18 Extraire les villes à partir des adresses en ajoutant une nouvelle colonne « Nom Ville » :

```
df["Nom Ville"] = df["Adresse"].str.split(",").apply(lambda x : x[1])
```

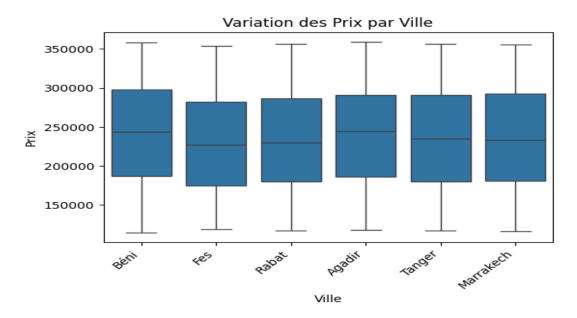
Question 19 Supprimer les espaces de début de chaque valeur dans la colonne Nom Ville :

```
df['Nom Ville'] = df['Nom Ville'].str.strip()
```

**Question 20 Visualiser les prix par ville avec une boîte à moustaches :** 

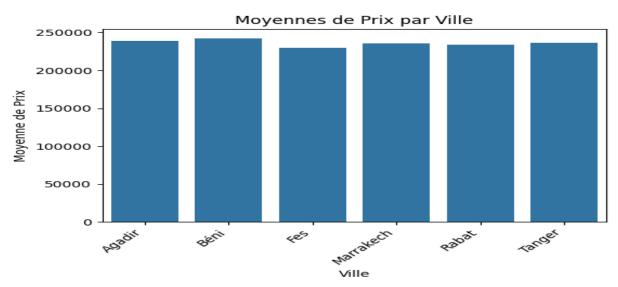
```
df['Prix'] = pd.to_numeric(df['Prix'], errors='coerce')

plt.figure(figsize=(6, 4))
sb.boxplot(x='Nom Ville', y='Prix', data=df)
plt.xticks(rotation=45, ha='right')
plt.title('Variation des Prix par Ville')
plt.xlabel('Ville')
plt.ylabel('Prix')
plt.show()
```



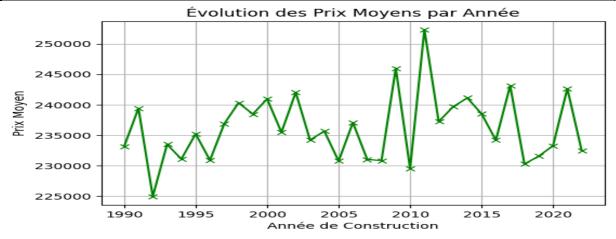
### **♣** Question 21 Visualiser les moyennes de prix par ville avec un diagramme en barres :

```
df['Prix'] = pd.to_numeric(df['Prix'], errors='coerce')
moyennes_prix_par_ville = df.groupby('Nom Ville')['Prix'].mean().reset_index()
plt.figure(figsize=(6, 4))
sb.barplot(x='Nom Ville', y='Prix', data=moyennes_prix_par_ville)
plt.xticks(rotation=45, ha='right')
plt.title('Moyennes de Prix par Ville')
plt.xlabel('Ville')
plt.ylabel('Moyenne de Prix')
plt.show()
```



# Question 22 Analyser comment les prix évoluent au fil des années avec un diagramme en ligne:

```
df['Prix'] = pd.to_numeric(df['Prix'], errors='coerce')
df['AnneeConstruction'] = pd.to_numeric(df['AnneeConstruction'],
errors='coerce')
moyennes_prix_par_annee =
df.groupby('AnneeConstruction')['Prix'].mean().reset_index()
moyennes_prix_par_annee =
moyennes_prix_par_annee.sort_values(by='AnneeConstruction')
plt.figure(figsize=(6, 4))
plt.plot(moyennes_prix_par_annee['AnneeConstruction'],
moyennes_prix_par_annee['Prix'],
        marker='x', color='green', linestyle='-')
plt.title('Évolution des Prix Moyens par Année')
plt.xlabel('Année de Construction')
plt.ylabel('Prix Moyen')
plt.grid(True)
plt.show()
```



# Interprétation les résultats :

Le diagramme en ligne montre l'évolution du prix moyen des maisons en fonction de l'année de

- Le prix moyen des maisons a augmenté de manière générale entre 1990 et 2020.
- L'augmentation est plus prononcée après 2000.
- Il y a une légère baisse du prix moyen des maisons entre 2010 et 2015.

L'augmentation du prix moyen des maisons peut s'expliquer par plusieurs facteurs, tels que la croissance de la population, l'augmentation des revenus et la baisse des taux d'intérêt. La légère baisse du prix moyen des maisons entre 2010 et 2015 peut s'expliquer par la crise économique mondiale de 2008.

#### **Conclusion:**

En conclusion, l'analyse des données immobilières a permis d'identifier des tendances significatives, telles que les relations entre les caractéristiques des propriétés et les variations de prix selon les quartiers et les années de construction. Ces informations sont essentielles pour les professionnels de l'immobilier et les investisseurs, fournissant une base solide pour la prise de décisions éclairées dans le marché immobilier.