



To import pandas we need the following commands

In [3]: `!pip install pandas`

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

In [4]: `import pandas as pd`

For reading the csv file below is the code

In [6]: `df=pd.read_csv(r"/content/Property Prices in Tunisia.csv")`

checking the datafram using the variable 'df'

In [8]: `df`

Out[8]:

	category	room_count	bathroom_count	size	type	price	c
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	100000.0	Ariège
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	316000.0	Ariège
2	Appartements	2.0	1.0	80.0	À Louer	380.0	Ariège
3	Locations de vacances	1.0	1.0	90.0	À Louer	70.0	Ariège
4	Appartements	2.0	2.0	113.0	À Vendre	170000.0	Ariège
...	...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	3200000.0	Tarbes
12744	Appartements	1.0	1.0	100.0	À Louer	600.0	Tarbes
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	1950000.0	Tarbes
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	240000.0	Tarbes
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	500.0	Tarbes

12748 rows × 9 columns

## DATA EXPLORATION

head(): Viewing the first few rows of a Dataframe

In [9]: `df.head()`

Out[9]:

	category	room_count	bathroom_count	size	type	price	city	re
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	1000000.0	Ariana	R
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	316000.0	Ariana	/
2	Appartements	2.0	1.0	80.0	À Louer	380.0	Ariana	/
3	Locations de vacances	1.0	1.0	90.0	À Louer	70.0	Ariana	/
4	Appartements	2.0	2.0	113.0	À Vendre	170000.0	Ariana	/

In [11]: df.head(15)

Out[11]:

	<b>category</b>	<b>room_count</b>	<b>bathroom_count</b>	<b>size</b>	<b>type</b>	<b>price</b>	<b>city</b>
<b>0</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	100000.0	Ariana
<b>1</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	316000.0	Ariana
<b>2</b>	Appartements	2.0	1.0	80.0	À Louer	380.0	Ariana
<b>3</b>	Locations de vacances	1.0	1.0	90.0	À Louer	70.0	Ariana
<b>4</b>	Appartements	2.0	2.0	113.0	À Vendre	170000.0	Ariana
<b>5</b>	Locations de vacances	1.0	1.0	70.0	À Louer	80.0	Ariana
<b>6</b>	Appartements	2.0	1.0	100.0	À Louer	670.0	Ariana
<b>7</b>	Magasins, Commerces et Locaux industriels	2.0	1.0	20.0	À Louer	650.0	Ariana
<b>8</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	180000.0	Ariana
<b>9</b>	Magasins, Commerces et Locaux industriels	1.0	1.0	40.0	À Louer	450.0	Ariana
<b>10</b>	Maisons et Villas	4.0	2.0	1208.0	À Vendre	700000.0	Ariana
<b>11</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	600000.0	Ariana
<b>12</b>	Maisons et Villas	5.0	3.0	220.0	À Louer	2000.0	Ariana
<b>13</b>	Appartements	3.0	2.0	138.0	À Vendre	340000.0	Ariana
<b>14</b>	Maisons et Villas	2.0	1.0	150.0	À Vendre	120000.0	Ariana

tail(): Viewing the last few rows of a Dataframe.

In [10]: `df.tail()`

Out[10]:

	category	room_count	bathroom_count	size	type	price	cit
<b>12743</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	3200000.0	Tun
<b>12744</b>	Appartements	1.0	1.0	100.0	À Louer	600.0	Tun
<b>12745</b>	Maisons et Villas	3.0	1.0	760.0	À Vendre	1950000.0	Tun
<b>12746</b>	Maisons et Villas	3.0	1.0	190.0	À Vendre	240000.0	Tun
<b>12747</b>	Maisons et Villas	2.0	1.0	70.0	À Louer	500.0	Tun

In [13]: `df.tail(15)`

Out[13]:

	category	room_count	bathroom_count	size	type	price	c
12733	Maisons et Villas	3.0	1.0	160.0	À Louer	1000.0	Tu
12734	Appartements	2.0	1.0	150.0	À Louer	2500.0	Tu
12735	Appartements	2.0	1.0	70.0	À Louer	140.0	Tu
12736	Magasins, Commerces et Locaux industriels	1.0	1.0	29.0	À Vendre	115000.0	Tu
12737	Magasins, Commerces et Locaux industriels	1.0	1.0	300.0	À Louer	1800.0	Tu
12738	Appartements	1.0	1.0	60.0	À Louer	90.0	Tu
12739	Magasins, Commerces et Locaux industriels	2.0	1.0	100.0	À Vendre	50000.0	Tu
12740	Appartements	2.0	1.0	68.0	À Vendre	110000.0	Tu
12741	Appartements	2.0	1.0	140.0	À Vendre	250000.0	Tu
12742	Maisons et Villas	5.0	3.0	340.0	À Vendre	23205455.0	Tu
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	3200000.0	Tu
12744	Appartements	1.0	1.0	100.0	À Louer	600.0	Tu
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	1950000.0	Tu
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	240000.0	Tu
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	500.0	Tu

info():Displaying information about the DataFrame.

In [14]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12748 entries, 0 to 12747
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   category         12748 non-null   object  
 1   room_count       12748 non-null   float64 
 2   bathroom_count  12748 non-null   float64 
 3   size             12748 non-null   float64 
 4   type             12748 non-null   object  
 5   price            12748 non-null   float64 
 6   city             12748 non-null   object  
 7   region           12748 non-null   object  
 8   log_price        12748 non-null   float64 
dtypes: float64(5), object(4)
memory usage: 896.5+ KB
```

describe():Generating summary statistics of numeric column

Now if you see only 3 columns are numerical and hence describe shows summary of those 3

```
In [19]: df.describe()
```

```
Out[19]:      room_count  bathroom_count          size        price      log_price
count    12748.000000    12748.000000  12748.000000  1.274800e+04  12748.000000
mean     1.759649        0.759884    130.896219  1.601575e+07  4.374245
std      2.171468        1.264812    184.074990  1.016644e+09  1.389788
min     -1.000000        -1.000000   -1.000000  1.000000e+01  1.000000
25%     -1.000000        -1.000000   -1.000000  8.500000e+02  2.929419
50%      2.000000        1.000000    95.000000  8.975000e+04  4.953033
75%      3.000000        1.000000   150.000000  2.600000e+05  5.414973
max     20.000000        10.000000   2000.000000 1.000000e+11  11.000000
```

shape: Getting the dimensions(rows and columns) of a Dataframe

```
In [20]: df.shape
```

```
Out[20]: (12748, 9)
```

column: Accessing the column names of a Dataframe

```
In [22]: df.values
```

```
Out[22]: array([['Terrains et Fermes', -1.0, -1.0, ..., 'Ariana', 'Raoued', 5.0],
   ['Terrains et Fermes', -1.0, -1.0, ..., 'Ariana', 'Autres villes',
    5.4996870826184034],
   ['Appartements', 2.0, 1.0, ..., 'Ariana', 'Autres villes',
    2.57978359661681],
   ...,
   ['Maisons et Villas', 3.0, 1.0, ..., 'Tunis', 'La Marsa',
    6.290034611362518],
   ['Maisons et Villas', 3.0, 1.0, ..., 'Tunis', 'La Marsa',
    5.380211241711606],
   ['Maisons et Villas', 2.0, 1.0, ..., 'Tunis', 'Sidi Bou Said',
    2.6989700043360187]], dtype=object)
```

types: Getting data types of columns:

```
In [23]: df.dtypes
```

```
Out[23]:
```

	0
<b>category</b>	object
<b>room_count</b>	float64
<b>bathroom_count</b>	float64
<b>size</b>	float64
<b>type</b>	object
<b>price</b>	float64
<b>city</b>	object
<b>region</b>	object
<b>log_price</b>	float64

**dtype:** object

Data Selection and Filtering

loc[]: Accessing rows and columns by label.

```
In [50]: df.loc(1)
```

```
Out[50]: <pandas.core.indexing._LocIndexer at 0x7bba4277f110>
```

iloc[]: Accessing rows and columns by integer index

```
In [26]: df.iloc[0,0]
```

```
Out[26]: 'Terrains et Fermes'
```

query(): Filtering rows using a query expression

```
In [48]: filtered_df=df.query("size > 100")
print(filtered_df)
```

```
          category  room_count  bathroom_count \
4           Appartements      2.0            2.0
10          Maisons et Villas    4.0            2.0
12          Maisons et Villas    5.0            3.0
13           Appartements      3.0            2.0
14          Maisons et Villas    2.0            1.0
...
12737  Magasins, Commerces et Locaux industriels    1.0            1.0
12741           Appartements      2.0            1.0
12742          Maisons et Villas    5.0            3.0
12745          Maisons et Villas    3.0            1.0
12746          Maisons et Villas    3.0            1.0

      size      type     price      city      region  log_price
4    113.0  À Vendre  170000.0  Ariana  Ariana Ville  5.230449
10   1208.0  À Vendre  700000.0  Ariana  La Soukra  5.845098
12    220.0  À Louer   2000.0  Ariana  Chotrana  3.301030
13   138.0  À Vendre  340000.0  Ariana Jardins D'el Menzah  5.531479
14   150.0  À Vendre  120000.0  Ariana  Raoued  5.079181
...
12737   300.0  À Louer   1800.0  Tunis  La Marsa  3.255273
12741   140.0  À Vendre  250000.0  Tunis  Tunis  5.397940
12742   340.0  À Vendre  23205455.0  Tunis  Autres villes  7.365590
12745   760.0  À Vendre  1950000.0  Tunis  La Marsa  6.290035
12746   190.0  À Vendre  240000.0  Tunis  La Marsa  5.380211
```

[5489 rows x 9 columns]

Data Manipulation

drop(): Removing rows or coluns from a Dataframe.

```
In [49]: df_dropped = df.drop(columns=['size'])
print(df_dropped)
```

```

category    room_count   bathroom_count      type      price
\\
0        Terrains et Fermes     -1.0           -1.0  À Vendre  1000000.0
1        Terrains et Fermes     -1.0           -1.0  À Vendre  3160000.0
2            Appartements       2.0            1.0  À Louer   380.0
3  Locations de vacances     1.0            1.0  À Louer    70.0
4            Appartements       2.0           2.0  À Vendre  1700000.0
...
12743      Terrains et Fermes     -1.0           -1.0  À Vendre 3200000.0
12744          Appartements       1.0            1.0  À Louer   600.0
12745      Maisons et Villas     3.0            1.0  À Vendre 1950000.0
12746      Maisons et Villas     3.0            1.0  À Vendre 2400000.0
12747      Maisons et Villas     2.0            1.0  À Louer   500.0

city        region  log_price
0    Ariana        Raoued  5.000000
1    Ariana  Autres villes  5.499687
2    Ariana  Autres villes  2.579784
3    Ariana  Autres villes  1.845098
4    Ariana  Ariana Ville  5.230449
...
12743    Tunis  Sidi Bou Said  6.505150
12744    Tunis  Autres villes  2.778151
12745    Tunis      La Marsa  6.290035
12746    Tunis      La Marsa  5.380211
12747    Tunis  Sidi Bou Said  2.698970

```

[12748 rows x 8 columns]

rename():Renaming columns

```
In [57]: df_renamed=df.rename(columns={'category':'Full categroy'})
print(df_renamed)
```

	Full categroy	room_count	bathroom_count	size	type	\
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
2	Appartements	2.0	1.0	80.0	À Louer	
3	Locations de vacances	1.0	1.0	90.0	À Louer	
4	Appartements	2.0	2.0	113.0	À Vendre	
...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
12744	Appartements	1.0	1.0	100.0	À Louer	
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	
	price	city	region	log_price		
0	1000000.0	Ariana	Raoued	5.000000		
1	3160000.0	Ariana	Autres villes	5.499687		
2	380.0	Ariana	Autres villes	2.579784		
3	70.0	Ariana	Autres villes	1.845098		
4	1700000.0	Ariana	Ariana Ville	5.230449		
...	...	...	...	...	...	
12743	3200000.0	Tunis	Sidi Bou Said	6.505150		
12744	600.0	Tunis	Autres villes	2.778151		
12745	1950000.0	Tunis	La Marsa	6.290035		
12746	240000.0	Tunis	La Marsa	5.380211		
12747	500.0	Tunis	Sidi Bou Said	2.698970		

[12748 rows x 9 columns]

same goes for rename to permanently rename the column name use  
`df_renamed=df.`

`df.rename(columns={'category':'Full categroy'}, inplace=True)`

In [58]: df

Out[58]:

	category	room_count	bathroom_count	size	type	price	c
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	100000.0	Ari
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	316000.0	Ari
2	Appartements	2.0	1.0	80.0	À Louer	380.0	Ari
3	Locations de vacances	1.0	1.0	90.0	À Louer	70.0	Ari
4	Appartements	2.0	2.0	113.0	À Vendre	170000.0	Ari
...	...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	3200000.0	Tu
12744	Appartements	1.0	1.0	100.0	À Louer	600.0	Tu
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	1950000.0	Tu
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	240000.0	Tu
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	500.0	Tu

12748 rows × 9 columns

sort\_values():Sorting a Dataframe by one or more columns.

In [60]:

```
df_sorted=df.sort_values(by='size')
df_sorted
```

Out[60]:

	category	room_count	bathroom_count	size	type	price	ci
<b>2355</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	55000.0	Ariane
<b>2341</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	185000.0	Ariane
<b>2344</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	100000.0	Ariane
<b>2346</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	78000.0	Ariane
<b>7704</b>	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	520000.0	Sous
...	...	...	...	...	...	...	...
<b>6688</b>	Maisons et Villas	4.0	2.0	2000.0	À Vendre	380000.0	Sous
<b>5469</b>	Maisons et Villas	2.0	2.0	2000.0	À Louer	2800.0	Mahc
<b>5667</b>	Maisons et Villas	3.0	1.0	2000.0	À Vendre	150000.0	Médenia
<b>8578</b>	Maisons et Villas	2.0	1.0	2000.0	À Vendre	380000.0	Sous
<b>11787</b>	Maisons et Villas	3.0	1.0	2000.0	À Vendre	350000.0	Tur

12748 rows × 9 columns

fillna():Filling missing values in a Dtaframe.

```
In [61]: df_filled=df.fillna(0)
df_filled
```

Out[61]:

	category	room_count	bathroom_count	size	type	price	c
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	100000.0	Ariège
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	316000.0	Ariège
2	Appartements	2.0	1.0	80.0	À Louer	380.0	Ariège
3	Locations de vacances	1.0	1.0	90.0	À Louer	70.0	Ariège
4	Appartements	2.0	2.0	113.0	À Vendre	170000.0	Ariège
...	...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	3200000.0	Tunisie
12744	Appartements	1.0	1.0	100.0	À Louer	600.0	Tunisie
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	1950000.0	Tunisie
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	240000.0	Tunisie
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	500.0	Tunisie

12748 rows × 9 columns

drop\_duplicates(): Removing duplicate rows.

In [62]:

```
df_unique=df.drop_duplicates()
print(df_unique)
```

	category	room_count	bathroom_count	size	type	\
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
2	Appartements	2.0	1.0	80.0	À Louer	
3	Locations de vacances	1.0	1.0	90.0	À Louer	
4	Appartements	2.0	2.0	113.0	À Vendre	
...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
12744	Appartements	1.0	1.0	100.0	À Louer	
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	
	price	city	region	log_price		
0	1000000.0	Ariana	Raoued	5.000000		
1	3160000.0	Ariana	Autres villes	5.499687		
2	380.0	Ariana	Autres villes	2.579784		
3	70.0	Ariana	Autres villes	1.845098		
4	1700000.0	Ariana	Ariana Ville	5.230449		
...	...	...	...	...	...	...
12743	3200000.0	Tunis	Sidi Bou Said	6.505150		
12744	600.0	Tunis	Autres villes	2.778151		
12745	1950000.0	Tunis	La Marsa	6.290035		
12746	240000.0	Tunis	La Marsa	5.380211		
12747	500.0	Tunis	Sidi Bou Said	2.698970		

[11135 rows x 9 columns]

replace():Replacing values in a Dataframe

```
In [65]: df_replaced=df.replace({'Terrains et Fermes':'fermes'})
print(df_replaced)
```

```

          category  room_count  bathroom_count    size      type \
0            fermes       -1.0           -1.0   -1.0  À Vendre
1            fermes       -1.0           -1.0   -1.0  À Vendre
2        Appartements        2.0            1.0  80.0  À Louer
3  Locations de vacances        1.0            1.0  90.0  À Louer
4        Appartements        2.0            2.0 113.0  À Vendre
...
12743            ...         ...         ...     ...
12743            fermes       -1.0           -1.0   -1.0  À Vendre
12744            Appartements        1.0            1.0 100.0  À Louer
12745        Maisons et Villas        3.0            1.0 760.0  À Vendre
12746        Maisons et Villas        3.0            1.0 190.0  À Vendre
12747        Maisons et Villas        2.0            1.0  70.0  À Louer

      price     city      region  log_price
0  1000000.0  Ariana      Raoued  5.000000
1  3160000.0  Ariana  Autres villes  5.499687
2    380.0  Ariana  Autres villes  2.579784
3     70.0  Ariana  Autres villes  1.845098
4  1700000.0  Ariana     Ariana Ville  5.230449
...
12743  3200000.0    Tunis  Sidi Bou Said  6.505150
12744    600.0    Tunis  Autres villes  2.778151
12745  1950000.0    Tunis      La Marsa  6.290035
12746  240000.0    Tunis      La Marsa  5.380211
12747    500.0    Tunis  Sidi Bou Said  2.698970

```

[12748 rows x 9 columns]

grouping and Aggregation

groupby():Grouping data by one or more columns for aggregation

```
In [67]: grouped_df=df.groupby('city')['price'].sum()
grouped_df
```

Out[67]:

price

city	price
Ariana	4.269905e+09
Ben arous	1.025832e+09
Bizerte	2.887079e+09
Béja	3.511091e+08
Gabès	1.002493e+11
Gafsa	1.693071e+08
Jendouba	1.079715e+07
Kairouan	2.223898e+08
Kasserine	9.853333e+08
Kébili	9.467200e+06
La manouba	5.988976e+09
Le kef	5.703807e+08
Mahdia	7.762563e+08
Monastir	2.267905e+09
Médenine	1.289848e+10
Nabeul	3.583874e+08
Sfax	1.895223e+09
Sidi bouzid	1.259995e+09
Siliana	1.161653e+07
Sousse	2.510973e+09
Tataouine	7.028610e+08
Tozeur	9.044530e+06
Tunis	6.250273e+10
Zaghouan	2.235485e+09

**dtype:** float64

agg():Appiying aggregation functions(e.g.,sum,mean)to grouped data

In [73]: `agg_df=df.groupby('category').agg({'size': 'mean'})  
print(agg_df)`

```
size
category
Appartements           107.616383
Bureaux et Plateaux    197.498915
Colocations             110.000000
Locations de vacances   97.959732
Magasins, Commerces et Locaux industriels 204.993856
Maisons et Villas        286.714060
Terrains et Fermes       -1.000000
```

count():Counting non-nullvalues in each group.

```
In [76]: count_df=df.groupby('category')['size'].count()
count_df
```

```
Out[76]:
```

category	size
<b>Appartements</b>	4700
<b>Bureaux et Plateaux</b>	461
<b>Colocations</b>	65
<b>Locations de vacances</b>	298
<b>Magasins, Commerces et Locaux industriels</b>	651
<b>Maisons et Villas</b>	3158
<b>Terrains et Fermes</b>	3415

**dtype:** int64

## Data Cleaning

dropna():Removing rows or columns with missing values

```
In [77]: df_cleaned=df.dropna()
print(df_cleaned)
```

	category	room_count	bathroom_count	size	type	\
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
2	Appartements	2.0	1.0	80.0	À Louer	
3	Locations de vacances	1.0	1.0	90.0	À Louer	
4	Appartements	2.0	2.0	113.0	À Vendre	
...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
12744	Appartements	1.0	1.0	100.0	À Louer	
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	
	price	city	region	log_price		
0	1000000.0	Ariana	Raoued	5.000000		
1	3160000.0	Ariana	Autres villes	5.499687		
2	380.0	Ariana	Autres villes	2.579784		
3	70.0	Ariana	Autres villes	1.845098		
4	1700000.0	Ariana	Ariana Ville	5.230449		
...	...	...	...	...	...	
12743	3200000.0	Tunis	Sidi Bou Said	6.505150		
12744	600.0	Tunis	Autres villes	2.778151		
12745	1950000.0	Tunis	La Marsa	6.290035		
12746	240000.0	Tunis	La Marsa	5.380211		
12747	500.0	Tunis	Sidi Bou Said	2.698970		

[12748 rows x 9 columns]

fillna():filling missing values with specified values or methodes.

```
In [78]: df_filled=df.fillna(0)
          print(df_filled)
```

	category	room_count	bathroom_count	size	type	\
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
2	Appartements	2.0	1.0	80.0	À Louer	
3	Locations de vacances	1.0	1.0	90.0	À Louer	
4	Appartements	2.0	2.0	113.0	À Vendre	
...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
12744	Appartements	1.0	1.0	100.0	À Louer	
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	
	price	city	region	log_price		
0	1000000.0	Ariana	Raoued	5.000000		
1	3160000.0	Ariana	Autres villes	5.499687		
2	380.0	Ariana	Autres villes	2.579784		
3	70.0	Ariana	Autres villes	1.845098		
4	1700000.0	Ariana	Ariana Ville	5.230449		
...	...	...	...	...		
12743	3200000.0	Tunis	Sidi Bou Said	6.505150		
12744	600.0	Tunis	Autres villes	2.778151		
12745	1950000.0	Tunis	La Marsa	6.290035		
12746	240000.0	Tunis	La Marsa	5.380211		
12747	500.0	Tunis	Sidi Bou Said	2.698970		

[12748 rows x 9 columns]

String operations

```
In [81]: df['contains_a']=df['category'].str.contains('D')
print(df)
```

	category	room_count	bathroom_count	size	type	\
0	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
1	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
2	Appartements	2.0	1.0	80.0	À Louer	
3	Locations de vacances	1.0	1.0	90.0	À Louer	
4	Appartements	2.0	2.0	113.0	À Vendre	
...	...	...	...	...	...	...
12743	Terrains et Fermes	-1.0	-1.0	-1.0	À Vendre	
12744	Appartements	1.0	1.0	100.0	À Louer	
12745	Maisons et Villas	3.0	1.0	760.0	À Vendre	
12746	Maisons et Villas	3.0	1.0	190.0	À Vendre	
12747	Maisons et Villas	2.0	1.0	70.0	À Louer	
	price	city	region	log_price	contains_a	
0	1000000.0	Ariana	Raoued	5.000000	False	
1	3160000.0	Ariana	Autres villes	5.499687	False	
2	380.0	Ariana	Autres villes	2.579784	False	
3	70.0	Ariana	Autres villes	1.845098	False	
4	1700000.0	Ariana	Ariana Ville	5.230449	False	
...	...	...	...	...	...	...
12743	3200000.0	Tunis	Sidi Bou Said	6.505150	False	
12744	600.0	Tunis	Autres villes	2.778151	False	
12745	1950000.0	Tunis	La Marsa	6.290035	False	
12746	240000.0	Tunis	La Marsa	5.380211	False	
12747	500.0	Tunis	Sidi Bou Said	2.698970	False	

[12748 rows x 10 columns]

str.strip():Remaning leading and trailing whitespace

```
In [82]: df['category']=df['category'].str.strip()
print(df)
```

```

          category  room_count  bathroom_count    size    type \
0      Terrains et Fermes     -1.0           -1.0   -1.0  À Vendre
1      Terrains et Fermes     -1.0           -1.0   -1.0  À Vendre
2          Appartements      2.0            1.0   80.0  À Louer
3  Locations de vacances     1.0            1.0   90.0  À Louer
4          Appartements      2.0            2.0  113.0  À Vendre
...
12743      Terrains et Fermes     -1.0           -1.0   -1.0  À Vendre
12744          Appartements      1.0            1.0  100.0  À Louer
12745      Maisons et Villas      3.0            1.0  760.0  À Vendre
12746      Maisons et Villas      3.0            1.0  190.0  À Vendre
12747      Maisons et Villas      2.0            1.0   70.0  À Louer

      price     city        region  log_price  contains_a
0  1000000.0  Ariana       Raoued  5.000000    False
1  3160000.0  Ariana  Autres villes  5.499687    False
2    380.0  Ariana  Autres villes  2.579784    False
3     70.0  Ariana  Autres villes  1.845098    False
4  1700000.0  Ariana  Ariana Ville  5.230449    False
...
12743   3200000.0    Tunis  Sidi Bou Said  6.505150    False
12744     600.0    Tunis  Autres villes  2.778151    False
12745  1950000.0    Tunis      La Marsa  6.290035    False
12746  240000.0     Tunis      La Marsa  5.380211    False
12747    500.0    Tunis  Sidi Bou Said  2.698970    False

```

[12748 rows x 10 columns]

### statistical Analysis

`corr()`:Calculating the correlation between columns.

`value_counts()`:countingunique values in a column.

```
In [84]: counts=df['category'].value_counts()
print(counts)
```

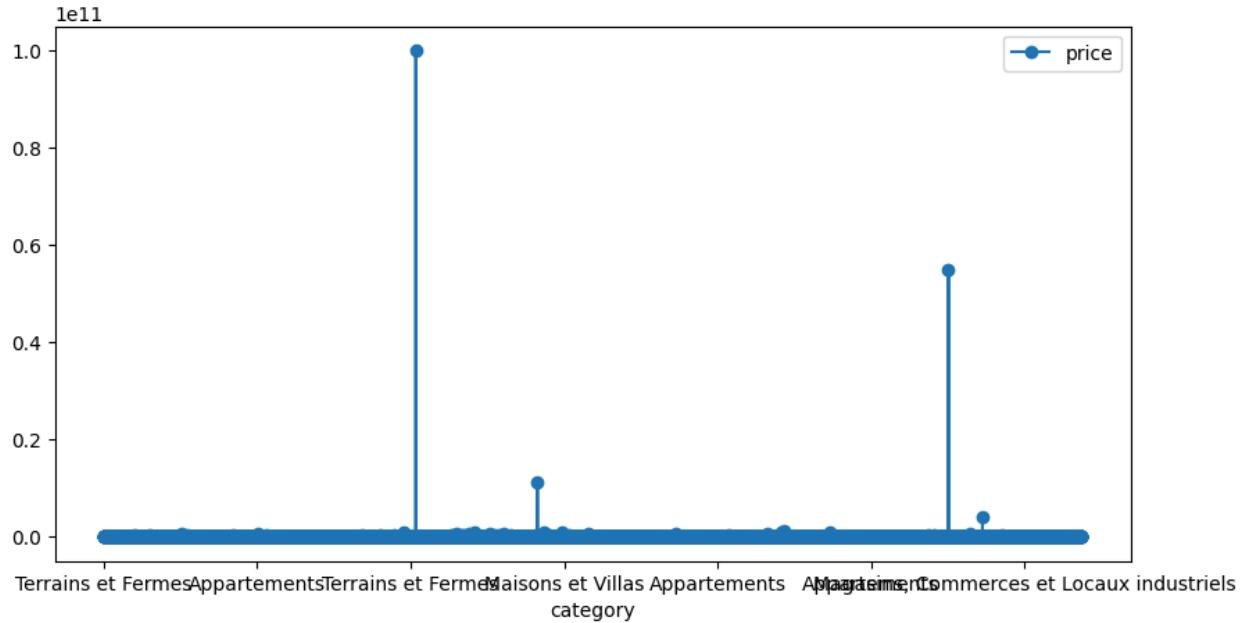
category	
Appartements	4700
Terrains et Fermes	3415
Maisons et Villas	3158
Magasins, Commerces et Locaux industriels	651
Bureaux et Plateaux	461
Locations de vacances	298
Colocations	65
Name: count, dtype: int64	

Data visualisation using pandas where backend is matplotlib itself

LINE PLOT:--A line plot is useful for visualizing trends over time.

```
In [89]: import matplotlib.pyplot as plt
df.plot(x='category',y='price',kind='line',marker='o',figsize=(10,5))
```

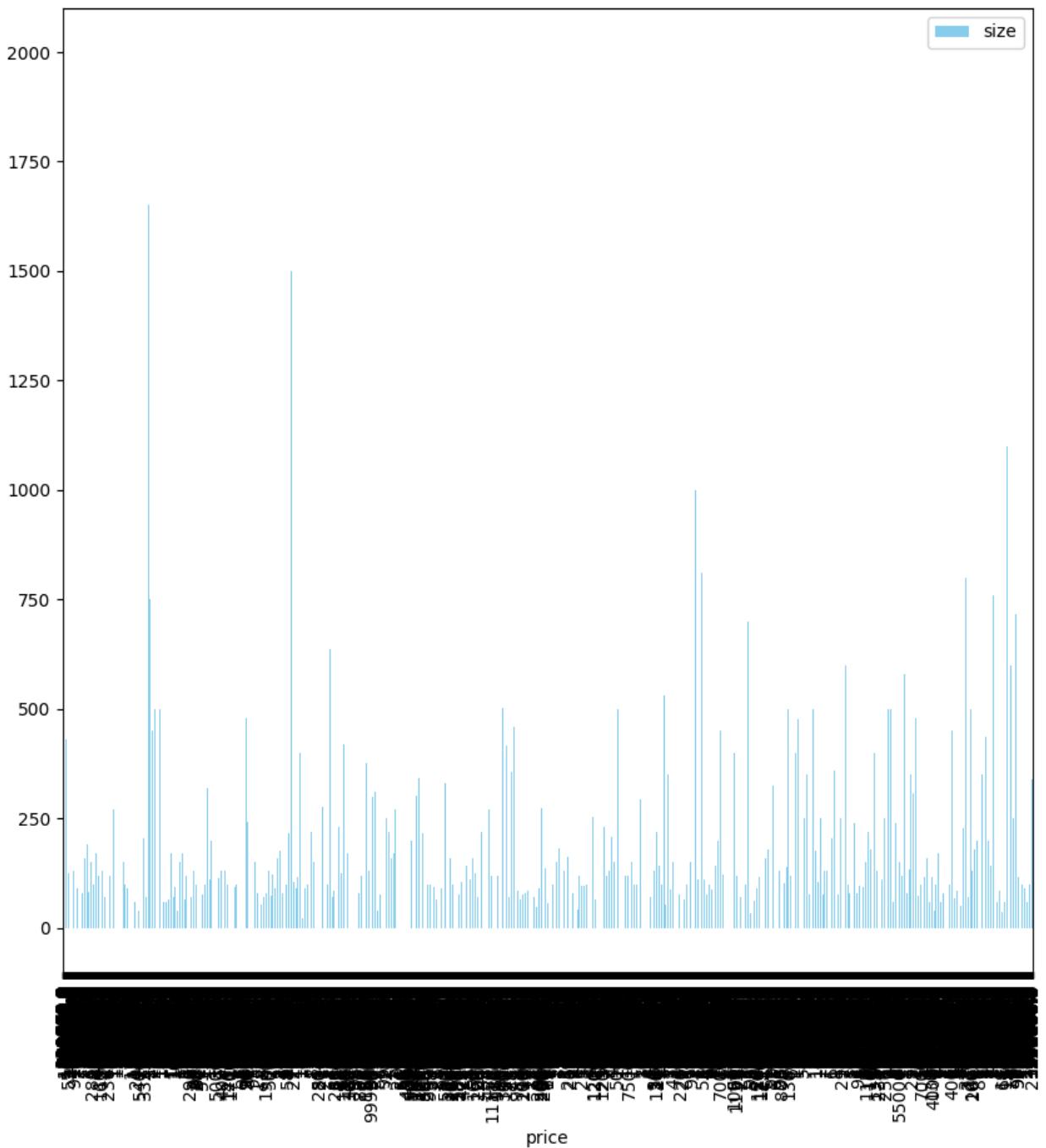
Out[89]: <Axes: xlabel='category'>



**BAR PLOT:**--A bar plot is used to compare the frequency or count of item across different categories.

```
In [97]: df.plot(x='price',y='size',kind='bar',  
color='skyblue',figsize=(10,10))
```

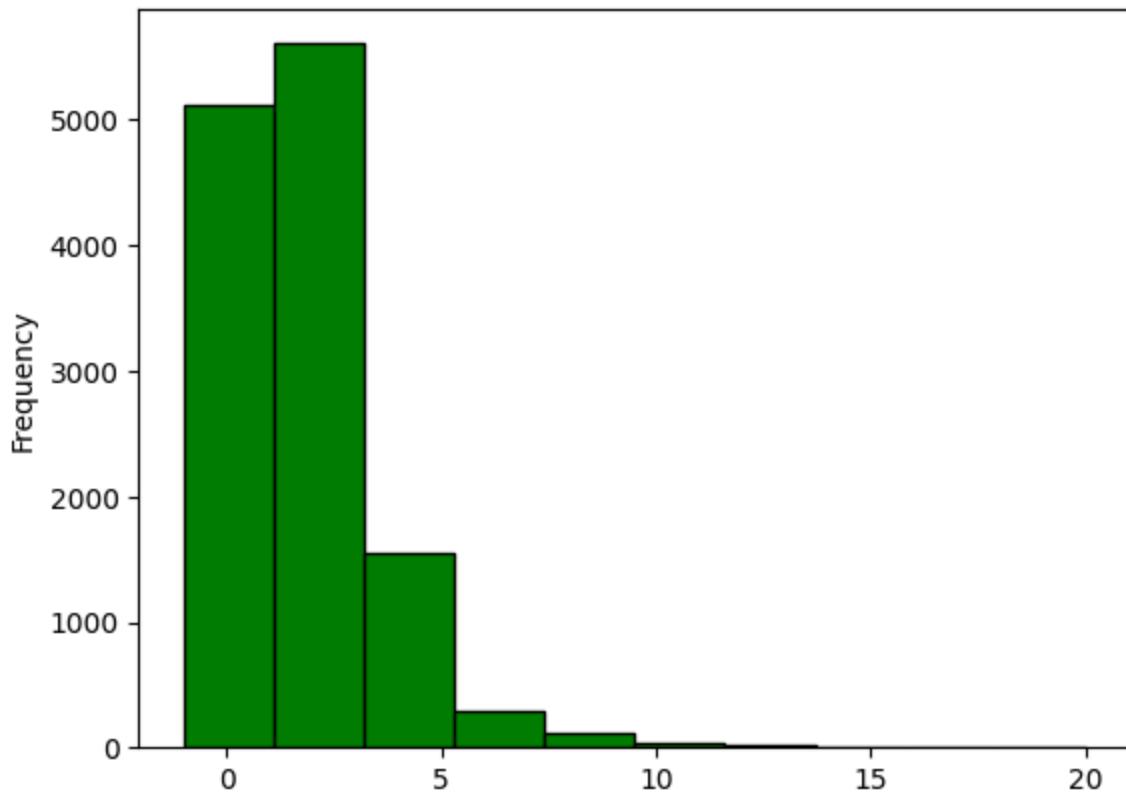
```
Out[97]: <Axes: xlabel='price'>
```



HISTOGRAM:--A histogram is used to visualize the distribution of a dataset.

```
In [96]: df['room_count'].plot(kind='hist',color='green',edgecolor='black')
```

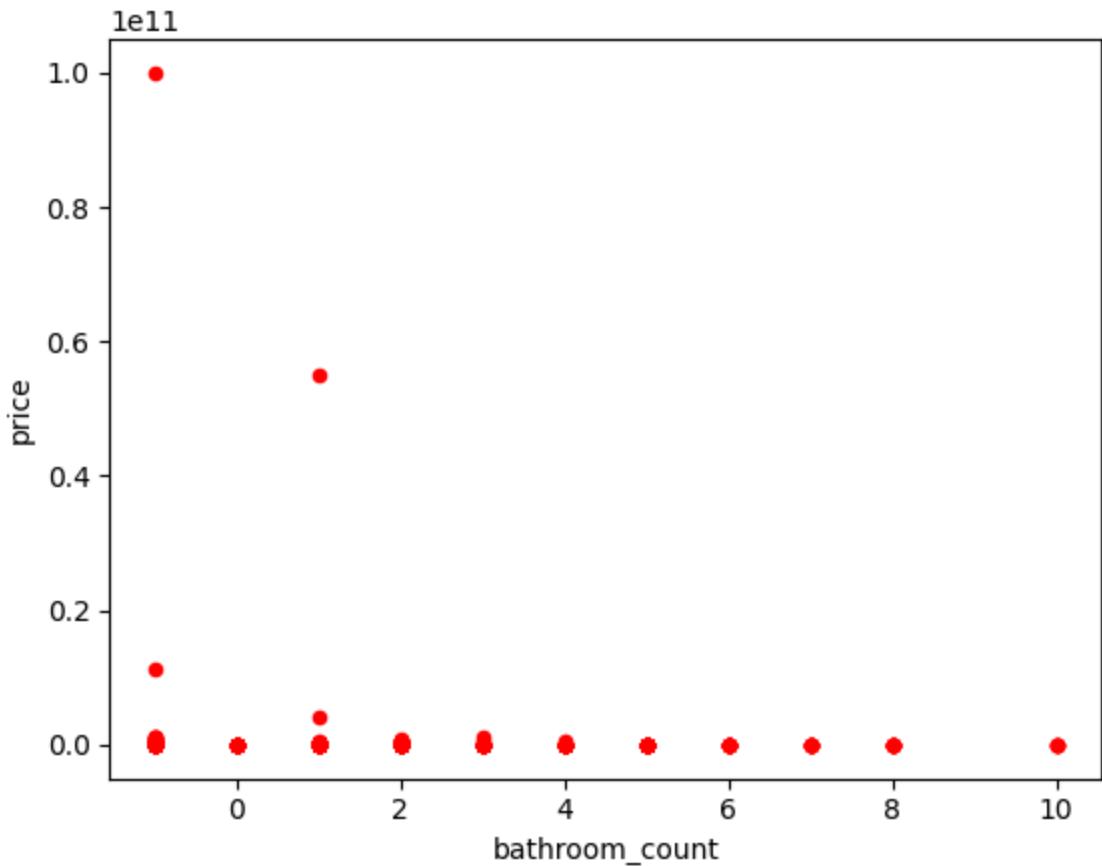
```
Out[96]: <Axes: ylabel='Frequency'>
```



SCATTER PLOT:-A scatter plot is useful for visualizing the relationship between two variables.

```
In [100]: df.plot(x='bathroom_count',y="price",kind='scatter', color='red')
```

```
Out[100]: <Axes: xlabel='bathroom_count', ylabel='price'>
```



In [ ]: