exploratory

October 21, 2025

1 Predictive Modeling of Property Sale Requisitions

1.1 Exploratory notebook

This notebook contains an initial exploration of the dataset, with the aim of generating and identifying the different data and links needed to achieve the objective.

1.1.1 Necessary libraries imports

[16]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import seaborn as sns

from IPython.display import display
```

1.1.2 Load datasets (2)

[13]:
```
# Source path
data_path = "../data/raw/"

# Create the two dataframes from the csv

df_poursuites = pd.read_csv(os.path.join(data_path,␣
 ↪"06_02_conjoncture_poursuites.csv"),sep=";", encoding="utf-8")
df_vacants = pd.read_csv(os.path.join(data_path,␣
 ↪"09_03_log_vacants_taux_des_1975.csv"),sep=";",encoding="utf-8")

# Print first 10 rows
display(df_poursuites.head(n=10))
display(df_vacants.head(n=10))
```

	Date	District Id	District	Bezirk	Réquisitions de poursuite
0	2025-07	1001	Broye	Broye	1328
1	2025-06	1001	Broye	Broye	1319
2	2025-05	1001	Broye	Broye	1589
3	2025-01	1001	Broye	Broye	812
4	2024-08	1001	Broye	Broye	1314

```
5  2024-05         1001    Broye  Broye                      1128
6  2023-11         1001    Broye  Broye                      1055
7  2023-10         1001    Broye  Broye                      1144
8  2023-09         1001    Broye  Broye                      1752
9  2023-07         1001    Broye  Broye                       926

   Réquisitions de continuer la poursuite  Réquisitions de vente  \
0                                     883                     4
1                                    1257                     8
2                                    1115                    17
3                                    1547                    11
4                                    1611                    15
5                                    1150                    33
6                                    1247                    23
7                                    1470                     2
8                                     996                     7
9                                     691                     5

   Actes de défaut de biens
0                       353
1                       638
2                       506
3                       495
4                       526
5                       543
6                       608
7                       759
8                       656
9                       354

   Année  District id District Bezirk  Commune id hist.  Commune id  \
0   1975         1001    Broye  Broye             12055        2022
1   1975         1001    Broye  Broye             12549        2035
2   1975         1001    Broye  Broye             12991        2043
3   1975         1002    Glâne  Glane             12113        2079
4   1975         1002    Glâne  Glane             12611        2067
5   1975         1002    Glâne  Glane             14103        2063
6   1975         1002    Glâne  Glane             14137        2114
7   1975         1002    Glâne  Glane             14476        2087
8   1975         1002    Glâne  Glane             14477        2099
9   1975         1002    Glâne  Glane             14480        2113

                Commune  Parc de logements au 31 décembre  \
0             Gletterens                                69
1                Nuvilly                                87
2                  Sévaz                                27
3             Grangettes                                39
4           Le Châtelard                               111
```

```
5         Billens-Hennens                                123
6           Villorsonnens                                241
7           Mézières (FR)                                175
8                Siviriez                                416
9  Vuisternens-dev-Romont                                437

   Nombre de logements vacants  Taux de logements vacants en %
0                            0                            0.00
1                            0                            0.00
2                            0                            0.00
3                            0                            0.00
4                            2                            1.80
5                            0                            0.00
6                            0                            0.00
7                            0                            0.00
8                            8                            1.92
9                            7                            1.60
```

[34]:
```
# Print column infos
display(df_poursuites.info())
display(df_vacants.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2667 entries, 0 to 2666
Data columns (total 8 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Date                                   2667 non-null   object
 1   district_id                            2667 non-null   int64
 2   District                               2667 non-null   object
 3   Bezirk                                 2667 non-null   object
 4   Réquisitions de poursuite              2667 non-null   int64
 5   Réquisitions de continuer la poursuite 2667 non-null   int64
 6   Réquisitions de vente                  2667 non-null   int64
 7   Actes de défaut de biens               2667 non-null   int64
dtypes: int64(5), object(3)
memory usage: 166.8+ KB
```

```
None
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6050 entries, 0 to 6049
Data columns (total 10 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Année                           6050 non-null   int64
 1   district_id                     6050 non-null   int64
 2   District                        6050 non-null   object
 3   Bezirk                          6050 non-null   object
```

```
 4   Commune id hist.                  6050 non-null   int64
 5   Commune id                        6050 non-null   int64
 6   Commune                           6050 non-null   object
 7   Parc de logements au 31 décembre  6050 non-null   int64
 8   Nombre de logements vacants       6050 non-null   int64
 9   Taux de logements vacants en %    6050 non-null   float64
dtypes: float64(1), int64(6), object(3)
memory usage: 472.8+ KB
```

None

[14]:
```
# Print column stats infos (overview of data)
display(df_poursuites.describe())
display(df_vacants.describe())
```

```
       District Id  Réquisitions de poursuite  \
count  2667.000000                2667.000000
mean   1004.000000                1169.576678
std       2.000375                1031.917576
min    1001.000000                   0.000000
25%    1002.000000                 556.000000
50%    1004.000000                 783.000000
75%    1006.000000                1269.500000
max    1007.000000                6341.000000

       Réquisitions de continuer la poursuite  Réquisitions de vente  \
count                              2667.000000            2667.000000
mean                                949.199850              28.539183
std                                 867.930693              29.045575
min                                   0.000000               0.000000
25%                                 437.000000              10.000000
50%                                 627.000000              19.000000
75%                                1023.500000              36.000000
max                                5297.000000             344.000000

       Actes de défaut de biens
count               2667.000000
mean                 460.641545
std                  467.553270
min                    0.000000
25%                  181.000000
50%                  303.000000
75%                  543.500000
max                 2800.000000

            Année  District id  Commune id hist.   Commune id  \
count  6050.00000  6050.000000       6050.000000  6050.000000
mean   2000.34000  1003.710744      13273.000000  2183.347107
std      14.66765     1.792870       1716.543371    95.569891
```

```
min    1975.00000  1001.000000      10162.000000  2008.000000
25%    1988.00000  1002.000000      11978.000000  2115.000000
50%    2000.50000  1004.000000      13080.000000  2177.000000
75%    2013.00000  1005.000000      14481.000000  2271.000000
max    2025.00000  1007.000000      16662.000000  2338.000000

       Parc de logements au 31 décembre  Nombre de logements vacants  \
count                        6050.000000                  6050.000000
mean                          887.422975                    10.694545
std                          1890.289404                    34.224023
min                            15.000000                     0.000000
25%                           235.000000                     0.000000
50%                           430.500000                     3.000000
75%                           844.000000                     9.000000
max                         22765.000000                   809.000000

       Taux de logements vacants en %
count                      6050.000000
mean                          1.091294
std                           1.448554
min                           0.000000
25%                           0.000000
50%                           0.660000
75%                           1.490000
max                          15.150000
```

The pandas describe() method provides a statistical summary of the numerical variables in a dataset, including count, mean, standard deviation, and minimum and maximum values.

- In the case of the Poursuites dataset, the statistics show strong variability between districts and over time, with some observations equal to zero and others very high, indicating possible outliers or significant economic fluctuations.
- For the Logements vacants dataset, the values describe a broad period (1975–2025) with an average vacancy rate of around 1%, consistent with the Swiss reality, but with some outliers above 10%.

1.1.3 Observations

Because it brings together complementary viewpoints on the same economic reality, combining the two datasets is beneficial. While the Vacants dataset offers structural information about housing market conditions, the Poursuites dataset uses monthly debt collection activity to capture short-term financial distress. By linking them, it is possible to analyze how financial tension and real estate dynamics interact across the different districts and time.

Note that the two datasets have different temporal granularities, where: - The **poursuites** dataset is monthly, with one row for each month $\times$ district. - The **vacants** dataset is annual, with one row for each year $\times$ district.

The common key district_id and the corresponding year will be used to merge the two datasets.

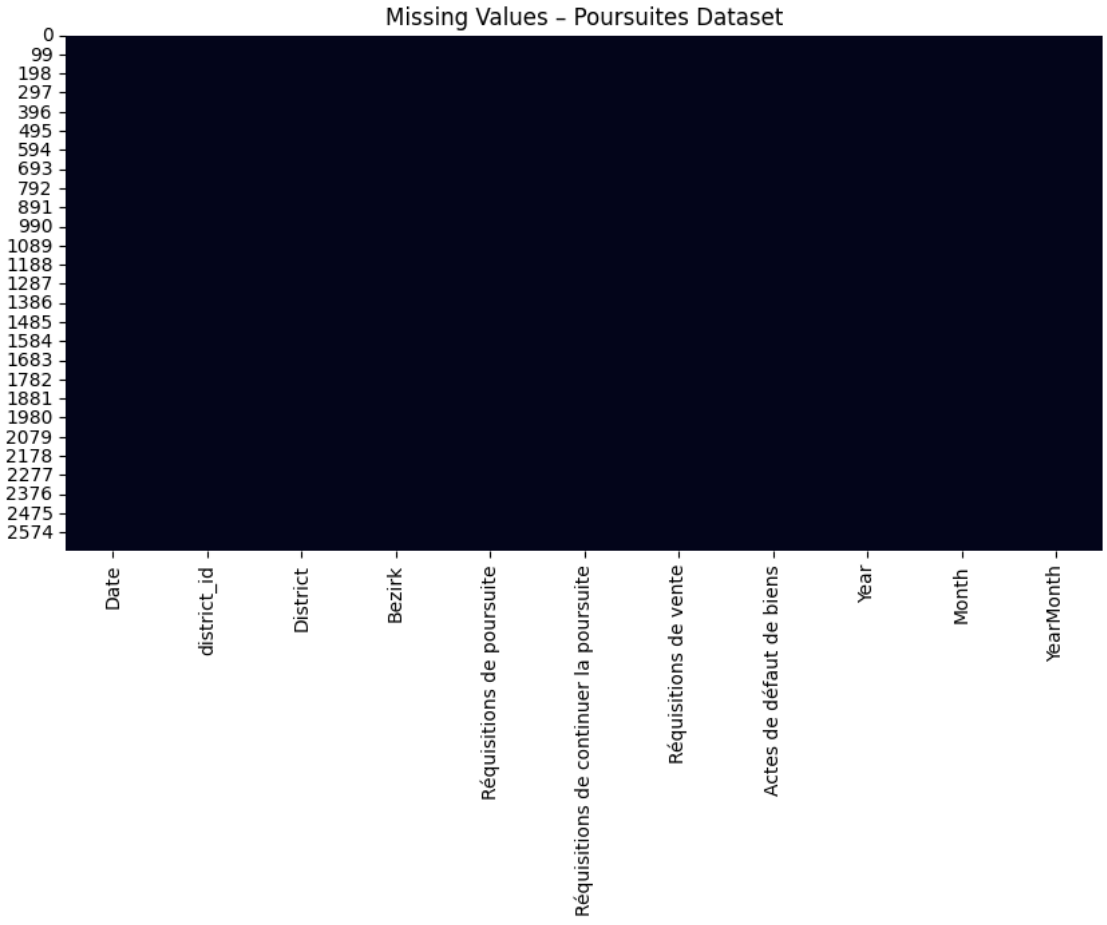
While maintaining the original temporal resolution of both sources, this method ensures that each monthly observation inherits the contextual information of its district and year, linking the monthly financial indicators of the Poursuites dataset with the annual vacancy data of the Vacants dataset.

1.1.4 Checking missing values

[63]:
```
import matplotlib.pyplot as plt

# Plot quantity of missing values in columns (white = missing value)
plt.figure(figsize=(10,5))
sns.heatmap(df_poursuites.isnull(), cbar=False)
plt.title("Missing Values - Poursuites Dataset")
plt.show()
display(df_poursuites.isna().sum())

plt.figure(figsize=(10,5))
sns.heatmap(df_vacants.isnull(), cbar=False)
plt.title("Missing Values - Vacants Dataset")
plt.show()
display(df_vacants.isna().sum())
```



```
Date                                    0
district_id                             0
District                                0
Bezirk                                  0
Réquisitions de poursuite               0
Réquisitions de continuer la poursuite  0
Réquisitions de vente                   0
Actes de défaut de biens                0
Year                                    0
Month                                   0
YearMonth                               0
dtype: int64
```


```
Année                   0
district_id             0
District                0
Bezirk                  0
```

```
Commune id hist.                    0
Commune id                          0
Commune                             0
Parc de logements au 31 décembre    0
Nombre de logements vacants         0
Taux de logements vacants en %      0
Date                                0
Year                                0
Month                               0
dtype: int64
```

The two datasets do not contain any missing data, so no specific processing or replacement will be necessary.

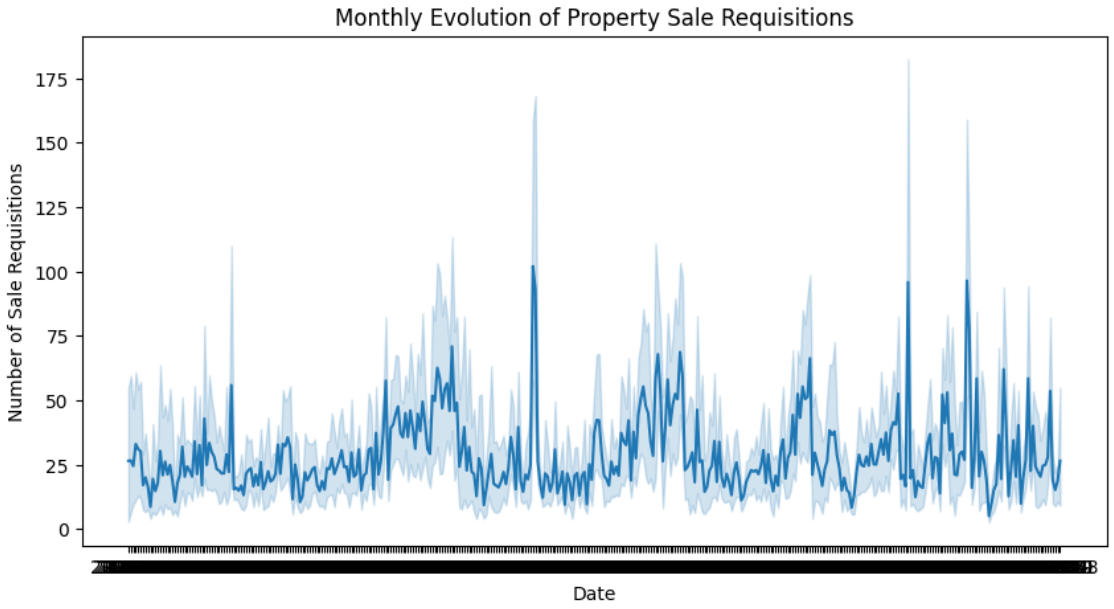
1.1.5 Evolution of monthly sale requisitions across all districts

[22]:
```
plt.figure(figsize=(10,5))

# Plot the monthly trend of sale requisitions over time
sns.lineplot(data=df_poursuites, x="Date", y="Réquisitions de vente")

# Add plot titles
plt.title("Monthly Evolution of Property Sale Requisitions")
plt.xlabel("Date")
plt.ylabel("Number of Sale Requisitions")


plt.show()
```



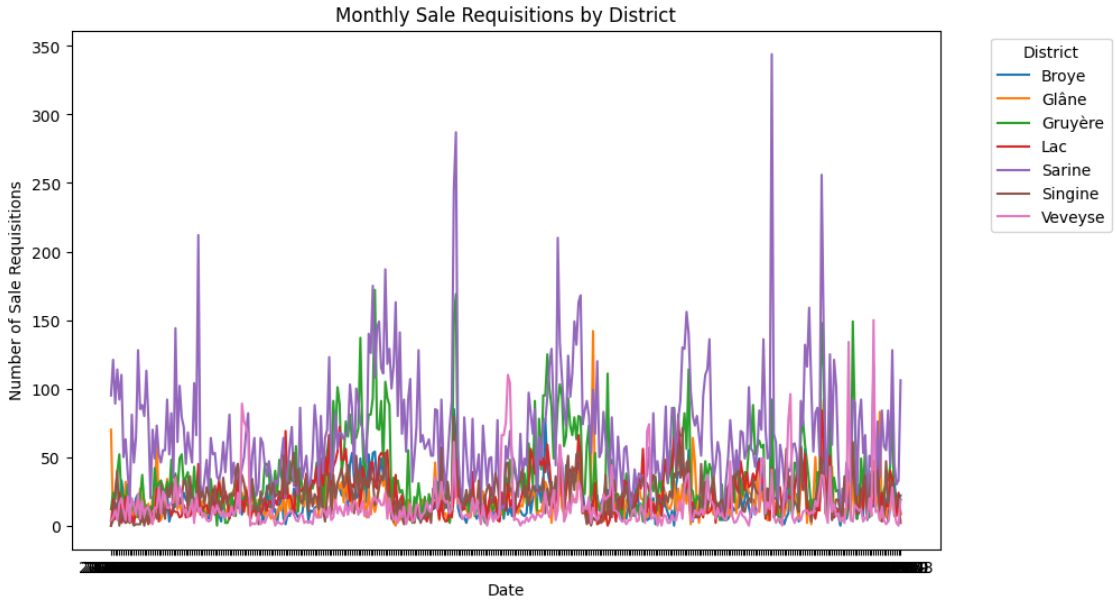
1.1.6 Compare trends across districts

[23]:
```
plt.figure(figsize=(10,6))

# Plot the monthly sale requisitions for each district
sns.lineplot(data=df_poursuites, x="Date", y="Réquisitions de vente",␣
 ↪hue="District")

# Add plot titles
plt.title("Monthly Sale Requisitions by District")
plt.xlabel("Date")
plt.ylabel("Number of Sale Requisitions")
plt.legend(title="District", bbox_to_anchor=(1.05, 1), loc="upper left") #␣
 ↪Position the legend outside the plot area

plt.show()
```



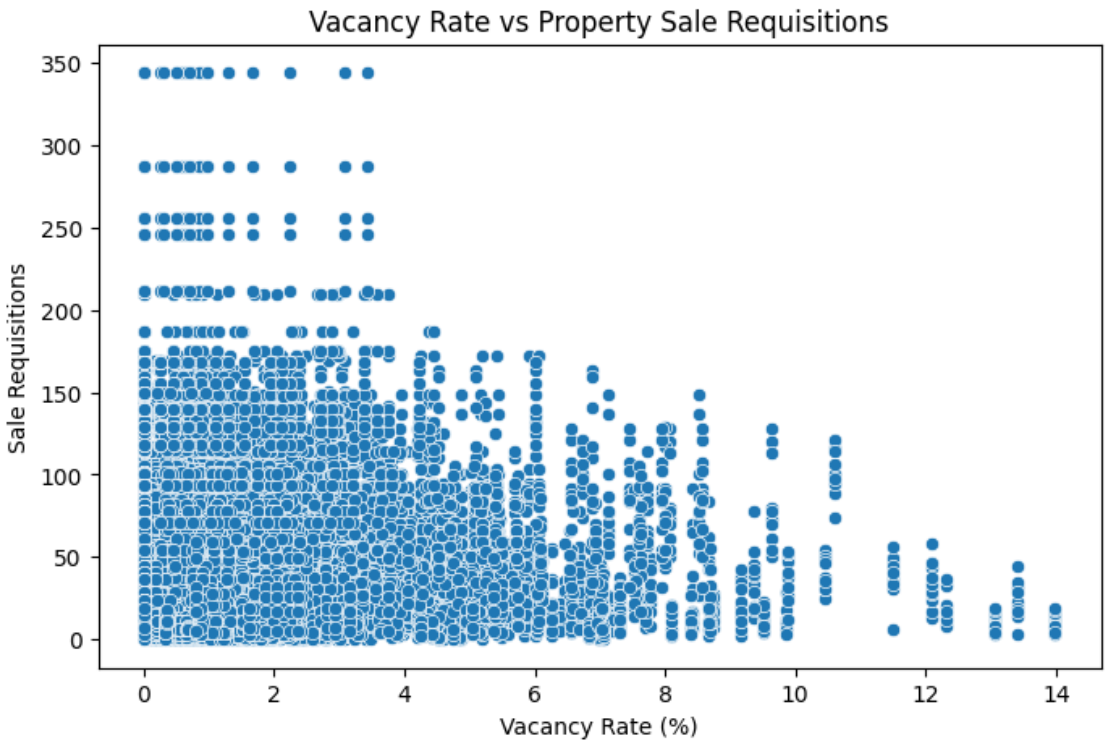
[64]:
```
plt.figure(figsize=(8,5))

# Plot the relationship between vacancy rate and sale requisitions
sns.scatterplot(data=df_merged, x='Taux de logements vacants en %',␣
 ↪y='Réquisitions de vente') # scatter --> see distributions

# Add plot titles
plt.title("Vacancy Rate vs Property Sale Requisitions")
```

```
plt.xlabel("Vacancy Rate (%)")
plt.ylabel("Sale Requisitions")

plt.show()
```



[65]:
```
# Extract the year from the Date column
df_poursuites['Year'] = df_poursuites['Date'].dt.year

# Total number of sales requisitions per year.
yearly = df_poursuites.groupby('Year')['Réquisitions de vente'].sum().
  ↪reset_index()

plt.figure(figsize=(10,5))
# Plot the total number of sale requisitions by year
sns.lineplot(data=yearly, x='Year', y='Réquisitions de vente', marker='o')

# Add plot titles
plt.title("Annual Evolution of Property Sale Requisitions")
plt.xlabel("Year")
plt.ylabel("Total Sale Requisitions")

plt.show()
```

1.1.7 Conclusions

The exploratory analysis reveals consistent and complete data across both sources, with no missing values. Temporal visualizations highlight cyclical patterns in property sale requisitions and clear differences across districts, reflecting local economic variability. The yearly trend suggests a gradual increase in sale requisitions, while the comparison with housing vacancy rates indicates a weak inverse relationship between housing market stability and financial distress. These findings confirm the dataset's suitability for time-series modeling to predictive tasks.