Tracking the Market: A Data-Driven Look at House Prices in Kungälv

25/5/2025

Project: Real Estate Data Mining and Visualization from Hemnet

Project Overview

In this project, I conducted a complete data extraction and analysis pipeline focused on the Swedish real estate market. I scraped and parsed HTML files containing housing data from Hemnet ,Sweden's leading property listing platform with the objective of generating structured insights from unstructured web data. The dataset, which included residential sale records from Kungälv as of October 2023, was originally provided in a compressed archive `kungalv_slutpriser.tar.gz`.

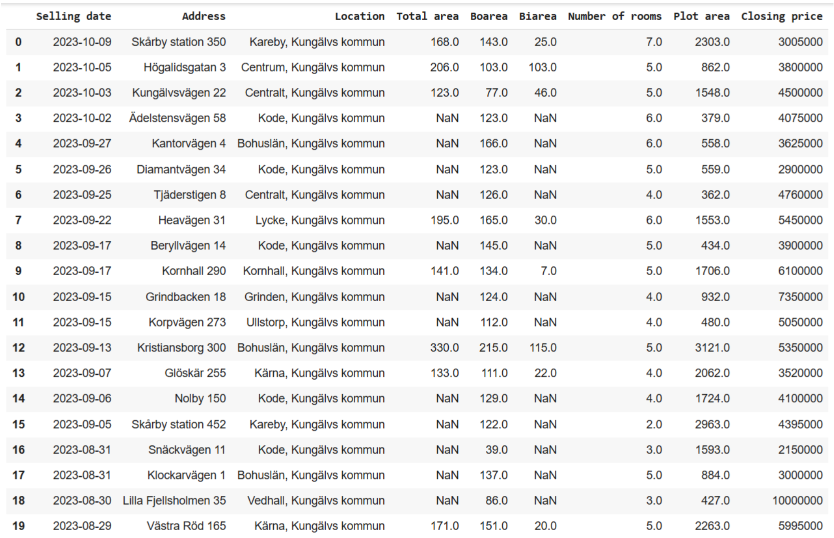
Using Python and Beautiful Soup, I extracted detailed attributes from each real estate listing, processed them into a structured format, and created a cleaned CSV file for subsequent data analysis.

Data Extraction and Preprocessing

Each listing in the HTML documents contained the following key data points:

- **Sale Date:** e.g., `Såld 28 oktober 2023`
- **Address or Plot Name:** e.g., `Strömgatan 4`
- **Location:** e.g., `Ytterby, Kungälvs kommun`
- **Area:** e.g., `105+10 m² (boarea + biarea)`
- **Number of Rooms:** e.g., `5 rum`
- **Plot Size:** e.g., `972 m² tomt`
- **Final Sale Price:** e.g., `5 750 000 kr`

Missing data points were handled gracefully and left blank as per the data integrity guidelines. The resulting CSV was used for exploratory data analysis in the second phase of the project.



	Selling date	Address	Location	Total area	Boarea	Biarea	Number of rooms	Plot area	Closing price
0	2023-10-09	Skårby station 350	Kareby, Kungälvs kommun	168.0	143.0	25.0	7.0	2303.0	3005000
1	2023-10-05	Högalidsgatan 3	Centrum, Kungälvs kommun	206.0	103.0	103.0	5.0	862.0	3800000
2	2023-10-03	Kungälvsvägen 22	Centralt, Kungälvs kommun	123.0	77.0	46.0	5.0	1548.0	4500000
3	2023-10-02	Ädelstensvägen 58	Kode, Kungälvs kommun	NaN	123.0	NaN	6.0	379.0	4075000
4	2023-09-27	Kantorvägen 4	Bohuslän, Kungälvs kommun	NaN	166.0	NaN	6.0	558.0	3625000
5	2023-09-26	Diamantvägen 34	Kode, Kungälvs kommun	NaN	123.0	NaN	5.0	559.0	2900000
6	2023-09-25	Tjäderstigen 8	Centralt, Kungälvs kommun	NaN	126.0	NaN	4.0	362.0	4760000
7	2023-09-22	Heavägen 31	Lycke, Kungälvs kommun	195.0	165.0	30.0	6.0	1553.0	5450000
8	2023-09-17	Beryllvägen 14	Kode, Kungälvs kommun	NaN	145.0	NaN	5.0	434.0	3900000
9	2023-09-17	Kornhall 290	Kornhall, Kungälvs kommun	141.0	134.0	7.0	5.0	1706.0	6100000
10	2023-09-15	Grindbacken 18	Grinden, Kungälvs kommun	NaN	124.0	NaN	4.0	932.0	7350000
11	2023-09-15	Korpvägen 273	Ullstorp, Kungälvs kommun	NaN	112.0	NaN	4.0	480.0	5050000
12	2023-09-13	Kristiansborg 300	Bohuslän, Kungälvs kommun	330.0	215.0	115.0	5.0	3121.0	5350000
13	2023-09-07	Glöskär 255	Kärna, Kungälvs kommun	133.0	111.0	22.0	4.0	2062.0	3520000
14	2023-09-06	Nolby 150	Kode, Kungälvs kommun	NaN	129.0	NaN	4.0	1724.0	4100000
15	2023-09-05	Skårby station 452	Kareby, Kungälvs kommun	NaN	122.0	NaN	2.0	2963.0	4395000
16	2023-08-31	Snäckvägen 11	Kode, Kungälvs kommun	NaN	39.0	NaN	3.0	1593.0	2150000
17	2023-08-31	Klockarvägen 1	Bohuslän, Kungälvs kommun	NaN	137.0	NaN	5.0	884.0	3000000
18	2023-08-30	Lilla Fjellsholmen 35	Vedhall, Kungälvs kommun	NaN	86.0	NaN	3.0	427.0	10000000
19	2023-08-29	Västra Röd 165	Kärna, Kungälvs kommun	171.0	151.0	20.0	5.0	2263.0	5995000

Figure 1: Extracted CSV: First 20 rows of structured housing data.

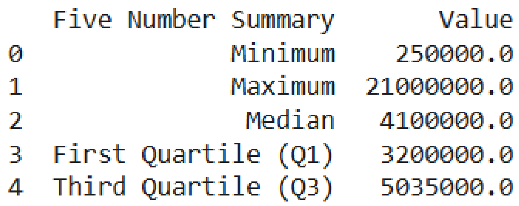
Data Analysis: Housing Market Insights for 2022

The second phase focused on analyzing housing transactions from 2022 using the extracted dataset. Below are key steps and insights:

- Generated the Five-number summary of sale prices: Minimum, Q1, Median, Q3, Maximum.
- Created a histogram to examine the price distribution using the Square Root Rule for bin selection.
- Constructed scatter plots to explore correlations between living area and price.
- Added a color dimension to the scatter plot based on the number of rooms to visualize multivariate trends.
- Interpreted visual trends and outliers to understand pricing behavior.

Five-Number Summary

The summary statistics of sale prices provide a foundational view of the market spread.



```
     Five Number Summary       Value
0                Minimum    250000.0
1                Maximum  21000000.0
2                 Median   4100000.0
3   First Quartile (Q1)   3200000.0
4   Third Quartile (Q3)   5035000.0
```

Figure 2: Five-number summary of closing prices (2022).

Histogram of Closing Prices

I applied the Square Root Rule:

$$k = \lceil \sqrt{n} \rceil$$

to determine the number of histogram bins, where $n = 190$ (data points) and $k \approx 14$. This allowed for effective visualization of the price distribution.

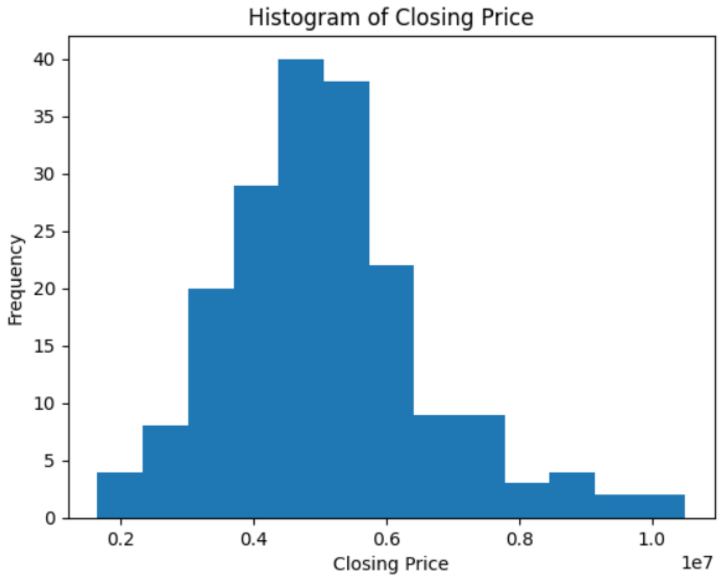


Figure 3: Histogram showing distribution of closing prices (2022).

Scatter Plot: Price vs Boarea

This visualization explored the relationship between house size (boarea) and sale price.

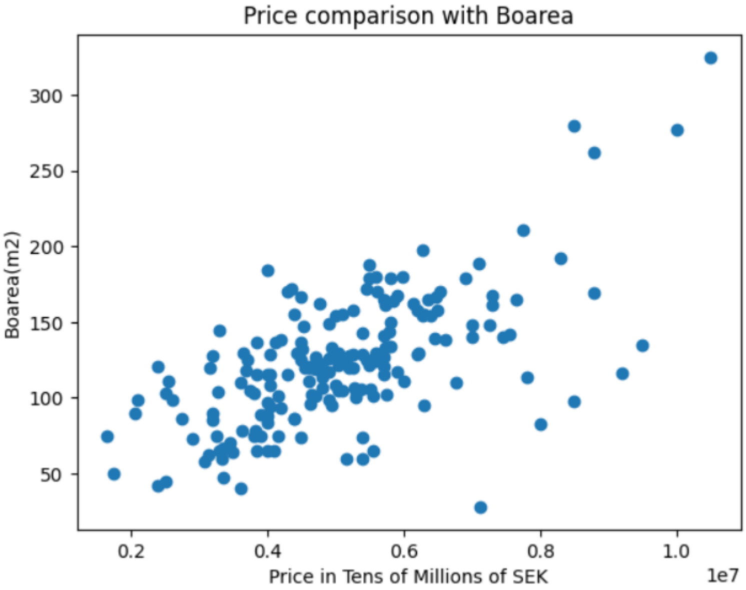


Figure 4: Scatter plot: Price vs Living Area (boarea).

Scatter Plot with Room-Based Color Coding

To add dimensionality, I color-coded data points by the number of rooms.

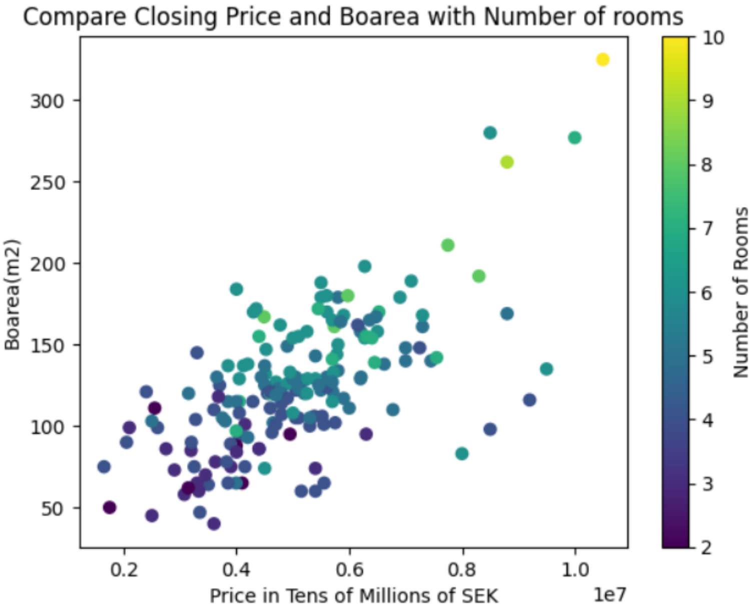


Figure 5: Color-coded scatter plot: Price vs Area by Room Count.

Insights and Observations

- There is a strong positive correlation between living area and final sale price.
- Properties with more rooms (lighter shades) tend to be both larger and more expensive.
- Smaller, darker-colored homes typically fall into the lower price and area ranges.
- Most homes are clustered between 2 to 6 million SEK and 50–200 m^2.
- High-end properties are rare and form outliers in the distribution.

Tools and Technologies Used

- **Python (Beautiful Soup, pandas, matplotlib, seaborn)**
- **HTML Parsing and Web Data Handling**
- **Exploratory Data Analysis (EDA)**
- **Data Visualization and Statistical Summarization**

Conclusion

This project combined data scraping, wrangling, and exploratory visualization to uncover market dynamics in the real estate sector. It not only strengthened my skills in data engineering and analysis but also gave me deeper insight into interpreting real-world datasets in housing and economics.

Appendix

Problem 1 - Scraping House Prices

Part 1: Setting up

```
#Installing BeautifulSoup4
!pip install beautifulsoup4

#Importing necessary packages
from bs4 import BeautifulSoup  #importing beautifulsoup
import tarfile
import re
import pandas as pd
import os
from datetime import datetime

#Connect to drive to be able to import files
from google.colab import drive
drive.mount('/content/drive')

#File path
file_path = '/content/drive/MyDrive/DAT565 - Data Science and AI/Assignment 2/kungalv_slutpriser.tar.gz'
```

Part 2: Opening and reading files

```
#Open the tarfile we have recieved for this assignment
with tarfile.open(file_path, "r") as tf:
    #List all the files
    file_names = tf.getnames()
    #Extracting files to current working directory
    tf.extractall()

print(*file_names, sep="\n")
```

Part 3: Extracting wanted data

```
#Empty list for all the listings
all_listings = []

#For-loop iterating through the files in the folder in the working directory
for files in sorted(os.listdir('kungalv_slutpriser')):
        #Opening and reading the html files
        with open(os.path.join('kungalv_slutpriser', files), 'r') as f:
            html = f.read()
            #Using beautiful soup to parse the data
            soup = BeautifulSoup(html, 'html.parser')

            #Finding all matching elements for sold results and adding to list
            for cell in soup.find_all('li',class_='sold-results__normal-hit'):
                all_listings.append(cell)

print("Number of listings:", len(all_listings))
print("First item:", all_listings[0])
```

```
#Function for converting the date from a Swedish string format to yyyy-mm-dd

def date_conversion(date_swe):
# Replace Swedish month names with English equivalents for parsing
  date_swe = date_swe.replace("januari", "January") \
                   .replace("februari", "February") \
                   .replace("mars", "March") \
                   .replace("april", "April") \
                   .replace("maj", "May") \
                   .replace("juni", "June") \
                   .replace("juli", "July") \
                   .replace("augusti", "August") \
                   .replace("september", "September") \
                   .replace("oktober", "October") \
                   .replace("november", "November") \
                   .replace("december", "December")

# Parse the modified date string and format it to yyyy-mm-dd
  date_eng = datetime.strptime(date_swe, "%d %B %Y")
  formatted_date = date_eng.strftime("%Y-%m-%d")

  return formatted_date
```

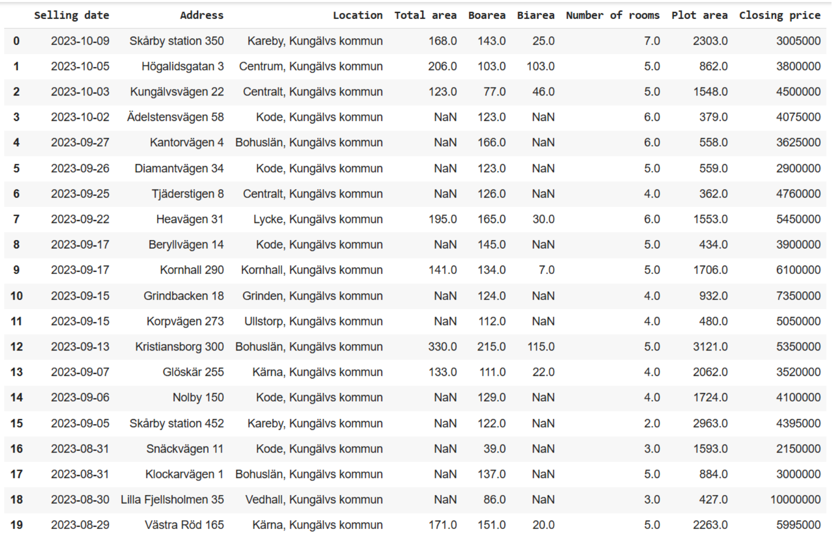
```
#Empty lists
boarea_list = []
biarea_list = []
total_area_list = []
nr_rooms_list = []

#Iterating through all listings
for listings in all_listings:

    #House areas and rooms (contains several things we want, area x2 and number of rooms)
    house_html = listings.find('div', {'class': 'sold-property-listing__subheading sold-property-
  listing__area'})
    #Finding childern/subcategories
    house_list_children = list(house_html.children)

    #If biarea exists
    if len(house_list_children) == 3:
      #Boarea
      boarea = house_list_children[0].text.strip().
  replace(',','.')
      boarea_float = float(boarea)
      #Biarea
      biarea = house_list_children[1].text.strip()
      biarea = re.findall(r'(\d+)', biarea)
      biarea = float(biarea[0])
      #Toatal area
      total_area = boarea_float + biarea
      #Number of rooms
      nr_rooms = house_list_children[2].text.strip()
      nr_rooms = re.findall(r"(\d+)\s*rum", nr_rooms)
      #If nr_room is NOT equal to empty string
      if nr_rooms != []:
        nr_rooms_float = float(nr_rooms[0])
      else:
        nr_rooms_float = None

    #If biarea doesn't exist
    else:
      #Boarea
      boarea = re.findall(r"(\d+)\s*m  ", house_html.
   text.strip())
      if boarea != []:
        boarea_float = float(boarea[0])
      else:
        boarea_float = None
      #Biarea
      biarea = None
      #Total area
      total_area = None
      #Rooms
      nr_rooms = house_html.text.strip()
      nr_rooms = re.findall(r"(\d+)\s*rum", nr_rooms)
      #If nr_room is NOT equal to empty string
      if nr_rooms != []:
        nr_rooms_float = float(nr_rooms[0])
      else:
        nr_rooms_float = None

    boarea_list.append(boarea_float)
    biarea_list.append(biarea)
    total_area_list.append(total_area)
```

```
    nr_rooms_list.append(nr_rooms_float)

print('boarea',boarea_list)
print('biarea',biarea_list)
print('total',total_area_list)
print('rooms',nr_rooms_list)
```

```
#Empty lists
plot_area_list = []
closing_price_list = []
selling_date_list = []
address_list = []
location_list = []

#Iterating through all listings
for listings in all_listings:

 #Plot area
    plot_area = listings.find('div', {'class': 'sold-
   property-listing__land-area'})
    if plot_area is not None:
        plot_area = int("".join(re.findall(r'(\d+)',
   plot_area.text.strip())))
    plot_area_list.append(plot_area)

    #Closing price
    closing_price = listings.find('span', {'class': '
   hcl-text hcl-text--medium'})
    closing_price = int("".join(re.findall(r'(\d+)',
   closing_price.text.strip())))
    closing_price_list.append(closing_price)

    #Selling date
    selling_date = listings.find('span', {'class': '
   hcl-label hcl-label--state hcl-label--sold-at'})
    selling_date = selling_date.text.strip().replace('
   S ld ','')
    #Using function date_conversion
    converted_date = date_conversion(selling_date)
    selling_date_list.append(converted_date)

    #Address
    address = listings.find('h2', {'class': 'sold-
   property-listing__heading qa-selling-price-title
   hcl-card__title'})
    address_text = address.text.strip()
    address_list.append(address_text)

    #Location
    location = listings.find('span', {'class': '
   property-icon property-icon--result’})
    location_text = location.next_sibling.strip()
    location_text = re.sub(r’\s+’, ’ ’, location_text)
    location_list.append(location_text)


print(’plot area’,plot_area_list)
print(’price’,closing_price_list)
print(’date’,selling_date_list)
print(’address’,address_list)
print(’location’,location_list)
```

Part 4: Summarizing into CSV file

```
data = pd.DataFrame({
    "Selling date": selling_date_list,
    "Address": address_list,
    "Location": location_list,
    "Total area": total_area_list,
    "Boarea": boarea_list,
    "Biarea": biarea_list,
    "Number of rooms": nr_rooms_list,
    "Plot area": plot_area_list,
    "Closing price": closing_price_list
})

data.to_csv(’Assignment2_Housingprices.csv’, index=
   False)

data.head(20)
```



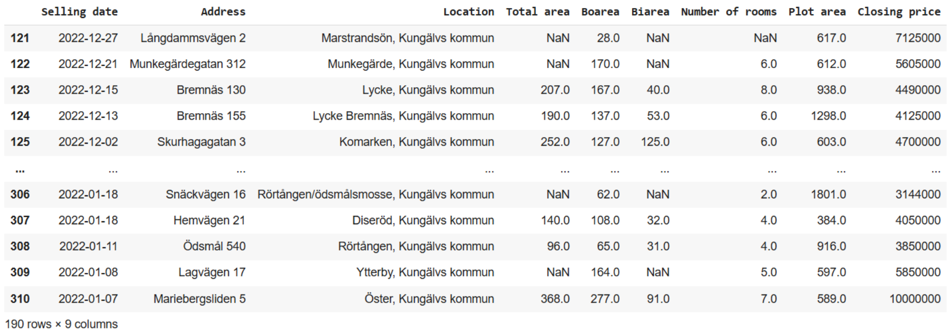
	Selling date	Address	Location	Total area	Boarea	Biarea	Number of rooms	Plot area	Closing price
0	2023-10-09	Skårby station 350	Kareby, Kungälvs kommun	168.0	143.0	25.0	7.0	2303.0	3005000
1	2023-10-05	Högalidsgatan 3	Centrum, Kungälvs kommun	206.0	103.0	103.0	5.0	862.0	3800000
2	2023-10-03	Kungälvsvägen 22	Centralt, Kungälvs kommun	123.0	77.0	46.0	5.0	1548.0	4500000
3	2023-10-02	Ädelstensvägen 58	Kode, Kungälvs kommun	NaN	123.0	NaN	6.0	379.0	4075000
4	2023-09-27	Kantorvägen 4	Bohuslän, Kungälvs kommun	NaN	166.0	NaN	6.0	558.0	3625000
5	2023-09-26	Diamantvägen 34	Kode, Kungälvs kommun	NaN	123.0	NaN	5.0	559.0	2900000
6	2023-09-25	Tjäderstigen 8	Centralt, Kungälvs kommun	NaN	126.0	NaN	4.0	362.0	4760000
7	2023-09-22	Heavägen 31	Lycke, Kungälvs kommun	195.0	165.0	30.0	6.0	1553.0	5450000
8	2023-09-17	Beryllvägen 14	Kode, Kungälvs kommun	NaN	145.0	NaN	5.0	434.0	3900000
9	2023-09-17	Kornhall 290	Kornhall, Kungälvs kommun	141.0	134.0	7.0	5.0	1706.0	6100000
10	2023-09-15	Grindbacken 18	Grinden, Kungälvs kommun	NaN	124.0	NaN	4.0	932.0	7350000
11	2023-09-15	Korpvägen 273	Ullstorp, Kungälvs kommun	NaN	112.0	NaN	4.0	480.0	5050000
12	2023-09-13	Kristiansborg 300	Bohuslän, Kungälvs kommun	330.0	215.0	115.0	5.0	3121.0	5350000
13	2023-09-07	Glöskär 255	Kärna, Kungälvs kommun	133.0	111.0	22.0	4.0	2062.0	3520000
14	2023-09-06	Nolby 150	Kode, Kungälvs kommun	NaN	129.0	NaN	4.0	1724.0	4100000
15	2023-09-05	Skårby station 452	Kareby, Kungälvs kommun	NaN	122.0	NaN	2.0	2963.0	4395000
16	2023-08-31	Snäckvägen 11	Kode, Kungälvs kommun	NaN	39.0	NaN	3.0	1593.0	2150000
17	2023-08-31	Klockarvägen 1	Bohuslän, Kungälvs kommun	NaN	137.0	NaN	5.0	884.0	3000000
18	2023-08-30	Lilla Fjellsholmen 35	Vedhall, Kungälvs kommun	NaN	86.0	NaN	3.0	427.0	10000000
19	2023-08-29	Västra Röd 165	Kärna, Kungälvs kommun	171.0	151.0	20.0	5.0	2263.0	5995000

Figure 6: First 20 rows of the CSV-file output

Problem 2 - Analyzing 2022 House Sales

Part 1: Setting up and filtering

```python
import matplotlib.pyplot as plt
from datetime import datetime
from tabulate import tabulate

# Load the dataset and filter 2022
df = pd.read_csv(’Assignment2_Housingprices.csv’)
df[’Selling date’] = pd.to_datetime(df[’Selling date’])
df_2022 = df[df[’Selling date’].dt.year == 2022]
df_2022
```



	Selling date	Address	Location	Total area	Boarea	Biarea	Number of rooms	Plot area	Closing price
121	2022-12-27	Långdammsvägen 2	Marstrandsön, Kungälvs kommun	NaN	28.0	NaN	NaN	617.0	7125000
122	2022-12-21	Munkegärdegatan 312	Munkegärde, Kungälvs kommun	NaN	170.0	NaN	6.0	612.0	5605000
123	2022-12-15	Bremnäs 130	Lycke, Kungälvs kommun	207.0	167.0	40.0	8.0	938.0	4490000
124	2022-12-13	Bremnäs 155	Lycke Bremnäs, Kungälvs kommun	190.0	137.0	53.0	6.0	1298.0	4125000
125	2022-12-02	Skurhagagatan 3	Komarken, Kungälvs kommun	252.0	127.0	125.0	6.0	603.0	4700000
...	...	...	...	...	...	...	...	...	...
306	2022-01-18	Snäckvägen 16	Rörtången/ödsmålsmosse, Kungälvs kommun	NaN	62.0	NaN	2.0	1801.0	3144000
307	2022-01-18	Hemvägen 21	Diseröd, Kungälvs kommun	140.0	108.0	32.0	4.0	384.0	4050000
308	2022-01-11	Ödsmål 540	Rörtången, Kungälvs kommun	96.0	65.0	31.0	4.0	916.0	3850000
309	2022-01-08	Lagvägen 17	Ytterby, Kungälvs kommun	NaN	164.0	NaN	5.0	597.0	5850000
310	2022-01-07	Mariebergsliden 5	Öster, Kungälvs kommun	368.0	277.0	91.0	7.0	589.0	10000000

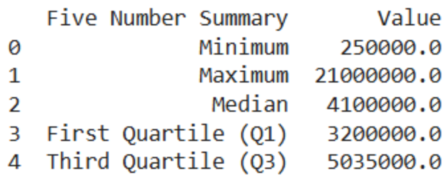
190 rows × 9 columns

Figure 7: Filtered the entries for the year 2022.

Part 2: Analyzing and plotting results

Five-number summary of closing prices

```python
#The five number summary of the closing prices
df[’Closing price’] = pd.to_numeric(df[’Closing price’])

#Calculating individual statistics
minimum = df[’Closing price’].min()
maximum = df[’Closing price’].max()
median = df[’Closing price’].median()
firstquartile = df[’Closing price’].quantile(0.25)
thirdquartile = df[’Closing price’].quantile(0.75)

stats = pd.DataFrame({
    ’Five Number Summary’: [’Minimum’, ’Maximum’, ’Median’, ’First Quartile (Q1)’, ’Third Quartile (Q3)’],
    ’Value’: [minimum, maximum, median, firstquartile, thirdquartile]
})

print(stats)
```



```
   Five Number Summary        Value
0              Minimum     250000.0
1              Maximum   21000000.0
2               Median    4100000.0
3  First Quartile (Q1)    3200000.0
4  Third Quartile (Q3)    5035000.0
```

Figure 8: Five number summary of closing prices 2022.

Histogram of the closing prices

```python
#Histogram of the closing prices - Square root rule
import math

plt.hist(df_2022['Closing price'], bins=(int(math.sqrt(190)) )) #Square root rule
plt.xlabel('Closing Price')
plt.ylabel('Frequency')
plt.title('Histogram of Closing Price')
plt.savefig("histogram.pdf")
plt.show()
```

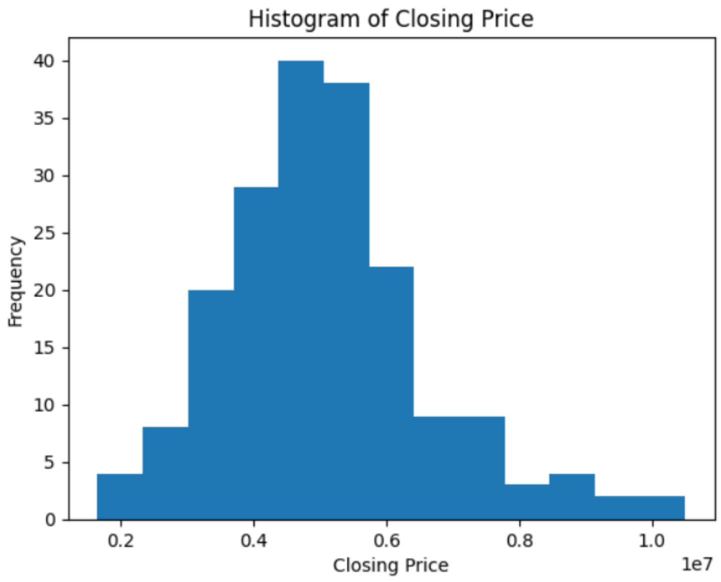


Figure 9: Histogram of closing prices 2022.

Scatter plot

```
#Scatter plot that shows the relationship of the
   closing price with the boarea of the house
ClosingPrice = df_2022['Closing price']
Boarea = df_2022['Boarea']
plt.scatter(ClosingPrice, Boarea)
plt.title("Price comparison with Boarea")
plt.xlabel("Price in Tens of Millions of SEK")
plt.ylabel("Boarea(m2)")
plt.savefig("scatterplot.pdf")
```

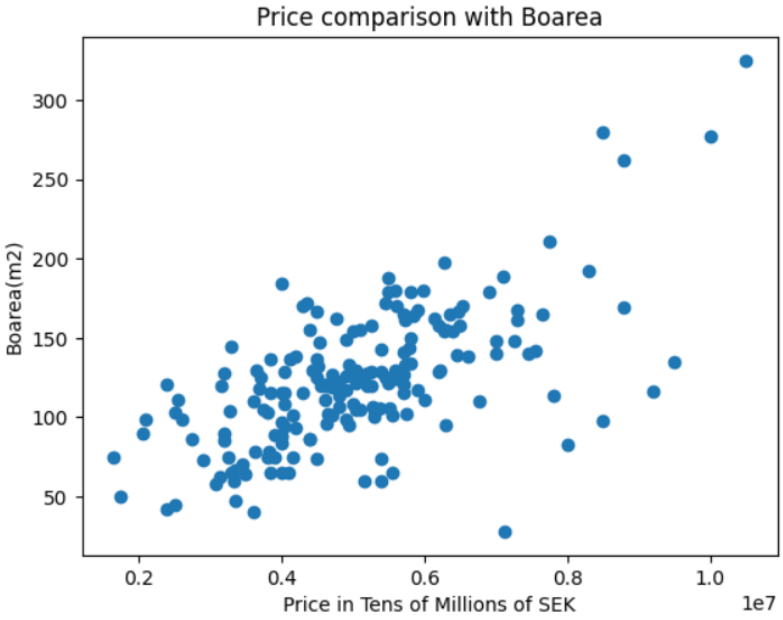


Figure 10: Scatter plot that shows the relationship of the closing price with the boarea of the house.

Colorizing the observations

```
#Colorizing the observations by the number of rooms in
    the house
ClosingPrice = df_2022['Closing price']
Boarea = df_2022['Boarea']
compare = plt.scatter(ClosingPrice, Boarea, c=df_2022[
    'Number of rooms'], cmap='viridis')
legend = plt.colorbar(compare)
legend.set_label("Number of Rooms")
plt.title("Compare Closing Price and Boarea with
   Number of rooms")
plt.xlabel("Price in Tens of Millions of SEK")
plt.ylabel("Boarea(m2)")
plt.savefig("scatterplot.pdf")
```

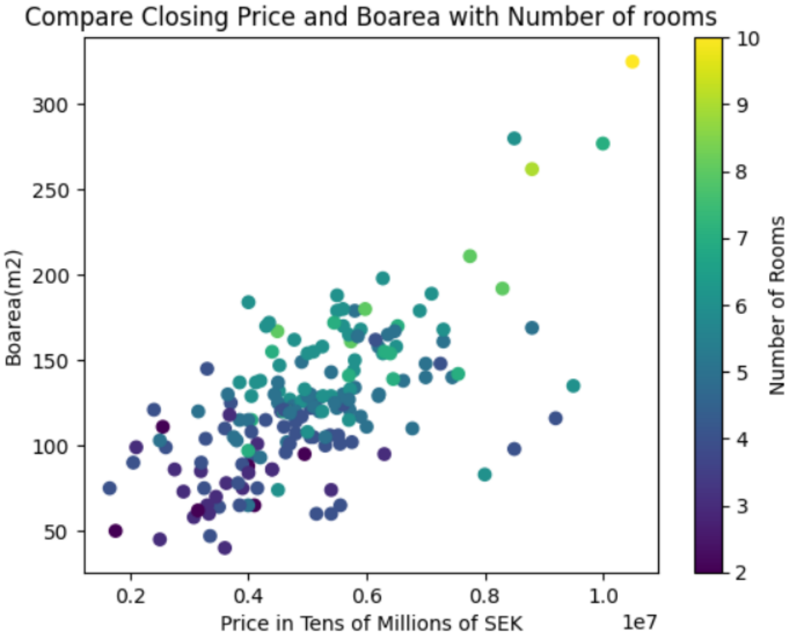


Figure 11: Colorizing the observations by the number of rooms in the house