DAT565/DIT407 Assignment 2

Connell Hagen
connellh@student.chalmers.se

Måns Redin
mansre@student.chalmers.se

2024-09-18

1 Problem 1: Web Scraping Hemnet

1.1 Introduction

Hemnet is a Swedish website produced to aid the housing market with buying and selling. In this problem the website was scraped, for data, using Python. The data was compiled into a dataframe of some of the features of each house. The, this data was later visualized and analyzed.

1.2 Methodology

In the scraping part, the main target was to extract data in the following categories:

- Date of sale
- Address
- Location of the estate
- Living area
- The number of rooms
- Area of the plot
- Closing price

The website was scraped using the Python library BeautifulSoup, which allowed the HTML code to be easier sifted through for information. The RegularExpression library was also used to help with extracting information from the HTML.

2 Problem 2: Plotting and Analyzing the Data

2.1 Five Number Summary of Closing Prices

In this part the five-number summary was requested to be calculated. The five-number summary consists of the minimum, maximum, medain, 1st and 3rd quartiles of the closing prices. The obtained data is in Table 1.

Minimum	1,650,000
1st Quartile	4,012,500
Median	5,000,000
3rd Quartile	5,795,000
Maximum	10,500,000

Table 1: 5 Number Summary for Closing Prices

2.2 Histogram of the Closing Prices

The closing of 2022 was then requested to be plotted in a histogram with the amplitude of the the amount being sold. Figure 1 show the data of the closing prices of 2022

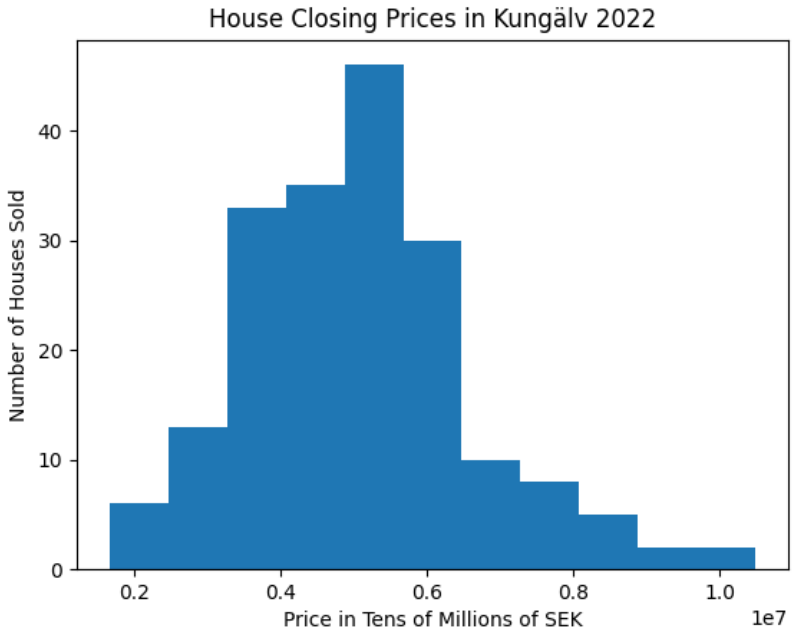

Figure 1: Closing Prices of Properties in Kungälv Municipality 2022

To make a good plot the bin width had to be considered. Equation 1, Scott's rule, is a relationship of how to calculate the bin width for a histogram.

$$h = \frac{3.49 \cdot \hat{\sigma}}{\sqrt[3]{n}} \tag{1}$$

2.3 Discussion

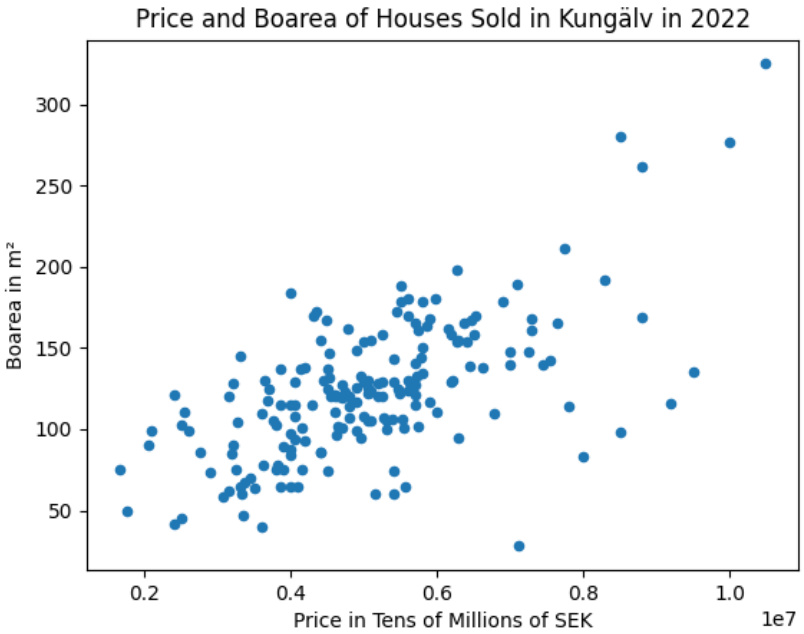

Figure 2: Prices and Boarea of Sold Houses in Kungälv

Figure 2 Shows a positive correlation between the cost of the house, and the total boarea that the house has. Overall, the correlation is somewhat weak, which shows that it really depends on the individual house what the relationship between price and area is. This could be due to factors like location, or how nice or new the house is.

Figure 3 show that there is correlation between the closing price and the amount of rooms, and it is a positive correlation. Also, the correlation between the boarea and the amount of rooms is a positive correlation, as well. There are some outliers which are more expensive, even though they have 4 or less rooms, as is shown with the orange dots on the graph. This could be due to similar factors as were discussed for the previous figure. The outliers in the lower price ranges are not as extreme, which may indicate a fairly standard minimum price for a house of any size.

A Code

Here is the code used for both scraping and plotting the data, we did this is 2 seperate scripts.

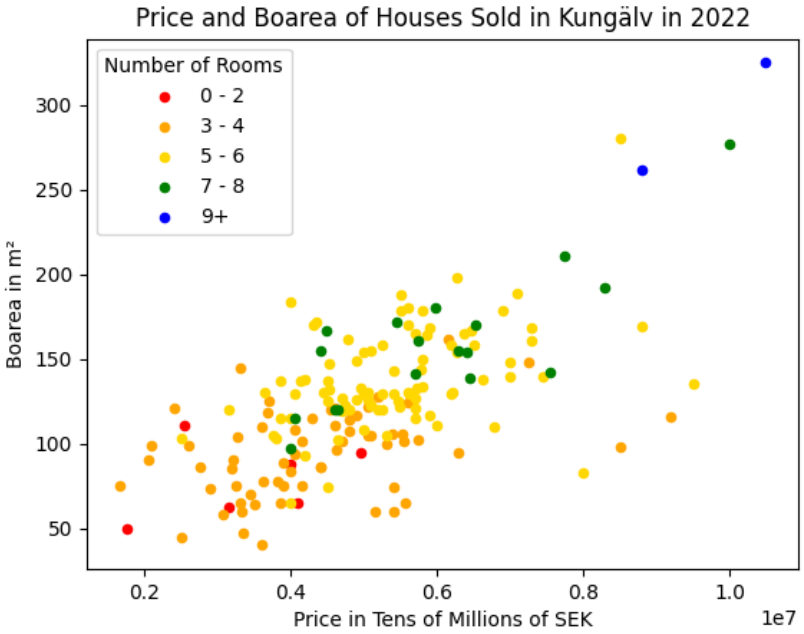

Figure 3: Prices and Boarea with Corresponding Number of Rooms

```
# Scraping the data
from bs4 import BeautifulSoup
import pandas as pd
import re

filenames = [];
for i in range(1, 41):
    if i < 10:
        filenames += [f"assignment_2/
            kungalv_slutpriser/kungalv_slutpris_page_0{
            i}.html"]
    else:
        filenames += [f"assignment_2/
            kungalv_slutpriser/kungalv_slutpris_page_{i
            }.html"]

values = []

for filename in filenames:
    with open(filename, 'r', encoding="utf-8") as f:
        html = f.read()

    soup = BeautifulSoup(html, 'html.parser')
```

```
20
21      for cell in soup.find_all('li', class_ = 'sold-
            results__normal-hit'):
22          card = cell.a.div
23
24          date_sold = card.find('span', class_ = 'hcl-
                label--sold-at').text.strip()
25          sold_tokens = date_sold.split('␣')
26          date_sold = sold_tokens[1] + "␣" + sold_tokens
                [2] + "␣" + sold_tokens[3]
27
28          address = card.find('h2', class_ = 'sold-
                property-listing__heading').text.strip()
29
30          location = card.find('div', class_ = 'sold-
                property-listing__location').div.text.strip
                ()[24:]
31          location_tokens = location.split(',')
32
33          for i in range (0, len(location_tokens)):
34              location_tokens[i] = re.search("(\\w+.*)",
                    location_tokens[i]).group(1)
35          location = location_tokens[0]
36          for i in range(1, len(location_tokens)):
37              location += (",␣" + location_tokens[i])
38
39          living_area_and_rooms = card.find('div',
                class_ = 'sold-property-listing__area')
40          contains_m2 = living_area_and_rooms.text.strip
                ().find('m  ')
41          laar_tokens = living_area_and_rooms.text.strip
                ().split('m  ')
42          living_area = ""
43
44
45          if len(laar_tokens) == 2:
46              living_areas_tokens = laar_tokens[0].split
                    ('+')
47              if (len(living_areas_tokens) == 1):
48                  living_area = re.search("(\\d+)",
                        living_areas_tokens[0]).group(1)
49              else:
50                  living_area = re.search("(\\d+)",
                        living_areas_tokens[0]).group(1) +
                        "+" + \
51                                  re.search("(\\d+).*$",
                                        living_areas_tokens[1])
                                        .group(1)
52
```

```
53              rooms = re.search('(\\d+)\xa0rum',
                    laar_tokens[1])
54              if (rooms == None):
55                  rooms = ""
56              else:
57                  rooms = rooms.group(1)
58
59          elif len(laar_tokens) != 2 and contains_m2 !=
                -1:
60              living_areas_tokens = laar_tokens[0].split
                    ('+')
61              if (len(living_areas_tokens) == 1):
62                  living_area = re.search("(\\d+)",
                        living_areas_tokens[0]).group(1)
63              else:
64                  living_area = re.search("(\\d+)",
                        living_areas_tokens[0]).group(1) +
                        "+" + \
65                                  re.search("(\\d+).*$",
                                        living_areas_tokens[1])
                                        .group(1)
66          else:
67              rooms = re.search('(\\d+)\xa0rum',
                    laar_tokens[0])
68              if (rooms == None):
69                  rooms = ""
70              else:
71                  rooms = rooms.group(1)
72
73          plot_area = card.find('div', class_ = "sold-
                property-listing__land-area")
74          if plot_area == None:
75              plot_area = ""
76          else:
77              plot_area = plot_area.text.strip()[:-8].
                    replace("\xa0", "")
78
79          closing_price = card.find('span', class_ = "
                hcl-text␣hcl-text--medium")
80          closing_price = closing_price.text.strip()
                [9:-3].replace("\xa0", "")
81
82          values.append([date_sold, address, location,
                living_area, rooms, plot_area,
                closing_price])
83
84  data = list()
85  for val in values:
86      row = {
87          "Date␣Sold" : val[0],
```

```
88          "Address" : val[1],
89          "Location" : val[2],
90          "Living Area" : val[3],
91          "Rooms" : val[4],
92          "Plot Area" : val[5],
93          "Closing Price" : val[6]
94      }
95      data.append(row)
96
97  data = pd.DataFrame(data)
98
99  data.to_csv("assignment_2/housing_data.csv", index =
        None)
100
101 #Plotting the data
102
103 import numpy as np
104 import pandas as pd
105 import matplotlib.pyplot as plt
106
107 df = pd.read_csv("assignment_2/housing_data.csv")
108 df_2022 = df.loc[df["Date Sold"].str.contains("2022")]
109
110 closing_price_data = {
111     "minmum": df_2022["Closing Price"].min(),
112     "maxmimum": df_2022["Closing Price"].max(),
113     "first_q": df_2022["Closing Price"].quantile(0.25)
            ,
114     "median": df_2022["Closing Price"].median(),
115     "third_q": df_2022["Closing Price"].quantile(0.75)
116 }
117
118 plt.hist(df_2022["Closing Price"], bins = 11)
119 plt.title("House Closing Prices in Kung lv 2022")
120 plt.xlabel("Price in Tens of Millions of SEK")
121 plt.ylabel("Number of Houses Sold")
122 plt.show()
123
124
125 closing_prices = df_2022["Closing Price"]
126 boareas = df_2022["Living Area"].str.split("+").str
        [0].astype("float64")
127 rooms = df_2022["Rooms"]
128
129 plt.scatter(closing_prices, boareas, s = 20)
130 plt.title("Price and Boarea of Houses Sold in Kung lv
        in 2022")
131 plt.xlabel("Price in Tens of Millions of SEK")
132 plt.ylabel("Boarea in m  ")
133 plt.show()
```

```
134
135 plt.scatter(closing_prices[(rooms <= 2)], boareas[(
        rooms <= 2)], s = 20, color = 'red', label = '0 - 2
    ')
136 plt.scatter(closing_prices[(rooms <= 4) & (rooms > 2)
    ], boareas[(rooms <= 4) & (rooms > 2)], s = 20,
    color = 'orange', label = '3 - 4')
137 plt.scatter(closing_prices[(rooms <= 6) & (rooms > 4)
    ], boareas[(rooms <= 6) & (rooms > 4)], s = 20,
    color = '#FFD700', label = '5 - 6')
138 plt.scatter(closing_prices[(rooms <= 8) & (rooms > 6)
    ], boareas[(rooms <= 8) & (rooms > 6)], s = 20,
    color = 'green', label = '7 - 8')
139 plt.scatter(closing_prices[(rooms >= 9)], boareas[(
        rooms >= 9)], s = 20, color = 'blue', label = '9+')
140 plt.title("Price and Boarea of Houses Sold in Kung lv
    in 2022")
141 plt.xlabel("Price in Tens of Millions of SEK")
142 plt.ylabel("Boarea in m  ")
143 plt.legend(title = "Number of Rooms")
144 plt.show()
```