

DAT565/ Assignment 2 / Group 96

David Duong
thuan@chalmers.se

Savinjith Walisadeera
savwal@chalmers.se

September 16, 2024

Problem 2

i)

Most expensive house in Kungälv municipality during the year 2022 was 10 500 000 and the cheapest 1 650 000. The median price was 5 000 000. First quartile was 4 012 500 and third quartile was 5 795 000.

ii)

We chose the number of bins from 'Square Root Rule' which states number of bins = \sqrt{n} . In our case we had 200 data points giving us $\sqrt{190} \approx 14$.

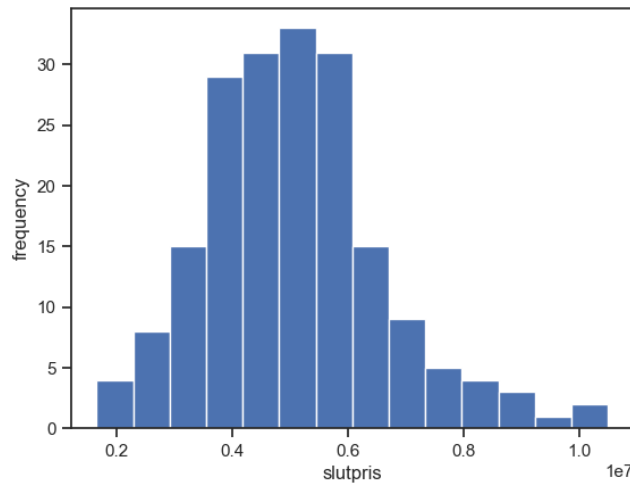


Figure 1: Histogram of closing prices and frequency in Kungälv municipality.

iii)

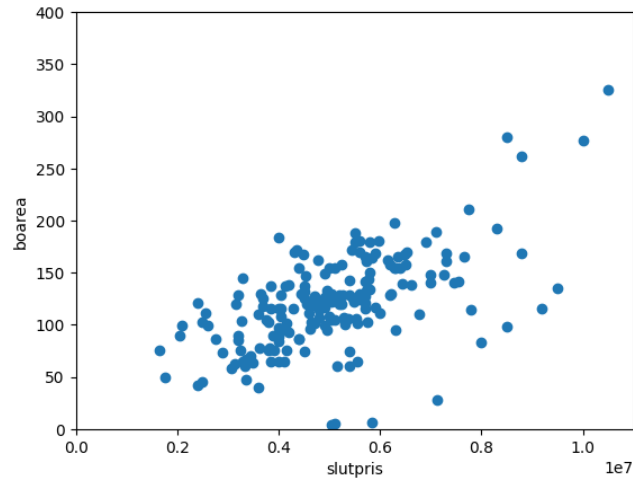


Figure 2: Scatter plot of *boarea* to closing prices.

iv)

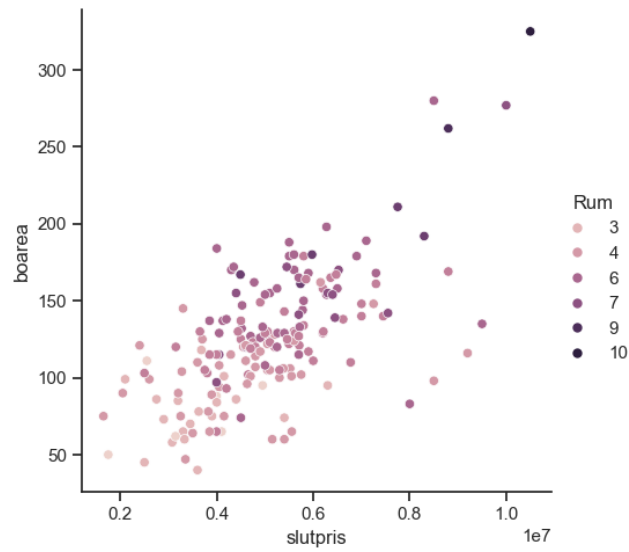


Figure 3: Scatter plot of *boarea* to closing prices with the number of rooms of the house defining the color of the point.

v)

From looking at the scatter plots in *Figure 2* and *Figure 3* we can see that *boarea* grows linearly to the closing price. As we look at growing closing prices in *Figure 3* we see that the colors of the points get darker as *boarea* rises. This makes sense as the number of rooms usually increase with *boarea*. Therefore we can conclude that the closing price is dependent on *boarea*.

A Code

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from bs4 import BeautifulSoup
5 import csv
6 import re
7 import seaborn as sns
8 import matplotlib
9 import matplotlib.colors
10
11 import pandas as pd
12 data = list()
13
14 months = {'januari':'01', 'februari':'02', 'mars':'03',
15           , 'april':'04', 'maj':'05', 'juni':'06', 'juli':'07',
16           , 'augusti':'08', 'september':'09', 'oktober':'10',
17           , 'november':'11', 'december':'12'}
18
19 for j in range(1,5):
20     with open("kungalv_slutpriser\
21             kungalv_slutpris_page_{}.html".format(j,0),
22             encoding = 'utf-8') as fp:
23         soup = BeautifulSoup(fp, 'html.parser')
24         sellings = soup.find_all(attrs={'class':'sold-
25             results__normal-hit'})
26         for sold in sellings:
27             sold_at = sold.find('span', class_='hcl-label_
28                 hcl-label--state_hcl-label--sold-at')
29             sold_at = sold_at.text.strip()[5:]
30             for month in months:
31                 sold_at = sold_at.replace(month, months[
32                     month])
33             sold_at = sold_at.split()
34
35             address = sold.find('h2', class_='sold-
36                 property-listing__heading_qa-selling-price-
37                 title_hcl-card__title')
38             address_text = address.text.strip()
39             location = address.parent.div
40             location.span.clear()
```

```

30     location = list(location.stripped_strings)
31     location = location[0].split(',')
32     for i in range(len(location)):
33         location[i] = location[i].strip()
34
35     try:
36         tomt = sold.find('div', class_="sold-
            property-listing__land-area").text.
            strip()[:-8]
37     except:
38         tomt = pd.NA
39
40     price = sold.find('span', class_='hcl-text_hcl
        -text--medium').text.strip()[9:-3]
41     price = int(re.sub('[^0-9]+', '', price))
42     rooms = sold.find('div', class_='sold-property
        -listing__subheading_sold-property-
        listing__area')
43     rooms = rooms.text.replace('_', '').replace('\
        n', '_')
44     rooms_val = re.sub('[^0-9_]+', '', rooms).
        split()
45
46     boarea = pd.NA
47     room_amount = pd.NA
48     biarea = pd.NA
49     if len(rooms_val) == 2:
50         if bool(re.search('(rum)', rooms)):
51             boarea = int(rooms_val[0])
52             room_amount = int(rooms_val[1])
53         else:
54             boarea = int(rooms_val[0])
55             biarea = int(rooms_val[1])
56     elif len(rooms_val) == 1:
57         boarea = int(rooms_val[0])
58     elif len(rooms_val) == 3:
59         boarea = int(rooms_val[0])
60         biarea = int(rooms_val[1])
61         room_amount = int(rooms_val[2])
62     elif len(rooms_val) == 0:
63         boarea = pd.NA
64         biarea = pd.NA
65         room_amount = pd.NA
66
67
68     row = {
69         'Year_of_sale': sold_at[2],
70         'Month_of_sale': sold_at[1],
71         'Day_of_sale': sold_at[0],
72         'Address': address_text,

```

```

73         'Ort': location[0],
74         'Kommun': location[1],
75         'boarea': boarea,
76         'biarea': biarea,
77         'Rum': room_amount,
78         'total_area': boarea+biarea,
79         'plot_area': tomt,
80         'slutpris': price
81     }
82     data.append(row)
83
84     for i in range(0,4):
85         for j in range(1,10):
86             with open("kungalv_slutpriser\
87                     kungalv_slutpris_page_{}_{}.html".format(i,j
88                     ), encoding = 'utf-8') as fp:
89                 soup = BeautifulSoup(fp, 'html.parser')
90                 sellings = soup.find_all(attrs={'class': 'sold-
91                     results__normal-hit'})
92                 for sold in sellings:
93                     sold_at = sold.find('span', class_='hcl-
94                         label_hcl-label--state_hcl-label--sold-
95                         at')
96                     sold_at = sold_at.text.strip()[5:]
97                     for month in months:
98                         sold_at = sold_at.replace(month,
99                             months[month])
100                     sold_at = sold_at.split()
101                     address = sold.find('h2', class_='sold-
102                         property-listing__heading_qa-selling-
103                         price-title_hcl-card__title')
104                     address_text = address.text.strip()
105                     location = address.parent.div
106                     location.span.clear()
107                     location = list(location.stripped_strings)
108                     location = location[0].split(',')
109                     for ind in range(len(location)):
110                         location[ind] = location[ind].strip()
111
112                     try:
113                         tomt = sold.find('div', class_="sold-
114                             property-listing__land-area").text.
115                             strip()[:-8]
116                     except:
117                         tomt = pd.NA
118
119                     price = sold.find('span', class_='hcl-text
120                         _hcl-text--medium').text.strip()[9:-3]
121                     price = int(re.sub('[^0-9]+', '', price))

```

```

111         rooms = sold.find('div', class_='sold-
112             property-listing__subheading_sold-
113             property-listing__area')
114         rooms = rooms.text.replace('␣', '').
115             replace('\n', '␣')
116         rooms_val = re.sub('[^0-9␣]+', '', rooms).
117             split()
118
119         boarea = pd.NA
120         room_amount = pd.NA
121         biarea = pd.NA
122         if len(rooms_val) == 2:
123             if bool(re.search('(rum)', rooms)):
124                 boarea = int(rooms_val[0])
125                 room_amount = int(rooms_val[1])
126             else:
127                 boarea = int(rooms_val[0])
128                 biarea = int(rooms_val[1])
129         elif len(rooms_val) == 3:
130             boarea = int(rooms_val[0])
131             biarea = int(rooms_val[1])
132             room_amount = int(rooms_val[2])
133         elif len(rooms_val) == 1:
134             boarea = int(rooms_val[0])
135         elif len(rooms_val) == 0:
136             boarea = pd.NA
137             biarea = pd.NA
138             room_amount = pd.NA
139
140         row = {
141             'Year_of_sale': sold_at[2],
142             'Month_of_sale': sold_at[1],
143             'Day_of_sale': sold_at[0],
144             'Address': address_text,
145             'Ort': location[0],
146             'Kommun': location[1],
147             'boarea': boarea,
148             'biarea': biarea,
149             'Rum': room_amount,
150             'total_area': boarea+biarea,
151             'plot_area': tomt,
152             'slutpris': price
153         }
154         data.append(row)
155     data = pd.DataFrame(data)
156     data.to_csv('kungalv_prices.csv', index=None)
157     sellings_2022 = data[data['Year_of_sale'] == '2022']

```

```

157 print(sellings_2022['slutpris'].max())
158 print(sellings_2022['slutpris'].min())
159 print(sellings_2022['slutpris'].median())
160 print(np.quantile(sellings_2022['slutpris'], 0.25))
161 print(np.quantile(sellings_2022['slutpris'], 0.75))
162 plt.hist(sellings_2022['slutpris'], int(np.ceil(np.
    sqrt(190))))
163
164 sellings_2022_boarea = sellings_2022.dropna(subset=['
    boarea'], inplace=False)
165
166 fig, ax = plt.subplots()
167 ax.scatter(sellings_2022_boarea['slutpris'],
    sellings_2022_boarea['boarea'])
168 plt.xlim(0,1.1*10**7)
169 plt.ylim(0,400)
170
171 sns.set_theme(style='ticks')
172 sns.relplot(data=sellings_2022_boarea, x='slutpris', y
    ='boarea', hue='Rum')
173 plt.show()

```