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 $\sqrt{190} \approx 14$.

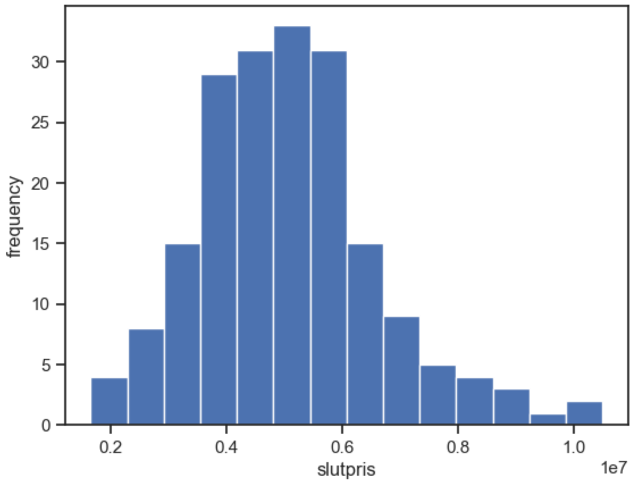

Figure 1: Histogram of closing prices and frequency in Kungälv municipality.

iii)

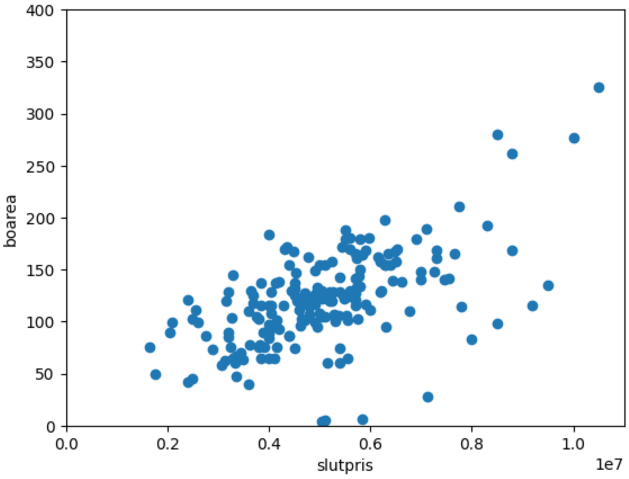

Figure 2: Scatter plot of *boarea* to closing prices.

iv)

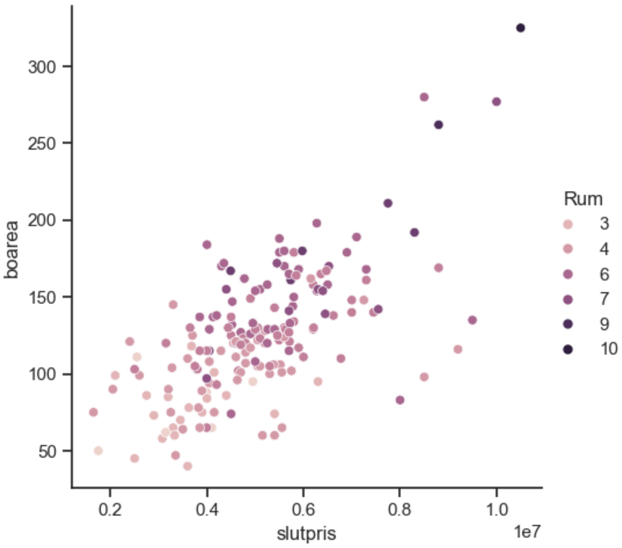

Figure 3: Scatter plot of *boarea* to closing prices with the number of rooms of the house defining the color of the point.

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

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
import csv
import re
import seaborn as sns
import matplotlib
import matplotlib.colors

import pandas as pd
data = list()

months = {'januari':'01', 'februari':'02', 'mars':'03', 'april':'04','maj':'05','juni':'06', 'juli':'07', 'augusti':'08', 'september':'09', 'oktober':'10','november':'11','december':'12'}
for j in range(1,5):
    with open("kungalv_slutpriser\kungalv_slutpris_page_{}{}.html".format(j,0), encoding = 'utf-8') as fp:
        soup = BeautifulSoup(fp, 'html.parser')
    sellings = soup.find_all(attrs={'class':'sold-results__normal-hit'})
    for sold in sellings:
        sold_at = sold.find('span', class_='hcl-label␣hcl-label--state␣hcl-label--sold-at')
        sold_at = sold_at.text.strip()[5:]
        for month in months:
            sold_at = sold_at.replace(month, months[month])
        sold_at = sold_at.split()

        address = sold.find('h2', class_='sold-property-listing__heading␣qa-selling-price-title␣hcl-card__title')
        address_text = address.text.strip()
        location = address.parent.div
        location.span.clear()
        location = list(location.stripped_strings)
        location = location[0].split(',')
        for i in range(len(location)):
            location[i] = location[i].strip()

        try:
            tomt = sold.find('div', class_="sold-property-listing__land-area").text.strip()[:-8]
        except:
            tomt = pd.NA

        price = sold.find('span', class_='hcl-text␣hcl-text--medium').text.strip()[9:-3]
        price = int(re.sub('[^0-9]+', '', price))
        rooms = sold.find('div', class_='sold-property-listing__subheading␣sold-property-listing__area')
        rooms = rooms.text.replace('␣', '').replace('\n', '␣')
        rooms_val = re.sub('[^0-9␣]+', '', rooms).split()

        boarea = pd.NA
        room_amount = pd.NA
        biarea = pd.NA
        if len(rooms_val) == 2:
            if bool(re.search('(rum)',rooms)):
                boarea = int(rooms_val[0])
                room_amount = int(rooms_val[1])
            else:
                boarea = int(rooms_val[0])
                biarea = int(rooms_val[1])
        elif len(rooms_val) == 1:
            boarea = int(rooms_val[0])
        elif len(rooms_val) == 3:
            boarea = int(rooms_val[0])
            biarea = int(rooms_val[1])
            room_amount = int(rooms_val[2])
        elif len(rooms_val) == 0:
            boarea = pd.NA
            biarea = pd.NA
            room_amount = pd.NA


        row = {
                'Year␣of␣sale':sold_at[2],
                'Month␣of␣sale':sold_at[1],
                'Day␣of␣sale':sold_at[0],
                'Address': address_text,
```

```python
                'Ort': location[0],
                'Kommun': location[1],
                'boarea': boarea,
                'biarea': biarea,
                'Rum': room_amount,
                'total_area': boarea+biarea,
                'plot_area': tomt,
                'slutpris': price
                    }
        data.append(row)

for i in range(0,4):
    for j in range(1,10):
        with open("kungalv_slutpriser\kungalv_slutpris_page_{}{}.html".format(i,j), encoding = 'utf-8') as fp:
            soup = BeautifulSoup(fp, 'html.parser')
        sellings = soup.find_all(attrs={'class':'sold-results__normal-hit'})
        for sold in sellings:
            sold_at = sold.find('span', class_='hcl-label hcl-label--state hcl-label--sold-at')
            sold_at = sold_at.text.strip()[5:]
            for month in months:
                sold_at = sold_at.replace(month, months[month])
            sold_at = sold_at.split()
            address = sold.find('h2', class_='sold-property-listing__heading qa-selling-price-title hcl-card__title')
            address_text = address.text.strip()
            location = address.parent.div
            location.span.clear()
            location = list(location.stripped_strings)
            location = location[0].split(',')
            for ind in range(len(location)):
                location[ind] = location[ind].strip()

            try:
                tomt = sold.find('div', class_="sold-property-listing__land-area").text.strip()[:-8]
            except:
                tomt = pd.NA

            price = sold.find('span', class_='hcl-text hcl-text--medium').text.strip()[9:-3]
            price = int(re.sub('[^0-9]+', '', price))
            rooms = sold.find('div', class_='sold-property-listing__subheading␣sold-property-listing__area')
            rooms = rooms.text.replace('␣', '').replace('\n', '␣')
            rooms_val = re.sub('[^0-9␣]+', '', rooms).split()

            boarea = pd.NA
            room_amount = pd.NA
            biarea = pd.NA
            if len(rooms_val) == 2:
                if bool(re.search('(rum)',rooms)):
                    boarea = int(rooms_val[0])
                    room_amount = int(rooms_val[1])
                else:
                    boarea = int(rooms_val[0])
                    biarea = int(rooms_val[1])
            elif len(rooms_val) == 3:
                boarea = int(rooms_val[0])
                biarea = int(rooms_val[1])
                room_amount = int(rooms_val[2])
            elif len(rooms_val) == 1:
                boarea = int(rooms_val[0])
            elif len(rooms_val) == 0:
                boarea = pd.NA
                biarea = pd.NA
                room_amount = pd.NA


            row = {
                'Year␣of␣sale':sold_at[2],
                'Month␣of␣sale':sold_at[1],
                'Day␣of␣sale':sold_at[0],
                'Address': address_text,
                'Ort': location[0],
                'Kommun': location[1],
                'boarea': boarea,
                'biarea': biarea,
                'Rum': room_amount,
                'total_area': boarea+biarea,
                'plot_area': tomt,
                'slutpris': price
                    }
            data.append(row)
data = pd.DataFrame(data)

data.to_csv('kungalv_prices.csv',index=None)

sellings_2022 = data[data['Year␣of␣sale'] == '2022']
```

```python
print(sellings_2022['slutpris'].max())
print(sellings_2022['slutpris'].min())
print(sellings_2022['slutpris'].median())
print(np.quantile(sellings_2022['slutpris'], 0.25))
print(np.quantile(sellings_2022['slutpris'], 0.75))
plt.hist(sellings_2022['slutpris'], int(np.ceil(np.sqrt(190))))

sellings_2022_boarea = sellings_2022.dropna(subset=['boarea'], inplace=False)

fig, ax = plt.subplots()
ax.scatter(sellings_2022_boarea['slutpris'], sellings_2022_boarea['boarea'])
plt.xlim(0,1.1*10**7)
plt.ylim(0,400)

sns.set_theme(style='ticks')
sns.relplot(data=sellings_2022_boarea, x='slutpris', y='boarea', hue='Rum')
plt.show()
```