DAT565/DIT407 Assignment 2

Louis PRUNIER
prunier.louis@icloud.com

Marco SPEZIALE
speziale@student.chalmers.se

2024-04-10

1 Figure 1

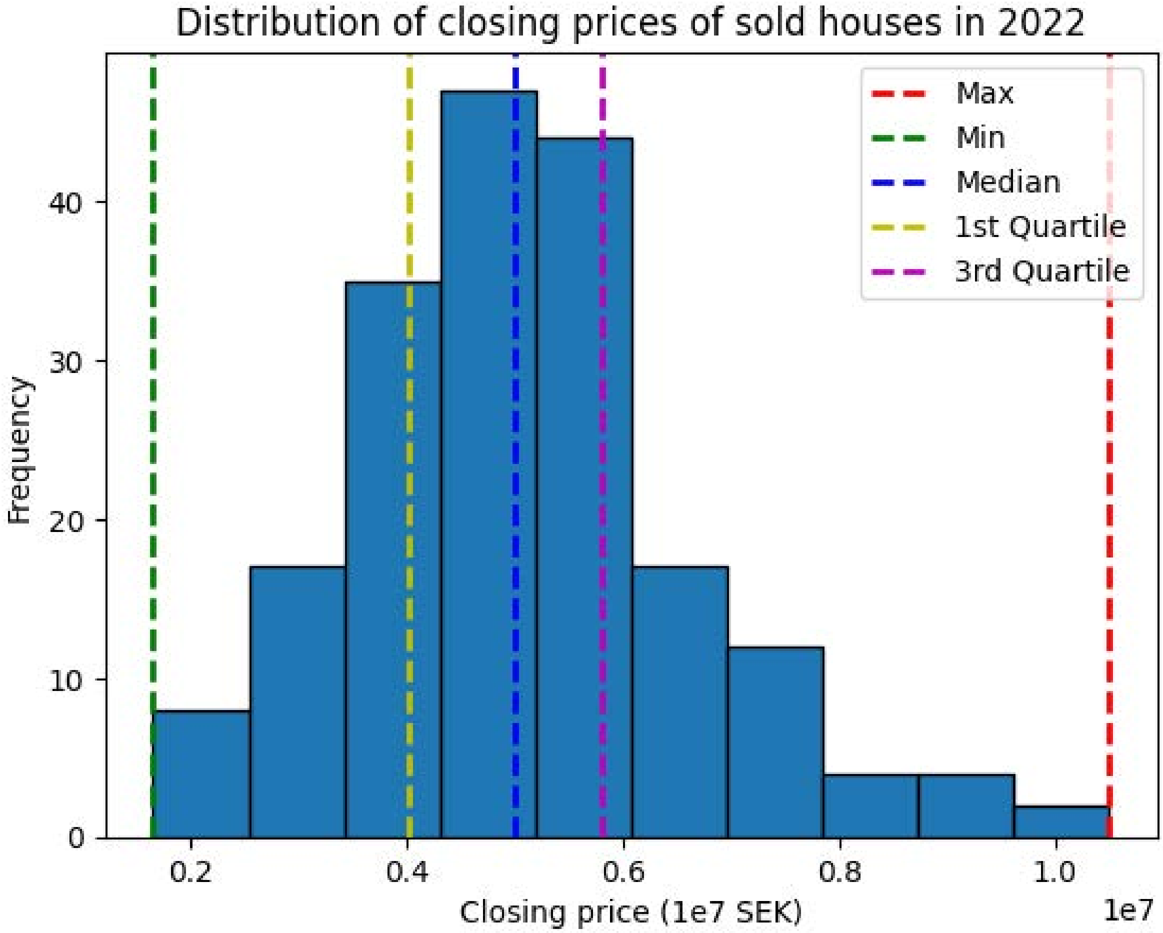

Figure 1: Distribution of Closing Prices in 2022

2 Figure 2

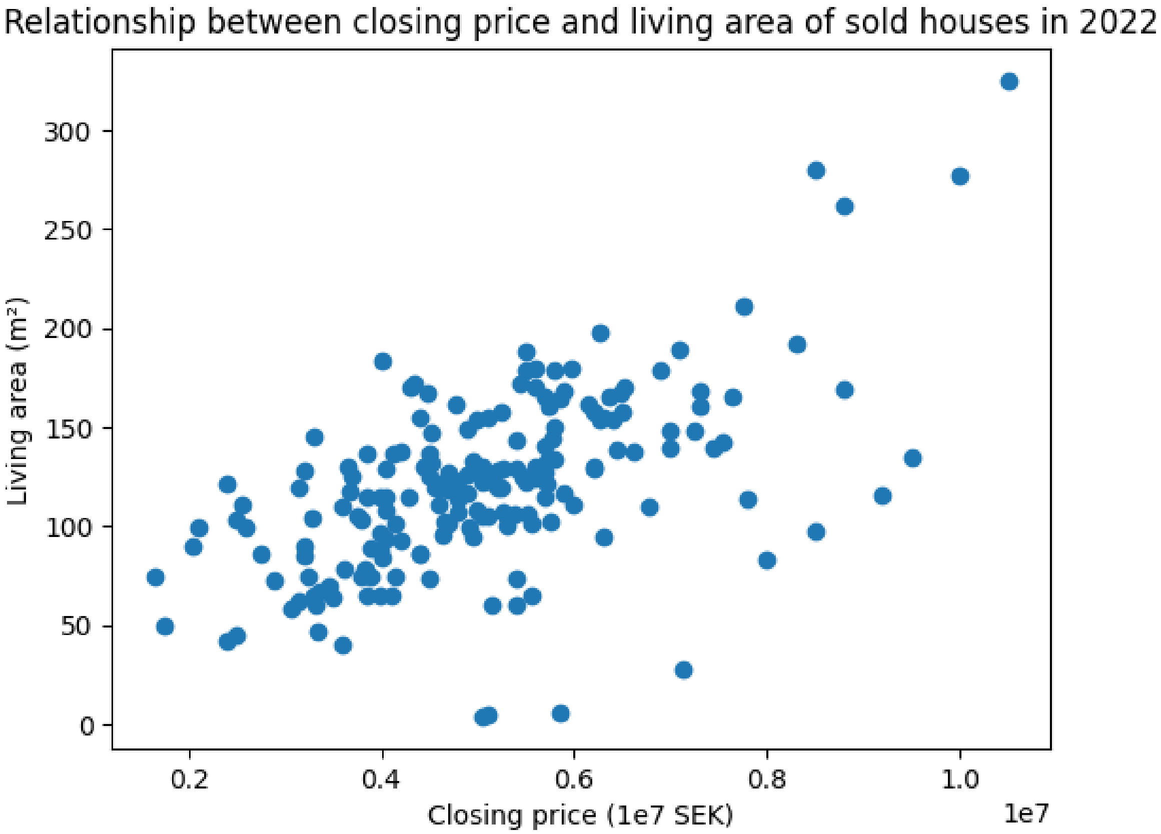

Figure 2: Relationship between house Boarea and closing price (2022)

3 Figure 3

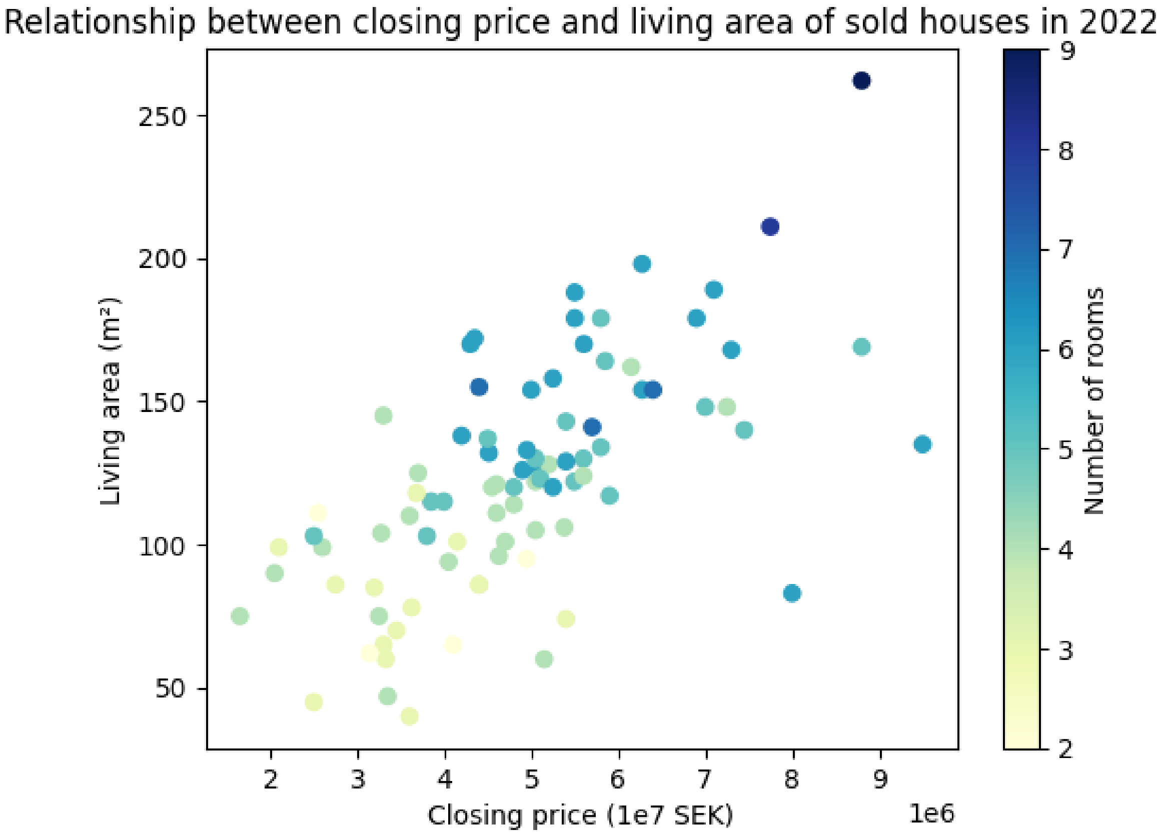

Figure 3: Relation between house boarea and closing price by number of rooms

4 Discussion

4.1 Figure 1

Figure 1 is a bar chart illustrating the distribution of house closing prices in 2022 in Kungälv. On this histogram, 5 key statistical concepts are added: the minimum value, the maximum value, the median, the first quartile and the third quartile. On this histogram, we can see that house sale prices in 2022 in Kungälv range from SEK 1,650,000 to a maximum of SEK 10,500,000. The median price is SEK 5,000,000. 25% of closing prices are below SEK 4,000,000 and conversely only 25% of closing prices are above SEK 5,795,000. We chose 10 as the number of bins to show the distribution as the prices range from approximately 1,000,000 to 10,000,000 SEK, and each bin can therefore represent (approximately) a step in millions of SEK. 10 is also enough bins to make each line corresponding to the five-number summary fit into one unique bin each, clearly showing the frequency associated with the given statistical concept.

4.2 Figure 2

Figure 2 is a scatter-plot showing the relationship between living areas (boarea) and the closing price of the house. We can see from the trend line that prices increase in a linear fashion as living area also increases, reflecting a fairly logical correlation between living area and house price. Figure 2 also shows that the high concentration of points (houses) is mainly between 100 and 150 m2 at a price of SEK 5,000,000 (median).

4.3 Figure 3

Finally, Figure 3 is identical to number 2, except that the number of bedrooms in the house is also indicated (colored dots). On this figure, we can see that the smallest houses (between 50 and 100 m2) have the fewest bedrooms (between 0 and 2 bedrooms), and conversely the largest houses have the most bedrooms (over 250 m2 and 8 to 10 bedrooms). The vast majority of houses (between 100 and 150 m2) have around 4 bedrooms (between 2 and 6 bedrooms to be broader).

4.4 Conclusion

So, the conclusions we can draw are that the closing price of a house in Kungälv is strongly correlated to the variable 'living area' (BoArea), even if there are disparities with some outliers. But also, and quite logically, the largest houses have the most bedrooms and vice versa for the smallest, which conversely doesn't really influence the closing price but depends more on the size of the house's living area.

5 Appendix

```
#%%
import os
from bs4 import BeautifulSoup
import pandas as pd


## PART 1 ##

# methods to clean up the extracted information into useful formats

def cleanup_date(date_info):
    split_date = date_info.split()
    pure_date = split_date[1:]
    return '␣'.join(pure_date)

def cleanup_address(address_info):
    return address_info.strip()

def cleanup_location(location_info):
    location_info_split = location_info.split()
    return '␣'.join(location_info_split)

def cleanup_bo_area(bo_area_info):
    bo_area = bo_area_info.strip()
    bo_area = bo_area.replace(',','.')
    if bo_area.isnumeric():
        return float(bo_area)
    else:
        return None

def cleanup_bi_area(bi_area_info):
    bi_area_split = bi_area_info.split()
    bi_area_pure = bi_area_split[1:2]
    bi_area = ''.join(bi_area_pure)
    bi_area = bi_area.replace(',','.')
    if bi_area.isnumeric():
         return float(bi_area)
    else:
         return None

def cleanup_rooms(nr_rooms_info):
    nr_rooms_split = nr_rooms_info.split()
    nr_rooms_pure = nr_rooms_split[:1]
    nr_rooms = ''.join(nr_rooms_pure)
    if nr_rooms.isnumeric():
        nr_rooms = int(nr_rooms)
    else:
        nr_rooms = None

def cleanup_area_room(area_and_room_info):
    area_room = []
    area_and_room_info_split = area_and_room_info.split()
    area_pure = area_and_room_info_split[:1]
    room_pure = area_and_room_info_split[2:3]

    area = ''.join(area_pure)
    area = area.replace(',','.')
    if area.isnumeric():
        area = float(area)
    else:
        area = None
    area_room.append(area)

    nr_rooms = ''.join(room_pure)
    if nr_rooms.isnumeric():
        nr_rooms = int(nr_rooms)
    else:
        nr_rooms = None
    area_room.append(nr_rooms)
    return area_room

def cleanup_plotarea(plot_area_info):
    plot_area = ''
    for char in plot_area_info:
         if char.isnumeric():
              plot_area += char
    plot_area = plot_area[:-1]
    plot_area.replace(',','.')
    return float(plot_area)

def cleanup_price(price_info):
    price = ''
    for char in price_info:
        if char.isnumeric():
            price += char
    return float(price)

```

```

# directory with html files
directory = 'kungalv_slutpriser'

data = []

# for each file in directory, extract the sought-out information
for filepath in os.listdir(directory):
    with open(os.path.join(directory, filepath), encoding='utf-8') as fp:
        soup = BeautifulSoup(fp, 'html.parser')

    result = soup.find_all('li', {'class': 'sold-results__normal-hit'})

    # find the relevant pieces of information
    for element in result:
        date_info = element.find('span', {'class': 'hcl-label␣hcl-label--state␣hcl-label--sold-at'}).text
        date = cleanup_date(date_info)

        address_info = element.find('h2', {'class': 'sold-property-listing__heading␣qa-selling-price-title␣hcl-card__title'}).text
        address = cleanup_address(address_info)

        location_info = element.find('div', {'class': 'sold-property-listing__location'}).contents[3].contents[2]
        location = cleanup_location(location_info)

        if len(element.find('div', {'class': 'sold-property-listing__subheading␣sold-property-listing__area'}).contents) > 1:
            bo_area_info = element.find('div', {'class': 'sold-property-listing__subheading␣sold-property-listing__area'}).contents[0]
            bo_area = cleanup_bo_area(bo_area_info)

            bi_area_info = element.find('div', {'class': 'sold-property-listing__subheading␣sold-property-listing__area'}).contents[1].text
```

```python
            bi_area = cleanup_bi_area(bi_area_info)

            if (bo_area and bi_area):
                area = bo_area + bi_area
            elif (bo_area):
                area = bo_area

            nr_rooms_info = element.find('div', {'class': 'sold-property-listing__subheading sold-property-listing__area'}).contents[2]
            nr_rooms = cleanup_rooms(nr_rooms_info)
        else:
            area_and_room_info = element.find('div', {'class': 'sold-property-listing__subheading sold-property-listing__area'}).contents[0]
            area_and_room = cleanup_area_room(area_and_room_info)

            area = area_and_room[0]
            bi_area = None
            bo_area = area

            nr_rooms = area_and_room[1]

        if element.find('div', {'class': 'sold-property-listing__land-area'}):
            plot_area_info = element.find('div', {'class': 'sold-property-listing__land-area'}).text
            plot_area = cleanup_plotarea(plot_area_info)
        else:
            plot_area = None

        price_info = element.find('span', {'class': 'hcl-text hcl-text--medium'}).text
        price = cleanup_price(price_info)

        data.append([date, address, location, bo_area, bi_area, area, nr_rooms, plot_area, price])

# turn data into dataframe
```

```python
df = pd.DataFrame(data, columns=['Date', 'Address', 'Location', 'Bo-area', 'Bi-area', 'Total area', 'Rooms', 'Plot', 'Price'])

# turn dataframe into csv file
csv = df.to_csv('houseprices.csv')

#%%
## PART 2 ##
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('houseprices.csv')
df = df.drop('Unnamed: 0', axis=1)

sold_2022 = df[df['Date'].str.contains('2022')]

closing_prices_2022 = sold_2022['Price']
closing_prices_2022

min_2022 = closing_prices_2022.min()
max_2022 = closing_prices_2022.max()
median_2022 = closing_prices_2022.median()
first_quartile_2022 = closing_prices_2022.quantile(0.25)
third_quartile_2022 = closing_prices_2022.quantile(0.75)

# histogram
plt.hist(closing_prices_2022, bins=10, edgecolor='black')

plt.axvline(max_2022, color='r', linestyle='dashed', linewidth=2, label='Max')
plt.axvline(min_2022, color='g', linestyle='dashed', linewidth=2, label='Min')
plt.axvline(median_2022, color='b', linestyle='dashed', linewidth=2, label='Median')
plt.axvline(first_quartile_2022, color='y', linestyle='dashed', linewidth=2, label='1st Quartile')
plt.axvline(third_quartile_2022, color='m', linestyle='dashed', linewidth=2, label='3rd Quartile')

plt.xlabel('Closing price (1e7 SEK)')
plt.ylabel('Frequency')
plt.title('Distribution␣of␣closing␣prices␣of␣sold␣
    houses␣in␣2022')

plt.legend()
plt.show()

# scatter plot
bo_areas_2022 = sold_2022['Bo-area']
bo_areas_2022

plt.scatter(closing_prices_2022, bo_areas_2022)
plt.ylabel('Living␣area␣(m^2)')
plt.xlabel('Closing␣price␣(1e7␣SEK)')
plt.title('Relationship␣between␣closing␣price␣and␣
    living␣area␣of␣sold␣houses␣in␣2022')

plt.show()

# colored version
nr_rooms_2022 = sold_2022['Rooms']

scatter = plt.scatter(closing_prices_2022,
    bo_areas_2022, c=nr_rooms_2022, cmap='YlGnBu')
plt.ylabel('Living␣area␣(m^2)')
plt.xlabel('Closing␣price␣(1e7␣SEK)')
plt.title('Relationship␣between␣closing␣price␣and␣
    living␣area␣of␣sold␣houses␣in␣2022')

plt.colorbar(scatter, label='Number␣of␣rooms')
plt.show()
```