Project Overview

Name of the project	Real estate – apartment for rent analysis
Author	Piotr Milner piotr.milner@outlook.com
Project description	The project concerns the analysis of the real estate market in Poland in terms of potential investment in the purchase of an apartment for rent. The goal was to obtain results showing the profitability of the indicated investment in cities with a population of over 500,000. In addition to the location, the main criteria determining the price were the number of rooms and area in square meters. The comparison includes offering prices of apartments for sales and their rents. A simplified Return On Investment (ROI) ratio was used as a comparison of investment profitability
Dataset source	The collected data comes from the Domiporta.pl, which is one of the largest websites in Poland that allows publishing real estate offers. Data obtained by web scrapping program written in Python.
GitHub	https://github.com/piotr-milner/Data-Projects/tree/main/Real%20Estate%20Project

Roadmap Table of Contents

1. Defining Questions

First step in our analysis is to define a goal including answering to the following questions:

1. In which city (over 500k population) will the purchase of an apartment for rent bring the highest ROI?
2. What number of square meters will bring the highest ROI?
3. What number of rooms will bring the highest ROI?

2. Collecting data - Web Scrapping

A special tool has been designed to extract data from a website: Domiporta.pl and saving the output to CSV file. Before the final selection of the site, the robots.txt subpage was checked.

Data that needs to be extracted:
- source of information (as a future the project will expand for additional data from other sources)
- location
- price/rent
- the number of square meters
- no. of rooms

All the above for apartments for sale and separately for apartments for rent currently available as an real offer.

Code in Python below:

```
# ==========================================================
# Web scrapping scrypt for real estate data analysis project, based on a BeautifulSoup library
# Current site: Domiporta.pl
# As a future plan the project will expand for additional data from other sources
# thus functions has been implemented
# ==========================================================

import re
import csv
import requests
from bs4 import BeautifulSoup

final_data = []
count = 0  # Variable used in loop to correctly updating final dict with scrapped data
```

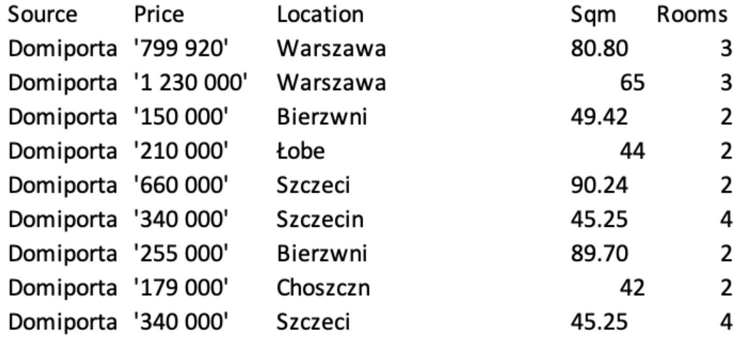
```
# Searching for all pages currently available on site
req = requests.get('https://www.domiporta.pl/mieszkanie/sprzedam?Rynek=Wtorny')
parse = BeautifulSoup(req.text, 'html.parser').select('.pagination')
all_pages = max([int(num) for num in re.split("[^0-9]", str(parse)) if num != '']) + 1

# Selecting and assigning price, square meters, location and rooms from HTML
for p in range(1, all_pages + 1):
    req_main = requests.get(
        f'https://www.domiporta.pl/mieszkanie/sprzedam?Rynek=Wtorny&PageNumber={p}')
    soup = BeautifulSoup(req_main.text, 'html.parser')
    price = soup.select('.sneakpeak__price_value')
    sqm = soup.select('.sneakpeak__details_item.sneakpeak__details_item--area')
    loc = soup.select('.sneakpeak__title--inblock')
    r = soup.select('.sneakpeak__details_item.sneakpeak__details_item--room')

    # Extracting a single price and adding to final list
    def extract_price(price):
        for idx, item in enumerate(price):
            if idx % 2 == 0:
                raw_str = price[idx].getText()
                sub1 = '">'
                sub2 = '</'
                idx1 = raw_str.find(sub1)
                idx2 = raw_str.find(sub2)
                res = raw_str[idx1 + 1: idx2 - 2].__repr__()
                final_data.append({'Source': 'Domiporta', 'Price': res.replace(r'\xa0', ' ')})
        return final_data
    extract_price(price)

    # Extracting a single name of location and adding to final dict
    def extract_loc(loc, count):
        for idx, item in enumerate(loc):
            raw_str = loc[idx].getText()
            sub1 = 'mieszkanie '
            sub2 = ','
            idx1 = raw_str.find(sub1)
            idx2 = raw_str.find(sub2)
            res = raw_str[idx1 + len(sub1): idx2]
            final_data[count].update({'Location': res})
            count += 1
        return final_data
    extract_loc(loc, count)

    # Extracting a single number of square meters and adding to final dict
    def extract_sqm(sqm, count):
        for idx, item in enumerate(sqm):
            if idx % 2 == 0:
                raw_str = sqm[idx].getText()
                sub1 = 'Powierzchnia">'
                sub2 = '<abbr'
                idx1 = raw_str.find(sub1)
                idx2 = raw_str.find(sub2)
                res = raw_str[idx1 + len(sub1) + 1: idx2 - 3].strip()
                final_data[count].update({'Sqm': res.replace(',', '.')})
                count += 1
        return final_data
    extract_sqm(sqm, count)

    # Extracting a single rooms number  and adding to final dict
    def extract_r(r, count):
        for idx, item in enumerate(r):
            if idx % 2 == 0:
                raw_str = r[idx].getText()
                sub1 = '>'
                sub2 = '<'
                idx1 = raw_str.find(sub1)
                idx2 = raw_str.find(sub2)
                res = raw_str[idx1 + len(sub1) + 1: idx2 - 5].strip()
                final_data[count].update({'Rooms': res})
                count += 1
        return final_data, count

    final_data, count = extract_r(r, count)

# Exporting dictionary to CSV file
csv_file = "final_data_sell.csv"
csv_columns = ['Source', 'Price', 'Location', 'Sqm', 'Rooms']
try:
    with open(csv_file, 'w') as csv_file:
        wrt = csv.DictWriter(csv_file, fieldnames=csv_columns)
        wrt.writeheader()
        for data in final_data:
            wrt.writerow(data)
except IOError:
    print("I/O error")
```

As output, two CSV files:

final_data_sell.csv with a volume of over 10,000 records and

final_data_rent.csv with a volume of over 7,000
The easiest way to analyse them will be importing to MS Excel.

3. Exploratory Data Analysis (EDA)



Source	Price	Location	Sqm	Rooms
Domiporta	'799 920'	Warszawa	80.80	3
Domiporta	'1 230 000'	Warszawa	65	3
Domiporta	'150 000'	Bierzwni	49.42	2
Domiporta	'210 000'	Łobe	44	2
Domiporta	'660 000'	Szczeci	90.24	2
Domiporta	'340 000'	Szczecin	45.25	4
Domiporta	'255 000'	Bierzwni	89.70	2
Domiporta	'179 000'	Choszczn	42	2
Domiporta	'340 000'	Szczeci	45.25	4

As a result of the Exploratory Data Analysis, the following information was obtained:

Final_data_sell.csv

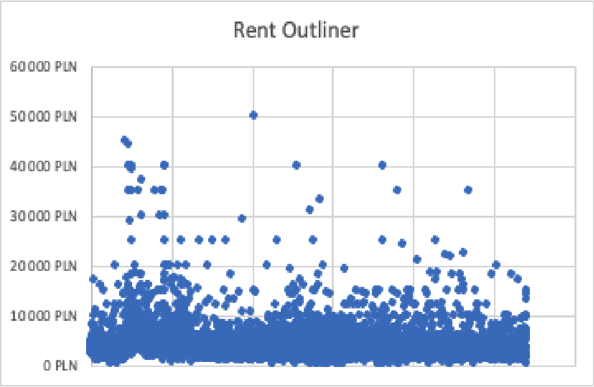
⇒ No. of rows: 10,001
⇒ No. of columns: 5
⇒ No. of rows containing empty cells: 29 - needs cleaning

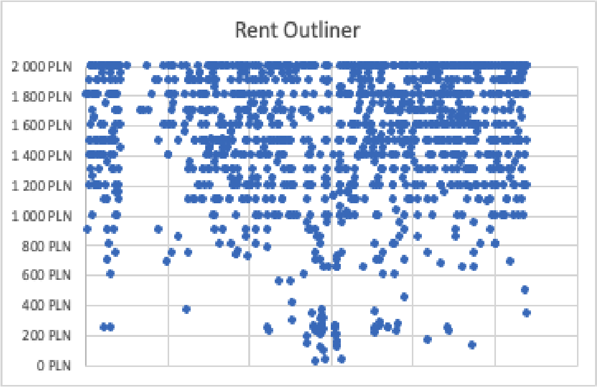
Final_data_rent.csv

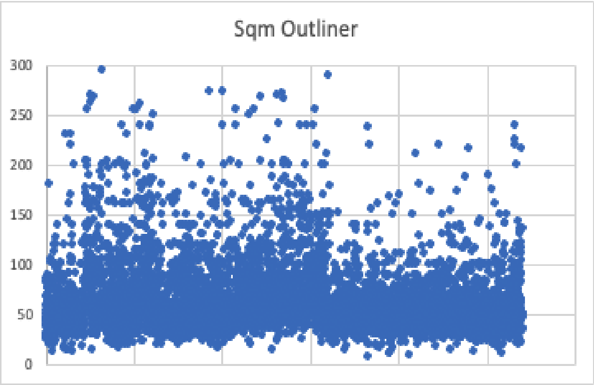
⇒ No. of rows: 7,233
⇒ No. of columns: 5
⇒ No. of rows containing empty cells: 25 - needs cleaning

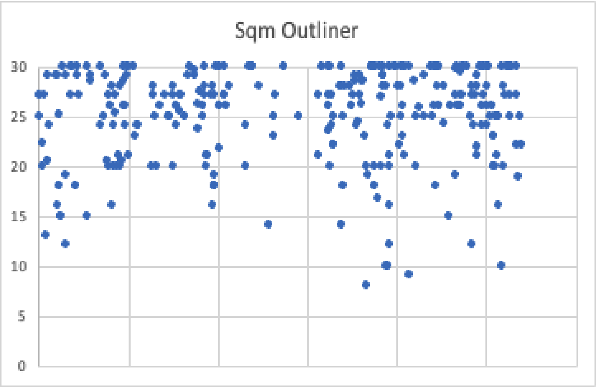
For both datasets:

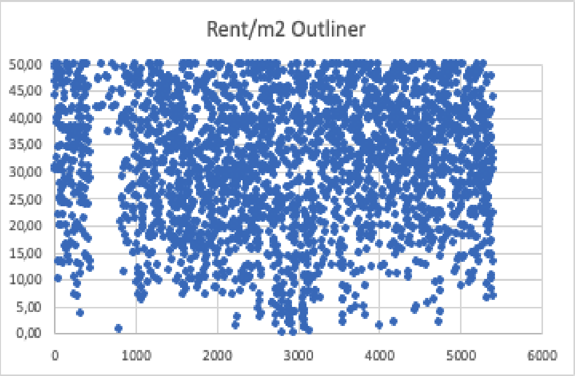
⇒ The “price” column was imported as a string because of the single quote in it – needs cleaning
⇒ Some values in “sqm” column were imported as a string because of dot sign in it– needs cleaning
⇒ Adding calculated fields “price(rent)/m2” to check all the outliners using scatterplot chart:

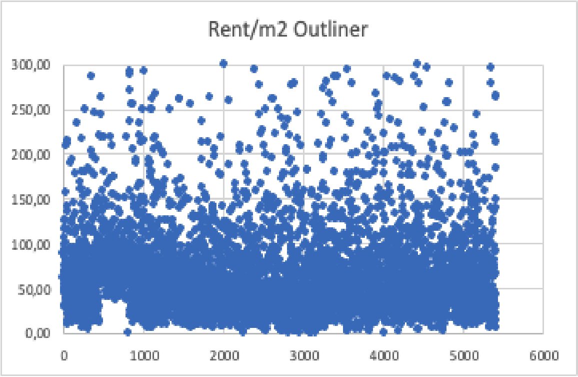

Rent Outliner
60 000 PLN
50 000 PLN
40 000 PLN
30 000 PLN
20 000 PLN
10 000 PLN
0 PLN

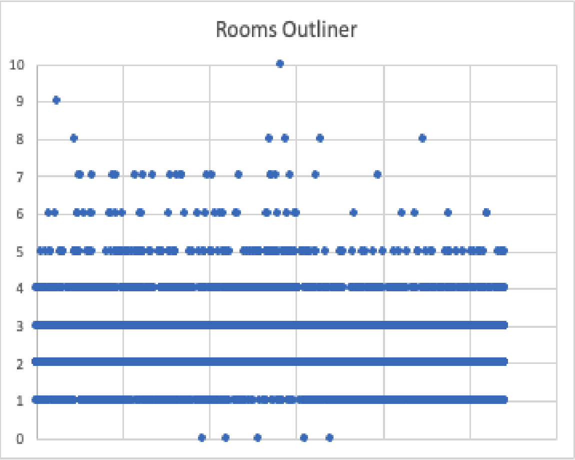

Rent Outliner
2 000 PLN
1 800 PLN
1 600 PLN
1 400 PLN
1 200 PLN
1 000 PLN
800 PLN
600 PLN
400 PLN
200 PLN
0 PLN

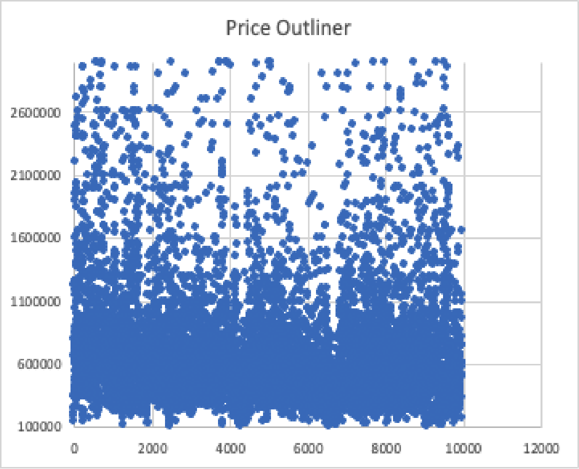

Sqm Outliner
300
250
200
150
100
50
0

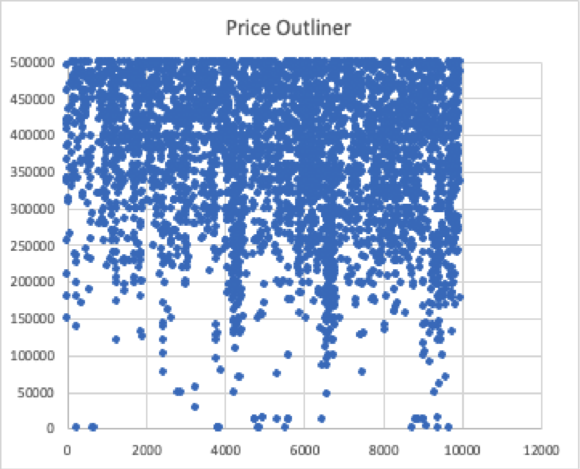

Sqm Outliner
30
25
20
15
10
5
0

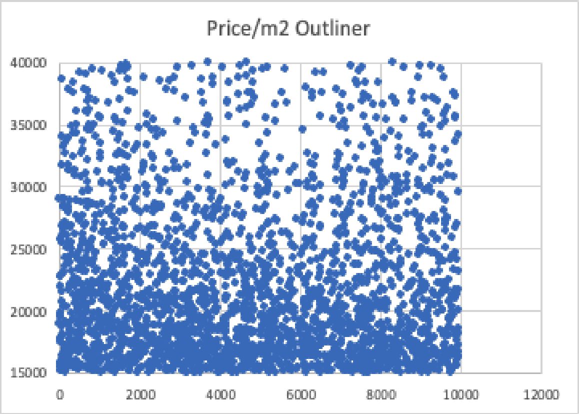

Rent/m2 Outliner
50,00
45,00
40,00
35,00
30,00
25,00
20,00
15,00
10,00
5,00
0,00
0
1000
2000
3000
4000
5000
6000

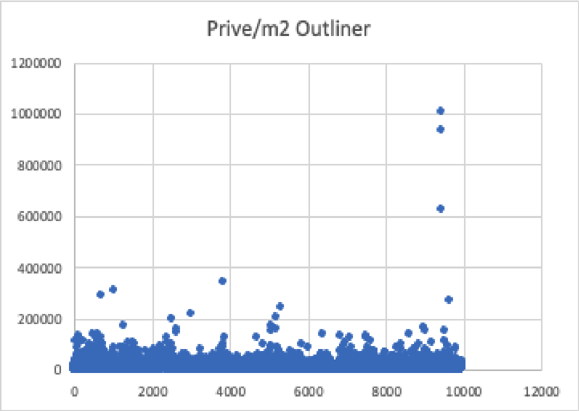

Rent/m2 Outliner
300,00
250,00
200,00
150,00
100,00
50,00
0,00
0
1000
2000
3000
4000
5000
6000


Rooms Outliner
10
9
8
7
6
5
4
3
2
1
0



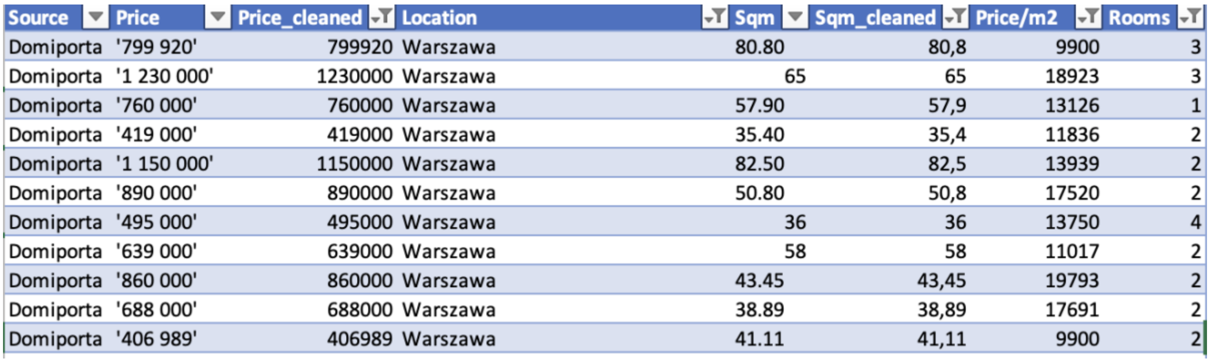






4. Data Cleaning

1. Deleting rows with empty cells
2. Filtering location column to cities over 500,000 population i.e.: Warszawa, Kraków, Wrocław, Łódź, Poznań
3. Deleting single quotes signs from "price" column and converting to value
4. In "sqm" column: converting dot sign to comma and converting to value
5. Filtering values to get rid of outliners



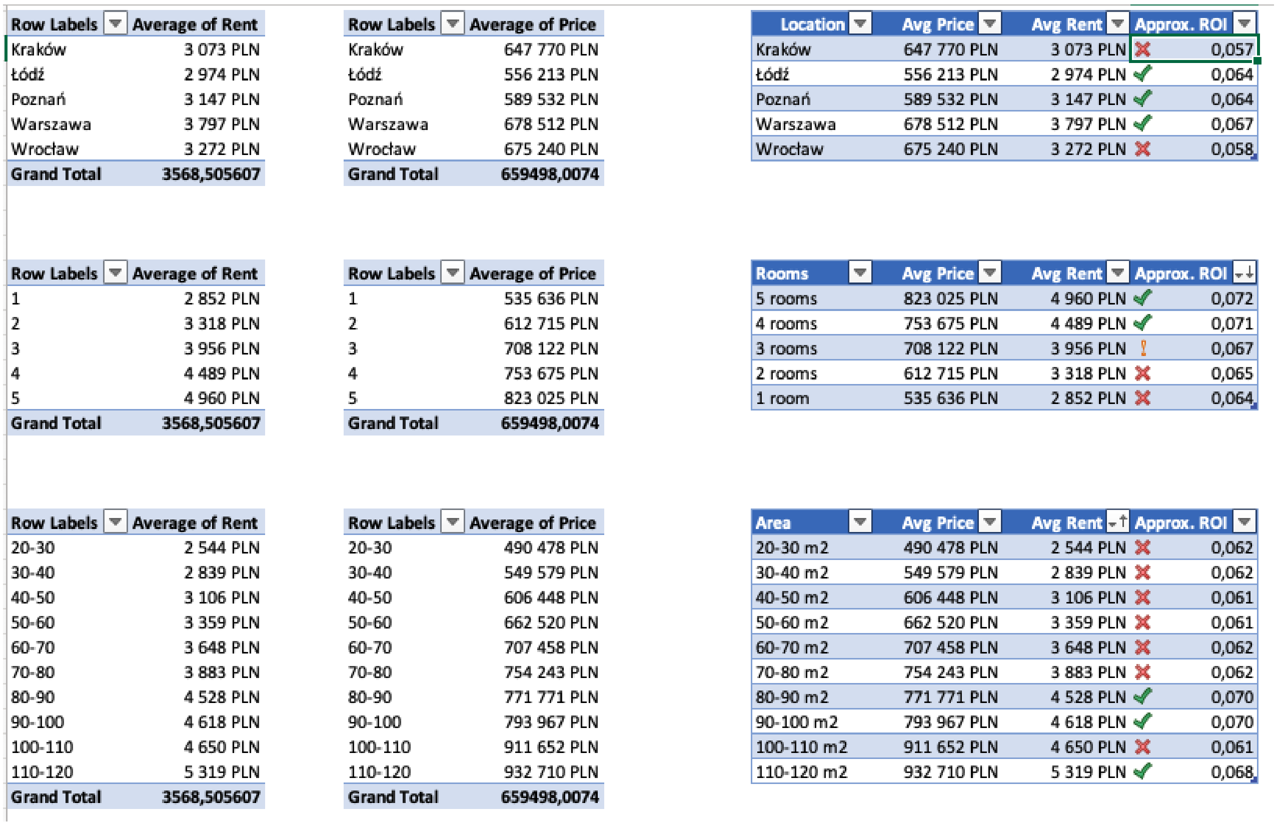
Source	Price	Price_cleaned	Location	Sqm	Sqm_cleaned	Price/m2	Rooms
Domiporta	'799 920'	799920	Warszawa	80.80	80,8	9900	3
Domiporta	'1 230 000'	1230000	Warszawa	65	65	18923	3
Domiporta	'760 000'	760000	Warszawa	57.90	57,9	13126	1
Domiporta	'419 000'	419000	Warszawa	35.40	35,4	11836	2
Domiporta	'1 150 000'	1150000	Warszawa	82.50	82,5	13939	2
Domiporta	'890 000'	890000	Warszawa	50.80	50,8	17520	2
Domiporta	'495 000'	495000	Warszawa	36	36	13750	4
Domiporta	'639 000'	639000	Warszawa	58	58	11017	2
Domiporta	'860 000'	860000	Warszawa	43.45	43,45	19793	2
Domiporta	'688 000'	688000	Warszawa	38.89	38,89	17691	2
Domiporta	'406 989'	406989	Warszawa	41.11	41,11	9900	2

After cleaning table has been copied and passed as value in other sheet.

5. Data Analysis

Using pivot tables calculating the average price of an apartment and the average rent broken down by location, number of rooms and area (ranges 10m2). Then the ROI was obtained according to the following formula. In addition to the price of the apartment, expenses such as transaction-related costs and taxes should also be added to the final costs. However, they may be omitted for this comparison

$$ROI = \frac{\text{avg. rent} * 12 \text{ months}}{\text{avg. price}}$$



Row Labels	Average of Rent
Kraków	3 073 PLN
Łódź	2 974 PLN
Poznań	3 147 PLN
Warszawa	3 797 PLN
Wrocław	3 272 PLN
Grand Total	**3568,505607**

Row Labels	Average of Price
Kraków	647 770 PLN
Łódź	556 213 PLN
Poznań	589 532 PLN
Warszawa	678 512 PLN
Wrocław	675 240 PLN
Grand Total	**659498,0074**

Location	Avg Price	Avg Rent	Approx. ROI
Kraków	647 770 PLN	3 073 PLN	✗ 0,057
Łódź	556 213 PLN	2 974 PLN	✓ 0,064
Poznań	589 532 PLN	3 147 PLN	✓ 0,064
Warszawa	678 512 PLN	3 797 PLN	✓ 0,067
Wrocław	675 240 PLN	3 272 PLN	✗ 0,058

Row Labels	Average of Rent
1	2 852 PLN
2	3 318 PLN
3	3 956 PLN
4	4 489 PLN
5	4 960 PLN
Grand Total	**3568,505607**

Row Labels	Average of Price
1	535 636 PLN
2	612 715 PLN
3	708 122 PLN
4	753 675 PLN
5	823 025 PLN
Grand Total	**659498,0074**

Rooms	Avg Price	Avg Rent	Approx. ROI
5 rooms	823 025 PLN	4 960 PLN	✓ 0,072
4 rooms	753 675 PLN	4 489 PLN	✓ 0,071
3 rooms	708 122 PLN	3 956 PLN	! 0,067
2 rooms	612 715 PLN	3 318 PLN	✗ 0,065
1 room	535 636 PLN	2 852 PLN	✗ 0,064

Row Labels	Average of Rent
20-30	2 544 PLN
30-40	2 839 PLN
40-50	3 106 PLN
50-60	3 359 PLN
60-70	3 648 PLN
70-80	3 883 PLN
80-90	4 528 PLN
90-100	4 618 PLN
100-110	4 650 PLN
110-120	5 319 PLN
Grand Total	**3568,505607**

Row Labels	Average of Price
20-30	490 478 PLN
30-40	549 579 PLN
40-50	606 448 PLN
50-60	662 520 PLN
60-70	707 458 PLN
70-80	754 243 PLN
80-90	771 771 PLN
90-100	793 967 PLN
100-110	911 652 PLN
110-120	932 710 PLN
Grand Total	**659498,0074**

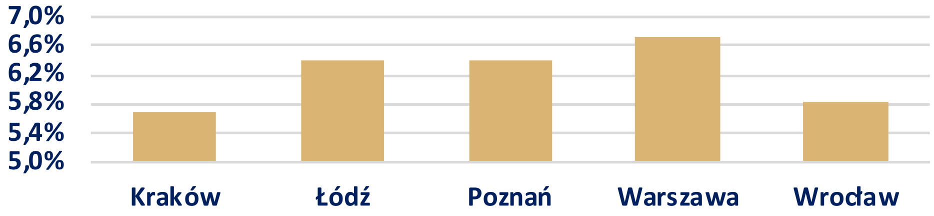
Area	Avg Price	Avg Rent	Approx. ROI
20-30 m2	490 478 PLN	2 544 PLN	✗ 0,062
30-40 m2	549 579 PLN	2 839 PLN	✗ 0,062
40-50 m2	606 448 PLN	3 106 PLN	✗ 0,061
50-60 m2	662 520 PLN	3 359 PLN	✗ 0,061
60-70 m2	707 458 PLN	3 648 PLN	✗ 0,062
70-80 m2	754 243 PLN	3 883 PLN	✗ 0,062
80-90 m2	771 771 PLN	4 528 PLN	✓ 0,070
90-100 m2	793 967 PLN	4 618 PLN	✓ 0,070
100-110 m2	911 652 PLN	4 650 PLN	✗ 0,061
110-120 m2	932 710 PLN	5 319 PLN	✓ 0,068

6. Data Visualisation

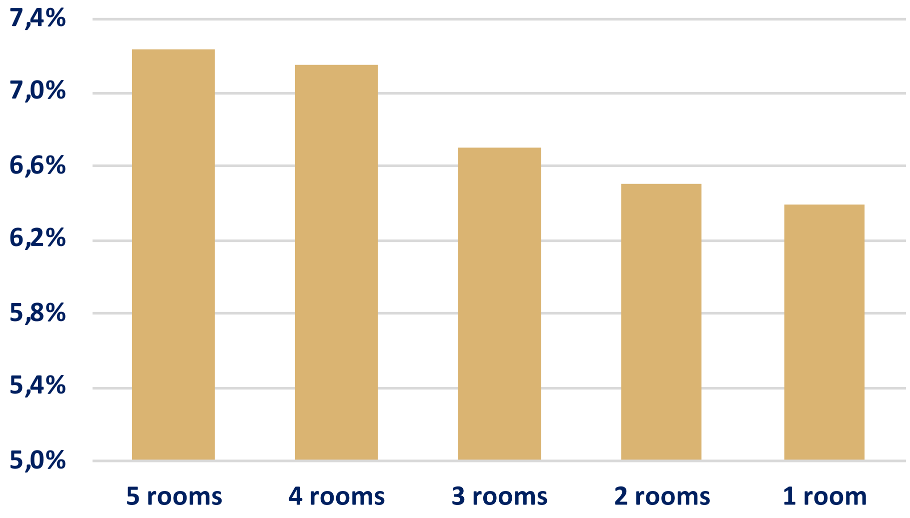
Apartment for rent analysis

Location	Average Price	Average Rent
Kraków	647 770 PLN	3 073 PLN
Łódź	556 213 PLN	2 974 PLN
Poznań	589 532 PLN	3 147 PLN
Warszawa	678 512 PLN	3 797 PLN
Wrocław	675 240 PLN	3 272 PLN

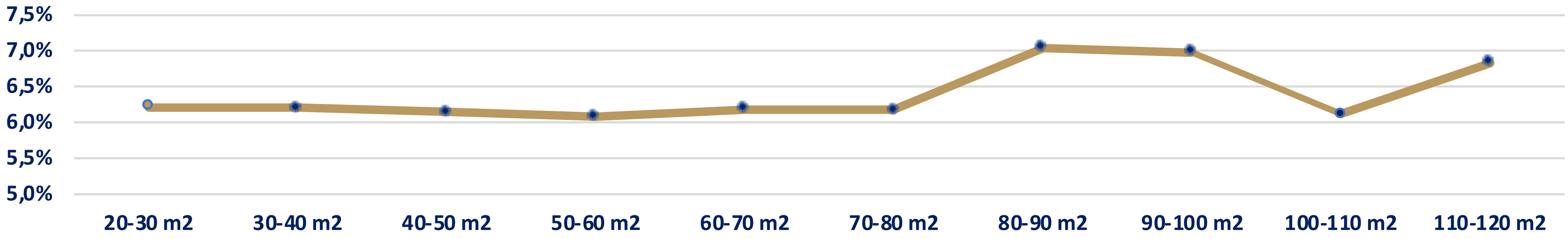
Approx. ROI compare to: location



Approx. ROI compare to: number of rooms



Approx. ROI compare to: area [m2]



7. Summary

Answering the initial questions asked:

1. In which city (over 500k population) will the purchase of an apartment for rent bring the highest ROI?
   Warszawa
2. What number of square meters will bring the highest ROI?
3. 80-100m2
4. What number of rooms will bring the highest ROI?
   5 rooms - due to the lack of sufficient data (lack of offers), ROI cannot be calculated for apartments with more than 5 rooms.

After the analysis, it is worth noting that the greater the number of rooms in the apartment, the greater the estimated ROI. The difference in ROI between a 1-room apartment and a 5-room apartment is about 0.8 percentage point. Therefore, it should be considered whether having a larger sum of money it would not be more profitable to buy two smaller flats than one large one, however this is not the subject of current considerations.