Аналитический отчет Парсинг данных:

```python
import cianparser
from time import sleep
a=0
while a < 52:
    data = cianparser.CianParser(location="Москва").get_flats(deal_type="sale", rooms=(3,4), with_saving_csv=True, with_extra_data=True, additional_settings={"start_page":1+a,
    "end_page":2+a})
    sleep(120)
    a+=2
```

Собираем все данные о квартирах с помощью библиотеки cian. После того как мы спарсим нужное кол-во строк, нам нужно произвести подготовку данных для анализа: проверка на пропуски, выбросы и ошибки. Обработать выявленные аномалии (удалить / заполнить)

1) Я решил удалить столбцы author, author_type, url, deal_type и accommodation_type из нашего CSV файла с данными о квартирах по следующим причинам:

А) Необходимость в релевантной информации: Основная цель анализа данных о квартирах заключается в оценке их стоимости и сравнении цен за квадратный метр. Столбцы, такие как author и author_type, содержат информацию о том, кто разместил объявление, что не влияет на фактическую стоимость недвижимости. Эти данные не имеют отношения к характеристикам квартиры, которые могут повлиять на цену.

Б) Фокус на ценовых показателях: Для анализа рынка недвижимости важнее сосредоточиться на факторах, которые непосредственно влияют на цену, таких как площадь квартиры, местоположение и другие характеристики. Столбцы deal_type и accommodation_type могут быть полезны в определенных контекстах, но для расчета цены за квадратный метр они не являются критически важными, так как цена может варьироваться в зависимости от других более значимых факторов.

```python
# Загрузка данных из CSV файла
file_path = r'C:\Users\user\Desktop\intensiv\merged_file.csv'  # Укажите путь к вашему CSV файлу
df = pd.read_csv(file_path)

# Удаление указанных столбцов
columns_to_drop = ['author', 'author_type', 'url', 'deal_type', 'accommodation_type']
df.drop(columns=columns_to_drop, inplace=True)

# Сохранение обновленного DataFrame обратно в CSV файл
df.to_csv('updated_apartments.csv', index=False)
```

```
Процент пропусков в колонках:
location                0.000000
deal_type               0.000000
accommodation_type      0.000000
floor                   0.000000
floors_count            0.000000
rooms_count             0.000000
total_meters            0.000000
price                   0.487673
district                9.428339
street                 14.020591
house_number           15.253319
underground            17.691682
residential_complex    25.995665
year_of_construction   25.020320
object_type            25.020320
house_material_type    25.020320
heating_type           25.020320
finish_type            25.020320
living_meters          25.020320
kitchen_meters         25.020320
price_per_meter         0.487673
dtype: float64
```

2)Анализируем какие колонки имеют пропуски, видим,что пропусков достаточно много. Я решил удалить строки, в которых отстутсвует год постройки·

```python
# Удаление строк, где отсутствует год постройки
df_cleaned = df.dropna(subset=['year_of_construction'])

# Сохранение очищенного DataFrame обратно в CSV файл
df_cleaned.to_csv('cleaned_apartments.csv', index=False)
```

3) После удаления строк, снова смотрим пусты значения.

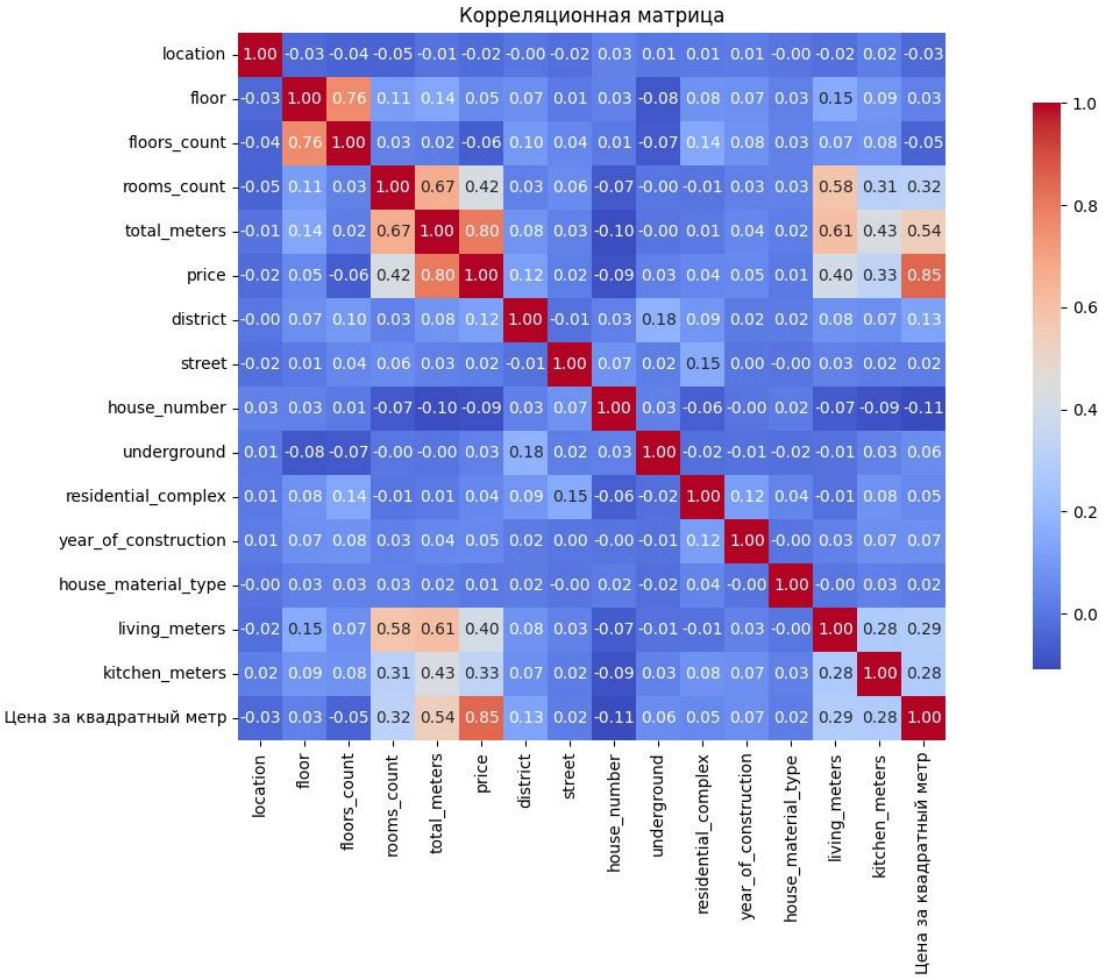
Видим, что в колонке метро есть много пропусков. Буду заполнять колонку underground. Часть процентов я заполнил модой, другую медианой. Колонку house_number заполним медианой, так как этот фактор не сильно влияет на цену квадратного метра.
Так же нужно заполнить значения -1 в колонках. Заполнил разные колонки по разному. Колонку с годом постройки, определил медиану и по ней заполнил. Квадратные метры кухни и жилой площади заполнял по среднему. Смотрел сколько квадратных метров квартира, и в процентном соотношении заполнил аномалии

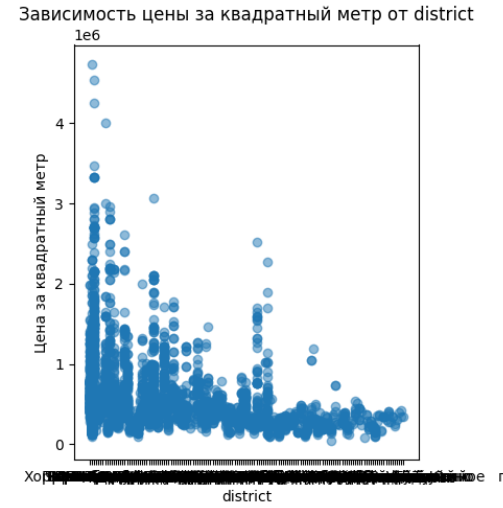
```
location                0.000000
deal_type               0.000000
accommodation_type      0.000000
floor                   0.000000
floors_count            0.000000
rooms_count             0.000000
total_meters            0.000000
price                   0.469738
district               11.833785
street                 15.627823
house_number           17.127371
underground            23.197832
residential_complex    27.588076
year_of_construction    0.000000
object_type             0.000000
house_material_type     0.000000
heating_type            0.000000
finish_type             0.000000
living_meters           0.000000
```

Вывод:

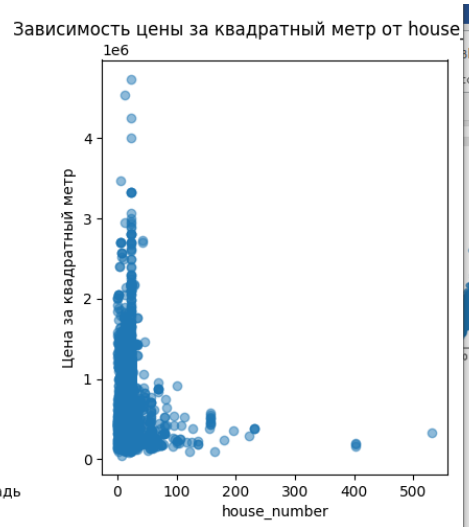
В процессе анализа рынка недвижимости мы пришли к важному выводу: цена за квадратный метр квартиры в значительной степени определяется тремя ключевыми факторами.

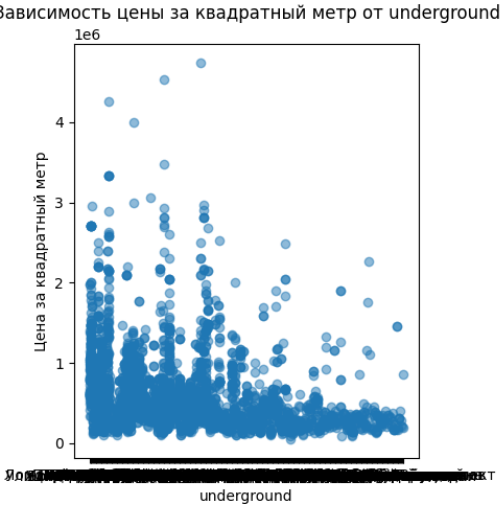
1) **Площадь квартиры**
2) **Количество комнат**
3) **Локация**




Зависимость цены за квадратный метр от district
1e6
Цена за квадратный метр
district


Зависимость цены за квадратный метр от street
1e6
Цена за квадратный метр
street


Зависимость цены за квадратный метр от house_
1e6
Цена за квадратный метр
house_number


Зависимость цены за квадратный метр от underground
1e6
Цена за квадратный метр
underground


Зависимость цены за квадратный метр от residential_complex
1e6
Цена за квадратный метр
residential_complex

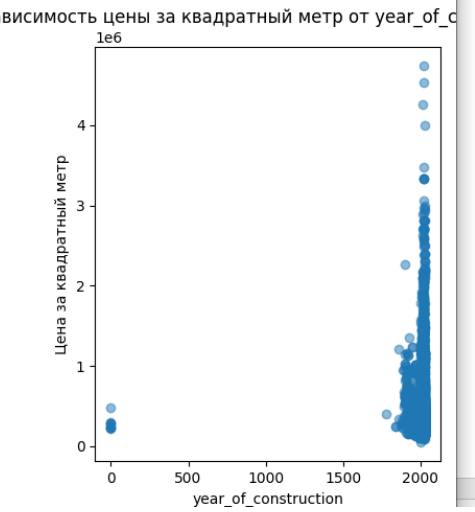
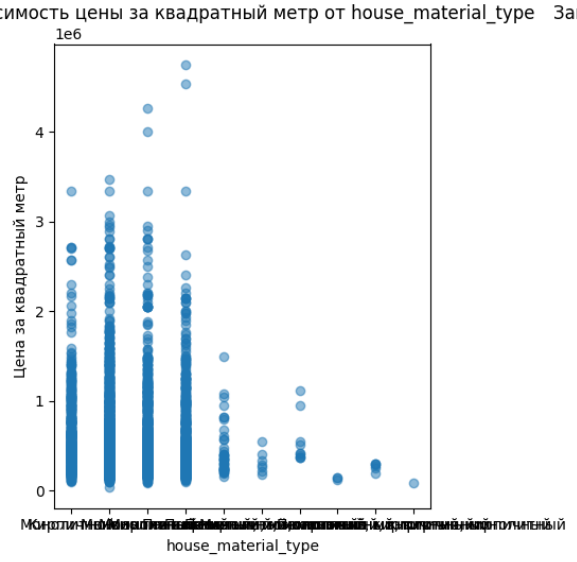
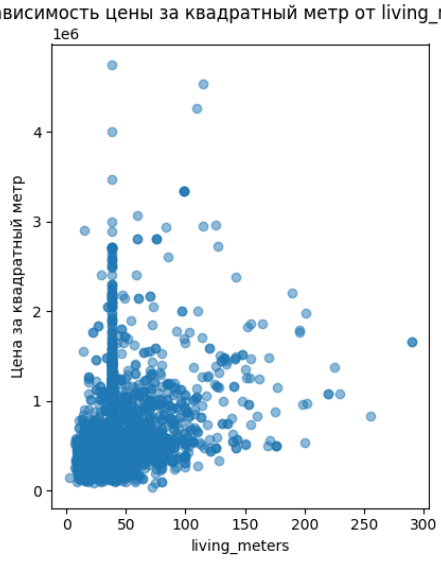
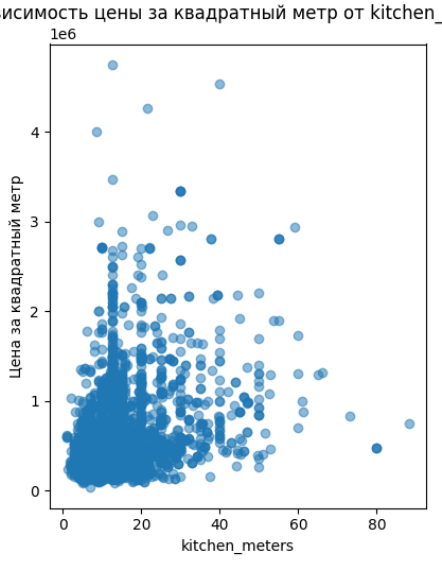
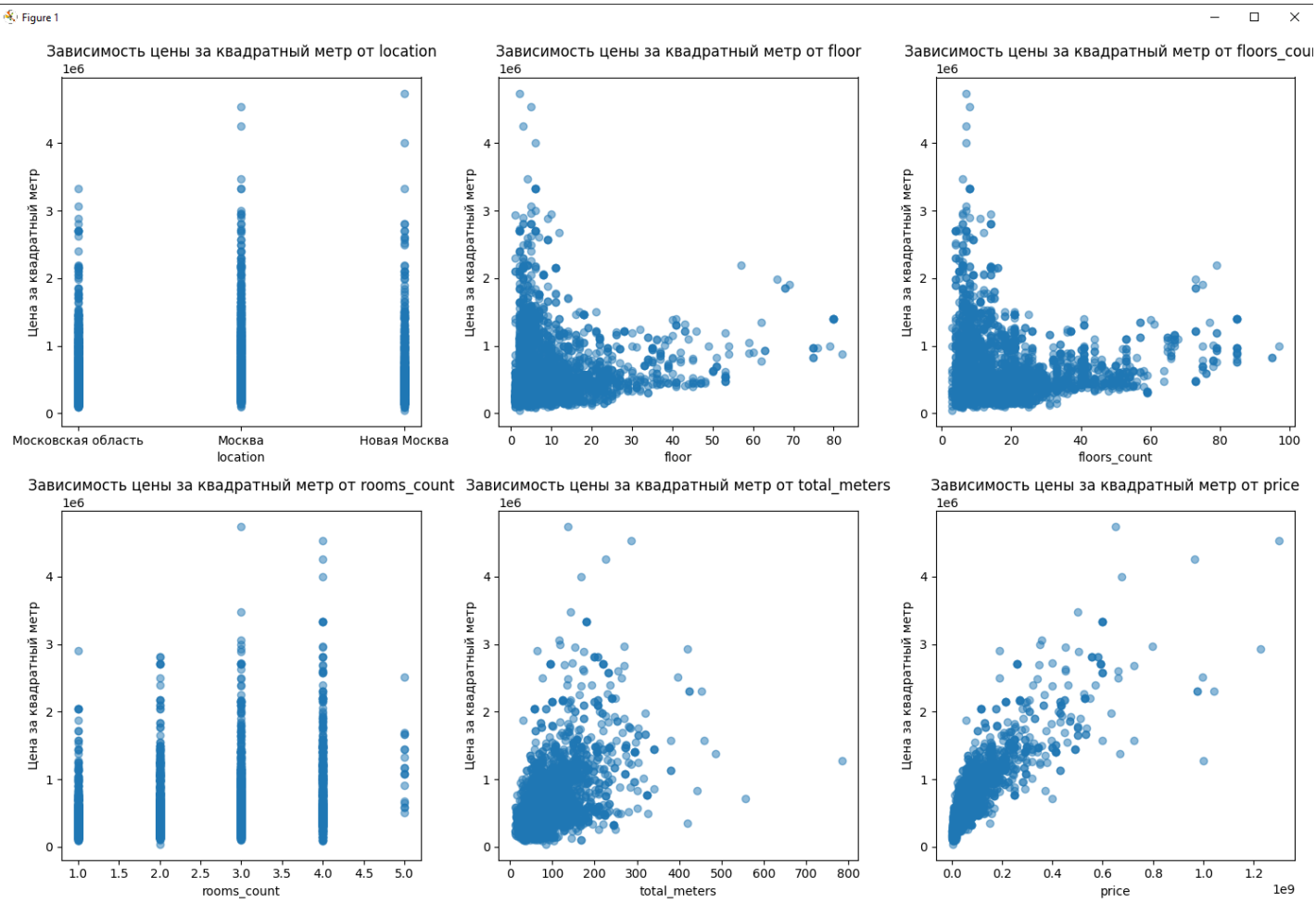

Зависимость цены за квадратный метр от year_of_c
1e6
Цена за квадратный метр
year_of_construction

Figure 1


симость цены за квадратный метр от house_material_type
1e6
Цена за квадратный метр
house_material_type


Зависимость цены за квадратный метр от living_meters
1e6
Цена за квадратный метр
living_meters


Зависимость цены за квадратный метр от kitchen_
1e6
Цена за квадратный метр
kitchen_meters



Рефлексия о проделанной работе

По окончанию работы можно провести рефлексию: что у меня получилось, что не получилось, что можно улучшить в представленной работе.

1. **Загрузка и очистка данных**: Я смог загрузить данные из CSV файла и провести их предварительную очистку, удалив ненужные столбцы и строки с пропущенными значениями. Это создало основу для дальнейшего анализа.
2. **Кодирование категориальных переменных**: Использование LabelEncoder для кодирования категориальных переменных прошло успешно. Это позволило преобразовать текстовые данные в числовые, что является важным шагом для последующего анализа.
3. **Визуализация данных**: Я создал тепловую карту для визуализации корреляционной матрицы и тепловую карту пропусков. Эти визуализации помогают лучше понять структуру данных и выявить взаимосвязи между переменными.

Что не получилось
Несмотря на достигнутые успехи, были и некоторые трудности:

1. **Обработка ошибок**: В процессе работы возникали ошибки, связанные с отсутствующими столбцами. Хотя я смог их устранить, это потребовало дополнительных усилий и времени. В будущем стоит заранее проверять наличие всех необходимых столбцов перед началом обработки данных.

2. **Оптимизация кода**: Код можно было бы сделать более компактным и читаемым. Например, можно было бы объединить некоторые шаги очистки данных в одну функцию, чтобы избежать дублирования кода.

Что можно улучшить

1. **Документация и комментарии**: Важно добавить больше комментариев к коду, чтобы сделать его более понятным для других пользователей. Это поможет лучше понять логику работы и упростит дальнейшую поддержку.
2. **Расширение анализа**: В будущем можно рассмотреть возможность проведения более глубокого анализа данных, включая использование других методов визуализации и статистических тестов для выявления значимых взаимосвязей.