Исследование объявлений о продаже квартир

В вашем распоряжении данные сервиса Яндекс.Недвижимость — архив объявлений о продаже квартир в Санкт-Петербурге и соседних населённых пунктов за несколько лет. Нужно научиться определять рыночную стоимость объектов недвижимости. Ваша задача — установить параметры. Это позволит построить автоматизированную систему: она отследит аномалии и мошенническую деятельность.

По каждой квартире на продажу доступны два вида данных. Первые вписаны пользователем, вторые — получены автоматически на основе картографических данных. Например, расстояние до центра, аэропорта, ближайшего парка и водоёма.

Ваша задача — выполнить предобработку данных и изучить их, чтобы найти интересные особенности и зависимости, которые существуют на рынке недвижимости. О каждой квартире в базе содержится два типа данных: добавленные пользователем и картографические. Например, к первому типу относятся площадь квартиры, её этаж и количество балконов, ко второму — расстояния до центра города, аэропорта и ближайшего парка.

Инструкция по выполнению проекта

Шаг 1. Откройте файл с данными и изучите общую информацию

- Путь к файлу: /datasets/real_estate_data.csv
- Скачать датасет
- Загрузите данные из файла в датафрейм.
- Изучите общую информацию о полученном датафрейме.
- Постройте общую гистограмму для всех числовых столбцов таблицы. Например, для датафрейма data это можно сделать командой data.hist(figsize=(15, 20)).

Шаг 2. Предобработка данных

Найдите и изучите пропущенные значения в столбцах:

- Определите, в каких столбцах есть пропуски.
- Заполните пропущенные значения там, где это возможно. Например, если продавец не указал число балконов, то, скорее всего, в его квартире их нет. Такие пропуски правильно заменить на 0. Если логичную замену предложить невозможно, то оставьте эти значения пустыми. Пропуски — тоже важный сигнал, который нужно учитывать.
- В ячейке с типом markdown укажите причины, которые могли привести к пропускам в данных.

Рассмотрите типы данных в каждом столбце:

- Найдите столбцы, в которых нужно изменить тип данных.
- Преобразуйте тип данных в выбранных столбцах.

В ячейке с типом markdown поясните, почему нужно изменить тип данных. Изучите уникальные значения в столбце с названиями и устраните неявные дубликаты. Например, «поселок Рябово» и «поселок городского типа Рябово», «поселок Тельмана» и «посёлок Тельмана» — это обозначения одних и тех же населённых пунктов. Вы можете заменить названия в существующем столбце или создать новый с названиями без дубликатов.

Найдите и устраните редкие и выбивающиеся значения. Например, в столбце ceiling_height может быть указана высота потолков 25 м и 32 м. Логично предположить, что на самом деле это вещественные значения: 2.5 м и 3.2 м.

Попробуйте обработать аномалии в этом и других столбцах.

Если природа аномалии понятна и данные действительно искажены, то восстановите корректное значение.

В противном случае удалите редкие и выбивающиеся значения.

В ячейке с типом markdown опишите, какие особенности в данных вы обнаружили.

Шаг 3. Добавьте в таблицу новые столбцы со следующими параметрами:

- цена одного квадратного метра;
- день недели публикации объявления (0 — понедельник, 1 — вторник и так далее);
- месяц публикации объявления;
- год публикации объявления;
- тип этажа квартиры (значения — «первый», «последний», «другой»);
- расстояние до центра города в километрах (переведите из м в км и округлите до целых значений).

Шаг 4. Проведите исследовательский анализ данных: Изучите следующие параметры объектов:

- общая площадь;
- жилая площадь;
- площадь кухни;
- цена объекта;
- количество комнат;
- высота потолков;
- этаж квартиры;
- тип этажа квартиры («первый», «последний», «другой»);
- общее количество этажей в доме;
- расстояние до центра города в метрах;
- расстояние до ближайшего аэропорта;
- расстояние до ближайшего парка;
- день и месяц публикации объявления.

Постройте отдельные гистограммы для каждого из этих параметров. Опишите все ваши наблюдения по параметрам в ячейке с типом markdown.

Изучите, как быстро продавались квартиры (столбец days_exposition). Этот параметр показывает, сколько дней было размещено каждое объявление.

Постройте гистограмму.

Посчитайте среднее и медиану.

В ячейке типа markdown опишите, сколько времени обычно занимает продажа. Какие продажи можно считать быстрыми, а какие — необычно долгими? Какие факторы больше всего влияют на общую (полную) стоимость объекта?

Изучите, зависит ли цена от:

- общей площади;

- жилой площади;
- площади кухни;
- количества комнат;
- этажа, на котором расположена квартира (первый, последний, другой);
- даты размещения (день недели, месяц, год).

Постройте графики, которые покажут зависимость цены от указанных выше параметров. Для подготовки данных перед визуализацией вы можете использовать сводные таблицы.

Посчитайте среднюю цену одного квадратного метра в 10 населённых пунктах с наибольшим числом объявлений. Выделите населённые пункты с самой высокой и низкой стоимостью квадратного метра. Эти данные можно найти по имени в столбце locality_name.

Ранее вы посчитали расстояние до центра в километрах. Теперь выделите квартиры в Санкт-Петербурге с помощью столбца locality_name и вычислите среднюю цену каждого километра. Опишите, как стоимость объектов зависит от расстояния до центра города.

Шаг 5. Напишите общий вывод

Опишите полученные результаты и зафиксируйте основной вывод проведённого исследования.

Описание данных

- airports_nearest — расстояние до ближайшего аэропорта в метрах (м)
- balcony — число балконов
- ceiling_height — высота потолков (м)
- cityCenters_nearest — расстояние до центра города (м)
- days_exposition — сколько дней было размещено объявление (от публикации до снятия)
- first_day_exposition — дата публикации
- floor — этаж
- floors_total — всего этажей в доме
- is_apartment — апартаменты (булев тип)
- kitchen_area — площадь кухни в квадратных метрах (м²)
- last_price — цена на момент снятия с публикации
- living_area — жилая площадь в квадратных метрах (м²)
- locality_name — название населённого пункта
- open_plan — свободная планировка (булев тип)
- parks_around3000 — число парков в радиусе 3 км
- parks_nearest — расстояние до ближайшего парка (м)
- ponds_around3000 — число водоёмов в радиусе 3 км
- ponds_nearest — расстояние до ближайшего водоёма (м)
- rooms — число комнат
- studio — квартира-студия (булев тип)
- total_area — общая площадь квартиры в квадратных метрах (м²)
- total_images — число фотографий квартиры в объявлении

Шаг 1 — Откройте файл с данными и изучите общую информацию.

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import matplotlib as mpl
from matplotlib import pyplot as plt
%matplotlib inline
import warnings #импортируем библиотеку

warnings.filterwarnings('ignore') # отключаем предупреждения
```

In [2]:
```
df = pd.read_csv("real_estate_data.csv", sep = '\t')
df
```

Out[2]:

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total
0	20	13000000.0	108.00	2019-03-07T00:00:00	3	2.70	16.0
1	7	3350000.0	40.40	2018-12-04T00:00:00	1	NaN	11.0
2	10	5196000.0	56.00	2015-08-20T00:00:00	2	NaN	5.0
3	0	64900000.0	159.00	2015-07-24T00:00:00	3	NaN	14.0
4	2	10000000.0	100.00	2018-06-19T00:00:00	2	3.03	14.0
...	...	...	...	...	...	...	...
23694	9	9700000.0	133.81	2017-03-21T00:00:00	3	3.70	5.0
23695	14	3100000.0	59.00	2018-01-15T00:00:00	3	NaN	5.0
23696	18	2500000.0	56.70	2018-02-11T00:00:00	2	NaN	3.0
23697	13	11475000.0	76.75	2017-03-28T00:00:00	2	3.00	17.0
23698	4	1350000.0	32.30	2017-07-21T00:00:00	1	2.50	5.0

23699 rows × 22 columns

Описание данных

- airports_nearest — расстояние до ближайшего аэропорта в метрах (м)
- balcony — число балконов
- ceiling_height — высота потолков (м)
- cityCenters_nearest — расстояние до центра города (м)
- days_exposition — сколько дней было размещено объявление (от публикации до снятия)
- first_day_exposition — дата публикации
- floor — этаж
- floors_total — всего этажей в доме
- is_apartment — апартаменты (булев тип)
- kitchen_area — площадь кухни в квадратных метрах (м²)
- last_price — цена на момент снятия с публикации
- living_area — жилая площадь в квадратных метрах (м²)
- locality_name — название населённого пункта
- open_plan — свободная планировка (булев тип)
- parks_around3000 — число парков в радиусе 3 км
- parks_nearest — расстояние до ближайшего парка (м)
- ponds_around3000 — число водоёмов в радиусе 3 км

- ponds_nearest — расстояние до ближайшего водоёма (м)
- rooms — число комнат
- studio — квартира-студия (булев тип)
- total_area — общая площадь квартиры в квадратных метрах (м²)
- total_images — число фотографий квартиры в объявлении

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23699 entries, 0 to 23698
Data columns (total 22 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   total_images          23699 non-null  int64  
 1   last_price            23699 non-null  float64
 2   total_area            23699 non-null  float64
 3   first_day_exposition  23699 non-null  object 
 4   rooms                 23699 non-null  int64  
 5   ceiling_height        14504 non-null  float64
 6   floors_total          23613 non-null  float64
 7   living_area           21796 non-null  float64
 8   floor                 23699 non-null  int64  
 9   is_apartment          2775 non-null   object 
 10  studio                23699 non-null  bool   
 11  open_plan             23699 non-null  bool   
 12  kitchen_area          21421 non-null  float64
 13  balcony               12180 non-null  float64
 14  locality_name         23650 non-null  object 
 15  airports_nearest      18157 non-null  float64
 16  cityCenters_nearest   18180 non-null  float64
 17  parks_around3000      18181 non-null  float64
 18  parks_nearest         8079 non-null   float64
 19  ponds_around3000      18181 non-null  float64
 20  ponds_nearest         9110 non-null   float64
 21  days_exposition       20518 non-null  float64
dtypes: bool(2), float64(14), int64(3), object(3)
memory usage: 3.7+ MB
```

In [4]:
```
df.shape
```

Out[4]: (23699, 22)

В файле 23699 строк и 22 столбца. 2 столбца - типа bool, 14 - float64, 3 - int64 и три переменных - категориальные (object). В столбцах есть пропуски, на них мы обратим внимание на втором шаге.

In [5]:
```
df.dtypes
```

Out[5]:
```
total_images              int64
last_price              float64
total_area              float64
first_day_exposition     object
rooms                     int64
ceiling_height          float64
floors_total            float64
living_area             float64
floor                     int64
is_apartment             object
studio                     bool
open_plan                  bool
kitchen_area            float64
balcony                 float64
locality_name            object
airports_nearest        float64
cityCenters_nearest     float64
parks_around3000        float64
parks_nearest           float64
ponds_around3000        float64
ponds_nearest           float64
days_exposition         float64
dtype: object
```

- first_day_exposition - object - должен быть тип datetime
- floors_total - должен быть тип int, т.к. количество этажей - целое значение
- is_apartment - должен быть тип bool
- parks_around3000, ponds_around3000 - тип должен быть int
- days_exposition - тип должен быть int
- last_price, balcony - перевести в int

In [6]:
```
df.describe()
```

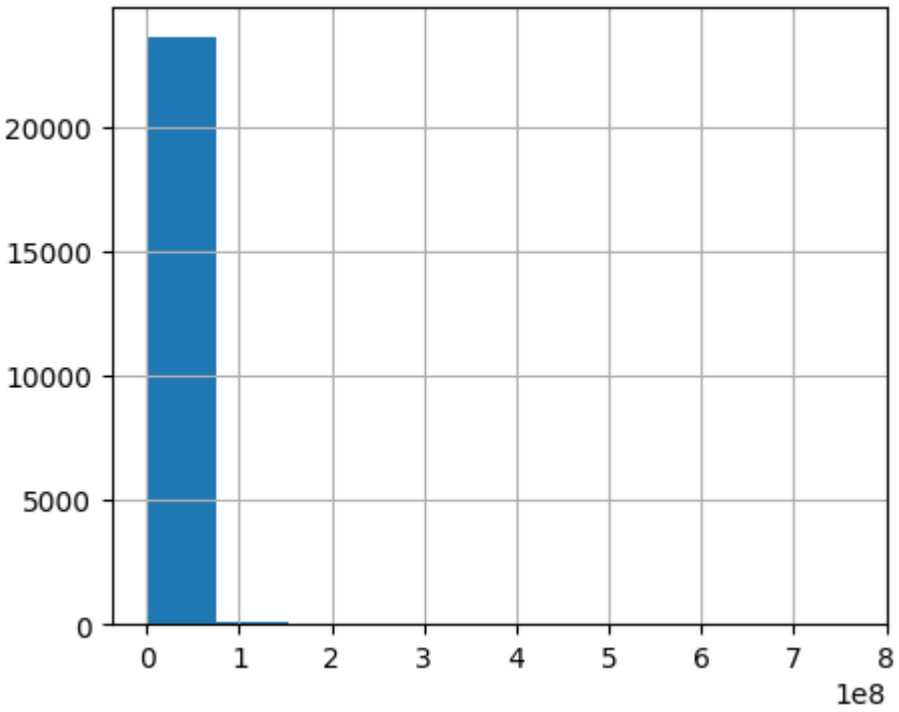
Out[6]:

	total_images	last_price	total_area	rooms	ceiling_height	floors_total	liv
count	23699.000000	2.369900e+04	23699.000000	23699.000000	14504.000000	23613.000000	2179
mean	9.858475	6.541549e+06	60.348651	2.070636	2.771499	10.673824	3
std	5.682529	1.088701e+07	35.654083	1.078405	1.261056	6.597173	2
min	0.000000	1.219000e+04	12.000000	0.000000	1.000000	1.000000	
25%	6.000000	3.400000e+06	40.000000	1.000000	2.520000	5.000000	1
50%	9.000000	4.650000e+06	52.000000	2.000000	2.650000	9.000000	3
75%	14.000000	6.800000e+06	69.900000	3.000000	2.800000	16.000000	4
max	50.000000	7.630000e+08	900.000000	19.000000	100.000000	60.000000	40

- total_images - максимамльное значение 50 (обратить внимание);
- total_area максимамльное значение 900 кв. м. обратить внимание. Данные скошены вправо;
- rooms максимамльное значение 19. обратить внимание. Данные скошены вправо;
- 100.000000 - высота потолка;
- 1580 - дней размещено объявление сильно завышено;
- есть значения 0 на них нужно посмотреть (например, rooms !!! ponds_around3000 ponds_nearest).

Есть данные которые распределены неравномерно, по видимому есть роскошные квартиры с большой площадью и ценой.

In [7]:
```
df['last_price'].hist(figsize=(5, 4));
```



In [ ]:
```
```

Обратить внимание на признаки у которых есть 0 значение.

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
0
```

Шаг 2 — Предобработка данных

In [9]:
```
df.columns.tolist()
```

Out[9]:
```
['total_images',
 'last_price',
 'total_area',
 'first_day_exposition',
 'rooms',
 'ceiling_height',
 'floors_total',
 'living_area',
 'floor',
 'is_apartment',
 'studio',
 'open_plan',
 'kitchen_area',
 'balcony',
 'locality_name',
 'airports_nearest',
 'cityCenters_nearest',
 'parks_around3000',
 'parks_nearest',
 'ponds_around3000',
 'ponds_nearest',
 'days_exposition']
```

Переименуем столбцы

In [10]:
```
df.rename(columns={'cityCenters_nearest' : 'city_centers_nearest'}, inplace=True)
```

In [11]:
```
df.columns.tolist()
```

Out[11]:
```
['total_images',
 'last_price',
 'total_area',
 'first_day_exposition',
 'rooms',
 'ceiling_height',
 'floors_total',
 'living_area',
 'floor',
 'is_apartment',
 'studio',
 'open_plan',
 'kitchen_area',
 'balcony',
 'locality_name',
 'airports_nearest',
 'city_centers_nearest',
 'parks_around3000',
 'parks_nearest',
 'ponds_around3000',
 'ponds_nearest',
 'days_exposition']
```

In [12]:
```
df['locality_name'].unique().tolist()
```

Out[12]:
```
['Санкт-Петербург',
 'посёлок Шушары',
 'городской посёлок Янино-1',
 'посёлок Парголово',
 'посёлок Мурино',
 'Ломоносов',
 'Сертолово',
 'Петергоф',
 'Пушкин',
 'деревня Кудрово',
 'Коммунар',
 'Колпино',
 'поселок городского типа Красный Бор',
 'Гатчина',
 'поселок Мурино',
 'деревня Фёдоровское',
 'Выборг',
 'Кронштадт',
 'Кировск',
 'деревня Новое Девяткино',
 'посёлок Металлострой',
 'посёлок городского типа Лебяжье',
 'посёлок городского типа Сиверский',
 'поселок Молодцово',
 'поселок городского типа Кузьмоловский',
 'садовое товарищество Новая Ропша',
 'Павловск',
 'деревня Пикколово',
 'Всеволожск',
 'Волхов',
 'Кингисепп',
 'Приозерск',
 'Сестрорецк',
 'деревня Куттузи',
 'посёлок Аннино',
 'поселок городского типа Ефимовский',
 'посёлок Плодовое',
 'деревня Заклинье',
 'поселок Торковичи',
 'поселок Первомайское',
 'Красное Село',
 'посёлок Понтонный',
 'Сясьстрой',
 'деревня Старая',
 'деревня Лесколово',
 'посёлок Новый Свет',
 'Сланцы',
 'село Путилово',
 'Ивангород',
 'Мурино',
 'Шлиссельбург',
 'Никольское',
 'Зеленогорск',
 'Сосновый Бор',
 'поселок Новый Свет',
 'деревня Оржицы',
 'деревня Кальтино',
 'Кудрово',
 'поселок Романовка',
 'посёлок Бугры',
 'поселок Бугры',
 'поселок городского типа Рощино',
 'Кириши',
 'Луга',
 'Волосово',
```

```
'Отрадное',
'село Павлово',
'поселок Оредеж',
'село Копорье',
'посёлок городского типа Красный Бор',
'посёлок Молодёжное',
'Тихвин',
'посёлок Победа',
'деревня Нурма',
'поселок городского типа Синявино',
'Тосно',
'посёлок городского типа Кузьмоловский',
'посёлок Стрельна',
'Бокситогорск',
'посёлок Александровская',
'деревня Лопухинка',
'Пикалёво',
'поселок Терволово',
'поселок городского типа Советский',
'Подпорожье',
'посёлок Петровское',
'посёлок городского типа Токсово',
'поселок Сельцо',
'посёлок городского типа Вырица',
'деревня Кипень',
'деревня Келози',
'деревня Вартемяги',
'посёлок Тельмана',
'поселок Севастьяново',
'городской поселок Большая Ижора',
nan,
'городской посёлок Павлово',
'деревня Агалатово',
'посёлок Новогорелово',
'городской посёлок Лесогорский',
'деревня Лаголово',
'поселок Цвелодубово',
'поселок городского типа Рахья',
'поселок городского типа Вырица',
'деревня Белогорка',
'поселок Заводской',
'городской посёлок Новоселье',
'деревня Большие Колпаны',
'деревня Горбунки',
'деревня Батово',
'деревня Заневка',
'деревня Иссад',
'Приморск',
'городской посёлок Фёдоровское',
'деревня Мистолово',
'Новая Ладога',
'поселок Зимитицы',
'поселок Барышево',
'деревня Разметелево',
'поселок городского типа имени Свердлова',
'деревня Пеники',
'поселок Рябово',
'деревня Пудомяги',
'поселок станции Корнево',
'деревня Низино',
'деревня Бегуницы',
'посёлок Поляны',
'городской посёлок Мга',
'поселок Елизаветино',
'посёлок городского типа Кузнечное',
'деревня Колтуши',
```

```
'поселок Запорожское',
'посёлок городского типа Рощино',
'деревня Гостилицы',
'деревня Малое Карлино',
'посёлок Мичуринское',
'посёлок городского типа имени Морозова',
'посёлок Песочный',
'посёлок Сосново',
'деревня Аро',
'поселок Ильичёво',
'посёлок городского типа Тайцы',
'деревня Малое Верево',
'деревня Извара',
'поселок станции Вещево',
'село Паша',
'деревня Калитино',
'посёлок городского типа Ульяновка',
'деревня Чудской Бор',
'поселок городского типа Дубровка',
'деревня Мины',
'поселок Войсковицы',
'посёлок городского типа имени Свердлова',
'деревня Коркино',
'посёлок Ропша',
'поселок городского типа Приладожский',
'посёлок Щеглово',
'посёлок Гаврилово',
'Лодейное Поле',
'деревня Рабитицы',
'поселок городского типа Никольский',
'деревня Кузьмолово',
'деревня Малые Колпаны',
'поселок Тельмана',
'посёлок Петро-Славянка',
'городской посёлок Назия',
'посёлок Репино',
'посёлок Ильичёво',
'поселок Углово',
'поселок Старая Малукса',
'садовое товарищество Рахья',
'поселок Аннино',
'поселок Победа',
'деревня Меньково',
'деревня Старые Бегуницы',
'посёлок Сапёрный',
'поселок Семрино',
'поселок Гаврилово',
'поселок Глажево',
'поселок Кобринское',
'деревня Гарболово',
'деревня Юкки',
'поселок станции Приветнинское',
'деревня Мануйлово',
'деревня Пчева',
'поселок Поляны',
'поселок Цвылёво',
'поселок Мельниково',
'посёлок Пудость',
'посёлок Усть-Луга',
'Светогорск',
'Любань',
'поселок Селезнёво',
'поселок городского типа Рябово',
'Каменногорск',
'деревня Кривко',
'поселок Глебычево',
```

'деревня Парицы',
'поселок Жилпосёлок',
'посёлок городского типа Мга',
'городской поселок Янино-1',
'посёлок Войскорово',
'село Никольское',
'посёлок Терволово',
'поселок Стеклянный',
'посёлок городского типа Важины',
'посёлок Мыза-Ивановка',
'село Русско-Высоцкое',
'поселок городского типа Лебяжье',
'поселок городского типа Форносово',
'село Старая Ладога',
'поселок Житково',
'городской посёлок Виллози',
'деревня Лампово',
'деревня Шпаньково',
'деревня Лаврики',
'посёлок Сумино',
'посёлок Возрождение',
'деревня Старосиверская',
'посёлок Кикерино',
'поселок Возрождение',
'деревня Старое Хинколово',
'посёлок Пригородный',
'посёлок Торфяное',
'городской посёлок Будогощь',
'поселок Суходолье',
'поселок Красная Долина',
'деревня Хапо-Ое',
'поселок городского типа Дружная Горка',
'поселок Лисий Нос',
'деревня Яльгелево',
'посёлок Стеклянный',
'село Рождествено',
'деревня Старополье',
'посёлок Левашово',
'деревня Сяськелево',
'деревня Камышовка',
'садоводческое некоммерческое товарищество Лесная Поляна',
'деревня Хязельки',
'поселок Жилгородок',
'посёлок городского типа Павлово',
'деревня Ялгино',
'поселок Новый Учхоз',
'городской посёлок Рощино',
'поселок Гончарово',
'поселок Почап',
'посёлок Сапёрное',
'посёлок Платформа 69-й километр',
'поселок Каложицы',
'деревня Фалилеево',
'деревня Пельгора',
'поселок городского типа Лесогорский',
'деревня Торошковичи',
'посёлок Белоостров',
'посёлок Алексеевка',
'поселок Серебрянский',
'поселок Лукаши',
'поселок Петровское',
'деревня Щеглово',
'поселок Мичуринское',
'деревня Тарасово',
'поселок Кингисеппский',
'посёлок при железнодорожной станции Вещево',

```
'поселок Ушаки',
'деревня Котлы',
'деревня Сижно',
'деревня Торосово',
'посёлок Форт Красная Горка',
'поселок городского типа Токсово',
'деревня Новолисино',
'посёлок станции Громово',
'деревня Глинка',
'посёлок Мельниково',
'поселок городского типа Назия',
'деревня Старая Пустошь',
'поселок Коммунары',
'поселок Починок',
'посёлок городского типа Вознесенье',
'деревня Разбегаево',
'посёлок городского типа Рябово',
'поселок Гладкое',
'посёлок при железнодорожной станции Приветнинское',
'поселок Тёсово-4',
'посёлок Жилгородок',
'деревня Бор',
'посёлок Коробицыно',
'деревня Большая Вруда',
'деревня Курковицы',
'посёлок Лисий Нос',
'городской посёлок Советский',
'посёлок Кобралово',
'деревня Суоранда',
'поселок Кобралово',
'поселок городского типа Кондратьево',
'коттеджный поселок Счастье',
'поселок Любань',
'деревня Реброво',
'деревня Зимитицы',
'деревня Тойворово',
'поселок Семиозерье',
'поселок Лесное',
'поселок Совхозный',
'поселок Усть-Луга',
'посёлок Ленинское',
'посёлок Суйда',
'посёлок городского типа Форносово',
'деревня Нижние Осельки',
'посёлок станции Свирь',
'поселок Перово',
'Высоцк',
'поселок Гарболово',
'село Шум',
'поселок Котельский',
'поселок станции Лужайка',
'деревня Большая Пустомержа',
'поселок Красносельское',
'деревня Вахнова Кара',
'деревня Пижма',
'коттеджный поселок Кивеннапа Север',
'поселок Коробицыно',
'поселок Ромашки',
'посёлок Перово',
'деревня Каськово',
'деревня Куровицы',
'посёлок Плоское',
'поселок Сумино',
'поселок городского типа Большая Ижора',
'поселок Кирпичное',
'деревня Ям-Тесово',
```

```
 'деревня Раздолье',
 'деревня Терпилицы',
 'посёлок Шугозеро',
 'деревня Ваганово',
 'поселок Пушное',
 'садовое товарищество Садко',
 'посёлок Усть-Ижора',
 'деревня Выскатка',
 'городской посёлок Свирьстрой',
 'поселок Громово',
 'деревня Кисельня',
 'посёлок Старая Малукса',
 'деревня Трубников Бор',
 'поселок Калитино',
 'посёлок Высокоключевой',
 'садовое товарищество Приладожский',
 'посёлок Пансионат Зелёный Бор',
 'деревня Ненимяки',
 'поселок Пансионат Зелёный Бор',
 'деревня Снегирёвка',
 'деревня Рапполово',
 'деревня Пустынка',
 'поселок Рабитицы',
 'деревня Большой Сабск',
 'деревня Русско',
 'деревня Лупполово',
 'деревня Большое Рейзино',
 'деревня Малая Романовка',
 'поселок Дружноселье',
 'поселок Пчевжа',
 'поселок Володарское',
 'деревня Нижняя',
 'коттеджный посёлок Лесное',
 'деревня Тихковицы',
 'деревня Борисова Грива',
 'посёлок Дзержинского']
```

In [13]:
```
df['locality_name'] = df['locality_name'].str.lower()
```

Проблемы с буквой ё

In [14]:
```
df['locality_name'] = df['locality_name'].str.replace('ё', 'е', regex=True)
```

In [15]:
```
df['locality_name'] = df['locality_name'].str.replace('городской поселок','поселок го
```

In [16]:
```
df['locality_name'].unique().tolist()
```

Out[16]:
```
['санкт-петербург',
 'поселок шушары',
 'поселок городского типа янино-1',
 'поселок парголово',
 'поселок мурино',
 'ломоносов',
 'сертолово',
 'петергоф',
 'пушкин',
 'деревня кудрово',
 'коммунар',
 'колпино',
 'поселок городского типа красный бор',
 'гатчина',
 'деревня федоровское',
 'выборг',
 'кронштадт',
 'кировск',
 'деревня новое девяткино',
 'поселок металлострой',
 'поселок городского типа лебяжье',
 'поселок городского типа сиверский',
 'поселок молодцово',
 'поселок городского типа кузьмоловский',
 'садовое товарищество новая ропша',
 'павловск',
 'деревня пикколово',
 'всеволожск',
 'волхов',
 'кингисепп',
 'приозерск',
 'сестрорецк',
 'деревня куттузи',
 'поселок аннино',
 'поселок городского типа ефимовский',
 'поселок плодовое',
 'деревня заклинье',
 'поселок торковичи',
 'поселок первомайское',
 'красное село',
 'поселок понтонный',
 'сясьстрой',
 'деревня старая',
 'деревня лесколово',
 'поселок новый свет',
 'сланцы',
 'село путилово',
 'ивангород',
 'мурино',
 'шлиссельбург',
 'никольское',
 'зеленогорск',
 'сосновый бор',
 'деревня оржицы',
 'деревня кальтино',
 'кудрово',
 'поселок романовка',
 'поселок бугры',
 'поселок городского типа рощино',
 'кириши',
 'луга',
 'волосово',
 'отрадное',
 'село павлово',
 'поселок оредеж',
```

```
'село копорье',
'поселок молодежное',
'тихвин',
'поселок победа',
'деревня нурма',
'поселок городского типа синявино',
'тосно',
'поселок стрельна',
'бокситогорск',
'поселок александровская',
'деревня лопухинка',
'пикалево',
'поселок терволово',
'поселок городского типа советский',
'подпорожье',
'поселок петровское',
'поселок городского типа токсово',
'поселок сельцо',
'поселок городского типа вырица',
'деревня кипень',
'деревня келози',
'деревня вартемяги',
'поселок тельмана',
'поселок севастьяново',
'поселок городского типа большая ижора',
nan,
'поселок городского типа павлово',
'деревня агалатово',
'поселок новогорелово',
'поселок городского типа лесогорский',
'деревня лаголово',
'поселок цвелодубово',
'поселок городского типа рахья',
'деревня белогорка',
'поселок заводской',
'поселок городского типа новоселье',
'деревня большие колпаны',
'деревня горбунки',
'деревня батово',
'деревня заневка',
'деревня иссад',
'приморск',
'поселок городского типа федоровское',
'деревня мистолово',
'новая ладога',
'поселок зимитицы',
'поселок барышево',
'деревня разметелево',
'поселок городского типа имени свердлова',
'деревня пеники',
'поселок рябово',
'деревня пудомяги',
'поселок станции корнево',
'деревня низино',
'деревня бегуницы',
'поселок поляны',
'поселок городского типа мга',
'поселок елизаветино',
'поселок городского типа кузнечное',
'деревня колтуши',
'поселок запорожское',
'деревня гостилицы',
'деревня малое карлино',
'поселок мичуринское',
'поселок городского типа имени морозова',
'поселок песочный',
```

```
'поселок сосново',
'деревня аро',
'поселок ильичево',
'поселок городского типа тайцы',
'деревня малое верево',
'деревня извара',
'поселок станции вещево',
'село паша',
'деревня калитино',
'поселок городского типа ульяновка',
'деревня чудской бор',
'поселок городского типа дубровка',
'деревня мины',
'поселок войсковицы',
'деревня коркино',
'поселок ропша',
'поселок городского типа приладожский',
'поселок щеглово',
'поселок гаврилово',
'лодейное поле',
'деревня рабитицы',
'поселок городского типа никольский',
'деревня кузьмолово',
'деревня малые колпаны',
'поселок петро-славянка',
'поселок городского типа назия',
'поселок репино',
'поселок углово',
'поселок старая малукса',
'садовое товарищество рахья',
'деревня меньково',
'деревня старые бегуницы',
'поселок саперный',
'поселок семрино',
'поселок глажево',
'поселок кобринское',
'деревня гарболово',
'деревня юкки',
'поселок станции приветнинское',
'деревня мануйлово',
'деревня пчева',
'поселок цвылево',
'поселок мельниково',
'поселок пудость',
'поселок усть-луга',
'светогорск',
'любань',
'поселок селезнево',
'поселок городского типа рябово',
'каменногорск',
'деревня кривко',
'поселок глебычево',
'деревня парицы',
'поселок жилпоселок',
'поселок войскорово',
'село никольское',
'поселок стеклянный',
'поселок городского типа важины',
'поселок мыза-ивановка',
'село русско-высоцкое',
'поселок городского типа форносово',
'село старая ладога',
'поселок житково',
'поселок городского типа виллози',
'деревня лампово',
'деревня шпаньково',
```

```
'деревня лаврики',
'поселок сумино',
'поселок возрождение',
'деревня старосиверская',
'поселок кикерино',
'деревня старое хинколово',
'поселок пригородный',
'поселок торфяное',
'поселок городского типа будогощь',
'поселок суходолье',
'поселок красная долина',
'деревня хапо-ое',
'поселок городского типа дружная горка',
'поселок лисий нос',
'деревня яльгелево',
'село рождествено',
'деревня старополье',
'поселок левашово',
'деревня сяськелево',
'деревня камышовка',
'садоводческое некоммерческое товарищество лесная поляна',
'деревня хязельки',
'поселок жилгородок',
'деревня ялгино',
'поселок новый учхоз',
'поселок гончарово',
'поселок почап',
'поселок саперное',
'поселок платформа 69-й километр',
'поселок каложицы',
'деревня фалилеево',
'деревня пельгора',
'деревня торошковичи',
'поселок белоостров',
'поселок алексеевка',
'поселок серебрянский',
'поселок лукаши',
'деревня щеглово',
'деревня тарасово',
'поселок кингисеппский',
'поселок при железнодорожной станции вещево',
'поселок ушаки',
'деревня котлы',
'деревня сижно',
'деревня торосово',
'поселок форт красная горка',
'деревня новолисино',
'поселок станции громово',
'деревня глинка',
'деревня старая пустошь',
'поселок коммунары',
'поселок починок',
'поселок городского типа вознесенье',
'деревня разбегаево',
'поселок гладкое',
'поселок при железнодорожной станции приветнинское',
'поселок тесово-4',
'деревня бор',
'поселок коробицыно',
'деревня большая вруда',
'деревня курковицы',
'поселок кобралово',
'деревня суоранда',
'поселок городского типа кондратьево',
'коттеджный поселок счастье',
'поселок любань',
```

```
 'деревня реброво',
 'деревня зимитицы',
 'деревня тойворово',
 'поселок семиозерье',
 'поселок лесное',
 'поселок совхозный',
 'поселок ленинское',
 'поселок суйда',
 'деревня нижние осельки',
 'поселок станции свирь',
 'поселок перово',
 'высоцк',
 'поселок гарболово',
 'село шум',
 'поселок котельский',
 'поселок станции лужайка',
 'деревня большая пустомержа',
 'поселок красносельское',
 'деревня вахнова кара',
 'деревня пижма',
 'коттеджный поселок кивеннапа север',
 'поселок ромашки',
 'деревня каськово',
 'деревня куровицы',
 'поселок плоское',
 'поселок кирпичное',
 'деревня ям-тесово',
 'деревня раздолье',
 'деревня терпилицы',
 'поселок шугозеро',
 'деревня ваганово',
 'поселок пушное',
 'садовое товарищество садко',
 'поселок усть-ижора',
 'деревня выскатка',
 'поселок городского типа свирьстрой',
 'поселок громово',
 'деревня кисельня',
 'деревня трубников бор',
 'поселок калитино',
 'поселок высокоключевой',
 'садовое товарищество приладожский',
 'поселок пансионат зеленый бор',
 'деревня ненимяки',
 'деревня снегиревка',
 'деревня рапполово',
 'деревня пустынка',
 'поселок рабитицы',
 'деревня большой сабск',
 'деревня русско',
 'деревня лупполово',
 'деревня большое рейзино',
 'деревня малая романовка',
 'поселок дружноселье',
 'поселок пчевжа',
 'поселок володарское',
 'деревня нижняя',
 'коттеджный поселок лесное',
 'деревня тихковицы',
 'деревня борисова грива',
 'поселок дзержинского']
```

In [17]:

```
t = df.locality_name[df.locality_name.notnull()]
name = set(t.unique().tolist())
namep= set()
named = set()
```

In [18]:

```
for i in name:
    if 'поселок' in i :
        namep.add(i)
namep = sorted(namep)
print(*namep, sep = '\n')
```

коттеджный поселок кивеннапа север
коттеджный поселок лесное
коттеджный поселок счастье
поселок александровская
поселок алексеевка
поселок аннино
поселок барышево
поселок белоостров
поселок бугры
поселок возрождение
поселок войсковицы
поселок войскорово
поселок володарское
поселок высокоключевой
поселок гаврилово
поселок гарболово
поселок гладкое
поселок глажево
поселок глебычево
поселок гончарово
поселок городского типа большая ижора
поселок городского типа будогощь
поселок городского типа важины
поселок городского типа виллози
поселок городского типа вознесенье
поселок городского типа вырица
поселок городского типа дружная горка
поселок городского типа дубровка
поселок городского типа ефимовский
поселок городского типа имени морозова
поселок городского типа имени свердлова
поселок городского типа кондратьево
поселок городского типа красный бор
поселок городского типа кузнечное
поселок городского типа кузьмоловский
поселок городского типа лебяжье
поселок городского типа лесогорский
поселок городского типа мга
поселок городского типа назия
поселок городского типа никольский
поселок городского типа новоселье
поселок городского типа павлово
поселок городского типа приладожский
поселок городского типа рахья
поселок городского типа рощино
поселок городского типа рябово
поселок городского типа свирьстрой
поселок городского типа сиверский
поселок городского типа синявино
поселок городского типа советский
поселок городского типа тайцы
поселок городского типа токсово
поселок городского типа ульяновка
поселок городского типа федоровское
поселок городского типа форносово
поселок городского типа янино-1
поселок громово
поселок дзержинского
поселок дружноселье
поселок елизаветино
поселок жилгородок
поселок жилпоселок
поселок житково
поселок заводской
поселок запорожское
поселок зимитицы

поселок ильичево
поселок калитино
поселок каложицы
поселок кикерино
поселок кингисеппский
поселок кирпичное
поселок кобралово
поселок кобринское
поселок коммунары
поселок коробицыно
поселок котельский
поселок красная долина
поселок красносельское
поселок левашово
поселок ленинское
поселок лесное
поселок лисий нос
поселок лукаши
поселок любань
поселок мельниково
поселок металлострой
поселок мичуринское
поселок молодежное
поселок молодцово
поселок мурино
поселок мыза-ивановка
поселок новогорелово
поселок новый свет
поселок новый учхоз
поселок оредеж
поселок пансионат зеленый бор
поселок парголово
поселок первомайское
поселок перово
поселок песочный
поселок петро-славянка
поселок петровское
поселок платформа 69-й километр
поселок плодовое
поселок плоское
поселок победа
поселок поляны
поселок понтонный
поселок почап
поселок починок
поселок при железнодорожной станции вещево
поселок при железнодорожной станции приветнинское
поселок пригородный
поселок пудость
поселок пушное
поселок пчевжа
поселок рабитицы
поселок репино
поселок романовка
поселок ромашки
поселок ропша
поселок рябово
поселок саперное
поселок саперный
поселок севастьяново
поселок селезнево
поселок сельцо
поселок семиозерье
поселок семрино
поселок серебрянский
поселок совхозный

```
поселок сосново
поселок станции вещево
поселок станции громово
поселок станции корнево
поселок станции лужайка
поселок станции приветнинское
поселок станции свирь
поселок старая малукса
поселок стеклянный
поселок стрельна
поселок суйда
поселок сумино
поселок суходолье
поселок тельмана
поселок терволово
поселок тесово-4
поселок торковичи
поселок торфяное
поселок углово
поселок усть-ижора
поселок усть-луга
поселок ушаки
поселок форт красная горка
поселок цвелодубово
поселок цвылево
поселок шугозеро
поселок шушары
поселок щеглово
```

In [19]:
```
df['locality_name'] = df['locality_name'].str.replace('поселок мурино','мурино', rege
```

In [20]:
```
df['locality_name'] = df['locality_name'].str.replace('коттеджный поселок','поселок',
```

Посмотрим пропущенные значение в процентном соотношение

In [21]:
```
MissingValue = df.isnull().sum().sort_values(ascending = False)
Percent = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = False)
MissingData = pd.concat([MissingValue, Percent], axis=1, keys=['Пропущенные значения'
MissingData
```

Out[21]:

	Пропущенные значения	Процент
is_apartment	20924	88.290645
parks_nearest	15620	65.909954
ponds_nearest	14589	61.559559
balcony	11519	48.605426
ceiling_height	9195	38.799105
airports_nearest	5542	23.384953
city_centers_nearest	5519	23.287902
ponds_around3000	5518	23.283683
parks_around3000	5518	23.283683
days_exposition	3181	13.422507
kitchen_area	2278	9.612220
living_area	1903	8.029875
floors_total	86	0.362885
locality_name	49	0.206760
total_images	0	0.000000
last_price	0	0.000000
studio	0	0.000000
floor	0	0.000000
rooms	0	0.000000
first_day_exposition	0	0.000000
total_area	0	0.000000
open_plan	0	0.000000

Переведем столбец с датой в формат даты без времени, т.к. время не указано

In [22]:
```
df['first_day_exposition_copy']=df['first_day_exposition'].copy()
df['first_day_exposition'] = pd.to_datetime(df['first_day_exposition']).dt.strftime('
#df['first_day_exposition'] = pd.to_datetime(df['first_day_exposition']).dt.strftime(
#df['first_day_exposition'] = pd.to_datetime(df['first_day_exposition'], format = '%Y
df['first_day_exposition'].tolist()
```

Out[22]:
```
['2019-03-07',
 '2018-12-04',
 '2015-08-20',
 '2015-07-24',
 '2018-06-19',
 '2018-09-10',
 '2017-11-02',
 '2019-04-18',
 '2018-05-23',
 '2017-02-26',
 '2017-11-16',
 '2018-08-27',
 '2016-06-30',
 '2017-07-01',
 '2016-06-23',
 '2017-11-18',
 '2017-11-23',
 '2016-09-09',
 '2017-01-27',
 '2019-01-09',
 '2017-09-28',
 '2018-03-14',
 '2017-04-24',
 '2016-10-29',
 '2015-10-31',
 '2015-10-01',
 '2017-04-28',
 '2017-05-12',
 '2017-12-13',
 '2016-04-09',
 '2018-02-19',
 '2017-10-26',
 '2016-05-22',
 '2018-10-15',
 '2018-02-04',
 '2017-06-26',
 '2017-01-25',
 '2017-10-28',
 '2018-03-29',
 '2018-11-29',
 '2017-03-15',
 '2016-05-04',
 '2015-07-08',
 '2017-01-10',
 '2018-11-18',
 '2018-12-02',
 '2019-01-31',
 '2018-10-18',
 '2017-10-03',
 '2018-11-22',
 '2017-11-13',
 '2017-10-17',
 '2017-09-22',
 '2017-11-10',
 '2017-04-14',
 '2018-03-24',
 '2016-03-28',
 '2017-10-25',
 '2017-07-05',
 '2017-03-06',
 '2018-01-01',
 '2017-08-24',
 '2017-09-17',
 '2018-09-11',
 '2017-11-07',
```

```
'2018-03-03',
'2015-12-16',
'2018-08-28',
'2017-08-04',
'2018-09-16',
'2018-09-05',
'2019-03-27',
'2018-12-09',
'2016-03-14',
'2018-06-14',
'2018-02-27',
'2016-05-14',
'2016-03-13',
'2018-02-22',
'2017-12-18',
'2018-02-23',
'2018-12-27',
'2018-01-15',
'2018-09-24',
'2019-03-13',
'2018-11-26',
'2018-07-17',
'2019-04-23',
'2019-04-09',
'2018-10-25',
'2017-09-16',
'2015-10-28',
'2017-11-17',
'2016-06-15',
'2019-04-26',
'2018-08-06',
'2018-08-24',
'2019-02-12',
'2014-12-10',
'2017-09-26',
'2018-02-22',
'2019-03-23',
'2016-04-15',
'2018-10-04',
'2018-04-18',
'2018-01-07',
'2017-05-25',
'2016-04-13',
'2017-10-30',
'2015-12-08',
'2017-09-29',
'2017-05-29',
'2018-03-27',
'2016-05-19',
'2019-02-07',
'2018-11-05',
'2017-11-13',
'2018-03-04',
'2019-02-24',
'2015-10-21',
'2017-10-20',
'2017-02-17',
'2019-04-06',
'2019-04-04',
'2019-03-10',
'2018-02-25',
'2019-04-23',
'2019-04-29',
'2019-02-13',
'2018-02-01',
'2018-06-23',
```

```
'2018-08-20',
'2017-10-13',
'2018-03-23',
'2017-07-10',
'2018-09-28',
'2018-08-01',
'2018-01-01',
'2017-11-18',
'2016-04-13',
'2016-12-09',
'2019-02-27',
'2017-09-27',
'2016-06-08',
'2017-03-30',
'2015-09-21',
'2018-03-31',
'2017-07-06',
'2018-11-04',
'2018-06-28',
'2017-07-01',
'2019-03-20',
'2018-08-08',
'2017-05-21',
'2018-11-06',
'2017-07-07',
'2018-06-25',
'2017-02-15',
'2017-11-07',
'2017-09-04',
'2017-07-30',
'2016-04-11',
'2018-07-11',
'2017-05-11',
'2018-06-25',
'2019-02-28',
'2017-12-02',
'2019-04-08',
'2018-11-14',
'2017-06-02',
'2018-03-31',
'2019-02-22',
'2018-02-08',
'2017-04-11',
'2018-10-16',
'2019-03-08',
'2019-03-14',
'2017-09-10',
'2019-03-30',
'2019-04-11',
'2018-08-02',
'2017-08-23',
'2017-10-10',
'2017-02-16',
'2018-02-14',
'2018-10-20',
'2018-10-02',
'2017-10-06',
'2019-04-16',
'2017-10-13',
'2016-06-01',
'2018-11-16',
'2017-04-17',
'2017-08-17',
'2019-02-07',
'2016-05-19',
'2017-06-29',
```

```
'2016-04-22',
'2015-02-04',
'2019-03-22',
'2018-06-30',
'2015-11-12',
'2017-04-29',
'2016-11-11',
'2018-01-12',
'2018-01-15',
'2019-04-23',
'2018-09-08',
'2019-04-17',
'2019-04-11',
'2016-06-15',
'2019-03-26',
'2015-01-29',
'2016-10-10',
'2017-07-20',
'2018-09-07',
'2018-02-27',
'2018-04-05',
'2019-04-01',
'2017-12-12',
'2018-07-09',
'2018-05-09',
'2018-05-08',
'2017-12-12',
'2016-03-09',
'2017-04-05',
'2019-04-24',
'2018-01-12',
'2017-11-25',
'2018-09-03',
'2018-04-03',
'2015-02-28',
'2018-03-08',
'2017-12-15',
'2019-04-18',
'2019-03-12',
'2017-09-22',
'2016-11-23',
'2019-04-16',
'2017-02-16',
'2018-11-05',
'2017-09-18',
'2018-12-13',
'2017-06-07',
'2016-02-04',
'2016-02-19',
'2018-08-12',
'2017-01-20',
'2015-08-02',
'2018-01-23',
'2019-01-22',
'2017-05-25',
'2018-01-10',
'2018-03-28',
'2017-06-30',
'2017-03-18',
'2018-02-01',
'2016-03-14',
'2017-08-10',
'2017-11-21',
'2014-12-09',
'2019-04-29',
'2019-01-24',
```

```
'2016-11-19',
'2016-06-03',
'2016-08-16',
'2016-04-19',
'2018-09-07',
'2018-03-15',
'2017-09-11',
'2019-03-18',
'2019-02-21',
'2018-09-21',
'2017-03-07',
'2016-02-29',
'2015-06-24',
'2018-11-15',
'2017-10-12',
'2019-03-11',
'2019-03-18',
'2017-04-26',
'2018-11-24',
'2018-11-15',
'2018-11-03',
'2018-02-06',
'2017-09-02',
'2016-04-06',
'2016-06-09',
'2018-07-10',
'2017-06-05',
'2018-02-22',
'2017-01-24',
'2016-03-11',
'2017-06-14',
'2018-10-14',
'2018-03-14',
'2017-08-09',
'2016-12-12',
'2017-04-25',
'2017-04-02',
'2018-06-21',
'2018-09-12',
'2016-04-05',
'2016-11-08',
'2018-02-01',
'2017-10-05',
'2018-08-23',
'2018-07-09',
'2018-02-01',
'2018-02-01',
'2018-01-05',
'2018-10-19',
'2018-08-07',
'2016-03-09',
'2017-11-29',
'2018-09-29',
'2017-03-13',
'2018-02-27',
'2017-07-10',
'2015-03-19',
'2016-01-30',
'2017-07-11',
'2017-02-07',
'2017-06-29',
'2015-09-06',
'2019-04-26',
'2017-09-22',
'2018-09-10',
'2019-04-04',
```

```
'2017-04-18',
'2016-04-20',
'2018-04-25',
'2018-11-01',
'2017-08-30',
'2016-05-14',
'2017-12-10',
'2019-03-11',
'2018-03-06',
'2018-09-03',
'2016-01-14',
'2017-06-15',
'2017-11-21',
'2019-05-01',
'2017-12-08',
'2018-07-30',
'2017-12-26',
'2016-08-18',
'2018-08-31',
'2018-03-12',
'2017-09-27',
'2019-03-25',
'2019-02-20',
'2016-04-25',
'2015-05-08',
'2017-11-20',
'2018-07-12',
'2018-10-05',
'2017-11-10',
'2019-02-28',
'2018-08-07',
'2018-08-22',
'2019-03-17',
'2019-03-19',
'2019-04-08',
'2018-09-06',
'2017-01-23',
'2017-12-21',
'2017-10-17',
'2018-09-21',
'2018-11-18',
'2018-03-26',
'2018-09-12',
'2018-03-22',
'2019-01-30',
'2015-04-24',
'2017-11-25',
'2018-06-26',
'2018-02-01',
'2015-10-27',
'2017-02-07',
'2018-12-12',
'2016-05-24',
'2018-02-01',
'2016-07-28',
'2018-01-11',
'2018-09-04',
'2018-03-06',
'2018-03-23',
'2017-09-19',
'2018-02-13',
'2018-10-26',
'2016-06-02',
'2018-01-19',
'2017-06-16',
'2017-08-23',
```

```
'2019-03-01',
'2016-09-08',
'2018-03-22',
'2014-12-09',
'2018-07-21',
'2018-08-25',
'2016-06-28',
'2015-01-20',
'2016-07-04',
'2016-04-27',
'2018-08-15',
'2018-02-02',
'2017-10-04',
'2017-03-21',
'2017-08-15',
'2017-10-19',
'2014-12-09',
'2018-02-09',
'2018-06-29',
'2018-03-31',
'2018-02-21',
'2016-07-04',
'2017-10-26',
'2018-02-21',
'2017-02-21',
'2017-09-20',
'2017-12-19',
'2018-03-24',
'2018-04-03',
'2016-05-10',
'2018-02-12',
'2018-02-01',
'2018-01-10',
'2018-09-09',
'2017-04-27',
'2017-05-02',
'2017-12-04',
'2018-09-10',
'2017-04-28',
'2017-08-03',
'2018-01-12',
'2018-02-01',
'2015-02-24',
'2017-06-29',
'2018-08-11',
'2018-03-12',
'2016-06-17',
'2018-02-07',
'2019-05-01',
'2018-08-17',
'2019-03-29',
'2019-04-24',
'2018-07-05',
'2017-06-30',
'2018-09-12',
'2017-02-03',
'2017-05-17',
'2017-08-04',
'2017-08-31',
'2017-11-10',
'2019-04-24',
'2017-06-07',
'2015-08-01',
'2018-10-05',
'2016-03-02',
'2018-02-01',
```

```
'2018-03-12',
'2017-08-08',
'2016-07-02',
'2015-06-11',
'2017-05-22',
'2018-01-11',
'2016-03-18',
'2016-09-23',
'2015-11-23',
'2018-07-01',
'2017-03-09',
'2018-09-28',
'2019-01-29',
'2017-06-25',
'2017-09-26',
'2017-11-10',
'2015-10-23',
'2018-01-12',
'2017-11-09',
'2015-11-18',
'2018-01-03',
'2018-10-08',
'2018-02-21',
'2018-06-04',
'2017-05-05',
'2015-08-20',
'2018-11-04',
'2017-10-26',
'2018-08-06',
'2018-12-14',
'2017-12-21',
'2017-12-05',
'2018-06-21',
'2017-05-12',
'2015-09-20',
'2019-02-27',
'2018-12-15',
'2016-02-16',
'2017-11-28',
'2017-07-19',
'2016-04-17',
'2017-01-26',
'2018-02-27',
'2018-02-25',
'2017-04-24',
'2018-03-24',
'2017-09-11',
'2017-03-28',
'2016-06-26',
'2018-10-29',
'2017-12-21',
'2014-12-09',
'2016-02-04',
'2018-11-12',
'2018-09-06',
'2018-09-21',
'2019-01-08',
'2015-07-30',
'2016-07-12',
'2016-11-02',
'2015-01-21',
'2015-10-31',
'2016-07-18',
'2017-05-18',
'2016-10-25',
'2017-05-30',
```

```
'2018-08-07',
'2018-01-09',
'2017-11-26',
'2018-07-17',
'2016-11-24',
'2014-12-16',
'2019-01-11',
'2018-08-30',
'2017-11-17',
'2019-03-27',
'2018-05-07',
'2018-02-01',
'2018-06-25',
'2017-02-21',
'2016-05-27',
'2016-04-28',
'2018-11-06',
'2018-02-21',
'2018-12-28',
'2018-11-16',
'2017-05-14',
'2018-07-24',
'2018-10-16',
'2018-04-21',
'2018-03-08',
'2017-03-30',
'2017-05-22',
'2017-09-12',
'2017-05-13',
'2018-06-06',
'2019-03-18',
'2017-10-27',
'2015-02-11',
'2015-12-21',
'2018-08-07',
'2016-04-19',
'2019-03-12',
'2017-06-18',
'2018-07-04',
'2018-11-01',
'2017-08-14',
'2016-05-18',
'2016-06-08',
'2016-07-01',
'2018-01-17',
'2017-04-05',
'2015-07-15',
'2016-02-02',
'2017-06-20',
'2017-09-11',
'2017-07-23',
'2018-07-19',
'2017-11-08',
'2019-02-07',
'2019-03-18',
'2017-07-10',
'2017-11-10',
'2018-02-07',
'2019-04-03',
'2018-02-01',
'2018-01-17',
'2016-05-03',
'2019-01-28',
'2018-09-03',
'2017-11-08',
'2015-07-30',
```

```
'2019-04-16',
'2018-03-06',
'2019-03-21',
'2018-03-27',
'2018-11-12',
'2017-11-07',
'2018-03-28',
'2018-12-08',
'2017-04-18',
'2017-06-06',
'2017-11-14',
'2018-12-12',
'2017-01-23',
'2018-04-03',
'2018-03-16',
'2019-02-20',
'2018-10-02',
'2018-08-04',
'2018-08-06',
'2017-06-17',
'2016-08-17',
'2017-08-01',
'2017-06-30',
'2019-01-28',
'2014-11-27',
'2018-02-25',
'2017-04-07',
'2019-04-04',
'2018-01-18',
'2018-11-16',
'2019-03-13',
'2017-10-17',
'2018-11-20',
'2017-05-18',
'2017-05-16',
'2017-04-14',
'2018-02-01',
'2018-07-19',
'2016-08-08',
'2017-11-14',
'2018-10-08',
'2018-03-19',
'2018-03-16',
'2018-11-20',
'2019-04-18',
'2019-04-12',
'2017-12-03',
'2018-04-06',
'2016-01-25',
'2018-08-21',
'2018-02-16',
'2015-11-25',
'2017-07-15',
'2017-09-29',
'2019-05-01',
'2018-07-01',
'2018-10-02',
'2016-02-02',
'2016-11-25',
'2018-03-26',
'2014-12-09',
'2017-04-18',
'2019-02-20',
'2018-10-11',
'2015-11-07',
'2015-11-02',
```

```
'2017-12-08',
'2017-10-04',
'2017-11-10',
'2017-11-21',
'2017-01-02',
'2018-11-15',
'2018-07-16',
'2018-08-30',
'2017-02-08',
'2017-09-26',
'2015-11-23',
'2017-08-08',
'2017-04-06',
'2019-02-26',
'2019-03-19',
'2019-02-11',
'2018-11-16',
'2019-04-27',
'2018-02-08',
'2017-04-25',
'2017-09-27',
'2017-10-31',
'2015-08-13',
'2019-02-28',
'2018-02-20',
'2014-12-09',
'2016-01-11',
'2018-11-28',
'2017-04-10',
'2015-12-08',
'2017-05-22',
'2016-04-29',
'2019-02-13',
'2017-10-13',
'2018-09-08',
'2015-03-26',
'2019-04-01',
'2014-11-27',
'2017-12-01',
'2018-08-21',
'2018-11-12',
'2015-06-09',
'2015-02-09',
'2018-07-19',
'2016-07-07',
'2015-12-17',
'2017-11-08',
'2019-03-02',
'2017-08-08',
'2016-03-01',
'2017-12-14',
'2017-09-05',
'2018-08-01',
'2017-12-22',
'2017-01-13',
'2015-09-20',
'2018-02-01',
'2017-06-07',
'2017-06-27',
'2018-07-30',
'2018-02-01',
'2018-06-05',
'2017-11-11',
'2017-08-05',
'2018-05-04',
'2018-03-02',
```

```
'2017-06-19',
'2016-05-25',
'2018-01-09',
'2018-07-10',
'2019-01-24',
'2018-03-14',
'2018-04-03',
'2018-08-03',
'2018-02-03',
'2017-06-06',
'2018-01-16',
'2016-08-05',
'2017-06-08',
'2017-11-23',
'2015-08-31',
'2018-03-25',
'2018-07-12',
'2016-05-22',
'2018-03-21',
'2016-03-09',
'2017-07-22',
'2018-10-25',
'2016-05-26',
'2017-11-28',
'2015-03-19',
'2017-03-04',
'2016-02-10',
'2018-03-13',
'2018-01-01',
'2017-04-24',
'2017-07-25',
'2019-03-21',
'2017-07-04',
'2018-07-04',
'2017-04-10',
'2017-12-13',
'2018-12-18',
'2019-04-16',
'2015-03-10',
'2018-05-05',
'2017-11-13',
'2018-07-24',
'2018-10-04',
'2018-02-01',
'2017-11-28',
'2018-03-26',
'2017-10-26',
'2017-12-06',
'2019-04-21',
'2018-10-15',
'2017-05-02',
'2018-10-01',
'2015-12-09',
'2019-03-28',
'2018-10-13',
'2018-02-20',
'2016-09-16',
'2016-11-14',
'2017-08-14',
'2016-04-04',
'2017-09-25',
'2017-10-18',
'2016-01-25',
'2019-04-24',
'2017-08-01',
'2018-10-17',
```

```
'2018-03-04',
'2018-08-24',
'2018-05-22',
'2016-04-29',
'2019-02-13',
'2015-08-23',
'2017-09-08',
'2015-11-25',
'2018-11-22',
'2017-04-10',
'2018-07-24',
'2018-11-10',
'2017-09-29',
'2019-01-14',
'2017-12-05',
'2017-12-13',
'2018-07-25',
'2017-08-16',
'2017-07-19',
'2018-09-05',
'2018-03-24',
'2017-11-10',
'2016-02-25',
'2017-07-26',
'2018-01-13',
'2015-05-19',
'2018-04-06',
'2016-04-25',
'2018-01-09',
'2018-10-09',
'2018-12-18',
'2018-06-15',
'2018-07-12',
'2018-11-06',
'2019-03-25',
'2015-02-19',
'2017-05-24',
'2017-05-30',
'2017-08-08',
'2017-04-04',
'2019-04-12',
'2018-11-15',
'2018-02-13',
'2018-10-05',
'2019-04-03',
'2016-04-13',
'2016-12-08',
'2018-02-11',
'2017-04-14',
'2017-10-19',
'2015-04-19',
'2018-07-09',
'2017-01-19',
'2017-09-06',
'2016-06-02',
'2017-11-21',
'2015-12-17',
'2018-08-17',
'2018-10-23',
'2015-10-29',
'2016-04-29',
'2019-03-08',
'2018-09-12',
'2018-08-10',
'2017-10-27',
'2018-03-26',
```

```
'2018-03-29',
'2018-02-13',
'2018-03-31',
'2017-04-14',
'2018-05-27',
'2018-10-11',
'2017-11-26',
'2018-11-07',
'2018-02-07',
'2017-12-23',
'2016-01-15',
'2017-07-31',
'2017-09-11',
'2015-11-18',
'2017-11-16',
'2017-06-02',
'2017-10-02',
'2018-07-11',
'2018-11-27',
'2015-09-24',
'2017-05-17',
'2014-12-09',
'2017-05-11',
'2018-09-19',
'2019-02-21',
'2016-05-01',
'2018-01-22',
'2016-05-10',
'2018-07-23',
'2018-09-28',
'2018-07-10',
'2019-03-06',
'2018-10-05',
'2017-12-19',
'2019-04-17',
'2017-12-19',
'2019-01-06',
'2017-12-04',
'2016-02-16',
'2015-10-10',
'2018-08-23',
'2015-12-01',
'2018-02-01',
'2018-06-18',
'2017-12-04',
'2016-03-29',
'2017-11-23',
'2018-09-05',
'2018-09-20',
'2017-07-10',
'2018-02-22',
'2016-03-20',
'2018-02-01',
'2017-06-29',
'2018-11-27',
'2015-10-21',
'2017-07-05',
'2017-10-04',
'2018-05-14',
'2018-01-11',
'2017-07-29',
'2015-11-10',
'2018-11-02',
'2018-11-12',
'2017-11-29',
'2017-10-16',
```

```
'2019-04-15',
'2017-10-04',
'2015-03-31',
'2019-03-20',
'2016-03-26',
'2017-11-11',
'2016-04-07',
'2018-01-16',
'2017-06-20',
'2017-12-06',
'2017-06-28',
'2018-06-15',
'2016-08-04',
'2017-11-08',
'2018-04-06',
'2017-09-11',
'2019-04-16',
'2018-02-28',
'2017-08-09',
'2018-12-05',
'2018-11-19',
'2018-03-23',
'2018-10-24',
'2016-09-27',
'2019-02-28',
'2017-06-15',
'2018-10-26',
'2017-09-01',
'2017-12-04',
'2017-09-10',
'2017-05-15',
'2017-07-07',
'2018-04-20',
'2017-05-10',
'2015-02-19',
'2019-04-15',
'2017-12-15',
'2018-09-13',
'2018-12-03',
'2017-11-07',
'2018-09-12',
'2018-08-16',
'2018-11-01',
'2017-02-16',
'2017-07-26',
'2017-07-09',
'2017-11-09',
'2019-03-30',
'2018-12-08',
'2017-07-31',
'2019-02-27',
'2018-03-26',
'2018-09-19',
'2017-07-21',
'2019-04-04',
'2018-03-23',
'2017-10-31',
'2018-07-25',
'2016-12-05',
'2018-09-26',
'2017-09-07',
'2017-09-12',
'2018-08-17',
'2015-02-19',
'2016-03-30',
'2016-03-21',
```

```
 '2019-02-12',
 '2017-02-15',
 '2017-06-03',
 '2017-03-02',
 '2017-12-18',
 '2018-01-11',
 '2017-11-26',
 '2017-06-03',
 '2018-01-15',
 '2017-04-26',
 '2016-10-18',
 ...]
```

Изменим цену на тип int для удобства просмотра

In [23]:
```
df['last_price'] = df['last_price'].astype('int')
```

Заменим пропуски в days_exposition и заменим тип данных

In [24]:
```
df['days_exposition'] = df['days_exposition'].fillna(0).astype('int')
```

Удалим столбцы, которые не заполнены более чем на 50 %

In [25]:
```
del df['parks_nearest']
del df['ponds_nearest']
del df['is_apartment']
#del df['ceiling_height']
```

Заменим пропуски в balcony на 0 и изменим тип данных на int

In [26]:
```
df['balcony'].value_counts()
df['balcony'] = df['balcony'].fillna(0)
df['balcony'] = df['balcony'].astype('int')
```

Удалим строки

In [27]:
```
df = df.dropna(subset=['locality_name'])
```

In [28]:
```
df= df.dropna(subset=['floors_total'])
```

Заменяем пустые значения жилой площади и площади кухни

In [29]:
```
living_ratio = df['living_area'].mean() / df['total_area'].mean()
kitchen_ratio = df['kitchen_area'].mean() / df['total_area'].mean()
df['living_area'].fillna(living_ratio * df['total_area'], inplace=True)
df['kitchen_area'].fillna(kitchen_ratio * df['total_area'], inplace=True)
```

Приводим в целочисленный тип количество парков. Заменяем пропущенные значения на 0

In [30]:
```
df['ponds_around3000'] = df['ponds_around3000'].fillna(value=0).astype(int)
```

Приводим в целочисленный тип количество парков. Заменяем пропущенные значения на 0

In [31]:
```
df['parks_around3000'] = df['parks_around3000'].fillna(value=0).astype(int)
```

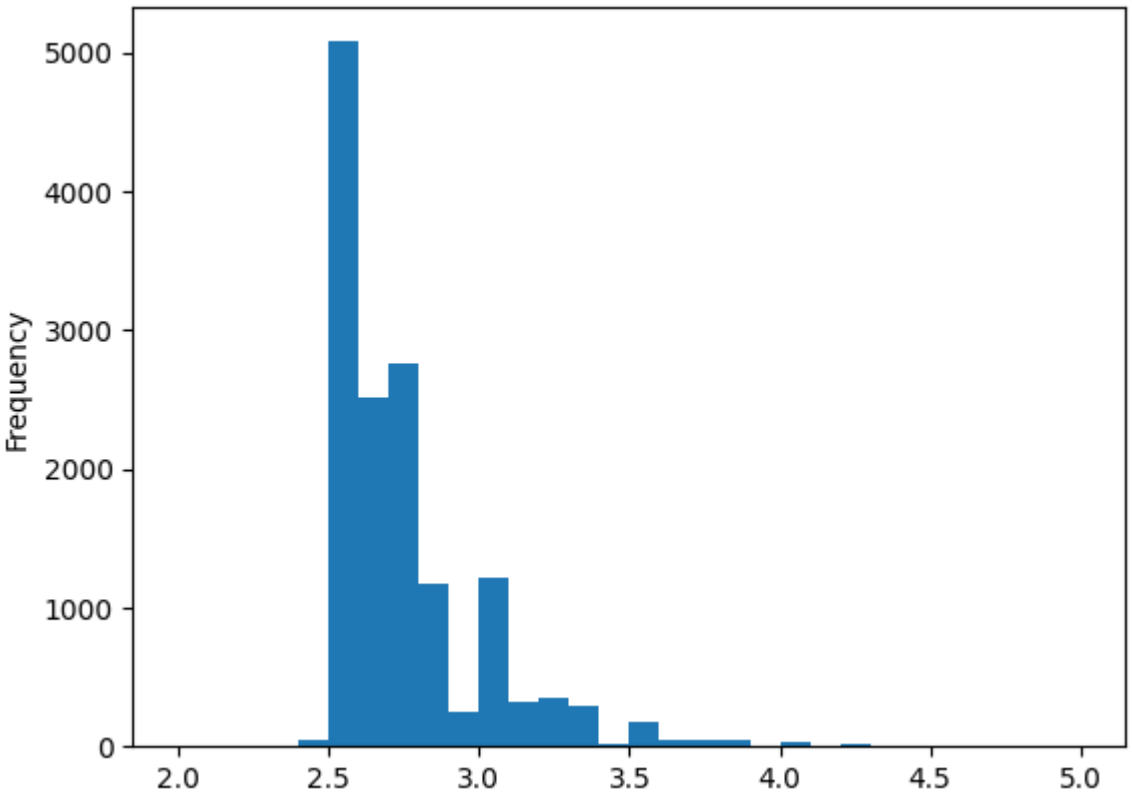
Посмотрим параметр ceiling_height, видим что есть необоснованно большие значения но пока оставим их как есть, медиана от среднего отличается не сильно из-за этих выбросов во всей выборке

In [32]:
```
df['ceiling_height'].sort_values().plot(y = 'ceiling_height', kind = 'hist', bins = 3
```

```
df['ceiling_height'].value_counts()
df['ceiling_height'].describe()
df[df['ceiling_height'] > 4].sort_values('ceiling_height').tail(20)
```

Out[32]:

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total
18545	6	3750000	43.0	2019-03-18	2	25.0	5.0
355	17	3600000	55.2	2018-07-12	2	25.0	5.0
6246	6	3300000	44.4	2019-03-25	2	25.0	5.0
14382	9	1700000	35.0	2015-12-04	1	25.0	5.0
11285	0	1950000	37.0	2019-03-20	1	25.0	5.0
4643	0	4300000	45.0	2018-02-01	2	25.0	9.0
9379	5	3950000	42.0	2017-03-26	3	25.0	5.0
5669	4	4400000	50.0	2017-08-08	2	26.0	9.0
5807	17	8150000	80.0	2019-01-09	2	27.0	36.0
5246	0	2500000	54.0	2017-10-13	2	27.0	5.0
21824	20	2450000	44.0	2019-02-12	2	27.0	2.0
22938	14	4000000	98.0	2018-03-15	4	27.0	2.0
4876	7	3000000	25.0	2017-09-27	0	27.0	25.0
10773	8	3800000	58.0	2017-10-13	2	27.0	10.0
20478	11	8000000	45.0	2017-07-18	1	27.0	4.0
17857	1	3900000	56.0	2017-12-22	3	27.0	5.0
21377	19	4900000	42.0	2017-04-18	1	27.5	24.0
22336	19	9999000	92.4	2019-04-05	2	32.0	6.0
3148	14	2900000	75.0	2018-11-12	3	32.0	3.0
22869	0	15000000	25.0	2018-07-25	1	100.0	5.0



Заменяем оставшиеся пустые значения высоты потолков на медианное значение по всему датасету

In [33]:
```
#df['ceiling_height'] = df.groupby(['locality_name',
#                                   'floors_total'])['ceiling_height'].apply(lambda x

#df['ceiling_height'].fillna(0, inplace=True)
print(df['ceiling_height'].median())
df['ceiling_height'].fillna(2.7,inplace=True)

#df['ceiling_height'].fillna(df['ceiling_height'].median(), inplace=True)
df.groupby(['locality_name','floors_total'])['ceiling_height'].describe()
```

```
2.65
```

Out[33]:

		count	mean	std	min	25%	50%	75%	max
locality_name	**floors_total**								
бокситогорск	**3.0**	6.0	3.083333	0.938971	2.70	2.70	2.70	2.70	5.00
	4.0	1.0	2.800000	NaN	2.80	2.80	2.80	2.80	2.80
	5.0	9.0	2.700000	0.000000	2.70	2.70	2.70	2.70	2.70
волосово	**1.0**	1.0	2.700000	NaN	2.70	2.70	2.70	2.70	2.70
	2.0	1.0	2.700000	NaN	2.70	2.70	2.70	2.70	2.70
...	**...**	...	...	...	...	...	...	...	...
шлиссельбург	**4.0**	6.0	2.600000	0.109545	2.50	2.50	2.60	2.70	2.70
	5.0	25.0	2.590000	0.092421	2.50	2.50	2.55	2.70	2.70
	9.0	24.0	2.643750	0.122752	2.50	2.50	2.70	2.75	2.80
	15.0	1.0	2.700000	NaN	2.70	2.70	2.70	2.70	2.70
	16.0	1.0	2.850000	NaN	2.85	2.85	2.85	2.85	2.85

1041 rows × 8 columns

Заполним пустые значения большим числом

In [34]:
```
df['city_centers_nearest'] = df['city_centers_nearest']/1000
df['city_centers_nearest'] = df['city_centers_nearest'].fillna(999999)
df['city_centers_nearest'] = df['city_centers_nearest'].astype('int')
df['airports_nearest'] = df['airports_nearest']/1000
df['airports_nearest'] = df['airports_nearest'].fillna(999999)
df['airports_nearest'] = df['airports_nearest'].astype('int')
```

In [35]:
```
MissingValue = df.isnull().sum().sort_values(ascending = False)
Percent = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = False)
MissingData = pd.concat([MissingValue, Percent], axis=1, keys=['Пропущенные значения'
MissingData
```

Out[35]:

	Пропущенные значения	Процент
total_images	0	0.0
last_price	0	0.0
days_exposition	0	0.0
ponds_around3000	0	0.0
parks_around3000	0	0.0
city_centers_nearest	0	0.0
airports_nearest	0	0.0
locality_name	0	0.0
balcony	0	0.0
kitchen_area	0	0.0
open_plan	0	0.0
studio	0	0.0
floor	0	0.0
living_area	0	0.0
floors_total	0	0.0
ceiling_height	0	0.0
rooms	0	0.0
first_day_exposition	0	0.0
total_area	0	0.0
first_day_exposition_copy	0	0.0

In [36]:
```
df.dtypes
```

Out[36]:
```
total_images                  int64
last_price                    int64
total_area                  float64
first_day_exposition         object
rooms                         int64
ceiling_height              float64
floors_total                float64
living_area                 float64
floor                         int64
studio                         bool
open_plan                      bool
kitchen_area                float64
balcony                       int64
locality_name                object
airports_nearest              int64
city_centers_nearest          int64
parks_around3000              int64
ponds_around3000              int64
days_exposition               int64
first_day_exposition_copy    object
dtype: object
```

Изменение типов во всех остальных случаях необходимо, чтобы сэкономить память.

In [37]:
```
df['floors_total'] =  df['floors_total'].astype('Int8')
```

Убирем анамальные значения

In [38]:
```
# Функция для подсчёта границ с учетом 1.5 размаха
def quartile_range(df, column):
    q1 = df[column].quantile(0.25)
    q3 = df[column].quantile(0.75)
    iqr = q3 - q1
    dfq = df.loc[(df[column] < q3 + 1.5*iqr) & (df[column] > q1 - 1.5*iqr), column]
    return dfq
# Столбцы с выбросами
list_features = ['last_price', 'total_area', 'kitchen_area', 'living_area', 'days_exp
for col in list_features:
    df[col] = quartile_range(df, col)
```

In [39]:
```
MissingValue = df.isnull().sum().sort_values(ascending = False)
Percent = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = False)
MissingData = pd.concat([MissingValue, Percent], axis=1, keys=['Пропущенные значения'
MissingData
```

Out[39]:

	Пропущенные значения	Процент
ceiling_height	2964	12.577976
last_price	2046	8.682368
days_exposition	2026	8.597496
kitchen_area	1315	5.580310
total_area	1239	5.257798
living_area	883	3.747083
rooms	194	0.823255
total_images	0	0.000000
locality_name	0	0.000000
ponds_around3000	0	0.000000
parks_around3000	0	0.000000
city_centers_nearest	0	0.000000
airports_nearest	0	0.000000
open_plan	0	0.000000
balcony	0	0.000000
studio	0	0.000000
floor	0	0.000000
floors_total	0	0.000000
first_day_exposition	0	0.000000
first_day_exposition_copy	0	0.000000

Выводы:

В ходе предобработки данных были:

- заполнены отсутствую значения,
- обработана некачественные названия столбцов
- изменены некорректные типы данных
- убоали нелогичная группировка столбцов между собой
- заменили прыгающий регистр в названии столбцов
- убрали выбросы

Шаг 3 — Посчитайте и добавьте в таблицу новые столбцы

посчитаем цену квадратного метра

In [40]:
```
df['price_per_square_meter'] = df['last_price'] / df['total_area']
df['price_per_square_meter']
```

Out[40]:
```
0                 NaN
1         82920.792079
2         92785.714286
3                 NaN
4        100000.000000
             ...
23694             NaN
23695     52542.372881
23696     44091.710758
23697    149511.400651
23698     41795.665635
Name: price_per_square_meter, Length: 23565, dtype: float64
```

создадим столбец дня недели

In [41]:
```
df['weekday_exposition'] = pd.to_datetime(df['first_day_exposition_copy']).dt.strftim
#df['weekday_exposition'] = df['first_day_exposition_copy'].dt.weekday
#df['weekday_exposition'] = df['first_day_exposition_copy'].map(lambda x: x.strftime(
df['weekday_exposition'].tolist()
```

Out[41]:
```
['Thursday',
 'Tuesday',
 'Thursday',
 'Friday',
 'Tuesday',
 'Monday',
 'Thursday',
 'Thursday',
 'Wednesday',
 'Sunday',
 'Thursday',
 'Monday',
 'Thursday',
 'Saturday',
 'Thursday',
 'Saturday',
 'Thursday',
 'Friday',
 'Friday',
 'Wednesday',
 'Thursday',
 'Wednesday',
 'Monday',
 'Saturday',
 'Saturday',
 'Thursday',
 'Friday',
 'Friday',
 'Wednesday',
 'Saturday',
 'Monday',
 'Thursday',
 'Sunday',
 'Monday',
 'Sunday',
 'Monday',
 'Wednesday',
 'Saturday',
 'Thursday',
 'Thursday',
 'Wednesday',
 'Wednesday',
 'Wednesday',
 'Tuesday',
 'Sunday',
 'Sunday',
 'Thursday',
 'Thursday',
 'Tuesday',
 'Thursday',
 'Monday',
 'Tuesday',
 'Friday',
 'Friday',
 'Friday',
 'Saturday',
 'Monday',
 'Wednesday',
 'Wednesday',
 'Monday',
 'Monday',
 'Thursday',
 'Sunday',
 'Tuesday',
 'Tuesday',
```

```
'Saturday',
'Wednesday',
'Tuesday',
'Friday',
'Sunday',
'Wednesday',
'Wednesday',
'Sunday',
'Monday',
'Thursday',
'Tuesday',
'Saturday',
'Sunday',
'Thursday',
'Monday',
'Friday',
'Thursday',
'Monday',
'Monday',
'Wednesday',
'Monday',
'Tuesday',
'Tuesday',
'Tuesday',
'Thursday',
'Saturday',
'Wednesday',
'Friday',
'Wednesday',
'Friday',
'Monday',
'Friday',
'Tuesday',
'Wednesday',
'Tuesday',
'Thursday',
'Saturday',
'Friday',
'Thursday',
'Wednesday',
'Sunday',
'Thursday',
'Wednesday',
'Monday',
'Tuesday',
'Friday',
'Monday',
'Tuesday',
'Thursday',
'Thursday',
'Monday',
'Monday',
'Sunday',
'Sunday',
'Wednesday',
'Friday',
'Friday',
'Saturday',
'Thursday',
'Sunday',
'Sunday',
'Tuesday',
'Monday',
'Wednesday',
'Thursday',
'Saturday',
```

```
'Monday',
'Friday',
'Friday',
'Monday',
'Friday',
'Wednesday',
'Monday',
'Saturday',
'Wednesday',
'Friday',
'Wednesday',
'Wednesday',
'Wednesday',
'Thursday',
'Monday',
'Saturday',
'Thursday',
'Sunday',
'Thursday',
'Saturday',
'Wednesday',
'Wednesday',
'Sunday',
'Tuesday',
'Friday',
'Monday',
'Wednesday',
'Tuesday',
'Monday',
'Sunday',
'Monday',
'Wednesday',
'Thursday',
'Monday',
'Thursday',
'Saturday',
'Monday',
'Wednesday',
'Friday',
'Saturday',
'Friday',
'Thursday',
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Sunday',
'Saturday',
'Thursday',
'Thursday',
'Wednesday',
'Tuesday',
'Thursday',
'Wednesday',
'Saturday',
'Friday',
'Tuesday',
'Friday',
'Wednesday',
'Friday',
'Monday',
'Thursday',
'Thursday',
'Thursday',
'Thursday',
'Friday',
```

```
'Wednesday',
'Friday',
'Saturday',
'Thursday',
'Saturday',
'Friday',
'Friday',
'Monday',
'Tuesday',
'Saturday',
'Wednesday',
'Thursday',
'Wednesday',
'Tuesday',
'Thursday',
'Monday',
'Thursday',
'Friday',
'Tuesday',
'Thursday',
'Monday',
'Tuesday',
'Monday',
'Wednesday',
'Tuesday',
'Tuesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Friday',
'Saturday',
'Monday',
'Tuesday',
'Saturday',
'Thursday',
'Friday',
'Thursday',
'Tuesday',
'Friday',
'Tuesday',
'Thursday',
'Monday',
'Monday',
'Thursday',
'Wednesday',
'Thursday',
'Friday',
'Sunday',
'Friday',
'Sunday',
'Tuesday',
'Tuesday',
'Thursday',
'Wednesday',
'Wednesday',
'Friday',
'Saturday',
'Thursday',
'Monday',
'Thursday',
'Tuesday',
'Tuesday',
'Monday',
'Thursday',
'Saturday',
'Friday',
```

```
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Monday',
'Monday',
'Thursday',
'Friday',
'Tuesday',
'Monday',
'Wednesday',
'Thursday',
'Thursday',
'Monday',
'Monday',
'Wednesday',
'Saturday',
'Thursday',
'Saturday',
'Tuesday',
'Saturday',
'Wednesday',
'Thursday',
'Tuesday',
'Monday',
'Thursday',
'Tuesday',
'Friday',
'Wednesday',
'Sunday',
'Wednesday',
'Wednesday',
'Monday',
'Tuesday',
'Sunday',
'Thursday',
'Wednesday',
'Tuesday',
'Tuesday',
'Thursday',
'Thursday',
'Thursday',
'Monday',
'Thursday',
'Thursday',
'Friday',
'Friday',
'Tuesday',
'Wednesday',
'Wednesday',
'Saturday',
'Monday',
'Tuesday',
'Monday',
'Thursday',
'Saturday',
'Tuesday',
'Tuesday',
'Thursday',
'Sunday',
'Friday',
'Friday',
'Monday',
'Thursday',
'Tuesday',
'Wednesday',
```

'Wednesday',
'Thursday',
'Wednesday',
'Saturday',
'Sunday',
'Monday',
'Tuesday',
'Monday',
'Thursday',
'Thursday',
'Tuesday',
'Wednesday',
'Friday',
'Monday',
'Tuesday',
'Thursday',
'Friday',
'Monday',
'Wednesday',
'Monday',
'Wednesday',
'Monday',
'Friday',
'Monday',
'Thursday',
'Friday',
'Friday',
'Thursday',
'Tuesday',
'Wednesday',
'Sunday',
'Tuesday',
'Monday',
'Thursday',
'Monday',
'Thursday',
'Tuesday',
'Friday',
'Sunday',
'Monday',
'Wednesday',
'Thursday',
'Wednesday',
'Friday',
'Saturday',
'Tuesday',
'Thursday',
'Tuesday',
'Tuesday',
'Wednesday',
'Tuesday',
'Thursday',
'Thursday',
'Thursday',
'Tuesday',
'Tuesday',
'Friday',
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Friday',
'Friday',
'Wednesday',
'Friday',
'Thursday',

```
'Thursday',
'Tuesday',
'Saturday',
'Saturday',
'Tuesday',
'Tuesday',
'Monday',
'Wednesday',
'Wednesday',
'Friday',
'Wednesday',
'Tuesday',
'Tuesday',
'Thursday',
'Tuesday',
'Friday',
'Friday',
'Saturday',
'Wednesday',
'Monday',
'Thursday',
'Wednesday',
'Tuesday',
'Wednesday',
'Tuesday',
'Saturday',
'Tuesday',
'Tuesday',
'Monday',
'Thursday',
'Wednesday',
'Sunday',
'Thursday',
'Tuesday',
'Monday',
'Monday',
'Friday',
'Thursday',
'Friday',
'Thursday',
'Tuesday',
'Thursday',
'Saturday',
'Monday',
'Friday',
'Wednesday',
'Wednesday',
'Friday',
'Friday',
'Wednesday',
'Thursday',
'Friday',
'Wednesday',
'Friday',
'Wednesday',
'Friday',
'Thursday',
'Friday',
'Wednesday',
'Wednesday',
'Friday',
'Wednesday',
'Thursday',
'Monday',
'Tuesday',
'Saturday',
```

'Thursday',
'Monday',
'Thursday',
'Friday',
'Friday',
'Monday',
'Sunday',
'Thursday',
'Friday',
'Tuesday',
'Sunday',
'Tuesday',
'Friday',
'Friday',
'Friday',
'Thursday',
'Wednesday',
'Wednesday',
'Monday',
'Wednesday',
'Monday',
'Friday',
'Thursday',
'Sunday',
'Thursday',
'Monday',
'Friday',
'Thursday',
'Tuesday',
'Thursday',
'Friday',
'Sunday',
'Wednesday',
'Saturday',
'Tuesday',
'Tuesday',
'Wednesday',
'Sunday',
'Thursday',
'Tuesday',
'Sunday',
'Monday',
'Saturday',
'Monday',
'Tuesday',
'Sunday',
'Monday',
'Thursday',
'Tuesday',
'Thursday',
'Monday',
'Thursday',
'Friday',
'Tuesday',
'Thursday',
'Tuesday',
'Wednesday',
'Wednesday',
'Saturday',
'Monday',
'Thursday',
'Tuesday',
'Tuesday',
'Tuesday',
'Tuesday',
'Sunday',

```
'Tuesday',
'Thursday',
'Tuesday',
'Friday',
'Thursday',
'Friday',
'Wednesday',
'Monday',
'Thursday',
'Monday',
'Tuesday',
'Friday',
'Thursday',
'Tuesday',
'Wednesday',
'Friday',
'Friday',
'Sunday',
'Tuesday',
'Tuesday',
'Saturday',
'Thursday',
'Thursday',
'Monday',
'Tuesday',
'Saturday',
'Wednesday',
'Monday',
'Friday',
'Wednesday',
'Monday',
'Tuesday',
'Tuesday',
'Tuesday',
'Sunday',
'Wednesday',
'Thursday',
'Monday',
'Wednesday',
'Wednesday',
'Friday',
'Wednesday',
'Wednesday',
'Wednesday',
'Tuesday',
'Tuesday',
'Monday',
'Sunday',
'Thursday',
'Wednesday',
'Thursday',
'Monday',
'Monday',
'Friday',
'Wednesday',
'Wednesday',
'Thursday',
'Wednesday',
'Tuesday',
'Monday',
'Monday',
'Wednesday',
'Thursday',
'Tuesday',
'Tuesday',
'Thursday',
```

```
'Tuesday',
'Monday',
'Tuesday',
'Wednesday',
'Saturday',
'Tuesday',
'Tuesday',
'Tuesday',
'Wednesday',
'Monday',
'Tuesday',
'Friday',
'Wednesday',
'Tuesday',
'Saturday',
'Monday',
'Saturday',
'Wednesday',
'Tuesday',
'Friday',
'Monday',
'Thursday',
'Sunday',
'Friday',
'Thursday',
'Thursday',
'Friday',
'Wednesday',
'Tuesday',
'Tuesday',
'Thursday',
'Tuesday',
'Friday',
'Thursday',
'Thursday',
'Monday',
'Tuesday',
'Monday',
'Monday',
'Friday',
'Tuesday',
'Thursday',
'Friday',
'Sunday',
'Friday',
'Monday',
'Tuesday',
'Friday',
'Wednesday',
'Saturday',
'Friday',
'Wednesday',
'Sunday',
'Tuesday',
'Tuesday',
'Friday',
'Monday',
'Tuesday',
'Tuesday',
'Wednesday',
'Thursday',
'Saturday',
'Monday',
'Friday',
'Wednesday',
'Friday',
```

```
'Tuesday',
'Monday',
'Thursday',
'Monday',
'Thursday',
'Wednesday',
'Tuesday',
'Monday',
'Tuesday',
'Tuesday',
'Tuesday',
'Monday',
'Friday',
'Saturday',
'Thursday',
'Tuesday',
'Wednesday',
'Tuesday',
'Thursday',
'Thursday',
'Tuesday',
'Tuesday',
'Monday',
'Wednesday',
'Monday',
'Tuesday',
'Monday',
'Friday',
'Wednesday',
'Friday',
'Saturday',
'Thursday',
'Monday',
'Thursday',
'Friday',
'Tuesday',
'Monday',
'Tuesday',
'Monday',
'Thursday',
'Thursday',
'Thursday',
'Wednesday',
'Saturday',
'Tuesday',
'Tuesday',
'Thursday',
'Tuesday',
'Wednesday',
'Friday',
'Friday',
'Sunday',
'Thursday',
'Wednesday',
'Tuesday',
'Monday',
'Thursday',
'Tuesday',
'Saturday',
'Saturday',
'Friday',
'Friday',
'Monday',
'Wednesday',
'Tuesday',
'Tuesday',
```

```
'Thursday',
'Wednesday',
'Tuesday',
'Friday',
'Saturday',
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Thursday',
'Monday',
'Sunday',
'Thursday',
'Sunday',
'Wednesday',
'Wednesday',
'Saturday',
'Thursday',
'Thursday',
'Tuesday',
'Thursday',
'Saturday',
'Wednesday',
'Tuesday',
'Monday',
'Monday',
'Tuesday',
'Thursday',
'Tuesday',
'Wednesday',
'Monday',
'Wednesday',
'Tuesday',
'Tuesday',
'Tuesday',
'Saturday',
'Monday',
'Tuesday',
'Thursday',
'Thursday',
'Tuesday',
'Monday',
'Thursday',
'Wednesday',
'Sunday',
'Monday',
'Tuesday',
'Monday',
'Wednesday',
'Thursday',
'Saturday',
'Tuesday',
'Friday',
'Monday',
'Monday',
'Monday',
'Monday',
'Wednesday',
'Monday',
'Wednesday',
'Tuesday',
'Wednesday',
'Sunday',
'Friday',
'Tuesday',
'Friday',
```

```
'Wednesday',
'Sunday',
'Friday',
'Wednesday',
'Thursday',
'Monday',
'Tuesday',
'Saturday',
'Friday',
'Monday',
'Tuesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Saturday',
'Friday',
'Thursday',
'Wednesday',
'Saturday',
'Tuesday',
'Friday',
'Monday',
'Tuesday',
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Tuesday',
'Monday',
'Thursday',
'Wednesday',
'Tuesday',
'Tuesday',
'Tuesday',
'Friday',
'Thursday',
'Tuesday',
'Friday',
'Wednesday',
'Wednesday',
'Thursday',
'Sunday',
'Friday',
'Thursday',
'Sunday',
'Monday',
'Thursday',
'Wednesday',
'Thursday',
'Tuesday',
'Thursday',
'Friday',
'Tuesday',
'Thursday',
'Friday',
'Friday',
'Wednesday',
'Friday',
'Friday',
'Monday',
'Thursday',
'Tuesday',
'Saturday',
'Friday',
```

```
'Sunday',
'Thursday',
'Sunday',
'Wednesday',
'Wednesday',
'Saturday',
'Friday',
'Monday',
'Monday',
'Wednesday',
'Thursday',
'Friday',
'Monday',
'Wednesday',
'Tuesday',
'Thursday',
'Wednesday',
'Tuesday',
'Thursday',
'Wednesday',
'Thursday',
'Sunday',
'Monday',
'Tuesday',
'Monday',
'Friday',
'Tuesday',
'Wednesday',
'Friday',
'Tuesday',
'Wednesday',
'Tuesday',
'Sunday',
'Monday',
'Tuesday',
'Saturday',
'Thursday',
'Tuesday',
'Thursday',
'Monday',
'Monday',
'Tuesday',
'Thursday',
'Wednesday',
'Thursday',
'Monday',
'Thursday',
'Sunday',
'Thursday',
'Thursday',
'Tuesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Monday',
'Thursday',
'Saturday',
'Tuesday',
'Friday',
'Monday',
'Wednesday',
'Monday',
'Monday',
'Wednesday',
'Tuesday',
'Wednesday',
```

```
'Saturday',
'Saturday',
'Thursday',
'Tuesday',
'Tuesday',
'Wednesday',
'Wednesday',
'Friday',
'Thursday',
'Wednesday',
'Friday',
'Monday',
'Tuesday',
'Wednesday',
'Wednesday',
'Wednesday',
'Monday',
'Friday',
'Wednesday',
'Tuesday',
'Thursday',
'Thursday',
'Friday',
'Friday',
'Monday',
'Sunday',
'Monday',
'Friday',
'Friday',
'Wednesday',
'Thursday',
'Monday',
'Friday',
'Thursday',
'Monday',
'Tuesday',
'Wednesday',
'Thursday',
'Thursday',
'Thursday',
'Wednesday',
'Sunday',
'Thursday',
'Saturday',
'Saturday',
'Monday',
'Wednesday',
'Monday',
'Wednesday',
'Friday',
'Thursday',
'Friday',
'Tuesday',
'Wednesday',
'Monday',
'Wednesday',
'Thursday',
'Tuesday',
'Friday',
'Thursday',
'Wednesday',
'Monday',
'Tuesday',
'Wednesday',
'Saturday',
'Thursday',
```

```
 'Monday',
 'Thursday',
 'Sunday',
 'Saturday',
 'Monday',
 'Wednesday',
 'Tuesday',
 'Friday',
 'Saturday',
 'Thursday',
 'Monday',
 ...]
```

Создадим столбец месяца

In [42]:
```
#df['month_exposition'] = df['first_day_exposition_copy'].dt.month
df['month_exposition'] = pd.to_datetime(df['first_day_exposition_copy']).dt.strftime(
df['month_exposition'].tolist()
```

Out[42]:
```
['03',
 '12',
 '08',
 '07',
 '06',
 '09',
 '11',
 '04',
 '05',
 '02',
 '11',
 '08',
 '06',
 '07',
 '06',
 '11',
 '11',
 '09',
 '01',
 '01',
 '09',
 '03',
 '04',
 '10',
 '10',
 '10',
 '04',
 '05',
 '12',
 '04',
 '02',
 '10',
 '05',
 '10',
 '02',
 '06',
 '01',
 '10',
 '03',
 '11',
 '03',
 '05',
 '07',
 '01',
 '11',
 '12',
 '01',
 '10',
 '10',
 '11',
 '11',
 '10',
 '09',
 '11',
 '04',
 '03',
 '03',
 '10',
 '07',
 '03',
 '01',
 '08',
 '09',
 '09',
 '11',
```

'03',
'12',
'08',
'08',
'09',
'09',
'03',
'12',
'03',
'06',
'02',
'05',
'03',
'02',
'12',
'02',
'12',
'01',
'09',
'03',
'11',
'07',
'04',
'04',
'10',
'09',
'10',
'11',
'06',
'04',
'08',
'08',
'02',
'12',
'09',
'02',
'03',
'04',
'10',
'04',
'01',
'05',
'04',
'10',
'12',
'09',
'05',
'03',
'05',
'02',
'11',
'11',
'03',
'02',
'10',
'10',
'02',
'04',
'04',
'03',
'02',
'04',
'04',
'02',
'02',
'06',

```
'08',
'10',
'03',
'07',
'09',
'08',
'01',
'11',
'04',
'12',
'02',
'09',
'06',
'03',
'09',
'03',
'07',
'11',
'06',
'07',
'03',
'08',
'05',
'11',
'07',
'06',
'02',
'11',
'09',
'07',
'04',
'07',
'05',
'06',
'02',
'12',
'04',
'11',
'06',
'03',
'02',
'02',
'04',
'10',
'03',
'03',
'09',
'03',
'04',
'08',
'08',
'10',
'02',
'02',
'10',
'10',
'04',
'10',
'06',
'11',
'04',
'08',
'02',
'05',
'06',
'04',
```

```
'02',
'03',
'06',
'11',
'04',
'11',
'01',
'01',
'04',
'09',
'04',
'04',
'06',
'03',
'01',
'10',
'07',
'09',
'02',
'04',
'04',
'12',
'07',
'05',
'05',
'12',
'03',
'04',
'04',
'01',
'11',
'09',
'04',
'02',
'03',
'12',
'04',
'03',
'09',
'04',
'02',
'11',
'09',
'12',
'06',
'02',
'02',
'08',
'01',
'08',
'01',
'01',
'05',
'01',
'03',
'06',
'03',
'02',
'03',
'08',
'11',
'12',
'04',
'01',
'11',
'06',
```

'08',
'04',
'09',
'03',
'09',
'03',
'02',
'09',
'03',
'02',
'06',
'11',
'10',
'03',
'03',
'04',
'11',
'11',
'11',
'02',
'09',
'04',
'06',
'07',
'06',
'02',
'01',
'03',
'06',
'10',
'03',
'08',
'12',
'04',
'04',
'06',
'09',
'04',
'11',
'02',
'10',
'08',
'07',
'02',
'02',
'01',
'10',
'08',
'03',
'11',
'09',
'03',
'02',
'07',
'03',
'01',
'07',
'02',
'06',
'09',
'04',
'09',
'09',
'04',
'04',
'04',

'04',
'11',
'08',
'05',
'12',
'03',
'03',
'09',
'01',
'06',
'11',
'05',
'12',
'07',
'12',
'08',
'08',
'03',
'09',
'03',
'02',
'04',
'05',
'11',
'07',
'10',
'11',
'02',
'08',
'08',
'03',
'03',
'04',
'09',
'01',
'12',
'10',
'09',
'11',
'03',
'09',
'03',
'01',
'04',
'11',
'06',
'02',
'10',
'02',
'12',
'05',
'02',
'07',
'01',
'09',
'03',
'03',
'09',
'02',
'10',
'06',
'01',
'06',
'08',
'03',
'09',

'03',
'12',
'07',
'08',
'06',
'01',
'07',
'04',
'08',
'02',
'10',
'03',
'08',
'10',
'12',
'02',
'06',
'03',
'02',
'07',
'10',
'02',
'02',
'09',
'12',
'03',
'04',
'05',
'02',
'02',
'01',
'09',
'04',
'05',
'12',
'09',
'04',
'08',
'01',
'02',
'02',
'06',
'08',
'03',
'06',
'02',
'05',
'08',
'03',
'04',
'07',
'06',
'09',
'02',
'05',
'08',
'08',
'11',
'04',
'06',
'10',
'03',
'02',
'03',
'08',
'07',

```
'06',
'05',
'01',
'03',
'09',
'11',
'07',
'03',
'09',
'01',
'06',
'09',
'11',
'10',
'01',
'11',
'11',
'01',
'10',
'02',
'06',
'05',
'08',
'11',
'10',
'08',
'12',
'12',
'12',
'06',
'05',
'09',
'02',
'12',
'02',
'11',
'07',
'04',
'01',
'02',
'02',
'04',
'03',
'09',
'03',
'06',
'10',
'12',
'12',
'02',
'11',
'09',
'09',
'01',
'07',
'07',
'11',
'01',
'10',
'07',
'05',
'10',
'05',
'08',
'01',
'11',
```

'07',
'11',
'12',
'01',
'08',
'11',
'03',
'05',
'02',
'06',
'02',
'05',
'04',
'11',
'02',
'12',
'11',
'05',
'07',
'10',
'04',
'03',
'03',
'05',
'09',
'05',
'06',
'03',
'10',
'02',
'12',
'08',
'04',
'03',
'06',
'07',
'11',
'08',
'05',
'06',
'07',
'01',
'04',
'07',
'02',
'06',
'09',
'07',
'07',
'11',
'02',
'03',
'07',
'11',
'02',
'04',
'02',
'01',
'05',
'01',
'09',
'11',
'07',
'04',
'03',
'03',

'03',
'11',
'11',
'03',
'12',
'04',
'06',
'11',
'12',
'01',
'04',
'03',
'02',
'10',
'08',
'08',
'06',
'08',
'08',
'06',
'01',
'11',
'02',
'04',
'04',
'01',
'11',
'03',
'10',
'11',
'05',
'05',
'04',
'02',
'07',
'08',
'11',
'10',
'03',
'03',
'11',
'04',
'04',
'12',
'04',
'01',
'08',
'02',
'11',
'07',
'09',
'05',
'07',
'10',
'02',
'11',
'03',
'12',
'04',
'02',
'10',
'11',
'11',
'12',
'10',
'11',

'11',
'01',
'11',
'07',
'08',
'02',
'09',
'11',
'08',
'02',
'03',
'02',
'11',
'04',
'02',
'04',
'09',
'10',
'08',
'02',
'02',
'12',
'01',
'11',
'04',
'12',
'05',
'04',
'02',
'10',
'09',
'03',
'04',
'11',
'12',
'08',
'11',
'06',
'02',
'07',
'07',
'12',
'11',
'03',
'08',
'03',
'12',
'09',
'08',
'12',
'01',
'09',
'02',
'06',
'06',
'07',
'02',
'06',
'11',
'08',
'05',
'03',
'06',
'05',
'01',
'07',

'01',
'03',
'04',
'08',
'02',
'06',
'01',
'08',
'06',
'11',
'08',
'03',
'07',
'05',
'03',
'03',
'07',
'10',
'05',
'11',
'03',
'03',
'02',
'03',
'01',
'04',
'07',
'03',
'07',
'07',
'04',
'12',
'12',
'04',
'03',
'05',
'11',
'07',
'10',
'02',
'11',
'03',
'10',
'12',
'04',
'10',
'05',
'10',
'12',
'03',
'10',
'02',
'09',
'11',
'08',
'04',
'09',
'10',
'01',
'04',
'08',
'10',
'03',
'08',
'05',
'04',

'02',
'08',
'09',
'11',
'11',
'04',
'07',
'11',
'09',
'01',
'12',
'12',
'07',
'08',
'07',
'09',
'03',
'11',
'02',
'07',
'01',
'05',
'04',
'04',
'01',
'10',
'12',
'06',
'07',
'11',
'03',
'02',
'05',
'05',
'08',
'04',
'04',
'11',
'02',
'10',
'04',
'04',
'12',
'02',
'04',
'10',
'04',
'07',
'01',
'09',
'06',
'11',
'12',
'08',
'10',
'10',
'04',
'03',
'09',
'08',
'10',
'03',
'03',
'02',
'03',
'04',

'05',
'10',
'11',
'11',
'02',
'12',
'01',
'07',
'09',
'11',
'11',
'06',
'10',
'07',
'11',
'09',
'05',
'12',
'05',
'09',
'02',
'05',
'01',
'05',
'07',
'09',
'07',
'03',
'10',
'12',
'04',
'12',
'01',
'12',
'02',
'10',
'08',
'12',
'02',
'06',
'12',
'03',
'11',
'09',
'09',
'07',
'02',
'03',
'02',
'06',
'11',
'10',
'07',
'10',
'05',
'01',
'07',
'11',
'11',
'11',
'11',
'10',
'04',
'10',
'03',
'03',

'03',
'11',
'04',
'01',
'06',
'12',
'06',
'06',
'08',
'11',
'04',
'09',
'04',
'02',
'08',
'12',
'11',
'03',
'10',
'09',
'02',
'06',
'10',
'09',
'12',
'09',
'05',
'07',
'04',
'05',
'02',
'04',
'12',
'09',
'12',
'11',
'09',
'08',
'11',
'02',
'07',
'07',
'11',
'03',
'12',
'07',
'02',
'03',
'09',
'07',
'04',
'03',
'10',
'07',
'12',
'09',
'09',
'09',
'08',
'02',
'03',
'03',
'02',
'02',
'06',
'03',

```
 '12',
 '01',
 '11',
 '06',
 '01',
 '04',
 '10',
 '06',
 '06',
 '09',
 '08',
 ...]
```

Создадим столбец года

In [43]:
```
df['year_exposition'] = pd.to_datetime(df['first_day_exposition_copy']).dt.strftime('
df['year_exposition'].tolist()
#df['year_exposition'] = df['first_day_exposition'].dt.year
```

Out[43]:
```
['2019',
 '2018',
 '2015',
 '2015',
 '2018',
 '2018',
 '2017',
 '2019',
 '2018',
 '2017',
 '2017',
 '2018',
 '2016',
 '2017',
 '2016',
 '2017',
 '2017',
 '2016',
 '2017',
 '2019',
 '2017',
 '2018',
 '2017',
 '2016',
 '2015',
 '2015',
 '2017',
 '2017',
 '2017',
 '2016',
 '2018',
 '2017',
 '2016',
 '2018',
 '2018',
 '2017',
 '2017',
 '2017',
 '2018',
 '2018',
 '2017',
 '2016',
 '2015',
 '2017',
 '2018',
 '2018',
 '2019',
 '2018',
 '2017',
 '2018',
 '2017',
 '2017',
 '2017',
 '2017',
 '2017',
 '2018',
 '2016',
 '2017',
 '2017',
 '2017',
 '2018',
 '2017',
 '2017',
 '2018',
 '2017',
```

'2018',
'2015',
'2018',
'2017',
'2018',
'2018',
'2019',
'2018',
'2016',
'2018',
'2018',
'2016',
'2016',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2019',
'2018',
'2018',
'2019',
'2019',
'2018',
'2017',
'2015',
'2017',
'2016',
'2019',
'2018',
'2018',
'2019',
'2014',
'2017',
'2018',
'2019',
'2016',
'2018',
'2018',
'2018',
'2017',
'2016',
'2017',
'2015',
'2017',
'2017',
'2018',
'2016',
'2019',
'2018',
'2017',
'2018',
'2019',
'2015',
'2017',
'2017',
'2019',
'2019',
'2019',
'2018',
'2019',
'2019',
'2019',
'2018',
'2018',

'2018',
'2017',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2017',
'2016',
'2016',
'2019',
'2017',
'2016',
'2017',
'2015',
'2018',
'2017',
'2018',
'2018',
'2017',
'2019',
'2018',
'2017',
'2018',
'2017',
'2018',
'2017',
'2017',
'2017',
'2017',
'2016',
'2018',
'2017',
'2018',
'2019',
'2017',
'2019',
'2018',
'2017',
'2018',
'2019',
'2018',
'2017',
'2018',
'2019',
'2019',
'2017',
'2019',
'2019',
'2018',
'2017',
'2017',
'2017',
'2018',
'2018',
'2017',
'2019',
'2017',
'2016',
'2018',
'2017',
'2017',
'2019',
'2016',
'2017',
'2016',

'2015',
'2019',
'2018',
'2015',
'2017',
'2016',
'2018',
'2018',
'2019',
'2018',
'2019',
'2019',
'2016',
'2019',
'2015',
'2016',
'2017',
'2018',
'2018',
'2018',
'2019',
'2017',
'2018',
'2018',
'2018',
'2017',
'2016',
'2017',
'2019',
'2018',
'2017',
'2018',
'2018',
'2015',
'2018',
'2017',
'2019',
'2019',
'2017',
'2019',
'2017',
'2018',
'2017',
'2018',
'2017',
'2016',
'2016',
'2018',
'2017',
'2015',
'2018',
'2019',
'2017',
'2018',
'2018',
'2017',
'2017',
'2018',
'2016',
'2017',
'2017',
'2014',
'2019',
'2019',
'2016',
'2016',

'2016',
'2016',
'2018',
'2018',
'2017',
'2019',
'2019',
'2018',
'2017',
'2016',
'2015',
'2018',
'2017',
'2019',
'2019',
'2017',
'2018',
'2018',
'2018',
'2018',
'2017',
'2016',
'2016',
'2018',
'2017',
'2018',
'2017',
'2016',
'2017',
'2018',
'2018',
'2017',
'2016',
'2017',
'2017',
'2018',
'2018',
'2016',
'2016',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2018',
'2018',
'2018',
'2016',
'2017',
'2018',
'2017',
'2018',
'2017',
'2015',
'2016',
'2017',
'2017',
'2017',
'2015',
'2019',
'2017',
'2018',
'2019',
'2017',
'2016',

'2018',
'2018',
'2017',
'2016',
'2017',
'2019',
'2018',
'2018',
'2016',
'2017',
'2017',
'2019',
'2017',
'2018',
'2017',
'2016',
'2018',
'2018',
'2017',
'2019',
'2019',
'2016',
'2015',
'2017',
'2018',
'2018',
'2017',
'2019',
'2018',
'2018',
'2019',
'2019',
'2019',
'2018',
'2017',
'2017',
'2017',
'2018',
'2018',
'2018',
'2018',
'2018',
'2019',
'2015',
'2017',
'2018',
'2018',
'2015',
'2017',
'2018',
'2016',
'2018',
'2016',
'2018',
'2018',
'2018',
'2018',
'2017',
'2018',
'2018',
'2016',
'2018',
'2017',
'2017',
'2019',
'2016',

'2018',
'2014',
'2018',
'2018',
'2016',
'2015',
'2016',
'2016',
'2018',
'2018',
'2017',
'2017',
'2017',
'2017',
'2014',
'2018',
'2018',
'2018',
'2018',
'2016',
'2017',
'2018',
'2017',
'2017',
'2017',
'2018',
'2018',
'2016',
'2018',
'2018',
'2018',
'2018',
'2017',
'2017',
'2017',
'2018',
'2017',
'2017',
'2018',
'2018',
'2015',
'2017',
'2018',
'2018',
'2016',
'2018',
'2019',
'2018',
'2019',
'2019',
'2018',
'2017',
'2018',
'2017',
'2017',
'2017',
'2017',
'2017',
'2019',
'2017',
'2018',
'2016',
'2018',
'2018',
'2017',
'2016',

'2015',
'2017',
'2018',
'2016',
'2016',
'2015',
'2018',
'2017',
'2018',
'2019',
'2017',
'2017',
'2017',
'2015',
'2018',
'2017',
'2015',
'2018',
'2018',
'2018',
'2018',
'2017',
'2015',
'2018',
'2017',
'2018',
'2018',
'2017',
'2017',
'2018',
'2017',
'2015',
'2019',
'2018',
'2016',
'2017',
'2017',
'2016',
'2017',
'2018',
'2018',
'2017',
'2018',
'2017',
'2017',
'2016',
'2018',
'2017',
'2014',
'2016',
'2018',
'2018',
'2018',
'2019',
'2015',
'2016',
'2016',
'2015',
'2015',
'2016',
'2017',
'2016',
'2017',
'2018',
'2018',
'2017',

'2018',
'2016',
'2014',
'2019',
'2018',
'2017',
'2019',
'2018',
'2018',
'2018',
'2017',
'2016',
'2016',
'2018',
'2018',
'2018',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2017',
'2017',
'2017',
'2017',
'2018',
'2019',
'2017',
'2015',
'2015',
'2018',
'2016',
'2019',
'2017',
'2018',
'2018',
'2017',
'2016',
'2016',
'2016',
'2018',
'2017',
'2015',
'2016',
'2017',
'2017',
'2017',
'2018',
'2017',
'2019',
'2019',
'2017',
'2017',
'2018',
'2019',
'2018',
'2018',
'2016',
'2019',
'2018',
'2017',
'2015',
'2019',
'2018',
'2019',

'2018',
'2018',
'2017',
'2018',
'2018',
'2017',
'2017',
'2017',
'2018',
'2017',
'2018',
'2018',
'2019',
'2018',
'2018',
'2018',
'2017',
'2016',
'2017',
'2017',
'2019',
'2014',
'2018',
'2017',
'2019',
'2018',
'2018',
'2019',
'2017',
'2018',
'2017',
'2017',
'2017',
'2018',
'2018',
'2016',
'2017',
'2018',
'2018',
'2018',
'2018',
'2019',
'2019',
'2017',
'2018',
'2016',
'2018',
'2018',
'2015',
'2017',
'2017',
'2019',
'2018',
'2018',
'2016',
'2016',
'2018',
'2014',
'2017',
'2019',
'2018',
'2015',
'2015',
'2017',
'2017',
'2017',

'2017',
'2017',
'2018',
'2018',
'2018',
'2017',
'2017',
'2015',
'2017',
'2019',
'2019',
'2019',
'2018',
'2019',
'2018',
'2017',
'2017',
'2017',
'2015',
'2019',
'2018',
'2014',
'2016',
'2018',
'2017',
'2015',
'2017',
'2016',
'2019',
'2017',
'2018',
'2015',
'2019',
'2014',
'2017',
'2018',
'2018',
'2015',
'2015',
'2018',
'2016',
'2015',
'2017',
'2019',
'2017',
'2016',
'2017',
'2017',
'2018',
'2017',
'2017',
'2015',
'2018',
'2017',
'2017',
'2018',
'2018',
'2018',
'2017',
'2017',
'2018',
'2018',
'2017',
'2016',
'2018',
'2018',

```
'2019',
'2018',
'2018',
'2018',
'2018',
'2017',
'2018',
'2016',
'2017',
'2017',
'2015',
'2018',
'2018',
'2016',
'2018',
'2016',
'2017',
'2018',
'2016',
'2017',
'2015',
'2017',
'2016',
'2018',
'2018',
'2017',
'2017',
'2019',
'2017',
'2018',
'2017',
'2017',
'2018',
'2019',
'2015',
'2018',
'2017',
'2018',
'2018',
'2018',
'2017',
'2018',
'2017',
'2017',
'2019',
'2018',
'2017',
'2018',
'2015',
'2019',
'2018',
'2018',
'2016',
'2016',
'2017',
'2016',
'2017',
'2017',
'2016',
'2019',
'2017',
'2018',
'2018',
'2018',
'2018',
'2016',
```

'2019',
'2015',
'2017',
'2015',
'2018',
'2017',
'2018',
'2018',
'2017',
'2019',
'2017',
'2017',
'2018',
'2017',
'2017',
'2018',
'2018',
'2017',
'2016',
'2017',
'2018',
'2015',
'2018',
'2016',
'2018',
'2018',
'2018',
'2018',
'2018',
'2018',
'2019',
'2015',
'2017',
'2017',
'2017',
'2017',
'2019',
'2018',
'2018',
'2018',
'2019',
'2016',
'2016',
'2018',
'2017',
'2017',
'2015',
'2018',
'2017',
'2017',
'2016',
'2017',
'2015',
'2018',
'2018',
'2015',
'2016',
'2019',
'2018',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2017',

'2018',
'2018',
'2017',
'2018',
'2018',
'2017',
'2016',
'2017',
'2017',
'2015',
'2017',
'2017',
'2017',
'2018',
'2018',
'2015',
'2017',
'2014',
'2017',
'2018',
'2019',
'2016',
'2018',
'2016',
'2018',
'2018',
'2018',
'2019',
'2018',
'2017',
'2019',
'2017',
'2019',
'2017',
'2016',
'2015',
'2018',
'2015',
'2018',
'2018',
'2017',
'2016',
'2017',
'2018',
'2018',
'2017',
'2018',
'2016',
'2018',
'2017',
'2018',
'2015',
'2017',
'2017',
'2018',
'2018',
'2017',
'2015',
'2018',
'2018',
'2017',
'2017',
'2019',
'2017',
'2015',
'2019',

'2016',
'2017',
'2016',
'2018',
'2017',
'2017',
'2017',
'2018',
'2016',
'2017',
'2018',
'2017',
'2019',
'2018',
'2017',
'2018',
'2018',
'2018',
'2018',
'2016',
'2019',
'2017',
'2018',
'2017',
'2017',
'2017',
'2017',
'2017',
'2018',
'2017',
'2015',
'2019',
'2017',
'2018',
'2018',
'2017',
'2018',
'2018',
'2018',
'2017',
'2017',
'2017',
'2017',
'2019',
'2018',
'2017',
'2019',
'2018',
'2018',
'2017',
'2019',
'2018',
'2017',
'2018',
'2016',
'2018',
'2017',
'2017',
'2018',
'2015',
'2016',
'2016',
'2019',
'2017',
'2017',
'2017',

```
 '2017',
 '2018',
 '2017',
 '2017',
 '2018',
 '2017',
 '2016',
 '2016',
 '2018',
 '2017',
 '2017',
 ...]
```

Напишем функцию категоризации по этажам

In [44]:

```
def floor_category(row):
    floors_total = row['floors_total']
    floor = row['floor']
    if floor == 1:
        return 'первый'
    elif floor == floors_total:
        return 'последний'
    elif 1 < floor < floors_total:
        return 'другой'

#категоризуем этажи с помощью функции
df['floor_category'] = df.apply(floor_category, axis = 1)
df['floor_category'].tolist()
```

Out[44]: ['другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'первый',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'первый',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'последний',

```
        'другой',
        'другой',
        'другой',
        'первый',
        'последний',
        'первый',
        'другой',
        'другой',
        'первый',
        'другой',
        'другой',
        'первый',
        'первый',
        'другой',
        'последний',
        'первый',
        'другой',
        'другой',
        'другой',
        'первый',
        'последний',
        'другой',
        'другой',
        'первый',
        'другой',
        'другой',
        'первый',
        'другой',
        'другой',
        'другой',
        'последний',
        'другой',
        'последний',
        'другой',
        'другой',
        'другой',
        'другой',
        'другой',
        'другой',
        'первый',
        'другой',
        'другой',
        'последний',
        'другой',
        'последний',
        'другой',
        'первый',
        'другой',
        'последний',
        'другой',
        'другой',
        'последний',
        'другой',
        'другой',
        'другой',
        'первый',
        'другой',
        'другой',
        'другой',
        'другой',
        'другой',
        'первый',
        'другой',
        'первый',
        'другой',
        'последний',
```

'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'последний',
'другой',
'другой',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'последний',
'последний',
'другой',
'последний',
'другой',
'первый',
'другой',
'первый',
'первый',
'последний',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'первый',
'другой',
'последний',
'другой',

```
'последний',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'первый',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'первый',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
```

```
'другой',
'другой',
'другой',
'последний',
'первый',
'другой',
'другой',
'последний',
'первый',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'последний',
'первый',
'другой',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'первый',
'последний',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'первый',
```

```
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'первый',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'первый',
'последний',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'последний',
'первый',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'другой',
```

```
'другой',
'другой',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'последний',
'другой',
```

'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'первый',
'последний',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'первый',
'другой',
'последний',
'последний',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'первый',
'другой',

```
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'первый',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
```

```
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'первый',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'последний',
'первый',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'другой',
'первый',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
```

```
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'первый',
'другой',
'другой',
'первый',
'первый',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'первый',
'первый',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'последний',
'последний',
'последний',
'первый',
'другой',
'другой',
```

```
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'первый',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'последний',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
```

```
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'последний',
            'другой',
            'последний',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'первый',
            'первый',
            'первый',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'последний',
            'другой',
            'последний',
            'последний',
            'последний',
            'последний',
            'другой',
            'другой',
            'другой',
            'другой',
            'первый',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'первый',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
            'другой',
```

```
'другой',
'другой',
'первый',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'последний',
'первый',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'первый',
'первый',
'другой',
'другой',
'последний',
'первый',
'другой',
'первый',
'другой',
'первый',
'другой',
'последний',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'первый',
'другой',
'другой',
'последний',
'последний',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'первый',
'первый',
```

```
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'последний',
'другой',
'другой',
'первый',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'первый',
'другой',
'другой',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
'другой',
'другой',
'другой',
'последний',
'другой',
```

```
 'первый',
 'другой',
 'другой',
 'первый',
 'другой',
 'первый',
 'другой',
 'другой',
 'другой',
 'первый',
 'другой',
 ...]
```

Проведите исследовательский анализ данных

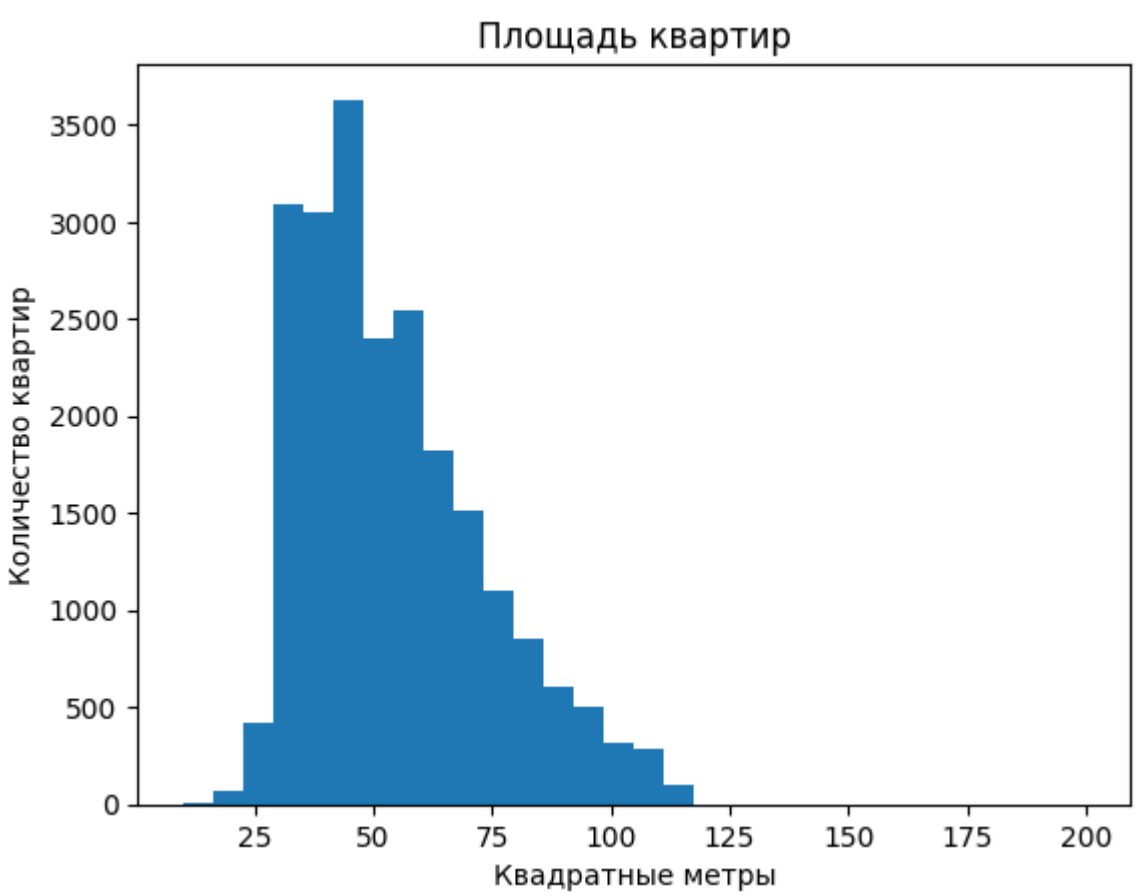
Распределение данных нормальное, есть выбросы но они не сильно влияют на данные, по гистограмме кроме распределения нечего больше указать, с помошью describe получил общее представление

In [45]:
```
df.describe()
```

Out[45]:

	total_images	last_price	total_area	rooms	ceiling_height	floors_total	livin
count	23565.000000	2.151900e+04	22326.000000	23371.000000	20601.000000	23565.0	22682.0
mean	9.878421	4.837115e+06	54.319704	2.030208	2.653136	10.675875	31.5
std	5.681156	2.215774e+06	19.331030	0.973563	0.089934	6.594823	13.7
min	0.000000	1.219000e+04	12.000000	0.000000	2.460000	1.0	2.0
25%	6.000000	3.300000e+06	39.400000	1.000000	2.600000	5.0	18.7
50%	9.000000	4.400000e+06	50.000000	2.000000	2.700000	9.0	30.0
75%	14.000000	6.000000e+06	65.500000	3.000000	2.700000	16.0	40.6
max	50.000000	1.186686e+07	114.200000	5.000000	2.850000	60.0	76.7

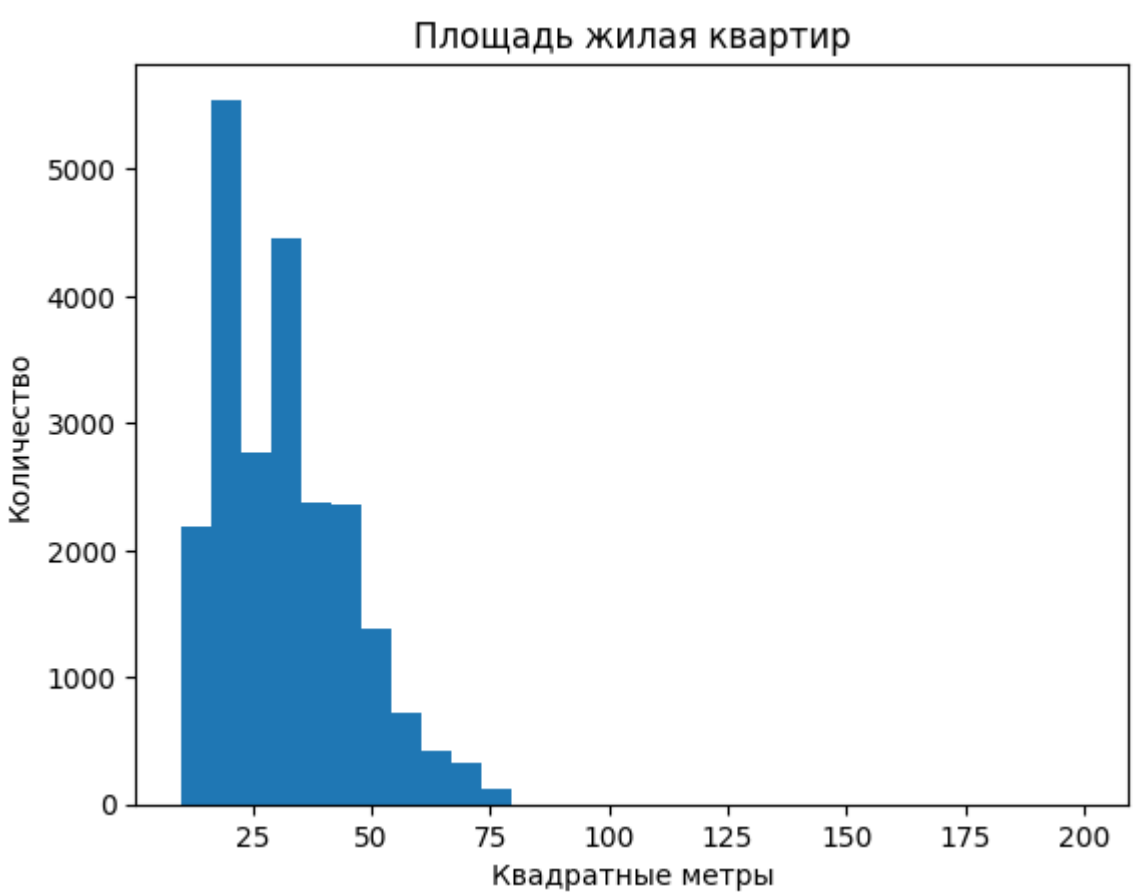
In [46]:
```
plt.hist(df['total_area'], bins=30, range=(10,200))
plt.title('Площадь квартир')
plt.xlabel('Квадратные метры')
plt.ylabel('Количество квартир')
plt.show()
print('Наибольшие по площади варианты:')
print(df['total_area'].sort_values(ascending=False).head(10))
print(df['total_area'].describe())
```




```
Наибольшие по площади варианты:
5978     114.2
19021    114.2
1170     114.2
4657     114.2
20477    114.2
5853     114.1
18122    114.0
6462     114.0
19959    114.0
8249     114.0
Name: total_area, dtype: float64
count    22326.000000
mean        54.319704
std         19.331030
min         12.000000
25%         39.400000
50%         50.000000
75%         65.500000
max        114.200000
Name: total_area, dtype: float64
```

"Общая площадь" в нашей выборке колеблется от 12 до 114 $м^2$, среднее значение 54$м^2$, а медиана 50 $м^2$. Распределение данных Гаусса.

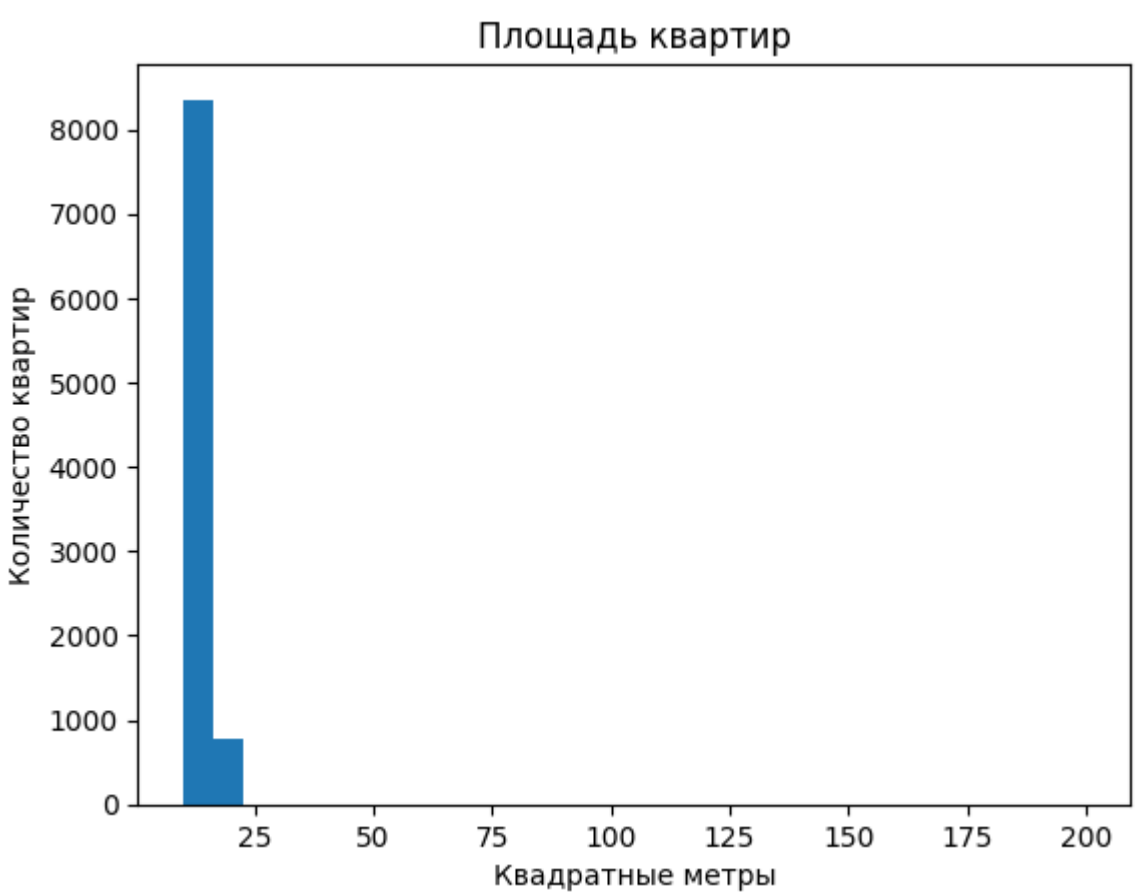
In [47]:
```
plt.hist(df['living_area'], bins=30, range=(10,200))
plt.title('Площадь жилая квартир')
plt.xlabel('Квадратные метры')
plt.ylabel('Количество')
plt.show()
print('Наибольшие по площади варианты:')
print(df['living_area'].sort_values(ascending=False).head(10))
print(df['living_area'].describe())
```




```
Наибольшие по площади варианты:
8470     76.7000
18109    76.6500
8533     76.5373
14540    76.5373
7091     76.5373
16618    76.5000
8933     76.5000
7202     76.4000
10386    76.4000
9361     76.4000
Name: living_area, dtype: float64
count    22682.000000
mean        31.526984
std         13.752739
min          2.000000
25%         18.700000
50%         30.000000
75%         40.600000
max         76.700000
Name: living_area, dtype: float64
```

"Жилая площадь" в нашей выборке колеблется от 2 до 46 $м^2$, среднее значение 32$м^2$, а медиана 30 $м^2$. Распределение данных Гаусса. Два пика графика вызваны, вероятно, тем, что это жилые площади для 1-комнатной, затем для 2-комнатной и после этого для 3-комнатной квартиры. Элитная недвижимость была удалена.

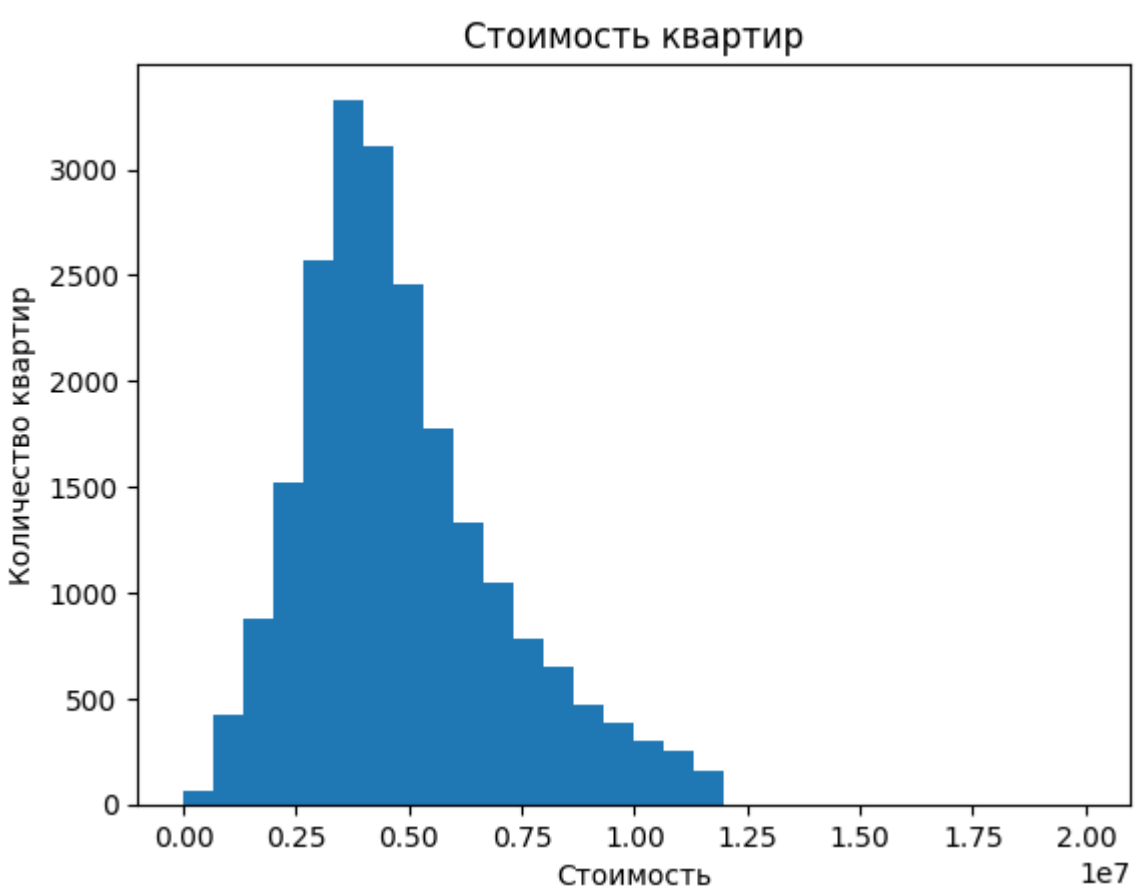
In [48]:
```
plt.hist(df['kitchen_area'], bins=30, range=(10,200))
plt.title('Площадь квартир')
plt.xlabel('Квадратные метры')
plt.ylabel('Количество квартир')
plt.show()
print('Наибольшие по площади варианты:')
print(df['kitchen_area'].sort_values(ascending=False).head(10))
print(df['kitchen_area'].describe())
```




```
Наибольшие по площади варианты:
9448     19.45000
16505    19.43328
8286     19.42000
14212    19.40000
417      19.40000
8901     19.40000
9723     19.40000
6721     19.40000
3056     19.40000
15365    19.40000
Name: kitchen_area, dtype: float64
count    22250.000000
mean         9.417840
std          3.177523
min          1.300000
25%          7.000000
50%          9.000000
75%         11.000000
max         19.450000
Name: kitchen_area, dtype: float64
```

"Площадь кухни" в нашей выборке колеблется от 1 до 20 $м^2$, среднее значение 9$м^2$, а медиана 9 $м^2$. Распределение данных Гаусса. Максимальные значения могут существовать.

In [49]:
```
# Работаем со стоимостью
plt.hist(df['last_price'], bins=30, range=(0,20000000))
plt.title('Стоимость квартир')
plt.xlabel('Стоимость')
plt.ylabel('Количество квартир')
plt.show()
print('Наибольшие по цене варианты:')
print(df['last_price'].sort_values(ascending=False).head(10))

print(df['last_price'].describe())
```

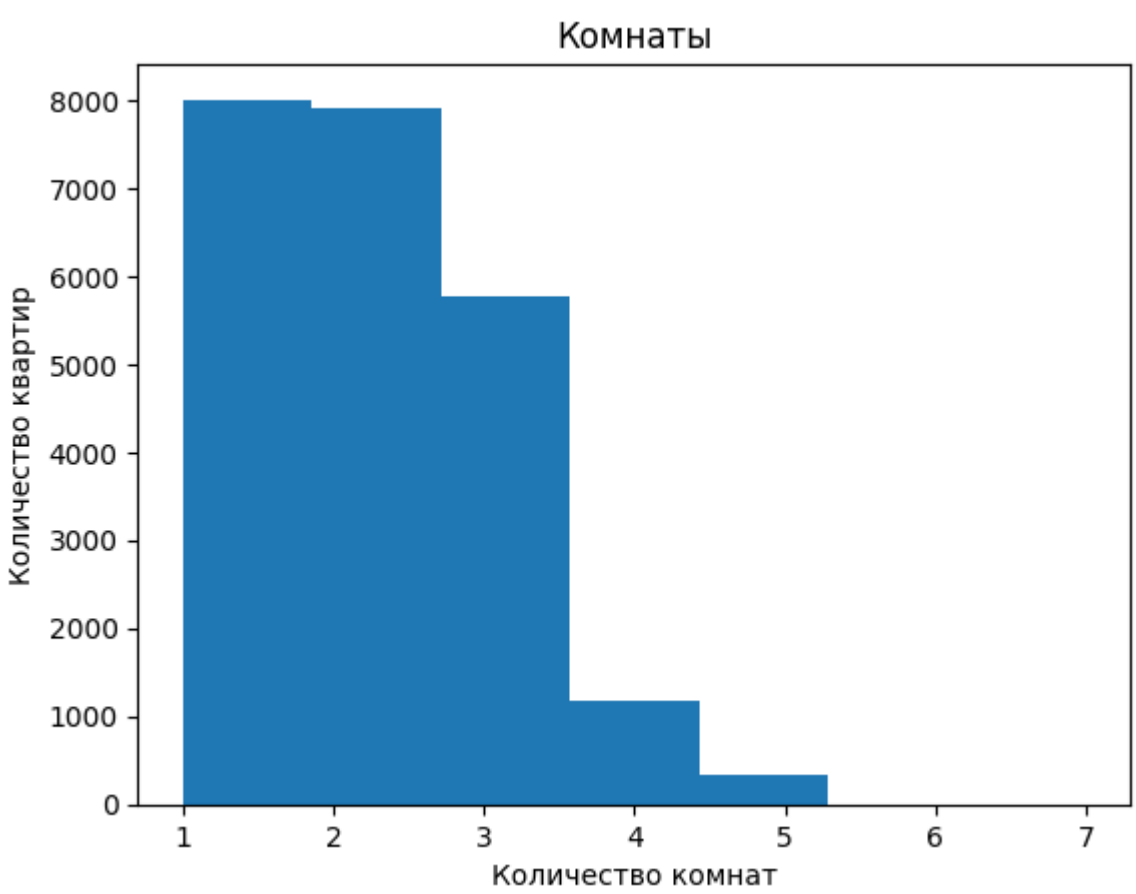



```
Наибольшие по цене варианты:
3165     11866860.0
21860    11858000.0
12437    11850000.0
10068    11840000.0
19454    11820000.0
20193    11809670.0
22326    11800000.0
3642     11800000.0
2507     11800000.0
2267     11800000.0
Name: last_price, dtype: float64
count    2.151900e+04
mean     4.837115e+06
std      2.215774e+06
min      1.219000e+04
25%      3.300000e+06
50%      4.400000e+06
75%      6.000000e+06
max      1.186686e+07
Name: last_price, dtype: float64
```

Цены на недвижимость имеют среднее значение в 6,5 млн руб., медиана 4,6 млн руб. Распределение данных Гаусса.

с данными все впорядке но значения выводятся в экспоненциальной записи чисел

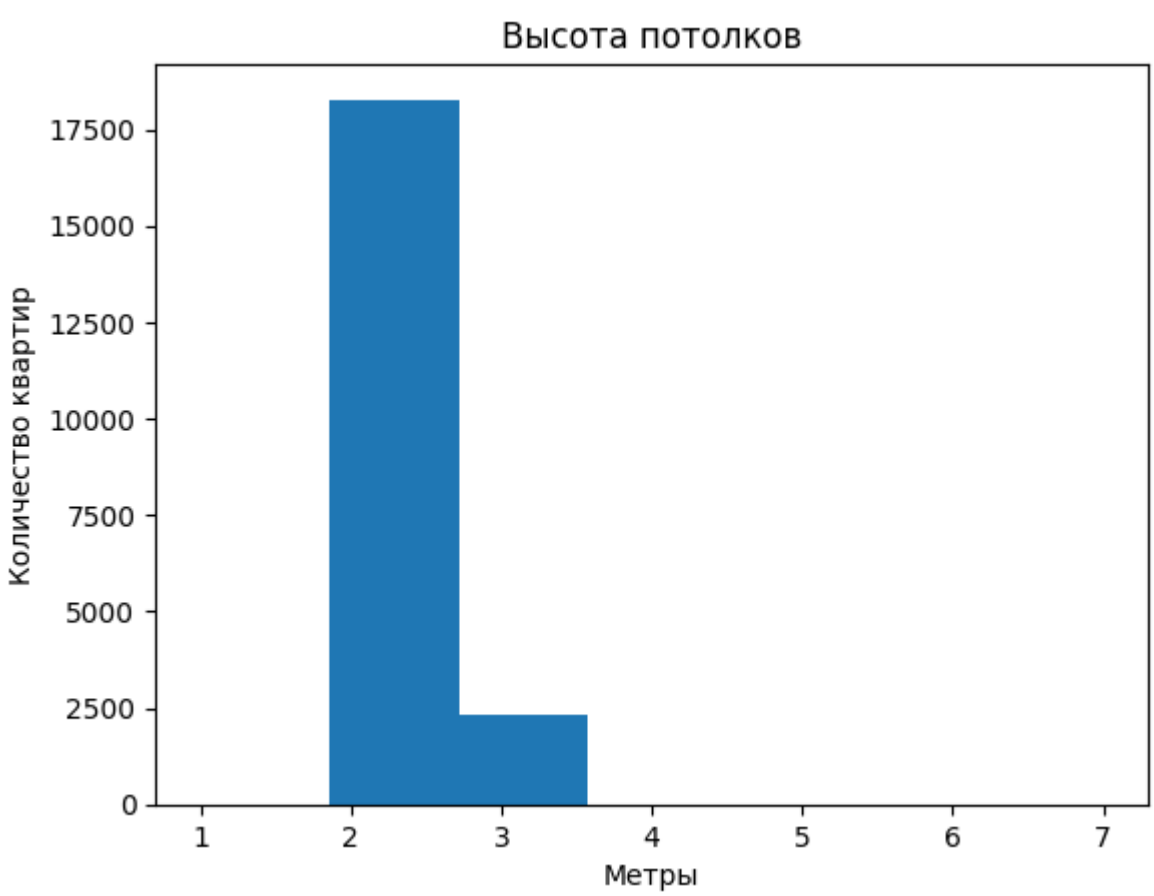
In [50]:
```
# Работаем с комнатами
plt.hist(df['rooms'], bins=7, range=(1,7))
plt.title('Комнаты')
plt.xlabel('Количество комнат')
plt.ylabel('Количество квартир')
plt.show()
print('Наибольшие по числу комнат варианты:')
print(df['rooms'].sort_values(ascending=False).head(10))
print(df['rooms'].describe())
```




```
Наибольшие по числу комнат варианты:
11851    5.0
15856    5.0
22494    5.0
22502    5.0
15999    5.0
5648     5.0
4035     5.0
18633    5.0
18651    5.0
9419     5.0
Name: rooms, dtype: float64
count    23371.000000
mean         2.030208
std          0.973563
min          0.000000
25%          1.000000
50%          2.000000
75%          3.000000
max          5.000000
Name: rooms, dtype: float64
```

Количество комнат колеблется от 1 до 5. Больше всего однокомнатных квартир. В основном квартиры состоят из 1, 2 и 3 комнат.

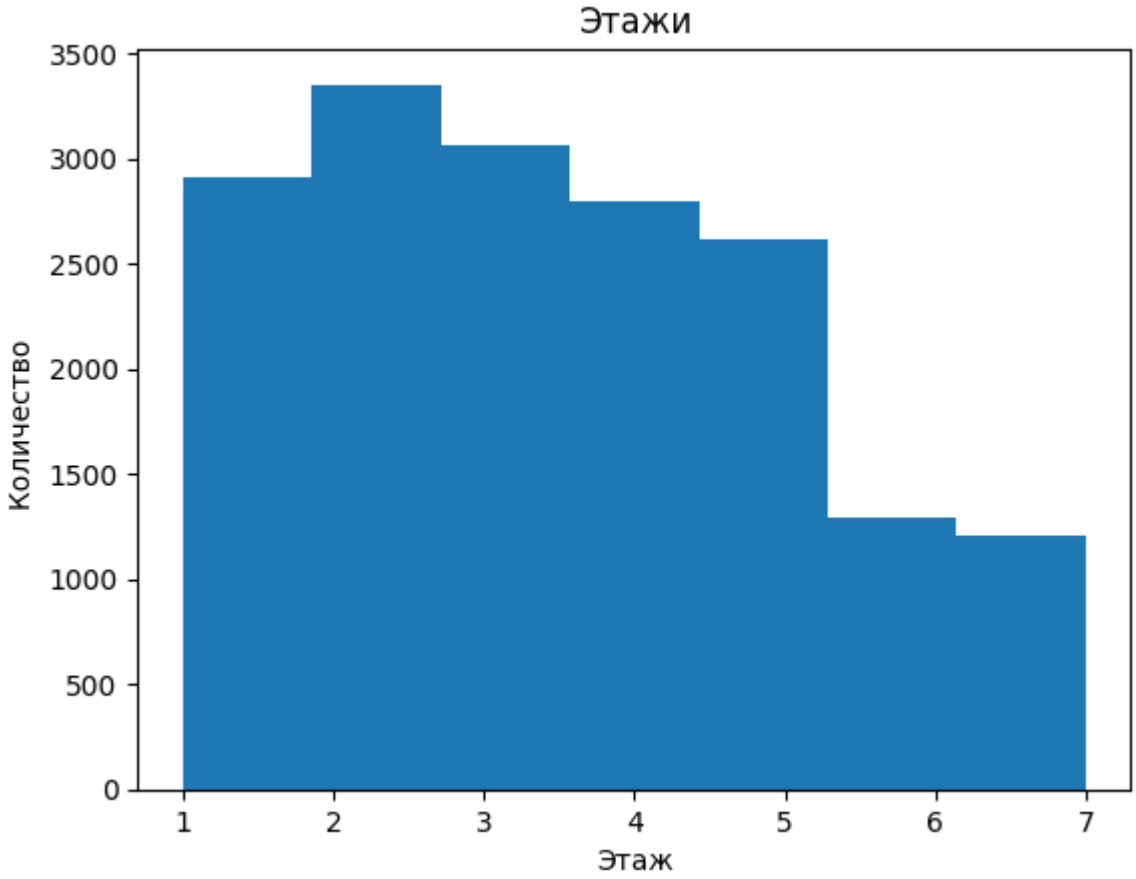
In [51]:
```
# Работаем с комнатами
plt.hist(df['ceiling_height'], bins=7, range=(1,7))
plt.title('Высота потолков')
plt.xlabel('Метры')
plt.ylabel('Количество квартир')
plt.show()
print('Наибольшие по высоте потолков варианты:')
print(df['ceiling_height'].sort_values(ascending=False).head(10))
print(df['ceiling_height'].describe())
```




```
Наибольшие по высоте потолков варианты:
9500     2.85
17580    2.85
17027    2.85
16120    2.85
1912     2.85
5435     2.85
22212    2.85
10462    2.85
13938    2.85
5405     2.85
Name: ceiling_height, dtype: float64
count    20601.000000
mean         2.653136
std          0.089934
min          2.460000
25%          2.600000
50%          2.700000
75%          2.700000
max          2.850000
Name: ceiling_height, dtype: float64
```

Большинство вариантов имеют высоту потолков от 2,5 до 2.75 метров. Больше 2.75 метров высота потолков выглядит подозрительно. Высота потолков в среднем составляет 2,65 м, медиана 2,65 м.

In [52]:
```
# Работаем с комнатами
plt.hist(df['floor'], bins=7, range=(1,7))
plt.title('Этажи')
plt.xlabel('Этаж')
plt.ylabel('Количество ')
plt.show()
print('Этажность:')
print(df['floor'].sort_values(ascending=False).head(10))
print(df['floor'].describe())
```




```
Этажность:
18218    33
11575    32
18629    31
1917     30
11079    29
397      28
12888    27
16644    27
23292    27
4091     27
Name: floor, dtype: int64
count    23565.000000
mean         5.878124
std          4.871485
min          1.000000
25%          2.000000
50%          4.000000
75%          8.000000
max         33.000000
Name: floor, dtype: float64
```

Ничего подозрительного нет. Хотя можно обратить внимание на 33 этаж.

In [53]:
```
df['floor'].sort_values().unique()
```

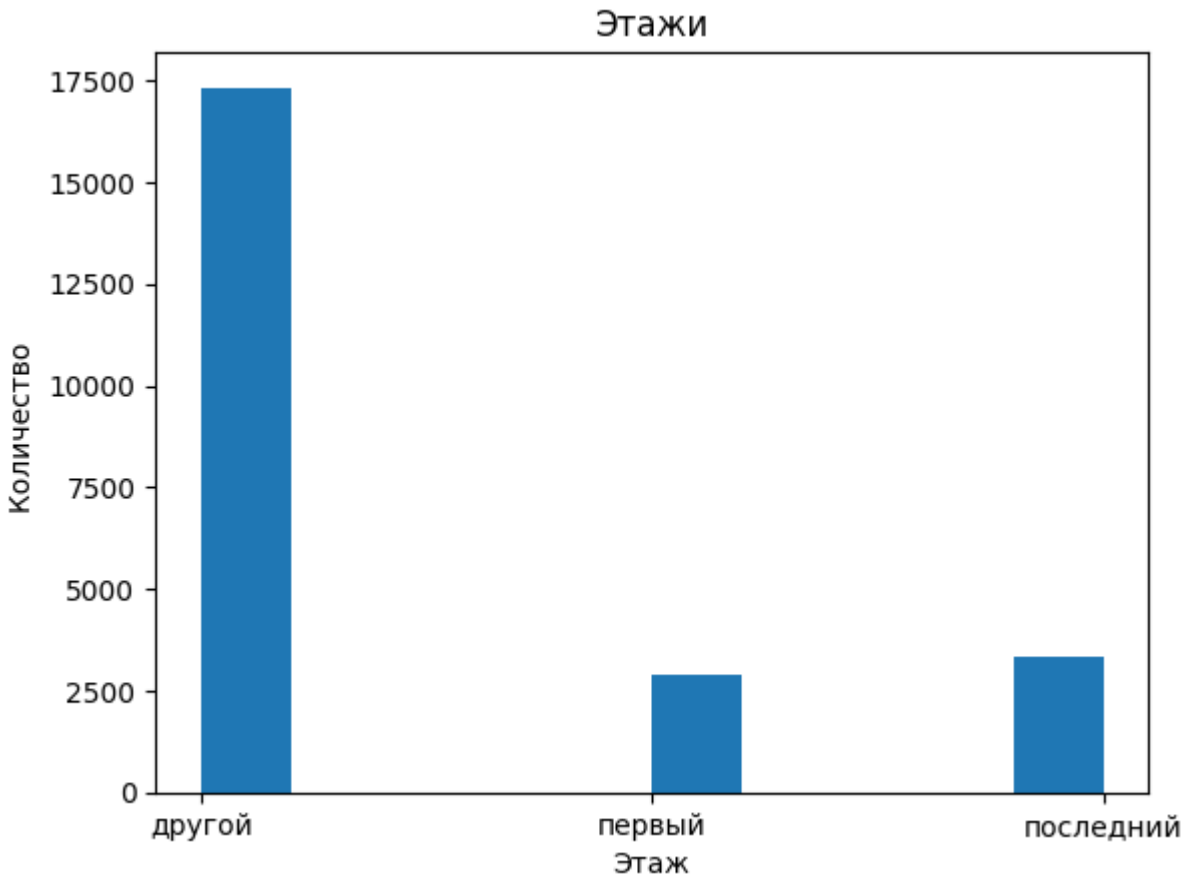
Out[53]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33])
```

In [54]:
```
df['floor_category'].sort_values().unique()
```

Out[54]:
```
array(['другой', 'первый', 'последний'], dtype=object)
```

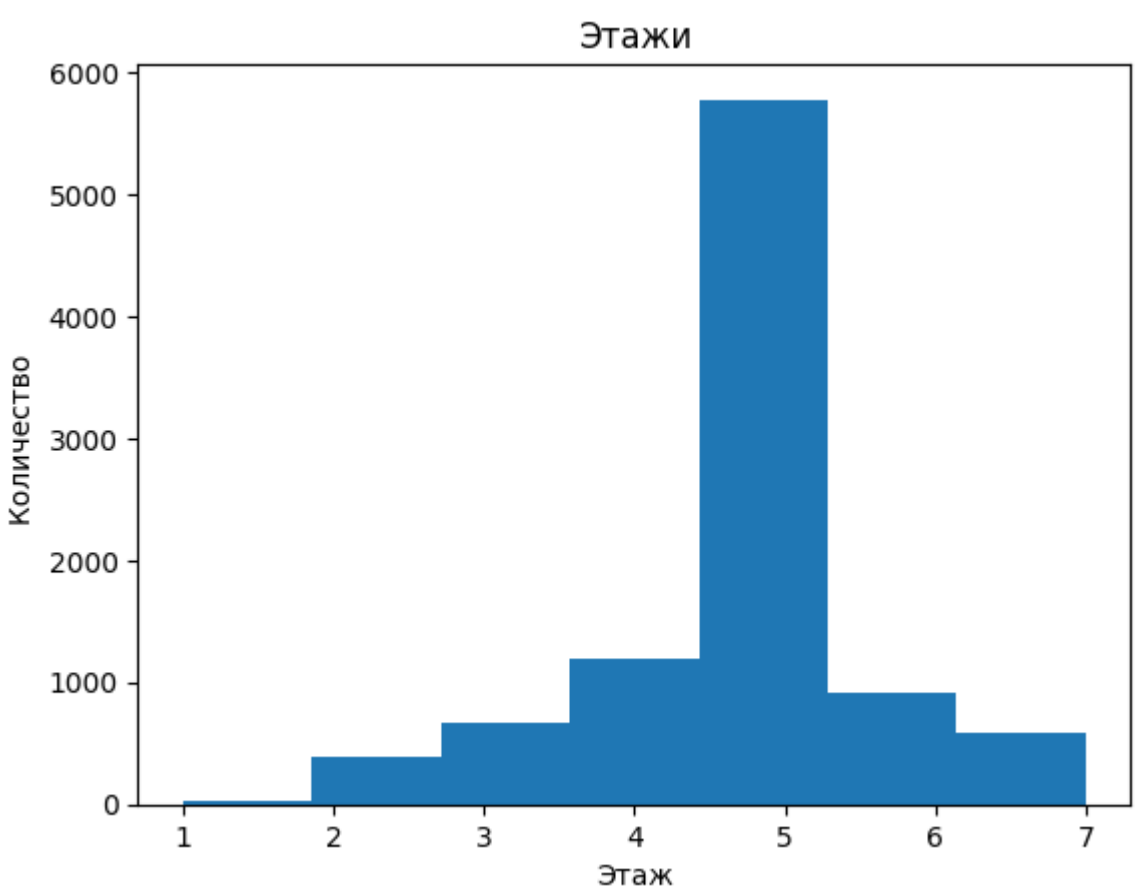
In [55]:
```
# Работаем с комнатами
plt.hist(df['floor_category'])
plt.title('Этажи')
plt.xlabel('Этаж')
plt.ylabel('Количество ')
```

```
plt.show()
print('Этажность:')
print(df['floor_category'].sort_values(ascending=False).head(10))
print(df['floor_category'].describe())
```



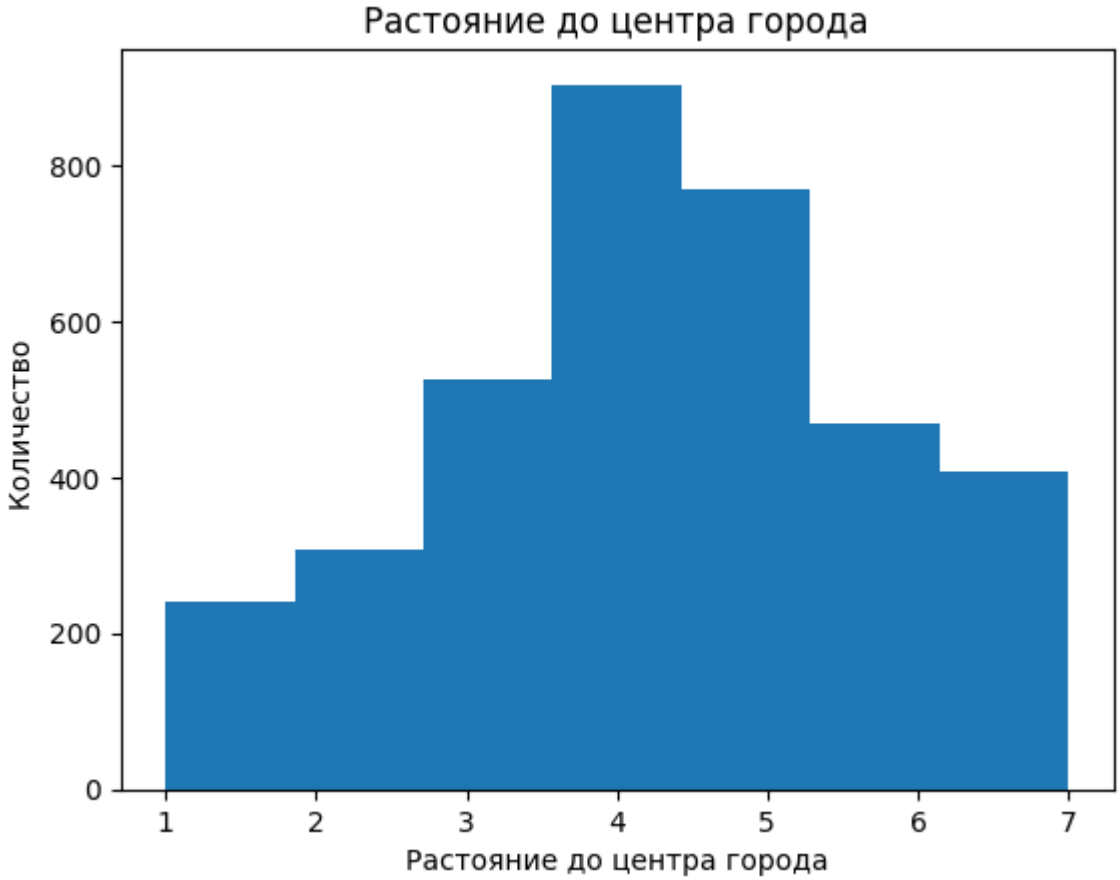

```
Этажность:
6813     последний
17303    последний
3054     последний
8675     последний
17284    последний
3060     последний
3063     последний
8669     последний
3067     последний
17290    последний
Name: floor_category, dtype: object
count      23565
unique         3
top       другой
freq       17326
Name: floor_category, dtype: object
```

In [56]:
```
plt.hist(df['floors_total'], bins=7, range=(1,7))
plt.title('Этажи')
plt.xlabel('Этаж')
plt.ylabel('Количество ')
plt.show()
print('Этажность:')
print(df['floors_total'].sort_values(ascending=False).head(10))
print(df['floors_total'].describe())
```



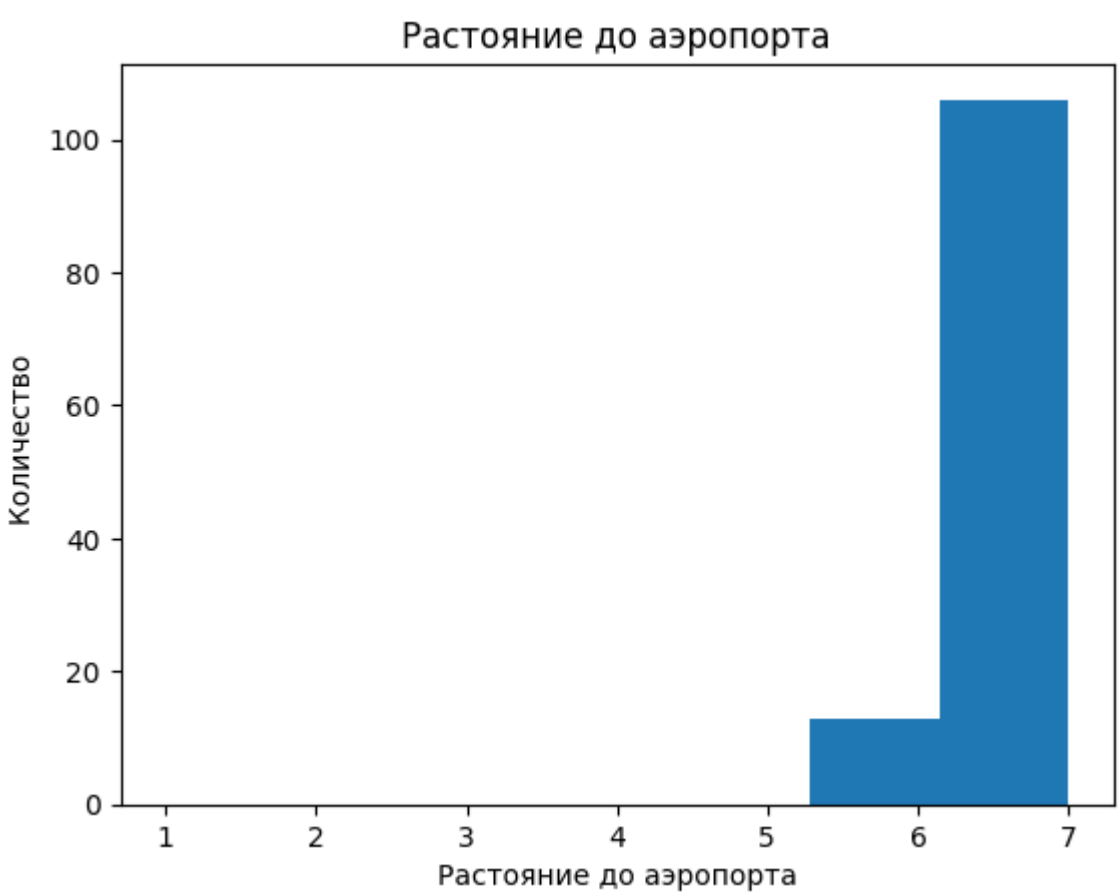

```
Этажность:
2253     60
16731    52
16934    37
5807     36
11079    36
397      36
13975    35
2966     35
9186     35
1917     35
Name: floors_total, dtype: Int8
count      23565.0
mean     10.675875
std       6.594823
min             1.0
25%             5.0
50%             9.0
75%            16.0
max            60.0
Name: floors_total, dtype: Float64
```

In [57]:
```
plt.hist(df['city_centers_nearest'], bins=7, range=(1,7))
plt.title('Растояние до центра города')
plt.xlabel('Растояние до центра города')
plt.ylabel('Количество ')
plt.show()
print('Наибольшее  растояние до центра города:')
print(df['city_centers_nearest'].sort_values(ascending=False).head(10))
print(df['city_centers_nearest'].describe())
```




```
Наибольшее  растояние до центра города:
23698    999999
11883    999999
4694     999999
8621     999999
17298    999999
17302    999999
17305    999999
4686     999999
8622     999999
17311    999999
Name: city_centers_nearest, dtype: int64
count     23565.000000
mean     233449.688394
std      423022.213340
min           0.000000
25%          10.000000
50%          14.000000
75%          35.000000
max      999999.000000
Name: city_centers_nearest, dtype: float64
```

In [58]:
```
plt.hist(df['airports_nearest'], bins=7, range=(1,7))
plt.title('Растояние до аэропорта')
plt.xlabel('Растояние до аэропорта')
plt.ylabel('Количество ')
plt.show()
print('Наибольшее растояние до аэропорта:')
print(df['airports_nearest'].sort_values(ascending=False).head(10))
print(df['airports_nearest'].describe())
```

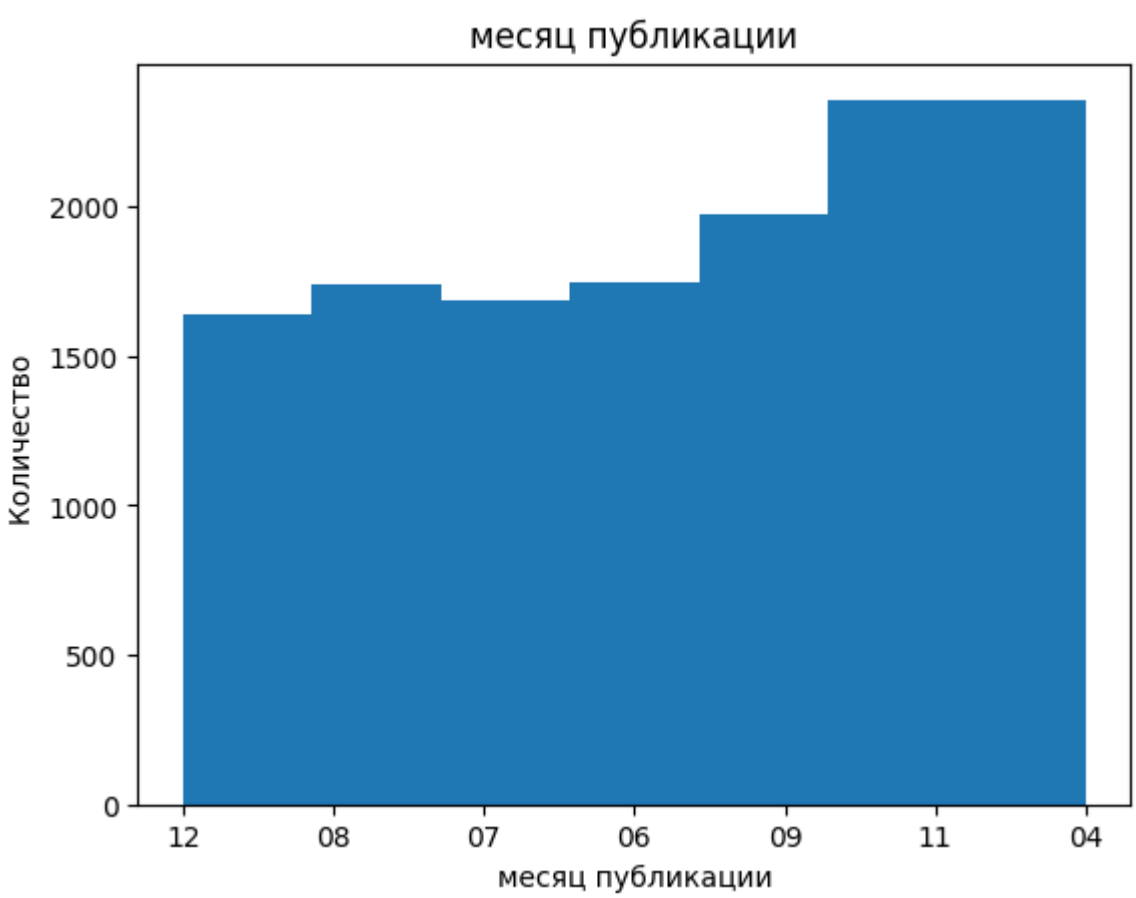



```
Наибольшее растояние до аэропорта:
23698    999999
8812     999999
8809     999999
20900    999999
20901    999999
20902    999999
20906    999999
20908    999999
8786     999999
8784     999999
Name: airports_nearest, dtype: int64
count     23565.000000
mean     234436.864375
std      423629.497058
min           0.000000
25%          21.000000
50%          33.000000
75%          54.000000
max      999999.000000
Name: airports_nearest, dtype: float64
```

В последних столбцах также нет анамальных значений или искажений.

Расмотрим день и месяц публикации

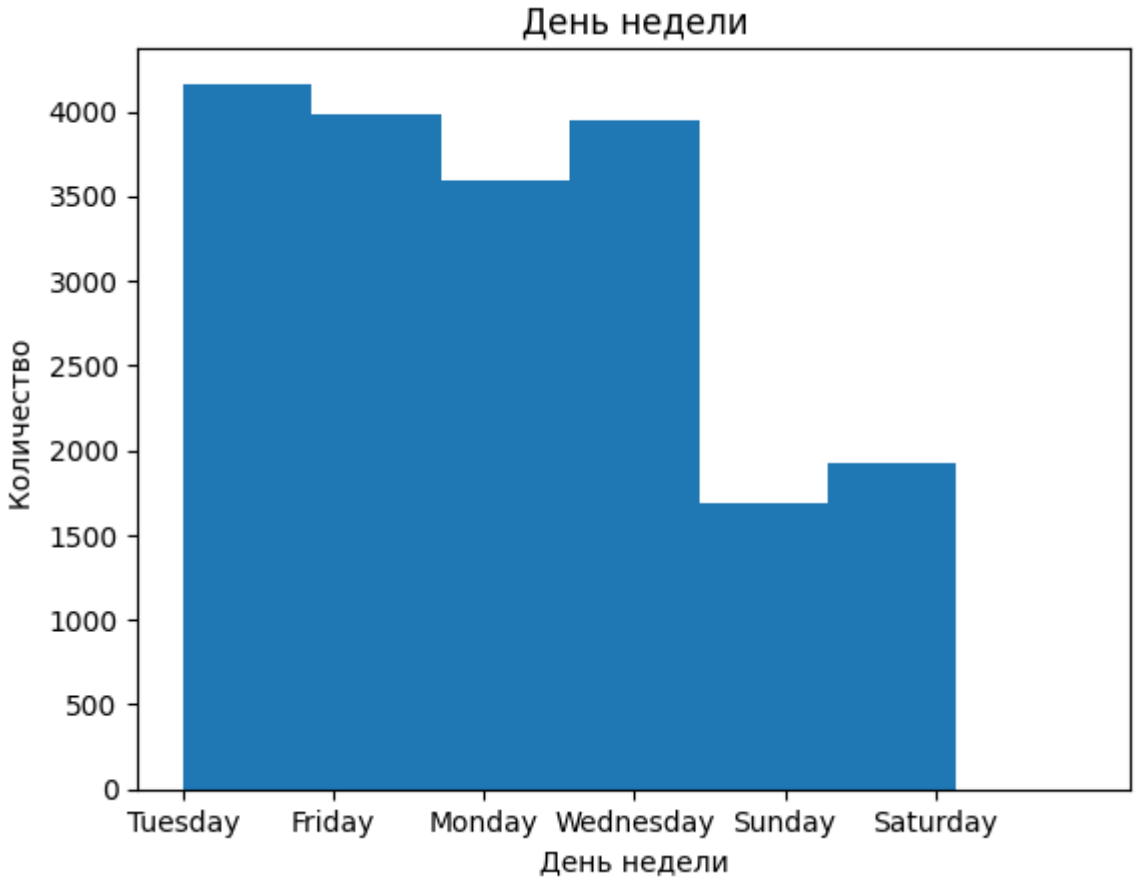
In [59]:
```
plt.hist(df['month_exposition'], bins=7, range=(1,7))
plt.title('месяц публикации')
plt.xlabel('месяц публикации')
plt.ylabel('Количество ')
plt.show()
print('месяц публикации')

print(df['month_exposition'].describe())
```




```
месяц публикации
count     23565
unique       12
top          02
freq       2636
Name: month_exposition, dtype: object
```

Самые частые месяца февраль, март , апрель и далее идет всплеск осенью.

In [60]:
```
plt.hist(df['weekday_exposition'], bins=7, range=(1,7))
plt.title('День недели')
plt.xlabel('День недели')
plt.ylabel('Количество ')
plt.show()
print('Самые частые дни недели:')
print(df['weekday_exposition'].sort_values(ascending=False).head(10))
print(df['weekday_exposition'].describe())
```




```
Самые частые дни недели:
2882     Wednesday
7973     Wednesday
17565    Wednesday
7994     Wednesday
2851     Wednesday
22376    Wednesday
12487    Wednesday
17560    Wednesday
17559    Wednesday
15480    Wednesday
Name: weekday_exposition, dtype: object
count        23565
unique           7
top       Thursday
freq          4276
Name: weekday_exposition, dtype: object
```

В основном пн и ср. - это дни в которые появляется наибольшее число объявлений

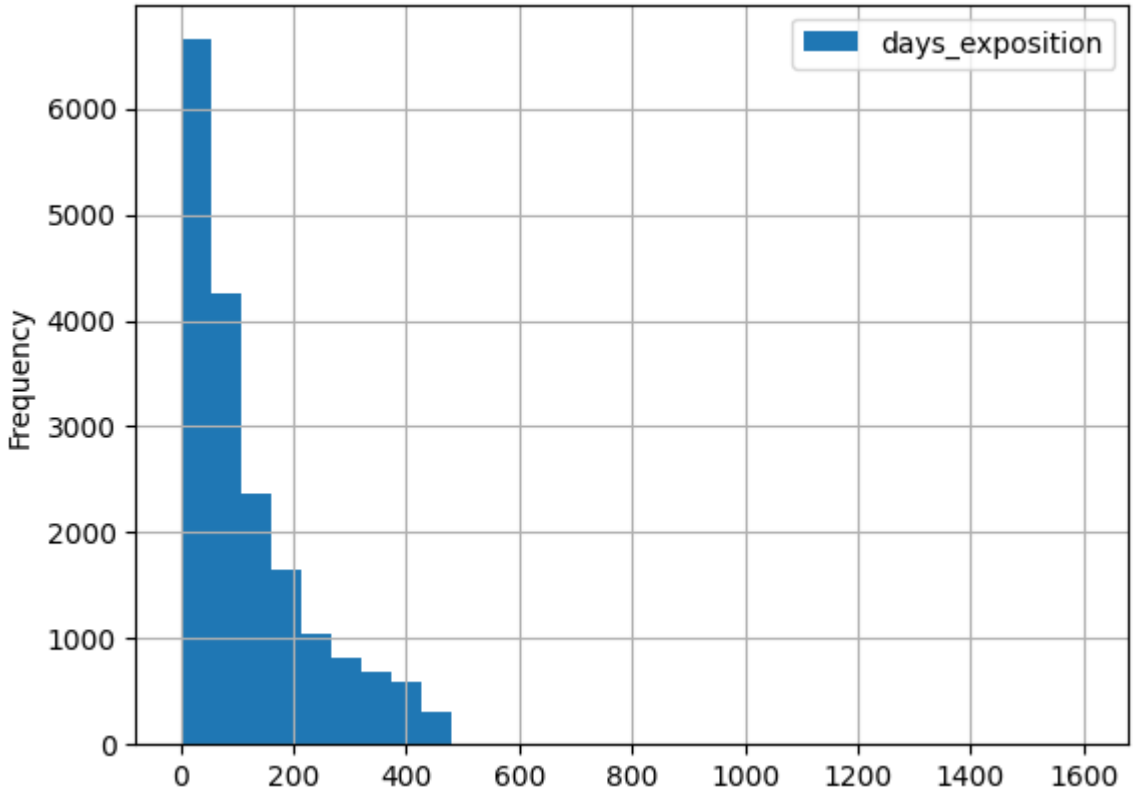
Изучим как продаются квартиры

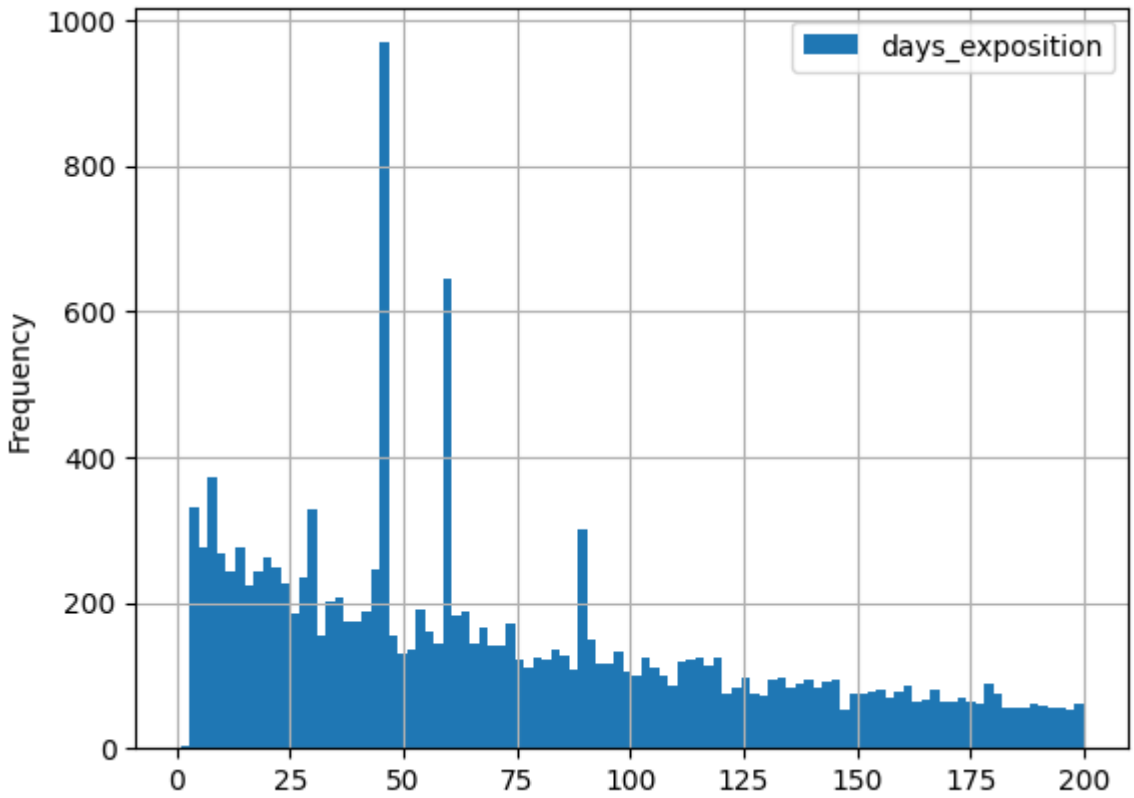
In [61]:

```
#диаграмма размаха
import matplotlib.pyplot as plt


df.plot(y = 'days_exposition', kind = 'hist', bins = 30, grid = True, range = (1,1600
df.plot(y = 'days_exposition', kind = 'hist', bins = 100, grid = True, range = (1,200

#среднее значение, медиана и межквартильный размах
df[df['days_exposition']!=0]['days_exposition'].describe()

print(f'Среднее время продажи квартиры в днях:',int(df['days_exposition'].mean()))
print('Медианное время продажи квартиры в днях:',int(df['days_exposition'].median()))

print('\n[Выбросы] Количество объявлений, которые сняты через:')
print('45 дней:',df[df['days_exposition']==45]['days_exposition'].count())
```

```
print('60 дней:',df[df['days_exposition']==60]['days_exposition'].count())
print('90 дней:',df[df['days_exposition']==90]['days_exposition'].count())
```

```
Среднее время продажи квартиры в днях: 103
Медианное время продажи квартиры в днях: 60

[Выбросы] Количество объявлений, которые сняты через:
45 дней: 879
60 дней: 538
90 дней: 200
```





Мы видим распределение Пуассона. Сделав гистограмму со значениями от 0 до 200 , обнаружили на 7, 30, 45, 60, 90 днях - это наиболее популярное количество дней до снятия объявления.

Мы видим "длинный хвост" квартир, которые продавались очень долго. Выбросы похожи на платные объявлени с истекшим сроком размещения или работу ситемы удаления неактивных

объявлений.

In [62]:
```
df['days_exposition'].describe()
```

Out[62]:
```
count    21539.000000
mean       103.573889
std        111.664697
min          0.000000
25%         18.000000
50%         60.000000
75%        153.000000
max        461.000000
Name: days_exposition, dtype: float64
```

Быстрыми продажи до 18 дней, а необычно долгими - свыше 153 дня.

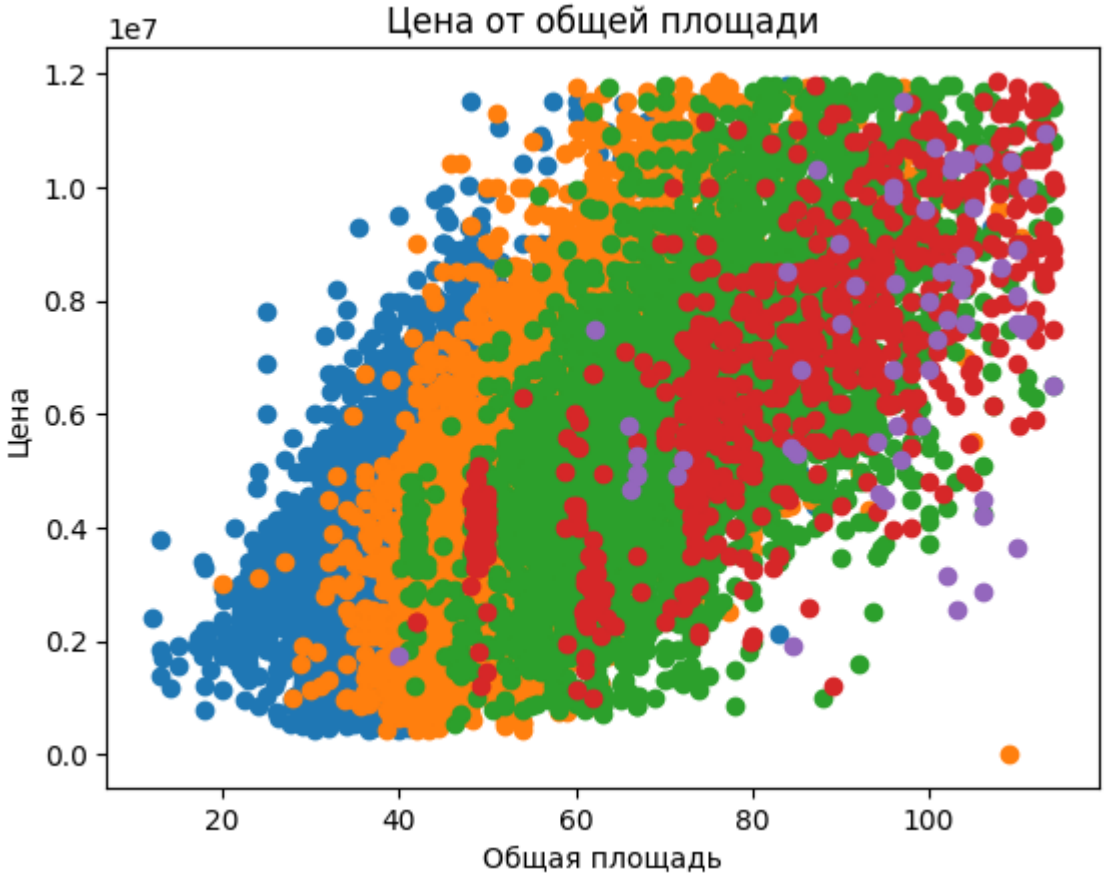
Изучите, зависит ли цена от:

- общей площади;
- жилой площади;
- площади кухни;
- количества комнат;
- этажа, на котором расположена квартира (первый, последний, другой);
- даты размещения (день недели, месяц, год).

Постройте графики, которые покажут зависимость цены от указанных выше параметров. Для подготовки данных перед визуализацией вы можете использовать сводные таблицы.

Цена от общей площади

In [72]:
```
#Построение графика

for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['total_area'],
            df[df['rooms']==i]['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('Общая площадь')
plt.ylabel('Цена')
plt.title('Цена от общей площади')

# Отображение графика
plt.show()
```



В общем цена растетет от общей площади(цветами отражены 1,2,3,4,5комнатныеквартиры).

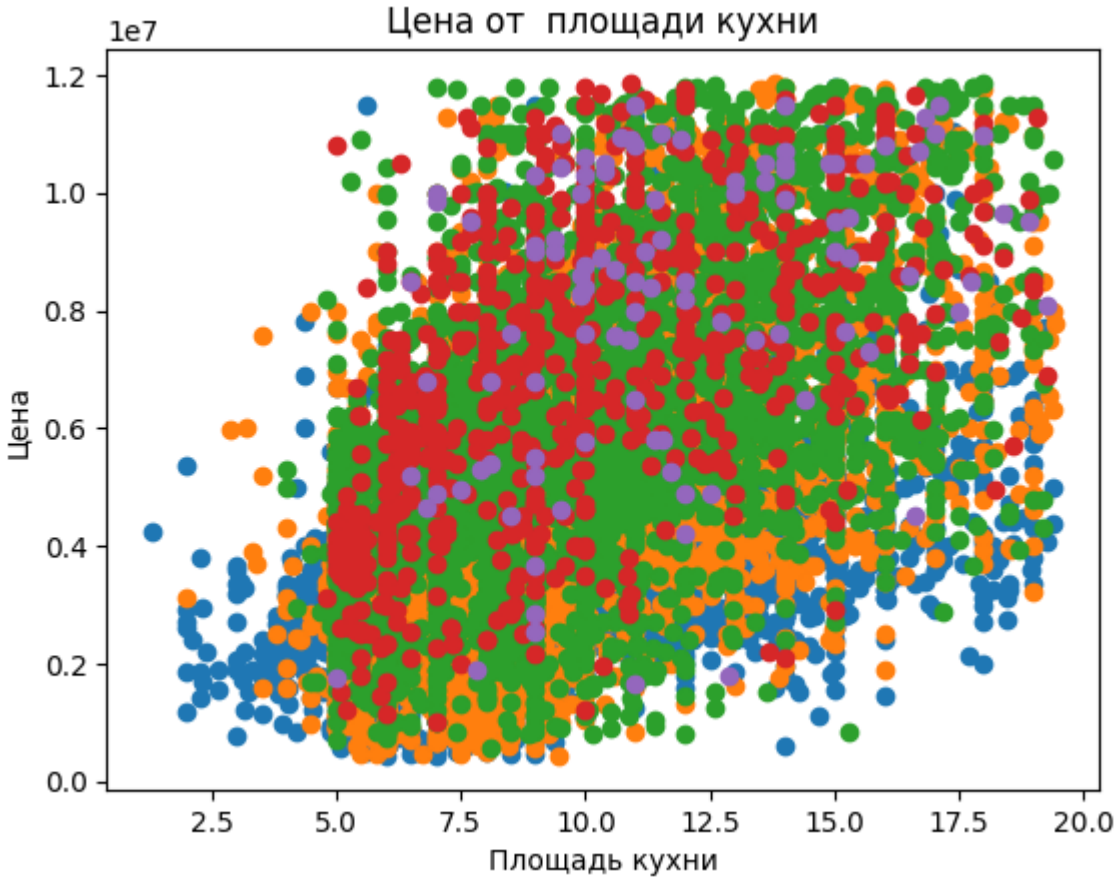
Цена от жилой площади

In [73]:
```
#Построение графика
for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['living_area'],
            df[df['rooms']==i]['last_price'])

#plt.scatter(df['living_area'], df['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('Жилая площадь')
plt.ylabel('Цена')
plt.title('Цена от жилой площади')

# Отображение графика
plt.show()
```

Цена от жилой площади проявляет сходную зависимость, как и полная, но для однокомнатных и для двухкомнатных(до 6млн) цена может быть разной при одной и тойже площади,т.е. зависит от других причин. Явное разделение цен жилой площади от количества комнат, чем больше комнат тем цена ниже.

Цена от площади кухни

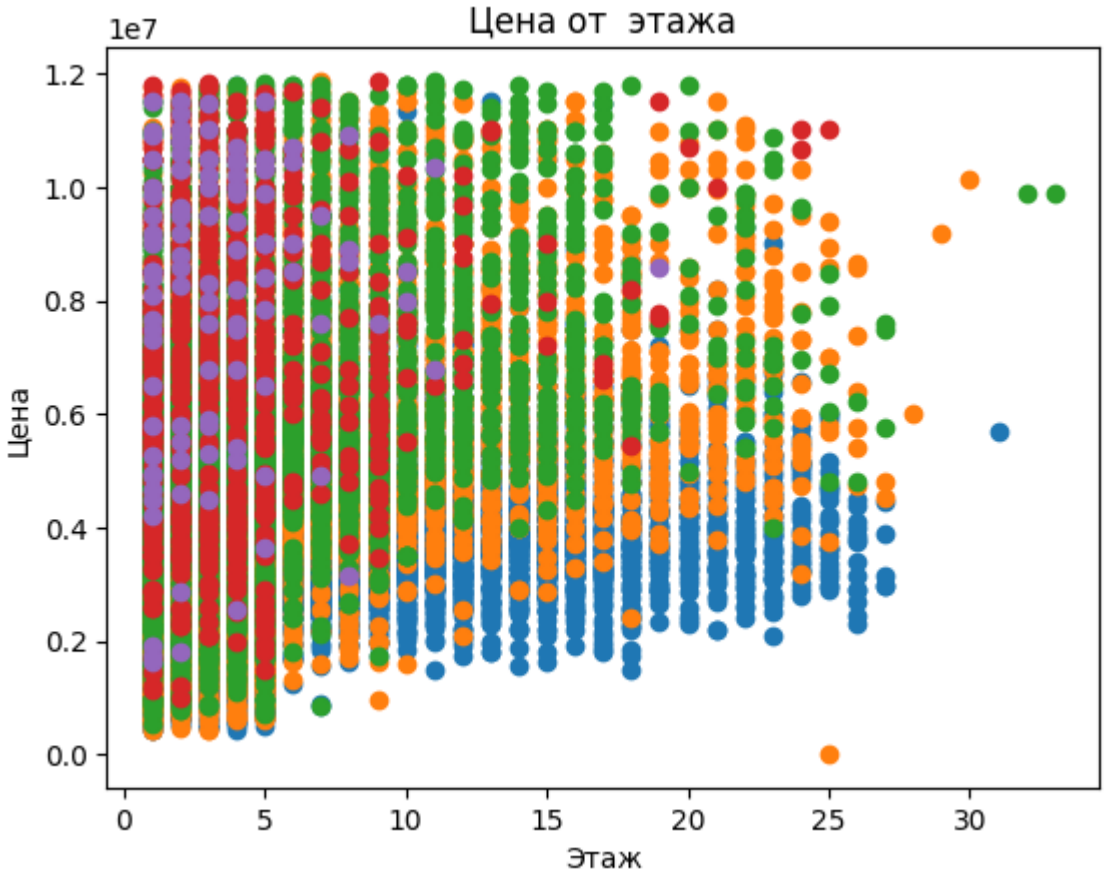
In [74]:
```
#Построение графика
for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['kitchen_area'],
            df[df['rooms']==i]['last_price'])


#plt.scatter(df['kitchen_area'], df['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('Площадь кухни')
plt.ylabel('Цена')
plt.title('Цена от  площади кухни')

# Отображение графика
plt.show()
```



In [ ]:
```
цена от площади кухни возрастает и почти не завист от колчества комнат
```

Цена от этажа

In [75]:
```
#Построение графика
for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['floor'],
            df[df['rooms']==i]['last_price'])


#plt.scatter(df['kitchen_area'], df['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('Этаж')
plt.ylabel('Цена')
plt.title('Цена от  этажа')

# Отображение графика
plt.show()
```

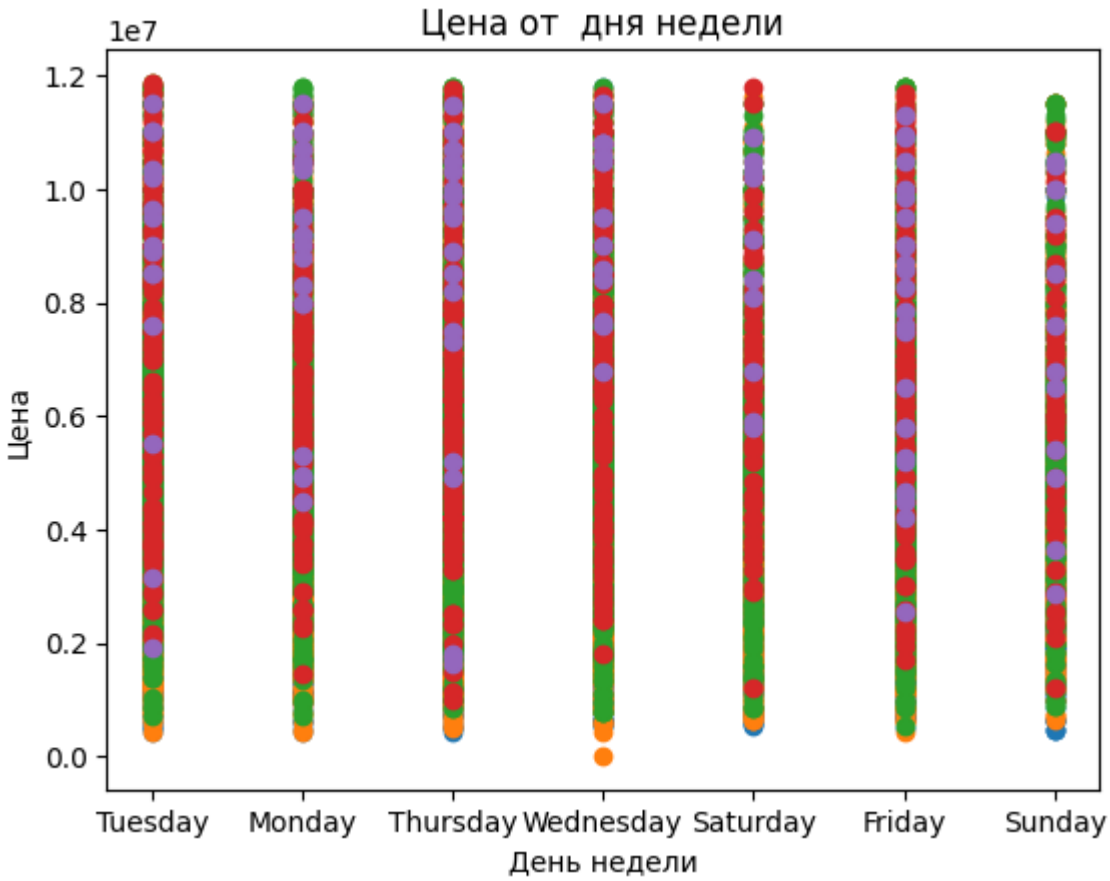


Цена 1-квартир от 1до 4,5 этажей почти не зависит от этажа и может быть различной от 2млн до 12млн Цена 1-квартир на последних этажах(10-25) почти не зависит от этажа и равна примерно 3млн 2-х комнатные квартиры предпочитают с 1 по 10 этаж, цена может различаться от1.5млн до 12млнт 3-х комнатные (кроме самых дешовых до 2млн) предпочитают от 6 до 18этажа 4-5 комнатные предпочитают с10 этажа и выше.

Цена от даты размещения

День недели

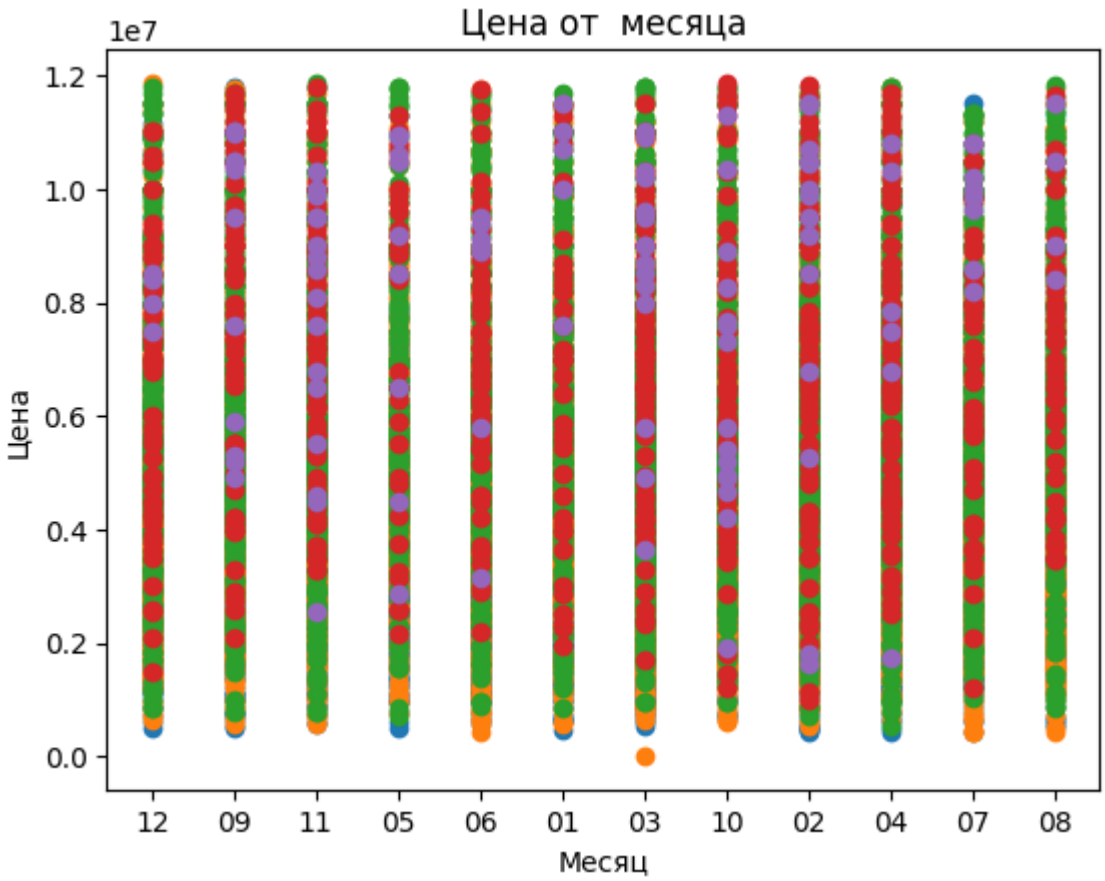
In [77]:
```
#Построение графика
for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['weekday_exposition'],
            df[df['rooms']==i]['last_price'])


#plt.scatter(df['kitchen_area'], df['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('День недели')
plt.ylabel('Цена')
plt.title('Цена от  дня недели')

# Отображение графика
plt.show()
```



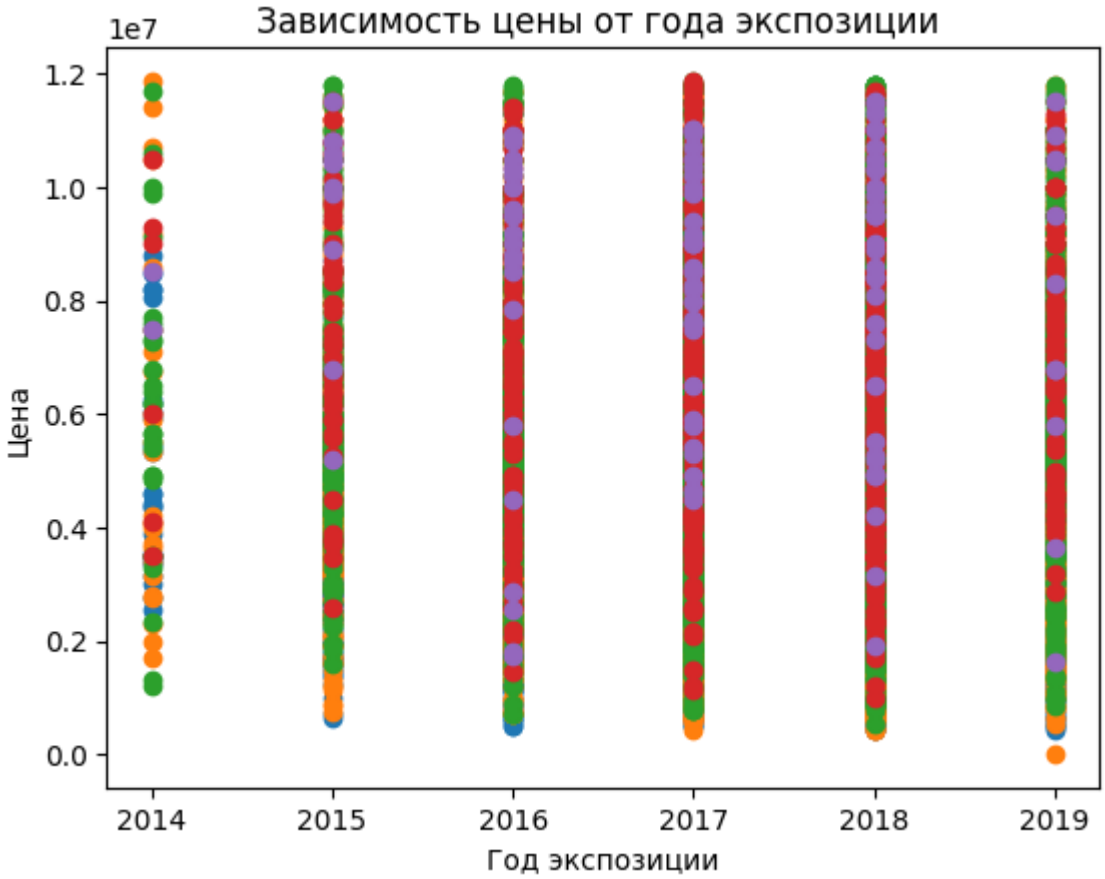
1-комнатные дорогие размещают каждый день 1 -комнатные дешевые -пятница, воскресеннье (еще немного понедельник и четверг) 2-х(нижний ценовой диапозон) и 3-х комнатные (средний и высокий диапозон) -каждый день. дорогие 3-х комнатны в основом -воскресенье(немного пятница и суббота)

Цена от месяца

In [79]:
```
#Построение графика
for i in range(1,6):
    plt.scatter(df[df['rooms']==i]['month_exposition'],
            df[df['rooms']==i]['last_price'])


#plt.scatter(df['kitchen_area'], df['last_price'])

# Настройка осей и заголовка графика
plt.xlabel('Месяц')
plt.ylabel('Цена')
plt.title('Цена от  месяца')

# Отображение графика
plt.show()
```



In [82]:
```
#Построение графика
for i in range(1,6):
    # Фильтрация данных и сортировка по столбцу 'year_exposition'
    x = df[df['rooms'] == i]['year_exposition']
    y = df[df['rooms'] == i]['last_price']
    sort_order = np.argsort(x)  # Получение индексов, отсортированных по 'year_exposi
    x_sorted = x.iloc[sort_order]  # Сортировка значений 'year_exposition'
    y_sorted = y.iloc[sort_order]  # Сортировка соответствующих значений 'last_price'
    plt.scatter(x_sorted, y_sorted)
plt.xlabel('Год экспозиции')
plt.ylabel('Цена')
plt.title('Зависимость цены от года экспозиции ')
plt.show()
```



In [ ]:
```
Из графиков видно что:
    рынок 1-х уменьшается и склонен к увелечению цены
    рынок 2-х комнатных переходитв средний и высокий ценовой диапозоны
    рынок 3-х комнатных разделился на низкой цены(до 3млн) и на высокой цены ( от 9мл
```