

# Исследование объявлений о продаже квартир

В вашем распоряжении данные сервиса Яндекс Недвижимость — архив объявлений о продаже квартир в Санкт-Петербурге и соседних населённых пунктах за несколько лет. Вам нужно научиться определять рыночную стоимость объектов недвижимости. Для этого проведите исследовательский анализ данных и установите параметры, влияющие на цену объектов. 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 и других объектов — эти данные автоматически получены из геосервисов. Количество парков и водоёмов также заполняется без участия пользователя.

## Откройте файл с данными и изучите общую информацию

```
In [520... import pandas as pd
import matplotlib.pyplot as plt
```

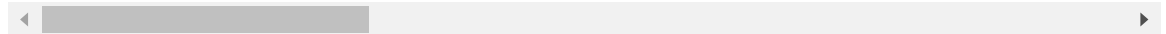
```
In [521... data = pd.read_csv('/datasets/real_estate_data.csv', sep='\t')

data.head(10)
```

Out[521...

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floc
0	20	13000000.0	108.00	2019-03-07T00:00:00	3	2.70	
1	7	3350000.0	40.40	2018-12-04T00:00:00	1	NaN	
2	10	5196000.0	56.00	2015-08-20T00:00:00	2	NaN	
3	0	64900000.0	159.00	2015-07-24T00:00:00	3	NaN	
4	2	10000000.0	100.00	2018-06-19T00:00:00	2	3.03	
5	10	2890000.0	30.40	2018-09-10T00:00:00	1	NaN	
6	6	3700000.0	37.30	2017-11-02T00:00:00	1	NaN	
7	5	7915000.0	71.60	2019-04-18T00:00:00	2	NaN	
8	20	2900000.0	33.16	2018-05-23T00:00:00	1	NaN	
9	18	5400000.0	61.00	2017-02-26T00:00:00	3	2.50	

10 rows × 22 columns



In [522...

```
data.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

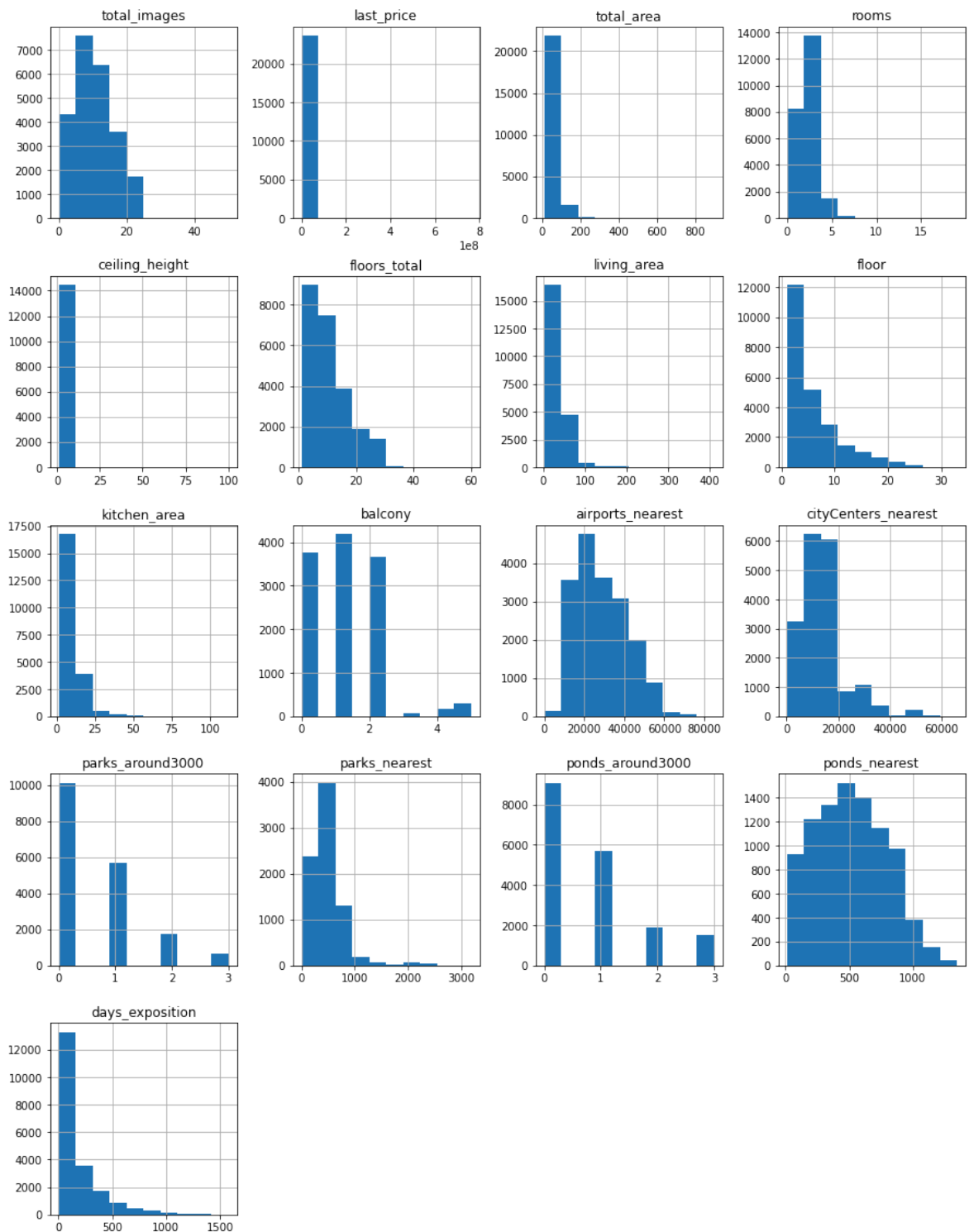
```

```
In [523... data.hist(figsize=(15, 20))# построим гистограммы
```

```

Out[523... array([[<AxesSubplot:title={'center':'total_images'}>,
  <AxesSubplot:title={'center':'last_price'}>,
  <AxesSubplot:title={'center':'total_area'}>,
  <AxesSubplot:title={'center':'rooms'}>],
[<AxesSubplot:title={'center':'ceiling_height'}>,
  <AxesSubplot:title={'center':'floors_total'}>,
  <AxesSubplot:title={'center':'living_area'}>,
  <AxesSubplot:title={'center':'floor'}>],
[<AxesSubplot:title={'center':'kitchen_area'}>,
  <AxesSubplot:title={'center':'balcony'}>,
  <AxesSubplot:title={'center':'airports_nearest'}>,
  <AxesSubplot:title={'center':'cityCenters_nearest'}>],
[<AxesSubplot:title={'center':'parks_around3000'}>,
  <AxesSubplot:title={'center':'parks_nearest'}>,
  <AxesSubplot:title={'center':'ponds_around3000'}>,
  <AxesSubplot:title={'center':'ponds_nearest'}>],
[<AxesSubplot:title={'center':'days_exposition'}>, <AxesSubplot:>,
  <AxesSubplot:>, <AxesSubplot:>]], dtype=object)

```



**Вывод:** Большое количество данных не имеют нормального распределения, возможно, это связано с выбросами. Надо проверить на аномалии `last_price`, `total_area`, `ceiling_height`, `living_area`, `parks_nearest`

Выполните предобработку данных

Проверим количество пропущенных значений

In [524...

```
data.isna().sum()
```

```
Out[524... total_images      0
last_price        0
total_area        0
first_day_exposition  0
rooms            0
ceiling_height    9195
floors_total      86
living_area       1903
floor            0
is_apartment      20924
studio           0
open_plan        0
kitchen_area     2278
balcony          11519
locality_name     49
airports_nearest  5542
cityCenters_nearest  5519
parks_around3000  5518
parks_nearest    15620
ponds_around3000  5518
ponds_nearest    14589
days_exposition  3181
dtype: int64
```

```
In [525... data.isna().sum()/data.shape[0]*100
```

```
Out[525... total_images      0.000000
last_price        0.000000
total_area        0.000000
first_day_exposition  0.000000
rooms            0.000000
ceiling_height    38.799105
floors_total      0.362885
living_area       8.029875
floor            0.000000
is_apartment      88.290645
studio           0.000000
open_plan        0.000000
kitchen_area     9.612220
balcony          48.605426
locality_name     0.206760
airports_nearest  23.384953
cityCenters_nearest  23.287902
parks_around3000  23.283683
parks_nearest    65.909954
ponds_around3000  23.283683
ponds_nearest    61.559559
days_exposition  13.422507
dtype: float64
```

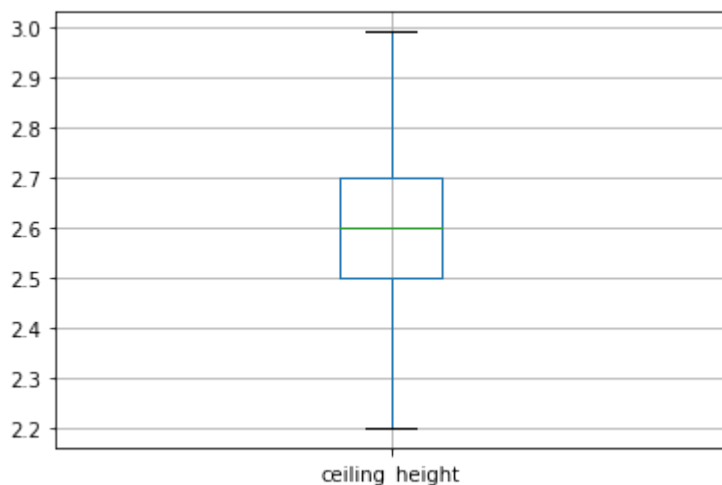
Большое количество пропусков в колонке `is_apartment` 88%, скорее всего связано с отсутствием непосредственно апартаментов. В колонке `balcony` по той же причине отсутствует 48% данных. `airports_nearest`, `cityCenters_nearest`, `ponds_nearest` и остальные похожие колонки заполнены автоматически. Пропуски `living_area`, `kitchen_area`, `floors_total`, `days_exposition` данные отсутствуют. Есть большое количество отсутствия данных в `ceiling_height` 38%

попробуем заполнить медианным значением. `locality_name` заполняем  
Неизвестно

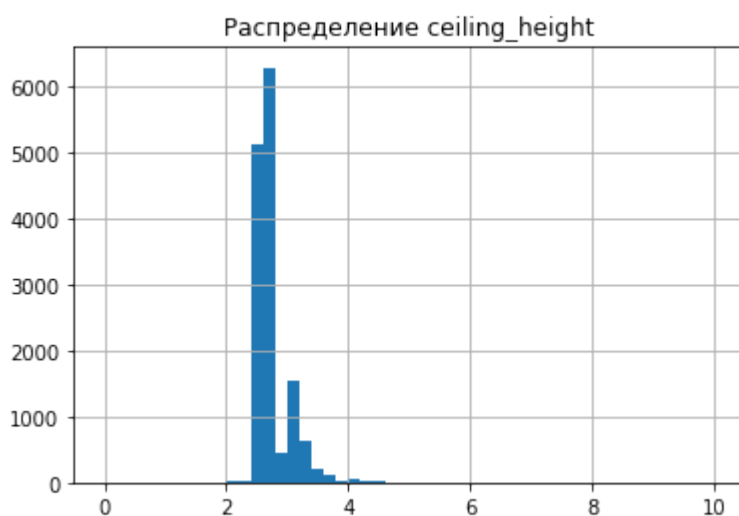
`ceiling_height`

```
In [526... data.query('ceiling_height < 3 and ceiling_height >= 2.2').boxplot(column='ceili
```

Out[526... <AxesSubplot:>



```
In [527... data['ceiling_height'].hist(bins=50,range=(0,10))
plt.title('Распределение ceiling_height')
plt.show()
```

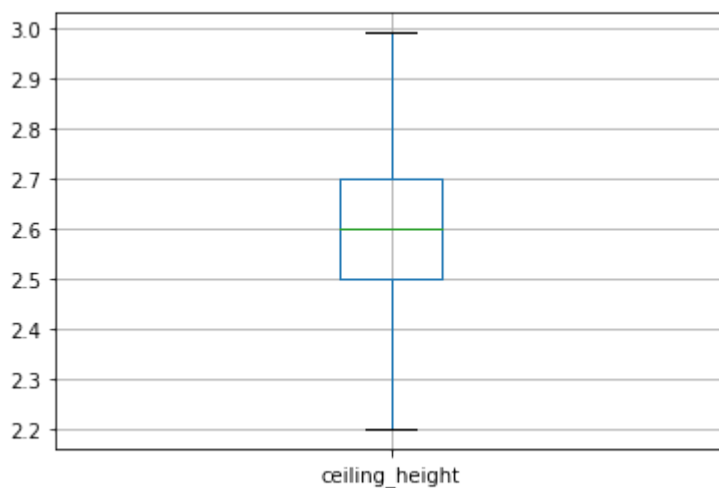


```
In [528... data['ceiling_height'].describe()
```

```
Out[528... count    14504.000000
mean       2.771499
std        1.261056
min        1.000000
25%        2.520000
50%        2.650000
75%        2.800000
max        100.000000
Name: ceiling_height, dtype: float64
```

```
In [529... data.query('ceiling_height < 3 and ceiling_height >= 2.2').boxplot(column='ceili
```

Out[529... <AxesSubplot:>



Получили нормальное распределение высоты потолков, заполним отсутствующие значения

```
In [530... data['ceiling_height'] = data['ceiling_height'].fillna(data['ceiling_height'].median())
data['ceiling_height'].isna().sum()
```

Out[530... 0

**Вывод:** Заполнили `ceiling_height` медианным значением

Заполнение пропусков в `is_apartment`

```
In [531... data['is_apartment']=data['is_apartment'].fillna(False)
```

Заполнение пропусков в `balcony`

```
In [532... data['balcony']=data['balcony'].fillna(0)
```

`locality_name` NaN заполняем на `Неизвестно`

```
In [533... data['locality_name'] = data['locality_name'].fillna('Неизвестно')
```

```
In [534... _list=['cityCenters_nearest','airports_nearest','parks_around3000','parks_nearest']
for row in _list:
    data[row]=data[row].fillna(0)
```

```
In [535... for row in _list:
    print(row,data[row].isna().sum())
```

```
cityCenters_nearest 0
airports_nearest 0
parks_around3000 0
parks_nearest 0
ponds_around3000 0
ponds_nearest 0
```

**Вывод** заменим пропущенные значения на 0 в колонках

```
'cityCenters_nearest', 'airports_nearest', 'parks_around3000', 'parks_nearest'
locality_name
```

 заполняли на *Неизвестно*

## Изменение типов данных

In [536...

```
data.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         23699 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           23699 non-null  bool
10  studio                 23699 non-null  bool
11  open_plan              23699 non-null  bool
12  kitchen_area           21421 non-null  float64
13  balcony                23699 non-null  float64
14  locality_name          23699 non-null  object
15  airports_nearest       23699 non-null  float64
16  cityCenters_nearest    23699 non-null  float64
17  parks_around3000       23699 non-null  float64
18  parks_nearest          23699 non-null  float64
19  ponds_around3000       23699 non-null  float64
20  ponds_nearest          23699 non-null  float64
21  days_exposition        20518 non-null  float64
dtypes: bool(3), float64(14), int64(3), object(2)
memory usage: 3.5+ MB
```

In [537...

```
data['first_day_exposition'] = pd.to_datetime(data['first_day_exposition'],forma
```

In [538...

```
data['first_day_exposition']#check
```

Out[538...

```
0      2019-03-07
1      2018-12-04
2      2015-08-20
3      2015-07-24
4      2018-06-19
...
23694   2017-03-21
23695   2018-01-15
23696   2018-02-11
23697   2017-03-28
23698   2017-07-21
Name: first_day_exposition, Length: 23699, dtype: datetime64[ns]
```

In [539...

```
data['last_price'].astype('int32')
```



```
Out[539... 0          13000000
1           3350000
2           5196000
3          64900000
4          10000000
...
23694       9700000
23695       3100000
23696       2500000
23697      11475000
23698       1350000
Name: last_price, Length: 23699, dtype: int32
```

```
In [540... data['balcony'].astype('int32')
```

```
Out[540... 0          0
1          2
2          0
3          0
4          0
..
23694       0
23695       0
23696       0
23697       2
23698       0
Name: balcony, Length: 23699, dtype: int32
```

Изменены типы данных в `first_day_exposition` на `datetime64` `last_price` на `int`, потому что работаем с ценой и количество балконов.

## Устранение явных и неявных дубликатов

```
In [541... sorted(data['locality_name'].unique())
```

```
Out[541... ['Бокситогорск',
'Волосово',
'Волхов',
'Всеволожск',
'Выборг',
'Высоцк',
'Гатчина',
'Зеленогорск',
'Ивангород',
'Каменногорск',
'Кингисепп',
'Кириши',
'Кировск',
'Колпино',
'Коммунар',
'Красное Село',
'Кронштадт',
'Кудрово',
'Лодейное Поле',
'Ломоносов',
'Луга',
'Любань',
'Мурино',
'Неизвестно',
'Никольское',
'Новая Ладога',
'Отрадное',
'Павловск',
'Петергоф',
'Пикалёво',
'Подпорожье',
'Приморск',
'Приозерск',
'Пушкин',
'Санкт-Петербург',
'Светогорск',
'Сертолово',
'Сестрорецк',
'Сланцы',
'Сосновый Бор',
'Сясьстрой',
'Тихвин',
'Тосно',
'Шлиссельбург',
'городской поселок Большая Ижора',
'городской поселок Янино-1',
'городской посёлок Будогощь',
'городской посёлок Виллози',
'городской посёлок Лесогорский',
'городской посёлок Мга',
'городской посёлок Назия',
'городской посёлок Новоселье',
'городской посёлок Павлово',
'городской посёлок Рощино',
'городской посёлок Свирьстрой',
'городской посёлок Советский',
'городской посёлок Фёдоровское',
'городской посёлок Янино-1',
'деревня Агалатово',
'деревня Аро',
```

'деревня Батово',  
'деревня Бегуницы',  
'деревня Белогорка',  
'деревня Большая Вруда',  
'деревня Большая Пустомержа',  
'деревня Большие Колпаны',  
'деревня Большое Рейзино',  
'деревня Большой Сабск',  
'деревня Бор',  
'деревня Борисова Грива',  
'деревня Ваганово',  
'деревня Вартемяги',  
'деревня Вахнова Кара',  
'деревня Вискатка',  
'деревня Гарболово',  
'деревня Глинка',  
'деревня Горбунки',  
'деревня Гостилицы',  
'деревня Заклинье',  
'деревня Заневка',  
'деревня Зимитицы',  
'деревня Извара',  
'деревня Иссад',  
'деревня Калитино',  
'деревня Кальтино',  
'деревня Камышовка',  
'деревня Каськово',  
'деревня Келози',  
'деревня Кипень',  
'деревня Кисельня',  
'деревня Колтуши',  
'деревня Коркино',  
'деревня Котлы',  
'деревня Кривко',  
'деревня Кудрово',  
'деревня Кузьмолowo',  
'деревня Курковицы',  
'деревня Куровицы',  
'деревня Куттузи',  
'деревня Лаврики',  
'деревня Лаголово',  
'деревня Лампово',  
'деревня Лесколово',  
'деревня Лопухинка',  
'деревня Лупполово',  
'деревня Малая Романовка',  
'деревня Малое Верево',  
'деревня Малое Карлино',  
'деревня Малые Колпаны',  
'деревня Мануйлово',  
'деревня Меньково',  
'деревня Мины',  
'деревня Мистолово',  
'деревня Ненимяки',  
'деревня Нижние Осельки',  
'деревня Нижняя',  
'деревня Низино',  
'деревня Новое Девяткино',  
'деревня Новолисино',  
'деревня Нурма',

'деревня Оржицы',  
'деревня Парицы',  
'деревня Пельгора',  
'деревня Пеники',  
'деревня Пижма',  
'деревня Пикколово',  
'деревня Пудомяги',  
'деревня Пустынка',  
'деревня Пчева',  
'деревня Рабитицы',  
'деревня Разбегаево',  
'деревня Раздолье',  
'деревня Разметелево',  
'деревня Рапполово',  
'деревня Реброво',  
'деревня Русско',  
'деревня Сижно',  
'деревня Снегирёвка',  
'деревня Старая',  
'деревня Старая Пустошь',  
'деревня Старое Хинколово',  
'деревня Старополье',  
'деревня Старосиверская',  
'деревня Старые Бегуницы',  
'деревня Суоранда',  
'деревня Сяськелево',  
'деревня Тарасово',  
'деревня Терпилицы',  
'деревня Тихковицы',  
'деревня Тойворово',  
'деревня Торосово',  
'деревня Торошковицы',  
'деревня Трубников Бор',  
'деревня Фалилеево',  
'деревня Фёдоровское',  
'деревня Хапо-Ое',  
'деревня Хязельки',  
'деревня Чудской Бор',  
'деревня Шпаньково',  
'деревня Щеглово',  
'деревня Юнки',  
'деревня Ялгино',  
'деревня Яльгелево',  
'деревня Ям-Тесово',  
'коттеджный посёлок Кивеннапа Север',  
'коттеджный посёлок Счастье',  
'коттеджный посёлок Лесное',  
'посёлок Аннино',  
'посёлок Барышево',  
'посёлок Бугры',  
'посёлок Возрождение',  
'посёлок Войсковичи',  
'посёлок Володарское',  
'посёлок Гаврилово',  
'посёлок Гарболово',  
'посёлок Гладкое',  
'посёлок Глажево',  
'посёлок Глебычево',  
'посёлок Гончарово',  
'посёлок Громово',

'поселок Дружноселье',  
'поселок Елизаветино',  
'поселок Жилгородок',  
'поселок Жилпосёлок',  
'поселок Житково',  
'поселок Заводской',  
'поселок Запорожское',  
'поселок Зимитицы',  
'поселок Ильичёво',  
'поселок Калитино',  
'поселок Каложицы',  
'поселок Кингисеппский',  
'поселок Кирпичное',  
'поселок Кобралово',  
'поселок Кобринское',  
'поселок Коммунары',  
'поселок Коробицыно',  
'поселок Котельский',  
'поселок Красная Долина',  
'поселок Красносельское',  
'поселок Лесное',  
'поселок Лисий Нос',  
'поселок Лукаши',  
'поселок Любань',  
'поселок Мельниково',  
'поселок Мичуринское',  
'поселок Молодцово',  
'поселок Мурино',  
'поселок Новый Свет',  
'поселок Новый Учхоз',  
'поселок Оредеж',  
'поселок Пансионат Зелёный Бор',  
'поселок Первомайское',  
'поселок Перово',  
'поселок Петровское',  
'поселок Победа',  
'поселок Поляны',  
'поселок Почап',  
'поселок Починок',  
'поселок Пушное',  
'поселок Пчевжа',  
'поселок Рабителицы',  
'поселок Романовка',  
'поселок Ромашки',  
'поселок Рябово',  
'поселок Севастьяново',  
'поселок Селезнёво',  
'поселок Сельцо',  
'поселок Семиозерье',  
'поселок Семрино',  
'поселок Серебрянский',  
'поселок Совхозный',  
'поселок Старая Малукса',  
'поселок Стекланный',  
'поселок Сумино',  
'поселок Суходолье',  
'поселок Тельмана',  
'поселок Терволово',  
'поселок Торковичи',  
'поселок Тёсово-4',

'поселок Углово',  
'поселок Усть-Луга',  
'поселок Ушаки',  
'поселок Цвелодубово',  
'поселок Цвылёво',  
'поселок городского типа Большая Ижора',  
'поселок городского типа Вырица',  
'поселок городского типа Дружная Горка',  
'поселок городского типа Дубровка',  
'поселок городского типа Ефимовский',  
'поселок городского типа Кондратьево',  
'поселок городского типа Красный Бор',  
'поселок городского типа Кузьмолровский',  
'поселок городского типа Лебяжье',  
'поселок городского типа Лесогорский',  
'поселок городского типа Назия',  
'поселок городского типа Никольский',  
'поселок городского типа Приладожский',  
'поселок городского типа Рахья',  
'поселок городского типа Рощино',  
'поселок городского типа Рябово',  
'поселок городского типа Синявино',  
'поселок городского типа Советский',  
'поселок городского типа Токсово',  
'поселок городского типа Форносово',  
'поселок городского типа имени Свердлова',  
'поселок станции Вещево',  
'поселок станции Корнево',  
'поселок станции Лужайка',  
'поселок станции Приветнинское',  
'посёлок Александровская',  
'посёлок Алексеевка',  
'посёлок Аннино',  
'посёлок Белоостров',  
'посёлок Бугры',  
'посёлок Возрождение',  
'посёлок Войскорово',  
'посёлок Высокоключевой',  
'посёлок Гаврилово',  
'посёлок Дзержинского',  
'посёлок Жилгородок',  
'посёлок Ильичёво',  
'посёлок Кикерино',  
'посёлок Кобралово',  
'посёлок Коробицыно',  
'посёлок Левашово',  
'посёлок Ленинское',  
'посёлок Лисий Нос',  
'посёлок Мельниково',  
'посёлок Металлострой',  
'посёлок Мичуринское',  
'посёлок Молодёжное',  
'посёлок Мурино',  
'посёлок Мыза-Ивановка',  
'посёлок Новогорелово',  
'посёлок Новый Свет',  
'посёлок Пансионат Зелёный Бор',  
'посёлок Парголово',  
'посёлок Перово',  
'посёлок Песочный',

'посёлок Петро-Славянка',  
'посёлок Петровское',  
'посёлок Платформа 69-й километр',  
'посёлок Плодовое',  
'посёлок Плоское',  
'посёлок Победа',  
'посёлок Поляны',  
'посёлок Понтонный',  
'посёлок Пригородный',  
'посёлок Пудость',  
'посёлок Репино',  
'посёлок Ропша',  
'посёлок Сапёрное',  
'посёлок Сапёрный',  
'посёлок Сосново',  
'посёлок Старая Малукса',  
'посёлок Стекланный',  
'посёлок Стрельна',  
'посёлок Суйда',  
'посёлок Сумино',  
'посёлок Тельмана',  
'посёлок Терволово',  
'посёлок Торфяное',  
'посёлок Усть-Ижора',  
'посёлок Усть-Луга',  
'посёлок Форт Красная Горка',  
'посёлок Шугозеро',  
'посёлок Шушары',  
'посёлок Щеглово',  
'посёлок городского типа Важины',  
'посёлок городского типа Вознесенье',  
'посёлок городского типа Вырица',  
'посёлок городского типа Красный Бор',  
'посёлок городского типа Кузнечное',  
'посёлок городского типа Кузьмоловский',  
'посёлок городского типа Лебяжье',  
'посёлок городского типа Мга',  
'посёлок городского типа Павлово',  
'посёлок городского типа Роцино',  
'посёлок городского типа Рябово',  
'посёлок городского типа Сиверский',  
'посёлок городского типа Тайцы',  
'посёлок городского типа Токсово',  
'посёлок городского типа Ульяновка',  
'посёлок городского типа Форносово',  
'посёлок городского типа имени Морозова',  
'посёлок городского типа имени Свердлова',  
'посёлок при железнодорожной станции Вещево',  
'посёлок при железнодорожной станции Приветнинское',  
'посёлок станции Громово',  
'посёлок станции Свирь',  
'садоводческое некоммерческое товарищество Лесная Поляна',  
'садовое товарищество Новая Ропша',  
'садовое товарищество Приладожский',  
'садовое товарищество Рахья',  
'садовое товарищество Садко',  
'село Копорье',  
'село Никольское',  
'село Павлово',  
'село Паша',

```
'село Путилово',  
'село Рождествено',  
'село Русско-Высоцкое',  
'село Старая Ладога',  
'село Шум']
```

```
In [542... data['locality_name'].nunique()
```

```
Out[542... 365
```

```
In [543... data['locality_name'] = (  
    data['locality_name']  
    .str.replace('ё', 'e')  
    .replace(['поселок при железнодорожной станции', 'городской поселок', 'коттедж  
)
```

```
In [544... data['locality_name'].nunique()
```

```
Out[544... 319
```

```
In [545... sorted(data['locality_name'].unique())
```



```
Out[545... ['Бокситогорск',  
            'Волосово',  
            'Волхов',  
            'Всеволожск',  
            'Выборг',  
            'Высоцк',  
            'Гатчина',  
            'Зеленогорск',  
            'Ивангород',  
            'Каменногорск',  
            'Кингисепп',  
            'Кириши',  
            'Кировск',  
            'Колпино',  
            'Коммунар',  
            'Красное Село',  
            'Кронштадт',  
            'Кудрово',  
            'Лодейное Поле',  
            'Ломоносов',  
            'Луга',  
            'Любань',  
            'Мурино',  
            'Неизвестно',  
            'Никольское',  
            'Новая Ладога',  
            'Отрадное',  
            'Павловск',  
            'Петергоф',  
            'Пикалево',  
            'Подпорожье',  
            'Приморск',  
            'Приозерск',  
            'Пушкин',  
            'Санкт-Петербург',  
            'Светогорск',  
            'Сертолово',  
            'Сестрорецк',  
            'Сланцы',  
            'Сосновый Бор',  
            'Сясьстрой',  
            'Тихвин',  
            'Тосно',  
            'Шлиссельбург',  
            'деревня Агалатово',  
            'деревня Аро',  
            'деревня Батово',  
            'деревня Бегуницы',  
            'деревня Белогорка',  
            'деревня Большая Вруда',  
            'деревня Большая Пустомержа',  
            'деревня Большие Колпаны',  
            'деревня Большое Рейзино',  
            'деревня Большой Сабск',  
            'деревня Бор',  
            'деревня Борисова Грива',  
            'деревня Ваганово',  
            'деревня Вартемяги',  
            'деревня Вахнова Кара',  
            'деревня Вискатка',
```

'деревня Гарболово',  
'деревня Глинка',  
'деревня Горбунки',  
'деревня Гостилицы',  
'деревня Заклинье',  
'деревня Заневка',  
'деревня Зимитицы',  
'деревня Извара',  
'деревня Иссад',  
'деревня Калитино',  
'деревня Кальтино',  
'деревня Камышовка',  
'деревня Каськово',  
'деревня Келози',  
'деревня Кипень',  
'деревня Кисельня',  
'деревня Колтуши',  
'деревня Коркино',  
'деревня Котлы',  
'деревня Кривко',  
'деревня Кудрово',  
'деревня Кузьмолово',  
'деревня Курковицы',  
'деревня Куровицы',  
'деревня Куттузи',  
'деревня Лаврики',  
'деревня Лаголово',  
'деревня Лампово',  
'деревня Лесколово',  
'деревня Лопухинка',  
'деревня Лупполово',  
'деревня Малая Романовка',  
'деревня Малое Верево',  
'деревня Малое Карлино',  
'деревня Малые Колпаны',  
'деревня Мануйлово',  
'деревня Меньково',  
'деревня Мины',  
'деревня Мистолово',  
'деревня Ненимяки',  
'деревня Нижние Осельки',  
'деревня Нижняя',  
'деревня Низино',  
'деревня Новое Девяткино',  
'деревня Новолисино',  
'деревня Нурма',  
'деревня Оржицы',  
'деревня Парицы',  
'деревня Пельгора',  
'деревня Пеники',  
'деревня Пижма',  
'деревня Пикколово',  
'деревня Пудомяги',  
'деревня Пустынка',  
'деревня Пчева',  
'деревня Рабитицы',  
'деревня Разбегаево',  
'деревня Раздолье',  
'деревня Разметелево',  
'деревня Рапполово',

'деревня Реброво',  
'деревня Русско',  
'деревня Сижно',  
'деревня Снегиревка',  
'деревня Старая',  
'деревня Старая Пустошь',  
'деревня Старое Хинколово',  
'деревня Старополье',  
'деревня Старосиверская',  
'деревня Старые Бегуницы',  
'деревня Суоранда',  
'деревня Сяськелево',  
'деревня Тарасово',  
'деревня Терпилицы',  
'деревня Тиховицы',  
'деревня Тойворово',  
'деревня Торосово',  
'деревня Торшковичи',  
'деревня Трубников Бор',  
'деревня Фалилеево',  
'деревня Федоровское',  
'деревня Хапо-Ое',  
'деревня Хязельки',  
'деревня Чудской Бор',  
'деревня Шпаньково',  
'деревня Щеглово',  
'деревня Юкки',  
'деревня Ялгино',  
'деревня Яльгелево',  
'деревня Ям-Тесово',  
'поселок Александровская',  
'поселок Алексеевка',  
'поселок Аннино',  
'поселок Барышево',  
'поселок Белоостров',  
'поселок Большая Ижора',  
'поселок Бугры',  
'поселок Будогощь',  
'поселок Важины',  
'поселок Вещево',  
'поселок Виллози',  
'поселок Вознесенье',  
'поселок Возрождение',  
'поселок Войсковицы',  
'поселок Войскорово',  
'поселок Володарское',  
'поселок Вырица',  
'поселок Высокоключевой',  
'поселок Гаврилово',  
'поселок Гарболово',  
'поселок Гладкое',  
'поселок Глажево',  
'поселок Глебычево',  
'поселок Гончарово',  
'поселок Громово',  
'поселок Дзержинского',  
'поселок Дружная Горка',  
'поселок Дружноселье',  
'поселок Дубровка',  
'поселок Елизаветино',

'поселок Ефимовский',  
'поселок Жилгородок',  
'поселок Жилпоселок',  
'поселок Житково',  
'поселок Заводской',  
'поселок Запорожское',  
'поселок Зимитицы',  
'поселок Ильичево',  
'поселок Калитино',  
'поселок Каложицы',  
'поселок Кивеннапа Север',  
'поселок Кикерино',  
'поселок Кингисеппский',  
'поселок Кирпичное',  
'поселок Кобралово',  
'поселок Кобринское',  
'поселок Коммунары',  
'поселок Кондратьево',  
'поселок Корнево',  
'поселок Коробицыно',  
'поселок Котельский',  
'поселок Красная Долина',  
'поселок Красносельское',  
'поселок Красный Бор',  
'поселок Кузнечное',  
'поселок Кузьмолровский',  
'поселок Лебяжье',  
'поселок Левашово',  
'поселок Ленинское',  
'поселок Лесное',  
'поселок Лесогорский',  
'поселок Лисий Нос',  
'поселок Лужайка',  
'поселок Лукаши',  
'поселок Любань',  
'поселок Мга',  
'поселок Мельниково',  
'поселок Металлострой',  
'поселок Мичуринское',  
'поселок Молодежное',  
'поселок Молодцово',  
'поселок Мурино',  
'поселок Мыза-Ивановка',  
'поселок Назия',  
'поселок Никольский',  
'поселок Новогорелово',  
'поселок Новоселье',  
'поселок Новый Свет',  
'поселок Новый Учхоз',  
'поселок Оредеж',  
'поселок Павлово',  
'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 Приветнинское',  
'поселок Пригородный',  
'поселок Приладожский',  
'поселок Пудость',  
'поселок Пушное',  
'поселок Пчевжа',  
'поселок Рабитицы',  
'поселок Рахья',  
'поселок Репино',  
'поселок Романовка',  
'поселок Ромашки',  
'поселок Ропша',  
'поселок Рощино',  
'поселок Рябово',  
'поселок Саперное',  
'поселок Саперный',  
'поселок Свирь',  
'поселок Свирьстрой',  
'поселок Севастьяново',  
'поселок Селезнево',  
'поселок Сельцо',  
'поселок Семиозерье',  
'поселок Семрино',  
'поселок Серебрянский',  
'поселок Сиверский',  
'поселок Синявино',  
'поселок Советский',  
'поселок Совхозный',  
'поселок Сосново',  
'поселок Старая Малукса',  
'поселок Стекланный',  
'поселок Стрельна',  
'поселок Суйда',  
'поселок Сумино',  
'поселок Суходолье',  
'поселок Счастье',  
'поселок Тайцы',  
'поселок Тельмана',  
'поселок Терволово',  
'поселок Тесово-4',  
'поселок Токсово',  
'поселок Торковичи',  
'поселок Торфяное',  
'поселок Углово',  
'поселок Ульяновка',  
'поселок Усть-Ижора',  
'поселок Усть-Луга',  
'поселок Ушаки',  
'поселок Федоровское',  
'поселок Фornosово',  
'поселок Форт Красная Горка',  
'поселок Цвелодубово',  
'поселок Цвылево',  
'поселок Шугозеро',

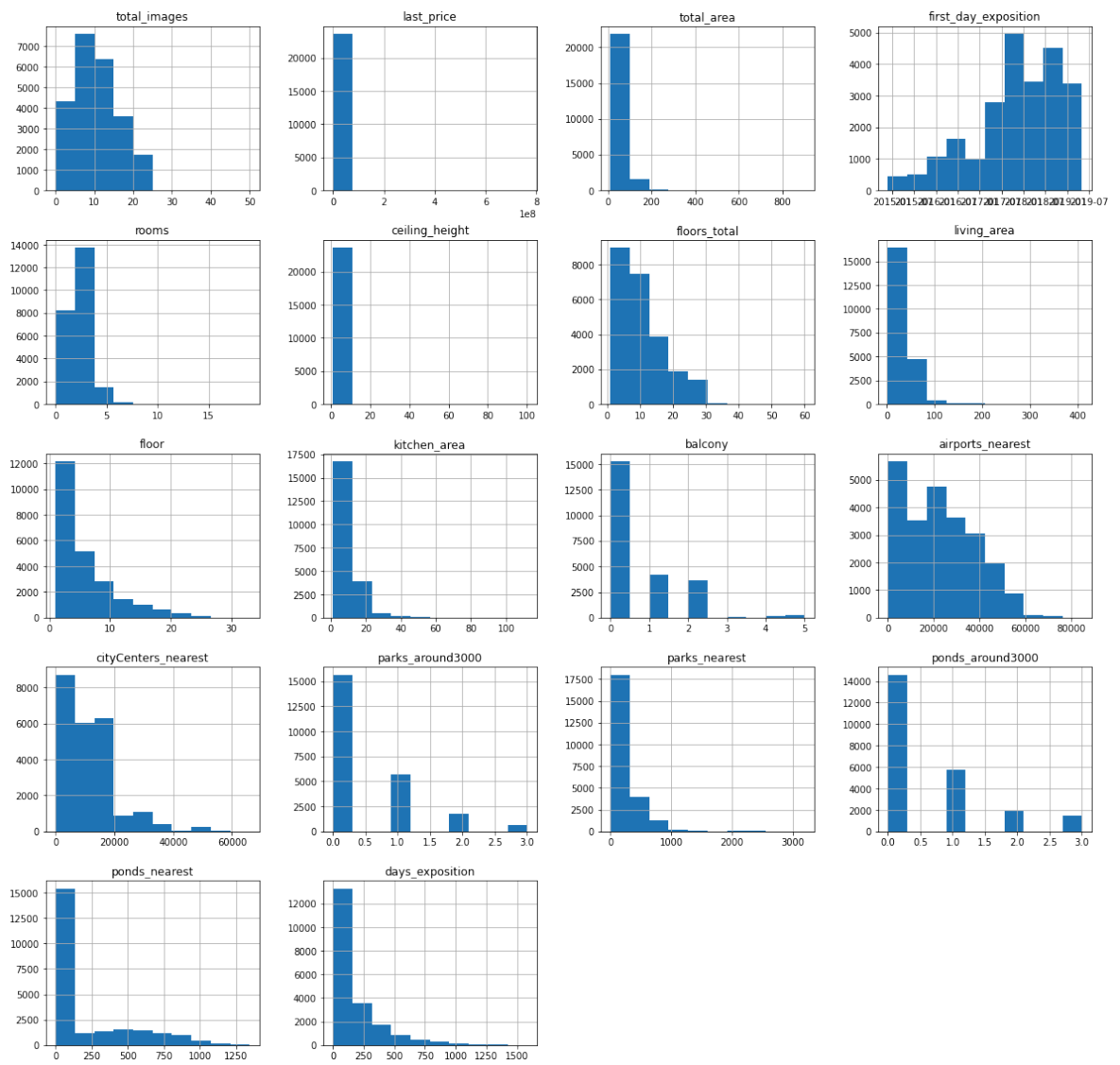
```
'поселок Шушары',  
'поселок Щеглово',  
'поселок Янино-1',  
'поселок имени Морозова',  
'поселок имени Свердлова',  
'садоводческое некоммерческое товарищество Лесная Поляна',  
'садовое товарищество Новая Ропша',  
'садовое товарищество Приладожский',  
'садовое товарищество Рахья',  
'садовое товарищество Садко',  
'село Копорье',  
'село Никольское',  
'село Павлово',  
'село Паша',  
'село Путилово',  
'село Рождествено',  
'село Русско-Высоцкое',  
'село Старая Ладога',  
'село Шум']
```

**Вывод:** Устранены неявные дубликаты '*поселок при железнодорожной станции*', '*городской поселок*', '*коттеджный поселок*', '*поселок станции*', '*поселок городского типа*'. Заменены на '*поселок*'

## Обработка аномалий

In [546...

```
data.hist(figsize=(20,20))  
plt.show()
```

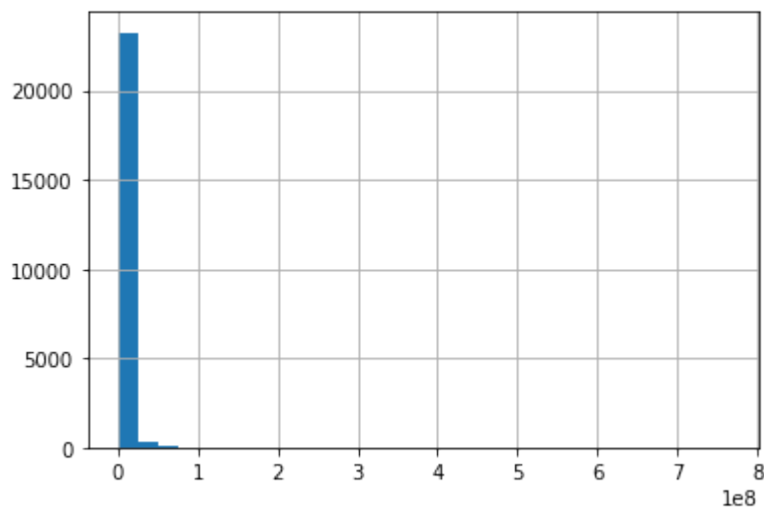


`last_price`

Посмотрим снова на гистограммы, проверим `last_price`

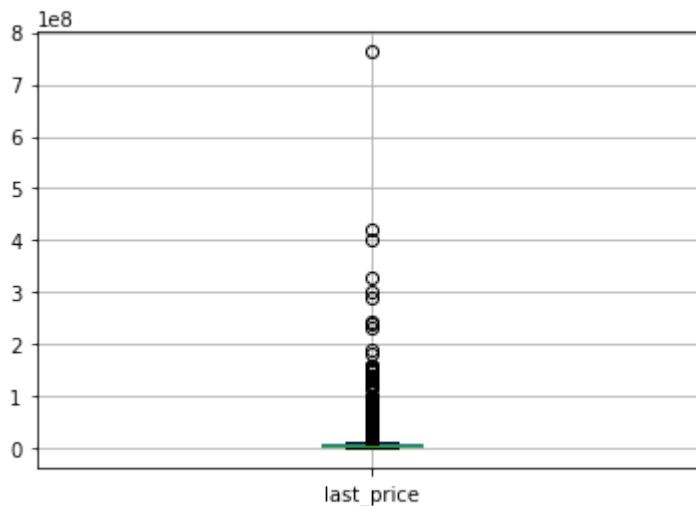
In [547... `data['last_price'].hist(bins=30)`

Out[547... `<AxesSubplot:>`



```
In [548... data.boxplot(column='last_price')
data['last_price'].describe()
```

```
Out[548... count    2.369900e+04
mean      6.541549e+06
std       1.088701e+07
min       1.219000e+04
25%       3.400000e+06
50%       4.650000e+06
75%       6.800000e+06
max       7.630000e+08
Name: last_price, dtype: float64
```



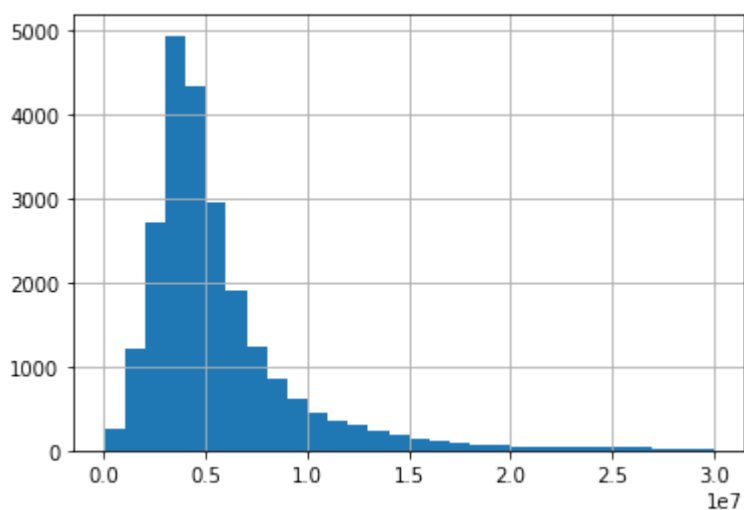
Виден сильный выброс, посмотрим какую долю занимают высокие цены

```
In [549... data_price = data['last_price'] > 2.00000e+07
data_price.mean()
```

```
Out[549... 0.02983248238322292
```

```
In [550... data.query('last_price < 3.00000e+07')['last_price'].hist(bins=30)
```

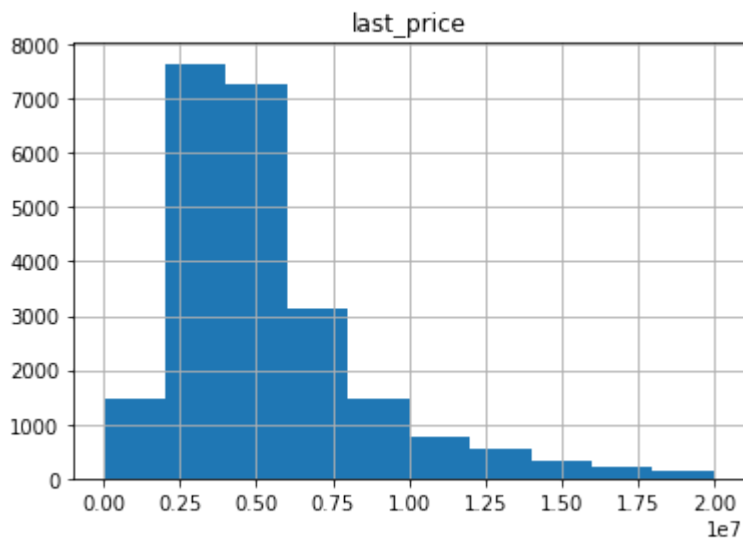
```
Out[550... <AxesSubplot:>
```



```
In [551... good_data = data[data['last_price'] < 2.00000e+07]
good_data.hist(column='last_price')
```



```
Out[551... array([[<AxesSubplot:title={'center':'last_price'}>]], dtype=object)
```



```
In [552... data_price = data['last_price'] > 1.50000e+07  
data_price.mean()
```

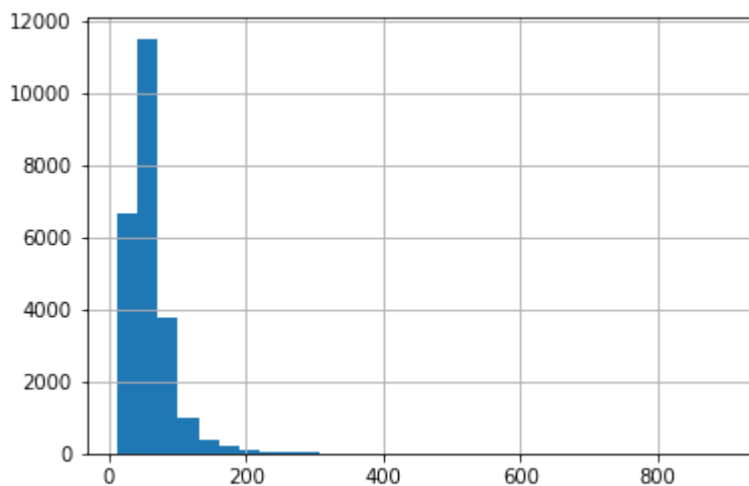
```
Out[552... 0.050803831385290515
```

**Вывод:** last\_price Виден большой хвост в стоимости квартир выше 20 миллионов, что составляет 3% от общей выборки. Это сильно влияет на остальные результаты. Убрали эти квартиры.

total\_area

```
In [553... data['total_area'].hist(bins=30)
```

```
Out[553... <AxesSubplot:>
```



```
In [554... data['total_area'].describe()
```

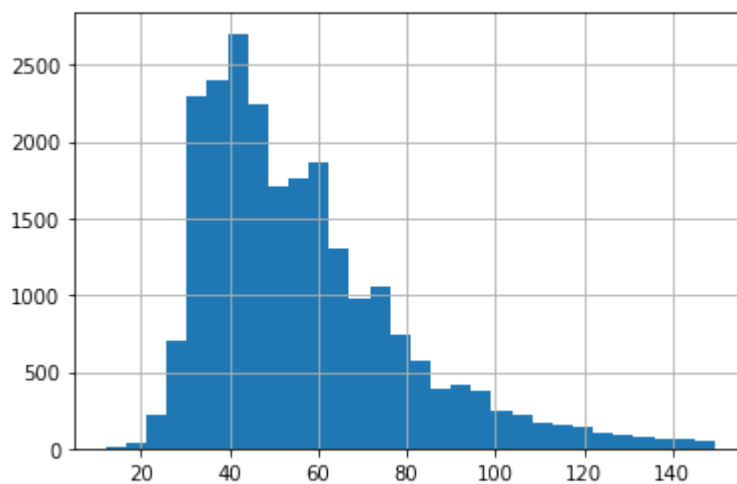
```
Out[554...] count    23699.000000
            mean      60.348651
            std       35.654083
            min       12.000000
            25%       40.000000
            50%       52.000000
            75%       69.900000
            max       900.000000
            Name: total_area, dtype: float64
```

```
In [555...] data_area = data['total_area'] > 150
            data_area.mean()
```

```
Out[555...] 0.02379847250938858
```

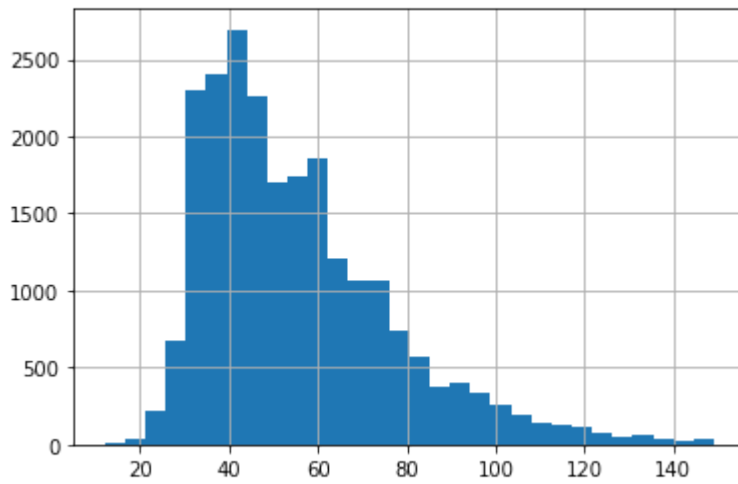
```
In [556...] data.query('total_area < 150')['total_area'].hist(bins=30)
```

```
Out[556...] <AxesSubplot:>
```



```
In [557...] good_data = good_data.query('total_area < 150')
            good_data['total_area'].hist(bins=30)
            good_data['total_area'].describe()
```

```
Out[557...] count    22777.000000
            mean      55.598621
            std       21.637601
            min       12.000000
            25%       39.700000
            50%       50.800000
            75%       66.700000
            max       149.200000
            Name: total_area, dtype: float64
```

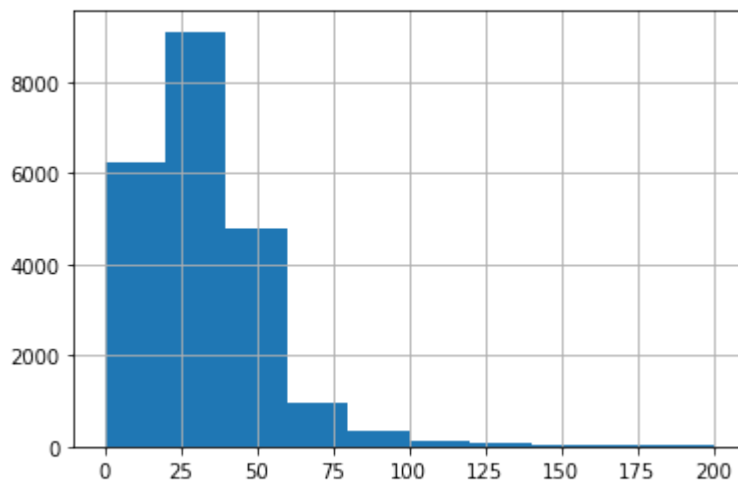


**Вывод:** Удалили квартиры с общей площадью больше 150, сильно влияли на результат при это составляли 2% от данных

living\_area

In [558... data['living\_area'].hist(range=(0,200))

Out[558... <AxesSubplot:>



In [559... data\_area = data['living\_area'] > 80  
data\_area.mean()

Out[559... 0.02894636904510739

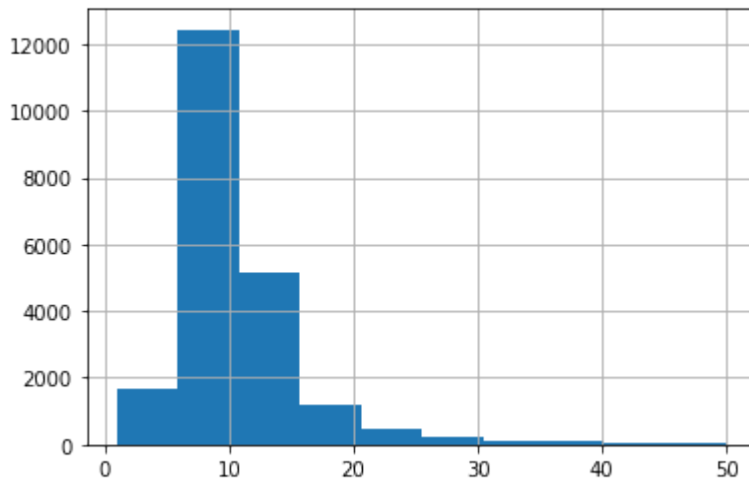
In [560... good\_data = good\_data.query('living\_area < 80')

**Вывод:** в колонке living\_area квартиры с жилой площадью больше 100 м<sup>2</sup> занимают 3%, уберем их из таблицы

kitchen\_area

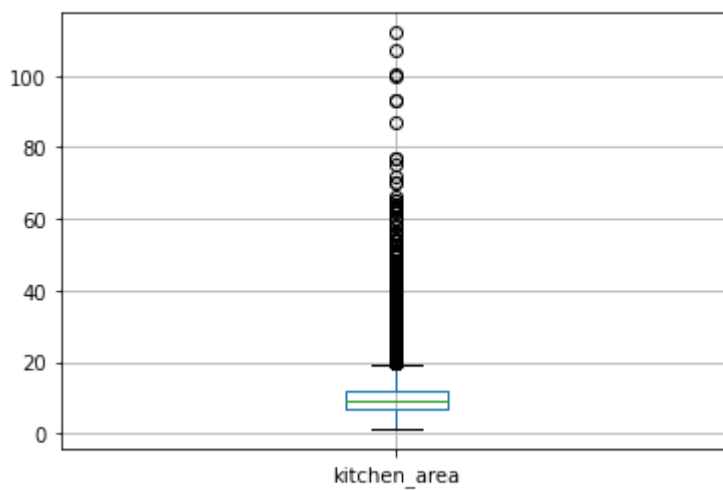
In [561... data['kitchen\_area'].hist(range=(1,50))

Out[561... <AxesSubplot:>



```
In [562...] data.boxplot(column='kitchen_area')
```

```
Out[562...] <AxesSubplot:>
```



```
In [563...] data_k = data['kitchen_area'] > 25
data_k.mean()
```

```
Out[563...] 0.02329212203046542
```

```
In [564...] good_data = good_data.query('kitchen_area < 25')
```

```
In [565...] good_data.shape
```

```
Out[565...] (19692, 22)
```

**Вывод:** в колонке `kitchen_area` квартиры с кухней площадью больше 25 м<sup>2</sup> занимают 2%, уберем их из таблицы

## Промежуточный вывод

В ходе преобработки данных постарались привести к нормальному распределению.

- Заполнили медианным значением в высоту квартир
- Заменим пропущенные значения на 0 в колонках `'cityCenters_nearest'`, `'airports_nearest'`, `'parks_around3000'`, `'parks_nearest'`, `'ponds_around'`
- `locality_name` заполняли на *Неизвестно*

- `first_day_exposition last_price balcony` привели к типу `datetime64`, `int32`, `int32` соответственно
- Убрали аномалии в `total_area`, `last_price`, `kitchen_area`, `living_area` составляющие более 3%

## Добавьте в таблицу новые столбцы

```
In [566... good_data['one_per_metr'] = good_data['last_price'] / good_data['total_area'] #ц
good_data['one_per_metr'] = good_data['one_per_metr'].round(2)
```

```
In [567... good_data['weekday'] = good_data['first_day_exposition'].dt.weekday#день недели
```

```
In [568... good_data['month'] = good_data['first_day_exposition'].dt.month
```

```
In [569... good_data['year'] = good_data['first_day_exposition'].dt.year
```

```
In [570... def floor_type(row):
    if row['floor'] == 1:
        return 'первый'
    elif row['floors_total'] == row['floor']:
        return 'последний'
    else:
        return 'другой'

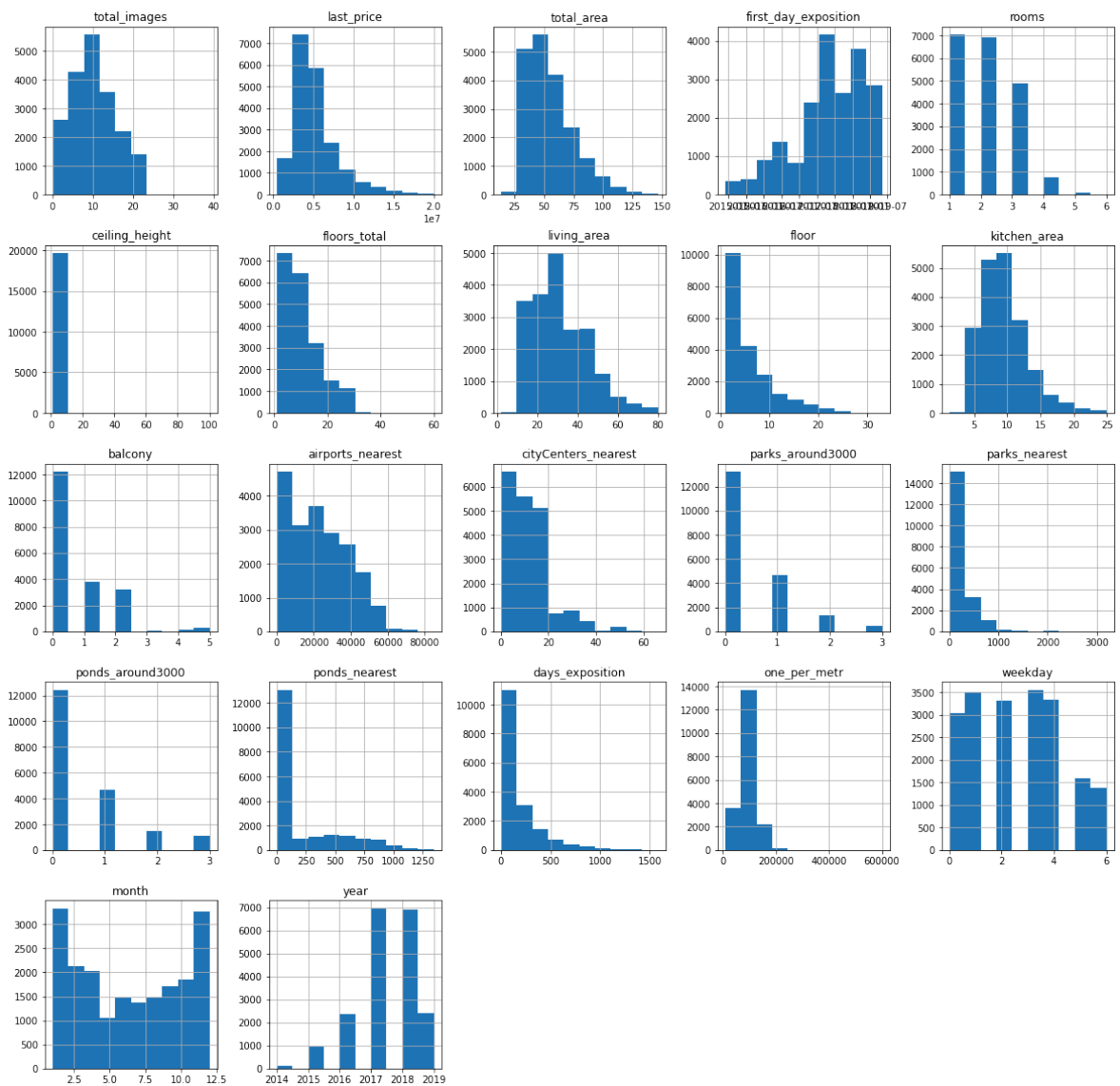
good_data['type_floor'] = good_data.apply(floor_type, axis=1)#тип этажа квартиры
```

```
In [571... good_data['cityCenters_nearest'] = good_data['cityCenters_nearest'] / 1000
good_data['cityCenters_nearest'] = good_data['cityCenters_nearest'].round()
good_data['cityCenters_nearest']
```

```
Out[571... 1      19.0
2      14.0
5       0.0
6      19.0
8       0.0
...
23692   0.0
23693  34.0
23694   4.0
23695   0.0
23698   0.0
Name: cityCenters_nearest, Length: 19692, dtype: float64
```

## Проведите исследовательский анализ данных

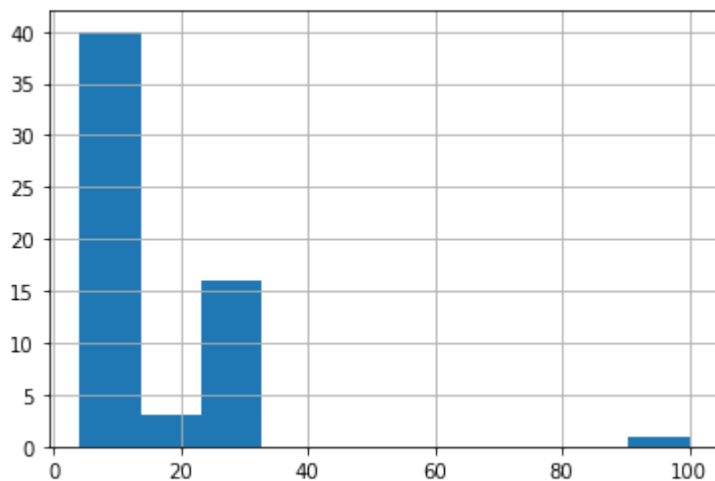
```
In [572... good_data.hist(figsize=(20,20))
plt.show()
```



**Вывод** После проведения предобработки данных, данные стали выглядеть лучше, кроме ceiling\_height. Исследуем колонку.

In [573... `good_data['ceiling_height'].hist(range=(4,100))`

Out[573... `<AxesSubplot:>`



In [574... `good_data = good_data.query('ceiling_height < 100')`

Большое количество данных указаны в другой размерности в колонках  
`ceiling_height`

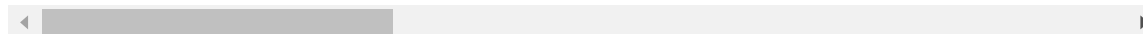
```
In [575... def fix_height(row):
    try:
        if row > 10:
            return row / 10
    except:
        print('smt wrong', row)
    return row

good_data['ceiling_height'] = good_data['ceiling_height'].apply(fix_height)
```

```
In [576... good_data.query('ceiling_height > 5')
```

```
Out[576...      total_images  last_price  total_area  first_day_exposition  rooms  ceiling_height
3474             3  1400000.0        30.0        2018-03-16         1             8.0
5863            20  5650000.0        45.0        2018-08-15         1             8.3
15743           14  5600000.0        67.0        2016-11-11         3             8.0
```

3 rows × 27 columns

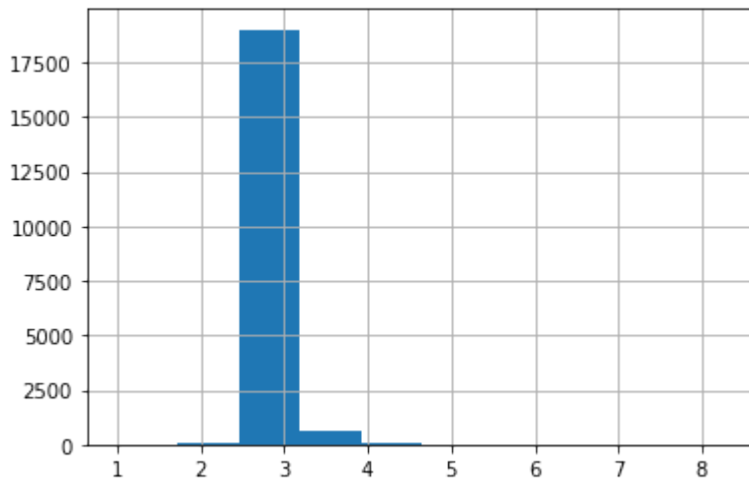


```
In [577... good_data['ceiling_height'].value_counts()
```

```
Out[577... 2.65    7894
2.50    3245
2.60    1522
2.70    1378
2.55     926
...
2.89         1
1.75         1
4.25         1
3.48         1
3.93         1
Name: ceiling_height, Length: 149, dtype: int64
```

```
In [578... good_data['ceiling_height'].hist()
```

```
Out[578... <AxesSubplot:>
```



**Вывод:** В колонке высота потолка оказались по разному заполнены данные, привели к общему виду

In [579...

```
good_data.loc[good_data['parks_nearest']>1000,'parks_nearest'] = 0
```

**Вывод:** Большая часть парков в диапазоне от 1 до 1000 метров от остальных можно избавиться.

## Исследование на сколько быстро продавались квартиры

In [596...

```
good_data['days_exposition'].hist(bins=100)
plt.title('Распределение days_exposition')
plt.show()
```



In [597...

```
good_data['days_exposition'].hist(bins=100,range=(1,400))
plt.title('Распределение days_exposition')
plt.show()
```

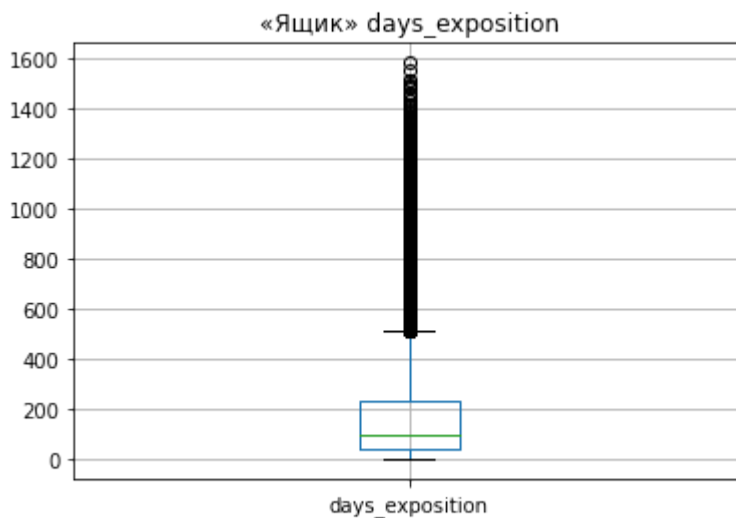




```
In [582... good_data['days_exposition'].describe()
```

```
Out[582... count    17123.000000
mean      179.843953
std       215.652403
min        1.000000
25%       44.000000
50%       98.000000
75%      231.000000
max     1580.000000
Name: days_exposition, dtype: float64
```

```
In [598... good_data.boxplot(column = 'days_exposition')
plt.title('«Ящик» days_exposition')
plt.show()
```



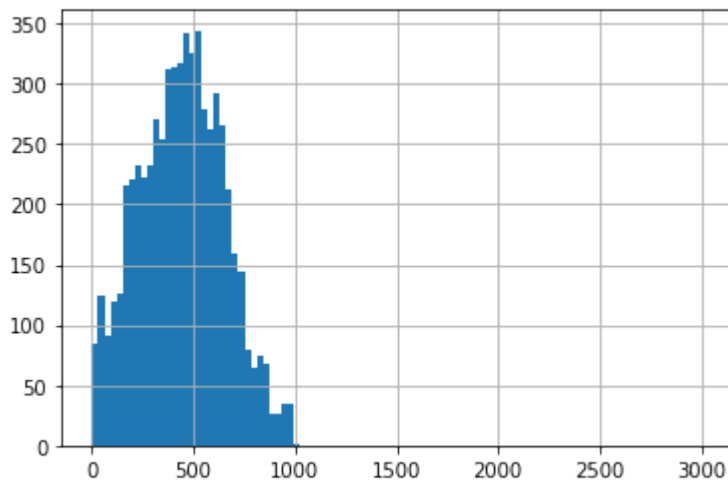
**Вывод:** Продажа квартиры согласуется с распределением Пуассона. В среднем за 200 дней распродаются 75% квартир. Больше 500 дней можно считать необычно долгими, а быстрые продажи занимают 50 дней.

```
In [584... good_data['cityCenters_nearest'].describe()
```

```
Out[584...] count    19691.000000
              mean      11.303946
              std        9.766335
              min         0.000000
              25%        3.000000
              50%       12.000000
              75%       15.000000
              max       66.000000
              Name: cityCenters_nearest, dtype: float64
```

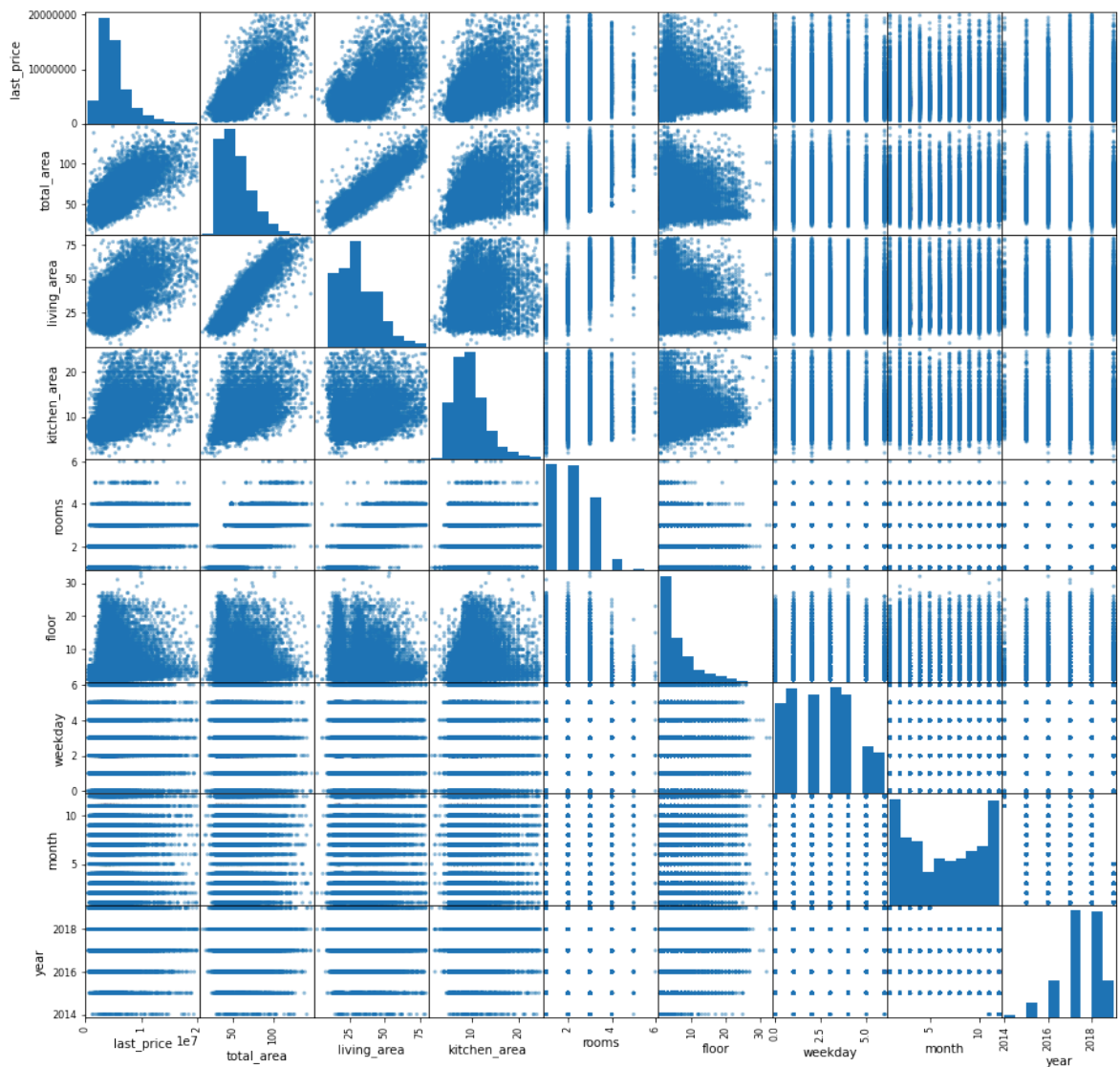
```
In [585...] good_data['parks_nearest'].hist(bins=100,range=(1,3000))
```

```
Out[585...] <AxesSubplot:>
```



## Исследование влияние стоимости на жилье

```
In [593...] column_s = ['last_price','total_area','living_area','kitchen_area','rooms','floc
df_s = good_data[column_s]
pd.plotting.scatter_matrix(df_s,figsize=(15,15))
plt.show()
```



**Вывод:** Что и следовало ожидать цена зависит от общей площади квартиры и жилой площади, коррелирует с площадью кухни и количеством комнат. Декабрские и январские покупки присутствуют больше, чем остальные месяцы. Стоимость жилья равномерно распределяется в зависимости дня недели, месяца, года и не коррелирует с ней. Количество продаж квартир увеличивается от года к году

**Посчитать среднюю цену одного квадратного метра в 10 населённых пунктах с наибольшим числом объявлений»:**

```
In [588... images = good_data.sort_values(by='total_images', ascending=False)
images.head(10)['one_per_metr'].mean()
```

```
Out[588... 106921.861
```

Вывод: Средняя цена одного квадратного метра с наибольшим числом объявлений равна 109987

**Населённые пункты с самой высокой и низкой стоимостью квадратного метра.**

```
In [644... pivot = good_data.pivot_table(index='locality_name', values='one_per_metr', aggf
pivot
```

Out[644...

	count	mean
locality_name		
Бокситогорск	10	16910.915000
Волосово	35	40144.608000
Волхов	91	35120.871319
Всеволожск	350	67196.921600
Выборг	181	58102.715580
...	...	...
село Путилово	2	39921.755000
село Рождествено	2	34553.885000
село Русско-Высоцкое	8	54991.515000
село Старая Ладога	1	36666.670000
село Шум	1	33898.310000

299 rows × 2 columns

In [643...

```
pivot.query('locality_name != "Неизвестно"]').sort_values(by='mean', ascending=False)
```

Out[643...

	count	mean
locality_name		
Санкт-Петербург	12888	107819.176779
Зеленогорск	21	106070.223810
Сестрорецк	155	101841.077032
Пушкин	311	101392.726592
Кудрово	148	99896.067432
деревня Мистолово	9	93294.498889
деревня Кудрово	223	92503.059507
Мурино	28	92200.776071
поселок Парголово	288	90756.002986
поселок Стрельна	42	89122.955952

In [642...

```
pivot.sort_values(by='mean', ascending=True).head(10)
```

Out[642...

	count	mean
locality_name		
деревня Старополье	1	10368.660
деревня Ям-Тесово	1	12328.770
деревня Выскатка	2	12335.480
поселок Совхозный	1	12555.560
поселок Ефимовский	1	12686.570
деревня Малая Романовка	1	12724.550
поселок Тесово-4	1	12931.030
деревня Сижно	1	13709.680
поселок Житково	2	14956.275
поселок Свирь	1	15000.000

**Вывод:** Населённый пункт с самой высокой стоимостью **Санкт-Петербург**.  
Населённый пункт с самой низкой стоимостью **деревня Старополье**.

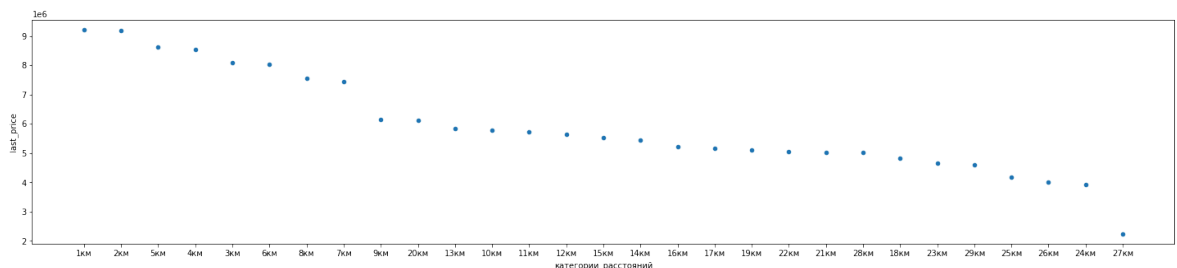
## Стоимость квартир в Санкт-Петербурге на разном удалении от центра

In [591...

```
# Выделите квартиры в Санкт-Петербурге с помощью столбца locality_name и вычислите
spb_data = good_data.copy()
spb_data = spb_data.query('locality_name == "Санкт-Петербург"')
def spb_price_cat(row):
    for i in range(1,30):
        if row['cityCenters_nearest'] <= i:
            return f'{i}км'
spb_data['категории_расстояний'] = spb_data.apply(spb_price_cat,axis=1)
plot_spb = spb_data.groupby('категории_расстояний')['last_price'].agg('mean').re
plot_spb.plot(kind='scatter',x='категории_расстояний',y='last_price',figsize=(25
spb_data['cityCenters_nearest'].corr(spb_data['last_price']))
```

Out[591...

-0.3789849810002749



In [592...

plot\_spb['last\_price'].diff().mean()

Out[592...

-248514.91839963835

**Вывод:** Построена таблица зависимости средней стоимости от центра города. Мы видим отрицательную корреляцию, чем дальше от центра, тем ниже стоимость. В

среднем цена падает на 260 814 рублей. Есть провалы между 8 и 9 км, 24 и 27 километрами, возможно это связано с недостаточной выборкой, но в общем тренд не меняется.

## Вывод

### Предобработка данных:

- Заполнили данные `is_apartment` 88%, скорее всего связано с отсутствием непосредственно апартаментов.
- В колонке `balcony` по той же причине отсутствует 48% данных.
- `airports_nearest`, `cityCenters_nearest`, `ponds_nearest` и остальные похожие колонки заполнили 0 из-за отсутствия данных
- Пропуски `last_price`, `total_area`, `living_area`, `kitchen_area` были убраны часть квартир они составляли менее 3% от общих данных и давали большой выброс в каждой из колонок
- Пропуски `floors_total`, `days_exposition` данные отсутствуют их не трогали
- Есть большое количество отсутствия данных в `ceiling_height` 38% заполнили медианным значением.
- `locality_name` заполнили Неизвестно

### Добавление столбцов

Добавили столбцы для анализа `one_per_metr`, день недели, месяц, год

### Исследование данных

#### Зависимость продаж от времени

В среднем за 200 дней распродаются 75% квартир. Больше 500 дней можно считать необычно долгими, а быстрые продажи занимают 50 дней.

#### Общие выводы между различными колонками

- Цена зависит от общей площади квартиры и жилой площади.
- Коррелирует с площадью кухни и количеством комнат.
- Декабрские и январские покупки присутствуют больше, чем остальные месяцы.
- Стоимость жилья равномерно распределяется в зависимости дня недели, месяца, года и не коррелирует с ней.
- Количество продаж квартир увеличивается от года к году.
- Средняя цена одного квадратного метра с наибольшим числом объявлений равна `109 987`
- Населённый пункт с самой высокой стоимостью Санкт-Петербург. Населённый пункт с самой низкой стоимостью деревня Старополье.

Есть ли зависимость квартир в Санкт-Петербурге от расстояния

Да гипотеза подтвердилась, в среднем цена падает на 260 814 рублей рублей за каждый километр от центра

## Общий вывод:

В ходе исследования проверяли рыночную стоимость квартир и какие параметры на нее влияют. Наибольший вклад вносят общая и жилая площадь. Площадь кухни и количество также влияют, но меньше. Цена не зависит от месяца и года покупки квартир. Также цена зависит от центра города, например в городе Санкт-Петербург цена в среднем падает 260т рублей.

## Чек-лист готовности проекта

Поставьте 'x' в выполненных пунктах. Далее нажмите Shift+Enter.

- ☒ Файл с данными открыт.
- ☒ Файл с данными изучен: выведены первые строки, использован метод `info()` , построены гистограммы.
- ☒ Найдены пропущенные значения.
- ☒ Пропущенные значения заполнены там, где это возможно.
- ☒ Объяснено, какие пропущенные значения обнаружены.
- ☒ В каждом столбце установлен корректный тип данных.
- ☒ Объяснено, в каких столбцах изменён тип данных и почему.
- ☐ Устранены неясные дубликаты в названиях населённых пунктов.
- ☐ Обработаны редкие и выбивающиеся значения (аномалии).
- ☐ В таблицу добавлены новые параметры: – цена одного квадратного метра; – день публикации объявления (0 - понедельник, 1 - вторник и т. д.); – месяц публикации объявления; – год публикации объявления; – тип этажа квартиры (значения — «первый», «последний», «другой»); – расстояние до центра города в километрах.
- ☐ Изучены и описаны параметры: - общая площадь; - жилая площадь; - площадь кухни; - цена объекта; - количество комнат; - высота потолков; - тип этажа квартиры («первый», «последний», «другой»); - общее количество этажей в доме; - расстояние до центра города в метрах; - расстояние до ближайшего парка.
- ☐ Выполнено задание «Изучите, как быстро продавались квартиры (столбец `days_exposition`)»:
  - построена гистограмма;
  - рассчитаны среднее и медиана;
  - описано, сколько обычно занимает продажа и указано, какие продажи можно считать быстрыми, а какие — необычно долгими.
- ☐ Выполнено задание «Определите факторы, которые больше всего влияют на общую (полную) стоимость объекта». Построены графики, которые показывают зависимость цены от параметров: - общая площадь; - жилая площадь; - площадь кухни; - количество комнат; - тип этажа, на котором расположена

квартира (первый, последний, другой); - дата размещения (день недели, месяц, год).

- ☐ Выполнено задание «Посчитайте среднюю цену одного квадратного метра в 10 населённых пунктах с наибольшим числом объявлений»:
  - выделены населённые пункты с самой высокой и низкой стоимостью квадратного метра.
- ☐ Выполнено задание «Выделите квартиры в Санкт-Петербурге с помощью столбца `locality_name` и вычислите их среднюю стоимость на разном удалении от центра»:
  - учтён каждый километр расстояния, известны средние цены квартир в одном километре от центра, в двух и так далее;
  - описано, как стоимость объекта зависит от расстояния до центра города;
  - построен график изменения средней цены для каждого километра от центра Петербурга.
- ☐ На каждом этапе сделаны промежуточные выводы.
- ☐ В конце проекта сделан общий вывод.