Содержание

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. 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.

Изучение данных из файла

In [1]:
```
# импортируем библиотеку pandas
import pandas as pd
import matplotlib.pyplot as plt
from plotly import graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"
import numpy as np
import seaborn as sns
sns.set(style="whitegrid")
colors =["#ef476f","#ffd166","#06d6a0","#118ab2","#073b4c"]
sns.set_palette(sns.color_palette(colors))
import re
from scipy import stats as st
import math as mth
import re
from scipy import stats as st
import math as mth
pd.set_option('display.float_format', '{:,.2f}'.format)
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# скачиваем нужный файл
data = pd.read_csv('/datasets/real_estate_data.csv',sep='\t')
data.head()
```

Out[2]:

total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	fl

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	fl
0	20	13,000,000.00	108.00	2019-03-07T00:00:00	3	2.70	16.00	51.00	
1	7	3,350,000.00	40.40	2018-12-04T00:00:00	1	NaN	11.00	18.60	
2	10	5,196,000.00	56.00	2015-08-20T00:00:00	2	NaN	5.00	34.30	
3	0	64,900,000.00	159.00	2015-07-24T00:00:00	3	NaN	14.00	NaN	
4	2	10,000,000.00	100.00	2018-06-19T00:00:00	2	3.03	14.00	32.00	

5 rows × 22 columns

In [3]:
```
# методом info() получаем таблицу.
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
```

- Имеем столбцы:
  - 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 ($м^2$)
  - 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 — площадь квартиры в квадратных метрах (м²)
- total_images — число фотографий квартиры в объявлении

In [4]:
```
data.describe()
```

Out[4]:

	total_images	last_price	total_area	rooms	ceiling_height	floors_total	living_area	floor	kitch
count	23,699.00	23,699.00	23,699.00	23,699.00	14,504.00	23,613.00	21,796.00	23,699.00	2
mean	9.86	6,541,548.77	60.35	2.07	2.77	10.67	34.46	5.89	
std	5.68	10,887,013.27	35.65	1.08	1.26	6.60	22.03	4.89	
min	0.00	12,190.00	12.00	0.00	1.00	1.00	2.00	1.00	
25%	6.00	3,400,000.00	40.00	1.00	2.52	5.00	18.60	2.00	
50%	9.00	4,650,000.00	52.00	2.00	2.65	9.00	30.00	4.00	
75%	14.00	6,800,000.00	69.90	3.00	2.80	16.00	42.30	8.00	
max	50.00	763,000,000.00	900.00	19.00	100.00	60.00	409.70	33.00	

Вывод

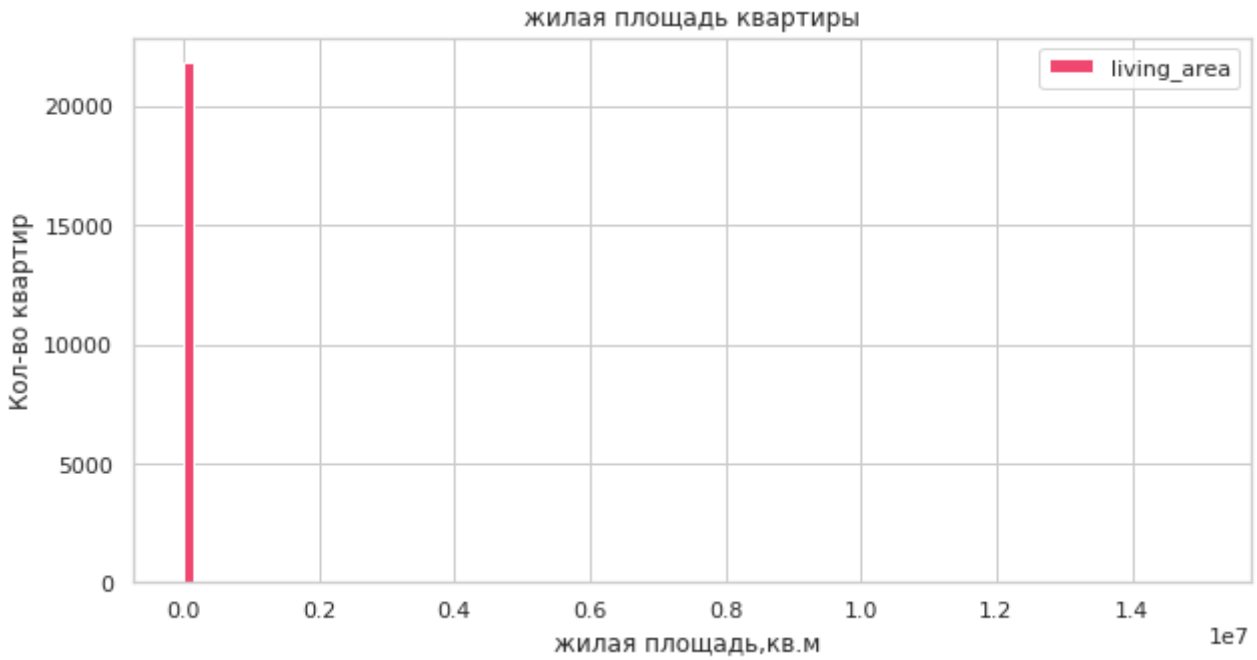
- в таблице есть пропущенные значения:
  - наличие балкона указано только в половине случаев,
  - общее кол-во этажей,
  - высота потолков,
  - жилая площадь,
  - аппартаменты,
  - площадь кухни,
  - название населенного пункта,
  - сколько дней размещено объявление и др.
- Так же некоторые типы данных неверные. Например, общая этажность, кол-во ближайших водоемов и парков, а так же наличие балкона имеют тип вещественных чисел.
- Имеем средние значения:
  - число фотографий квартиры в объявлении - 9
  - цена на момент снятия с публикации - 4650000.00
  - площадь квартиры в квадратных метрах (м²)- 52
  - число комнат - 2
  - высота потолков (м)- 2.65
  - всего этажей в доме - 9
  - жилая площадь в квадратных метрах (м²) - 30
  - этаж - 4

- площадь кухни в квадратных метрах (м^2) - 9.1
- число балконов - 1
- расстояние до ближайшего аэропорта в метрах (м)- 26726.00
- расстояние до центра города (м) - 13098.50
- число парков в радиусе 3 км - 0
- расстояние до ближайшего парка (м) - 455.00
- число водоёмов в радиусе 3 км - 1
- расстояние до ближайшего водоёма (м) - 502.00
- сколько дней было размещено объявление (от публикации до снятия) - 95.00

Заметили следующее:

- 1.first_day_exposition - object - должен быть тип datetime,
- 2.ceiling_height - есть пропущенные значения ,
- 3.floors_total - должен быть тип int, т.к. количество этажей - целое значение + пропущенные значения,
- 4.is_apartment - должен быть тип bool + пропущенные значения,
- 5.living_area - пропущенные значения,
- 6.kitchen_area - пропущенные значения,
- 7.balcony - пропущенные значения,
- 8.locality_name - пропущенные значения,
- 9.airports_nearest - пропущенные значения,
- 10.cityCenters_nearest - пропущенные значения,
- 11.parks_around3000 - тип должен быть int + пропущенные значения,
- 12.parks_nearest - пропущенные значения,
- 13.ponds_around3000 - тип должен быть int + пропущенные значения,
- 14.ponds_nearest - пропущенные значения,
- 15.days_exposition - тип должен быть int + пропущенные значения,

In [5]:
```
data.plot(y = 'living_area', kind = 'hist', bins = 100, grid=True, range = (0,15000000), figsiz
plt.title('жилая площадь квартиры')
plt.xlabel('жилая площадь,кв.м ')
plt.ylabel('Кол-во квартир')
plt.show()
```



Предобработка данных

In [6]:
```
# пропуски в balcony
data['balcony'].isna().sum()
```

Out[6]:
```
11519
```

пропущенные значения в столбце " балкон" скорее всего означают отсутствие балкона. поэтому можно заполнить их 0.

In [7]:
```
#заменим пропуски в balcony на 0 и изменим тип данных на int
data['balcony'].value_counts()
data['balcony'] = data['balcony'].fillna(0)
data['balcony'] = data['balcony'].astype('int')
print('пропуски после', data['balcony'].isna().sum())
```

```
пропуски после 0
```

In [8]:
```
# определяем пропущенные значения по locality_name
print('пропуски ',data['locality_name'].isna().sum())
```

```
пропуски  49
```

In [9]:
```
# доля пропусков в наименовании населенного пункта
print('доля пропусков',49/23650)
```

```
доля пропусков 0.002071881606765328
```

пропусков очень мало(49),меньше 1%, можно пренебречь

In [10]:
```
# удаляем строки с пропущенными значениями название населённого пункта методом dropna()
data = data.dropna(subset=['locality_name'])
print('пропуски после:',data['locality_name'].isna().sum()) #проверяем
```

```
пропуски после: 0
```

In [11]:
```
#посмотрим значения в столбцах для выявления нестандартных значений на первый взгляд и ошибок,
#все эти столбца оставляем как есть
data['total_images'].value_counts()
data['ceiling_height'].value_counts()
data['floor'].value_counts()
data['is_apartment'].value_counts()
data['total_area'].value_counts()
data['rooms'].value_counts()
data['airports_nearest'].value_counts()
data['cityCenters_nearest'].value_counts()
data['parks_around3000'].value_counts()
data['parks_nearest'].value_counts()
data['ponds_around3000'].value_counts()
data['locality_name'].value_counts().head()
```

Out[11]:
```
Санкт-Петербург    15721
посёлок Мурино       522
посёлок Шушары       440
Всеволожск           398
Пушкин               369
Name: locality_name, dtype: int64
```

In [12]:
```
data['total_images'].value_counts().head()
```

Out[12]:
```
10    1793
9     1723
20    1690
```

```
8     1581
7     1515
Name: total_images, dtype: int64
```

In [13]:
```
#столбец апартаменты видимо появился не вместе с началом ведения записей, все пропущенные значе
#апартаменты появились относительно недавно
data['is_apartment'] = data['is_apartment'].fillna(False)
data['is_apartment'].head()
```

Out[13]:
```
0    False
1    False
2    False
3    False
4    False
Name: is_apartment, dtype: bool
```

In [14]:
```
#изменим цену на тип int для удобства просмотра
data['last_price'] = data['last_price'].astype('int')
data['last_price'].head()
```

Out[14]:
```
0    13000000
1     3350000
2     5196000
3    64900000
4    10000000
Name: last_price, dtype: int64
```

In [15]:
```
#переведем столбец с датой в формат даты без времени, т.к. время не указано
data['first_day_exposition'] = pd.to_datetime(data['first_day_exposition'], format = '%Y-%m-%d'
data['first_day_exposition'].head()
```

Out[15]:
```
0   2019-03-07
1   2018-12-04
2   2015-08-20
3   2015-07-24
4   2018-06-19
Name: first_day_exposition, dtype: datetime64[ns]
```

In [16]:
```
data['days_exposition'].head()
```

Out[16]:
```
0      NaN
1    81.00
2   558.00
3   424.00
4   121.00
Name: days_exposition, dtype: float64
```

In [17]:
```
# проверяем пропуски floors_total
print('пропуски до:',data['floors_total'].isna().sum())
```

```
пропуски до: 85
```

Доля пропусков минимальна, можно удалить пропущенные строки

In [18]:
```
# удаляем строки с пропущенными значениями общей этажности методом dropna()
data = data.dropna(subset=['floors_total'])
print('пропуски после:',data['floors_total'].isna().sum()) #проверяем
```

```
пропуски после: 0
```

Определил и изучил пропущенные значения. Там, где это необходимо, заменил типы данных на необходимые для удобной работы

Расчёты и добавление результатов в таблицу

рассчитываем стоимость квадратного метра

In [19]:
```
#рассчитываем стоимость квадратного метра
data['last_price_area'] = data['last_price'] / data['total_area']
#для удобства просмотра приведем к типу int
data['last_price_area'] = data['last_price_area'].astype('int')
print(data['last_price_area'].head(5))
```

```
0    120370
1     82920
2     92785
3    408176
4    100000
Name: last_price_area, dtype: int64
```

находим и создаем столбцы дня недели,месяца и года

In [20]:
```
# находим и создаем столбец дня недели
data['weekday_exposition'] = data['first_day_exposition'].dt.weekday
data['weekday_exposition'].head(5)
```

Out[20]:
```
0    3
1    1
2    3
3    4
4    1
Name: weekday_exposition, dtype: int64
```

In [21]:
```
#создаем столбец месяца
data['month_exposition'] = data['first_day_exposition'].dt.month
data['month_exposition'].head(5)
```

Out[21]:
```
0     3
1    12
2     8
3     7
4     6
Name: month_exposition, dtype: int64
```

In [22]:
```
#создаем столбец года
data['year_exposition'] = data['first_day_exposition'].dt.year
data['year_exposition'].head(5)
```

Out[22]:
```
0    2019
1    2018
2    2015
3    2015
4    2018
Name: year_exposition, dtype: int64
```

рассчитываем отношение жилой площади к общей и площади кухни к общей площади

In [23]:
```
# рассчитываем отношение жилой площади к общей и площади кухни к общей площади
data['living_area_total'] = data['living_area'] / data['total_area']
print(data['living_area_total'].head(5))
```

```
0   0.47
1   0.46
2   0.61
3    NaN
4   0.32
Name: living_area_total, dtype: float64
```

In [24]:
```
# рассчитываем отношение  площади кухни к общей площади
data['kitchen_area_total'] = data['kitchen_area'] / data['total_area']
print(data['kitchen_area_total'].head(5))
```

```
0   0.23
1   0.27
2   0.15
3    NaN
4   0.41
Name: kitchen_area_total, dtype: float64
```

In [25]:
```
#заменим пропуски в 'kitchen_area_total' медианой
print('пропуски до:',data['kitchen_area_total'].isna().sum())
data['kitchen_area_total'] = data['kitchen_area_total'].median()
print('пропуски после:',data['kitchen_area_total'].isna().sum())
```

```
пропуски до: 2222
пропуски после: 0
```

In [26]:
```
#этаж квартиры; варианты — первый, последний, другой;
##напишем функцию категоризации по этажам, используя метод apply применимо к каждой строке дата

def floor_category(row):
    if row['floor'] == 1: return 'первый'
    if row['floor'] == row['floors_total']: return 'последний'
    return 'другой'
data['floor_category'] = data.apply(floor_category, axis = 1)
data['floor_category'].head(5)
```

Out[26]:
```
0    другой
1    первый
2    другой
3    другой
4    другой
Name: floor_category, dtype: object
```

In [27]:
```
# добавляем в таблицу новые столбцы: стоимость кв.м, отношение жилой площади к общей, отношение

data[["last_price_area", 'living_area_total', 'kitchen_area_total', 'first_day_exposition', 'we
data.head()
```

Out[27]:

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	floor
0	20	13000000	108.00	2019-03-07	3	2.70	16.00	51.00	8
1	7	3350000	40.40	2018-12-04	1	NaN	11.00	18.60	1
2	10	5196000	56.00	2015-08-20	2	NaN	5.00	34.30	4
3	0	64900000	159.00	2015-07-24	3	NaN	14.00	NaN	9
4	2	10000000	100.00	2018-06-19	2	3.03	14.00	32.00	13

5 rows × 29 columns

Посчитал и добавил в таблицу цену квадратного метра жилья,соотношение жилой и общей площади, а также отношение площади кухни к общей. Вывел из даты дни недели, месяцы и года размещения объявлений, добавил категории квартир по этажам.

Исследовательский анализ данных

строим гистограмму жилой площади

In [28]:
```
#изучим жилую площадь
# строим гистограмму жилой площади
data.plot(y = 'living_area', kind = 'hist', bins = 100, grid=True, figsize = (10,5), range = (0
plt.title('жилая площадь')
plt.xlabel('жилая площадь,кв.м')
```

```
plt.ylabel('Кол-во квартир')
plt.show()
```

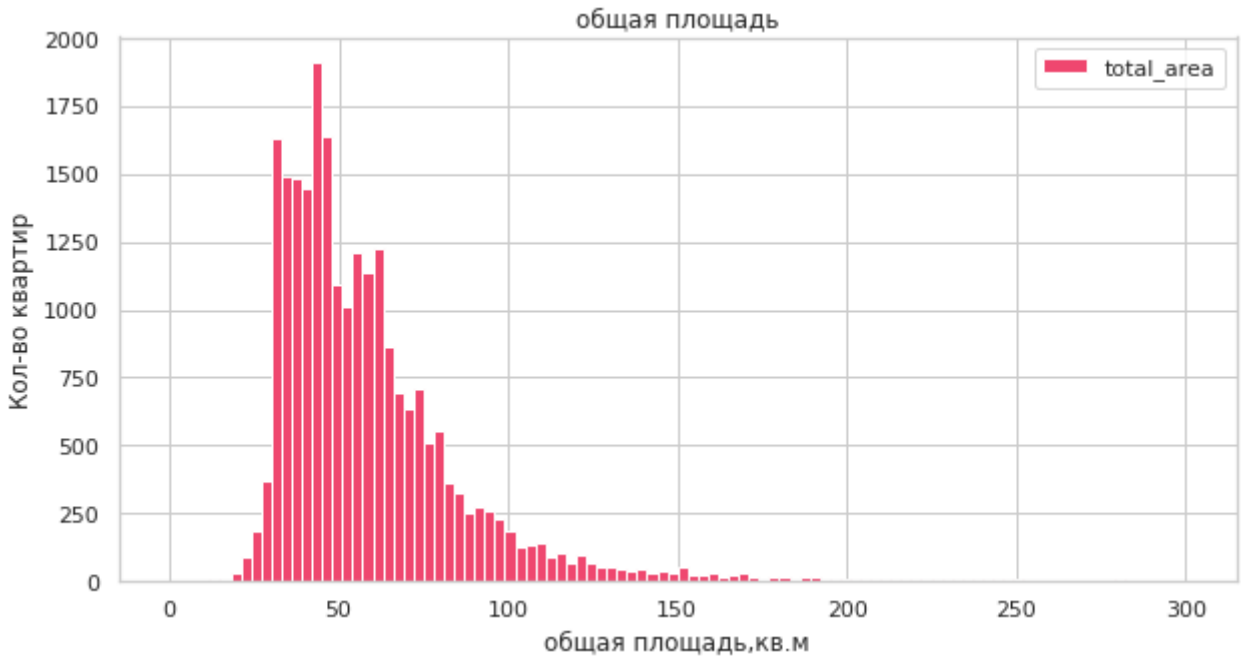
In [29]:
```
print(data['living_area'].describe())
```

```
count   21,700.00
mean        34.45
std         22.05
min          2.00
25%         18.60
50%         30.00
75%         42.30
max        409.70
Name: living_area, dtype: float64
```

Наибольшее предложение приходится на квартиры жилой площадью менее 43 кв.м

строим гистограмму общей площади

In [30]:
```
#изучим общую площадь
#строим гистограмму общей площади
data.plot(y = 'total_area', kind = 'hist', bins = 100, grid=True, figsize = (10,5), range = (0,
plt.title('общая площадь')
plt.xlabel('общая площадь,кв.м')
plt.ylabel('Кол-во квартир')
plt.show()
```



In [31]:
```
print(data['total_area'].describe())
```

```
count   23,565.00
mean        60.32
std         35.66
min         12.00
25%         40.00
50%         52.00
75%         69.70
max        900.00
Name: total_area, dtype: float64
```

подавляющее количество квартир - до 70 кв.м

строим гистограмму кол-ва комнат

In [32]:
```
#строим гистограмму кол-ва комнат
data.plot(y = 'rooms', kind = 'hist', bins = 20, grid=True, figsize = (10,5),range = (0,8))
plt.title('кол-во комнат')
plt.xlabel('кол-во комнат')
plt.ylabel('Кол-во квартир')
plt.show()
```

In [33]:
```
print(data['rooms'].describe())
```

```
count   23,565.00
mean         2.07
std          1.08
min          0.00
25%          1.00
50%          2.00
75%          3.00
max         19.00
Name: rooms, dtype: float64
```

Наибольшее предложение -1, 2-х и 3-х комнатных квартир. И практически нет комнат и квартир 6-комнатных и более.

строим гистограмму высоты потолков

In [34]:

```
#строим гистограмму высоты потолков
data.plot(y = 'ceiling_height', kind = 'hist', bins = 20, range = (2.5,3), grid=True, figsize =
plt.title('высота потолков')
plt.xlabel('высота потолков,м')
plt.ylabel('Кол-во квартир')
plt.show()
```

In [35]:

```
print(data['ceiling_height'].describe())
```

```
count   14,481.00
mean         2.77
std          1.26
min          1.00
25%          2.51
50%          2.65
75%          2.80
max        100.00
Name: ceiling_height, dtype: float64
```

Квартир с высотой потолков 2.65 наибольшее кол-во

строим гистограмму цены на момент снятия публикации

In [36]:

```
#строим гистограмму цены на момент снятия публикации
data.plot(y = 'last_price', kind = 'hist', bins = 100, grid=True, range = (0,15000000), figsize
plt.title('цена на момент снятия публикации')
plt.xlabel('цена на момент снятия публикации')
plt.ylabel('Кол-во квартир')
plt.show()
```

In [37]:
```
print(data['last_price'].describe())
```

```
count        23,565.00
mean      6,540,058.26
std      10,910,934.72
min          12,190.00
25%       3,400,000.00
50%       4,646,000.00
75%       6,790,000.00
max     763,000,000.00
Name: last_price, dtype: float64
```

В большинстве случаев стоимость кв.м от 3000000 до 6000000

строим гистограмму цены кв.м на момент снятия публикации

In [38]:
```
#стоимость кв.м
print(data['last_price_area'].describe())
```

```
count      23,565.00
mean       99,405.39
std        50,389.44
min           111.00
25%        76,566.00
50%        95,000.00
75%       114,213.00
max     1,907,500.00
Name: last_price_area, dtype: float64
```

In [39]:
```
#строим гистограмму цены  кв.м на момент снятия публикации
data.plot(y = 'last_price_area', kind = 'hist', bins = 100, grid=True, range = (0,190000), figs
plt.title('цена кв.м на момент снятия публикации')
plt.xlabel('цена кв.м на момент снятия публикации')
plt.ylabel('Кол-во квартир')
plt.show()
```

В большинстве случаев стоимость кв.м от 76000 до 114000

строим гистограмму времени продажи квартиры

In [40]:
```
#строим гистограмму времени продажи квартиры
data.plot(y = 'days_exposition', kind = 'hist', bins = 100, grid = True, range = (1,200), figsi
plt.title('время продажи квартиры')
plt.xlabel('время продажи квартиры')
plt.ylabel('Кол-во квартир')
plt.show()
```

In [41]:
```
# исследуем время продажи квартиры методом describe()
pd.set_option('display.float_format', '{:,.2f}'.format)
data['days_exposition'].describe()
```

Out[41]:
```
count   20,394.00
mean       180.74
std        219.73
min          1.00
25%         45.00
50%         95.00
75%        231.00
max      1,580.00
Name: days_exposition, dtype: float64
```

Среднее время продажи квартир 71 день. Но были итакие, которые продавались более 298 дней.
Продажу можно считать быстрой, если продалась за 50 дней,более 200 дней - долго.

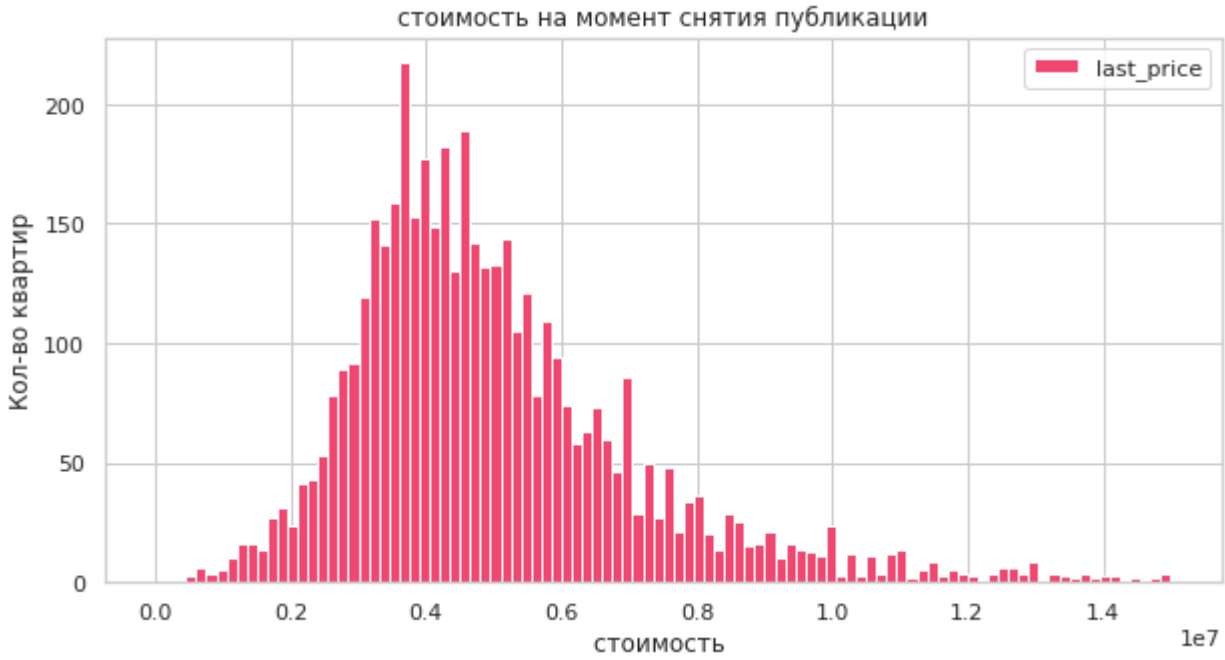
In [42]:
```
# убираем выбивающиеся значения методом query()
data = data.query(' days_exposition < 200')
```

Исследуем цену продажи квартир

In [43]:
```
#исследуем цену продажи квартир методом describe()
data['last_price'].describe()
```

Out[43]:
```
count        14,547.00
mean      5,777,485.99
std       9,955,875.56
min          12,190.00
25%       3,350,000.00
50%       4,400,000.00
75%       6,200,000.00
max     763,000,000.00
Name: last_price, dtype: float64
```

In [95]:
```
#строим гистограмму цены на момент снятия публикации
data.plot(y = 'last_price', kind = 'hist', bins = 100, grid=True, range = (0,15000000), figsize
plt.title('стоимость на момент снятия публикации')
plt.xlabel('стоимость ')
plt.ylabel('Кол-во квартир')
plt.show()
```



Исследуем жилую площадь квартиры

In [45]:
```
#исследуем жилую площадь квартиры методом describe()
data['living_area'].describe()
```

Out[45]:
```
count   13,178.00
mean        31.87
std         19.42
min          2.00
25%         18.00
50%         29.00
75%         39.58
max        409.70
Name: living_area, dtype: float64
```

In [46]:
```
#строим гистограмму жилой площади квартиры
```

```
data.plot(y = 'living_area', kind = 'hist', bins = 100, grid=True, range = (0,150), figsize = (
plt.title('жилая площадь квартиры')
plt.xlabel('жилая площадь,кв.м ')
plt.ylabel('Кол-во квартир')
plt.show()
```

имеются выбивающиеся значения площадь жилая min 2 кв.м и max 409.7

In [47]:
```
# убираем выбивающиеся значения методом query()
data = data.query('living_area < 78')
```

In [48]:
```
#исследуем общую площадь квартиры методом describe()
data['total_area'].describe()
```

Out[48]:
```
count   12,895.00
mean        53.10
std         20.22
min         13.00
25%         38.20
50%         48.00
75%         63.00
max        413.50
Name: total_area, dtype: float64
```

имеются выбивающиеся значения площадь общая min 12 кв.м и max 413

In [49]:
```
# убираем выбивающиеся значения методом query()
data = data.query('total_area < 100')
```

In [50]:
```
data['total_area'].describe()
```

Out[50]:
```
count   12,521.00
mean        51.19
std         16.70
min         13.00
25%         38.00
50%         47.00
75%         61.80
max         99.88
Name: total_area, dtype: float64
```

In [51]:
```
#исследуем кол-во комнат  методом describe()
data['rooms'].describe()
```

Out[51]:
```
count   12,521.00
mean         1.84
std          0.85
min          0.00
25%          1.00
50%          2.00
75%          2.00
max          6.00
Name: rooms, dtype: float64
```

имеются выбивающиеся значения : кол-во комнат 19

In [52]:
```
# убираем выбивающиеся значения методом query()
data = data.query('rooms < 6')
```

In [53]:
```
data['rooms'].describe()
```

Out[53]:
```
count   12,520.00
mean         1.84
std          0.85
min          0.00
25%          1.00
50%          2.00
75%          2.00
max          5.00
Name: rooms, dtype: float64
```

In [54]:
```
#исследуем высоту потолков квартиры методом describe()
data['ceiling_height'].describe()
```

Out[54]:
```
count   7,411.00
mean        2.73
std         1.46
min         1.00
25%         2.50
50%         2.60
75%         2.75
max       100.00
Name: ceiling_height, dtype: float64
```

имеются выбивающиеся значения - высота потолков 1 м и 1оо м

In [55]:
```
# убираем выбивающиеся значения методом query()
data = data.query('2.52 < ceiling_height < 3')
data['ceiling_height'].describe()
```

Out[55]:
```
count   4,473.00
mean        2.67
std         0.09
min         2.53
25%         2.60
50%         2.68
75%         2.75
max         2.98
Name: ceiling_height, dtype: float64
```

In [56]:
```
#проверим
data.head()
```

Out[56]:

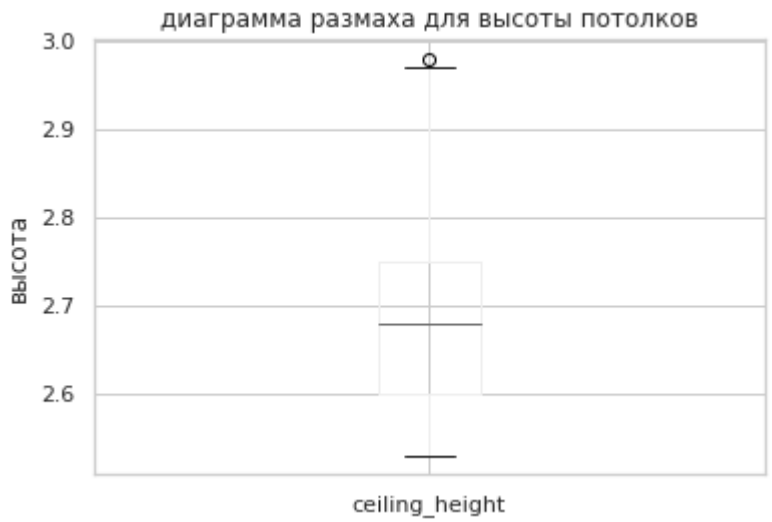
	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	floo
10	5	5050000	39.60	2017-11-16	1	2.67	12.00	20.30	3
20	12	6120000	80.00	2017-09-28	3	2.70	27.00	48.00	1
22	20	5000000	58.00	2017-04-24	2	2.75	25.00	30.00	1

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	floo
27	20	7100000	70.00	2017-05-12	3	2.60	17.00	49.00	1
28	8	4170000	44.00	2017-12-13	1	2.90	6.00	20.80	

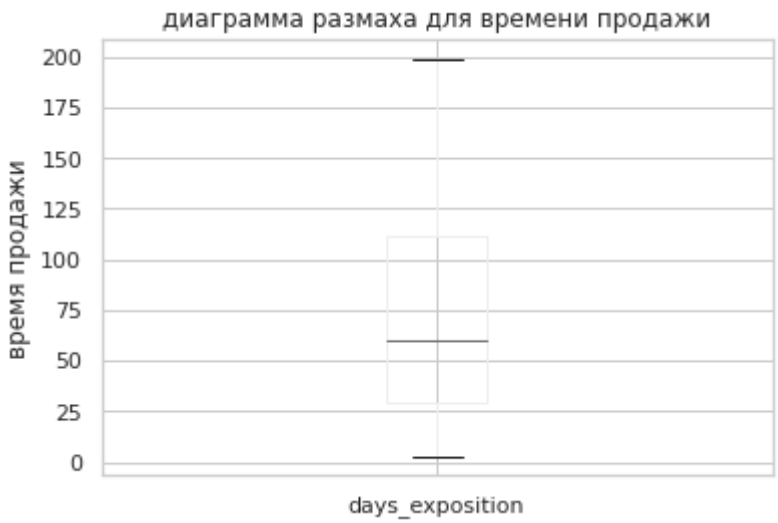
5 rows × 29 columns

стандартные отклонения минимальны в столбцах " кол-во комнат", "высота потолков" и "цена". в столбце "время продажи" - максимальная 219

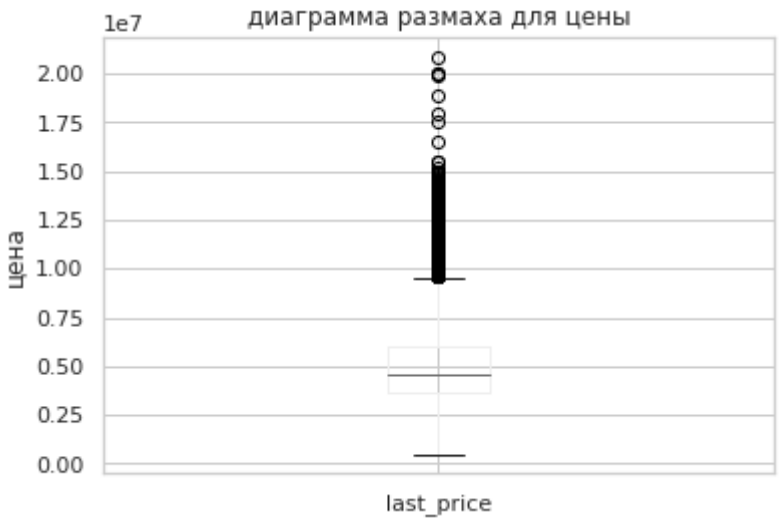
In [57]:
```
# строим диаграмму размаха для высоты потолков
data.boxplot('ceiling_height')
plt.title('диаграмма размаха для высоты потолков')
plt.ylabel('высота')
plt.show()
```



In [58]:
```
# строим диаграмму размаха для времени продажи
data.boxplot('days_exposition')
plt.title('диаграмма размаха для времени продажи')
plt.ylabel('время продажи')
plt.show()
```

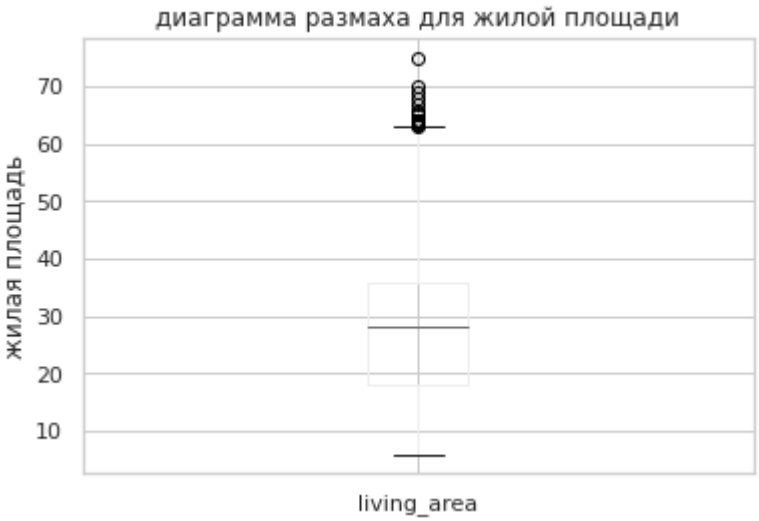


In [59]:
```
# строим диаграмму размаха для цены
data.boxplot('last_price')
plt.title('диаграмма размаха для цены')
plt.ylabel('цена')
plt.show()
```



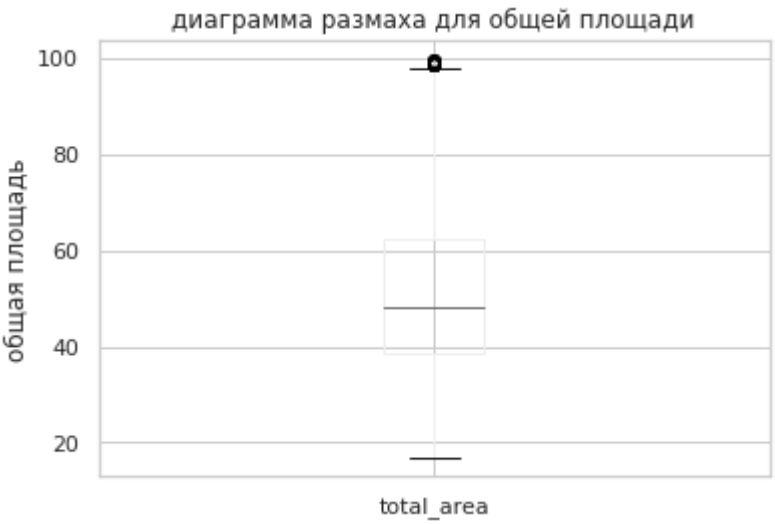
In [60]:

```
# строим диаграмму размаха для жилой площади
data.boxplot('living_area')
plt.title('диаграмма размаха для жилой площади')
plt.ylabel('жилая площадь')
plt.show()
```



In [61]:

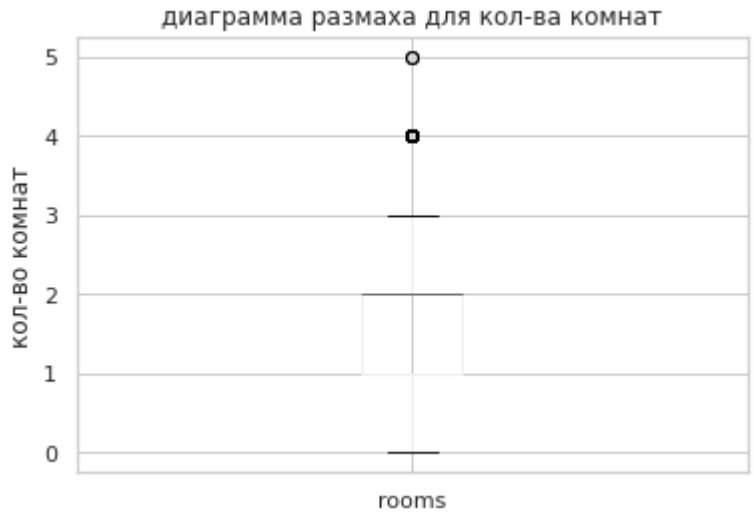
```
# строим диаграмму размаха для общей площади
data.boxplot('total_area')
plt.title('диаграмма размаха для общей площади')
plt.ylabel('общая площадь')
plt.show()
```



In [62]:

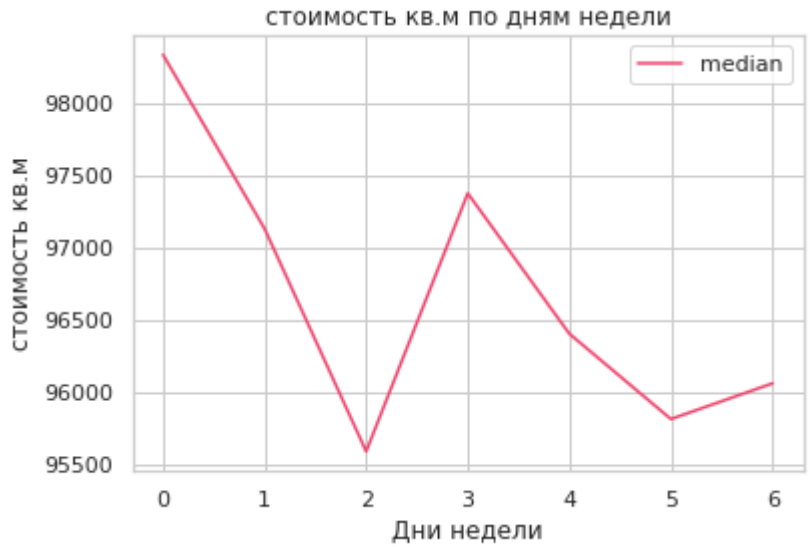
```
# строим диаграмму размаха для кол-ва комнат
```

```
data.boxplot('rooms')
plt.title('диаграмма размаха для кол-ва комнат')
plt.ylabel('кол-во комнат')
plt.show()
```



стоимость кв.м по дням недели

In [63]:
```
pivot_table_weekday_exposition = data.pivot_table(index = 'weekday_exposition', values = 'last_
pivot_table_weekday_exposition.columns = ['mean', 'count', 'median']
pivot_table_weekday_exposition.plot(y = 'median')
plt.title('стоимость кв.м по дням недели')
plt.xlabel('Дни недели')
plt.ylabel('стоимость кв.м')
plt.show()
```



In [64]:
```
pivot_table_weekday_exposition.sort_values('median', ascending = False)
```

Out[64]:

weekday_exposition	mean	count	median
0	97,828.46	678	98,333.00
3	98,664.98	856	97,374.00
1	97,700.12	834	97,128.50
4	99,021.65	718	96,402.50
6	97,327.30	319	96,059.00
5	94,725.97	339	95,812.00

	mean	count	median
weekday_exposition			
2	96,210.47	729	95,588.00

стоимость кв.м зависит от дня размещения объявления не существенно. в субботу минимальна

стоимость кв.м по месяцам

In [65]:
```
pivot_table_month_exposition = data.pivot_table(index = 'month_exposition', values = 'last_pric
pivot_table_month_exposition.columns = ['mean', 'count', 'median']
pivot_table_month_exposition.plot(y = 'median')
plt.title('стоимость кв.м  по месяцам')
plt.xlabel('Месяцы')
plt.ylabel('стоимость кв.м')
plt.show()
```

In [66]:
```
pivot_table_month_exposition.sort_values('median', ascending = False)
```

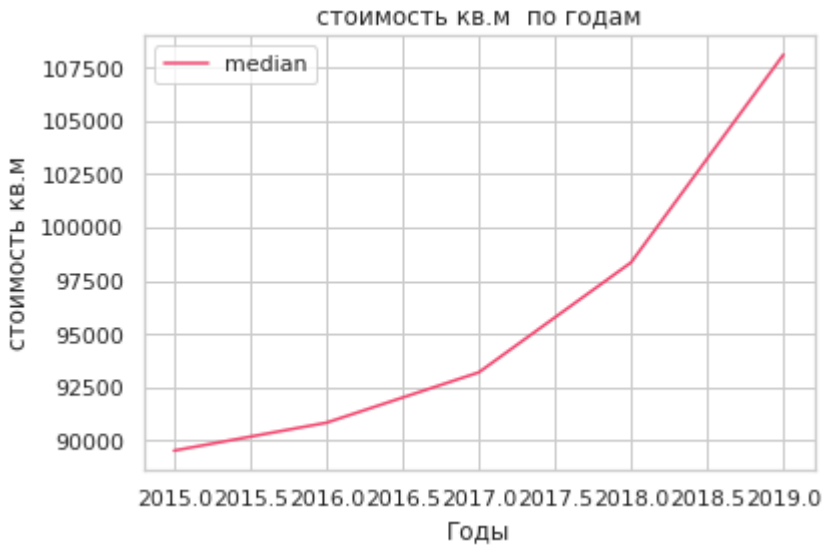
Out[66]:

	mean	count	median
month_exposition			
9	100,027.02	443	98,557.00
3	97,673.97	434	98,401.00
1	97,914.95	264	98,175.50
11	98,277.51	512	97,649.50
12	97,236.80	264	97,128.00
10	97,728.51	482	97,075.00
2	98,969.34	532	96,796.50
8	98,471.59	394	96,762.00
7	95,964.24	363	96,692.00
6	94,348.25	284	95,101.50
4	97,256.36	282	95,027.50
5	93,652.20	219	93,333.00

стоимость кв.м зависит от месяца размещения объявления существенно. есть минимум в июне. макс в апреле Зависимость существенная

стоимость кв.м по годам

In [67]:
```
pivot_table_year_exposition = data.pivot_table(index = 'year_exposition', values = 'last_price_
pivot_table_year_exposition.columns = ['mean', 'count', 'median']
pivot_table_year_exposition.plot(y = 'median')
plt.title('стоимость кв.м  по годам')
plt.xlabel('Годы')
plt.ylabel('стоимость кв.м')
plt.show()
```



In [68]:
```
pivot_table_year_exposition.sort_values('median', ascending = False)
```

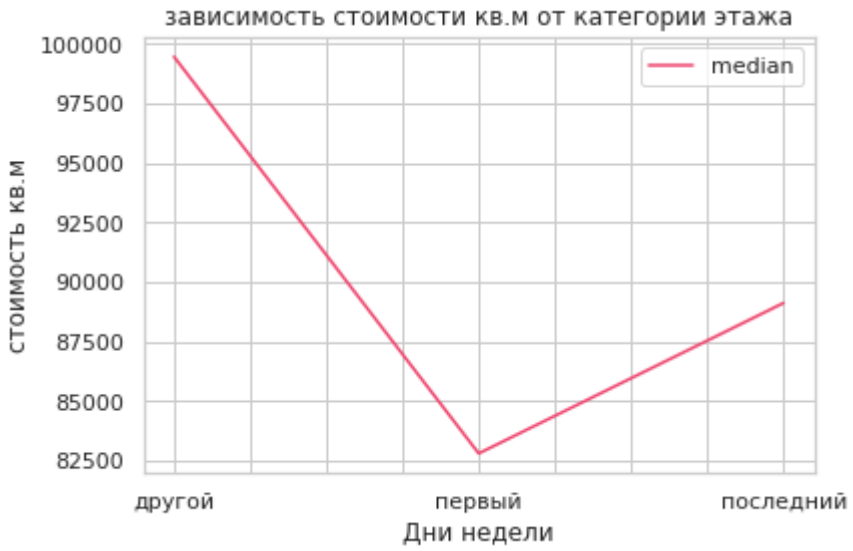
Out[68]:

year_exposition	mean	count	median
2019	108,605.44	417	108110
2018	98,439.50	2223	98360
2017	94,455.40	1536	93198
2016	92,472.13	295	90837
2015	89,528.00	2	89528

стоимость кв.м в течении рассматриваемого периода значительно росла, за исключением периода с 2015 по 2017г Зависимость существенная

зависимость стоимости кв.м от категории этажа

In [69]:
```
# зависимость стоимости кв.м от категории этажа
pivot_table_year_exposition = data.pivot_table(index = 'floor_category', values = 'last_price_a
pivot_table_year_exposition.columns = ['mean', 'count', 'median']
pivot_table_year_exposition.plot(y = 'median')
plt.title('зависимость стоимости кв.м от категории этажа')
plt.xlabel('Дни недели')
plt.ylabel('стоимость кв.м')
plt.show()
```



In [70]:
```
pivot_table_year_exposition.sort_values('median', ascending = False)
```

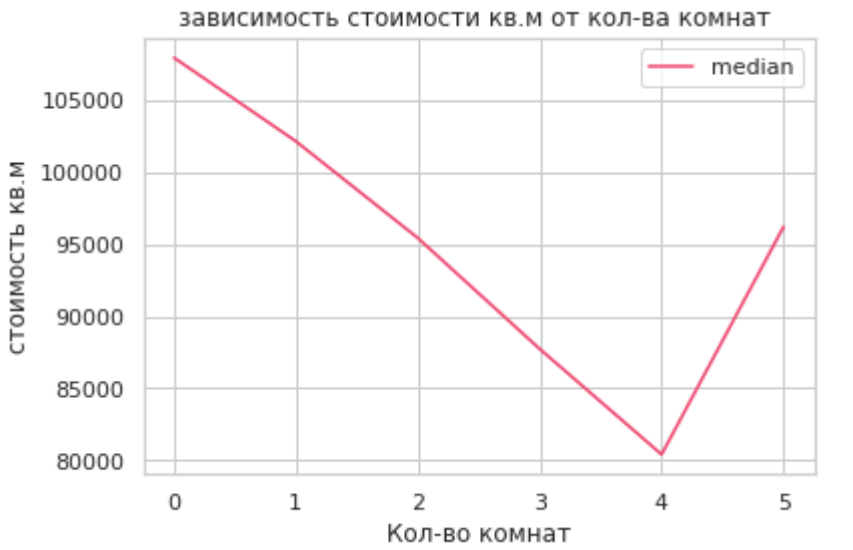
Out[70]:

floor_category	mean	count	median
другой	100,678.22	3577	99455
последний	89,469.67	465	89108
первый	81,048.52	431	82788

минимальная стоимость кв.м на первых этажах .

зависимость стоимости кв.м от кол-ва комнат

In [71]:
```
# зависимость стоимости кв.м от кол-ва комнат
pivot_table_year_exposition = data.pivot_table(index = 'rooms', values = 'last_price_area', agg
pivot_table_year_exposition.columns = ['mean', 'count', 'median']
pivot_table_year_exposition.plot(y = 'median')
plt.title('зависимость стоимости кв.м от кол-ва комнат ')
plt.xlabel('Кол-во комнат')
plt.ylabel('стоимость кв.м')
plt.show()
```



In [72]:
```
pivot_table_year_exposition.sort_values('median', ascending = False)
```

Out[72]:

	mean	count	median

rooms	mean	count	median

rooms			
0	107,604.53	47	107971
1	102,662.17	1909	102162
5	96,218.00	2	96218
2	96,109.95	1576	95444
3	89,993.81	852	87738
4	83,739.94	87	80405

с увеличением кол-ва комнат, стоимость кв.м снижается.После 4-х комнат, стоимость растет

Зависимость стоимости квартир от расстояния до центра города

In [73]:
```
data.plot(x='cityCenters_nearest', y='last_price_area', style='o', ylim=(0, 250000), grid=True,
plt.title('Зависимость стоимости квартир от расстояния до центра города ')
plt.xlabel('Расстояния ,м')
plt.ylabel('стоимость,руб')
plt.show()
```

максимальная стоимость у квартир, расстояние от которых до центра города не более 15 км

Зависимость стоимости квартир по годам продажи

In [74]:
```
data.plot(x='first_day_exposition', y='last_price_area', style='o', ylim=(0, 250000), grid=True
plt.title('Зависимость стоимости квартир по годам продажи ')
plt.xlabel('Годы')
plt.ylabel('Стоимость,руб')
plt.show()
```

после 2017 года стоимость квартир подросла и увеличились продажи. В январе 2017 и июне 2018 был спад продаж и снижение стоимости.

Зависимость стоимости кв.м от жилой площади квартиры

In [75]:
```
data.plot(x='living_area', y='last_price_area', style='o', ylim=(0, 250000), grid=True, figsize
plt.title('Зависимость стоимости кв.м от жилой площади квартиры' )
plt.xlabel('Площадь,кв.м')
plt.ylabel('Стоимость,руб')
plt.show()
```

считаем коэффициент корреляции

In [76]:
```
#считаем коэффициент корреляции
data['living_area'].corr(data['last_price_area'])
```

Out[76]: -0.1292579683208693

стоимость кв.м с увеличением жилой площади снижается незначительно.

Зависимость стоимости кв.м от общей площади квартиры

In [77]:
```
data.plot(x='total_area', y='last_price_area', style='o', ylim=(0, 250000), grid=True, figsize=
plt.title('Зависимость стоимости кв.м от общей площади квартиры' )
plt.xlabel('Площадь,кв.м')
plt.ylabel('Стоимость,руб')
plt.show()
```

In [78]:
```
#считаем коэффициент корреляции
data['total_area'].corr(data['last_price_area'])
```

Out[78]: -0.04275190509036562

стоимость квадратного метра с увеличением общей площади не меняется.

Выберем 10 населённых пунктов с наибольшим числом объявлений

- Выберем 10 населённых пунктов с наибольшим числом объявлений.
- Посчитаем среднюю цену квадратного метра в этих населённых пунктах.
- Выделим населённые пункты с самой высокой и низкой стоимостью жилья.

In [79]:
```
#считаем количество объявлений по городам
id_name = data.pivot_table(index='locality_name', values='last_price', aggfunc=['count'])
id_name.columns = ['count']
id_name.head()
```

Out[79]:

locality_name	count
Волосово	3
Волхов	11
Всеволожск	106
Выборг	14
Гатчина	32

In [80]:
```
#создаем список из верхних топ-10 городов
top_cities = id_name.sort_values(by='count', ascending=False).head(10)
top_cities.head()
```

Out[80]:

locality_name	count
Санкт-Петербург	3083
посёлок Мурино	129
посёлок Шушары	108

locality_name	count
Всеволожск	106
посёлок Парголово	79

In [81]:
```
locality_pivot_table = data.pivot_table(index = 'locality_name', values = 'last_price_area', ag
locality_pivot_table.columns = ['count', 'mean']
locality_pivot_table = locality_pivot_table.sort_values('count', ascending = False).head(10)
locality_pivot_table
#самая высокая стоимость
locality_pivot_table[locality_pivot_table['mean']==locality_pivot_table['mean'].max()]
```

Out[81]:

locality_name	count	mean
Санкт-Петербург	3083	107,825.95

самая высокая средняя стоимость ожидаемо в Санкт-Петербурге-108600

In [82]:
```
#самая низкая стоимость
locality_pivot_table[locality_pivot_table['mean']==locality_pivot_table['mean'].min()]
```

Out[82]:

locality_name	count	mean
Всеволожск	106	67,355.59

самая низкая стоимость во Всеволжске - 66795 руб

In [83]:
```
#Изучим предложения квартир
data['cityCenters_nearest_km'] = data['cityCenters_nearest']/1000
data['cityCenters_nearest_km'] = data['cityCenters_nearest_km'].fillna(999999)
data['cityCenters_nearest_km'] = data['cityCenters_nearest_km'].astype('int')
pivot_table_km = data.query('locality_name == "Санкт-Петербург" and cityCenters_nearest_km !=99
pivot_table_km.head(10)
```

Out[83]:

cityCenters_nearest_km	last_price_area
0	111,488.40
1	130,256.88
2	116,701.25
3	131,418.07
4	124,772.41
5	135,512.68
6	116,058.13
7	127,179.08
8	120,681.86
9	112,592.57

In [84]:

```
pivot_table_km.plot()
plt.title('Зависимость стоимости кв.м от расстояния до центра города ')
plt.xlabel('Расстояния ,км')
plt.ylabel('стоимость,руб')
plt.show()
```

In [85]:

```
data.plot(x='cityCenters_nearest', y='last_price_area', style='o', ylim=(0, 250000), grid=True,
plt.title('Зависимость стоимости кв.м от расстояния до центра города ')
plt.xlabel('Расстояния ,м')
plt.ylabel('стоимость,руб')
plt.show()
```

Вывод: судя по последним графикам, центром считать будем радиус в 19 километров

In [86]:

```
#выделим квартиры в центре, беря за радиус 19 км
center_spb_data = data.query('cityCenters_nearest_km <= 19 and locality_name == "Санкт-Петербур
center_spb_data.head()
```

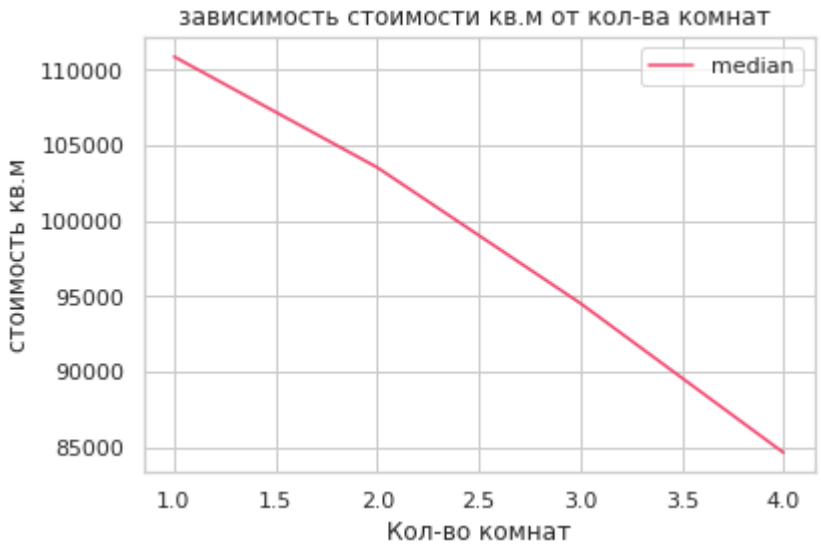
Out[86]:

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	floo
10	5	5050000	39.60	2017-11-16	1	2.67	12.00	20.30	3
27	20	7100000	70.00	2017-05-12	3	2.60	17.00	49.00	1
31	8	7200000	67.90	2017-10-26	2	2.80	16.00	38.10	4
34	3	3290000	33.00	2018-02-04	1	2.55	16.00	14.00	3
39	15	5200000	54.40	2018-11-29	2	2.75	9.00	29.70	2

5 rows × 30 columns

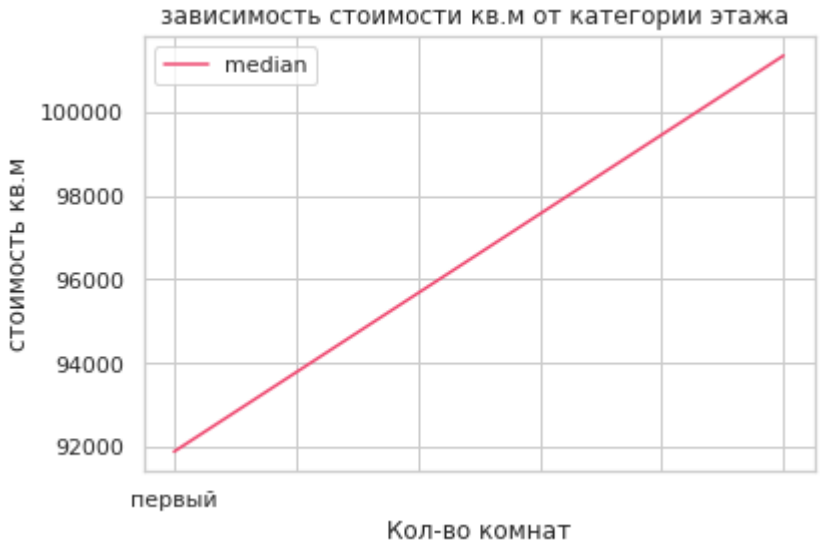
Зависимость стоимости от числа комнат

In [87]:
```
#зависимость стоимости от числа комнат
center_spb_rooms = center_spb_data.pivot_table(index = 'rooms', values = 'last_price_area', agg
center_spb_rooms.columns = ['mean', 'count', 'median']
center_spb_rooms.query('count > 50').plot(y = 'median')

center_spb_rooms.query('count > 50').sort_values('median', ascending = False)

center_spb_data['rooms'].corr(center_spb_data['last_price_area'])
plt.title('зависимость стоимости кв.м от кол-ва комнат ')
plt.xlabel('Кол-во комнат')
plt.ylabel('стоимость кв.м')
plt.show()
```



с увеличением числа комнат, стоимость кв.м падает. Так же как и в общей выборке.

In [88]:
```
#зависимость стоимости от категории этажа
center_spb_floor_category = center_spb_data.query('floor_category != "другой"').pivot_table(ind
center_spb_floor_category.columns = ['mean', 'count', 'median']
center_spb_floor_category.plot(y = 'median')
center_spb_floor_category
plt.title('зависимость стоимости кв.м от категории этажа ')
plt.xlabel('Кол-во комнат')
plt.ylabel('стоимость кв.м')
plt.show()
```

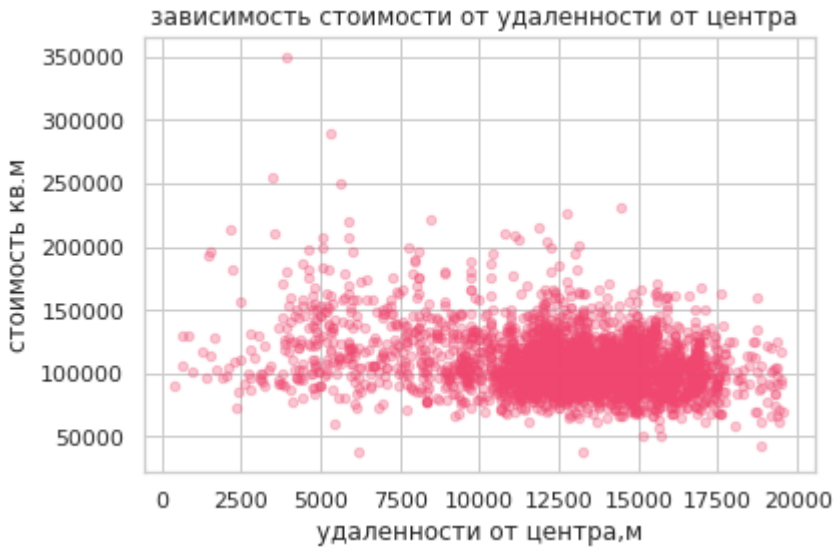


на первых этажах центре СПБ стоимость минимальна. Так же как и в общей выборке.

зависимость стоимости от удаленности от центра

In [89]:
```
#зависимость стоимости от удаленности от центра
center_spb_data.plot(kind = 'scatter', y = 'last_price_area', x = 'cityCenters_nearest', alpha
plt.title('зависимость стоимости от удаленности от центра ')
plt.xlabel('удаленности от центра,м')
plt.ylabel('стоимость кв.м')
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value
-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color
* keyword-argument or provide a 2-D array with a single row if you intend to specify the same R
GB or RGBA value for all points.
```



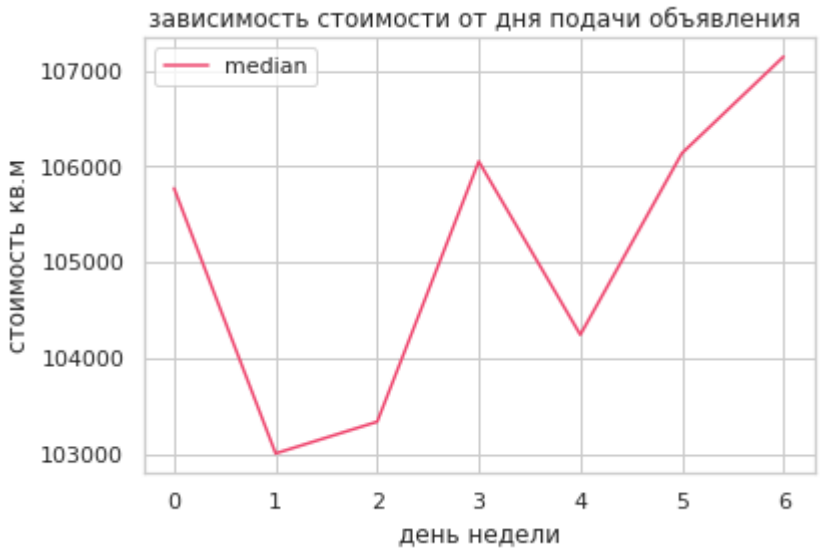
In [90]:
```
center_spb_data['cityCenters_nearest'].corr(center_spb_data['last_price_area'])
```

Out[90]:
```
-0.28181632931094364
```

коэффициент корреляции низкий и отрицательный. Зависимости четкой нет, т.к уже делали выборку по центру города.

зависимость стоимости от дня подачи объявления

In [91]:
```
#зависимость стоимости от дня подачи объявления
center_spb_weekday_exposition = center_spb_data.pivot_table(index = 'weekday_exposition', value
center_spb_weekday_exposition.columns = ['mean', 'count', 'median']
center_spb_weekday_exposition.plot(y = 'median')
plt.title('зависимость стоимости от дня подачи объявления ')
plt.xlabel('день недели')
plt.ylabel('стоимость кв.м')
plt.show()
center_spb_weekday_exposition.sort_values('median', ascending = False)
```



Out[91]:

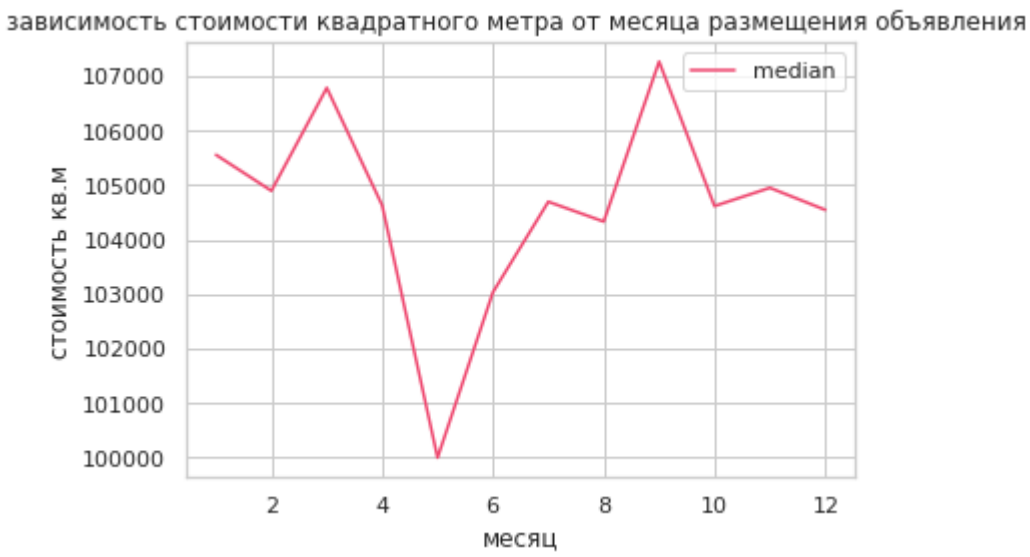
weekday_exposition	mean	count	median
6	110,548.03	201	107,142.00
5	106,113.07	214	106,134.50
3	109,193.26	586	106,049.00
0	110,105.85	419	105,769.00
4	108,484.54	502	104,241.00
2	107,764.37	475	103,333.00
1	107,121.11	570	103,002.00

у объявлений, размещенных во вторник минимальна цена, в четверг - максимальна. Отличается от общей выборки.Зависимость несущественная

Зависимость стоимости квадратного метра от месяца размещения объявления.

In [92]:

```
#Зависимость стоимости квадратного метра от месяца размещения объявления.
center_spb_month_exposition = center_spb_data.pivot_table(index = 'month_exposition', values =
center_spb_month_exposition.columns = ['mean', 'count', 'median']
center_spb_month_exposition.plot(y = 'median')
plt.title('зависимость стоимости квадратного метра от месяца размещения объявления ')
plt.xlabel('месяц')
plt.ylabel('стоимость кв.м')
plt.show()
center_spb_month_exposition.sort_values('median', ascending = False)
```



Out[92]:

month_exposition	mean	count	median
9	112,060.60	286	107,265.00
3	109,255.38	283	106,783.00
1	109,022.99	169	105,555.00
11	110,010.31	337	104,950.00
2	109,729.96	355	104,895.00
7	107,325.99	232	104,694.50
4	107,453.27	183	104,629.00
10	106,738.22	339	104,615.00
12	109,428.26	171	104,545.00
8	107,925.97	288	104,330.50
6	105,788.40	173	103,035.00
5	102,589.82	151	100,000.00

стоимости квадратного метра центра СПБ минимальна в мае, максимальна в марте и сентябре. Зависимость существенная

Зависимость стоимости квадратного метра от года размещения объявления

In [93]:

```
#Зависимость стоимости квадратного метра от года размещения объявления.
center_spb_year_exposition = center_spb_data.pivot_table(index = 'year_exposition', values = 'l
center_spb_year_exposition.columns = ['mean', 'count', 'median']
center_spb_year_exposition.query('count > 50').plot(y = 'median')
plt.title('зависимость стоимости квадратного метра от года размещения объявления ')
plt.xlabel('год')
plt.ylabel('стоимость кв.м')
plt.show()
center_spb_year_exposition.query('count > 50').sort_values('median', ascending = False)
```

Out[93]:

year_exposition	mean	count	median
2019	116,765.03	312	113,718.00
2018	109,821.21	1446	106,357.50
2017	104,926.03	1029	100,656.00

	mean	count	median
year_exposition			
2016	103,227.28	179	99,726.00

Зависимость стоимости квадратного метра от года размещения объявления d центрt СПБ отличается от общей. Если в общей график резко растет до 2017 года, потом становится плавнее, то здесь наоборот. С 2017 года резкий рост стоимости.

Общий вывод

Обработал полученный архив объявлений о продаже квартир в Санкт-Петербурге и соседних населённых пунктах за несколько лет. Определил и изучил пропущенные значения. Там, где это необходимо, заменил типы данных на необходимые для удобной работы. Посчитал и добавил в таблицу цену квадратного метра жилья,соотношение жилой и общей площади, а также отношение площади кухни к общей. Вывел из даты дни недели, месяцы и года размещения объявлений, добавил категории квартир по этажам. Изучил следующие параметры на наличие выбивающихся значений - площадь, цена, число комнат, высота потолков. Определил аномалии в данных параметрах. Выявил при помощи диаграммы размаха, что нормальные значения продажи квартир варьируются от 1 до 200 дней.Выявил, что на стоимость квадратного метра квартиры больше всего влияют количество комнат, этаж квартиры, близость к центру.На стоимость квадратного метра также влияют день, месяц, год размещения. Общая площадь влияет незначительно.Имеет место постоянное удорожание стоимости квартир.По графику выявил центр города в радиусе 19 км.

Чек-лист готовности проекта

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

- [x] открыт файл
- [x] файлы изучены (выведены первые строки, метод info())
- [x] определены пропущенные значения
- [x] заполнены пропущенные значения
- [x] есть пояснение, какие пропущенные значения обнаружены
- [x] изменены типы данных
- [x] есть пояснение, в каких столбцах изменены типы и почему
- [x] посчитано и добавлено в таблицу: цена квадратного метра
- [x] посчитано и добавлено в таблицу: день недели, месяц и год публикации объявления
- [x] посчитано и добавлено в таблицу: этаж квартиры; варианты — первый, последний, другой
- [x] посчитано и добавлено в таблицу: соотношение жилой и общей площади, а также отношение площади кухни к общей
- [x] изучены следующие параметры: площадь, цена, число комнат, высота потолков
- [x] построены гистограммы для каждого параметра
- [x] выполнено задание: "Изучите время продажи квартиры. Постройте гистограмму. Посчитайте среднее и медиану. Опишите, сколько обычно занимает продажа. Когда можно считать, что продажи прошли очень быстро, а когда необычно долго?"
- [x] выполнено задание: "Уберите редкие и выбивающиеся значения. Опишите, какие особенности обнаружили."
- [x] выполнено задание: "Какие факторы больше всего влияют на стоимость квартиры? Изучите, зависит ли цена от квадратного метра, числа комнат, этажа (первого или последнего), удалённости от центра. Также изучите зависимость от даты размещения: дня недели, месяца и года. "Выберите 10 населённых пунктов с наибольшим числом объявлений. Посчитайте среднюю цену

квадратного метра в этих населённых пунктах. Выделите населённые пункты с самой высокой и низкой стоимостью жилья. Эти данные можно найти по имени в столбце '*locality_name*'. "

- [x] выполнено задание: "Изучите предложения квартир: для каждой квартиры есть информация о расстоянии до центра. Выделите квартиры в Санкт-Петербурге ('*locality_name*'). Ваша задача — выяснить, какая область входит в центр. Создайте столбец с расстоянием до центра в километрах: округлите до целых значений. После этого посчитайте среднюю цену для каждого километра. Постройте график: он должен показывать, как цена зависит от удалённости от центра. Определите границу, где график сильно меняется — это и будет центральная зона. "
- [x] выполнено задание: "Выделите сегмент квартир в центре. Проанализируйте эту территорию и изучите следующие параметры: площадь, цена, число комнат, высота потолков. Также выделите факторы, которые влияют на стоимость квартиры (число комнат, этаж, удалённость от центра, дата размещения объявления). Сделайте выводы. Отличаются ли они от общих выводов по всему городу?"
- [x] в каждом этапе есть выводы
- [x] есть общий вывод