**Исследование объявлений о продаже квартир**

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

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

**1  Откройте файл с данными и изучите общую информацию.**

Ввод [1]:

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

​

​

data **=** pd.read\_csv('/datasets/real\_estate\_data.csv', sep**=**'\t')

​

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

В данных имеются пропуски, есть проблемы с типами данных, всего 23699 строчки.

Ввод [2]:

*#построим общую гистограмму для всех столбцов таблицы*

data.hist(figsize**=**(15, 20))

Out[2]:

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'}>,

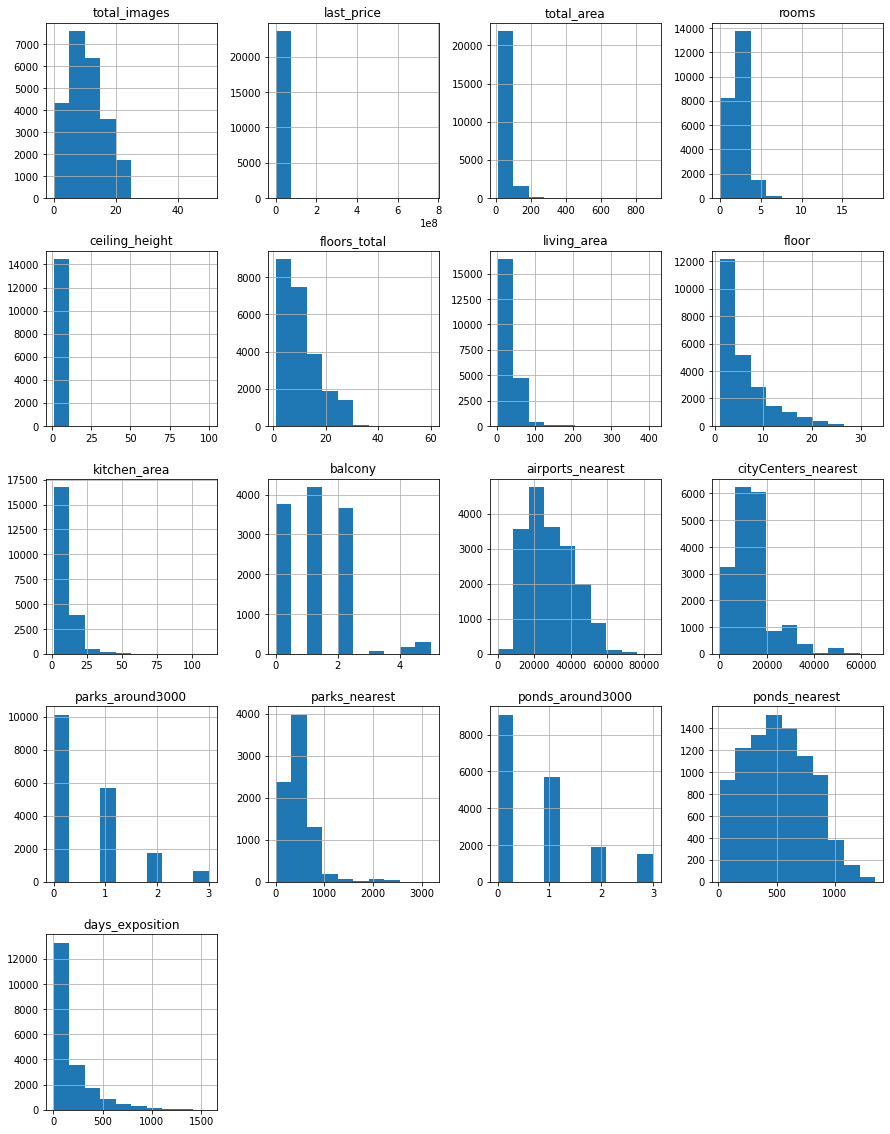
<AxesSubplot:title={'center':'parks\_nearest'}>,

<AxesSubplot:title={'center':'ponds\_around3000'}>,

<AxesSubplot:title={'center':'ponds\_nearest'}>],

[<AxesSubplot:title={'center':'days\_exposition'}>, <AxesSubplot:>,

<AxesSubplot:>, <AxesSubplot:>]], dtype=object)



При беглом рассмотрении полученных данных вижу наличие выбросов (в большую сторону либо пик нулевых значений) в значениях столбцов total\_area, rooms, living\_area, kitchen\_area, balcony, days\_exposition. Рассмотрю их внимательнее при дальнейшем исследовании данных.

Оставила выше по вашей рекоммендации.

**2  Предобработка данных**

Ввод [3]:

data.isna().sum()

Out[3]:

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

Большинство попущенных значений находятся в столбцах: высота потолков, апартаменты, число балконов, расстояние до ближайшего водоёма или парка.

Ввод [4]:

*#заполняю пропуски в столбце ceiling\_height медианным значением*

data['ceiling\_height'] **=** data['ceiling\_height'].fillna(data['ceiling\_height'].median())

​

*#заполняю пропуски в столбце balcony значением ноль, тк возможно у объекта просто не балкона*

data['balcony'] **=** data['balcony'].fillna(0)

​

*#заполняю пропуски в столбце living\_area значением ноль, тк возможно объект - это офисное помещение и площадь в нем просто не считается жилой*

data['living\_area'] **=** data['living\_area'].fillna(0)

​

*#заменяю пропуски в столбце is\_apartment на unknown, тк скорее всего собственник не знает статус объекта (в новых домах например могут располагаться как квартиры студии так и апартаменты - иногда сложно разобраться)*

data['is\_apartment'] **=** data['is\_apartment'].fillna('unknown')

​

​

​

​

Заменяю пропуски в ceiling\_height медианным значением, тк оно менее всего искажает данные, которые могут понадобиться мне для дальнейших рассчетов. Заполняю пропуски в столбце balcony значением ноль, тк возможно у объекта просто не балкона. Заполняю пропуски в столбце living\_area значением ноль, тк возможно объект - это офисное помещение и площадь в нем просто не считается жилой. Заменяю пропуски в столбце is\_apartment на unknown, тк скорее всего собственник не знает статус объекта (в новых домах например могут располагаться как квартиры студии так и апартаменты - иногда сложно разобраться).

​

​

​

(доработала в соответствии с замечаниями)

Пропуски в данных появляются разными путями. Например, пользователь не указал какие-нибудь сведения или платформа, на которой размещены объявления дала сбой. Иногда пропуски оставляют умышленно, рассчитывая на автозаполнение спецсимволами.

Ввод [5]:

*#меняю типы данных*

data['last\_price'] **=** data['last\_price'].astype(int)

data['balcony'] **=** data['balcony'].astype(int)

data['first\_day\_exposition'] **=** pd.DatetimeIndex(data['first\_day\_exposition'])

*#data.isna().sum()*

Изменение типов данных необходимо для того чтобы иметь возможность по-другому манипулировать значениями (в случае с датой - для облегчения визуального восприятия). Например, нам может потребоваться объединить числовые значения со строками или представить десятичные дроби в виде целых чисел.

Ввод [6]:

*#ищу уникальные значения в столбце locality\_name*

data['locality\_name'].unique()

​

*#из списка уникальных значений выбираю те, на которые заменю их неявные дубликаты и делаю это*

data['locality\_name']**=**data['locality\_name'].replace('городской посёлок Янино-1', 'Поселок Янино-1')

data['locality\_name']**=**data['locality\_name'].replace('Мурино', 'посёлок Мурино')

data['locality\_name']**=**data['locality\_name'].replace('Кудрово', 'деревня Кудрово')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Красный Бор', 'посёлок городского типа Красный Бор')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Лебяжье', 'посёлок городского типа Лебяжье')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Кузьмоловский', 'посёлок городского типа Кузьмоловский')

data['locality\_name']**=**data['locality\_name'].replace('поселок Аннино', 'посёлок Аннино')

data['locality\_name']**=**data['locality\_name'].replace('поселок Новый Свет', 'посёлок Новый Свет')

data['locality\_name']**=**data['locality\_name'].replace('Никольское', 'село Никольское')

data['locality\_name']**=**data['locality\_name'].replace('поселок Бугры', 'посёлок Бугры')

data['locality\_name']**=**data['locality\_name'].replace('село Павлово', 'посёлок городского типа Павлово')

data['locality\_name']**=**data['locality\_name'].replace('поселок Победа', 'посёлок Победа')

data['locality\_name']**=**data['locality\_name'].replace('поселок Терволово', 'посёлок Терволово')

data['locality\_name']**=**data['locality\_name'].replace('городской посёлок Советский', 'поселок городского типа Советский')

data['locality\_name']**=**data['locality\_name'].replace('поселок Петровское', 'посёлок Петровское')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Токсово', 'посёлок городского типа Токсово')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Вырица', 'посёлок городского типа Вырица')

data['locality\_name']**=**data['locality\_name'].replace('поселок Тельмана', 'посёлок Тельмана')

data['locality\_name']**=**data['locality\_name'].replace('городской поселок Большая Ижора' , 'поселок городского типа Большая Ижора')

data['locality\_name']**=**data['locality\_name'].replace('городской посёлок Лесогорский', 'поселок городского типа Лесогорский')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа имени Свердлова', 'посёлок городского типа имени Свердлова')

data['locality\_name']**=**data['locality\_name'].replace(['поселок Рябово' 'поселок городского типа Рябово'], 'посёлок городского типа Рябово')

data['locality\_name']**=**data['locality\_name'].replace('поселок Поляны', 'посёлок Поляны')

data['locality\_name']**=**data['locality\_name'].replace('городской посёлок Мга', 'посёлок городского типа Мга')

data['locality\_name']**=**data['locality\_name'].replace('поселок Мичуринское', 'посёлок Мичуринское')

data['locality\_name']**=**data['locality\_name'].replace('поселок Ильичёво', 'посёлок Ильичёво')

data['locality\_name']**=**data['locality\_name'].replace('поселок станции Вещево', 'посёлок при железнодорожной станции Вещево')

data['locality\_name']**=**data['locality\_name'].replace('поселок Гаврилово', 'посёлок Гаврилово')

data['locality\_name']**=**data['locality\_name'].replace('городской посёлок Назия', 'поселок городского типа Назия')

data['locality\_name']**=**data['locality\_name'].replace('поселок Старая Малукса', 'посёлок Старая Малукса')

data['locality\_name']**=**data['locality\_name'].replace('поселок станции Приветнинское', 'посёлок при железнодорожной станции Приветнинское')

data['locality\_name']**=**data['locality\_name'].replace('поселок Мельниково', 'посёлок Мельниково')

data['locality\_name']**=**data['locality\_name'].replace('поселок Усть-Луга', 'посёлок Усть-Луга')

data['locality\_name']**=**data['locality\_name'].replace('Любань', 'поселок Любань')

data['locality\_name']**=**data['locality\_name'].replace('поселок Стеклянный', 'посёлок Стеклянный')

data['locality\_name']**=**data['locality\_name'].replace('поселок городского типа Форносово', 'посёлок городского типа Форносово')

data['locality\_name']**=**data['locality\_name'].replace('поселок Сумино', 'посёлок Сумино')

data['locality\_name']**=**data['locality\_name'].replace('поселок Возрождение', 'посёлок Возрождение')

data['locality\_name']**=**data['locality\_name'].replace('поселок Лисий Нос', 'посёлок Лисий Нос')

data['locality\_name']**=**data['locality\_name'].replace('поселок Жилгородок', 'посёлок Жилгородок')

data['locality\_name']**=**data['locality\_name'].replace('поселок Коробицыно', 'посёлок Коробицыно')

data['locality\_name']**=**data['locality\_name'].replace('поселок Кобралово', 'посёлок Кобралово')

data['locality\_name']**=**data['locality\_name'].replace('коттеджный посёлок Лесное', 'поселок Лесное')

data['locality\_name']**=**data['locality\_name'].replace('поселок Перово', 'посёлок Перово')

data['locality\_name']**=**data['locality\_name'].replace('поселок Пансионат Зелёный Бор', 'посёлок Пансионат Зелёный Бор')

​

*#data['locality\_name'].head(20)*

​

​

Ввод [7]:

data.describe()

Out[7]:

|  | **total\_images** | **last\_price** | **total\_area** | **rooms** | **ceiling\_height** | **floors\_total** | **living\_area** | **floor** | **kitchen\_area** | **balcony** | **airports\_nearest** | **cityCenters\_nearest** | **parks\_around3000** | **parks\_nearest** | **ponds\_around3000** | **ponds\_nearest** | **days\_exposition** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 23699.000000 | 2.369900e+04 | 23699.000000 | 23699.000000 | 23699.000000 | 23613.000000 | 23699.000000 | 23699.000000 | 21421.000000 | 23699.000000 | 18157.000000 | 18180.000000 | 18181.000000 | 8079.000000 | 18181.000000 | 9110.000000 | 20518.000000 |
| **mean** | 9.858475 | 6.541549e+06 | 60.348651 | 2.070636 | 2.724358 | 10.673824 | 31.690930 | 5.892358 | 10.569807 | 0.591080 | 28793.672193 | 14191.277833 | 0.611408 | 490.804555 | 0.770255 | 517.980900 | 180.888634 |
| **std** | 5.682529 | 1.088701e+07 | 35.654083 | 1.078405 | 0.988298 | 6.597173 | 23.109666 | 4.885249 | 5.905438 | 0.959298 | 12630.880622 | 8608.386210 | 0.802074 | 342.317995 | 0.938346 | 277.720643 | 219.727988 |
| **min** | 0.000000 | 1.219000e+04 | 12.000000 | 0.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 | 1.300000 | 0.000000 | 0.000000 | 181.000000 | 0.000000 | 1.000000 | 0.000000 | 13.000000 | 1.000000 |
| **25%** | 6.000000 | 3.400000e+06 | 40.000000 | 1.000000 | 2.600000 | 5.000000 | 17.800000 | 2.000000 | 7.000000 | 0.000000 | 18585.000000 | 9238.000000 | 0.000000 | 288.000000 | 0.000000 | 294.000000 | 45.000000 |
| **50%** | 9.000000 | 4.650000e+06 | 52.000000 | 2.000000 | 2.650000 | 9.000000 | 29.000000 | 4.000000 | 9.100000 | 0.000000 | 26726.000000 | 13098.500000 | 0.000000 | 455.000000 | 1.000000 | 502.000000 | 95.000000 |
| **75%** | 14.000000 | 6.800000e+06 | 69.900000 | 3.000000 | 2.700000 | 16.000000 | 41.100000 | 8.000000 | 12.000000 | 1.000000 | 37273.000000 | 16293.000000 | 1.000000 | 612.000000 | 1.000000 | 729.000000 | 232.000000 |
| **max** | 50.000000 | 7.630000e+08 | 900.000000 | 19.000000 | 100.000000 | 60.000000 | 409.700000 | 33.000000 | 112.000000 | 5.000000 | 84869.000000 | 65968.000000 | 3.000000 | 3190.000000 | 3.000000 | 1344.000000 | 1580.000000 |

Для того, чтобы выявить аномали в данных я воспользовалась методом describe(): выбивающиеся значения были обнаружены в столбцах ceiling\_height, total\_area, rooms. Построив гистограмму для этих столбцов заметила, что это 1 значение для каждого: т.к. доля подобных значений крайне мала - просто удалю их.

Ввод [8]:

*#перезаписываю data исключая из него единичные выбросы*

data **=** data.query('ceiling\_height < 10 and total\_area < 400 and rooms <10')

*#data.hist(bins=50)*

**3  Посчитайте и добавьте в таблицу новые столбцы**

Ввод [9]:

*#цена одного квадратного метра*

data['price\_per\_m2'] **=** data['last\_price'] **/** data['total\_area']

*#день недели публикации объявления*

data['day'] **=** data['first\_day\_exposition'].dt.weekday

*#месяц публикации объявления*

data['month'] **=** data['first\_day\_exposition'].dt.month

*#год публикации объявления*

data['year'] **=** data['first\_day\_exposition'].dt.year

*#тип этажа квартиры*

**def** categorize\_floors(row):

**if** row['floor'] **==** 1:

**return** 'первый'

**elif** row['floor'] **==** row['floors\_total']:

**return** 'последний'

**else**:

**return** 'другой'

​

data['type\_floor'] **=** data.apply(categorize\_floors, axis**=**1)

​

​

​

*#расстояние до центра города в километрах*

data['cityCenters\_nearest\_km'] **=** (data['cityCenters\_nearest']**/**1000).round()

​

*#data*

**4  Проведите исследовательский анализ данных**

Ввод [10]:

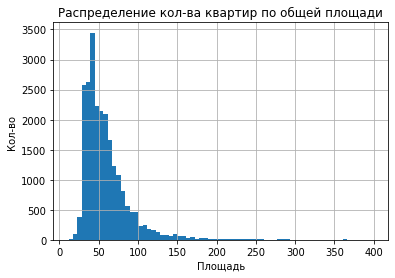
ax **=** data['total\_area'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по общей площади', grid**=True**)

ax.set\_xlabel('Площадь')

ax.set\_ylabel('Кол-во')

Out[10]:

Text(0, 0.5, 'Кол-во')



Ввод [11]:

data['total\_area'].describe()

Out[11]:

count 23639.000000

mean 59.797936

std 32.159745

min 12.000000

25% 40.000000

50% 52.000000

75% 69.600000

max 399.000000

Name: total\_area, dtype: float64

Площадь большинства квартир в продаже составляет около 50м, чуть меньше квартир в группе 30-40 м (распределение поддается логике, тк население растет, а площади под застройку не увеличиваются, растет этажность домов и уменьшается сама площадь квартир), очень маленькое количество квартир имеет площадь от 150м, встречаются единичные объекты площадью от 300м и более (вероятно принадлежат к старому жилому фонду, тк во времена их постройки маленьких квартир не строили).

Ввод [12]:

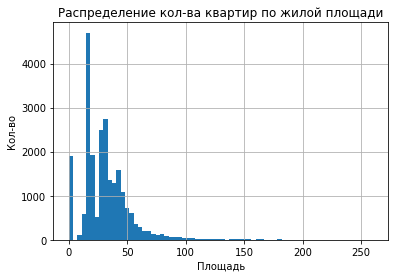
ax **=** data['living\_area'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по жилой площади', grid**=True**)

ax.set\_xlabel('Площадь')

ax.set\_ylabel('Кол-во')

Out[12]:

Text(0, 0.5, 'Кол-во')



Ввод [13]:

data['living\_area'].describe()

Out[13]:

count 23639.000000

mean 31.448308

std 21.770473

min 0.000000

25% 17.800000

50% 29.000000

75% 41.000000

max 260.000000

Name: living\_area, dtype: float64

Распределение пиков на гистограмме выглядит нормально за исключением нулевого пика вначале(около 2 тысяч): я предполагаю что это количество объектов, которые представляют собой офисные помещения или апартаменты, тк на уровне законодательства их площадь не считается "жилой" и собственники обязаны указывать эту информацию об объекте при продаже.

Ввод [14]:

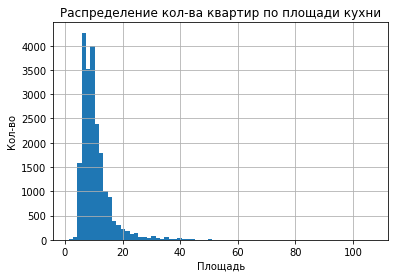
ax **=** data['kitchen\_area'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по площади кухни', grid**=True**)

ax.set\_xlabel('Площадь')

ax.set\_ylabel('Кол-во')

Out[14]:

Text(0, 0.5, 'Кол-во')



Ввод [15]:

data['kitchen\_area'].describe()

Out[15]:

count 21371.000000

mean 10.533171

std 5.754619

min 1.300000

25% 7.000000

50% 9.100000

75% 12.000000

max 107.000000

Name: kitchen\_area, dtype: float64

Расределение выглядит нормальным и похоже на данные по общей площади квартир (кухни есть во всех выставляемых объектах).

Ввод [16]:

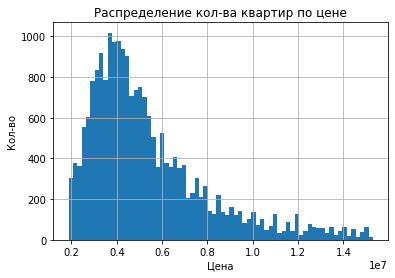
ax **=** data['last\_price'].plot(kind**=**'hist', bins**=**70, range**=**(1870000,15300000), title**=**'Распределение кол-ва квартир по цене', grid**=True**)

ax.set\_xlabel('Цена')

ax.set\_ylabel('Кол-во')

Out[16]:

Text(0, 0.5, 'Кол-во')



Ввод [17]:

data['last\_price'].describe()

Out[17]:

count 2.363900e+04

mean 6.381065e+06

std 8.207500e+06

min 1.219000e+04

25% 3.400000e+06

50% 4.644000e+06

75% 6.752530e+06

max 3.300000e+08

Name: last\_price, dtype: float64

Стоимость большинства квартир находится в районе 4 млн.руб.

Ввод [18]:

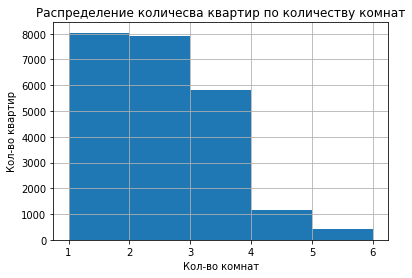
ax **=** data['rooms'].plot(kind**=**'hist', bins**=**5, range**=**(1,6), title**=**'Распределение количесва квартир по количеству комнат', grid**=True**)

ax.set\_xlabel('Кол-во комнат')

ax.set\_ylabel('Кол-во квартир')

Out[18]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [19]:

data['rooms'].describe()

Out[19]:

count 23639.000000

mean 2.061932

std 1.043745

min 0.000000

25% 1.000000

50% 2.000000

75% 3.000000

max 9.000000

Name: rooms, dtype: float64

Самые распространненые-это однокомнатные и двухкомнатные квартиры.

Ввод [20]:

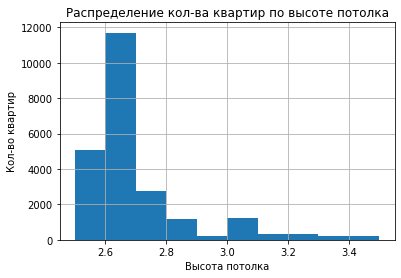
ax **=** data['ceiling\_height'].plot(kind**=**'hist', bins**=**10, range**=**(2.5,3.5), title**=**'Распределение кол-ва квартир по высоте потолка', grid**=True**)

ax.set\_xlabel('Высота потолка')

ax.set\_ylabel('Кол-во квартир')

Out[20]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [21]:

data['rooms'].describe()

Out[21]:

count 23639.000000

mean 2.061932

std 1.043745

min 0.000000

25% 1.000000

50% 2.000000

75% 3.000000

max 9.000000

Name: rooms, dtype: float64

Чаще всего встречается высота потолка 2.7, вероятно из-за массовой застройки нового однотипного жилья.

Ввод [22]:

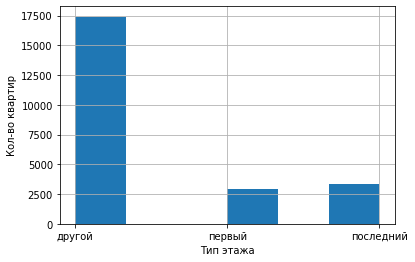
data['type\_floor'].hist(bins**=**6, grid**=True**)

plt.xlabel('Тип этажа')

plt.ylabel('Кол-во квартир')

Out[22]:

Text(0, 0.5, 'Кол-во квартир')



Большинство квартир из списка объявлений располагается на "других" этажах, то есть не на первлм и не на последнем. Распределение по каждому представленному этажу можно увидееть на графике ниже (значения убывают от меньшего этажа к большему возможжно в связи с тем, что из-за большой площади исторического центра города в котором запрещена застройка многоэтажными домами).

Ввод [23]:

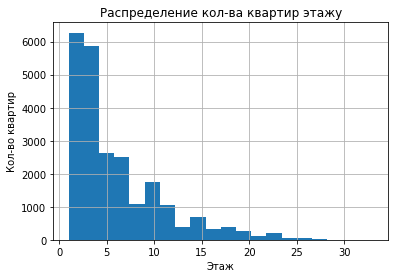
ax **=** data['floor'].plot(kind**=**'hist', bins**=**20, title**=**'Распределение кол-ва квартир этажу', grid**=True**)

ax.set\_xlabel('Этаж')

ax.set\_ylabel('Кол-во квартир')

Out[23]:

Text(0, 0.5, 'Кол-во квартир')



Значительная часть квартир продается на 1-2 этажах(это более 6тыс квартир), следующая по объему группа располагается на 3-4 этажах - около 6 тыс квартир.

Ввод [24]:

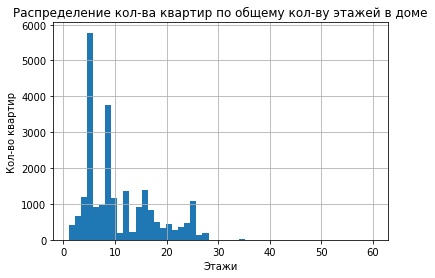
ax **=** data['floors\_total'].plot(kind**=**'hist', bins**=**50, title**=**'Распределение кол-ва квартир по общему кол-ву этажей в доме', grid**=True**)

ax.set\_xlabel('Этажи')

ax.set\_ylabel('Кол-во квартир')

Out[24]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [25]:

data['floors\_total'].describe()

Out[25]:

count 23553.000000

mean 10.679489

std 6.596204

min 1.000000

25% 5.000000

50% 9.000000

75% 16.000000

max 60.000000

Name: floors\_total, dtype: float64

Большая часть продаваемых объектов сосредоточена в 5 этажных домах, далее идут 9 этажки и 3-е место делят между собой 12-ти и 16-ти этажные дома.

Ввод [26]:

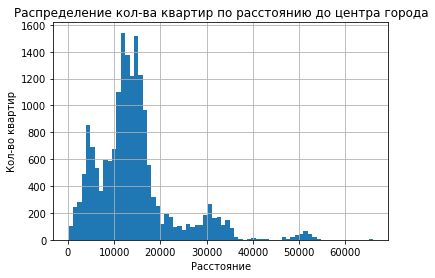
ax **=** data['cityCenters\_nearest'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по расстоянию до центра города', grid**=True**)

ax.set\_xlabel('Расстояние')

ax.set\_ylabel('Кол-во квартир')

Out[26]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [27]:

data['floors\_total'].describe()

Out[27]:

count 23553.000000

mean 10.679489

std 6.596204

min 1.000000

25% 5.000000

50% 9.000000

75% 16.000000

max 60.000000

Name: floors\_total, dtype: float64

Большая часть продаваемых объектов удалена от центра на 12 и 13 тысяч метров.Есть небольшое значение объектов которые отдалены более чем на 60 тысяч метров.

Ввод [28]:

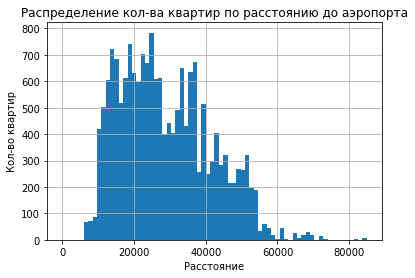
ax **=** data['airports\_nearest'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по расстоянию до аэропорта', grid**=True**)

ax.set\_xlabel('Расстояние')

ax.set\_ylabel('Кол-во квартир')

Out[28]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [29]:

data['airports\_nearest'].describe()

Out[29]:

count 18109.000000

mean 28797.350213

std 12636.248654

min 0.000000

25% 18575.000000

50% 26736.000000

75% 37288.000000

max 84869.000000

Name: airports\_nearest, dtype: float64

Большинство продаваемых объектов располагается на расстоянии 25 тысяч метров от Пулково, следующая по величине группа располагается на расстоянии приблизительно 19 тысяч метров от аэропорта. Есть небольшое значение объектов которые отдалены более чем на 80 тысяч метров от аэропорта.

Ввод [30]:

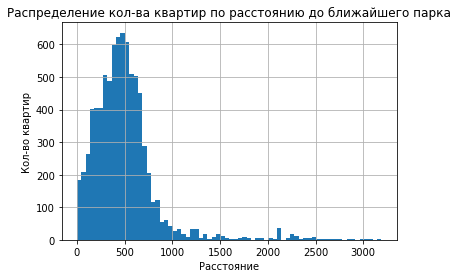
ax **=** data['parks\_nearest'].plot(kind**=**'hist', bins**=**70, title**=**'Распределение кол-ва квартир по расстоянию до ближайшего парка', grid**=True**)

ax.set\_xlabel('Расстояние')

ax.set\_ylabel('Кол-во квартир')

Out[30]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [31]:

data['parks\_nearest'].describe()

Out[31]:

count 8049.000000

mean 490.935396

std 342.669098

min 1.000000

25% 288.000000

50% 455.000000

75% 612.000000

max 3190.000000

Name: parks\_nearest, dtype: float64

Самая больша группа объектов располагается в 500 м от ближайшего парка.

Ввод [32]:

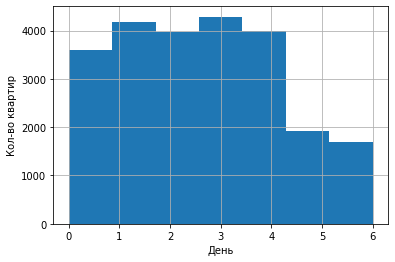
data['day'].hist(bins**=**7, grid**=True**)

plt.xlabel('День')

plt.ylabel('Кол-во квартир')

Out[32]:

Text(0, 0.5, 'Кол-во квартир')



Чаще всего объявления выставляют во второй и четвертый день (предположительно во вторник и четверг - возможно из-за снижения пиков занятости).

Ввод [48]:

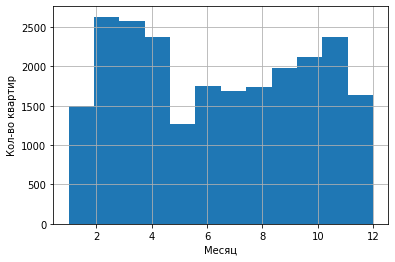
data['month'].hist(bins**=**12, grid**=True**)

plt.xlabel('Месяц')

plt.ylabel('Кол-во квартир')

Out[48]:

Text(0, 0.5, 'Кол-во квартир')



И в феврале, когда все возвращаются с новогодних каникул.Так же следующий пик располагается в ноябре перед последним месяцем в году, когда занятость людей достигает своего пика.

Ввод [50]:

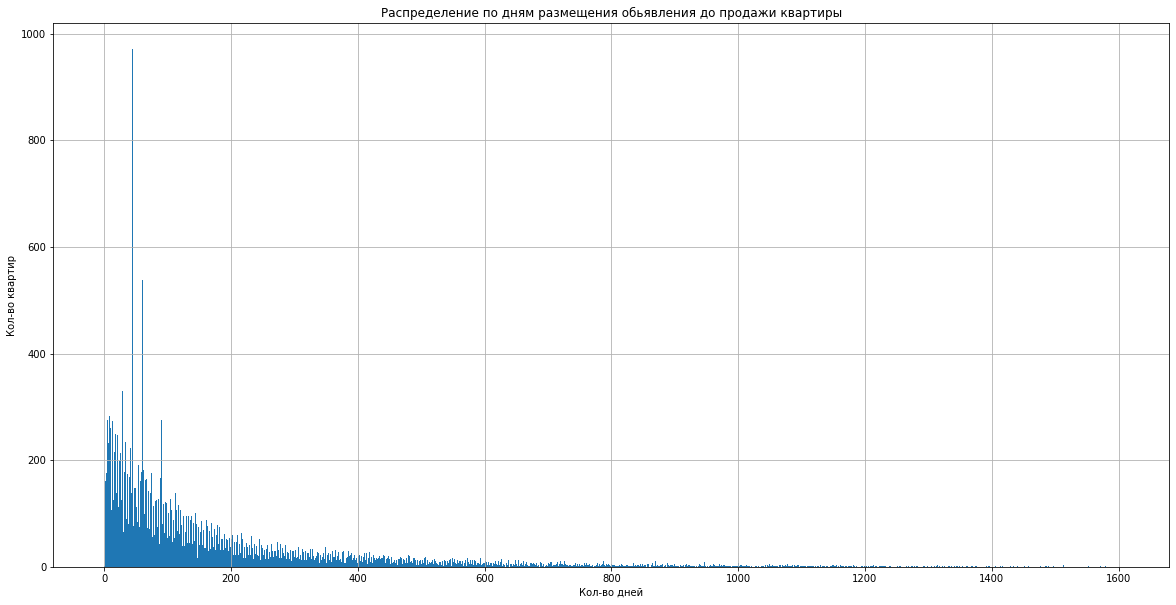
ax **=** data['days\_exposition'].plot(kind**=**'hist', bins**=**1000, range **=** (0,1600), figsize**=**(20, 10), title**=**'Распределение по дням размещения обьявления до продажи квартиры', grid**=True**)

ax.set\_xlabel('Кол-во дней')

ax.set\_ylabel('Кол-во квартир')

Out[50]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [51]:

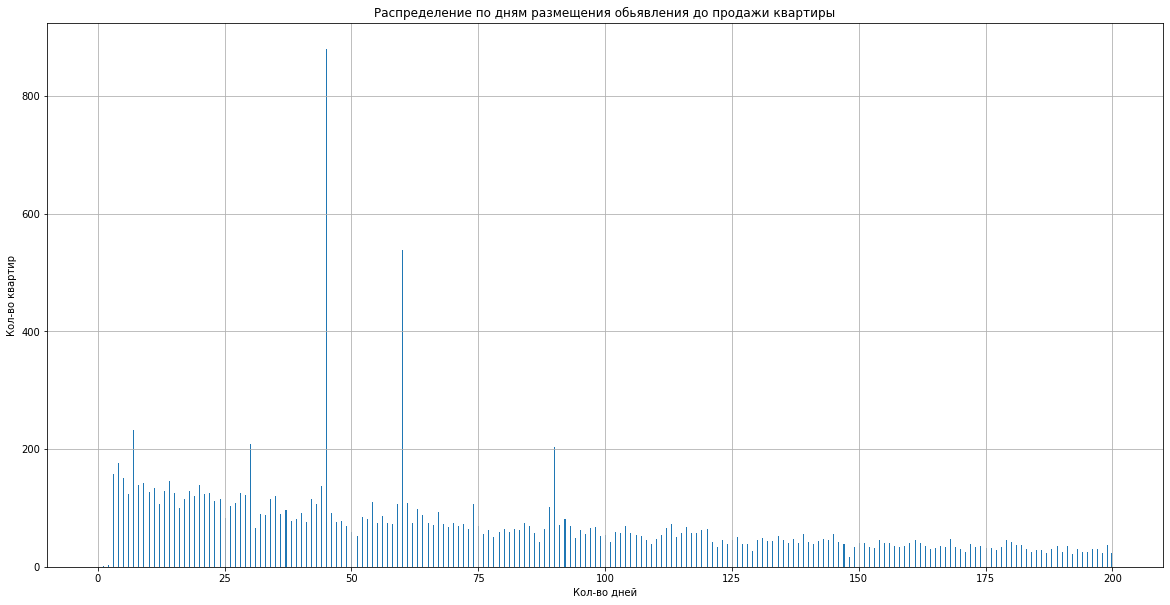
ax **=** data['days\_exposition'].plot(kind**=**'hist', bins**=**1000, range **=** (0,200), figsize**=**(20, 10), title**=**'Распределение по дням размещения обьявления до продажи квартиры', grid**=True**)

ax.set\_xlabel('Кол-во дней')

ax.set\_ylabel('Кол-во квартир')

Out[51]:

Text(0, 0.5, 'Кол-во квартир')



Ввод [54]:

data['days\_exposition'].mean(), data['days\_exposition'].median()

​

Out[54]:

(180.82914081961607, 95.0)

Ввод [53]:

data['days\_exposition'].quantile(0.05), data['days\_exposition'].quantile(0.95)

​

Out[53]:

(9.0, 647.0)

На первом графике видно, что основная масса квартир продается в первые 100 дней экспозиции. На втором графике мы можем увидеть, что квартиры чаще продаются на 45 (можно считать быстрой продажей) и 60-ый дни. Так же есть объекты, которые находятся на экспозиции около 4 лет - такую продажу можно считать аномально долгой.

Ввод [36]:

*#считаю коэффициент Пирсона для параметров: total\_area ,living\_area, kitchen\_area, rooms, day, month, year*

data[['last\_price', 'total\_area', 'living\_area', 'kitchen\_area', 'rooms', 'type\_floor', 'day', 'month', 'year']].corr()

Out[36]:

|  | **last\_price** | **total\_area** | **living\_area** | **kitchen\_area** | **rooms** | **day** | **month** | **year** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **last\_price** | 1.000000 | 0.688399 | 0.516557 | 0.527574 | 0.399993 | -0.005094 | 0.001848 | -0.051747 |
| **total\_area** | 0.688399 | 1.000000 | 0.791961 | 0.595224 | 0.776752 | -0.008598 | 0.008023 | -0.089639 |
| **living\_area** | 0.516557 | 0.791961 | 1.000000 | 0.364581 | 0.745041 | -0.021551 | 0.007352 | -0.093072 |
| **kitchen\_area** | 0.527574 | 0.595224 | 0.364581 | 1.000000 | 0.246363 | -0.009676 | 0.003921 | -0.057517 |
| **rooms** | 0.399993 | 0.776752 | 0.745041 | 0.246363 | 1.000000 | -0.005461 | -0.002801 | -0.056802 |
| **day** | -0.005094 | -0.008598 | -0.021551 | -0.009676 | -0.005461 | 1.000000 | 0.011390 | -0.004672 |
| **month** | 0.001848 | 0.008023 | 0.007352 | 0.003921 | -0.002801 | 0.011390 | 1.000000 | -0.269210 |
| **year** | -0.051747 | -0.089639 | -0.093072 | -0.057517 | -0.056802 | -0.004672 | -0.269210 | 1.000000 |

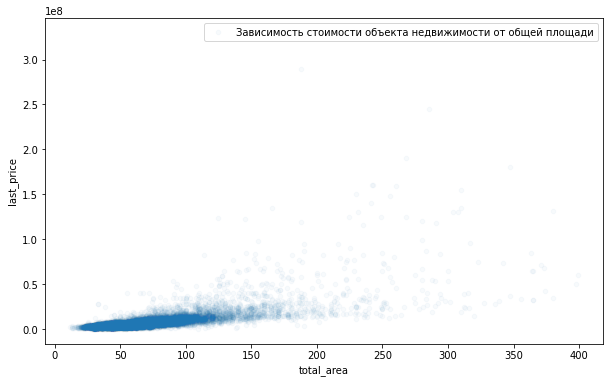
Ввод [37]:

*#графики зависимости цены от указанных выше параметров*

data.plot(x**=**'total\_area', y**=**'last\_price', kind**=**'scatter', alpha**=**0.03, label**=**'Зависимость стоимости объекта недвижимости от общей площади', figsize**=**(10,6))

Out[37]:

<AxesSubplot:xlabel='total\_area', ylabel='last\_price'>



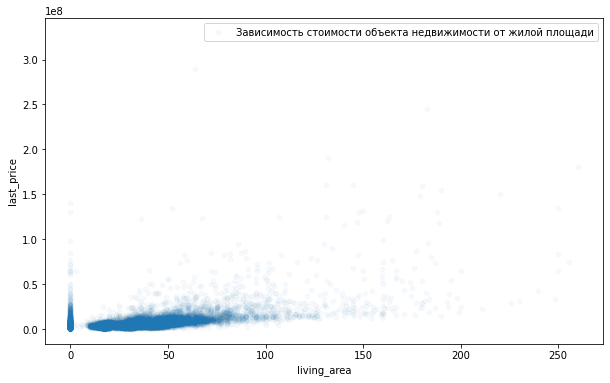
На графике рассеяния можно заметить, что чем больше площадь квартиры, тем больше ее цена (что подстверждается и значением коэф-та Пирсона по данной зависимости он составляет - 0.6).

Ввод [38]:

data.plot(x**=**'living\_area', y**=**'last\_price', kind**=**'scatter', alpha**=**0.03, label**=**'Зависимость стоимости объекта недвижимости от жилой площади', figsize**=**(10,6))

Out[38]:

<AxesSubplot:xlabel='living\_area', ylabel='last\_price'>



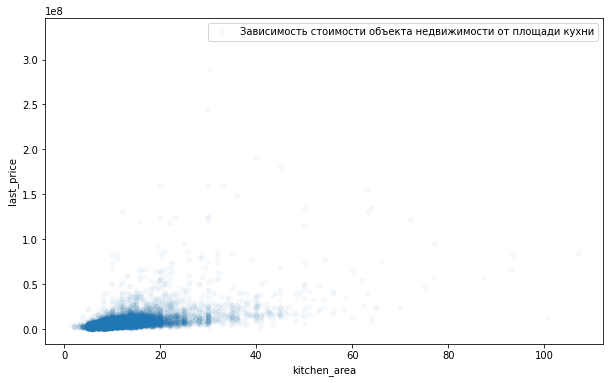
Та же тендения прослеживается и с зависимостью от объема жилой площади (коэф-т Пирсона - 0.5 - чуть меньше в связи с имеющимся пиком в начале оси х - 0 м жилой площади, который я связываю с ниличием в продаже офисных помещений или апартаментов, где жилая площадь не выделена по документации, если исключить эти значения мы получим коэф-т зависимости равный 0.6).

Ввод [39]:

data.plot(x**=**'kitchen\_area', y**=**'last\_price', kind**=**'scatter', alpha**=**0.03, label**=**'Зависимость стоимости объекта недвижимости от площади кухни', figsize**=**(10,6))

Out[39]:

<AxesSubplot:xlabel='kitchen\_area', ylabel='last\_price'>



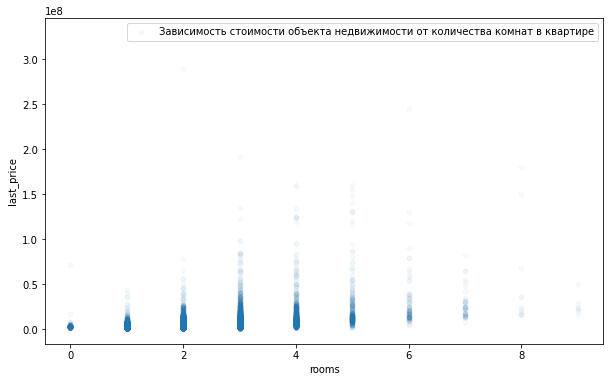
На графике видно, что чем больше площадь кухни, тем больше цена объекта недвижимости(что подтверждается и значением коэф-та Пирсона по данной зависимости он составляет 0.5).

Ввод [40]:

data.plot(x**=**'rooms', y**=**'last\_price', kind**=**'scatter', alpha**=**0.03, label**=**'Зависимость стоимости объекта недвижимости от количества комнат в квартире', figsize**=**(10,6))

Out[40]:

<AxesSubplot:xlabel='rooms', ylabel='last\_price'>



Коэф-т Пирсона при рассчете зависимости стоимости квартиры от количества комнат составил всего 0.3, то есть нельзя сказать что количество комнат в квартире всегда однозначно оказывает влияние на ее стоимость: как мы видим на графике при росте количества комнат от 1 до 4 наблюдается и рост цены объекта, но далее мы можем наблюдать снижение стоимости, затем небольшой рост в цене на 8-ми комнатные квартиры и снова спад.

Ввод [41]:

*#график зависимости средней цены квартиры от типа этажа*

e**=**data.groupby('type\_floor')['last\_price'].agg('mean')

print(e)

e.plot.bar()

​

type\_floor

другой 6.626323e+06

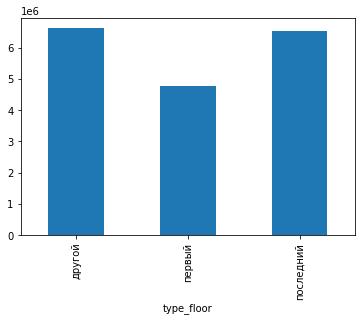
первый 4.758714e+06

последний 6.518541e+06

Name: last\_price, dtype: float64

Out[41]:

<AxesSubplot:xlabel='type\_floor'>



Тип этажа так же оказывает влияние на стоимость квартиры: люди предпочитаю покупать квартиры находящиеся на средних этажах, тк там меньше риск затопления с крыши при осадках или затопления от прорыва канализиционных труб.

Ввод [42]:

*#график зависимости средней цены квартиры от дня публикации*

e**=**data.groupby('day')['last\_price'].mean()

print(e)

e.plot.bar()

day

0 6.366513e+06

1 6.338435e+06

2 6.390930e+06

3 6.656358e+06

4 6.357675e+06

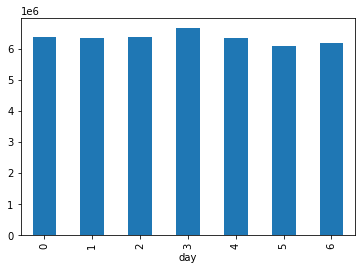
5 6.102318e+06

6 6.169800e+06

Name: last\_price, dtype: float64

Out[42]:

<AxesSubplot:xlabel='day'>



Прослеживается совсем небольшая зависимость (что подтверждается величиной коэ-та Пирсона 0,01): стоимость квартир в объявлениях размещенных по четвергам чуть больше, чем в остальные дни.

Ввод [43]:

*#график зависимости средней цены квартиры от месяца публикации*

e**=**data.groupby('month')['last\_price'].mean()

print(e)

e.plot.bar()

month

1 6.527365e+06

2 6.271710e+06

3 6.246637e+06

4 6.543343e+06

5 6.329995e+06

6 6.227845e+06

7 6.633744e+06

8 6.351352e+06

9 6.426382e+06

10 6.129610e+06

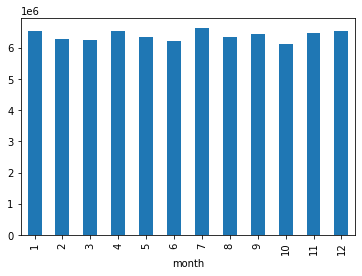
11 6.462865e+06

12 6.527004e+06

Name: last\_price, dtype: float64

Out[43]:

<AxesSubplot:xlabel='month'>



По месяцам размещения так же прослеживается совсем небольшая зависимость (что подтверждается величиной коэ-та Пирсона 0,01 и графиком зависимости): стоимость квартир в объявлениях размещенных в январе, апреле и июле чуть больше, чем в остальные дни.

Ввод [44]:

*#график зависимости средней цены квартиры от года публикации*

e**=**data.groupby('year')['last\_price'].mean()

print(e)

e.plot.bar()

year

2014 1.208714e+07

2015 8.498354e+06

2016 6.611455e+06

2017 6.279777e+06

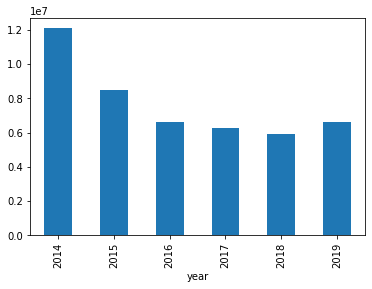
2018 5.934431e+06

2019 6.621434e+06

Name: last\_price, dtype: float64

Out[44]:

<AxesSubplot:xlabel='year'>



По графику и значению коэф-та Пирсона (-0.5) мы можем видеть значительное падение цен на недвижимость с 2014 по 2018гг (связан с продолжающимся экономическим кризисом и отказом банков выдавать ипотечные ссуды) и затем небольшой рост в 2019г (среди основных драйверов роста цен в 2019 году - переход на эскроу-счета, они предполагали развитие кредитования застройщиков, а значит, у девелоперов появились дополнительные расходы, которые они заложили в конечную цену жилья).

Ввод [45]:

top\_locality **=** data['locality\_name'].value\_counts().head(10)

locality\_price **=** data.pivot\_table(index**=**'locality\_name', values**=**'price\_per\_m2')

top\_10 **=** locality\_price.merge(top\_locality, how**=**'right', left\_index**=True**, right\_index**=True**)

top\_10.sort\_values(by**=**'price\_per\_m2', ascending**=False**)

Out[45]:

|  | **price\_per\_m2** | **locality\_name** |
| --- | --- | --- |
| **Санкт-Петербург** | 114521.613089 | 15671 |
| **Пушкин** | 103125.819291 | 369 |
| **деревня Кудрово** | 95324.930544 | 472 |
| **посёлок Парголово** | 90175.913458 | 327 |
| **посёлок Мурино** | 86154.688847 | 554 |
| **посёлок Шушары** | 78677.364217 | 440 |
| **Колпино** | 75424.579098 | 338 |
| **Гатчина** | 68757.678644 | 306 |
| **Всеволожск** | 68654.473986 | 398 |
| **Выборг** | 58141.909153 | 237 |

Ввод [46]:

price\_by\_city\_center **=** (data[data['locality\_name'] **==** 'Санкт-Петербург'].pivot\_table(index**=**'cityCenters\_nearest\_km', values**=**'last\_price', aggfunc**=**'mean').reset\_index())

price\_by\_city\_center

Out[46]:

|  | **cityCenters\_nearest\_km** | **last\_price** |
| --- | --- | --- |
| **0** | 0.0 | 3.144912e+07 |
| **1** | 1.0 | 2.099337e+07 |
| **2** | 2.0 | 1.639876e+07 |
| **3** | 3.0 | 1.089569e+07 |
| **4** | 4.0 | 1.211507e+07 |
| **5** | 5.0 | 1.316289e+07 |
| **6** | 6.0 | 1.475988e+07 |
| **7** | 7.0 | 1.470735e+07 |
| **8** | 8.0 | 9.251954e+06 |
| **9** | 9.0 | 6.950659e+06 |
| **10** | 10.0 | 6.506353e+06 |
| **11** | 11.0 | 6.252387e+06 |
| **12** | 12.0 | 5.818162e+06 |
| **13** | 13.0 | 6.084458e+06 |
| **14** | 14.0 | 5.630439e+06 |
| **15** | 15.0 | 5.857246e+06 |
| **16** | 16.0 | 5.374296e+06 |
| **17** | 17.0 | 5.267783e+06 |
| **18** | 18.0 | 5.091331e+06 |
| **19** | 19.0 | 5.037675e+06 |
| **20** | 20.0 | 5.905662e+06 |
| **21** | 21.0 | 5.493469e+06 |
| **22** | 22.0 | 5.415524e+06 |
| **23** | 23.0 | 4.985376e+06 |
| **24** | 24.0 | 3.854430e+06 |
| **25** | 25.0 | 4.052840e+06 |
| **26** | 26.0 | 3.973013e+06 |
| **27** | 27.0 | 8.300000e+06 |
| **28** | 28.0 | 5.026429e+06 |
| **29** | 29.0 | 4.236667e+06 |

Ввод [57]:

j**=**price\_by\_city\_center.pivot\_table(index**=**'cityCenters\_nearest\_km', values **=** 'last\_price')

display(j)

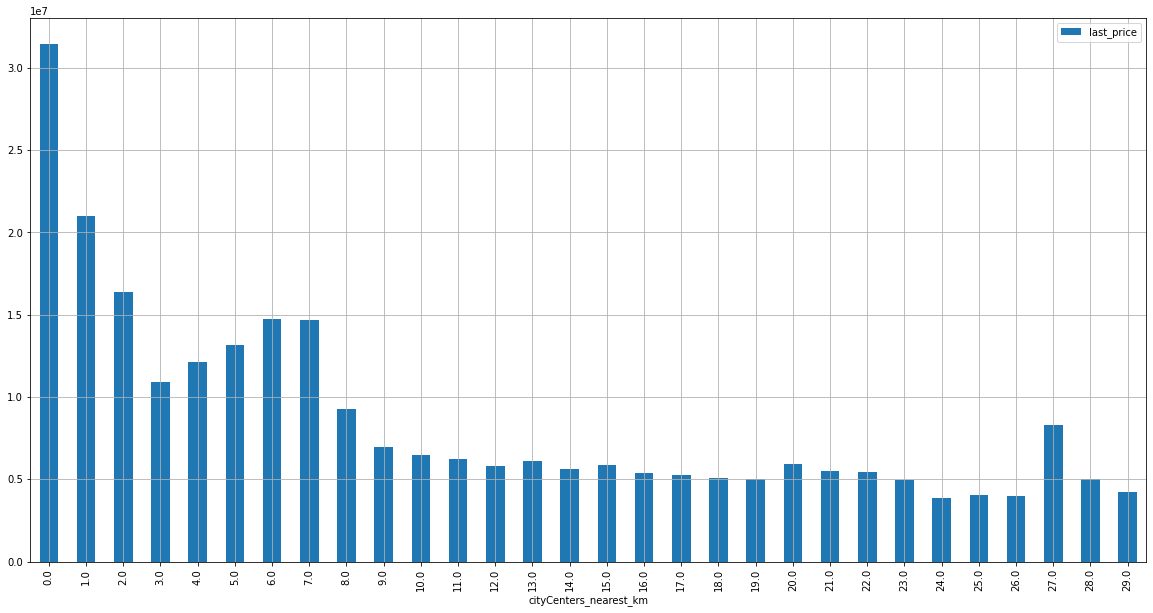
j.plot.bar(figsize**=**(20, 10), grid**=True**)

​

|  | **last\_price** |
| --- | --- |
| **cityCenters\_nearest\_km** |  |
| **0.0** | 3.144912e+07 |
| **1.0** | 2.099337e+07 |
| **2.0** | 1.639876e+07 |
| **3.0** | 1.089569e+07 |
| **4.0** | 1.211507e+07 |
| **5.0** | 1.316289e+07 |
| **6.0** | 1.475988e+07 |
| **7.0** | 1.470735e+07 |
| **8.0** | 9.251954e+06 |
| **9.0** | 6.950659e+06 |
| **10.0** | 6.506353e+06 |
| **11.0** | 6.252387e+06 |
| **12.0** | 5.818162e+06 |
| **13.0** | 6.084458e+06 |
| **14.0** | 5.630439e+06 |
| **15.0** | 5.857246e+06 |
| **16.0** | 5.374296e+06 |
| **17.0** | 5.267783e+06 |
| **18.0** | 5.091331e+06 |
| **19.0** | 5.037675e+06 |
| **20.0** | 5.905662e+06 |
| **21.0** | 5.493469e+06 |
| **22.0** | 5.415524e+06 |
| **23.0** | 4.985376e+06 |
| **24.0** | 3.854430e+06 |
| **25.0** | 4.052840e+06 |
| **26.0** | 3.973013e+06 |
| **27.0** | 8.300000e+06 |
| **28.0** | 5.026429e+06 |
| **29.0** | 4.236667e+06 |

Out[57]:

<AxesSubplot:xlabel='cityCenters\_nearest\_km'>



Чем ближе к центру города тем выше цена за объект и чаще выбросы по цене (возможно это связано с увеличением количества объектов со статусом "памятник культуры", тк цены на них не регулируются стандартами ценообразования).

**5  Общий вывод**

В ходе работы на проектом мною были изучены и обработаны данные, а именно: заменены пропуски в ceiling\_height медианным значением, тк оно менее всего искажает данные, которые могут понадобиться мне для дальнейших рассчетов; заполнены пропуски в столбце balcony значением ноль, тк возможно у объекта просто не балкона; заполнены пропуски в столбце living\_area значением ноль, тк возможно объект - это офисное помещение и площадь в нем просто не считается жилой; Заменены пропуски в столбце is\_apartment на значение unknown, тк скорее всего собственник не знает статус объекта (в новых домах например могут располагаться как квартиры студии так и апартаменты - иногда сложно разобраться). Так же рассмотрены и заменены (по необходимости типы данных. Изучены уникальные значения в столбце с названиями населенных пунктов и устранены неявные дубликаты. Найдены и устранены редкие и выбивающиеся значения. Добавлены новые столбцы. Проведен исследовательский анализ параметров объектов. Изучена скорость продажи квартир. Изучены зависимости цены квартиры от таких параметров как: Так же исследована зависимость цены квартиры от общей площади, жилой площади, площади кухни, количества комнат, этажа, на котором расположена квартира (первый, последний, другой),даты размещения (день недели, месяц, год).

Подводя итоги проведенного иследования можно сказать, что в центре Санкт-Петербурга приобладают квартиры с большей площадью, высотой потолков и стоимостью за квадратный метр, тк это в основном историческая застройка; дальше от центра города число объектов недвижимости со стандартными параметрами увеличивается (снижается метраж/высота потолка/количество балконов/цена за квадратный метр). Так же исходя из данных по времени экспозиции можно сделать вывод, что она достигает наибольшего значения для квартир в центре. Больше всего влияют на стоимость объектов близость к центру города и площадь квартир, чуть меньше этаж и количество комнат. Большим спросом пользуются квартиры находящиеся на любом этаже, кроме первого и последнего.

Могу предложить коллегам так же обратить внимание на изменение стоимости на объект в зависимости от его близисти к аэропотру.