Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА _____СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Анализ объявлений о продаже квартир

Студент __ИУ5-52Б_____ _____ ___**Пустовалов Г.В.**____
(Группа) (Подпись, дата). (И.О.Фамилия)

Руководитель научно-исследовательской работы _____ ___**К. Ю. Масленников**___
(Подпись, дата). (И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ___ИУ5____
(Индекс)
_____ ___В.И. Терехов__
(И.О.Фамилия)
«_04_» ____сентября_____ 2023 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по дисциплине ___Оперативный анализ данных _____

Студент группы _ИУ5-52Б__

_____Пустовалов Григорий Владимирович_____
(Фамилия, имя, отчество)

Тема научно-исследовательской работы Проектирования операционного устройства и его составляющих – операционного автомата и управляющего устройства

Направленность КР (учебная, исследовательская, практическая, производственная, др.) _____УЧЕБНАЯ_

Источник тематики (кафедра, предприятие, НИР) _____КАФЕДРА_

График выполнения работы: 25% к __ нед., 50% к __ нед., 75% к __ нед., 100% к __ нед.

Задание _____Спроектировать операционное устройство и его составляющие – операционный автомат и управляющее устройство

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _____ листах формата А4.

Дата выдачи задания «_04_» ____сентября_____ 2023 г.

Руководитель научно-исследовательской работы _____ ____К. Ю.Масленников_____
(Подпись, дата) (И.О.Фамилия)

Студент _____ _____Г.В. Пустовалов____
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Введение

Необходимо исследовать базу данных и выявить закономерности в стоимости квартир.

Цели:

1) определение данных;
2) формулирование гипотез;
3) загрузка данных в Python;
4) проверка данных;
5) очистка данных;
6) преобразование данных;
7) выбор данных для анализа;
8) агрегирование данных;
9) визуализация данных;
10) подтверждение или опровержение поставленных гипотез;
11) формулирование ограничений и выводов.

1. Определение данных для анализа

В качестве данных для анализа был выбран датасет «объявлений о продаже квартир»

Сейчас покупка, продажа и аренда квартиры стали тяжелым процессом для каждого человека. Особенно это касается крупного города, где особенно сильно варьируются варианты: как по ценам, так и по инфраструктуре.

Учитывая это, было сделано решение выбрать этот датасет. В датасете представлены данные о квартирах в Санкт-Петербурге и Ленинградской области: их стоимости, площади, близости парков и множестве других параметров.

1. Описание данных

Для анализа были собраны данные о продаваемых в 2018 году в Москве квартирах. В наборе данных содержатся:

1) **total_images** – количество изображний в объявлении
2) **last_price** – последняя цена, указанная в объявлении
3) **total_area**– общая площадь квартиры, кв.м.
4) **first_day_exposition** – день, в который объявление было выложено.
5) **rooms**– количество комнат
6) **ceiling_height**– высота потолков
7) **floors_total** – количество этажей в доме
8) **living_area**– жилая площадь квартиры, кв.м.
9) **floor –** этаж
10) **is_apartament –** является ли апартаментами
11) **kitchen_area –** кухня площадь, кв.м.
12) **locality_name –** название населенного пункта
13) **cityCenter_nearest –** расстояние до центра
14) **parks_nearest –** ближайший парк
15) **days_exposition –** количество дней объявлению

1. **Формулирование гипотез**

В ходе первичного анализа были выдвинуты следующие гипотезы:
Гипотеза 1: чем больше комнат в квартире, тем она дороже.
Гипотеза 2: стоимость квадратного метра увеличивается каждый год
Гипотеза 3: чем ближе квартира к центру города, тем выше ее стоимость квадратного метра.
Гипотеза 4: если квартира находится в радиусе 9км от центра, то на стоимость квартиры не влияет расстояние до центра.

1. Изучение общей информации

Загружаем датасет, подключаем необходимые библиотеки:

```
import pandas as pd
```

Откроем файл с данными.

```
df = pd.read_csv('/real_estate_data.csv', sep='\t')

df.head(10)
```

	total_images	last_price	total_area	first_day_exposition	rooms	ceiling_height	floors_total	living_area	floor	is_apartment	...	kitchen_area	balcony
0	20	13000000.0	108.00	2019-03-07T00:00:00	3	2.70	16.0	51.00	8	NaN	...	25.00	NaN
1	7	3350000.0	40.40	2018-12-04T00:00:00	1	NaN	11.0	18.60	1	NaN	...	11.00	2.0
2	10	5196000.0	56.00	2015-08-20T00:00:00	2	NaN	5.0	34.30	4	NaN	...	8.30	0.0
3	0	64900000.0	159.00	2015-07-24T00:00:00	3	NaN	14.0	NaN	9	NaN	...	NaN	0.0
4	2	10000000.0	100.00	2018-06-19T00:00:00	2	3.03	14.0	32.00	13	NaN	...	41.00	NaN
5	10	2890000.0	30.40	2018-09-10T00:00:00	1	NaN	12.0	14.40	5	NaN	...	9.10	NaN
6	6	3700000.0	37.30	2017-11-02T00:00:00	1	NaN	26.0	10.60	6	NaN	...	14.40	1.0
7	5	7915000.0	71.60	2019-04-18T00:00:00	2	NaN	24.0	NaN	22	NaN	...	18.90	2.0
8	20	2900000.0	33.16	2018-05-23T00:00:00	1	NaN	27.0	15.43	26	NaN	...	8.81	NaN
9	18	5400000.0	61.00	2017-02-26T00:00:00	3	2.50	9.0	43.60	7	NaN	...	6.50	2.0

Получим информацию о датасете:

```
df.info()
```

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

Проверим на дубликаты.

```
df.duplicated().sum()
```

```
df.duplicated().sum()

0
```

Займемся пропусками. Пропущенные значения количества балконов заменим на 0, так как, если человек не указал число балконов — скорее всего, их нет.

```
df['balcony'] = df['balcony'].fillna(0)
df['balcony'].isna().sum()
```

Аналогично поступим с количеством парков и прудов в радиусе 3 000 м. Если поле осталось пустым, скорее всего их просто нет.

```
df['parks_around_3000'] = df['parks_around_3000'].fillna(0)
df['ponds_around_3000'] = df['ponds_around_3000'].fillna(0)
```

Посмотрим, какие населенные пукты представленны.

```
df['locality_name'].unique()
```

```
array(['Санкт-Петербург', 'посёлок Шушары', 'городской посёлок Янино-1',
       'посёлок Парголово', 'посёлок Мурино', 'Ломоносов', 'Сертолово',
       'Петергоф', 'Пушкин', 'деревня Кудрово', 'Коммунар', 'Колпино',
       'поселок городского типа Красный Бор', 'Гатчина', 'поселок Мурино',
       'деревня Фёдоровское', 'Выборг', 'Кронштадт', 'Кировск',
       'деревня Новое Девяткино', 'посёлок Металлострой',
       'посёлок городского типа Лебяжье',
       'посёлок городского типа Сиверский', 'поселок Молодцово',
       'поселок городского типа Кузьмоловский',
       'садовое товарищество Новая Ропша', 'Павловск',
       'деревня Пикколово', 'Всеволожск', 'Волхов', 'Кингисепп',
       'Приозерск', 'Сестрорецк', 'деревня Куттузи', 'посёлок Аннино',
       'поселок городского типа Ефимовский', 'посёлок Плодовое',
       'деревня Заклинье', 'поселок Торковичи', 'поселок Первомайское',
       'Красное Село', 'посёлок Понтонный', 'Сясьстрой', 'деревня Старая',
       'деревня Лесколово', 'посёлок Новый Свет', 'Сланцы',
       'село Путилово', 'Ивангород', 'Мурино', 'Шлиссельбург',
       'Никольское', 'Зеленогорск', 'Сосновый Бор', 'поселок Новый Свет',
       'деревня Оржицы', 'деревня Кальтино', 'Кудрово',
       'поселок Романовка', 'посёлок Бугры', 'поселок Бугры',
       'поселок городского типа Рощино', 'Кириши', 'Луга', 'Волосово',
       'Отрадное', 'село Павлово', 'поселок Оредеж', 'село Копорье',
       'посёлок городского типа Красный Бор', 'посёлок Молодёжное',
       'Тихвин', 'посёлок Победа', 'деревня Нурма',
       'поселок городского типа Синявино', 'Тосно',
       'посёлок городского типа Кузьмоловский', 'посёлок Стрельна',
       'Бокситогорск', 'посёлок Александровская', 'деревня Лопухинка',
       'Пикалёво', 'поселок Терволово',
       'поселок городского типа Советский', 'Подпорожье',
       'посёлок Петровское', 'посёлок городского типа Токсово',
       'поселок Сельцо', 'посёлок городского типа Вырица',
       'деревня Кипень', 'деревня Келози', 'деревня Вартемяги',
       'посёлок Тельмана', 'поселок Севастьяново',
       'городской поселок Большая Ижора', nan,
```

Слово "Поселок" записано по-разному, через буквы "е" и "ё". Заменим все буквы "ё" на "е".

```
len(df['locality_name'].unique())
```

365

```
df['locality_name'] = df['locality_name'].str.replace('ё', 'е')
len(df['locality_name'].unique())
```

Сделаем формат столбцов "balcony" и "last_price" целочисленным, так удобнее воспринимать информацию из них. Есть ещё столбцы, которые можно было бы сделать целочисленным, но они имеют пропуски, которые нельзя заполнить логически.

```
df['balcony'] = df['balcony'].astype('int')
df['last_price'] = df['last_price'].astype('int')
df.head()
```

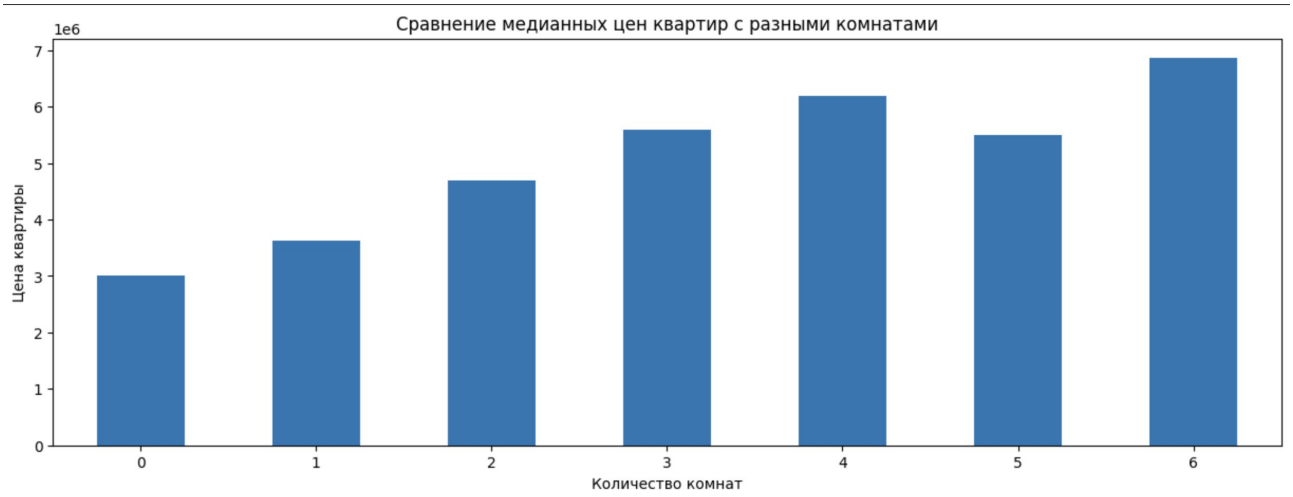
Добавим в таблицу стобцы со значением цены за квадратный метр "price_per_m2" и годом "year" размещения объявления.

```
df['price_per_m2'] = df['last_price'] / df['total_area']
df['year'] = df['first_day_exposition'].dt.year
df.head()
```

1. Исследовательский анализ данных

1.1. Сравнение медианных цен с разными комнатами

```
df.groupby('rooms')['last_price'].median()\
    .plot(kind = 'bar',\
        figsize = (15,5),\
        title = 'Сравнение медианных цен квартир с разными комнатами')
plt.xticks(rotation = 0)
plt.xlabel('Количество комнат')
plt.ylabel('Цена квартиры')
plt.show()
```



Гипотеза 1: соответствует действительности

1.1. Зависимость стоимости квадратного метра от года

Гипотеза 2: возможно, в 2015 году был кризис недвижимости

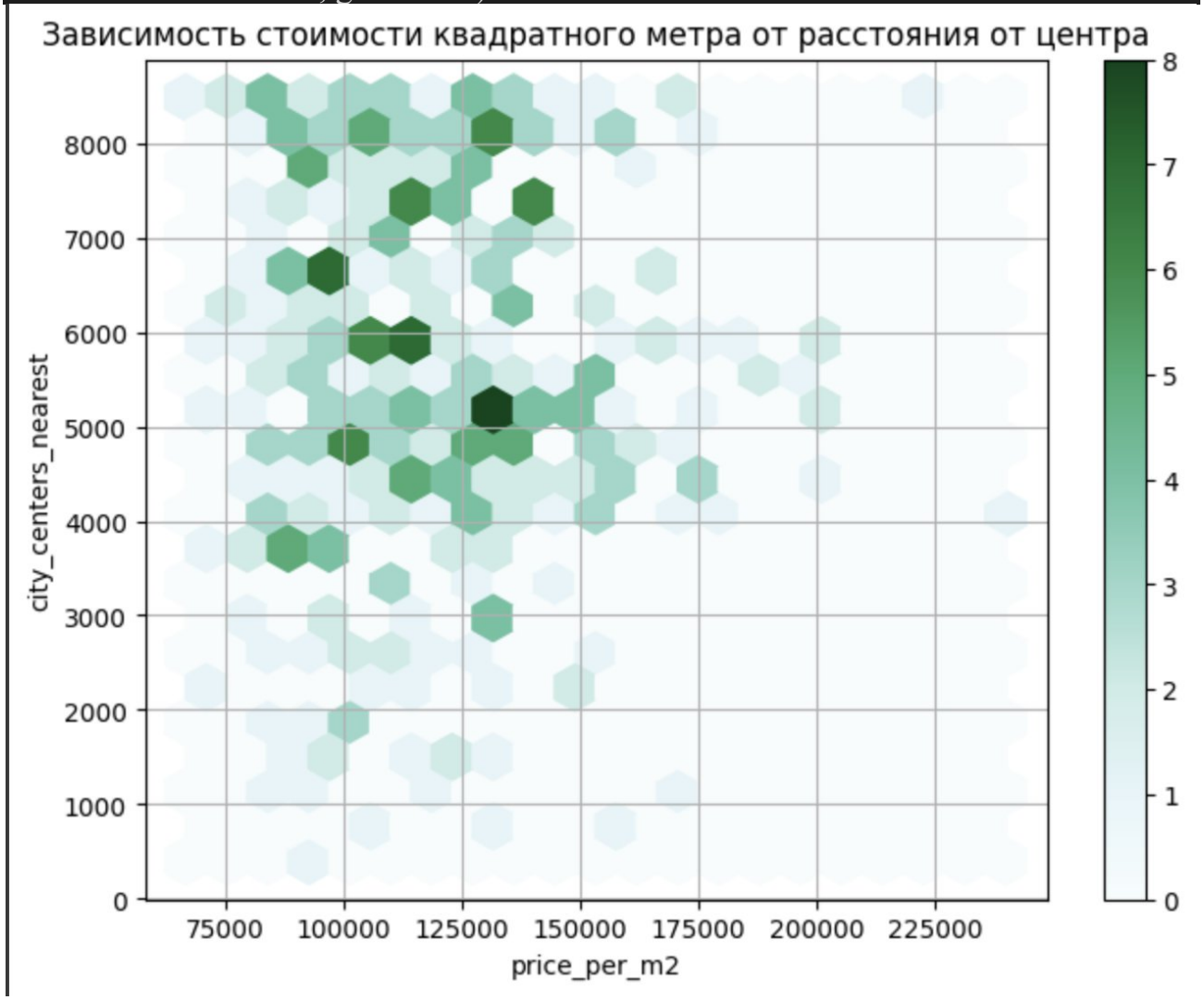
1.1. Зависимость стоимости квадратного метра от расстояния до центра

Гипотеза 3: на цену квартиры также влияет близость метро и прочих благ жизни.

Проанализируем сегмент квартир в центре.

```
df_center = df.query('city_centers_nearest_km < 9')
df_center.head()
```

```
df_center.plot(x='price_per_m2',
               y='city_centers_nearest',
               title='Зависимость стоимости квадратного метра от расстояния от центра',
               kind='hexbin',
               gridsize=20,
               figsize=(8, 6),
               sharex=False, grid=True)
```



Гипотеза 4: соответствует действительности