

Проектная работа

Рынок заведений общественного питания Москвы

Цель исследования:

- Исследовать рынок общественного питания в Москве.
- Презентовать результаты, которые в будущем помогут в выборе подходящего инвесторам места.

Описание данных:

Доступен датасет с заведениями общественного питания Москвы, составленный на основе данных сервисов Яндекс Карты и Яндекс Бизнес на лето 2022 года. **Файл `moscow_places.csv`:** \ • `name` — название заведения; \ • `address` — адрес заведения; \ • `category` — категория заведения, например «кафе», «пиццерия» или «кофейня»; \ • `hours` — информация о днях и часах работы; \ • `lat` — широта географической точки, в которой находится заведение; \ • `lng` — долгота географической точки, в которой находится заведение; \ • `rating` — рейтинг заведения по оценкам пользователей в Яндекс Картах (высшая оценка — 5.0); \ • `price` — категория цен в заведении, например «средние», «ниже среднего», «выше среднего» и так далее; \ • `avg_bill` — строка, которая хранит среднюю стоимость заказа в виде диапазона, например: \ ◦ «Средний счёт: 1000–1500 ₽»; \ ◦ «Цена чашки капучино: 130–220 ₽»; \ ◦ «Цена бокала пива: 400–600 ₽». \ и так далее; \ • `middle_avg_bill` — число с оценкой среднего чека, которое указано только для значений из столбца `avg_bill`, начинающихся с подстроки «Средний счёт»: \ ◦ Если в строке указан ценовой диапазон из двух значений, в столбец войдёт медиана этих двух значений. \ ◦ Если в строке указано одно число — цена без диапазона, то в столбец войдёт это число. \ ◦ Если значения нет или оно не начинается с подстроки «Средний счёт», то в столбец ничего не войдёт. \ • `middle_coffee_cup` — число с оценкой одной чашки капучино, которое указано только для значений из столбца `avg_bill`, начинающихся с подстроки «Цена одной чашки капучино»: \ ◦ Если в строке указан ценовой диапазон из двух значений, в столбец войдёт медиана этих двух значений. \ ◦ Если в строке указано одно число — цена без диапазона, то в столбец войдёт это число. \ ◦ Если значения нет или оно не начинается с подстроки «Цена одной чашки капучино», то в столбец ничего не войдёт. \ • `chain` — число, выраженное 0 или 1, которое показывает, является ли заведение сетевым (для маленьких сетей могут встречаться ошибки): \ ◦ 0 — заведение не является сетевым \ ◦ 1 — заведение является сетевым \ • `district` — административный район, в котором находится заведение, например Центральный административный округ; \ • `seats` — количество посадочных мест. \

План работы:

Шаг 1. Загрузить данные и изучить общую информацию

Шаг 2. Выполнить предобработку данных

Изучите, есть ли дубликаты в данных. Поищите пропуски: встречаются ли они, в каких столбцах? Можно ли их обработать или оставить как есть?

Выполните предобработку данных:

- Создайте столбец `street` с названиями улиц из столбца с адресом.
- Создайте столбец `is_24/7` с обозначением, что заведение работает ежедневно и круглосуточно (24/7):
 - логическое значение `True` — если заведение работает ежедневно и круглосуточно;

- логическое значение False – в противоположном случае.

Шаг 3. Анализ данных

- Какие категории заведений представлены в данных? Исследуйте количество объектов общественного питания по категориям: рестораны, кофейни, пиццерии, бары и так далее. Постройте визуализации. Ответьте на вопрос о распределении заведений по категориям.
- Исследуйте количество посадочных мест в местах по категориям: рестораны, кофейни, пиццерии, бары и так далее. Постройте визуализации. Проанализируйте результаты и сделайте выводы.
- Рассмотрите и изобразите соотношение сетевых и несетевых заведений в датасете. Каких заведений больше?
- Какие категории заведений чаще являются сетевыми? Исследуйте данные и ответьте на вопрос графиком.
- Сгруппируйте данные по названиям заведений и найдите топ-15 популярных сетей в Москве. Под популярностью понимается количество заведений этой сети в регионе. Постройте подходящую для такой информации визуализацию. Знакомы ли вам эти сети? Есть ли какой-то признак, который их объединяет? К какой категории заведений они относятся?
- Какие административные районы Москвы присутствуют в датасете? Отобразите общее количество заведений и количество заведений каждой категории по районам. Попробуйте проиллюстрировать эту информацию одним графиком.
- Визуализируйте распределение средних рейтингов по категориям заведений. Сильно ли различаются усреднённые рейтинги в разных типах общепита?
- Постройте фоновую картограмму (хороплет) со средним рейтингом заведений каждого района. Границы районов Москвы, которые встречаются в датасете, хранятся в файле `admin_level_geomap.geojson` (скачать файл для локальной работы).
- Отобразите все заведения датасета на карте с помощью кластеров средствами библиотеки `folium`.
- Найдите топ-15 улиц по количеству заведений. Постройте график распределения количества заведений и их категорий по этим улицам. Попробуйте проиллюстрировать эту информацию одним графиком.
- Найдите улицы, на которых находится только один объект общепита. Что можно сказать об этих заведениях?
- Значения средних чеков заведений хранятся в столбце `middle_avg_bill`. Эти числа показывают примерную стоимость заказа в рублях, которая чаще всего выражена диапазоном. Посчитайте медиану этого столбца для каждого района. Используйте это значение в качестве ценового индикатора района. Постройте фоновую картограмму (хороплет) с полученными значениями для каждого района. Проанализируйте цены в центральном административном округе и других. Как удалённость от центра влияет на цены в заведениях?
- Необязательное задание: проиллюстрируйте другие взаимосвязи, которые вы нашли в данных. Например, по желанию исследуйте часы работы заведений и их зависимость от расположения и категории заведения. Также можно исследовать особенности заведений с плохими рейтингами, средние чеки в таких местах и распределение по категориям заведений.
- Соберите наблюдения по вопросам выше в один общий вывод.

Шаг 4. Детализация исследования: открытие кофейни

Основателям фонда «Shut Up and Take My Money» не даёт покоя успех сериала «Друзья». Их мечта — открыть такую же крутую и доступную, как «Central Perk», кофейню в Москве. Будем считать, что заказчики не боятся конкуренции в этой сфере, ведь кофеен в больших городах уже достаточно. Попробуйте определить, осуществима ли мечта клиентов. Ответьте на следующие вопросы:

- Сколько всего кофеен в датасете? В каких районах их больше всего, каковы особенности их расположения?
- Есть ли круглосуточные кофейни?
- Какие у кофеен рейтинги? Как они распределяются по районам?

- На какую стоимость чашки капучино стоит ориентироваться при открытии и почему?

По желанию вы можете расширить список вопросов для исследования, добавив собственные. Постройте визуализации. Попробуйте дать рекомендацию для открытия нового заведения. Это творческое задание: здесь нет правильного или неправильного ответа, но ваше решение должно быть чем-то обосновано. Объяснить свою рекомендацию можно текстом с описанием или маркерами на географической карте.

Шаг 5. Подготовка презентации

Подготовьте презентацию исследования для инвесторов. Отвечая на вопросы о московском общепите, вы уже построили много диаграмм, и помещать каждую из них в презентацию не нужно. Выберите важные тезисы и наблюдения, которые могут заинтересовать заказчиков. Для создания презентации используйте любой удобный инструмент, но отправить презентацию нужно обязательно в формате PDF. Приложите ссылку на презентацию в markdown-ячейке в формате: Презентация: <ссылка на облачное хранилище с презентацией> Следуйте принципам оформления из темы «Подготовка презентации».

Оглавление:

Загрузка данных:

Шаг 1. Загрузите данные и изучите общую информацию Загрузите данные о заведениях общественного питания Москвы.

```
In [1]: import pandas as pd
import numpy as np
import plotly.express as px
from plotly import graph_objects as go
from plotly.offline import iplot
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('/datasets/moscow_places.csv')
```

```
In [3]: display(df)
```

	name	category	address	district	hours	lat	lng	rating	pr
0	WoWfli	кафе	Москва, улица Дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	N
1	Четыре комнаты	ресторан	Москва, улица Дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	4.5	высший средний
2	Хазри	кафе	Москва, Клязьминская улица, 15	Северный административный округ	пн-чт 11:00–02:00; пт,сб 11:00–05:00; вс 11:00–...	55.889146	37.525901	4.6	средний
3	Dormouse Coffee Shop	кофейня	Москва, улица Маршала Федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	5.0	N
4	Иль Марко	пиццерия	Москва, Правобережная улица, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средний
...
8401	Суши Мания	кафе	Москва, Профсоюзная	Юго-Западный административный	ежедневно, 09:00–02:00	55.670021	37.552480	4.4	N

	name	category	address	district	hours	lat	lng	rating	pr
			улица, 56	округ					
8402	Миславнес	кафе	Москва, Пролетарский проспект, 19, корп. 1	Южный административный округ	ежедневно, 08:00–22:00	55.640875	37.656553	4.8	N
8403	Самовар	кафе	Москва, Люблинская улица, 112А, стр. 1	Юго-Восточный административный округ	ежедневно, круглосуточно	55.648859	37.743219	3.9	N
8404	Чайхана Sabr	кафе	Москва, Люблинская улица, 112А, стр. 1	Юго-Восточный административный округ	ежедневно, круглосуточно	55.648849	37.743222	4.2	N
8405	Kebab Time	кафе	Москва, Россошанский проезд, 6	Южный административный округ	ежедневно, круглосуточно	55.598229	37.604702	3.9	N

8406 rows × 14 columns

```
In [4]: # Изучим информацию о датасете:
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8406 entries, 0 to 8405
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   name                   8406 non-null   object
1   category               8406 non-null   object
2   address                8406 non-null   object
3   district               8406 non-null   object
4   hours                  7870 non-null   object
5   lat                    8406 non-null   float64
6   lng                    8406 non-null   float64
7   rating                 8406 non-null   float64
8   price                  3315 non-null   object
9   avg_bill               3816 non-null   object
10  middle_avg_bill        3149 non-null   float64
11  middle_coffee_cup      535 non-null    float64
12  chain                   8406 non-null   int64
13  seats                  4795 non-null   float64
dtypes: float64(6), int64(1), object(7)
memory usage: 919.5+ KB
```

```
In [5]: df.describe()
```

	lat	lng	rating	middle_avg_bill	middle_coffee_cup	chain	seats
count	8406.000000	8406.000000	8406.000000	3149.000000	535.000000	8406.000000	4795.000000
mean	55.750109	37.608570	4.229895	958.053668	174.721495	0.381275	108.421689
std	0.069658	0.098597	0.470348	1009.732845	88.951103	0.485729	122.833396
min	55.573942	37.355651	1.000000	0.000000	60.000000	0.000000	0.000000
25%	55.705155	37.538583	4.100000	375.000000	124.500000	0.000000	40.000000
50%	55.753425	37.605246	4.300000	750.000000	169.000000	0.000000	75.000000
75%	55.795041	37.664792	4.400000	1250.000000	225.000000	1.000000	140.000000
max	55.928943	37.874466	5.000000	35000.000000	1568.000000	1.000000	1288.000000

```
In [6]: print('В датасете представлено', len(df['category'].unique()), 'категорий заведений')
```

В датасете представлено 8 категорий заведений

```
In [7]: print('Количество уникальных заведений = ', len(df['name']))
```

Количество уникальных заведений = 8406

Выводы:

- Таблица содержит информацию об 8406 заведениях, которые относятся к 8 категориям.
- Медианное значение количества посадочных заведений равно 75, среднее - 108
- Средний рейтинг исследуемых заведений в Москве - 4.2, медианный - 4.3
- Типы данных соответствуют содержимому столбцов.
- В таблице есть пропуски. Подробнее рассмотрим ситуацию с пропусками в следующей части "Предобработка данных".

Предобработка данных

Изучите, есть ли дубликаты в данных. Поищите пропуски: встречаются ли они, в каких столбцах? Можно ли их обработать или оставить как есть? Выполните предобработку данных:

- Создайте столбец `street` с названиями улиц из столбца с адресом.
- Создайте столбец `is_24/7` с обозначением, что заведение работает ежедневно и круглосуточно (24/7):

- логическое значение `True` — если заведение работает ежедневно и круглосуточно;
- логическое значение `False` — в противоположном случае.

Дубликаты в данных:

Встречаются ли дубликаты в данных?

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

Out[8]: 0

Полных дубликатов не обнаружено.

```
In [9]: #Поиск дубликатов по столбцам name category address:
df.duplicated(subset = ['name', 'category', 'address']).sum()
```

Out[9]: 0

В исследуемом датасете полные дубликаты не были обнаружены

```
In [10]: # Поиск неявных дубликатов:
# приводим столбец к нижнему регистру
df['name'] = df['name'].str.lower()
df['address'] = df['address'].str.lower()
#Удаление неявных дубликатов:
df.duplicated(subset=['name', 'address']).sum()
print(len(df))
df = df.drop_duplicates(subset=['name', 'address'])
print(len(df))
```

8406

8402

Пропуски в данных:

Встречаются ли пропуски в данных? в каких столбцах? Можно ли их обработать или оставить как есть?

```
In [11]: gaps = df.isna().sum()
print(gaps)
```

```
name          0
category      0
```

```

address          0
district         0
hours            535
lat              0
lng              0
rating           0
price            5087
avg_bill         4586
middle_avg_bill  5253
middle_coffee_cup 7867
chain            0
seats            3610
dtype: int64

```

Пропуски имеются в следующих столбцах:

- hours
- price
- avg_bill
- middle_avg_bill
- middle_coffee_cup
- seats

Редко когда бывает необходимо заполнить пропуски медианными значениями моды. Такой подход к обработке данных может заметно исказить результаты анализа данных. Поэтому все пропуски оставляем так, как есть, и ничего не трогаем.

Создание столбцов:

- Создайте столбец `street` с названиями улиц из столбца с адресом.
- Создайте столбец `is_24/7` с обозначением, что заведение работает ежедневно и круглосуточно (24/7):
 - логическое значение `True` — если заведение работает ежедневно и круглосуточно;
 - логическое значение `False` — в противоположном случае.

Создадим столбец `street` с названиями улиц из столбца с адресом:

```

In [12]: df['street'] = df['address'].str.split(',', expand=True)[1]
df['street'] = df['street'].str.strip()

```

Создадим столбец `is_24/7` с обозначением, что заведение работает ежедневно и круглосуточно (24/7):

- логическое значение `True` — если заведение работает ежедневно и круглосуточно;
- логическое значение `False` — в противоположном случае.

```

In [13]: df['works24/7'] = df['hours'].str.contains('ежедневно, круглосуточно')
# преобразование значений в логические
df['works24/7'] = df['works24/7'].astype(bool)
df.columns

```

```

Out[13]: Index(['name', 'category', 'address', 'district', 'hours', 'lat', 'lng',
               'rating', 'price', 'avg_bill', 'middle_avg_bill', 'middle_coffee_cup',
               'chain', 'seats', 'street', 'works24/7'],
              dtype='object')

```

```

In [14]: # Проверяем:
display(df.head(5))

```

	name	category	address	district	hours	lat	lng	rating	price
0	wowфли	кафе	москва, улица дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN

	name	category	address	district	hours	lat	lng	rating	price
1	четыре комнаты	ресторан	москва, улица дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	4.5	выше среднего
2	хазри	кафе	москва, клязьминская улица, 15	Северный административный округ	пн-чт 11:00–02:00; пт,сб 11:00–05:00; вс 11:00–11:00...	55.889146	37.525901	4.6	средние
3	dormouse coffee shop	кофейня	москва, улица маршала федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	5.0	NaN
4	иль марко	пиццерия	москва, правобережная улица, 16	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние

Вывод:

- Осуществили поиск дубликатов. Удалили неявные дубликаты.
- Нашли пропуски в датасете и решили оставить их без изменения.
- Создали столбец street с названиями улиц из столбца с адресом.
- Создали столбец works_24/7 с обозначением, что заведение работает ежедневно и круглосуточно (24/7):

Анализ данных

Какие категории заведений представлены в данных?

- Исследуйте количество объектов общественного питания по категориям: рестораны, кофейни, пиццерии, бары и так далее.
- Постройте визуализации. Ответьте на вопрос о распределении заведений по категориям.

```
In [15]: print('В датасете представлено', len(df['category'].unique()), 'категорий заведений')
```

В датасете представлено 8 категорий заведений

```
In [16]: print('В данных представлены следующие категории заведений:', df['category'].unique())
```

В данных представлены следующие категории заведений: ['кафе' 'ресторан' 'кофейня' 'пиццерия' 'бар, паб' 'быстрое питание' 'булочная' 'столовая']

```
In [17]: category = df['category'].value_counts().to_frame(name='values').reset_index()
category['percent'] = round(category['values'] / sum(category['values']) * 100, 2)

display(category)
```

	index	values	percent
0	кафе	2376	28.28
1	ресторан	2042	24.30
2	кофейня	1413	16.82
3	бар,паб	764	9.09
4	пиццерия	633	7.53
5	быстрое питание	603	7.18
6	столовая	315	3.75

	index	values	percent
7	булочная	256	3.05

In [18]:

```
fig = px.bar(category, x='index', y='values', color="index", text='values')
fig.update_layout(title='Количество заведений по категориям',
                  xaxis_title='Категория',
                  yaxis_title='Количество заведений')
fig.show()
```



- Наиболее распространенным объектом является категория "кафе", таких заведений в исследуемых данных - 28,3%.
- Наименее распространены заведения, относящиеся к категории "булочная" - их всего 3%.
- Кафе, рестораны и кофейни вместе занимают достаточно весомую часть от совокупности всех заведений - 69,4(Почти 70%).

Количество посадочных мест в местах по категориям

рестораны, кофейни, пиццерии, бары и так далее. Постройте визуализации. Проанализируйте результаты и сделайте выводы.

In [19]:

```
seats = df.groupby('category')['seats'].describe().round(2).reset_index()
```

In [20]:

```
seats_median = df[df['seats'] != 0].groupby('category')['seats'].median().reset_index().sort_values
seats_median
```

Out[20]:

	category	seats
6	ресторан	90.0
0	бар,паб	84.0

	category	seats
4	кофейня	80.0
7	столовая	80.0
2	быстрое питание	75.0
3	кафе	60.0
5	пиццерия	56.0
1	булочная	52.0

In [21]:

```
fig = px.box(df, x="category", y="seats", color = 'category')
fig.update_layout(title='Количество посадочных мест по категориям, ',
                  xaxis_title='Категория',
                  yaxis_title='Количество посадочных мест')
fig.show()
```



В данном пункте мы рассмотрели распределение посадочных мест по категориям и выяснили:

- Как правило, максимальным количеством мест для посадки располагают бары, пабы, рестораны и кофейни.
- Меньше всего посадочных мест в столовых и булочных. Распределение посадочных мест имеет определенную закономерность: \ Бары, пабы, рестораны и кофейни часто посещают для довольно длительных встреч, бронируют столики заранее, выбирают наиболее комфортную обстановку, располагающую к общению. В таких заведениях платят не только за пищу, но и за интерьер, атмосферу, качественное обслуживание. \ Со столовыми, булочными и другими заведениями, ситуация обратная, они предназначены для быстрого потребления пищи. Владельцы таких заведений часто принимают различные меры для того, чтобы посетитель как можно меньше времени занимал свой столик: недостаточно удобная мебель, частично открытая кухня, минималистично оформленные интерьеры самого заведения и т.д. Это сделано для того, чтобы принять максимальное количество гостей, продавая им больше блюд.

Соотношение сетевых и несетевых заведений в датасете.

- Каких заведений больше?
- Какие категории заведений чаще являются сетевыми?
- Исследуйте данные и ответьте на вопрос графиком.

```
In [22]: # Сгруппируем данные по типу объекта и сетевому признаку, подсчитав количество заведений в каждой
chain_info = df.groupby(['category', 'chain'])['name'].count().reset_index()
# выделим несетевые заведения:
separate = chain_info.query ('chain == 0')
# выделим сетевые заведения:
chain = chain_info.query ('chain == 1')

#chain_info = chain_info['chain'].replace(['0'], ' Несетевые')
#chain_info = chain_info['chain'].replace(['1'], ' Сетевые')
chain_info['chain'] = chain_info['chain'].replace([0, 1], ['Несетевые', 'Сетевые'])
```

```
In [23]: chain_info
```

```
Out[23]:
```

	category	chain	name
0	бар,паб	Несетевые	596
1	бар,паб	Сетевые	168
2	булочная	Несетевые	99
3	булочная	Сетевые	157
4	быстрое питание	Несетевые	371
5	быстрое питание	Сетевые	232
6	кафе	Несетевые	1597
7	кафе	Сетевые	779
8	кофейня	Несетевые	693
9	кофейня	Сетевые	720
10	пиццерия	Несетевые	303
11	пиццерия	Сетевые	330
12	ресторан	Несетевые	1313
13	ресторан	Сетевые	729
14	столовая	Несетевые	227
15	столовая	Сетевые	88

```
In [24]: fig = px.pie(chain_info, values=chain_info['name'], names=chain_info['chain'], title='Категории заведений по типу объекта')
fig.update_layout(
    title=dict(
        text="Отношение сетевых и несетевых заведений в датасете",
        font=dict(size=24),
        x=0.5,
        xref="paper"
    )
)
fig.show()
```

Отношени

Круговая диаграмма наглядно иллюстрирует соотношение сетевых и несетевых заведений в датасете - 38,1% против 61,9%.

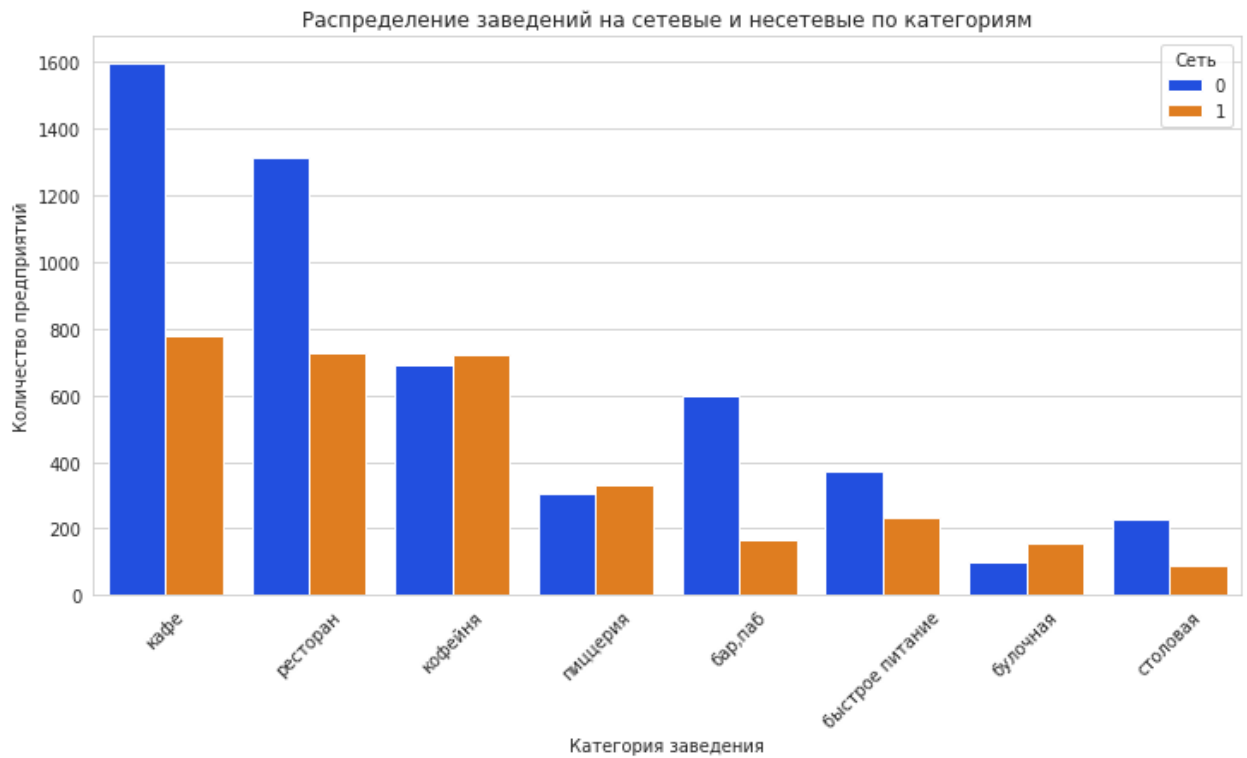
```
In [25]: for_plot = pd.pivot_table(chain_info,
                                   index = ['chain'],
                                   columns = ['category']).reset_index()
```

```
In [26]: display(for_plot)
```

		chain							name
category		бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
0	Несетевые	596	99	371	1597	693	303	1313	227
1	Сетевые	168	157	232	779	720	330	729	88

```
In [27]: sns.set_style('whitegrid')
sns.set_palette('bright')

plt.figure(figsize=(12,6))
sns.countplot(x='category', hue='chain', data=df)
plt.title('Распределение заведений на сетевые и несетевые по категориям')
plt.xlabel('Категория заведения')
plt.ylabel('Количество предприятий')
plt.xticks(rotation=45)
plt.legend(title='Сеть', loc='upper right')
plt.show()
# Напомним, что 0 - несетевые, 1 - сетевые
```



```
In [28]: category_count = df.pivot_table(index = 'category', values = 'name', aggfunc = 'count').sort_value
category_count
```

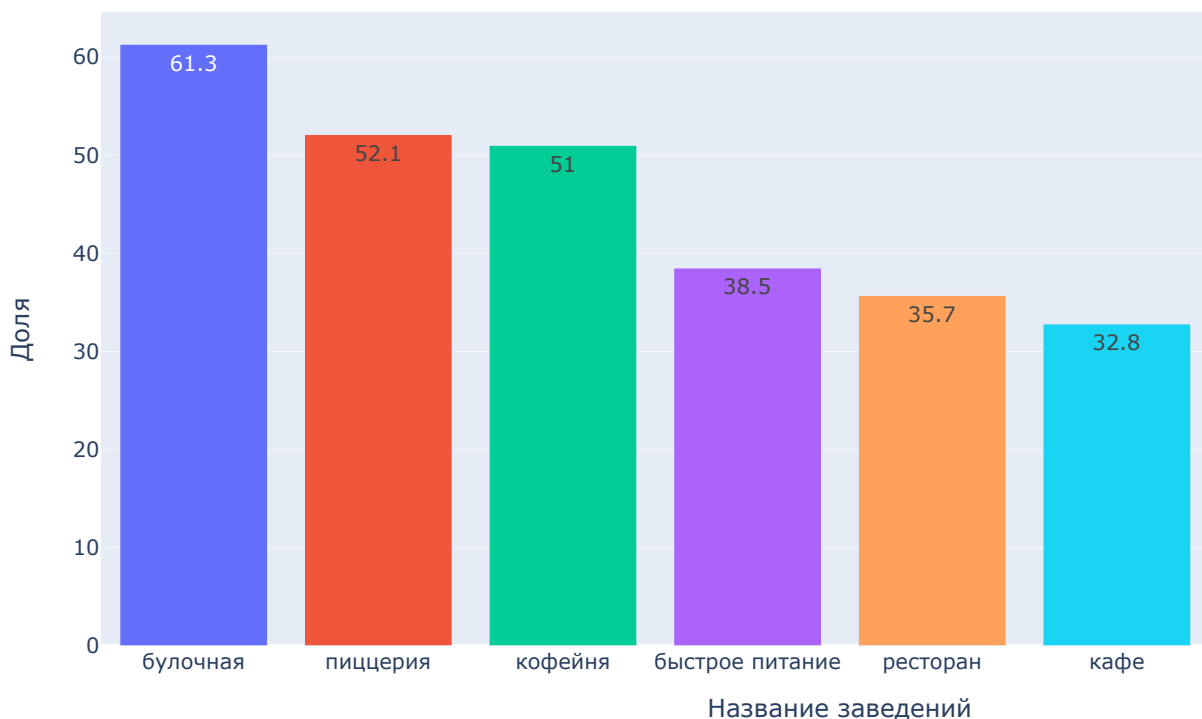
```
Out[28]:
```

category	name
кафе	2376
ресторан	2042
кофейня	1413
бар, паб	764
пиццерия	633
быстрое питание	603
столовая	315
булочная	256

```
In [29]: chain_category = df.query('chain == 1').pivot_table(index = 'category', values = 'name', aggfunc =
chain_category_merge = category_count.merge(chain_category, how='left', on='category').reset_index
chain_category_merge.columns = ['category', 'count', 'chain']
chain_category_merge['percent'] = round(chain_category_merge['chain']/chain_category_merge['count']
chain_category_merge = chain_category_merge.sort_values(by = 'percent', ascending = False)
```

```
In [30]: fig = px.bar(chain_category_merge,
                    x='category',
                    y='percent',
                    text='percent',
                    color='category',
                    height=500,
                    width=1000
                )
fig.update_layout(title = '% Сетевых заведений от общего количества заведений каждого типа',
                  xaxis_title = 'Название заведений',
                  yaxis_title = 'Доля',
                  showlegend = False)
fig.show()
```

% Сетевых заведений от общего количества заведений каждого типа



- В категории сетевых заведений наиболее часто оказываются булочные, пиццерии и кофейни.
- Более 50% этих заведений являются сетевыми.

Топ-15 популярных сетей в Москве

Сгруппируйте данные по названиям заведений и найдите топ-15 популярных сетей в Москве. Под популярностью понимается количество заведений этой сети в регионе. Постройте подходящую для такой информации визуализацию. Знакомы ли вам эти сети? Есть ли какой-то признак, который их объединяет? К какой категории заведений они относятся?

```
In [31]: # Группируем данные по названиям заведений и найдем топ-15 популярных сетей в Москве.

data_chain = df[df['chain'] == 1]
top_15 = data_chain.groupby('name').agg({'rating': 'median', 'category': pd.Series.mode, 'district': 'count'})
top_15 = top_15.rename(columns={'district': 'count'})
top_15 = top_15.sort_values('count', ascending = False).reset_index().head(15)
top_15
```

```
Out[31]:
```

	name	rating	category	count
0	шоколадница	4.20	кофейня	120
1	домино'с пицца	4.20	пиццерия	76
2	додо пицца	4.30	пиццерия	74
3	one price coffee	4.20	кофейня	71
4	яндекс лавка	4.00	ресторан	69
5	sofix	4.10	кофейня	65
6	prime	4.20	ресторан	50
7	хинкальная	4.40	кафе	44
8	кофепорт	4.20	кофейня	42
9	кулинарная лавка братьев караваевых	4.40	кафе	39

	name	rating	category	count
10	теремок	4.10	ресторан	38
11	чайхана	4.10	кафе	37
12	cofest	4.05	кофейня	32
13	буханка	4.40	булочная	32
14	му-му	4.30	кафе	27

In [32]:

```
# Визуализируем таблицу с самыми популярными сетями в Москве:
fig = px.bar(top_15,
             x='count',
             y='name',
             text='count',
             color='name',
             height=500,
             width=1000
            )
fig.update_layout(title = 'Топ-15 популярных сетей Москвы',
                  xaxis_title = 'Количество заведений',
                  yaxis_title = 'Название заведений',
                  showlegend = True)
fig.show()
```

Топ-15 популярных сетей Москвы



- Вышеприведенные сети, конечно, знакомы многим жителям Москвы и тем, кто бывал в этом городе.
- Подавляющее большинство популярных заведений относится к категориям: кафе, кофейни, рестораны, пиццерии.
- Лидирующую позицию в нашем топ-15 занимает кофейня "Шоколадница". В Москве целых 120 заведений данной сети.
- Остальные сети значительно отстают от лидера: в них число заведений меньше на 40% и более.

In [33]:

```
print('В датасете присутствуют следующие административные районы Москвы:', df['district'].unique())
```

В датасете присутствуют следующие административные районы Москвы: ['Северный административный округ', 'Северо-Восточный административный округ', 'Северо-Западный административный округ', 'Западный административный округ', 'Центральный административный округ', 'Восточный административный округ', 'Юго-Восточный административный округ', 'Южный административный округ', 'Юго-Западный административный округ']

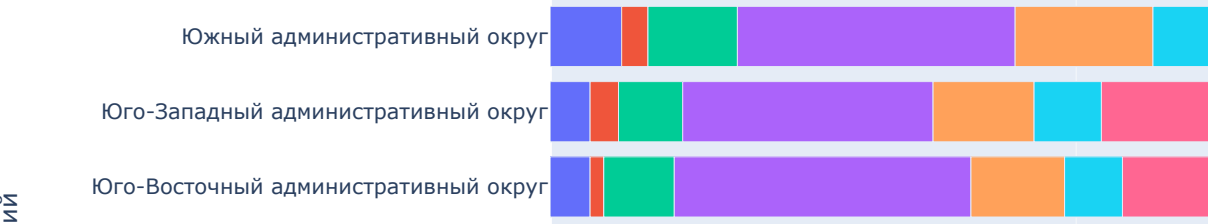
```
In [34]: pie_district = pd.pivot_table(df, index = ['district', 'category'], values = ['name'], aggfunc = ['count'])
pie_district.columns = ['district', 'category', 'count']
display(pie_district)
```

	district	category	count
0	Восточный административный округ	бар,паб	53
1	Восточный административный округ	булочная	25
2	Восточный административный округ	быстрое питание	71
3	Восточный административный округ	кафе	272
4	Восточный административный округ	кофейня	105
...
67	Южный административный округ	кафе	264
68	Южный административный округ	кофейня	131
69	Южный административный округ	пиццерия	73
70	Южный административный округ	ресторан	202
71	Южный административный округ	столовая	44

72 rows x 3 columns

```
In [35]: # Визуализируем количество заведений и количество заведений каждой категории по районам
fig = px.bar(pie_district, x="count", y="district", color="category")
fig.update_layout(title = 'Количество заведений в зависимости от категории по округам',
                    xaxis_title = 'Название административного округа',
                    yaxis_title = 'Количество заведений',
                    showlegend = True)
fig.show()
```

Количество заведений в зависимости от категории по округам



```
In [36]: abc = pd.crosstab(df.district, df.category)
abc
```

Out[36]:

	category	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
	district								
	Восточный административный округ	53	25	71	272	105	72	160	40
	Западный административный округ	50	37	62	238	150	71	218	24
	Северный административный округ	68	39	58	234	193	77	188	41
	Северо-Восточный административный округ	62	28	82	269	159	68	182	40
	Северо-Западный административный округ	23	12	30	115	62	40	109	18
	Центральный административный округ	364	50	87	464	428	113	670	66
	Юго-Восточный административный округ	38	13	67	282	89	55	145	25
	Юго-Западный административный округ	38	27	61	238	96	64	168	17
	Южный административный округ	68	25	85	264	131	73	202	44

```
In [37]: abc = pd.crosstab(index = df.district,
                        columns = df.category,
                        normalize = 'index').reset_index()

abc
```

Out[37]:

	category	district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
0	Восточный административный округ		0.066416	0.031328	0.088972	0.340852	0.131579	0.090226	0.200501	0.050125
1	Западный административный округ		0.058824	0.043529	0.072941	0.280000	0.176471	0.083529	0.256471	0.028235
2	Северный административный округ		0.075724	0.043430	0.064588	0.260579	0.214922	0.085746	0.209354	0.045657
3	Северо-Восточный административный округ		0.069663	0.031461	0.092135	0.302247	0.178652	0.076404	0.204494	0.044944
4	Северо-Западный административный округ		0.056235	0.029340	0.073350	0.281174	0.151589	0.097800	0.266504	0.044010
5	Центральный административный округ		0.162355	0.022302	0.038805	0.206958	0.190901	0.050401	0.298840	0.029438

category	district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
6	Юго-Восточный административный округ	0.053221	0.018207	0.093838	0.394958	0.124650	0.077031	0.203081	0.035014
7	Юго-Западный административный округ	0.053597	0.038082	0.086037	0.335684	0.135402	0.090268	0.236953	0.023977
8	Южный административный округ	0.076233	0.028027	0.095291	0.295964	0.146861	0.081839	0.226457	0.049327

In [38]:

```
fig = go.Figure()
fig.add_trace(go.Bar(
    x=abc['бар,паб'],
    y=abc.district,
    name='бар,паб',
    orientation='h',
    marker=dict(
        color='#7eb0d5',
        #line=dict(color='#7eb0d5', width=3)
    )
))
fig.add_trace(go.Bar(
    x=abc['булочная'],
    y=abc.district,
    name='булочная',
    orientation='h',
    marker=dict(
        color='#b2e061',
        #line=dict(color='#b2e061', width=3)
    )
))
fig.update_layout(title = 'Количество заведений в зависимости от категории по округам',
    barmode='stack')
fig.show()
```

Количество заведений в зависимости от категории по округам



- На приведенных выше графиках можем отметить, что максимальное количество заведений расположилось в Центральном административном округе.
- Именно здесь число предприятий общественного питания более чем в 2 раза превышает их число в любом другом Московском районе.

Заведения по административным районам Москвы. Отобразите общее количество заведений и количество заведений каждой категории по районам. Попробуйте проиллюстрировать эту информацию одним графиком.

Рейтинги по категориям заведений.

- Визуализируйте распределение средних рейтингов по категориям заведений.
- Сильно ли различаются усреднённые рейтинги в разных типах общепита?

In [39]:

	name	category	address	district	hours	lat	lng	rating	price
0	wowfli	кафе	москва, улица дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN
1	четыре комнаты	ресторан	москва, улица дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	4.5	выше среднего
2	хазри	кафе	москва, клязьминская улица, 15	Северный административный округ	пн-чт 11:00–02:00; пт,сб 11:00–05:00; вс 11:00–...	55.889146	37.525901	4.6	средние сч
3	dormouse coffee shop	кофейня	москва, улица маршала федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	5.0	NaN L кап
4	иль марко	пиццерия	москва, правобережная улица, 16	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние сч

In [40]:

	category	mean_rating
0	бар,паб	4.39
5	пиццерия	4.30
6	ресторан	4.29
4	кофейня	4.28
1	булочная	4.27
7	столовая	4.21
3	кафе	4.12
2	быстрое питание	4.05

In [41]:

```
fig = px.bar(rating_cat,
             x='mean_rating',
             y='category',
             text='mean_rating',
             color='category'
            )
fig.update_layout(title='Распределение средних рейтингов по категориям заведений',
                  xaxis_title='Рейтинг',
                  yaxis_title='Название категорий')
fig.update_xaxes(range=[4, 4.5])
fig.show()
```

Распределение средних рейтингов по категориям заведений



- Усредненные рейтинги общепита различаются.
- Наивысших рейтингов удостоилась категория бар/паб, а в аутсайдеры попали предприятия быстрого питания

Хороплет со средним рейтингом заведений каждого района.

- Границы районов Москвы, которые встречаются в датасете, хранятся в файле `admin_level_geomap.geojson` (скачать файл для локальной работы).
- Отобразите все заведения датасета на карте с помощью кластеров средствами библиотеки `folium`.
- Найдите топ-15 улиц по количеству заведений. Постройте график распределения количества заведений и их категорий по этим улицам.
- Попробуйте проиллюстрировать эту информацию одним графиком. Найдите улицы, на которых находится только один объект общепита. Что можно сказать об этих заведениях?

In [42]:

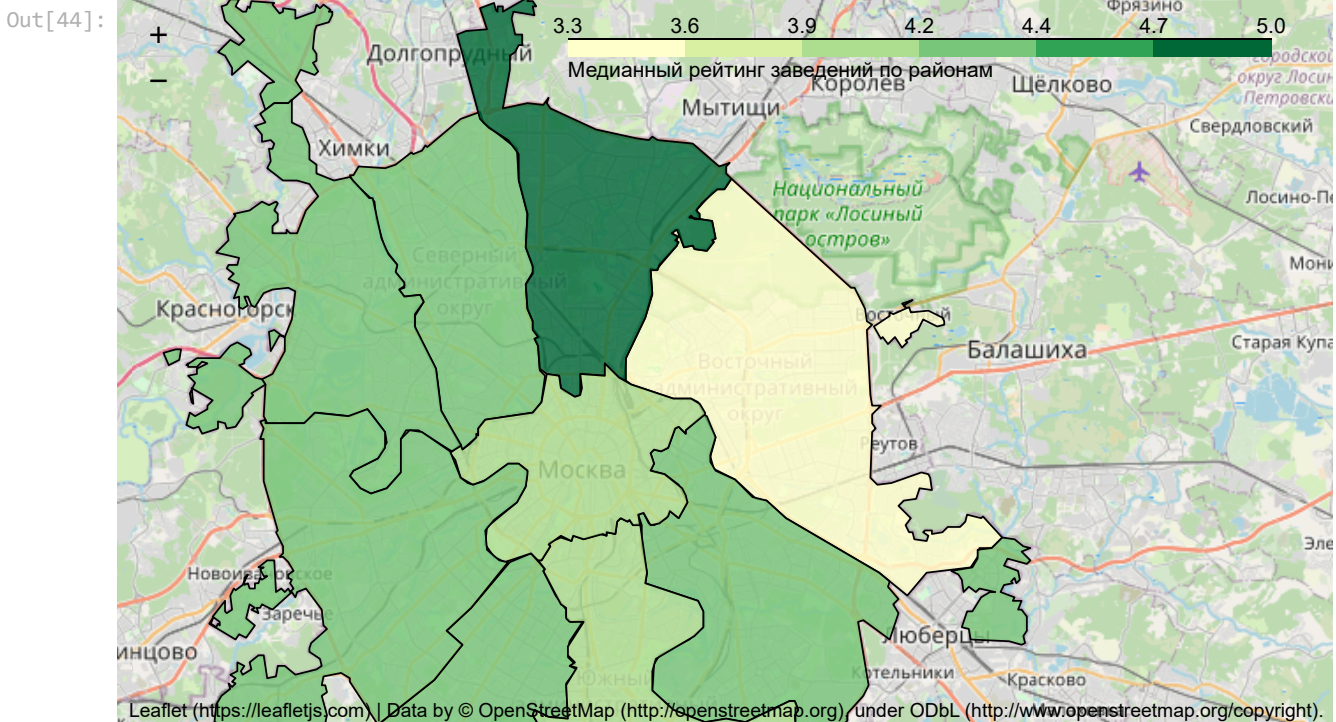
```
# подключаем модуль для работы с JSON-форматом
import json
# читаем файл и сохраняем в переменной
with open('/datasets/admin_level_geomap.geojson', 'r') as f:
    geo_json = json.load(f)

#print(json.dumps(geo_json, indent=2, ensure_ascii=False, sort_keys=True))
```

```
In [43]: df['district'].unique()
```

```
Out[43]: array(['Северный административный округ',  
        'Северо-Восточный административный округ',  
        'Северо-Западный административный округ',  
        'Западный административный округ',  
        'Центральный административный округ',  
        'Восточный административный округ',  
        'Юго-Восточный административный округ',  
        'Южный административный округ',  
        'Юго-Западный административный округ'], dtype=object)
```

```
In [44]: # импортируем карту и хороплет  
from folium import Map, Choropleth  
  
# загружаем JSON-файл с границами округов Москвы  
state_geo = '/datasets/admin_level_geomap.geojson'  
# moscow_lat - широта центра Москвы, moscow_lng - долгота центра Москвы  
moscow_lat, moscow_lng = 55.751244, 37.618423  
  
# создаём карту Москвы  
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10)  
  
# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту  
Choropleth(  
    geo_data=state_geo,  
    data=df,  
    columns=['district', 'rating'],  
    key_on='feature.name',  
    fill_color='YlGn',  
    fill_opacity=0.8,  
    legend_name='Медианный рейтинг заведений по районам',  
) .add_to(m)  
  
# выводим карту  
m
```



Проанализировав данные, отображенные на хороплете, можем сделать следующий вывод:

- Наивысшим средним рейтингом отличаются заведения в северо-восточном административном округе Москвы(между 4,7 и 5,0 баллов).

- Самым низким рейтингом отличился восточный административный округ (его оценка на уровне 3,3 - 3,6 баллов).
- Рейтинг заведений в остальных районах держится на среднем уровне.

In [45]:

```
# импортируем карту и маркер
from folium import Map, Marker
# импортируем кластер
from folium.plugins import MarkerCluster

# moscow_lat - широта центра Москвы, moscow_lng - долгота центра Москвы
moscow_lat, moscow_lng = 55.751244, 37.618423

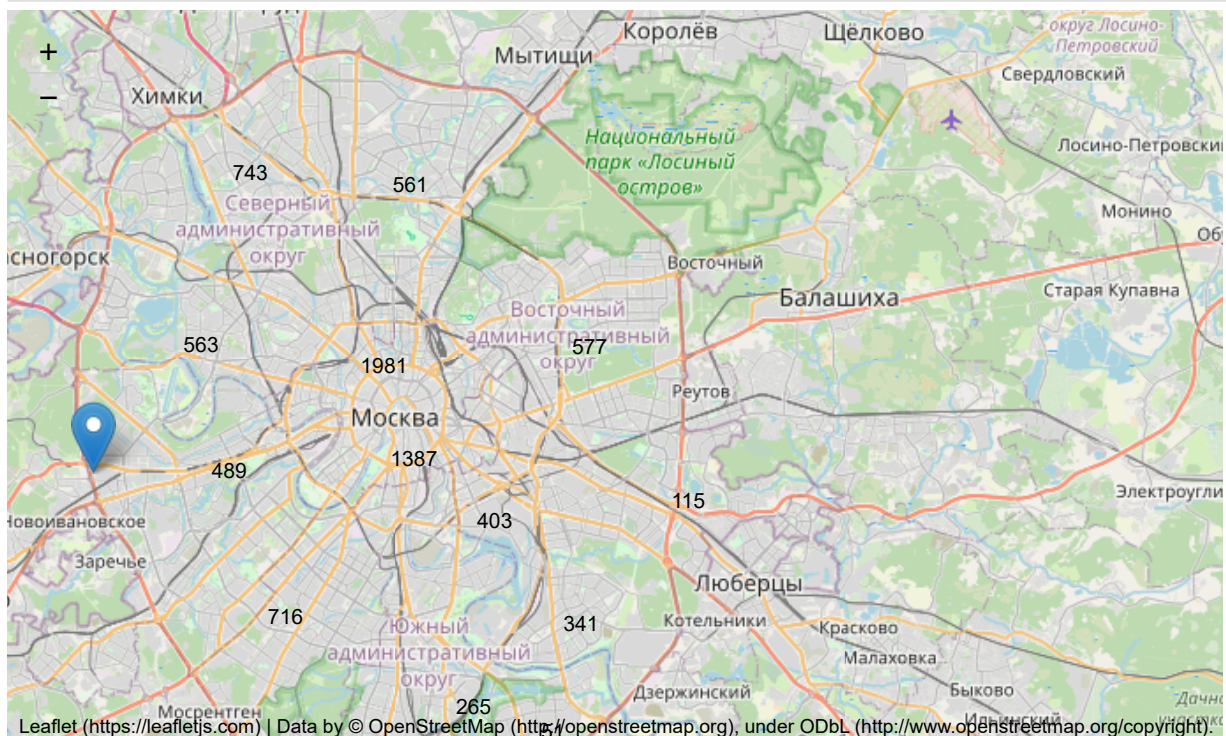
# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаём пустой кластер, добавляем его на карту
marker_cluster = MarkerCluster().add_to(m)

# пишем функцию, которая принимает строку датафрейма,
# создаёт маркер в текущей точке и добавляет его в кластер marker_cluster
def create_clusters(row):
    Marker(
        [row['lat'], row['lng']],
        popup=f"{row['name']} {row['rating']}",
    ).add_to(marker_cluster)

# применяем функцию create_clusters() к каждой строке датафрейма
df.apply(create_clusters, axis=1)

# выводим карту
m
```

Out[45]:



Отобразили все заведения датасета на карте с помощью кластеров средствами библиотеки folium. Отметили следующую закономерность: По мере движения от центра к окраине Москвы, количество заведений общественного питания постепенно уменьшается.

In [46]:

```
# Топ-15 улиц по количеству заведений
msc_streets = df[df['street'].notnull()]
top15_streets = msc_streets['street'].value_counts().reset_index().head(15)
top15_streets.columns = ['street', 'count']
top15_streets
```


Out[46]:

	street	count
0	проспект мира	183
1	профсоюзная улица	122
2	проспект вернадского	108
3	ленинский проспект	107
4	ленинградский проспект	95
5	дмитровское шоссе	88
6	каширское шоссе	77
7	варшавское шоссе	76
8	ленинградское шоссе	70
9	мкад	65
10	люблинская улица	60
11	улица вавилова	55
12	кутузовский проспект	54
13	улица миклухо-маклая	49
14	пятницкая улица	48

In [47]:

```
# таблица улица - категория - количество
streets_category = df.groupby(['street', 'category'])['name'].count().reset_index()
streets_category.columns = ['street', 'category', 'count']
streets_category.sort_values('count', ascending=False)
```

Out[47]:

	street	category	count
2137	проспект мира	кафе	53
1520	мкад	кафе	45
2140	проспект мира	ресторан	45
2138	проспект мира	кофейня	36
2149	профсоюзная улица	кафе	35
...
1536	молодёжная улица	кофейня	1
1538	монтажная улица	кофейня	1
1539	монтажная улица	ресторан	1
1540	москворецкая набережная	кофейня	1
3820	№ 7	кафе	1

3821 rows × 3 columns

In [48]:

```
# улицы из топ15
streets_top15_category = streets_category[streets_category['street'].isin(top15_streets['street'])]
streets_top15_category
```

Out[48]:

	street	category	count
698	варшавское шоссе	бар,паб	6
699	варшавское шоссе	быстрое питание	7
700	варшавское шоссе	кафе	18
701	варшавское шоссе	кофейня	14

	street	category	count
702	варшавское шоссе	пиццерия	4
...
3174	улица миклухо-маклая	быстрое питание	4
3175	улица миклухо-маклая	кафе	21
3176	улица миклухо-маклая	кофейня	4
3177	улица миклухо-маклая	пиццерия	2
3178	улица миклухо-маклая	ресторан	15

111 rows × 3 columns

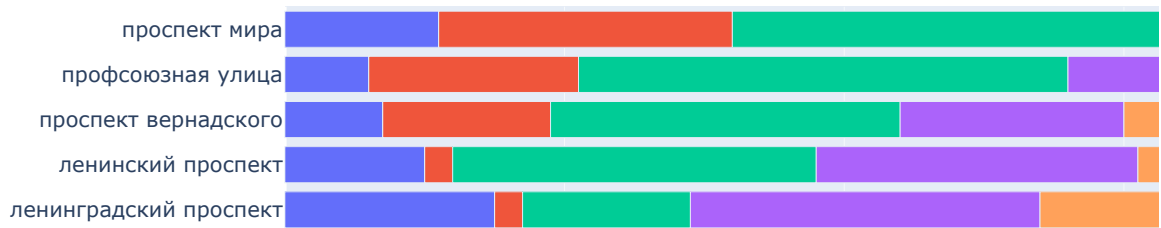
In [49]:

```
# Построим график распределения количества заведений и их категорий по этим улицам на одном графике
fig = px.bar(streets_top15_category,
             x='count',
             y='street',
             color='category'
            )

fig.update_layout(title='Количество заведений каждой категории по районам',
                  xaxis_title='Количество заведений',
                  yaxis_title='Название улиц',
                  yaxis={'categoryorder':'total ascending'})

fig.show()
```

Количество заведений каждой категории по районам



- Самой "богатой" на предприятия общепита в Москве является улица "проспект Мира". Именно здесь находится максимальное количество кафе, кофейен, ресторанов, и заведений быстрого питания.

In [50]:

```
# Найдем улицы, на которых находится только один объект общепита.

one_cafe = df['street'].value_counts().reset_index()
```

```
one_cafe.columns = ['street', 'count']
one_cafe = one_cafe[one_cafe['count']==1]
one_cafe
```

```
Out[50]:
```

	street	count
990	стромынский переулок	1
991	молдавская улица	1
992	1-й новокузнецкий переулок	1
993	2-й красносельский переулок	1
994	12-я парковая улица	1
...
1442	братеевский парк	1
1443	поперечный просек	1
1444	улица седова	1
1445	2-я улица марьиной рощи	1
1446	улица саморы машела	1

457 rows × 2 columns

```
In [51]: # Посмотрим на категории:
one_cafe_category = streets_category[streets_category['street'].isin(one_cafe['street'])]
one_cafe_category = one_cafe_category.groupby('category')['street'].count().sort_values(ascending=
one_cafe_category
```

```
Out[51]: category
кафе          159
ресторан       93
кофейня       84
бар, паб       39
столовая       36
быстрое питание 23
пиццерия       15
булочная       8
Name: street, dtype: int64
```

- Всего было найдено 458 улиц, на которых находится только один объект общепита. Такие заедения наиболее часто относятся к категории "кафе", и реже всего к категории "булочная".
- Вероятнее всего, на этих улицах всего один объект общепита т.к.они обладают совсем небольшими размерами и протяженностью.

Значения средних чеков заведений хранятся в столбце `middle_avg_bill`. Эти числа показывают примерную стоимость заказа в рублях, которая чаще всего выражена диапазоном. Посчитайте медиану этого столбца для каждого района. Используйте это значение в качестве ценового индикатора района. Постройте фоновую картограмму (хороплет) с полученными значениями для каждого района. Проанализируйте цены в центральном административном округе и других. Как удалённость от центра влияет на цены в заведениях?

```
In [52]: #Расчитаем медианный счет для каждого района:
median_bill = pd.pivot_table(df, index = ['district'], values = ['middle_avg_bill'], aggfunc = ['m
median_bill.columns = ['district', 'median_bill']
median_bill
```

```
Out[52]:
```

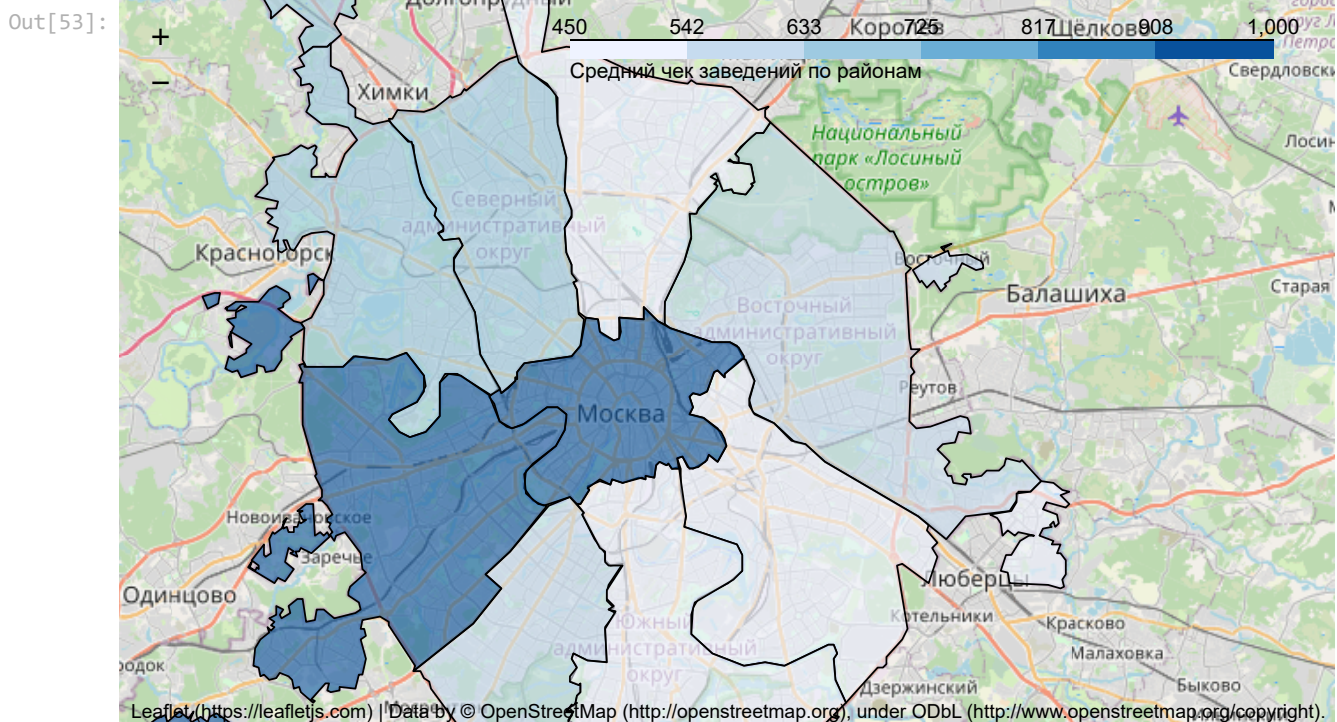
	district	median_bill
0	Восточный административный округ	575.0

	district	median_bill
1	Западный административный округ	1000.0
2	Северный административный округ	650.0
3	Северо-Восточный административный округ	500.0
4	Северо-Западный административный округ	700.0
5	Центральный административный округ	1000.0
6	Юго-Восточный административный округ	450.0
7	Юго-Западный административный округ	600.0
8	Южный административный округ	500.0

```
In [53]: # Построим фоновую картограмму (хороплет) с полученными медианными значениями чеков для каждого ра
mm = Map(location=[moscow_lat, moscow_lng], zoom_start=10)

# хороплет
Choropleth(
    geo_data=state_geo,
    data = median_bill,
    columns = ['district', 'median_bill'],
    key_on = 'feature.name',
    legend_name = 'Средний чек заведений по районам',
).add_to(mm)

mm
```



Глядя на хороплет, мы можем убедиться в том, что самые дорогие средние чеки в заведениях в центральном и западном округах.

- Соберите наблюдения по вопросам выше в один общий вывод.

Детализация исследования: открытие кофейни

Сколько всего кофеен в датасете? В каких районах их больше всего, каковы особенности их расположения?

```
In [54]: df.head(5)
```

Out[54]:

	name	category	address	district	hours	lat	lng	rating	price
0	wowфли	кафе	москва, улица дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN
1	четыре комнаты	ресторан	москва, улица дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	4.5	выше среднего
2	хазри	кафе	москва, клязьминская улица, 15	Северный административный округ	пн-чт 11:00–02:00; пт,сб 11:00–05:00; вс 11:00...	55.889146	37.525901	4.6	средние сч
3	dormouse coffee shop	кофейня	москва, улица маршала федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	5.0	NaN L кап
4	иль марко	пиццерия	москва, правобережная улица, 16	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние сч



In [55]:

```
coffee = df[df['category'] == 'кофейня']
print('Количество кофеен:', len(coffee))
```

Количество кофеен: 1413

In [56]:

```
# создаем карту
m_coffee = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаем пустой кластер и добавляем его на карту
marker_cluster = MarkerCluster().add_to(m_coffee)

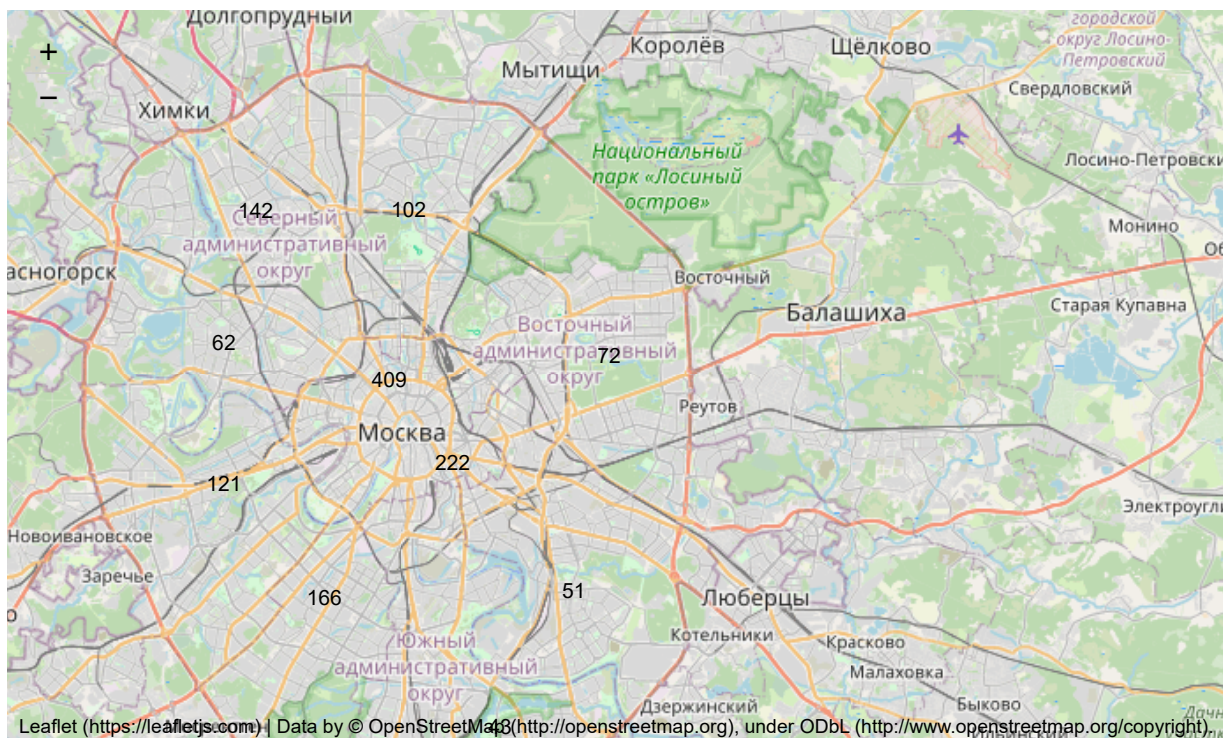
# функция, которая принимает строку датафрейма,
# создаёт маркер в текущей точке и добавляет его в кластер marker_cluster

def create_clusters(row):
    Marker(
        [row['lat'], row['lng']],
        popup=f"{row['name']} {row['rating']}",
    ).add_to(marker_cluster)

# применяем функцию create_clusters() к каждой строке датафрейма
coffee.apply(create_clusters, axis=1)

# выводим карту
m_coffee
```

Out[56]:



Вывод

- Всего в датасете 1413 кофеен.
- Больше всего заведений из этой категории расположено в центральной части города.
- Меньше всего кофеен находится в южном, юго-восточном и восточном округах.

Есть ли круглосуточные кофейни?

In [57]:

```
coffee_24_7 = coffee[coffee['works24/7'] == True]
print('Количество круглосуточных кофеен:', len(coffee_24_7))
```

Количество круглосуточных кофеен: 74

In [58]:

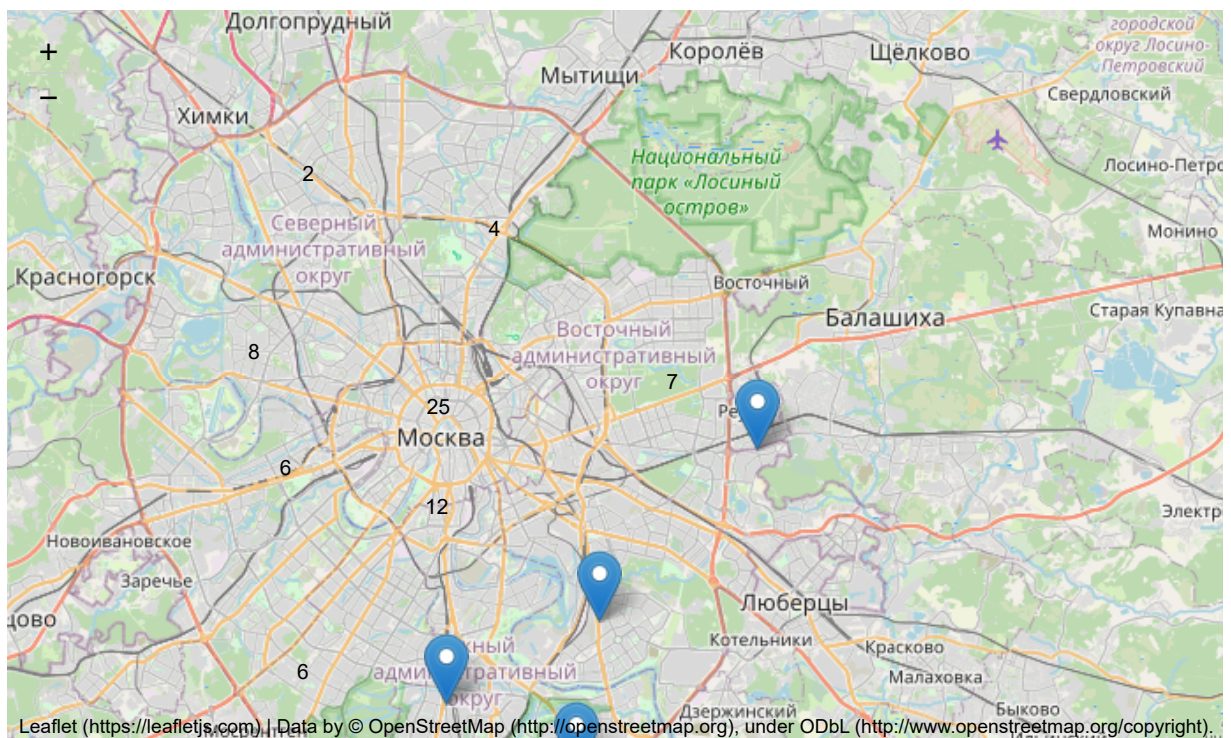
```
# создаём карту Москвы
m_coffee = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаём пустой кластер, добавляем его на карту
marker_cluster = MarkerCluster().add_to(m_coffee)

# пишем функцию, которая принимает строку датафрейма,
# создаёт маркер в текущей точке и добавляет его в кластер marker_cluster
def create_clusters(row):
    Marker(
        [row['lat'], row['lng']],
        popup=f"{row['name']} {row['rating']}",
    ).add_to(marker_cluster)

# применяем функцию create_clusters() к каждой строке датафрейма
coffee_24_7.apply(create_clusters, axis=1)

# выводим карту
m_coffee
```


Out[58]:



Вывод

- Мы обнаружили в датасете 59 круглосуточных кофеен.
- Из них максимальное количество - 59 - расположились в центре Москвы.
- Лишь по одной из кофеен находятся в южном, юго-восточном и северном округах.

Какие у кофеен рейтинги? Как они распределяются по районам?

In [59]:

```
coffee_rating = coffee.groupby('district', as_index=False)['rating'].agg('mean').round(2).sort_val  
coffee_rating
```

Out[59]:

	district	rating
5	Центральный административный округ	4.34
4	Северо-Западный административный округ	4.33
2	Северный административный округ	4.29
0	Восточный административный округ	4.28
7	Юго-Западный административный округ	4.28
6	Юго-Восточный административный округ	4.23
8	Южный административный округ	4.23
3	Северо-Восточный административный округ	4.22
1	Западный административный округ	4.20

Вывод

- Рейтинги кофеен находятся в пределах от 4.2 до 4.34
- Любителям кофе и десертов достаточно сложно угодить.

На какую стоимость чашки капучино стоит ориентироваться при открытии и почему?

In [60]:

```
cup = coffee.groupby('district', as_index=False)['middle_coffee_cup'].agg('mean').round(2).sort_val  
display(cup)  
coffee_price = round(cup['middle_coffee_cup'].mean(), 2)  
coffee_price2 = round(cup['middle_coffee_cup'].median(), 2)
```

```
print(f'Средняя стоимость чашки кофе в Москве:', coffee_price)
print(f'Медианная стоимость чашки кофе в Москве:', coffee_price2)
```

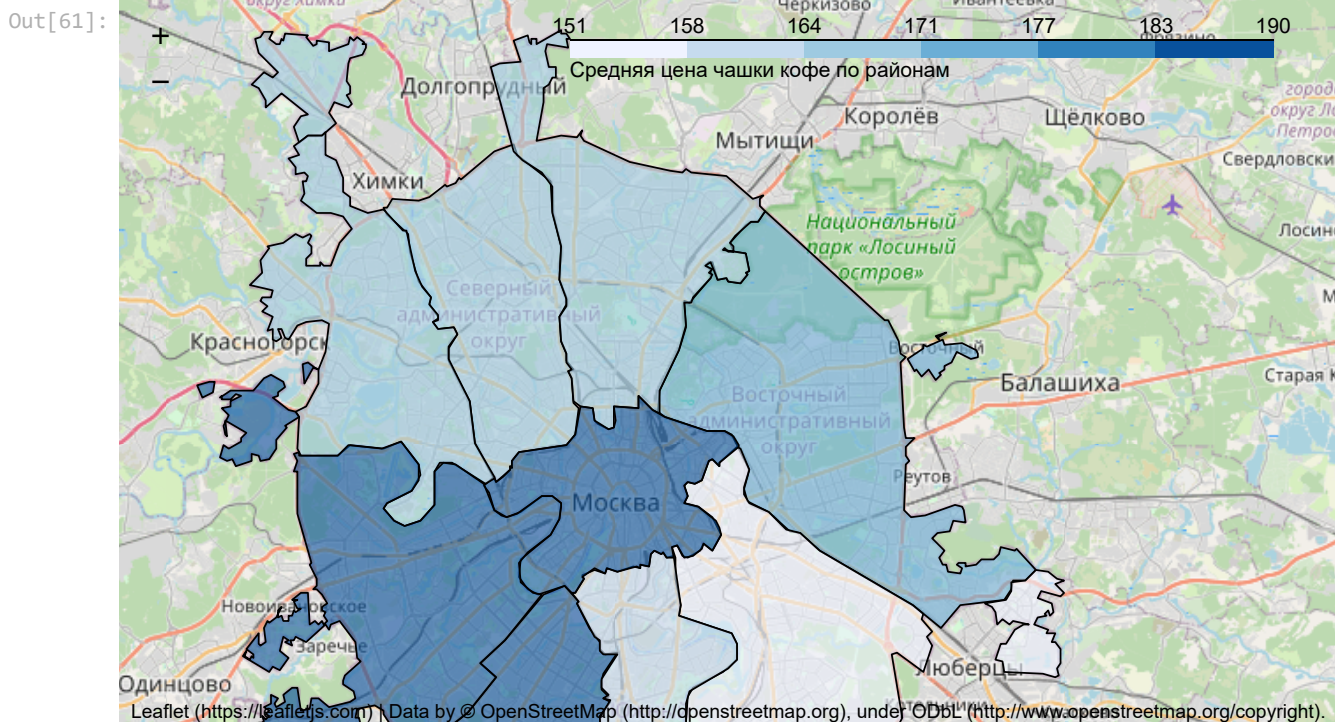
	district	middle_coffee_cup
1	Западный административный округ	189.94
5	Центральный административный округ	187.52
7	Юго-Западный административный округ	184.18
0	Восточный административный округ	174.02
2	Северный административный округ	165.79
4	Северо-Западный административный округ	165.52
3	Северо-Восточный административный округ	165.33
8	Южный административный округ	158.49
6	Юго-Восточный административный округ	151.09

Средняя стоимость чашки кофе в Москве: 171.32
Медианная стоимость чашки кофе в Москве: 165.79

```
In [61]: m_cup = Map(location=[moscow_lat, moscow_lng], zoom_start=10)

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=cup,
    columns=['district', 'middle_coffee_cup'],
    key_on='feature.name',
    legend_name='Средняя цена чашки кофе по районам',
).add_to(m_cup)

# выводим карту
m_cup
```



Вывод

- При открытии кофейни стоит ориентироваться на стоимость чашки кофе в пределах от 150 до 190 рублей.
- Стоимость чашки капучино в заведении также будет зависеть от района и округа, в котором будет открыто заведение: Так, в районах с более дорогой, элитной недвижимостью или широко развитой

туристической инфраструктурой цена чашки кофе будет равна 183-190 рублям. В спальных районах города можно ориентироваться на цену 151-158 рублей за чашку капучино.

- Средняя стоимость чашки кофе в Москве: 171.32
- Медианная стоимость чашки кофе в Москве: 165.79

Рекомендации:

Без обратной связи и большей информации от инвесторов и учредителей сложно дать конкретные рекомендации. Как правило, для того, чтобы принять такое важное решение об открытии собственного заведения и выбора для него места - необходима сплоченная работа заказчика, аналитиков и многих других сотрудников. Наше исследование получилось очень поверхностным. Было бы неплохо обратить внимание на перспективные районы с малым количеством заведений на текущий момент и соотнести их с местами скопления большого количества людей, остановками общественного транспорта и метро. Много зависит от других факторов, которые мы не учитываем: например, от бюджета на открытие заведения, необходимых сроков окупаемости. На данный момент на рынке общепита более популярны кафе. Если разместить объект на одной из топ-15 улиц, то поток покупателей будет обеспечен, т.к. это многолюдные популярные, большие улицы, которые находятся в центре и усеяны бизнес-центрами, достопримечательностями и местами туристической инфраструктуры. Основываясь на высокой стоимости чашки капучино, стоит обратить внимание на Центральный, Западный и Юго-Западный административные районы. Здесь посетителям более привычно видеть высокий ценник. Заведение окупится быстрее. Возможно стоит обратить внимание на район с самыми низкими рейтингами и сделать будущее кафе самым популярным и качественным, если работать на хороший рейтинг. Самый низкий средний рейтинг в Западном административном округе (ниже 4.2). Лучше всего открывать кофейню возле мест высокой проходимости, например, метро, либо возле учебных заведений, офисов.

Ссылка на презентацию:

https://disk.yandex.ru/i/zjK2ODhb__usZQ