

Рынок заведений общественного питания Москвы

Описание проекта

Инвесторы из фонда «Shut Up and Take My Money» решили попробовать себя в новой области и открыть заведение общественного питания в Москве. Заказчики ещё не знают, что это будет за место: кафе, ресторан, пиццерия, паб или бар, — и какими будут расположение, меню и цены.

Для начала они просят вас — аналитика — подготовить исследование рынка Москвы, найти интересные особенности и презентовать полученные результаты, которые в будущем помогут в выборе подходящего инвесторам места.

Постарайтесь сделать презентацию информативной и лаконичной. Её структура и оформление сильно влияют на восприятие информации читателями вашего исследования. Выбирать инструменты (matplotlib, seaborn и другие) и типы визуализаций вы можете самостоятельно.

Вам доступен датасет с заведениями общественного питания Москвы, составленный на основе данных сервисов Яндекс Карты и Яндекс Бизнес на лето 2022 года. Информация, размещённая в сервисе Яндекс Бизнес, могла быть добавлена пользователями или найдена в общедоступных источниках. Она носит исключительно справочный характер.

Оглавление

1. Описание проекта
 - 1.1 [Описание данных](#)
 - 1.2 [Выполнение проекта](#)
2. [Импорт библиотек](#)
3. [Выгрузка датасета](#)
 - 3.1 [Краткий вывод](#)
4. [Предобработка](#)
 - 4.1 [Краткий вывод](#)
5. [Анализ данных](#)
 - 5.1 [Категории заведений](#)
 - 5.2 [Количество мест по категориям](#)
 - 5.3 [Сетевые заведения](#)
 - 5.4 [Средний рейтинг](#)
 - 5.5 [Все заведения на карте](#)
 - 5.6 [Топ-15 улиц](#)
 - 5.7 [Средний чек](#)
 - 5.8 [Анализ круглосуточных заведений](#)

- 5.9 [Другие зависимости](#)
- 5.10 [Краткий вывод](#)
- 6. [Детали кофеен](#)
 - 6.1 Количество и расположение
 - 6.2 Круглосуточные
 - 6.3 Рейтинги по районам
 - 6.4 Чашка кофе
 - 6.5 [Непосредственные конкуренты в СЗАО](#)
 - 6.6 [Краткий вывод](#)
- 7. [Вывод](#)

Описание данных

Файл `_moscowplaces.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);
`district` — административный район, в котором находится заведение, например Центральный административный округ; `seats` — количество посадочных мест.

Выполнение проекта

Шаг 1. Загрузите данные и изучите общую информацию

Загрузите данные о заведениях общественного питания Москвы.

Изучите общую информацию о датасете. Сколько заведений представлено? Что можно сказать о каждом столбце? Значения какого типа они хранят?

Шаг 2. Выполните 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. 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? Отобразите общее количество заведений и количество заведений каждой категории по районам.
- 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`.
 - 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. Детализируем исследование: 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».

Оформление Основное задание выполните в Jupyter Notebook, программный код заполните в ячейках типа `code`, текстовые пояснения — в ячейках типа `markdown`. Примените форматирование и заголовки. Презентацию можно выполнить с помощью любого удобного вам инструмента, главное — экспортировать её в PDF-формат.

Импорт библиотек

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly import graph_objects as go
# подключаем модуль для работы с JSON-форматом
import json
# импортируем маркер, карту и хороплет
from folium import Marker, Map, Choropleth
# импортируем кластер
from folium.plugins import MarkerCluster
```

Выгрузка moscow_places.csv

```
In [2]: def watch_basics(df):
        """
        Отображает базовую информацию о датасете
        """
        p = 20
        print('*'*p, 'Общая информация', '*'*p)
        display(df.head())
        df.info()
        print('*'*p, 'Пропуски', '*'*p)
        if not (df.isna().sum() > 0).any():
            print('*'*p, 'Пропусков не найдено', '*'*p)
        else:
            display(df.isna().sum())
        print('*'*p, 'Явные дубликаты', '*'*p)
        display(df[df.duplicated()].count())
```

```
In [3]: df_cafe = pd.read_csv('/datasets/moscow_places.csv', sep=',')
watch_basics(df_cafe)
```

```
***** Общая информация *****
```

	name	category	address	district	hours	lat	lng	ra
0	WoWfli	кафе	Москва, улица Дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	
1	Четыре комнаты	ресторан	Москва, улица Дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	
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	
3	Dormouse Coffee Shop	кофейня	Москва, улица Маршала Федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	
4	Иль Марко	пиццерия	Москва, Правобережная улица, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	

```
<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
***** Пропуски *****
name                0
category            0
address             0
district            0
hours               536
lat                 0
lng                 0
rating              0
price               5091
avg_bill            4590
middle_avg_bill     5257
middle_coffee_cup   7871
chain               0
seats               3611
dtype: int64
***** Явные дубликаты *****
```

name	0
category	0
address	0
district	0
hours	0
lat	0
lng	0
rating	0
price	0
avg_bill	0
middle_avg_bill	0
middle_coffee_cup	0
chain	0
seats	0

В датасете нет явных дубликатов. Всего 8406 строк, однако много пропусков в столбцах hours, price, avg_bill, middle_avg_bill, middle_coffee_cup и seats. Возможно, это результат слияния нескольких таблиц. middle_avg_bill и middle_coffee_cup основаны на avg_bill, причем начинающихся со строк «Средний счёт» или «Цена одной чашки капучино» соответственно - поэтому не удивительно, что пропусков здесь много больше, чем в avg_bill. У части заведений отсутствует рейтинг в Яндекс-картах. Возможно, это какие-то новые заведения, не успевшие набрать более 5 оценок - а потому еще не имеют таких данных.

```
In [4]: # у заведения может быть несколько точек. Посмотрим тогда только на уникальные
print('В датасете представлено', len(df_cafe['name'].unique()), 'уникальных заведе-
```

В датасете представлено 5614 уникальных заведений.

```
In [5]: for col in df_cafe.columns:
pct_missing = np.mean(df_cafe[col].isnull())
print('{} - {}'.format(col, round(pct_missing*100)))
```

```
name - 0%
category - 0%
address - 0%
district - 0%
hours - 6%
lat - 0%
lng - 0%
rating - 0%
price - 61%
avg_bill - 55%
middle_avg_bill - 63%
middle_coffee_cup - 94%
chain - 0%
seats - 43%
```

6% заведений из датасета не предоставили информацию о часах работы, таких 536 строк. У 61% отсутствует категория цен. Обычно ее загружает сама компания. У 55% отсутствует средний чек. Это меньше, чем каталог цен. Похоже, посетители активнее, чем сами заведения. У 43% также отсутствуют данные о посадочных местах.

Краткий вывод

Всего 8406 строк, есть пропуски в 6 столбцах. Возможно, это результат слияния нескольких таблиц. middle_avg_bill и middle_coffee_cup основаны на avg_bill, а потому

меют 63% и 94% пропусков. У части заведений отсутствует рейтинг в Яндекс-картах. Возможно, это какие-то новые заведения, не успевшие набрать более 5 оценок. Нет явных дубликатов. Хранить данные в object и float64 не целесообразно с точки зрения памяти. В данной работе переопределять их не будем - но стоит обратить внимание в выводе.

Предобработка

```
In [6]: # посмотрим есть ли дубликаты по адресу и имени заведения
print('Количество дубликатов:', df_cafe[df_cafe.duplicated(subset=['name' , 'address'])])

Количество дубликатов: 0

In [7]: # создадим столбец с только улицей
df_cafe['street'] = df_cafe['address'].str.split(', ').str[1]

In [8]: # проверим как кафе объявляют, что работают круглосуточно
for name in df_cafe[~df_cafe['hours'].isna()]['hours'].unique():
    if name.find('круг') != -1:
        print(name)
```


ежедневно, круглосуточно
 пн 00:01–12:00, перерыв 12:00–13:30; вт-чт 13:30–12:00; пт 13:30–00:00; сб,вс круглосуточно
 вт-вс круглосуточно
 сб круглосуточно
 пт-вс круглосуточно
 пн,ср,чт,пт,сб,вс круглосуточно
 пн 10:00–00:00; вт-сб круглосуточно; вс 00:00–23:00
 пн-чт 07:30–23:00; пт 07:30–00:00; сб круглосуточно; вс 00:00–23:00
 пн круглосуточно; вт-чт 12:00–00:00; пт 12:00–02:00; сб 11:00–02:00; вс 14:00–00:00
 пн-чт 08:00–23:00; пт,сб круглосуточно; вс 08:00–23:00
 пн-ср 07:00–23:00; чт 07:00–00:00; пт,сб круглосуточно; вс 00:00–23:00
 пн-чт 08:00–23:00; пт 08:00–00:00; сб круглосуточно; вс 00:00–23:00
 пн 08:00–23:00; вт-пт 08:00–00:00; сб круглосуточно; вс 00:00–23:00
 пн-ср 09:00–00:00; чт-вс круглосуточно
 пн-чт круглосуточно; пт 00:00–05:00, перерыв 05:00–07:00; сб 07:00–05:00; вс 07:00–00:00
 пн-чт 11:00–23:00; пт 11:00–00:00; сб круглосуточно; вс 00:00–23:00
 пн,вт 08:00–22:00; ср,чт 08:00–23:00; пт,сб круглосуточно; вс 00:00–22:00
 пн-чт 07:00–23:00; пт,сб круглосуточно; вс 08:00–23:00
 пн,вт 07:30–23:00; ср-вс круглосуточно
 пн-ср 08:00–22:00; чт 08:00–23:00; пт,сб круглосуточно; вс 00:00–22:00
 пн-чт 09:00–00:00; пт,сб круглосуточно; вс 09:00–00:00
 пн-чт 07:00–22:00; пт,сб круглосуточно; вс 00:00–22:00
 пн-пт 09:00–17:00; сб,вс круглосуточно
 пн-ср 08:00–23:00; чт 08:00–00:00; пт,сб круглосуточно; вс 00:00–23:00
 пн-чт 08:00–00:00; пт,сб круглосуточно; вс 08:00–00:00
 пн-чт 08:00–22:00; пт 08:00–00:00; сб круглосуточно; вс 00:00–22:00
 пн,вт 10:00–00:00; ср-вс круглосуточно
 пн 06:00–00:00; вт-вс круглосуточно
 пн-чт 10:00–00:00; пт-вс круглосуточно
 пн-чт 07:00–23:00; пт 07:00–00:00; сб круглосуточно; вс 00:00–23:00
 пн-чт 10:00–23:00; пт 10:00–00:00; сб круглосуточно; вс 00:00–23:00
 пн 08:00–23:00; вт-вс круглосуточно
 чт круглосуточно, перерыв 10:00–20:00; сб круглосуточно
 пн-пт круглосуточно; сб 09:00–22:00; вс круглосуточно
 пн-чт 10:00–00:00; пт,сб круглосуточно; вс 10:00–00:00
 пн-пт круглосуточно; сб,вс 00:00–01:00
 пн,вт,ср,чт,сб,вс круглосуточно
 пн-ср 08:00–00:00; чт-сб круглосуточно; вс 08:00–00:00
 пн-чт 07:00–00:00; пт,сб круглосуточно; вс 07:00–00:00

```
In [9]: # единственный подходящий вариант - 'ежедневно, круглосуточно'
# на его основе создадим столбец is_24/7 с обозначением, что заведение работает ежедневно
df_cafe['is_24/7'] = df_cafe['hours'] == 'ежедневно, круглосуточно'
```

```
In [10]: #посмотрим какие административные районы мск в принципе есть
county = list(df_cafe['district'].unique())
county
```

```
Out[10]: ['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']
```

Неудобно и длинно - лучше переименовать

```
In [11]: district = dict()
for count in county:
    district[count] = ''.join([c for c in count.title() if c.isupper()])
district
```

```
Out[11]: {'Северный административный округ': 'CAO',
'Sеверо-Восточный административный округ': 'CBAO',
'Sеверо-Западный административный округ': 'CZA0',
'Западный административный округ': 'ZA0',
'Центральный административный округ': 'ЦАО',
'Восточный административный округ': 'BAO',
'Юго-Восточный административный округ': 'ЮBAO',
'Южный административный округ': 'ЮАО',
'Юго-Западный административный округ': 'ЮЗАO'}
```

```
In [12]: for key,value in district.items():
df_cafe.loc[df_cafe['district'] == key, 'district'] = value
df_cafe.head(1)
```

```
Out[12]:
```

	name	category	address	district	hours	lat	lng	rating	price	avg_bill
0	WoWФли	кафе	Москва, улица Дыбенко, 7/1	CAO	ежедневно, 10:00–22:00	55.878494	37.47886	5.0	NaN	NaN

```
In [13]: df_cafe['category'] = df_cafe['category'].str.title()
```

Краткий вывод

Создан столбец с названиями улиц, на которых расположены заведения, столбец в обозначении работает ли заведение ежедневно и круглосуточно. Сокращены названия округов. В данной работе пропуски не мешают анализу заведений общепита, не будем обрабатывать.

Анализ данных

Категории заведений

```
In [14]: # готовим данные для графика
fig = px.histogram(df_cafe, # загружаем данные
                  x='category', # указываем столбец с данными для оси X
                  title='Заведения по категориям', # указываем заголовок
                  nbins=1000, # назначаем число корзин
                  barmode='overlay') # выбираем «полупрозрачный» тип отображения
fig.update_xaxes(title_text='Категории') # подпись для оси X
fig.update_yaxes(title_text='Количество') # подпись для оси Y
fig.show() # выводим график
```

Заведения по категориям



В датасете больше всего кафеен(2378шт.), следом с небольшим отставанием ретораны(2043). Меньше всего булочных и столовых. Похоже, кафейни очень популярны. Надеемся, выручка у них тоже соответствующая - хотябы за счет количества посетителей. Доступа к этим данным у нас нет, зато клиент вернется если доволен - поэтому посмотрим на медиану оценок в каждой категории и на средний чек

```
In [15]: df_cafe.pivot_table(index='category', values=['rating', 'middle_avg_bill', 'middle_
```

Out[15]:

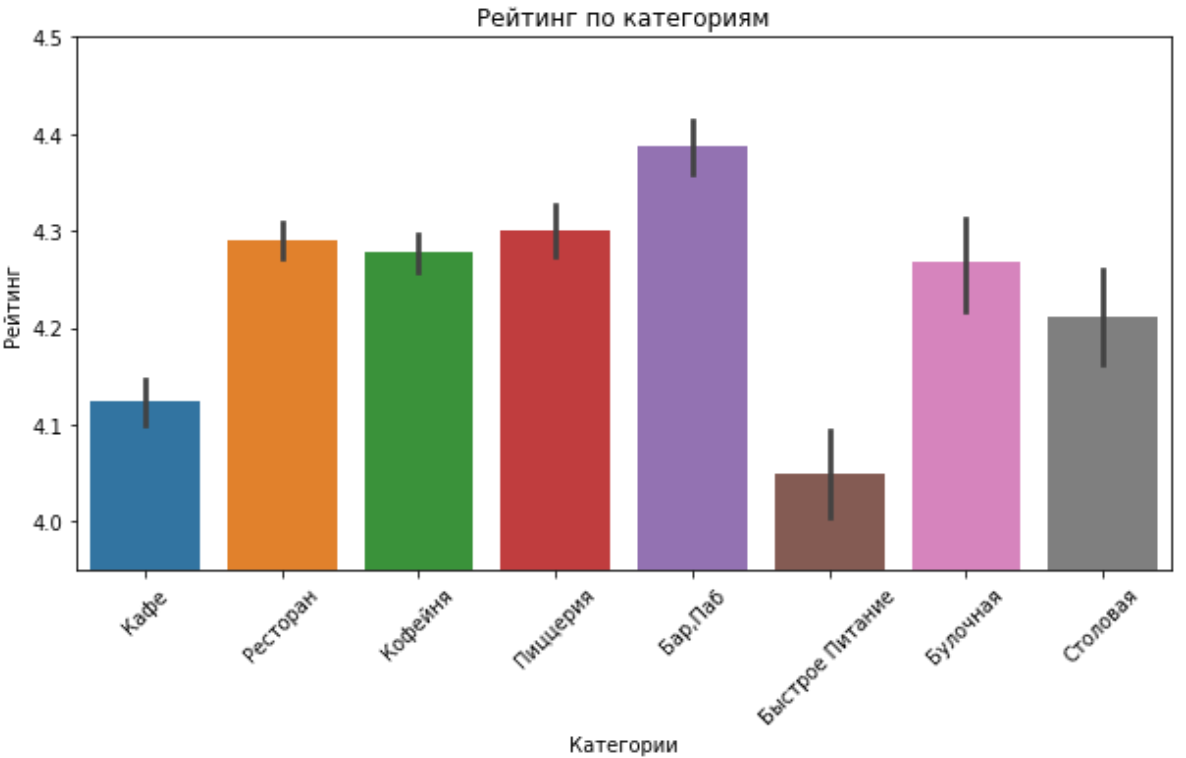
category	mean			media		
	middle_avg_bill	middle_coffee_cup	rating	middle_avg_bill	middle_coffee_cup	rating
Бар,Паб	1338.762178	208.333333	4.387712	1250.0	202.5	4.3
Булочная	658.773585	NaN	4.268359	450.0	NaN	4.3
Быстрое Питание	445.763713	140.000000	4.050249	375.0	140.0	4.3
Кафе	707.753602	105.500000	4.123886	550.0	111.0	4.3
Кофейня	614.210000	175.055662	4.277282	400.0	170.0	4.3
Пиццерия	789.377215	153.333333	4.301264	600.0	150.0	4.3
Ресторан	1367.881731	NaN	4.290357	1250.0	NaN	4.3
Столовая	335.348066	NaN	4.211429	300.0	NaN	4.3

Везде примерно одинаковая оценка 4.3. Зато больше всего денег за раз оставляю в барах и ресторанах - логично. Странно, что для булочной не нашлось цены за чашку кофе - кто же откажется от кофе с круассаном? Видно, что медиана и среднее отличаются. Возможно, есть какие-то выбросы.

In [16]:

```
plt.figure(figsize=(10,5))
ax = sns.barplot(x='category', y='rating', data=df_cafe)

plt.ylim(3.95,4.5)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
plt.xlabel('Категории')
plt.ylabel('Рейтинг')
plt.title('Рейтинг по категориям')
plt.show()
```



Среднее значение тоже не сильно отличается: от 4,13 до 4,4. У булочных, столовых и быстрого питания больше других среднеквадратичное отклонение, а у ресторанов и кофеен - самое маленькое.

Количество мест по категориям

```
In [17]: df_cafe.pivot_table(index='category', values='seats', aggfunc=['min','median','ma
```

Out[17]:

	min	median	max
	seats	seats	seats
category			
Бар,Паб	0.0	82.5	1288.0
Булочная	0.0	50.0	625.0
Быстрое Питание	0.0	65.0	1040.0
Кафе	0.0	60.0	1288.0
Кофейня	0.0	80.0	1288.0
Пиццерия	0.0	55.0	1288.0
Ресторан	0.0	86.0	1288.0
Столовая	0.0	75.5	1200.0

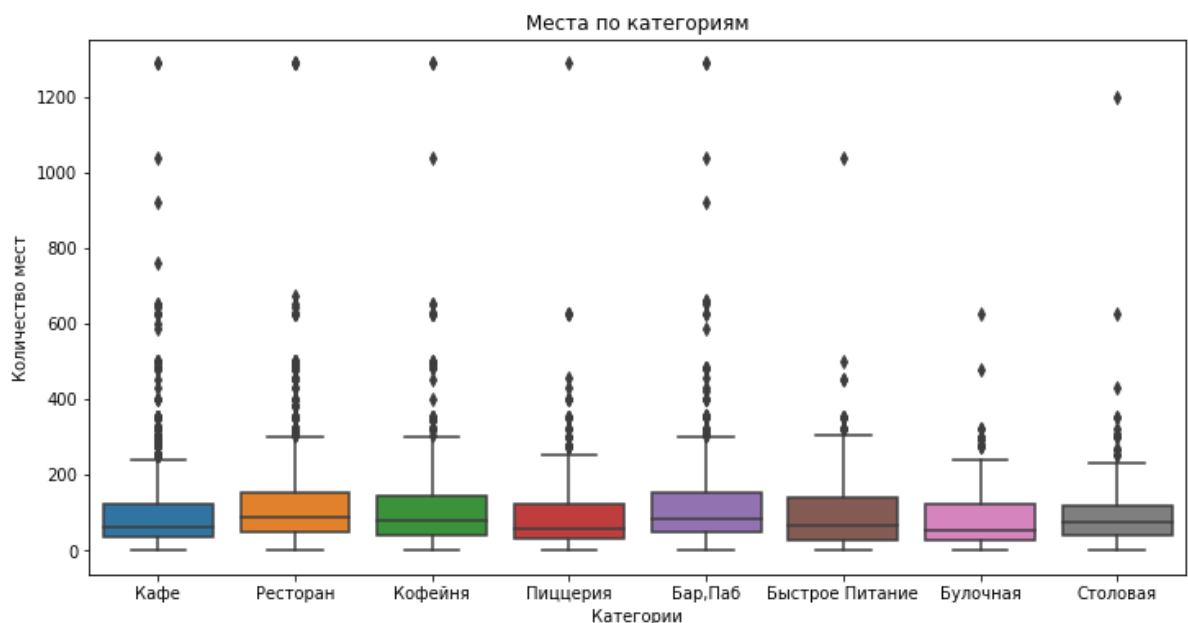
Медиана варьируется от 50 о 86 посаточных мест. Заведения с 0 стульев? Посмотрим что это

```
In [18]: df_cafe[df_cafe['seats']==0]['name'].unique()
```

```
Out[18]: array(['Meat Doner Kebab', 'Арамье', 'Донер-Шашлык', 'Тандыр № 1',
      'Неаполитан пицца', 'Пекарня&Донер', 'Центр Плов', 'Шаурма',
      'Everest Coffee', 'Пекарня Маковка', 'Паб 28/13', 'Рандеву',
      'Огонек', 'Кофе с собой A&M', 'Пекарня 24', 'Sushi-das.ru',
      'Халяль', 'Стейки Bar-B-Que', 'Дом плова', 'Кулинария',
      'Нуш донер', 'Домино'с Пицца", 'Кулинария Виктория',
      'Сладкая параллель', 'Рожь Хлеб и Кофе', 'Афросиаб', 'ПекарняУз',
      'Шаурма и Таук', 'I-cup', 'French Bakery', 'Тимир', 'Cofix',
      'Wild Bean Cafe', '9 Bar Coffee', 'Сказка Египта', 'Street coffee',
      'Coffee in', 'O! Фобо', 'Намшон', 'Пицца и гирос', 'Шаверма',
      'Кофе с собой', 'Вьетнамская кухня', 'Органик', 'Здоровое Питание',
      'КлинКом', 'Andy Coffee', 'Моремэй', 'Детилэнд', 'СушиСтор',
      'Чайхана', 'КИНОпицца', 'Sugarbey', 'Семетей', 'Яндекс Лавка',
      'Чайхана Семетей', 'Мангал', 'Стумари', 'Таманно', 'Signature',
      'Мысли кофе', 'Elephantkids', 'Столовая на Шаболовке',
      'Донер кебаб', 'Piccolo Coffee', 'Кулинариум', 'Бистро 24',
      'Шашлычная77', 'Пиццерия Пауло Виктория', 'Пицца Паоло', 'БроКофе',
      'Shawarma', 'Лайфхакер кофе', 'Система', 'Хлеб да обед',
      'Чайхана Халва', 'ШашлыкоFF', 'Выдра кофе', 'Вьет Лотос', 'Bỏ',
      'Мясо на углях', 'One Price Coffee', "Manny's Burger",
      'Monkey Pizza', 'Дон Хулио', 'Здрасте', 'Адыгская кухня',
      'Take and Wake', 'MYration', 'Иссык-Куль', 'Main Food',
      'Крошка Картошка', 'Додо Пицца', 'Теремок',
      'Азербайджанская кухня', 'Плов лагман', 'Восточный уголок',
      'Сладко', '1-я Креветочная', 'Кафе', 'Японская кухня',
      'Роллы суши и десерт', 'Чайхана Ташкент', 'Мишель',
      'Кафе Халяль Плов № 1', 'Bravos', 'Четыре Пекаря', 'Баам-кафе',
      'Хинкали хачапури', 'Суши Хай', 'Лига Шашлыков', 'Куманёк',
      'Wild Bean', 'Орхан', 'Бико', 'Чайхана УЧ Кудук', 'Масса кофе',
      'Сочная шаурма в Кузьминках', 'Достор'], dtype=object)
```

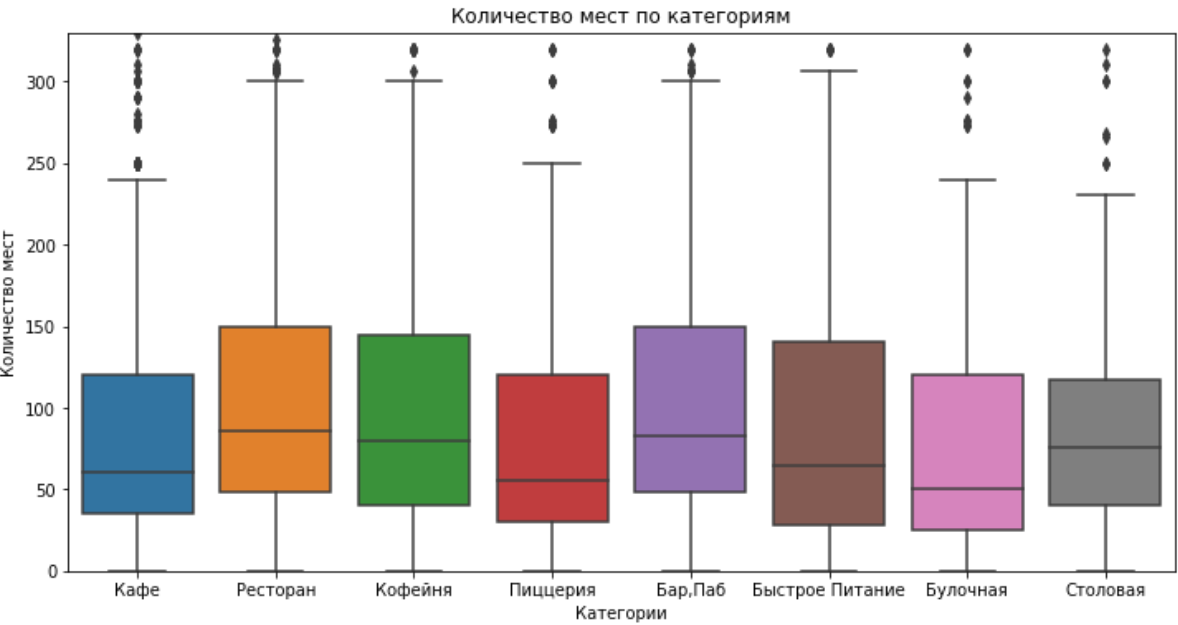
Meat Doner Kebab? Шаверма? Похоже, это заведения с окошком для выдачи - у них и не должно быть сидений. Посмотрим на "Ящик с усами"

```
In [19]: plt.figure(figsize=(12,6))
#plt.ylim(0,1.6)
sns.boxplot(x='category',y='seats',data=df_cafe)
plt.xlabel('Категории')
plt.ylabel('Количество мест')
plt.title('Места по категориям')
plt.show()
```



Видно, что почти в каждой категории есть выбросы. Однако, если их не учитывать, все выглядит очень похоже. Взглянем поближе

```
In [20]: plt.figure(figsize=(12,6))
plt.ylim(0,330)
sns.boxplot(x='category',y='seats',data=df_cafe)
plt.xlabel('Категории')
plt.ylabel('Количество мест')
plt.title('Количество мест по категориям')
plt.show()
```



Теперь видно различие. Как и могло бы ожидаться - больше всего мест будет в ресторанах и барах, зато в булочной/пиццерии/кафе очень часто дают на вынос. Не удивительно, что первые квартили у них самые низкие.

Сетевые заведения

```
In [21]: temp = df_cafe.groupby('chain')['name'].agg('count').rename(index = {0:'Несетевые'},
temp = pd.DataFrame([temp, temp / df_cafe.shape[0] ], index=['Количество', 'Процент']
# как победить 5 нулей в "Количество" и оставить процент во 2 столбце?
temp
```

Out[21]:

	Количество	Процент
chain		
Несетевые	5201.000000	61.87%
Сетевые	3205.000000	38.13%

Одиночных заведений - 61.87% датасета - явно больше.

```
In [22]: def one_plot_creator(df, x, y, color, text=None,
title='default tittle', xaxis_title='default xaxis tittle', ya
legend_title='default legend tittle', barmode='stack',
orientation='v', height=500, showlegend=True, for_export=False

fig = px.bar(df, x=x, y=y, color=color, text=text, barmode=barmode)
```

```

fig.update_layout(
    height=height,
    showlegend=showlegend,
    title=title,
    xaxis_title=xaxis_title,
    yaxis_title=yaxis_title,
    legend_title=legend_title
)

if for_export == True:
    fig.update_layout({
        'plot_bgcolor': 'rgba(0, 0, 0, 0)',
        'paper_bgcolor': 'rgba(0, 0, 0, 0)',
    })

return fig

```

```

In [23]: chains = df_cafe.groupby(by=['category', 'chain'], as_index=False).agg(count=('name', 'count'))
chains.loc[chains['chain'] == 0, 'chain'] = 'Не сетевой'
chains.loc[chains['chain'] == 1, 'chain'] = 'Сетевой'
for i in range(len(chains)):
    chains.at[i, 'ratio'] = str(round(chains.loc[i]['count']/len(df_cafe[df_cafe['category'] == chains.loc[i]['category']], 2)))

```

```

In [24]: fig = one_plot_creator(
df=chains,
x='category', y='count', color='chain', text='ratio',
title='Соотношение сетевых и не сетевых заведений по категориям',
xaxis_title='Категория', yaxis_title='Количество',
legend_title='Обозначение', height=450
)
fig.show()

```

Соотношение сетевых и не сетевых заведений по категориям



Кафе представлено в датасете большим количеством, в нем также больше всего не сетевых заведений. А вот в кофейнях соотношение 50/50. Во всех категориях

несетевых больше, кроме булочных: 60/40.

Топ-15 сетевых

```
In [25]: # отфильтруем данные, сгруппируем по имени и посчитаем объявления
df_loc_count = df_cafe.loc[df_cafe['chain'] == 1].groupby('name')[['name']].count()
# переименуем столбец
df_loc_count.columns = ['total_count']
# отсортируем и оставим лидеров
df_loc_count = df_loc_count.reset_index().sort_values(by='total_count', ascending=False)

# строим столбчатую диаграмму
fig = px.bar(df_loc_count.sort_values(by='total_count', ascending=True), # загружаем
             x='total_count', # указываем столбец с данными для оси X
             y='name', # указываем столбец с данными для оси Y
             text='total_count' # добавляем аргумент, который отобразит текст с информацией
             # о количестве объявлений внутри столбца графика
             )
# оформляем
fig.update_layout(title='ТОП-15 заведений',
                  xaxis_title='Количество заведений',
                  yaxis_title='Заведение')
fig.show()
```

ТОП-15 заведений



Похоже, больше всего заведений открыла Шоколадница - целых 120 в Москве. За ним пара пиццерий: Домино'с и Додо пицца. Все более-менее знакомые

Посмотрим на категории заведений к которым относятся топ-15 сетевых.

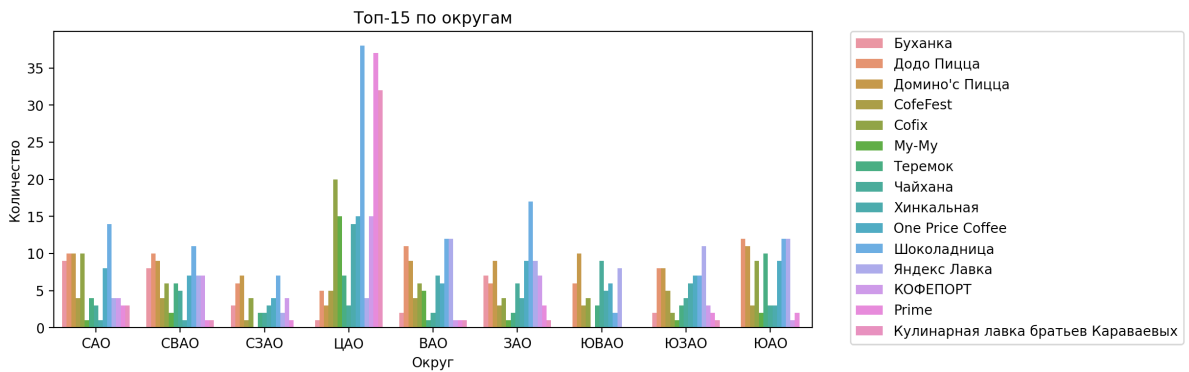
```
In [26]: chain_top15 = df_cafe[df_cafe['name'].isin(df_loc_count['name'])]
temp = chain_top15.pivot_table(index='name', values='category',aggfunc='first')
display(temp)
temp.value_counts()
```

category	
name	
CofeFest	Кофейня
Cofix	Кофейня
One Price Coffee	Кофейня
Prime	Ресторан
Буханка	Булочная
Додо Пицца	Пиццерия
Домино'с Пицца	Пиццерия
КОФЕПОРТ	Кофейня
Кулинарная лавка братьев Караваевых	Кафе
Му-Му	Кафе
Теремок	Ресторан
Хинкальная	Быстрое Питание
Чайхана	Кафе
Шоколадница	Кофейня
Яндекс Лавка	Ресторан

```
Out[26]: category
Кофейня      5
Кафе         3
Ресторан     3
Пиццерия     2
Булочная    1
Быстрое Питание  1
dtype: int64
```

Итак, мы имеем 5 кофеен, 3 кафе или ресторана, 2 пиццерии и по 1 булочной ли быстрого питания. Шоколадница - самое массовое заведение, является кофейней

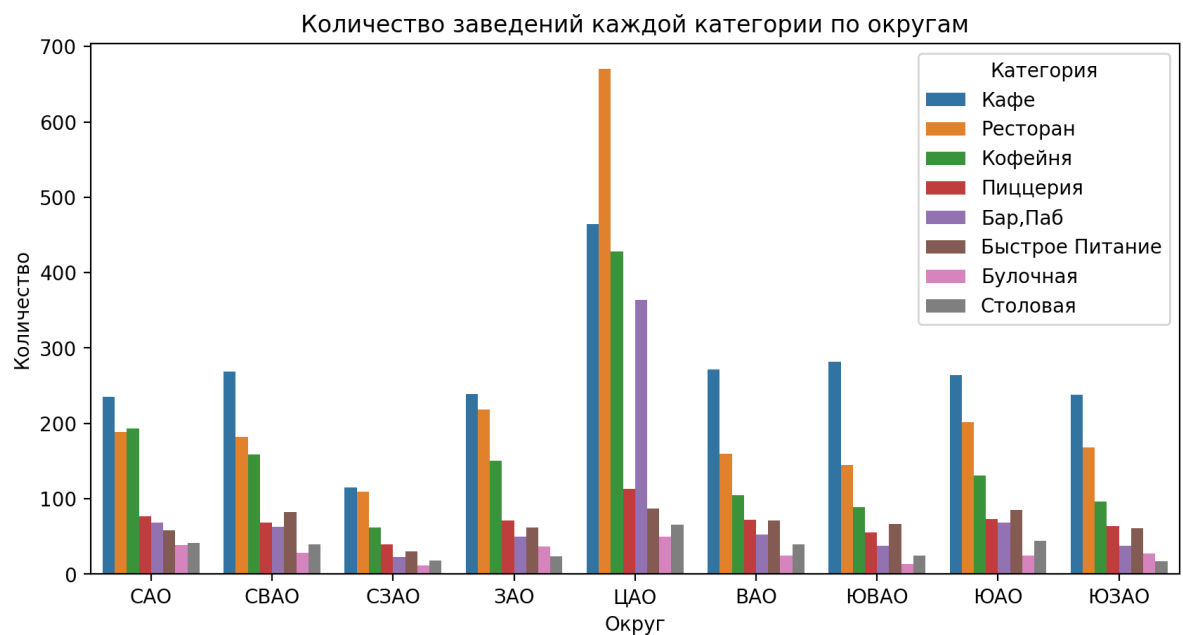
```
In [27]: plt.figure(figsize=(10,4),dpi=200)
my_plot = sns.countplot(x='district', data=chain_top15, hue='name')
plt.xlabel('Округ')
plt.ylabel('Количество')
plt.title('Топ-15 по округам')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



В ЦАО самым многочисленным является Шоколадница, следом - Prime, Кулинарная лавка братьев Караваевых. Наиболее малочисленная - Буханка. Зато ее заведений много относительно других в CAO, BAO и ЮАО.

Общий обзор сетевых и несетевых

```
In [28]: plt.figure(figsize=(10,5),dpi=200)
my_plot = sns.countplot(x='district',data=df_cafe,hue='category');
#my_plot.set_xticklabels(my_plot.get_xticklabels(), rotation=45)
plt.xlabel('Округ')
plt.ylabel('Количество')
plt.title('Количество заведений каждой категории по округам')
my_plot.legend(title='Категория');
```



Большинство заведений находится в центре - и это рестораны, также много кафе и кофеен. Да и целом, в какой округ не посмотри - везде лидируют кафе. Также примечательно, что в центре популярны бары, пабы - в отличие от других округов. В целом динамика очень похожа в каждом округе: сначала кафе, потом ресторан, кофейня, пиццерия/быстрое питание, бар, паб, столовая/булочная. Разве что в CAO рестораны чуть популярнее кофеен.

Хороплет со средним рейтингом

```
In [29]: # сформируем таблицу для визуализации
rating_df = df_cafe.groupby('district', as_index=False)['rating'].agg('mean')
```

```
for key,value in district.items():
    rating_df.loc[rating_df['district'] == value, 'district'] = key
rating_df
```

Out[29]:

	district	rating
0	Восточный административный округ	4.174185
1	Западный административный округ	4.181551
2	Северный административный округ	4.239778
3	Северо-Восточный административный округ	4.148260
4	Северо-Западный административный округ	4.208802
5	Центральный административный округ	4.377520
6	Южный административный округ	4.184417
7	Юго-Восточный административный округ	4.101120
8	Юго-Западный административный округ	4.172920

In [30]:


```
# 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, tiles='Cartodb Positron')

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=rating_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
```

Out[30]: Make this Notebook Trusted to load map: File → Trust Notebook



Медианный рейтинг заведений по районам

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

Похоже, в центре все же более щедры на хорошие оценки, хотя везде не менее 4. Ну или же конкуренция там больше. Самый низкооцениваемый - ЮВАО: 4.10. Однако, факт остается: в ЦАО почти 4.38.

Все заведения на карте

```
In [31]: # создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles="Cartodb Positron")
# 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_cafe.apply(create_clusters, axis=1)

# выводим карту
m
```

out[31]: Make this Notebook Trusted to load map: File -> Trust Notebook

+

—

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

```
In [32]: # отфильтруем данные, сгруппируем по имени и посчитаем объявления
df_loc_count = df_cafe.loc[df_cafe['district'] == 'ЦАО'].groupby('category')[['cate
# переименуем столбец
df_loc_count.columns = ['total_count']
# отсортируем и оставим лидеров
df_loc_count = df_loc_count.reset_index().sort_values(by='total_count', ascending=F

# строим столбчатую диаграмму
fig = px.bar(df_loc_count.sort_values(by='total_count', ascending=True), # загружае
             x='total_count', # указываем столбец с данными для оси X
             y='category', # указываем столбец с данными для оси Y
             text='total_count' # добавляем аргумент, который отобразит текст с инф
                               # о количестве объявлений внутри столбца графика
             )
# оформляем
fig.update_layout(title='Количество заведений по категориям в ЦАО',
                  xaxis_title='Категория',
                  yaxis_title='Заведение')
fig.show()
```

Количество заведений по категориям в ЦАО



```
In [33]: print(df_cafe.loc[df_cafe['district'] == 'ЦАО'].shape)
print(df_cafe.loc[df_cafe['district'] != 'ЦАО'].shape)

(2242, 16)
(6164, 16)
```

Топ-15 улиц с заведениями

```
In [34]: streets = df_cafe['street'].value_counts()
streets

Out[34]: проспект Мира      184
Профсоюзная улица      122
проспект Вернадского    108
Ленинский проспект     107
Ленинградский проспект   95
...
Композиторская улица      1
Токмаков переулок        1
Новоясеневский тупик      1
Курсовой переулок        1
Новошуйкинская улица      1
Name: street, Length: 1448, dtype: int64

In [35]: street_cat = df_cafe[df_cafe['street'].isin(streets.head(15).index)].groupby(['street',
count=('name', 'count'))).sort_values('count', ascending=False)

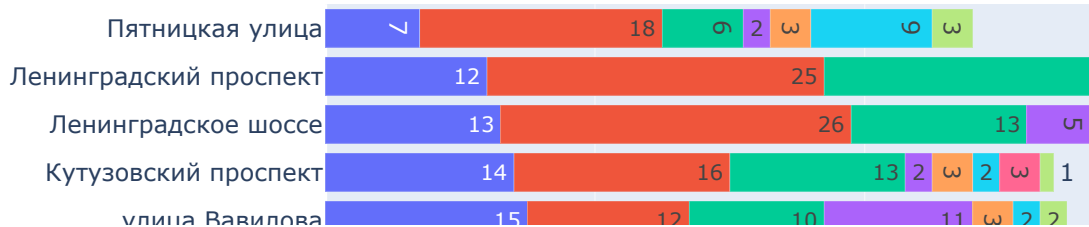
fig = one_plot_creator(
df=street_cat,
x='count', y='street', color='category', text='count',
```

```

title='Топ-15 улиц по количеству заведений',
xaxis_title='Количество заведений', yaxis_title='Улица',
legend_title='Категория',
orientation='h'
)
fig.show()

```

Топ-15 улиц по количеству заведений



Похоже, больше всего заведений на Проспекте Мира. Это не удивительно: улица длинная и недалеко от центра. На каждой улице по количеству лидируют либо кафе, либо рестораны.

Теперь рассмотрим заведения, которые по каким-то причинам оказались в 1 экземпляре на 1 улице.

```

In [36]: names = df_cafe['street'].value_counts(ascending=True)
just_one_cafe = df_cafe[df_cafe['street'].isin(names[names < 2].index)]
just_one_cafe.head()

```


Out[36]:

	name	category	address	district	hours	lat	lng	rating	
15	Дом обеда	Столовая	Москва, улица Бусиновская Горка, 2	САО	пн-пт 08:30–18:30; сб 10:00–20:00	55.885890	37.493264	4.1	ср
21	7/12	Кафе	Москва, Прибрежный проезд, 7	САО	ежедневно, 10:00–22:00	55.876805	37.464934	4.5	
25	В парке вкуснее	Кофейня	Москва, парк Левобережный	САО	ежедневно, 10:00–21:00	55.878453	37.460028	4.3	
58	Coffeekaldi's	Кофейня	Москва, Угличская улица, 13, стр. 8	СВАО	ежедневно, 09:00–22:00	55.900316	37.570558	4.1	ср
60	Чебуречная история	Кофейня	Москва, ландшафтный заказник Лианозовский	СВАО	ежедневно, 10:00–22:00	55.899845	37.570488	4.9	

На первый взгляд ничего необычного. Возможно, эти заведения находятся на окраинах - поэтому из по одному?

In [37]:

```
# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles="Cartodb Positron")
# 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
just_one_cafe.apply(create_clusters, axis=1)

# выводим карту
m
```

Out[37]: Make this Notebook Trusted to load map: File -> Trust Notebook

Leaflet (https://leafletjs.com) | © OpenStreetMap (http://www.openstreetmap.org/copyright) contributors © CartoDB (http://cartodb.com/attributions), CartoDB attributions (http://cartodb.com/attributions)

При общем взгляде снова ничего необычного. Однако, если приближать карту - видно, что большинство заведений находятся в переулках или тупиках - то есть улицах со своими небольшими названиями. В таких укромных местах выгодно открыть одиночную кофейню или кафе. Проверим

```
In [38]: chains = just_one_cafe.groupby(by=['category', 'chain'], as_index=False).agg(count=
chains.loc[chains['chain'] == 0, 'chain'] = 'Одиночный'
chains.loc[chains['chain'] == 1, 'chain'] = 'Сеть'
for i in range(len(chains)):
    chains.at[i, 'ratio'] = str(round(
        chains.loc[i]['count']/len(just_one_cafe[just_one_cafe['category'] == chain
```

```
In [39]: fig = one_plot_creator(
    df=chains,
    x='category', y='count', color='chain', text='ratio',
    title='Категории на одиночных улицах',
    xaxis_title='Категория', yaxis_title='Количество',
    legend_title='Обозначение', height=450
)
fig.show()
```

Категории на одиночных улицах



Действительно - очень много несетевых кафе и кафеен. Однако также довольно много ресторанов - это, скорее всего, потому что многие находятся в центре.

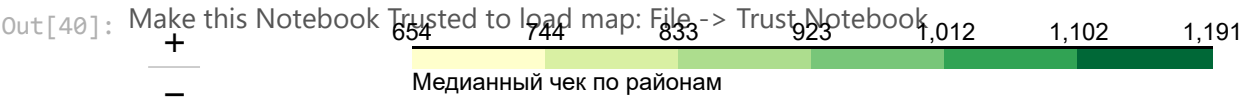
Исследование о среднем чеке

```
In [40]: avg_bill_df = df_cafe.groupby('district', as_index=False)['middle_avg_bill'].agg('n
for key,value in district.items():
    avg_bill_df.loc[avg_bill_df['district'] == value, 'district'] = key

# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles='Cartodb Positron')

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=avg_bill_df,
    columns=['district', 'middle_avg_bill'],
    key_on='feature.name',
    fill_color='YlGn',
    fill_opacity=0.8,
    legend_name='Медианный чек по районам',
).add_to(m)

# выводим карту
m
```



Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

```
In [41]: for key,value in district.items():
        avg_bill_df.loc[avg_bill_df['district'] == value, 'district'] = key
avg_bill_df
```

Out[41]:

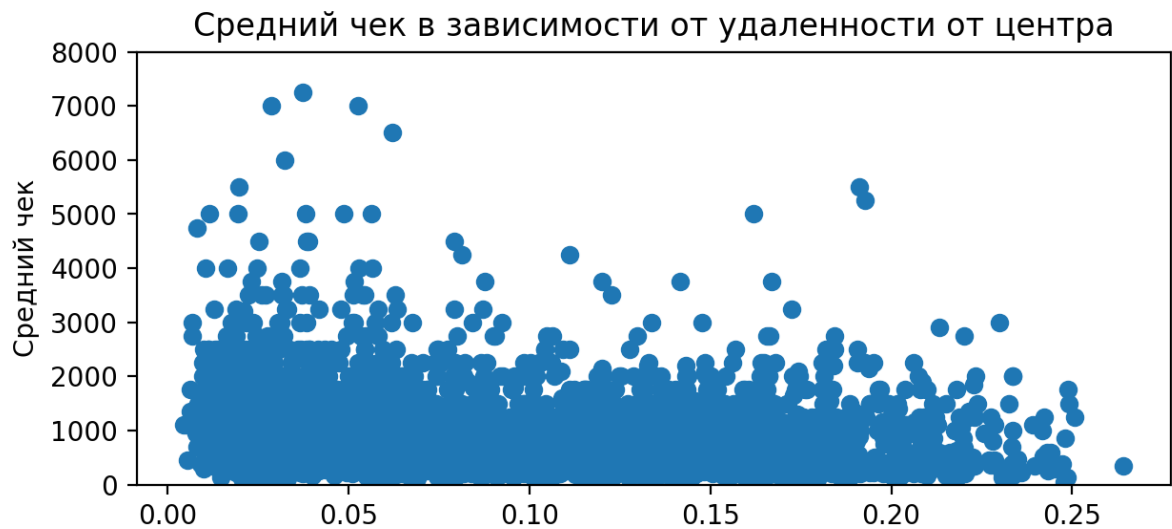
	district	middle_avg_bill
0	Восточный административный округ	820.626923
1	Западный административный округ	1053.225490
2	Северный административный округ	927.959627
3	Северо-Восточный административный округ	716.611296
4	Северо-Западный административный округ	822.222930
5	Центральный административный округ	1191.057547
6	Южный административный округ	834.398089
7	Юго-Восточный административный округ	654.097938
8	Юго-Западный административный округ	792.561702

Центр и ЗАО считаются хорошими районами даже по стоимости жилья - не удивительно, что у них и средний чек выше. Однако, самый дорогой район все же центральный.

Посмотрим на зависимость среднего чека от удаленности от центра. Для этого воспользуемся координатами

```
In [42]: # сместили центр, посчитали гипотенузу
df_cafe['radius_from_center'] = np.sqrt((df_cafe['lat'] - moscow_lat) ** 2 + (df_ca
```

```
In [43]: plt.figure(figsize=(7,3),dpi=200)
plt.ylabel('Средний чек')
plt.xlabel('')
plt.ylim(0,8000)
plt.title('Средний чек в зависимости от удаленности от центра')
plt.scatter(df_cafe['radius_from_center'], df_cafe['middle_avg_bill']);
```



Есть ожидаемая зависимость: чем дальше от центра, тем, в среднем, дешевле.

```
In [44]: print(df_cafe['radius_from_center'].corr(df_cafe['middle_avg_bill']))
-0.14708116993017434
```

Анализ круглосуточных заведений

```
In [45]: # отфильтруем данные, сгруппируем по имени и посчитаем объявления
df_loc_count = df_cafe[df_cafe['is_24/7']].groupby('category')[['category']].count()
# переименуем столбец
df_loc_count.columns = ['total_count']
# отсортируем и оставим пять лидеров
df_loc_count = df_loc_count.reset_index().sort_values(by='total_count', ascending=False)

# строим столбчатую диаграмму - использовать вместо того кода, что ниже
fig = px.bar(df_loc_count.sort_values(by='total_count', ascending=True), # загружаем
             x='total_count', # указываем столбец с данными для оси X
             y='category', # указываем столбец с данными для оси Y
             text='total_count' # добавляем аргумент, который отобразит текст с инф
                               # о количестве объявлений внутри столбца графика
             )
# оформляем график
fig.update_layout(title='Категории, работающие круглосуточно',
                  xaxis_title='Количество заведений',
                  yaxis_title='Категория')
fig.show() # выводим график
```

Категории, работающие круглосуточно



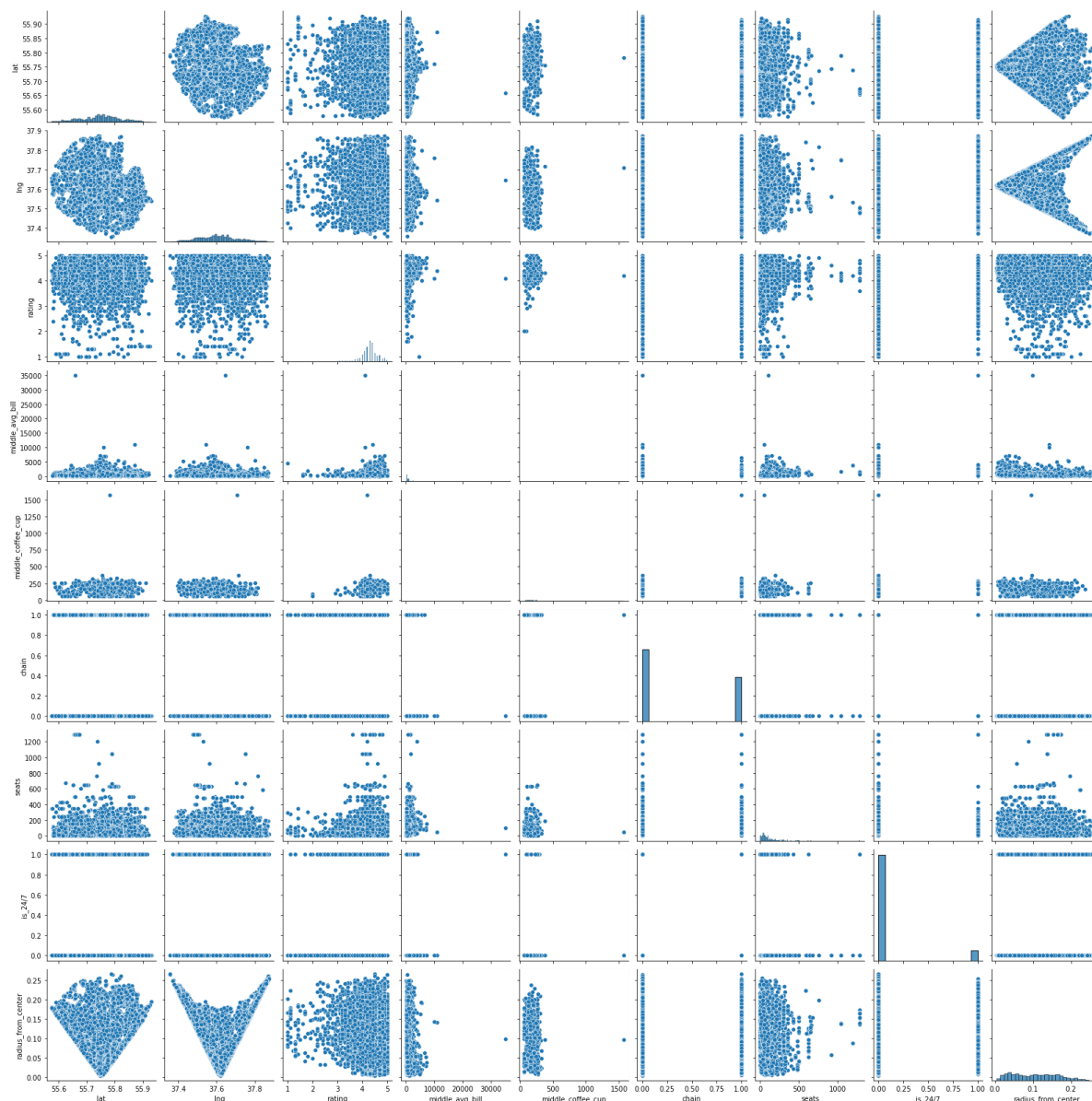
Больше всего круглосуточно работают кафе и быстрое питание.

Другие зависимости

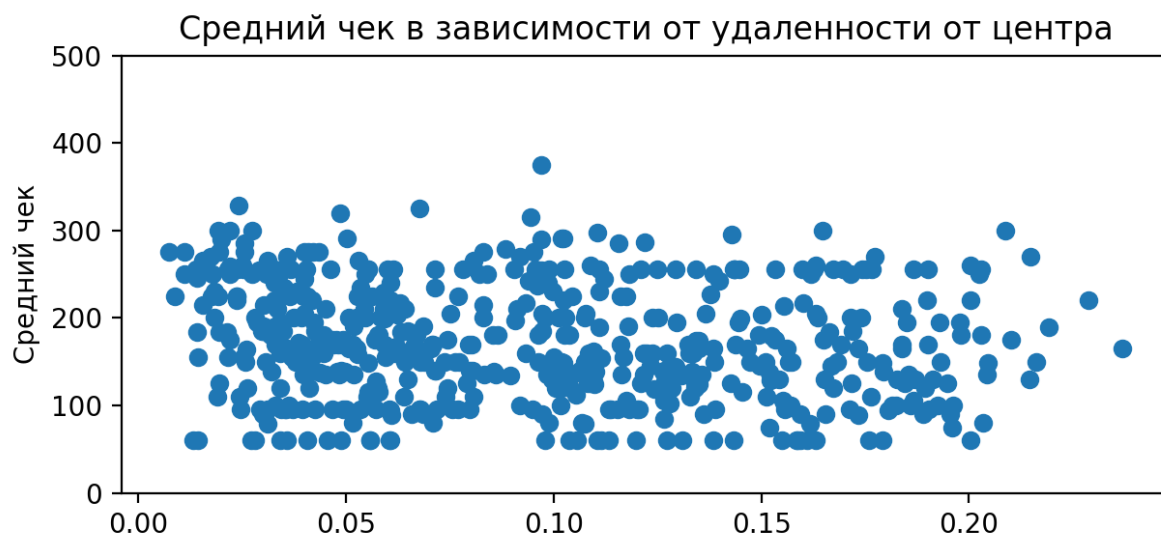
```
In [46]: # посмотрим какие другие интересные зависимости можно рассмотреть  
sns.pairplot(df_cafe)
```

```
<__array_function__ internals>:5: RuntimeWarning:  
Converting input from bool to <class 'numpy.uint8'> for compatibility.  
  
<__array_function__ internals>:5: RuntimeWarning:  
Converting input from bool to <class 'numpy.uint8'> for compatibility.
```

```
Out[46]: <seaborn.axisgrid.PairGrid at 0x7fc8e56fb730>
```



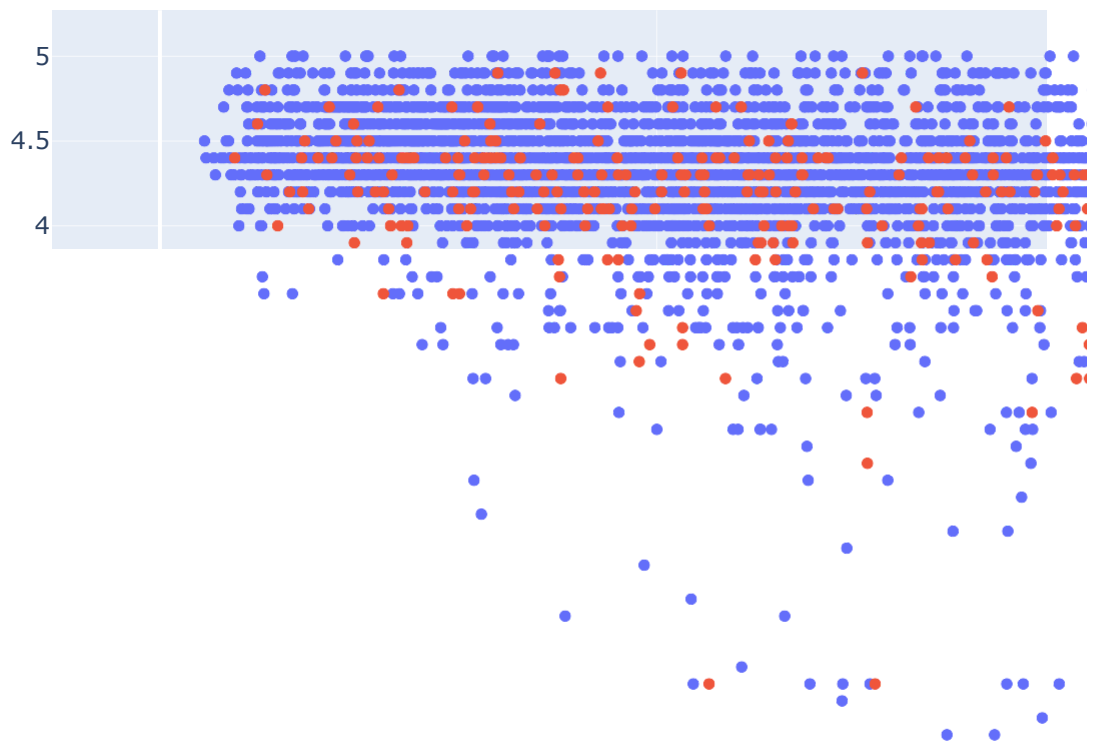
```
In [47]: plt.figure(figsize=(7,3),dpi=200)
plt.ylabel('Средний чек')
plt.xlabel('')
plt.ylim(0,500)
plt.title('Средний чек в зависимости от удаленности от центра')
plt.scatter(df_cafe['radius_from_center'], df_cafe['middle_coffee_cup']);
```



Интересно, что стоимость чашки кофе не зависит от удаленности от центра.

```
In [48]: # cmpoum scatter
fig = px.scatter(df_cafe.rename(columns = {'is_24/7': 'Круглосуточный'}),
                 x='radius_from_center', # указываем столбец с данными для оси X
                 y="rating", # указываем столбец с данными для оси Y
                 color='Круглосуточный') # обозначаем категорию для разделения цветов
# оформляем график
fig.update_layout(title='Зависимость рейтинга от удаленности от центра',
                  yaxis_title='Рейтинг',
                  xaxis_title='Удаленность от центра')
my_plot.legend(title='Категория');
fig.show() # выводим график
```

Зависимость рейтинга от удаленности от центра

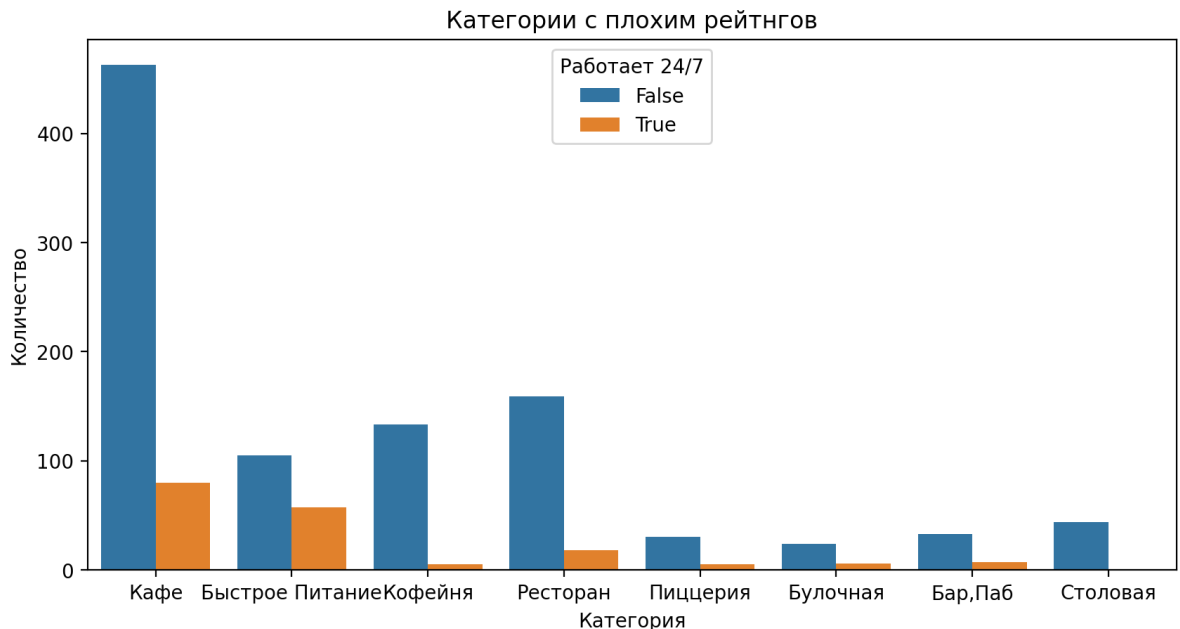


Также в средней удаленности от центра чаще других ставят низкие оценки.

```
In [49]: failed_rat = df_cafe.query('rating < 4')
print("Средний чек в заведениях с рейтинком меньше 4 = {0:.2f} руб.".format(failed_rat['avg_check'].mean()))

Средний чек в заведениях с рейтинком меньше 4 = 544.26 руб.
```

```
In [50]: plt.figure(figsize=(10,5),dpi=200)
my_plot = sns.countplot(x='category',data=failed_rat,hue='is_24/7');
#my_plot.set_xticklabels(my_plot.get_xticklabels(), rotation=45)
plt.xlabel('Категория')
plt.ylabel('Количество')
plt.title('Категории с плохим рейтингом')
my_plot.legend(title='Работает 24/7');
```

Кафе в датасете в принципе много. Быстрое питание почти сравнялось с количеством заведений, работающих 24/7. Очень много ресторанов.

Краткий вывод

Проанализированы **категории предоставленных** данных:

- больше всего кафеен(2378шт) и реторанов(2043шт)
- меньше всего булочных и столовых
- больше всего денег за раз оставляют в барах и ресторанах

Была построена **гистограмма средних оценок по категориям**, по ней замечено, что **средняя оценка** колеблется **от 4,13 до 4,4**. У булочных, столовых и быстрого питания больше других среднеквадратичное отклонение, а у ресторанов и кофеен - самое маленькое.

Было проанализировано **количество посадочных мест в зависимости от категории**

- медианы варьируются от 50 до 86 посадочных мест
- построен ящик с усами: Есть выбросы с количеством стульев более 1200. **Больше всего мест в ресторанах и барах**, зато в булочной/пиццерии/кафе очень часто дают на вынос - первые квартили у них самые низкие.

Проанализировано **количество сетевых и несетевых заведений**.

- несетевых 61.87% датасета
- построен график соотношения сетевых и несетевых заведений по категориям. Категория "кафе" представлено в датасете большим количеством, в нем также больше всего несетевых заведений. А вот в кофенях соотношение 50/50. Во всех категориях несетевых больше, кроме булочных: 60/40.

Далее выявили **топ-15 наиболее многочисленных сетевых точек** в Москве:

- больше всего заведений открыла Шоколадница - целых 120 . За ним пара пиццерий: Домино'с и Додо пицца.

- из 15 заведений 5 кофеен, 3 кафе или ресторана, 2 пиццерии и по 1 булочной и быстрого питания. Шоколадница - самое массовое заведение, является кофейней
- ЦАО самым многочисленным является Шоколадница, следом - Prime, Кулинарная лавка братьев Караваевых. Наиболее малочисленная - Буханка. Зато ее заведений много относительно в CAO, BAO и ЮАО.

Если же смотреть **общее количество заведений каждой категории по округам**

- большинство находится в центре: это рестораны, кафе и кофейни
- везде много кафе
- в центре особенно популярны бары, пабы - в отличие от других округов
- динамика повторяется в каждом округе: очень много кафе, потом ресторанов, кофеен, пиццерий/быстрого питания, баров, пабов, столовых/булочных. Разве что в CAO рестораны чуть популярнее кофеен.

Также построен **хороплет со средним рейтингом**: в центре все же более щедры на хорошие оценки. Самый **низкооцениваемый** - ЮВАО: 4.10. Однако, факт остается: в **ЦАО почти 4.38**.

С помощью кластера были **отмечены на карте Москвы все заведения**, присутствующие в датасете. Была построена **гистограмма для топ-15 улиц по количеству заведений**:

- лидирует Проспект Мира. Это не удивительно: улица длинная и недалеко от центра
- на каждой улице по количеству лидируют либо кафе, либо рестораны. Далее посмотрели на **улицы**, на которых находится **1 заведение**. Выяснено, что большинство заведений находятся **в переулках или тупиках** - то есть улицах со своими небольшими названиями, либо же далеко от центра. В таких укромных местах выгодно открыть одиночную кофейню или кафе.

Построен **хороплет с медианным чеком по районам**: у центра и ЗАО **средний чек выше других**. Самый дорогой район - центральный. Построен график рассеяния: чем дальше от центра, тем в среднем, дешевле.

Также были выявлены другие интересные закономерности. Так, стоимость чашки кофе не зависит от удаленности от центра. В средней удаленности от центра чаще других ставят низкие оценки. Средний чек в заведениях с рейтингом меньше четырех составляет 544 руб. Построена гистограмма для категорий заведений с плохим рейтингом. Кафе в датасете в принципе много, поэтому их чаще других оценивают плохо. Быстрое питание, работающее 24/7, относительно чаще других оценивают плохо. Вероятно, люди уставшие.

Детали кофеен

```
In [51]: # сначала отделим кофейни от кафе и прочих категорий
data = df_cafe[df_cafe['category'] == 'Кофейня']
```

Количество и расположение

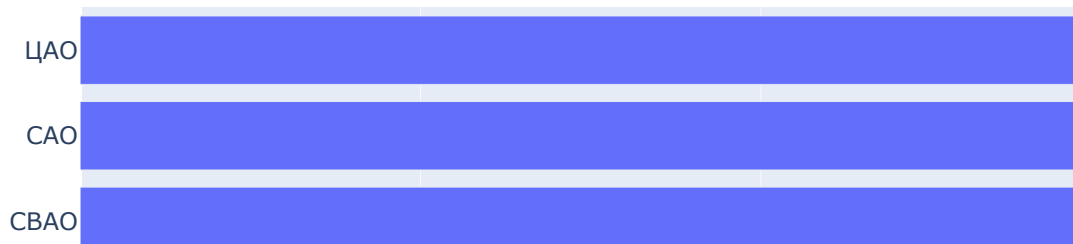
```
In [52]: print('Строк с кофейнями в датасете:', data.shape[0])
```

Строк с кофейнями в датасете: 1413

```
In [53]: # отфильтруем данные, сгруппируем по имени и посчитаем объявления
df_loc_count = data.groupby('district')[['district']].count()
# переименуем столбец
df_loc_count.columns = ['total_count']
# отсортируем и оставим пять лидеров
df_loc_count = df_loc_count.reset_index().sort_values(by='total_count', ascending=False)

# строим столбчатую диаграмму - использовать вместо того кода, что ниже
fig = px.bar(df_loc_count.sort_values(by='total_count', ascending=True), # загружаем
             x='total_count', # указываем столбец с данными для оси X
             y='district', # указываем столбец с данными для оси Y
             text='total_count' # добавляем аргумент, который отобразит текст с инф
                               # о количестве объявлений внутри столбца графика
             )
# оформляем график
fig.update_layout(title='Количество кофеен по районам',
                  xaxis_title='Количество заведений',
                  yaxis_title='Округ')
fig.show() # выводим график
```

Количество кофеен по районам



Как и ожидалось, намного больше всего кофеен в центре по отношению к любому другому округу. Чтобы оценить особенности расположения - посмотрим на карту

```
In [54]: # создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles="Cartodb Positron")
```

```
# 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
data.apply(create_clusters, axis=1)

# выводим карту
m
```

Out[54]: Make this Notebook Trusted to load map: File -> Trust Notebook

+

—

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

Почти все находятся рядом с большими улицами

Круглосуточные

```
In [55]: df_loc = data[data['is_24/7'] == 1].groupby('district')[['district']].count()
# переименуем столбец
df_loc.columns = ['total']
df_loc_count = df_loc.reset_index().merge(df_loc, on='district', how='left')
df_loc_count['total_coun'] = df_loc_count['total'] / df_loc_count['total_count']
for key, value in district.items():
    df_loc_count.loc[df_loc_count['district'] == value, 'district'] = key

# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles='Cartodb Positron')

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=df_loc_count,
    columns=['district', 'total_coun'],
    key_on='feature.name',
    fill_color='YlGn',
    fill_opacity=0.8,
```

```

legend_name='Отношение работающих круглосуточных к общему числу кофеен в этом с
).add_to(m)

# выводим карту
m

```

Out[55]: Make this Notebook Trusted to load map: File -> Trust Notebook

Отношение работающих круглосуточных к общему числу кофеен в этом окр

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

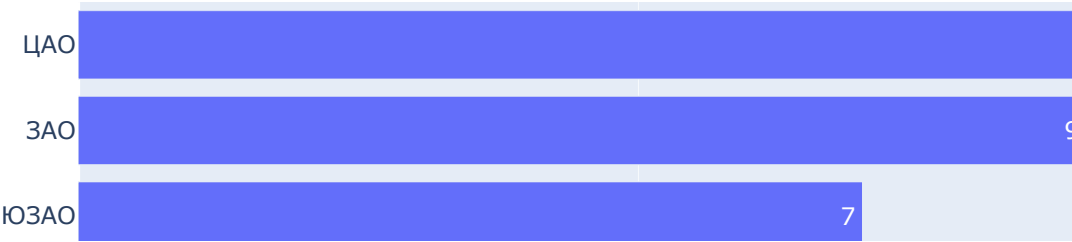
```

In [56]: # отфильтруем данные, сгруппируем по имени и посчитаем объявления
df_loc_count = df_loc
# отсортируем и оставим пять лидеров
df_loc_count = df_loc_count.reset_index().sort_values(by='total', ascending=False)

# строим столбчатую диаграмму - использовать вместо того кода, что ниже
fig = px.bar(df_loc_count.sort_values(by='total', ascending=True), # загружаем данн
             x='total', # указываем столбец с данными для оси X
             y='district', # указываем столбец с данными для оси Y
             text='total' # добавляем аргумент, который отобразит текст с информаци
                        # о количестве объявлений внутри столбца графика
            )
# оформляем график
fig.update_layout(title='Количество кофеен по районам',
                  xaxis_title='Количество заведений',
                  yaxis_title='Округ')
fig.show() # выводим график

```

Количество кофеен по районам



Круглосуточных кофеен не так много: 26 из 59 работают в ЦАО. Меньше их всего - на юге столицы. Посмотрим на распределение рейтингов кофеен по районам

Рейтинги по районам

```
In [57]: rating_df = data.groupby('district', as_index=False)['rating'].agg('mean')
rating_df.style.format({'rating': '{:2f}'}) # как вывести столбец с 2 знаками после
```

Out[57]:

	district	rating
0	БАО	4.282857
1	ЗАО	4.195333
2	САО	4.291710
3	СВАО	4.216981
4	СЗАО	4.325806
5	ЦАО	4.336449
6	ЮАО	4.232824
7	ЮВАО	4.225843
8	ЮЗАО	4.283333

Средний рейтинг кофеен варьируется от 4.19 до 4.34.

```
In [58]: rating_df = data.groupby('district', as_index=False)['rating'].agg('mean')
for key,value in district.items():
    rating_df.loc[rating_df['district'] == value, 'district'] = key
rating_df

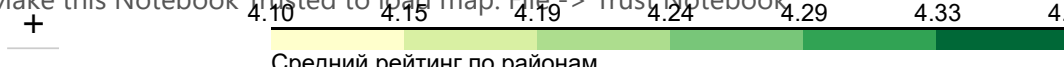
rating_df = df_cafe.groupby('district', as_index=False)['rating'].agg('mean')
for key,value in district.items():
    rating_df.loc[rating_df['district'] == value, 'district'] = key

# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles='Cartodb Positron')

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=rating_df,
    columns=['district', 'rating'],
    key_on='feature.name',
    fill_color='YlGn',
    fill_opacity=0.8,
    legend_name='Средний рейтинг по районам',
).add_to(m)

# выводим карту
m
```

Out[58]: Make this Notebook Trusted to load map: File → Trust Notebook



Средний рейтинг по районам

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

Как обычно - самые высокие рейтинги у ЦАО. Но что интересно - следующим идет САО и СЗАО. В них также мало круглосуточных. Стоит присмотреться внимательней.

```
In [59]: # нашла и составила датафрейм с начелением по округам на 2022г.
df_dist = {'district': ['BAO', 'ЗАО', 'САО', 'СВАО', 'СЗАО', 'ЦАО', 'ЮВАО', 'ЮЗАО'],
            'people': [1514420, 1383853, 1175229, 1427597, 1009217, 779086, 1432839, 1432839]}

df_dist = pd.DataFrame(df_dist)
df_dist
```

Out[59]:

	district	people
0	BAO	1514420
1	3AO	1383853
2	CAO	1175229
3	CBAO	1427597
4	C3AO	1009217
5	ЦАО	779086
6	ЮВАО	1432839
7	ЮЗАО	1442971
8	ЮАО	1773425

```
In [60]: # оценим хватает ли кофеен в каждом округе
df_dist = data.groupby('district', as_index=False)['name'].agg('count').merge(df_dist)
df_dist['rel'] = (df_dist['name'] / df_dist['people'])
df_dist
```

Out[60]:

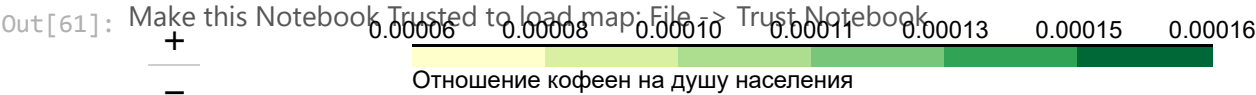
	district	name	people	rel
0	BAO	105	1514420	0.000069
1	3AO	150	1383853	0.000108
2	CAO	193	1175229	0.000164
3	CBAO	159	1427597	0.000111
4	C3AO	62	1009217	0.000061
5	ЦАО	428	779086	0.000549
6	ЮАО	131	1773425	0.000074
7	ЮВАО	89	1432839	0.000062
8	ЮЗАО	96	1442971	0.000067

```
In [61]: rating_df = df_dist[df_dist['district'] != 'ЦАО'].loc[:, ['district', 'rel']]
for key, value in district.items():
    rating_df.loc[rating_df['district'] == value, 'district'] = key

# создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles='Cartodb Positron')

# создаём хороплет с помощью конструктора Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=rating_df,
    columns=['district', 'rel'],
    key_on='feature.name',
    fill_color='YlGn',
    fill_opacity=0.8,
    legend_name='Отношение кофеен на душу населения',
).add_to(m)

# выводим карту
m
```

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

Очевидно, что центр будет наиболее богат на кофейни - сюда люди стремятся отдохнуть и в выходные, и в будни, поэтому при анализе его исключим. В CAO довольно много кофеен на душу населения этого округа по сравнению с остальными округами. А вот в СЗАО кофеен не так много, но рейтинги ставят весьма неплохие.

Чашечка кофе

In [62]:

rating_df = data.groupby('district', as_index=False)['middle_coffee_cup'].agg('median')
rating_df

Out[62]:

	district	middle_coffee_cup
8	ЮЗАО	198.0
5	ЦАО	190.0
1	ЗАО	189.0
4	СЗАО	165.0
3	СВАО	162.5
2	CAO	159.0
6	ЮАО	150.0
7	ЮВАО	147.5
0	ВАО	135.0

Чашка кофе стоит по-разному в зависимости от округа. Как и ожидалось, один из самых дорогих кофе в центре. Однако, больше всего указывает на юго-запад. Это странно. Если же планировать открывать кофейню в СЗАО - капучино лучше бы стоить не менее 165р.

Непосредственные конкуренты в СЗАО

Поскольку «Shut Up and Take My Money» собирается открыть кофейню как в сериале "Друзья", то стоит посмотреть на конкурентов. Должны быть места для посадки, доступная ценаю Средний чек в этом районе 822.222930, а чашка кофе 165руб. Оставим неизвестные данные

In [63]:

```
SVAO = data[(data['district'] == 'СЗАО') & (~data['price'].isna()) & (data['seats']
& ((data['middle_avg_bill']<=822.222930) | (data['middle_avg_bill'].isna()
& ((data['middle_coffee_cup']<=165) | (data['middle_coffee_cup'].isna()
print(SVAO.shape)
SVAO
```

(5, 17)

Out[63]:

	name	category	address	district	hours	lat	lng	rating
403	Дон Тантуни	Кофейня	Москва, Туристская улица, 6, стр. 1	СЗАО	ежедневно, 10:00–23:00	55.848467	37.423566	4.7
1181	The Buffet	Кофейня	Москва, улица Кулакова, 20, корп. 1	СЗАО	пн-пт 08:00– 20:00	55.803189	37.390862	4.4
1229	Столовая 33	Кофейня	Москва, Таманская улица, 33	СЗАО	ежедневно, 08:30–21:00	55.780793	37.436796	4.5
1291	Шоколадница	Кофейня	Москва, улица Народного Ополчения, 49, корп. 1	СЗАО	ежедневно, круглосуточно	55.794815	37.494834	4.2
3238	Роскофейня.РФ	Кофейня	Москва, улица Мнёвники, 13	СЗАО	пн-пт 07:00– 22:00; сб,вс 08:00–21:00	55.773344	37.485226	4.5

И посленднее: кафе Central Perk небольшое и уютное, поэтому конкурентами не будут те, что вмещают много людей. Оставим условно тех, что имеют количество посадочным мест до 50

In [64]:

```
SVAO = SVAO[SVAO['seats'] <= 50]
SVAO
```

Out[64]:

	name	category	address	district	hours	lat	lng	rating
1229	Столовая 33	Кофейня	Москва, Таманская улица, 33	СЗАО	ежедневно, 08:30–21:00	55.780793	37.436796	4.5
3238	Роскофейня.РФ	Кофейня	Москва, улица Мнёвники, 13	СЗАО	пн-пт 07:00– 22:00; сб,вс 08:00–21:00	55.773344	37.485226	4.5

```
In [65]: # создаём карту Москвы
m = Map(location=[moscow_lat, moscow_lng], zoom_start=10, tiles='Cartodb Positron')

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

# применяем функцию для создания маркера ко всем строкам датафрейма
SVAO.apply(create_marker, axis=1)

# выводим карту
m
```

```
Out[65]: Make this Notebook Trusted to load map: File -> Trust Notebook
+
-

```

Leaflet (<https://leafletjs.com>) | © OpenStreetMap (<http://www.openstreetmap.org/copyright>) contributors © CartoDB (<http://cartodb.com/attributions>), CartoDB attributions (<http://cartodb.com/attributions>)

Удивительно, но осталось критично мало конкурентов

Краткий вывод

Всего строк с кофейнями в датасете: 1413. В ЦАО находится 428 кофеен, в САО - 62, из которых 25 и 2 круглосуточные соответственно. Средний рейтинг кофеен по округам варьируется от 4.19 до 4.34. Самые высокие рейтинги у ЦАО. Но что интересно - следующим идет САО и СЗАО. В них также мало круглосуточных. Был создан датасет с количеством человек, проживающих в округе, а затем посчитано отношение кофеен на человека. В САО довольно много заведений на душу населения этого округа по сравнению с остальными. А вот в СЗАО кофеен не так много, но рейтинги ставят весьма неплохие - поэтому рекомендуется открывать дело в этом округе. The Buffet или Шоколадница являются крупными игроками, к тому же кофейня Central Perk в сериале ориентирована на другую аудиторию - поэтому их в расчет как конкурентов брать не будем. Дон Тантуни вмещает большое количество человек, тогда как Central Perk предполагает небольшое уютное пространство - поэтому тоже не рассматривается. Столовая 33, несмотря на звание кофейни, является все же столовой(к тому же к

настоящему моменту закрылась). Таким образом, непосредственным конкурентом остается только Роскофейня.РФ.

Дополнительно рекомендуется провести детальный анализ возможных доходов и расходов в этом округе.

Вывод

В предоставленном датасете 8406 строк, есть пропуски в 6 столбцах. Возможно, это результат слияния нескольких таблиц. `middle_avg_bill` и `middle_coffee_cup` основаны на `avg_bill`, а потому имеют 63% и 94% пропусков. У части заведений отсутствует рейтинг в Яндекс-картах. Возможно, это какие-то новые заведения, не успевшие набрать более 5 оценок. Хранить данные в `object` и `float64` не целесообразно с точки зрения памяти. Помимо имеющихся данных был создан столбец с названиями улиц, на которых расположены заведения, столбец с обозначением работает ли заведение ежедневно и круглосуточно. Сокращены названия округов. В данной работе пропуски не мешают анализу заведений общепита и, соответственно, не обрабатываются.

Проанализированы категории предоставленных данных: больше всего кофеен(2378шт.), следом с небольшим отставанием рестораны(2043шт.). Меньше всего булочных и столовых. Также выяснено, что больше всего денег за раз оставляют в барах и ресторанах - это логично. Была построена гистограмма средних оценок по категориям, по ней замечено, что средняя оценка колеблется от 4,13 до 4,4. У булочных, столовых и быстрого питания больше других среднее квадратичное отклонение, а у ресторанов и кофеен - самое маленькое.

Было проанализировано количество посадочных мест в зависимости от категории, их медианы варьируются от 50 до 86. Дополнительно был построен ящик с усами: количество мест по категориям. Есть выбросы с количеством стульев более 1200. Больше всего мест в ресторанах и барах, зато в булочной/пиццерии/кафе очень часто дают на вынос - первые квартили у них самые низкие.

Проанализировано количество сетевых и несетевых заведений. Последних явно больше - 61.87% датасета. Также построен график соотношения сетевых и несетевых заведений по категориям. Категория "кафе" представлена в датасете большим количеством, в нем также больше всего несетевых заведений. А вот в кофейнях соотношение 50/50. Во всех категориях несетевых больше, кроме булочных: 60/40.

Далее выявили топ-15 наиболее многочисленных сетевых точек в Москве, выяснили, что больше всего заведений открыла Шоколадница(кофейня) - целых 120. За ней пара пиццерий: Домино'с и Додо пицца. Также посмотрели эти сети по округам. В ЦАО самым многочисленным является Шоколадница, следом - Prime, Кулинарная лавка братьев Караваевых. Наиболее малочисленная - Буханка. Зато ее заведений относительно много в САО, ВАО и ЮАО.

Если же смотреть общее количество заведений каждой категории по округам, большинство из них находится в центре: это рестораны, кафе и кофейни.

Примечательно, что в центре особенно популярны бары, пабы - в отличие от других

округов. В целом динамика повторяется в каждом округе: очень много кафе, потом ресторанов, кофеен, пиццерий/быстрого питания, баров, пабов, столовых/булочных. Разве что в CAO рестораны чуть популярнее кофеен.

Также построен Хороплет со средним рейтингом: в центре все же более щедры на хорошие оценки. Самый низкооцениваемый - ЮВАО: 4.10.

С помощью кластера были отмечены на карте Москвы все заведения, присутствующие в датасете. Была построена гистограмма для топ-15 улиц по количеству заведений: лидирует Проспект Мира. Это не удивительно: улица длинная и недалеко от центра. На каждой улице по количеству лидируют либо кафе, либо рестораны.

Построен хороплет с медианным чеком по районам: у центра и ЗАО средний чек выше других. Самый дорогой район - центральный. Построен график рассеяния: чем дальше от центра, тем в среднем, дешевле.

Также были выявлены другие интересные закономерности. Так, стоимость чашки кофе не зависит от удаленности от центра. В средней удаленности от центра чаще других ставят низкие оценки. Средний чек в заведениях с рейтингом меньше четырех составляет 544 руб. Построена гистограмма для категорий заведений с плохим рейтингом. Кафе в датасете в принципе много, поэтому их чаще других оценивают плохо. Быстрое питание, работающее 24/7, относительно чаще других оценивают плохо. Вероятно, люди уставшие.

Проведено исследование кофеен для оценки расположения для открытия места Central Perk из "Друзья" Всего строк с кофейнями в датасете: 1413. В ЦАО находится 428 кофеен, в CAO - 62, из которых 25 и 2 круглосуточные соответственно. Средний рейтинг кофеен по округам варьируется от 4.19 до 4.34. Самые высокие рейтинги у ЦАО. Но что интересно - следующим идет CAO и СЗАО. В них также мало круглосуточных. Был создан датасет с количеством человек, проживающих в округе, а затем посчитано отношение. В CAO довольно много заведений на душу населения этого округа по сравнению с остальными. А вот в СЗАО кофеен не так много, но рейтинги ставят весьма неплохие - поэтому рекомендуется открывать дело в этом округе. По результатам анализа данного округа был выявлен 1 непосредственный конкурент: Роскофейня.РФ.

Дополнительно рекомендуется провести детальный анализ возможных доходов и расходов в этом округе.

In []: