Рынок заведений общественного питания Москвы

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.

Цель исследования: найти интересные особенности и презентовать полученные результаты, которые в будущем помогут в выборе подходящего инвесторам места

Ход исследования:

- Изучение общей информации датасета
- Предобработка данных
- Анализ данных
- Детализация данных открытие кофейни

Описание данных

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.

Откроем файл и изучим общую информацию

In [1]:
```
!pip install plotly==5.13.0
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import graph_objects as go
import plotly.express as px
import warnings
import folium
import json
from folium import Map, Choropleth, Marker
from folium.plugins import MarkerCluster
```

```
Requirement already satisfied: plotly==5.13.0 in c:\programdata\anaconda3\lib\site-packages (5.13.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda3\lib\site-packages (from plotly==5.13.0) (8.0.1)
```

In [2]:
```
try:
    moscow_exp = pd.read_csv(r'C:\Users\niksmns\Desktop\moscow_places\moscow_places.csv')
    state_map = r'https://code.s3.yandex.net/data-analyst/admin_level_geomap.geojson'
except:
    moscow_exp = pd.read_csv('/datasets/moscow_places.csv')
    state_map = '/datasets/admin_level_geomap.geojson'
```

In [3]:
```
def general_info(data):
    display(data.head(10))
    display(data.info())
    display(pd.DataFrame(round(data.isna().mean()*100,)).style.background_gradient('coolwarm'))
    display('Количество дубликатов:', data.duplicated().sum())
    display(data.columns)
    display(data.describe())
```

In [4]:
```
general_info(moscow_exp)
```

	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
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, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние
5	Sergio Pizza	пиццерия	Москва, Ижорская улица, вл8Б	Северный административный округ	ежедневно, 10:00–23:00	55.888010	37.509573	4.6	средние
6	Огни города	бар,паб	Москва, Клязьминская улица, 9, стр. 3	Северный административный округ	пн 15:00–04:00; вт-вс 15:00–05:00	55.890752	37.524653	4.4	средние
7	Mr. Уголёк	быстрое питание	Москва, Клязьминская улица, 9, стр. 3	Северный административный округ	пн-чт 10:00–22:00; пт,сб 10:00–23:00; вс 10:00...	55.890636	37.524303	4.7	средние
8	Donna Maria	ресторан	Москва, Дмитровское шоссе, 107, корп. 4	Северный административный округ	ежедневно, 10:00–22:00	55.880045	37.539006	4.8	средние
9	Готика	кафе	Москва, Ангарская улица, 39	Северный административный округ	ежедневно, 12:00–00:00	55.879038	37.524487	4.3	средние

```
<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
None
```

	0
name	0.000000
category	0.000000
address	0.000000
district	0.000000
hours	6.000000
lat	0.000000
lng	0.000000
rating	0.000000
price	61.000000
avg_bill	55.000000
middle_avg_bill	63.000000
middle_coffee_cup	94.000000
chain	0.000000
seats	43.000000

```
'Количество дубликатов:'
0
Index(['name', 'category', 'address', 'district', 'hours', 'lat', 'lng',
       'rating', 'price', 'avg_bill', 'middle_avg_bill', 'middle_coffee_cup',
       'chain', 'seats'],
      dtype='object')
```

	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

Изучили информацию о датасете, было найдено много пропусков. Также в ходе ручной проверки на достоверность данных было выявлено много некоректоной информации. Скорее всего этому послужила ошибка ввода данных и неверно настроенные алгоритмы программы

Предобработка данных

In [5]:
```
# создадим колонку с улицами разделив строки методом split
moscow_exp['street'] = moscow_exp['address'].str.split(',').str[1]
moscow_exp['street']
```

Out[5]:
```
0                  улица Дыбенко
1                  улица Дыбенко
2              Клязьминская улица
3         улица Маршала Федоренко
4             Правобережная улица
                   ...
8401            Профсоюзная улица
8402        Пролетарский проспект
8403             Люблинская улица
8404             Люблинская улица
8405          Россошанский проезд
Name: street, Length: 8406, dtype: object
```

In [6]:
```
# посмотрим какие есть значения
moscow_exp['hours'].value_counts()
```

Out[6]:
```
ежедневно, 10:00–22:00                                                759
ежедневно, круглосуточно                                              730
ежедневно, 11:00–23:00                                                396
ежедневно, 10:00–23:00                                                310
ежедневно, 12:00–00:00                                                254
                                                                     ...
пн-пт 17:00–03:00; сб,вс 17:00–05:00                                    1
пн,вт 09:00–21:00; ср-пт 09:00–22:00; сб 10:00–22:00; вс 10:00–21:00    1
пн-пт 12:00–01:00                                                       1
пн-пт 10:30–21:30; сб,вс 10:30–22:30                                    1
пн-сб 10:30–21:30                                                       1
Name: hours, Length: 1307, dtype: int64
```

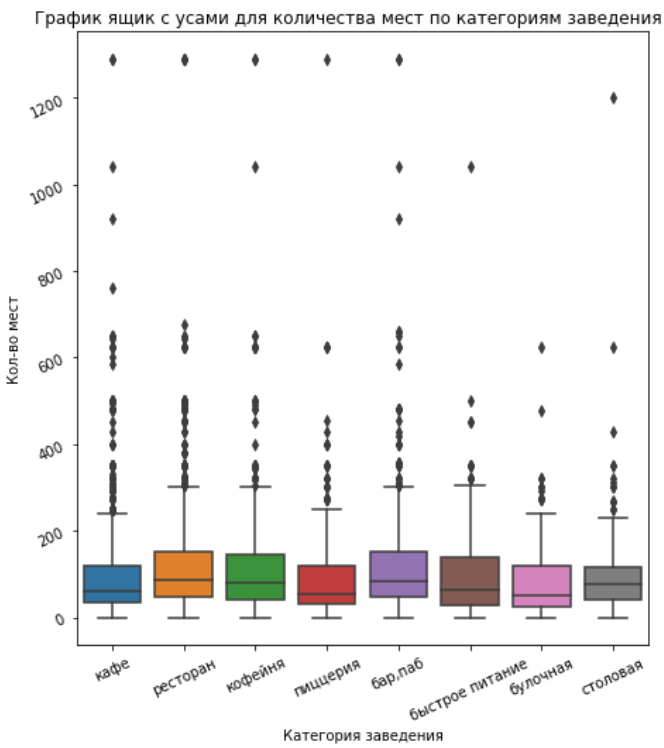
In [7]:
```
# функция для категорий графика работы 24/7
def hours_category(row):
    try:
        if 'ежедневно' and 'круглосуточно' in row:
            return True
        else:
            return False
    except:
        return 'неизвестно'
```

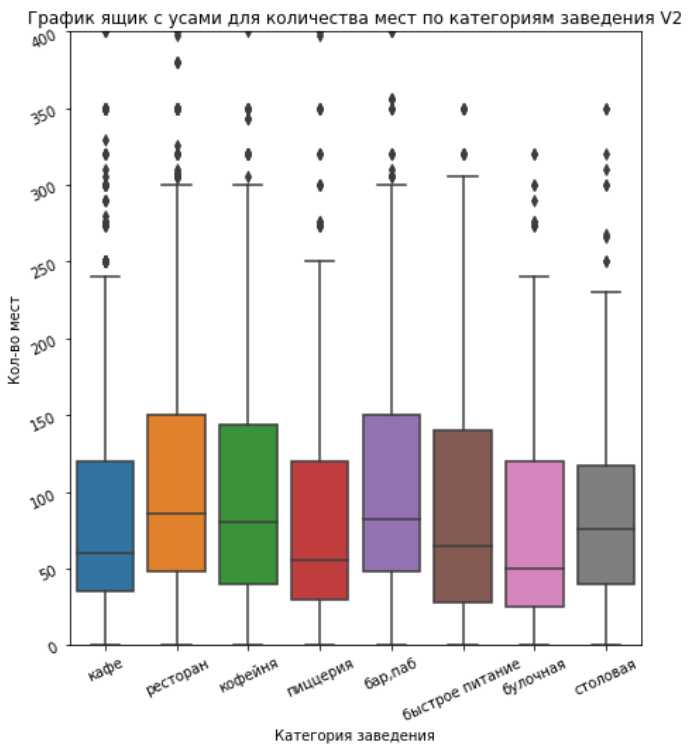
In [8]:
```
# применяем нашу функцию
moscow_exp['is_24/7'] = moscow_exp['hours'].apply(hours_category).astype('bool')
# проверяем
#moscow_exp[['hours', 'is_24/7']]
```

In [9]:
```
# проверяем корректность функции
#moscow_exp[['hours', 'is_24/7']].loc[moscow_exp['is_24/7'] == 'неизвестно'].head(5)
```

In [10]:
```
# ставим заглушки в пропусках
moscow_exp['hours'] = moscow_exp['hours'].fillna('неизвестно')
moscow_exp['price'] = moscow_exp['price'].fillna('неизвестно')
moscow_exp['avg_bill'] = moscow_exp['avg_bill'].fillna('неизвестно')
moscow_exp['seats'] = moscow_exp['seats'].astype(int, errors='ignore')
moscow_exp['middle_avg_bill'] = moscow_exp['middle_avg_bill'].astype(int, errors='ignore')
moscow_exp['middle_coffee_cup'] = moscow_exp['middle_coffee_cup'].astype(int, errors='ignore')
```

In [11]:
```
fig, axs = plt.subplots(1, 2, figsize=(16, 8))

sns.boxplot(data=moscow_exp, x='category', y='seats', ax=axs[0])
axs[0].tick_params(axis='x', rotation=25)
axs[0].tick_params(axis='y', rotation=25)
axs[0].set_title('График ящик с усами для количества мест по категориям заведения V1')
axs[0].set_xlabel('Категория заведения')
axs[0].set_ylabel('Кол-во мест')
sns.boxplot(data=moscow_exp, x='category', y='seats', ax=axs[1]).set(ylim=(0, 400))
axs[1].tick_params(axis='x', rotation=25)
axs[1].tick_params(axis='y', rotation=25)
axs[1].set_title('График ящик с усами для количества мест по категориям заведения V2')
axs[1].set_xlabel('Категория заведения')
axs[1].set_ylabel('Кол-во мест')
plt.show()
```





In [12]:
```
# создаем столбец для поиска неявных дубликатов
moscow_exp['address'] = moscow_exp['address'].str.lower()
moscow_exp['name'] = moscow_exp['name'].str.lower()
moscow_exp['duplicate'] = moscow_exp['address'] + moscow_exp['name']
```

In [13]:
```
# смотрим неявные дубликаты
moscow_exp[moscow_exp['duplicate'].duplicated() == True]['duplicate']
```

Out[13]:
```
215                      москва, парк ангарские прудыкафе
1511     москва, волоколамское шоссе, 11, стр. 2more poke
2420    москва, проспект мира, 118раковарня клешни и х...
3109            москва, ярцевская улица, 19хлеб да выпечка
Name: duplicate, dtype: object
```

In [14]:
```
moscow_exp['duplicate'] = moscow_exp[moscow_exp['duplicate'].duplicated() == False]['duplicate']
```

In [15]:
```
moscow_exp = moscow_exp.drop(labels=[215, 1511, 2420, 3109], axis=0)
moscow_exp = moscow_exp.drop(columns='duplicate')
```

In [16]:
```
# выявляем неявные дубликаты
moscow_exp[moscow_exp['name'].str.contains('домино')]['name'].unique()
# исправляем
moscow_exp['name'] = moscow_exp['name'].replace(['доминос пицца', "домино'с" ], "домино'с пицца"
# проверяем
moscow_exp[moscow_exp['name'].str.contains('домино')]['name'].unique()
```

Out[16]:
```
array(["домино'с пицца"], dtype=object)
```

In [17]:
```
# выявляем неявные дубликаты
moscow_exp[moscow_exp['name'].str.contains('яндекс')]['name'].unique()
# исправляем
moscow_exp['name'] = moscow_exp['name'].replace('яндекс.лавка', 'яндекс лавка')
# проверяем
moscow_exp[moscow_exp['name'].str.contains('яндекс')]['name'].unique()
```

Out[17]:
```
array(['яндекс лавка', 'яндекс еда'], dtype=object)
```

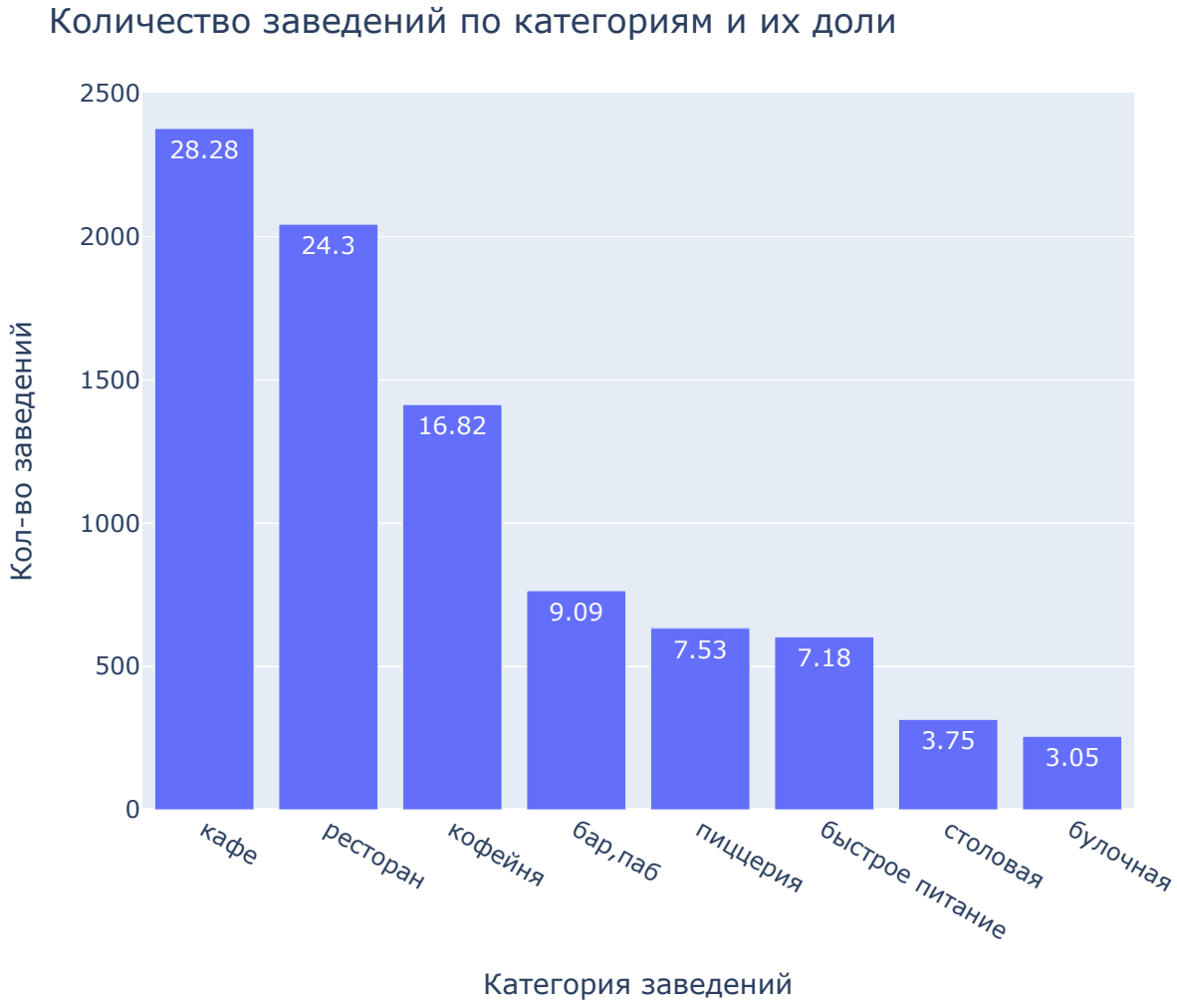
Поменяли тип данных у колонок где смогли. Заполнили пропуски заглушками. Также добавили столбцы улиц заведений и столбец работы 24/7 и удалили неявные дубликаты

Анализ данных

Количество заведений по категориям

In [18]:
```
mos_category_cnt = (
    moscow_exp.pivot_table(index='category', values='name', aggfunc='count')
    .reset_index())
mos_category_cnt.sort_values(by='name', ascending=False)
mos_category_cnt['percent'] = round(mos_category_cnt['name'] / mos_category_cnt['name'].sum() * 
mos_category_cnt = mos_category_cnt.sort_values(by='percent', ascending=False)
```

In [19]:
```
fig = px.bar(mos_category_cnt, x=mos_category_cnt['category'], y=mos_category_cnt['name'] ,text=

fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Количество заведений по категориям и их доли',
                  xaxis_title='Категория заведений',
                  yaxis_title='Кол-во заведений')
```



Видим, что больше всего заведений в Москве - это кафе, рестораны и кофейни. Кафе и

рестораны занимают ~1/4 заведений Москвы, также видим, что у кофеен большая доля заведений, вероятно это связано с "четвертой волной", когда стало переосмысление кофе как продукта, и оно превратилось в науку, спорт и искусство. Люди стали чаще открывать кофейни и кофейные точки, где-то осталась философия этой четвертой волны, а где-то осталась только коммерция. Также открытие простенькой кофейни не столько финансово затратно, как открытие других категорий заведений.

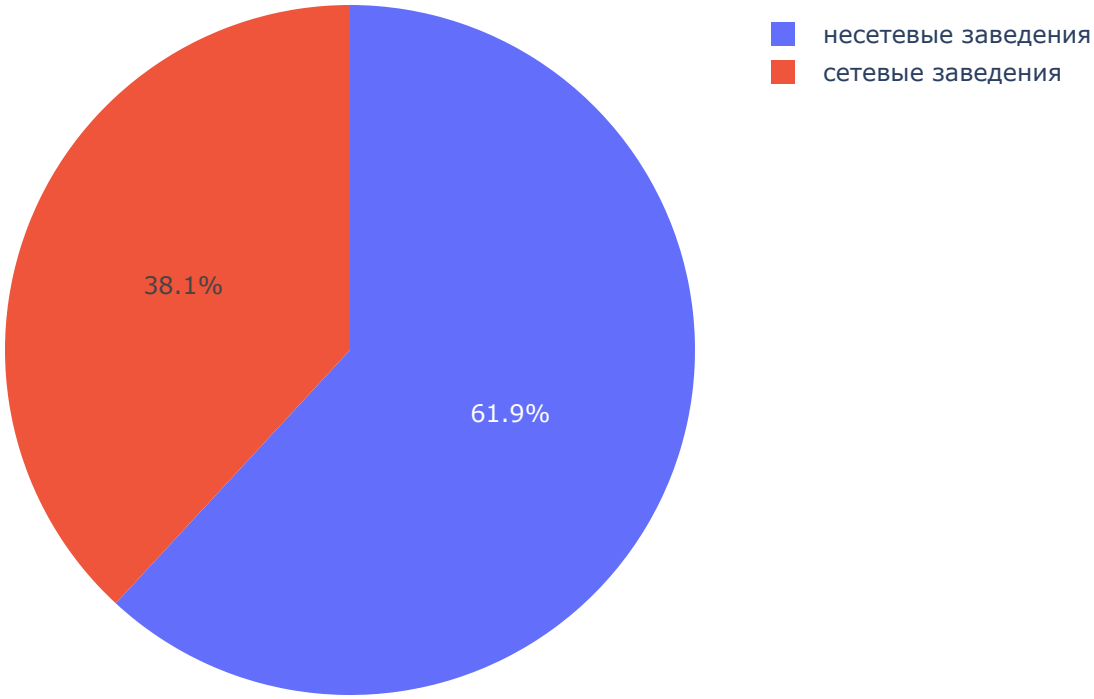
Посадочные места заведений по категориям

In [20]:
```
mos_seats = moscow_exp.pivot_table(index='category', values='seats', aggfunc='median').reset_inde
fig = px.bar(mos_seats, x=mos_seats['category'], y=mos_seats['seats'], text_auto='.0f')

fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Среднее количество мест по категориям',
                  xaxis_title='Категория заведений',
                  yaxis_title='Среднее кол-во мест')
```

Поскольку в данных явно какие-то неккоретные аномальные сведения о количестве посадочных мест, выявлили среднее количество посадочных мест через медиану. Выяснилось, что больше всего мест имеют рестораны, бары/пабы и кофейни.

Сетевые и несетевые заведения

In [21]:
```
mos_chain = moscow_exp.pivot_table(index='chain', values='name', aggfunc='count').reset_index()
fig = go.Figure(data=[go.Pie(labels=mos_chain['chain'].map({1: 'сетевые заведения', 0: 'несетевы
                             values=mos_chain['name'])])
fig.update_layout(title_text='Доли сетевых/несетевых заведений')
fig.show()
```



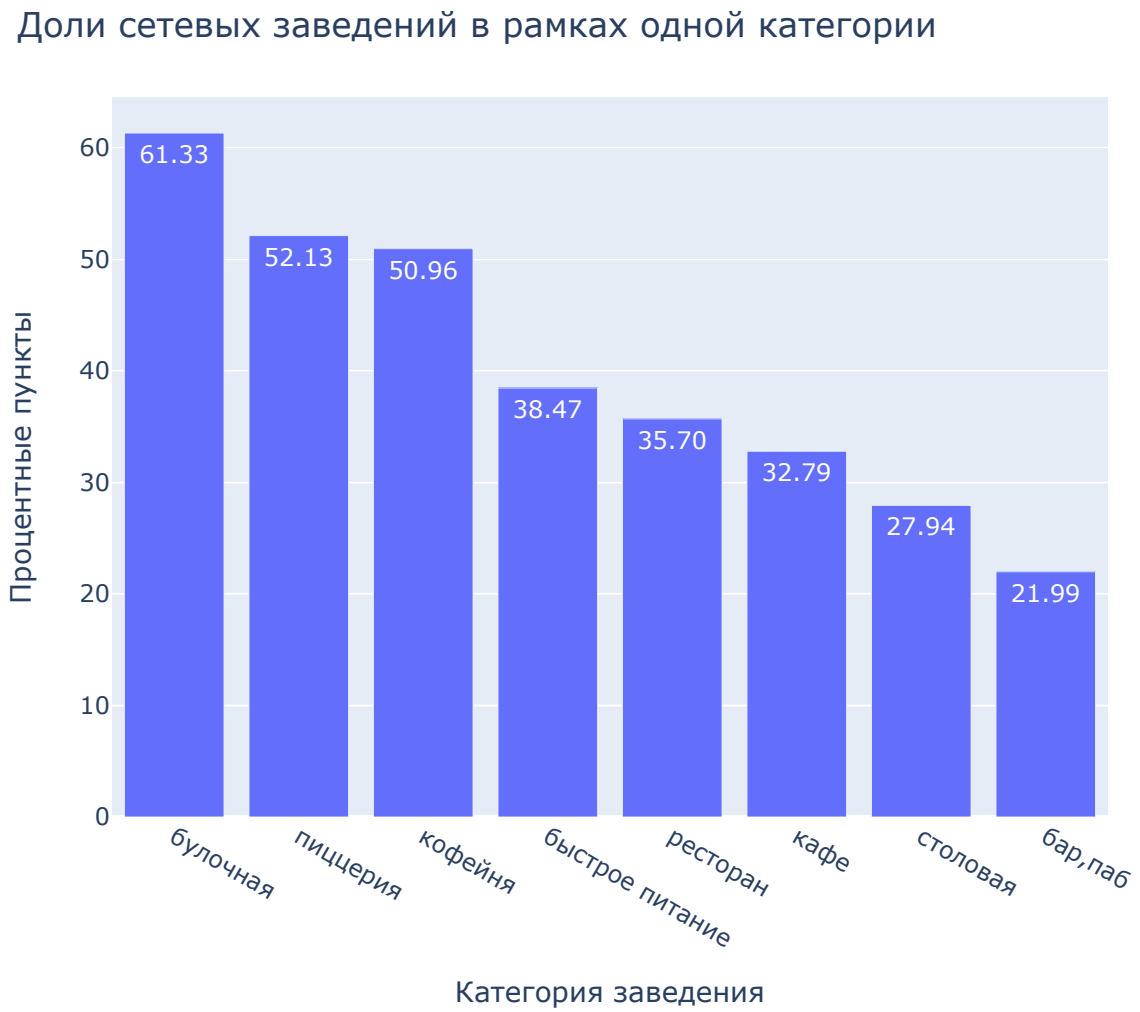
Несетевых заведений в Москве больше, процентная доля их составляет - 61.9%, сетевых же заведений поменьше, их доля составляет - 38.1%

In [22]:
```
mos_chain_categ = moscow_exp.pivot_table(index='category', values='name', columns='chain', aggfu
mos_chain_categ['total_cnt'] = mos_chain_categ[0] + mos_chain_categ[1]
mos_chain_categ['percent_chain'] = round(mos_chain_categ[1] / mos_chain_categ['total_cnt'] * 100
mos_chain_categ = mos_chain_categ.sort_values(by='percent_chain', ascending=False)
mos_chain_categ
```

Out[22]:

chain	category	0	1	total_cnt	percent_chain
1	булочная	99	157	256	61.33
5	пиццерия	303	330	633	52.13
4	кофейня	693	720	1413	50.96
2	быстрое питание	371	232	603	38.47
6	ресторан	1313	729	2042	35.70
3	кафе	1597	779	2376	32.79
7	столовая	227	88	315	27.94
0	бар,паб	596	168	764	21.99

In [23]:
```
fig = px.bar(mos_chain_categ, x=mos_chain_categ['category'], y=mos_chain_categ['percent_chain'],

fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Доли сетевых заведений в рамках одной категории',
                  xaxis_title='Категория заведения',
                  yaxis_title='Процентные пункты')
```



Булочные, пиццерии и кофейни занимают большую часть сетевых заведений по отношению не к сетевым

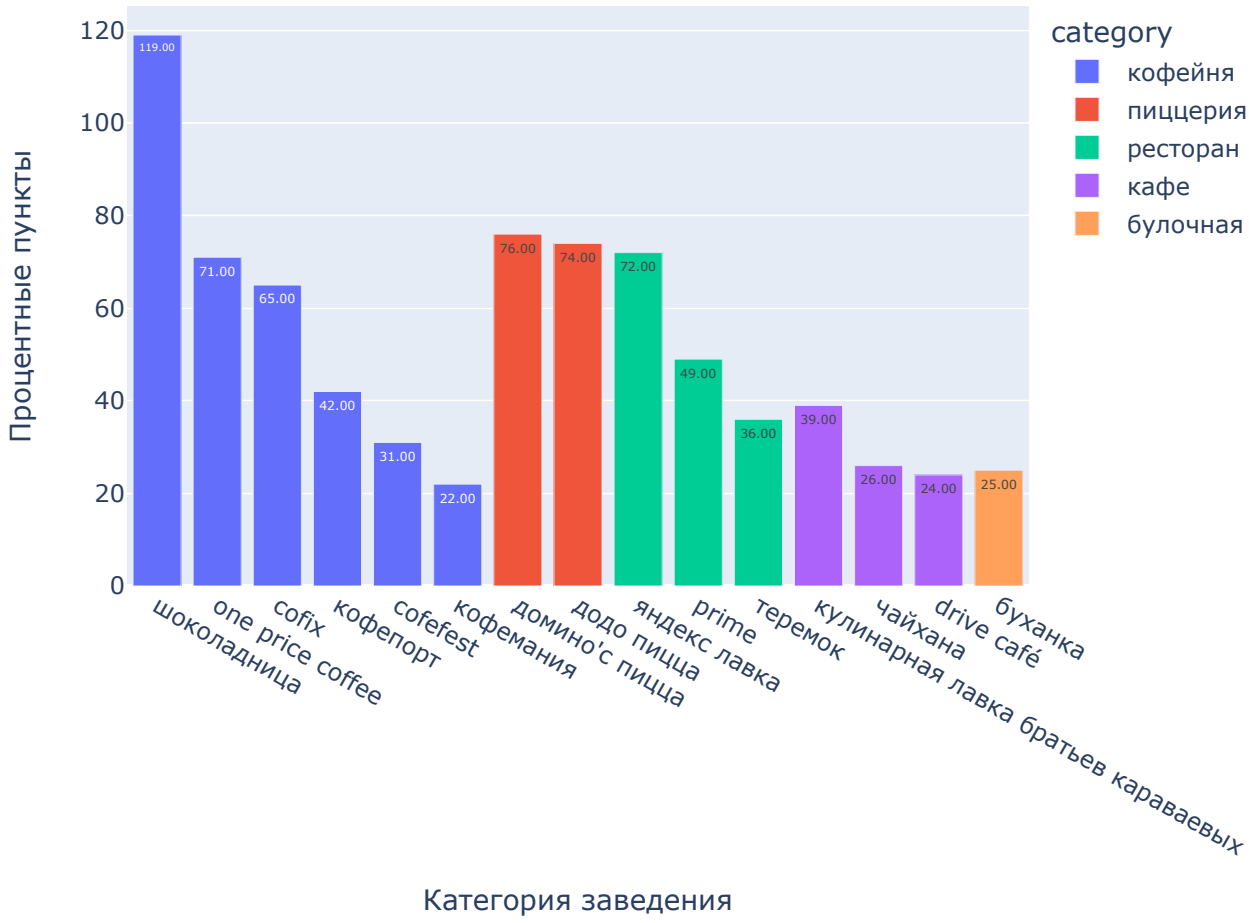
Топ-15 сетевых заведений

In [24]:
```
mos_chain_top_15 = (
    moscow_exp.query('chain == 1')
    .groupby(['category','name']).agg({'name':'count'}))
mos_chain_top_15.columns = ['count']
mos_chain_top_15 = mos_chain_top_15.sort_values(by='count', ascending=False).head(15).reset_inde
mos_chain_top_15
```

Out[24]:

	category	name	count
0	кофейня	шоколадница	119
1	пиццерия	домино'с пицца	76
2	пиццерия	додо пицца	74
3	ресторан	яндекс лавка	72
4	кофейня	one price coffee	71
5	кофейня	cofix	65
6	ресторан	prime	49
7	кофейня	кофепорт	42
8	кафе	кулинарная лавка братьев караваевых	39
9	ресторан	теремок	36
10	кофейня	cofefest	31
11	кафе	чайхана	26
12	булочная	буханка	25
13	кафе	drive café	24
14	кофейня	кофемания	22

In [25]:
```
fig = px.bar(mos_chain_top_15, x='name', y='count', text_auto='.2f', color='category')

fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Доли сетевых заведений в рамках одной категории',
                  xaxis_title='Категория заведения',
                  yaxis_title='Процентные пункты')
```



Как видим, здесь очень много знакомых нам сетевых заведений, среди них очень много кофеен. Также большую часть занимает сетевые пиццерии и рестораны

Административные районы

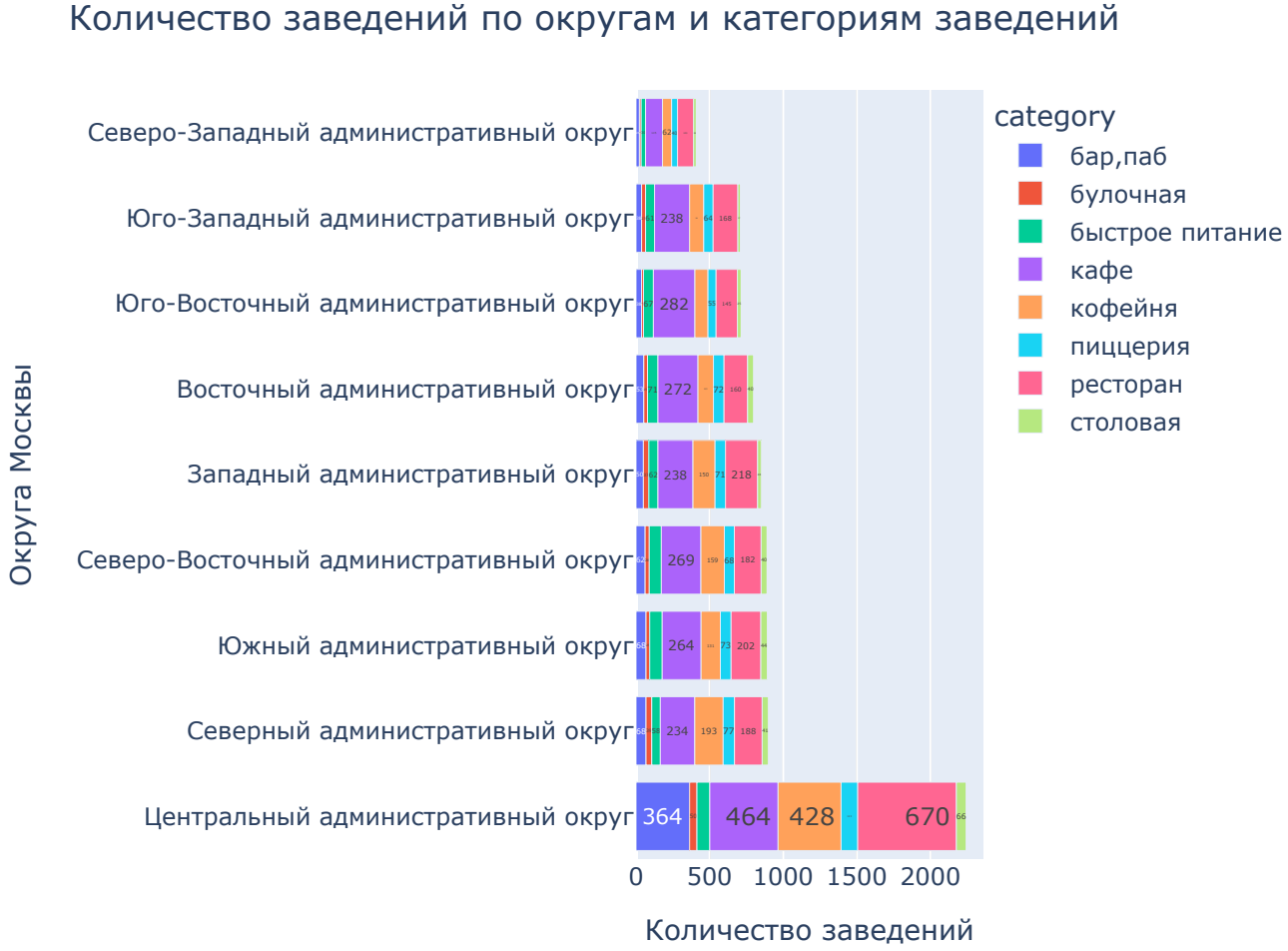
In [26]:
```
mos_district_categ = moscow_exp.pivot_table(index='district', columns='category', values='name',
mos_district_categ['total'] = mos_district_categ[['бар,паб','булочная','быстрое питание','кафе',
mos_district_categ = mos_district_categ.sort_values(by='total', ascending=False)
mos_district_categ
```

Out[26]:

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая	total
Центральный административный округ	364	50	87	464	428	113	670	66	2242
Северный административный округ	68	39	58	234	193	77	188	41	898
Южный административный округ	68	25	85	264	131	73	202	44	892
Северо-Восточный административный округ	62	28	82	269	159	68	182	40	890
Западный административный округ	50	37	62	238	150	71	218	24	850
Восточный административный округ	53	25	71	272	105	72	160	40	798
Юго-Восточный административный округ	38	13	67	282	89	55	145	25	714
Юго-Западный административный округ	38	27	61	238	96	64	168	17	709
Северо-Западный административный округ	23	12	30	115	62	40	109	18	409

In [27]:
```
mos_district_categ = mos_district_categ.drop(columns='total', axis=1)
```

In [28]:
```
fig = px.bar(mos_district_categ, text_auto='.0f', orientation='h')
fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Количество заведений по округам и категориям заведений',
                  xaxis_title='Количество заведений',
                  yaxis_title='Округа Москвы')
```



Можем заметить, что в центре больше всего количества заведений каждого типа, в других же округах плюс-минус распределение одинаковое за исключением Северо-Западного административного округа, там меньше всего заведений почти всех категорий.

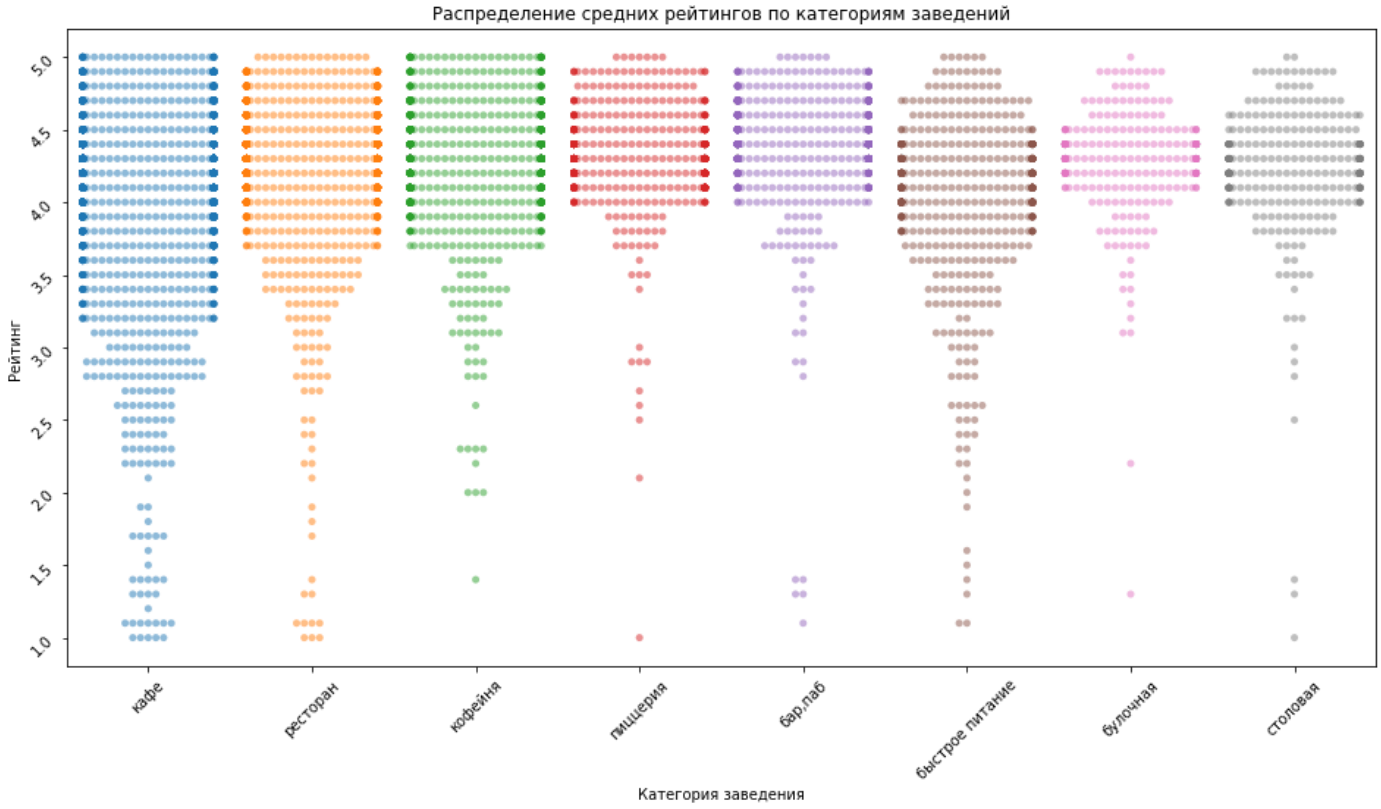
Распределение средних рейтингов по категориям заведений

In [30]:
```
moscow_exp.pivot_table(index='category', values='rating', aggfunc=['count', 'mean'])
```

Out[30]:

category	count rating	mean rating
бар,паб	764	4.387696
булочная	256	4.268359
быстрое питание	603	4.050249
кафе	2376	4.124285
кофейня	1413	4.277282
пиццерия	633	4.301264
ресторан	2042	4.290402
столовая	315	4.211429

In [31]:
```
with warnings.catch_warnings():# воспользуемся библиотекой warnings, чтобы убрать сообщения об ou
    warnings.simplefilter("ignore", category=UserWarning)
    plt.figure(figsize=(16,8))
    ax = sns.swarmplot(y='rating', x='category', alpha=0.5, data=moscow_exp) # воспользуемся кате
    ax.set_title('Распределение средних рейтингов по категориям заведений')
    ax.set_xlabel('Категория заведения')
    ax.set_ylabel('Рейтинг')
    ax.tick_params(axis='x', rotation=45)
    ax.tick_params(axis='y', rotation=45)
```

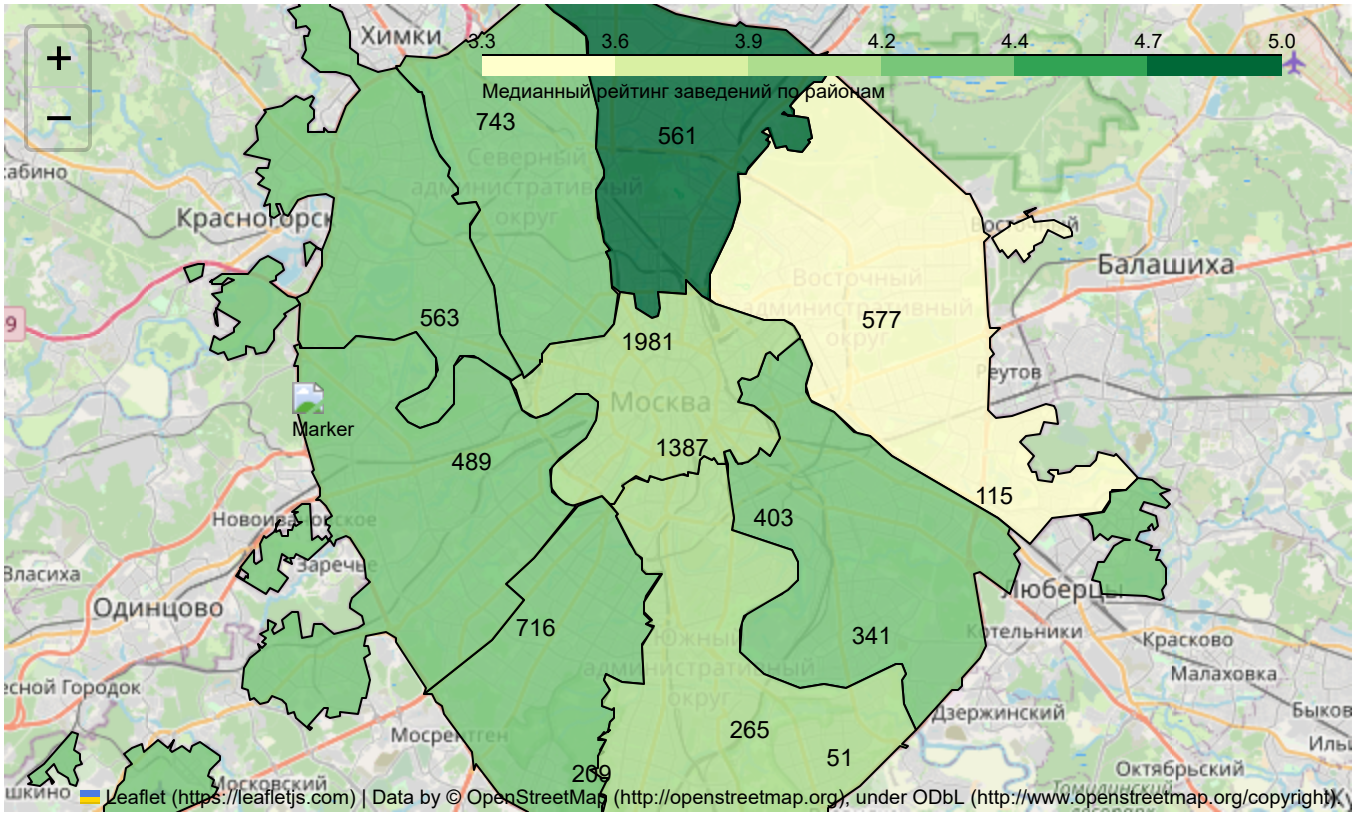


Видим, что основное количество оценок приходится на кафе, рестораны и кофейни. Разнообразие же в рейтинге обладают кафе, рестораны и заведения быстрого питания. Усредненные же рейтиги различаются не очень сильно по всем категориям, но булочные, столовые и заведения быстрого питания меньше всего получают самые высокие оценки.

Фоновая картограмма

In [32]:
```
# координаты центра Москвы
moscow_lat, moscow_lng = 55.751244, 37.618423
# создаем карту Москвы
mos_map = Map(location=[moscow_lat, moscow_lng], zoom_start=10)

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

marker_cluster = MarkerCluster().add_to(mos_map)

# 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
moscow_exp.apply(create_clusters, axis=1)

# выводим нашу карту
mos_map
```

Out[32]:



Можем выделить, что почти все регионы имеет среднюю оценку в 4+. Выделяются у нас тут два региона: Северо-Восточный административный округ, который имеет больше всего положительных оценок заведений, и Восточный административный округ, который имеет меньше положительных оценок и средний рейтинг можно назвать удовлетворительным.

Топ-15 улиц по количеству заведений

In [33]:
```
mos_street_top15 = (
    moscow_exp.pivot_table(index='street', columns='category', values='name', aggfunc='count')
)
mos_street_top15['total'] = mos_street_top15[['бар,паб','булочная','быстрое питание','кафе', 'ко
mos_street_top15 = mos_street_top15.sort_values(by='total', ascending=False).head(15)
mos_street_top15
```

Out[33]:

category	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая	total
street									
проспект Мира	11.0	4.0	21.0	53.0	36.0	11.0	45.0	2.0	183.0
Профсоюзная улица	6.0	4.0	15.0	35.0	18.0	15.0	26.0	3.0	122.0
проспект Вернадского	7.0	1.0	12.0	25.0	16.0	12.0	33.0	2.0	108.0
Ленинский проспект	10.0	3.0	2.0	26.0	23.0	5.0	33.0	5.0	107.0
Ленинградский проспект	15.0	4.0	2.0	12.0	25.0	9.0	25.0	3.0	95.0
Дмитровское шоссе	6.0	2.0	10.0	23.0	11.0	8.0	24.0	4.0	88.0
Каширское шоссе	2.0	NaN	10.0	20.0	16.0	5.0	19.0	5.0	77.0
Варшавское шоссе	6.0	NaN	7.0	18.0	14.0	4.0	20.0	7.0	76.0
Ленинградское шоссе	5.0	2.0	5.0	13.0	13.0	3.0	26.0	3.0	70.0
МКАД	1.0	NaN	9.0	45.0	4.0	NaN	5.0	1.0	65.0
Люблинская улица	5.0	NaN	5.0	26.0	11.0	1.0	10.0	2.0	60.0
улица Вавилова	2.0	2.0	11.0	15.0	10.0	3.0	12.0	NaN	55.0
Кутузовский проспект	2.0	1.0	2.0	14.0	13.0	3.0	16.0	3.0	54.0
улица Миклухо-Маклая	3.0	NaN	4.0	21.0	4.0	2.0	15.0	NaN	49.0
Пятницкая улица	9.0	3.0	2.0	7.0	6.0	3.0	18.0	NaN	48.0

In [34]:
```
mos_street_top15 = mos_street_top15.drop(columns=['total'], axis=1)
fig = px.bar(mos_street_top15, text_auto='.0f', orientation='h')
fig.update_traces(textposition='inside', textangle=0)
```

```
fig.update_layout(title='Количество заведений по улицами и категориям',
                  xaxis_title='Округа Москвы',
                  yaxis_title='Сумма средних чеков')
```

Больше всего находится заведений на проспекте Мира, меньше же всего находится на Пятницкой улицы. Думаю такое количество связано с местоположением и протяженностью улиц

Улицы с одним объектом общепита

In [35]:
```
msones = moscow_exp
msones['name_cnt'] = moscow_exp['name']
mos_one_street = msones.pivot_table(index='street', values='name_cnt', aggfunc='count')
mos_one_street = mos_one_street.reset_index().query('name_cnt == 1')
mac = moscow_exp[['name','lat', 'lng', 'street', 'rating', 'category', 'district']]
merge_one_st = mos_one_street.merge(mac)
merge_one_st
```

Out[35]:

	street	name_cnt	name	lat	lng	rating	category	district
0	1-й Автозаводский проезд	1	чайхана азия	55.704847	37.657065	4.2	кафе	Южный административный округ
1	1-й Балтийский переулок	1	хуан хэ	55.810418	37.518824	4.4	ресторан	Северный административный округ
2	1-й Варшавский проезд	1	колизей	55.648674	37.627979	4.0	кафе	Южный административный округ
3	1-й Вешняковский проезд	1	deli by shell	55.723152	37.794014	3.4	кафе	Юго-Восточный административный округ
4	1-й Голутвинский переулок	1	shelby	55.739600	37.613494	4.1	бар,паб	Центральный административный округ
...	...	...	...	...	...	...	...	...
454	улица Шкулёва	1	мираж	55.693340	37.746231	4.7	ресторан	Юго-Восточный административный округ
455	улица Шкулёва 4	1	cofefest	55.693299	37.749927	4.4	кофейня	Юго-Восточный административный округ
456	улица Шухова	1	scirocco	55.716123	37.608472	4.1	ресторан	Южный административный округ
457	улица Юннатов	1	кафе-столовая	55.802610	37.558759	4.7	столовая	Северный административный округ
458	№ 7	1	енот	55.679064	37.615015	4.8	кафе	Южный административный округ

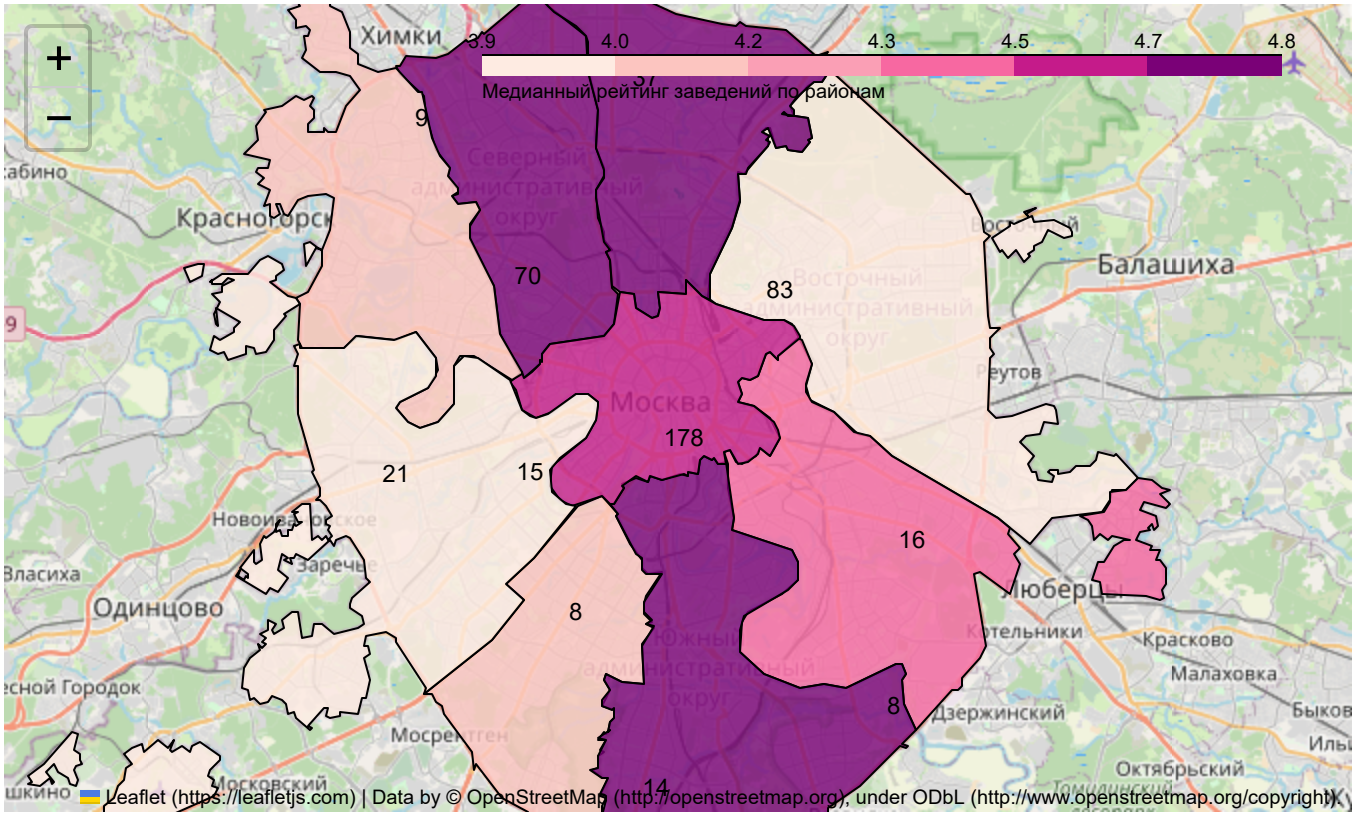
459 rows × 8 columns

In [36]:

```
moscow_lat, moscow_lng = 55.751244, 37.618423

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

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

Choropleth(
    geo_data=state_map,
    data=merge_one_st,
    columns=['district', 'rating'],
    key_on='feature.name',
    fill_color='RdPu',
    fill_opacity=0.8,
    legend_name='Медианный рейтинг заведений по районам',
).add_to(mos_map)
```

```
marker_cluster = MarkerCluster().add_to(mos_map)

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

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

# выводим карту
mos_map
```

Out[36]:



Как заметно, в центре больше всего скоплений одного заведения на одну улицу, скорее всего это связано с большим количеством коротких проездов и переулков, да еще и играет роль, что в центре естественно будет больше заведений. Далее по мере удаления от ЦФО, наблюдаем тенденцию уменьшения кол-ва заведения и улицы становятся длиннее из-за этого вероятность одного заведения на одной улице значительно уменьшается, да и заведений становится меньше. Также заметно, что в парковых, набережных и железнодорожных зонах чаще бывает одно заведение.

Значения средних чеков по округам

In [37]:

```
median_district = moscow_exp.pivot_table(index='district', values='middle_avg_bill', aggfunc='me
median_district.columns = ['district', 'middle_avg_district']
merge_median_dis = median_district.merge(moscow_exp)
merge_median_dis
```

Out[37]:

	district	middle_avg_district	name	category	address	hours	lat
0	Восточный административный округ	575.0	кафе	кафе	москва, поперечный просек, 11, стр. 2	ежедневно, 10:00–21:00	55.802843
1	Восточный административный округ	575.0	сирень	бар,паб	москва, песочная аллея, 1	ежедневно, 12:00–23:00	55.793336
2	Восточный административный округ	575.0	сеть итальянских кафе меркато	ресторан	москва, улица сокольнический вал, 1, стр. 1	пн-чт 10:00–22:00; пт-вс 10:00–23:00	55.797544
3	Восточный административный округ	575.0	green park sokolniki	кафе	москва, проезд сокольнического круга, 2	ежедневно, 11:00–23:00	55.794250
4	Восточный административный округ	575.0	чинар-а	ресторан	москва, улица сокольнический вал, 22	ежедневно, 11:00–23:00	55.791688
...	...	...	...	...	...	...	...
8397	Южный административный округ	500.0	маленький француз	быстрое питание	москва, мытная улица, 74	ежедневно, 08:00–21:00	55.711995
8398	Южный административный округ	500.0	суши рай	кафе	москва, коломенская улица, 17	ежедневно, 10:00–23:00	55.678265
8399	Южный административный округ	500.0	тачки	столовая	москва, нагорный проезд, 26	пн-сб 10:00–19:00	55.689027
8400	Южный административный округ	500.0	миславнес	кафе	москва, пролетарский проспект, 19, корп. 1	ежедневно, 08:00–22:00	55.640875
8401	Южный административный округ	500.0	kebab time	кафе	москва, россошанский проезд, 6	ежедневно, круглосуточно	55.598229

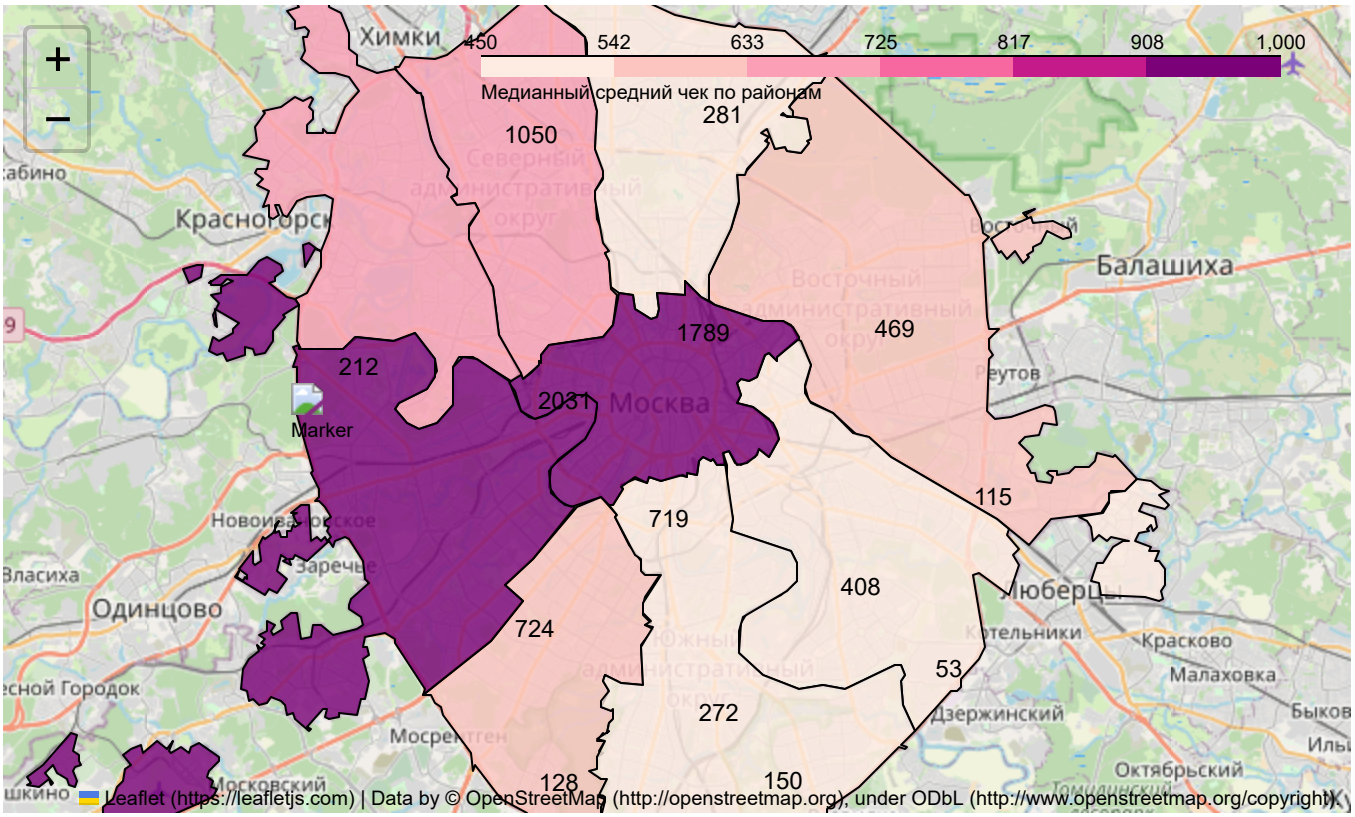
8402 rows × 18 columns

In [38]:

```
moscow_lat, moscow_lng = 55.751244, 37.618423

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

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

Choropleth(
    geo_data=state_map,
    data=merge_median_dis,
    columns=['district', 'middle_avg_district'],
    key_on='feature.name',
    fill_color='RdPu',
    fill_opacity=0.8,
```

```
    legend_name='Медианный средний чек по районам',
).add_to(mos_dis)

marker_cluster = MarkerCluster().add_to(mos_dis)

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

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

# выводим карту
mos_dis
```

Out[38]:



In [39]:

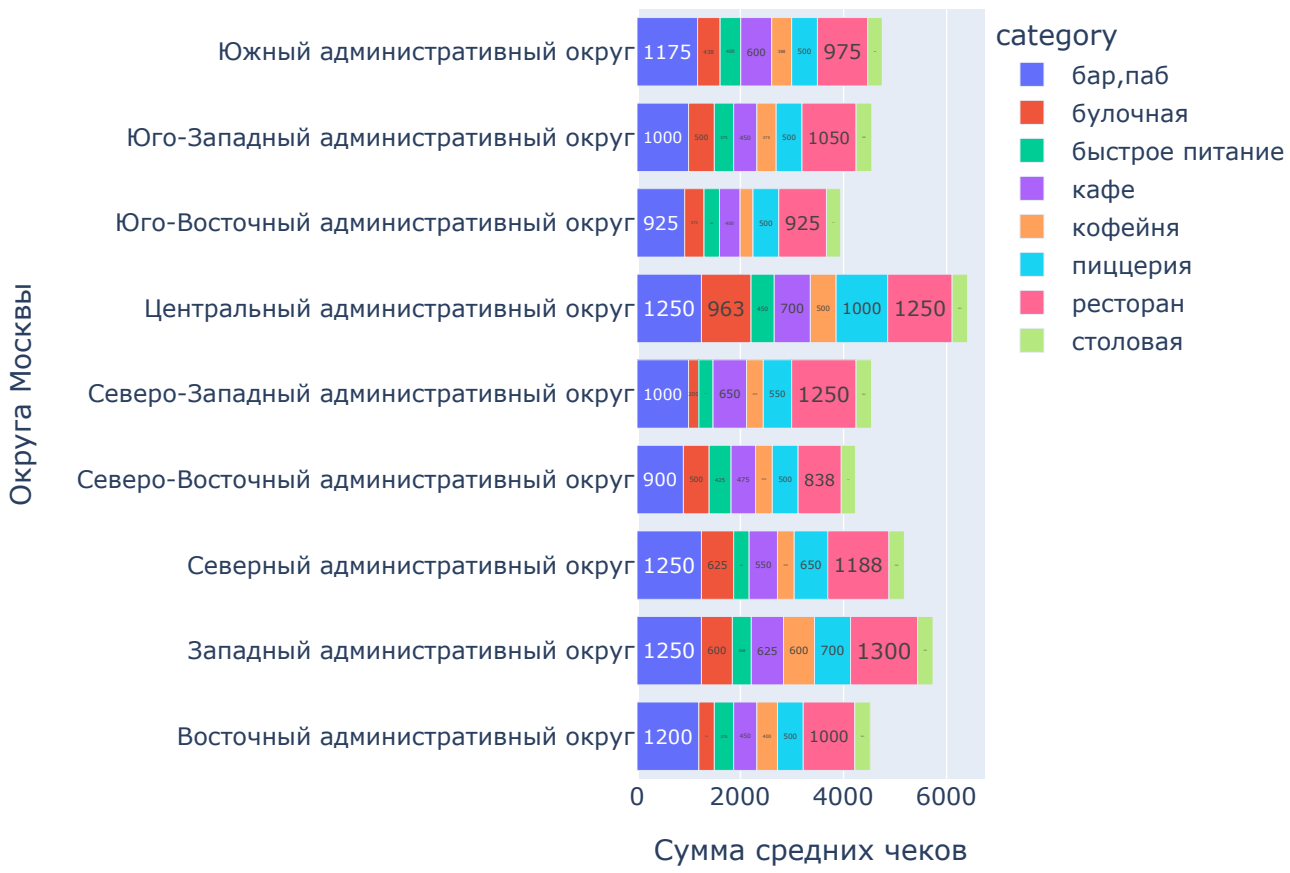
```
median_dis_categ = merge_median_dis.pivot_table(index='district', columns='category', values='mi
median_dis_categ
```

Out[39]:

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
Восточный административный округ	1200.0	300.0	375.0	450.0	400.0	500.0	1000.0	300.0
Западный административный округ	1250.0	600.0	367.5	625.0	600.0	700.0	1300.0	300.0
Северный административный округ	1250.0	625.0	300.0	550.0	325.0	650.0	1187.5	300.0
Северо-Восточный административный округ	900.0	500.0	425.0	475.0	325.0	500.0	837.5	275.0
Северо-Западный административный округ	1000.0	200.0	275.0	650.0	325.0	549.5	1250.0	300.0
Центральный административный округ	1250.0	962.5	450.0	700.0	500.0	1000.0	1250.0	300.0
Юго-Восточный административный округ	925.0	375.0	300.0	400.0	250.0	500.0	925.0	275.0
Юго-Западный административный округ	1000.0	500.0	375.0	450.0	375.0	500.0	1050.0	305.0
Южный административный округ	1175.0	437.5	400.0	600.0	387.5	500.0	975.0	282.5

In [40]:

```
fig = px.bar(median_dis_categ, text_auto='.0f', orientation='h')
fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Средний чек по округам и категориям заведений',
                  xaxis_title='Сумма средних чеков',
                  yaxis_title='Округа Москвы')
```



In [41]:
```
display(moscow_exp['seats'].corr(moscow_exp['middle_avg_bill']))
display(moscow_exp['rating'].corr(moscow_exp['middle_avg_bill']))
display(moscow_exp['lng'].corr(moscow_exp['middle_avg_bill']))
display(moscow_exp['lat'].corr(moscow_exp['middle_avg_bill']))
```

```
0.0824289311743684
0.18323797147624563
-0.05266191026894847
-0.006489025489014967
```

Проверка пр корреляции особо ничего нам не дала, а значит средний чек зависит от особенностей районов Москвы. Видим, что ЦАО и ЗАО имеют наибольший средний чек, скорее всего это связано с тем, что это самые престижные округа в Москве, на них расположены много важных объектов и достопримечательностей. Из-за этого там живут чаще состоятельные люди, да и много людей приезжает в такие места. Самыми же худшими округами являются СВАО, ЮАО и ЮВАО в этих регионах. Они являются одними из самых густонаселенных округов где построено много новостроек, и имеет проблемы с экологией из-за предприятий. Поэтому в этих районах меньше дорогих ресторанов, да и покупательская способность другая

Итак, выяснили, что в Москве:

- больше всего заведений: кафе, рестораны, кофейни
- больше всего мест в заведениях: рестораны, бары/пабы и столовые
- больше всего несетевых заведений(61.9%), сетевых меньше(38.1%)
- больше всего по количеству сетевых заведений являются кофейни, далее пиццерии, а за ними и рестораны
- самая большая сетка из Топ-15 сетевых - Шоколадница, самая маленькая Кофемания
- самый большой округ по заведениям Центральный административный округ, а самый маленький Северо-Западный административный округ
- усредненные рейтиги различаются не очень сильно по всем категориям. Булочные, столовые и заведения быстрого питания меньше всего получают самые высокие оценки
- северо-Восточный административный округ имеет больше всего положительных оценок заведений. Восточный административный округ, имеет меньше положительных оценок
- местоположение и протяженность улиц влияет на количество заведений на них
- на коротких улицах, проездах и переулках чаще можно встретить только одно заведение, особенно в центре
- на средний чек в округах влияет особенности этих округов

Открытие кофейни

Количество кофеен и их рейтинг по округам

In [42]:
```
moscow_coffee = moscow_exp.query('category == "кофейня"')
print('В Москве', moscow_coffee['name'].count(), 'кофеен')
```

```
В Москве 1413 кофеен
```

In [43]:
```
mc_pivot_cnt = (
    moscow_coffee.pivot_table(index='district', values='name', aggfunc='count')
    .reset_index())

mc_pivot_rating = (
    moscow_coffee.pivot_table(index='district', values='rating', aggfunc=['mean','median'])
    .reset_index()
    .droplevel(1, axis=1))
mc_pivot = mc_pivot_cnt.merge(mc_pivot_rating)
mc_pivot.columns = ['Округ', 'Количество кофеен', 'Средний рейтинг', 'Медианный рейтинг']
mc_pivot.sort_values(by='Средний рейтинг', ascending=False)
```

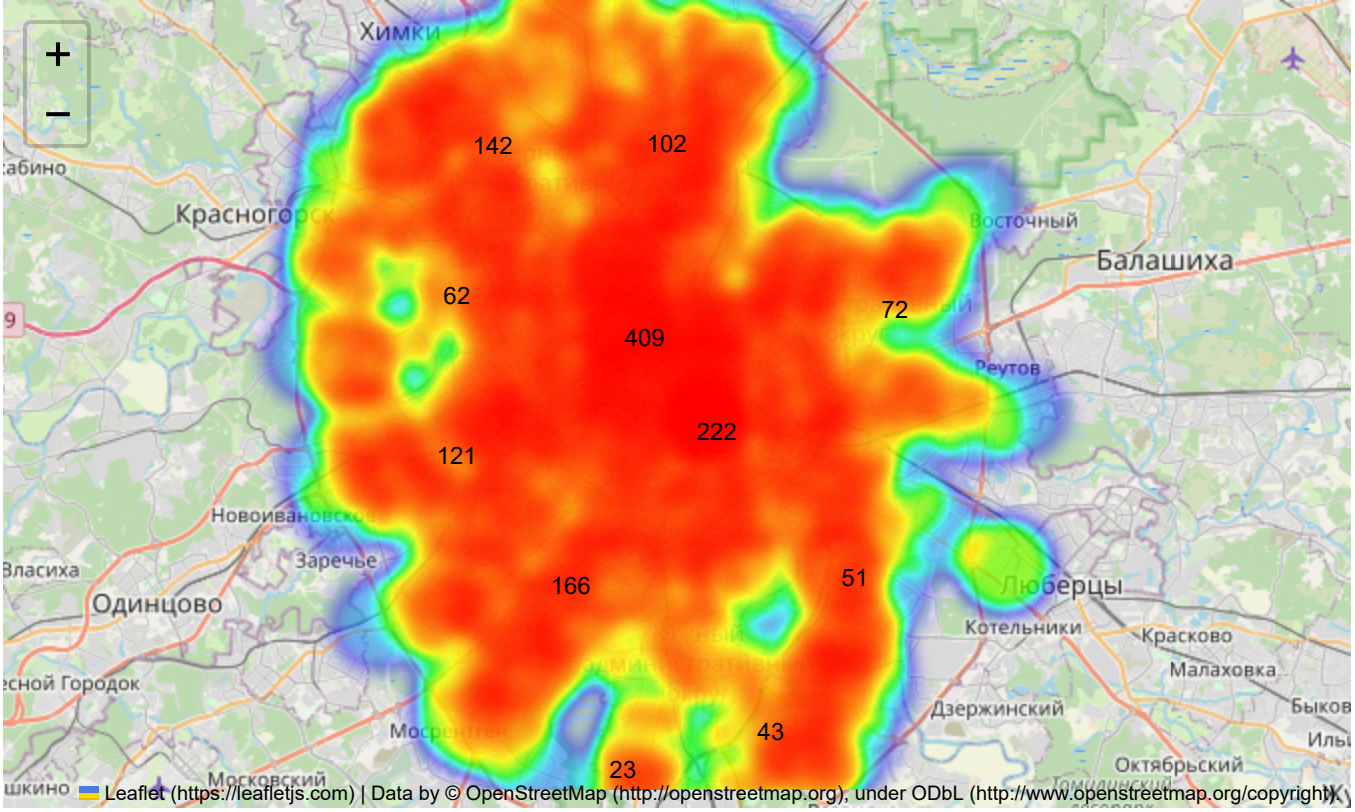
Out[43]:

	Округ	Количество кофеен	Средний рейтинг	Медианный рейтинг
5	Центральный административный округ	428	4.336449	4.3
4	Северо-Западный административный округ	62	4.325806	4.3
2	Северный административный округ	193	4.291710	4.3
7	Юго-Западный административный округ	96	4.283333	4.3
0	Восточный административный округ	105	4.282857	4.3
8	Южный административный округ	131	4.232824	4.3
6	Юго-Восточный административный округ	89	4.225843	4.3
3	Северо-Восточный административный округ	159	4.216981	4.3
1	Западный административный округ	150	4.195333	4.2

In [44]:

```
moscow_lat, moscow_lng = 55.751244, 37.618423

# создаём карту Москвы
mos_cofeee = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаём пустой кластер, добавляем его на карту

marker_cluster = MarkerCluster().add_to(mos_cofeee)

def create_clusters(row):
    Marker(
        [row['lat'], row['lng']],
        popup=f"{row['name']}  {row['rating']} {row['middle_coffee_cup']} {row['is_24/7']}",
    ).add_to(marker_cluster)

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

heatmap = moscow_coffee[['lat', 'lng']]

folium.plugins.HeatMap(heatmap).add_to(mos_cofeee)
# выводим карту
mos_cofeee
```

Out[44]:



Как видим, в Москве 1413 кофееен. Больше всего кофеен сосредоточено в Центральном, Северном и Северо-Восточном административном округе, меньше всего кофеен находится в Северо-Западном, Юго-Восточном, Юго-Западном административном округе. Из особенностей размещение видим, что много кофеен открывают у жилищного комплекса, ТЦ и ТРЦ, учебных заведений, бизнес-центрах и арт-пространств, то есть где всегда есть поток людей. Средний рейтинг округа больше всего в ЦАО, а меньше всего в ЗАО.

Круглосуточные кофейни

In [45]:
```
mos_cof_24_7 = moscow_exp[(moscow_exp['is_24/7'] == True) & (moscow_exp['category'] == 'кофейня'
print('Кофеен 24/7:', mos_cof_24_7['name'].count())
```

```
Кофеен 24/7: 91
```

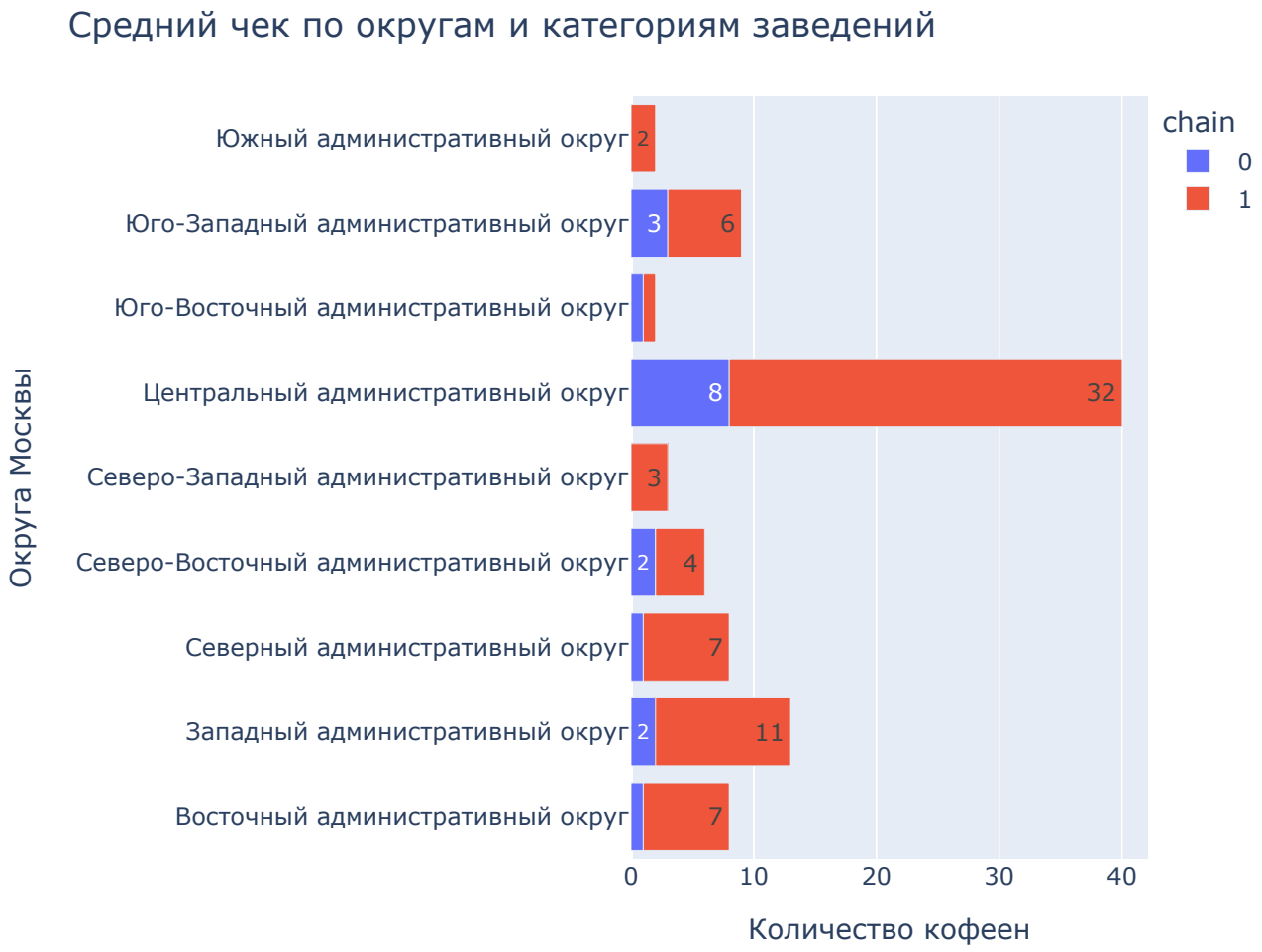
In [46]:
```
mos_chain = mos_cof_24_7.pivot_table(index='district', columns='chain', values='name', aggfunc='
mos_chain['total'] = mos_chain[[0, 1]].sum(axis=1)
mos_chain.sort_values(by='total', ascending=False)
```

Out[46]:

chain district	0	1	total
Центральный административный округ	8.0	32.0	40.0
Западный административный округ	2.0	11.0	13.0
Юго-Западный административный округ	3.0	6.0	9.0
Восточный административный округ	1.0	7.0	8.0
Северный административный округ	1.0	7.0	8.0
Северо-Восточный административный округ	2.0	4.0	6.0
Северо-Западный административный округ	NaN	3.0	3.0
Юго-Восточный административный округ	1.0	1.0	2.0
Южный административный округ	NaN	2.0	2.0

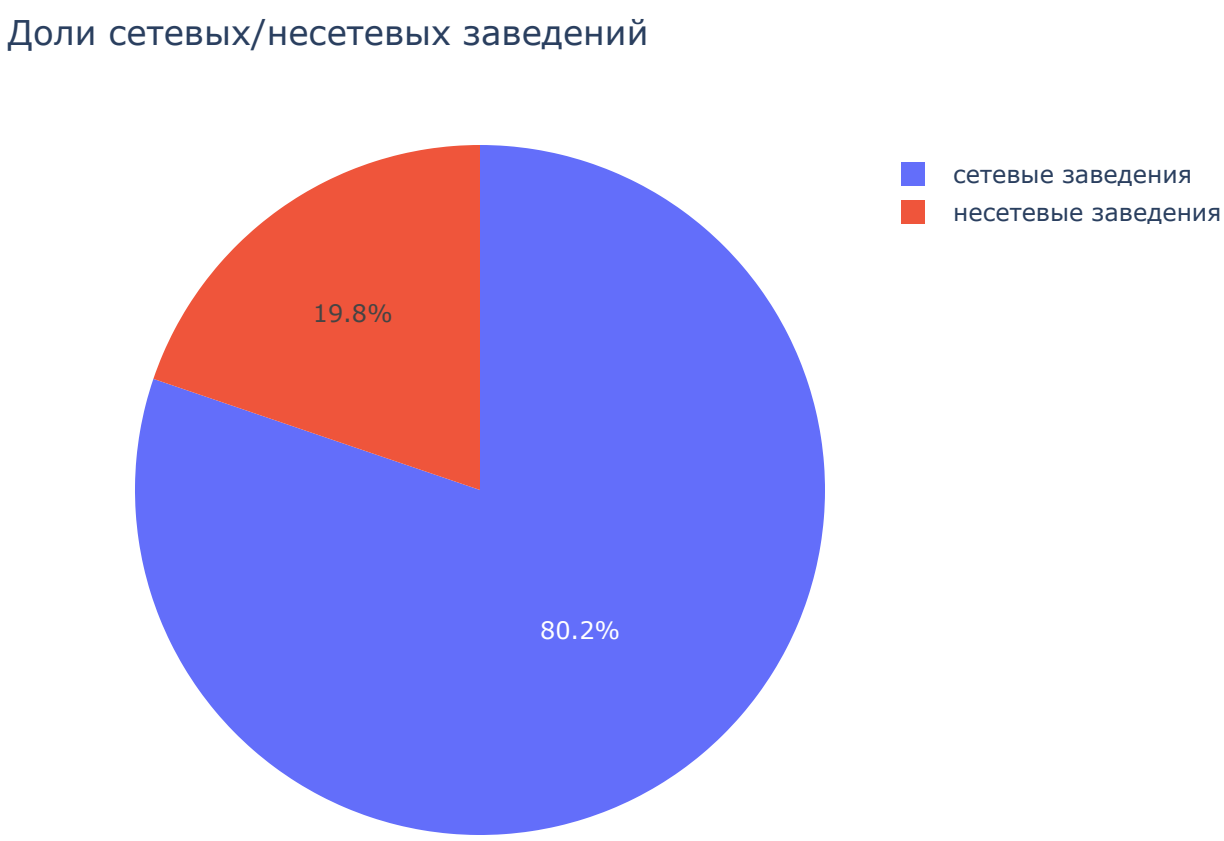
In [47]:
```
mos_chain = mos_chain.drop(columns='total', axis=1)
```

In [48]:
```
fig = px.bar(mos_chain, text_auto='.0f', orientation='h')
fig.update_traces(textposition='inside', textangle=0)
fig.update_layout(title='Средний чек по округам и категориям заведений',
                  xaxis_title='Количество кофеен',
                  yaxis_title='Округа Москвы')
```



In [49]:
```
mos_chain_pie = mos_cof_24_7.pivot_table(index='chain', values='name', aggfunc='count').reset_in
fig = go.Figure(data=[go.Pie(labels=mos_chain_pie['chain'].map({1: 'сетевые заведения', 0: 'несе
```

```
                                values=mos_chain_pie['name'])])
fig.update_layout(title_text='Доли сетевых/несетевых заведений')
fig.show()
```



91 - это число кофеен, которые работают 24/7. Из них видим, что сетевые кофейни занимают значительно большую часть - 80.2% в Центральном округе

Стоимость чашки капучино при открытии

In [50]:
```
mos_cof_cup = moscow_coffee.pivot_table(index='district', values='middle_coffee_cup', aggfunc='m
mos_cof_cup.columns = ['district', 'median_coffee_cup_dis']
mos_cof_merge = mos_cof_cup.merge(moscow_coffee)
mos_cof_cup.sort_values(by='median_coffee_cup_dis', ascending=False)
```

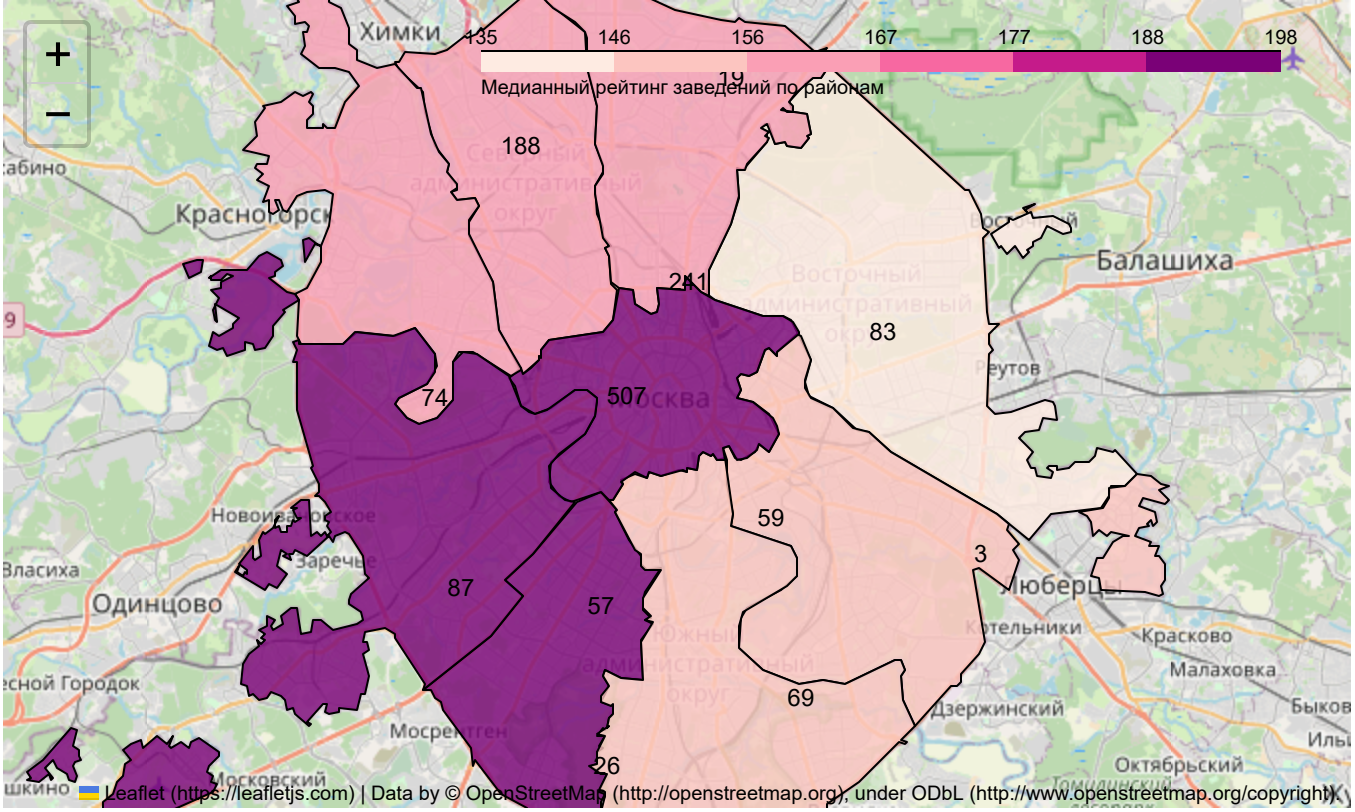
Out[50]:

	district	median_coffee_cup_dis
7	Юго-Западный административный округ	198.0
5	Центральный административный округ	190.0
1	Западный административный округ	189.0
4	Северо-Западный административный округ	165.0
3	Северо-Восточный административный округ	162.5
2	Северный административный округ	159.0
8	Южный административный округ	150.0
6	Юго-Восточный административный округ	147.5
0	Восточный административный округ	135.0

In [51]:

```
moscow_lat, moscow_lng = 55.751244, 37.618423

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

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

Choropleth(
    geo_data=state_map,
    data=mos_cof_merge,
    columns=['district', 'median_coffee_cup_dis'],
    key_on='feature.name',
    fill_color='RdPu',
    fill_opacity=0.8,
    legend_name='Медианный рейтинг заведений по районам',
).add_to(mos_cofeee)

marker_cluster = MarkerCluster().add_to(mos_cofeee)

def create_clusters(row):
    Marker(
        [row['lat'], row['lng']],
        popup=f"{row['name']}  {row['rating']} {row['middle_coffee_cup']} {row['is_24/7']}",
    ).add_to(marker_cluster)

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

# выводим карту
mos_cofeee
```

Out[51]:



Наблюдаем, что самые высокие цены за чашку капучино в ЦАО, ЗАО и ЮЗАО, а самые маленькие цены в ВАО.

Вывод

1. Открыли файлы и изучили данные

- было обнаружено множество пропусков
- были обнаружено много некорректных данных на первый взгляд

2. Провели предобработку

- привели названия колонок к нижнему регистру и переименовали как удобно
- заполнили пропуски заглушками где смогли
- ввели два новых столбца `is_24/7` и `street`
- избавились от неявных дубликатов, где смогли найти
- исправили аномальные значения и заполнили пропуски медианой по категориям

3. Проанализировали данные

- выяснили количество заведений по категориям
- выяснили минимальную и максимальную дату и отрезок проведения эксперимента
- выяснили наиболее часто встречающиеся количество мест в заведениях по категориям
- выяснили доли сетевых/несетевых заведений
- выяснили доли сетевых заведений по категориям

- выяснили ТОП-15 сетевых заведений
- выяснили какое количество заведений в категориях по административным округам
- сделали распределение средних рейтингов по категориям заведений
- построили фоновую картограмму заведений
- выяснили ТОП-15 улиц по количеству заведений
- выяснили какие улицы с одним объектом общепита
- выяснили средние чеки по административным округам

4. Детализировали исследование для открытия кофейни

- выяснили количество кофеен
- выяснили рейтинг кофеен по административным округам
- выяснили количество кофеен, которые работают 24/7
- выяснили количество сетевых и несетевых кофеен, которые работают 24/7 и рассчитали их долю
- выяснили какую стоимость определять для чашки капучино при открытие кофейни

Целью исследования было: найти интересные особенности и презентовать полученные результаты, которые в будущем помогут в выборе подходящего инвесторам места

В ходе проведения исследования сделали выводы:

1. Среди категорий заведений популярны такие категории: кафе, ресторан, кофейни. Рынок заведений полон этими категориями, следовательно там больше конкуренции.
2. Рестораны, бары/пабы и кофейни обычно требуют больше всего посадочных мест, значит площадь заведения должна быть большая.
3. Несетевых заведений в Москве больше, процентные пункты составляют `61.9%`
4. Булочные, пиццерии и кофейни занимают большую часть сетевых заведений
5. В Центральном административном округе больше всего количества заведений, то есть конкуренция там высокая. В Северо-Западном административном округе меньше всего заведений следовательно и конкуренции там будет меньше.
6. Рейтинг заведения по категориям выглядит примерно одинаковым по всем категориям, но булочные, столовые и заведения быстрого питания меньше всего получают самые высокие оценки.
7. Протяженность улицы зависит на количество на ней заведений, чем длиннее улица, тем больше заведений на ней.
8. В парковых, набережных и зонах рядом железной дорогой чаще бывает одно заведение.
9. Средний чек в заведениях чаще всего зависит от особенностей округа

Рекомендации при открытие кофейни:
Кофейню лучше всего открывать в местах, где лучше всего проходимость. Проходимость в Центральном округе больше всего.

Среди самых проходимых улиц(данные из исследования Геомаркетингового агенства One by One | 1by1):

- ул. Маросейка
- пр. Мира
- ул. Бауманская
- ул. Тверская

Необязательно открывать кофейню именно на этих улицах, можно присмотреться к близлежайшим улицам, переулкам, проездам.

В Центральном административном округе средняя цена за чашку капучино 190р. Так как заведение будет новым, рекомендую снизить цену на 10-20 рублей, а в дальнейшем с течением времени уже обновить прайс.

Делать круглосуточную кофейню не стоит, так как эту нишу заняли сетевые кофейные у которых уже все процессы настроены. Открываться по френчайзингу в теории можно, есть выгода за счёт настроенных процессов, рекламы, бренда, но вы лишаетесь индивидуальности и регламентов, которые могут быть неподходящими под вашу философию. При открытие открытие по френчайзингу - это заведение не будет вашим детищем, а вы там будете как инвестор с долей менеджмента.

Ссылка на презентацию: ТЫК