Учебный проект №11

Рынок заведений общественного питания Москвы

студента когорты DA61
Алиева Рустама

Цель: отработка навыков работы с инструментами визуализации, а также создания презентационных материалов на основе исследования

Контекст: проведение исследования рынка общественного питания Москвы для инвесторов, предполагающих открытие кофейни

Описание исходных данных: имеем в наличии датасет, содержащий информацию о различных заведениях общепиты Мск (название, категория), включая геолокацию (адрес, координаты) и данные, заполненные пользователями (рейтинг, уровень цен, режим работы, принадлежность к сети)

План работы:

- знакомство с данными и предобработка
- дополнение необходимыми для работы данными
- поиск и анализ разнообразных закономерностей в данных с построением разнообразных визуализаций
- детализация результатов исследования применительно к открытию кофейни
- выводы

In [1]:
```
# блок загрузки библиотек
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from numpy import median, mean
import plotly.express as px
from plotly import graph_objects as go

# для геоданных
try:
    from folium import Map, Choropleth, Marker
    from folium.plugins import MarkerCluster
except:
    !pip install folium
    from folium import Map, Choropleth, Marker
    from folium.plugins import MarkerCluster

# установим стиль для снс
sns.set_style('darkgrid')

# скрываем предупреждения
import warnings
warnings.filterwarnings('ignore')
```

Загрузка и общее знакомство с данными

In [2]:
```
# загружаем данные локально или с сервера
try:
    df = pd.read_csv('moscow_places.csv')
except:
    df = pd.read_csv('network path hidden')
```

Изучим общую информацию о ДФ

In [3]:
```
df.info()
```

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

In [4]:
```
df.head()
```

Out[4]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_avg_bill	middle_coffee_cup
0	WoWфли	кафе	Москва, улица Дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN	NaN	NaN	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	Средний счёт:1500–1600 ₽	1550.0	NaN
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	Средний счёт:от 1000 ₽	1000.0	NaN
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	Цена чашки капучино:155–185 ₽	NaN	170.0
4	Иль Марко	пиццерия	Москва, Правобережная улица, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние	Средний счёт:400–600 ₽	500.0	NaN

Исходно имеем дф с 14 столбцами, строк в таблице 8406 (без учёта вероятных дублей, это количество заведений общепита). Типы данных, на первый взгляд, соответствуют имеющимся данным. В таблице много текстовой информации, заполненной пользователями, что осложняет анализ данных, часть цифровой информации задана одним числом (средний счёт), при этом, другая - диапазоном.
Очень много пропусков в части цен, среднего чека, посадочных мест.

Предобработка данных

Поиск дубликатов

Поиск явных дубликатов

In [5]:
```
df.duplicated().sum()
```

Out[5]:
```
0
```

Явные дубликаты отсутствуют.
Попробуем найти неявные дубли, для начала создадим копии столбцов с адресом и названием в нижнем регистре. Затем по ним попробуем поискать дубликаты.

In [6]:
```
df['name_low'] = df['name'].str.lower()
df['address_low'] = df['address'].str.lower()
```

In [7]:
```
df[df[['name_low', 'address_low']].duplicated()]['name'].count()
```

Out[7]:
```
4
```

Обнаружилось 4 пары дублей. Адреса и названия совпадают - очевидно, это дубли, вызванные ошибками ввода. Можно удалить.

In [8]:
```
df = df[~df[['name_low', 'address_low']].duplicated()]
```

Поиск аномалий и выбросов

Рассмотрим типы заведений, нет ли там аномалий

In [9]:
```
df['category'].value_counts()
```

Out[9]:
```
category
кафе               2376
ресторан           2042
кофейня            1413
бар,паб             764
пиццерия            633
быстрое питание     603
столовая            315
булочная            256
Name: count, dtype: int64
```

Повторяющихся значений и ошибок нет. Переходим к району.

In [10]:
```
df['district'].value_counts()
```

Out[10]:
```
district
Центральный административный округ          2242
Северный административный округ              898
Южный административный округ                 892
Северо-Восточный административный округ      890
Западный административный округ              850
Восточный административный округ             798
Юго-Восточный административный округ         714
Юго-Западный административный округ          709
Северо-Западный административный округ       409
Name: count, dtype: int64
```

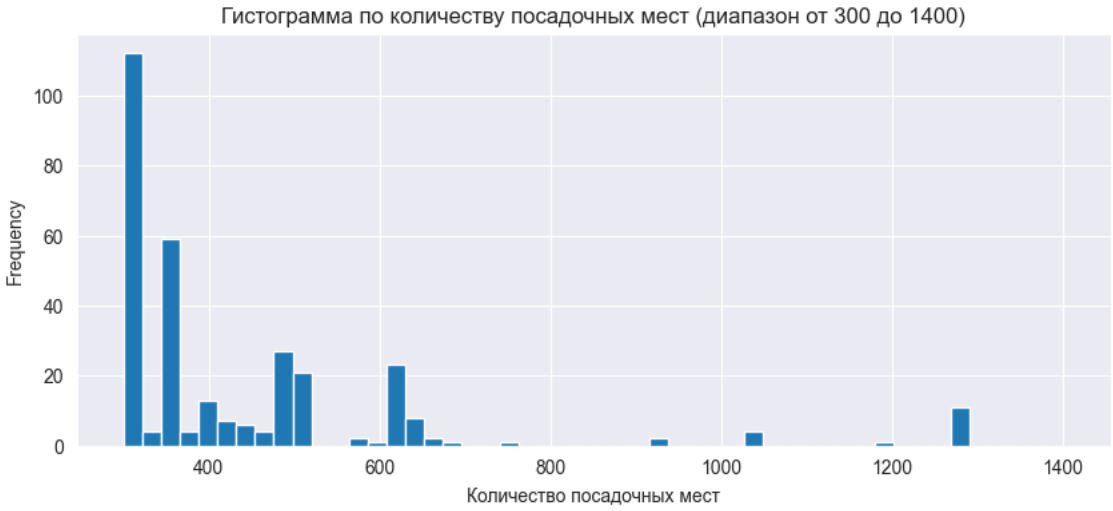
Ошибок также нет, можно анализировать следующий столбец. Посмотрим на количество посадочных мест.

In [11]:
```
df['seats'].describe()
```

Out[11]:
```
count    4792.000000
mean      108.361436
std       122.841130
min         0.000000
25%        40.000000
50%        75.000000
75%       140.000000
max      1288.000000
Name: seats, dtype: float64
```

1288 даже для очень крупного заведения - очень крупно. Возможно, в данных какая-то ошибка. Построим гистограмму, сразу подсветим "подозрительные" зоны.

In [12]:
```
df['seats'].plot(
        kind='hist'
       ,bins=50
       ,range=(300,1400)
       ,figsize=(10,4)
       ,title='Гистограмма по количеству посадочных мест (диапазон от 300 до 1400)'
    )
plt.xlabel('Количество посадочных мест')
plt.show()
```



In [13]:
```
df['seats'].plot(
        kind='hist'
       ,bins=40
       ,range=(0,40)
       ,figsize=(10,4)
       ,title='Гистограмма по количеству посадочных мест (диапазон от 0 до 40)'
    )
plt.xlabel('Количество посадочных мест')
plt.show()
```

Можно предположить, что часть заведений находится в ТРЦ. И в качестве количества посадочных мест указано общее количеств мест на фудкорте. Основная часть заведений ограничена границами межквартильного размаха: от 40 до 140. Заведения с количеством мест больше 650 можно считать выбросами и удалить из данных. Что касается заведений с количеством мест 0, то это могут быть заведения в формате "eat'n'go", где "рассиживаться" негде, всё готовится "навынос".
Оценим, сколько данных планируется к удалению.

In [14]:
```
display(df.query('seats > 650 and ~seats.isna()')['name'].count())
```


Величина незначительная для более чем 8000 записей

In [15]:
```
# удаляем выбросы по местам
df = df.query('seats <= 650 or seats.isna()')
```

Сетевые заведения

Часть заведений относятся к сетевым. Об этом говорит флаг "chain". Создадим список сетевых заведений

In [16]:
```
chain_list = df.query('chain == 1')['name'].sort_values().unique()
print('Всего сетевых заведений: {}'.format(len(chain_list)))
```

```
Всего сетевых заведений: 762
```

Чтобы заполнить часть пропусков по ценам, можно допустить, что все заведения одной сети должны относиться к одной ценовой категории и должны плюс-минус быть равны по средним ценам (на практике, это не всегда так, конечно).
Заполним пропуски по среднему чеку и средней стоимости кофе средним значением по "сети".

In [17]:
```
print('Количество пропусков в графе Средний чек до заполнения: {}'.format(df['middle_avg_bill'].isna().sum()))
print('Количество пропусков в графе Средняя стоимости чашки кофе до заполнения: {}'.format(df['middle_coffee_cup'].isna().sum()))
```

```
Количество пропусков в графе Средний чек до заполнения: 5237
Количество пропусков в графе Средняя стоимости чашки кофе до заполнения: 7845
```

In [18]:
```
# будем заполнять, соблюдая изначальную логику:
# если есть информация о среднем чеке, то нет информации о средней стоимости чашки, и наоборот
for i in chain_list:
    avg_bill = df.query('name == @i')['middle_avg_bill'].mean()
    avg_cup = df.query('name == @i')['middle_coffee_cup'].mean()

    df.loc[(df['name'] == i) & (df['middle_coffee_cup'].notna()), 'middle_avg_bill'] =\
        df.loc[(df['name'] == i) & (df['middle_coffee_cup'].notna()), 'middle_avg_bill'].fillna(avg_bill)

    df.loc[(df['name'] == i) & (df['middle_avg_bill'].notna()), 'middle_coffee_cup'] =\
        df.loc[(df['name'] == i) & (df['middle_avg_bill'].notna()), 'middle_coffee_cup'].fillna(avg_cup)
```

In [19]:
```
print('Количество пропусков в графе Средний чек *ПОСЛЕ* заполнения: {}'.format(df['middle_avg_bill'].isna().sum()))
print('Количество пропусков в графе Средняя стоимости чашки кофе *ПОСЛЕ* заполнения: {}'.format(df['middle_coffee_cup'].isna().sum()))
```

```
Количество пропусков в графе Средний чек *ПОСЛЕ* заполнения: 5149
Количество пропусков в графе Средняя стоимости чашки кофе *ПОСЛЕ* заполнения: 7820
```

Анализ данных чеков

Рассмотрим данные по среднему чеку в заведениях разного ценового уровня.

In [20]:
```
(
    df.pivot_table(
         index='price'
        ,values=['middle_avg_bill', 'middle_coffee_cup']
        ,aggfunc=['min', 'max']
        )
    .sort_values(by=('max','middle_avg_bill'))
)
```

Out[20]:

	min		max	
	middle_avg_bill	middle_coffee_cup	middle_avg_bill	middle_coffee_cup
price				
низкие	90.0	60.0	679.166667	256.000000
средние	165.0	60.0	2150.000000	1568.000000
выше среднего	375.0	150.0	4500.000000	277.193548
высокие	0.0	250.0	35000.000000	290.000000

К сожалению, информацию не везде полная (нет информации по мин. среднему чеку заведений с высокими ценами), а в некоторых случаях и противоречивая: максимальная средняя стоимость чашки кофе в заведении среднего уровня больше, чем в заведениях высокого ценового уровня. Но в целом, картина целостная и логичная.

Столбец street

На основе адреса заведения выделим название улицы и запишем в отдельный столбец

In [21]:
```
df['street'] = df['address'].apply(lambda x: x.split(', ')[1])
# сразу можно посмотреть на самые популярные улицы
df['street'].value_counts().head(10)
```

Out[21]:
```
street
проспект Мира             183
Профсоюзная улица         122
Ленинский проспект        107
проспект Вернадского       97
Ленинградский проспект     95
Дмитровское шоссе          88
Каширское шоссе            77
Варшавское шоссе           76
Ленинградское шоссе        70
МКАД                       65
Name: count, dtype: int64
```

Сразу выявляются самые "питательные" улицы - это проспект Мира, Профсоюзная и Ленинский проспект - количество заведений на них превышает 100.

Заведения 24/7

Выделим "флажком" заведения, которые оказывают услуги круглосуточно и ежедневно.

In [22]:
```
df['is_24/7'] = (df['hours'].str.contains('ежеднев')) & (df['hours'].str.contains('круглос'))
print('Количество круглосуточных заведений общественного питания в базе: {}'.format(df['is_24/7'].sum()))
print('Что составляет: {:.2%} от общего числа заведений'.format(df['is_24/7'].sum() / df.shape[0]))
```

```
Количество круглосуточных заведений общественного питания в базе: 728
Что составляет: 8.69% от общего числа заведений
```

In [23]:
```
# посмотрим, что получилось с сходным дф
df.head()
```

Out[23]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_avg_bill	middle_coffee_cup
0	WoWфли	кафе	Москва, улица Дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN	NaN	NaN	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	Средний счёт:1500–1600 ₽	1550.0	NaN
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	Средний счёт:от 1000 ₽	1000.0	NaN
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	Цена чашки капучино:155–185 ₽	NaN	170.0
4	Иль Марко	пиццерия	Москва, Правобережная улица, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние	Средний счёт:400–600 ₽	500.0	NaN

Анализ данных

Категории заведений

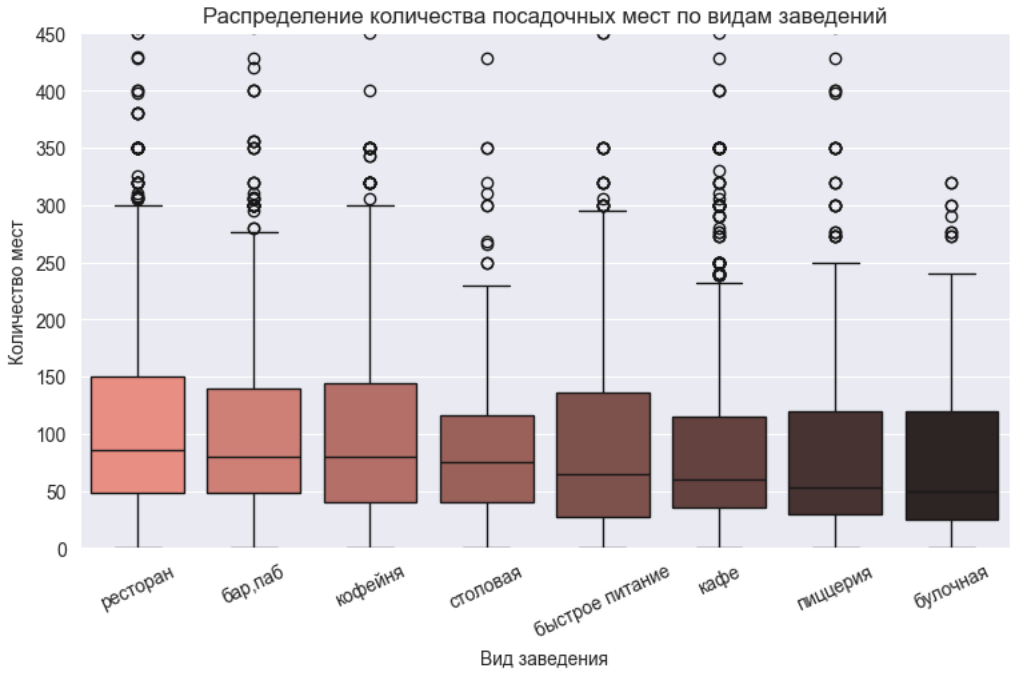
In [24]:
```
# строим график
fig = px.histogram(df
                   ,x='category'
                   ,text_auto=True
                   )
fig.update_layout(title='Количество заведений по категориям в Москве'
                  ,width=900
                  ,height=500
                  )
fig.update_xaxes(title_text='Категории заведений', categoryorder='total descending')
fig.update_yaxes(title_text='Количество заведений')
fig.show()
```

В наибольшем количестве на рынке МСК представлены классические кафе (более 2300 заведений), за ними в рейтинге идут рестораны (около 2 тыс.) и кофейни (чуть меньше 1,5 тыс). Меньше всех булочных и столовых (примерно по 300 заведений каждого вида).

Количество посадочных мест в местах по категориям

Оценим на сколько мест расчитаны заведения общепита в МСК в разрезе их категорий.

In [25]:
```
plt.figure(figsize=(9, 5))
sns.boxplot(
     x='category'
    ,y='seats'
    ,data=df
    ,palette='dark:salmon_r'
    ,order=(df.groupby('category', as_index=False)
            ['seats'].median().sort_values(by='seats', ascending=False)
            ['category']
           )
       )
plt.title('Распределение количества посадочных мест по видам заведений')
plt.xlabel('Вид заведения')
plt.ylabel('Количество мест')
plt.ylim(0,450) # подрежем максимальные значения, чтобы сделать акцент на главном
plt.xticks(rotation=25);
```

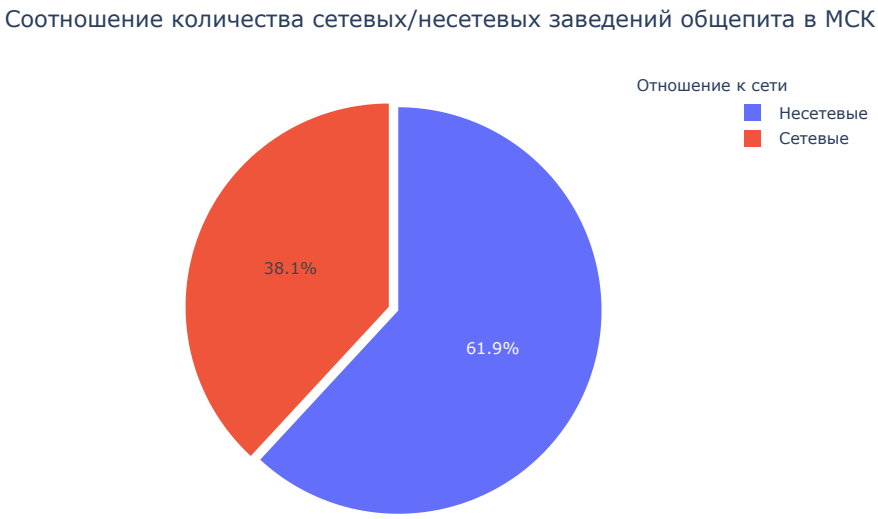


Боксплот хорошо показывает, что для всех типов заведений медианным количеством посадочных мест является диапазон от 50 до 100. Межквартильный размах, как упоминалось раньше, находится для всех заведений в диапазоне 40-140 мест. Другие значения - очень большие и достаточно маленькие - встречаются намного реже и могут быть связаны как особенностью конкретной локации, так и с ошибками ввода.

Соотношение сетевых и несетевых заведений

Посмотрим, как соотносится количество сетевых и несетвых заведений в целом по рынку, не разбивая их на категории

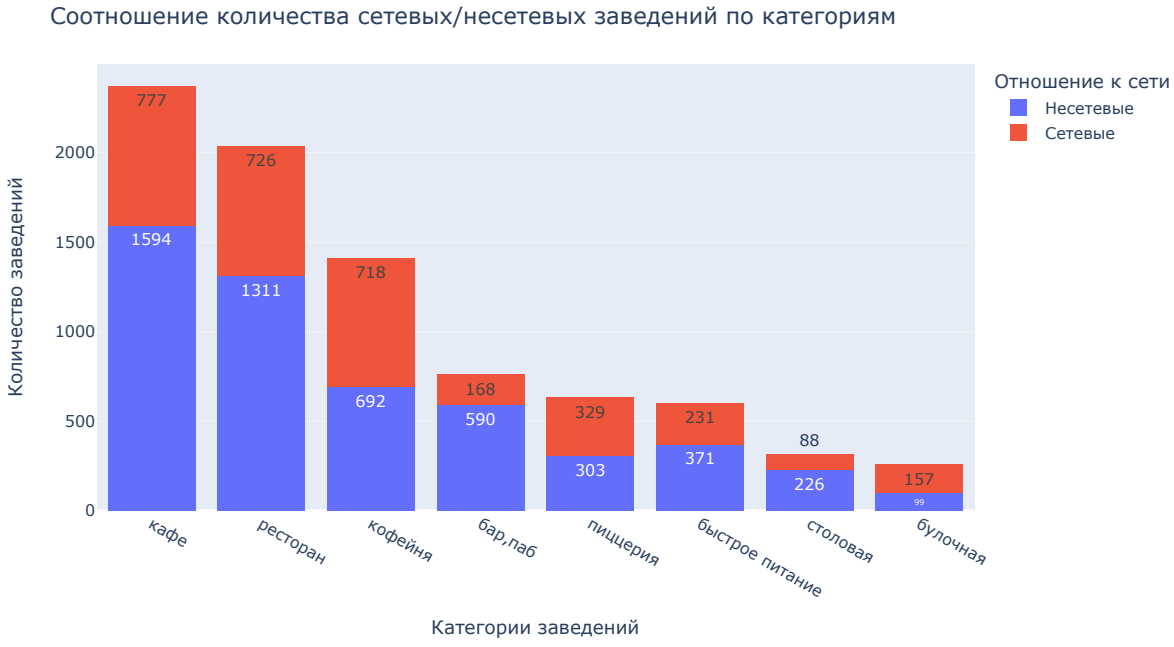
In [26]:
```
# подгтовка дф
df_chain = pd.DataFrame(df['chain'].value_counts()).reset_index()
df_chain = df_chain.reset_index()

# столбик с категорией
df_chain['cat'] = df_chain['index'].apply(lambda x: 'Несетевые' if x == 0 else 'Сетевые')

# строим пай
fig = go.Figure(data=[go.Pie(labels=df_chain['cat']
                             ,values=df_chain['count']
                             ,pull = [0.05, 0])])
fig.update_layout(title='Соотношение количества сетевых/несетевых заведений общепита в МСК'
                  ,width=700
                  ,height=500
                  ,annotations=[dict(x=1.12
                                     ,y=1.05
                                     ,text='Отношение к сети'
                                     ,showarrow=False)]
                 )
fig.show()
```



Анализ общего соотношения заведений показывает, что почти 2/3 заведений не являются сетевыми и представлены в единственном "экземпляре".

Следующим этам проанализируем соотношение сетевых и несетевых организаций по каждой категории.

In [27]:
```
# для красоты отображения добавим столбик в дф
df['Отношение к сети']= df['chain'].apply(lambda x: 'Несетевые' if x == 0 else 'Сетевые')
# строим график
fig = px.histogram(df
                   ,x='category'
                   ,color='Отношение к сети'
                   ,text_auto=True
                   )
fig.update_layout(title='Соотношение количества сетевых/несетевых заведений по категориям'
                  ,width=900
                  ,height=500
                  )
fig.update_xaxes(title_text='Категории заведений', categoryorder='total descending')
fig.update_yaxes(title_text='Количество заведений')
fig.show()
```

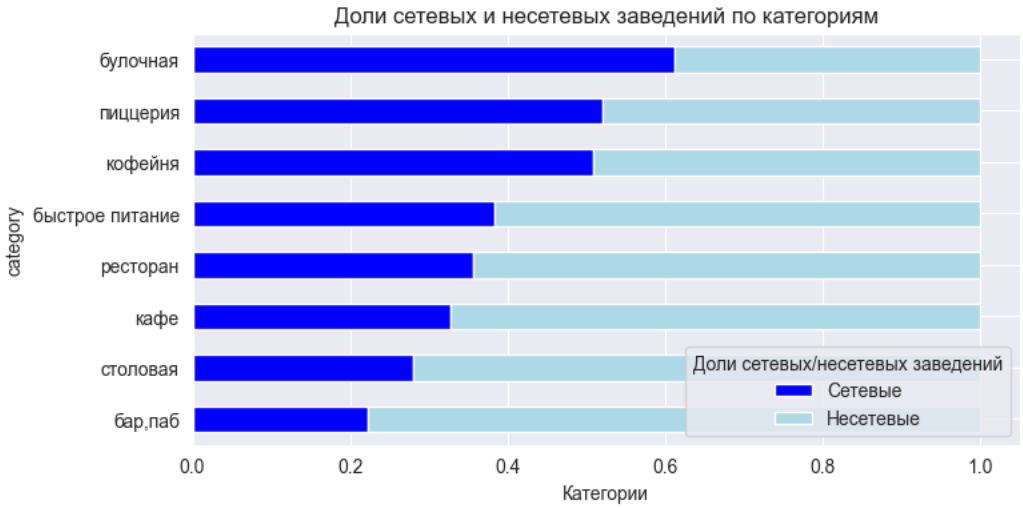


Категории заведений, которые чаще являются сетевыми

Посмотрим на относительные данные, чтобы выявить однозначных лидеров по присутствию сетевых организаций

In [28]:
```
df_chain = (
    df.groupby(by='category')
    .agg({'chain':'mean'})
    .round(3)
    .sort_values(by='chain')
    )
df_chain['solo'] = 1 - df_chain['chain']
df_chain.columns=['Сетевые', 'Несетевые']

ax = (
      df_chain
        .plot
```

```
        .barh(
              figsize=(8, 4)
              ,title='Доли сетевых и несетевых заведений по категориям'
              ,xlabel='Категории'
              ,stacked=True
              ,color=['blue', 'lightblue']
             )
       )
plt.legend(loc='lower right', title='Доли сетевых/несетевых заведений')
plt.show()
```



Можно отметить, что кафейни, пиццерии и булочные в большей части случаев являются сетевыми. Менее всех в сети объединены бары/пабы и столовые.
По графику с относительными показателями однозначно можно выделить лидера по сетевым заведениям - это булочные.

Топ-15 популярных сетей в Москве

Найдём топ-15 популярных сетей в Москве (популярность - количество заведений этой сети)

In [29]:
```
# сгруппируем дф
df_top15 = df.query('chain == 1')[['name', 'category']].value_counts()
df_top15 = df_top15.reset_index().head(15)
# сразу создадим дф с популярными категорями в топ-15
df_top15_share = df_top15.groupby('category')['count'].sum().reset_index().sort_values(by='count', ascending=False)
```

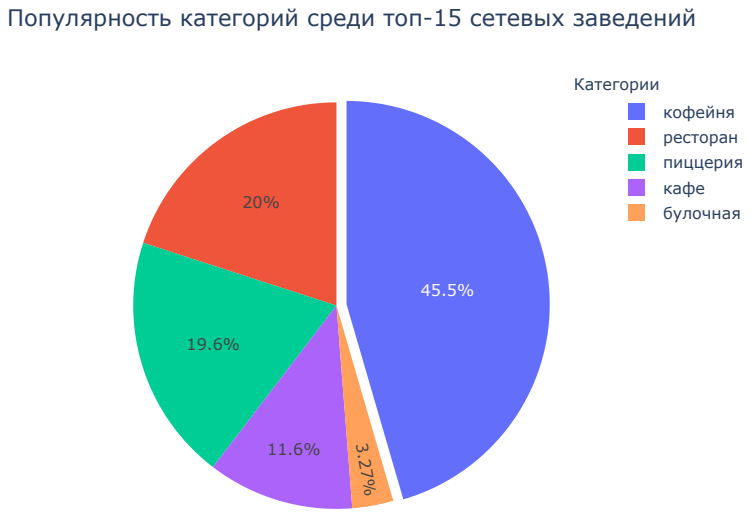
In [30]:
```
df_top15.columns=['Имя', 'Категория', 'Количество заведений']

# строим график
fig = px.bar(df_top15.sort_values(by='Количество заведений', ascending=True)
             ,y='Имя'
             ,x='Количество заведений'
             ,text_auto=True
             )
fig.update_layout(title='Топ-15 сетевых заведений МСК по популярности'
                  ,width=1000
                  ,height=500
                 )
fig.update_xaxes(title_text='Количество заведений')
fig.update_yaxes(title_text='Имена сетей')
fig.show()
```

Наиболее популярная сеть заведений - Шоколадница, в неё входит 118 заведений категории "кофейня".
Необходимо учитывать, что подобная группировка является лишь одним из возможных вариантов, т.к. не учитывает тот факт, что у некоторых сетей разные

"точки" могут фигурировать под разными категориями. Но смысл существенно не меняется (проверял!)

Посмотрим, на то, какие вообще категории присущи топ-15 сетевых заведений

In [31]:
```
# строим пай
fig = go.Figure(data=[go.Pie(labels=df_top15_share['category']
                             ,values=df_top15_share['count']
                             ,pull = [0.05, 0])])
fig.update_layout(title='Популярность категорий среди топ-15 сетевых заведений'
                  ,width=600
                  ,height=500
                  ,annotations=[dict(x=1.12
                                     ,y=1.05
                                     ,text='Категории'
                                     ,showarrow=False)]
                 )
fig.show()
```



Почти половина (45,5%) из всех самых популярных сетевых заведений - это кофейни. Также популярны пиццерии и рестораны. Полагаю, что главный признак наиболее популярных сетей - это формат быстрого питания и устоявшихся стандартов обслуживания. Для потребителя важно - за вменяемые деньги быстро перекусить, будучи уверенным в постоянстве качества вне зависимости от местоположения. На эти принципах живут большинство мировых сетей быстрого питания (досужее мнение).

Заведения общепита по административным районам Москвы

Проанализируем, как распределены заведения по районам МСК.
В дата-сете представлены следующие районы Москвы:

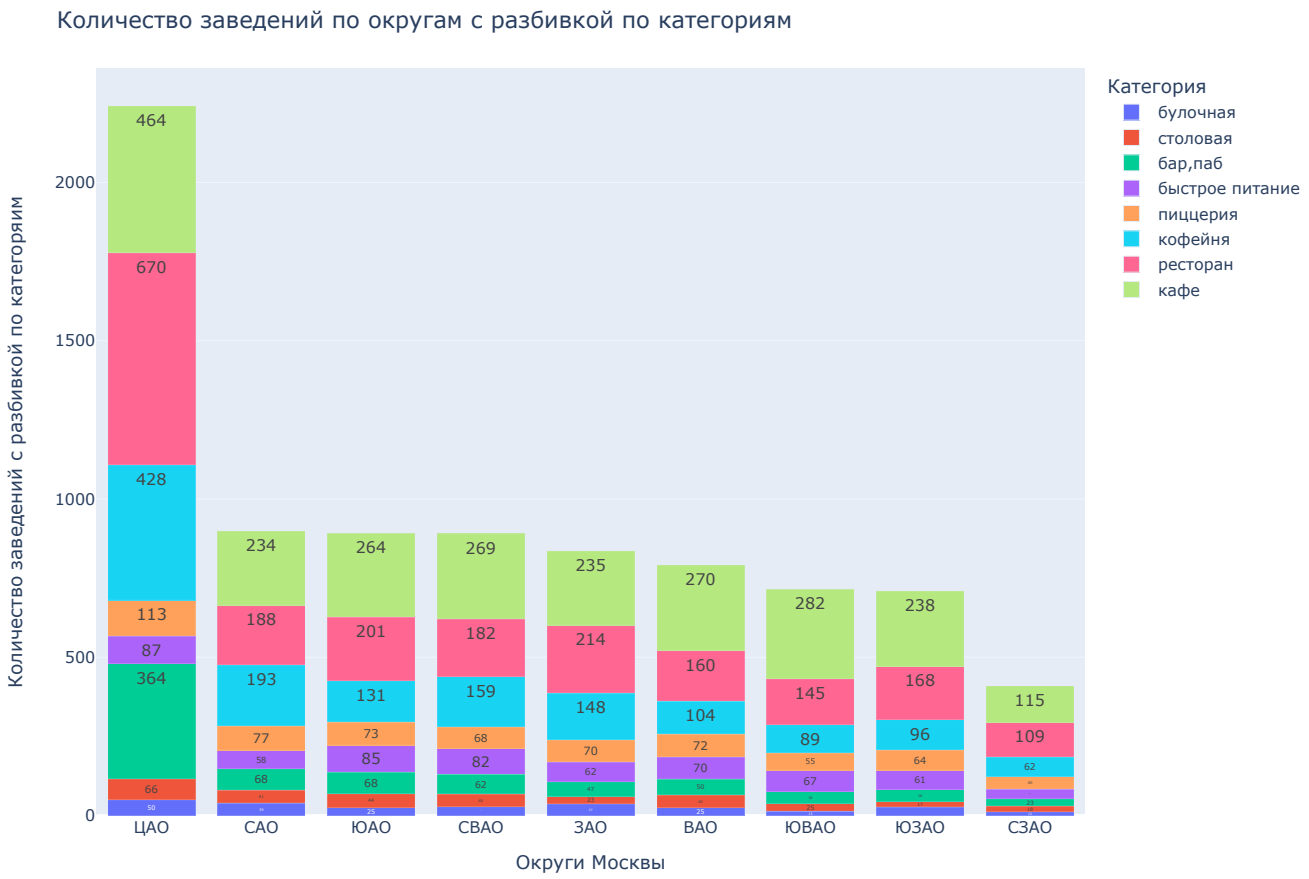
In [32]:
```
df['district'].unique()
```

Out[32]:
```
array(['Северный административный округ',
       'Северо-Восточный административный округ',
       'Северо-Западный административный округ',
       'Западный административный округ',
       'Центральный административный округ',
       'Восточный административный округ',
       'Юго-Восточный административный округ',
       'Южный административный округ',
       'Юго-Западный административный округ'], dtype=object)
```

In [33]:
```
# создадим столбец с сокращенными значениями округов, для красоты на графиках
replace_distr = {'Северный административный округ':'САО',
                 'Северо-Восточный административный округ':'СВАО',
                 'Северо-Западный административный округ':'СЗАО',
                 'Западный административный округ':'ЗАО',
                 'Центральный административный округ':'ЦАО',
                 'Восточный административный округ':'ВАО',
                 'Юго-Восточный административный округ':'ЮВАО',
                 'Южный административный округ':'ЮАО',
                 'Юго-Западный административный округ':'ЮЗАО'
                }
df['district_short']=df['district'].replace(replace_distr)
```

In [34]:
```
# сгруппируем дф для построения визуализации

df_distr=(
    df[['district_short', 'category']]
    .value_counts()
    .reset_index(name='count')
    .sort_values(by='count', ascending=True)
)
df_distr.columns=['Округ', 'Категория', 'Количество']
```

In [35]:
```
fig = px.bar(data_frame=df_distr
             ,x='Округ'
             ,y='Количество'
             ,color="Категория"
             ,barmode="stack"
             ,text_auto=True
            )
fig.update_layout(title='Количество заведений по округам с разбивкой по категориям'
                  ,width=1000
                  ,height=700
                  )
fig.update_xaxes(title_text='Округи Москвы', categoryorder='total descending')
fig.update_yaxes(title_text='Количество заведений с разбивкой по категориям')

fig.show()
```

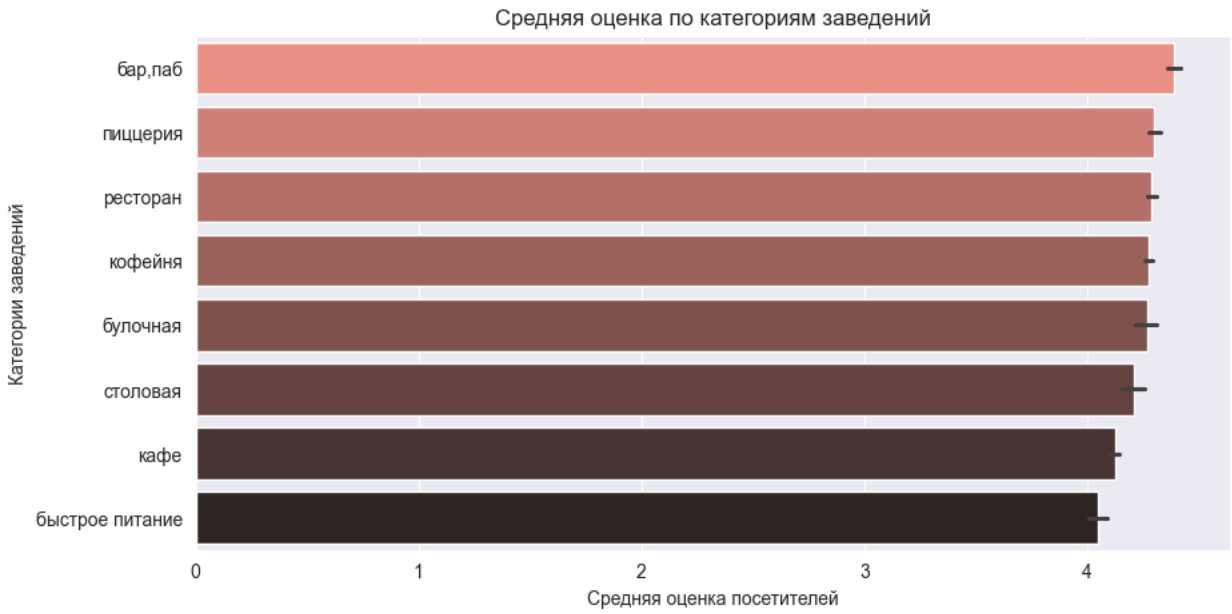


На графике показано количество разных заведений в разбивке по округам Москвы. Лидер по количеству заведений - ЦАО. Наименьшее количество заведений в СЗАО. В структуре заведений во всех округах явно выделяется "кафе", "ресторан" и "кофейня". В ЦАО отмечается большое количество "пабов-баров".

Средние рейтинги по категориям заведений

Оценим, как оценивают посетители в среднем заведения разных категорий

In [36]:
```
# строим график в снс
plt.figure(figsize=(10, 5))
ax=sns.barplot(
               y='category'
              ,x='rating'
              ,data=df
              ,estimator=mean
              ,palette='dark:salmon_r'
# сортировка оценок
              ,order=(df.groupby('category',as_index=False)
                      ['rating']
                          .agg('mean')
                          .sort_values(by='rating', ascending=False)
                      ['category']
                     )
              )
plt.title('Средняя оценка по категориям заведений')
plt.xlabel('Средняя оценка посетителей')
plt.ylabel('Категории заведений')
plt.show()
```



Необходимо отметить, что средняя (не медианная) оценка по всем типам довольно высокая - больше 4! Больше всех высоких оценок у баров-пабов. Наименьшие оценки - у заведений быстрого обслуживания (наверно, скорость не устраивает, таки).

Средний рейтинг заведений по округам на хороплете

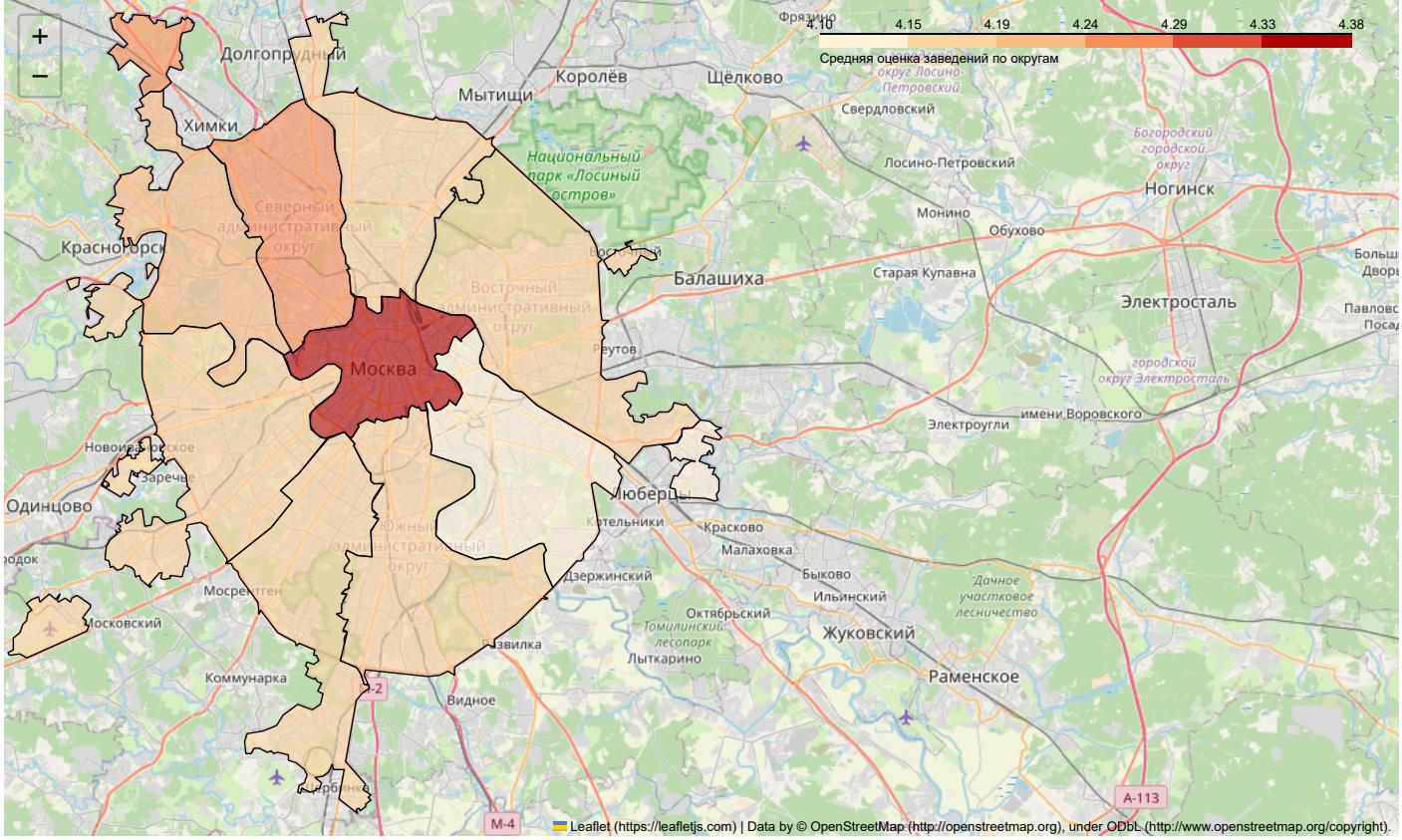
Построим фоновую картограмму со средним рейтингом заведений каждого района

In [37]:
```
# подготовка дф
df_distr_rating = (
                    df.groupby('district', as_index=False)
                      .agg({'rating':'mean'})
                   )
```

In [38]:
```
# загружаем JSON-файл с границами округов Москвы локально или с сервера
try:
    state_geo = 'admin_level_geomap_ansi.geojson'
except:
    state_geo = 'https://code.s3.yandex.net/data-analyst/admin_level_geomap.geojson'

# прицел на Москву
moscow_lat, moscow_lng = 55.751244, 37.618423
```

In [39]:
```
map_rating = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаём хороплет с рейтингами
Choropleth(
    geo_data=state_geo,
    data=df_distr_rating,
    columns=['district', 'rating'],
    key_on='feature.name',
    fill_color='OrRd',
    fill_opacity=0.6,
    legend_name='Средняя оценка заведений по округам',
).add_to(map_rating)

# выводим карту
map_rating
```

Out[39]:



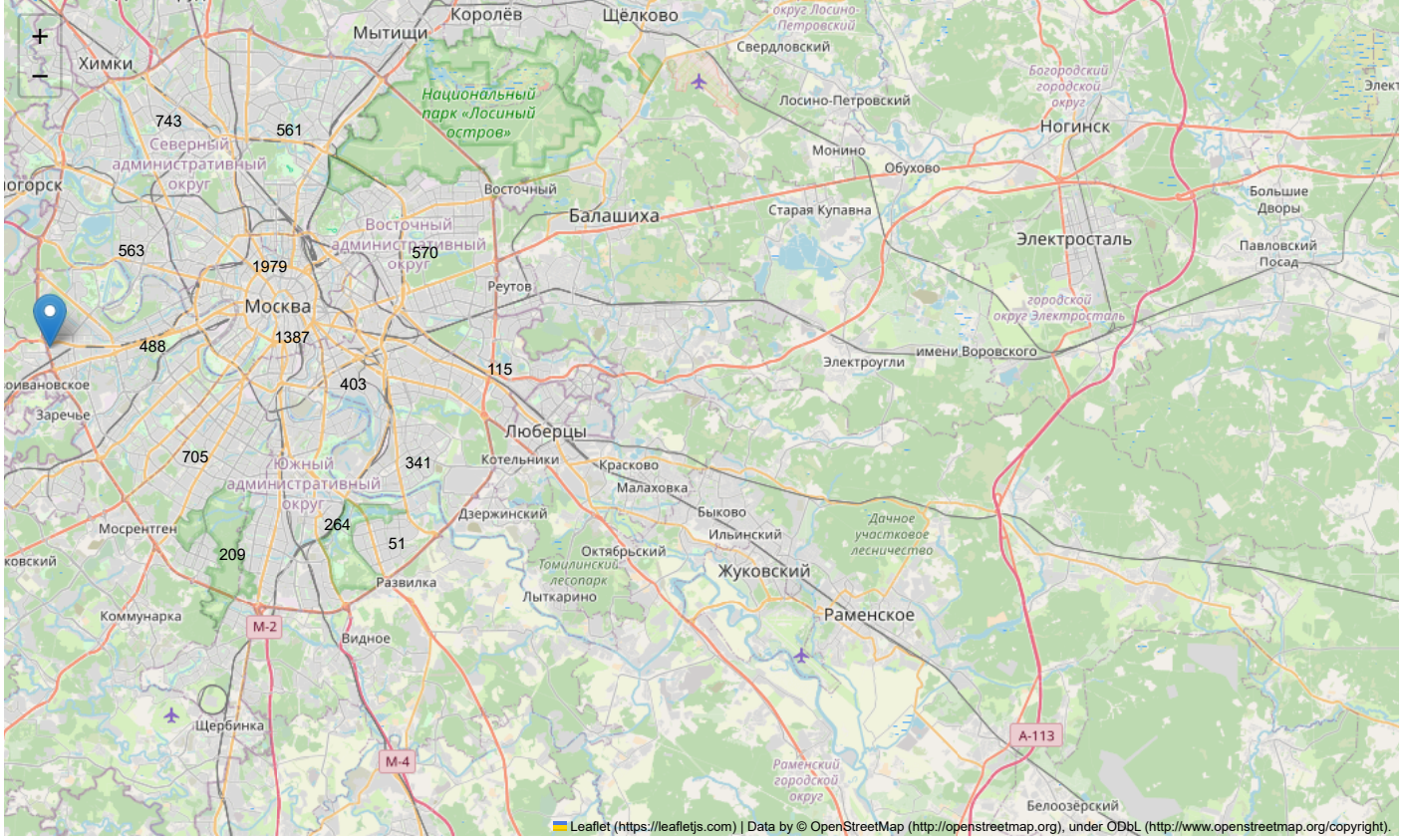
Самые высокие оценки посетителей у заведения ЦАО. Похоже, их вытягивают высокие оценки баров и пабов, которые в большом количестве разместились именно там.

Заведения на карте Москвы (кластеры)

In [40]:

```
# новая карта
all_places_map = Map(location=[moscow_lat, moscow_lng], zoom_start=10)
# создаём пустой кластер, добавляем его на карту
marker_cluster = MarkerCluster().add_to(all_places_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['category']}",
    ).add_to(marker_cluster)

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

# выводим карту
all_places_map
```

Out[40]:



Кажется, что общепита нет только на территориях леса и парков... Но и там, уверен, они есть!))

Топ-15 улиц по количеству заведений

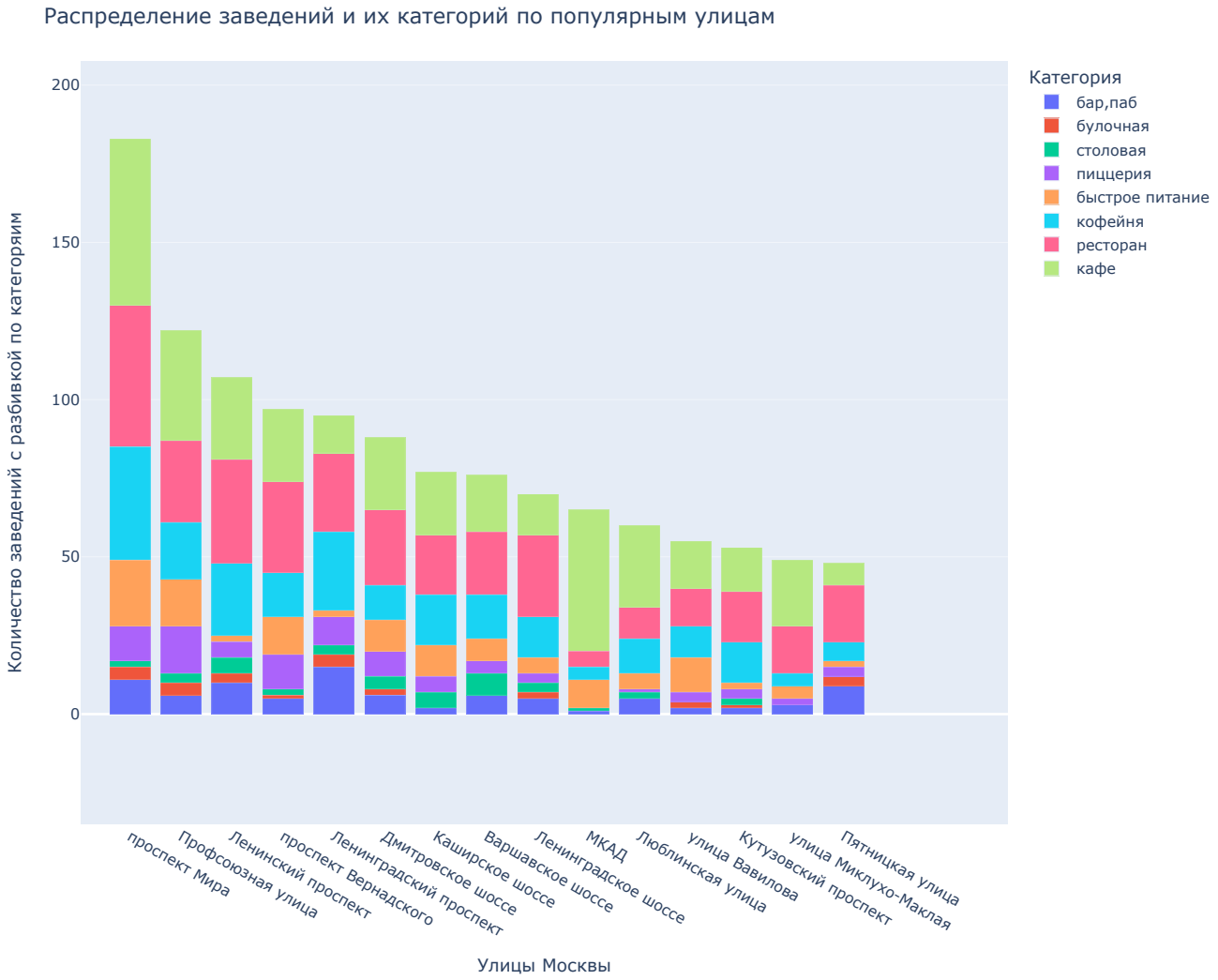
Посмотрим, какие улицы - самые "притягательные", чтобы поесть!

In [41]:
```
# создадим список из 15 улиц с наибольшим количеством заведений
street_top15 = (
                df.groupby('street', as_index=False)
                .agg({'name':'count'})
                .sort_values(by='name', ascending=False)
                .head(15)
                ['street']
                .to_list()
                )
```

Построим график распределения количества заведений и их категорий по этим улицам

In [42]:
```
# подготовка данных
df_15street=(
    df[df['street'].isin(street_top15)]
    [['street', 'category']]
    .value_counts()
    .reset_index(name='count')
    .sort_values(by='count', ascending=True)
)
df_15street.columns=['Улица', 'Категория', 'Количество']
```

In [43]:
```
# строим график
fig = px.bar(data_frame=df_15street
             , x='Улица'
             , y='Количество'
             , color="Категория"
             , barmode="stack")
fig.update_layout(title='Распределение заведений и их категорий по популярным улицам'
                  ,width=1000
                  ,height=800
                  )
fig.update_xaxes(title_text='Улицы Москвы', categoryorder='total descending')
fig.update_yaxes(title_text='Количество заведений с разбивкой по категориям')

fig.show()
```



В номинации "Вкусная улица №1" побеждает проспект Мира! Ура! Более 180 заведений: преобладают кафе, рестораны и кофейни.
Очевидно, что все улицы из топ-15 - центральные и довольно длинные (сами названия: проспект, шоссе)
Из всего списка в глаза бросается МКАД, в структуре которого сильно преобладают кафе (2/3 от общего числа)

Улицы с единственным заведением общепита

Поищем улицы, на которых представлены по одному заведению общепита

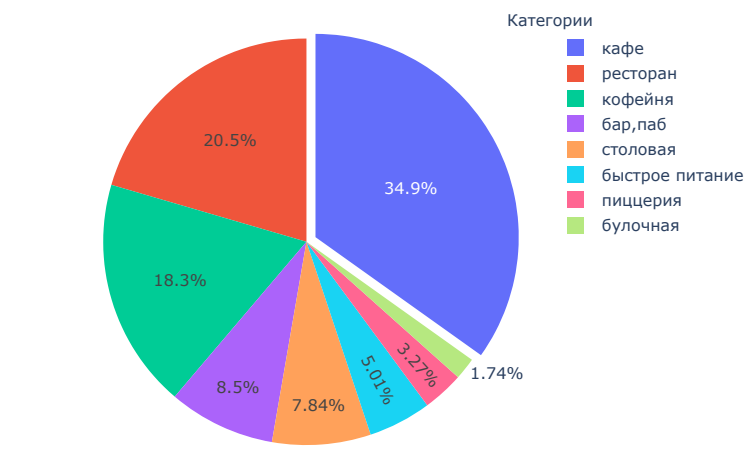
In [44]:
```
street_solo = (
               df.groupby('street', as_index=False)
               .agg({'name':'count'})
               .query('name == 1')['street']
               .to_list()
               )
print('Количество улиц с единственным заведением общепита в Москве: {}'
      .format(len(street_solo)))
print('Общее количество улиц в Москве, представленных в ДС: {}'
      .format(len(df['street'].unique())))
```

```
Количество улиц с единственным заведением общепита в Москве: 459
Общее количество улиц в Москве, представленных в ДС: 1448
```

Таким образом, на 30% улиц из представленных в исходных данных расположены по одному заведению общественного питания. Посмотрим, что это за заведения:

In [45]:
```
# подготовим  данные по категориям заведений
df_solo = df[(df['street'].isin(street_solo))]
df_solo_share = (df_solo.groupby('category')['name']
                 .count()
                 .reset_index()
                 .sort_values(by='name', ascending=False))
```

In [46]:
```
# строим пай
fig = go.Figure(data=[go.Pie(labels=df_solo_share['category']
                             ,values=df_solo_share['name']
                             ,pull = [0.05, 0])])
fig.update_layout(title='Популярность категорий единственных на улице заведений'
                  ,width=600
                  ,height=500
                  ,annotations=[dict(x=1.12
                                     ,y=1.05
                                     ,text='Категории'
                                     ,showarrow=False)]
                 )
fig.show()
```



Из выборки можно сделать вывод, что как и везде, кафе являются самыми популярными заведениями, среди заведений, представленных на улице в единственном числе. Оценки примерно равны, среди остальных показателей, явно выделяющихся признаков также не отмечается.

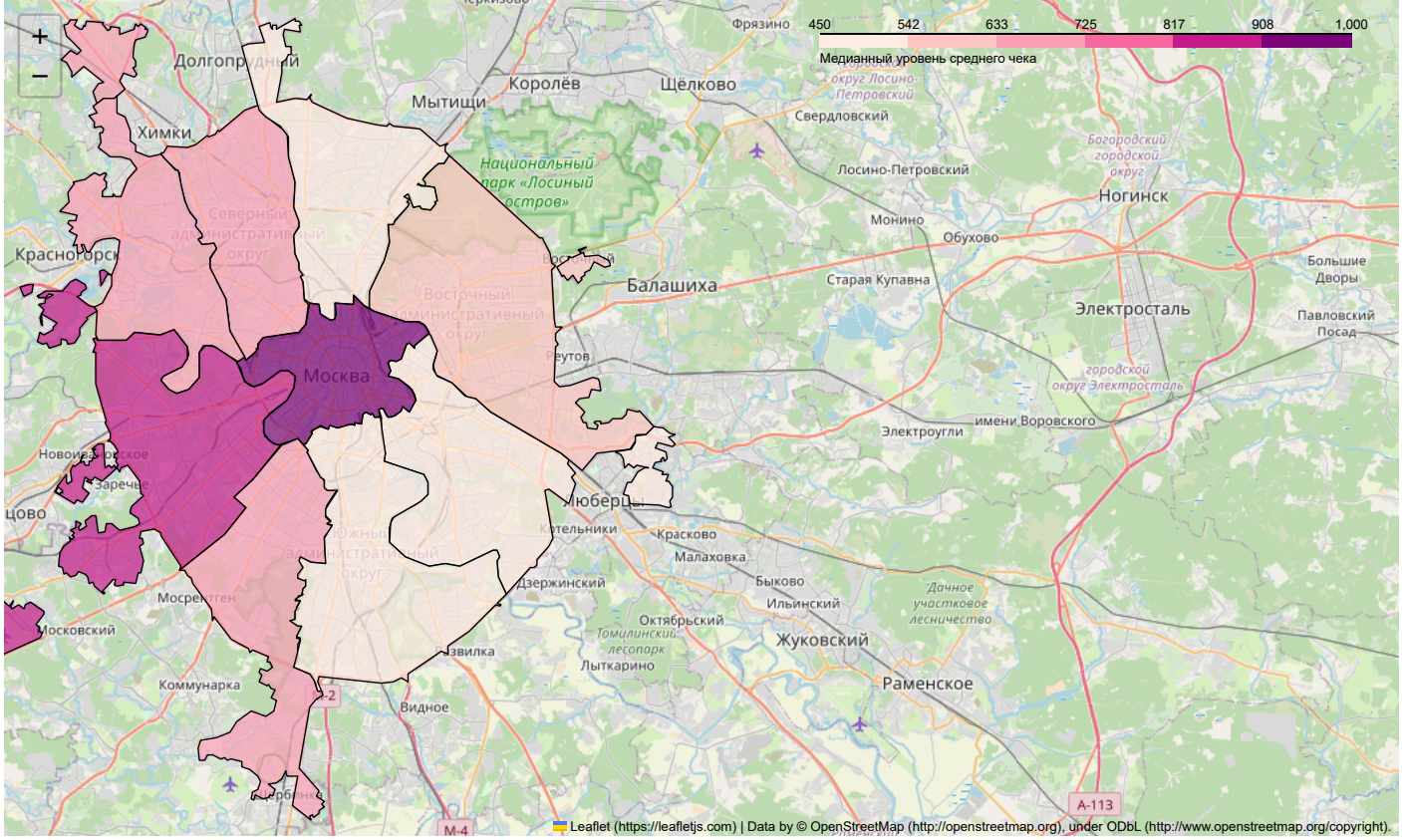
Анализ средних чеков заведений по округам

Несмотря на то, что в данных по среднему чеку достотачно много пропусков, попробуем сгруппировать имеющиеся данные по округам Москвы и выявить закономерности в данных

In [47]:
```
# подготовка данных
df_distr_price = (
                  df.groupby(['district', 'district_short'], as_index=False)
                    .agg({'middle_avg_bill':'median'})
                    .sort_values(by='middle_avg_bill', ascending=False)
                  )
```

In [48]:
```
# создаём карту
map_avg_bill = Map(location=[moscow_lat, moscow_lng], zoom_start=10)

# создаём хороплет с помощью Choropleth и добавляем его на карту
Choropleth(
    geo_data=state_geo,
    data=df_distr_price,
    columns=['district', 'middle_avg_bill'],
    key_on='feature.name',
    fill_color='RdPu',
    fill_opacity=0.7,
    legend_name='Медианный уровень среднего чека',
).add_to(map_avg_bill)

# выводим карту
map_avg_bill
```

Out[48]:

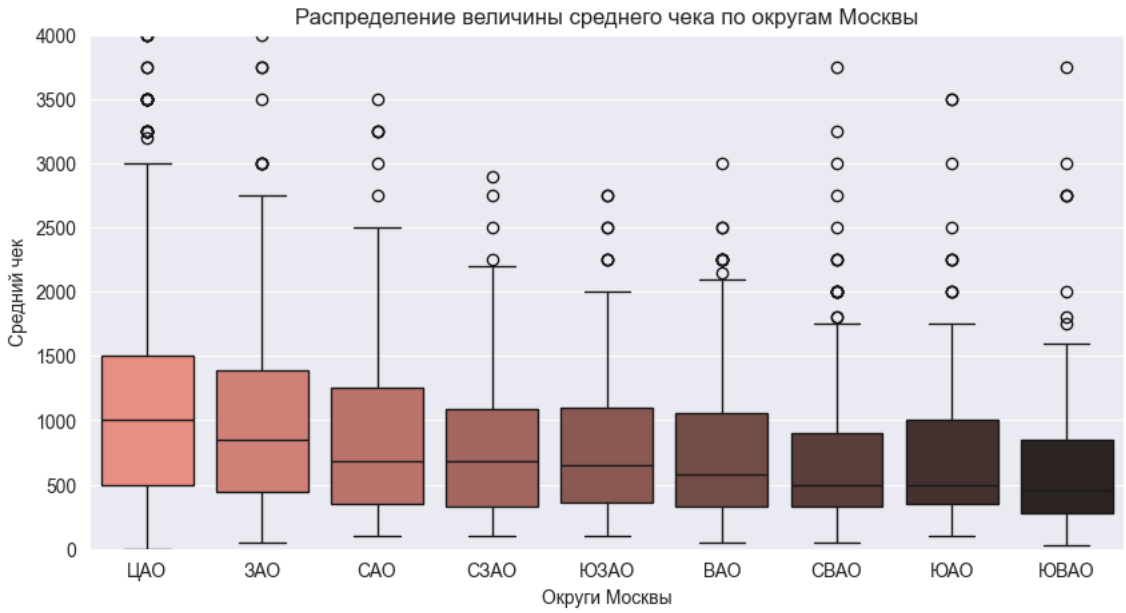


Самая высокая медиана среднего чека ожидаемо - в ЦАО. Самые низкие - в ЮВАО.

Проанализируем уровень цен по округам Москвы.

In [49]:

```
plt.figure(figsize=(10, 5))
sns.boxplot(
     x='district_short'
    ,y='middle_avg_bill'
    ,data=df
    ,palette='dark:salmon_r'
    ,order=df_distr_price['district_short']
    )
plt.title('Распределение величины среднего чека по округам Москвы')
plt.xlabel('Округи Москвы')
plt.ylabel('Средний чек')
plt.ylim(0,4000)
plt.show()
```



Боксплоты наглядно подтверждают вывод о величине среднего чека в разрезе округов: в ЦАО и медиана выше, и "разлёт" цен выше (95 перцентиль достигает 3000, тогда как в СВАО и ЮАО не достигает и 2000). Очевидно, что по мере приближения к центру уровень цен растёт, на окраинах - цены минимальны. Высокими ценами также выделяется элитный ЗАО.

Выводы

В ходе анализа имеющейся информации по заведениям общепита Москвы можно сделать несколько выводов:

- общее количество заведений в базе превышает 8400
- из них порядка 9% круглосуточные;
- около 33% из всех заведений - кафе, 20% - рестораны, 15% - кофейни, остальные категорри менее представительны;

- количество мест варьируется от 0 до 600-1000 (смотря что взять за границу), но большинство располагают от 40 до 140 мест
- 2/3 всех заведений не относится к сетям, сетевыми же чаще являются: булочные, пиццерии и кофейни
- самые популярные сети: Шоколадница, Домино'с пицца и Додо пицца
- среди округов Москвы самый "насыщенный" заведениями питания - ЦАО, он же лидер по количеству самых высоких оценок, высоких цен и количеству пабов среди округов
- самые насыщенные заведениями улицы - пр-т Мира, Профсоюзная ул. и Ленинский пр-т. Очевидно, что все топовые улицы - в центре столицы
- структура заведений, которые представлены в единственном числе на какой-либо улице особо ничем не выделяется от общей структуры заведений
- по оценке потребителей:
    1. низкому уровню цен соответствует заведение со средним чеком от 90 до 680р (чашка кофе - от 60 до 256р),
    2. среднему уровню соответствует вилка чека от 165 до 2150 (чашка кофе - от 60 до 1568р)
    3. выше среднего - средний чек 375 - 4500 (чашка кофе 150 - 277р)
    4. высокий уровень цен - средний чек 0 - 35000 (чашка кофе от 250 до 290)

Детализируем исследование: открытие кофейни

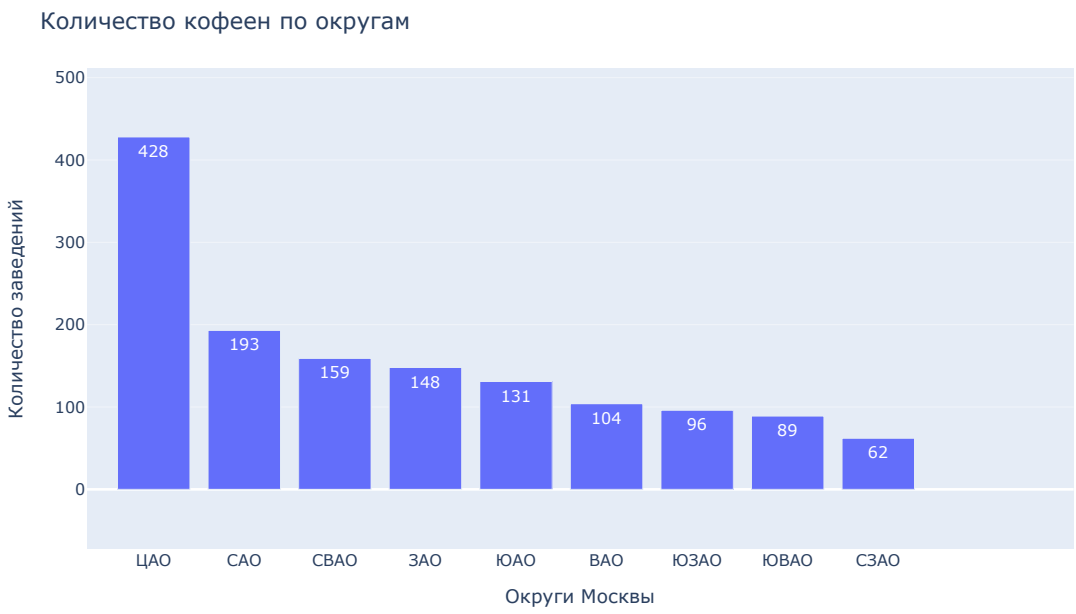
Количество в ДС и распределение кофеен по округам Москвы

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

In [50]:
```
print('Всего в датасете {}  кофеен'
      .format(df.query('category == "кофейня"').shape[0]))
print('Что составляет {:.2%} от общего числа заведений'
      .format(df.query('category == "кофейня"').shape[0] / df.shape[0]))
```

```
Всего в датасете 1410  кофеен
Что составляет 16.83% от общего числа заведений
```

In [51]:
```
fig = px.bar(data_frame=df_distr.query('Категория == "кофейня"')
             ,x='Округ'
             ,y='Количество'
             ,text_auto=True
            )
fig.update_layout(title='Количество кофеен по округам'
                  ,width=900
                  ,height=500
                  )
fig.update_xaxes(title_text='Округи Москвы', categoryorder='total descending')
fig.update_yaxes(title_text='Количество заведений')
fig.show()
```



Кофейни расположены преимущественно в "богатом" ЦАО, по мере отдаления от центра - уменьшается и количество. Для кофейни важно наличие офисного или студенческого контингента. Работники пром.предприятий в такие заведения редко наведываются.

Круглосуточные кофейни?

In [52]:
```
# строим пай по круглосуточным кофейням
fig = go.Figure(data=[go.Pie(labels=['Да', 'Нет']
                             ,values=(df.query('category == "кофейня"')
                                        .groupby('is_24/7')
                                      ['name'].count().sort_values()
                                      )
                             ,pull = [0.05, 0])])
fig.update_layout(title='Доля кругосуточных и некруглосуточных кофеен'
                  ,width=600
                  ,height=500
                  ,annotations=[dict(x=1.12
                                     ,y=1.05
                                     ,text='24/7 ?'
                                     ,showarrow=False)]
```

```
            )
fig.show()
```

Очевидно, что количество и, соответственно, популярность круглосуточных кофеен невелика! Это вообще "на любителя" - пить условное кофе ночью!

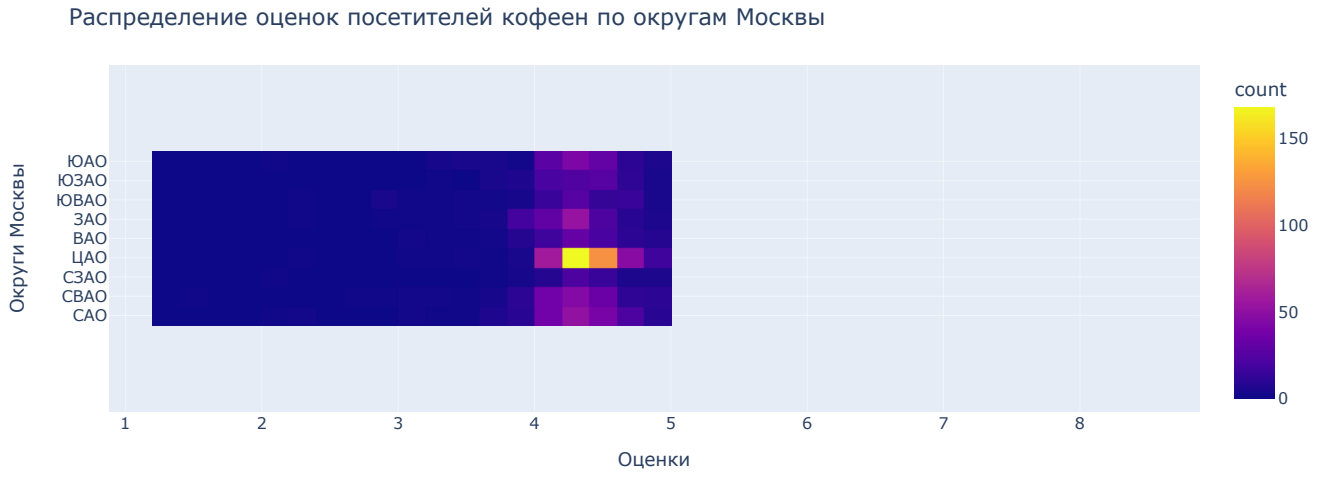
Рейтинги у кофеен. Распределение рейтингов кофеен по районам

In [53]:
```
plt.figure(figsize=(10, 5))
sns.histplot(
    df.query('category == "кофейня"')['rating']
    ,bins=35
    ,palette='dark:salmon_r'
    ,kde=True
           )
plt.title('Распределение оценок кофеен посетителями')
plt.xlabel('Оценка посетителей')
plt.ylabel('Количество оценок')
plt.show()
```

Как видно, кофейни, как правило, получают высокую оценку - от 4 до 4,5. Оценки ниже 4 почти не встречаются.

Для анализа оценок посетителей кофеен в разбивке по округам построим тепловую карту (её в нашем "зоопарке картинок" ещё не было).

In [54]:
```
# хитмэп от px
fig = px.density_heatmap(
                            df.query('category == "кофейня"')
                                ,x='rating'
                                ,y='district_short'
#                                 ,text_auto=True
                        )
fig.update_layout(title='Распределение оценок посетителей кофеен по округам Москвы'
                  ,width=1000
                  ,height=400
                  )
fig.update_xaxes(title_text='Оценки', range=(1,5))
fig.update_yaxes(title_text='Округи Москвы')
fig.show()
```

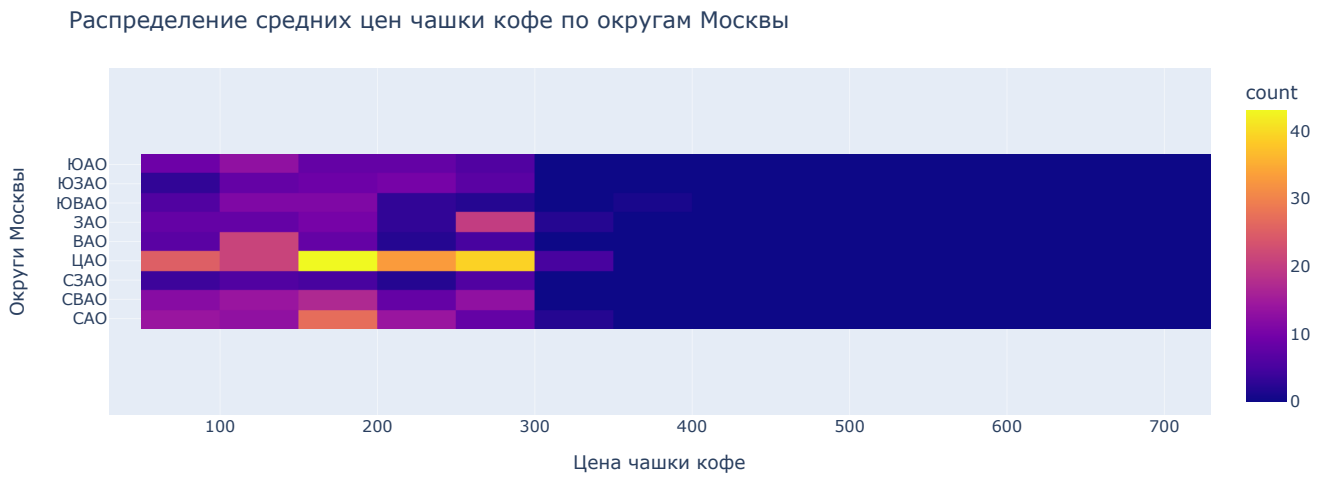


Тепловая карта также показывает, что основные оценки в диапазоне 4 - 4.5
Наибольшие оценки кофейни получают в ЦАО, холодней всего отношение к кофейням - в СЗАО.

Стоимость чашки капучино - ориентир при открытии

Стоимость чашки капучино, также как и средний чек, зависит от района работы предполагаемой кофейни.
Для наглядности не откажу себе влепить ещё одну тепловую карту и график распределения цены чашки кофе от количества посадочных мест: возможно, небольшие уютные места с харизмой могут позволить себе более высокий прайс?

In [55]:
```
fig = px.density_heatmap(
                            df.query('category == "кофейня"')
                                ,x='middle_coffee_cup'
                                ,y="district_short"
#                                ,text_auto=True
                        )
fig.update_layout(title='Распределение средних цен чашки кофе по округам Москвы'
                  ,width=1000
                  ,height=400
                 )
fig.update_xaxes(title_text='Цена чашки кофе', range=(0,350))
fig.update_yaxes(title_text='Округи Москвы')
fig.show()
```



Здесь картина не удивляет - ЦАО вновь лидер по цене: наиболее тёплая часть карты смещена вправо, в сторону максимальных цен, в то время, как в остальных округах основное (тепленькое) предложение "левее", т.е. цены ниже.
Интересен провал цен в диапазоне 200-250р за чашку: ниже много предложения и выше - тоже. Вероятно, это пограничная цена для разных "ценовых уровней" заведений, которые есть в разных округах.

In [56]:
```
# попробуем оценить зависимость цен на кофе от количества мест
sns.relplot(
    data=df.query('category == "кофейня"')
    ,x='seats'
    ,y='middle_coffee_cup'
    ,kind='scatter'
    ,palette='dark:salmon_r'
    ,height=4
)
plt.title('Зависимость цены чашки кофе от количества мест')
plt.xlabel('Количество мест')
plt.ylabel('Средний чек')
plt.ylim(50,400)
plt.show()
```

Гипотеза не подтвердилась - явной зависимости цен от количества мест не наблюдается. Цены варьируются во всем диапазоне (от 50 до 300р) при количестве мест от 0 (предположили, что это "кофе-он-гоу") до 150-200.

Таким образом, рынок кофеен отличается большой насыщенностью и конкуренцией. Для открытия нового заведения (сети) рекомендуется:

- выбрать локацию на одной из улиц из топ-15, т.к. они обеспечивают максимальный трафик целевой аудитории формата кофейни
- уровень цен необходимо выбирать, исходя из локации, для ЦАО это диапазон 150-300р за чашку. При этом необходимо учитывать высокую эластичность спроса по цене при такой высокой конкуренции: при отсутствии прочих преимуществ (изюминок) более высокая цена сильно сократит поток клиентов
- часы работы выбираются также, ориентируясь на основной трафик, режим кофейни 24/7, вероятней всего, не оправдан, особенно, если поблизости есть конкуренты, спрос - невелик.

Презентация: https://disk.yandex.ru/i/ttD5JI4DDprGrQ