Исследования рынка общепита в Москве для принятия решения об открытии нового заведения

Февраль 2022г

Общие выводы по исследованию.

Доля кафе в заведениях общепита около 40%. Преобладают несетевые заведения. Среднее количество посадочных мест в кафе около 40. По сетевому формату открывают заведения категории "кафе" в 23% случаев. Наибольшее количество объектов питания расположены на главных улицах города.

Описание проекта

Выяснить текущее положение дел на рынке общественного питания в Москве. Исследовать вопрос: будет ли успешным и популярным на долгое время кафе, в котором гостей обслуживают роботы-официанты. По результатам анализа подготовить презентация для инвесторов с рекомендациями.

Инструменты и навыки

```
Python
Pandas
Seaborn
Plotly
визуализация данных
```

1 Загрузка данных и подготовка их к анализу

1.1 Загрузка данных

Загрузим данные о заведениях общественного питания Москвы. Убедимся, что тип данных в каждой колонке — правильный, а также отсутствуют пропущенные значения и дубликаты. При необходимости обработаем их. Путь к файлу: /datasets/rest_data.csv

In [1]:
```
#загружаем необходимые библиотеки
import json
import math
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import requests
import scipy.stats as stats
import seaborn as sns
import warnings
import plotly.express as px
from io import BytesIO
from scipy import stats as st
from pandas.io.json import json_normalize
from pprint import pprint
sns.set(rc={'figure.figsize':(10, 8)})
pd.options.display.float_format = '{:,.2f}'.format
%config InlineBackend.figure_format = 'retina'
def fxn():
    warnings.warn("deprecated", DeprecationWarning)
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    fxn()
pd.set_option('display.max_colwidth',1000)
```

In [2]:
```
#Загружаем данные:
df = pd.read_csv('/datasets/rest_data.csv')
```

- Описание данных
  - Таблица rest_data:
    - id — идентификатор объекта;
    - object_name — название объекта общественного питания;
    - chain — сетевой ресторан;
    - object_type — тип объекта общественного питания;
    - address — адрес;
    - number — количество посадочных мест.

1.2 Предобработка данных

In [3]:
```
#получаем информацию
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15366 entries, 0 to 15365
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   id           15366 non-null  int64 
 1   object_name  15366 non-null  object
 2   chain        15366 non-null  object
 3   object_type  15366 non-null  object
 4   address      15366 non-null  object
 5   number       15366 non-null  int64 
dtypes: int64(2), object(4)
memory usage: 720.4+ KB
```

In [4]:
```
df.columns# проверка результатов - перечень названий столбцов
```

Out[4]:
```
Index(['id', 'object_name', 'chain', 'object_type', 'address', 'number'], dtype='objec
t')
```

названия столбцов указаны корректно

In [5]:
```
#изменим названием столбца с посадочными местами  откорректируем тип данных
df['number'] = df['number'].astype('int16')
df = df.rename(columns={'number':'number_of_seats'})
```

In [6]:
```
# откроем таблицу
df.head()
```

Out[6]:

	id	object_name	chain	object_type	address	number_of_seats
0	151635	СМЕТАНА	нет	кафе	город Москва, улица Егора Абакумова, дом 9	48
1	77874	Родник	нет	кафе	город Москва, улица Талалихина, дом 2/1, корпус 1	35
2	24309	Кафе «Академия»	нет	кафе	город Москва, Абельмановская улица, дом 6	95
3	21894	ПИЦЦЕТОРИЯ	да	кафе	город Москва, Абрамцевская улица, дом 1	40
4	119365	Кафе «Вишневая метель»	нет	кафе	город Москва, Абрамцевская улица, дом 9, корпус 1	50

In [7]:
```
# есть проблемы с регистром названий - исправляем их
df['object_name'] = df['object_name'].str.lower()
df['object_type'] = df['object_type'].str.lower()
df['address'] = df['address'].str.lower()
```

In [8]:
```
# проверим
df.head()
```

Out[8]:

	id	object_name	chain	object_type	address	number_of_seats
0	151635	сметана	нет	кафе	город москва, улица егора абакумова, дом 9	48
1	77874	родник	нет	кафе	город москва, улица талалихина, дом 2/1, корпус 1	35
2	24309	кафе «академия»	нет	кафе	город москва, абельмановская улица, дом 6	95
3	21894	пиццетория	да	кафе	город москва, абрамцевская улица, дом 1	40
4	119365	кафе «вишневая метель»	нет	кафе	город москва, абрамцевская улица, дом 9, корпус 1	50

In [9]:
```
# проверим на дубликаты
df.duplicated(subset=['object_name', 'address', 'chain', 'object_type', 'number_of_seats
```

Out[9]:
```
85
```

In [10]:
```
df.groupby('object_name')['id']
df.head()
```

Out[10]:

	id	object_name	chain	object_type	address	number_of_seats
0	151635	сметана	нет	кафе	город москва, улица егора абакумова, дом 9	48
1	77874	родник	нет	кафе	город москва, улица талалихина, дом 2/1, корпус 1	35
2	24309	кафе «академия»	нет	кафе	город москва, абельмановская улица, дом 6	95
3	21894	пиццетория	да	кафе	город москва, абрамцевская улица, дом 1	40
4	119365	кафе «вишневая метель»	нет	кафе	город москва, абрамцевская улица, дом 9, корпус 1	50

Можно предположить, что по одному адресу не может быть заведений с одинаковыми названиями и разным числом посадочных мест, поэтому среди таких записей можно оставить запись с максимальным id, как самую актуальную (такое предположение может быть неверным, но идея с тем, что id хронологически увеличиваются полезна).

Проверили на дубликаты. Дубликатов обнаружено 85. Исключим их.

In [11]:
```
# удаляем дубликаты
df.drop_duplicates(subset=['object_name', 'address', 'chain', 'object_type', 'number_of_
```

In [12]:
```
#проверяем
df.duplicated(subset=['object_name', 'address', 'chain', 'object_type', 'number_of_seats
```

Out[12]:
```
0
```

In [13]:
```
# проверим на пропуски
df.isna().sum()
```

Out[13]:
```
id                0
object_name       0
chain             0
object_type       0
address           0
number_of_seats   0
dtype: int64
```

пропусков не обнаружено

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15281 entries, 0 to 15365
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   id               15281 non-null  int64 
 1   object_name      15281 non-null  object
 2   chain            15281 non-null  object
 3   object_type      15281 non-null  object
 4   address          15281 non-null  object
 5   number_of_seats  15281 non-null  int16 
dtypes: int16(1), int64(1), object(4)
memory usage: 746.1+ KB
```

In [15]:
```
synonyms = {'mcdonalds': ['МАКДОНАЛДС', "Ресторан 'Макдоналдс'", "Макдоналдс"] , "kfc"
def check_synonyms(cell):
    for name,syn in synonyms.items():
        if cell in syn: return name
    return cell
df['object_name_new'] = df['object_name'].apply(check_synonyms)
df[['object_name','object_name_new']].query('object_name_new == "kfc"').head()
```

Out[15]:

	object_name	object_name_new
142	kfc	kfc
301	kfc	kfc
339	kfc	kfc
726	kfc	kfc
853	kfc	kfc

In [16]:
```
df[['object_name','object_name_new']].query('object_name_new == "mcdonalds"').head()
```

Out[16]:

	object_name	object_name_new
14147	mcdonalds	mcdonalds

1.2.1 Вывод.

В процессе предобработки: привел к нижнему регистру текстовые данные в столбцах, переименовал столбец с количеством посадочных мест, проверил на дубли и отсутствующие значения. Тип данных в каждой колонке — правильный Дубликаты устранены. Пропущенных значений нет.

Тут также можно обратить внимание на то, что названия заведений написаны вразнобой. Можно попробовать привести всё к одному виду (это также будет полезно, т.к. далее надо группировать заведения по сетям), типа макдональс, макдоналдс, mcdonalds → макдоналдс.

2 Анализ данных

2.1 Исследование соотношения видов объектов общественного питания по количеству.

In [17]:
```
# виды объектов общественного питания по количеству.
df.groupby('object_type')['object_name'].count().sort_values()
```

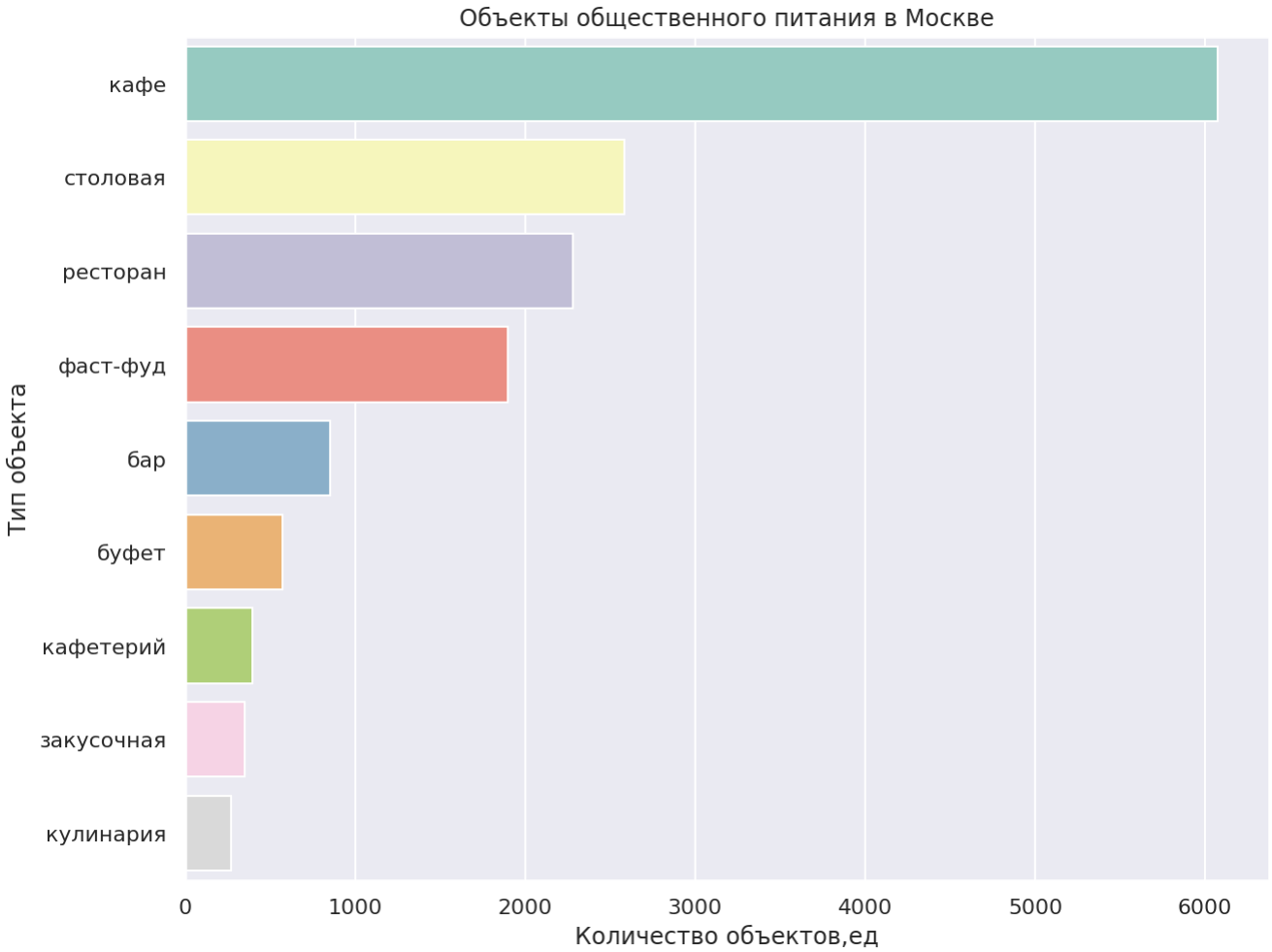
Out[17]:
```
object_type
магазин (отдел кулинарии)             273
закусочная                             348
кафетерий                              395
буфет                                  576
бар                                    855
предприятие быстрого обслуживания     1897
ресторан                              2282
столовая                              2584
кафе                                  6071
Name: object_name, dtype: int64
```

In [18]:
```
#приведем длинные названия типов объектов к более понятным
df['object_type'] = df['object_type'].str.replace('предприятие быстрого обслуживания',
df['object_type'] = df['object_type'].str.replace('магазин \(отдел кулинарии\)', 'кулина
```

```
/tmp/ipykernel_724/2991126020.py:3: FutureWarning: The default value of regex will cha
nge from True to False in a future version.
  df['object_type'] = df['object_type'].str.replace('магазин \(отдел кулинарии\)', 'ку
линария')
```

2.2 Построим график

In [19]:
```
# график распределения объектов общественного питания по типам в Москве
temp = df.groupby('object_type').count().reset_index()
ax = sns.barplot(x='number_of_seats', y='object_type', data=temp.sort_values('number_of_
ax.set_title('Объекты общественного питания в Москве')
ax.set_xlabel('Количество объектов,ед')
ax.set_ylabel('Тип объекта')
plt.show()
```



Всего зарегистрировано 15281 заведений.Подавляющее большинство заведений - 'кафе' (6071). Затем идут 'столовые'(2584), 'рестораны'(2282) и 'фаст-фуд'(1897).Доля остальных заведений незначительна.

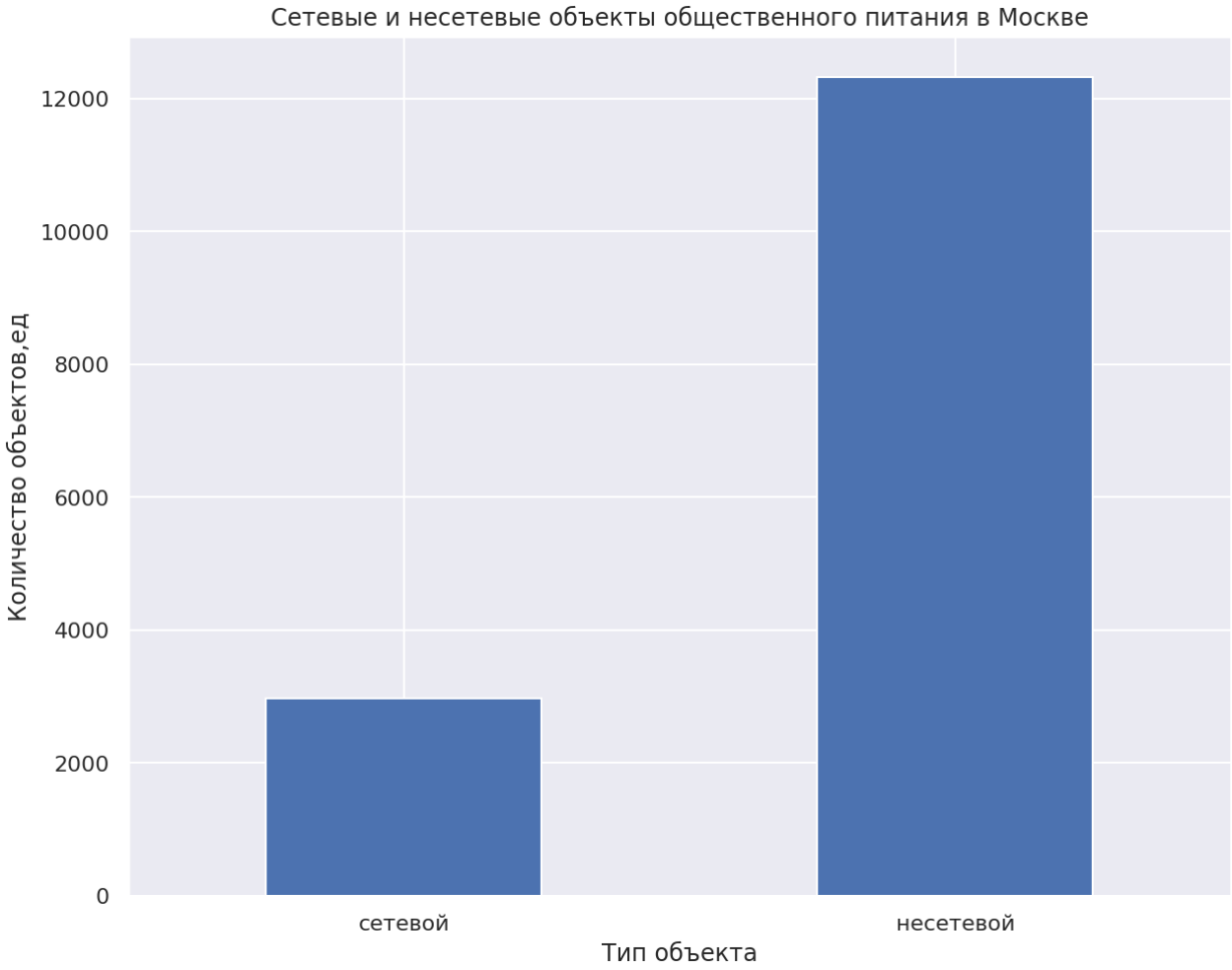
2.3 Соотношение сетевых и несетевых заведений по количеству

In [20]:
```
# Соотношение сетевых и несетевых заведений по количеству
df['chain'] = df['chain'].map({'нет':'несетевой','да':'сетевой'})
hh =df.groupby('chain')['object_name'].count().sort_values()
hh
```

Out[20]:
```
chain
сетевой       2964
несетевой    12317
Name: object_name, dtype: int64
```

2.4 Строим график

In [21]:
```
# график соотношения сетевых и несетевых заведений по количеству
ax = df.groupby('chain')['object_name'].count().sort_values().plot(kind='bar')
ax.set_title('Сетевые и несетевые объекты общественного питания в Москве')
ax.set_xlabel('Тип объекта')
ax.set_ylabel('Количество объектов,ед')
plt.xticks(rotation=0)
plt.show()
```



Количество несетевых заведений более чем в 4 раза превышает количество сетевых: 12317 против 2964.

2.5 Определим,для какого вида объекта общественного питания характерно сетевое распространение.

In [22]:
```
# Создадим сводную таблицу
df_pivot = df.pivot_table(index='object_type', values='id', columns='chain', aggfunc='co
df_pivot['share_chain %'] = ((df_pivot['сетевой'] / (df_pivot['сетевой'] + df_pivot['нес
df_pivot.drop(df_pivot.columns[[1,2]], axis=1, inplace=True)
df_pivot = df_pivot.sort_values('share_chain %', ascending = False)
```

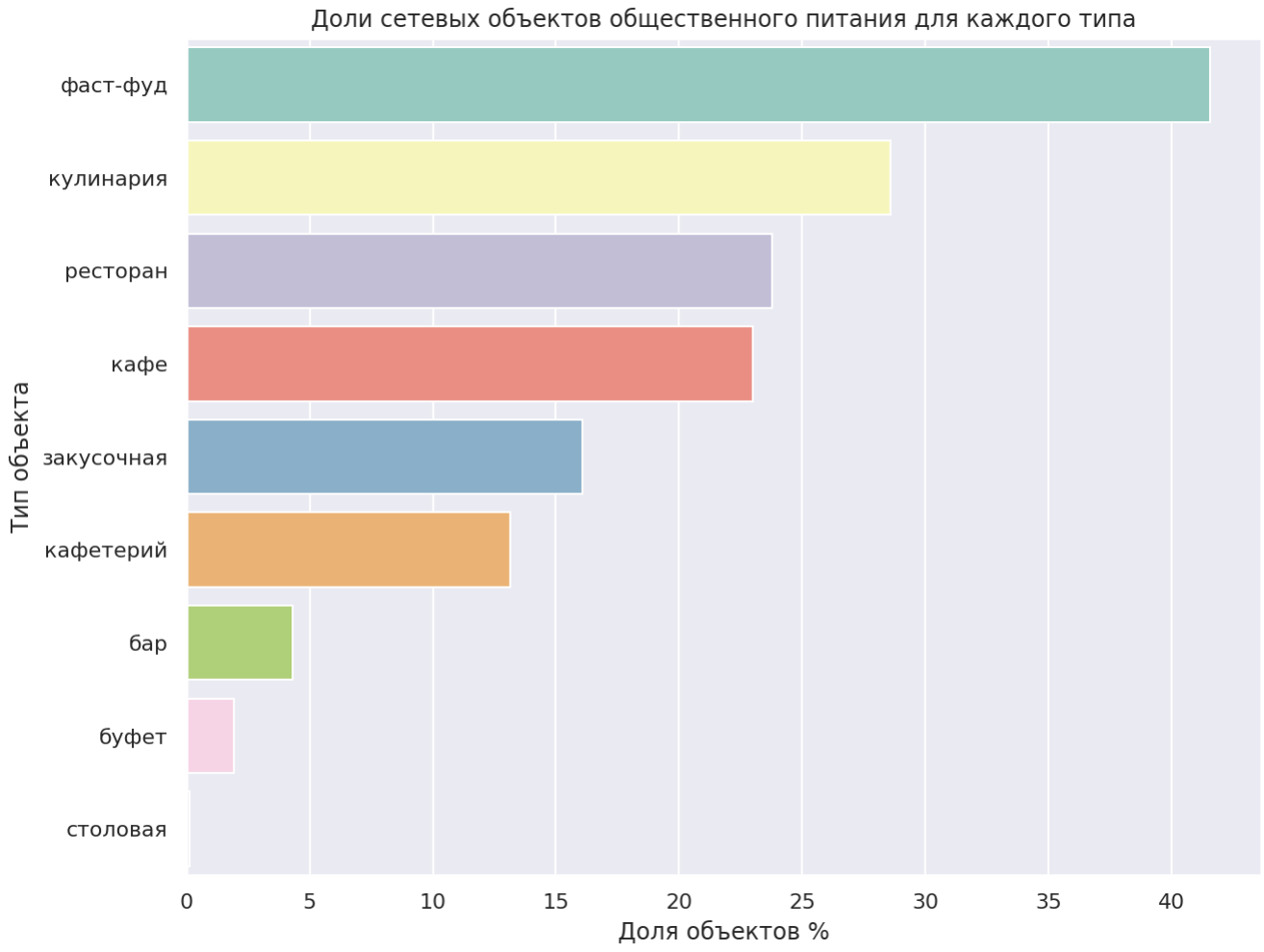
In [23]:
```
# открываем
df_pivot
```

Out[23]:

chain	object_type	share_chain %
8	фаст-фуд	41.54
5	кулинария	28.57
6	ресторан	23.79
3	кафе	22.99
2	закусочная	16.09
4	кафетерий	13.16
0	бар	4.33
1	буфет	1.91
7	столовая	0.12

2.6 Строим график

In [24]:
```
# график долей сетевых объектов общественного питания для каждого типа'
ax = sns.barplot(x='share_chain %', y='object_type', data=df_pivot.sort_values('share_ch
ax.set_title('Доли сетевых объектов общественного питания для каждого типа')
ax.set_xlabel('Доля объектов %')
ax.set_ylabel('Тип объекта')
plt.show()
```



У "фаст-фуда" доля сетевых заведений составляет более 40%.Среди сетевых лидируют

так же форматы объектов: кулинария, ресторан и кафе. Практически нет сетевых столовых, хотя в Москве они вторые по численности - более 2500.

In [25]:
```
df.groupby('object_type').apply(lambda x : pd.Series(

        {'сетевых' : (x['chain'] == 'сетевой').sum(),
         'несетевых' : (x['chain'] == 'несетевой').sum(),
         'доля сетевых' : (x['chain'] == 'сетевой').mean()
        }
    ))
```

Out[25]:

object_type	сетевых	несетевых	доля сетевых
бар	37.00	818.00	0.04
буфет	11.00	565.00	0.02
закусочная	56.00	292.00	0.16
кафе	1,396.00	4,675.00	0.23
кафетерий	52.00	343.00	0.13
кулинария	78.00	195.00	0.29
ресторан	543.00	1,739.00	0.24
столовая	3.00	2,581.00	0.00
фаст-фуд	788.00	1,109.00	0.42

2.7 Разберемся,что характерно для сетевых заведений: много заведений с небольшим числом посадочных мест в каждом или мало заведений с большим количеством посадочных мест

In [26]:
```
# определим среднее число мест для разных типов сетевых заведений
chain_object = df.query('chain =="сетевой"')
chain_group = (chain_object.groupby('object_name')
               .agg({'id': 'count', 'number_of_seats' : 'mean'})
               .reset_index()
               .rename(columns={'object_name':'название заведения','id':'число заведений
               .sort_values(by='число заведений', ascending=False))

chain_group.head(10)
```

Out[26]:

	название заведения	число заведений	среднее число мест
563	шоколадница	157	57.18
25	kfc	155	55.34
330	макдоналдс	150	87.70
109	бургер кинг	137	46.65
521	теремок	94	25.61
311	крошка картошка	90	21.86
159	домино'с пицца	90	18.34
339	милти	72	1.33
505	суши wok	72	6.71
367	папа джонс	51	22.04

In [27]:
```
#Разобьем все сети на 4 группы по признакам много/мало ресторанов, много/мало посадочных
chain_group.loc[chain_group['число заведений'] >= 50, 'много объектов'] = 'больше 50'
chain_group.loc[chain_group['число заведений'] < 50, 'мало объектов'] = 'меньше 50'
chain_group.loc[chain_group['среднее число мест'] < 40, 'мало мест'] = 'меньше 40'
chain_group.loc[chain_group['среднее число мест'] >= 40, 'много мест'] = 'больше 40'
chain_group.head(10)
```

Out[27]:

	название заведения	число заведений	среднее число мест	много объектов	мало объектов	мало мест	много мест
563	шоколадница	157	57.18	больше 50	NaN	NaN	больше 40
25	kfc	155	55.34	больше 50	NaN	NaN	больше 40
330	макдоналдс	150	87.70	больше 50	NaN	NaN	больше 40
109	бургер кинг	137	46.65	больше 50	NaN	NaN	больше 40
521	теремок	94	25.61	больше 50	NaN	меньше 40	NaN
311	крошка картошка	90	21.86	больше 50	NaN	меньше 40	NaN
159	домино'с пицца	90	18.34	больше 50	NaN	меньше 40	NaN
339	милти	72	1.33	больше 50	NaN	меньше 40	NaN
505	суши wok	72	6.71	больше 50	NaN	меньше 40	NaN
367	папа джонс	51	22.04	больше 50	NaN	меньше 40	NaN

2.8 Строим график

In [28]:
```
# график среднего числа мест для разных типов сетевых заведений
fig = px.scatter(chain_group, x="число заведений", y="среднее число мест",color="среднее
                 render_mode="webgl", width=800, height=600)

fig.show()
```

Результат неоднозначен.В формате "столовая" и "ресторан" характерно большое количество посадочных мест. Притом, что столовые занимают последнюю строчку по численности сетевых заведений - 3шт. "Рестораны" занимают третье место по количеству.Это объяснимо - столица, все-таки. Среди остальных сетевых заведений характерно большое количество точек с небольшим количеством посадочных мест.И есть группа сетевых заведений с числом посадочных мест менее 20.

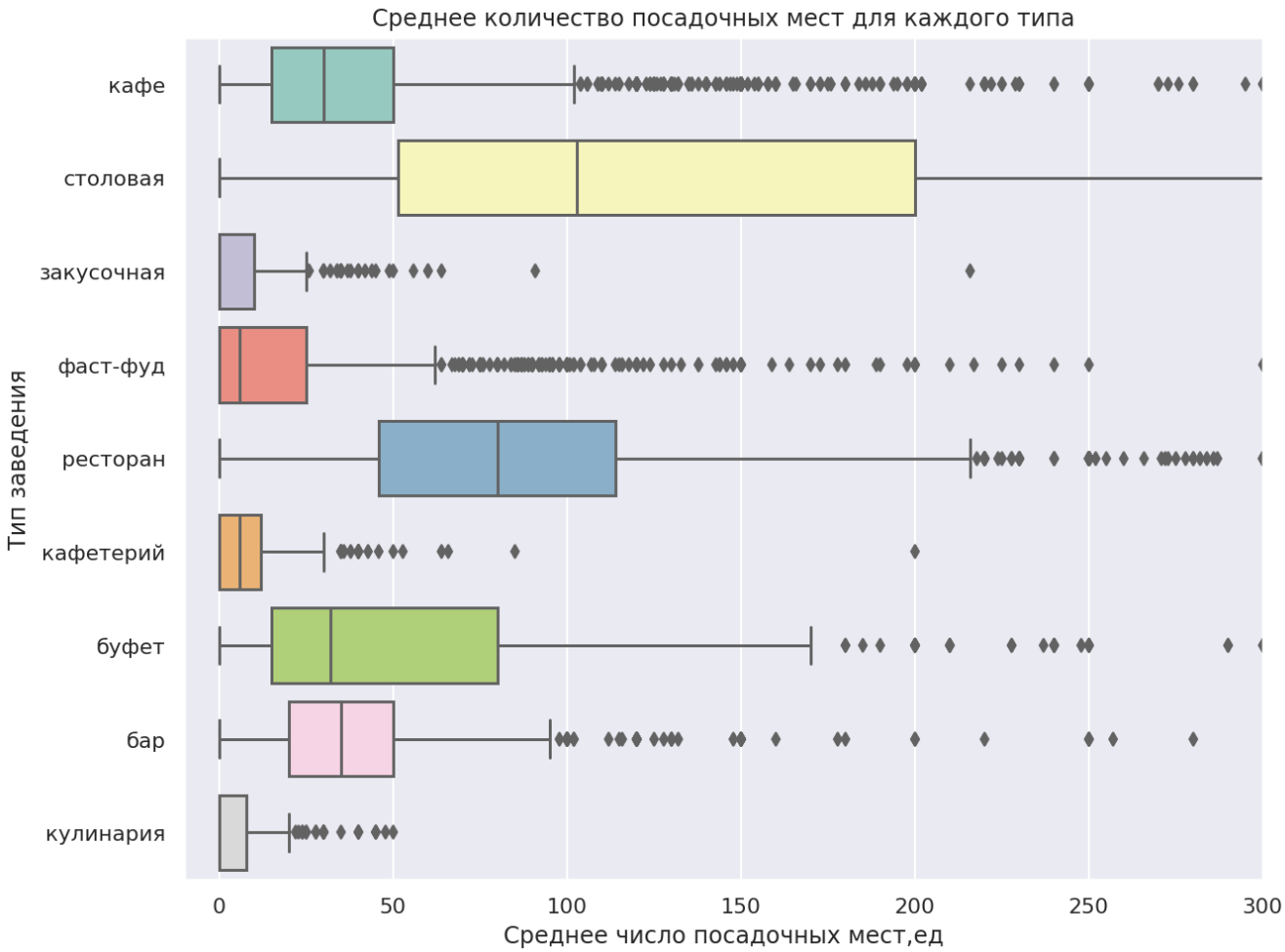
In [29]:
```
chain_group_copy = chain_group.loc[:,'название заведения' : 'среднее число мест'].copy()

#Разобьем все сети на 4 группы по признакам много/мало ресторанов, много/мало посадочных
chain_group_copy.loc[(chain_group['число заведений'] >= 50)
                     & (chain_group['среднее число мест'] < 40), 'категория'] = 'много з

chain_group_copy.loc[(chain_group['число заведений'] >= 50)
                     & (chain_group['среднее число мест'] >= 40), 'категория'] = 'много

chain_group_copy.loc[(chain_group['число заведений'] < 50)
                     & (chain_group['среднее число мест'] < 40), 'категория'] = 'мало за

chain_group_copy.loc[(chain_group['число заведений'] < 50)
                     & (chain_group['среднее число мест'] >= 40), 'категория'] = 'мало з


# график среднего числа мест для разных типов сетевых заведений
fig = px.scatter(chain_group_copy, x="число заведений", y="среднее число мест",color="ка
                 render_mode="webgl", width=800, height=600, hover_data=['название завед

fig.show()

chain_group_copy['категория'].value_counts()
```

Out[29]:
```
мало заведений - много мест     319
мало заведений - мало мест      248
много заведений - мало мест       6
```

```
много заведений - много мест      4
Name: категория, dtype: int64
```

2.9 Среднее количество посадочных мест для каждого вида объекта общественного питания

In [30]:
```
# строим диаграмму
ax = sns.boxplot(x='number_of_seats', y='object_type', data=df, palette="Set3")
ax.set_title('Среднее количество посадочных мест для каждого типа')
ax.set_xlabel('Среднее число посадочных мест,ед')
ax.set_ylabel('Тип заведения')
ax.set_xlim(-10,300)
plt.show()
```

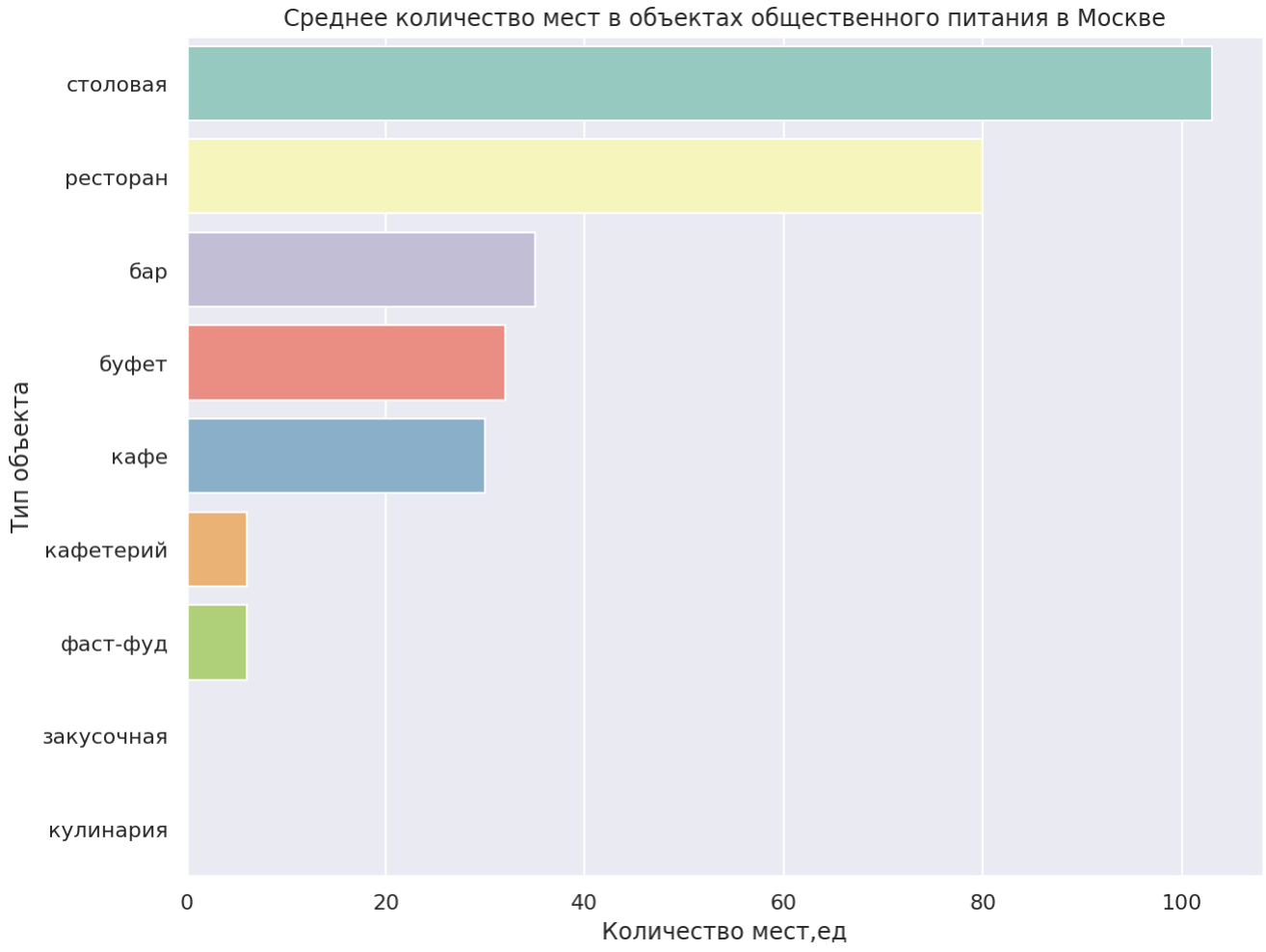


In [31]:
```
#получае таблицу
df.groupby('object_type')['number_of_seats'].describe().sort_values(by='50%')
```

Out[31]:

object_type	count	mean	std	min	25%	50%	75%	max
закусочная	348.00	7.85	16.88	0.00	0.00	0.00	10.00	216.00
кулинария	273.00	5.59	9.87	0.00	0.00	0.00	8.00	50.00
кафетерий	395.00	9.18	14.68	0.00	0.00	6.00	12.00	200.00
фаст-фуд	1,897.00	20.81	38.56	0.00	0.00	6.00	25.00	580.00
кафе	6,071.00	39.79	37.75	0.00	15.00	30.00	50.00	533.00
буфет	576.00	51.43	56.51	0.00	15.00	32.00	80.00	320.00
бар	855.00	43.53	67.11	0.00	20.00	35.00	50.00	1,700.00
ресторан	2,282.00	96.88	94.78	0.00	46.00	80.00	114.00	1,500.00
столовая	2,584.00	130.34	95.19	0.00	51.50	103.00	200.00	1,400.00

In [32]:
```
#график
temp = df.groupby('object_type').agg({'number_of_seats' : 'median'}).sort_values(by = 'n
ax = sns.barplot(x='number_of_seats', y='object_type', data=temp, palette="Set3")
ax.set_title('Среднее количество мест в объектах общественного питания в Москве')
ax.set_xlabel('Количество мест,ед')
ax.set_ylabel('Тип объекта')
plt.xticks(rotation=0)
plt.show()
```



В среднем самое большое количество посадочных мест представляет формат"столовая", немного меньше -"ресторан".Бары и буфеты и кафе имеют почти одинаковое количество посадочных мест - около 40.

2.10 Информация о расположении заведений

In [33]:
```
#проведем некоторые преобразования чтобы проще было выделить улицу
symbols = [',', '«', '»', '(', ')', '"', '  ']
for s in symbols:
    df['address'] = df['address'].str.replace(s, ' ')
df['address'] = df['address'].str.replace('ё','е')
```

```
/tmp/ipykernel_724/1213433674.py:4: FutureWarning:

The default value of regex will change from True to False in a future version. In addi
tion, single character regular expressions will*not* be treated as literal strings whe
n regex=True.
```

In [34]:
```
address = df['address'].to_list()
streets = []

for street in address:
    start = street.find('город москва')
    end = street.find('дом ')
    streets.append(street[start+12:end-1])

df['street_name'] = streets
df.head(10)
```

Out[34]:

	id	object_name	chain	object_type	address	number_of_seats	object_name_new	s
0	151635	сметана	несетевой	кафе	город москва улица егора абакумова дом 9	48	сметана	
1	77874	родник	несетевой	кафе	город москва улица талалихина дом 2/1 корпус 1	35	родник	
2	24309	кафе «академия»	несетевой	кафе	город москва абельмановская улица дом 6	95	кафе «академия»	абел
3	21894	пиццетория	сетевой	кафе	город москва абрамцевская улица дом 1	40	пиццетория	аб
4	119365	кафе «вишневая метель»	несетевой	кафе	город москва абрамцевская улица дом 9 корпус 1	50	кафе «вишневая метель»	аб
5	27429	стол. при гоу сош № 1051	несетевой	столовая	город москва абрамцевская улица дом 15 корпус 1	240	стол. при гоу сош № 1051	аб
6	148815	брусника	сетевой	кафе	город москва переулок сивцев вражек дом 6/2	10	брусника	сив
7	20957	буфет мтуси	несетевой	столовая	город москва авиамоторная улица дом 8 строение 1	90	буфет мтуси	ав
8	20958	кпф семья-1	несетевой	столовая	город москва авиамоторная улица дом 8 строение 1	150	кпф семья-1	ав
9	28858	столовая мтуси	несетевой	столовая	город москва авиамоторная улица дом 8 строение 1	120	столовая мтуси	ав

2.11 Топ-10 улиц по количеству объектов общественного питания.

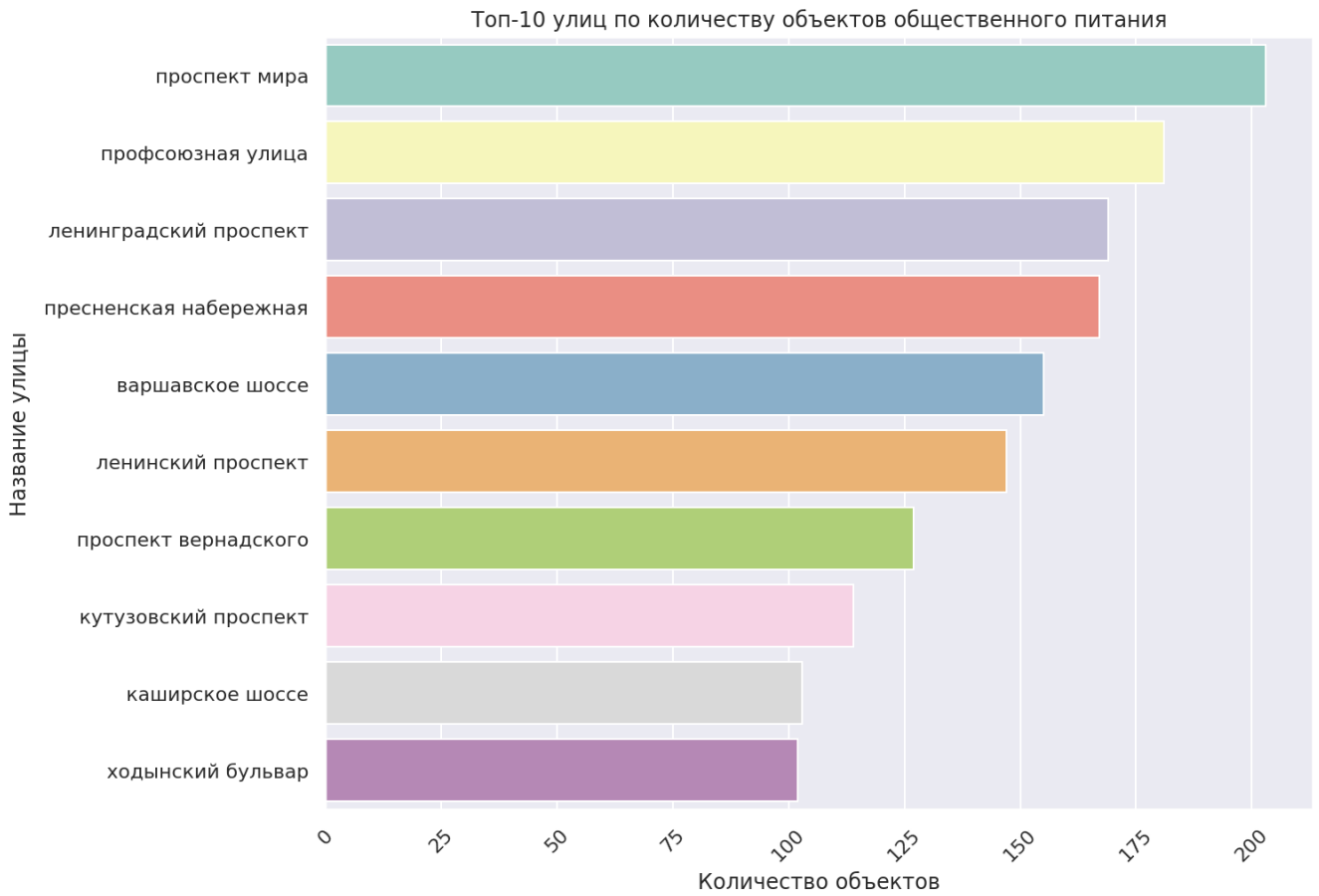
In [35]:
```
# найдем улицы с наибольшим числом объектов общепита
temp = df.groupby('street_name').count().sort_values(by='id', ascending=False).head(10)
top_10 = list(temp['street_name'])
temp[['street_name','id']]
```

Out[35]:

	street_name	id
0	проспект мира	203
1	профсоюзная улица	181
2	ленинградский проспект	169
3	пресненская набережная	167
4	варшавское шоссе	155
5	ленинский проспект	147
6	проспект вернадского	127
7	кутузовский проспект	114
8	каширское шоссе	103
9	ходынский бульвар	102

2.12 График топ-10 улиц по количеству объектов общественного питания.

In [36]:
```
#строим график
ax = sns.barplot(x='object_type', y='street_name', data=temp, palette="Set3")
ax.set_title('Топ-10 улиц по количеству объектов общественного питания')
ax.set_xlabel('Количество объектов')
ax.set_ylabel('Название улицы')
plt.xticks(rotation=45)
plt.show()
```



Наибольшее количество объектов питания расположены на главных артериях города: Ленинградском проспекте, Просоюзной улице и проспекте Мира.

Для того, чтобы узнать к каким районам принадлежат эти улицы, воспользуемся данными Мосгаза

In [37]:
```
# получаем необходимую информацию
spreadsheet_id = '1jB0T2q4XbQMUOSVmjCMS8cJDhBjdmhJcWcwsV_WcsQM'
file_name = 'https://docs.google.com/spreadsheets/d/{}/export?format=csv'.format(spreads
r = requests.get(file_name)
moscow_streets = pd.read_csv(BytesIO(r.content))
moscow_streets.head()
```

Out[37]:

	streetname	areaid	okrug	area
0	Выставочный переулок	17	ЦАО	Пресненский район
1	улица Гашека	17	ЦАО	Пресненский район
2	Большая Никитская улица	17	ЦАО	Пресненский район
3	Глубокий переулок	17	ЦАО	Пресненский район
4	Большой Гнездниковский переулок	17	ЦАО	Пресненский район

In [38]:
```
#переименуем столбцы
moscow_streets = moscow_streets.rename(columns={'streetname':'street_name'})
moscow_streets = moscow_streets.rename(columns={'areaid':'area_id'})
temp['street_name'] = temp['street_name'].str.strip()
```

In [39]:
```
# приведем названия улиц, округов и районов к нижнему регистру.
moscow_streets['okrug'] = moscow_streets['okrug'].str.lower()
moscow_streets['area'] = moscow_streets['area'].str.lower()
moscow_streets['street_name'] = moscow_streets['street_name'].str.lower()
```

In [40]:
```
#получаем таблицу
moscow_streets.head()
```

Out[40]:

	street_name	area_id	okrug	area
0	выставочный переулок	17	цао	пресненский район
1	улица гашека	17	цао	пресненский район
2	большая никитская улица	17	цао	пресненский район
3	глубокий переулок	17	цао	пресненский район
4	большой гнездниковский переулок	17	цао	пресненский район

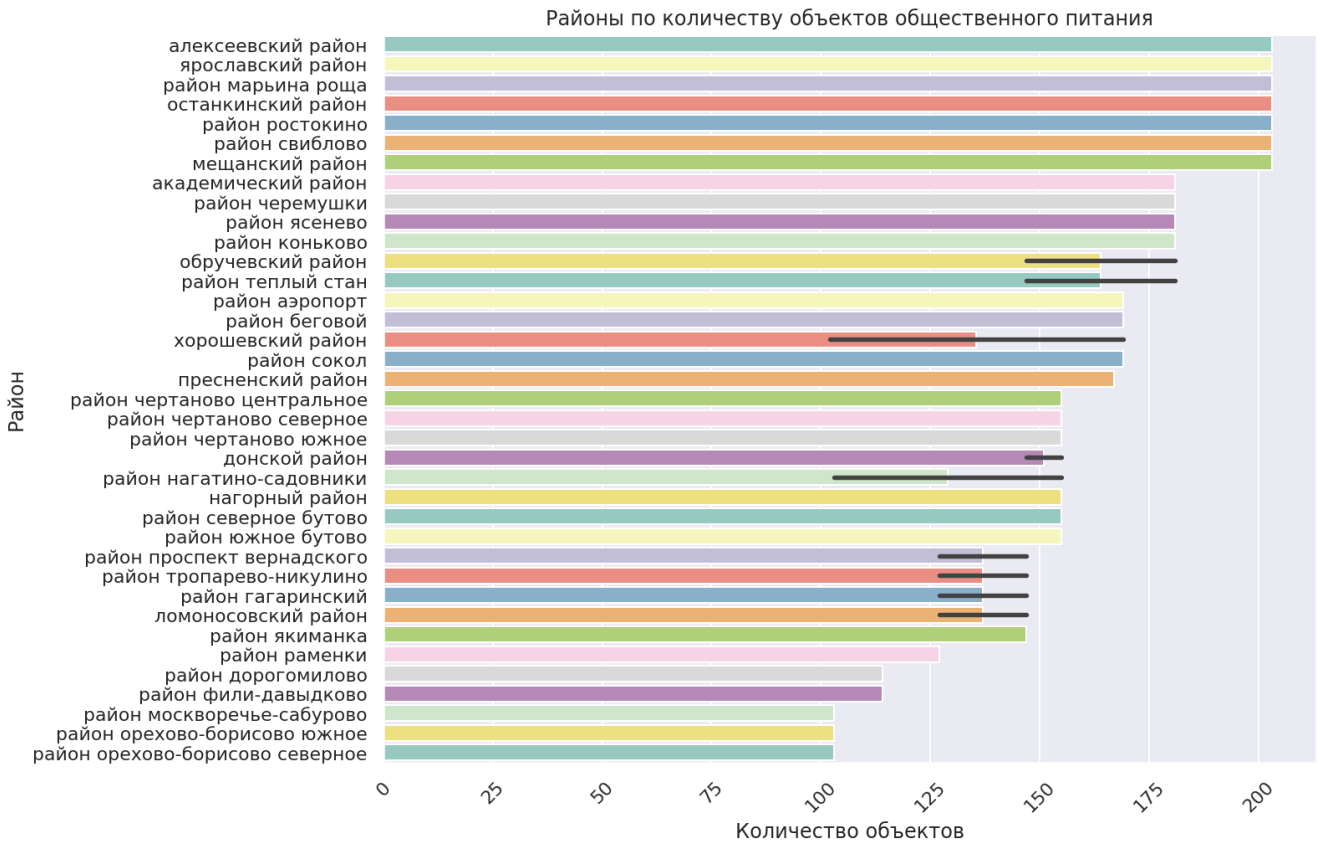
Методом merge объединим таблицы:

In [41]:
```
district = temp.merge(moscow_streets, on='street_name', how='left')
district = district.drop(['id','area_id', 'object_name', 'chain', 'address', 'number_of_
district = district[['street_name','okrug', 'area','object_type']]
district.head(10)
```

Out[41]:

	street_name	okrug	area	object_type
0	проспект мира	свао	алексеевский район	203
1	проспект мира	свао	ярославский район	203
2	проспект мира	свао	район марьина роща	203
3	проспект мира	свао	останкинский район	203
4	проспект мира	свао	район ростокино	203
5	проспект мира	свао	район свиблово	203
6	проспект мира	цао	мещанский район	203
7	профсоюзная улица	юзао	академический район	181
8	профсоюзная улица	юзао	район черемушки	181
9	профсоюзная улица	юзао	район ясенево	181

In [42]:
```
#найдем районы с наибольшим числом объектов общепита
ax = sns.barplot(x='object_type', y='area', data=district, palette="Set3")
ax.set_title('Районы по количеству объектов общественного питания')
ax.set_xlabel('Количество объектов')
ax.set_ylabel('Район')
plt.xticks(rotation=45)
plt.show()
```



Как видим, топ-10 улиц располагается в большом количестве районов и в разных частях города.

2.13 Число улиц с одним объектом общественного питания.

In [43]:
```
# Сгруппируем данные в датафрейм с 1-м объектом общественного питания
one = df.groupby('street_name').agg({'object_name': 'count'}).sort_values(by = 'object_
one = one.query('object_name == 1')
one = one.drop('object_name',1)
one = one.reset_index()
one.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 829 entries, 0 to 828
Data columns (total 1 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   street_name  829 non-null    object
dtypes: object(1)
memory usage: 6.6+ KB
```

829 улиц с одним объектом питания

In [44]:
```
#Добавим район и округ и удалим сразу дубликаты
one['street_name'] = one['street_name'].str.strip()
one = one.merge(moscow_streets, on='street_name', how='left')
one.head()
```

Out[44]:

	street_name	area_id	okrug	area
0	5-й проезд марьиной рощи	78.00	свао	район марьина роща
1	5-й рощинский проезд	98.00	юао	даниловский район
2	осенняя улица владение 2	NaN	NaN	NaN
3	улица перерва владение 3	NaN	NaN	NaN
4	улица паперника	120.00	ювао	рязанский район

In [45]:
```
area_nunique = one['area'].drop_duplicates()
area_nunique = area_nunique.dropna()
len(area_nunique)
```

Out[45]: 98

в 98 районах

In [46]:
```
# получаем список районов
area_nunique.head(10)
```

Out[46]:
```
0       район марьина роща
1        даниловский район
4          рязанский район
5          район хамовники
8        район богородское
9      район замоскворечье
10          тверской район
12         таганский район
15    район фили-давыдково
17         район лефортово
Name: area, dtype: object
```

In [53]:
```
# найдем, в каких районах больше улиц с 1 заведением
c = one.groupby('area').agg({'area_id': 'count'}).sort_values(by = 'area_id', ascending
c.head()
```

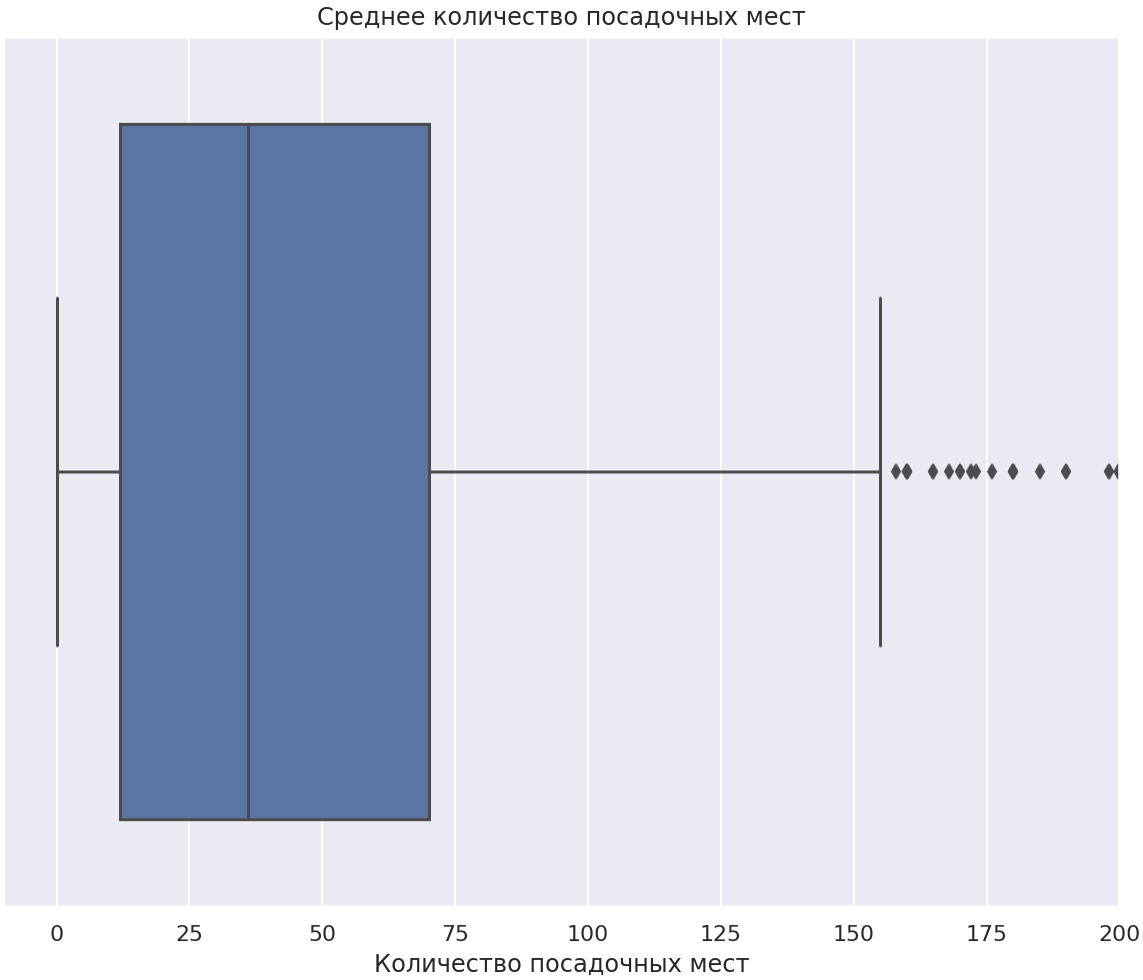
Out[53]:

area	area_id
таганский район	27
район хамовники	24
басманный район	21
тверской район	20
пресненский район	19

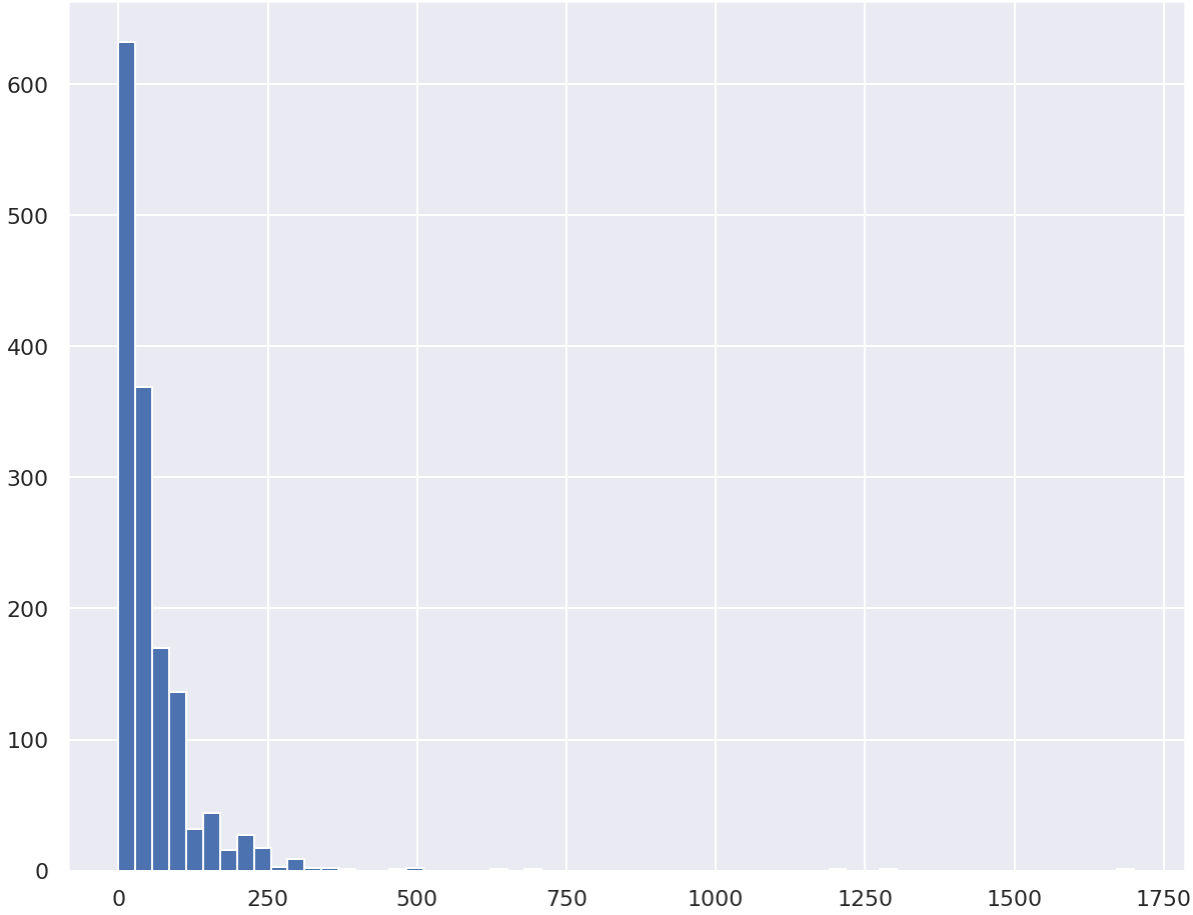
На основе полученных данных можно сделать вывод, что в Москве: 829 улиц с одним объектом общественного питания, которые находятся в 98 районах города.

2.14 Распределение количества посадочных мест для улиц с большим количеством объектов общественного питания

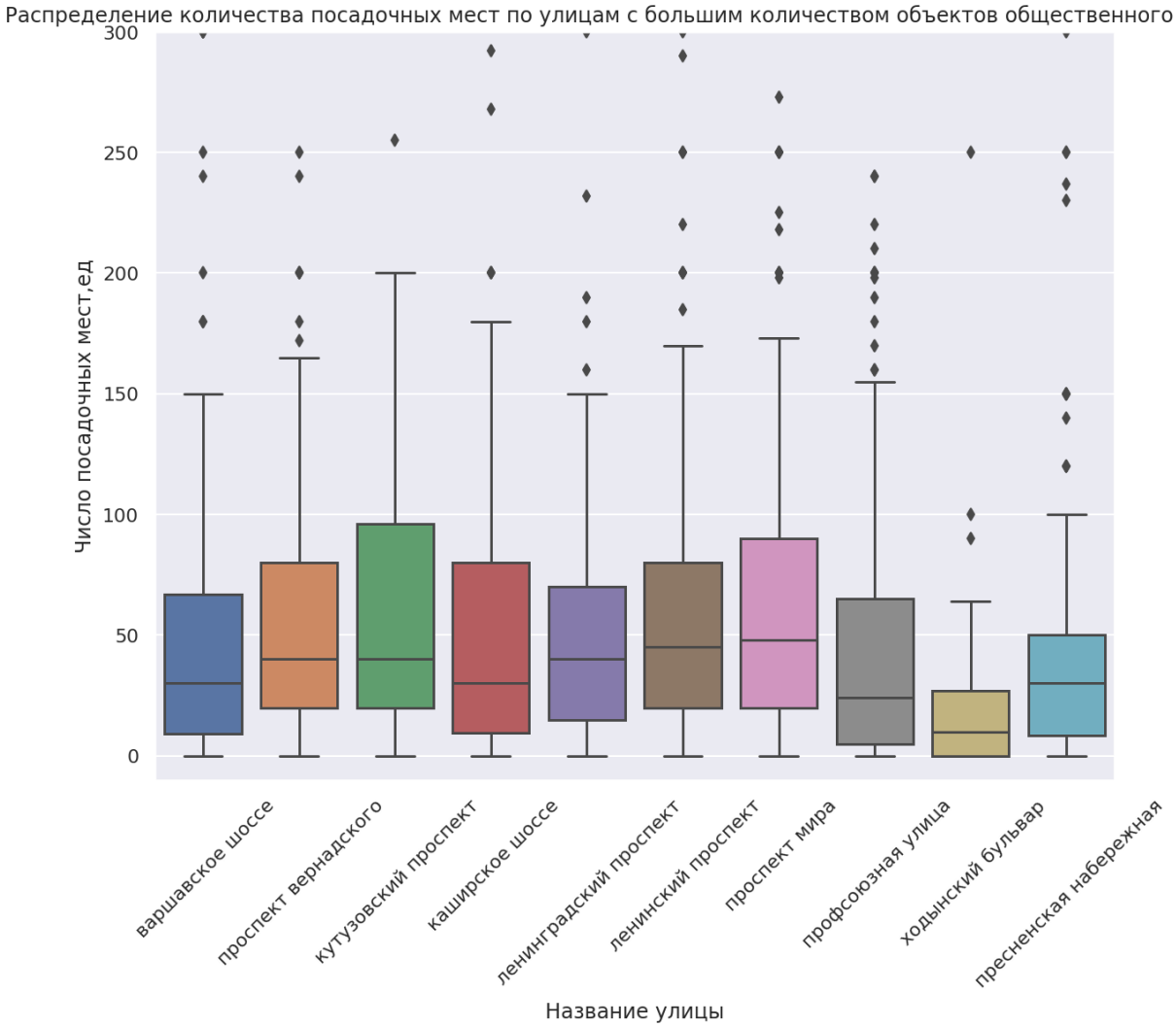
In [49]:
```
# строим диаграмму
temp = df[df['street_name'].isin(top_10)]
ax = sns.boxplot(x=temp['number_of_seats'])
ax.set_title('Среднее количество посадочных мест')
ax.set_xlabel('Количество посадочных мест')
ax.set_xlim(-10,200)
plt.show()
```



In [50]:
```
# строим гистограмму распределения количества посадочных мест
temp['number_of_seats'].hist(bins=60)
ax.set_title('Распределение количества посадочных мест ')
ax.set_xlabel('Количество посадочных мест')
plt.show()
```



In [51]:
```
# график распределения количества посадочных мест по улицам с большим количеством объект
ax = sns.boxplot(x="street_name", y="number_of_seats", data=temp, orient='v')
plt.xticks(rotation=45)
ax.set_title(' Распределение количества посадочных мест по улицам с большим количеством
ax.set_xlabel('Название улицы')
ax.set_ylabel('Число посадочных мест,ед')
ax.set_ylim(-10,300)
plt.show()
```



In [52]:
```
temp['number_of_seats'].describe()
```

Out[52]:
```
count   1,468.00
mean       55.78
std        89.11
min         0.00
25%        12.00
50%        36.00
75%        70.00
max     1,700.00
Name: number_of_seats, dtype: float64
```

Очень заметна тенденция: на популярных улицах среднее количество посадочных мест в заведениях небольшое,около 40.Есть улицы с большим количеством заведений, но с маленьким числом посадочных мест, например, "Профсоюзная". Скорее всего сказывается очень высокая цена на аренду коммерческой недвижимости.

3 Общий вывод

В ходе проекта обработал полученные данные.Первым делом проведена предобработка данных на наличие пропусков, дубликатов.Удалил дубликаты. Там, где это необходимо, заменил типы данных на необходимые для удобной работы. Использовал различные типы графиков: boxplot, гистограмму,barplot и scatterlot. Для работы с графиками использовал библиотеку seaborn.Так же использоал Python,Pandas,Plotly и визуализацию данных.

Проанализировал полученные результаты по объектам общественного питания Москвы:

Всего зарегистрировано 15281 заведений.Подавляющее большинство заведений - 'кафе' (6071). Затем идут 'столовые'(2584), 'рестораны'(2282) и 'фаст-фуд'(1897).Доля остальных заведений незначительна.

Количество несетевых заведений более чем в 4 раза превышает количество сетевых: 12317 против 2964.

У "фаст-фуда" доля сетевых заведений составляет более 40%.Среди сетевых лидируют так же форматы объектов: кулинария, ресторан и кафе. Практически нет сетевых столовых, хотя в Москве они вторые по численности - более 2500.

Наибольшое количество посадочных мест характерно в формате "столовая" и "ресторан". Притом, что столовые занимают последнюю строчку по численности сетевых заведений - 3шт. "Рестораны" занимают третье место по количеству. Среди остальных сетевых заведений характерно большое количество точек с небольшим количеством посадочных мест.И есть группа сетевых заведений с числом посадочных мест менее 20.

В среднем самое большое количество посадочных мест представляет формат"столовая", немного меньше -"ресторан".Бары и буфеты и кафе имеют почти одинаковое количество посадочных мест - около 40.

Топ-10 улиц по количеству заведений общепита располагается в большом количестве районов и в разных частях города.

В Москве 829 улиц с одним объектом общественного питания, которые находятся в 98 районах города.

Очень заметна тенденция: на популярных улицах среднее количество посадочных мест в заведениях небольшое,около 40.Есть улицы с большим количеством заведений, но с маленьким числом посадочных мест, например, "Профсоюзная". Скорее всего сказывается очень высокая цена на аренду коммерческой недвижимости.

Наибольшее количество объектов питания расположены на главных артериях города: Ленинградском проспекте, Просоюзной улице и проспекте Мира.

Рекомендации:

для такого оригинального кафе(гостей должны обслуживать роботы)следует обратить внимание на центральную часть города(постоянный случайный поток посетителей), в частности те улицы, где минимальное количество объектов общественного питания.
Формат для заведения следует выбрать "кафе" с количеством посадочных мест 30-50.
Но для более конкретных рекомендаций представленных данных недостаточно. Необходимо продолжить исследования- изучить соотношение числа посадочных мест к плотности населения(в спальных районах) или к трафику посетителей(в деловых). На улицах с одним заведением может быть как кафе(конкурент), так и кулинария, например.Так же изучить локацию, ведь на соседней улице,за углом, может быть с десяток заведений общепита.

Презентация:https://www.canva.com/design/DAE5WHraq6I/pLIcMOBPOISgPlNWCst9tw/edit (https://www.canva.com/design/DAE5WHraq6I/pLIcMOBPOISgPlNWCst9tw/edit)