

Исследование рынка общественного питания в Москве

Презентация: <https://yadi.sk/i/AqOokLv-9Mdwtw> Цель проекта - изучение рынка общественного питания в Москве для открытия нового кафе, в котором гостей будут обслуживать роботы-официанты.

В нашем распоряжении данные о заведениях общественного питания Москвы.

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.

План работы

1. Загрузка данных и подготовка к анализу
2. Анализ данных
 - Исследование соотношения видов объектов общественного питания по количеству
 - Исследование соотношения сетевых и несетевых заведений по количеству
 - Для какого вида объекта общественного питания характерно сетевое распространение?
 - Что характерно для сетевых заведений: много заведений с небольшим числом посадочных мест в каждом или мало заведений с большим количеством посадочных мест?
 - Среднее количество мест для каждого вида объекта общественного питания
 - Топ-10 улиц по количеству объектов общественного питания
 - Улицы с одним объектом общественного питания
 - Распределение количества посадочных мест для улиц с большим количеством объектов общественного питания
3. Вывод. Презентация

Загрузка данных и подготовка к анализу

```
In [1]: 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 numpy as np
from pymystem3 import Mystem
```

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

```
In [3]: rest_data.head()
```

Out[3]:

	id	object_name	chain	object_type		address	number
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 [4]:

rest_data.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 [5]:

rest_data.duplicated().sum()

Out[5]: 0

Пропущенных значений нет, дубликатов нет. Можно приступать к анализу

Анализ данных

Исследование соотношения видов объектов общественного питания по количеству

In [6]:

rest_data['object_type'].unique()

```
Out[6]: array(['кафе', 'столовая', 'закусочная',
   'предприятие быстрого обслуживания', 'ресторан', 'кафетерий',
   'буфет', 'бар', 'магазин (отдел кулинарии)'],
  dtype=object)
```

Для исследования соотношения видов объектов общепита по количеству лучше всего подойдет круговая диаграмма

In [7]:

types_count = rest_data.groupby('object_type').agg({'id':'nunique'}).reset_index()

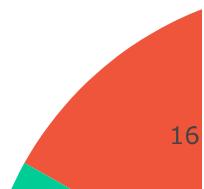
In [8]:

types_count.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   object_type 9 non-null     object 
 1   id          9 non-null     int64  
dtypes: int64(1), object(1)
memory usage: 144.0+ KB
```

```
dtypes: int64(1), object(1)
memory usage: 272.0+ bytes
```

```
In [9]: fig = go.Figure(data=[go.Pie(labels=types_count['object_type'], values=types_count['
fig.show()
```



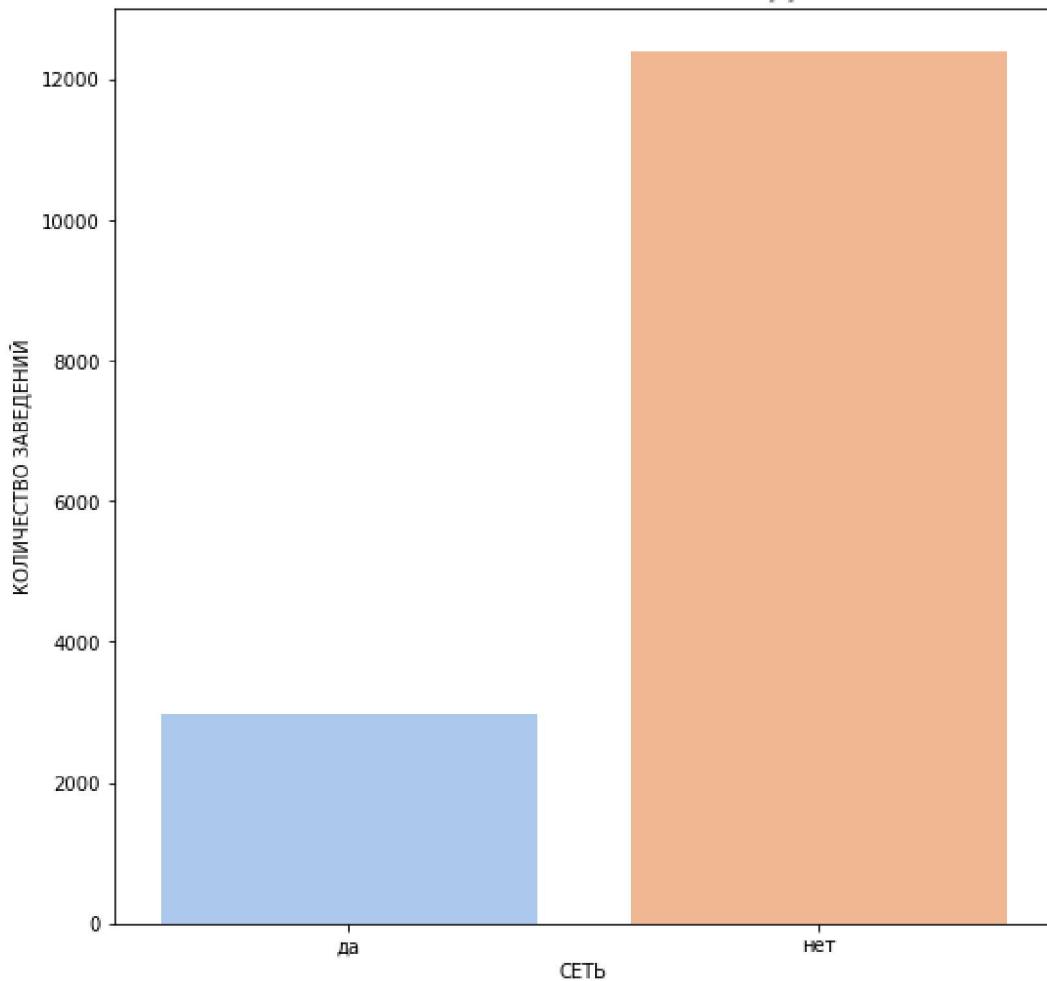
Большую часть заведений общепита Москвы составляют кафе, далее столовые, рестораны, предприятия быстрого обслуживания, бары и другие

Исследование соотношения сетевых и несетевых заведений по количеству

```
In [10]: chain_count = rest_data.groupby('chain').agg({'id':'nunique'}).reset_index()
```

```
In [11]: sns.set_palette('pastel')
plt.figure(figsize = (9, 9))
sns.barplot(data=chain_count, x= 'chain', y='id')
plt.title('Соотношение сетевых и несетевых заведений по количеству', color="grey", f
plt.xlabel('СЕТЬ')
plt.ylabel('КОЛИЧЕСТВО ЗАВЕДЕНИЙ')
plt.show()
```

Соотношение сетевых и несетевых заведений по количеству



Несетевых заведений намного больше, чем сетевых

Для какого вида объекта общественного питания характерно сетевое распространение?

```
In [12]: chain_object = rest_data[rest_data['chain']=='да'].groupby('object_type').agg({'id':
```

```
In [13]: objects_count = rest_data.groupby('object_type').agg({'id':'nunique'})
```

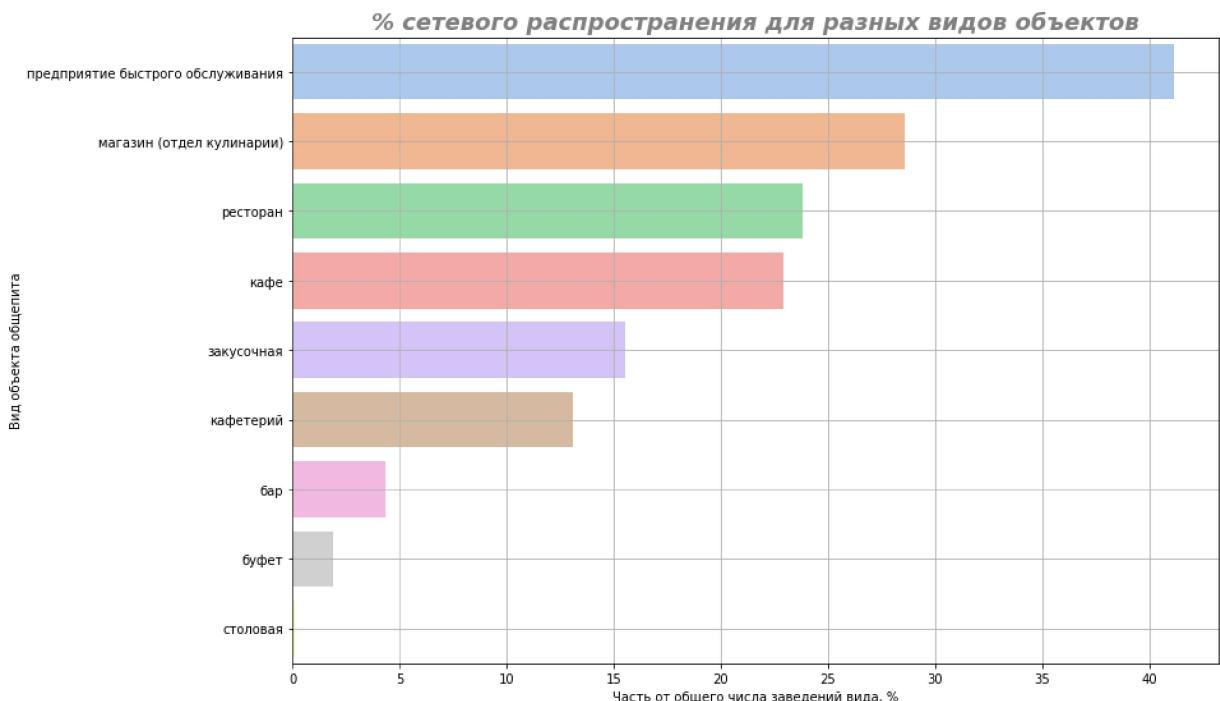
```
In [14]: chain_object = ((chain_object / objects_count) * 100).reset_index().sort_values('id'
```

```
In [15]: chain_object
```

	object_type	id
6	предприятие быстрого обслуживания	41.133645
5	магазин (отдел кулинарии)	28.571429
7	ресторан	23.807440
3	кафе	22.888998
2	закусочная	15.555556
4	кафетерий	13.065327
0	бар	4.322430
1	буфет	1.880342

	object_type	id
8	столовая	0.115964

```
In [16]: plt.figure(figsize=(13,9))
sns.barplot(x='id', y='object_type', data = chain_object)
plt.grid()
plt.title('% сетевого распространения для разных видов объектов', color="grey", font
plt.xlabel('Часть от общего числа заведений вида, %')
plt.ylabel('Вид объекта общепита')
plt.show()
```



Сетевое распространение характерно для предприятий быстрого обслуживания. Более 40% предприятий быстрого обслуживания являются сетевыми

Что характерно для сетевых заведений: много заведений с небольшим числом посадочных мест в каждом или мало заведений с большим количеством посадочных мест?

Небольшим считается заведение с количеством посадочных мест не более 50. Возьмем 50 посадочных мест за границу разбивки

Приведем все названия к нижнему регистру, уберем кавычки, слова 'кафе', 'ресторан'

```
In [17]: chain_rest_data = rest_data.loc[rest_data['chain'] == 'да'].reset_index()
```

```
In [18]: chain_rest_data['object_name'] = chain_rest_data['object_name'].str.lower()
```

```
In [19]: chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('«', '')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('»', '')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('кафе ', 'кафе')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace(' кафе', 'кафе-за')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('рестора', 'рестора')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('предпри', 'предпри')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('subway', 'subway')
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('кофейня', 'кофейня')
```

```

chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('. экспр
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('шоколад
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('пиццери
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('ростикс
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('starbus
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('кондите
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('prime п
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('хофе-ха
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('старбак
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('кулинар
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('праймпр
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('прайм-п
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('прайм-к
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('да пино
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('пицца п
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('детское
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('детский
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('бургер
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('vietcaf
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('старина
chain_rest_data['object_name'] = chain_rest_data['object_name'].str.replace('бар ', '

```

```

m=MyStem() rest_chain_lemmas =
chain_rest_data['object_name'].apply(m.lemmatize).value_counts() rest_chain_lemmas.tail(55)

```

На этом закончим обработку названий. Посмотрим, сколько заведений в сетях всего, и какое среднее количество посадочных мест

```

In [20]: chain_grouped = chain_rest_data.groupby('object_name').agg({'id':'count', 'number':'

```

```

In [21]: chain_grouped['number'] = chain_grouped['number'].round().astype('int')

```

```

In [22]: chain_grouped.head()

```

	object_name	id	number
0	beverly hills diner	1	88
1	bierloga	1	75
2	black & white	1	40
3	bocconcino	3	67
4	boobo	1	46

```

Out[22]:

```

```

In [23]: chain_grouped = chain_grouped.groupby('id').agg({'number':'mean'}).reset_index()

```

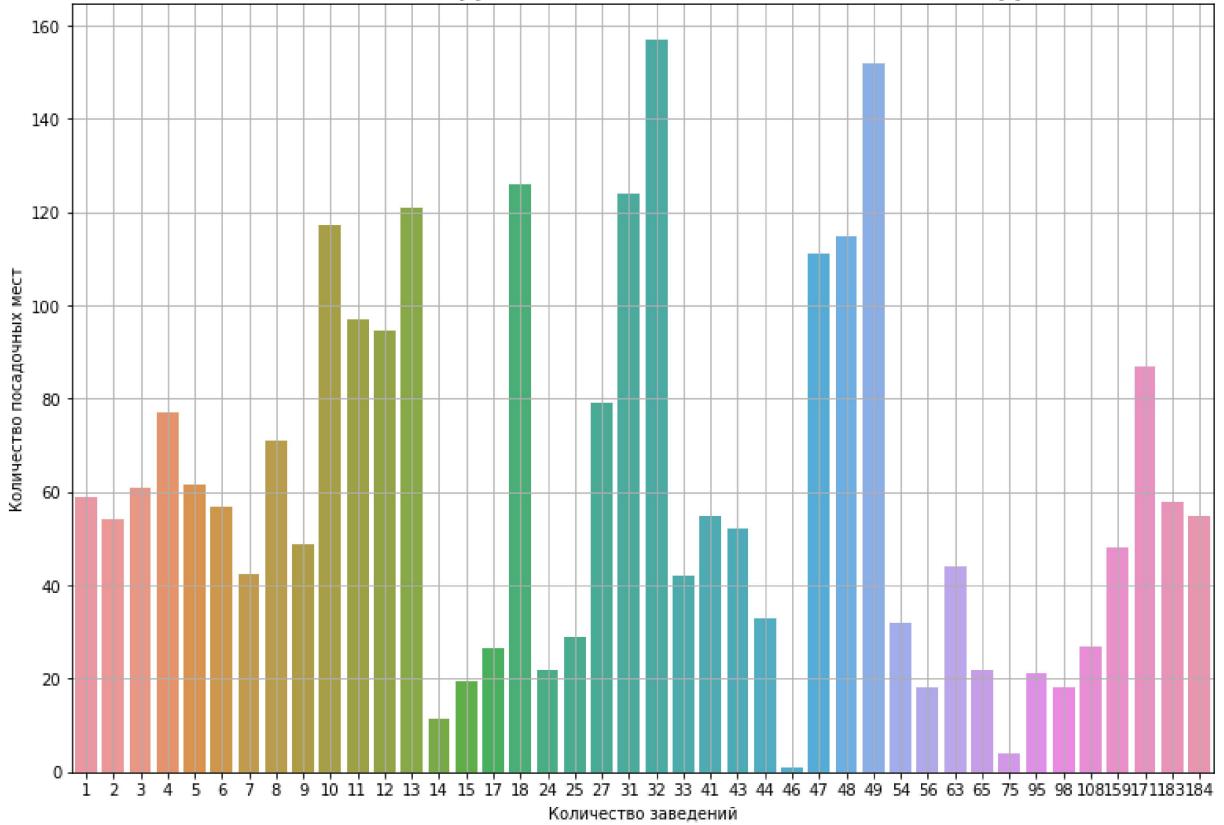
```

In [24]: plt.figure(figsize = (13, 9))
sns.barplot(x='id', y = 'number', data =chain_grouped)
plt.title('Количество посадочных мест на количество заведений', color="k", fontsize=
plt.ylabel("Количество посадочных мест")
plt.xlabel('Количество заведений')
plt.grid()

plt.show()

```

Количество посадочных мест на количество заведений

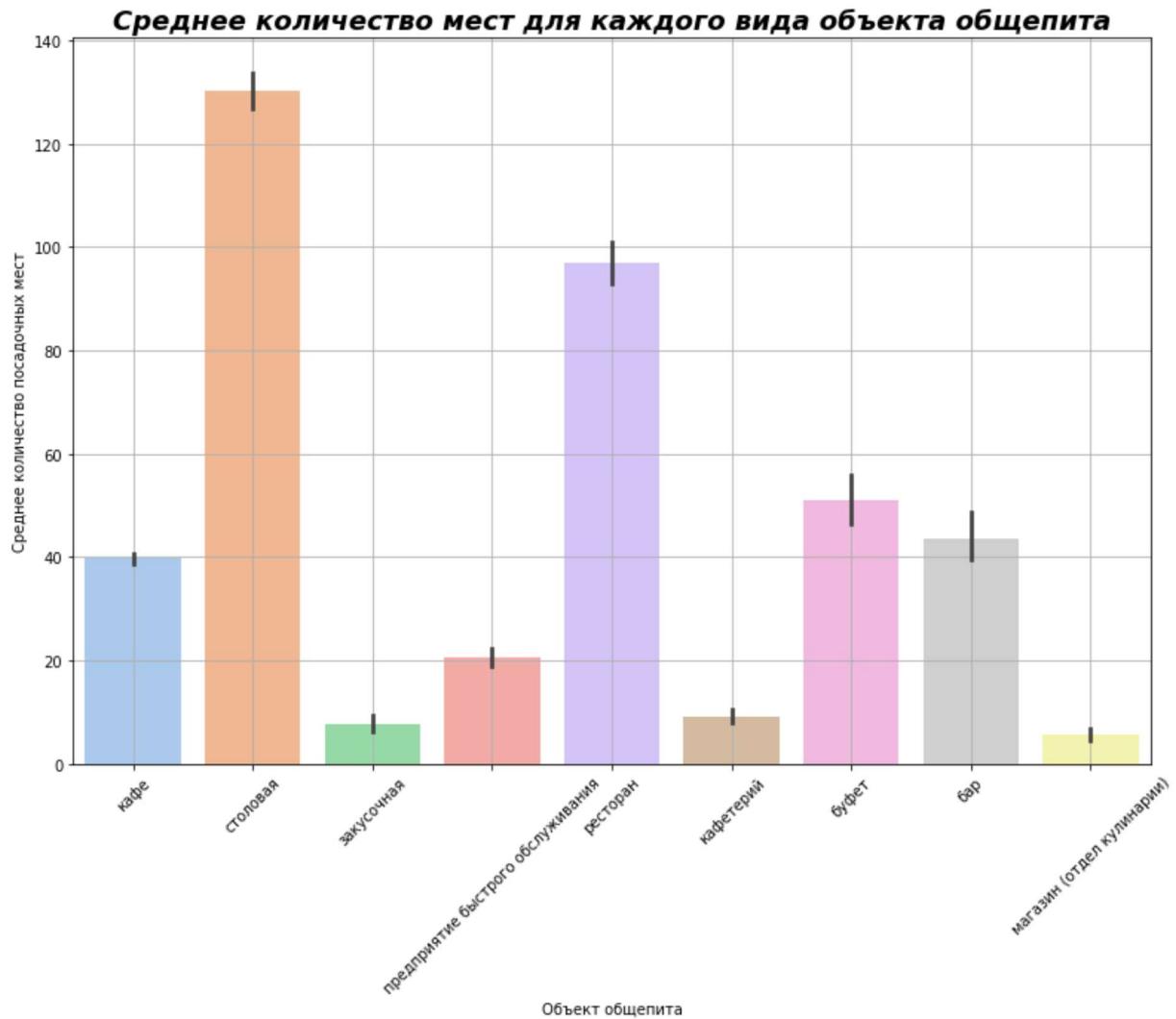


Из графика видно, что для сети, в которой много заведений (от 50), характерно меньшее количество посадочных мест, чем для сетей, в которых меньше заведений

Среднее количество мест для каждого вида объекта общепита

```
In [25]: plt.figure(figsize=(13,9))
plt.title('Среднее количество мест для каждого вида объекта общепита', color="k", fontweight='bold')
sns.barplot(x='object_type', y='number', data = rest_data)
plt.xlabel("Объект общепита")
plt.ylabel('Среднее количество посадочных мест')
plt.xticks(rotation=45)

plt.grid()
plt.show()
```



В среднем самое большое количество посадочных мест в столовых и в ресторанах

Топ-10 улиц по количеству объектов общественного питания

In [26]: `rest_data.head()`

	id	object_name	chain	object_type	address	number
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 [27]: `rest_data['street'] = rest_data['address'].str.replace('город Москва, ', ', ')
rest_data['street'] = rest_data['street'].str.replace('город Зеленоград, ', ', ')
rest_data['street'] = rest_data['street'].str.replace('поселение Сосенское, ', ', ')`

In [28]: `def street(street):
 street = street[:street.find(',')]
 return street`

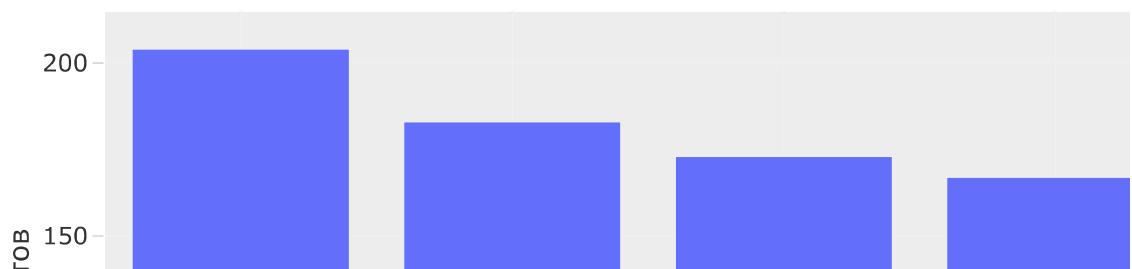
```
In [29]: rest_data['street'] = rest_data['street'].apply(street)
```

```
In [30]: top_streets = rest_data.groupby('street').agg({'id':'nunique'}).sort_values('id', as
```

```
In [31]: top_streets.columns = ['street', 'n_objects']
```

```
In [32]: fig = px.bar(top_streets, x='street', y='n_objects', title='Топ-10 улиц по количеству объектов')
fig.update_layout(template="ggplot2", yaxis_title="Количество объектов")
fig.update_xaxes(title_text="Улица", row=1, col=1)
fig.show()
```

Топ-10 улн



Улицы из топа достаточно крупные, нет ни одного переулка или тупика

```
In [33]: streets=pd.read_csv('https://raw.githubusercontent.com/Distans/temp/main/mosgaz-streets.csv')
```

```
In [34]: streets.head()
```

	streetname	areaid	okrug	area
0	Выставочный переулок	17	ЦАО	Пресненский район
1	улица Гашека	17	ЦАО	Пресненский район

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

```
In [35]: streets.columns = ['street', 'area_id', 'okrug', 'area']
```

```
In [36]: top_streets_areas = top_streets.merge(streets, on = 'street' )
```

```
In [37]: top_streets_areas
```

	street	n_objects	area_id	okrug	area
0	проспект Мира	204	70	CBAO	Алексеевский район
1	проспект Мира	204	86	CBAO	Ярославский Район
2	проспект Мира	204	78	CBAO	Район Марьина роща
3	проспект Мира	204	79	CBAO	Останкинский район
4	проспект Мира	204	81	CBAO	Район Ростокино
5	проспект Мира	204	82	CBAO	Район Свиблово
6	проспект Мира	204	16	ЦАО	Мещанский район
7	Профсоюзная улица	183	123	ЮЗАО	Академический район
8	Профсоюзная улица	183	132	ЮЗАО	Район Черемушки
9	Профсоюзная улица	183	134	ЮЗАО	Район Ясенево
10	Профсоюзная улица	183	126	ЮЗАО	Район Коньково
11	Профсоюзная улица	183	129	ЮЗАО	Обручевский район
12	Профсоюзная улица	183	131	ЮЗАО	Район Теплый Стан
13	Ленинградский проспект	173	54	CAO	Район Аэропорт
14	Ленинградский проспект	173	55	CAO	Район Беговой
15	Ленинградский проспект	173	69	CAO	Хорошевский район
16	Ленинградский проспект	173	66	CAO	Район Сокол
17	Пресненская набережная	167	17	ЦАО	Пресненский район
18	Варшавское шоссе	162	109	ЮАО	Район Чертаново Центральное
19	Варшавское шоссе	162	108	ЮАО	Район Чертаново Северное
20	Варшавское шоссе	162	110	ЮАО	Район Чертаново Южное
21	Варшавское шоссе	162	99	ЮАО	Донской район
22	Варшавское шоссе	162	102	ЮАО	Район Нагатино-Садовники
23	Варшавское шоссе	162	104	ЮАО	Нагорный район
24	Варшавское шоссе	162	130	ЮЗАО	Район Северное Бутово
25	Варшавское шоссе	162	133	ЮЗАО	Район Южное Бутово
26	Ленинский проспект	148	44	ЗАО	Район Проспект Вернадского

	street	n_objects	area_id	okrug	area
27	Ленинский проспект	148	47	ЗАО	Район Тропарево-Никулино
28	Ленинский проспект	148	99	ЮАО	Донской район
29	Ленинский проспект	148	124	ЮЗАО	Район Гагаринский
30	Ленинский проспект	148	128	ЮЗАО	Ломоносовский район
31	Ленинский проспект	148	129	ЮЗАО	Обручевский район
32	Ленинский проспект	148	131	ЮЗАО	Район Теплый Стан
33	Ленинский проспект	148	21	ЦАО	Район Якиманка
34	проспект Вернадского	132	44	ЗАО	Район Проспект Вернадского
35	проспект Вернадского	132	45	ЗАО	Район Раменки
36	проспект Вернадского	132	47	ЗАО	Район Тропарево-Никулино
37	проспект Вернадского	132	124	ЮЗАО	Район Гагаринский
38	проспект Вернадского	132	128	ЮЗАО	Ломоносовский район
39	Кутузовский проспект	114	38	ЗАО	Район Дорогомилово
40	Кутузовский проспект	114	49	ЗАО	Район Фили-Давыдково
41	Каширское шоссе	112	101	ЮАО	Район Москворечье-Сабурово
42	Каширское шоссе	112	102	ЮАО	Район Нагатино-Садовники
43	Каширское шоссе	112	106	ЮАО	Район Орехово-Борисово Южное
44	Каширское шоссе	112	105	ЮАО	Район Орехово-Борисово Северное
45	Кировоградская улица	110	109	ЮАО	Район Чертаново Центральное
46	Кировоградская улица	110	108	ЮАО	Район Чертаново Северное
47	Кировоградская улица	110	110	ЮАО	Район Чертаново Южное

In [38]: `top_streets_areas.pivot_table(index = ['area', 'street'], values = 'n_objects')`

Out[38]:

area	street	n_objects
Академический район	Профсоюзная улица	183
Алексеевский район	проспект Мира	204
Донской район	Варшавское шоссе	162
	Ленинский проспект	148
Ломоносовский район	Ленинский проспект	148
	проспект Вернадского	132
Мещанский район	проспект Мира	204
Нагорный район	Варшавское шоссе	162
Обручевский район	Ленинский проспект	148
	Профсоюзная улица	183
Останкинский район	проспект Мира	204

n_objects

area	street	
Пресненский район	Пресненская набережная	167
Район Аэропорт	Ленинградский проспект	173
Район Беговой	Ленинградский проспект	173
Район Гагаринский	Ленинский проспект	148
	проспект Вернадского	132
Район Дорогомилово	Кутузовский проспект	114
Район Коньково	Профсоюзная улица	183
Район Марьина роща	проспект Мира	204
Район Москворечье-Сабурово	Каширское шоссе	112
Район Нагатино-Садовники	Варшавское шоссе	162
	Каширское шоссе	112
Район Орехово-Борисово Северное	Каширское шоссе	112
Район Орехово-Борисово Южное	Каширское шоссе	112
Район Проспект Вернадского	Ленинский проспект	148
	проспект Вернадского	132
Район Раменки	проспект Вернадского	132
Район Ростокино	проспект Мира	204
Район Свиблово	проспект Мира	204
Район Северное Бутово	Варшавское шоссе	162
Район Сокол	Ленинградский проспект	173
Район Теплый Стан	Ленинский проспект	148
	Профсоюзная улица	183
Район Тропарево-Никулино	Ленинский проспект	148
	проспект Вернадского	132
Район Фили-Давыдково	Кутузовский проспект	114
Район Черемушки	Профсоюзная улица	183
Район Чертаново Северное	Варшавское шоссе	162
	Кировоградская улица	110
Район Чертаново Центральное	Варшавское шоссе	162
	Кировоградская улица	110
Район Чертаново Южное	Варшавское шоссе	162
	Кировоградская улица	110
Район Южное Бутово	Варшавское шоссе	162
Район Якиманка	Ленинский проспект	148
Район Ясенево	Профсоюзная улица	183

n_objects

area	street	n_objects
Хорошевский район	Ленинградский проспект	173
Ярославский Район	проспект Мира	204

Топ-10 улиц превратились в 48 районов Москвы, так как, как уже говорилось, топовые улицы достаточно крупные, захватываюx сразу несколько районов и даже округов

Улицы и районы с одним объектом общественного питания

```
In [39]: single_streets = rest_data.groupby('street').agg({'id':'nunique'}).reset_index()
```

```
In [40]: single_streets = single_streets.loc[single_streets['id'] == 1]
```

```
In [41]: single_streets
```

```
Out[41]:
```

	street	id
2	1-й Балтийский переулок	1
3	1-й Басманный переулок	1
4	1-й Ботанический проезд	1
7	1-й Вешняковский проезд	1
9	1-й Голутвинский переулок	1
...
1952	улица Шеногина	1
1955	улица Шумилова	1
1956	улица Шумкина	1
1960	улица Щорса	1
1965	шоссе Фрезер	1

593 rows x 2 columns

593 улицы, на которых всего по 1 объекту общественному питанию

```
In [42]: single_streets = single_streets.merge(streets, on = 'street')
```

```
In [43]: single_streets.pivot_table(index = 'area', values = 'street', aggfunc = 'count').res
```

```
Out[43]:
```

	area	street
96	Таганский район	27
82	Район Хамовники	26
3	Басманный район	25
21	Пресненский район	20
97	Тверской район	20
52	Район Марьина роща	18

	area	street
22	Район Арбат	18
13	Мещанский район	15
76	Район Сокольники	15
34	Район Замоскворечье	14

Больше всего таких улиц в центральных районах Москвы

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

In [44]: rest_data

	id	object_name	chain	object_type	address	number	street
0	151635	СМЕТАНА	нет	кафе	город Москва, улица Егора Абакумова, дом 9	48	улица Егора Абакумова
1	77874	Родник	нет	кафе	город Москва, улица Талалихина, дом 2/1, корпус 1	35	улица Талалихина
2	24309	Кафе «Академия»	нет	кафе	город Москва, Абельмановская улица, дом 6	95	Абельмановская улица
3	21894	ПИЦЦЕТОРИЯ	да	кафе	город Москва, Абрамцевская улица, дом 1	40	Абрамцевская улица
4	119365	Кафе «Вишневая метель»	нет	кафе	город Москва, Абрамцевская улица, дом 9, корпус 1	50	Абрамцевская улица
...
15361	208537	Шоколадница	да	кафе	город Москва, 3-й Крутицкий переулок, дом 18	50	3-й Крутицкий переулок
15362	209264	Шоколадница	да	кафе	город Москва, улица Земляной Вал, дом 33	10	улица Земляной Вал
15363	209186	Шоколадница	да	кафе	город Москва, улица Земляной Вал, дом 33	20	улица Земляной Вал
15364	221900	Шоколадница	да	кафе	город Москва, поселение Московский, Киевское ш...	36	поселение Московский
15365	222535	Шоколадница	да	кафе	город Москва, Ходынский бульвар, дом 4	10	Ходынский бульвар

15366 rows × 7 columns

```
In [45]: top_streets = top_streets.set_index('street')

In [46]: new_data = rest_data.set_index('street').join(top_streets)

In [47]: new_data = new_data.reset_index()

In [48]: new_data['n_objects'] = new_data['n_objects'].fillna(0)

In [49]: new_data = new_data[new_data['n_objects'] != 0]

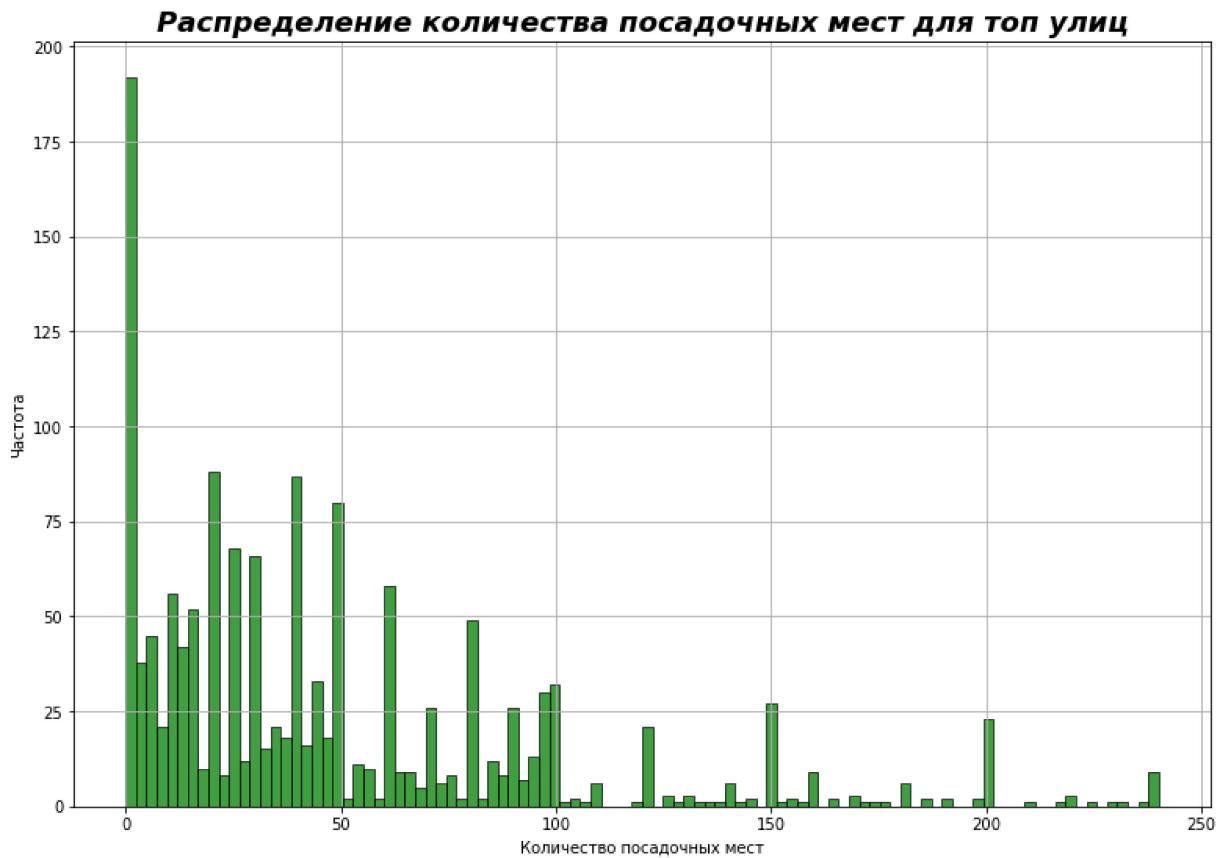
In [50]: new_data = new_data[['street', 'number']]

In [51]: plt.figure(figsize=(13,9))
plt.title('Распределение количества посадочных мест для топ улиц', color="k", fontsize=14)
sns.histplot(new_data['number'], color = 'g', bins = 100)
plt.grid()
plt.xlabel('Количество посадочных мест')
plt.ylabel('Частота')
plt.show()
```



Большинство значений до 250, уменьшим зону рассмотрения

```
In [52]: plt.figure(figsize=(13,9))
plt.title('Распределение количества посадочных мест для топ улиц', color="k", fontsize=14)
sns.histplot(new_data.loc[new_data['number'] < 250]['number'], color = 'g', bins = 100)
plt.grid()
plt.xlabel('Количество посадочных мест')
plt.ylabel('Частота')
plt.show()
```



Самое частое количество посадочных мест - 0. Скорее всего это бары без посадочных мест, заведения, работающие только на вынос

Вывод. Презентация

Вывод:

- кафе с роботами - отличная идея, но стоит учитывать, что тип заведения "кафе" составляет большинство на рынке общественного питания Москвы (~40%)
- несетевые заведения составляют большую часть заведений общепита. Если кафе с роботами превратиться в сеть, то шанс на удержание аудитории выше (сетевые кафе составляют всего 15%)
- что касается размеров - зависит от количества открываемых кафе. Меньше точек больше посадочных мест, больше точек - меньше посадочных мест
- в среднем на кафе приходится 40 посадочных мест
- место: в центральных районах Москвы более 25 улиц с 1 заведением на каждой. Стоит обратить на них внимание, чтобы не перенаселять топ-улицы количеством заведений общепита
- НО: учитывая, что на топ-улицах много заведений без посадочных мест (скорее всего бары), то стоит рассмотреть и открытие кафе на топ-улице с посадкой ниже среднего (менее 40 мест)