# Рынок заведений общественного питания Москвы

## Описание проекта

Мы решили открыть небольшое кафе в Москве. Оно оригинальное — гостей должны обслуживать роботы. Проект многообещающий, но дорогой. Вместе с партнёрами мы решились обратиться к инвесторам. Их интересует текущее положение дел на рынке — сможем ли мы снискать популярность на долгое время, когда все зеваки насмотрятся на роботов-официантов?
Мы — гуру аналитики, и партнёры просят нас подготовить исследование рынка. У нас есть открытые данные о заведениях общественного питания в Москве.

## Оглавление



## Описание данных и задачи

[к Оглавлению](#0.0)

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

## Задачи

#### Шаг 1. Загрузить данные и подготовить их к анализу

- Убедиться, что тип данных в каждой колонке — правильный, а также отсутствуют пропущенные значения и дубликаты. При необходимости обработать их.

#### Шаг 2. Анализ данных

- Исследуем соотношение видов объектов общественного питания по количеству. Построим график.
- Исследуем соотношение сетевых и несетевых заведений по количеству. Построим график.
- 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?
- Для каждого вида объекта общественного питания опишем среднее количество посадочных мест. Какой вид предоставляет в среднем самое большое количество посадочных мест? Построим графики.
- Выделим в отдельный столбец информацию об улице из столбца `address`.
- Построим график топ-10 улиц по количеству объектов общественного питания. Воспользуемся внешней информацией и ответьтим на вопрос — в каких районах Москвы находятся эти улицы?
- Найдем число улиц с одним объектом общественного питания. Воспользуемся внешней информацией и ответим на вопрос — в каких районах Москвы находятся эти улицы?
- Посмотрим на распределение количества посадочных мест для улиц с большим количеством объектов общественного питания. Узнаем какие закономерности можно выявить?

#### Шаг 3. Общий вывод

- Сделаем общий вывод и дадим рекомендации о виде заведения, количестве посадочных мест, а также районе расположения. Прокомментируем возможность развития сети.

#### Шаг 4. Подготовка презентации

- Подготовим презентацию исследования для инвесторов. Для создания презентации используем любой удобный инструмент и отправим презентацию обязательно в формате pdf.

- Приложим ссылку на презентацию в markdown-ячейке в формате: `Презентация: <ссылка на облачное хранилище с презентацией>`

## 1. Загрузка данных и подготовка их к анализу

[к Оглавлению](#0.0)

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from IPython.display import display
import numpy as np
import plotly.express as px
from plotly import graph_objects as go
from plotly.offline import plot, iplot

#!pip install psutil
#!conda install -c plotly plotly-orca
import plotly.graph_objects as go

from io import BytesIO
import requests
from bs4 import BeautifulSoup
import re
```

In [2]:

```
!pip install plotly==4.10.0
```

```
Requirement already satisfied: plotly==4.10.0 in c:\users\use
r\appdata\local\programs\python\python38\lib\site-packages
(4.10.0)
Requirement already satisfied: six in c:\users\user\appdata\l
ocal\programs\python\python38\lib\site-packages (from plotly=
=4.10.0) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\us
er\appdata\local\programs\python\python38\lib\site-packages
(from plotly==4.10.0) (1.3.3)
```

In [3]:

```
API_key = 'c5320408-6d41-4fbe-a245-9edd413765b8'
```

In [4]:

```
df = pd.read_csv('rest_data.csv')
```

In [5]:

```
#df = pd.read_csv('/datasets/rest_data.csv')
```

**Изучим общую информацию о данных.** Посмотрим таблицу. Изучим описание данных для

числовых колонок.
Убедимся, что тип данных в каждой колонке — правильный, а также отсутствуют пропущенные значения и дубликаты. При необходимости обработем их.

In [6]:

```
display(df.head(), df.info(), df.describe())
```

```
<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
```

| | 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 |

```
None
```

| | id | number |
|---|---|---|
| **count** | 15366.000000 | 15366.000000 |
| **mean** | 119720.066901 | 59.547182 |
| **std** | 73036.130732 | 74.736833 |
| **min** | 838.000000 | 0.000000 |
| **25%** | 28524.000000 | 12.000000 |
| **50%** | 144974.500000 | 40.000000 |
| **75%** | 184262.250000 | 80.000000 |
| **max** | 223439.000000 | 1700.000000 |

**Посмотрим есть ли пропущенные значения в наших данных**

In [7]:

```
df.isnull().sum()
```

Out[7]:

```
id             0
object_name    0
chain          0
object_type    0
address        0
number         0
dtype: int64
```

**Посмотри наличие дубликатов**

In [8]:

```
df['object_name'].duplicated().sum()
```

Out[8]:

```
4973
```

**Изучим названия заведений общественного питания и посчитаем количество названий**

In [9]:

```
df['object_name'].value_counts()
```

Out[9]:

```
Столовая                                  267
Кафе                                      236
Шаурма                                    234
KFC                                       155
Шоколадница                               142
                                         ...
Консерватория им. П.И. Чайковского          1
ФИЛИАЛ ШБС ЛЮБЛИНО ШК. 1146                 1
ЗАКРЫТЫЙ КЛУБ «ГАРАЖ»                       1
СТОЛОВАЯ ПРИ ГОУ ЭШВСМ «СЕВЕРНЫЙ»           1
Буфет при ГОУ СОШ №1114                     1
Name: object_name, Length: 10393, dtype: int64
```

**Изучим тип заведений общественного питания и посчитаем количество названий**

In [10]:

```
df['object_type'].value_counts()
```

Out[10]:

```
кафе                                  6099
столовая                              2587
ресторан                              2285
предприятие быстрого обслуживания    1923
```

```
бар                                  856
буфет                                585
кафетерий                            398

закусочная                           360
магазин (отдел кулинарии)            273
Name: object_type, dtype: int64
```

**Посчитаем количество сетевых заведений общественного питания**

In [11]:

```
df['chain'].value_counts()
```

Out[11]:

```
нет    12398
да      2968
Name: chain, dtype: int64
```

**Посчитаем количество посадочных мест в заведениях общественного питания**

In [12]:

```
df['number'].value_counts()
```

Out[12]:

```
0      1621
40      835
20      727
30      685
10      644
       ...
491       1
675       1
455       1
167       1
495       1
Name: number, Length: 315, dtype: int64
```

Очень много заведений, в которых не указано количество мест, надо узнать, что это за заведения и где находятся.

In [13]:

```
df['address'].value_counts()
```

Out[13]:

```
город Москва, Ходынский бульвар, дом 4                  95
город Москва, Пресненская набережная, дом 2             63
город Москва, проспект Мира, дом 211, корпус 2          60
город Москва, Кировоградская улица, дом 13А             53
город Москва, площадь Киевского Вокзала, дом 2          48
                                                        ..
город Москва, улица Гарибальди, дом 27, корпус 4         1
город Москва, Клинская улица, дом 20                     1
```

```
город Москва, улица Солянка, дом 11/6, строение 1        1
город Москва, улица Островитянова, дом 16, корпус 5      1


город Москва, улица Дмитриевского, дом 23                1
```

Интересно, почему аж 95 заведений расположены по одному адресу. Разберемся.

# 2. Анализ данных

## 2.1. Соотношение видов объектов общественного питания по количеству

[к Оглавлению](#0.0)

Сгруппируем данные о видах обектов общественного питания и посчитаем их количество. Построим график.

In [14]:

```
object_type_count = df.groupby('object_type').agg({'id': ['count']}).reset
_index()
object_type_count.columns = ['object_type','count']
object_type_count['percent'] = round((object_type_count['count']/object_ty
pe_count['count'].sum())*100,1)
object_type_count = object_type_count.sort_values(by='count', ascending=Fa
lse)
object_type_count
```

Out[14]:

| | object_type | count | percent |
|---|---|---|---|
| **3** | кафе | 6099 | 39.7 |
| **8** | столовая | 2587 | 16.8 |
| **7** | ресторан | 2285 | 14.9 |
| **6** | предприятие быстрого обслуживания | 1923 | 12.5 |
| **0** | бар | 856 | 5.6 |
| **1** | буфет | 585 | 3.8 |
| **4** | кафетерий | 398 | 2.6 |
| **2** | закусочная | 360 | 2.3 |
| **5** | магазин (отдел кулинарии) | 273 | 1.8 |

In [15]:

```
fig = go.Figure(data=[go.Bar(x=object_type_count['object_type'], y=object_
type_count['count'])])
fig.update_layout(barmode='group', title_text='График количества видов объ
```

```
ектов общественного питания (ООП)',
                  xaxis_title="Вид объекта общественного питания", yaxis_t
itle="Количество ООП, ед.",)
fig.show()
```

Кафе являются лидерами по количеству открытых заведений - 6099 из 15366 или 40% рынка.
Столовая, ресторан и предприятие быстрого обслуживания составляют после кафе тройку лидеров по популярности.
Аутсайдеры - магазины (отдел кулинарии) - и это логично, ведь чашка кофе и круассан, предлагаемые посетителям, - не могут составить конкуренцию меню кафе или ресторана. Впрочем, для магазинов эта статья доходов не является ключевой

## 2.2. Соотношение сетевых и несетевых заведений по количеству

[к Оглавлению](#0.0)

Сгруппируем данные о количестве сетевых и несетевых обектах общественного питания.
Построим круговой график.

In [16]:

```
df['chain'] = df['chain'].replace('да', "сетевое заведение").replace('нет'
, "несетевое заведение")
chain_count = df.groupby('chain').agg({'id': ['count']}).reset_index()
chain_count.columns = ['chain','count']
chain_count['percent'] = round((chain_count['count']/chain_count['count'].
sum())*100,1)
chain_count = chain_count.sort_values(by='count', ascending=False)
chain_count
```

Out[16]:

| | chain | count | percent |
|---|---|---|---|
| **0** | несетевое заведение | 12398 | 80.7 |
| **1** | сетевое заведение | 2968 | 19.3 |

In [17]:

```
fig = go.Figure(data=[go.Pie(labels=['Несетевые заведения', 'Сетевые завед
ения'], values=chain_count['count'])])
fig.update_layout(#legend_orientation="h",
                  #legend=dict(x=.5, xanchor="center"),
                  title="График количества сетевых и несетевых заведений")
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_si
ze=20,
                  marker=dict(line=dict(color='#000000', width=2)))
fig.show()
```

Большинство заведений общественного питания - несетевые, 12398 заведений, их доля составляет 80.7%.
Сетевые кафе и рестораны, 2968 предприятий, занимают оставшиеся 19.3% рынка

## 2.3. Для какого вида объекта общественного питания характерно сетевое распространение?

[к Оглавлению](#0.0)

Сгруппируем данные о количестве сетевых и несетевых обектах общественного питания по типу ООП. Построим гистограмму график.

In [18]:

```
type_chain = df.pivot_table(index='object_type',columns='chain', values='id', aggfunc='count').reset_index()
type_chain['percent'] = (type_chain['сетевое заведение'] / type_chain['сетевое заведение'].sum() * 100).round(1)
type_chain['chain_ratio'] = (type_chain['сетевое заведение'] / type_chain['несетевое заведение'] * 100).round(1)
type_chain = type_chain.sort_values(by='сетевое заведение', ascending=False)
type_chain
```

Out[18]:

| chain | object_type | несетевое заведение | сетевое заведение | percent | chain_ratio |
|---|---|---|---|---|---|
| 3 | кафе | 4703 | 1396 | 47.0 | 29.7 |
| 6 | предприятие быстрого обслуживания | 1132 | 791 | 26.7 | 69.9 |
| 7 | ресторан | 1741 | 544 | 18.3 | 31.2 |
| 5 | магазин (отдел кулинарии) | 195 | 78 | 2.6 | 40.0 |
| 2 | закусочная | 304 | 56 | 1.9 | 18.4 |
| 4 | кафетерий | 346 | 52 | 1.8 | 15.0 |
| 0 | бар | 819 | 37 | 1.2 | 4.5 |
| 1 | буфет | 574 | 11 | 0.4 | 1.9 |
| 8 | столовая | 2584 | 3 | 0.1 | 0.1 |

In [19]:

```
fig = go.Figure(data=[go.Bar(name='Сетевые заведения', x=type_chain['object_type'], y=type_chain['сетевое заведение']),
                      go.Bar(name='Несетевые заведения', x=type_chain['object_type'], y=type_chain['несетевое заведение'])])
fig.update_layout(barmode='group', title_text='Распространение сетевых заведений по видам объектов общественного питания',
                  xaxis_title="Объект общественного питания", yaxis_title=
```

```
"Количество ООП, ед.",)
fig.show()
```

По абсолютному количеству сетевых заведений лидируют **Кафе** - 1396 заведение **Предприятия быстрого обслуживания** - 791 заведение **Рестораны** - 544 заведения

In [20]:

```
type_chain_sorted = type_chain.sort_values(by='percent')
fig = px.bar(type_chain_sorted, x='percent', y='object_type', orientation=
'h', text='percent', height=600, width=1000,
             title='Процент распространения сетевых заведений по видам объ
ектов общественного питания',
             labels={'percent':'Доля сетевых заведений', 'object_type':'Ви
д объекта общественного питания'})
fig.update_traces(textposition='outside')
fig.show()
```

Если посмотреть на относительную величину сетевого распространения среди заведений общепита, то на первом месте предприятия быстрого обслуживания - 70% среди них сетевые. На втором месте отделы кулинарии в магазинах - 40% сетевых заведений. Третье место разделили между собой рестораны и кафе - 31% и 30% соответственно

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

[к Оглавлению](#0.0)

Сделаем срез по сетевым заведениям общественного питания.

In [21]:

```
df_chain_yes = df.query('chain == "сетевое заведение"')
```

**Изучим данные по названиям заведений и посмотрим распределение мест и количество заведений**

In [22]:

```
df_name_mean = (df_chain_yes.groupby('object_name').agg({'number':['mean'
],'id': ['count']}).reset_index())
df_name_mean.columns = ['object_name', 'number', 'count']
df_name_mean['number'] = df_name_mean['number'].astype('int')
df_name_mean = df_name_mean.sort_values(by='count', ascending=False)
df_name_mean = df_name_mean.query('count>1') # Заведение будем считать сет
```

```
евым, когда их более 1.
df_name_mean.tail(10)
```

Out[22]:

| | object_name | number | count |
|---|---|---|---|
| **213** | Зодиак | 72 | 2 |
| **114** | Барбекю | 22 | 2 |
| **216** | ИЛЬ ПАТИО | 56 | 2 |
| **207** | Закусочная «Бургер Кинг» | 20 | 2 |
| **218** | Изба | 17 | 2 |
| **13** | Correas | 50 | 2 |
| **444** | НИЯМА | 69 | 2 |
| **88** | Азбука вкуса | 19 | 2 |
| **116** | Бенто WOK | 18 | 2 |
| **652** | Темпл бар | 147 | 2 |

**Посмотрим распределение количества сетевых заведений**

In [23]:

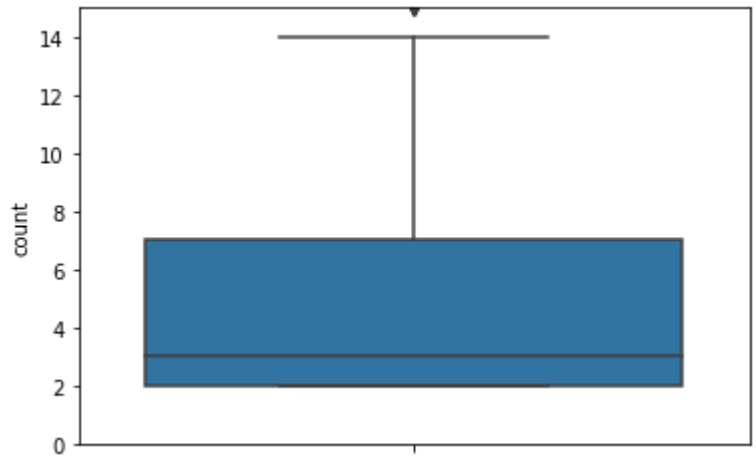
```
display(df_name_mean['count'].reset_index().describe(percentiles=[0.1, 0.25, 0.36, 0.4, 0.50, 0.6, 0.75, 0.8, 0.9, 0.99]).T[1:])
```

| | count | mean | std | min | 10% | 25% | 36% | 40% | 50% | 60% | 75% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 274.0 | 9.189781 | 19.502431 | 2.0 | 2.0 | 2.0 | 2.28 | 3.0 | 3.0 | 4.0 | 7.0 |

Узнаем нормальное распределение количества заведений в сети

In [24]:

```
sns.boxplot(y=df_name_mean['count'])
plt.ylim(0,15)
plt.show()
```



Нормальное распределение количества заведении в сети - от 2 до 14

**Посмотрим распределение количества мест в сетевых заведениях**

In [25]:

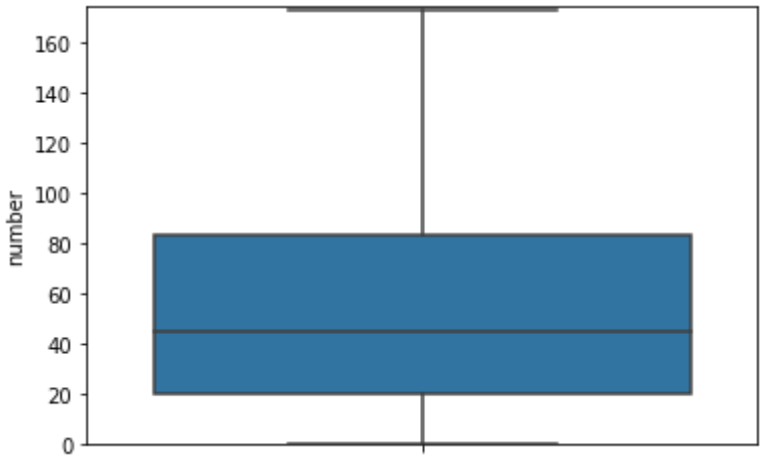
```
display(df_name_mean['number'].reset_index().describe(percentiles=[0.1, 0.
25, 0.3, 0.4, 0.5, 0.6, 0.75, 0.8, 0.9, 0.99]).T[1:])
```

| | count | mean | std | min | 10% | 25% | 30% | 40% | 50% | 60% | 75 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **number** | 274.0 | 57.437956 | 48.487355 | 0.0 | 9.3 | 20.0 | 22.0 | 32.2 | 44.5 | 56.0 | 82.1 |

Узнаем нормальное распределение количества посадочных мест в сети заведений

In [26]:

```
sns.boxplot(y=df_name_mean['number'])
plt.ylim(0,174)
plt.show()
```




Нормальное распределение количества заведении в сети - от 0 до 174

Медиана количества заведений в каждой сети - три заведения.
Медиана количества мест в заведении - 44,5.
75% сетей имеет не более 7 заведений с 174 посадочными местами

In [27]:

```
fig = go.Figure(data=[go.Bar(name='Количество посадочных мест', x=df_name_
mean['object_name'][:30], y=df_name_mean['number']),
                      go.Bar(name='Количество заведений', x=df_name_mean[
'object_name'][:30], y=df_name_mean['count'])])
fig.update_layout(barmode='group', title_text='Количество сетевых заведени
й по названиям объектов общественного питания',
                  xaxis_title="Объект общественного питания", yaxis_title=
"Количество заведенийООП/посадочных мест, ед.",)
fig.show()
```

Отсортируем даные по количеству посадочных мест в сетевых заведениях

In [28]:

```
df_name_mean_c = df_name_mean.sort_values(by='number', ascending=False)
df_name_mean_c.head(10)
```

Out[28]:

| | object_name | number | count |
|---|---|---|---|
| **458** | ПИЛЗНЕР | 245 | 2 |
| **332** | Кафе Пронто | 222 | 2 |
| **597** | СТАРИНА МЮЛЛЕР Старина Миллер | 215 | 2 |
| **105** | Бакинский бульвар | 213 | 2 |
| **295** | Кафе «Му-Му» | 203 | 3 |
| **287** | Кафе «Кружка» | 194 | 4 |
| **708** | ЯКИТОРИЯ | 185 | 5 |
| **214** | Золотая вобла | 173 | 3 |
| **161** | Грабли | 166 | 8 |
| **410** | МУ-МУ | 164 | 8 |

In [29]:

```
fig = go.Figure(data=[go.Bar(name='Количество посадочных мест', x=df_name_
mean_c['object_name'][:30], y=df_name_mean_c['number']),
                      go.Bar(name='Количество заведений', x=df_name_mean_c
['object_name'][:30], y=df_name_mean_c['count'])])
fig.update_layout(barmode='group', title_text='Количество посадочных мест
 в сетевых заведениях объектов общественного питания',
                  xaxis_title="Объект общественного питания", yaxis_title=
"Количество посадочных мест/заведений, ед.",)
fig.show()
```

Построим график количества заведений к чису посадочных мест, ограничив данные верхним усом графиков боксплот

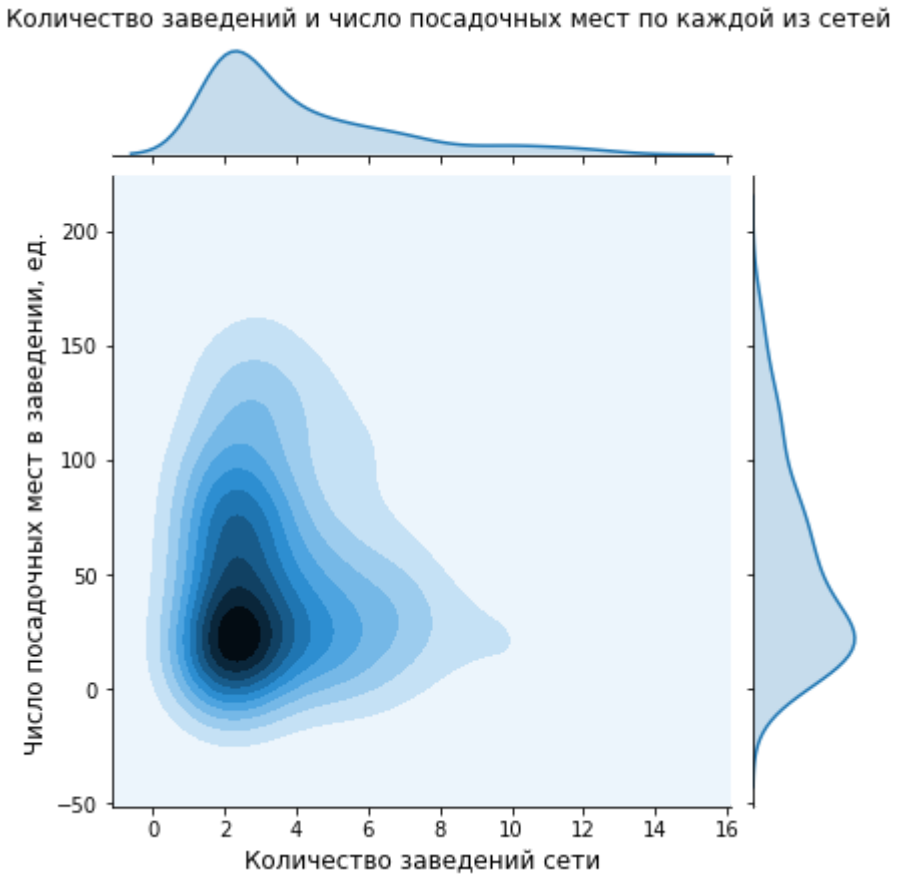
In [30]:

```
df_name_mean = df_name_mean.query('number < 174 & count < 14')
```

In [31]:

```
p = sns.jointplot(x=df_name_mean["count"], y=df_name_mean["number"], kind=
'kde')

p.fig.suptitle('Количество заведений и число посадочных мест по каждой из
 сетей', y=1.02)
p.set_axis_labels("Количество заведений сети", "Число посадочных мест в за
ведении, ед.", fontsize=12)
plt.show()
```




In [32]:

```
print('Среднее количество сетевых заведений - ', df_name_mean_c['count'].median(), 'ед.')
print('Среднее количество посадочных мест в сетевых заведениях - ', df_name_mean_c['number'].median(), 'ед.')
```

```
Среднее количество сетевых заведений -  3.0 ед.
Среднее количество посадочных мест в сетевых заведениях -  44.5 ед.
```

**Вывод:**

- Большинство сетей имеет не более 7 - и заведений и в них до 174 посадочных мест.
- Первая четверка сетевых заведений по количеству заведений, имеет в среднем 50 посадочных мест.
- Первая пятерка сетевых заведений по количеству посадочных мест, имеет в среднем 2 сетевых заведения.
- Для сетевых заведений характерно три заведения с количеством посадочных мест 40-50. На графике это область самая темная.

## 2.5. Среднее количество посадочных мест. Какой вид предоставляет в среднем самое большое количество посадочных мест?

[к Оглавлению](#0.0)

Сделаем табличку с типами заведений и посмотрим распределение мест и количество заведений, такую же сделаем и для названий объектов

In [33]:

```
df_type_mean = (df_chain_yes.groupby('object_type').agg({'number':['mean'
],'id': ['count']}).reset_index())
df_type_mean.columns = ['object_type', 'number', 'count']
df_type_mean['number'] = df_type_mean['number'].astype('int')
df_type_mean = df_type_mean.sort_values(by='number', ascending=False)
df_type_mean['percent'] = (df_type_mean['number'] / df_type_mean['number']
.sum() * 100).round(1)
df_type_mean
```

Out[33]:

| | object_type | number | count | percent |
|---|---|---|---|---|
| **8** | столовая | 112 | 3 | 28.1 |
| **7** | ресторан | 100 | 544 | 25.1 |
| **0** | бар | 53 | 37 | 13.3 |
| **3** | кафе | 47 | 1396 | 11.8 |
| **6** | предприятие быстрого обслуживания | 40 | 791 | 10.1 |
| **2** | закусочная | 14 | 56 | 3.5 |
| **1** | буфет | 12 | 11 | 3.0 |
| **4** | кафетерий | 12 | 52 | 3.0 |
| **5** | магазин (отдел кулинарии) | 8 | 78 | 2.0 |

In [34]:

```
fig = go.Figure(data=[go.Bar(name='Количество посадочных мест', x=df_type_
mean['object_type'], y=df_type_mean['number'])])
fig.update_layout(barmode='group', title_text='Количество посадочных мест
 в сетевых заведениях по видам ООП',
                  xaxis_title="Объект общественного питания", yaxis_title=
"Количество посадочных мест, ед.",)
fig.show()
```

**Вывод:**

- Наибольшее количество посадочных мест в среднем у объекта общественного питания - **столовая**, 112 мест или 28%. В столовых низкие цены, но большая проходимость - за счет нее формируется выручка. Для большого количества посетителей нужно много мест.
- На втором месте - **ресторан**, 100 мест или 25%. Здесь средний чек гораздо выше столовой, поэтому выручка не страдает от меньшего количества мест. Тем более, что чем меньше мест в ресторане - тем предполагается выше сервис, а следовательно и выше цена.
- Третье, четвертое и пятое место у объектов общественного питания **бар** (53 места, 13%), **кафе** (47 мест, 12%) и **предприятие быстрого обслуживания** (40 мест, 10%) - самые демократичные и по ценам, и по качеству заведения.
- Меньше всего посадочных мест в **закусочных**, **буфетах** и **кафетериях** - не более 15 мест и менее 3%. Как правило эти типы заведений располагаются они обычно в аэропортах или вокзалах. Для них эта статья доходов не является основной, а закусочные и кафетерии расчитаны на другой формат обслуживания посетителей. Он подразумевает не посадочные места, а высокие стойки без стульев.
- Замыкает рейтинг видов общественного питания **отдел кулинарии в магазинах** не более 9 мест и 2%. Смылс такой же как и объктов описанных выше, для магазинов эта статья доходов не является основной.

## 2.6. Выделим в отдельный столбец информацию об улице из столбца address

[к Оглавлению](#0.0)

Нам нужно создать новый столбец с названием улиц. Применим к строкам столбца `address` метод `split` и сохраним в новом столбце.

In [35]:

```
df['street_1'] = df['address'].str.split(', ')
df.head()
```

Out[35]:

| | id | object_name | chain | object_type | address | number | |
|---|---|---|---|---|---|---|---|
| **0** | 151635 | СМЕТАНА | несетевое заведение | кафе | город Москва, улица Егора Абакумова, дом 9 | 48 | [гор<br>ул<br>Абаку |
| **1** | 77874 | Родник | несетевое заведение | кафе | город Москва, улица Талалихина, дом 2/1, корпус 1 | 35 | [гор<br>Та<br>дом |
| **2** | 24309 | Кафе «Академия» | несетевое заведение | кафе | город Москва, Абельмановская улица, дом 6 | 95 | [гор<br>Абель<br>ули |
| **3** | 21894 | ПИЦЦЕТОРИЯ | сетевое заведение | кафе | город Москва, Абрамцевская улица, дом 1 | 40 | [гор<br>Абр<br>ули |
| **4** | 119365 | Кафе «Вишневая метель» | несетевое заведение | кафе | город Москва, Абрамцевская улица, дом 9, корпус 1 | 50 | [гор<br>Абр<br>ули |

Изучим какие есть характеристики улиц, для подбора нужных слов.

In [36]:

```
df['address'] = df.address.apply(lambda x: x[0:-1].split(',')[0]) # Переве
дем столбец из строк в списки в 0 элементе

street_names = pd.DataFrame(df['address']) # создадим DF
street_names = street_names.explode('address') # Разобъем наши строки на с
лова

street_names_count = street_names['address'].value_counts() # Посчтитаем к
оличество уникальных слов
street_names_count = pd.DataFrame(list(street_names_count.items()), column
s=['name', 'count']) # создадим DF и посмотрим какие есть улицы
street_names_count.head(5)
```

Out[36]:

| | name | count |
|---|---|---|
| **0** | город Москва | 15295 |
| **1** | улица Ленинская Слобода | 4 |
| **2** | Николоямская улица | 4 |
| **3** | город Московский | 3 |

| | | |
|---|---|---|
| **4** | Солянский тупик | 3 |

Применим функцию, которая ищет нужные нам слова с названием улиц и записывает их в столбец `street`

In [37]:

```
street_variant = ['улица', 'переулок', 'площадь', 'проезд', 'проспект', 'н
абережная', 'шоссе', 'бульвар', 'аллея']
def get_street(row):
    for i in street_variant:
        if i in row['street_1'][0]:
            return row['street_1'][0]
        elif i in row['street_1'][1]:
            return row['street_1'][1]
        try:
            if i in row['street_1'][2]:
                return row['street_1'][2]
        except: pass
        try:
            if i in row['street_1'][3]:
                return row['street_1'][3]
        except: pass

df['street'] = df.apply(get_street, axis = 1)
df = df.drop(['street_1'], axis=1) # удалим не нужный столбец
df.head()
```

Out[37]:

| | id | object_name | chain | object_type | address | number | stre |
|---|---|---|---|---|---|---|---|
| **0** | 151635 | СМЕТАНА | несетевое заведение | кафе | город Москва | 48 | улица Его Абакумо |
| **1** | 77874 | Родник | несетевое заведение | кафе | город Москва | 35 | ули Талалихи |
| **2** | 24309 | Кафе «Академия» | несетевое заведение | кафе | город Москва | 95 | Абельмановск ули |
| **3** | 21894 | ПИЦЦЕТОРИЯ | сетевое заведение | кафе | город Москва | 40 | Абрамцевск ули |
| **4** | 119365 | Кафе «Вишневая метель» | несетевое заведение | кафе | город Москва | 50 | Абрамцевск ули |

Посмотрим сколько заведений не имеют свого адреса.

In [38]:

```
df['street'].isnull().sum()
```

Out[38]:

```
421
```

In [39]:

```
df.query('street.isnull()')['address'].value_counts()
```

Out[39]:

```
город Москва                        413
Солянский тупик                       3
поселение "Мосрентген"                1
поселение Михайлово-Ярцевское         1
Нижний Таганский тупик                1
поселение Сосенское                   1
поселение Марушкинское                1
Name: address, dtype: int64
```

Мы не смогли заполнить 421 строчку адресами. В адресе указан только город или поселение без уточнения адреса.

In [40]:

```
df_name_street = (df.groupby('street').agg({'id': ['count']}).reset_index
())
df_name_street.columns = ['street', 'count']
df_name_street['percent'] = (df_name_street['count'] / df_name_street['cou
nt'].sum() * 100).round(2)
df_name_street = df_name_street.sort_values(by='count', ascending=False)
df_name_street.head(10)
```

Out[40]:

| | street | count | percent |
|---|---|---|---|
| **1405** | проспект Мира | 204 | 1.37 |
| **1002** | Профсоюзная улица | 183 | 1.22 |
| **676** | Ленинградский проспект | 173 | 1.16 |
| **986** | Пресненская набережная | 167 | 1.12 |
| **390** | Варшавское шоссе | 165 | 1.10 |
| **679** | Ленинский проспект | 148 | 0.99 |
| **1401** | проспект Вернадского | 132 | 0.88 |
| **666** | Кутузовский проспект | 114 | 0.76 |
| **590** | Каширское шоссе | 112 | 0.75 |
| **597** | Кировоградская улица | 110 | 0.74 |

In [41]:

```
print('Топ 10 улиц с наибольшим количеством заведений, занимают всего - ',
df_name_street['percent'].head(10).sum(), '% от всех заведений Москвы')
print('Количество уникальных названий улиц с наибольшим количеством заведе
ний - ', len(df_name_street['percent']), 'ед.')
```

```
Топ 10 улиц с наибольшим количеством заведений, занимают всег
о -  10.09 % от всех заведений Москвы
Количество уникальных названий улиц с наибольшим количеством
заведений -  1884 ед.
```

## 2.7. Топ-10 улиц по количеству объектов общественного питания. В каких районах Москвы находятся эти улицы?

[к Оглавлению](#0.0)

**Узнаем в каких районах Москвы находятся эти улицы, а потом посчитаем топ 10 этих улиц**

Создадим отдельную табличку, где разместим улицы и их `id`. К ним применим функцию для поиска районов
Для уменьшения времени обработки данных при запрос, сделаем срез на топ 10 улиц по количеству заведений и улицы с 1 заведением

In [42]:

```
df_streets = (df.groupby('street').agg({'id': ['count']}).reset_index())
df_streets.columns = ['street', 'count']
df_streets = df_streets.query('count == 1 | count > 100').sort_values(by=
'count', ascending=False)
df_streets.head(15)
```

Out[42]:

| | street | count |
|---|---|---|
| **1405** | проспект Мира | 204 |
| **1002** | Профсоюзная улица | 183 |
| **676** | Ленинградский проспект | 173 |
| **986** | Пресненская набережная | 167 |
| **390** | Варшавское шоссе | 165 |
| **679** | Ленинский проспект | 148 |
| **1401** | проспект Вернадского | 132 |
| **666** | Кутузовский проспект | 114 |
| **590** | Каширское шоссе | 112 |
| **597** | Кировоградская улица | 110 |
| **1268** | Ходынский бульвар | 102 |
| **1160** | Стройковская улица | 1 |
| **1156** | Стрельбищенский переулок | 1 |
| **1143** | Старомарьинское шоссе | 1 |
| **1169** | Сумская улица | 1 |

Сделаем запрос через к Сервис Яндекс.Карты к HTTP API Геокодер с поиском по адресу. Геокодер API позволяет определять координаты топонима по его адресу, или адрес точки по её координатам.

Найдем координаты по адресу и подставим в таблицу

In [43]:

```
# def get_coords(address):
#     r = requests.get("https://geocode-maps.yandex.ru/1.x/?apikey=" + API
_key + "&format=xml&geocode=Москва," + address)
#     soup=BeautifulSoup(r.text, 'lxml')
#     geo = soup.find_all('pos')
#     for row in geo:
#         return row.text

# df_streets['coords'] = df_streets['street'].apply(get_coords)
```

Проверим на отсутствующие значения.

In [44]:

```
# df_streets.isnull().sum()
```

In [45]:

```
# df_streets = df_streets.dropna()
# df_streets.info()
```

По полученым координатам найдем район, в котором находится улица.

In [46]:

```
# def get_raion(coords):
#     r = requests.get("https://geocode-maps.yandex.ru/1.x/?apikey=" + API
_key + "&format=xml&geocode=" + coords + "&kind=district")
#     soup=BeautifulSoup(r.text, 'lxml')
#     geo = soup.find_all('name')
#     wrong_lines_content = []
#     try:
#         for row in geo[5]:
#             return row
#     except:
#         for row in geo:
#             return wrong_lines_content.append(row)

# df_streets['area'] = df_streets['coords'].apply(get_raion)
```

In [47]:

```
# df_streets = df_streets.fillna(0)
# df_streets.info()
```

Сохраним полученную табличку в файл csv и будем пользоваться им.

In [48]:

```
# df_streets.to_csv('df_streets_srez.csv', sep = ';', index= False)
```

https://docs.google.com/spreadsheets/d/1qAyK_elQ_RQ8i5aXgxqx66UxjBRSaTeKdVmGksMqsU4/edi
usp=sharing

Мы загрузили таблицу с данными на Google Sheets, откуда и будем ее считывать.

In [49]:

```
from io import BytesIO
import requests
spreadsheet_id = '1qAyK_elQ_RQ8i5aXgxqx66UxjBRSaTeKdVmGksMqsU4'
file_name = 'https://docs.google.com/spreadsheets/d/{}/export?format=csv'.
format(spreadsheet_id)
r = requests.get(file_name)
df_area = pd.read_csv(BytesIO(r.content), sep = ',')
df_area.head()
```

Out[49]:

| | street | count | coords | area |
|---|---|---|---|---|
| **0** | проспект Мира | 204 | 37.637937 55.812368 | Алексеевский район |
| **1** | Профсоюзная улица | 183 | 37.532511 55.649525 | район Коньково |
| **2** | Ленинградский проспект | 173 | 37.545626 55.794285 | Хорошёвский район |
| **3** | Пресненская набережная | 167 | 37.540982 55.746791 | Пресненский район |
| **4** | Варшавское шоссе | 165 | 37.603954 55.599799 | район Чертаново Южное |

Построим таблицу и посчитаем топ-10 улиц по количеству объектов общественного питания

In [50]:

```
df_street_top = df_area.sort_values(by='count', ascending=False).head(10)
df_street_top
```

Out[50]:

| | street | count | coords | area |
|---|---|---|---|---|
| **0** | проспект Мира | 204 | 37.637937<br>55.812368 | Алексеевский район |
| **1** | Профсоюзная улица | 183 | 37.532511<br>55.649525 | район Коньково |
| **2** | Ленинградский проспект | 173 | 37.545626<br>55.794285 | Хорошёвский район |
| **3** | Пресненская набережная | 167 | 37.540982<br>55.746791 | Пресненский район |
| **4** | Варшавское шоссе | 165 | 37.603954<br>55.599799 | район Чертаново Южное |
| **5** | Ленинский проспект | 148 | 37.537137 55.68263 | Ломоносовский район |
| **6** | проспект Вернадского | 132 | 37.515335<br>55.681656 | Ломоносовский район |
| **7** | Кутузовский проспект | 114 | 37.50911 55.735013 | район Дорогомилово |
| **8** | Каширское шоссе | 112 | 37.683194<br>55.641244 | район Москворечье-Сабурово |
| **9** | Кировоградская улица | 110 | 37.605023<br>55.614455 | район Чертаново Центральное |

In [51]:

```
fig = go.Figure(data=[go.Bar(name='Количество посадочных мест', x=df_stree
t_top['street'], y=df_street_top['count'])])
fig.update_layout(barmode='group', title_text='ТОП-10 улиц по количеству о
бъектов общественного питания',
                  xaxis_title="Улица", yaxis_title="Количество заведений,
 ед.",)
fig.show()
```

**Вывод:**

- На первом месте по количеству заведений проспект Мира с 204 заведениями.
- Если смотреть по районам, то в Ломоносовский район входит две улицы и их суммарное количество заведений 280 ед.

## 2.8. Количество улиц с одним объектом общественного питания. В каких районах Москвы находятся эти улицы?

[к Оглавлению](#0.0)

Соберем таблицу по улице и сделаем срез по количеству ООП равным одному и посмотрим на

каких улицах они расположены. Посчитаем их количество.

In [52]:

```
df_one_street = df_area.query('count == 1')
df_one_street.head()
```

Out[52]:

| | street | count | coords | area |
|---|---|---|---|---|
| **11** | Стройковская улица | 1 | 37.674382 55.73371 | Таганский район |
| **12** | Стрельбищенский переулок | 1 | 37.539464 55.761001 | Пресненский район |
| **13** | Старомарьинское шоссе | 1 | 37.611994 55.803717 | район Марьина Роща |
| **14** | Сумская улица | 1 | 37.605607 55.623651 | район Чертаново Северное |
| **15** | Стромынский переулок | 1 | 37.694738 55.792245 | район Сокольники |

In [53]:

```
print('Количество улиц с одним объектом общественного питания - ', len(df_
one_street), 'ед. или', round(len(df_one_street)/len(df_name_street)*100,1
), '%')
```

```
Количество улиц с одним объектом общественного питания -  537
ед. или 28.5 %
```

Посмотрим какие районы преобладают в количестве улиц с одним заведением ООП

In [54]:

```
df_area_pivot = df_area.pivot_table(index='area', values='count', aggfunc=
'count').sort_values(by='count', ascending=False).reset_index()
df_area_pivot['percent'] = ((df_area_pivot['count']/df_area_pivot['count']
.sum())*100).round(1)
df_area_pivot.head(10)
```

Out[54]:

| | area | count | percent |
|---|---|---|---|
| **0** | Таганский район | 26 | 4.7 |
| **1** | район Хамовники | 23 | 4.2 |
| **2** | Басманный район | 23 | 4.2 |
| **3** | Тверской район | 21 | 3.8 |
| **4** | Пресненский район | 20 | 3.6 |
| **5** | район Марьина Роща | 17 | 3.1 |
| **6** | район Замоскворечье | 15 | 2.7 |
| **7** | Мещанский район | 15 | 2.7 |

| | | | |
|---|---|---|---|
| **8** | район Арбат | 12 | 2.2 |
| **9** | район Сокольники | 11 | 2.0 |

График распределений районов по количеству улиц с одним ООП

In [55]:

```
fig = go.Figure(data=[go.Bar(name='Количество улиц', x=df_area_pivot['are
a'][:30], y=df_area_pivot['count'])])
fig.update_layout(barmode='group', title_text='Распределений районов по ко
личеству улиц с одним ООП',
                  xaxis_title="Район", yaxis_title="Количество заведений,
 ед.",)
fig.show()
```

In [56]:

```
display(df_area_pivot['count'].reset_index().describe(percentiles=[0.1, 0.
27, 0.3, 0.4, 0.5, 0.6, 0.75, 0.8, 0.9, 0.99]).T[1:])
```

| | count | mean | std | min | 10% | 27% | 30% | 40% | 50% | 60% | 75% | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 116.0 | 4.724138 | 5.050331 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 | 3.0 | 4.0 | 6.0 | |

In [57]:

```
print('Количество районов с одним объектом общественного питания -', len(df_area_pivot), 'ед.')
```

```
Количество районов с одним объектом общественного питания - 1
16 ед.
```

**Вывод:**

- Количество улиц с одним объектом общественного питания - 537 ед.
- Количество районов 116 ед.
- Наибольшее количество улиц с одним заведением находится в Таганском районе - 26 ед.
- Топ 5 Районов с количестовом заведений Таганский район - 26, район Хамовники - 23, Басманный район- 23, Тверской район - 21, Пресненский район - 20
- По распределению видно, что районы с одной улицей составляют 27%. Районы с 10 и более улиц составляют 10%.

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

[к Оглавлению](#0.0)

Найдем улицы с большим количеством заведений ООП и посмотрим количество посадочных мест в них.

In [58]:

```
df_number_street = (df.groupby('street').agg({'id': ['count'], 'number':['mean']}).reset_index())
df_number_street.columns = ['street', 'count', 'number']
```

In [59]:

```
display(df_number_street['count'].reset_index().describe(percentiles=[0.1, 0.5, 0.6, 0.75, 0.8, 0.9, 0.95, 0.99]).T[1:])
```

| | count | mean | std | min | 10% | 50% | 60% | 75% | 80% | 90% | 95% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 1884.0 | 7.93259 | 15.939351 | 1.0 | 1.0 | 3.0 | 4.0 | 7.0 | 9.0 | 18.0 | 29.85 |

За большое количество заведений возьмем количество более 30 заведений.

In [60]:

```
df_number_street = df_number_street.query('count >=30').round().sort_values(by='count', ascending=False)
df_number_street.head(10)
```

Out[60]:

| | street | count | number |
|---|---|---|---|
| **1405** | проспект Мира | 204 | 63.0 |

| | | | |
|---|---|---|---|
| **1002** | Профсоюзная улица | 183 | 47.0 |
| **676** | Ленинградский проспект | 173 | 52.0 |
| **986** | Пресненская набережная | 167 | 46.0 |
| **390** | Варшавское шоссе | 165 | 52.0 |
| **679** | Ленинский проспект | 148 | 63.0 |
| **1401** | проспект Вернадского | 132 | 67.0 |
| **666** | Кутузовский проспект | 114 | 85.0 |
| **590** | Каширское шоссе | 112 | 55.0 |
| **597** | Кировоградская улица | 110 | 60.0 |

In [61]:

```
print('Количество посадочных мест для улиц с большим количеством ООП по ме
диане -', df_number_street['number'].median(), 'ед.')
```

```
Количество посадочных мест для улиц с большим количеством ООП
по медиане - 50.0 ед.
```

Посмотрим как наши данные выглядят на гистограмме

In [62]:

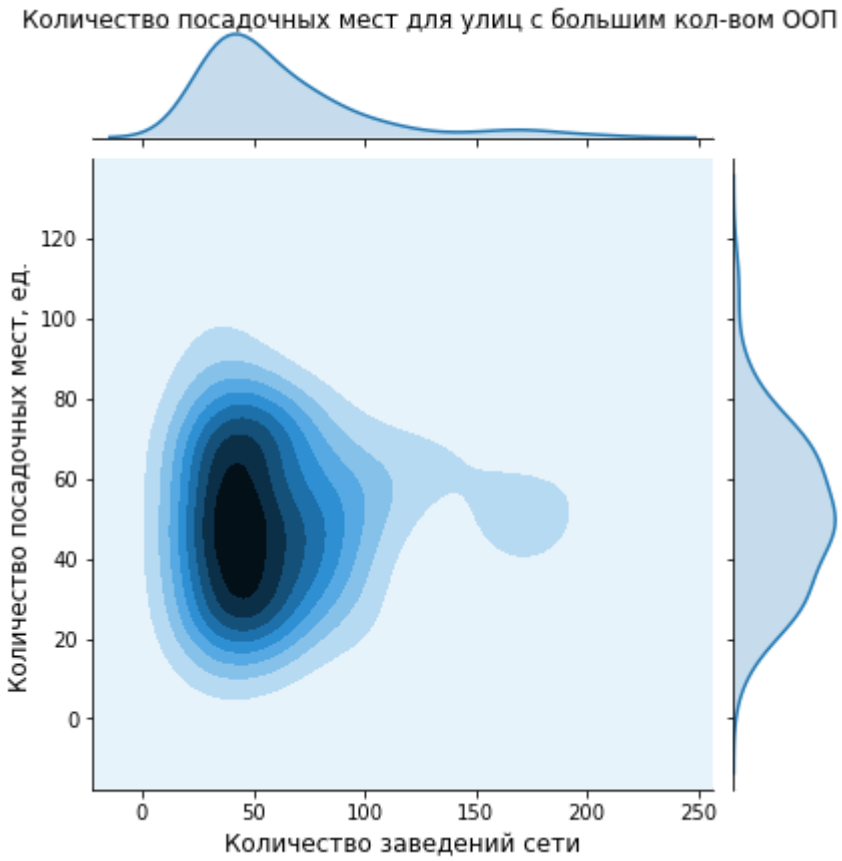
```
fig = go.Figure(data=[go.Bar(name='Количество посадочных мест', x=df_numbe
r_street['street'][:30], y=df_number_street['number']),
                      go.Bar(name='Количество заведений', x=df_number_stre
et['street'][:30], y=df_number_street['count'])])
fig.update_layout(barmode='group', title_text='Количество посадочных мест
 для улиц с большим кол-вом ООП',
                  xaxis_title="Улица", yaxis_title="Количество заведений О
ОП/посадочных мест, ед.",)
fig.show()
```

Посмотрим на то, как связаны между собой количество заведений на улице и количество мест на этих улицах.

In [63]:

```
p = sns.jointplot(x=df_number_street["count"], y=df_number_street["number"
], kind='kde')

p.fig.suptitle('Количество посадочных мест для улиц с большим кол-вом ООП'
, y = 1.0)
p.set_axis_labels("Количество заведений сети","Количество посадочных мест,
ед.", fontsize=12)
plt.show()
```




In [64]:

```
hundred_cafe_sorted = df_number_street.sort_values(by='number', ascending=
False)
fig = px.scatter(hundred_cafe_sorted, x="street", y="number", size="count"
, hover_name="street")
fig.update_layout(title_text='Количество посадочных мест в заведениях на у
лицах с большим количеством ООП',
                  xaxis_title="Улица", yaxis_title="Количество посадочных
```

```
 мест, ед.",)
fig.show()
```

**Вывод:**

- Улицы с наибольшим количеством заведений имеют от 40 до 70 посадочных мест.
- Показатели количества мест в заведениях нормально распределены вокруг медианного значения - 50 посадочных мест.

## 3. Общий вывод

[к Оглавлению](#0.0)

- Количество всех заведений общественного питания в Москве 15 366 ед. из них Кафе - 40% , Столовые - 17%, Рестораны - 15%, Предприятия быстрого обслуживани - 12,5%, Бары - 6%.
- Из всех заведений ОП только 2968 ед. являются сетевыми или 19%.
- Сетевое распределение характерно для Кафе - 1396 ед. 47%, Предприятий быстрого питания 791 ед. 26%, ресторанов и отделов кулинарии 544 ед. 18%.
- Для ресторанов характерно больше мест (в среднем 97), но меньше заведений. В кафе и предприятиях быстрого обслуживания (40 и 20 мест в среденем) все наоборот: меньше мест, но больше точек.

- Самое большое среднее число мест в столовых - 112
- Топ 10 улиц с наибольшим количеством заведений, занимают всего - 10 % от количества всех заведений Москвы.
- Количество уникальных названий улиц с наибольшим количеством заведений - 1884 ед.
- Среднее число посадочных мест в заведениях общественного питания на улицах Москвы с высокой плотностью кафе и ресторанов составляет от 40 до 70 посадочных мест.

Подавляющее большинство улиц с одним кафе или рестораном - небольшие переулки Москвы или один из километров МКАД, проходящие вдоль одного района. Проходимость и/или популярность этих улиц у гурманов минимальны, поэтому открывать там новое кафе нерентабельно.

Можно рекомендовать к открытию кафе с количеством 40-50 посадочных мест вдоль одного из шоссе или проспекта Москвы с высокой проходимостью и транспортной доступностью. Возможность развития сети кофеен пока не рекомендуется. Если и рассматривать сетевое развитие - то в долгосрочной перспективе, после выхода кафе на плато по прибыли.

Дополнительно необходимо было бы провести анализ по расположению мест скопления большого количества людей, кто регулярно покупает бизнес-ланчи - ВУЗы, крупные предприятия, бизнес-центры. Желательно учитывать по текущим объектам расстояния до метро, какое метро, средние чеки в заведениях. (посмотреть зависимость среднего чека от типов объекта, расстояния от метро, расположениях в топовых районах). Желательно выделить наиболеее интересные здания для размещения заведений - высотки, исторические здания, топовые отели и гостиницы. Плюс желательно подумать об иностранных туристах заранее, ведь можно примерно сказать, где они проводят большую часть времени в Москве.

## 4. Презентация

[к Оглавлению](#0.0)

Презентация - https://yadi.sk/i/Oi3NUmBFByuMaw