Рынок заведений общественного питания Москвы

Исследования рынка общественного питания Москвы на основе открытых данных.

Работа состоит из нескольких основных пунктов

1. Подготовка и преобразование данных.
2. Соотношение видов объектов общественного питания по количеству.
3. Исследование соотношений сетевых и несетевых заведений по количеству.
4. Характерные виды объектов общественного питания для сетевых объектов.
5. Характеристика сетевых заведений в разрезе посадочных мест.
6. На каких улицах Москвы располагаются заведения.
7. Топ 10 улиц и районов Москвы с большим числом точек общественного питания
8. Количество улиц с одним объектом общественного питания. Районы с этими улицам и
9. Распределение количества посадочных мест для улиц с большим количеством объек тов общественного питания.

Для работы над проектом использовались данные из открытых источников.

Итогом работы считается определение основных характеристик объектов общественного питания города Москва и разработка рекомендаций для открытия нового объекта.

Файл был заменен на найденый в открытых источниках на "Портале открытых данных Правительства Москвы" по ссылке (https://data.mos.ru/opendata/7710881420-obshchestvennoe-pitanie-v-moskve/data/table?versionNumber=1&releaseNumber=15)

Ссылка на презентацию в формате pdf по ссылке (https://)

Подготовка и преобразование данных

In [17]:

```
# подгрузим нужные нам библиотеки
import pandas as pd
import datetime as dt
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:

```
# так как у нас исследование по открытым данным то я нашел нормальный файл а не ваш вот этот:))
# Спасибо, что я очень интересно провел один день, когда искал соответствия райнов и ул иц. Дошел до КЛАДРа и кодов ОКАТО
# (кстати КЛАДР дал только 50% данных по районам)
```

In [267]:

```
url_file = 'https://github.com/DmitryShinkarev/Project_8/blob/master/moscow_rest.xlsx?raw=true'
```

In [19]:

```
# Подгружаю файл найденый в открытых источниках, который копирует файл задания но содержит полную информацию
#df = pd.read_excel('moscow_rest.xlsx')
df = pd.read_excel(url_file)
```

In [20]:

```
df.head()
```

Out[20]:

	ID	Name	global_id	IsNetObject	OperatingCompany	TypeObject	
0	151635	СМЕТАНА	637376221	нет	NaN	кафе	Северо-В администрат
1	77874	Родник	637376331	нет	NaN	кафе	Цен администрат
2	24309	Кафе «Академия»	637376349	нет	NaN	кафе	Цен администрат
3	21894	ПИЦЦЕТОРИЯ	637376381	да	Пиццетория	кафе	Северо-В администрат
4	119365	Кафе «Вишневая метель»	637376403	нет	NaN	кафе	Северо-В администрат

In [21]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15366 entries, 0 to 15365
Data columns (total 15 columns):
ID                  15366 non-null int64
Name                15366 non-null object
global_id           15366 non-null int64
IsNetObject         15366 non-null object
OperatingCompany    2968 non-null object
TypeObject          15366 non-null object
AdmArea             15366 non-null object
District            15366 non-null object
Address             15366 non-null object
PublicPhone         15366 non-null object
SeatsCount          15366 non-null int64
SocialPrivileges    15366 non-null object
Longitude_WGS84     15366 non-null float64
Latitude_WGS84      15366 non-null float64
geoData             15366 non-null object
dtypes: float64(2), int64(3), object(10)
memory usage: 1.8+ MB
```

In [22]:

```
df.columns
```

Out[22]:

```
Index(['ID', 'Name', 'global_id', 'IsNetObject', 'OperatingCompany',
       'TypeObject', 'AdmArea', 'District', 'Address', 'PublicPhone',
       'SeatsCount', 'SocialPrivileges', 'Longitude_WGS84', 'Latitude_WGS8
4',
       'geoData'],
      dtype='object')
```

In [23]:

```
# Изменим название столбцов
df.columns = ['id', 'name', 'global_id', 'is_net_object', 'operating_company',
       'type_object', 'adm_area', 'district', 'address', 'public_phone',
       'seats_count', 'social_privileges', 'longitude_WGS84', 'latitude_WGS84',
       'geo_data']
```

In [24]:

```
# Удалим дубликаты, хотя это наверное лишнее так как данные качественные
df.drop_duplicates();
```

In [25]:

```
# Проверим таблицу на наличе null в столбцах
df.isnull().sum()
```

Out[25]:

```
id                      0
name                    0
global_id               0
is_net_object           0
operating_company    12398
type_object             0
adm_area                0
district                0
address                 0
public_phone            0
seats_count             0
social_privileges       0
longitude_WGS84         0
latitude_WGS84          0
geo_data                0
dtype: int64
```

In [26]:

```
# заполним все значения null  в столбике с названием сетовой компании на значенение sel
f
df['operating_company'] = df['operating_company'].fillna('self')
```

In [244]:

```
# Для удобства расчетов добавим столбик с значением 1 для группировки и консолидации
df['agregate'] = 1
```

In [245]:

```
# Таблица удовлетворяет потребностям дальнейшего исследования
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15366 entries, 0 to 15365
Data columns (total 17 columns):
id                   15366 non-null int64
name                 15366 non-null object
global_id            15366 non-null int64
is_net_object        15366 non-null object
operating_company    15366 non-null object
type_object          15366 non-null object
adm_area             15366 non-null object
district             15366 non-null object
address              15366 non-null object
public_phone         15366 non-null object
seats_count          15366 non-null int64
social_privileges    15366 non-null object
longitude_WGS84      15366 non-null float64
latitude_WGS84       15366 non-null float64
geo_data             15366 non-null object
agregate             15366 non-null int64
street               15366 non-null object
dtypes: float64(2), int64(4), object(11)
memory usage: 2.0+ MB
```

Вывод: Файл содержит исчерпывающую информацию по объектам общественного питания в том числе геоданные, названия районов, название сети и является ли ресторан сетевым. Столбцы от исходного файла были переименованы в нижний регистр, но названия столбцов изменены не были. Значения nan в столбце сетевого имени ресторана на значение "self". Добавил колонку `agregate` для количественных функций группировки и подсчета.

Соотношение видов объектов общественного питания по количеству

В данном исследование возьмем количественный разрез по району, количеству сетевых ресторанов, классификации места (ресторан, кафе)

In [248]:

```
# Рассмотрим какие виды объектов вообще есть
df['type_object'].unique()
```

Out[248]:

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

In [249]:

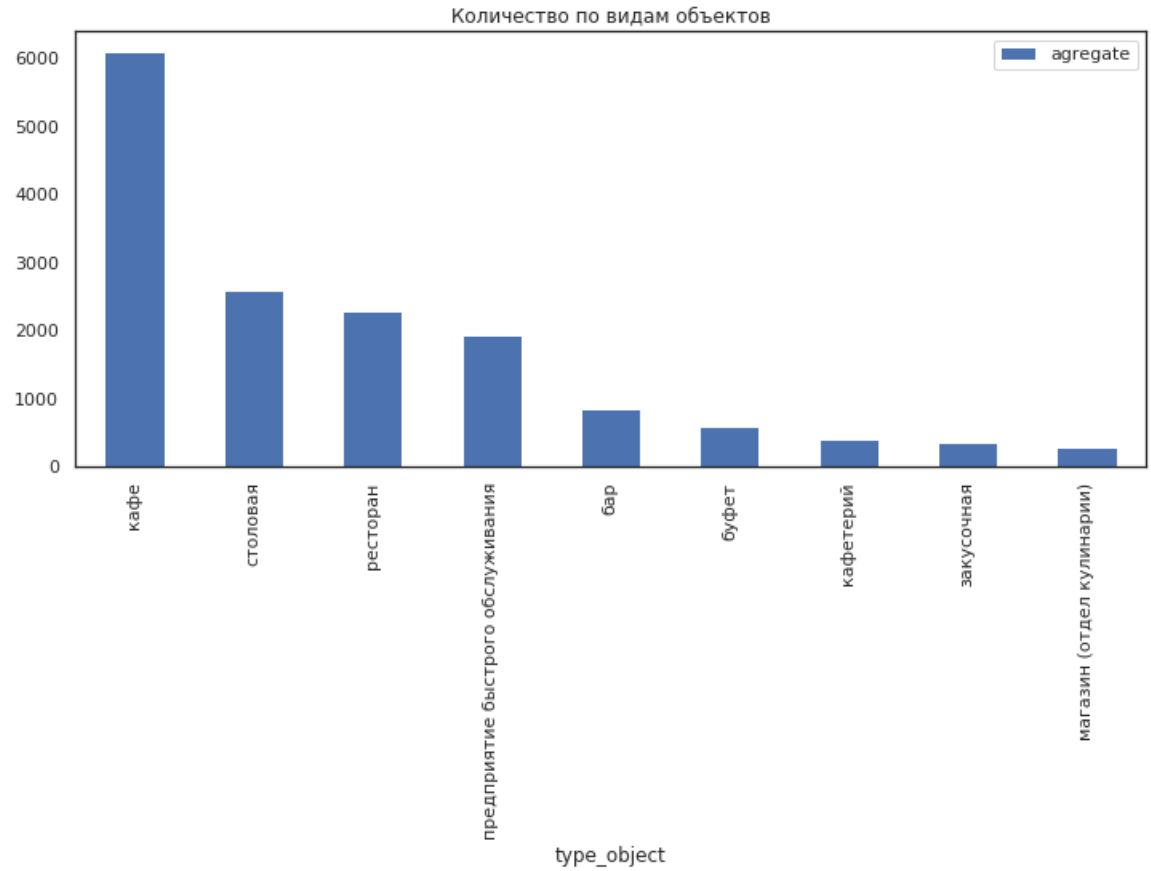
```
# Сгруппируем данные по виду объекта
df_type_object = df.groupby('type_object')['agregate'].sum()
df_type_object = df_type_object.reset_index().sort_values('agregate',ascending=False);
df_type_object
```

Out[249]:

	type_object	agregate
3	кафе	6099
8	столовая	2587
7	ресторан	2285
6	предприятие быстрого обслуживания	1923
0	бар	856
1	буфет	585
4	кафетерий	398
2	закусочная	360
5	магазин (отдел кулинарии)	273

In [250]:

```
# Представим данные в виде столбчатой диаграммы для наглядности
df_type_object = df_type_object.pivot_table(values='agregate', index='type_object', aggfunc='sum')
df_type_object = df_type_object.sort_values('agregate', ascending = False)
title_n = 'Количество по видам объектов'
df_type_object.plot.bar(figsize=(12, 5), title = title_n);
```



Вывод: Самым распространненым видом объектом общественного питания является кафе с количественным весом в 6099 шт, дальше идет столовая и ресторан с похожими результатми и немного отстает "предприятия быстрого обслуживания"

- кафе (6099)
- столовая (2587)
- ресторан (2285)
- предприятие быстрого обслуживания (1923)

Исследование соотношений сетевых и несетевых заведений по количеству.

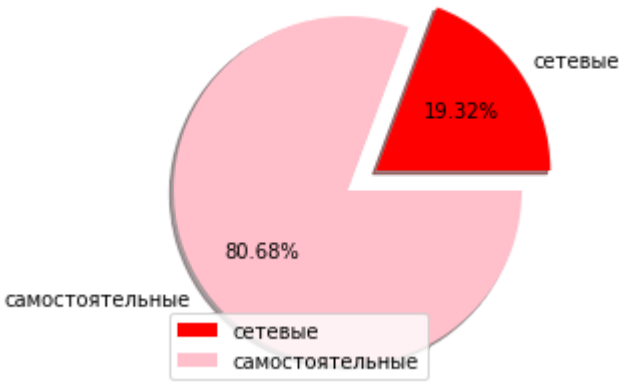
Необходимо найти долю сетевых заведений из общего числа объетков

In [32]:

```
df_net_object = df.groupby('is_net_object')['agregate'].sum()
df_net_object = df_net_object.reset_index()
df_net_object['is_net_object'] = df_net_object['is_net_object'].replace('да','сетевые')
df_net_object['is_net_object'] = df_net_object['is_net_object'].replace('нет','самостоя
тельные')
```

In [33]:

```
size = list(df_net_object['agregate'])
colors = ['red', 'pink']
labels = "сетевые", "самостоятельные"
explode = [0, 0.2]

plt.pie(size, colors = colors, labels = labels, autopct = '%.2f%%', explode = explode,
shadow = True)
plt.legend()
plt.show()
```



Вывод Как видим из общего числа заведений, сетевые составляют только 20 процентов от общего числа. Но из-ща назойливой рекламы и пестрых вывесок может сложится ощущение, что они занимают все 90 процентов рынка.

Характерные виды объектов общественного питания для сетевых объектов

Проведем характерные признаки сетевого предприятия общественного питания в разрезе видов объектов : ресторан, кафе и т.д.

In [239]:

```
#Для какого вида объекта общественного питания характерно сетевое распространение?
df_corp_type = df.pivot_table(values='agregate', index='type_object', columns='is_net_o
bject', aggfunc='sum').reset_index()
df_corp_type.columns = ['type', 'net', 'self']
df_corp_type.drop('self', axis=1)
df_corp_type = df_corp_type.sort_values('net',ascending=False)
df_corp_type
```

Out[239]:

	type	net	self
3	кафе	1396	4703
6	предприятие быстрого обслуживания	791	1132
7	ресторан	544	1741
5	магазин (отдел кулинарии)	78	195
2	закусочная	56	304
4	кафетерий	52	346
0	бар	37	819
1	буфет	11	574
8	столовая	3	2584

In [240]:

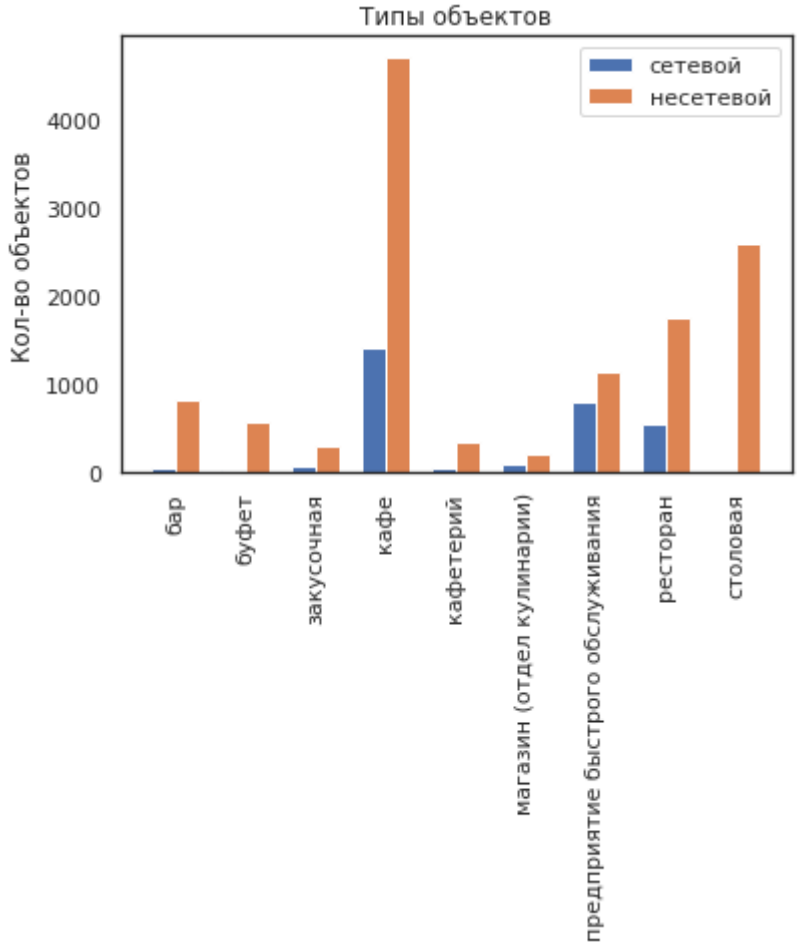
```
#Для какого вида объекта общественного питания характерно сетевое распространение?
df_corp_type_x = df.pivot_table(values='agregate', index='is_net_object', columns='type
_object', aggfunc='sum').reset_index()
```

In [241]:

```
# Функция делает из массива список для посроения графика
def creat_array(row_array):

    col = []

    for i in row_array:

        if i == 'is_net_object' or i == 'да' or i == 'нет':
            continue
        else:
            col.append(i)
    return col
```

In [243]:

```
# Воспользуемся библиотекой matplotlib для построения графика весов
labels = creat_array(df_corp_type_x.columns)
men_means = creat_array(np.array(df_corp_type_x[:1])[0])
women_means = creat_array(np.array(df_corp_type_x[1:])[0])

x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, men_means, width, label='сетевой')
rects2 = ax.bar(x + width/2, women_means, width, label='несетевой')


# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Кол-во объектов')
ax.set_title('Типы объектов')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')


#autolabel(rects1)
#autolabel(rects2)

#fig.tight_layout()

plt.xticks(rotation=90)
plt.show();
```


Типы объектов
сетевой
несетевой
Кол-во объектов
0
1000
2000
3000
4000
бар
буфет
закусочная
кафе
кафетерий
магазин (отдел кулинарии)
предприятие быстрого обслуживания
ресторан
столовая

Вывод : График не сильно информативен в угоду размерности всех столбиков, но наи интересно только три вида объектов это кафе, ресторан и предприятия быстрого питания. Кафе вообще больше всего как представителей общественного питания. Видно, что сетевой бизнес здесь имеет в три раза меньше точек чем самостоятельные кафе но всё ровно этот показатель лидирующий для обеих категорий. Самое инетересное, что самый большой противовес сетевые кафе состояляют в разделе "предприятия быстрого питания" и почти, что догоняют самостоятельные заведения. Третья категория это ресторан соответсвенно и совсем представители сетевого бизнеса не представлены в разделе "столовая" наверное потому что просто по определению

Характеристика сетевых заведений в разрезе посадочных мест

In [251]:

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

In [252]:

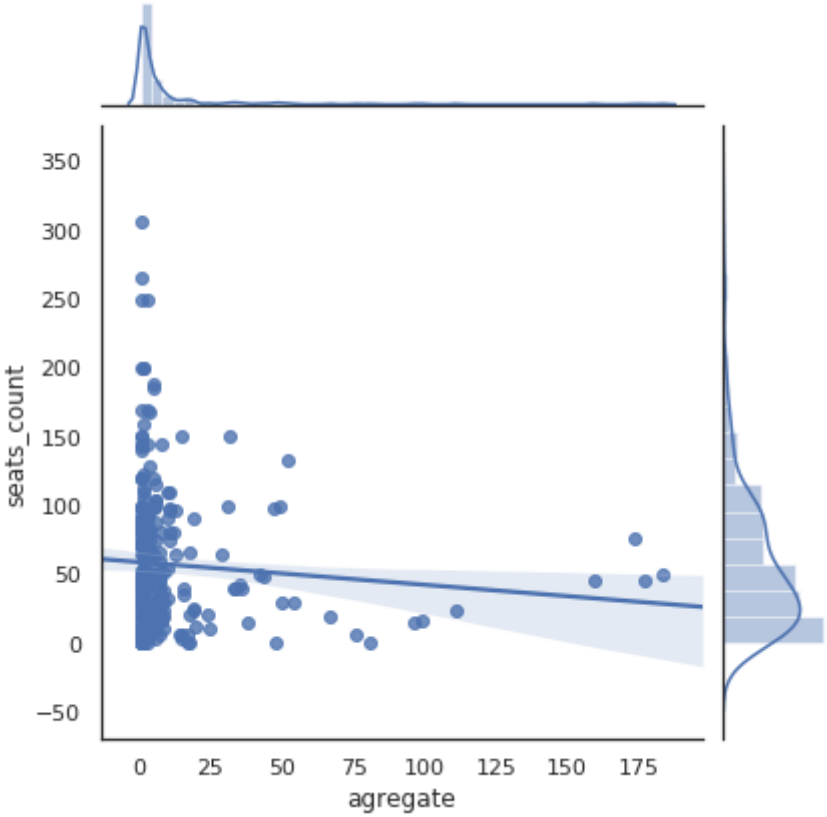
```
df_filt = df.query('is_net_object == "да"')
df_character_net = df_filt.groupby('operating_company')\
                          .agg({'agregate': 'sum', 'seats_count': 'median'}).reset_in
dex()
df_character_net = df_character_net.sort_values('agregate',ascending=False).reset_index
()
df_character_net.head()
```

Out[252]:

	index	operating_company	agregate	seats_count
0	272	Шоколадница	183	50.0
1	22	KFC	177	46.0
2	159	Макдоналдс	173	76.0
3	73	Бургер Кинг	159	45.0
4	246	Теремок	111	24.0

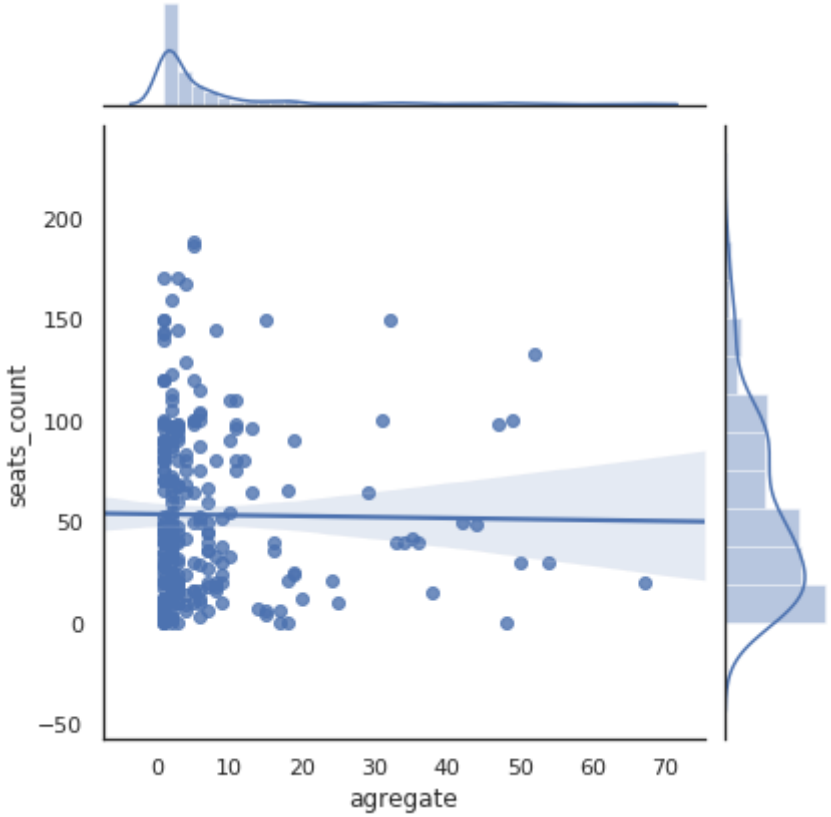
In [253]:

```
count_rest = df_character_net['agregate']
median_seats = df_character_net['seats_count']
sns.set(style="white", color_codes=True)
g = sns.jointplot(x=count_rest, y=median_seats, data=df_character_net, kind='reg')
plt.show()
```



In [254]:

```
df_character_net_filt = df_character_net.query('agregate < 75 & seats_count <200')
count_rest = df_character_net_filt['agregate']
median_seats = df_character_net_filt['seats_count']
sns.set(style="white", color_codes=True)
g = sns.jointplot(x=count_rest, y=median_seats, data=df_character_net_filt, kind='reg')
plt.show()
```



Вывод: Прямой зависимости между количеством посадочным мест в заведение и количеством сетевых объектов не прослеживаестся. Даже если убрать выбросы видим, что большее количество находится в квадрате от 0 до 10 заведений и от 0 до 100 мест. Остальные представители скорее исключение.

На каких улицах Москвы распологаются заведения

In [196]:

```
# Выделите в отдельный столбец информацию об улице из столбца address .
def find_street_df(row):

    adr = row['address']

    list_st = ['улица', 'проезд', 'бульвар', 'шоссе', 'переулок', 'проспект', 'квартал'
, 'тупик', 'набережная', 'площадь'\
               , 'аллея', 'линия', 'парк', 'сквер']

    f = adr.split(',')
    for i in f:
        for r in list_st:
            if i.find(r) != -1:
                #i = i.replace(r,'')
                return i.strip()
```

In [197]:

```
df['street'] = df.apply(find_street_df, axis=1)
```

In [198]:

```
# Не получилось определить около 400 улиц которые и улицами не считаются такие как квар
талы г. Зеленоград
df['street'] = df['street'].fillna('unknown')
```

In [256]:

```
df_not_null = df.query('street != "unknown"')
df_not_null.head()
```

Out[256]:

	id	name	global_id	is_net_object	operating_company	type_object	
0	151635	СМЕТАНА	637376221	нет	self	кафе	Северо админи
1	77874	Родник	637376331	нет	self	кафе	Ц админи
2	24309	Кафе «Академия»	637376349	нет	self	кафе	Ц админи
3	21894	ПИЦЦЕТОРИЯ	637376381	да	Пиццетория	кафе	Северо админи
4	119365	Кафе «Вишневая метель»	637376403	нет	self	кафе	Северо админи

Вывод: Были вычеслены улицы из полного адреса. Неполучилось определить улицы города Зеленоград который является частью Москвы и застраивался экспериментальной квартальной застройкой и неуоторые части МКАД.

Топ 10 улиц и районов Москвы с большим числом точек общественного питания

Агрегируем количество ресторанов в пределах одной улицы и выведем первую 10

In [201]:

```
df_top10_street = df_not_null.groupby('street')['agregate'].sum().reset_index()\
                                    .sort_values('agregate',ascending=False).head(10)
df_top10_street
```

Out[201]:

	street	agregate
1425	проспект Мира	204
1013	Профсоюзная улица	183
686	Ленинградский проспект	173
997	Пресненская набережная	167
399	Варшавское шоссе	165
689	Ленинский проспект	148
1421	проспект Вернадского	132
676	Кутузовский проспект	114
599	Каширское шоссе	112
606	Кировоградская улица	110

Вывод: Это не стало удивлением корреляция на лицо, чем длинее улица и ближе к центру тем больше вдоль нее находится точек общественного питания. Рекодтсмен Проспект Мира с 204 заведениями а дальше идет Профсоюзная улица и Ленинградский проспект.

Количество улиц с одним объектом общественного питания. Районы с этими улицами

Выбрать общее количество улиц на которых находится не больше одного объекта общественного питания выяснить районы в которых они распологаются.

In [210]:

```
pd.options.display.max_colwidth = 100

df_down_10_street = df_not_null.groupby('street')['agregate'].sum().reset_index()\
                                    .sort_values('agregate',ascending=False)


count_str = df_down_10_street.query('agregate == 1').count()
count_str_wthout_rest = count_str[0]
print(f'Количество улиц с одним объектом общественного питания {count_str_wthout_rest}'
)
```

```
Количество улиц с одним объектом общественного питания 551
```

In [204]:

```
# Найдите число улиц с одним объектом общественного питания. Воспользуйтесь внешней инф
ормацией и
# ответьте на вопрос – в каких районах Москвы находятся эти улицы?

pd.options.display.max_colwidth = 100

df_down_10_street = df_not_null.groupby('street')['agregate'].sum().reset_index()\
                                    .sort_values('agregate',ascending=False).tail(10)

df_district_one = df_down_10_street.merge(df, on = 'street', how = 'left')
df_district_one[['district','street']]
```

Out[204]:

	district	street
0	район Южное Тушино	Штурвальная улица
1	район Арбат	Шубинский переулок
2	район Северное Измайлово	Щёлковский проезд
3	Пресненский район	Электрический переулок
4	Мещанский район	Звонарский переулок
5	Рязанский район	Зарайская улица
6	район Измайлово	Заводской проезд
7	Даниловский район	Жуков проезд
8	район Отрадное	Юрловский проезд
9	район Лефортово	Перовский проезд

Вывод: Всего улиц с одним объектом 551. В 10 районов в которых ниболшее число улиц с одним рестораном находится в районах Южное Тушино, Район Арбат, Серерное измайлово, Пресненский район. Если с первый все более менее понятно, это спальный район Москвы то с Арбатом не так всё ясно, наверное из за того что между новым и старым арбтом находится полно небольших переулков.

Чаще всего это не совсем улицы а переулки, тупики или проезды.

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

Решил собрать все данные которые хоть как-то характирзовали заведения в зависимости от количества мест. Разобьем таблицу на улицы и найем медиану, среднию, кол-во заведений без посадочных мест, колчичество заведений с посадкой а так же разобъем все по категориям и построим корелляционую матрицу.

In [231]:

```
# Посмотрите на распределение количества посадочных мест для улиц с большим количеством
объектов
# общественного питания. Какие закономерности можно выявить?
df_test1 = df_not_null.groupby('street').agg({'agregate': 'sum', 'seats_count': 'sum'})
.reset_index()\
                                                                .sort_values('agregate',asce
nding=False).head(100)
```

In [232]:

```
df_sits_top_streets_0 =  df_not_null.query('seats_count == 0')
df_sits_top_streets_1 =  df_not_null.query('seats_count != 0')

df_sits_top_streets_0 = df_sits_top_streets_0.groupby('street')['agregate'].sum().reset
_index()
df_sits_top_streets_1 = df_sits_top_streets_1.groupby('street')['agregate'].sum().reset
_index()

df_test1 = df_test1.merge(df_sits_top_streets_0, on = 'street', how = 'left')
df_test1 = df_test1.merge(df_sits_top_streets_1, on = 'street', how = 'left')

df_test1.columns = ['street','count_obj','sum_seats_count','without_seats','with_seats'
]
```

In [213]:

```
df_sits_mean = df_not_null.groupby('street')['seats_count'].mean().round().reset_index
()
df_test1 = df_test1.merge(df_sits_mean, on = 'street', how = 'left')
```

In [214]:

```
df_sits_median = df_not_null.groupby('street')['seats_count'].median().round().reset_in
dex()
df_test1 = df_test1.merge(df_sits_median, on = 'street', how = 'left')
#df_test1 = df_test1.drop('seats_count_y' ,axis = 1)
df_test1 = df_test1.rename({'seats_count_x': 'mean_seats', 'seats_count_y': 'median_sea
ts'}, axis=1)
df_test1 = df_test1.fillna(0)
df_test1.head()
```

Out[214]:

	street	count_obj	sum_seats_count	without_seats	with_seats	mean_seats	medi
0	проспект Мира	204	12790	12.0	192	63.0	
1	Профсоюзная улица	183	8667	40.0	143	47.0	
2	Ленинградский проспект	173	9042	13.0	160	52.0	
3	Пресненская набережная	167	7656	11.0	156	46.0	
4	Варшавское шоссе	165	8626	21.0	144	52.0	

In [215]:

```
df_test1['without_seats'] = df_test1['without_seats'].astype('int')
df_test1['mean_seats'] = df_test1['mean_seats'].astype('int')
df_test1['median_seats'] = df_test1['median_seats'].astype('int')
```

In [216]:

```
df_type_obj = df_not_null.pivot_table(values='seats_count', index='street', columns='ty
pe_object', aggfunc='sum').reset_index()
df_type_obj_top = df_test1.merge(df_type_obj,  on = 'street', how = 'left')
df_type_obj_top = df_type_obj_top.fillna(0)
```

In [217]:

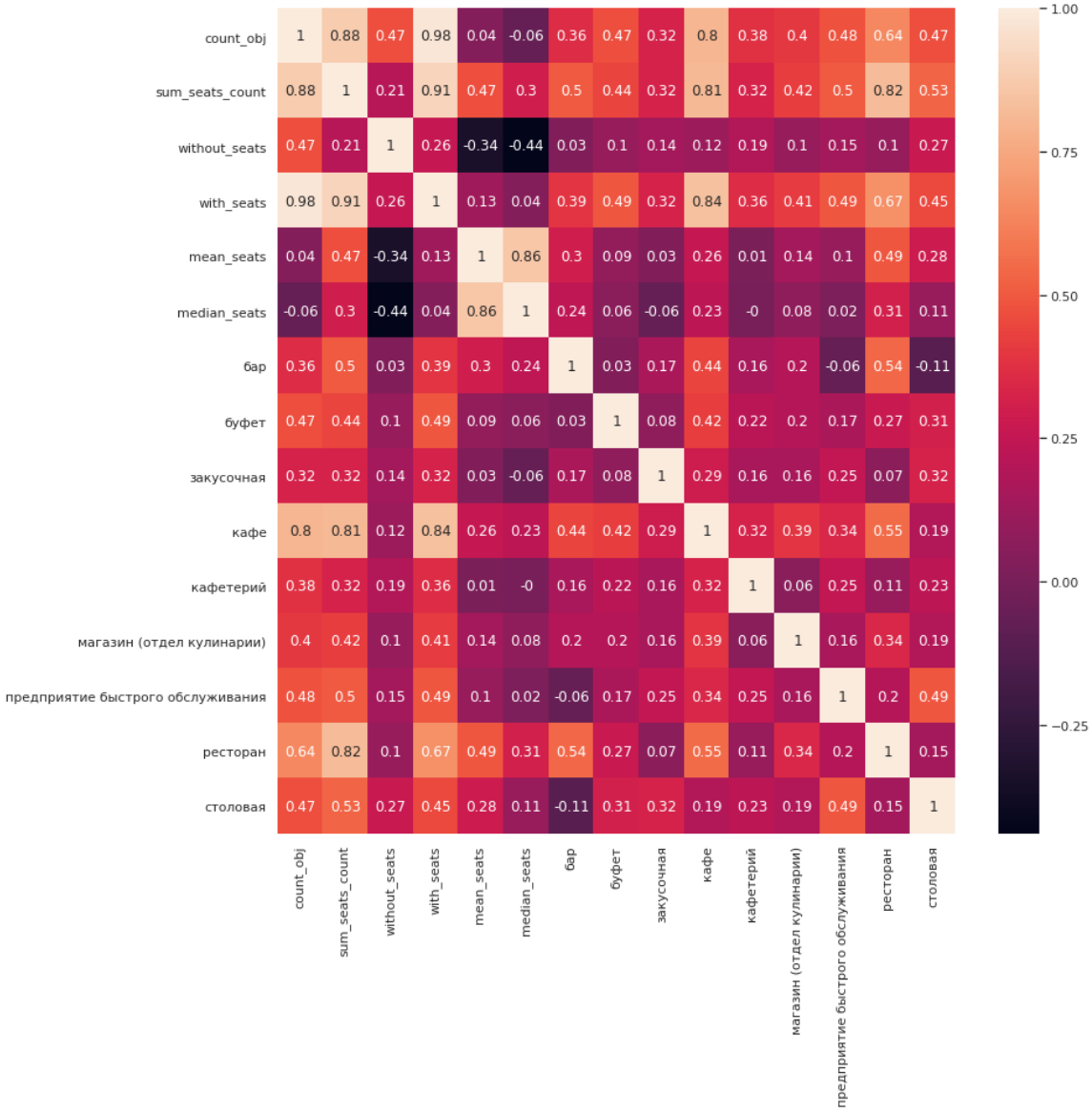
```
df_type_obj_top.head()
```

Out[217]:

	street	count_obj	sum_seats_count	without_seats	with_seats	mean_seats	medi
0	проспект Мира	204	12790	12	192	63	
1	Профсоюзная улица	183	8667	40	143	47	
2	Ленинградский проспект	173	9042	13	160	52	
3	Пресненская набережная	167	7656	11	156	46	
4	Варшавское шоссе	165	8626	21	144	52	

In [218]:

```
f, ax = plt.subplots(figsize=(13, 13))
correlation_matrix = df_type_obj_top.corr(method ='pearson').round(2)
sns.heatmap(data=correlation_matrix, annot=True)
plt.show()
```


count_obj 1 0.88 0.47 0.98 0.04 -0.06 0.36 0.47 0.32 0.8 0.38 0.4 0.48 0.64 0.47
sum_seats_count 0.88 1 0.21 0.91 0.47 0.3 0.5 0.44 0.32 0.81 0.32 0.42 0.5 0.82 0.53
without_seats 0.47 0.21 1 0.26 -0.34 -0.44 0.03 0.1 0.14 0.12 0.19 0.1 0.15 0.1 0.27
with_seats 0.98 0.91 0.26 1 0.13 0.04 0.39 0.49 0.32 0.84 0.36 0.41 0.49 0.67 0.45
mean_seats 0.04 0.47 -0.34 0.13 1 0.86 0.3 0.09 0.03 0.26 0.01 0.14 0.1 0.49 0.28
median_seats -0.06 0.3 -0.44 0.04 0.86 1 0.24 0.06 -0.06 0.23 -0 0.08 0.02 0.31 0.11
бар 0.36 0.5 0.03 0.39 0.3 0.24 1 0.03 0.17 0.44 0.16 0.2 -0.06 0.54 -0.11
буфет 0.47 0.44 0.1 0.49 0.09 0.06 0.03 1 0.08 0.42 0.22 0.2 0.17 0.27 0.31
закусочная 0.32 0.32 0.14 0.32 0.03 -0.06 0.17 0.08 1 0.29 0.16 0.16 0.25 0.07 0.32
кафе 0.8 0.81 0.12 0.84 0.26 0.23 0.44 0.42 0.29 1 0.32 0.39 0.34 0.55 0.19
кафетерий 0.38 0.32 0.19 0.36 0.01 -0 0.16 0.22 0.16 0.32 1 0.06 0.25 0.11 0.23
магазин (отдел кулинарии) 0.4 0.42 0.1 0.41 0.14 0.08 0.2 0.2 0.16 0.39 0.06 1 0.16 0.34 0.19
предприятие быстрого обслуживания 0.48 0.5 0.15 0.49 0.1 0.02 -0.06 0.17 0.25 0.34 0.25 0.16 1 0.2 0.49
ресторан 0.64 0.82 0.1 0.67 0.49 0.31 0.54 0.27 0.07 0.55 0.11 0.34 0.2 1 0.15
столовая 0.47 0.53 0.27 0.45 0.28 0.11 -0.11 0.31 0.32 0.19 0.23 0.19 0.49 0.15 1
count_obj
sum_seats_count
without_seats
with_seats
mean_seats
median_seats
бар
буфет
закусочная
кафе
кафетерий
магазин (отдел кулинарии)
предприятие быстрого обслуживания
ресторан
столовая
1.00
0.75
0.50
0.25
0.00
−0.25

Вывод: Прямых зависимостей мы не видим. Заметно, что столовая и предприятия быстрого питания чаще всего не имеют посадочных, или если сказать правильнее "в них этот показатель встречается чаще". А в магазине и отделе кулинарии чаще всего оъекты имеют посадочные места. Похожее число посадочных мест имееют рестораны, кафе и бары. Бары в свою очередь имееют обратную зависимость и по всей видимости чаще встречаются с посадочными местами.

Общий вывод

Большее количество пунктов общественного питания находятся на улицах в центре города, Больше весего по типам недвижимости кафе и ресторанов, самая большая доля представителей сейтей находятся в разделе предприятий быстрого питания и кафе и ресторанов. Зависимостей от количество ресторнов сети и количество посадочных мест не выявлено. Самое большое количество ресторанов находится на длинных улицах чаще всего проспектах только из-за того, что они протяженнее.

Лучшее место для нового ресторана это улица с большим количеством ресторнаов но не проспект. Чаще всего это улицы которые из себя представляют место где полюбому можно найти хороший/ хорошое кафе/ресторан и свободное место. Если точку поставить на место где только одно заведения то и посетители будут случайными. В отличии от первого варианта.

In [ ]: