

# Ведение

В этом исследовании нам необходимо проанализировать имеющиеся открытык данные о заведениях общепита Москвы и выяснить какие имеются закономерности.

Важное предуведомление.

- 1. при выделении улиц из адреса и анализа распределения объектов по улицам и районам, я принял, для адресов населенных пунктов-спутников Москвы улица=район=название населенного пункта. Т.е. все объекты например из Зеленограда имеют улица - Зелиноград, район - Зелиноград.
- 2. Я совсем не доволен своей презентацией, но проблема в том, что я ненавижу такой жанр. Если бы мне надо было бы делать доклад, то на слайдах я бы оставил только графики, а весь текст читал с бумаг в руках.
- 3. презентация лежит на моем личном яндекс-диске. Надеюсь проблем с доступом не будет.

## импорт

Импортируем библиотеки и данные

```
Ввод [1]: import pandas as pd
import numpy as np
import datetime as dt
import plotly.express as px
from plotly import graph_objects as go
from plotly.subplots import make_subplots
import requests
```

```
Ввод [9]: df=pd.read_csv('rest_data.csv')
```

```
Ввод [10]: df.info()
display(df.sample(10))
```

```
<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
7952	152169	ГБОУ Многопрофильный лицей №1501	нет	столовая	город Москва, Тихвинский переулок, дом 3	140
194	77929	Subway	да	кафе	город Москва, 2-я Владимирская улица, дом 36	14
7861	69631	Образ	нет	кафе	город Москва, Марксистская улица, дом 34, корп...	50
69	22139	ХуанХэ	нет	ресторан	город Москва, 1-й Балтийский переулок, дом 3/25	60
9388	167660	Хачапуropff	нет	закусочная	город Москва, Первомайская улица, дом 81	0
12889	200591	УРБАН КОФИКС РАША	нет	кафетерий	город Москва, Профсоюзная улица, дом 61А	15
950	154938	Суши Вок	да	кафе	город Москва, Профсоюзная улица, дом 113, корп...	10
14012	210377	Бургерная Электроугли	нет	кафе	город Москва, Театральный проезд, дом 5, строе...	40
11151	178086	Грузинская кухня	нет	кафе	город Москва, Открытое шоссе, дом 9, строение 14А	12
9290	166909	Чойхана	нет	кафе	город Москва, Дмитровское шоссе, дом 95А	49

Пустых значение нет. Проверим дубликаты.

```
Ввод [11]: print(df.duplicated().sum())

0
```

Дубликатов нет. Посмотрим на типы объектов.

```
Ввод [12]: print(df.object_type.unique())

['кафе' 'столовая' 'закусочная' 'предприятие быстрого обслуживания'
 'ресторан' 'кафетерий' 'буфет' 'бар' 'магазин (отдел кулинарии)']
```

Можно поменять тип данных на категориальный. Что и сделаем.

```
Ввод [13]: df.object_type=df.object_type.astype('category')
```

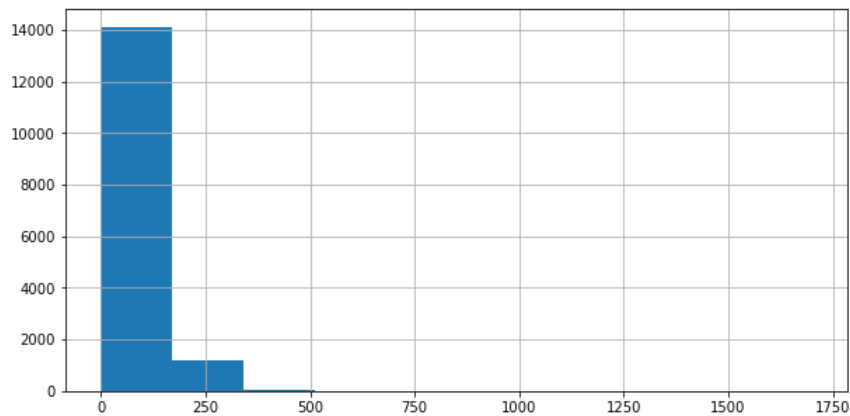
Посмотрим на цифры в столбце содержащем число посадочных мест.

```
Ввод [14]: print(df.number.describe())
```

```
count    15366.000000
mean       59.547182
std       74.736833
min         0.000000
25%       12.000000
50%       40.000000
75%       80.000000
max      1700.000000
Name: number, dtype: float64
```

```
Ввод [15]: df['number'].hist(figsize=(10,5))
```

Out[15]: <AxesSubplot:>



Интересный разброс. Есть с нулем?

```
Ввод [16]: display(len(df[df.number==0]))
```

1621

Да есть, посмотрим к каким типам заведений относятся

```
Ввод [17]: display(df[df.number==0].groupby('object_type').agg({'id':'count'}))
```

id	
object_type	
бар	28
буфет	68
закусочная	189
кафе	310
кафетерий	113
магазин (отдел кулинарии)	145
предприятие быстрого обслуживания	704
ресторан	45
столовая	19

Я бы еще понял отдел кулинарии или предприятие быстрого обслуживания с 0 посадочных мест, продают только на вынос, но ресторан, работающий только на вынос это сильно. Количество посадочных мест является параметром участвующем в анализе, поэтому представляется разумным, удалить эти записи, как недостоверные. Хотя, конечно это 10% данных.

```
Ввод [18]: df=df.drop(df[df['number']==0].index)
```

Посмотри на заведения с большим числом посадочных мест.

Ввод [19]:

display(df[df.number>600].sort\_values(by='number',ascending=False))

	id	object_name	chain	object_type	address	number
12723	199696	Arena by Soho Family	нет	бар	город Москва, Кутузовский проспект, дом 36, ст...	1700
8148	80291	Банкетный зал Шелк	нет	ресторан	город Москва, Большой Саввинский переулок, дом...	1500
8345	27750	СТОЛОВАЯ-БУФЕТ	нет	столовая	город Москва, улица Волхонка, дом 15	1400
9064	19719	КОМБИНАТ ПИТАНИЯ «УПРАВЛЕНИЕ АКАДЕМСЕРВИС»	нет	столовая	город Москва, проспект Вернадского, дом 84, ст...	1288
3686	27026	РУМ СЕРВИС	нет	ресторан	город Москва, площадь Европы, дом 2	1200
4480	27024	РУМ СЕРВИС	нет	ресторан	город Москва, Кутузовский проспект, дом 2/1, с...	1200
2313	26560	Ресторан «АЛЬФА»	нет	ресторан	город Москва, Измайловское шоссе, дом 71, корп...	1040
9955	171116	EATALY	нет	ресторан	город Москва, Киевская улица, дом 2	920
5649	29232	СТОЛОВАЯ МОСКОВСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ	нет	столовая	город Москва, улица Юности, дом 5, строение 1	760
13299	200905	White Hall	нет	ресторан	город Москва, Ферганская улица, дом 8	700
4110	29320	Столовая	нет	столовая	город Москва, Каширское шоссе, дом 31	680
7280	26269	Ресторан «Шахин Шах»	нет	ресторан	город Москва, улица Маршала Захарова, дом 6, к...	675
2317	73306	Ресторан «БЕТА»	нет	ресторан	город Москва, Измайловское шоссе, дом 71, корп...	660
7807	66234	Стамбул	нет	ресторан	город Москва, улица Скульптора Мухиной, дом 11	650
3401	135197	ДУРДИНЬ	нет	ресторан	город Москва, Мичуринский проспект, дом 8, стр...	650
6077	29073	СТОЛОВАЯ 16 ПРИ РУДН	нет	столовая	город Москва, улица Миклухо-Маклая, дом 6	644
7104	27033	Банкетный	нет	ресторан	город Москва, Ленинградский проспект, дом 31А,...	625

Я посмотрел в интернете описание некоторых заведений из этого списка, число посадочных мест более-менее соответствует. Поэтому оставляю данные как есть.

Проверка данных завершена, приступаем к анализу.

## Анализ

Ввод [ ]:

Ввод [ ]:

### Количество объектов по типам

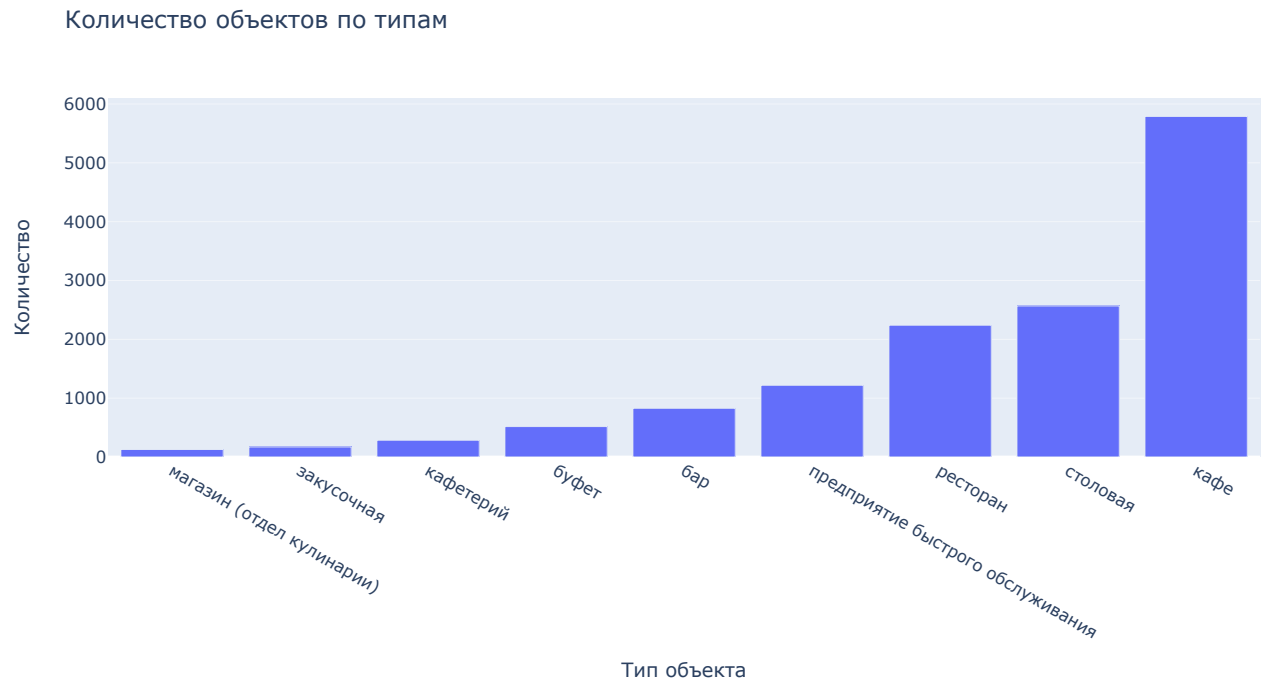
Посмотрим, каково колчество объектов разного типа. Подготовка данных.

Ввод [20]:

dftmp=df.groupby('object\_type').id.count().reset\_index()  
dftmp.columns=['name','count']  
dftmp=dftmp.sort\_values(by='count')

Отображение

```
Ввод [21]: fig = go.Figure(data=[go.Bar(x=dftmp['name'], y=dftmp['count'])])
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Количество')
fig.update_layout(title_text="Количество объектов по типам")
fig.show()
```



```
Ввод [22]: print(df[df.object_type=='кафе'].number.describe())
```

```
count    5789.000000
mean      41.842805
std       37.550917
min        1.000000
25%       18.000000
50%       35.000000
75%       50.000000
max       533.000000
Name: number, dtype: float64
```

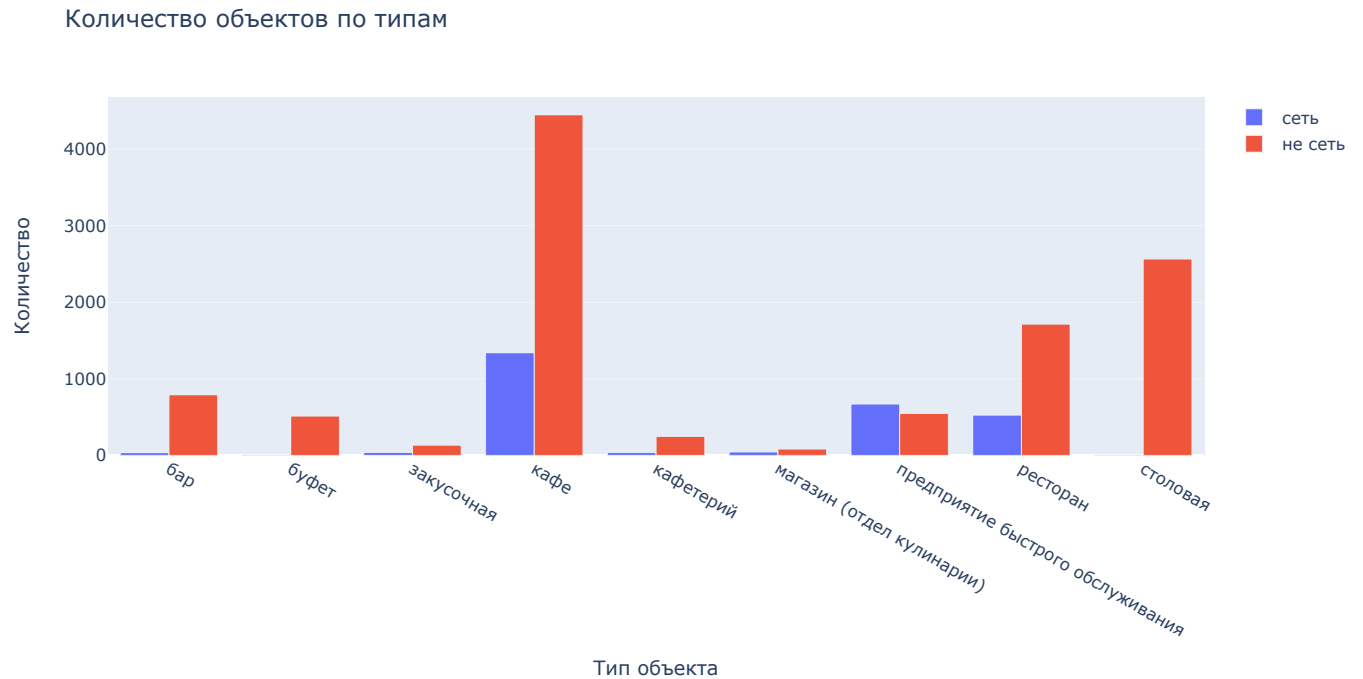
Подводящее число заведений относится к типу кафе. Причем большая часть кафе имеет число посадочных мест до 50, примерно.

## сетевые и не сетевые

Посмотрим как соотносятся сетевые и не сетевые заведения

```
Ввод [23]: dftmp=df.groupby(['object_type', 'chain']).id.count().reset_index()
dftmp.columns=['oname', 'ochain', 'ocount']
```

```
Ввод [24]: bar1=dftmp[dftmp.ochain=='да']
bar2=dftmp[dftmp.ochain=='нет']
fig = go.Figure(data=[go.Bar(x=bar1['oname'], y=bar1['ocount'],name='сеть')])
fig.add_trace(go.Bar(x=bar2['oname'], y=bar2['ocount'],name='не сеть'))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Количество')
fig.update_layout(title_text="Количество объектов по типам")
fig.show()
```



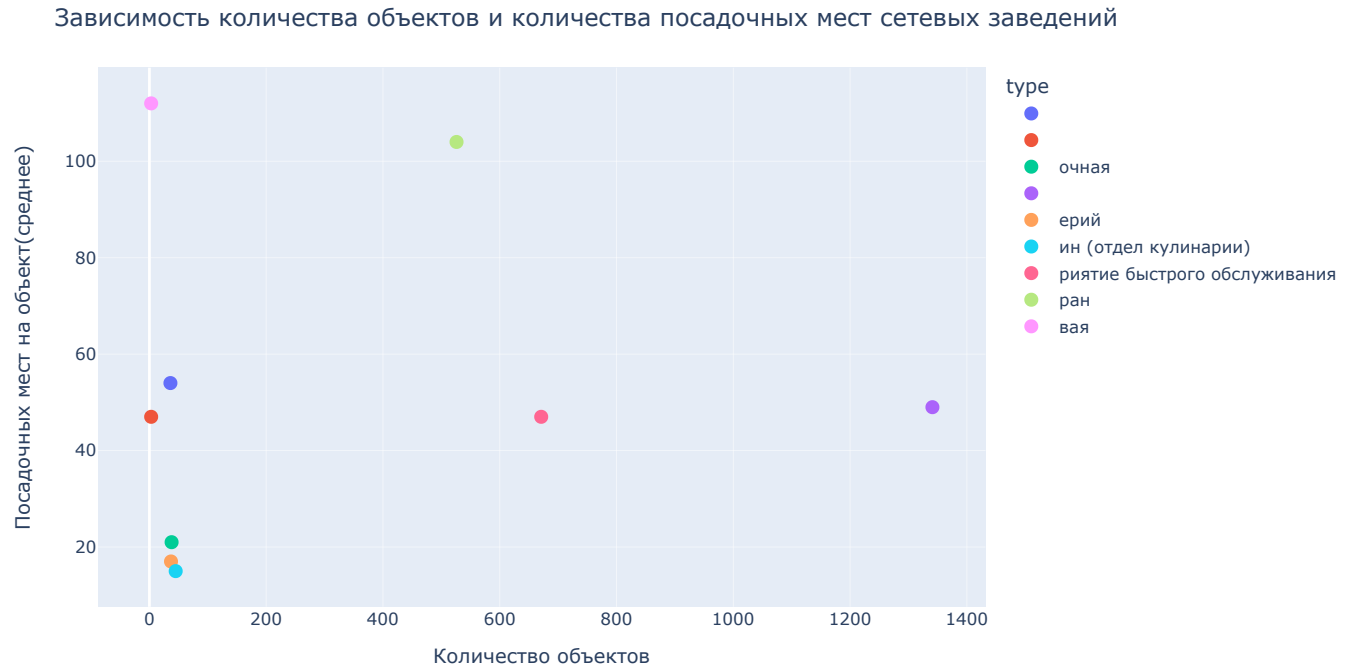
Только в категории предприятий быстрого обслуживания количество сетевых объектов превышает количество не сетевых. Во всех остальных категориях преобладают несетевые объекты.

## Характеристики сетевых объектов

Посмотрим, как распределенно количество объектов и посадочных мест в сетевых заведениях.

```
Ввод [25]: dftmp=df[df.chain=='да'].groupby('object_type').agg({'id':'count','number':'sum'}).reset_index()
dftmp.columns=['type','kolvo','seats']
dftmp['seat_per_id']=round(dftmp.seats/dftmp.kolvo).astype('int')
```

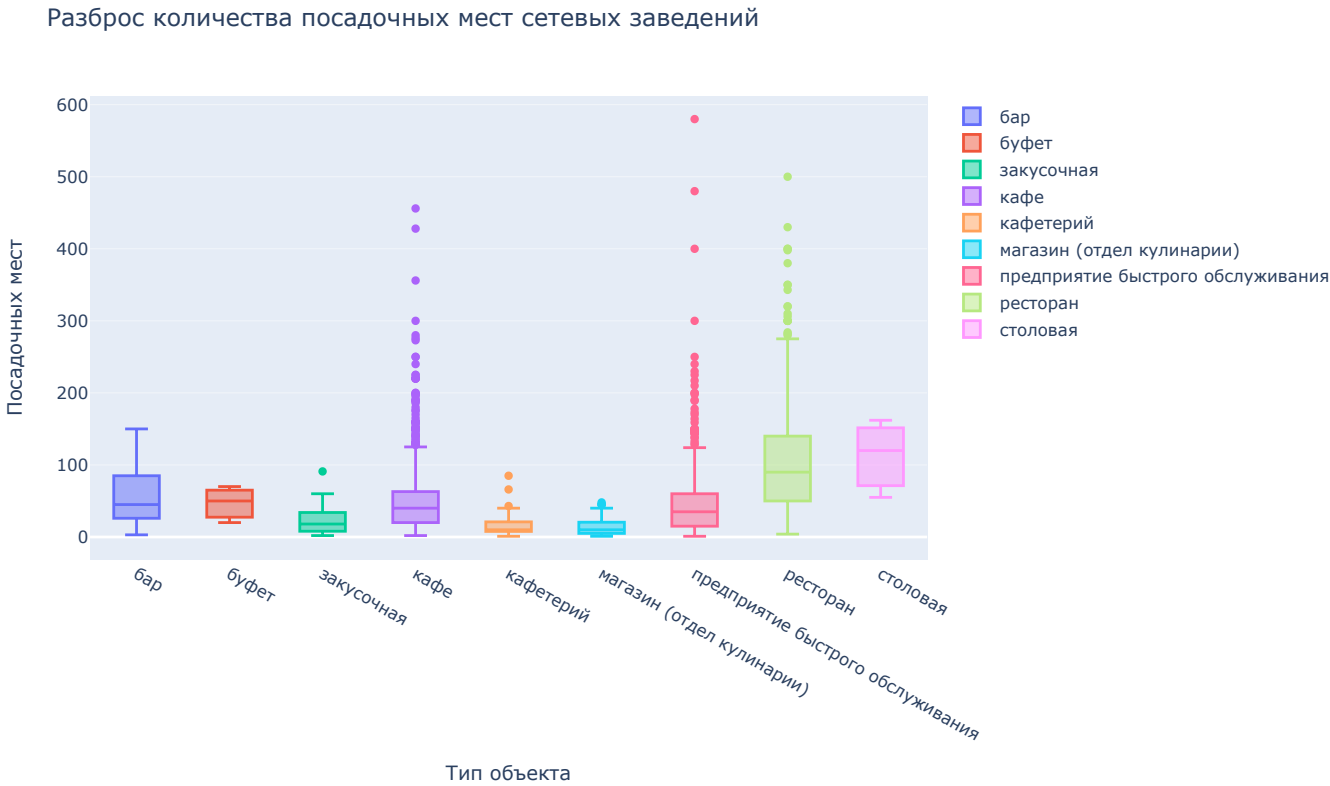
```
Ввод [26]: fig=px.scatter(dftmp,x='koIvo', y='seat_per_id',color='type')
fig.for_each_trace(lambda t: t.update(name=t.name[5:]))
fig.update_traces(marker_size=10)
fig.update_xaxes(title_text='Количество объектов')
fig.update_yaxes(title_text='Посадочных мест на объект(среднее)')
fig.update_layout(title_text="Зависимость количества объектов и количества посадочных мест сетевых заведений")
fig.show()
```



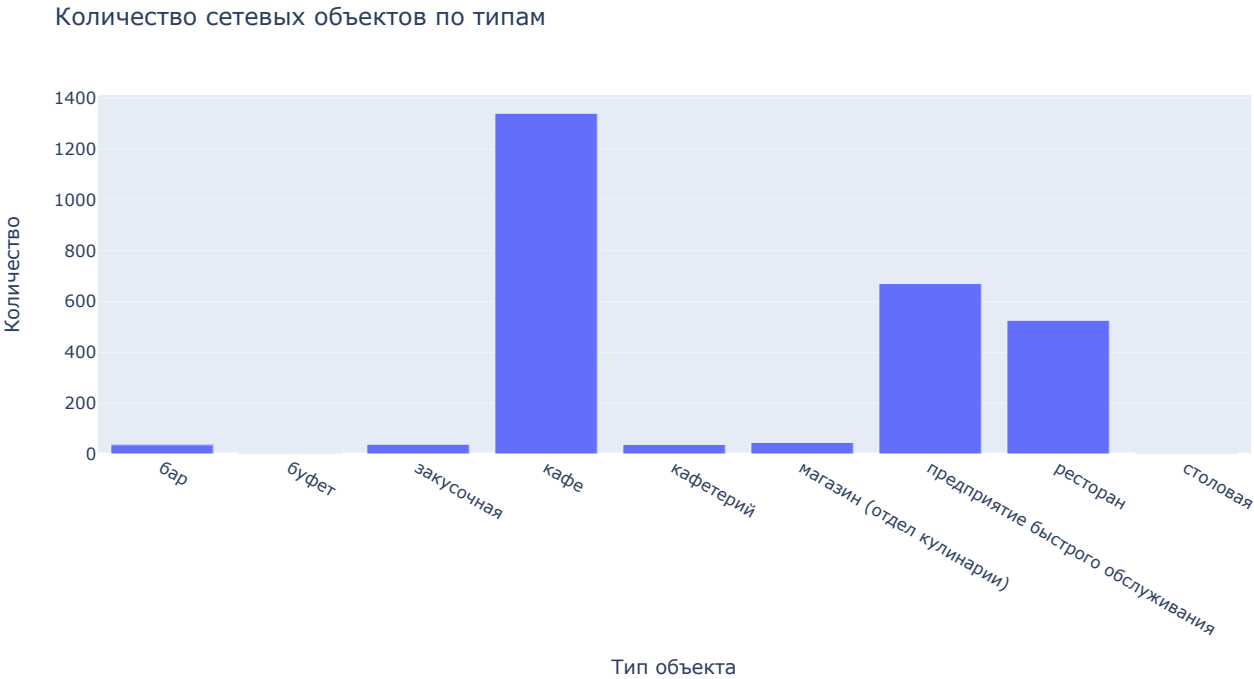
Можно сказать, что сетевые заведения тяготеют к большому количеству заведений, при небольшом количестве посадочных мест. Из этой закономерности выбиваются только столовые и рестораны.

Посмотрим еще по-другому.

```
Ввод [27]: fig = go.Figure()
ls=np.sort(df.object_type.unique())
for name in ls:
    data=df[(df.object_type==name) & (df.chain=='да')].number
    fig.add_trace(go.Box(y=data,name=name))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Посадочных мест')
fig.update_layout(
    title_text="Разброс количества посадочных мест сетевых заведений",
    height=600)
fig.show()
```



```
Ввод [28]: dftmp=df[df.chain=='да'].groupby('object_type').agg({'id':'count'}).reset_index()
fig = go.Figure(data=[go.Bar(x=dftmp['object_type'], y=dftmp['id'])])
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Количество')
fig.update_layout(title_text="Количество сетевых объектов по типам")
fig.show()
```



Тут мы еще раз видим, что наиболее массовые по количеству типы объектов: кафе и предприятие быстрого обслуживания тяготеют к небольшому (примерно

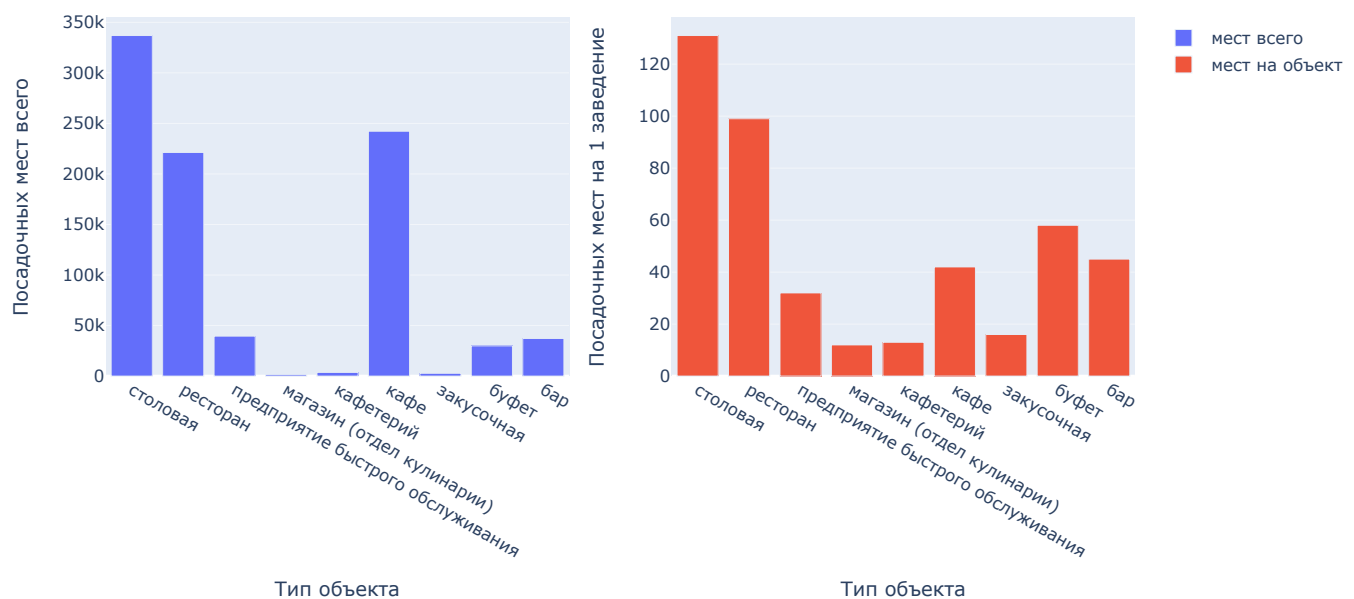
до 50) числу посадочных мест. И даже третья по массовости категория, рестораны, несмотря на выбросы, в среднем тяготеет к числу посадочных мест около 90.

## Характеристики всех объектов

```
Ввод [29]: dftmp=df.groupby('object_type').agg({'id':'count','number':'sum'}).reset_index()
dftmp.columns=['type','kolvo','seats']
dftmp['seat_per_id']=round(dftmp.seats/dftmp.kolvo).astype('int')
```

```
Ввод [30]: fig = make_subplots(rows=1, cols=2)
bar1=dftmp.sort_values(by='type',ascending=False)
bar2=dftmp.sort_values(by='type',ascending=False)
fig.add_trace(go.Bar(x=bar1['type'], y=bar1['seats'],name='мест всего'),row=1,col=1)
fig.add_trace(go.Bar(x=bar2['type'], y=bar2['seat_per_id'],name='мест на объект'),row=1,col=2)
fig.update_xaxes(title_text='Тип объекта',row=1,col=1)
fig.update_yaxes(title_text='Посадочных мест всего',row=1,col=1)
fig.update_xaxes(title_text='Тип объекта',row=1,col=2)
fig.update_yaxes(title_text='Посадочных мест на 1 заведение',row=1,col=2)
fig.update_layout(title_text="Количество посадочных мест и среднее число посадочных мест на 1 заведение по типам")
fig.show()
```

Количество посадочных мест и среднее число посадочных мест на 1 заведение по типам

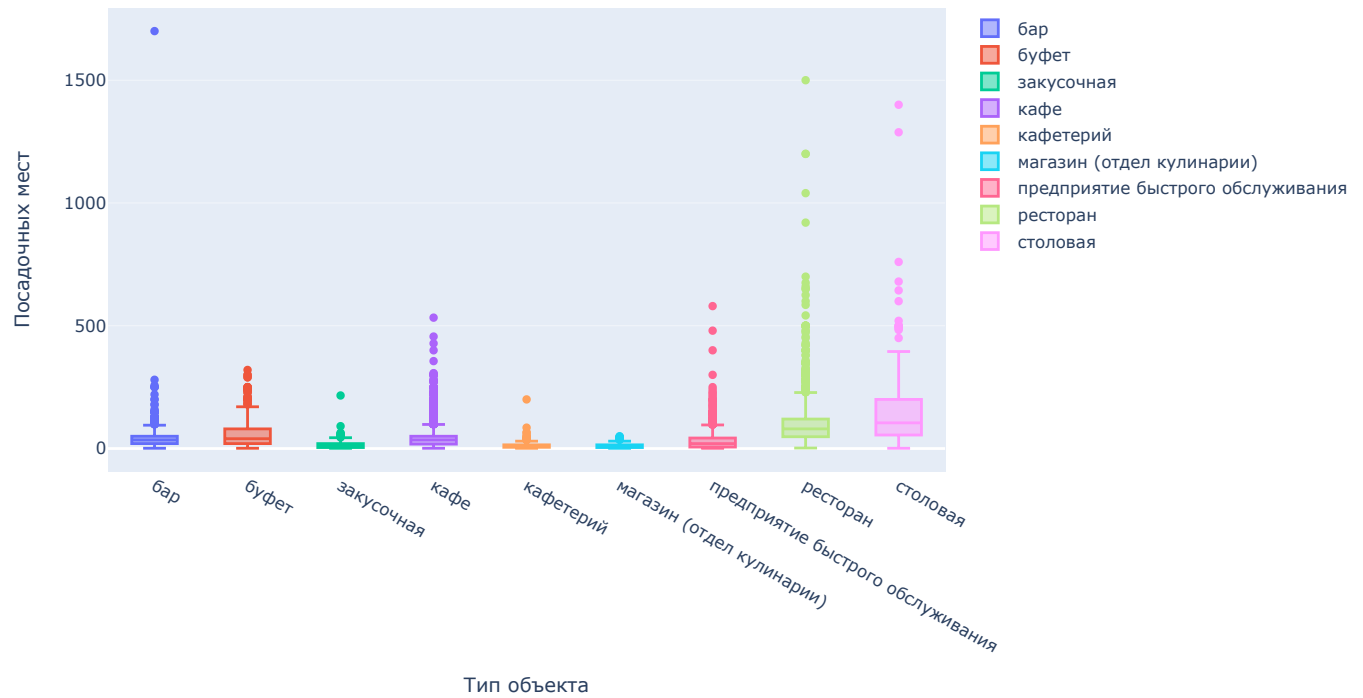


На этих диаграммах мы видим, что столовые и рестораны имеют самое большое количество посадочных мест в целом и на одно заведение. Все остальные объекты имеют гораздо меньше мест на объект. Кафе, за счет своей массовости, выделяются огромным числом посадочных мест в целом, при том, что на одно заведение приходится около 40 мест.



```
Ввод [31]: fig = go.Figure()
ls=np.sort(df.object_type.unique())
for name in ls:
    data=df[df.object_type==name].number
    fig.add_trace(go.Box(y=data,name=name))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Посадочных мест')
fig.update_layout(
    title_text="Количество посадочных мест по типам объектов",
    height=600)
fig.show()
```

Количество посадочных мест по типам объектов



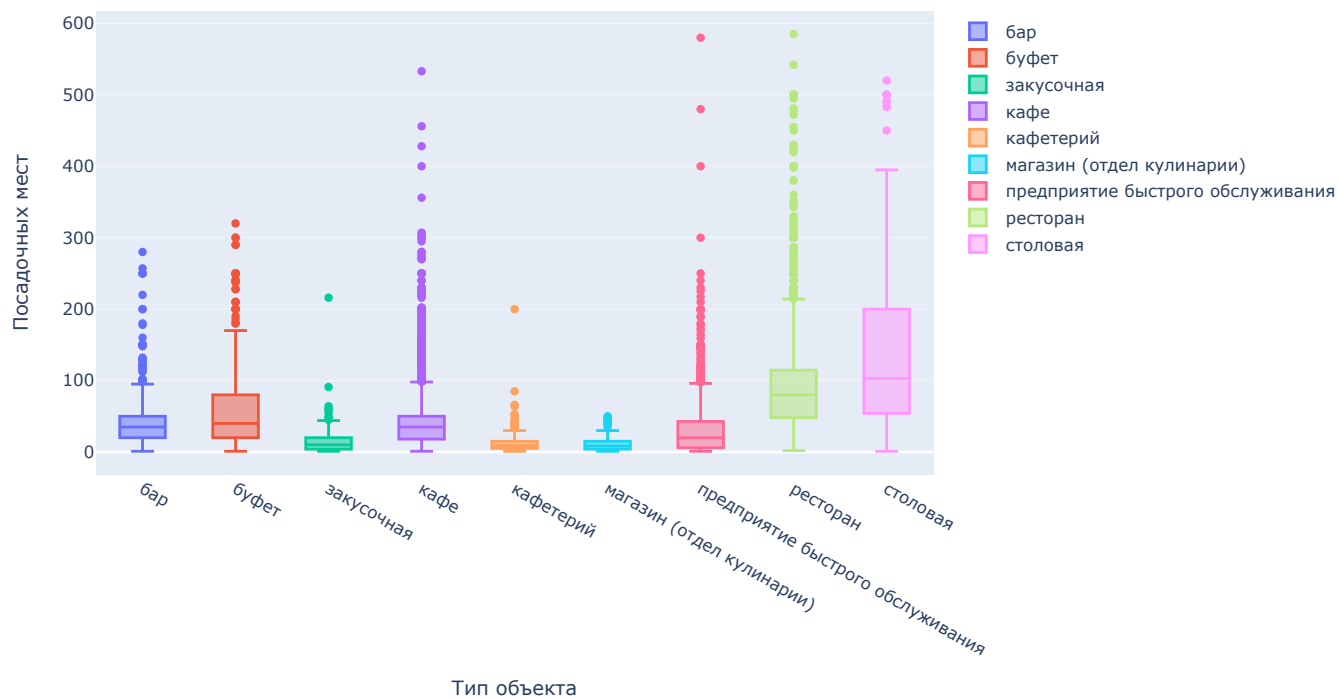
Выбросы портят масштаб. Сделаю то-же самое отбросив объекты с числом посадочных мест >600

```

Ввод [32]: fig = go.Figure()
ls=np.sort(df.object_type.unique())
for name in ls:
    data=df[(df.object_type==name) & (df.number<600)].number
    fig.add_trace(go.Box(y=data,name=name))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Посадочных мест')
fig.update_layout(
    title_text="Количество посадочных мест по типам объектов",
    height=600)
fig.show()

```

Количество посадочных мест по типам объектов

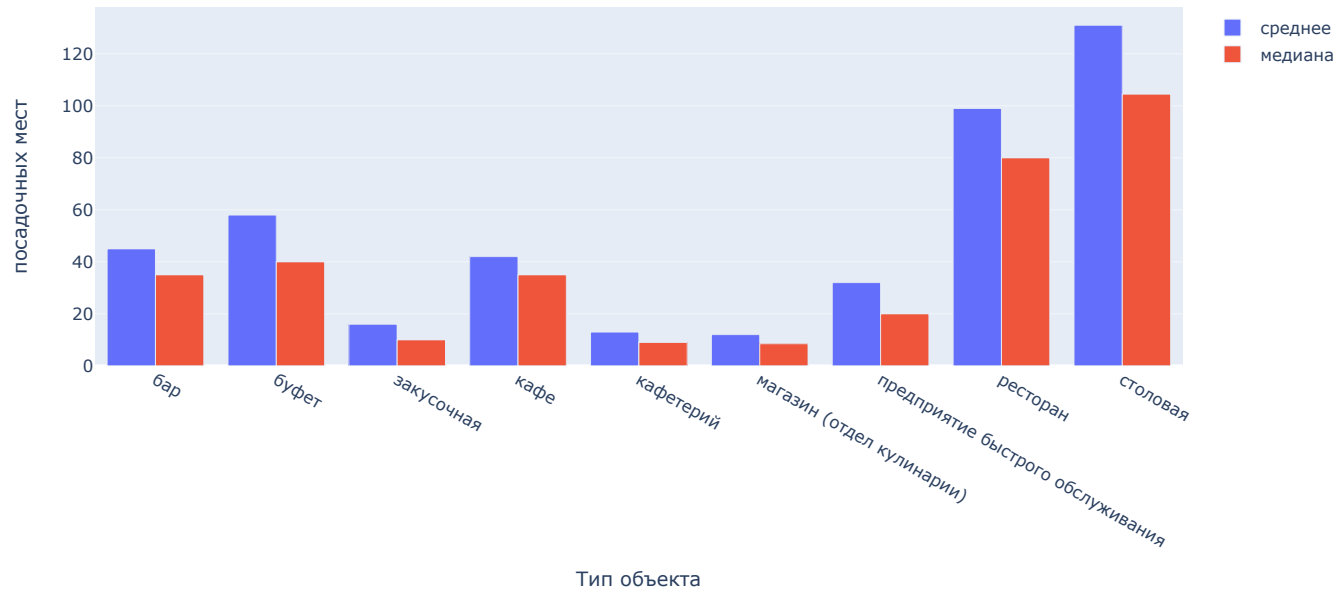


Закусочная, кафетерий, отдел кулинарии образуют группу с небольшим числом посадочных мест и небольшим разбросом. Кафе, буфет, бар, закусочная и предприятия быстрого обслуживания образуют следующую группу с большим числом посадочных мест и умеренным разбросом, не считая бар на 1700 мест. И столовая и рестораны образуют следующую группу, с большим числом посадочных мест и большим разбросом значений.

Вычислим среднее число посадочных мест по каждому типу объекта

```
Ввод [33]: dftmp=df.groupby('object_type').agg({'id':'count','number':['sum','median']}).reset_index()
dftmp.columns=['type','kolvo','seats','median_seats']
dftmp['seat_per_id']=round(dftmp.seats/dftmp.kolvo).astype('int')
fig = go.Figure(data=[go.Bar(x=dftmp['type'], y=dftmp['seat_per_id'],name='среднее')])
fig.add_trace(go.Bar(x=dftmp['type'], y=dftmp['median_seats'],name='медиана'))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='посадочных мест')
fig.update_layout(title_text='Среднее число посадочных мест')
fig.show()
```

Среднее число посадочных мест



За счет выбросов медиана везде немного меньше среднего. Только две категории, рестораны и столовые, выбиваются из общей группы объектов с небольшим средним (меньше 40) числом посадочных мест.

Анализ объектов в разрезе адресных данных

В этой части нам необходимо выделить из адреса улицу и проанализировать расположение объектов в разрезе улиц и районов. Для того, чтобы получить информацию по районам, я использую данные мосгаза, по ссылке внизу. Эти данные содержат наименование улицы, административного округа и района города.

```
Ввод [34]: url = 'https://frs.noosphere.ru/xmlui/bitstream/handle/20.500.11925/714058/mosgaz-streets.csv?sequence=1&isAllowed=y'
info=pd.read_csv(url)
display(info.sample(5))
```

	streetname	areaid	okrug	area
2320	Анадырский проезд	72	СВАО	Бабушкинский район
3592	улица Нижние Поля	115	ЮВАО	Район Люблино
4133	1-й Кадашёвский переулок	21	ЦАО	Район Якиманка
3440	Сосинская улица	122	ЮВАО	Южнопортовый Район
2766	Малое кольцо Московской железной дороги (МКМЖД)	92	СЗАО	Район Хорошево-Мневники

Вставим в наш датафрейм столбец для улицы

```
Ввод [35]: df['street']=''
```

Следующая функция выделяет из строки адреса название улицы.

Логика ее работы следующая:

адрес представляет собой строку, разделенную запятыми, в которой в начале идет/или не идет название города. если строка начинается с названия города, то после запятой идет название улицы, или города-спутника. если строка не начинается с названия города, то сразу идет название улицы.

Таким образом, после применения этой функции мы получим в новом столбце названия улиц, или название населенного пункта-спутника Москвы.

С такими данными, все заведения расположенные в населенных пунктах-спутниках будут иметь одно и то-же значение в строке адреса. Я считаю, что нас это устраивает, т.к. с одной стороны, мы не можем определить район подобных улиц по данным мосгаза, там этих данных нет. А с другой стороны название населенного пункта-спутника Москвы само служит названием района.

```
Ввод [36]: def extr_street(val):
    st=val.split(',')
    if st[0].startswith('город'):
        ul=st[1].lstrip() # адресная строка начинается с названия города
    else:
        ul=st[0].lstrip() # адресная строка начинается с названия улицы
    return(ul)

df['street']=df.address.apply(extr_street)
```

Следующая функция определяет название района по данным мосгаза, исходя из названия улицы. Принято следующее допущение: если улица проходит через несколько районов, то берется первое, встретившееся в данных мосгаза наименование района.

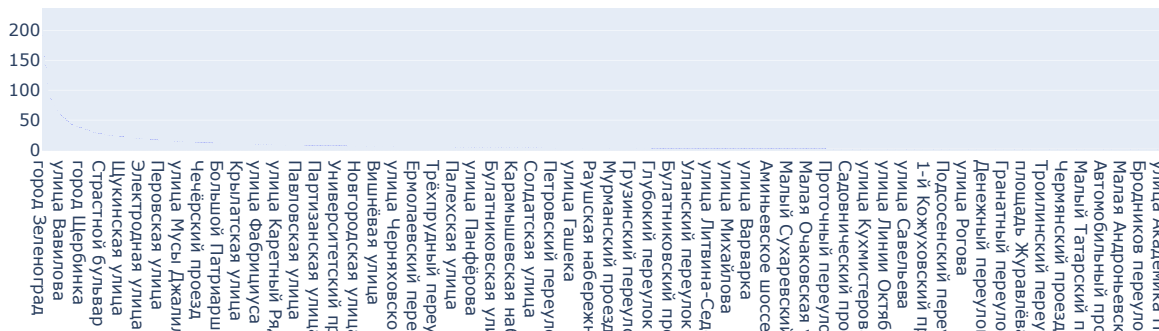
```
Ввод [37]: def def_area(row):
    st=row.street
    tdf=info[info.streetname==st]
    if (st.startswith('город')==True) or (st.startswith('поселение')==True) or (st.startswith('деревня')==True):
        are=st
        okr=st
    else:
        if len(tdf)==0:
            okr='не найдено'
            are='не найдено'
        else:
            okr=tdf.okrug.to_numpy()[0]
            are=tdf.area.to_numpy()[0]
    row.area=are
    row.A0=okr
    return(row)
```

Построим датафрейм, содержащий для каждой улицы количество заведений и количество посадочных мест в разрезе типов объектов.

```
Ввод [38]: dfstreet=df.groupby(['street', 'object_type']).agg({'id': 'count', 'number': 'sum'}).reset_index()
```

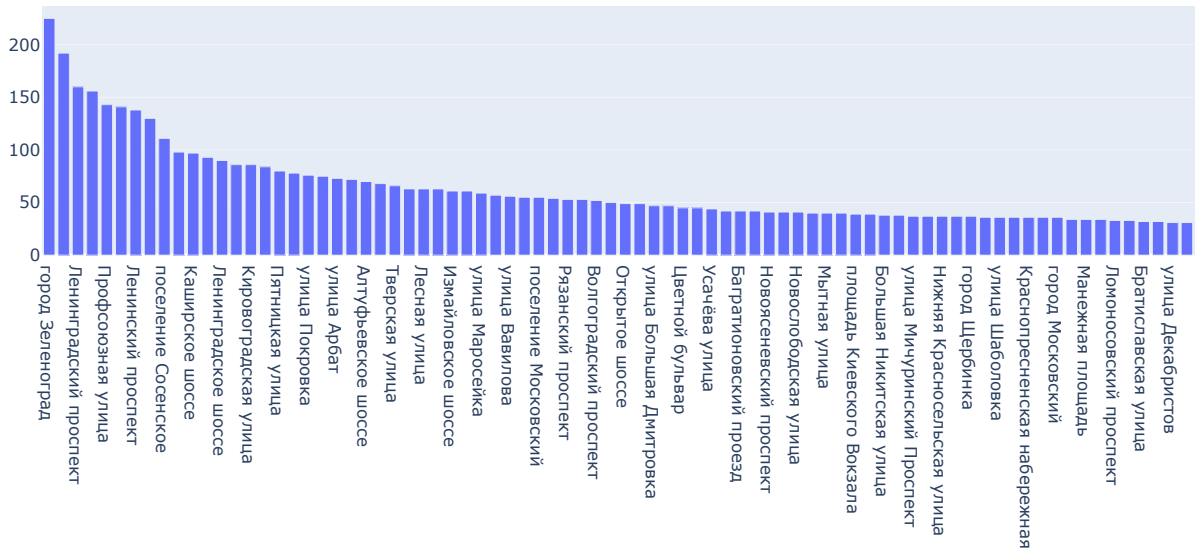
Хочу посмотреть как вообще распределены объекты по улицам

```
Ввод [39]: dftmp=dfstreet.groupby('street').agg({'id': 'sum'}).reset_index().sort_values(by='id', ascending=False)
fig = go.Figure(data=[go.Bar(x=dftmp['street'], y=dftmp['id'])])
fig.update_xaxes(title_text='')
fig.update_yaxes(title_text='')
fig.update_layout(title_text='')
fig.show()
print('Всего улиц {}'.format(len(dftmp)))
```



Это какая-то экспонента. Посмотрю улицы с числом объектов до 30.

```
Ввод [40]: dftmp=dfstreet.groupby('street').agg({'id':'sum'}).reset_index().sort_values(by='id',ascending=False)
dftmp=dftmp[dftmp.id>30]
fig = go.Figure(data=[go.Bar(x=dftmp['street'], y=dftmp['id'])])
fig.update_xaxes(title_text='')
fig.update_yaxes(title_text='')
fig.update_layout(title_text='')
fig.show()
print('Улиц в этом графике'.format(len(dftmp)))
```



Улиц в этом графике

Держа в уме, что я оставил Зеленоград, получается, что максимум и составляют 10 улиц. Всего 80 улиц с числом объектов до 30 из 1830. Глядя на огромный хвост вправо, можно сказать, что заведения распределены довольно равномерно по всему городу и имеется небольшая область со сверхконцентрацией. Для анализ нужно смотреть распределение непосредственно на карте.

Построим датафрейм, содержащий данные по 15 улиц с наибольшим числом заведений и заполним для них данные по районам, применив функцию `def_area()` описанную выше.

```
Ввод [41]: dftop=dfstreet.groupby('street').agg({'id':'sum'}).reset_index() # формируем список улиц с числом заведений без разбивки по типам
dftop=dftop.sort_values(by='id',ascending=False).head(15) # оставляем 15 улиц с наибольшим числом заведений
dftop['area']=''
dftop['A0']=''

dftop=dftop.apply(def_area,axis=1) # заносим данные по районам
```

Посмотрим на эти улицы

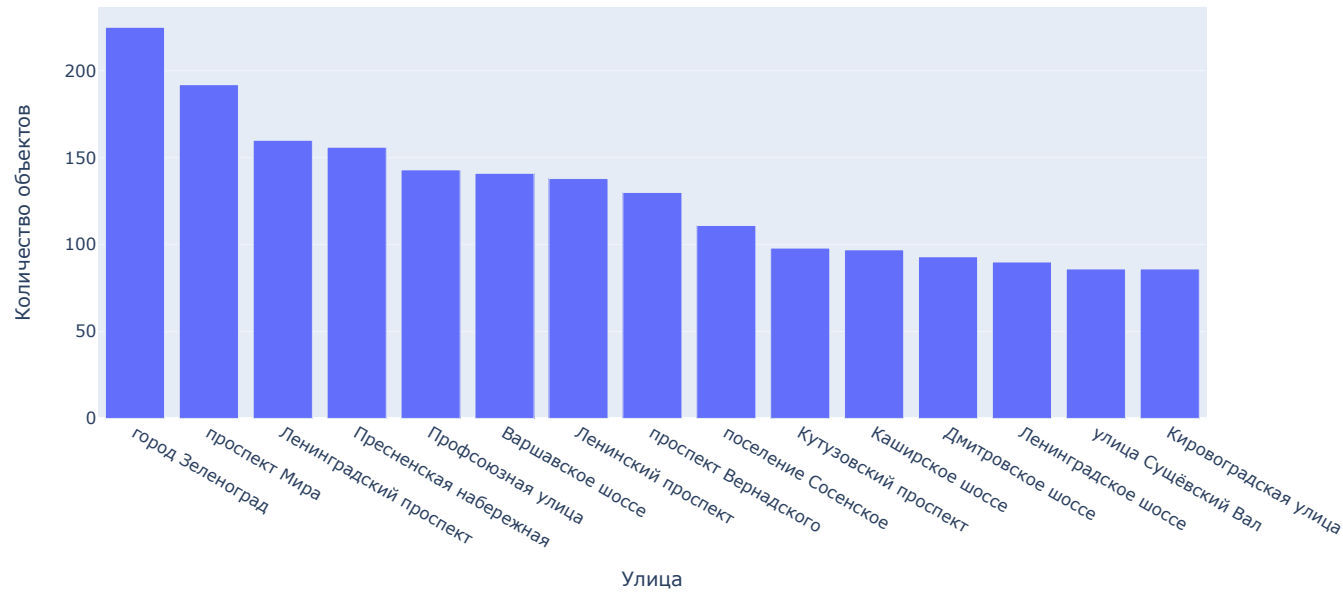
Ввод [42]: `display(dftop)`

	street	id	area	АО
1318	город Зеленоград	225	город Зеленоград	город Зеленоград
1388	проспект Мира	192	Алексеевский район	СВАО
677	Ленинградский проспект	160	Район Аэропорт	САО
973	Пресненская набережная	156	Пресненский район	ЦАО
987	Профсоюзная улица	143	Академический район	ЮЗАО
404	Варшавское шоссе	141	Район Чертаново Центральное	ЮАО
680	Ленинский проспект	138	Район Проспект Вернадского	ЗАО
1385	проспект Вернадского	130	Район Проспект Вернадского	ЗАО
1358	поселение Сосенское	111	поселение Сосенское	поселение Сосенское
668	Кутузовский проспект	98	Район Дорогомилово	ЗАО
594	Каширское шоссе	97	Район Москворечье-Сабурово	ЮАО
495	Дмитровское шоссе	93	Бескудниковский Район	САО
678	Ленинградское шоссе	90	Район Матушкино-Савелки	Зеленоград
1782	улица Сущёвский Вал	86	Тверской район	ЦАО
599	Кировоградская улица	86	Район Чертаново Центральное	ЮАО

Как мы видим, в список ТОП-10 попали два населенных пункта-спутника, Зеленоград и Сосенское. Тут есть два варианта - убрать эти два наименования, т.к. фактически это не Москва и обрезать список до 10 улиц, или оставить как есть. Я решил оставить как есть.

Ввод [43]: `fig = go.Figure(data=[go.Bar(x=dftop['street'], y=dftop['id'])])  
fig.update_xaxes(title_text='Улица')  
fig.update_yaxes(title_text='Количество объектов')  
fig.update_layout(title_text="Улицы с наибольшим количеством объектов")  
fig.show()`

Улицы с наибольшим количеством объектов



Держи в уме, что я оставил Зеленоград, хотя это не Москва, то улицей с наибольшим числом объектов будет проспект Мира.

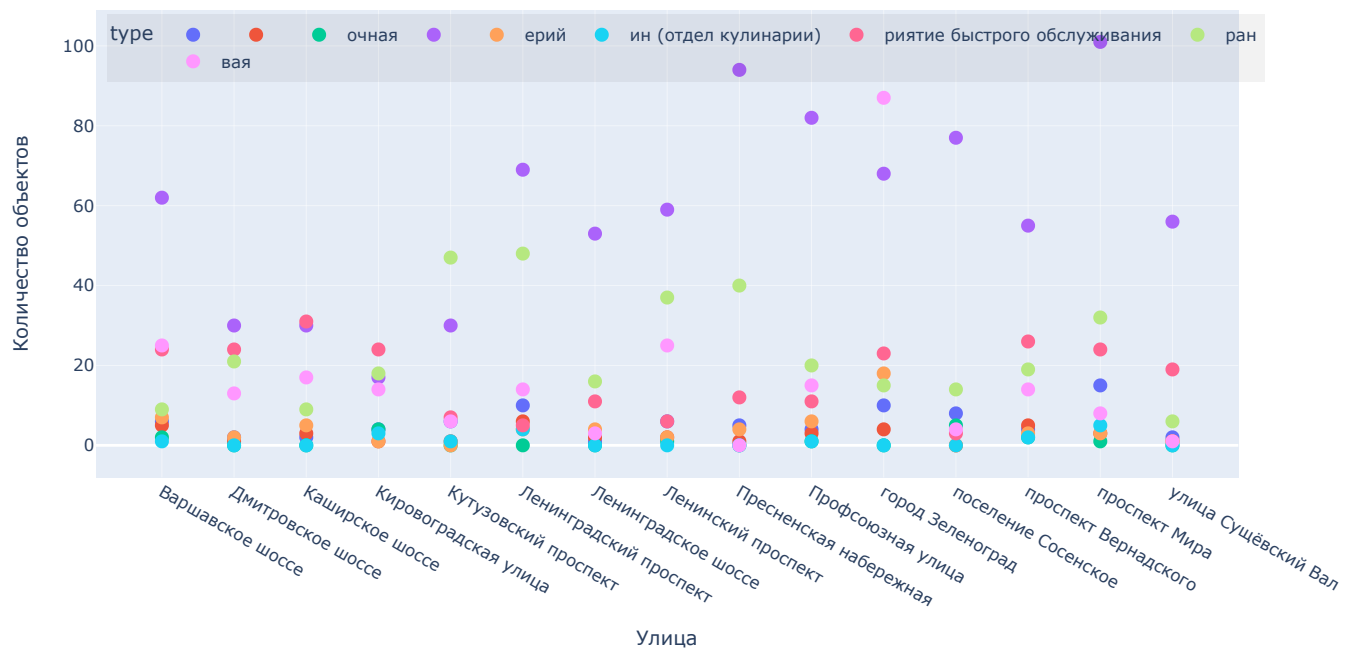
Посмотрим сколько объектов разного типа на ТОП-15 улицах.

```

Ввод [44]: dftmp=dfstreet[dfstreet.street.isin(dftop.street)]
dftmp.columns=['street','type','id','number']
fig=px.scatter(dftmp,x='street', y='id',color='type')
fig.update_traces(marker_size=10)
fig.update_xaxes(title_text='Улица')
fig.update_yaxes(title_text='Количество объектов')
fig.for_each_trace(lambda t: t.update(name=t.name[5:]))
fig.update_layout(
    title_text="Распределение количества объектов по типам для улиц с наибольшим числом объектов",
    legend=dict(
        orientation="h",
        bgcolor = 'rgba(0,0,0,0.05)',
        yanchor="top",
        y=0.99,
        xanchor="left",
        x=0.01))
fig.show()

```

Распределение количества объектов по типам для улиц с наибольшим числом объектов

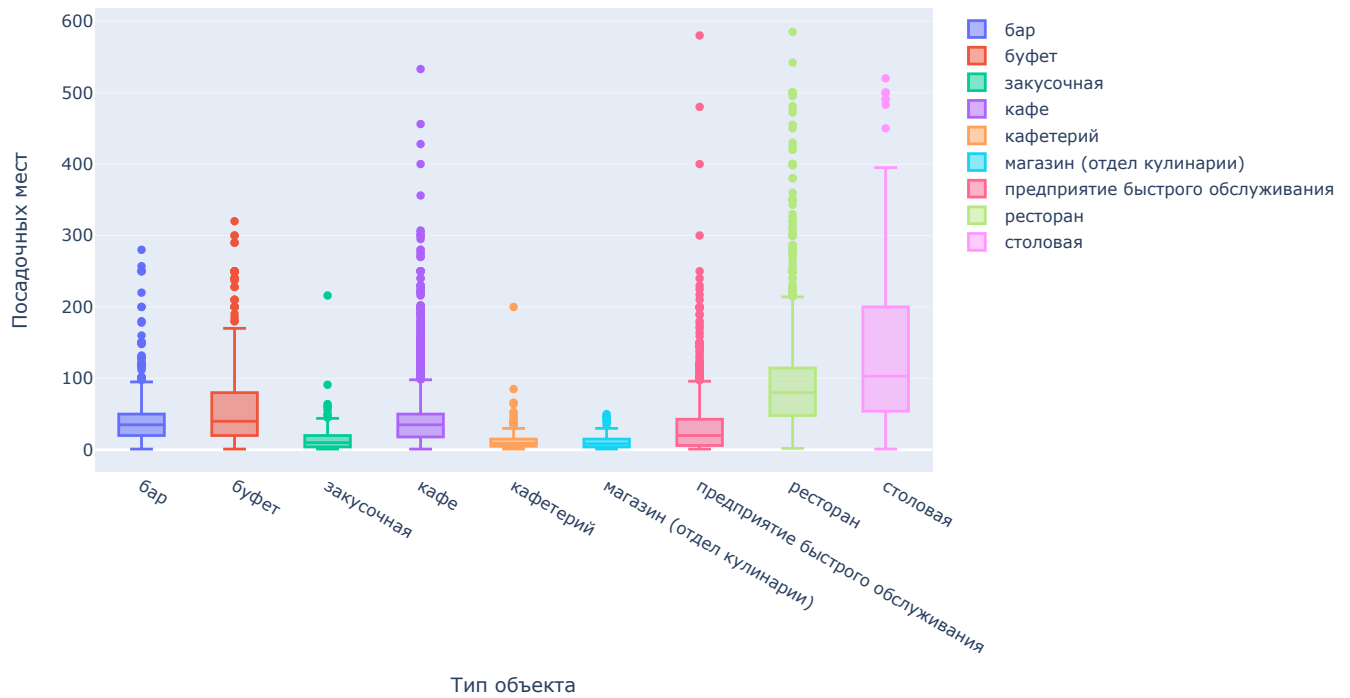


Наибольшее число объектов типа кафе, что и очевидно, т.к. как мы видели ранее это в целом наиболее распространенный тип объекта. Интересно, что в Зеленограде преобладают столовые, на Каширском шоссе и Кировоградской улице - предприятия общественного питания, а на Кутузовском проспекте - рестораны.

Посмотрим на распределение посадочных мест по типам объектов для ТОП-10 улиц

```
Ввод [45]: dftmp=dfstreet[dfstreet.street.isin(dftop.street)]
fig = go.Figure()
ls=np.sort(df.object_type.unique())
for name in ls:
    data=df[(df.object_type==name) & (df.number<600)].number
    fig.add_trace(go.Box(y=data,name=name))
fig.update_xaxes(title_text='Тип объекта')
fig.update_yaxes(title_text='Посадочных мест')
fig.update_layout(
    title_text="Количество посадочных мест для ТОП-10 улиц по типам объектов",
    height=600)
fig.show()
```

Количество посадочных мест для ТОП-10 улиц по типам объектов

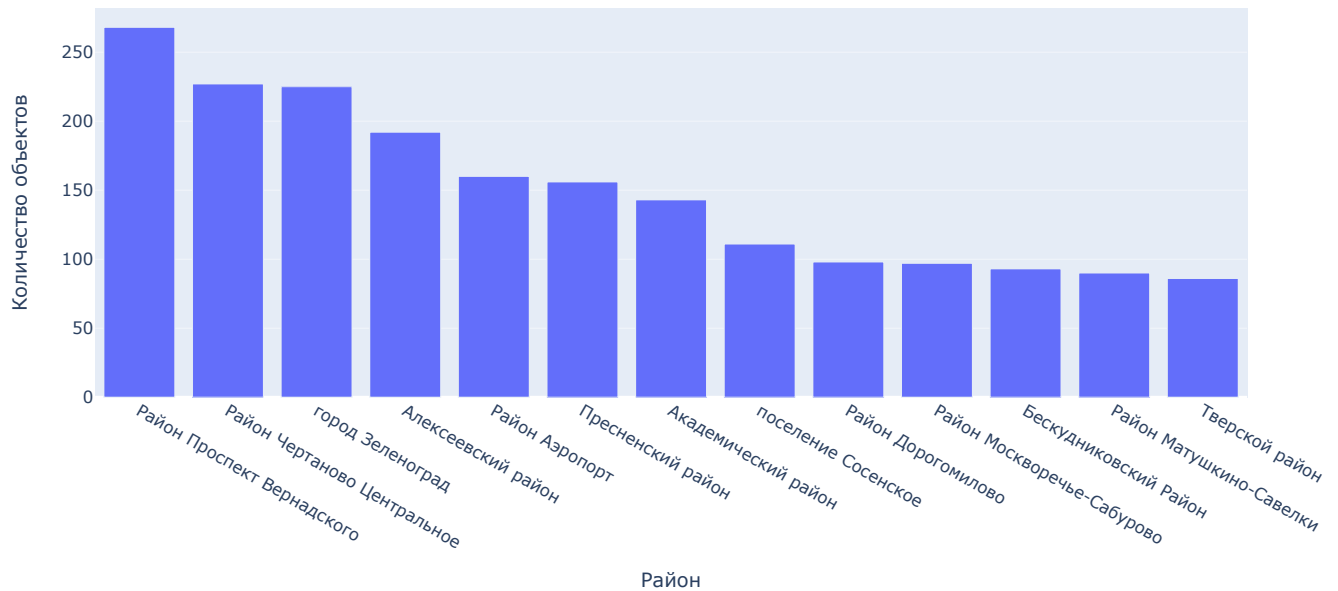


Посмотрим на общее число объектов по районам.



```
Ввод [46]: dftmp=dftop.groupby('area').agg({'id':'sum'}).reset_index().sort_values(by='id',ascending=False)
fig = go.Figure(data=[go.Bar(x=dftmp['area'], y=dftmp['id'])])
fig.update_xaxes(title_text='Район')
fig.update_yaxes(title_text='Количество объектов')
fig.update_layout(title_text="Районы с наибольшим количеством объектов")
fig.show()
```

Районы с наибольшим количеством объектов



Мы видим, что наибольшее число объектов в районе проспекта Вернадского. На втором месте район Чертаново Центральное, и город Зеленоград.

## улица с 1 объектом

Определим число улиц с одним объектом и заодно заполним для них значения района расположения.

```
Ввод [47]: dfbot=dfstreet.groupby('street').agg({'id':'sum'}).reset_index()
dfbot=dfbot[dfbot.id==1]
print('Число улиц с одним объектом : {}'.format(len(dfbot)))

dfbot['area']=''
dfbot['AO']=''

dfbot=dfbot.apply(def_area,axis=1) # определяем район расположения улицы

Число улиц с одним объектом : 552
```

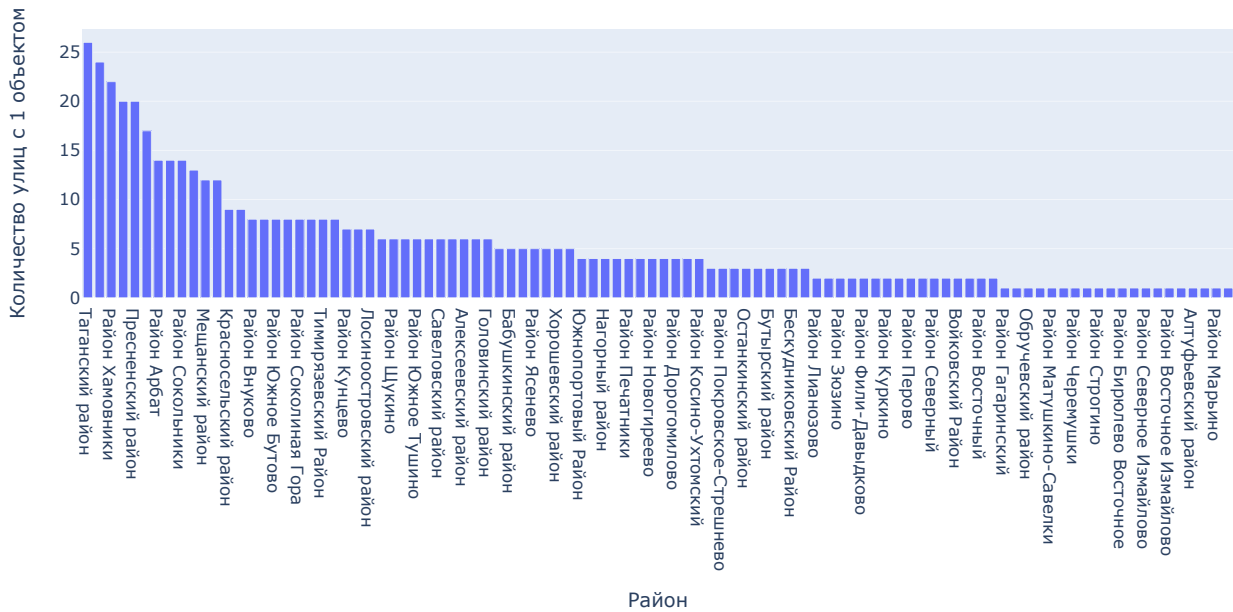
Посмотрим в каких районах расположены эти улицы:

```
Ввод [48]: display('Количество районов, в которых имеется улица с единственным объектом {}'.format(len(np.sort(dfbot.area.unique()))))

'Количество районов, в которых имеется улица с единственным объектом 99'
```

```
Ввод [49]: dftmp=dftbot[dfbot.area!='не найдено'].groupby('area').agg({'id':'sum'}).reset_index().sort_values(by='id',ascending=False)
fig = go.Figure(data=[go.Bar(x=dftmp['area'], y=dftmp['id'])])
fig.update_xaxes(title_text='Район')
fig.update_yaxes(title_text='Количество улиц с 1 объектом')
fig.update_layout(title_text="Количество улиц с 1 объектом по районам")
fig.show()
```

Количество улиц с 1 объектом по районам



На первом месте по числу улиц с 1 объектом - Таганский район, далее не считая тех улиц, для которых не удалось определить район, идут Тверской, Марьяна роща и Арбат.

Ссылка на презентацию

<https://disk.yandex.ru/i/XNwbqgUP4W6ZsA> (https://disk.yandex.ru/i/XNwbqgUP4W6ZsA)

Заключение

Изучив предоставленные данные, можно сделать следующие выводы:

Характеристики объектов.

- 1. Во всех категориях объектов, кроме предприятий быстрого обслуживания, не сетевые заведения преобладают над сетевыми.
- 2. Самая массовая по числу точек категория - кафе.
- 3. Число посадочных мест для кафе имеет большой разброс, от 1 до 500, медиана - 35
- 4. Самая массовая по числу мест категория - столовая. Она же является второй по количеству заведений.
- 5. Медианное число посадочных мест для столовых - 103
- 6. Третьей категорией по числу объектов и второй по количеству посадочных мест являются рестораны, медиана - 80.

Расположение объектов

- 1. Имеется около 10 улиц с большой концентрацией объектов.
- 2. Самый распространенный тип заведения для этих ТОП-10 улиц - кафе
- 3. В целом, 1830 улиц характеризуется небольшим (до 30) количеством объектов.
- 4. Имеется окло 500 улиц с одним объектом

Вывод

В целом, недостаточно данных, что-бы давать осмысленные рекомендации. Необходимы данные о числе посетителей, среднем чеке, площади заведения, средней стоимости аренды, расстоянии до ближайшего заведения подобного типа, расстояние до объектов транспортной инфраструктуры.

Но анализируя то, что мы имеем, можно рекомендовать кафе с числом посадочных мест около 40, как самый массовый тип объектов.

