

Анализ рынка заведений общественного питания Москвы

Оглавление

- 1 Общая информация
- 2 Шаг №1. Загрузка данных и подготовка их к анализу
- 3 Шаг №2. Анализ данных
 - 3.1 Соотношение видов объектов общественного питания по количеству.
 - 3.2 Соотношение сетевых и несетевых заведений по количеству.
 - 3.3 Анализ наиболее распространенных видов сетевых заведений
 - 3.4 Что характерно для сетевых заведений: много заведений с небольшим числом посадочных мест в каждом или мало заведений с большим количеством посадочных мест?
 - 3.5 Для каждого вида объекта общественного питания опишите среднее количество посадочных мест. Какой вид предоставляет в среднем самое большое количество посадочных мест?
 - 3.6 Выделите в отдельный столбец информацию об улице из столбца address
 - 3.7 Постройте график топ-10 улиц по количеству объектов общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?
 - 3.8 Найдите число улиц с одним объектом общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?
- 4 Шаг №3. Подготовка презентации
- 5 Выводы

Общая информация

[Вернуться к оглавлению](#)

Описание проекта

Вы решили открыть небольшое кафе в Москве. Оно оригинальное — гостей должны обслуживать роботы. Проект многообещающий, но дорогой. Вместе с партнёрами вы решились обратиться к инвесторам. Их интересует текущее положение дел на рынке — сможете ли вы снискать популярность на долгое время, когда все зеваки посмотрят на роботов-официантов?

Входные данные:

Файл `/datasets/rest_data.csv` с данными о заведениях общественного питания Москвы.

Цель: для инвесторов подготовить исследование рынка заведений общественного питания в Москве и дать рекомендации о виде заведения, количестве посадочных мест, а также районе расположения.

Задачи

Шаг №1. Загрузите данные и подготовьте их к анализу

Загрузите данные о заведениях общественного питания Москвы. Убедитесь, что тип данных в каждой колонке — правильный, а также отсутствуют пропущенные значения и дубликаты. При необходимости обработайте их.

Путь к файлу: /datasets/rest_data.csv.

Шаг №2. Анализ данных

- Исследуйте соотношение видов объектов общественного питания по количеству. Постройте график.
- Исследуйте соотношение сетевых и несетевых заведений по количеству. Постройте 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 опишите среднее количество посадочных мест. Какой вид предоставляет в среднем самое большое количество посадочных мест? Постройте графики.
- Выделите в отдельный столбец информацию об улице из столбца address .
- Постройте график топ-10 улиц по количеству объектов общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?
- Найдите число улиц с одним объектом общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?

Сделайте общий вывод и дайте рекомендации о виде заведения, количестве посадочных мест, а также районе расположения. Прокомментируйте возможность развития сети.

Шаг №3. Подготовка презентации

Подготовьте презентацию исследования для инвесторов.

Шаг №1. Загрузка данных и подготовка их к анализу

[Вернуться к оглавлению](#)

```
In [1]: import pandas as pd
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
from plotly import graph_objects as go
import seaborn as sns
import re
import warnings
warnings.filterwarnings("ignore", 'This pattern has match groups')
warnings.filterwarnings("ignore", 'This pattern is interpreted as a regular expression,
```

Чтение данных rest_data.csv

```
In [2]: # чтение данных
rest_data = pd.read_csv('rest_data.csv')
```

```
In [3]: # информация по датафрейму
rest_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15366 entries, 0 to 15365
```

```
Data columns (total 6 columns):
#      Column      Non-Null Count  Dtype
---  -
0     id           15366 non-null   int64
1     object_name   15366 non-null   object
2     chain          15366 non-null   object
3     object_type    15366 non-null   object
4     address        15366 non-null   object
5     number         15366 non-null   int64
dtypes: int64(2), object(4)
memory usage: 720.4+ KB
```

```
In [4]: # первые пять строк датафрейма
rest_data.head()
```

```
Out[4]:
```

	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

В столбце `chain` заменим `да/нет` на булев тип `True/False`

```
In [5]: # заменим в столбце chain да/нет на булев тип true/false
rest_data['chain'] = rest_data['chain'].replace('нет', False)
rest_data['chain'] = rest_data['chain'].replace('да', True)
```

Проверим уникальные названия типов заведений:

```
In [6]: # проверка уникальных типов заведений
rest_data['object_type'].unique()
```

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

Для удобства заменим тип заведения `магазин (отдел кулинарии)` на `кулинария`, а `предприятие быстрого обслуживания` на `фаст-фуд`.

```
In [7]: rest_data['object_type'] = rest_data['object_type'].replace('магазин (отдел кулинарии)',
rest_data['object_type'] = rest_data['object_type'].replace('предприятие быстрого обслуж
```

В датафрейме присутствуют неявные дубликаты названий заведений. Например кафе "Шоколадница" имеет 9 различных наименований:

```
In [8]: rest_data.loc[(rest_data['object_name'].str.contains('Шоколадница')), 'object_name'].uni
```

```
Out[8]: array(['Шоколадница', 'Шоколадница Кофе Хаус', 'Кофейня «Шоколадница»',
        'Кафе «Шоколадница»', 'КАФЕ «Шоколадница»', 'кафе «Шоколадница»',
        'Кофейня Шоколадница', 'Шоколадница Экспресс',
        'Шоколадница Кофемания'], dtype=object)
```

Для начала приведем все названия заведений к нижнему регистру:

```
In [9]: # приведем строки к нижнему регистру
rest_data['object_name'] = rest_data['object_name'].str.lower()
```

Некоторые строки содержат названия двух объектов. Исследуем такие записи и по возможности откорректируем их:

```
In [10]: # некоторые строки содержат названия двух объектов. очистим такие данные:

# fridays kfc
# 15233 211435 fridays kfc True предприятие быстрого обслуживания город Мо
# по адресу находится оба заведения. Но т.к. тип заведения "предприятие быстрого питания"
rest_data['object_name'] = rest_data['object_name'].replace('fridays kfc', 'kfc')

# сабвей сушивок
# 2671 124655 сабвей сушивок True кафе город Москва, Ботаническая улица, дом 29
# по адресу находится оба заведения. Но т.к. тип заведения "кафе" и сушивок позиционирую
# то уберем из строки "сушивок"
rest_data['object_name'] = rest_data['object_name'].replace('сабвей сушивок', 'сабвей')

# кафе «сабвей» старбакс
# 5742 59091 кафе «сабвей» старбакс True предприятие быстрого обслуживания
# 8022 96154 кафе «сабвей» старбакс True предприятие быстрого обслуживания
# по указанным адресам находятся заведения сабвей
rest_data['object_name'] = rest_data['object_name'].replace('кафе «сабвей» старбакс', 'сабвей')

# якитория и ян примус
# 2328 24716 якитория и ян примус True ресторан город Москва, Спартакос
# по указанному адресу находится якитория, ян приус находится в соседнем здании
rest_data['object_name'] = rest_data['object_name'].replace('якитория и ян примус', 'якитория')

# советские времена чебуречная ссср
# 13460 29742 советские времена чебуречная ссср True закусочная город Москва
# по данному адресу находится заведение советские времена
rest_data['object_name'] = rest_data['object_name'].replace('советские времена чебуречная ссср', 'советские времена')

# шоколадница кофе хаус
# 273 25141 шоколадница кофе хаус True кафе город Москва, Тверская улица, дом 27
# по данному адресу находится шоколадница
rest_data['object_name'] = rest_data['object_name'].replace('шоколадница кофе хаус', 'шоколадница')

# black burger чайхона №1
# 8450 74880 black burger чайхона №1 True ресторан город Москва, Садовая-Садовья
# по данному адресу до 2015 года была чайхона, затем там открылся burger heroes. Оставим
# burger по этому адресу нет
rest_data['object_name'] = rest_data['object_name'].replace('black burger чайхона №1', 'burger')

# павлин мавлин чайхона №1
# павлин мавлин - чайхана, а не чайхона №1
rest_data['object_name'] = rest_data['object_name'].replace('павлин мавлин чайхона №1', 'павлин мавлин')

# кафе пекарня хачапури
# 970 124103 кафе пекарня хачапури True кафе город Москва, Пятницкая улица, дом 97
# по данному адресу находится хачапурия
rest_data['object_name'] = rest_data['object_name'].replace('кафе пекарня хачапури', 'хачапури')

# хачапури, одесса-мама
# 9281 166809 хачапури, одесса-мама True ресторан город Москва, улица Шабо
# оставим сетевое заведение хачапури
rest_data['object_name'] = rest_data['object_name'].replace('хачапури, одесса-мама', 'хачапури')

# нияма. пицца пи
# пицца пи является частью сети нияма
rest_data['object_name'] = rest_data['object_name'].replace('нияма. пицца пи', 'нияма')
```

```

# граци рагацци зю кафе
# 4767 120924 граци рагацци зю кафе True ресторан город Москва, улица Новы
# оставим сетевое заведение зю кафе
rest_data['object_name'] = rest_data['object_name'].replace('граци рагацци зю кафе', 'зю

# шикари иль патио
# 7646 149858 шикари иль патио True бар город Москва, Большая Тульская у
# по данному адресу находятся оба заведения. однако в базе шикари больше не встречается.
# поэтому оставим только иль патио
rest_data['object_name'] = rest_data['object_name'].replace('шикари иль патио', 'иль пати

# метро к&к - не является заведением общепита и встречается в базе один раз. удалим стро
rest_data = rest_data.loc[rest_data['object_name'] != 'метро к&к'].reset_index(drop=True)

# суши тун, хруст pizza, кофе тун
# 11994 187626 суши тун, хруст pizza, кофе тун True кафе город Москва, Большая Ту
# суши тун - кофе тун - это одна сеть. назовем ее суши тун
rest_data['object_name'] = rest_data['object_name'].replace('суши тун, хруст pizza, кофе

# шоколадница кофемания
# 8085 25599 шоколадница кофемания True кафе город Москва, улица Арбат, дом 1
# по данному адресу находится шоколадница
rest_data['object_name'] = rest_data['object_name'].replace('шоколадница кофемания', 'шок

# кофейня кофемания, пиццерия бармалини
# 8175 24108 кофейня кофемания, пиццерия бармалини True кафе город Москва, Са
# по данному адресу находятся оба заведения. Но бармалини не является сетевым заведением
rest_data['object_name'] = rest_data['object_name'].replace('кофейня кофемания, пиццерия

# пицца паоло и бенто wok
# 15182 205739 пицца паоло и бенто wok True кафе город Москва, город Зеленоград,
# по данному адресу находятся оба заведения. Оставим в качестве кафе пицца паоло
rest_data['object_name'] = rest_data['object_name'].replace('пицца паоло и бенто wok', 'п

# баскин роббинс & стардогс
# 12036 174346 закусочная «баскин роббинс & стардогс» True закусочная город Мо
# по данному адресу находятся оба заведения. В качестве закусочной оставим стардогс
rest_data['object_name'] = rest_data['object_name'].replace('баскин роббинс & стардогс',

```

Теперь найдем записи, в которых заведения указаны в кавычках. Извлечем названия из кавычек, а остальную информацию удалим из названия.

```

In [11]: # извлечем названия в кавычках и удалим остальную информацию из строки
for i in rest_data['object_name'].index:
    line = re.findall('<«[A-zA-я0-9 -]+»>', rest_data.loc[i, 'object_name'])
    if len(line) != 0:
        rest_data.loc[i, 'object_name'] = line[0][1:-1]

```

Приведем разные названия заведений одной сети к единому названию. Чтобы избежать случайного "схлопывания" названий заведений из разных сетей, например шоколадница может объединится с кафе шоколад, исключим слова шоколад, хинкальная, сити, кофе, хачапури, гурман и кафе при азс.

```

In [12]: for line in (rest_data[rest_data['chain'] == True]['object_name'].unique()):
    if (line != 'шоколад') \
        & (line != 'хинкальная') \
        & (line != 'сити') \
        & (line != 'кофе') \
        & (line != 'хачапури') \
        & (line != 'гурман') \
        & (line != 'кафе при азс'):
        rest_data.loc[(rest_data['object_name'].str.contains(line)) & (rest_data['chain'

```

В заключение, объединим названия заведений с латинскими и русскими символами. А также исправим названия заведений, в которых есть ошибки в окончаниях или содержат пробелы и дефисы.

```
In [13]: # переименуем дубликаты названий латинские-русские названия
rest_data['object_name'] = rest_data['object_name'].replace('суши вок', 'суши wok')
rest_data['object_name'] = rest_data['object_name'].replace('кафе шоколад', 'шоколад')
rest_data['object_name'] = rest_data['object_name'].replace('сабвей', 'subway')
rest_data['object_name'] = rest_data['object_name'].replace(['братья караваевы', 'братья
rest_data['object_name'] = rest_data['object_name'].replace('иль-патио', 'иль патио')
rest_data['object_name'] = rest_data['object_name'].replace('krispy creme', 'krispy krem')
rest_data['object_name'] = rest_data['object_name'].replace('старбакс', 'starbucks')
rest_data['object_name'] = rest_data['object_name'].replace('кофетун', 'кофе тун')
rest_data['object_name'] = rest_data['object_name'].replace('maki maki', 'маки маки')
rest_data['object_name'] = rest_data['object_name'].replace('dunkin donuts', 'данкин дона
rest_data['object_name'] = rest_data['object_name'].replace('павлин-мавлин', 'павлин мавл
```

В датафрейме у заведений одной сети могут быть разные типы, например "Макдональдс":

```
In [14]: # проблема в том, что одно и то же сетевое заведение имеет разный тип
rest_data.query('object_name=="макдоналдс"]').object_type.value_counts()
```

```
Out[14]: фаст-фуд      117
ресторан      31
кафе          25
Name: object_type, dtype: int64
```

Приведем типы заведений так, чтобы у заведений одной сети был одинаковый тип. Найдем самый популярный тип заведения в сети и применим его ко всем заведениям одной сети.

```
In [15]: # функция находит самый популярный тип сетевого заведения и применяет этот тип ко всей с
for line in (rest_data[rest_data['chain'] == True]['object_name'].unique()):
    rest_data.loc[rest_data['object_name']== line, 'object_type'] = \
    rest_data.query('object_name==@line').object_type.value_counts().index[0]
```

```
In [16]: # выведем случайные сетевые заведения и проверим, что у них один тип
rest_data[rest_data['chain']==True][['object_name', 'object_type', 'chain']] \
.pivot_table(index=['object_name', 'object_type']).sample(n=10)
```

```
Out[16]:
```

		chain
object_name	object_type	
лукойл	кафе	1.0
роснефть	фаст-фуд	1.0
кофе-бин	кафе	1.0
пиццетория	кафе	1.0
burger club	кафе	1.0
изба	фаст-фуд	1.0
bubbleology	кафе	1.0
grand cru	кафе	1.0
luciano	кафе	1.0
шоколадница	кафе	1.0

```
In [17]: # Выведем уникальные названия сетевых заведений
rest_data[rest_data['chain']==True]['object_name'].unique()
```

```
Out[17]: array(['брусника', 'алло пицца', 'суши wok', 'тануки',  
      'домино'с пицца', 'готика', 'му-му', 'хлеб насущный', 'tajj mahal',  
      'данкин донатс', 'вареничная №1', 'шоколадница', 'теремок',  
      'хинкальная', 'шантимель', 'хинкальная city', 'кружка',  
      'примавера', 'виктория', 'академия', 'чебуречная ссср',  
      'макдоналдс', 'grand cru', 'чайхона №1', 'панчо пицца', 'kfc',  
      'subway', 'якитория', 'советские времена', 'андерсон', 'суши сет',  
      'шоколад', 'тирольские пироги', 'гамбринус', 'пицца фабрика',  
      'сити пицца', 'кофе хаус', 'кулинарная лавка братьев караваевых',  
      'прайм', 'пицца экспресс', 'николай', 'магнолия', 'кофе с собой',  
      'джаганнат', 'волконский', 'moskalyan', 'гино-но-таки',  
      'тратория semplice', 'ньокки', 'хижина', 'додо пицца',  
      'крошка картошка', 'бургер кинг', 'папа джонс', 'две палочки',  
      'джон джоли', 'ваби-саби', 'кофемания', 'простые вещи',  
      'павлин мавлин', 'тапчан', 'штолле', 'бабай клаб', 'кактус',  
      'темпл бар', 'хлеб&со', 'кофе-бин', 'французская выпечка',  
      'планета суши', 'де марко', 'илья муромец', 'тарас бульба',  
      'гудман', 'иль патио', 'мюнгер', 'ботик петра', 'чин чин',  
      'правда кофе', 'сити', 'krispy krem', 'азбука вкуса', 'пивко',  
      'брудер', 'мимино', 'кофепорт', 'цинандали хинкальная',  
      'иль forno', 'travelers coffe', 'рецептор', 'сушишоп', 'кофе-хаус',  
      'баскин роббинс', 'барашка', 'пицца хат', 'мята', 'costa coffee',  
      'ёрш', 'стардого', 'upside down', 'хачапурия',  
      'домашнее кафе сеть городских кафе', 'добрынинский и партнёры',  
      'менза', 'колбасофф', 'крепери де пари', 'коста кофе', 'starbucks',  
      'венечия', 'жан жак', 'гурмания', 'bierloga', 'штирбирлиц',  
      'маки-маки', 'хачапури', 'в&в бургер', 'jeffreys coffee',  
      'дабл би', 'золотая вобла', 'пилзнер', 'перекресток', 'руккола',  
      'милано пицца', 'зодиак', 'торро гриль', 'world class', 'роллофф',  
      'кулинарное бюро', 'билла', 'урюк', 'tokyo bay', 'пиппони',  
      'пицца пипони', 'сытая утка', 'икура паб', 'сварня',  
      'пиццерия пиу дель чибо', 'ичибан боши', 'макс бреннер',  
      'бутчер бизон', 'магбургер', 'prime', 'то да сё', 'пицца pomodoro',  
      'козловица', 'грабли', 'вьеткафе', 'оникс', 'да пино',  
      'старина миллер', 'хинкальная №1', 'пронто', 'паоло', 'лепешка',  
      'гурман', 'территория', 'городские автокофейни', 'порто мальтезе',  
      'квартира 44', 'хинкальная кинто', 'суп кафе', 'florentini',  
      'поль бейкери', 'glowsubs sandwiches', 'ми пьяче',  
      'пиццерия донателло', 'нияма', 'марукамэ', 'бир хаус', 'спб',  
      'кебаб хаус', 'paul поль', 'шашлык-машлык', 'генацвали', 'зю кафе',  
      'виват-пицца', 'starlite diner', 'кофе тун', 'ташир пицца',  
      'ресторан хинкальная', 'елки-палки', 'тамаси суши', 'burger club',  
      'фантоцци рус', 'изба', 'воккер', 'ганс и марта', 'кофе',  
      'ливан-хаус', 'лето', 'ямми микс', 'баракат', 'кафе при азс',  
      'космик', 'сушиман', 'япоша', 'барбарис', 'yogurt frenzy',  
      'bosconscino', 'помидор', 'дюшес', 'урожай', 'ян примус',  
      'шварцвальд', 'tutti frutti', 'меленка', 'сбарро',  
      'восточный базар', 'каро', 'white rabbite (белый кролик)',  
      'бакинский бульвар', 'healthy food', 'sushilka', 'coffeeshop',  
      'маки маки', 'барбекю', 'обжорный ряд', 'тайм авеню', 'wok & box',  
      'кафе при азс газпромнефть', 'торнадо', 'бенто wok', 'бургер клаб',  
      'br', 'wokker', 'ариана', 'ванвок', 'il patio', 'пражечка',  
      'блинная', 'correas', 'мистер картошка', 'сим-сим', 'югос',  
      'марчеллис', 'кафе песто и митлес', 'молли гвинз',  
      'дорогая я перезвоню', 'm safe хинкальная', 'marrakesh хинкальная',  
      'the terrace', 'пончиковое кафе икеа ikea', 'ресторан икеа ikea',  
      'black & white', 'роснефть', 'cookhouse',  
      'мск московская сеть кальянных', 'sofix', 'пикколо', 'милти',  
      'васаби', 'грузинская кухня эзо хинкальная',  
      'хачапури, одесса -мама', 'fridays', 'panda express',  
      'ирландский паб', 'сказка', 'вкусняшка', 'суши тун', 'кофейня',  
      'лукойл', 'мираторг', 'bubbleology', 'boobo', 'luciano',  
      'кальянная f-lounge', 'beverly hills diner', 'lavkalavka',  
      'хлебница пекарня', 'osteria mario', 'мясорооб',  
      'газпромнефтьцентр', 'my box', 'сувлаки', 'movenpick', 'marmalato'],  
      dtype=object)
```

```
In [18]: # проверка явных дубликатов строк
print(
    'Количество явных дубликатов равно:',
    rest_data.duplicated().sum()
)
```

Количество явных дубликатов равно: 0

```
In [19]: # проверка неявных дубликатов
print(
    'Количество неявных дубликатов равно:',
    rest_data[['object_name', 'chain', 'object_type', 'address', 'number']].duplicated()
)

rest_data = rest_data[~(rest_data[['object_name', 'chain', 'object_type', 'address', 'number']].duplicated())]
rest_data.reset_index(drop=True)
```

Количество неявных дубликатов равно: 88

```
In [20]: rest_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15277 entries, 0 to 15276
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   id              15277 non-null  int64
 1   object_name     15277 non-null  object
 2   chain           15277 non-null  bool
 3   object_type     15277 non-null  object
 4   address         15277 non-null  object
 5   number          15277 non-null  int64
dtypes: bool(1), int64(2), object(3)
memory usage: 611.8+ KB
```

Описание данных

Таблица rest_data:

- **id** — идентификатор объекта;
- **object_name** — название объекта общественного питания;
- **chain** — сетевой ресторан;
- **object_type** — тип объекта общественного питания;
- **address** — адрес;
- **number** — количество посадочных мест.

В таблице rest_data 15277 записей.

Пропусков и явных дубликатов не обнаружено. Без учета ID было обнаружено и удалено 88 дубликатов.

Тип данных в столбце chain был заменен на bool. Тип данных в остальных столбцах оставлен как есть.

Названия столбцов соответствуют стилистическим нормам и оставлены как есть.

Названия сетевых заведений были обработаны, приведены к единому названию: убраны кавычки, названия из латинских и русских символов приведены к единому формату, тип заведений установлен одинаковым для всей сети.

Чтение данных из внешнего источника

```
In [21]: # считываем данные с информацией по районам из файла mosgaz-streets.csv
streets = pd.read_csv('https://hubofdata.ru/dataset/4fee7193-2ead-4a49-ac2d-63928ba7a0f9')
```



```
In [22]: # информация по датафрейму
streets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4398 entries, 0 to 4397
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   streetname  4398 non-null   object
1   areaid      4398 non-null   int64
2   okrug       4398 non-null   object
3   area        4398 non-null   object
dtypes: int64(1), object(3)
memory usage: 137.6+ KB
```

```
In [23]: # переименуем название столбцов для удобства объединения по столбцу street
streets.columns = ['street', 'areaid', 'okrug', 'area']
```

```
In [24]: # проверка дубликатов строк
streets.duplicated().sum()
```

```
Out[24]: 8
```

```
In [25]: # удаление дубликатов
streets = streets.drop_duplicates().reset_index(drop=True)
streets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4390 entries, 0 to 4389
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  -
0   street  4390 non-null   object
1   areaid  4390 non-null   int64
2   okrug   4390 non-null   object
3   area    4390 non-null   object
dtypes: int64(1), object(3)
memory usage: 137.3+ KB
```

В таблице streets 4390 записей.

Пропусков не обнаружено. Дубликаты удалены.

Типы данных в столбцах оставлены как есть. Названия столбцов соответствуют стилистическим нормам.

Шаг №2. Анализ данных

Соотношение видов объектов общественного питания по количеству.

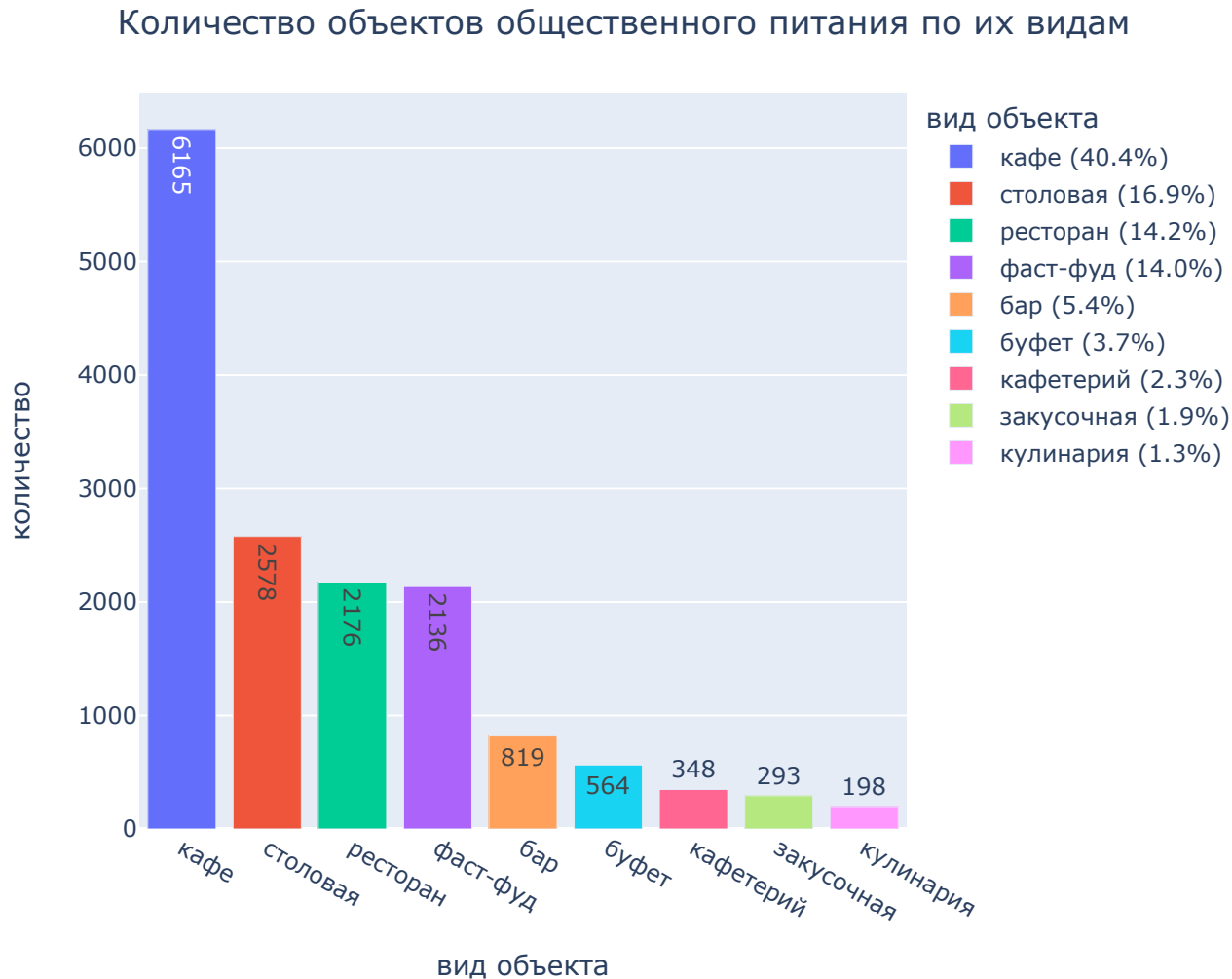
[Вернуться к оглавлению](#)

```
In [26]: # построим столбчатые диаграммы по видам заведений
fig = px.bar(
    rest_data.groupby('object_type', as_index=False).agg({'id': 'count'})[['object_type',
    x='object_type',
    y='id',
    color='object_type',
    labels=dict(object_type='вид объекта', id='количество'),
    text = 'id'
    )
```

```

fig.update_layout(
    title={'text': 'Количество объектов общественного питания по их видам', 'x': 0.5}
)
for trace, percent in zip(fig.data, (rest_data['object_type'].value_counts()/len(rest_da
    trace.name = trace.name + ' (' + percent + ') '
fig.show()

```

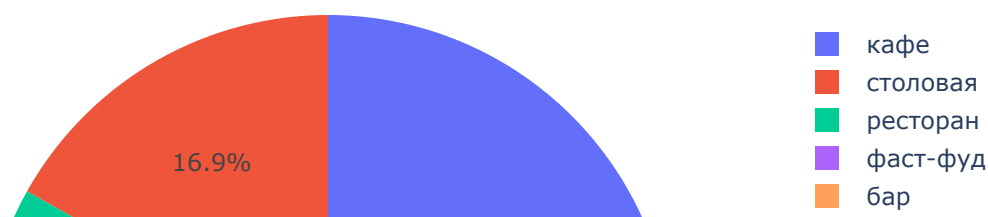


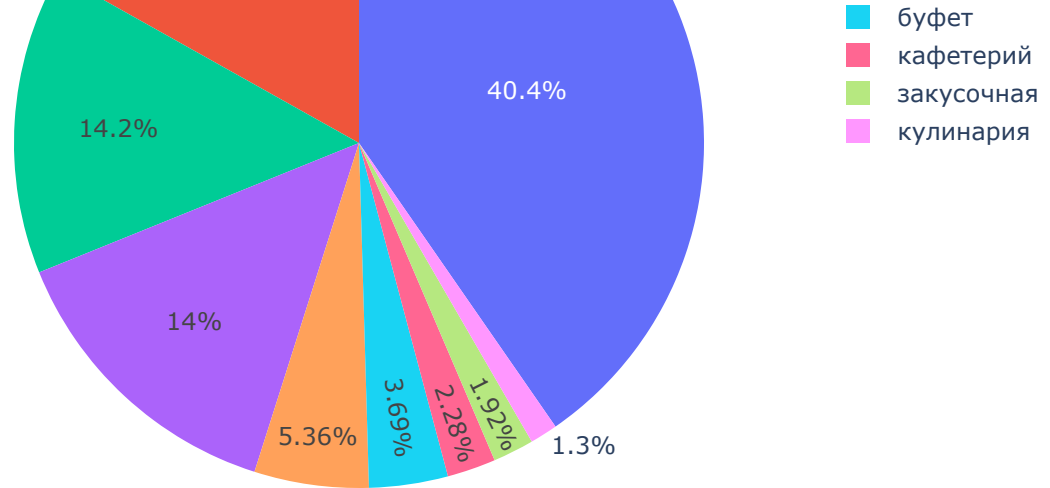
```

In [27]: # построим круговую диаграмму по видам заведений
fig = go.Figure(
    data=go.Pie(
        labels=rest_data['object_type'].value_counts().reset_index()['index'],
        values=rest_data['object_type'].value_counts()
    )
)
fig.update_layout(
    title={'text': 'Соотношение объектов общественного питания по их видам', 'x': 0.5}
)
fig.show()

```

Соотношение объектов общественного питания по их видам





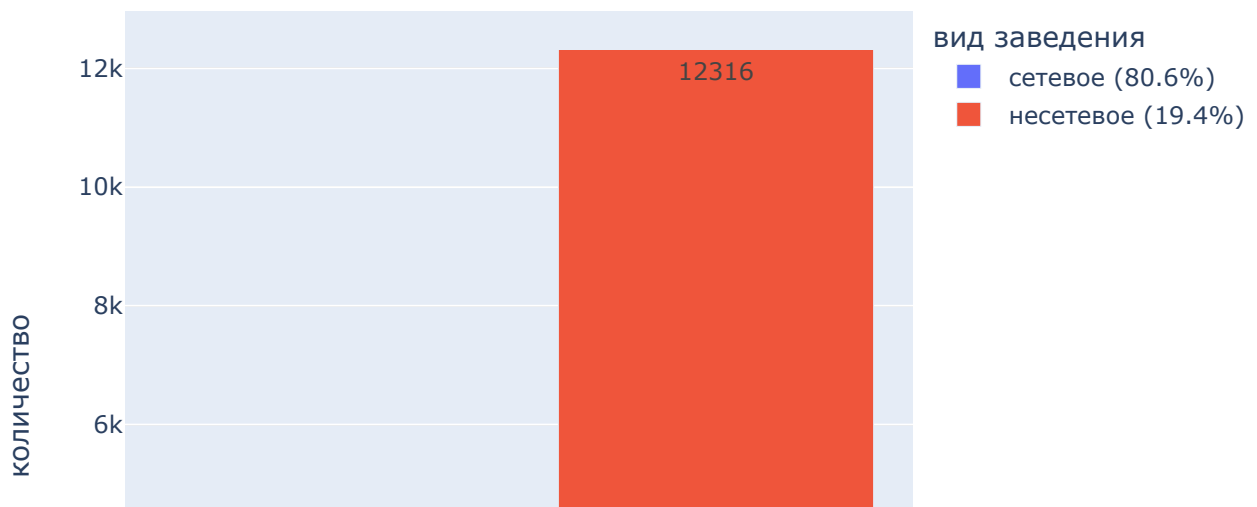
Из графиков следует, что больше всего заведений общественного питания вида кафе(40.5%), столовые(16.9%) и рестораны(14.3%). Кулинарии представлены в наименьшем количестве.

Соотношение сетевых и несетевых заведений по количеству.

[Вернуться к оглавлению](#)

```
In [28]: # столбчатая диаграмма сетевых / несетевых заведений
fig = px.bar(
    rest_data.groupby('chain', as_index=False).agg({'id': 'count'})[['chain', 'id']].sort(
        x=['сетевое', 'несетевое'],
        y='id',
        color=['сетевое', 'несетевое'],
        labels=dict(x='вид заведения', id='количество', color='вид заведения'),
        text = 'id'
    )
fig.update_layout(
    title={'text': 'Соотношение сетевых и несетевых заведений', 'x': 0.5}
)
for trace, percent in zip(fig.data, (rest_data['chain'].value_counts() / len(rest_data) * 100)):
    trace.name = trace.name + ' (' + percent + '%)'
fig.show()
```

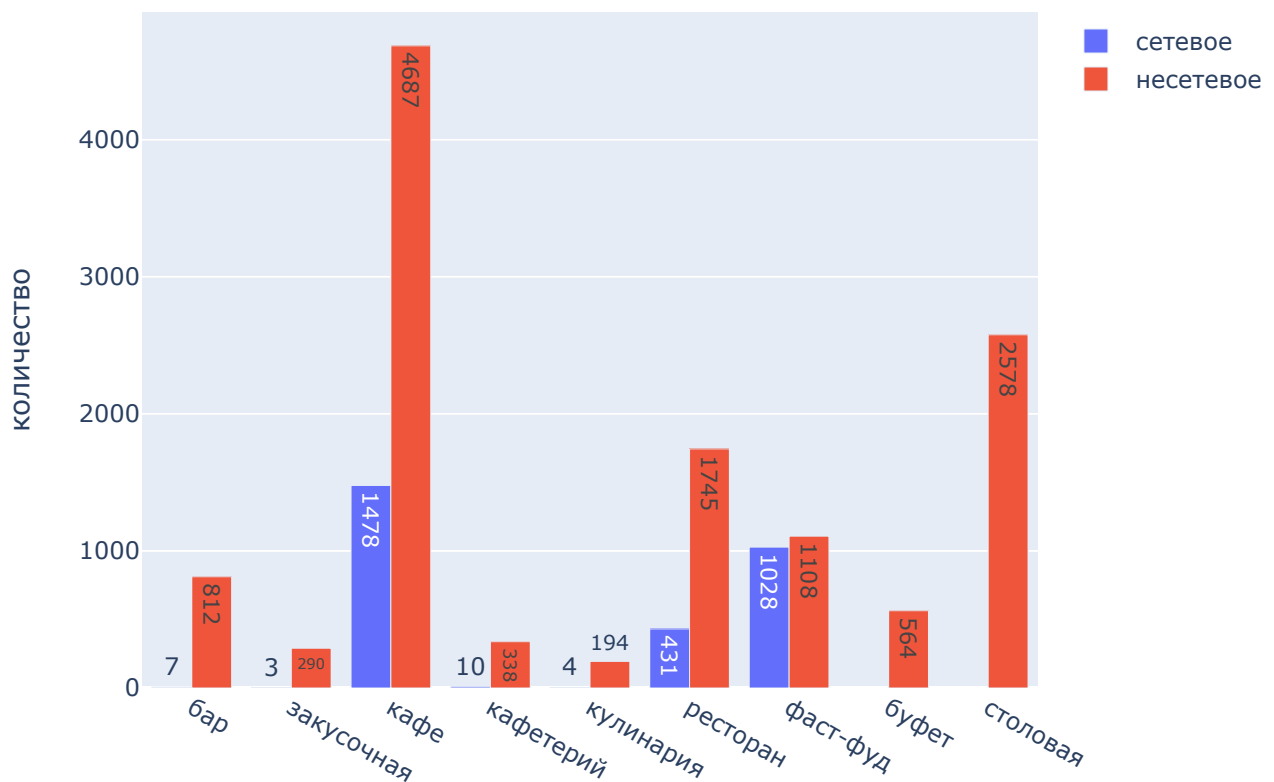
Соотношение сетевых и несетевых заведений





```
In [29]: # столбчатая диаграмма сетевых/несетевых заведений по категориям
fig = go.Figure()
fig.add_trace(go.Bar(
    x = rest_data[rest_data['chain']==True].groupby('object_type').agg('count')['id'].index,
    y = rest_data[rest_data['chain']==True].groupby('object_type').agg('count')['id'],
    text = rest_data[rest_data['chain']==True].groupby('object_type').agg('count')['id'].values,
    name='сетевое'
))
fig.add_trace(go.Bar(
    x = rest_data[rest_data['chain']==False].groupby('object_type').agg('count')['id'].index,
    y = rest_data[rest_data['chain']==False].groupby('object_type').agg('count')['id'],
    text = rest_data[rest_data['chain']==False].groupby('object_type').agg('count')['id'].values,
    name='несетевое'
))
fig.update_layout(
    title={'text': 'Количество сетевых и несетевых заведений по количеству и виду', 'x':0},
    xaxis_title="вид заведения",
    yaxis_title="количество"
)
fig.show()
```

Количество сетевых и несетевых заведений по количеству и виду



Несетевых заведений почти в четыре раза больше чем сетевых. Всего несетевых 12316 заведений, сетевых - 2961. По категориям несетевых заведений также больше.

При этом сетевые заведения в основном представлены в виде кафе, ресторанов, фаст-фудов.

Несетевых кафе больше сетевых в 3,2 раза.

Несетевых ресторанов больше сетевых в 4,0 раза.

Несетевых фаст-фудов больше сетевых в 1,1 раза.

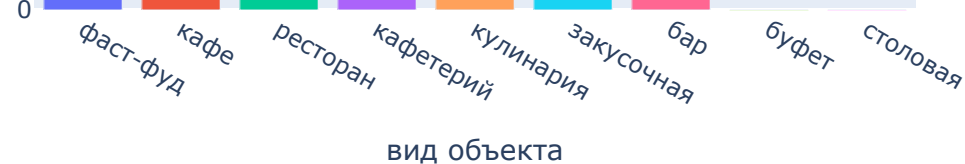
Анализ наиболее распространенных видов сетевых заведений

[Вернуться к оглавлению](#)

```
In [30]: # соберем датафрейм с % сетевых заведений по их типам
chain_rate = pd.DataFrame(rest_data[rest_data['chain']==True]['object_type'].value_counts()
                           value_counts()*100).round(1).sort_values(by='object_type', asc
chain_rate.columns = ['object_type', 'rate']
chain_rate['rate'] = chain_rate['rate'].fillna(0)
```

```
In [31]: # построим столбчатые диаграммы по % заведений
fig = px.bar(
    chain_rate,
    x='object_type',
    y='rate',
    color='object_type',
    labels=dict(object_type='вид объекта', rate='% заведений'),
    text = chain_rate['rate'].astype('str')+'%'
)
fig.update_layout(
    title={'text': 'Количество объектов общественного питания по их видам', 'x':0.5}
)
fig.show()
```





Сетевая принадлежность наиболее характерна для:

- фаст-фудов - 47,7%
- кафе - 23,8%
- ресторанов - 20,7%.

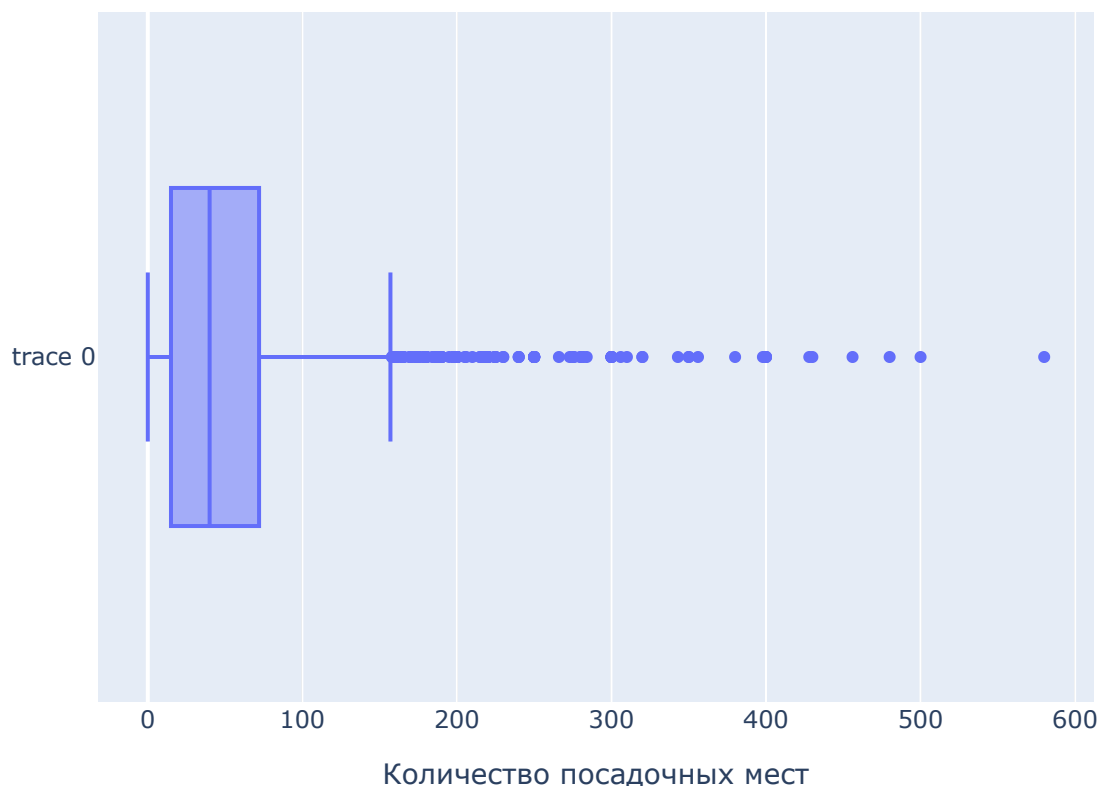
Столовые и буфеты обычно не представлены в виде сетевых заведений. Это связано с тем, что данные типы заведений являются объектами обслуживающих производств и хозяйств и находятся на территории предприятий, бизнес-центров, в учреждениях культуры и досуга и т.д.

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

[Вернуться к оглавлению](#)

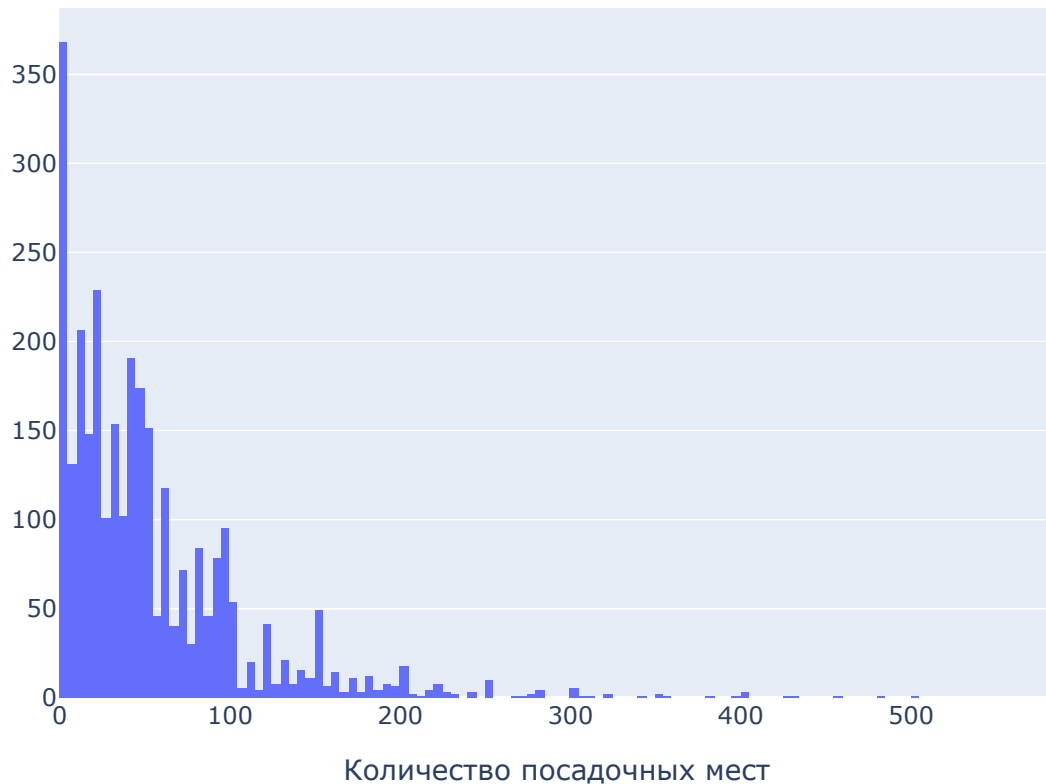
```
In [32]: # построим диаграмму размаха посадочных мест
fig = go.Figure(go.Box(x=rest_data[rest_data['chain']==True]['number']))
fig.update_layout(title=('text': 'Диаграмма размаха посадочных мест сетевых заведений', '
                        xaxis_title="Количество посадочных мест"))
fig.show()
```

Диаграмма размаха посадочных мест сетевых заведений



```
In [33]: # построим гистограмму посадочных мест сетевых заведений
fig = go.Figure(go.Histogram(x=rest_data[rest_data['chain']==True]['number']))
fig.update_layout(title={'text': 'Гистограмма посадочных мест сетевых заведений', 'x':0.5},
                  xaxis_title="Количество посадочных мест")
fig.show()
```

Гистограмма посадочных мест сетевых заведений



```
In [34]: # определим категории кол-ва заведений и кол-ва посадочных мест
chain_type = rest_data[rest_data['chain']==True].groupby('object_name', as_index=False).
            agg({'id':'count', 'number':'median'})

chain_type.columns = ['object_name', 'count', 'number']

# примем границу разделения много-мало 50% квантиль:
q_count = np.percentile(chain_type['count'], 50)
q_number = np.percentile(rest_data[rest_data['chain']==True]['number'], 50)

def chain_sort(row):
    global q_count
    global q_number
    if (row['count'] > q_count) & (row['number'] > q_number):
        return 'Много заведений и много посадочных мест'
    if (row['count'] > q_count) & (row['number'] <= q_number):
        return 'Много заведений и мало посадочных мест'
    if (row['count'] <= q_count) & (row['number'] > q_number):
        return 'Мало заведений и много посадочных мест'
    if (row['count'] <= q_count) & (row['number'] <= q_number):
        return 'Мало заведений и мало посадочных мест'

chain_type['category'] = chain_type.apply(chain_sort, axis=1)
```

```
print('Медианное количество заведений:', q_count)
print('Медианное количество посадочных мест:', q_number)
```

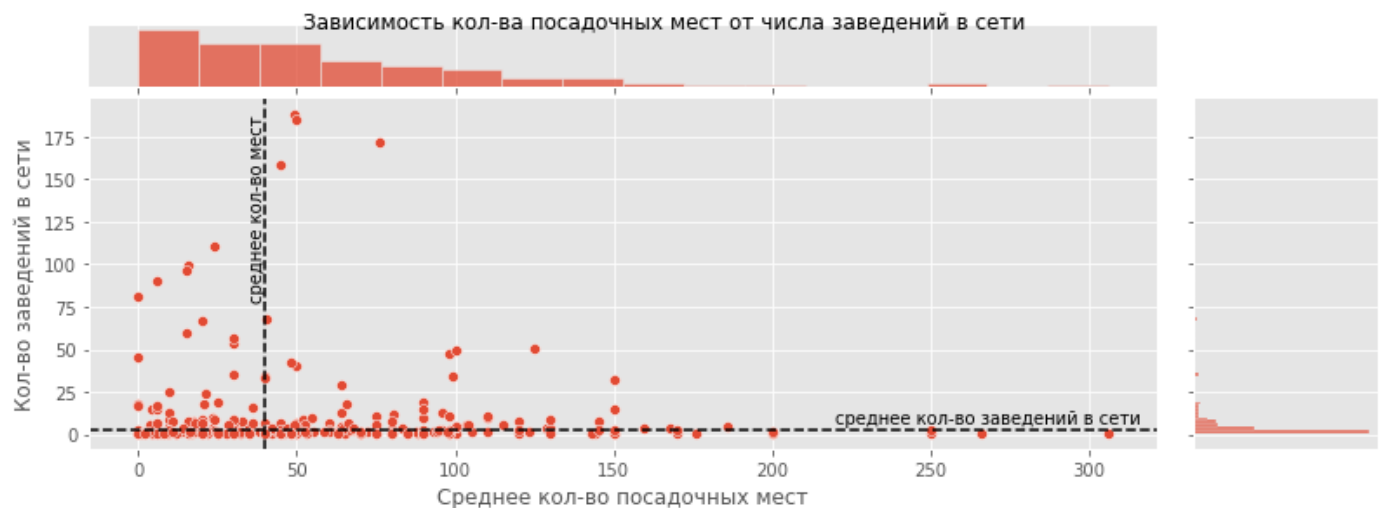
```
chain_type['category'].value_counts()
```

```
Медианное количество заведений: 3.0
Медианное количество посадочных мест: 40.0
Мало заведений и много посадочных мест      91
Мало заведений и мало посадочных мест        76
Много заведений и много посадочных мест      58
Много заведений и мало посадочных мест       51
Name: category, dtype: int64
```

Out[34]:

In [35]:

```
# Построим совместную диаграмму зависимости числа мест от количества заведений в сети
plt.style.use('ggplot')
p = sns.jointplot(x='number', y='object_name', data=rest_data[rest_data['chain']==True].
                  agg({'object_name': 'count', 'number': 'median'}))
p.ax_joint.axhline(np.percentile(chain_type['count'], 50), linestyle='--', color='black')
p.ax_joint.axvline(np.percentile(rest_data[rest_data['chain']==True]['number'], 50), lin
p.ax_joint.text(220, 7, 'среднее кол-во заведений в сети')
p.ax_joint.text(35, 80, 'среднее кол-во мест', rotation=90)
p.set_axis_labels('Среднее кол-во посадочных мест', 'Кол-во заведений в сети')
p.fig.suptitle('Зависимость кол-ва посадочных мест от числа заведений в сети')
p.fig.subplots_adjust(top=0.95)
p.fig.set_figwidth(12)
p.fig.set_figheight(4)
```



Из анализа следует, что для сетевых заведений характерно мало заведений с большим числом посадочных мест в каждом. Медианное количество мест в заведении равно 40. Медианное количество заведений в сети равно трем.

Для каждого вида объекта общественного питания опишите среднее количество посадочных мест. Какой вид предоставляет в среднем самое большое количество посадочных мест?

[Вернуться к оглавлению](#)

In [36]:

```
# построим столбчатые диаграммы по видам заведений
fig = px.bar(
    rest_data.groupby('object_type', as_index=False).agg({'number': 'mean'})[['object_type',
    x='object_type',
    y='number',
    color='object_type',
    labels=dict(object_type='тип заведения', number='среднее количество мест'),
    text = rest_data.groupby('object_type', as_index=False).agg({'number': 'mean'})[['numb
```

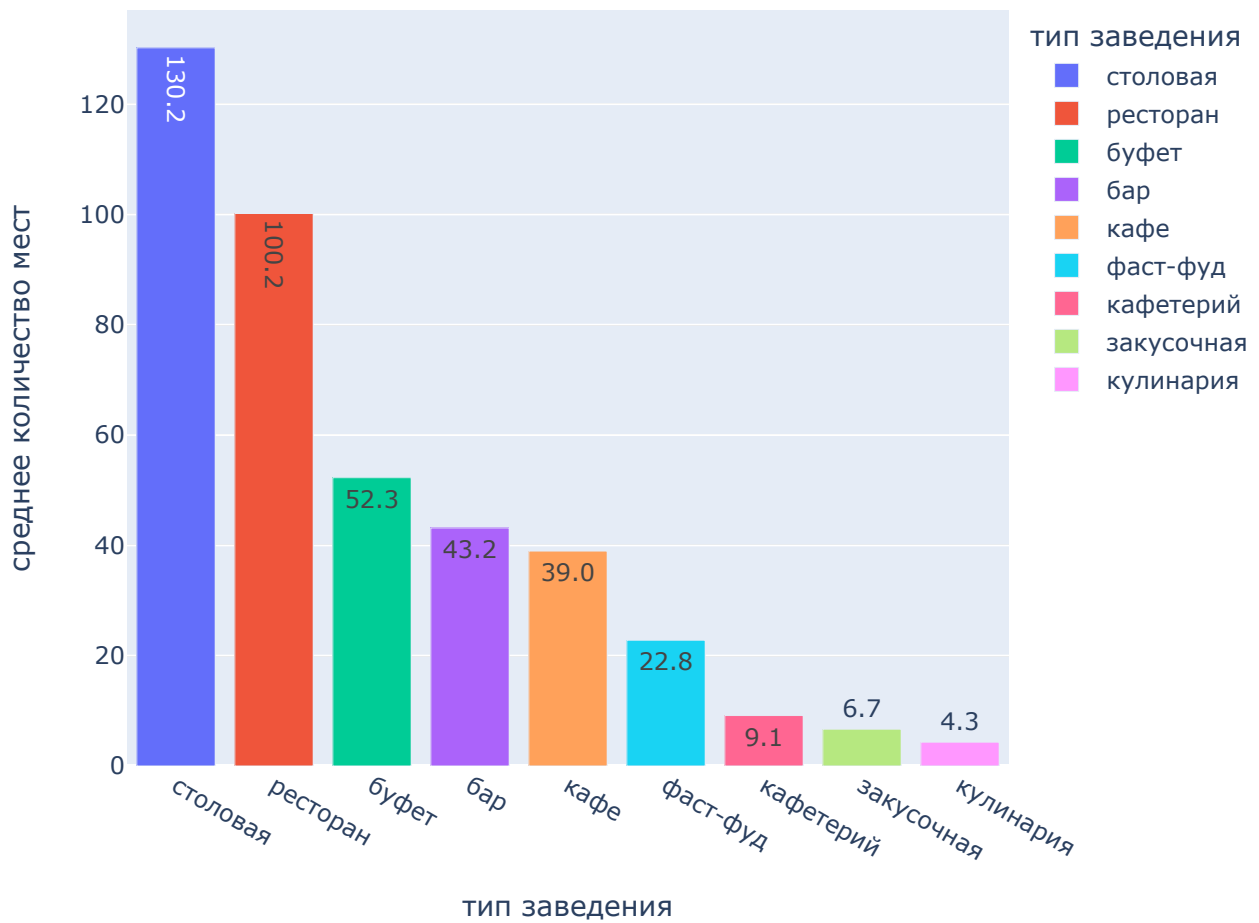


```

        .sort_values(ascending=False).astype('str')
    )
    fig.update_layout(
        title={'text': 'Среднее количество посадочных мест в заведении', 'x': 0.5}
    )
    fig.show()

```

Среднее количество посадочных мест в заведении

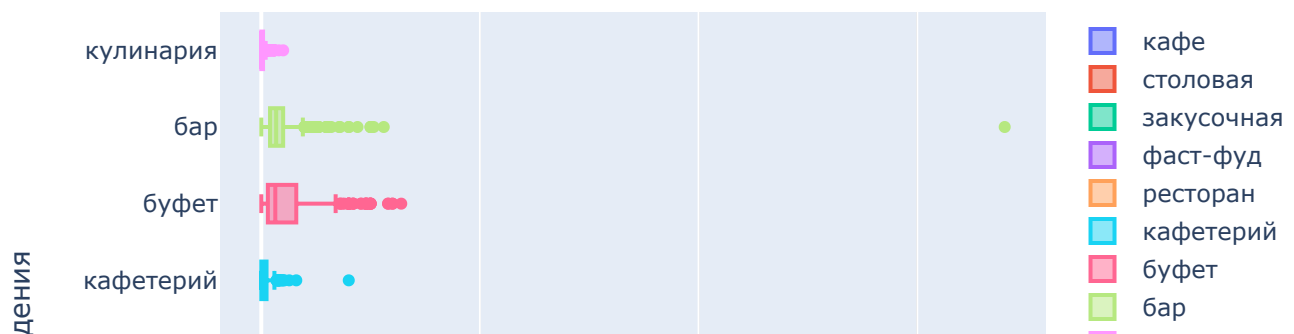


```

In [37]: # построим диаграммы разброса посадочных мест по типам заведений
fig = go.Figure()
for i in list(rest_data['object_type'].unique()):
    fig.add_trace(go.Box(x=rest_data.query('object_type==@i')['number'], name=i))
fig.update_layout(title={'text': 'Среднее количество посадочных мест в заведении', 'x': 0.5},
                  yaxis_title="Тип заведения",
                  xaxis_title="Количество посадочных мест")
fig.show()

```

Среднее количество посадочных мест в заведении





```
In [38]: # описание количества мест по типам заведений
rest_data.groupby('object_type')['number'].describe().sort_values(by='mean')
```

	count	mean	std	min	25%	50%	75%	max
object_type								
кулинария	198.0	4.323232	8.105683	0.0	0.0	0.0	4.0	50.0
закусочная	293.0	6.675768	16.325897	0.0	0.0	0.0	8.0	216.0
кафетерий	348.0	9.146552	15.062170	0.0	0.0	6.0	12.0	200.0
фаст-фуд	2136.0	22.804307	39.403825	0.0	0.0	6.0	30.0	580.0
кафе	6165.0	38.970479	37.058004	0.0	15.0	30.0	50.0	533.0
бар	819.0	43.158730	68.000965	0.0	20.0	34.0	50.0	1700.0
буфет	564.0	52.271277	56.713413	0.0	15.0	32.0	80.0	320.0
ресторан	2176.0	100.165901	96.587944	0.0	48.0	80.0	120.0	1500.0
столовая	2578.0	130.206749	94.964693	0.0	50.5	103.0	200.0	1400.0

В среднем наибольшее количество посадочных мест имеют столовые - 130 мест и рестораны - 100 мест. Наименьшее количество мест имеют кулинарии и закусочные.

Выделите в отдельный столбец информацию об улице из столбца address

[Вернуться к оглавлению](#)

```
In [39]: # выделим названия улиц в отдельный столбец
rest_data['street'] = rest_data['address'].apply(lambda x: x.split(',')[1])
rest_data['street'] = rest_data['street'].map(str.strip)
rest_data.head()
```

	id	object_name	chain	object_type	address	number	street
0	151635	сметана	False	кафе	город Москва, улица Егора Абакумова, дом 9	48	улица Егора Абакумова
1	77874	родник	False	кафе	город Москва, улица Талалихина, дом 2/1, корпус 1	35	улица Талалихина
2	24309	академия	False	кафе	город Москва, Абельмановская улица,	95	Абельмановская

3	21894	пиццетория	True	кафе	город Москва, Абрамцевская улица, дом 1	40	Абрамцевская улица
4	119365	вишневая метель	False	кафе	город Москва, Абрамцевская улица, дом 9, корпус 1	50	Абрамцевская улица

Постройте график топ-10 улиц по количеству объектов общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?

[Вернуться к оглавлению](#)

```
In [40]: # выберем топ-10 улиц по количеству заведений
top10_streets = rest_data.groupby('street').agg({'id': 'count'}).sort_values(by = 'id',
top10_streets = top10_streets[top10_streets['street'] != 'город Зеленоград']
top10_streets = top10_streets[top10_streets['street'] != 'поселение Сосенское']
top10_streets = top10_streets.head(10)
top10_streets.columns = ['street', 'number']
```

```
In [41]: # выведем график топ-10 улиц
fig = px.bar(
    top10_streets,
    x='street',
    y='number',
    color='street',
    labels=dict(street='улица', number='количество'),
    title='График топ-10 улиц по количеству объектов общественного питания',
    text = top10_streets['number']
)
fig.show()
```

График топ-10 улиц по количеству объектов общественного питания



```
In [42]: # выделим номера зданий для определения их положения на улице
building = rest_data['address'].apply(lambda x: re.findall('\d+', x.split('дом')[-1]))
for i in building.index:
    if len(building[i]) == 0:
        building[i] = np.nan
    else:
        building[i] = building[i][0]

rest_data['building'] = building.astype(int)
```

```
In [43]: # получим датасет с топ-10 улиц и их адресами
rest_sort = rest_data[(rest_data['street']=='проспект Мира') | \
    (rest_data['street']=='Профсоюзная улица') | \
    (rest_data['street']=='Ленинградский проспект') | \
    (rest_data['street']=='Пресненская набережная') | \
    (rest_data['street']=='Варшавское шоссе') | \
    (rest_data['street']=='Ленинский проспект') | \
    (rest_data['street']=='проспект Вернадского') | \
    (rest_data['street']=='Кутузовский проспект') | \
    (rest_data['street']=='Каширское шоссе') | \
    (rest_data['street']=='Кировоградская улица')]
```

```
In [44]: # выведем плотность распределения зданий вдоль улицы
fig = px.strip(rest_sort, x='building', y='street', color='street')
fig.update_layout(title={'text': 'Плотность распределения заведений вдоль улицы', 'x': 0.5},
    yaxis_title="улица",
    xaxis_title="номер здания",
    showlegend=False)
```

Плотность распределения заведений вдоль улицы



```
In [45]: # объединим топ-10 улиц с датафреймом мосгаза для получения районов
streets_okrug = top10_streets.merge(streets, on='street', how='left')
```

```
In [46]: # список районов, в которых расположены топ 10 улиц по заведениям
streets_okrug['area'].unique()
```

```
Out[46]: array(['Алексеевский район', 'Ярославский Район', 'Район Марьино',
                'Останкинский район', 'Район Ростокино', 'Район Свиблово',
                'Мещанский район', 'Академический район', 'Район Черемушки',
                'Район Ясенево', 'Район Коньково', 'Обручевский район',
                'Район Теплый Стан', 'Район Аэропорт', 'Район Беговой',
                'Хорошевский район', 'Район Сокол', 'Пресненский район',
                'Район Чертаново Центральное', 'Район Чертаново Северное',
                'Район Чертаново Южное', 'Донской район',
                'Район Нагатинно-Садовники', 'Нагорный район',
                'Район Северное Бутово', 'Район Южное Бутово',
                'Район Проспект Вернадского', 'Район Тропарево-Никулино',
                'Район Гагаринский', 'Ломоносовский район', 'Район Якиманка',
                'Район Раменки', 'Район Дорогомилово', 'Район Фили-Давыдково',
                'Район Москворечье-Сабурово', 'Район Орехово-Борисово Южное',
                'Район Орехово-Борисово Северное'], dtype=object)
```

Из графиков следует, что наибольшее количество объектов располагается на улицах:

- проспект Мира (203 заведения)
- Профсоюзная улица (181 заведение)
- Ленинградский проспект (170 заведений)

При определении положения объекта на улице следует учитывать плотность заведений вдоль улицы. Наиболее часто расположены заведения:

- на проспекте Мира между зданиями 26-79 и 91-128
- на Профсоюзной улице между зданиями 2-27, 56-83 и 96-132
- на Ленинградском проспекте между зданиями 9-17, 30-48 и 55-80

Найдите число улиц с одним объектом общественного питания. Воспользуйтесь внешней информацией и ответьте на вопрос — в каких районах Москвы находятся эти улицы?

[Вернуться к оглавлению](#)

```
In [47]: # получим список улиц с одним заведением и объединим с датафреймом мосгаза для получения
one_rest_street = rest_data.groupby('street').agg({'id': 'count'}).sort_values(by = 'id')
one_rest_street.columns = ['street', 'number']
one_rest_okrug = one_rest_street.merge(streets, on='street', how='left')
one_rest_okrug = one_rest_okrug.loc[one_rest_okrug['number']==1]
one_rest_okrug = one_rest_okrug.dropna().reset_index(drop=True)
print('Число улиц с одним объектом общественного питания равно:', len(one_rest_okrug))
```

Число улиц с одним объектом общественного питания равно: 560

```
In [48]: # список районов, в которых расположены топ 10 улиц по заведениям
one_rest_okrug['area'].unique()
```

```
Out[48]: array(['Район Кунцево', 'Район Аэропорт', 'Мещанский район',
                'Район Кузьминки', 'Район Измайлово', 'Район Соколиная Гора',
                'Район Сокольники', 'Район Южное Бутово', 'Район Марьино',
                'Нижегородский район', 'Район Якиманка', 'Район Солнцево',
```

```
'Район Свиблдово', 'Район Митино', 'Район Хорошево-Мневники',  
'Район Люблино', 'Район Выхино-Жулебино', 'Басманный район',  
'Район Покровское-Стрешнево', 'Район Северное Бутово',  
'Рязанский район', 'Красносельский район', 'Тверской район',  
'Район Сокол', 'Бутырский район', 'Тимирязевский Район',  
'Даниловский район', 'Хорошевский район', 'Район Замоскворечье',  
'Район Печатники', 'Район Внуково', 'Район Лефортово',  
'Район Дорогомилово', 'Таганский район', 'Район Перово',  
'Бескудниковский Район', 'Район Щукино', 'Район Богородское',  
'Район Раменки', 'Район Хамовники', 'Район Ростокино',  
'Район Текстильщики', 'Алексеевский район', 'Район Арбат',  
'Район Преображенское', 'Донской район', 'Останкинский район',  
'Войковский Район', 'Пресненский район', 'Район Северный',  
'Район Фили-Давыдково', 'Район Очаково-Матвеевское',  
'Район Филевский Парк', 'Лосиноостровский район',  
'Савеловский район', 'Район Чертаново Южное', 'Южнопортовый Район',  
'Головинский район', 'Район Царицыно', 'Район Косино-Ухтомский',  
'Район Москворечье-Сабурово', 'Район Гольяново',  
'Район Ново-Переделкино', 'Район Ясенево',  
'Район Северное Измайлово', 'Район Восточное Измайлово',  
'Можайский Район', 'Район Котловка', 'Район Беговой',  
'Ярославский Район', 'Бабушкинский район',  
'Район Чертаново Северное', 'Алтуфьевский район',  
'Район Новогиреево', 'Дмитровский район', 'Район Коптево',  
'Нагорный район', 'Район Зюзино', 'Район Куркино',  
'Район Бирюлево Восточное', 'Район Западное Дегунино',  
'Район Ивановское', 'Район Марьино', 'Район Левобережный',  
'Молжаниновский район', 'Район Южное Тушино', 'Район Отрадное',  
'Район Гагаринский', 'Район Матушкино-Савелки', 'Район Строгино',  
'Район Черемушки', 'Район Северное Медведково', 'Район Восточный',  
'Район Нагатино-Садовники', 'Район Метрогородок', 'Район Коньково',  
'Район Марфино', 'Обручевский район', 'Район Южное Медведково'],  
dtype=object)
```

Получен список из 560 улиц, на которых находится по одному заведению общественного питания. При определении положения будущего заведения следует избегать выбора этих улиц, как наименее популярных среди посетителей.

Шаг №3. Подготовка презентации

[Вернуться к оглавлению](#)

Презентация: https://disk.yandex.ru/i/_JhQyognRcfzWg

Выводы

[Вернуться к оглавлению](#)

С целью подготовить исследование рынка заведений общественного питания в Москве и дать рекомендации о виде заведения, количестве посадочных мест, а также районе расположения в данном анализе были проделаны следующие задачи:

- исследованы соотношения видов объектов общественного питания по количеству и построены графики;
- исследованы соотношения сетевых и несетевых заведений по количеству и построены графики;
- определены виды объектов общественного питания, для которых характерно сетевое распространение;

- определено параметры, характерные для сетевых заведений;
- для каждого вида объекта общественного питания описано среднее количество посадочных мест. Проведена оценка, какой вид заведений предоставляет в среднем самое большое количество посадочных мест и построены графики;
- выделена в отдельный столбец информация об улице из столбца address;
- построен график топ-10 улиц по количеству объектов общественного питания и определены районы, в которых находятся эти улицы;
- найдено число улиц с одним объектом общественного питания и определены районы, в которых находятся эти улицы.

По результатам анализа было определено, что больше всего заведений общественного питания вида кафе(40.3%), столовые(16.9%) и рестораны(14.4%). Кулинарии представлены в наименьшем количестве.

Несетевых заведений почти в четыре раза больше чем сетевых. Всего несетевых 12316 заведений, сетевых - 2961. По категориям несетевых заведений также больше.

При этом сетевые заведения в основном представлены в виде кафе, ресторанов, фаст-фудов.

- Несетевых кафе больше сетевых в 3,2 раза.
- Несетевых ресторанов больше сетевых в 4,0 раза.
- Несетевых фаст-фудов больше сетевых в 1,1 раза.

Сетевая принадлежность наиболее характерна для:

- фаст-фудов - 47,7%
- кафе - 23,8%
- ресторанов - 20,7%.

Столовые и буфеты обычно не представлены в виде сетевых заведений. Это связано с тем, что данные типы заведений являются объектами обслуживающих производств и хозяйств и находятся на территории предприятий, бизнес-центров, в учреждениях культуры и досуга и т.д.

Для сетевых заведений характерно мало заведений с большим числом посадочных мест в каждом. Медианное количество мест в заведении равно 40. Медианное количество заведений в сети равно трем.

В среднем наибольшее количество посадочных мест имеют столовые - 130 мест и рестораны - 100 мест. Наименьшее количество мест имеют кулинарии и закусочные.

Наибольшее количество объектов располагается на улицах:

- проспект Мира (203 заведения)
- Профсоюзная улица (181 заведение)
- Ленинградский проспект (170 заведений)

При определении положения объекта на улице следует учитывать плотность заведений вдоль улицы. Наиболее плотно находятся заведения:

- на проспекте Мира между зданиями 26-79 и 91-128
- на Профсоюзной улице между зданиями 2-27, 56-83 и 96-132
- на Ленинградском проспекте между зданиями 9-17, 30-48 и 55-80

Рекомендации по выбору типа заведения

При определении типа заведения, в котором гостей обслуживают роботы, наиболее

предпочтительным является кафе со средним количеством мест – 38.

Можно запланировать создание сети кафе до трех заведений.

Для выбора района рекомендуется рассмотреть топ-10 улиц с наибольшим количеством заведений, т.к. кафе на этих улицах пользуются спросом у населения. Наиболее популярные улицы: проспект Мира, Профсоюзная улица, Ленинградский проспект. При выборе места объекта необходимо учитывать плотность распределения заведений. Вероятно, что зоны с наибольшей плотностью заведений наиболее популярны среди посетителей.