Оглавление

Исследование рынка заведений общественного питания Москвы. =

Проведем исследование рынка заведений общественного питания Москвы. Основание - датасет с заведениями общественного питания Москвы, составленный на основе данных сервисов Яндекс Карты и Яндекс Бизнес на лето 2022 года. Информация, размещённая в сервисе Яндекс Бизнес, могла быть добавлена пользователями или найдена в общедоступных источниках. Она носит исключительно справочный характер.

Откроем файл с данными и изучим информацию

In [1]:
```
#pip install matplotlib --upgrade
#!pip install folium
```

In [2]:
```
#pip install pandas --upgrade
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
import random as r
import numpy as np
import folium
from folium.plugins import MarkerCluster
import warnings
warnings.filterwarnings(action = 'ignore')
pd.options.display.max_columns = 40
```

In [4]:
```
# загрузим набор с данными, посмотрим информацию о датасете.
df = pd.read_csv('C:/Users/User/Downloads/moscow_places.csv',encoding = 'utf8')
```

In [5]:
```
# посмотрим на первые 10 строк
df.head(10)
```

Out[5]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_avg_bi
0	WoWфли	кафе	Москва, улица Дыбенко, 7/1	Северный административный округ	ежедневно, 10:00–22:00	55.878494	37.478860	5.0	NaN	NaN	Na
1	Четыре комнаты	ресторан	Москва, улица Дыбенко, 36, корп. 1	Северный административный округ	ежедневно, 10:00–22:00	55.875801	37.484479	4.5	выше среднего	Средний счёт:1500–1600 ₽	1550.
2	Хазри	кафе	Москва, Клязьминская улица, 15	Северный административный округ	пн-чт 11:00–02:00; пт,сб 11:00–05:00; вс 11:00...	55.889146	37.525901	4.6	средние	Средний счёт:от 1000 ₽	1000.
3	Dormouse Coffee Shop	кофейня	Москва, улица Маршала Федоренко, 12	Северный административный округ	ежедневно, 09:00–22:00	55.881608	37.488860	5.0	NaN	Цена чашки капучино:155–185 ₽	Na
4	Иль Марко	пиццерия	Москва, Правобережная улица, 1Б	Северный административный округ	ежедневно, 10:00–22:00	55.881166	37.449357	5.0	средние	Средний счёт:400–600 ₽	500.
5	Sergio Pizza	пиццерия	Москва, Ижорская улица, вл8Б	Северный административный округ	ежедневно, 10:00–23:00	55.888010	37.509573	4.6	средние	NaN	Na
6	Огни города	бар,паб	Москва, Клязьминская улица, 9, стр. 3	Северный административный округ	пн 15:00–04:00; вт-вс 15:00–05:00	55.890752	37.524653	4.4	средние	Средний счёт:199 ₽	199.
7	Mr. Уголёк	быстрое питание	Москва, Клязьминская улица, 9, стр. 3	Северный административный округ	пн-чт 10:00–22:00; пт,сб 10:00–23:00; вс 10:00...	55.890636	37.524303	4.7	средние	Средний счёт:200–300 ₽	250.
8	Donna Maria	ресторан	Москва, Дмитровское шоссе, 107, корп. 4	Северный административный округ	ежедневно, 10:00–22:00	55.880045	37.539006	4.8	средние	Средний счёт:от 500 ₽	500.
9	Готика	кафе	Москва, Ангарская улица, 39	Северный административный округ	ежедневно, 12:00–00:00	55.879038	37.524487	4.3	средние	Средний счёт:1000–1200 ₽	1100.

In [6]:
```
# выведем информацию о столбцах и типах данных
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8406 entries, 0 to 8405
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   name               8406 non-null   object 
 1   category           8406 non-null   object 
 2   address            8406 non-null   object 
 3   district           8406 non-null   object 
 4   hours              7870 non-null   object 
 5   lat                8406 non-null   float64
 6   lng                8406 non-null   float64
 7   rating             8406 non-null   float64
 8   price              3315 non-null   object 
 9   avg_bill           3816 non-null   object 
 10  middle_avg_bill    3149 non-null   float64
 11  middle_coffee_cup  535 non-null    float64
 12  chain              8406 non-null   int64  
 13  seats              4795 non-null   float64
dtypes: float64(6), int64(1), object(7)
memory usage: 919.5+ KB
```

In [7]:
```
# проверим данные на явные дубликаты, 0 - дубликаты отсутствуют
df.duplicated().sum()
```

Out[7]:
```
0
```

- Вывод:

В данных представлено 8406 заведений, с указанием информации о категории заведения `category` (тип данных object), адрес заведений `address` , с указанием администртивного округа `district` (тип данных object), ширина и долгота координат нахождения `lat` , `lng` (число с плавающей точкой, тип данных float64), рейтинг `rating` (число с плавающей точкой, тип данных float64) и принадлежность к сетевым/несетевым заведениям `chain` (числовой тип int64). Столбец `price` информация о категории цен в заведении, содержит пропуски (тип данных object). Средняя стоимость заказа в виде диапазона

`avg_bill` (тип данных object) содержит пропуски. Столбец оценкой среднего чека `middle_avg_bill` (тип данных float64), которое указано только для значений из столбца avg_bill, начинающихся с подстроки «Средний счёт». Столбец `middle_coffee_cup` (float64) — число с оценкой одной чашки капучино, которое указано только для значений из столбца avg_bill, начинающихся с подстроки «Цена одной чашки капучино». Число посадочных мест `seats`, тип данных float64, содержит пропуски значений.

Предобработка данных

In [8]:
```
df.query("category == 'кафе' and middle_avg_bill > 3000")
```

Out[8]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_avg_b
1125	На мангале	кафе	Москва, проспект Мира, 119, стр. 503	Северо-Восточный административный округ	ежедневно, 10:00–21:00	55.830405	37.617687	1.0	высокие	Средний счёт:4000–5000 ₽	450
1800	Promenade Lounge	кафе	Москва, Ленинградский проспект, 36, стр. 33	Северный административный округ	ежедневно, круглосуточно	55.787416	37.566735	4.4	высокие	Средний счёт:3000–4000 ₽	350
7391	Рай-ON	кафе	Москва, Донецкая улица, 33	Юго-Восточный административный округ	ежедневно, 12:00–01:00	55.644529	37.711499	4.1	высокие	Средний счёт:2500–5000 ₽	375

Пропуски

In [9]:
```
# нипишем функцию для визуализации пропусков
def null_value(data):
    try:
        (
            (data.isna().mean()*100)
            .to_frame()
            .rename(columns = {0:'space'})
            .query('space > 0')
            .sort_values(by = 'space', ascending = True)
            .plot(kind= 'barh', figsize=(20,5), rot = 0, legend = False, fontsize = 16)
            .set_title('Пропуски данных' + "\n", fontsize = 22, color = 'SteelBlue')
        );
    except:
        print('Пропусков нет')
```

In [10]:
```
# применим функцию к датафрейму
null_value(df)
```

Пропуски в столбцах `seats, hour, avg_bill, middle_avg_bill, middle_coffee_cup` невозможно заполнить, так как требуется дополнительная информация. Пропуски столбца `price` при информации в столбце `middle_avg_bill` имеются, возможно можно заполнить пропуски. Посмотрим на распределение имеющихся значений `middle_avg_bill` в разрезе категорий `price`

In [11]:
```
#построим сводную таблицу,сгруппируем данные по категориям столбца price и посчитаем некоторые метрики столбца
df.pivot_table(index = 'price', values = 'middle_avg_bill', \
               aggfunc = ['min','max','mean','median',lambda x:x.quantile(.25)])\
.rename(columns ={'min': 'Минимум', 'max': 'Максимум','mean': 'Среднее', 'median':'Медиана', \
                  '<lambda>':'1-Квартиль','middle_avg_bill':'' }).sort_values(by = 'Минимум')
```

Out[11]:

	Минимум	Максимум	Среднее	Медиана	1-Квартиль
price					
высокие	0.0	35000.0	2472.631579	2000.0	1750.0
низкие	90.0	600.0	217.311828	180.0	150.0
средние	165.0	2150.0	598.908273	500.0	350.0
выше среднего	375.0	4500.0	1344.189189	1250.0	1250.0

Видим пересечение минимумов и максимумов, большой разброс значений, корректное заполнение пропусков в столбце `price` не представляется возможным, так как принцип категоризации столбца не основан на распределении значений столбца `middle_avg_bill` ,при заполнении пропусков результат исследования может исказиться.

Неявные дубликаты

- Рассмотрим столбец `name`. Для оценки различной синтаксической стилистики выберем 20 уникальных наименований

In [12]:
```
# посчитаем количество уникальных значений столбца
df['name'].nunique()
```

Out[12]:
```
5614
```

In [13]:
```
# выберем 20 уникальных значений
r.choices(df['name'].unique(), k = 20)
```

Out[13]:
```
['Пош бар',
 'Пиво пицца',
 'Время Кофе',
 'Ё-ланч',
 'Гастробар БеловЪ',
 'Кебар нон стоп',
 'DelonixCafe',
 'Пироги по-домашнему, Халяль',
 'Чайхана24',
 'Кафе Ипполит Матвеевич',
 'Человек и кофе',
 'Кебаб-хауз',
 'Ритм и Вкус',
 'Вкус кофе',
 'Moriarty Bar&Kitchen',
 'Пицца фабрика',
 'Шейх Палас',
 'Кофе On',
 'Нам',
 'Yura Restaurant & Bar']
```

Очевидны различия в регистрах, приведем к одному виду, исправим некоторые синтаксические различия

In [14]:
```
# заменим первый символ слова на заглавную букву
df['name'] = df['name'].str.title()
```

In [15]:
```
# исправим синтаксические различия
df['name'] = df['name'].str.replace('\'', '’')
df['name'] = df['name'].str.replace(' ’', '’')
df['name'] = df['name'].str.replace(' - ', '-')
df['name'] = df['name'].str.replace(' & ', '&')
df['name'] = df['name'].str.replace('’S', '’s')
```

Поиск грамматических ошибок или различий написания по всему списку уникальных значений столбца `name` не является целесообразным, так как названия заведений могут повторяться, иметь похожие названия и в то же время не быть связанными друг с другом. Имеет смысл проверить названия заведений, отмеченных как сети, то есть имеющих значение равное 1 в столбце `chain`

In [16]:
```
# создадим список уникальных наименований сетевых(согласно данным столбца chain) заведений
name = df.loc[df['chain'] == 1, 'name'].unique()
# создадим словарь dict, куда ключами добавим части строк начинающиеся с заглавной буквы,
# а значениями будут количество повторений такой строки во всех строках столбца name
dict = {}
for uni_name in name:
    for d in range(len(uni_name)):
        if uni_name[d].isupper():
            dict[uni_name[d:]] = dict.get(uni_name[d:],0)+1
        else:
            continue
# далее создадим результирующий словарь result, в котором ключами будут - ключи словаря dict со значением больш
# (более 2 найденных повторений в значениях подстроки), а значениями список наименований заведений , где встреч
```

```
result = {}
for key in dict:
    if dict[key]>=2:
        for j in sorted(name):
            if key in j:
                result[key] = result.get(key,[]) + [j]
```

In [17]:
```
#выведем полученный словарь, поищем неявные дубликаты наименований, при необходимости используем поисковую сист
# уточнения предположений
result
```

Out[17]:
```
{'Буханка': ['Буханка', 'Пекарня Буханка'],
 'Пекарня': ['Кафе-Пекарня',
  'Маленькая Пекарня Журавлевых',
  'Пекарня',
  'Пекарня Буханка',
  'Пекарня № 1',
  'Французская Пекарня'],
 'Город': ['Городок', 'Городское', 'Кушай Город', 'Старый Город'],
 'Пицца': ['Алло! Пицца',
  'Виват Пицца',
  'Додо Пицца',
  'Домино'С Пицца',
  'Нью-Йорк Пицца И Гриль',
  'Пицца И Канноли',
  'Пицца На Районе',
  'Пицца Паоло',
  'Пицца Экспресс',
  'Пиццаменто',
  'Сити Пицца',
  'Суши-Пицца 312',
  'Ташир Пицца',
  'Чао-Пицца'],
 'Coffee': ['9 Bar Coffee',
  'Abc Coffee Roasters',
  'Air Coffee',
  'Americano Black Coffee&Food',
  'Coffee And The City',
  'Coffee Bean',
  'Coffee Break',
  'Coffee Guru',
  'Coffee In',
  'Coffee Like',
  'Coffee Moose',
  'Coffee Music',
  'Coffee Party',
  'Coffee Point',
  'Coffee Way',
  'Coffeebar'17',
  'Coffeebrain',
  'Coffeekaldi's',
  'Coffeeshop Company',
  'Coffeeshots',
  'Coffeesphere',
  'Crop. Coffee&Smoothie Bar',
  'Demi Coffee Shop',
  'Gentleman Coffee',
  'Hq! Coffee',
  'Jeffrey's Coffee',
  'Jeffrey's Coffeeshop',
  'Kaya Coffee Shop',
  'Noba Coffee',
  'Omg Coffee',
  'One Price Coffee',
  'Skuratov, Coffee Roasters',
  'Sova Coffee',
  'You&Coffee'],
 'Шашлык': ['Мир Шашлыков',
  'Хочу Шашлык',
  'Шашлык',
  'Шашлык Хаус',
  'Шашлыки',
  'Шашлыкoff',
  'Шик Шашлык'],
 'Веранда': ['Веранда', 'Летняя Веранда'],
 'Халяль': ['Халяль', 'Чайхана Халяль'],
 'Хаус': ['Гриль Хаус',
  'Донер Хаус',
  'Лагман Хаус',
  'Ливан Хаус',
  'Масала Хаус',
  'Шашлык Хаус'],
 'Суши': ['Кофетун-Сушитун',
```

```
 'Островок Суши',
 'Суши Love',
 'Суши Wok',
 'Суши Сет',
 'Суши Таун',
 'Суши-Пицца 312',
 'Сушистор',
 'Я Люблю Суши'],
'Pho': ['Ho Chu Pho',
 'Pho',
 'Pho Bo',
 'Pho City',
 'Pho Hanoi',
 'Pho Ngon',
 'Pho Oanh',
 'Pho Street',
 'Pho U',
 'Pho Viet',
 'Phobo',
 'Vua Pho',
 'Wok Pho Mi'],
'Бургер': ['Бургер Кинг',
 'Рубим Бургер',
 'Стейк&Бургер',
 'Франклинс Бургер',
 'Шеф Бургер'],
'Лавка': ['Дагестанская Лавка',
 'Кулинарная Лавка Братьев Караваевых',
 'Турецкая Лавка',
 'Яндекс Лавка',
 'Яндекс.Лавка'],
'Wild Bean Cafe': ['The Wild Bean Cafe', 'Wild Bean Cafe'],
'Bean Cafe': ['The Wild Bean Cafe', 'Wild Bean Cafe'],
'Cafe': ['Cafe',
 'Cafe Inn',
 'Cassette Cafe',
 'Grao De Cafe',
 'Ipho Cafe',
 'Saperavi Cafe',
 'Star Hit Cafe',
 'Sub Cafe',
 'The Wild Bean Cafe',
 'Well’s Home Cafe',
 'White Fox Cafe',
 'Wild Bean Cafe'],
'На Углях': ['Мясо На Углях', 'Шаурма На Углях'],
'Углях': ['Мясо На Углях', 'Шаурма На Углях'],
'Кебаб': ['Донер Кебаб', 'Хан Кебаб'],
'Ош': ['Ош', 'Чайхана Ош'],
'Bean': ['Coffee Bean',
 'Milk&Beans',
 'The Wild Bean Cafe',
 'Wild Bean',
 'Wild Bean Cafe'],
'Кухня': ['Cvc Китайская Кухня',
 'Восточная Кухня',
 'Вьетнамская Кухня',
 'Грузинская Кухня',
 'Китайская Кухня',
 'Кухня Полли',
 'Японская Кухня'],
'Lounge': ['Библиотека Shisha Lounge', 'Мск Lounge', 'Мята Lounge'],
'Дворик': ['Бакинский Дворик',
 'Дворик',
 'Старый Дворик',
 'Шашлычный Дворик',
 'Южный Дворик'],
'Самарканд': ['Самарканд', 'Самарканд Сити', 'Чайхана Самарканд'],
'Легенда': ['Легенда', 'Хинкальная Легенда'],
'City': ['City Life', 'Coffee And The City', 'Pho City'],
'Ngon': ['Ngon', 'Pho Ngon', 'Viet Ngon'],
'Айва': ['Айва', 'Чайхона Айва'],
'Баракат': ['Баракат', 'Чайхана Баракат'],
'Выпечка': ['Выпечка',
 'Горячая Выпечка',
 'Свежая Выпечка',
 'Хлеб Да Выпечка'],
'Food': ['Americano Black Coffee&Food',
 'Arcus Bar And Food',
 'Food Low Cost Sushi',
 'Foodband.Ru',
 'Halal Food',
 'Life Food'],
```

```
'Bakery': ['Bakery',
 'French Bakery',
 'French Bakery Sedelice',
 'Max Bakery',
 'Remy Kitchen Bakery'],
'Паб': ['Изи Паб', 'Кружка Паб'],
'Вкус': ['В Парке Вкуснее',
 'Вкус Востока',
 'Вкус Дня',
 'Вкус Индии',
 'Галерея Вкуса',
 'Домашний Вкус',
 'Столичный Вкус'],
'Chick': ['Chicko', 'Korean Chick', 'Топ Chick'],
'Ланч': ['Ё-Ланч', 'Гурмэ Ланч', 'Ланч Поинт'],
'Бар': ['Бар-Ресторан Территория',
 'Баракат',
 'Барбарис',
 'Барбариста',
 'Кафе Бар',
 'Кафе-Бар',
 'Праймбиф Бар',
 'Чайхана Баракат'],
'Co': ['9 Bar Coffee',
 'Abc Coffee Roasters',
 'Air Coffee',
 'Americano Black Coffee&Food',
 'Cofefest',
 'Coffee And The City',
 'Coffee Bean',
 'Coffee Break',
 'Coffee Guru',
 'Coffee In',
 'Coffee Like',
 'Coffee Moose',
 'Coffee Music',
 'Coffee Party',
 'Coffee Point',
 'Coffee Way',
 'Coffeebar’17',
 'Coffeebrain',
 'Coffeekaldi’s',
 'Coffeeshop Company',
 'Coffeeshots',
 'Coffeesphere',
 'Coffprice',
 'Cofix',
 'Conversation',
 'Cosmic Latte',
 'Crop. Coffee&Smoothie Bar',
 'Demi Coffee Shop',
 'Dimsum&Co',
 'Food Low Cost Sushi',
 'Free&Co',
 'Gentleman Coffee',
 'Hq! Coffee',
 'Jeffrey’s Coffee',
 'Jeffrey’s Coffeeshop',
 'Kaya Coffee Shop',
 'Noba Coffee',
 'Omg Coffee',
 'One Price Coffee',
 'Plov.Com',
 'Skuratov, Coffee Roasters',
 'Sova Coffee',
 'You&Coffee',
 'Хлеб&Co'],
'Кофе': ['Бамбл Кофе',
 'Кофе',
 'Кофе Пью',
 'Кофе С Собой',
 'Кофе Твой Друг',
 'Кофе Хауз',
 'Кофе&Moloko',
 'Кофедей',
 'Кофейник',
 'Кофемания',
 'Кофепорт',
 'Кофети',
 'Кофетун-Сушитун',
 'Может, Кофе?',
 'Правда Кофе',
 'Роко Бэй — Мох И Кофе',
```

```
 'Твой Кофе'],
'Еда': ['Да, Еда', 'Домашняя Еда', 'Еда', 'Еда Greek'],
'Ссср': ['Ссср', 'Чебуречная Ссср'],
'Кафе': ['Бизнес-Кафе',
 'Вьетнамское Кафе',
 'Кафе Бар',
 'Кафе-Бар',
 'Кафе-Пекарня',
 'Кафе-Столовая',
 'Кафетерий',
 'Кафетериус',
 'Кафетеррия',
 'Кафешка',
 'Литературное Кафе',
 'Мое Кафе',
 'Моё Кафе',
 'Музейное Кафе',
 'Семейное Кафе',
 'Столовая-Кафе Росинка',
 'Татнефть Кафе',
 'Эль Кафе'],
'Халва': ['Халва', 'Халва, Сеть Почтоматов', 'Чайхана Халва'],
'Манас': ['Манас', 'Чайхана Манас'],
'В Пите': ['Донер В Пите', 'Шаурма В Пите'],
'Пите': ['Донер В Пите', 'Питербургер', 'Шаурма В Пите'],
'Очаг': ['Очаг', 'Семейный Очаг'],
'Хауз': ['Кофе Хауз', 'Пирог Хауз'],
'Двор': ['Бакинский Дворик',
 'Дворик',
 'Марков Двор',
 'Старый Дворик',
 'Шашлычный Двор',
 'Шашлычный Дворик',
 'Южный Дворик'],
'House': ['Poke House', 'Shawarma Vip House'],
'Pizza': ['Camorra Pizza E Birra',
 'Easy Pizza',
 'Frankie Pizza',
 'Pizza Express 24',
 'Pizza Hut',
 'Zotman Pizza'],
'Трапеза': ['Монастырская Трапеза', 'Поминальная Трапеза'],
'Go': ['Asia Gourmet', 'Deli2Go', 'Raw To Go'],
'Точка': ['Индийская Точка', 'Точка'],
'Poke': ['More Poke', 'Poke House', 'Wave California Poke'],
'Sedelice': ['French Bakery Sedelice', 'Sedelice'],
'Bar': ['9 Bar Coffee',
 'Arcus Bar And Food',
 'Boston Seafood&Bar',
 'Crop. Coffee&Smoothie Bar',
 'Temple Bar'],
'Burgers': ['Bb&Burgers', 'The Best Burgers'],
'Coffee Shop': ['Demi Coffee Shop', 'Kaya Coffee Shop'],
'Shop': ['Demi Coffee Shop', 'Kaya Coffee Shop'],
'Брусника': ['Брусника', 'Кондитерская-Кулинария Брусника'],
'Парк': ['В Парке Вкуснее', 'Гриль Парк', 'Парк'],
'Ли': ['Лао Ли',
 'Ли',
 'Ливан Хаус',
 'Лимонадница',
 'Линдфорс',
 'Литературное Кафе'],
'Coffee Roasters': ['Abc Coffee Roasters', 'Skuratov, Coffee Roasters'],
'Roasters': ['Abc Coffee Roasters', 'Skuratov, Coffee Roasters'],
'Пиццы': ['Бюро Пиццы', 'Империя Пиццы'],
'Лагман': ['Лагман', 'Лагман Хаус', 'Уйгурский Лагман'],
'Баку': ['Баку', 'Старый Баку'],
'Burger': ['Bb&Burgers',
 'Black Star Burger',
 'Burger Club',
 'Burger Heroes',
 'Manny’s Burger',
 'The Best Burgers'],
'Вино': ['Есть Хинкали&Пить Вино',
 'Хачапури И Вино',
 'Хинкали И Вино',
 'Хлеб И Вино'],
'И Вино': ['Хачапури И Вино', 'Хинкали И Вино', 'Хлеб И Вино'],
'Китайская Кухня': ['Cvc Китайская Кухня', 'Китайская Кухня'],
'Сити': ['Самарканд Сити', 'Сити Лайф', 'Сити Пицца', 'Чайхана Бишкек Сити'],
'Экспресс': ['Пицца Экспресс', 'Хинкальная Экспресс']}
```

In [18]:
```
# произведем замену
```

```
df['name'] = df['name'].str.replace('Чайхона','Чайхана')
df['name'] = df['name'].replace('Пекарня Буханка','Буханка')
df['name'] = df['name'].replace('Домино'С Пицца','Домино'с Пицца')
df['name'] = df['name'].replace('Чайхана Халяль','Халяль')
df['name'] = df['name'].replace('Jeffrey's Coffee','Jeffrey's Coffeeshop')
df['name'] = df['name'].replace('Phobo','Pho Bo')
df['name'] = df['name'].replace('Яндекс Лавка','Яндекс.Лавка')
df['name'] = df['name'].replace('Wild Bean Cafe','The Wild Bean Cafe')
df['name'] = df['name'].replace('Phobo','Pho Bo')
df['name'] = df['name'].replace('Чайхана Айва','Айва')
df['name'] = df['name'].replace('Чайхана Чайхана Айва','Айва')
df['name'] = df['name'].replace('Чайхана Баракат','Баракат')
df['name'] = df['name'].replace('Мое Кафе','Моё Кафе')
df['name'] = df['name'].replace('French Bakery Sedelice','Sedelice')
df['name'] = df['name'].replace('Кондитерская-Кулинария Брусника','Брусника')
```

In [19]:
```
# проверим количество уникальных значений
df['name'].nunique()
```

Out[19]: 5487

В столбце `name` найдены неявные дубликаты, произведенена работа по замене на корректные названия

- Произведем поиск синтаксических ошибок в столбце `address`

In [20]:
```
# посчитаем количество уникальных значений столбца
df['address'].nunique()
```

Out[20]: 5753

In [21]:
```
# выведем случайные значения столбца для оценки возможных ошибок
r.choices(df['address'].unique(), k = 20)
```

Out[21]:
```
['Москва, Конный переулок, 4',
 'Москва, улица Искры, 17А, стр. 3',
 'Москва, улица Дмитрия Ульянова, 9А, стр. 1',
 'Москва, Расторгуевский переулок, 3А',
 'Москва, улица Красного Маяка, 16, стр. 2',
 'Москва, Новощукинская улица, 10, корп. 1',
 'Москва, Варшавское шоссе, 47, корп. 4',
 'Москва, улица Римского-Корсакова, 11, корп. 1',
 'Москва, Большая Сухаревская площадь, 16/18с2',
 'Москва, 2-й Кабельный проезд, 1, корп. 2',
 'Москва, улица Мичуринский Проспект, Олимпийская Деревня, 4, корп. 3',
 'Москва, Кутузовский проспект, 24',
 'Москва, Балаклавский проспект, 16А',
 'Москва, Тверская улица, 27, стр. 2',
 'Москва, улица 1905 года, 10с1',
 'Москва, Большая Садовая улица, 5, корп. 1',
 'Москва, 1-й Хорошёвский проезд, 16, корп. 1',
 'Москва, Тверская улица, 27, стр. 2',
 'Москва, Звенигородская улица, 8, корп. 1',
 'Москва, Дербеневская набережная, 11, корп. В']
```

Напишем функцию для замены 'к' на 'корп.' и 'с' на 'стр.', там где это необходимо

In [22]:
```
# функция принимает строку с адресом,разбивает ее по разделителю "," и  создает список с элементами, проверяет,
# содержит ли последний элемент списка буквенно - цифровое значение или символ(проверяем, что элемент содержит
# номер корпуса и/или строения) и производит замену 'к' на  'корп.' и 'с' на 'стр.', там где необходимо, и возвр
# исправленную  строку
def chang(row):
    l = row.split(',')
    if (all(x.isalpha() or x.isalnum() for x in l[-1].strip())) or ('/' in l[-1]):
        if 'с' in l[-1] and 'стр' not in l[-1]:
            l[-1] = l[-1].replace('с', ', стр. ')
        elif 'к' in l[-1] and 'корп' not in l[-1]:
            l[-1] = l[-1].replace('к', ', корп. ')
    return ','.join(l)
```

In [23]:
```
# применим функцию к столбцу
df['address'] = df['address'].apply(chang)
```

In [24]:
```
# проверим количество уникальных значений
len(df['address'].unique())
```

Out[24]: 5597

Проверим данные на дубликаты по наименованию и адресу

In [25]:
```
df[df.duplicated(['name', 'address'])]
```

Out[25]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_avg_bill
1509	Шоколадница	кофейня	Москва, Ленинградское шоссе, 16А, стр. 4	Северный административный округ	ежедневно, 10:00–23:00	55.823029	37.498030	4.1	NaN	NaN	NaN
1511	More Poke	ресторан	Москва, Волоколамское шоссе, 11, стр. 2	Северный административный округ	пн-чт 09:00–18:00; пт,сб 09:00–21:00; вс 09:00...	55.806307	37.497566	4.2	NaN	NaN	NaN
2420	Раковарня Клешни И Хвосты	бар,паб	Москва, проспект Мира, 118	Северо-Восточный административный округ	пн-чт 12:00–00:00; пт,сб 12:00–01:00; вс 12:00...	55.810677	37.638379	4.4	NaN	NaN	NaN
3109	Хлеб Да Выпечка	кафе	Москва, Ярцевская улица, 19	Западный административный округ	NaN	55.738449	37.410937	4.1	NaN	NaN	NaN

In [26]:
```
# удалим дубликаты
df.drop_duplicates(subset = ['name', 'address'], inplace = True)
```

Добавление параметров

- Добавим столбец `street` с названием улицы

In [27]:
```
# напишем функцию для нахождения подстрок в строке
def get_street(row):
    l = [str(i).strip() for i in row.split(',')]
    try:
        for elem in l:
            for elem_2 in ['парк','шоссе','проезд','бульвар','проспект','бул','площадь','километр','линия','пер
                           'просек',
                           'набережная','тупик','заказник', 'улица','сквер', 'аллея','территория','заповедник',
                           'музей','сад','квартал','микрорайон', 'тоннель', 'памятник', 'ул.', 'пр-т', 'жилой к
                if elem_2 in elem:
                    return elem
    except:
        return 'ошибка'
```

In [28]:
```
# применим функцию и добавим новый столбец в датафрейм
df['street'] = df['address'].apply(get_street)
```

In [29]:
```
# для адресов с ошибочными или неполными данными заполним пропуски строкой 'не указано'
df['street'] = df['street'].fillna('не указано')
```

In [30]:
```
# проверим какие строки заполнились строкой 'не указано'
df.loc[df['street'] == 'не указано', 'address'].unique()
```

Out[30]:
```
array(['Москва, Северный административный округ, Головинский район',
       'Москва, Северо-Западный административный округ, район Северное Тушино',
       'Москва, Северо-Восточный административный округ, район Отрадное',
       'Москва, Северо-Восточный административный округ, Останкинский район, Выставка достижений народного хозя
йства, Кольцевая дорога',
       'Москва, Северо-Восточный административный округ, район Ростокино',
       'Москва, Северо-Западный административный округ, район Строгино',
       'Москва, Третье транспортное кольцо',
       'Москва, Западный административный округ, район Крылатское',
       'Москва, Центральный административный округ, район Якиманка',
       'Москва, Восточный административный округ, район Измайлово',
       'Москва, Западный административный округ, район Проспект Вернадского',
       'Москва, № 7',
       'Москва, Юго-Западный административный округ, Академический район',
       'Москва, Юго-Восточный административный округ, район Капотня',
       'Москва, Сумская, 2/12',
       'Москва, Юго-Восточный административный округ, район Кузьминки',
       'Москва, Юго-Восточный административный округ, Нижегородский район'],
      dtype=object)
```

Видим, что для таких строк адрес не указан или указан ошбочно, оставим отметку 'не указано' для столбца `street`

- Добавим столбец `is_24/7` , с логическим значеним True/False для обозначения еждневной и круглосуточной работы кафе

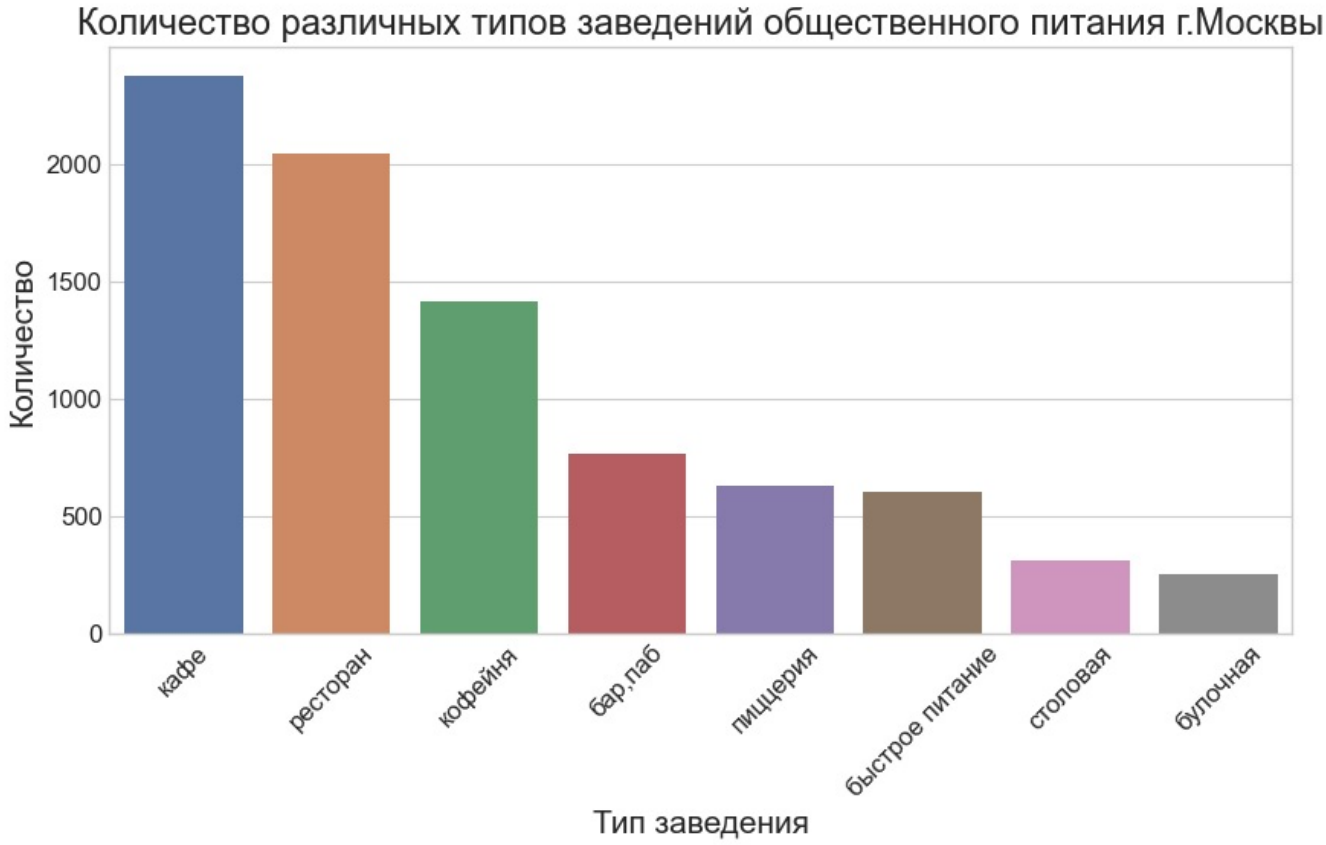
In [31]:
```
df['is_24/7'] = df[df['hours'].isna() == False]['hours'].apply(lambda x: 1 if x == 'ежедневно, круглосуточно' e
```

Анализ данных

Категориальные значения столбца `category`

- Рассмотрим распредение категориальных значений столбца `category`

In [32]:
```
df_0 = df.pivot_table(index = 'category', values = 'name', aggfunc = 'count').reset_index().sort_values(by = 'na
                                                                                                        ascendin
# построим столбчатую диаграмму столбца category
plt.figure(figsize = (10,5))
plt.style.use('seaborn-whitegrid')
sns.set_palette('deep')
sns.barplot( x = 'category',y = 'name', data =df_0 )
plt.title('Количество различных типов заведений общественного питания г.Москвы',size = 17)
plt.xlabel('Тип заведения', size = 15)
plt.ylabel('Количество',size = 15)
plt.xticks(rotation = 45,size = 12)
plt.yticks(size = 12)
plt.show;
```



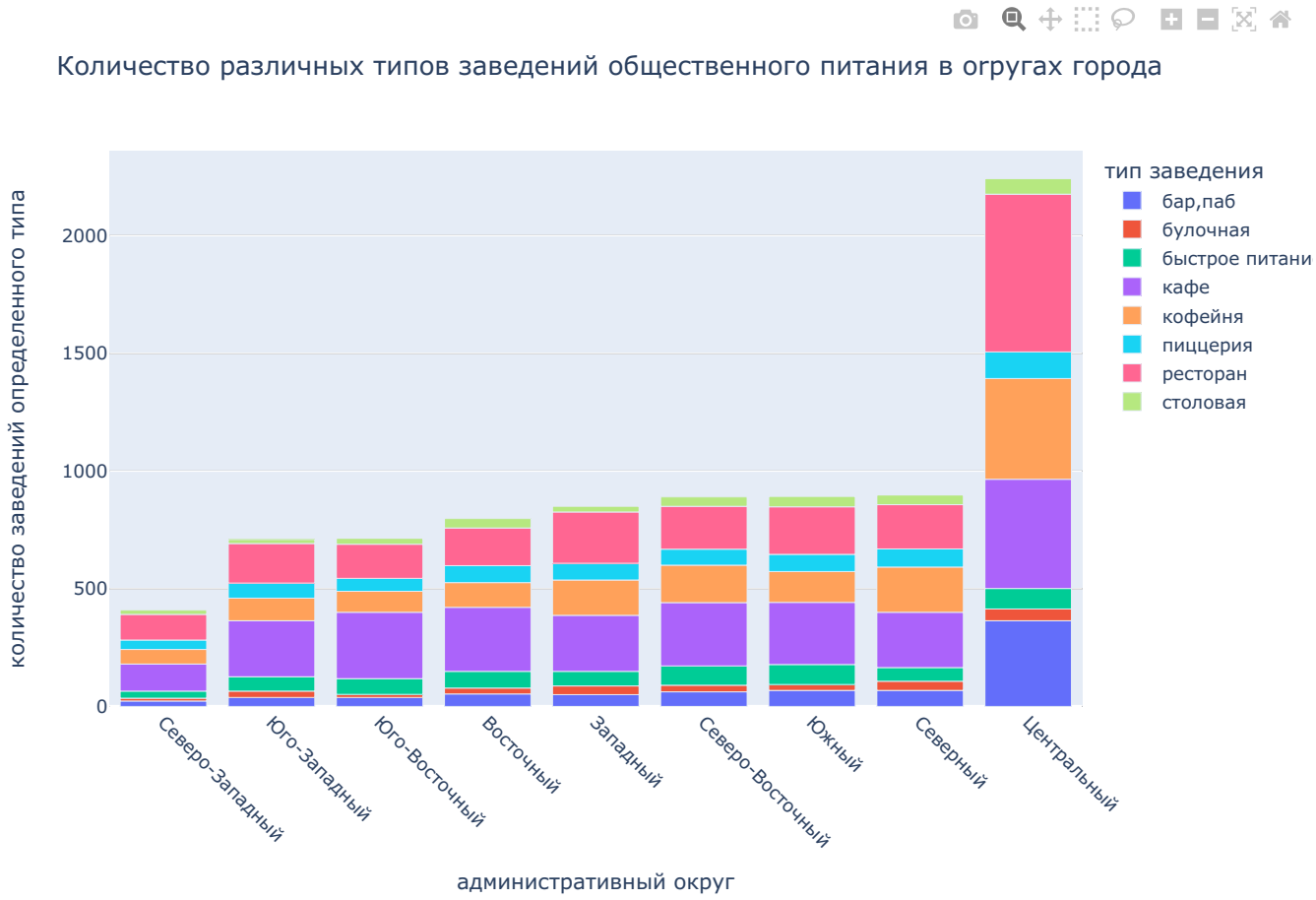
Количественное преимущество у заведений типа "кафе" и "ресторан", меньше всего нишевых заведений по типу "булочная" и "столовая". Посмотрим на распределение типов заведений в разрезе административных округов.

In [33]:
```
# создадим сводную таблицу, сгруппировав данные по округу посчитаем количество заведений каждого типа
df2 = df.pivot_table(index = 'district',columns = 'category', values = 'name', aggfunc = 'count')

display(df2)
```

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
Восточный административный округ	53	25	71	272	105	72	160	40
Западный административный округ	50	37	62	238	150	71	218	24
Северный административный округ	68	39	58	235	192	77	188	41
Северо-Восточный административный округ	62	28	82	269	159	68	182	40
Северо-Западный административный округ	23	12	30	115	62	40	109	18
Центральный административный округ	364	50	87	464	428	113	670	66
Юго-Восточный административный округ	38	13	67	282	89	55	145	25
Юго-Западный административный округ	38	27	61	238	96	64	168	17
Южный административный округ	68	25	85	264	131	73	202	44

Визуализируем данные, построим линейный график где по оси X будут административные округи, а по У количество заведений определенного типа, а также общее количество всех заведений

In [34]:
```
# для удобства сократим подписи оси X, создадим список на замену
labels =['Восточный','Западный','Северный','Северо-Восточный','Северо-Западный','Центральный','Юго-Восточный','Ю
         'Южный']
# построим график визуализации
fig = px.bar(df2, title = 'Количество различных типов заведений общественного питания в округах города',width=9(
fig.update_xaxes(title_text = 'административный округ',ticktext = labels,tickvals = df2.index, tickangle = 45 )
fig.update_xaxes(categoryorder = 'total ascending')
fig.update_yaxes(title_text = 'количество заведений определенного типа')
fig.update_layout(legend_title_text='тип заведения')
fig.show()
```



- ВЫВОД:

Очевидно лидирующим по количеству заведений общественного питания в городе является Центральный административный округ.Интересной особенность - это превышение, причем ощутимое, количества ресторанов над количеством кафе, вопреки общей тенденции. А также большее количество баров-пабов и типа "кофейня". В общей картине города общественные заведения типа "кафе" преобладают по количественному признаку, далее с небольшим количественным отстованием находятся рестораны. Меньше всего булочных и столовых.

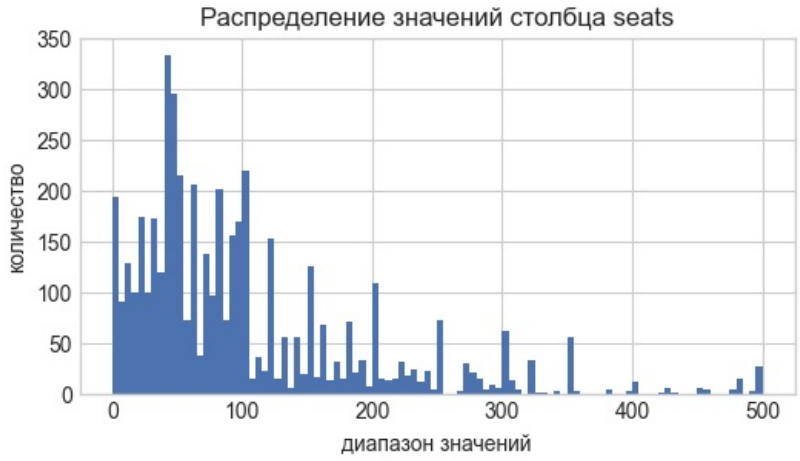
In [35]:
```
#КОД РЕВЬЮЕРА

data = df.groupby(['district', 'category']).agg({'name': 'count'}).reset_index()
districts = {'Северный административный округ': 'САО',
             'Северо-Восточный административный округ': 'СВАО',
             'Западный административный округ': 'ЗАО',
             'Центральный административный округ': 'ЦАО',
             'Восточный административный округ': 'ВАО',
             'Юго-Восточный административный округ': 'ЮВАО',
             'Южный административный округ': 'ЮАО',
             'Юго-Западный административный округ': 'ЮЗАО',
             'Северо-Западный административный округ': 'СЗАО'}

data['district_short'] = data['district'].map(districts)


fig = px.bar(data, y='district_short', x='name', color='category', title = 'Деление заведений общественного пита

fig.update_yaxes(title_text='Район', categoryorder='total ascending')
fig.update_xaxes(title_text='Количество заведений ')
```

```
fig.update_layout(
                legend_orientation='h', legend_yanchor='bottom', legend_y=1.01,
                            legend_xanchor='left', legend_x=.001,
                            legend_title=None
                )
fig.update_traces(hovertemplate='В <i>%{y}<i> %{x:.0f} заведений(-я, -е) <br> данного типа')

fig.show()
```

Посадочные места столбец `seats`

Рассмотрим значения столбца `seats`, построим гистограмму

In [36]:
```
df['seats'].hist(bins =100, figsize = (6,3), range = (0,500))

plt.xlabel('диапазон значений')
plt.ylabel('количество')
plt.title('Распределение значений столбца seats');
```



На графике видно, что около 180 заведений имеют значение посадочных мест равное 0, скорее всего это ошибочное заполнение, означающее отсутствие информации,либо отсутствие посадочных мест, исправим данные. Также стоит отметить довольно длинный хвост гистограммы, предположительно данные неоднородны.

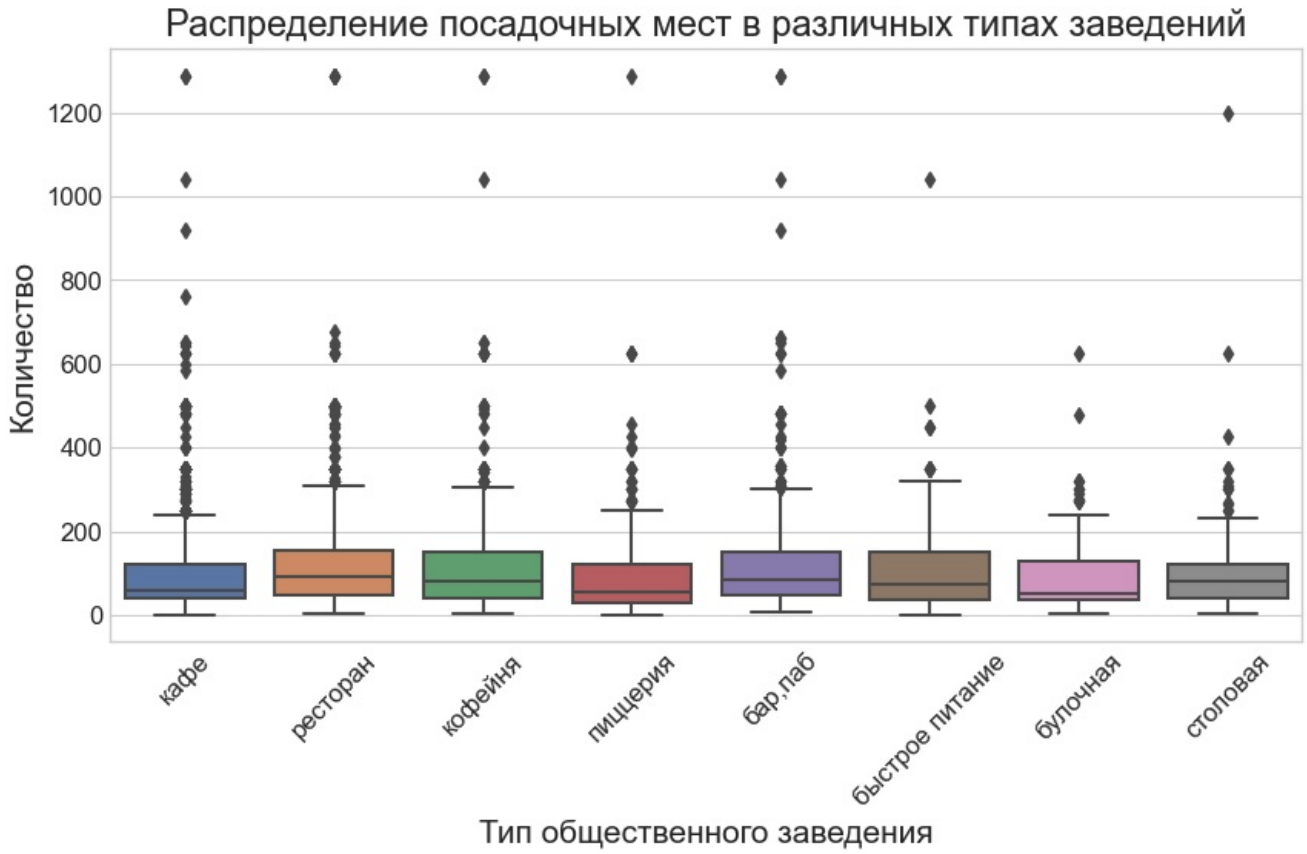
In [37]:
```
df['seats'] = df['seats'].replace({0.0: np.nan})
```

In [38]:
```
# построим сводную таблицу, сгруппировав данные по типу заведения, подсчитаем медианное значение, максимум и мин
```

In [38]:
```
# построим сводную таблицу, сгруппировав данные по типу заведения, подсчитаем медианное значение, максимум и мини
df_seats = df.pivot_table(index = 'category', values = 'seats', aggfunc = ['median','mean','max','min']).\
rename(columns = {'median':'медианное значение','mean':'среднее значение','max':'максимальное значение',\
                  'min':'минимальное значение'}).sort_values(by = ('медианное значение','seats'), ascending = Fals
df_seats.columns = df_seats.columns.droplevel([1])
df_seats['среднее значение'] = round(df_seats['среднее значение'],1)
display(df_seats)
```

category	медианное значение	среднее значение	максимальное значение	минимальное значение
ресторан	90.0	123.8	1288.0	2.0
бар,паб	84.0	125.6	1288.0	6.0
кофейня	80.0	114.9	1288.0	2.0
столовая	80.0	102.9	1200.0	4.0
быстрое питание	75.0	104.3	1040.0	1.0
кафе	60.0	101.0	1288.0	1.0
пиццерия	56.0	96.8	1288.0	1.0
булочная	52.0	96.6	625.0	3.0

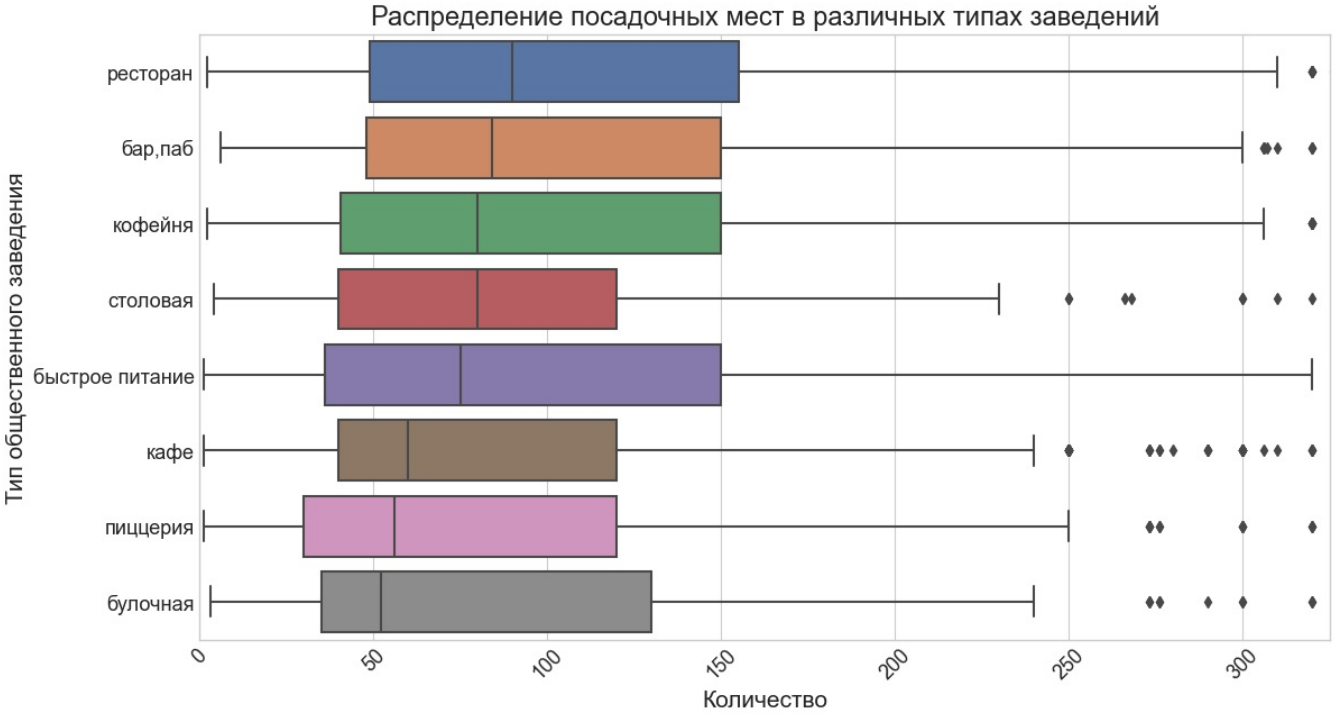
In [39]:
```
# сначала посторим график распределения посадочных мест в разных типах заведений
plt.style.use('seaborn-whitegrid')
sns.set_palette('deep')
plt.figure(figsize = (10,5))
sns.boxplot(data = df.loc[:,['category', 'seats']], x = 'category', y = 'seats')
plt.title('Распределение посадочных мест в различных типах заведений',size = 17)
plt.xlabel('Тип общественного заведения', size = 15)
plt.ylabel('Количество',size = 15)
plt.xticks(rotation = 45,size = 12)
plt.yticks(size = 12)
plt.show;
```



Выбросы сильно влияют на оценку, откорректируем график и отсортируем по медианному значению количество в распределении

In [40]:
```
# построим откорректированный и отсортированный график распределения значений
plt.style.use('seaborn-whitegrid')
sns.set_palette('deep')
plt.figure(figsize = (13,7))
sns.boxplot(data = df.loc[:,['category', 'seats']], y = 'category', x = 'seats', order =df_seats.index )
plt.title('Распределение посадочных мест в различных типах заведений',size = 17)
plt.xlabel('Количество', size = 15)
plt.ylabel('Тип общественного заведения',size = 15)
plt.xticks(rotation = 45,size = 13)
plt.yticks(size = 13)
```

```
plt.xlim(0,325)
plt.show;
```



Видим неоднородность данных, для оценки количества посадочных мест будет использоваться медианное значение

In [41]:
```
#посторим график визуализации
fig = px.line(df_seats.iloc[:,:2],
              title = 'Медианное и среднее значения количества посадочных мест в разных категориях заведений', t
fig.update_xaxes(title_text = 'Тип заведения')
fig.update_yaxes(title_text = 'Количество посадочных мест')
fig.update_layout(legend_title_text='')
fig.show()
```

- ВЫВОД

Медианное и среднее значения имеют существенные различия, очевидно на среднее влияют аномальные значения, распределение неоднородно, что подтвержается показателеями минимума и максимума значений количества посадочных мест в разрезе типа заведения (максимум для всех типов, кроме булочных на уровне выше 1000 посадочных мест). Наибольшее количество посадочных мест имеют рестораны, дальше по количественному признаку идут бары-пабы. Наименьшее количество мест имеют булочные, медианное значение чуть выше 50 мест, в то время как рестораны с макимальным медианным

значением имеют около 90 посадочных мест.Заведения быстрого питания и столовые, а также кофейни, имеют достаточно высокое количество посадочных мест, так как большой поток в пики наполняемости заведения, требует и большего количества посадочных мест.

Сетевые и несетевые заведения общественного питания

Отметка о сетевых/несетевых заведения общественного питания содержится в столбце `chain`. Посчитаем количество заведений , которые относятся к указанным категориям и построим график для наглядности.

In [ ]:

In [42]:
```
# создадим сводную таблицу, посчитаем количество сетевых/несетевых заведений
df_chain = df.pivot_table(index = 'chain', values = 'name', aggfunc = 'count').rename(index = {0:'несетевые', 1
# построим столбчатую диаграмму
fig = px.bar(df_chain, x = df_chain.index, y = 'name', text = 'name', title = 'Общее количество сетевых и несет
 заведений общепита', width=600, height=500)
fig.update_xaxes(title_text = 'признак принадлежности к сетевым заведениям',ticktext = ['несетевые', 'сетевые']
fig.update_yaxes(title_text = 'количество заведений')

fig.show()
```

In [43]:
```
# посчитаем количество сетевых заведений общественного питания сгруппировав данные по уникальному наименованию
print('Количество сетей (уникальных наименований) :',df.loc[df['chain'] == 1].groupby('name')['name'].agg('coun
```

```
Количество сетей (уникальных наименований) : 734
```

Данные содержат 734 уникальных наименований, отмеченных как сетевые заведения.В данных могут быть ошибки, посчитаем количество заведений, отмеченных как сетевые, но имеющие всего 1 заведение общественного питания

In [44]:
```
df3 = pd.DataFrame(df.loc[df['chain'] == 1].groupby('name')['name'].agg('count'))
print('Количество заведений, отмеченных как сетевые,но имеющие всего 1 заведение общественного питания: ',\
      int(df3.loc[df3['name'] == 1].count()))
```

```
Количество заведений, отмеченных как сетевые,но имеющие всего 1 заведение общественного питания:  63
```

Посчитаем долю сетевых/несетевых заведений в разрезе категорий заведений

In [45]:
```
# создадим сводную таблицу, посчитаем долю сетевых/несетевых заведений исходя из категорий
df_categ_chain1 = df.pivot_table(index='category', values = 'chain', aggfunc = 'mean').rename(columns = {'chain
sort_values(by = 'сетевые')
df_categ_chain1['несетевые'] = 1-df_categ_chain1['сетевые']
df_categ_chain1
```

Out[45]:

category	сетевые	несетевые
бар,паб	0.219895	0.780105
столовая	0.279365	0.720635
кафе	0.327724	0.672276
ресторан	0.357003	0.642997
быстрое питание	0.384743	0.615257
кофейня	0.509207	0.490793
пиццерия	0.521327	0.478673
булочная	0.613281	0.386719

In [46]:
```
# построим столбчатую диаграмму
fig = px.bar(df_categ_chain1,
             title = 'Доля сетевых/несетевых заведений в общем количестве этих типов заведений')
fig.update_xaxes(title_text = 'количество заведений')
fig.update_yaxes(title_text = 'типы заведений')
fig.update_layout(legend_title_text='')

fig.update_traces(texttemplate='%{value:.0%}') #КОД РЕВЬЮЕРА - добавляем метки данных в %
fig.show()
```

Доля сетевых заведений больше у булочных , примерно одинаковое количество у кофеен и пиццерий, меньше всего доля сетевых заведений у баров-пабов.

In [47]:
```
# создадим сводную таблицу, посчитаем количество сетевых/несетевых заведений исходя из категорий
df_categ_chain = df.pivot_table(index = 'category',columns = 'chain',values = 'name', aggfunc = 'count').rename
                                    {0:'несетевые', 1:'сетевые'}).sort_values(by = 'несетевые')
# построим столбчатую диаграмму
fig = px.bar(df_categ_chain, x = df_chain.index, \
             title = 'Количество сетевых/несетевых заведений в разрезе типов заведений', text = 'value')
fig.update_xaxes(title_text = 'количество заведений')
fig.update_yaxes(title_text = 'типы заведений')
fig.update_layout(legend_title_text='')
fig.show()
```

Сгруппируем данные по названиям заведений и найдем топ-15 популярных сетей в Москве. Под популярностью понимается количество заведений этой сети в регионе, одна сеть может иметь несколько типов заведений. Рассмотрим оба случая,построим подходящие для такой информации визуализации.

In [48]:
```
# создаем нужный датафрейм, где рассмотрим наименование сети и без категоризации заведений
df_top_0 = pd.DataFrame(df.loc[df['chain'] == 1].groupby(['name'])['name'].agg('count')\
                        .sort_values(ascending = False)).rename(columns = {'name':'количество'}).reset_index().hea
display(df_top_0)
```

	name	количество
0	Шоколадница	119
1	Домино'с Пицца	76
2	Додо Пицца	74
3	Яндекс.Лавка	72
4	One Price Coffee	71
5	Cofix	65
6	Prime	50
7	Чайхана	45
8	Хинкальная	44
9	Кофепорт	42
10	Кулинарная Лавка Братьев Караваевых	39
11	Теремок	38
12	Буханка	36
13	Cofefest	32
14	Му-Му	27

In [49]:
```
# строим столбчатую диаграмму для визуализации данных
fig = px.bar(df_top_0, x = 'name',y = 'количество',\
             title = 'ТОП-15 сетевых заведений без указания типа заведения', text = 'количество')
fig.update_xaxes(title_text = 'наименование сетей')
fig.update_yaxes(title_text = 'количество заведений')

fig.show()
```

Безусловный лидер - это "Шоколадница" - 119 объектов.Первую пятерку замыкает "One Price Coffee" с 65 объектами.
Наименьшее количество в ТОПе занимает сеть "My-My"

Теперь найдем самые популярные типы заведений среди ТОП-15 заведений.

In [50]:
```
# создадим список топ-15 самых популярных сетей
top_list = df_top_0['name'].unique()
```

In [51]:
```
# создаем нужный датафрейм
df_top = pd.DataFrame(df.query('chain == 1 and name in @top_list').groupby(['category'])['name'].agg('count')\
                      .sort_values(ascending = False)).rename(columns = {'name':'количество'}).reset_index().hea
display(df_top)
```

	category	количество
0	кофейня	336
1	ресторан	193
2	пиццерия	151
3	кафе	103
4	булочная	29
5	быстрое питание	12
6	бар,паб	4
7	столовая	2

In [52]:
```
# строим столбчатую диаграмму для визуализации данных
fig = px.bar(df_top,x = 'category', y = 'количество',\
             title = 'Популярные типы заведений в ТОП-15 сетевых заведений', text = 'количество')
fig.update_xaxes(title_text = 'тип заведения')
fig.update_yaxes(title_text = 'количество заведений')

fig.show()
```

Самый популярный тип заведения среди ТОП-15 - это кофейня, значительно меньше баров-пабов и столовых, всего 4 и 2 объекта соответственно. Довольно большое количество ресторанов, что может быть связано с включением в категорию ресторанов по доставке еды.

- ВЫВОД:

В данных преобладает количественное превосходство несетевых заведений общественного питания. Представлено 734 уникальных наименований сетей(63 из них иогут быть ошибочно помечены как сети). В разрезе категорий как правило преобладают несетевые заведения, исключения - это булочные и пиццерии, с преобладанием сетевых заведений. Тип "кофейня" имеет примерно одинаковое разделение, сетевых заведений немногим больше.В ТОП -15 самых популярных сетей входят достаточно крупные сети, больше всего в списке представлены кофейни, самая популярная "Шоколадница", 2 пиццерии "Додо Пицца" и "Доминос Пицца". 72 обьекта у ресторана по доставке еды Яндекс.Лавка. Меньше всего в ТОП-15 баров и столовых. Заведения быстрого питания также небольшое количество, всего 12 объектов

Рейтинги заведений

В данных имеется информация о рейтингах-оценках мест общественного питания. Проведем анализ распределения средних рейтингов по категориям заведений. Используем усредненное значение рейтинга для анализа, ввиду небольшого диапазона значений, ожидаемой однородности данных.

In [53]:
```
# создадим сводную таблицу
df_ind = df.pivot_table(index = 'category', values = 'rating',aggfunc = 'mean').reset_index().rename(columns = 
                        'тип заведения', "rating":'средний рейтинг'}).sort_values(by = 'средний рейтинг')
df_ind['средний рейтинг'] = round(df_ind['средний рейтинг'],2)
display(df_ind)
```

	тип заведения	средний рейтинг
2	быстрое питание	4.05
3	кафе	4.12
7	столовая	4.21
1	булочная	4.27
4	кофейня	4.28
6	ресторан	4.29
5	пиццерия	4.30
0	бар,паб	4.39

In [54]:
```
# построим столбчатую диаграмму
fig = px.bar(df_ind, x = 'средний рейтинг', y = 'тип заведения',\
             title = 'Средний рейтинг в разрезе типов заведений', width=600, height=400, text = 'средний рейтин
fig.update_xaxes(title_text = 'средний рейтинг' )
fig.update_yaxes(title_text = 'тип заведений')
```

```
fig.show()
```

Самая высокая оценка заведений у баров-пабов, быстрое питание имеет самые низкие рейтинги.Вообще средний рейтинг достаточно высокий во всех категориях, учитывая тот факт, что человек скорее оставляет отрицательный, а не положительный отзыв.

In [55]:
```
# создадим сводную таблицу,в которой посчитаем средний рейтинг по типу заведения
df_ind2 = df.pivot_table(index = 'district', columns = 'category', values = 'rating',aggfunc = 'mean')\
.apply(lambda x : round(x,3)).sort_values(by = 'бар,паб', ascending = False)

display(df_ind2)
```

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
Центральный административный округ	4.486	4.372	4.234	4.302	4.336	4.412	4.416	4.315
Западный административный округ	4.398	4.265	3.969	4.076	4.195	4.289	4.257	4.113
Северо-Западный административный округ	4.391	4.283	3.950	4.048	4.326	4.338	4.293	4.189
Юго-Западный административный округ	4.350	4.156	4.087	4.037	4.283	4.341	4.226	4.241
Северный административный округ	4.328	4.254	3.976	4.181	4.293	4.292	4.290	4.217
Восточный административный округ	4.315	4.168	4.044	4.098	4.283	4.269	4.187	4.232
Южный административный округ	4.276	4.340	4.101	4.094	4.233	4.259	4.212	4.261
Юго-Восточный административный округ	4.200	4.038	3.925	4.048	4.226	4.185	4.157	4.104
Северо-Восточный административный округ	4.187	4.343	4.033	4.054	4.217	4.256	4.210	4.082

In [56]:
```
# построим график визуализации
fig = px.scatter(df_ind2,width=950, height=600, title = 'Зависимость показателя среднего рейтинга заведения\
 и района нахождения')
fig.update_xaxes(title_text = 'административный округ',\
                 ticktext = ['Центральный','Западный','Северо-Западный','Юго-Западный','Северный','Восточный','Юг
                             'Юго-Восточный','Северо-Восточный'],tickvals = df_ind2.index )
fig.update_yaxes(title_text = 'средний рейтинг заведений')
fig.update_layout(legend_title_text='тип заведений ')
fig.show()
```

Можно заметить, что заведения Центрального округа получают в целом более высокие оценки. В Южном и Северо-Восточном огругах выше оценивают булочные. Быстрое питание в Юго-Восточном округе имеет самые низкие оценки среди всех заведений всех округов, выше всего в округе ценят кофейни. Хорошие оценки ставят барам-пабам охотнее.В Северном округе намного хеже обстоят дела с заведениями быстрого питания, чем любыми другими.В Юго-Западном, Восточном и Южном округах столовые имеют относительно повышенный рейтинг, чем в среднем по городу.

Визуализируем количество заведений, их типы, для объектов, имеющих низкий рейтинг. Низкий рейтинг опредилим в пределах 3.4 баллов.

In [57]:
```
# создадим датафрейм с заведениями имеющими низкие рейтинги, ниже 3.5, посчитаем их количество в разрезе типов
df_low_rat = df.loc[df['rating'] < 3.5].pivot_table(index = 'district', columns = 'category', values = 'name',\
                        aggfunc = 'count').sort_values(by = 'кафе')
df_low_rat
```

Out[57]:

category	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
district								
Центральный административный округ	1.0	NaN	3.0	10.0	4.0	NaN	2.0	NaN
Северо-Западный административный округ	NaN	1.0	3.0	12.0	1.0	NaN	4.0	NaN
Северный административный округ	2.0	1.0	6.0	18.0	6.0	1.0	3.0	NaN
Западный административный округ	1.0	1.0	11.0	23.0	5.0	2.0	9.0	1.0
Юго-Западный административный округ	1.0	2.0	5.0	25.0	1.0	NaN	6.0	NaN
Южный административный округ	3.0	1.0	7.0	25.0	8.0	1.0	13.0	1.0
Северо-Восточный административный округ	4.0	NaN	11.0	31.0	8.0	1.0	8.0	3.0
Восточный административный округ	NaN	1.0	10.0	32.0	4.0	1.0	10.0	4.0
Юго-Восточный административный округ	3.0	1.0	9.0	38.0	8.0	4.0	11.0	2.0

In [58]:
```
# построим график визуализации
fig = px.bar(df_low_rat, title = 'Заведения с низкими рейтингами',)
fig.update_xaxes(title_text = 'административный округ',ticktext = ['Центральный','Северо-Западный','Северный','
'Юго-Западный','Южный','Северо-Восточный','Восточный','Юго-Восточный'],tickvals = df_low_rat.index, tickangle =
fig.update_yaxes(title_text = 'количество')
fig.update_xaxes(categoryorder = 'total ascending')
fig.update_layout(legend_title_text='тип заведений')
fig.show()
```

Меньше всего заведений с плохими рейтингами в Центральном районе, больше всего в Юго-Восточном округе.Тип заведения 'кафе' лидирует по количеству плохих отзывов, с булочными ситуация обратная, таких заведений с низкими рейтингами меньше всего.

- ВЫВОД:

Учитывая тот факт, что в данных количество заведений в Центральном районе почти в 3 раза превышает количество заведений в других огругах города, хорошие рейтинги говорят о высоком уровне оценки потребителем заведений общественного питания в Центральном округе.Вообще по городу самый высокий средний рейтинг у баров-пабов, пиццерии и рестораны имеют почти одинаковый средний рейтинг, столовые, кафе и заведения быстрого питания находятся в конце списка средних рейтингов. Анализируя заведения с самыми низкими рейтингами можно утверждать, что в восточной части города содержится больше таких заведений, чем в других районах города, типы таких заведений обычно 'кафе', 'ресторан' или заведение быстрого питания.

Фоновая картограмма

Построим фоновую картограмму (хороплет) со средним рейтингом заведений каждого района.

In [59]:
```
# создадим датафрейм, сгруппируем данные по округу и посчитаем средний рейтинг
df_rat = df.groupby('district', as_index=False)['rating'].agg('mean')
df_rat['rating'] = round(df_rat['rating'],2)
df_rat
```

Out[59]:

	district	rating
0	Восточный административный округ	4.17
1	Западный административный округ	4.18
2	Северный административный округ	4.24
3	Северо-Восточный административный округ	4.15
4	Северо-Западный административный округ	4.21
5	Центральный административный округ	4.38
6	Юго-Восточный административный округ	4.10
7	Юго-Западный административный округ	4.17
8	Южный административный округ	4.18

In [60]:
```
# загрузим файл с границами районов
try:
    district_geo = 'C:/Users/User/Downloads/admin_level_geomap.geojson_adm'
except:
    district_geo = '/datasets/admin_level_geomap.geojson'
# создаем карту города
m_lat, m_lng = 55.751244, 37.618423
m = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
```

```
# создаем хороплет
k = folium.Choropleth(geo_data = district_geo, data = df_rat,columns = ['district','rating'], key_on = 'feature
          fill_color = 'RdPu', fill_opacity = 0.8, legend_name = 'Средний рейтинг').add_to(m)
# добавим интерактивное отражение названий округов
k.geojson.add_child(folium.features.GeoJsonTooltip(['name'], labels=False))

folium.LayerControl().add_to(m)

m
```

Out[60]: Make this Notebook Trusted to load map: File -> Trust Notebook

Теперь можно визуально оценить распределение среднего рейтинга. Самый высокий ожидаемо в Центральном районе, после идут Северный и Северо-Западный округа, самый низкий рейтинг в Юго-Восточном округе.

Отобразим на карте города все заведения общественного питания, для удобства обьединим их в кластеры, для корректной визуальной оценки

In [61]:
```
# создаем карту города
m_lat, m_lng = 55.751244, 37.618423
m1 = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
marker_cluster = MarkerCluster().add_to(m1)
# пишем функцию, которая принимает строку датафрейма,создаёт маркер в текущей точке и добавляет его в кластер ma
def create_clusters(row):
    folium.Marker([row['lat'], row['lng']],popup=f"{row['name']} {row['rating']} {row['category']}",).add_to(ma

# применяем функцию create_clusters() к каждой строке датафрейма
df.apply(create_clusters, axis=1)

# выводим карту
m1
```

Out[61]: Make this Notebook Trusted to load map: File -> Trust Notebook

Самый большие кластер по количеству, согласно установленному масштабировапнию конечно в Центральном районе.При изменении масштаба картина логично меняется

Топ-15 улиц с наибольшим количеством заведений. Единичные заведения.

Проанализируем распределение типов заведений, их количество, находящихся на 15 улицах с наибольшим количеством заведений общественного питания.

In [62]:
```
# создадим датафрейм, с Топ-15 улиц, с наибольшим количеством заведений
df_top_street = df.pivot_table(index = 'street', values = 'name', aggfunc = 'count')\
.sort_values(by ='name',ascending = False).head(15).reset_index().rename(columns = {'name':'количество'})
df_top_street
```

Out[62]:

	street	количество
0	проспект Мира	183
1	Профсоюзная улица	122
2	проспект Вернадского	108
3	Ленинский проспект	107
4	Ленинградский проспект	95
5	Дмитровское шоссе	88
6	Каширское шоссе	77
7	Варшавское шоссе	76
8	Ленинградское шоссе	69
9	Люблинская улица	60
10	улица Вавилова	55
11	Кутузовский проспект	54
12	улица Миклухо-Маклая	49
13	Пятницкая улица	48
14	Алтуфьевское шоссе	47

In [63]:
```
# строим столбчатую диаграмму для визуализации данных
fig = px.bar(df_top_street, x = 'количество',y = 'street',\
             title = 'ТОП-15 улиц с наибольшим количеством заведений', text = 'количество')
fig.update_xaxes(title_text = 'название улицы')
fig.update_yaxes(title_text = 'количество заведений')

fig.show()
```

Лидирует по количеству заведений проспект Мира, которая является одной из самых протяженных магистралей города, с 183 заведениями, замыкает топ Алтуфьевское шоссе с 47 заведениями общественного питания. Построим график визуализации Топ-15 улиц в разрезе типов заведений.

In [64]:
```
# создадим список улиц с наибольшим количеством заведений
top_street = list(df_top_street['street'])
# создадим датафрейм,с количеством заведений в разрезе типов на улицах с наибольшим количеством заведений
df_15 = df.query('street in @top_street').pivot_table(index = 'street',columns = 'category',values = 'name', ag
df_15['всего заведений'] = df_15.loc[:,:].sum(axis = 1)
df_15 =df_15.sort_values(by = 'всего заведений', ascending = False)

df_15
```

Out[64]:

category street	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая	всего заведений
проспект Мира	11.0	4.0	21.0	53.0	36.0	11.0	45.0	2.0	183.0
Профсоюзная улица	6.0	4.0	15.0	35.0	18.0	15.0	26.0	3.0	122.0
проспект Вернадского	7.0	1.0	12.0	25.0	16.0	12.0	33.0	2.0	108.0
Ленинский проспект	10.0	3.0	2.0	26.0	23.0	5.0	33.0	5.0	107.0
Ленинградский проспект	15.0	4.0	2.0	12.0	25.0	9.0	25.0	3.0	95.0
Дмитровское шоссе	6.0	2.0	10.0	23.0	11.0	8.0	24.0	4.0	88.0
Каширское шоссе	2.0	NaN	10.0	20.0	16.0	5.0	19.0	5.0	77.0
Варшавское шоссе	6.0	NaN	7.0	18.0	14.0	4.0	20.0	7.0	76.0
Ленинградское шоссе	5.0	2.0	5.0	13.0	12.0	3.0	26.0	3.0	69.0
Люблинская улица	5.0	NaN	5.0	26.0	11.0	1.0	10.0	2.0	60.0
улица Вавилова	2.0	2.0	11.0	15.0	10.0	3.0	12.0	NaN	55.0
Кутузовский проспект	2.0	1.0	2.0	14.0	13.0	3.0	16.0	3.0	54.0
улица Миклухо-Маклая	3.0	NaN	4.0	21.0	4.0	2.0	15.0	NaN	49.0
Пятницкая улица	9.0	3.0	2.0	7.0	6.0	3.0	18.0	NaN	48.0
Алтуфьевское шоссе	5.0	1.0	5.0	13.0	9.0	3.0	10.0	1.0	47.0

In [65]:
```
# строим график визулизации
fig = px.bar(df_15.loc[:,:'столовая'],\
             title = 'Распределение количества заведений,их категорий по улицам с наибольшим количеством заведе
             text = 'value',width=980, height=650 )
fig.update_xaxes(title_text = 'название улицы')
fig.update_yaxes(title_text = 'количество заведений')
fig.update_layout(legend_title_text='тип заведений')

fig.show()
```

На улицах, с большим количеством заведений общепита чаще всего расположены кафе, кофейни и рестораны, меньше всего столовых и булочных.

Найдем улицы, на которых находится только один объект общепита.

In [66]:
```
# создадим датафрейм, где по названию улицы сгруппируем данные и посчитаем количество заведений
df_low_street = df.pivot_table(index = 'street',columns = 'category',values = 'name', aggfunc = 'count')
df_low_street['всего заведений'] = df_low_street.loc[:,:].sum(axis = 1)
# оставим только те строки,в которых только 1 заведение на протяжении улицы
df_low_street = df_low_street.loc[(df_low_street['всего заведений'] == 1)].reset_index()
# создадим список таких улиц
street_low = df_low_street['street'].unique()
```

In [67]:
```
# создадим датафрейм, отфильтровав только те данные, которые относятся к единственным заведениям на улице
df_low = df.query('street in @street_low')
```

Отобразим такие заведения на карте города

In [68]:
```
# создаем карту города
m_lat, m_lng = 55.751244, 37.618423
m2 = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
marker_cluster = MarkerCluster().add_to(m2)
# пишем функцию, которая принимает строку датафрейма,создаёт маркер в текущей точке и добавляет его в кластер ma
def create_clusters(row):
    folium.Marker([row['lat'], row['lng']],popup=f"{row['name']} {row['rating']}",).add_to(marker_cluster)

# применяем функцию create_clusters() к каждой строке датафрейма
df_low.apply(create_clusters, axis=1)

# выводим карту
m2
```

Out[68]: Make this Notebook Trusted to load map: File -> Trust Notebook

При приближении карты можно заметить, что единичные заведения расположены на окраинах города, но есть и в центральных районах, возможно это связано также с длиной улицы.Проверим на каких типах улиц находятся такие заведения

In [69]:
```
#напишем функцию которая принимает строку и возвращает тип улицы
def type_sreet(row):
    l_1 = row.split()
    for element in l_1:
        if element == element.lower() and element.isalpha():
            return element
```

In [70]:
```
# создадим новый столбец, куда запишем тип улицы
df_low['type_st'] = df_low['street'].apply(type_sreet)
```

In [71]:
```
# посчитаем количество уникальных значений столбца
df_low['type_st'].value_counts()
```

Out[71]:
```
улица           212
переулок        109
проезд           57
парк             22
километр         11
набережная        7
тупик             7
площадь           7
проспект          5
аллея             5
шоссе             5
квартал           4
линия             4
территория        3
просек            3
бульвар           3
сквер             2
жилой             2
сад               2
ботанический      1
мост              1
лесопарк          1
ландшафтный       1
микрорайон        1
пешеходный        1
тоннель           1
Name: type_st, dtype: int64
```

Как видно, предположение подтвердилось, заведения расположены в местах,которые имеют небольшую протяженность исходя из типа улицы (переулок, тупик, проезд) или имеют ограниченную локацию(парк, площадь, аллея). И конечно встречаются удаленные места без адреса(километр).Посмторим на тип таких заведений

In [72]:
```
# посчитаем количество разных типов единичных по адресу заведений
df_low_t = (pd.DataFrame(df_low['category'].value_counts())).reset_index()
```

```
# построим график визуализации
fig = px.bar(df_low_t, x = 'index', y = 'category',\
             title = 'Количество заведений  в разрезе их типов, для единичных заведений на улице',\
             width=800, height=400, text = 'category')
fig.update_xaxes(title_text = 'тип заведений' )
fig.update_yaxes(title_text = 'количество')

fig.show()
```

Большинство единичных заведений имеют тип "кафе", меньше всего булочных и пиццерий.

- ВЫВОД:

Единичные заведения общепита на улице рпсположены не только на удаленном расстоянии от центра города, но и непосредственно в центре города, что обусловлено наличием исторической застройки и коротких по протяженности улиц, именуемых переулками, проездами, аллеями.Также некоторые единичные заведения расположены в скверах, парках, что логично ввиду ограниченной локации.Типы таких заведений чаще кафе. Улицы с самым большим количеством заведений в большей степени представлены кафе, кофейнями и ресторанами, находящимися на протяженных улица(шоссе, проспекты) ближе к центральной части города.

> **Комментарий ревьюера, ревью 1**
> Да, верно. Находятся не только на окраинах, но и близко в центру, на коротких улицах. Распределение по типам - похоже на общее распределение всех заведений - ничего особенного.

Анализ стоимости заказа

Значения средних чеков заведений хранятся в столбце middle_avg_bill. Эти числа показывают примерную стоимость заказа в рублях, которая чаще всего выражена диапазоном. Посчитаем медиану этого столбца для каждого района.Для исключения влияния выбросов, ошибочных данных используем медианный показатель для оценки стоимости заказа, чтобы рассчитать наиболее приближенную к реалиям стоимость заказа.

In [73]:
```
# создадим датафрейм, сгруппируем данные по округу и посчитаем медианное значение примерной стоимости заказа в
df_bill = df.groupby('district', as_index=False)['middle_avg_bill'].agg('median')

df_bill
```

Out[73]:

	district	middle_avg_bill
0	Восточный административный округ	575.0
1	Западный административный округ	1000.0
2	Северный административный округ	650.0
3	Северо-Восточный административный округ	500.0
4	Северо-Западный административный округ	700.0
5	Центральный административный округ	1000.0
6	Юго-Восточный административный округ	450.0
7	Юго-Западный административный округ	600.0
8	Южный административный округ	500.0

In [74]:
```
# создаем карту города
m3 = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
# создаем хороплет
bill = folium.Choropleth(geo_data = district_geo, data = df_bill,columns = ['district','middle_avg_bill'], \
                 key_on = 'feature.name',fill_color = 'Blues',fill_opacity = 0.8, legend_name = 'медианная сто
.add_to(m3)
# добавим интерактивное отражение названий округов
bill.geojson.add_child(folium.features.GeoJsonTooltip(['name'], labels=False))

folium.LayerControl().add_to(m3)

m3
```

Out[74]: Make this Notebook Trusted to load map: File -> Trust Notebook

В Центральном округе и Западном округе самое высокое медианное значение стоимости заказа равное 1000р. Самое низкое в Юго-Восточном огруге - 450 р.Посмотрим на заведения с плохим рейтингом.

In [75]:
```
# посчитаем медианную стоимость заказов, для заведений с плохим рейтингом и имеющейся информации по стоимости
df_low_rat_bil = df.loc[df['rating'] < 3.5].pivot_table(index = 'district', values = 'middle_avg_bill',\
                        aggfunc = ['median','count']).sort_values(by = ('count','middle_avg_bill'))
df_low_rat_bil
```

Out[75]:

	median	count
	middle_avg_bill	middle_avg_bill
district		
Северный административный округ	500.0	3
Юго-Западный административный округ	325.0	3
Северо-Западный административный округ	2000.0	5
Центральный административный округ	550.0	5
Южный административный округ	485.0	8
Восточный административный округ	335.0	9
Западный административный округ	417.5	10
Юго-Восточный административный округ	700.0	13
Северо-Восточный административный округ	400.0	14

Заведения с низким рейтингом и имеющие данные о стоимости заказа имеют медианное значение чека достаточно сильно отличающийся от медианного по городу, можно с уверенностью сказать, что таких заведений слишком мало и данных для адекватной оценки и сравнения недостаточно.

- ВЫВОД:

Согласно построенной тепловой карте сравниться с Центральными округом по стоимости заказа может только Западный, что соответствует сложившейся исторической и современной действительностью. Южный, Северо-Восточный и Юго-Восточный огруга входят в нижний диапазон по цене - 450-500р, чуть дороже платят в Восточном округе - 575 рублей. Оставшиеся районы

составляют средний костяк с медианной стоимостью 600-700 рублей.Можно условно разделить город на центральную и западную части, где стоимость заказа выше, чем в восточной части города.

Анализ заведений с круглосуточным режимом работы

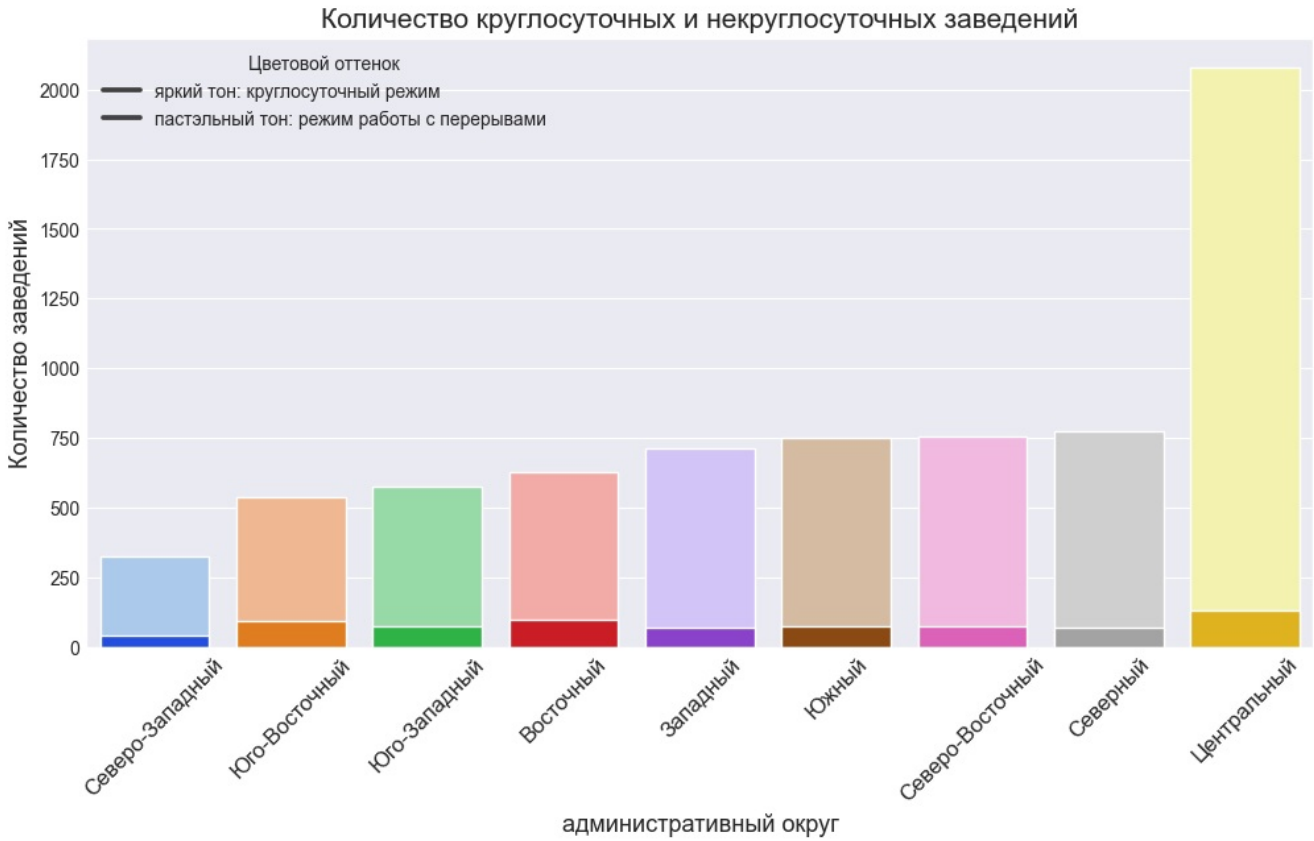
Исследуем часы работы заведений и их зависимость от расположения и категории заведения.

In [76]:
```
# создадим датасет и посчитаем количество заведений в разрезе режима работы
df_h = df.pivot_table(index = 'district',columns = 'is_24/7', values = 'name', aggfunc = 'count').reset_index(
rename(columns = {False:'с перерывом в работе', True: 'круглосуточные'}).sort_values(by = 'с перерывом в работ

df_h
```

Out[76]:

is_24/7	district	с перерывом в работе	круглосуточные
4	Северо-Западный административный округ	326	43
6	Юго-Восточный административный округ	536	93
7	Юго-Западный административный округ	577	73
0	Восточный административный округ	626	97
1	Западный административный округ	713	72
8	Южный административный округ	752	75
3	Северо-Восточный административный округ	754	75
2	Северный административный округ	773	71
5	Центральный административный округ	2080	131

In [77]:
```
spisok = ['Северо-Западный','Юго-Восточный','Юго-Западный','Восточный', 'Западный','Южный','Северо-Восточный',
          'Центральный',]
sns.set_style('darkgrid')
sns.set_palette('pastel')
plt.figure(figsize = (12,6))
sns.barplot(data = df_h, x = 'district', y ='с перерывом в работе' )
sns.set_palette('bright')
sns.barplot(data = df_h, x = 'district', y ='круглосуточные')
plt.title('Количество круглосуточных и некруглосуточных заведений',fontsize = 15)
plt.xlabel('административный округ', fontsize = 13 )
plt.ylabel('Количество заведений', fontsize = 13)
plt.legend(labels=["яркий тон: круглосуточный режим","пастэльный тон: режим работы с перерывами"], title = 'Цв
plt.xticks(ticks = range(0,len(df_h.index)),labels = spisok, rotation = 45, fontsize = 12)

plt.show;
```



Хоть количество круглосуточных заведений в Центральном районе свмое высокой в городе, из-за большого общего количества заведений находящихся в округе, доля таких заведений будет очевидно невелика.Восточный и Юго-Восточный округа при

среднем количестве заведений имеют высокую долю круглосуточных заведений по сравнению с остальными округами.

Посмотрим на соотношение круглосуточных заведений круглосуточных заведний в разрезе типов в каждом округе города

In [78]:
```
# создадим датафрейм, посчитаем количество круглосуточных заведений исходя из типа и округа
df_hours1 = df.pivot_table(index = 'district',columns = 'category', values = 'is_24/7', aggfunc = 'sum')

df_hours1
```

Out[78]:

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
Восточный административный округ	1.0	6.0	21.0	33.0	5.0	8.0	21.0	2.0
Западный административный округ	5.0	0.0	18.0	25.0	9.0	2.0	12.0	1.0
Северный административный округ	4.0	5.0	14.0	28.0	5.0	2.0	11.0	2.0
Северо-Восточный административный округ	4.0	4.0	17.0	31.0	3.0	3.0	11.0	2.0
Северо-Западный административный округ	0.0	3.0	7.0	12.0	2.0	1.0	18.0	0.0
Центральный административный округ	29.0	1.0	14.0	30.0	26.0	2.0	26.0	3.0
Юго-Восточный административный округ	5.0	1.0	15.0	48.0	1.0	9.0	13.0	1.0
Юго-Западный административный округ	0.0	1.0	20.0	34.0	7.0	0.0	10.0	1.0
Южный административный округ	4.0	3.0	24.0	26.0	1.0	4.0	13.0	0.0

In [79]:
```
# построим график визуализации
fig = px.bar(df_hours1, title = 'Распределение круглосуточных заведений в огругах города')
fig.update_xaxes(title_text = 'административный округ',ticktext = ['Восточный','Западный','Северный','Северо-В
'Северо-Западный','Центральный','Юго-Восточный','Юго-Западный','Южный'],tickvals = df_hours1.index, tickangle
fig.update_yaxes(title_text = 'количество')
fig.update_xaxes(categoryorder = 'total ascending')
fig.update_layout(legend_title_text = 'тип заведения', font_size = 12)
fig.show()
```

При ночной прогулке встретить на пути можно скорее круглосуточное кафе, исключение - это Северо-Западный округ, там преобладают круглосуточные рестораны. В Северо-Западном и Юго-Западном не найти круглосуточный ресторан.В Западном округе нет булочных работающих без перерыва, а в Юго-Западном - пиццерий.В Южном округе преобладают кафе и заведения быстрого питания, но совсем нет круглосуточных столовых, Центральный округ имеет практически одинаковое количество круглосуточных баров, кафе, кофеен и ресторанов и всего 1 круглосуточную булочную.

Посчитаем какую долю среди определенного типа заведений составляют круглосуточные заведения такого же типа.

In [80]:
```
# создадим сводную таблицу, сгруппировав данные по округу посчитаем долю круглосуточных заведений каждого типа
# в разных районах города
df_hours = df.pivot_table(index = 'district',columns = 'category', values = 'is_24/7', aggfunc = 'mean')
df_hours = round(df_hours*100,1)
df_hours
```

Out[80]:

category district	бар,паб	булочная	быстрое питание	кафе	кофейня	пиццерия	ресторан	столовая
Восточный административный округ	2.0	26.1	31.8	15.1	4.9	11.3	13.7	5.1
Западный административный округ	10.4	0.0	30.0	13.1	6.1	2.8	5.8	4.2
Северный административный округ	6.2	13.5	25.9	13.6	2.6	2.6	6.2	5.3
Северо-Восточный административный округ	7.1	14.3	21.5	13.5	1.9	4.5	6.3	5.0
Северо-Западный административный округ	0.0	25.0	25.9	14.0	3.2	2.5	17.6	0.0
Центральный административный округ	8.0	2.0	16.5	6.8	6.1	1.8	3.9	4.6
Юго-Восточный административный округ	13.5	8.3	24.6	22.0	1.1	16.4	9.6	4.3
Юго-Западный административный округ	0.0	3.8	35.1	17.3	7.4	0.0	6.3	5.9
Южный административный округ	6.2	12.0	29.6	12.1	0.8	5.5	6.7	0.0

In [81]:
```
# визуализируем данные
plt.figure(figsize = (10,6))
sns.heatmap(df_hours, annot=True, cmap='Blues')
plt.title('Доля круглосуточных заведений среди заведений определенного типа \
в округах города в процентах\n',fontsize = 17)
plt.xlabel('тип заведения', fontsize = 13 )
plt.ylabel('административный округ', fontsize = 13)
plt.xticks(fontsize = 11)
plt.yticks(fontsize = 11,ticks = range(0,len(df_hours.index)),labels = ['Восточный','Западный','Северный','Сев
'Северо-Западный','Центральный','Юго-Восточный','Юго-Западный','Южный'])
plt.show()
```

Процент круглосуточных заведений быстрого питания среди заведений быcрого питания во всех округах один из самых высоких. В Восточном и Северо-Западном округам примерно 25 % булочных - круглосуточные.Меньше всего доля круглосуточных заведений от завдений такого же типа в центре города. Доля круглосуточных кофеен среди кофеен в Западном, Северном и Северо-Восточном округах около 13%.Большая доля круглосуточных ресторанов среди ресторанов в Северо-Западном огруге.Круглосуточных столовых как правило мало, например в Северо-Западном округе их вообще нет, как и пабов-баров.В Юго-Восточном округе и Восточном относительно высокий процент пиццерий среди пиццерий по сравнения с другими округами, в Юго-Западном таких заведений нет вообще.

- Вывод:

Анализируя время работы заведений общественного питания можно придти к выводу, что скорее выделяется Центральный район, там большее количество заведений общественного питания работающих круглосуточно, но процент от общего числа заведений - один из самых низких, отсюда и невысокие показатели доли круглосуточных заведений исходя из типа заведений в этом округе.Меньше всего круглосуточных заведений в Северо-Западном округе. Лидируют по количеству заведений чаще кафе, а вот в разрезе типов заведений как правило в округах больше круглосуточных заведений быстрого питания.

Вывод

Вывод

В городе Москва согласно имеющемуся набору данных пробладают заведения общественного питания по типу "кафе", немного меньшее количество у ресторанов. Меньше всего в городе булочных и столовых, что скорее всего связано с очевидными ограничениями по меню. В разрезе административных округов лидирующее по количеству заведений общественного питания является Центральный административный округ, в котором вопреки общей тенденции преобладают рестораны, а не кафе. Также количество баров-пабов и пиццерий существенно больше, чем в любом другом округе города.

Наибольшее количество посадочных мест имеют заведения, которые предполагают долгое времяпрепровождение - это рестораны, пабы-бары. Булочные и пиццерии,кафе обычно имеют формат небольших заведений, число посадочных мест не больше 60 в медианном значении. Количество таких мест в столовых и заведениях быстрого питания, когда предполагается быстрый поток потребителей, ожидаемо выше.

Общее количество сетевых заведений 3202, несетевых - 5200, причем уникальных сетей, (наименований) - 734(63 из них могут иметь ошибочную метку как сеть, так как представлены единственным заведением). В разрезе категорий преобладают также несетевые заведения, исключения составляют булочные, кофейни и пиццерии, где больше сетей.Меньше всего доля сетей у столовых - такие заведения как правило бывают в единичном экземпляре.

В ТОП-15 самых популярных сетевых заведений ожидаемо входят крупные федеральные игроки, такие как кофейня "Шоколадница", пиццерии "Додо Пицца" и "Доминос Пицца", ресторан "Яндекс.Лавка" и "Теремок". Отличительной особенностью является наличие сетей с развитой системой по доставке еды, а не только традиционное посещение заведения общественного питания.

Рейтинги заведений имеют в среднем достаточно высокие оценки, выше 4 баллов.Рейтинги у пабов-баров самые высокие - 4.39, пиццерии, булочные,кофейни и рестораны имеют отличия в сотые доли процента. Заведения быстрого питания имеют самые низкие рейтинги - 4.05, в разрезе административных округов тенденция прослеживается, только в Южном округе кафе имеют оценку ниже границы рейтинга быстрого питания.В Южном, Юго-Восточном, Северо-Восточном округах высокие оценки ставят булочным и кофейни.Рейтинги заведений Центрального района выше, чем в любом другом районе города. Средний рейтинг всех общественных заведений по округам ожидаемо самый высокий у Центрального округа - 4.38 балла, у Северного - 4.24 балла, у Северо-Западного - 4.21, самый низкий средний рейтинг у заведений Юго-Восточного округа - 4.10 балла.

На больших магистралях-артериях города находятся улицы с наибольшим количеством заведений, самая протяженная из них имеет и большее количество точек общепита это 183 объекта на проспекте Мира.Первую тройку замыкает проспект Вернандского с 108 заведениями.Чаще на таких улицах можно встретить кафе, кофейни и рестораны, реже булочные и столовые. Улицы с единственным заведением находятся не только на отдалении от культурно-развлекательных мест города, но и в части исторической застройки с маленькой физической протяженностью улиц, например на аллеях, переулках, тупиках и проездах.

Медианная стоимость заказа условно делит город на восточную и западную с центральной части. Самая высокая медианная стоимость в Центральном и Западном округах - 1000 рублей. Самая низкая же стоимость в Юго-Восточном округе - 450 рублей.

Наибольшее количество круглосуточных заведений находятся в центре города - 131 обьект, среди них чаще можно встретить кафе, рестораны, бары и кофейни. Меньше всего круглосуточных заведений в Северо-Западном округе - 43 обьекта, там чаще можно найти ресторан, но нет баров и столовых.В остальных районах количество круглосуточных заведений варируется от 71 до 97.

Можно с уверенностью говорить о развитой системе общественного питания в городе.Хотя конкуренция среди заведений достаточно жесткая, особенно в Центральном районе, при наличии конкурентных преимуществ(уникальности в интерьере(рецептуре), талантливый бариста/шеф-повар, ценовая политика, удачное расположение) существует шанс открыть успешный бизнес в эпоху снятия постковидных ограничений и некоторых ослаблений в напряженной борьбе за своего покупателя.

Анализ кофеен города, рекомендации по соисканию местоположения новой кофейни.

Количество кофеен в городе, их расположение

In [82]:
```
#посторим датасет , только с типом заведения - кофейня
df_coffee = df.loc[df['category'] == 'кофейня']
```

In [83]:
```
#посчитаем количество кофеен в городе
print('Количество кофеен в городе: ', df_coffee['name'].count())
```

```
Количество кофеен в городе:  1412
```

In [84]:
```
# создаем карту города
m_lat, m_lng = 55.751244, 37.618423
m4 = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
marker_cluster_1 = MarkerCluster().add_to(m4)
# пишем функцию, которая принимает строку датафрейма,создаёт маркер в текущей точке и добавляет его в кластер
def create_clusters(row):
    folium.Marker([row['lat'], row['lng']],popup=f"{row['name']} {row['rating']}",).add_to(marker_cluster_1)

# применяем функцию create_clusters() к каждой строке датафрейма
df_coffee.apply(create_clusters, axis=1)

# выводим карту
```

```
m4
```

Out[84]: Make this Notebook Trusted to load map: File -> Trust Notebook

На первый взгляд в центре города, а также на севере и западе сосредоточено наибольшее количество заведений, также отчетливо прослеживается тенденция на снижение количества кофеен при удалении от центра города.Построим тепловую карту, посчитаем количество кофеен в округах Москвы.

In [85]:
```
# создадим датафрейм, посчитаем количество кофеен
df_count = df_coffee.pivot_table(index = 'district', values = 'name', aggfunc = 'count').reset_index()
df_count
```

Out[85]:

	district	name
0	Восточный административный округ	105
1	Западный административный округ	150
2	Северный административный округ	192
3	Северо-Восточный административный округ	159
4	Северо-Западный административный округ	62
5	Центральный административный округ	428
6	Юго-Восточный административный округ	89
7	Юго-Западный административный округ	96
8	Южный административный округ	131

In [86]:
```
# создаем карту города
m5 = folium.Map(location = [m_lat, m_lng], zoom_start = 10)
# создаем хороплет
cof_count = folium.Choropleth(geo_data = district_geo, data = df_count,columns = ['district','name'], \
                 key_on = 'feature.name',fill_color = 'YlGn',fill_opacity = 0.8, legend_name = 'Количество коф
# добавим интерактивное отражение названий округов
cof_count.geojson.add_child(folium.features.GeoJsonTooltip(['name'], labels=False))

folium.LayerControl().add_to(m5)

m5
```

Out[86]: Make this Notebook Trusted to load map: File -> Trust Notebook

Большее количество кофеен находится в центральном районе города - 428 объектов, в Восточном, Юго-Восточном, Юго-Западном, Северо -Западном (62 кофейни) округах наименьшее количество заведений

Рейтинги кофеен

Посчитаем средние рейтинги кофеен в различных округах города

In [87]:
```
# создадим датафрейм, посчитаем средний рейтинг
df_c = df_coffee.pivot_table(index = 'district', columns = 'chain', values = 'rating', aggfunc = 'mean').\
rename(columns ={0:'несетевые', 1:'сетевые'}).sort_values(by = 'несетевые')
df_c = round(df_c,2)
df_c
```

Out[87]:

chain	несетевые	сетевые
district		
Западный административный округ	4.24	4.17
Северо-Восточный административный округ	4.28	4.15
Южный административный округ	4.32	4.15
Юго-Восточный административный округ	4.32	4.03
Юго-Западный административный округ	4.32	4.25
Восточный административный округ	4.35	4.21
Северный административный округ	4.38	4.20
Центральный административный округ	4.40	4.28
Северо-Западный административный округ	4.49	4.19

In [88]:
```
fig = px.line(df_c, y = df_chain.index, width=900, height=500, \
              title = 'Средний рейтинг сетевых/несетевых кофеен по административным округам',text = 'value')
fig.update_xaxes(title_text = 'административный округ',ticktext = ['Западный','Северо-Восточный','Южный',\
 'Юго-Восточный','Юго-Западный','Восточный','Северный','Центральный','Северо-Западный'],tickvals = df_c.index)
fig.update_yaxes(title_text = 'средний рейтинг')
fig.update_layout(legend_title_text='')
fig.show()
```

Средний рейтинг несетевых кофеен выше во всех округах города, самый высокий в Северо-Западном регионе, самый низкий в Западном - 4.24 балла.Самый низкий рейтинг сетевых кофеен в Юго-Восточном округе - 4.03 балла, самый высокий в Центральном - 4.28

Режим работы кофеен

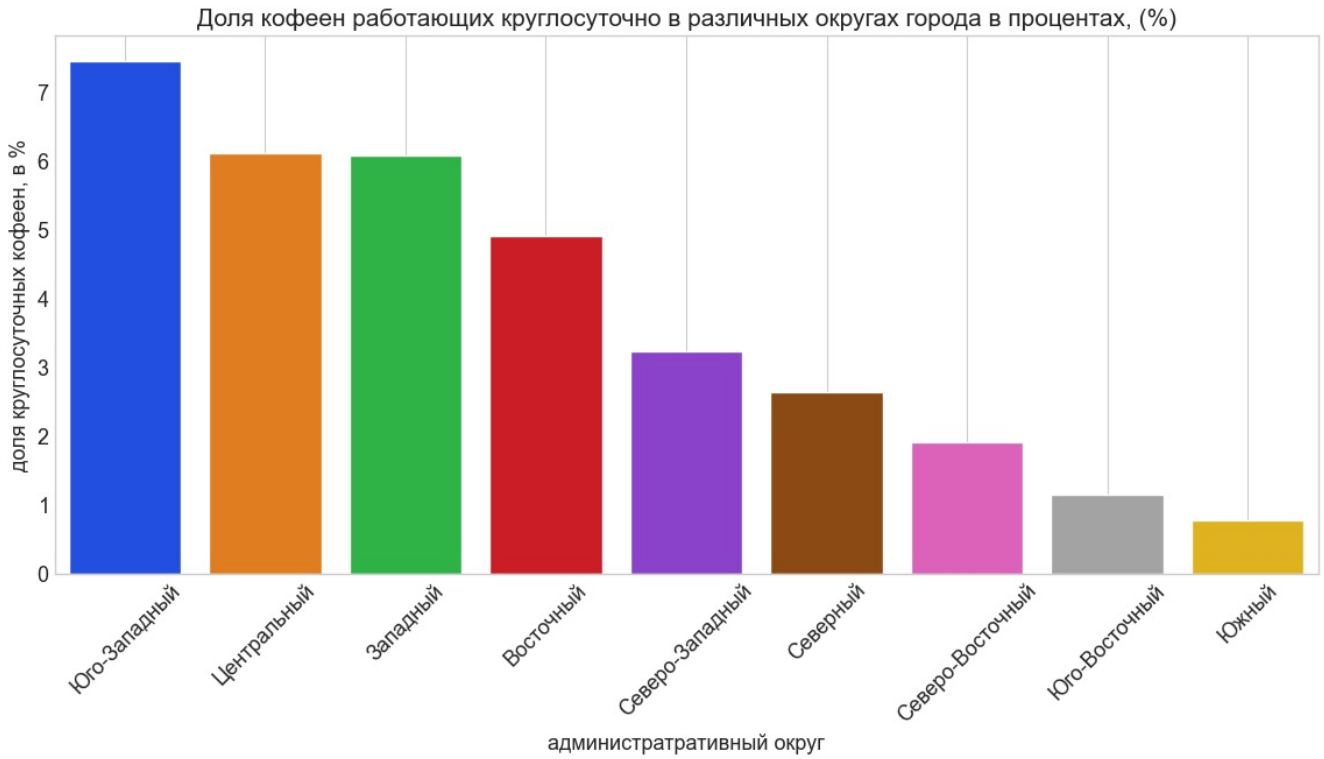
Посмотрим в каких районах есть круглосуточные кофейни, посчитаем процент круглосуточных кофеен от общего количества кофеен, для которых имеется информация о режиме работы

In [89]:
```
# создадим датафрейм, посчитаем процент круглосуточных кофеен
df_cof_hour = df_coffee.pivot_table(index = 'district',values = 'is_24/7', aggfunc = 'mean' )\
.sort_values( by = 'is_24/7', ascending = False)
df_cof_hour = round(df_cof_hour*100,2)
df_cof_hour
```

Out[89]:

district	is_24/7
Юго-Западный административный округ	7.45
Центральный административный округ	6.10
Западный административный округ	6.08
Восточный административный округ	4.90
Северо-Западный административный округ	3.23
Северный административный округ	2.63
Северо-Восточный административный округ	1.91
Юго-Восточный административный округ	1.14
Южный административный округ	0.77

In [90]:
```
# построим график визуализации
labs = ['Юго-Западный','Центральный','Западный','Восточный','Северо-Западный','Северный','Северо-Восточный',\
       'Юго-Восточный','Южный']
sns.set_style('whitegrid')
plt.figure(figsize=(14, 6))
sns.barplot(data = df_cof_hour, x =df_cof_hour.index, y = 'is_24/7' )
plt.title('Доля кофеен работающих круглосуточно в различных округах города в процентах, (%)', fontsize = 15)
plt.xlabel('администратративный округ', fontsize = 13)
plt.ylabel('доля круглосуточных кофеен, в %', fontsize = 13)
plt.xticks(ticks = range(0,len(df_cof_hour.index)),labels = labs,rotation=45, fontsize = 13)
plt.yticks(fontsize = 13)
plt.grid()

plt.show()
```



В Юго-Западном районе процент заведений работающих круглосуточно сымый высокий (выше 7 %), наименьшее количество в Южном и Юго-Восточном округе, около 1% и менее.Вообще количество круглосуточных заведений оносительно общего количества невелико, в округах, насыщенных местами культурно-развлекательного отдыха, с городскими достопримечательностями, кофеен, работающих без перерыва логично больше.

Стоимость чашки кофе

Посчитаем медианную стоимость чашки кофе в различных районах города, рассмотрим количество указанной информации для кофеен и других заведений общепита

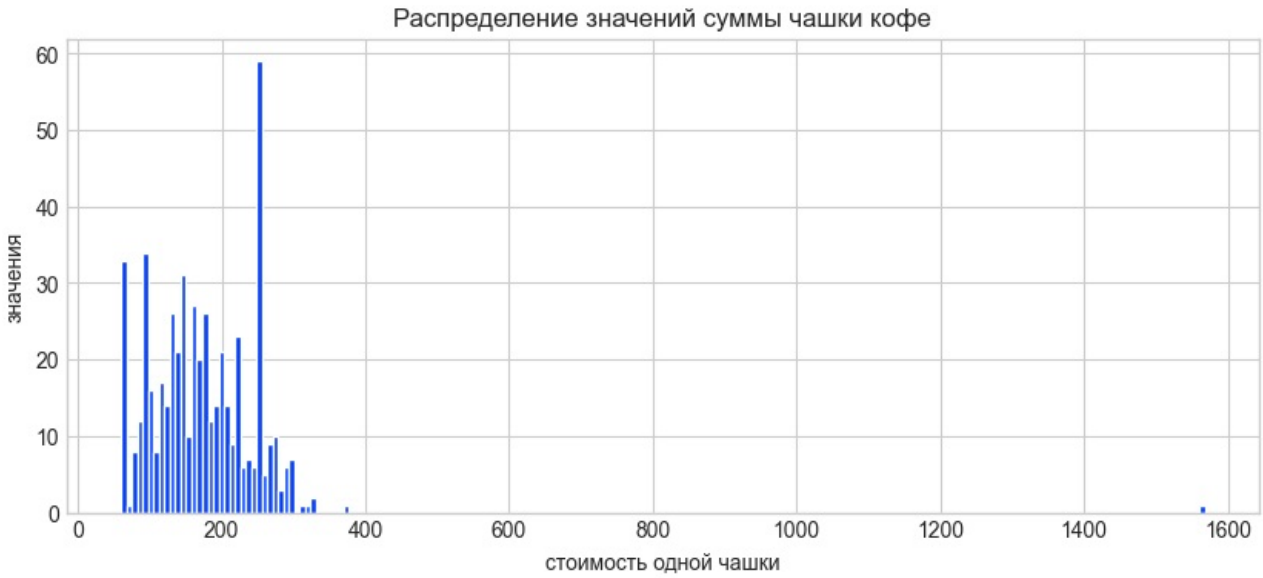
In [91]:
```
print('Количество заведений (кроме кофеен), с указанной информацией о стоимости чашки кофе: ',\
      df.loc[(df['category']!='кофейня')&(df['middle_coffee_cup'].isna()==False)]['name'].count())

print('Количество заведений (только кофейни), с указанной информацией о стоимости чашки кофе: ',\
      df.loc[(df['category']=='кофейня')&(df['middle_coffee_cup'].isna()==False)]['name'].count())
```

```
Количество заведений (кроме кофеен), с указанной информацией о стоимости чашки кофе:  14
Количество заведений (только кофейни), с указанной информацией о стоимости чашки кофе:  521
```

Имеет смысл рассматривать только стоимость чашки кофе в кофейнях

Рассмотрим распределение столбца `middle_coffee_cup`

In [92]:
```
plt.figure(figsize=(10, 4))
df_coffee['middle_coffee_cup'].hist(bins = 200)
plt.title('Распределение значений суммы чашки кофе')
plt.xlabel('стоимость одной чашки')
plt.ylabel('значения');
```



На распределении видим длинный хвост, проверим на ошибочные значения, проверим стоимость чашки кофе больше 600 рублей

In [93]:
```
df_coffee[(df_coffee['middle_coffee_cup']> 600)]
```

Out[93]:

	name	category	address	district	hours	lat	lng	rating	price	avg_bill	middle_av
2859	Шоколадница	кофейня	Москва, Большая Семёновская улица, 27, корп. 1	Восточный административный округ	ежедневно, 08:00–23:00	55.782268	37.709022	4.2	средние	Цена чашки капучино:230–2907 ₽	

Очевидно ошибочное значение, исправим значение

In [94]:
```
df_coffee.loc[(df_coffee['middle_coffee_cup']> 600),'middle_coffee_cup' ] = 260
```

Рассчитаем меданную стоимость чашки кофе. Медиана позволит оценить стоимость чашки кофе, которая встречается чаще всего. По ней, например, можно понять какую сумму на кофе обычно готов потратить потребитель в том или ином округе.

In [95]:
```
# создадим датафрейм , посчитаем медианную стоимость чашки кофе в разных округах города в сетевых/несетевых ко
df_cof_bill = df_coffee.pivot_table(index = 'district', columns = 'chain',values ='middle_coffee_cup', aggfunc
rename(columns ={0:'несетевые', 1:'сетевые'}).sort_values(by = 'несетевые')
df_cof_bill = round(df_cof_bill,2)
df_cof_bill
```

Out[95]:

chain	несетевые	сетевые
district		
Восточный административный округ	137.12	152.25
Юго-Восточный административный округ	151.77	149.83
Северо-Западный административный округ	161.78	168.33
Северо-Восточный административный округ	165.31	165.35
Северный административный округ	173.98	155.12
Южный административный округ	175.91	138.45
Юго-Западный административный округ	187.56	181.17
Западный административный округ	196.60	187.00
Центральный административный округ	198.73	175.73

In [96]:
```
# построим график визуализации
fig = px.line(df_cof_bill, y = df_chain.index, width = 800, height = 500, \
              title = 'Медианная стоимость чашки кофе в кофейнях по административным округам', markers = True)
fig.update_xaxes(title_text = 'административный округ',ticktext = ['Восточный','Юго-Восточный','Северо-Западны
  'Северо-Восточный','Северный','Южный','Юго-Западный','Западный','Центральный'],tickvals = df_cof_bill.index)
fig.update_yaxes(title_text = 'медианная стоимость')
fig.update_layout(legend_title_text = '', hovermode = "y unified", font_size = 12)

fig.show()
```

In [ ]:

Видна интересная особенность - в Юго-Восточном, Северо-Восточном округах медианная стоимость чашки кофе у сетевых и несетевых кофеен практически одинакова. Самая большая разница стоимости, в Южном округе. Напротив небольшая разница в Западном, Юго-Западном и Северо-Западном округах. Стоимость чашки кофе может сильно изменяться, на стоимость могут влиять не только заложенные расходы, но уникальность напитка: обжарка, рецептура или особая подача. Поэтому в расчетах стоит использовать медианное значение цены, максимально исключающее наличие выбросов(уникальностей). Стоимость чашки кофе в сетевых заведениях как правило одинаковая и ниже стоимости несетевых, исключение составляют Восточный и Северо-Западные округа

Посчитаем какая ценовая категория наиболее распространена среди кофеен

In [97]:
```
# сгруппируем данные по ценовому признаку, посчитаем количество и медианное значение стоимости чашки кофе
df_coffee.pivot_table(index = 'price', values = 'middle_coffee_cup', aggfunc = ['median', 'count']).dropna()
```

Out[97]:

	median	count
	middle_coffee_cup	middle_coffee_cup
price		
выше среднего	256.0	1
низкие	139.0	57
средние	200.0	211

Количество кофеен в среднем ценовом диапазоне преобладает, причем медианная стоимость чашки кофе в таких заведениях 200 рублей, в категории "низкие" стоимость чашки кофе равна 139 рублям. В категории "высокие" всего 1 заведение.

Вывод:

Так как заказчик в качестве основных описательных предложений использовал фразу "открыть такую же крутую и доступную, как «Central Perk», кофейню в Москве.", попробуем определить целевую аудиторию кофейни и некоторую концепцию. Предполагается небольшая кофейня с посадочными местами, наличие десертов(завтраков и т.д), подающихся на посуде (непластиковая, многоразовая).Посадочный места - барные, столики для одиночек/пар, несколько больших, уютных мест для компаний до 5 человек.Цены в категории низкие-средние. Ожидается большой процент продаж coffee_to_go. Исходя из концепции целевая аудитория - это люди идущие с/на работу, учебу, мамы с детьми. Исходя из описанных вводных следует выбирать места с высокой проходимостью, рядом с учебными заведениями, большими бизнес-центрами, на пути от дома до целевой точки. Локацию желательно выбирать вблизи больших транспортных узлов(метро, остановки общественного трансорта).Наличие поблизости крупных парков,городских достопримечательностей. Желательно, чтобы заведение находилось на первом этаже, количество ступенек - минимальное. Соседство поблизости конкурирующих заведений вполне возможно при отсутствии схожести в концепции.Важно размещение на первой линии улицы.

Отталкиваясь от предполагаемой стоимости чашки кофе, определим подходящие районы. Восточный, Юго-Восточный, Северо-Западный и Северо-Восточный имеют похожую ценовую политику, что немаловажно стоимость чашки кофе в сетевых заведениях в этих округах больше или равна стоимости несетевых. Рассмторим рейтинги кофеен. В Юго-Восточном округе

самый низкий рейтинг у сетевых заведений, а значит возможно страдает качество, что дает приемущество несетевым заведениям.В Северо-Западном высокий рейтинг у несетевых,можно предположить как высокую конкуренцию, так и высокий уровень спроса. При этом именно в Юго-Восточном и Северо-Западном округах небольшое количество кофеен. Для размещения кофейни рекомендуется выбирать Северо-Западный, Юго-Восточный округ, локацией ближе к центральному. Центральный округ также рекомендуется, с оговоркой на тщательное исследование конкуретных заведений поблизости выбранного помещения.

In [ ]: