In [1]:

**import** pandas **as** pd  
**import** numpy **as** np  
**import** plotly.express **as** px  
**import** warnings  
warnings**.**filterwarnings('ignore')

In [2]:

*# Импортируем основной файл с данными по ресторанам (фастфуд) за 2019 год*  
df **=** pd**.**read\_csv(r'C:\Users\chich\Desktop\CL-3\Данные\Datafiniti\_Fast\_Food\_Restaurants\_May19.csv', sep **=** ',')  
*# Создаем дополнительный столбец - "связка" город+провинция т.к. может быть несколько городов с одним названием*  
df['city\_pr'] **=** df['city']**+**'\_'**+**df['province']

In [3]:

*# Импортируем данные с информацией по районам*  
df\_dist **=** pd**.**read\_csv(r'C:\Users\chich\Desktop\CL-3\Данные\district.csv', sep **=** ';')

In [4]:

*# Импортируем дополнительные данные из внешних источников (численность населения, туристы, площадь городов, координаты городов)*  
df\_pl **=** pd**.**read\_csv(r'C:\Users\chich\Desktop\CL-3\Данные\plotnost.csv', sep **=** ';')

In [5]:

*# Проверяем основной файл на наличие дубликатов*  
df**.**duplicated()**.**sum()

Out[5]:

0

In [6]:

*# Проверяем основной файл на наличие null*  
df**.**isnull()**.**sum()

Out[6]:

id 0  
dateAdded 0  
dateUpdated 0  
address 0  
categories 0  
primaryCategories 0  
city 0  
country 0  
keys 0  
latitude 0  
longitude 0  
name 0  
postalCode 0  
province 0  
sourceURLs 0  
websites 82  
city\_pr 0  
dtype: int64

In [7]:

*# Оставляем в основном датафрейме только нужные нам поля*  
df\_col **=** df[['id','address','categories','city','province','city\_pr','latitude','longitude','name','postalCode']]

In [8]:

*# Формируем список рассматриваемых городов*  
city **=** ['Birmingham\_AL', 'Cincinnati\_OH', 'Cleveland\_OH', 'Columbus\_OH', 'Indianapolis\_IN', 'Portland\_OR', 'Miami\_FL', 'Las Vegas\_NV']  
*# Фильтруем наш датафрейм по списку рассматриваемых городов*  
df\_city **=** df\_col[df\_col['city\_pr']**.**isin(city)]

In [9]:

df\_city**.**info() *# вместе с Las Vegas и Miami*

<class 'pandas.core.frame.DataFrame'>  
Index: 514 entries, 49 to 9992  
Data columns (total 10 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 id 514 non-null object   
 1 address 514 non-null object   
 2 categories 514 non-null object   
 3 city 514 non-null object   
 4 province 514 non-null object   
 5 city\_pr 514 non-null object   
 6 latitude 514 non-null float64  
 7 longitude 514 non-null float64  
 8 name 514 non-null object   
 9 postalCode 514 non-null object   
dtypes: float64(2), object(8)  
memory usage: 44.2+ KB

In [10]:

*# Джойним к нашему датафрейму информацию по районам*   
df\_city **=** df\_city**.**merge(df\_dist, how**=**'left', on**=**'id')

In [11]:

*# По Лас Вегасу и Маями ничего не подтянулось, заполняем null значением Unknown*  
df\_city['district'] **=** df\_city['district']**.**fillna('Unknown')  
df\_city['county'] **=** df\_city['county']**.**fillna('Unknown')

In [12]:

*# Формируем сводную таблицу по городам, которая будет основой для визуализации*  
pivot **=** df\_city**.**groupby('city\_pr')**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()

In [13]:

*# Оформляем визуализацию*  
fig **=** px**.**bar(pivot, x**=**'cnt', y**=**'city\_pr', title **=** 'Кол-во ресторанов фастфуд по городам')  
fig**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Город')  
fig**.**show()

Indianapolis и Columbus являются лидерами по количеству точек общепита

In [15]:

*# Формируем сводную таблицу, которая будет основой для визуализации кол-ва точек по районам конкретного города*  
*# Информации много, в связи с чем на одной визуализации данные анализировать сложно, каждый город был отображен отдельно.*  
pivot2 **=** df\_city[df\_city['city\_pr']**==**'Columbus\_OH']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
*# Прописываем условие cnt>=2, чтобы не показывать районы с незначительным кол-вом точек фастфуда*  
pivot2 **=** pivot2[pivot2['cnt']**>=**2]  
*# Формируем визуализацию*  
fig2 **=** px**.**bar(pivot2, x**=**'cnt', y**=**'district', title **=** 'Columbus')  
fig2**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**700,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig2**.**show()

По городу Columbus предлагается брать в расчет ТОП-5 районов

In [17]:

*# Аналогично по городу Birmingham*  
pivot3 **=** df\_city[df\_city['city\_pr']**==**'Birmingham\_AL']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
pivot3 **=** pivot3[pivot3['cnt']**>=**2]  
fig3 **=** px**.**bar(pivot3, x**=**'cnt', y**=**'district', title **=** 'Birmingham')  
fig3**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**350,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig3**.**show()

По городу Birmingham предлагается брать в расчет ТОП-4 районов

In [19]:

*# Аналогично по городу Cincinnati*  
pivot4 **=** df\_city[df\_city['city\_pr']**==**'Cincinnati\_OH']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
pivot4 **=** pivot4[pivot4['cnt']**>=**2]  
fig4 **=** px**.**bar(pivot4, x**=**'cnt', y**=**'district', title **=** 'Cincinnati')  
fig4**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**500,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig4**.**show()

По городу Cincinnati предлагается брать в расчет ТОП-5 районов

In [21]:

*# Аналогично по городу Cleveland*  
pivot5 **=** df\_city[df\_city['city\_pr']**==**'Cleveland\_OH']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
pivot5 **=** pivot5[pivot5['cnt']**>=**2]  
fig5 **=** px**.**bar(pivot5, x**=**'cnt', y**=**'district', title **=** 'Cleveland')  
fig5**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**300,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig5**.**show()

По городу Cleveland предлагается брать в расчет ТОП-5 районов

In [23]:

*# Аналогично по городу Indianapolis*  
pivot6 **=** df\_city[df\_city['city\_pr']**==**'Indianapolis\_IN']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
pivot6 **=** pivot6[pivot6['cnt']**>=**2]  
fig6 **=** px**.**bar(pivot6, x**=**'cnt', y**=**'district', title **=** 'Indianapolis')  
fig6**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**500,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig6**.**show()

По городу Indianapolis предлагается брать в расчет ТОП-5 районов

In [25]:

*# Аналогично по городу Portland*  
pivot7 **=** df\_city[df\_city['city\_pr']**==**'Portland\_OR']**.**groupby('district')\  
**.**agg(cnt**=**('id','count'))**.**sort\_values(by**=**'cnt', ascending**=True**)**.**reset\_index()  
pivot7 **=** pivot7[pivot7['cnt']**>=**2]  
fig7 **=** px**.**bar(pivot7, x**=**'cnt', y**=**'district', title **=** 'Portland')  
fig7**.**update\_layout(  
xaxis\_title**=**'Кол-во точек общепита',  
yaxis\_title**=**'Район',  
autosize**=False**,  
 width**=**1100,  
 height**=**600,  
 margin**=**dict(l**=**30,r**=**30,b**=**30,t**=**30,pad**=**3))  
fig7**.**show()

По городу Portland предлагается брать в расчет ТОП-2 районов

In [27]:

*# Формируем сводную таблицу по городам с расчетом количества точек общепита и кол-ва районов в городе*  
df\_gr **=** df\_city**.**groupby('city\_pr')**.**agg(places**=**('id','count'), districts**=**('district','nunique'))**.**reset\_index()  
*# Из рассмотрения убираем Лас Вегас и Маями т.к. эти города не являются "типичными" для Америки + не прошли по unit экономике*  
df\_gr **=** df\_gr[(df\_gr['city\_pr'] **!=** 'Miami\_FL') **&** (df\_gr['city\_pr'] **!=** 'Las Vegas\_NV')]**.**reset\_index(drop**=True**)

In [28]:

*# Указываем необходимый коэффициент для дальнейших расчетов*  
K**=**0.33  
*# Джойним нашу сводную таблицу с таблицей, включающую данные из внешних источников (см. блок по импорту)*  
df\_cal **=** df\_gr**.**merge(df\_pl, how**=**'left',on**=**'city\_pr') *# DF с расчетом плотности*  
*# Рассчитываем следующие дополнительные поля:*  
df\_cal['population\_t'] **=** df\_cal['population'] **+** df\_cal['turists'] *# Население+туристы*  
df\_cal['ca\_per\_year'] **=** (df\_cal['population\_t']**\***K)**.**round(2) *# Размер ЦА в год*  
df\_cal['plot\_per\_dist'] **=** (df\_cal['places']**/**df\_cal['districts'])**.**round(2) *# Плотность на Район, шт*  
df\_cal['plot\_per\_km2'] **=** (df\_cal['places']**/**df\_cal['sqr, km2'])**.**round(2) *# Плотность на км2, шт*  
df\_cal['pop\_per\_1place'] **=** (df\_cal['population']**/**df\_cal['places'])**.**round(2) *# Население на 1 ресторан, чел.*  
df\_cal['pop\_t\_per\_1place'] **=** (df\_cal['population\_t']**/**df\_cal['places'])**.**round(2) *# Население+туристы на 1 ресторан, чел.*  
df\_cal['ca\_per\_1place'] **=** (df\_cal['ca\_per\_year']**/**df\_cal['places'])**.**round(2) *# ЦА на 1 ресторан, чел.*  
df\_cal *# Получилась следующая таблица*

Out[28]:

|  | **city\_pr** | **places** | **districts** | **population** | **turists** | **sqr, km2** | **lat** | **lon** | **population\_t** | **ca\_per\_year** | **plot\_per\_dist** | **plot\_per\_km2** | **pop\_per\_1place** | **pop\_t\_per\_1place** | **ca\_per\_1place** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Birmingham\_AL | 68 | 32 | 208889 | 3960000 | 393 | 33.543682 | -86.779633 | 4168889 | 1375733.37 | 2.12 | 0.17 | 3071.90 | 61307.19 | 20231.37 |
| **1** | Cincinnati\_OH | 60 | 37 | 304445 | 26100000 | 206 | 39.103119 | -84.512016 | 26404445 | 8713466.85 | 1.62 | 0.29 | 5074.08 | 440074.08 | 145224.45 |
| **2** | Cleveland\_OH | 50 | 30 | 381386 | 29200000 | 214 | 41.505493 | -81.681290 | 29581386 | 9761857.38 | 1.67 | 0.23 | 7627.72 | 591627.72 | 195237.15 |
| **3** | Columbus\_OH | 90 | 43 | 899908 | 51200000 | 578 | 39.983334 | -82.983330 | 52099908 | 17192969.64 | 2.09 | 0.16 | 9998.98 | 578887.87 | 191033.00 |
| **4** | Indianapolis\_IN | 94 | 37 | 875694 | 29200000 | 954 | 39.791000 | -86.148003 | 30075694 | 9924979.02 | 2.54 | 0.10 | 9315.89 | 319954.19 | 105584.88 |
| **5** | Portland\_OR | 52 | 33 | 653378 | 12100000 | 377 | 45.523064 | -122.676483 | 12753378 | 4208614.74 | 1.58 | 0.14 | 12564.96 | 245257.27 | 80934.90 |

In [29]:

*# По геоданным городов строим следующую визуализацию:*  
*# Цвет точек отвечает за кол-во ресторанов общепита в городе, размер точки за площадь города*  
fig\_m1 **=** px**.**scatter\_mapbox(df\_cal, lat**=**'lat', lon**=**'lon', color**=**'places', \  
 size**=**'sqr, km2', title**=**'Площадь города, кол-во точек', hover\_name**=**'city\_pr', mapbox\_style**=**'carto-positron', zoom**=**3)  
fig\_m1**.**update\_layout(autosize**=False**, width**=**1100, height**=**600)  
fig\_m1**.**update\_layout(margin**=**{"r": 0, "t": 30, "l": 0, "b": 0})  
fig\_m1**.**show()

У Cincinnati площадь менее, чем у других городов, однако, мест общепита больше, чем в Portland и Cleveland

In [31]:

*# По геоданным городов строим следующую визуализацию:*  
*# Цвет точек отвечает за кол-во туристов в городе, размер точки за количество жителей города*  
fig\_m2 **=** px**.**scatter\_mapbox(df\_cal, lat**=**'lat', lon**=**'lon', color**=**'turists', \  
 size**=**'population', title**=**'Кол-во жителей и туристов города', hover\_name**=**'city\_pr', mapbox\_style**=**'carto-positron', zoom**=**3)  
fig\_m2**.**update\_layout(autosize**=False**, width**=**1100, height**=**600)  
fig\_m2**.**update\_layout(margin**=**{"r": 0, "t": 30, "l": 0, "b": 0})  
fig\_m2**.**show()

Columbus - второй по площади (из нашего списка. по площади почти в 2 раза меньше Indianapolis), но первый по населению и туристам. Cleveland имеет небольшую площадь и наименьшее кол-во точек, но кол-во туристов на уровне Indianapolis

In [33]:

*# По геоданным городов строим следующую визуализацию:*  
*# Цвет точек отвечает за кол-во целевой аудитории на 1 ресторан по конкретному городу, размер точки за плотность расположения точек на один район города*  
fig\_m3 **=** px**.**scatter\_mapbox(df\_cal, lat**=**'lat', lon**=**'lon', color**=**'ca\_per\_1place', \  
 size**=**'plot\_per\_dist', title**=**'Плотность на район vs ЦА на 1 ресторан', hover\_name**=**'city\_pr', mapbox\_style**=**'carto-positron', zoom**=**3)  
fig\_m3**.**update\_layout(autosize**=False**, width**=**1100, height**=**600)  
fig\_m3**.**update\_layout(margin**=**{"r": 0, "t": 30, "l": 0, "b": 0})  
fig\_m3**.**show()

Исходя из вышеобозначенных наблюдений мы пришли к выводу, что город Columbus приоритетным для открытия блинного ресторана

In [35]:

*# По геоданным точек общепита строим следующую визуализацию:*  
*# Расположение точек общепита в городе Columbus с возможностью слияния в кластеры, а также с отображением количества точек в каждом кластере*  
df\_columbus **=** df\_city[df\_city['city\_pr']**==**'Columbus\_OH']  
fig\_m4 **=** px**.**scatter\_mapbox(df\_columbus, lat**=**'latitude', lon**=**'longitude', title**=**'Columbus, распределение конкурентов (кластеры)', hover\_name**=**'city\_pr',\  
 hover\_data**=**['district', 'address'], mapbox\_style**=**'carto-positron', zoom**=**9.5)  
fig\_m4**.**update\_traces(cluster**=**dict(enabled**=True**))  
fig\_m4**.**update\_layout(autosize**=False**, width**=**1100, height**=**600)  
fig\_m4**.**update\_layout(margin**=**{"r": 0, "t": 30, "l": 0, "b": 0})  
fig\_m4**.**show()

In [36]:

*# По геоданным точек общепита строим следующую визуализацию:*  
*# Расположение точек общепита в городе Columbus без слияния в кластеры*  
fig\_m5 **=** px**.**scatter\_mapbox(df\_columbus, lat**=**'latitude', lon**=**'longitude', title**=**'Columbus, распределение конкурентов', hover\_name**=**'city\_pr',\  
 hover\_data**=**['district', 'address'], mapbox\_style**=**'carto-positron', zoom**=**9.5)  
*#fig\_m5.update\_traces(cluster=dict(enabled=True))*  
fig\_m5**.**update\_layout(autosize**=False**, width**=**1100, height**=**600)  
fig\_m5**.**update\_layout(margin**=**{"r": 0, "t": 30, "l": 0, "b": 0})  
fig\_m5**.**show()