**Лабораторная работа № 7**

**Тема: «Базовая визуализация гео-пространственных данных на языке Python с помощью библиотеки plotly»**

**Цель работы**: изучить базовые способы визуализации гео-пространственных данных в библиотеке plotly.

**Теоретическая справка**

plotly.express.choropleth\_mapbox - хороплетная карта, с возможностью центрировать и ограничивать регионы карты. Имеются следующие параметры:

* data\_frame - принимает датафрейм с данными, содержащими индексы гео-объектов, подписи, значения для окраски регионов
* geojson - принимает словарь (гео-джейсон) с геометрией.
* locations - столбец с индексом полигонов из датафрейма
* featureidkey - принимает столбец с ключом из геоджейсона, отвечающим за индекс полигона
* color - столбец по которому нужно сделать визуализацию. Если его не указать, то все регионы будут одного цвета
* mapbox\_style -принимает строку, стиль карты
* color\_continuous\_scale - шкала изменения цвета
* Можно найти в документации
* range\_color - принимает кортеж из минимума и максимума шкалы изменения цвета
* opacity - число от 0 до 1 - прозрачность раскрашенных областей
* labels- позволяет менять названия лейблов. принимает на вход словарь с соответствием названий столбцов в датафрейме и желаемым названием
* hover\_name - принимает столбец с названиями областей, они показываются при наведении на область

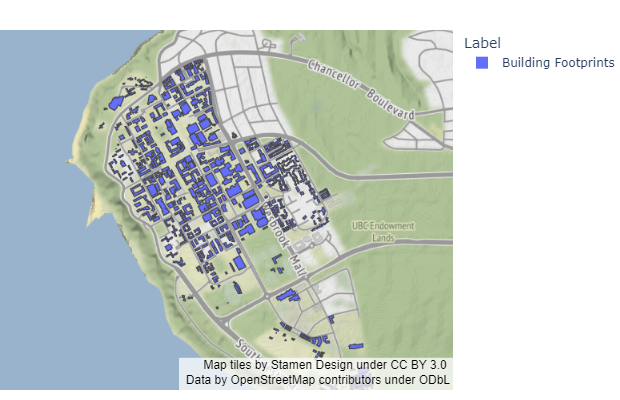
Например, нижеследующий код выведет карту, с подложкой (стилем) и нанесенными заполненными контурами зданий:

fig = px.choropleth\_mapbox(ubc, geojson=ubc.geometry, locations=ubc.index, color="Label",

                           center={"lat": 49.261, "lon": -123.246}, zoom=12.5,

                           mapbox\_style="stamen-terrain")

fig.update\_layout(margin=dict(l=0, r=0, t=30, b=10))



Для работы с геопандас датафреймами (которые позволяют манипулировать гео-данными), требуется его установить (в колабе по умолчанию отсутствует) с помощью команды:

!pip install -q geopandas

И импортировать данные, например в такой форме:

import geopandas as gpd

gpd.read\_file('/content/administrative-divisions-of-moscow/moscow\_adm.shp')

для превращения гео-датафрейма в гео-джейсон, можно использовать команду:

import json

json.loads(data.to\_json(to\_wgs84=True))

Для построения точечной/пузырьковой гео-карты можно использовать:

fig = px.scatter\_geo(df, locations="iso\_alpha",

                     size="pop” )

Для расчета центроида внутри полигона в геопандас, можно использовать команду

GeoSeries.centroid

**Самостоятельное задание**

1. Импортировать данные lab7.xlsx в соответствии с кодом:

|  |
| --- |
| import pandas as pd  df=pd.read\_excel(' lab7\_geo.xlsx) |

1. С помощью plotly.express построить хороплетную карту, показывающую плотность населения внутри административных границ районов Москвы
2. С помощью plotly.express построить хороплетную карту, показывающую количество малых и средних предприятий внутри административных границ районов Москвы
3. С помощью plotly.express построить хороплетную карту, показывающую площадь жилых помещений на 1 жителя внутри административных границ районов Москвы
4. С помощью plotly.express построить хороплетную карту, показывающую процент парков и лесов от площади внутри административных границ районов Москвы
5. Заменить на карте из пп.2 палитру на расходящуюся (синий-красный)
6. Вывести на карте из пп.3 во всплывающей информации название административного округа.
7. Используя гео-пространственные методы geopandas склеить полигоны в административные округа и рассчитать в них показатель – соотношение между плотностью населения и количеством малых предприятий. Построить по полученным данным хороплетную карту.
8. Построить точечную гео-карту (добавить на карту Москвы маркеры, соответствующие зданиям МГУ, МИФИ, МАИ, РАНХИГС). Каждому маркеру добавить подпись.
9. Построить пузырьковую гео-карту (добавить на карту Москвы маркеры, соответствующие центроидам административных районов Москвы). Каждому маркеру добавить подпись. Размер маркера определить из плотности населения.