CINIMEX
Вместе быть лучшими!

Обработка геоданных в Python

Мещеряков Александр

Сентябрь 2024

www.cinimex.ru

Я приехал в Питер, что делать?

Сегодня мы узнаем

Как получить геоданные?

Как правильно рассчитывать расстояния

Как объединять пространственные данные

Как можно производить агрегацию геоданных

CINIMEX
Вместе быть лучшими!

Как получить геоданные?

www.cinimex.ru

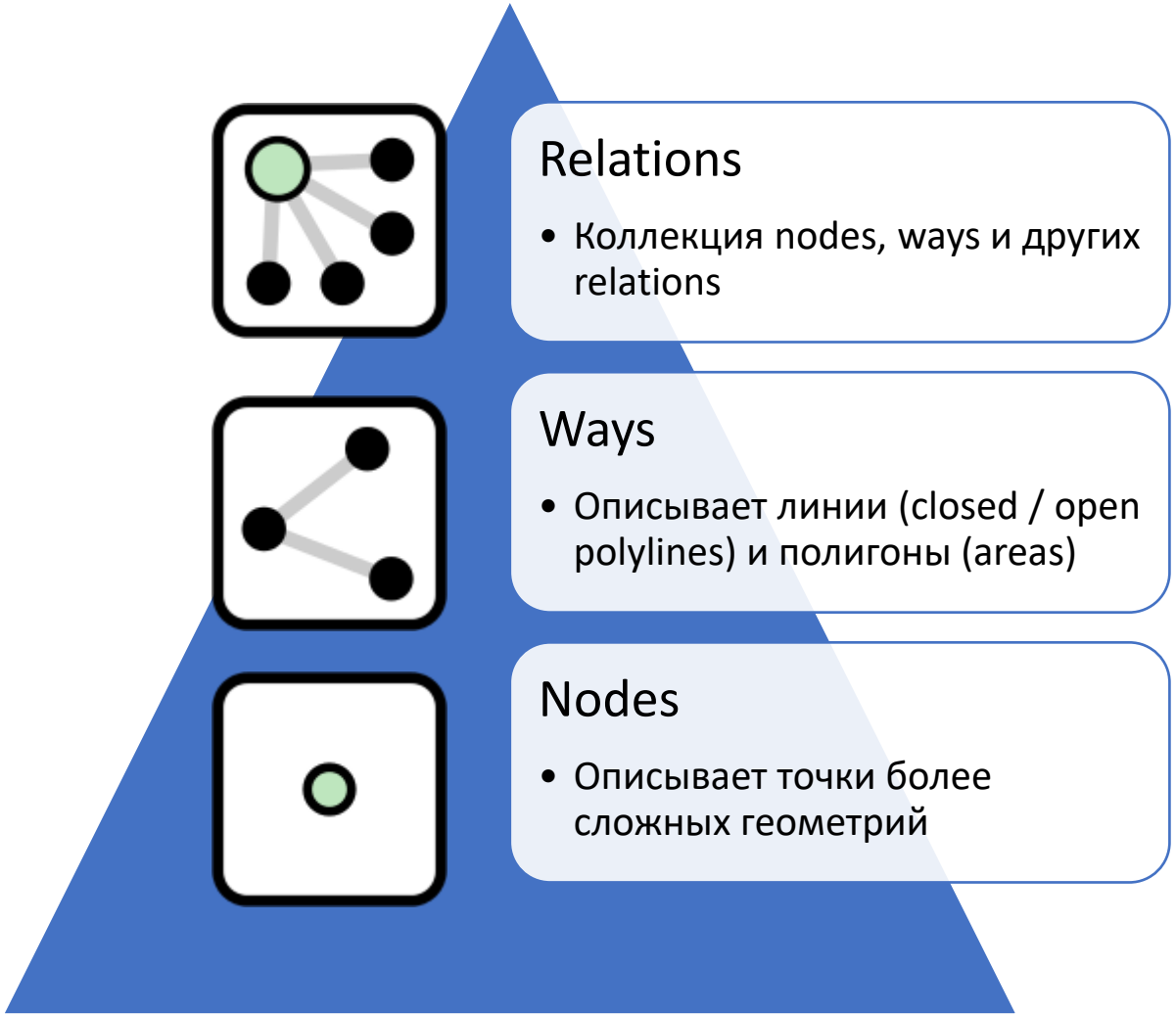
Источники данных

Яндекс Карты

Google Maps

OpenStreetMap

OSM: nodes, ways, relations



Relations

- Коллекция nodes, ways и других relations

Ways

- Описывает линии (closed / open polylines) и полигоны (areas)

Nodes

- Описывает точки более сложных геометрий

OSM тэги

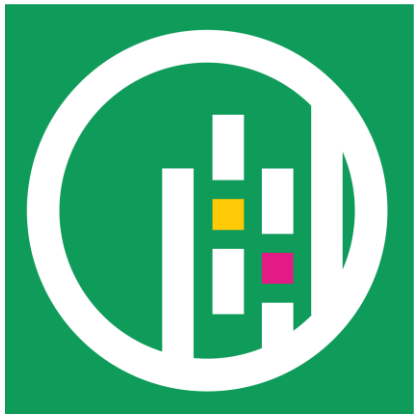
Сегодня нам потребуются:

- **amenity** — точки интереса (cafe, pub, restaurant)
- **building** — контуры зданий
- **admin_level** — уровень административного деления

НЕКОТОРЫЕ ПОПУЛЯРНЫЕ КЛЮЧИ

www.cinimex.ru

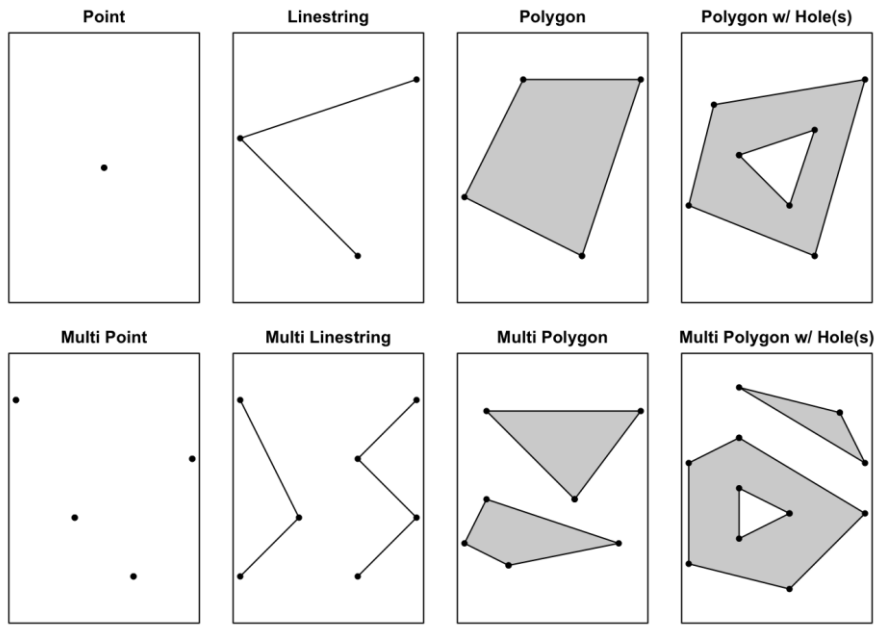
Работа с геоданными на стороне python: GeoPandas и shapely

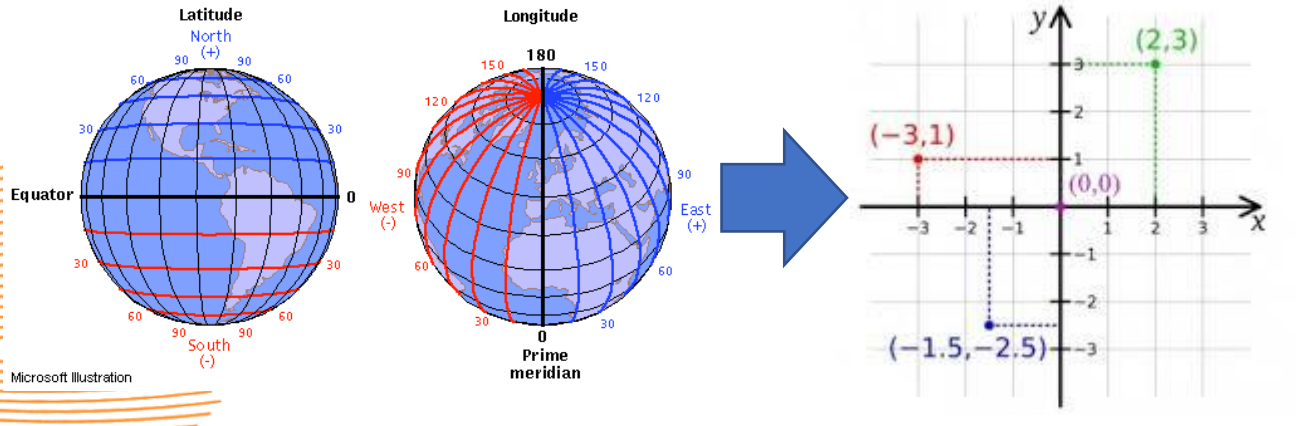


GeoPandas — расширение для более удобного анализа геоданных. Добавляет новые типы данных к *pandas* и позволяет производить пространственные операции.

Для геометрических операций *GeoPandas* использует *shapely*

Представления геометрий в WKT



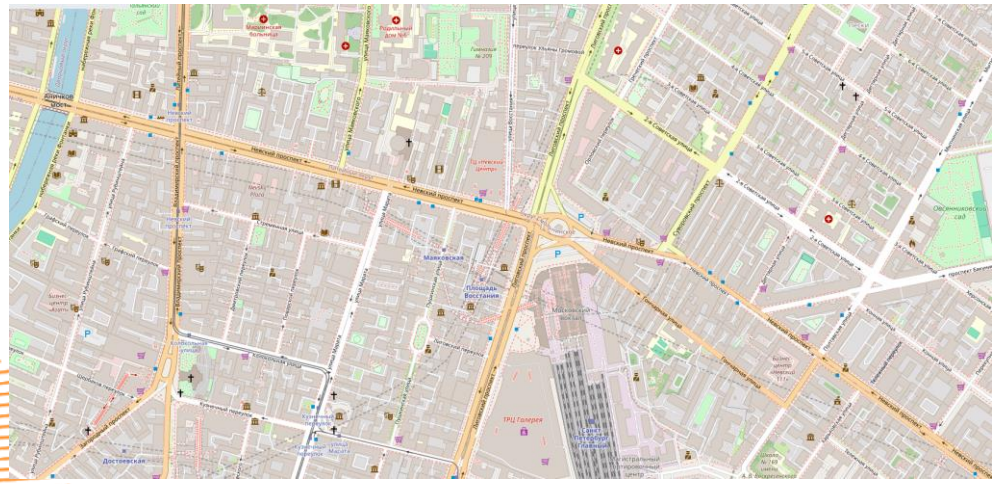


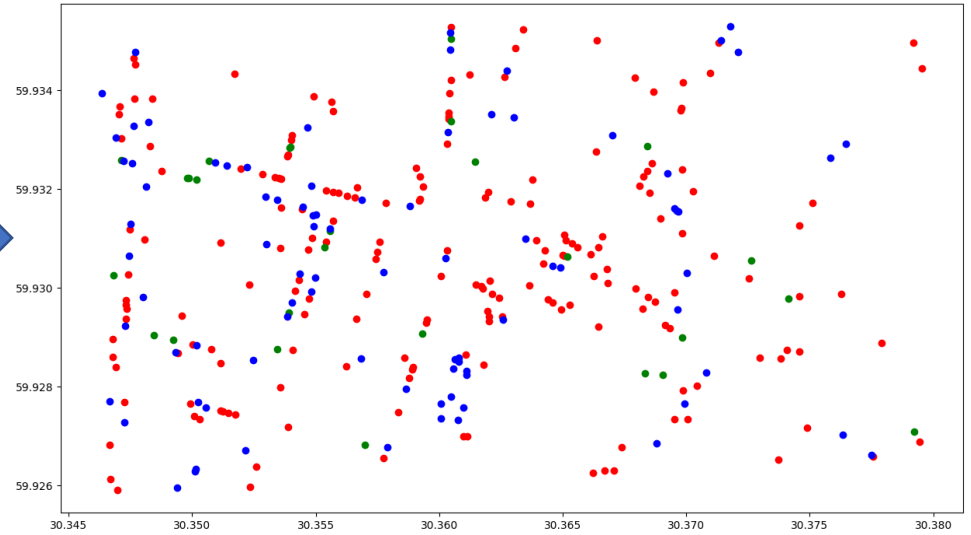
Пример текста в WKT
POINT(1 2)
LINESTRING(1.5 2.45,3.21 4)
POLYGON((1 2,1 4,3 4,3 2,1 2))
POLYGON((0.5 0.5,5 0,5 5,0 5,0.5 0.5), (1.5 1,4 3,4 1,1.5 1))
MULTIPOINT(0 0,1 1)
MULTILINESTRING((0 0,–1 –2,–3 –4),(2 3,3 4,6 7))
MULTIPOLYGON(((0 1,3 0,4 3,0 4,0 1)), ((3 4,6 3,5 5,3 4)), ((0 0,–1 –2,–3 –2,–2 –1,0 0)))
GEOMETRYCOLLECTION(POINT(5 8), LINESTRING(–1 3,1 4))

Загрузка из Overpass API

```
features = osmnx.features_from_bbox(
    bbox=(59.9355, 59.9259, 30.3798, 30.3461),
    tags={
        "amenity":["cafe", "pub", "restaurant"],
        }
)
features.head(3)
```

		amenity	check_date	contact:vk	cuisine	name	opening_hours	payment:jcb	payment:maestro	payment:mastercard	payment:unionpay	...
element_type	osmid											
node	444776758	cafe	2022-12-09	wahaha.tea.coffee	chinese	Wahaha	09:00-22:00	yes	yes	yes	yes	...
	455297491	pub	NaN	https://vk.com/thepub	NaN	Тара Бруч	24/7	NaN	NaN	NaN	NaN	...
	497695708	restaurant	NaN	NaN	japanese	Мисо	NaN	NaN	NaN	NaN	NaN	...





Геокодирование

```
import geopandas as gpd
df_adresses = gpd.GeoDataFrame(
    {
        "addresses": [
            "Санкт-Петербург, ул. Ташкентская, д. 4",
            "Санкт-Петербург, ул. Стартовая, 6, литера А",
            "Санкт-Петербург, Невский проспект, 85",
        ]
    }
)
df_adresses
```

```
df_geocoded = gpd.tools.geocode(
    df_adresses["addresses"],
    provider="nominatim",
    user_agent="piterpy_test",
    timeout=10
)
df_geocoded
```

	addresses
0	Санкт-Петербург, ул. Ташкентская, д. 4
1	Санкт-Петербург, ул. Стартовая, 6, литера А
2	Санкт-Петербург, Невский проспект, 85

	geometry	address
0	POINT (30.31496 59.89188)	Ташкентская улица, Ближняя Рогатка, округ Моск...
1	POINT EMPTY	None
2	POINT (30.36098 59.92974)	Сбербанк, 85 литВ, Невский проспект, округ Лиг...

CINIMEX
Вместе быть лучшими!

Планарные и геодезические расстояния

www.cinimex.ru

Расчет планарных расстояний

```
df_buildings_mercator = df_buildings.to_crs(3857)
...
train_station_centroid = df_buildings_big.query(
    "osmid == 2404499",
).geometry.centroid.values[0]
train_station_centroid.wkt
```

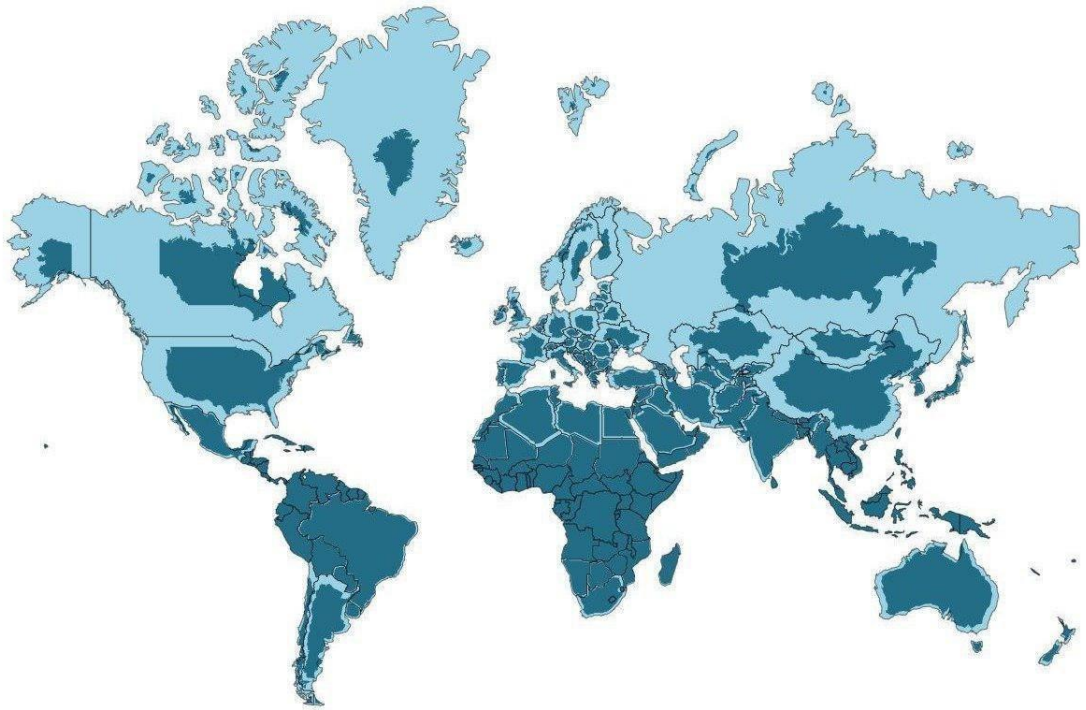
POINT (3379907.3673304603 8384099.920298464)

```
df_mercator_amenities = features_nodes.to_crs(3857)
df_distances = gpd.GeoDataFrame()
df_distances["planar_mercator"] = df_mercator_amenities.distance(
    train_station_centroid
)
```

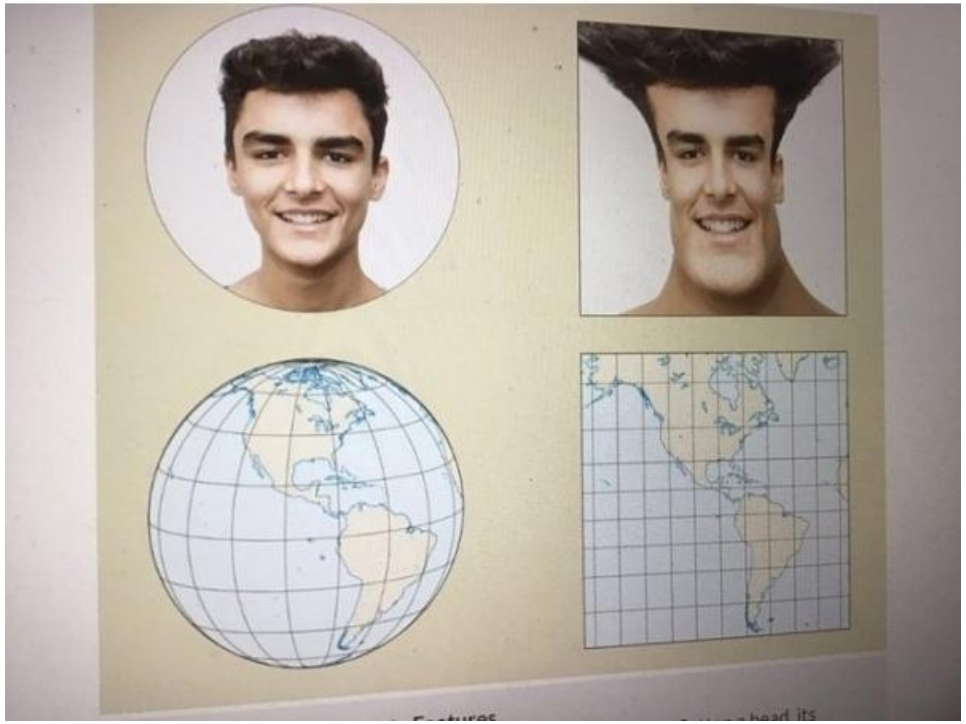
EPSG
GEODETIC PARAMETER DATASET
Managed by IOGP's Geomatics Committee

SRID — Позволяет определить уникальную комбинацию систему координат, допуска, разрешения и единиц измерения. Зная SRID можно произвести переход из одной проекции в другую.

Расчет планарных расстояний

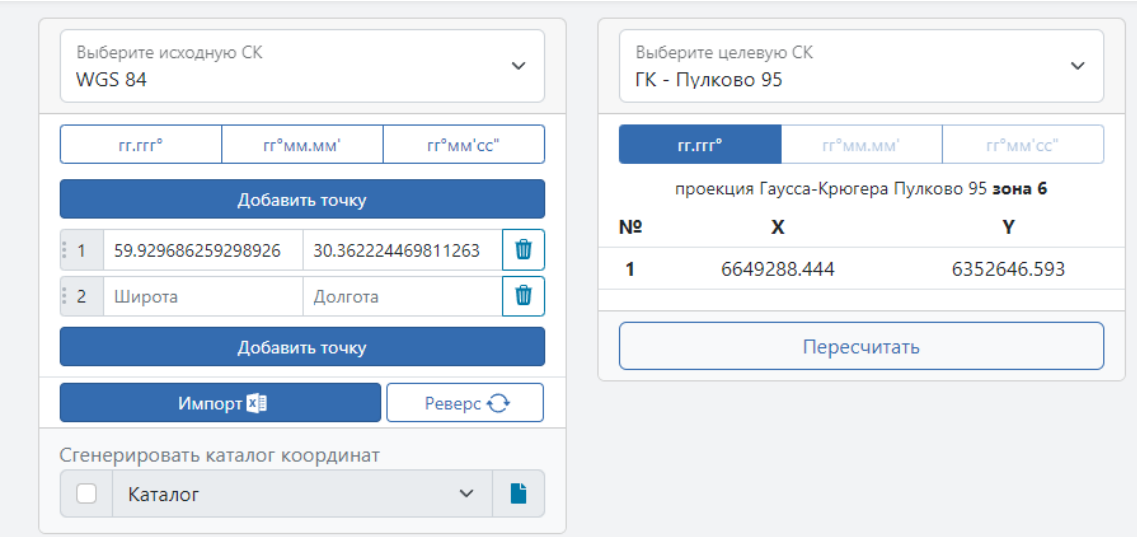


Как просто объяснить, что такое проекция Меркатора



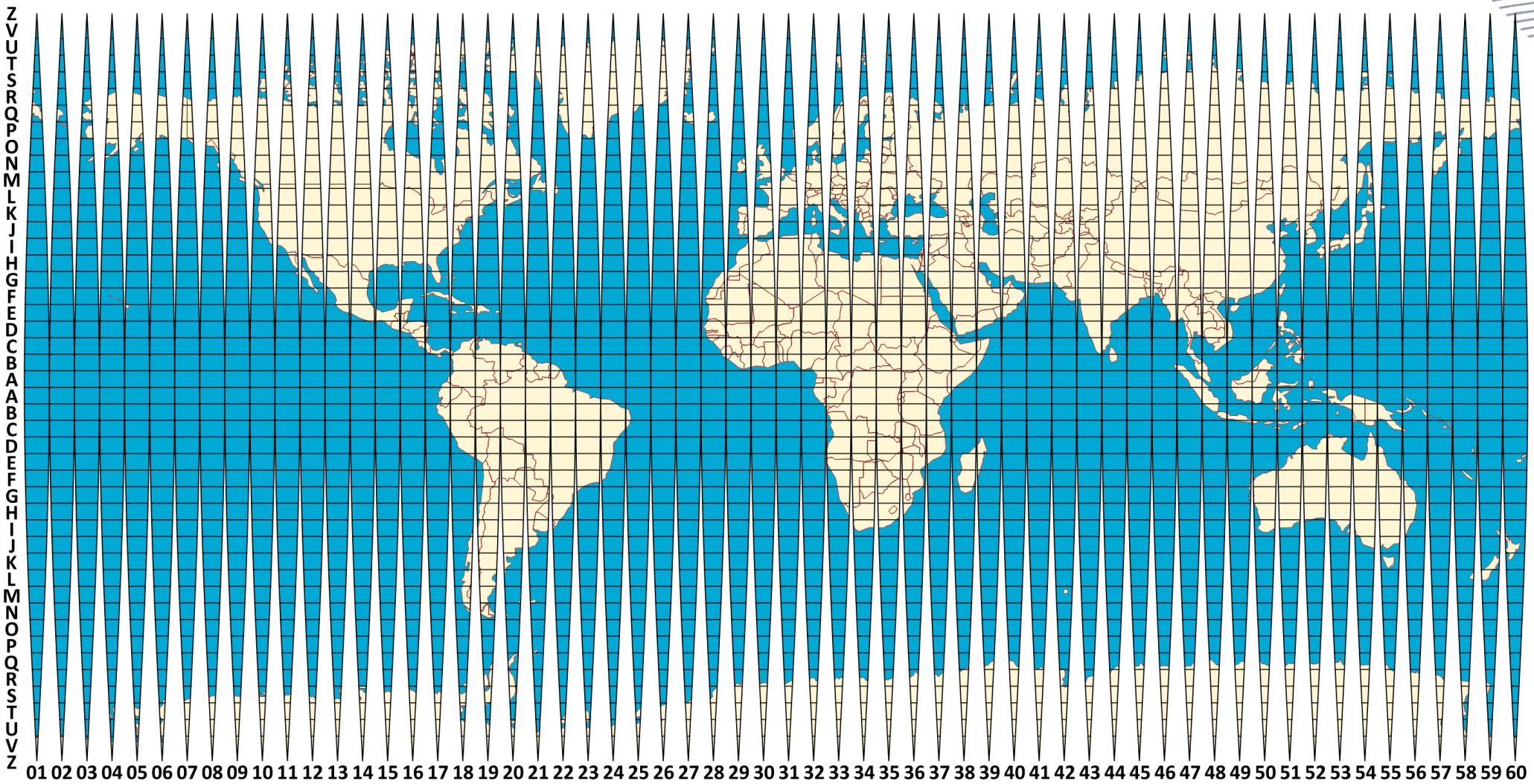
Расчет планарных расстояний

```
import pyproj
from shapely.ops import transform

pulkovo = pyproj.CRS(20006)
mercator = pyproj.CRS(3857)

project = pyproj.Transformer.from_crs(mercator, pulkovo,
always_xy=True).transform
train_station_centroid_pulkovo = transform(project, train_station_centroid)

df_distances["planar_pulkovo"] =
df_mercator_amenities.to_crs(20006).distance(
    train_station_centroid_pulkovo
)
```



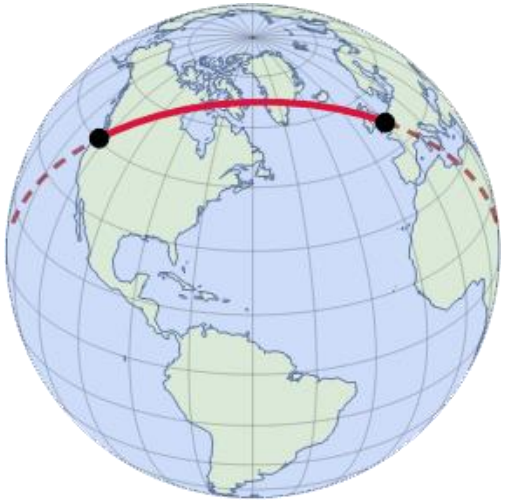
https://geoproj.ru/

Расчет планарных расстояний



Геодезические расстояния: гаверсинусное расстояние

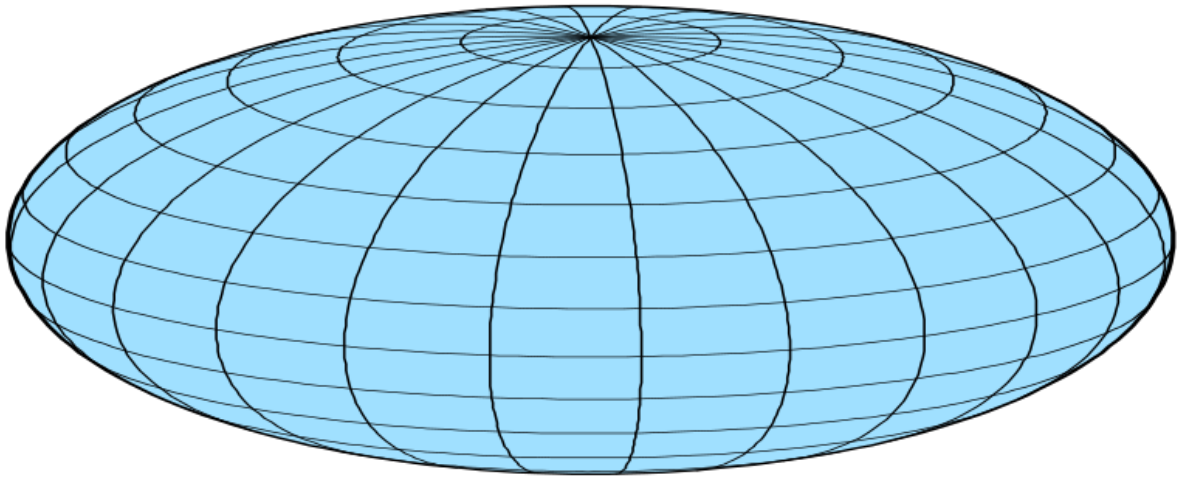
```python
from sklearn.metrics.pairwise import haversine_distances
from math import radians

def calculate_haversine_distance(first, second):
    first_in_radians = [[radians(_[0]) for _ in first.xy[::-1]]]
    second_in_radians = [[radians(_[0]) for _ in second.xy[::-1]]]
    result = haversine_distances(first_in_radians, second_in_radians)
    return result[0][0] * 6371000

df_distances["haversine"] =
df_mercator_amenities.to_crs(4326).geometry.apply(
    lambda x: calculate_haversine_distance(x, train_station_centroid_wgs)
)
```

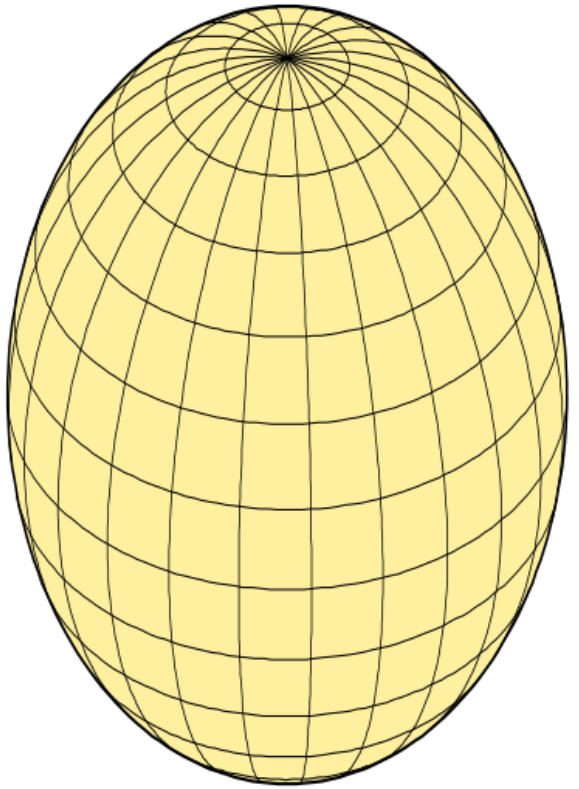


Погрешность измерения около 1–2%

Расчет геодезического расстояния

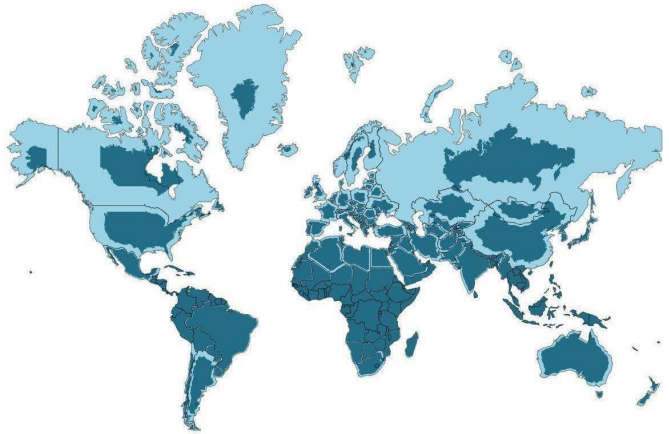
```python
import geopy.distance

def calculate_geodesic_distance(first, second):
    first_list = [_[0] for _ in first.xy[::-1]]
    second_list = [_[0] for _ in second.xy[::-1]]
    return geopy.distance.geodesic(first_list, second_list).m

df_distances["geodesic"] = df_mercator_amenities.to_crs(4326).geometry.apply(
    lambda x: calculate_geodesic_distance(x, train_station_centroid_wgs)
)
```

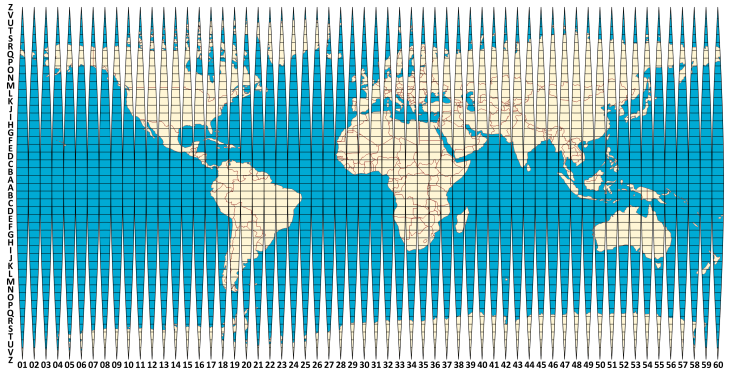


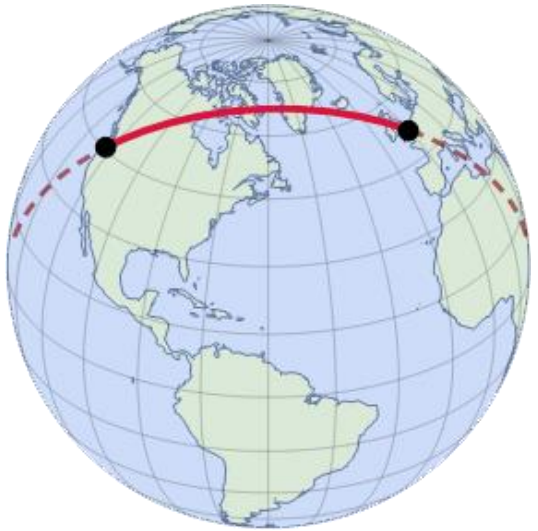


Сравнение расстояний: MAE расстояний

Псевдо-Меркатор	Пулково 1995. Зона 6	Гаверсинусное расстояние
506,2 м	0,1 м	1,6 м







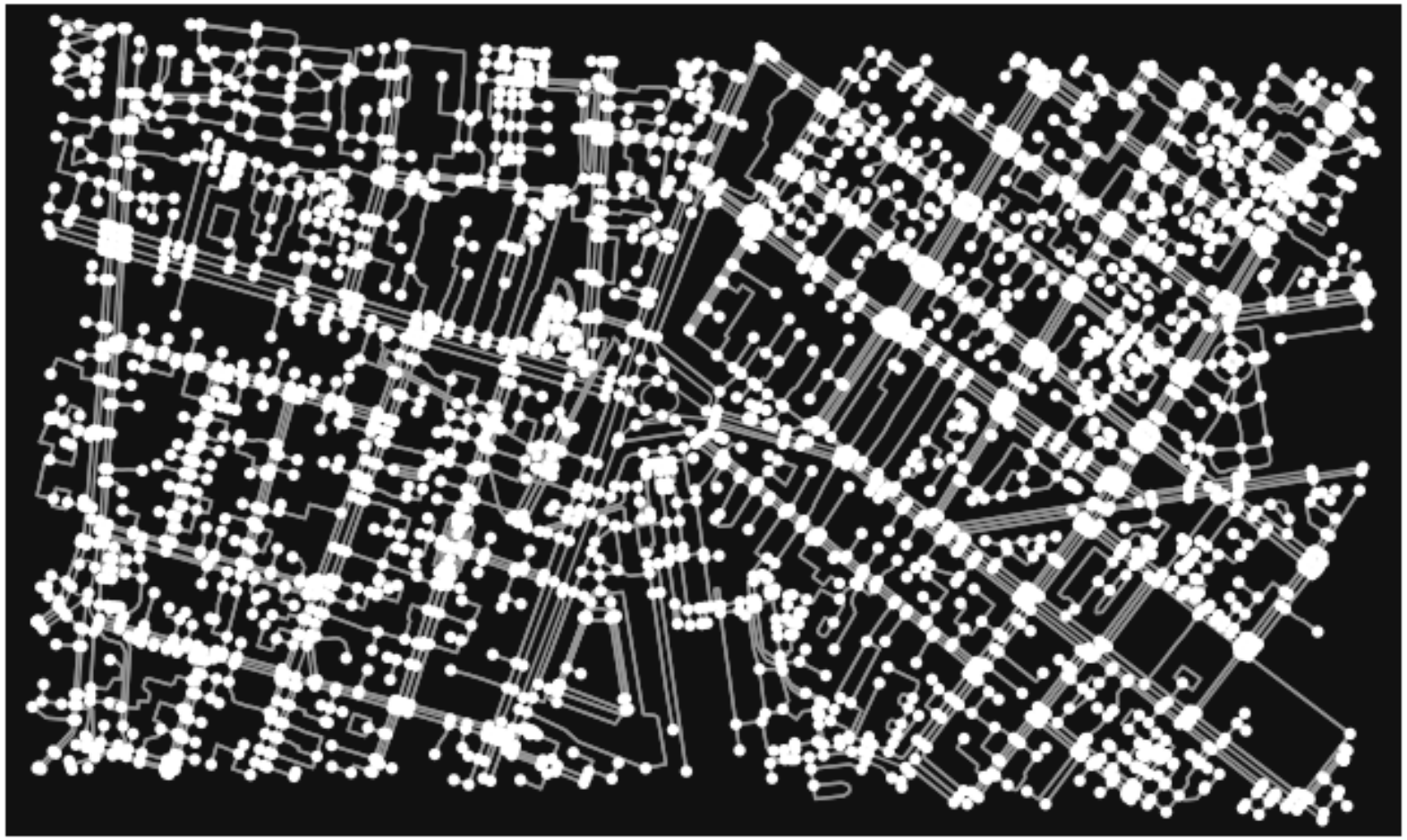
CINIMEX
Вместе быть лучшими!

Расстояние маршрута

www.cinimex.ru

Расчет расстояния маршрута

```
import osmnx

graph = osmnx.graph_from_bbox(bbox=(59.9355, 59.9259, 30.3798, 30.3461))
graph_pulkovo = ox.project_graph(graph, to_crs=20006)
osmnx.plot_graph(graph_pulkovo)
```

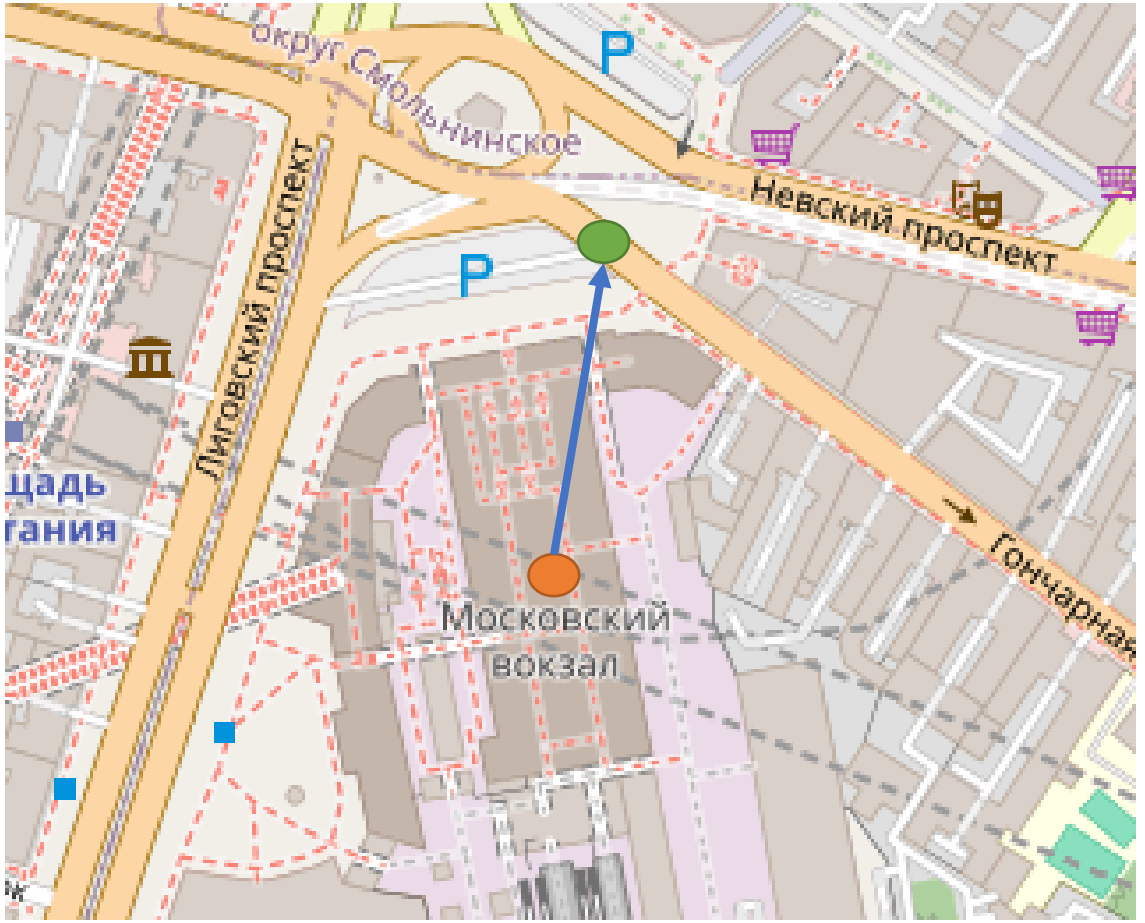


Расчет расстояния маршрута

```
df_pulkovo_amenities = df_mercator_amenities.to_crs(20006)
df_pulkovo_amenities.geometry
```

```
   element_type  osmid
   node          444776758      POINT (6352761.310 6649373.312)
                 455297491      POINT (6353232.484 6649361.507)
                 497695708      POINT (6353183.312 6649860.008)
                 654105840      POINT (6352905.622 6649355.163)
                 992360955      POINT (6352392.477 6648974.327)
                                          ...
                 11747735387    POINT (6352199.616 6649658.810)
                 11747735389    POINT (6352815.731 6649387.394)
                 11928137690    POINT (6352220.100 6649518.366)
                 11930817914    POINT (6351961.475 6649222.176)
                 11932151847    POINT (6352382.227 6648948.671)
   Name: geometry, Length: 308, dtype: geometry
```

```
node_train_station, dist_train_station = osmnx.nearest_nodes(
    graph_pulkovo,
    train_station_centroid_pulkovo.xy[0],
    train_station_centroid_pulkovo.xy[1],
    return_dist=True,
)
node_amenities, dist_amenities = osmnx.nearest_nodes(
    graph_pulkovo,
    df_pulkovo_amenities.geometry.apply(lambda x: x.xy[0][0]),
    df_pulkovo_amenities.geometry.apply(lambda x: x.xy[1][0]),
    return_dist=True,
)
```



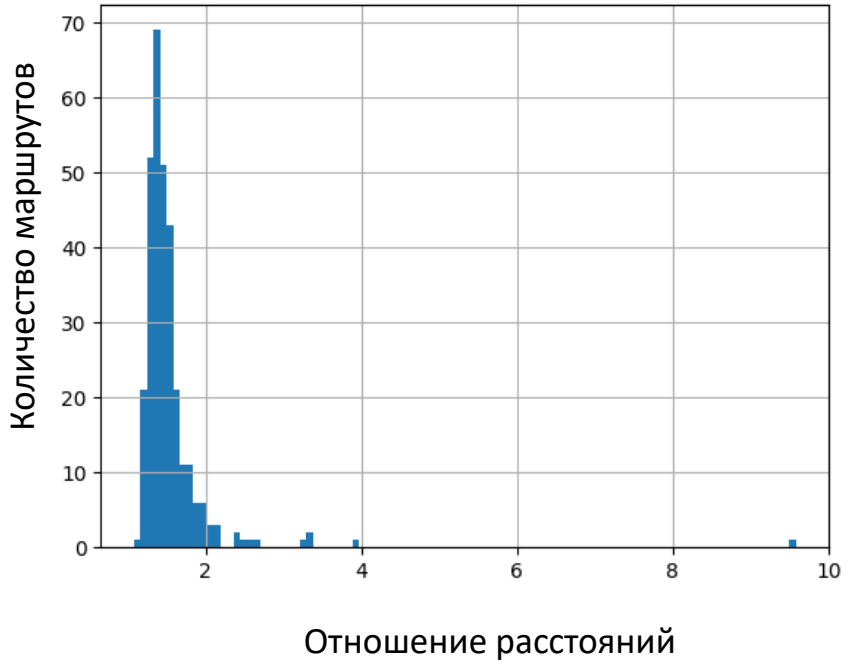
Расчет расстояния маршрута

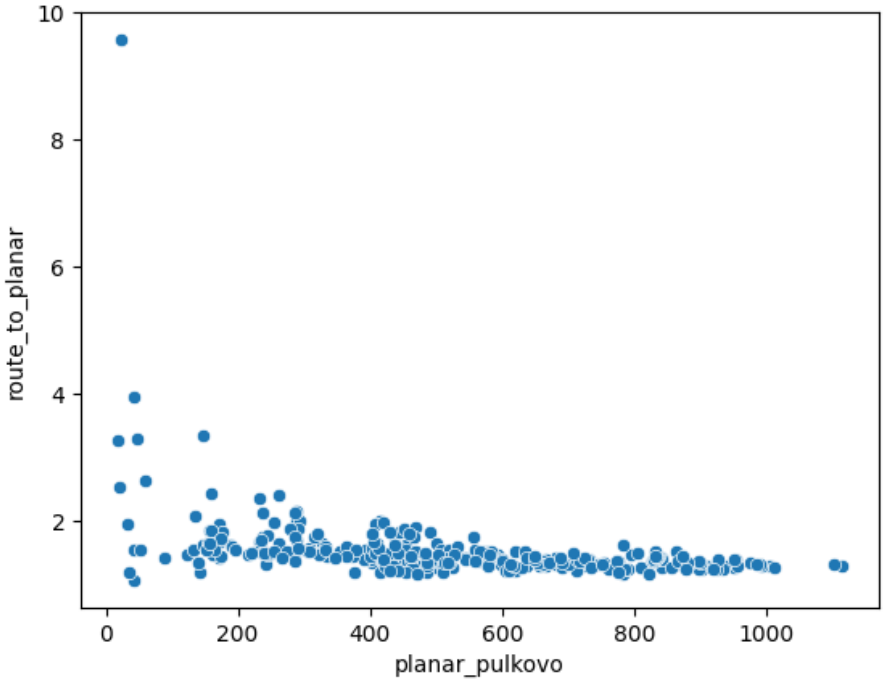
```
import numpy as np

route = ox.shortest_path(
    graph_pulkovo,
    np.repeat(node_train_station, len(node_amenities)),
    node_amenities,
    weight="length"
)

def calculate_route_length(routes, graph):
    route_length = []
    for route in routes:
        route_length.append(
            ox.routing.route_to_gdf(
                graph,
                route,
                weight="length"
            )["length"].sum()
        )
    return route_length

route_length = calculate_route_length(route, graph_pulkovo)
```

Маршрут и кратчайшее расстояние

```
df_distances["route_pulkovo"] = route_length
df_distances["route_pulkovo"] += dist_train_station[0]
df_distances["route_pulkovo"] += dist_amenities
df_distances["route_to_planar"] = df_distances["route_pulkovo"] /
df_distances["planar_pulkovo"]

df_distances["route_to_planar"].describe()
```

```
count    308.000000
mean       1.539978
std        0.562641
min        1.077376
25%        1.342092
50%        1.443546
75%        1.552086
max        9.582337
Name: route_to_planar, dtype: float64
```





CINIMEX
Вместе быть лучшими!

Пространственные объединения

www.cinimex.ru

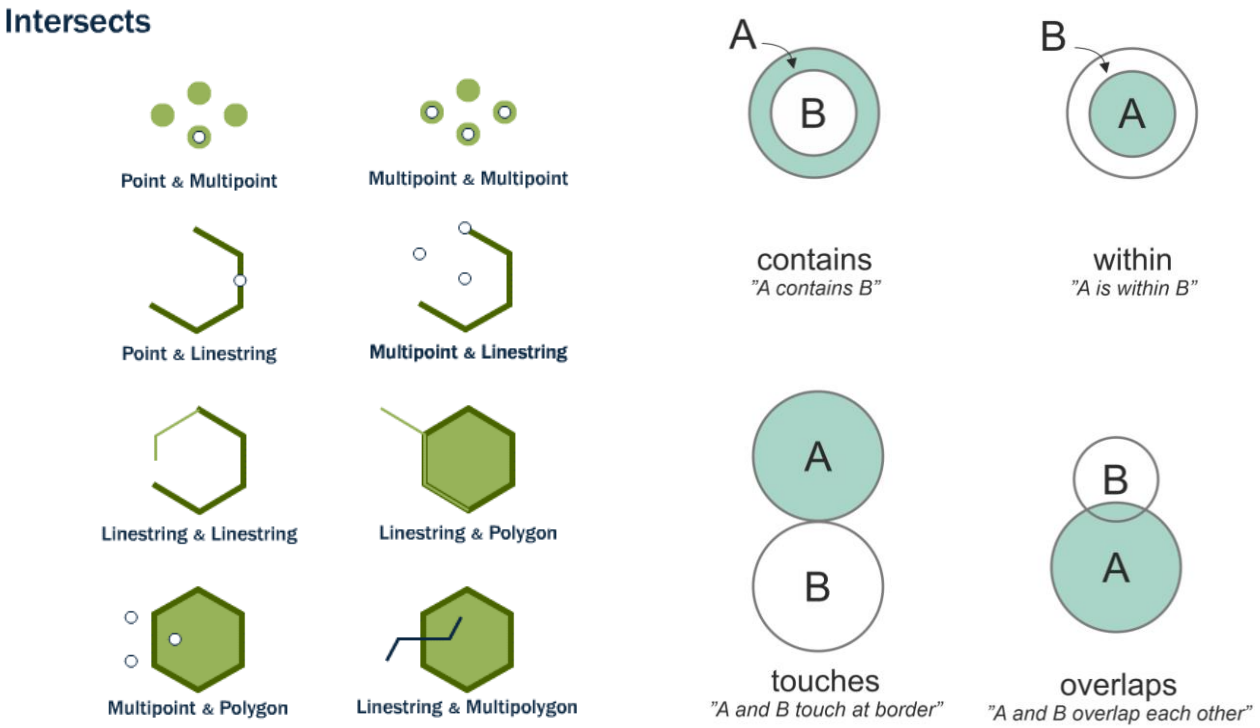
Первый закон географии Тоблера

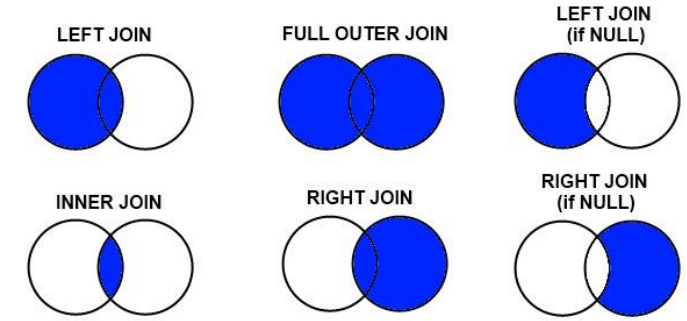
"Все связано со всем, но близкие вещи связаны сильнее, чем далекие"

Waldo R. Tobler

Пространственные объединения

```
df_amenities_in_buildings = df_buildings_mercator[["geometry",
"name"]].sjoin(
    df_mercator_amenities[["geometry", "amenity"]],
    how="left",
    predicate="contains",
)
```





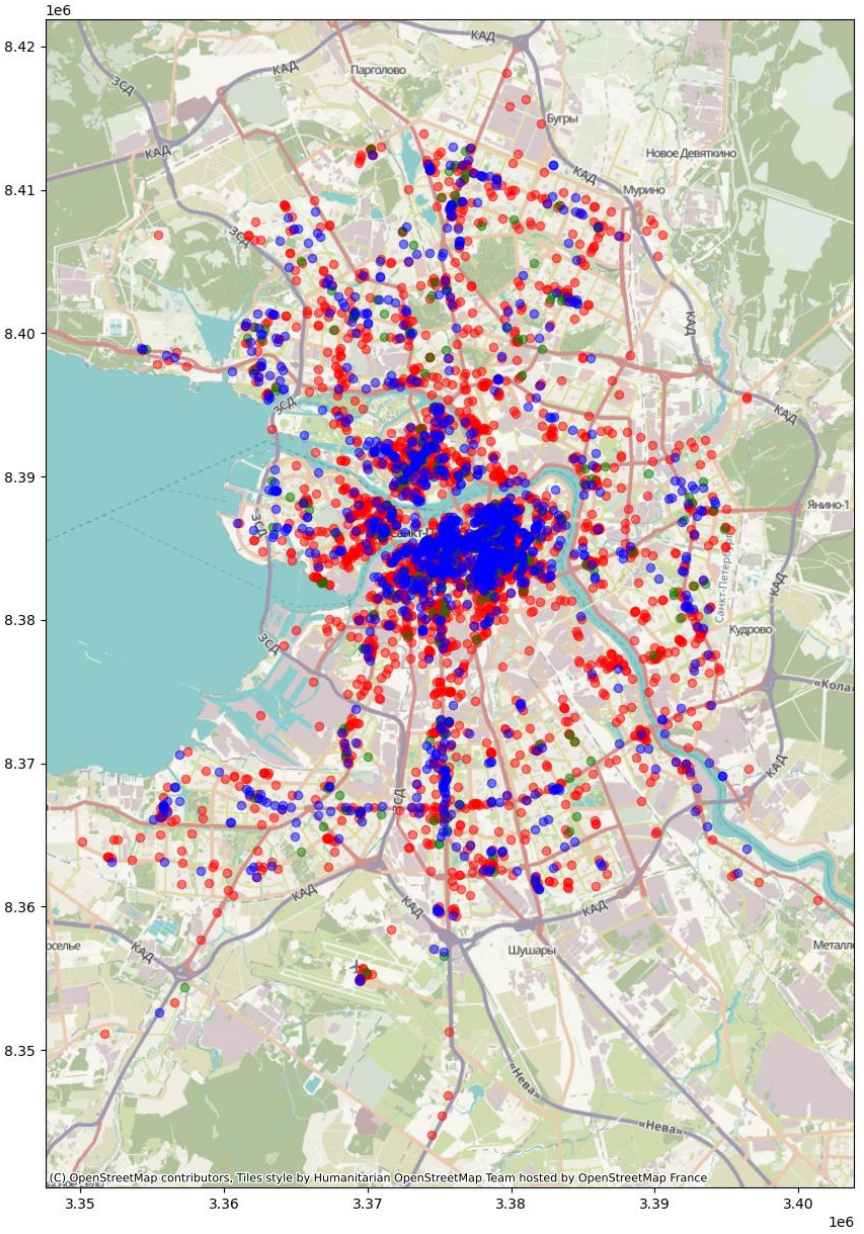
CINIMEX
Вместе быть лучшими!

Агрегация точек

www.cinimex.ru

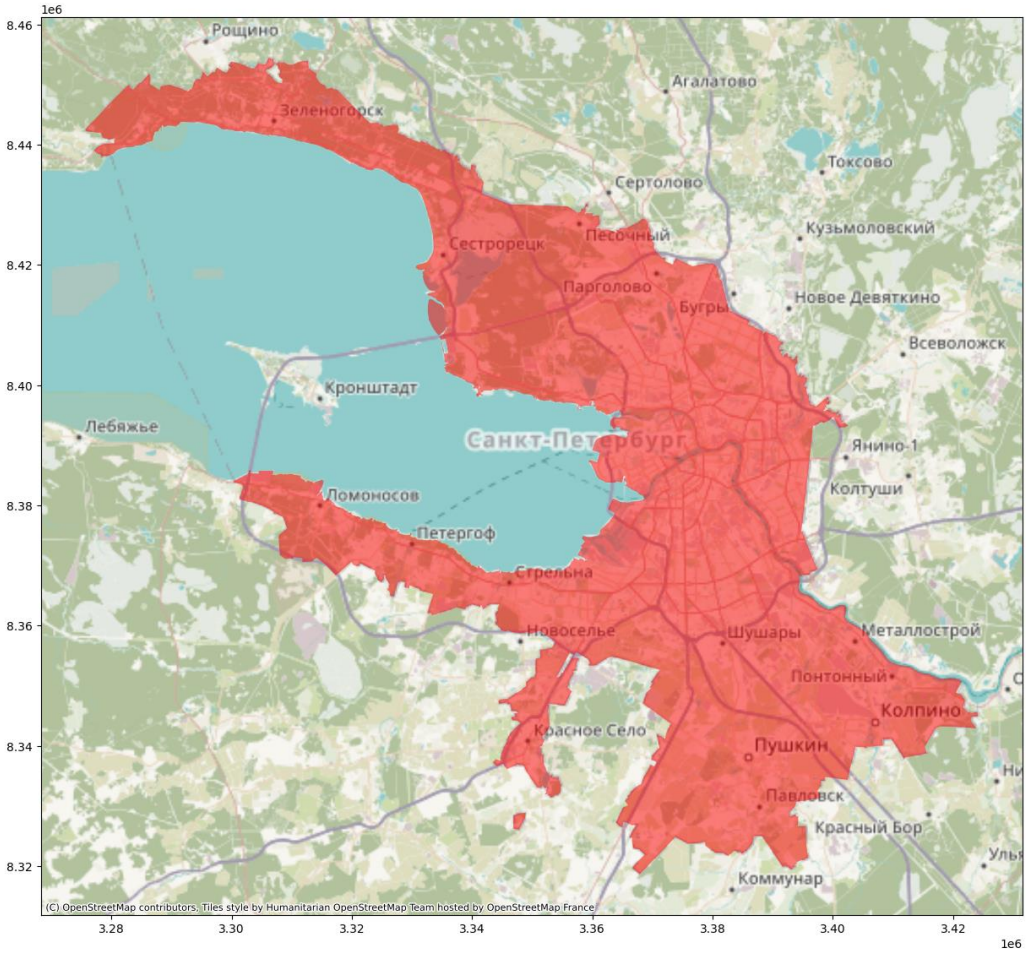
Зачем нам агрегировать данные?

```
df_all_amenities = osmnx.features_from_place(
    "Санкт-Петербург",
    tags={
        "amenity": ["cafe", "pub", "restaurant"],
        }
).query("element_type == 'node'")
df_all_amenities.head(5)
```



Административное деление

```
df_spb_subdivision = osmnx.features_from_place(
    "Санкт-Петербург",
    tags={
        "admin_level": "5",
        }
).query("element_type == 'relation'")
```

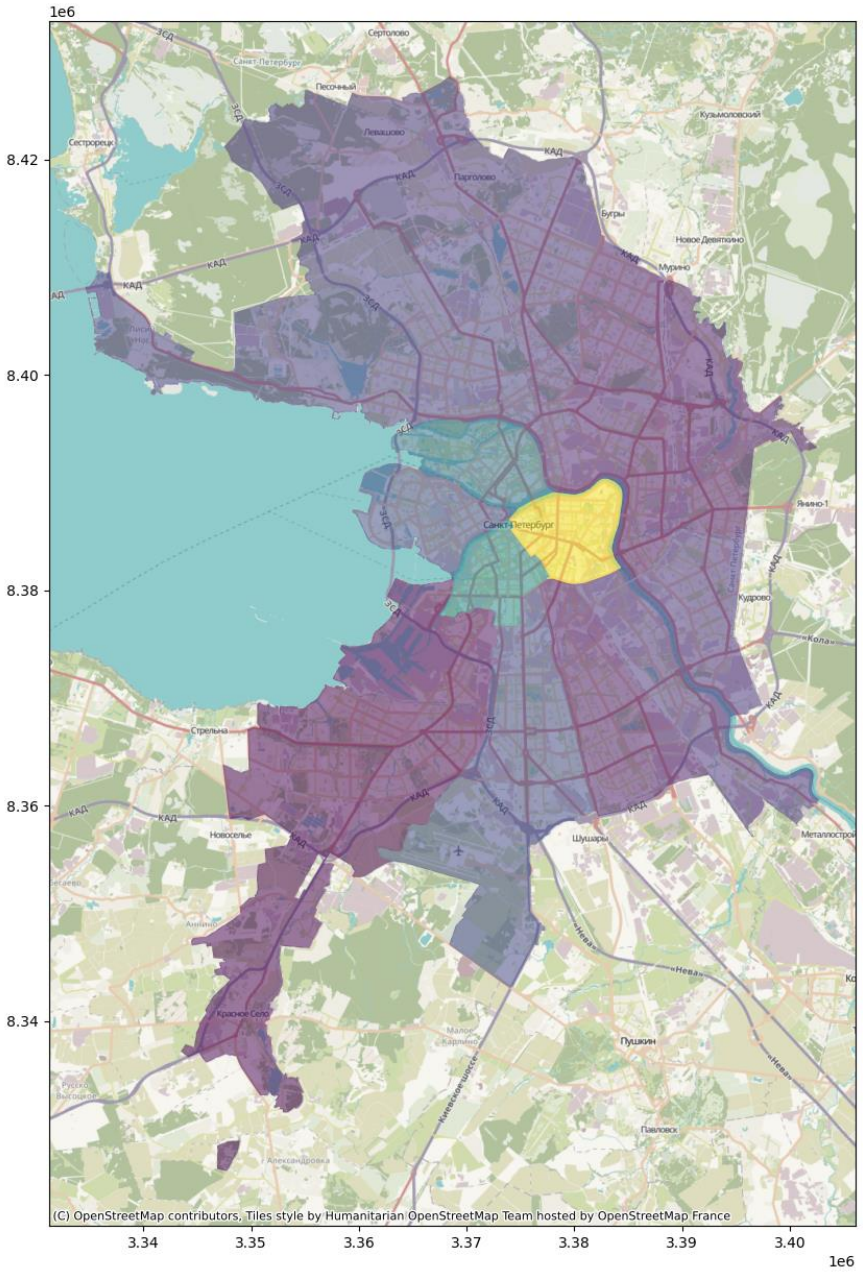



```
df_amenities_by_subregions = df_spb_subdivision[
    ["name", "geometry"]
].sjoin(
    df_all_amenities[["name", "amenity", "geometry"]],
    how="left",
    predicate="contains",
    rsuffix="amenity",
    lsuffix="subregion",
)
df_amenities_by_subregions.head(5)
```

element_type	osmid	name_subregion	geometry	index_amenity0	index_amenity1	name_amenity	amenity
relation	337424	Колпинский район	POLYGON ((30.44112 59.82937, 30.44333 59.82980...	NaN	NaN	NaN	NaN
	338635	Пушкинский район	POLYGON ((30.21822 59.67307, 30.21918 59.67361...	NaN	NaN	NaN	NaN
	338636	Московский район	POLYGON ((30.19877 59.80161, 30.19901 59.80180...	node	1.890210e+09	NaN	cafe
	338636	Московский район	POLYGON ((30.19877 59.80161, 30.19901 59.80180...	node	1.156439e+10	Пилотаж	pub
	338636	Московский район	POLYGON ((30.19877 59.80161, 30.19901 59.80180...	node	1.414117e+09	Евразия	restaurant

Административное деление

```
fig, ax = plt.subplots(figsize=(15, 15))
df_aggregated_amenities.query("amenity == 'cafe'").to_crs(3857).plot(ax=ax,
column="count", alpha=0.5)
cx.add_basemap(ax)
```



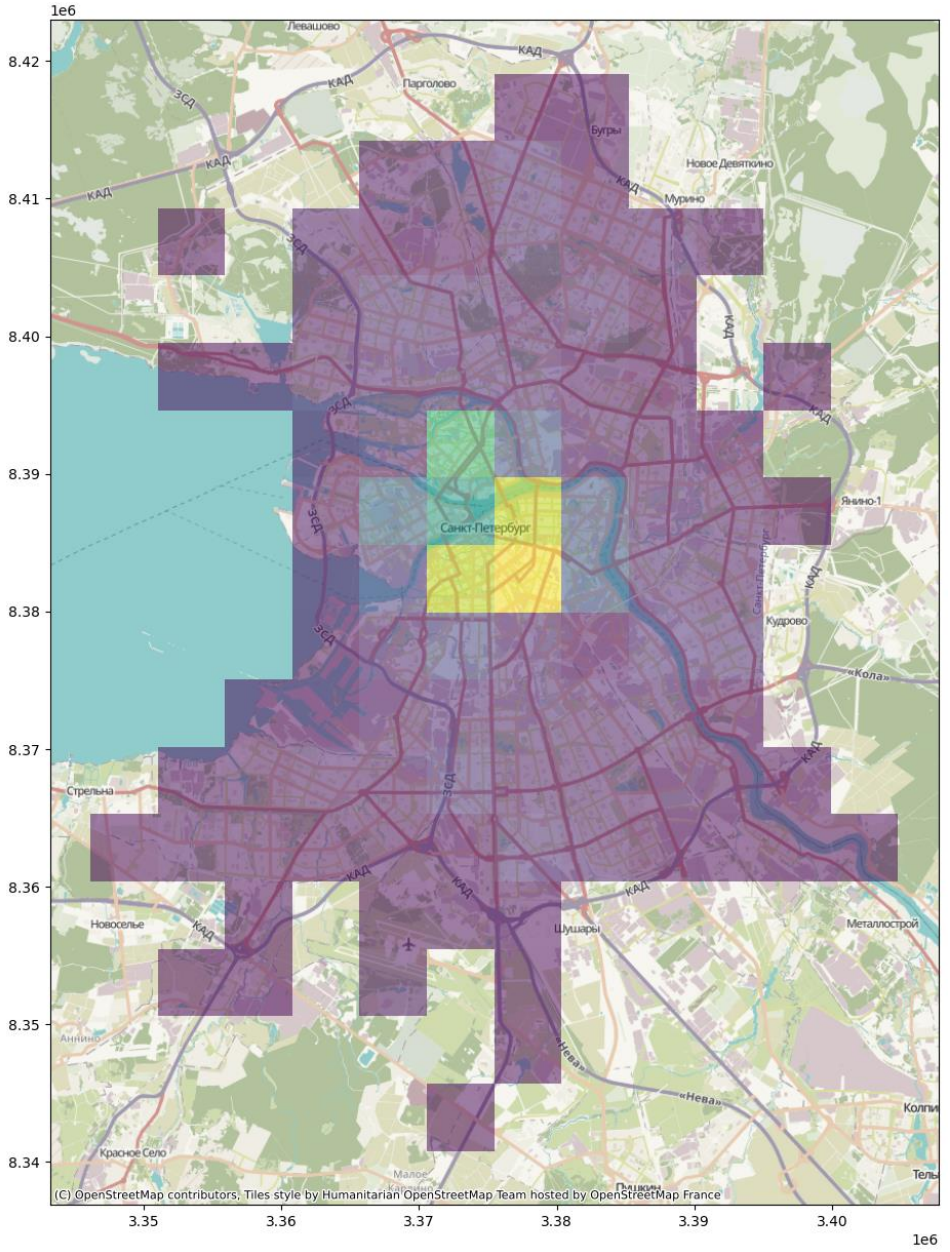
Тесселяция тайлами

```
from tesspy import Tessellation

spb_tesselation = Tessellation("Санкт-Петербург")
df_spb_squares = spb_tesselation.squares(13)
df_spb_squares.head(5)
```

	geometry	quadkey
0	POLYGON ((30.05859 59.75640, 30.05859 59.77852...	1201212121213
1	POLYGON ((30.10254 60.02095, 30.10254 60.04290...	1201212103102
2	POLYGON ((30.10254 59.99899, 30.10254 60.02095...	1201212103120
3	POLYGON ((30.10254 59.97701, 30.10254 59.99899...	1201212103122
4	POLYGON ((30.10254 59.84481, 30.10254 59.86688...	1201212121102

Тесселяция тайлами

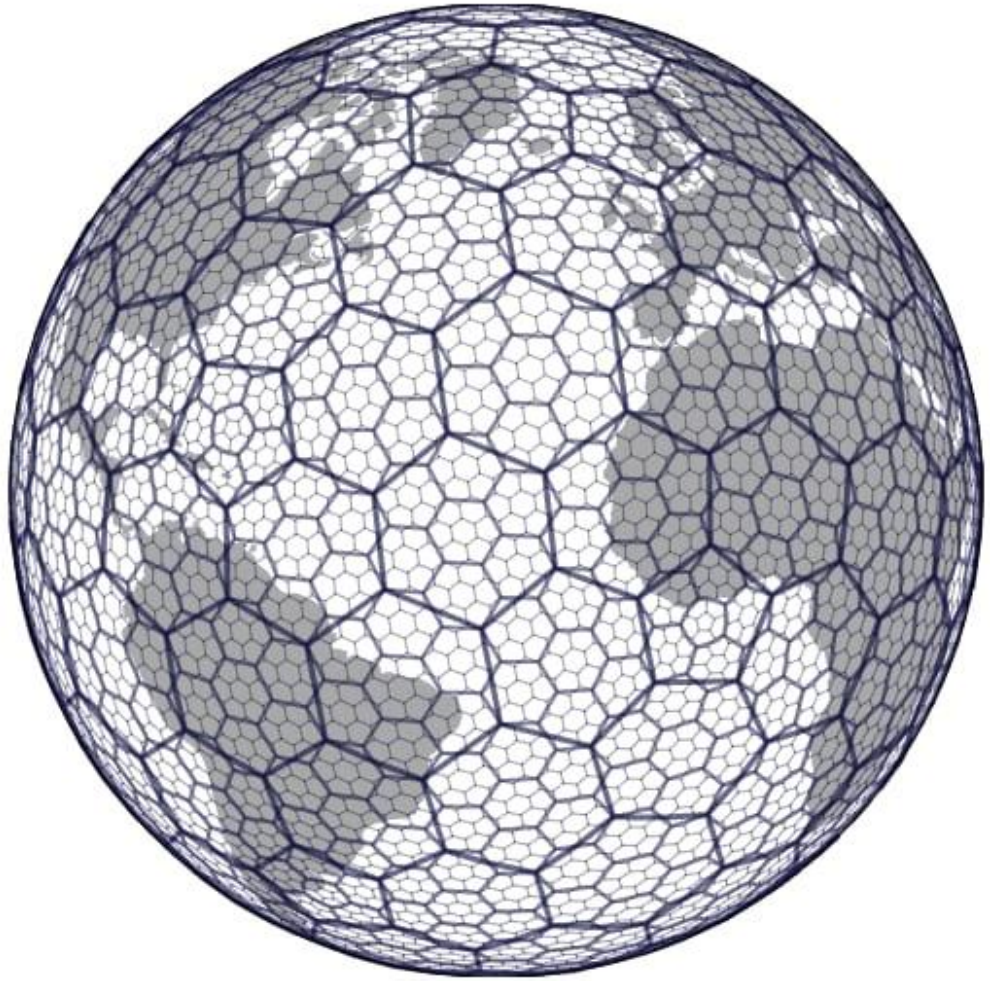
```
df_amenities_by_squares = df_spb_squares.sjoin(
    df_all_amenities[["name", "amenity", "geometry"]],
    how="left",
    predicate="contains",
    rsuffix="amenity",
    lsuffix="square",
)

df_aggregated_amenities_squares = df_amenities_by_squares.groupby(
    ["quadkey"]
)["amenity"].value_counts().to_frame()

df_aggregated_amenities_squares =
df_aggregated_amenities_squares.reset_index(
).merge(
    df_spb_squares[["quadkey", "geometry"]],
    how="left",
    on="quadkey",
)
df_aggregated_amenities_squares = gpd.GeoDataFrame(
    df_aggregated_amenities_squares,
    geometry="geometry",
)

fig, ax = plt.subplots(figsize=(15, 15))
df_aggregated_amenities_squares.query("amenity == 'cafe'").to_crs(3857).plot(
    ax=ax,
    column="count",
    alpha=0.5,
)
cx.add_basemap(ax)
```



H3 индекс

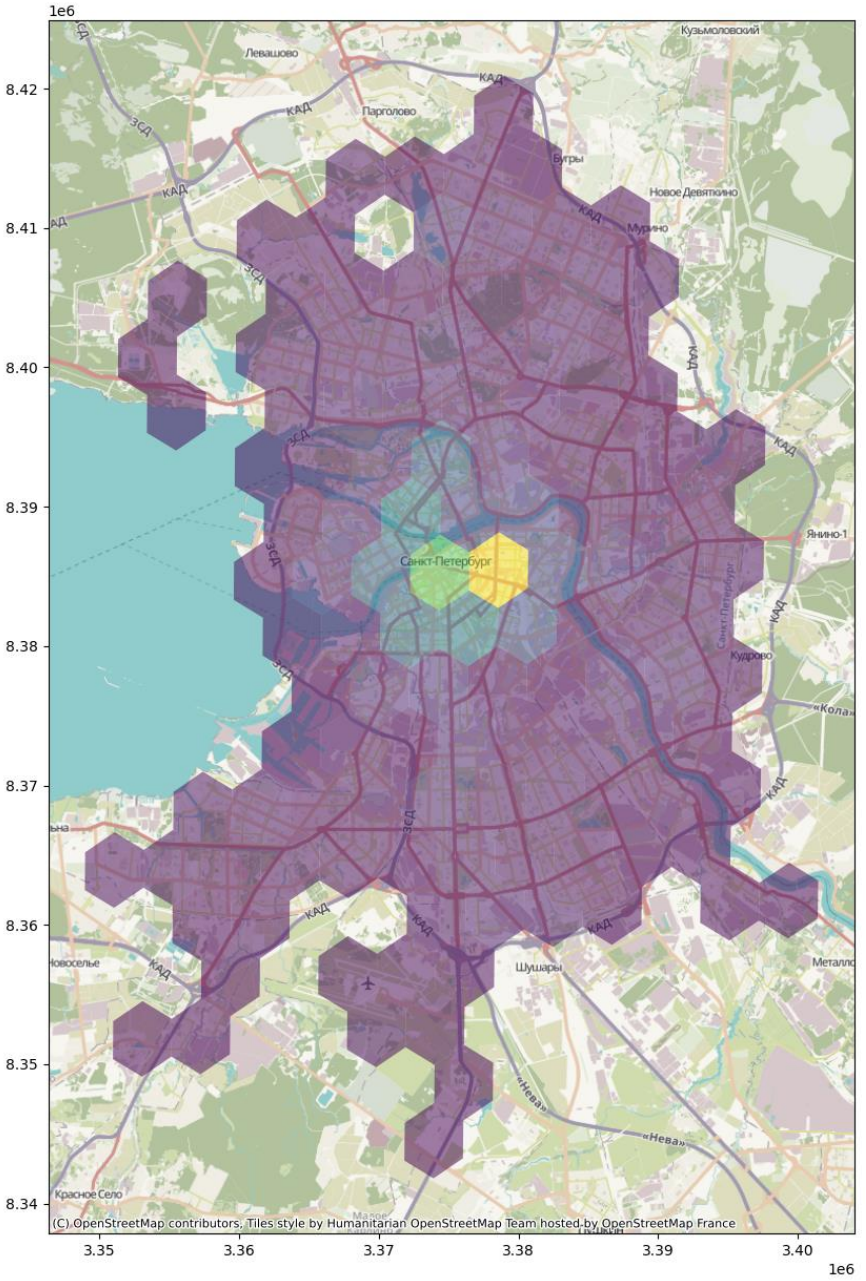
```
import h3pandas
RESOLUTION = 7
df_h3_amenities = df_all_amenities.h3.geo_to_h3(RESOLUTION)
df_h3_amenities.head(5)
```

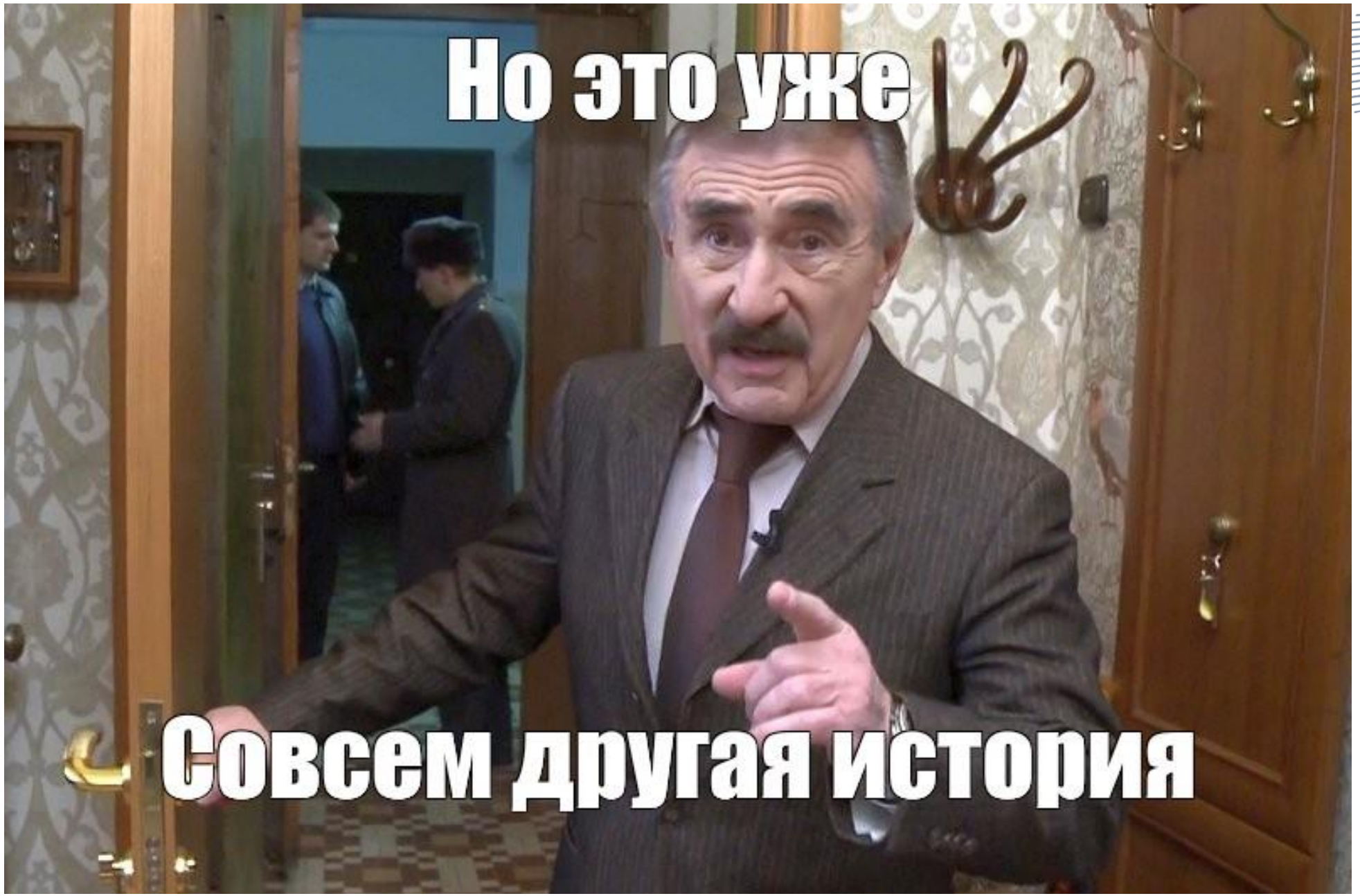
h3_07	amenity	cuisine	name	geometry	brand	brand:wikidata
87110614dffffff	restaurant	fish	Русская рыбалка	POINT (30.23327 59.96932)	NaN	NaN
871106a96ffffff	restaurant	italian;pizza;pasta	Mama Roma	POINT (30.30687 59.96281)	Mama Roma	Q109947151
871106a96ffffff	restaurant	jewish	Снежинка	POINT (30.30168 59.96316)	NaN	NaN
871106149ffffff	restaurant	georgian	Хачо и Пури	POINT (30.28602 59.96669)	NaN	NaN
871106065ffffff	cafe	coffee_shop	Больше Кофе	POINT (30.32051 59.95450)	NaN	NaN



H3 индекс

```
df_h3_amenities_aggregated = df_h3_amenities.groupby(
    "h3_{:>02}".format(RESOLUTION)
)["amenity"].value_counts().to_frame()

fig, ax = plt.subplots(figsize=(15, 15))
df_h3_amenities_aggregated.reset_index().set_index(
    "h3_{:>02}".format(RESOLUTION)
).h3.h3_to_geo_boundary().query(
    "amenity == 'cafe'"
).to_crs(3857).plot(ax=ax, column="count", alpha=0.5)
cx.add_basemap(ax)
```




Но это уже
Совсем другая история

CINIMEX
Вместе быть лучшими!

СПАСИБО ЗА ВНИМАНИЕ!

www.cinimex.ru

Готов ответить на ваши вопросы

Мещеряков Александр Олегович

Синимекс

E-mail: amecheryakov@cinimex.ru