

Отчет по проекту "Анализ городской транспортной среды как графа". Трамвайная сеть города Санкт-Петербург.

Содержание:

1. Введение
2. Цели и задачи проекта
3. Объект исследования и данные
4. Методология
5. Заключение
6. Список используемой литературы
7. Команда проекта

1. Введение:

Транспортная система мегаполиса - это сложный механизм, эффективность которого напрямую зависит от правильного распределения нагрузки всех видов общественного транспорта, в том числе и трамваев. В данном проекте мы рассматриваем трамвайную сеть города Санкт-Петербург как граф. Это позволяет применить алгоритмы анализа графов для выявления ключевых узлов и оценки структуры городской логистики.

2. Цели и задачи проекта:

Провести комплексный анализ трамвайной сети города Санкт-Петербург и определить:

- 1) топ-маршруты
- 2) самые длинные маршруты
- 3) количество компонентов связности графа

### 3. Объект исследования и данные

Исследуемые параметры:

Остановки: ID, название, географические координаты (широта/долгота), район города.

Маршруты: ID маршрута, номер маршрута, тип маршрута транспорта.

В процессе обработки данных были вычислены дополнительные метрики: количество маршрутов через каждую остановку и общая протяженность каждого маршрута в километрах.

### 4. Методология

```
import pandas as pd
from collections import defaultdict, deque

df = pd.read_csv('routes_raw_data - routes_raw_data (2).csv')

# df = df[df['Направление движения'] == 2 ]

df = df[df['Тип транспорта'] == 'Трамвай']

print("=== АНАЛИЗ СВЯЗНОСТИ МАРШРУТОВ ===")
print(f"Всего записей: {len(df)}")
print(f"Уникальных маршрутов: {df['Идентификатор маршрута'].nunique()}")
print(f"Уникальных остановок: {df['ID остановки'].nunique()}\n")

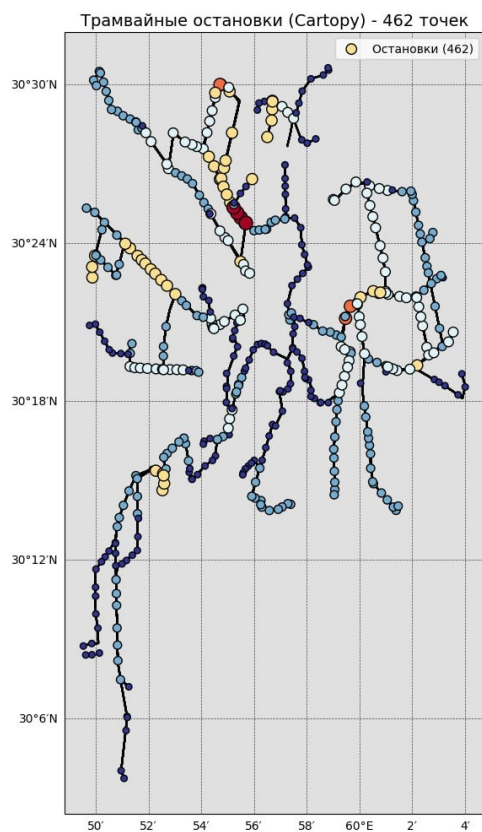
# 1. Построение графа связности
graph = defaultdict(set)
route_stops = defaultdict(set) # остановки по маршрутам

for _, row in df.iterrows():
    stop_id = row['ID остановки']
    next_stop = row['ID следующей остановки']
    route_id = f"{row['Идентификатор маршрута']}_{row['Числовой номер маршрута']}"

    # Добавляем связь между текущей и следующей остановкой
    if next_stop:
        graph[stop_id].add(next_stop)
        graph[next_stop].add(stop_id) # для неориентированного графа
```

Проект реализовывался в несколько этапов:

Моделирование: Представление трамвайной сети в виде графа. Каждая трамвайная остановка стала вершиной, а отрезок пути между двумя соседними остановками в рамках одного маршрута - ребром. С помощью программных средств (структуры данных на Python) сопоставили таблицы остановок с маршрутными листами.



Анализ:

- 1) Для поиска самых загруженных остановок мы подсчитали количество входящих в них ребер (маршрутов).
- 2) Для анализа длин маршрутов был проведен расчет количества остановок в каждом из них.
- 3) Для проверки связности мы использовали алгоритмы обхода графа, чтобы определить, на сколько изолированных групп (компонент) разбита сеть.

```

for i, row in tram_routes.iterrows():
    stop_id = row['ID остановки']
    coordinates = row['Координаты остановки']

    stops_dict[stop_id] = coordinates

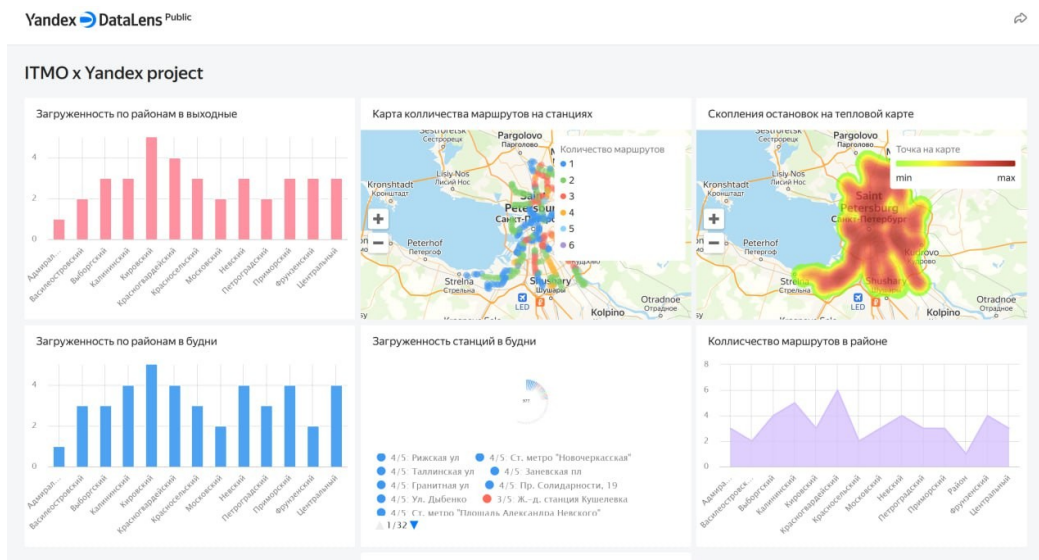
for i, row in tram_routes.iterrows():
    current_route_id = row['Идентификатор маршрута']
    current_stop_id = row['ID остановки']
    next_stop_id = row['ID следующей остановки']

# 4. Для каждой строки (каждой остановки) записываем её саму
results.append({
    'ID маршрута': current_route_id,
    'Номер маршрута': row['Числовой номер маршрута'],
    'ID остановки': current_stop_id,
    'Координаты': stops_dict.get(current_stop_id, 'Не найдено'),
})

# 5. Если есть следующая остановка, записываем и её (из словаря)
if pd.notna(next_stop_id) and next_stop_id in stops_dict:
    results.append({
        'ID маршрута': current_route_id,
        'Номер маршрута': row['Числовой номер маршрута'],
        'ID остановки': next_stop_id,
        'Координаты': stops_dict[next_stop_id],
    })

```

Визуализация: Данные были экспортированы в Yandex DataLens. Размер и цвет каждой точки зависели от её значимости (количества проходящих маршрутов).



## 5. Заключение

В ходе работы над проектом были получены следующие результаты:

Топ-остановок: Выявлены 5 ключевых узлов, являющихся главными пересадочными пунктами города. Эти остановки обслуживают максимальное количество трамвайных линий.

Длины маршрутов: Определены самый протяженный маршрут (с наибольшим числом остановок) и самый короткий (локальный).

Связность сети: перераспределили количество компонентов связности. Это позволило понять, является ли трамвайная сеть единым целым или она разделена на несколько автономных сегментов, то есть не связанных друг с другом рельсами.

## 6. Список используемой литературы

Для проекта используются открытые GTFS-данные \*(General Transit Feed Specification)\*, опубликованные на государственном портале открытых данных Санкт-Петербурга

- <https://data.gov.spb.ru/>

Используемые ресурсы:

Маршруты городского транспорта (трамваи) - <https://data.gov.spb.ru/irsi/7830001067-marshruty-dvizheniya-gorodskogo-transporta/>

Перечень остановочных пунктов с координатами

<https://data.gov.spb.ru/irsi/7830001067-perechen-ostanovochnyh-punktov-s-ukazaniem-vida-transporta-i-s-koordinatami-ih-mestopolozheniya/>

Распакованный архив с сырыми данными, выгруженный с data.gov.spb.ru - [https://github.com/artiomkonovalov/Transport-Analysis/tree/main/raw\\_data/](https://github.com/artiomkonovalov/Transport-Analysis/tree/main/raw_data/)

## 7. Команда проекта

### 1. Артём Коновалов

Координация команды, распределение задач и контроль их выполнения. Помощь другим участникам команды

### 2. Амгалан Гомбоев

Выбор источников, сбор данных

### 3. Арина Муравьёва

Анализ данных. Выбор данных для исследования, анализ источников, проектирование схемы данных

### 4. Максим Лобанов

Ведение документации, оформление отчёта, презентации

### 5. Андрей Языков

Представление данных в Yandex DataLens