

# PATHFINDER-API

01219335 Data Acquisition and Integration



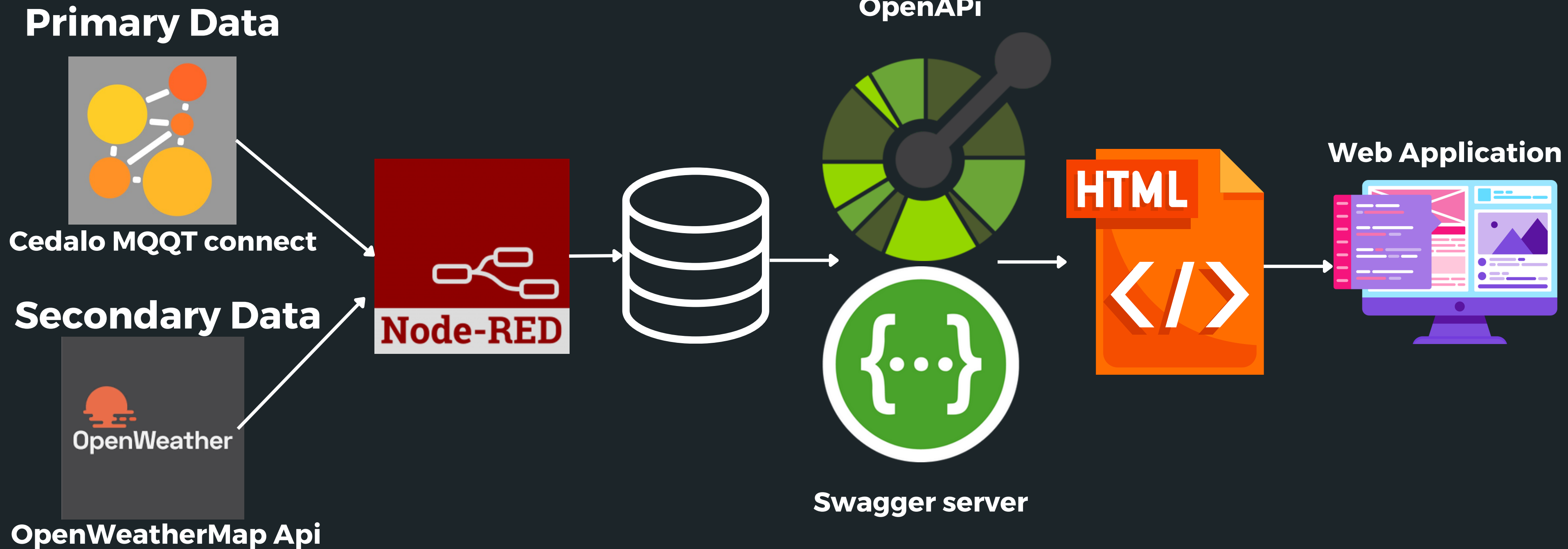


# Project Overview

Our project is aim to investigate the relation between traffic and weather by collecting the data on traffic like linearX, linearY, linearZ, Latitude, Longitude ,and Timestamp in realtime and change into acceleration and time spent for each time we travel to University. To answer the question “Is it always traffic jam when it’s raining?”.



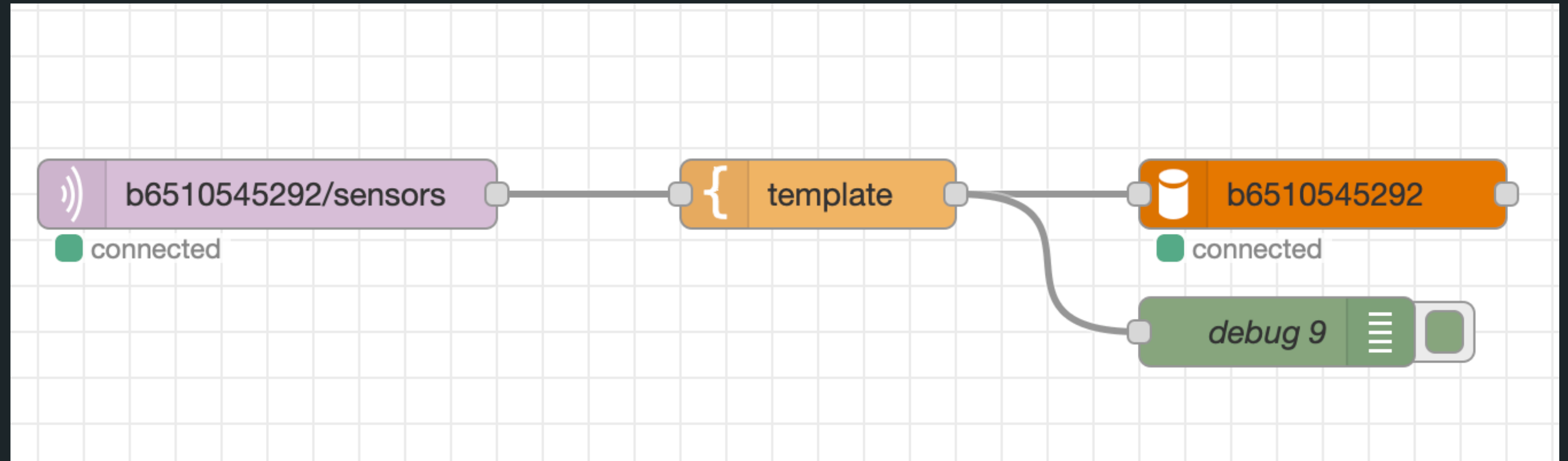
# Architecture



# Primary Data



Cedalo MQTT connect




Our primary data is collected by using Cedalo MQTT Connect which sent LinearX, LinearY, LinearZ , Lat, and Lon from our mobile phone in real time, and using NODE-RED to sent it to the database. Then we calculate the Acceleration and Time-Spent for each time we travel near University.



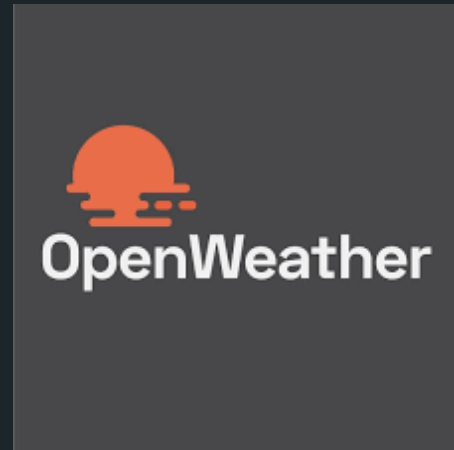
# Primary Data

| ID | LinearX | LinearY | LinearZ | Altitude | Latitude | Longitude | Timestamp           | ▲ 1 | DeviceID                             |
|----|---------|---------|---------|----------|----------|-----------|---------------------|-----|--------------------------------------|
| 57 | -3      | 1       | 3       | 8.21803  | 13.8456  | 100.567   | 2024-04-19 17:56:56 |     | 6FBF00F7-0101-472D-86EA-42D2B47459F0 |
| 58 | 2       | -3      | -2      | 8.52828  | 13.8451  | 100.567   | 2024-04-19 17:57:16 |     | 6FBF00F7-0101-472D-86EA-42D2B47459F0 |
| 59 | 0       | 3       | -2      | 8.17294  | 13.8445  | 100.568   | 2024-04-19 17:57:36 |     | 6FBF00F7-0101-472D-86EA-42D2B47459F0 |
| 60 | 0       | -2      | -2      | 8.27447  | 13.8441  | 100.569   | 2024-04-19 17:57:56 |     | 6FBF00F7-0101-472D-86EA-42D2B47459F0 |
| 61 | 0       | 2       | 1       | 8.66387  | 13.8436  | 100.57    | 2024-04-19 17:58:16 |     | 6FBF00F7-0101-472D-86EA-42D2B47459F0 |

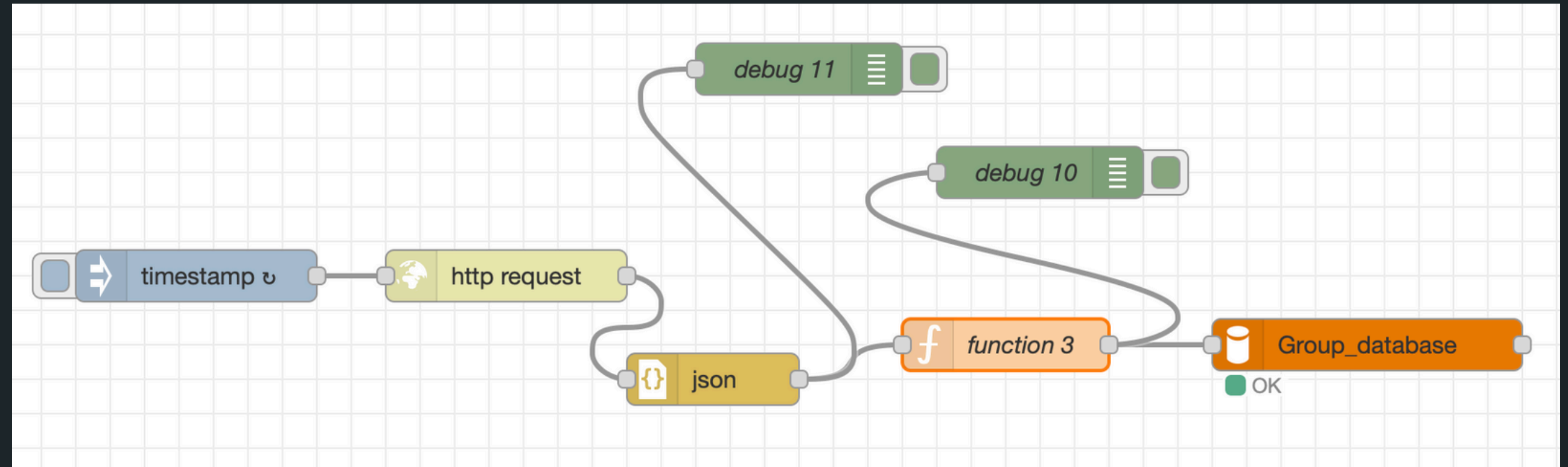


| Timestamp           | Latitude | Longitude | DeviceID                             | Acceleration | TravelID  1 |
|---------------------|----------|-----------|--------------------------------------|--------------|--|
| 2024-04-19 18:02:36 | 13.84    | 100.575   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 1            | 1  |
| 2024-04-19 18:01:16 | 13.8407  | 100.574   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 3.74166      | 1  |
| 2024-04-19 17:59:56 | 13.841   | 100.573   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 0            | 1  |
| 2024-04-19 18:02:56 | 13.84    | 100.575   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 0            | 1  |
| 2024-04-19 18:02:16 | 13.84    | 100.575   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 0            | 1  |
| 2024-04-19 17:59:36 | 13.841   | 100.573   | 6FBF00F7-0101-472D-86EA-42D2B47459F0 | 1            | 1  |

# Secondary Data





OpenWeatherMap Api



Our secondary data is collected by using API from OpenWeatherMap Api to collect current weather elements and conditions, and using NODE-RED to fetch the data.

# Database Schema

| Updated_GPS_tracker  |              |
|--|--------------|
|  <b>ID</b> | int          |
| Timestamp  | datetime?    |
| Latitude   | float?       |
| Longitude  | float?       |
| DeviceID   | varchar(40)? |
| Acceleration   | float?       |
| TravelID   | int?         |

| API_weather   |             |
|---|-------------|
|  <b>id</b> | int         |
| temp  | float       |
| hum   | int         |
| pres  | int         |
| dp  | float       |
| uni   | float       |
| cloud   | int         |
| vis   | int         |
| wmain   | varchar(40) |
| wdes  | varchar(40) |
| wicon   | varchar(5)  |
| Timestamp   | datetime?   |

# DATA SHARING API

## 1. Get/weather/{date}

Return weather elements from the specified date.

```
{
  "Timestamp": "2024-05-10T00:04:15Z",
  "cloud": 20,
  "dp": 24.02,
  "hum": 67,
  "pres": 1010,
  "temp": 30.86,
  "uvi": 0,
  "vis": 10000,
  "wdes": "few clouds",
  "wmain": "Clouds"
},
{
  "Timestamp": "2024-05-10T00:14:15Z",
  "cloud": 20,
  "dp": 24.26,
  "hum": 68,
  "pres": 1010,
  "temp": 30.86,
  "uvi": 0,
  "vis": 10000,
  "wdes": "few clouds",
  "wmain": "Clouds"
},
```

```
{
  "Timestamp": "2024-05-10T00:24:15Z",
  "cloud": 20,
  "dp": 23.69,
  "hum": 66,
  "pres": 1010,
  "temp": 30.78,
  "uvi": 0,
  "vis": 10000,
  "wdes": "few clouds",
  "wmain": "Clouds"
},
{
  "Timestamp": "2024-05-10T00:34:14Z",
  "cloud": 20,
  "dp": 23.72,
  "hum": 66,
  "pres": 1010,
  "temp": 30.81,
  "uvi": 0,
  "vis": 10000,
  "wdes": "few clouds",
  "wmain": "Clouds"
},
```



# DATA SHARING API

## 2. Get/weather/average/{date}

Return average weather elements for the specified date.

```
[
  {
    "avg_cloud": 26.3415,
    "avg_dp": 23.755,
    "avg_hum": 60.439,
    "avg_pres": 1009.3415,
    "avg_temp": 32.6485,
    "avg_uvi": 2.6278,
    "avg_vis": 10000,
    "occurrence_percentage": 86.014,
    "wmain": "Clouds"
  },
  {
    "avg_cloud": 20,
    "avg_dp": 23.1215,
    "avg_hum": 62.15,
    "avg_pres": 1010.8,
    "avg_temp": 31.2785,
    "avg_uvi": 0,
    "avg_vis": 10000,
    "occurrence_percentage": 13.986,
    "wmain": "Rain"
  }
]
```

# DATA SHARING API

## 3. Get/weather/average/rainPercent

Return average weather elements and rain percent for every date.

```
[
  {
    "avg_cloud": 20,
    "avg_dp": 24.1454,
    "avg_hum": 59,
    "avg_pres": 1006.3551,
    "avg_temp": 34.1131,
    "avg_uvi": 2.9439,
    "avg_vis": 10000,
    "date": "2024-04-25",
    "rain_percentage": 2.17
  },
  {
    "avg_cloud": 20,
    "avg_dp": 25.3674,
    "avg_hum": 63.1181,
    "avg_pres": 1005.8819,
    "avg_temp": 33.9353,
    "avg_uvi": 2.8871,
    "avg_vis": 10000,
    "date": "2024-04-26",
    "rain_percentage": 0
  },

```

```
{
  "avg_cloud": 20.4167,
  "avg_dp": 27.3244,
  "avg_hum": 68.0764,
  "avg_pres": 1005.4375,
  "avg_temp": 34.4527,
  "avg_uvi": 2.8321,
  "avg_vis": 10000,
  "date": "2024-04-27",
  "rain_percentage": 4.17
},
{
  "avg_cloud": 22.9167,
  "avg_dp": 27.2421,
  "avg_hum": 69,
  "avg_pres": 1006.4861,
  "avg_temp": 34.1031,
  "avg_uvi": 2.8087,
  "avg_vis": 10000,
  "date": "2024-04-28",
  "rain_percentage": 14.58
},

```

# DATA SHARING API

## 4. Get/traffic/details

Returns the average acceleration and time spent for every TravelID(ID for each time we travel).

```
[
  {
    "avg_acceleration": 1.9357,
    "endTime": "2024-04-19T18:10:36Z",
    "startTime": "2024-04-19T17:56:56Z",
    "time_spent": 13,
    "travelID": 1
  },
  {
    "avg_acceleration": 5.4665,
    "endTime": "2024-04-25T13:03:48Z",
    "startTime": "2024-04-25T12:56:08Z",
    "time_spent": 7,
    "travelID": 4
  },
  {
    "avg_acceleration": 2.6593,
    "endTime": "2024-05-03T12:01:00Z",
    "startTime": "2024-05-03T11:52:00Z",
    "time_spent": 9,
    "travelID": 12
  },
]
```

```
{
  "avg_acceleration": 3.3972,
  "endTime": "2024-04-25T20:15:04Z",
  "startTime": "2024-04-25T20:12:24Z",
  "time_spent": 2,
  "travelID": 5
},
{
  "avg_acceleration": 8.58,
  "endTime": "2024-05-03T15:22:43Z",
  "startTime": "2024-05-03T15:13:43Z",
  "time_spent": 9,
  "travelID": 13
},
{
  "avg_acceleration": 2.4656,
  "endTime": "2024-04-25T20:52:22Z",
  "startTime": "2024-04-25T20:44:42Z",
  "time_spent": 7,
  "travelID": 6
},
]
```



# DATA SHARING API

## 5. Get/traffic/details/{travelID}

Returns the average acceleration and time spent from the specified travelID(ID for each time we travel).

```
[
  {
    "avg_acceleration": 1.3299,
    "endTime": "2024-05-01T16:30:03Z",
    "startTime": "2024-05-01T16:22:43Z",
    "time_spent": 7,
    "travelID": 10
  }
]
```

# DATA SHARING API

## 6. Get/traffic/{TravelID}

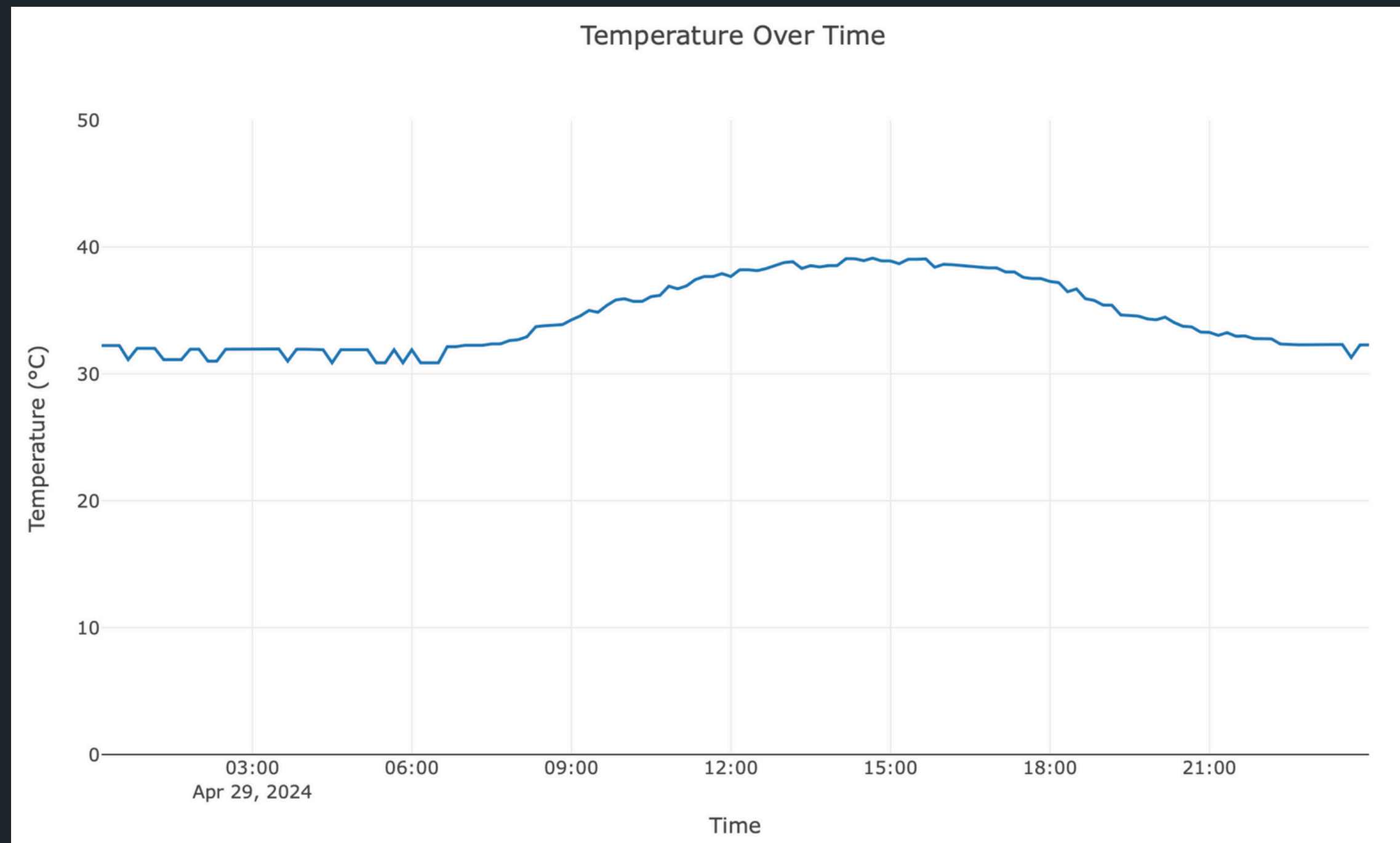
Returns the all traffic statistics for the specified travelID.

```
[
  {
    "DeviceID": "6FBF00F7-0101-472D-86EA-42D2B47459F0",
    "Latitude": 13.8423,
    "Longitude": 100.572,
    "Timestamp": "2024-05-01T16:22:43Z",
    "acceleration": 1.73205
  },
  {
    "DeviceID": "6FBF00F7-0101-472D-86EA-42D2B47459F0",
    "Latitude": 13.8421,
    "Longitude": 100.572,
    "Timestamp": "2024-05-01T16:23:03Z",
    "acceleration": 3.60555
  },
  {
    "DeviceID": "6FBF00F7-0101-472D-86EA-42D2B47459F0",
    "Latitude": 13.8412,
    "Longitude": 100.573,
    "Timestamp": "2024-05-01T16:23:23Z",
    "acceleration": 6.78233
  },
]
```

# Data Visualization

## 1. Line plot

Visualize the trend of data through different time

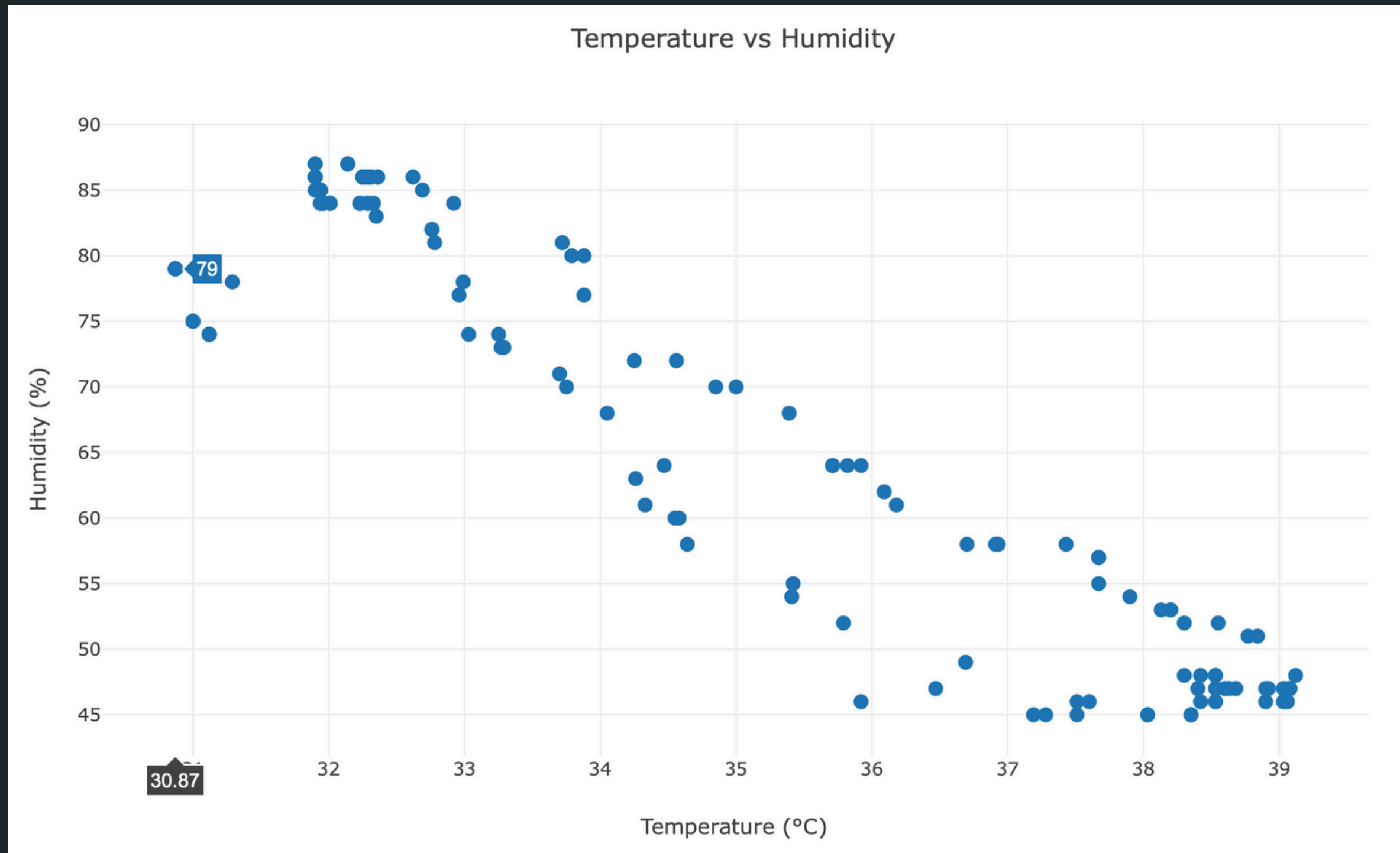




# Data Visualization

## 2. Scatter plot

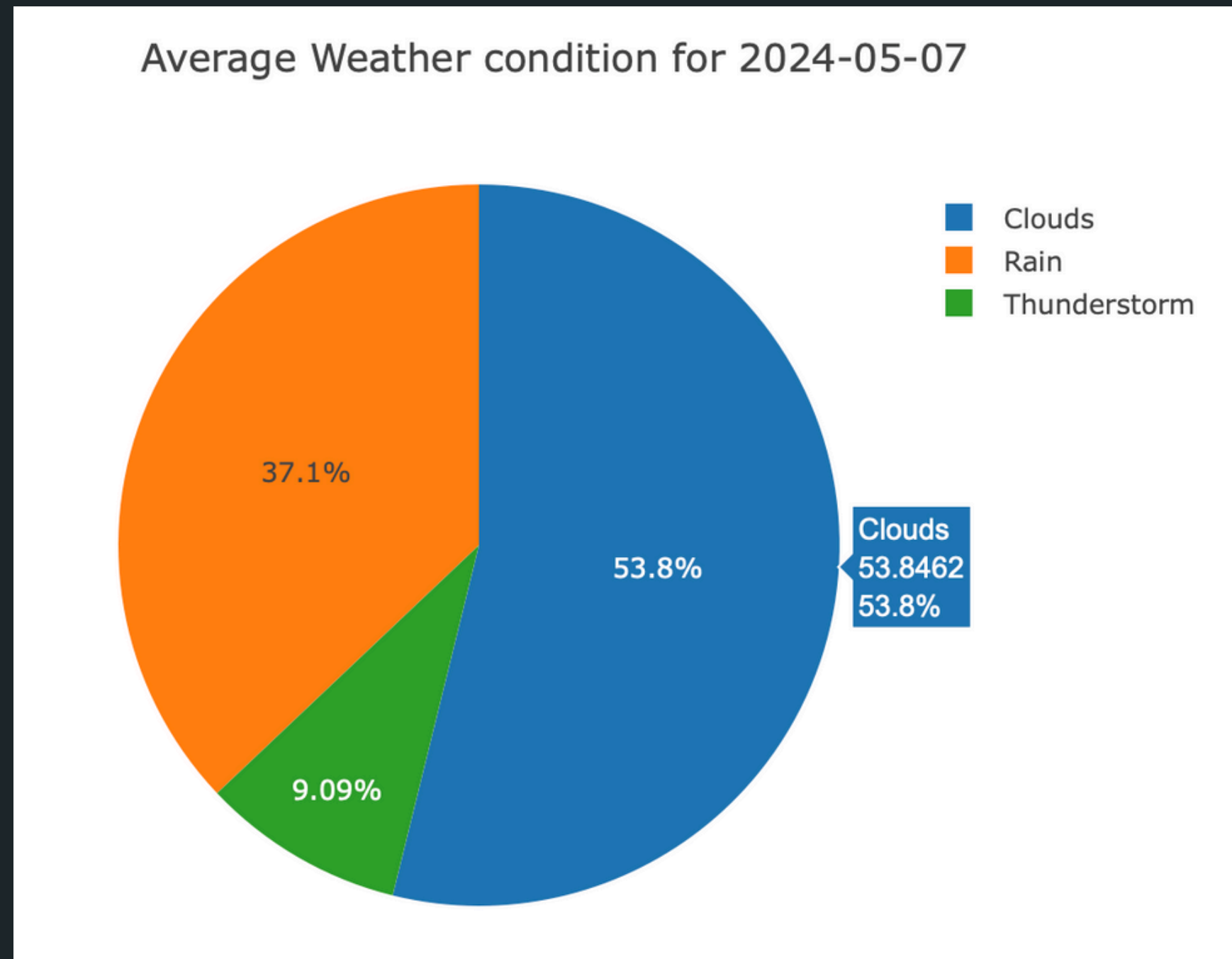
Visualize the relationship of 2 data



# Data Visualization

## 3. Pie plot

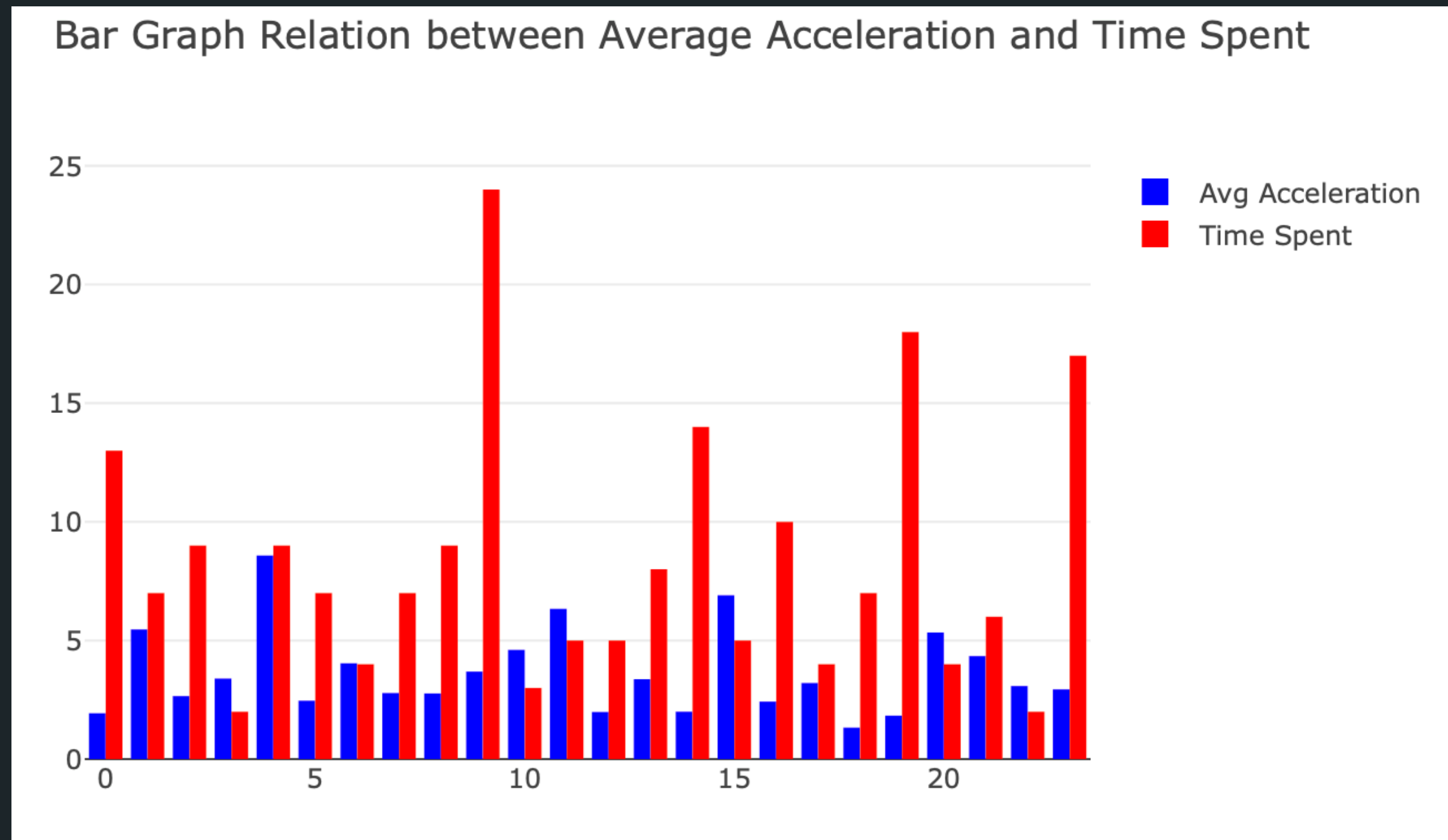
Visualize the ratio of the data



# Data Visualization

## 3. Bar plot

Visualize the relationship between two data







**DEMO**