**Mini project 2 - Estimating Market Penetration rate for Connected Vehicle Data in Iowa**

**Miguel Diaz**

**First, I defined the data format and field types to map this into Athena in AWS.**

1 - CV data Inspection and Analysis

| Variable | Description | Wejo-Specific Description | Field Type |
|---|---|---|---|
| **datapointid** | Unique identifier for the data point | ID associated with Wejo's connected vehicle data point | varchar(50) |
| **journeyid** | Unique identifier for the journey | ID representing a unique journey within Wejo's dataset | varchar(50) |
| **capturedtimestamp** | Timestamp when the data was captured | Time recorded for vehicle data collection in Wejo system | timestamp |
| **latitude** | Latitude coordinate of the data point | Vehicle's latitude recorded in Wejo data | double |
| **longitude** | Longitude coordinate of the data point | Vehicle's longitude recorded in Wejo data | double |
| **route_id** | Identifier for the route | Route identifier in Wejo's connected vehicle data | varchar(50) |
| **segment_start_measure** | Starting measure of the route segment | Measure indicating the starting point of the vehicle route segment | double |
| **hour** | Hour of the day | Hour extracted from the timestamp, useful for temporal aggregation | int |

2 - Wavetronix Sensor data and location data Inspection and Analysis

| Variable | Description | Field Type |
|---|---|---|
| **device_id** | Unique identifier for the Wavetronix sensor | varchar(50) |
| **lane_id** | Unique identifier for each lane monitored | varchar(50) |
| **lane_count** | Count of vehicles in the lane | int |
| **link_direction** | Direction of the monitored traffic link | varchar(50) |
| **parsed_cst_time** | Timestamp in Central Standard Time | timestamp |

| Variable | Description | Field Type |
|---|---|---|
| **device_id** | Unique identifier for the Wavetronix sensor | varchar(50) |
| **routeid** | Route identifier associated with the sensor | varchar(50) |
| **latitude** | Latitude coordinate of the sensor | float |
| **longitude** | Longitude coordinate of the sensor | float |

3 - Reading the data in AWS and reading data in Athena

Initial CV data format

| datapointid | journeyid | capturedtimestamp | latitude | longitude | route_id | segment_start_mea |
|---|---|---|---|---|---|---|
| "68b0984a-4845-471c-b0ba-a4c93fff707c" | "cedacf9ea3b75f26d11bcb36c421aff8905ed361" | "2023-05-02 14:06:19.000" | "41.630116" | "-93.759828" | "M787546530W" | "2.903" |

Reading process

```
CREATE EXTERNAL TABLE IF NOT EXISTS `miniproject_2_miguel`.`CV_Data` (
    `datapointid` varchar(50),
    `journeyid` varchar(50),
    `capturedtimestamp` string,
    `latitude` string,
    `longitude` string,
    `route_id` varchar(50),
    `segment_start_measure` string,
    `hour` string
)
COMMENT "Miguel miniproject 2 SQL AWS"
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim' = ',')
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://reactorlab/Miguel/CV_data/'
TBLPROPERTIES ('classification' = 'csv',
    'skip.header.line.count' = '1');
```

Data reformatting

```
CREATE TABLE "miniproject_2_miguel"."miguel_cv_data"
WITH (
    external_location = 's3://reactorlab/Miguel/miguel_cv_data/',
    format = 'PARQUET'
) AS
SELECT REPLACE(datapointid, '"', '') AS datapointid,
  REPLACE(journeyid, '"', '') AS journeyid,
  CAST(REPLACE(capturedtimestamp, '"', '') AS timestamp) AS capturedtimestamp,
  CAST(REPLACE(latitude, '"', '') AS double) AS latitude,
  CAST(REPLACE(longitude, '"', '') AS double) AS longitude,
  REPLACE(route_id, '"', '') AS route_id,
  CAST(
    REPLACE(segment_start_measure, '"', '') AS double
  ) AS segment_start_measure,
  CAST(REPLACE(hour, '"', '') AS int) AS hour
FROM (
    SELECT *
    FROM "miniproject_2_miguel"."cv_data"
    WHERE TRY_CAST(REPLACE(latitude, '"', '') AS double) IS NOT NULL
      AND TRY_CAST(REPLACE(longitude, '"', '') AS double) IS NOT NULL
      AND TRY_CAST(
        REPLACE(segment_start_measure, '"', '') AS double
      ) IS NOT NULL
      AND TRY_CAST(REPLACE(hour, '"', '') AS int) IS NOT NULL
  ) AS filtered_data;
```

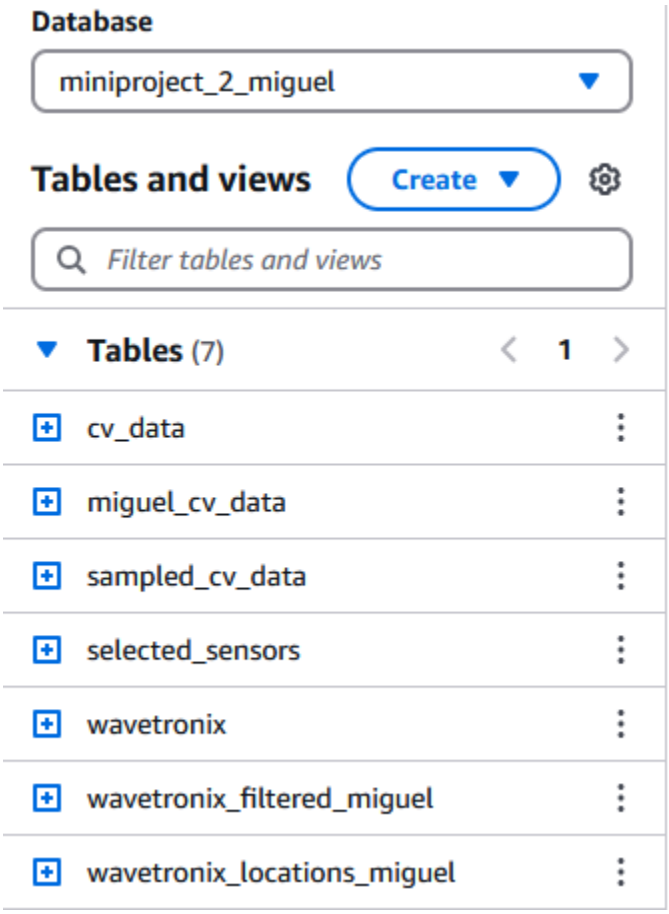
4 - Athena table data column types

Wavetronix location data

```
CREATE EXTERNAL TABLE IF NOT EXISTS `miniproject_2_miguel`.`Wavetronix_locations_Miguel` (
    `device_id` varchar(50),
    `routeid` varchar(50),
    `latitude` float,
    `longitude` float
) COMMENT "Miguel miniproject 2 SQL AWS"
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim' = ',')
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://reactorlab/Miguel/Wavetronix_locations/' --s3://reactorlab/Miguel/Wavetronix_locations/
TBLPROPERTIES ('classification' = 'csv',
    'skip.header.line.count' = '1');
```

Wavetronix sensors data

```
CREATE EXTERNAL TABLE IF NOT EXISTS `miniproject_2_miguel`.`Wavetronix_filtered_Miguel` (
    `device_id` varchar(50),
    `lane_id` varchar(50),
    `lane_count` int,
    `link_direction` varchar(50),
    `cst_time` string
)
COMMENT "Miguel miniproject 2 SQL AWS"
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim' = ',')
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://reactorlab/Miguel/Wavetronix_filtered_Miguel/'
TBLPROPERTIES ('classification' = 'csv',
    'skip.header.line.count' = '1');
```



| # | datapointid | journeyid | capturedtimestamp |
|---|---|---|---|
| 1 | 360438c9-93a6-4000-88d1-36fc0227f74e | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:44.000 |
| 2 | 0358cb37-82a8-491a-9b5a-278123332a4c | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:47.000 |
| 3 | ab32ae3c-7700-4c2b-9ecd-cbd733d60377 | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:50.000 |
| 4 | 83ac2da0-cfc1-41d4-9758-7f1ee2d06cce | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:53.000 |
| 5 | 389e2ece-b57e-44af-ab2b-4dc579d0aca0 | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:56.000 |
| 6 | 7cd47c90-22c8-47fe-a5f3-fc0dc15ead8a | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:46:59.000 |
| 7 | 907c9a7b-2cb1-4231-b297-928b06b5d778 | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:47:02.000 |
| 8 | 1dc8fdd7-6fc8-4de0-850c-22d33ef11cd7 | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:47:05.000 |
| 9 | 1063c909-4591-4057-b7f7-06910844d2fb | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:47:08.000 |
| 10 | cc260c87-a7c8-4fdf-8d3b-653cd1d78bbf | cbbe3fc4d1d7a7ffa0129947f8e0ce9b852e3557 | 2023-05-02 22:47:11.000 |

| | capturedtimestamp | latitude | longitude | route_id | segment_start_measure | hour |
|---|---|---|---|---|---|---|
| 2e3557 | 2023-05-02 22:46:44.000 | 41.67879 | -93.737347 | M382746870E | 0.21 | 22 |
| 2e3557 | 2023-05-02 22:46:47.000 | 41.678741 | -93.737745 | M382746870E | 0.192 | 22 |
| 2e3557 | 2023-05-02 22:46:50.000 | 41.67863 | -93.738123 | M382746870E | 0.162 | 22 |
| 2e3557 | 2023-05-02 22:46:53.000 | 41.678478 | -93.738504 | M382746870E | 0.063 | 22 |
| 2e3557 | 2023-05-02 22:46:56.000 | 41.67831 | -93.738896 | M382746870E | 0.063 | 22 |
| 2e3557 | 2023-05-02 22:46:59.000 | 41.678135 | -93.739303 | M382746870E | 0.063 | 22 |
| 2e3557 | 2023-05-02 22:47:02.000 | 41.677966 | -93.739709 | M382746870E | 0.063 | 22 |
| 2e3557 | 2023-05-02 22:47:05.000 | 41.677796 | -93.740111 | M382746870E | 0.05 | 22 |
| 2e3557 | 2023-05-02 22:47:08.000 | 41.677648 | -93.740508 | M382746870E | 0.025 | 22 |
| 2e3557 | 2023-05-02 22:47:11.000 | 41.677623 | -93.740832 | M382746870E | 0.011 | 22 |

Wavetronix sensors data parsed string to timestamp

```
CREATE TABLE "miniproject_2_miguel"."Wavetronix"
WITH (
    external_location = 's3://reactorlab/Miguel/Wavetronix/',
    format = 'PARQUET'
) AS
SELECT
    device_id,
    lane_id,
    lane_count,
    link_direction,
    date_parse(cst_time, '%Y-%m-%d-%H-%i-%s') AS parsed_cst_time
FROM "miniproject_2_miguel"."wavetronix_filtered_miguel";
```

5 - Identification of CVs within a 2-mile buffer of each sensor

- A WITH clause (cv_with_sensor) is used to join cv_data (CV data) and wavetronix_locations (sensor locations) based on the route_id.
- It calculates the approximate distance between CVs and sensors using the formula:

$$\text{distance} = \sqrt{(\text{ latitude difference })^2 + (\text{ longitude difference })^2}$$

```sql
-- Step 1: Identify CVs within 2-mile buffer of each sensor
WITH cv_with_sensor AS (
    SELECT
        cv.datapointid,
        cv.journeyid,
        cv.capturedtimestamp,
        cv.latitude AS cv_latitude,
        cv.longitude AS cv_longitude,
        cv.route_id,
        cv.hour,
        wl.device_id AS sensor_device_id,
        wl.latitude AS sensor_latitude,
        wl.longitude AS sensor_longitude,
        -- Approximation for buffer (2-mile ~ 0.03 latitude/longitude difference)
        sqrt(power(cv.latitude - wl.latitude, 2) + power(cv.longitude - wl.longitude, 2)) AS distance
    FROM
        raghu_cv_data AS cv
    JOIN
        raghu_wavetronix_locations AS wl
    ON
        cv.route_id = wl.route_id
),
```

6 - Aggregation of Wavetronix vehicles count by hour and sensor

**Grouping and Aggregation**

Data grouped by:

- device_id: The unique identifier of each Wavetronix sensor.
- hour: Extracted from the timestamp field (cst_time) in the data.

A SUM(lane_count) is performed to aggregate the vehicle counts across all lanes for each sensor-hour combination.

**Field Selection**

- device_id: The unique identifier of the sensor.
- hour: The time of data capture, derived from the timestamp.
- total_vehicle_count: The sum of vehicle counts across all lanes for the sensor-hour combination.

**Output**

(wavetronix_hourly_count) is generated with the following fields:

- device_id: The unique identifier of the sensor.
- hour: The hour of data capture.
- total_vehicle_count: The total number of vehicles detected across all lanes for the specified sensor and hour.

```
-- Step 3: Aggregate Wavetronix Counts by Hour and Sensor
wavetronix_hourly_count AS (
    SELECT
        device_id,
        hour(cst_time) AS hour,
        SUM(lane_count) AS total_vehicle_count
    FROM
        raghu_wavetronix_data
    GROUP BY
        device_id, hour(cst_time)
),
```

7 - Penetration Rate calculation

**Data Sources**

Combines the outputs from Step 2 (cv_hourly_count) and Step 3 (wavetronix_hourly_count).

- cv_hourly_count: Contains the count of unique CV journeys (cv_count) per sensor and hour.
- wavetronix_hourly_count: Contains the total number of vehicles (total_vehicle_count) per sensor and hour.

**Joining Tables**

- A join is performed between the two tables on:
- device_id: The unique identifier of the sensor.
- hour: The hour of data capture.
- This ensures that penetration rates are calculated for corresponding sensor-hour combinations.

**Penetration Rate Formula**

Penetration Rate (%):

- $\begin{cases} \left(\frac{\text{CV Count}}{\text{Total Vehicle Count}}\right) \times 100, & \text{if Total Vehicle Count } > 0 \\ 0, & \text{if Total Vehicle Count } = 0 \end{cases}$

**Output**

(penetration_rate) is generated with the following fields:

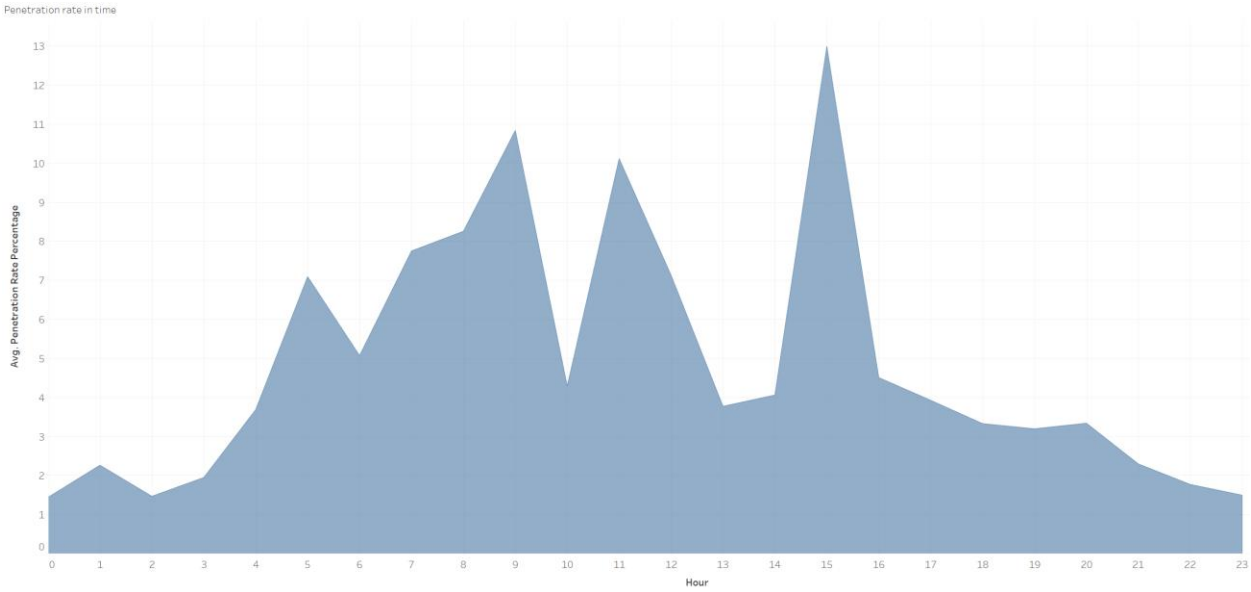
- **device_id**: The unique identifier of the sensor.
- **hour**: The hour of data capture.
- **cv_count**: The total number of CVs detected.
- **total_vehicle_count**: The total number of vehicles detected by the sensor.
- **penetration_rate_percentage**: The penetration rate of CVs as a percentage of total vehicles.

```
-- Step 4: Calculate Penetration Rate
penetration_rate AS (
    SELECT
        cv.device_id,
        cv.hour,
        cv.cv_count,
        wt.total_vehicle_count,
        -- Penetration Rate Calculation
        IF(wt.total_vehicle_count > 0,
           (CAST(cv.cv_count AS DECIMAL(10, 4)) / CAST(wt.total_vehicle_count AS DECIMAL(10, 4))) * 100,
           0) AS penetration_rate_percentage
    FROM
        cv_hourly_count AS cv
    JOIN
        wavetronix_hourly_count AS wt
    ON
        cv.device_id = wt.device_id
        AND cv.hour = wt.hour
)
```

```
-- Step 5: Output Results
SELECT
    device_id,
    hour,
    cv_count,
    total_vehicle_count,
    penetration_rate_percentage
FROM
    penetration_rate
ORDER BY
    device_id, hour;
```

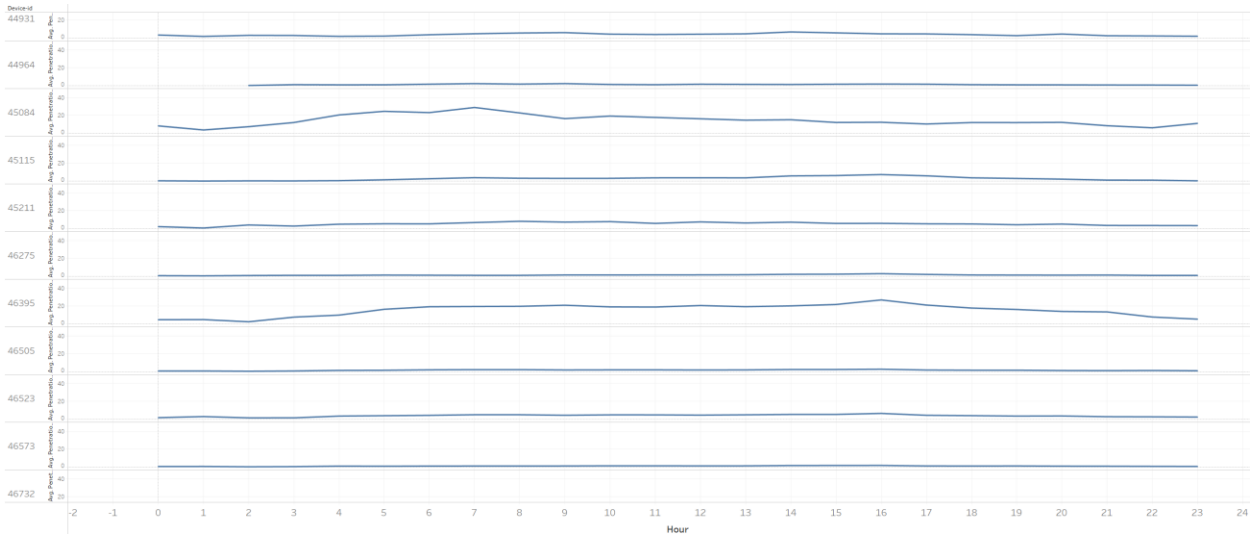
8 – Results

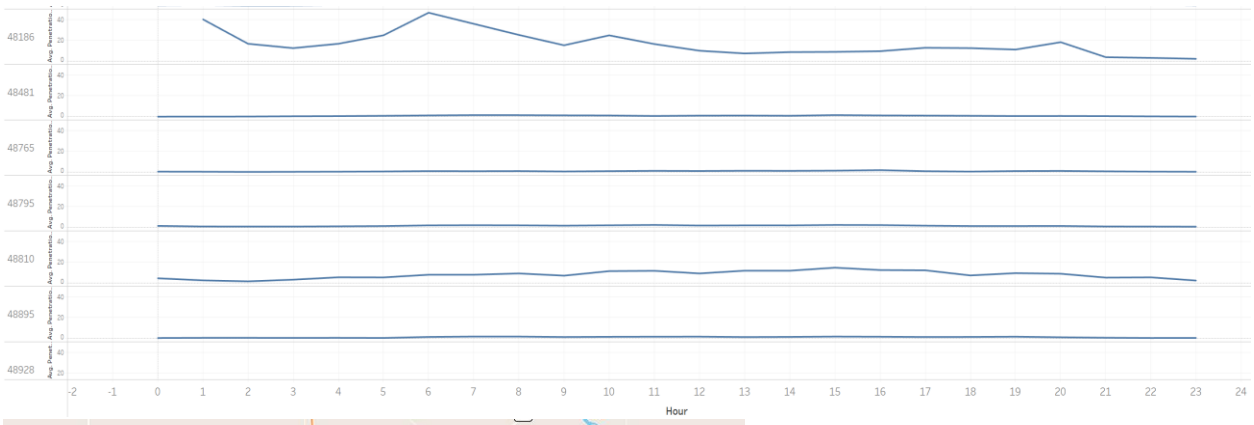
**Average penetration rates patterns in time**




- The highest peak for the penetration rate is presented at around 15:00 hours.
- The highest counted portion is presented during the morning hours

**Average penetration rates patterns in time by device ID**

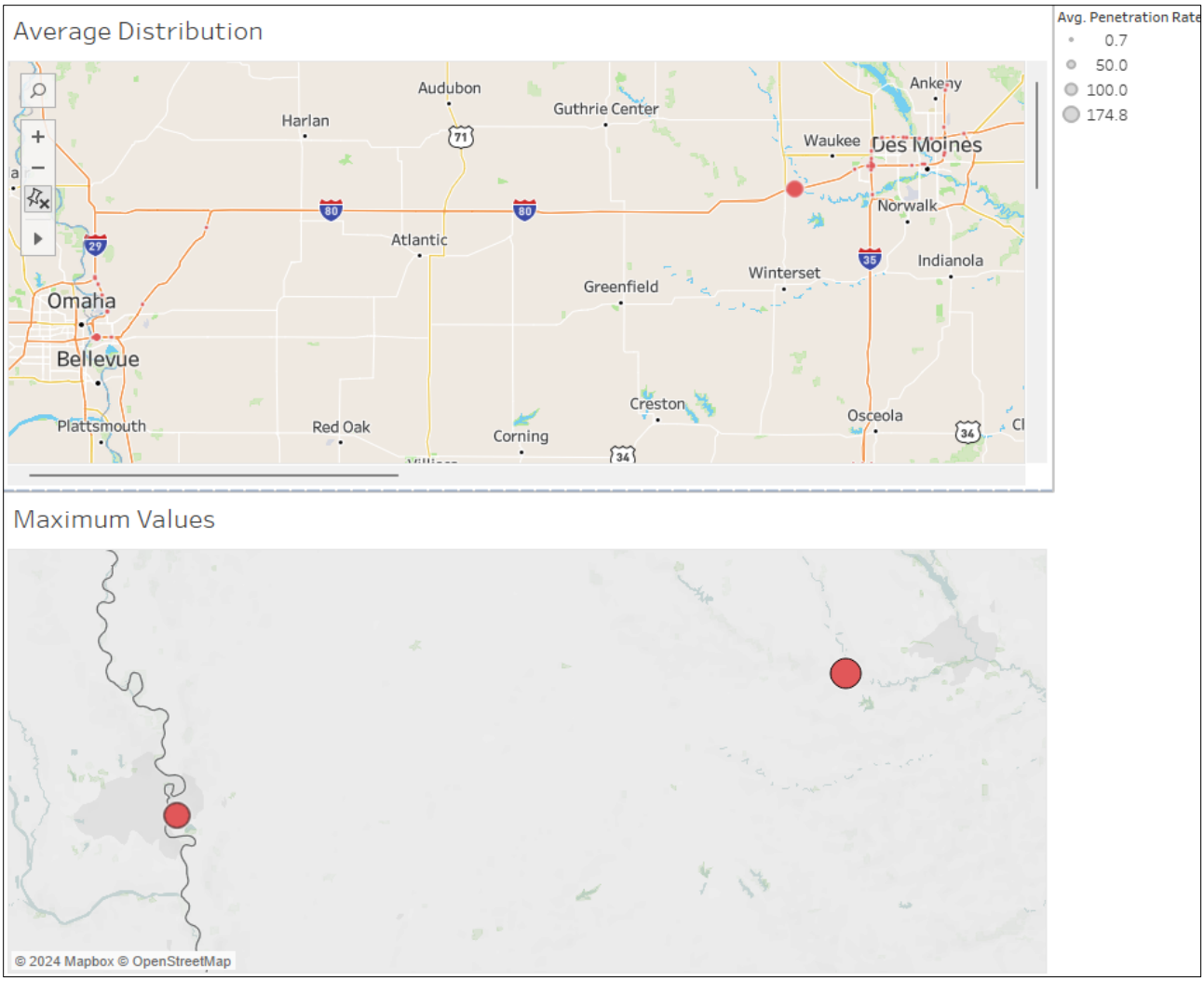






**Locations with the highest penetration rates**

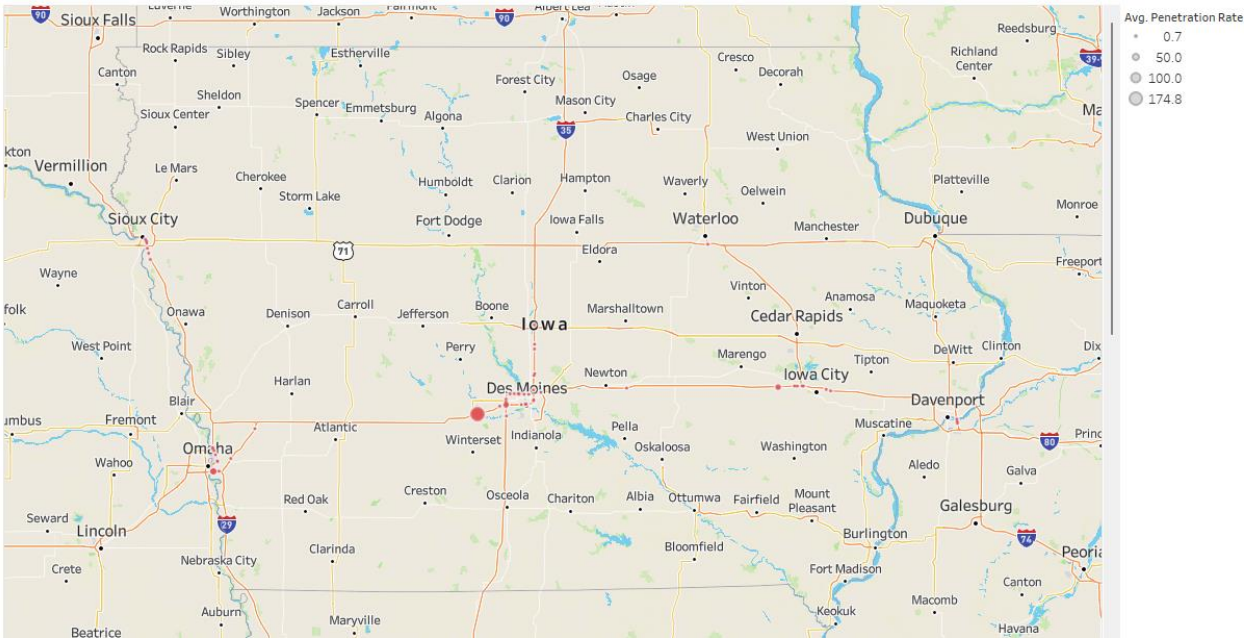
- The highest penetration rates appear to be concentrated in areas around Des Moines, Iowa.
- The second location with high penetration was identified around Omaha limits with Iowa.




**Sensors with average penetration rate at each location**

- The highest penetration rates appear to be concentrated in areas around Des Moines, Iowa.
- The smallest penetration rates are int the Sioux City region.

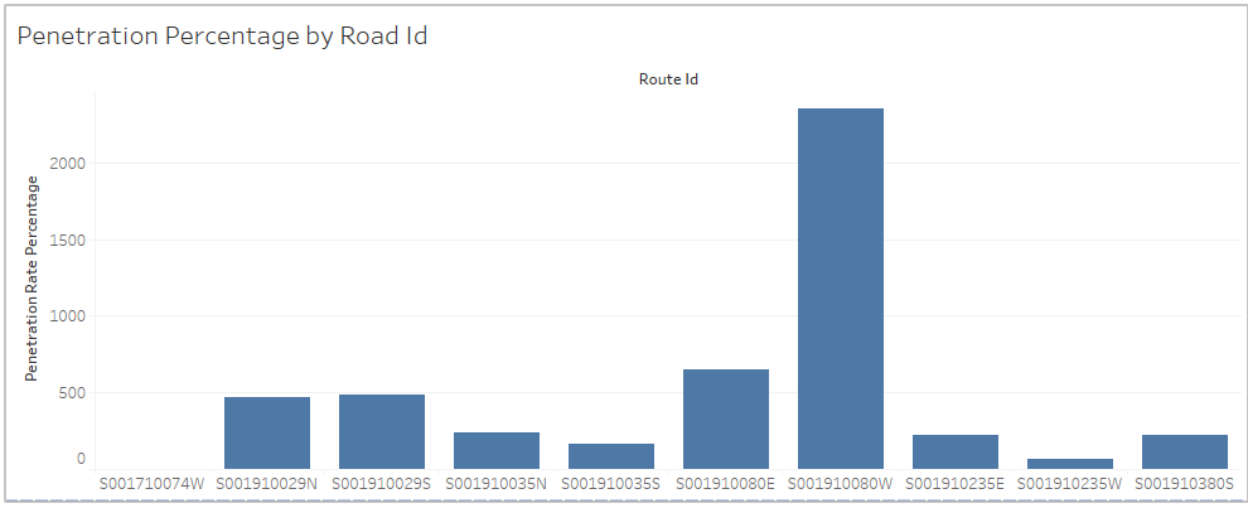
Locations of Average Penetration rate percentage

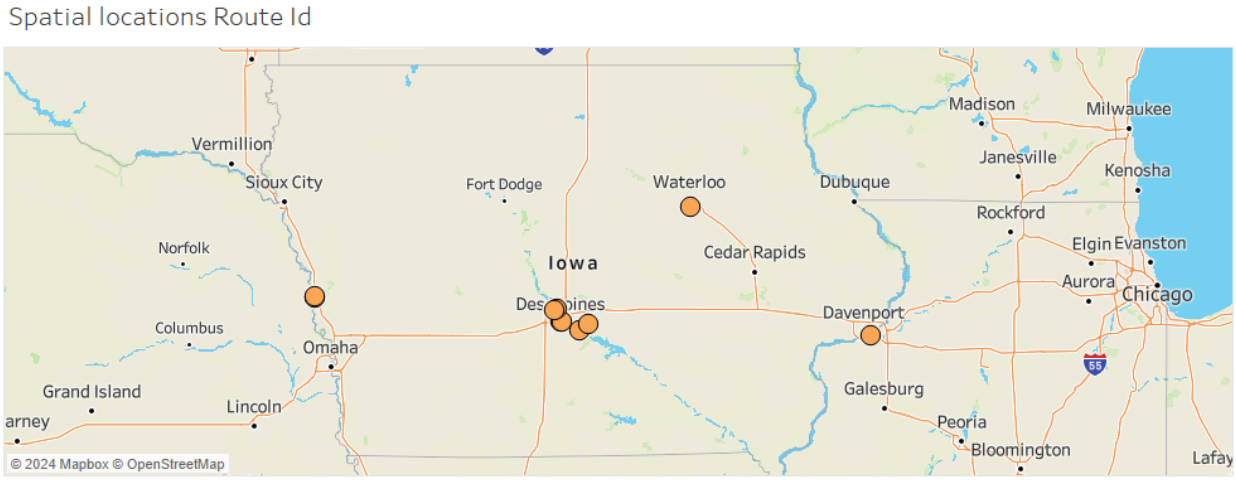



## Routes spatial penetration distribution

Spatial locations Route Id


Spatial locations Route Id



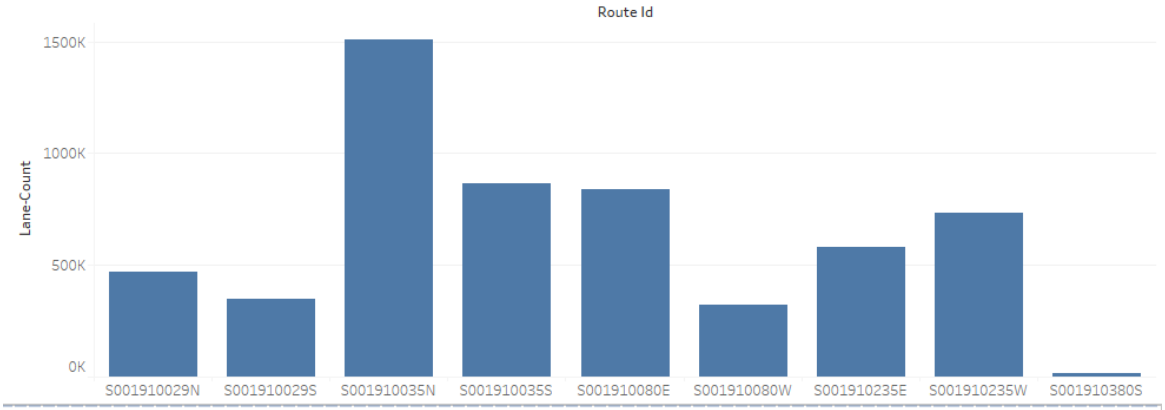




- Highest penetration rates along routes located around Des Moines city

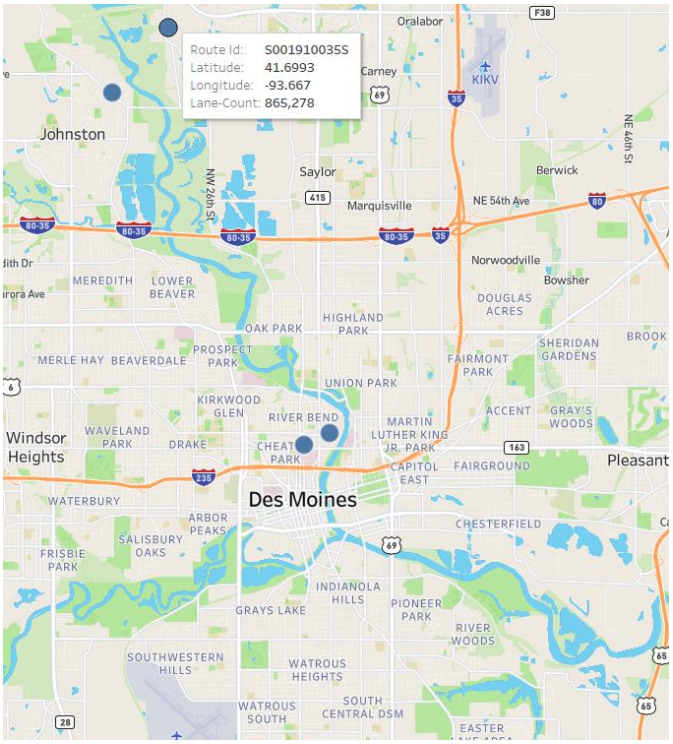
**Lane count spatial distribution**

- Highest count in lines located around Des Moines city

## Lane count by Route ID




## Lane count spartial distribution by route ID

**9 – Conclusion**

1. The penetration rate shows significant peaks around specific hours of the day, particularly around 8-9 AM and 3-4 PM, as indicated in the penetration rate time series.
2. These peaks likely correspond to morning and afternoon traffic rush hours, indicating a higher proportion of connected vehicles.
3. The highest penetration rates are concentrated in the Des Moines metropolitan area, as shown on the spatial map and supported by the bar chart of penetration percentage by Route ID.
4. Des Moines is a significant hub for connected vehicle traffic, suggesting advanced adoption or implementation of connected vehicle technology in this area.
5. Specific routes, such as S001910008W, show significantly higher penetration rates than others. These routes may be critical corridors for connected vehicle operations, possibly due to traffic density, infrastructure, or technology adoption along these paths.
6. Route S001910035N has the highest lane count among the routes, indicating it is a major regional arterial road. Spatial distribution maps align this route with high-traffic volume areas near Des Moines. Higher lane counts correlate with increased vehicle detection, providing robust traffic and penetration rate analysis data.
7. Regions like Cedar Rapids, Iowa City, and Dubuque have moderate penetration rates compared to Des Moines, with penetration declining as we move toward rural areas.
8. Urban areas show greater adoption of connected vehicle technology than rural areas, likely due to better infrastructure and higher vehicle density.
9. Higher adoption in metropolitan areas like Des Moines suggests these locations could serve as testing grounds for future connected vehicle initiatives.