NYC Transportation Pattern Analysis

RBDA Analytics Symposium - Fall 2023

Team Members:
Guanshi Wang (gw2310) | Haowei Tu (ht2397) | Wenbo Bao (wb2128) | Yueyan Lu (yl6211)

12.13.23

Abstract

In this project, we conduct a comprehensive analysis of NYC mobility patterns using data from taxis, for-hire vehicles (Uber/Lyft), city bikes, and real-time traffic speed records. The analysis starts from understanding how traffic speeds fluctuate throughout the day to identify peak and off-peak periods. In addition, we also aim to uncover insights into user behavior and preferences in choosing transportation modes, as well as to explore the correlation between traffic velocity and the demand for ride-hailing services. By drawing connections between these variables, our goal is to provide efficient recommendations for individuals in selecting their preferred transportation method, ensuring personalized and optimal urban mobility solutions.

Platform(s) where the application runs: Google Dataproc

Motivation

User-centric Solution: By analyzing traffic patterns, we gain valuable insights into user behavior and preferences, which is instrumental in avoiding congested routes, reducing stress, and tailoring efficient transportation solutions aligned with the needs of diverse populations.

Operational Efficiency for Companies: Companies can optimize their logistics and delivery routes by understanding traffic flow patterns. This leads to reduced travel time and lower fuel consumption, enhancing operational efficiency.

Efficient Urban Planning: Understanding of traffic patterns throughout the day aides city planners in optimizing infrastructure, road network, and public transportation systems.

Goodness

Data uniformity: The data analysis is performed on data with consistent geometric and temporal dimensions, ensuring uniformity access the analytical process.
Multi-Source Data Cross-referencing: The analytic is derived through referencing and cross-comparing data from diverse sources such as traffic sensors, private and public transport records records, enhancing the depth and reliability of the findings.
User Feedback Incorporation: The results of the analysis are compared with individual observations and experiences of traffic. This step ensures that the data aligns with what people commonly experience on the roads, enhancing the reliability of the analytic.
Integration with External Reports: The analytic also incorporates with external reports such as news articles and urban planning documents. This layer of validation adds an additional dimension to the analysis, aligning data-driven insights with real-world changes and expert perspectives in urban development.

PART 01

Data Cleaning & Profiling

Data Sources

Name: Taxi & Limousine Commission Trip Record Data
Description: NYC Yellow taxi trip records with trip information such as pickup/dropoff datetime, location and trip fare stored in parquet format
Size of data: 1.2 GB

Name: NYC Real-Time Traffic Speed Data (Collected on Main Roads)
Description: NYC traffic speed on main roads, specifically focusing on columns for LINK_POINTS (locations of the sensor links), DATA_AS_OF and SPEED
Size of data: 27 GB

Name: For-Hire Vehicle Trip Records
Description: This data dictionary describes FHV trip data. Each row represents a single trip in an FHV.
Size of data: 5.38 GB

Name: NYC Citi Bike Trip Data
Description: NYC Citi Bike trip records with trip information such as trip start/end datetime, station latitude and longitude stored in csv format
Size of data: 1.1 GB

Data Sample 1: TLC Trip Record Data

Sample taxi trip record (parquet format)

VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	airport_fee
1	2021-01-01 00:30:10	2021-01-01 00:36:12	1.0	2.1	1.0	N	142	43	2	8.0	3.0	0.5	0.0	0.0	0.3	11.8	2.5	null
1	2021-01-01 00:51:20	2021-01-01 00:52:19	1.0	0.2	1.0	N	238	151	2	3.0	0.5	0.5	0.0	0.0	0.3	4.3	0.0	null
1	2021-01-01 00:43:30	2021-01-01 01:11:06	1.0	14.7	1.0	N	132	165	1	42.0	0.5	0.5	8.65	0.0	0.3	51.95	0.0	null
1	2021-01-01 00:15:48	2021-01-01 00:31:01	0.0	10.6	1.0	N	138	132	1	29.0	0.5	0.5	6.05	0.0	0.3	36.35	0.0	null
2	2021-01-01 00:31:49	2021-01-01 00:48:21	1.0	4.94	1.0	N	68	33	1	16.5	0.5	0.5	4.06	0.0	0.3	24.36	2.5	null
1	2021-01-01 00:16:29	2021-01-01 00:24:30	1.0	1.6	1.0	N	224	68	1	8.0	3.0	0.5	2.35	0.0	0.3	14.15	2.5	null
1	2021-01-01 00:00:28	2021-01-01 00:17:28	1.0	4.1	1.0	N	95	157	2	16.0	0.5	0.5	0.0	0.0	0.3	17.3	0.0	null
1	2021-01-01 00:12:29	2021-01-01 00:30:34	1.0	5.7	1.0	N	90	40	2	18.0	3.0	0.5	0.0	0.0	0.3	21.8	2.5	null
1	2021-01-01 00:39:16	2021-01-01 01:00:13	1.0	9.1	1.0	N	97	129	4	27.5	0.5	0.5	0.0	0.0	0.3	28.8	0.0	null
1	2021-01-01 00:26:12	2021-01-01 00:39:46	2.0	2.7	1.0	N	263	142	1	12.0	3.0	0.5	3.15	0.0	0.3	18.95	2.5	null
2	2021-01-01 00:15:52	2021-01-01 00:38:07	3.0	6.11	1.0	N	166	255	1	20.5	0.5	0.5	0.0	0.0	0.3	24.3	2.5	null
2	2021-01-01 00:46:36	2021-01-01 00:53:45	2.0	1.21	1.0	N	255	80	1	7.0	0.5	0.5	2.49	0.0	0.3	10.79	0.0	null
1	2021-01-01 00:10:46	2021-01-01 00:32:58	2.0	7.4	1.0	N	138	166	2	24.5	2.5	0.5	0.0	6.12	0.3	33.92	0.0	null
2	2021-01-01 00:31:06	2021-01-01 00:38:52	5.0	1.7	1.0	N	142	50	1	8.0	0.5	0.5	2.36	0.0	0.3	14.16	2.5	null
2	2021-01-01 00:42:11	2021-01-01 00:44:24	5.0	0.81	1.0	N	50	142	2	4.5	0.5	0.5	0.0	0.0	0.3	8.3	2.5	null
2	2021-01-01 00:17:48	2021-01-01 00:21:55	1.0	1.01	1.0	N	236	237	1	5.5	0.5	0.5	1.0	0.0	0.3	10.3	2.5	null
2	2021-01-01 00:33:38	2021-01-01 00:38:37	1.0	0.73	1.0	N	142	239	1	5.5	0.5	0.5	2.79	0.0	0.3	12.09	2.5	null
2	2021-01-01 00:47:56	2021-01-01 00:52:53	1.0	1.17	1.0	N	238	166	1	6.5	0.5	0.5	2.06	0.0	0.3	12.36	2.5	null
2	2021-01-01 00:04:21	2021-01-01 00:07:58	1.0	0.78	1.0	N	239	238	1	4.5	0.5	0.5	1.66	0.0	0.3	9.96	2.5	null
2	2021-01-01 00:18:36	2021-01-01 00:27:10	2.0	1.66	1.0	N	151	142	2	8.5	0.5	0.5	0.0	0.0	0.3	12.3	2.5	null

Data Schema

```
root
 |-- VendorID: long (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- tpep_dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: double (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: double (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: long (nullable = true)
 |-- DOLocationID: long (nullable = true)
 |-- payment_type: long (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
 |-- airport_fee: double (nullable = true)
```

NYU

PART 01

Data Sample 2: NYC Real-Time Traffic Speed Data

Original Speed Data in csv format

```
gw2310_nyu_edu@nyu-dataproc-m:~/rbda_proj$ du -h DOT_Traffic_Speeds_NBE.csv
27G     DOT_Traffic_Speeds_NBE.csv
gw2310_nyu_edu@nyu-dataproc-m:~/rbda_proj$ cat DOT_Traffic_Speeds_NBE.csv | head -n 5
ID,SPEED,TRAVEL_TIME,STATUS,DATA_AS_OF,LINK_ID,LINK_POINTS,ENCODED_POLY_LINE,ENCODED_POLY_LINE_LVLS,OWNER,TRANSCOM_ID,BOROUGH,LINK_NAME
262,34.8,359,0,06/02/2017 11:41:59 PM,4616319,"40.6332305,-74.016151 40.63391,-74.01613 40.6343505,-74.016241 40.63485,-74.016501 40.63533,-74.01694 40.63622,-74.01827 40.6375704,-74.02046 40.
63803,-74.021071 40.6385,-74.02146 40.63908,-74.02173 40.63967,-74.02185 40.64036,-74.02172 40.64103,-74.02143",ud_wF|gwbMgCCwATcBr@_BvAqDhGmGtL{AxB}AlAsBt@uBViCYeCy@wBqAkCeCeOcP_WgXm~@aaAmb@y
c@cBuAwCqAmCUoDPuFdBsX`OwFlCaFzC{BvA,BBBBBBBBBBBBBBBBBBBBBBBBBB,NYC_DOT_LIC,4616319,Brooklyn,GOW S 9TH STREET - 7TH AVENUE
204,55.92,155,0,06/02/2017 11:41:59 PM,4616320,"40.7894406,-73.786291  40.78918,-73.78792  40.78876,-73.789521  40.7882704,-73.790631  40.7872906,-73.79263  40.7866804,-73.7941  40.7864,-73.79
4901  40.7861204,-73.796241  40.78602,-73.79693  40.78596,-73.79837  40.78615,-73.80182  40.786331,-73.80525  4",_u}wFhkjaMr@dI~A~HtA|EbEnKxBdHv@~Cv@jGRhCJ~G[pTm@lT}@nS[vB@p@]dHiAlPo@tEiA|Fu@n
CoDvKiAhFa@`DSxCAbAFfAb@lB`AbBjBjA??xHfBpOnBzJ`A|Cn@pEfBz[|O|HzEtBdBrEjFjEjHhAzB,BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB,NYC_DOT_LIC,4616320,Queens,CIP N TNB - Whitestone Expwy S Exit 14 (Li
nden Pl)
106,39.77,159,0,06/02/2017 11:41:59 PM,4616323,"40.77158,-73.994441 40.7713004,-73.99455 40.77085,-73.99467 40.76997,-73.99481 40.7701604,-73.99477 40.76986,-73.994831 40.7695406,-73.99496 40.
769341,-73.99508 40.768311,-73.9958 40.768311,-73.9958 40.76623,-73.99733 40.76623,-73.99733 40.76547,-73.9979",kezwFf`sbMv@TxAVnDZe@Gz@J~@Xf@VlEnC??~KpH??vCpB??zEbD??dC`B,BBBBBBBBBBBBBBBBB,NY
C_DOT_LIC,4616323,Manhattan,12th Ave S 57th St - 45th St
184,65.24,39,0,06/03/2017 04:46:59 AM,4616253,"40.8347204,-73.86593 40.83357,-73.86199 40.8327305,-73.8592 40.8312,-73.85409 40.83017,-73.85069 40.83015,-73.85071 40.8301304,-73.850731 40.8301
105,-73.850731",_pfxF`}yaMdFsWfDmPpH}^lEgTBBBBB,BBBBB,NYC_DOT_LIC,4616253,Bronx,CBE E TAYLOR AVENUE - CASTLE HILL AVENUE
```

Original Taxi Zone Data

NYC Taxi Zones

Based on NYC Taxi Zones

This map shows the NYC Taxi Zones, which correspond to the pickup and drop-off zones, or LocationIDs, included

More Views | Export | Discuss | Embed | About

OBJECTID	Shape_Leng	the_geom	Shape_Area	zone	LocationID	borough
1	0.116357453189	MULTIPOLYGON (((-74.184452999...	0.0007823067885	Newark Airport	1	EWR
2	0.43346966679	MULTIPOLYGON (((-73.823375972...	0.0048634037837	Jamaica Bay	2	Queens
3	0.0843411059012	MULTIPOLYGON (((-73.847926140...	0.000314414156821	Allerton/Pelham Gardens	3	Bronx
4	0.0435665270921	MULTIPOLYGON (((-73.971774109...	0.000111871946192	Alphabet City	4	Manhattan

Processed Result in FileOutputFormat

```
gw2310_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat rbda_proj/filter2/part-m-00000 | head -n 5
43.49,01/27/2021 06:33:03 AM,235
49.08,01/27/2021 06:33:03 AM,239
43.49,01/27/2021 06:33:03 AM,6
0.00,01/27/2021 06:33:03 AM,6
33.55,01/27/2021 06:33:04 AM,74
```

NYU

Data Sample 3

2015-2022 For-Hire Vehicle(FHV) Data - Parquet Format

Field Name	Description
Dispatching_base_num	The TLC Base License Number of the base that dispatched the trip
Pickup_datetime	The date and time of the trip pick-up
DropOff_datetime	The date and time of the trip dropoff
PULocationID	TLC Taxi Zone in which the trip began
DOLocationID	TLC Taxi Zone in which the trip ended
SR_Flag	Indicates if the trip was a part of a shared ride chain offered by a High Volume FHV company (e.g. Uber Pool, Lyft Line). For shared trips, the value is **1**. For non-shared rides, this field is **null**.

Process: output parquet files only retain date/time and zone columns, filter out entries with null value

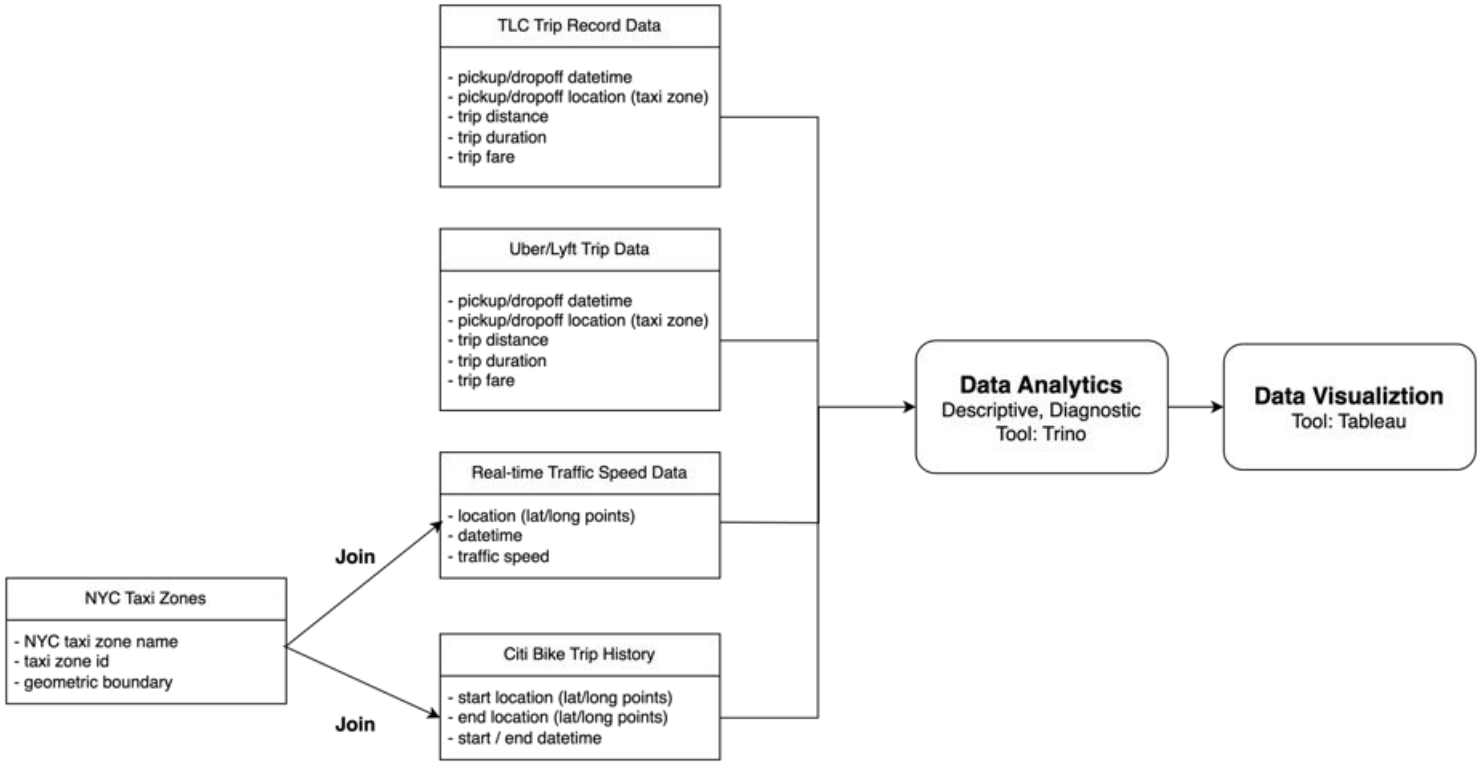
PART 01

Data Sample 4: NYC Citi Bike Trip Data

Original Speed Data in csv format

tripduration	starttime	stoptime	start station id	start station name	start station latitude	start station longitude	end station id	end station name	end station latitude	end station longitude	bikeid	usertype	birth year
789	2020-01-01 00:00:55.390	2020-01-01 00:14:05.147	504	1 Ave & E 16 St	40.732219	-73.981656	307	Canal St & Rutgers St	40.714275	-73.989900	30326	Subscriber	1992
1541	2020-01-01 00:01:08.102	2020-01-01 00:26:49.178	3423	West Drive & Prospect Park West	40.661063	-73.979453	3300	Prospect Park West & 8 St	40.665147	-73.976376	17105	Customer	1969
1464	2020-01-01 00:01:42.140	2020-01-01 00:26:07.011	3687	E 33 St & 1 Ave	40.743227	-73.974498	259	South St & Whitehall St	40.701221	-74.012342	40177	Subscriber	1963
592	2020-01-01 00:01:45.561	2020-01-01 00:11:38.155	346	Bank St & Hudson St	40.736529	-74.006180	490	8 Ave & W 33 St	40.751551	-73.993934	27690	Subscriber	1980
702	2020-01-01 00:01:45.788	2020-01-01 00:13:28.240	372	Franklin Ave & Myrtle Ave	40.694546	-73.958014	3637	Fulton St & Waverly Ave	40.683239	-73.965996	32583	Subscriber	1982

Design Diagram



Code Challenge 1

Inconsistent parquet schema across files

- Different capitalization of the header record
- Inconsistency in data types—mixing of Int, Long and Double
- Define a new output schema, set default value for fields that has null value

```
// Define the output parquet file schema
String writeSchema = "message new_schema { " +
        "required int64 vendor_id; " +
        "required int64 pickup_datetime (TIMESTAMP(MICROS,false)); " +
        "required int64 dropoff_datetime (TIMESTAMP(MICROS,false)); " +
        "required int64 passenger_count; " +
        "required double trip_distance; " +
        "required int64 pickup_locationId; " +
        "required int64 dropoff_locationId; " +
        "required int64 payment_type; " +
        "required double fare_amount; " +
        "required double extra; " +
        "required double mta_tax; " +
        "required double tip_amount; " +
        "required double tolls_amount; " +
        "required double improvement_surcharge; " +
        "required double total_amount; " +
        "required double congestion_surcharge; " +
        "required double airport_fee; " +
        "}";
MessageType schema = MessageTypeParser.parseMessageType(writeSchema);
LOG.info(schema);
GroupWriteSupport.setSchema(schema, getConf());
```

```
if (value.getFieldRepetitionCount(field:"VendorID") > 0) {
    if (inputType.getType(fieldName:"VendorID").isPrimitive() &&
            inputType.getType(fieldName:"VendorID").asPrimitiveType().getPrimitiveTypeName() == PrimitiveType.PrimitiveTypeName.INT32) {
        vendor_id = (long) value.getInteger(field:"VendorID", index:0);
    } else {
        vendor_id = value.getLong(field:"VendorID", index:0);
    }
}
```

```
if (inputType.containsField(name:"airport_fee") && value.getFieldRepetitionCount(field:"airport_fee") > 0) {
    airport_fee = value.getDouble(field:"airport_fee", index:0);
} else if (inputType.containsField(name:"Airport_fee") && value.getFieldRepetitionCount(field:"Airport_fee") > 0) {
    airport_fee = value.getDouble(field:"Airport_fee", index:0);
}
```

Code Challenge 2

Maven build with 3rd party library

- Leverage Apache parquet library to parse the parquet files in the MapReduce job
- Instead of supplying all the dependent jars in hadoop job submission, I updated the pom.xml to build a fat jar.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>3.2.4</version> <!-- Use the latest version available -->
            <executions>
                <execution>
                    <phase>package</phase>
                    <goals>
                        <goal>shade</goal>
                    </goals>
                    <configuration>
                        <createDependencyReducedPom>false</createDependencyReducedPom>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

PART 01

Code Challenge 3

- Calculate Stdev of Double Using Welford's online algorithm

✘ $\mathrm{Var}(X) = E[X^2] - (E[X])^2$

```
@Override
public void reduce(NullWritable key, Iterable<Text> values, Context context)
        throws IOException, InterruptedException {
    double max = Double.MIN_VALUE;
    double min = Double.MAX_VALUE;
    double count = 0;
    double mean = 0;
    double M2 = 0;

    for (Text val : values) {
        double value = Double.parseDouble(val.toString());
        max = Math.max(max, value);
        min = Math.min(min, value);
        count++;
        double delta = value - mean;
        mean += delta / count;
        double delta2 = value - mean;
        M2 += delta * delta2;
    }

    double variance = count > 1 ? M2 / count : 0;
    double stdev = Math.sqrt(variance);

    String output = "Max: " + max + ", Min: " + min + ", Count: " + count + ", Mean: " + mean + ", Stdev: " + stdev;
    context.write(NullWritable.get(), new Text(output));
}
```

NYU

PART 01

Code Challenge 4

- Convert lat/lon points to taxi zones using a replicated join

```
private STRtree strTree;
private GeometryFactory geometryFactory;
private WKTReader reader;
private HashMap<MultiPolygon, Integer> polygonToId;

@Override
public void setup(Context context) throws IOException, InterruptedException {
    strTree = new STRtree();
    geometryFactory = new GeometryFactory();
    reader = new WKTReader(geometryFactory);
    polygonToId = new HashMap<MultiPolygon, Integer>();

    URI[] files = context.getCacheFiles();

    for (URI file : files) {
        if (!file.toString().endsWith(".txt")) {
            continue;
        }
        Path path = new Path(file.getPath());
        BufferedReader bf = new BufferedReader(new FileReader(path.getName()));
        String line = null;
        while ((line = bf.readLine()) != null) {
            String[] parts = line.split("@");
            String wktPolygon = parts[0];
            int id = Integer.parseInt(parts[1]);
            MultiPolygon polygon;
            try {
                polygon = (MultiPolygon) reader.read(wktPolygon);
            } catch (org.locationtech.jts.io.ParseException e) {
                throw new RuntimeException("Parsing polygon error, stop!", e);
            }
            strTree.insert(polygon.getEnvelopeInternal(), polygon);
            polygonToId.put(polygon, id);
        }
    }
}
```

NYU

PART 01

Code Challenge 4

- Convert lat/lon points to taxi zones using a replicated join

```
@Override
public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
    String[] parts = value.toString().split(",");
    double lat = Double.parseDouble(parts[2]);
    double lon = Double.parseDouble(parts[3]);
    Point point = geometryFactory.createPoint(new Coordinate(lon, lat));
    List<MultiPolygon> queryResults = strTree.query(point.getEnvelopeInternal());
    for (MultiPolygon polygon : queryResults) {
        if (polygon.contains(point)) {
            StringBuilder sb = new StringBuilder();
            sb.append(parts[0]);
            sb.append(",");
            sb.append(parts[1]);
            sb.append(",");
            sb.append(Integer.toString(polygonToId.get(polygon)));
            context.write(NullWritable.get(), new Text(sb.toString()));
            break;
        }
    }
}

//@Override
//public void cleanup(Context context) throws IOException, InterruptedException {
//    context.write(NullWritable.get(), new Text(Integer.toString(polygonToId.size())));
//}
```

Use cleanup to print

NYU

Code Challenge 5

- Creating heatmap

```
import pandas as pd
from datetime import datetime
from folium.plugins import HeatMapWithTime

# create new base map for heat map with time
base_map_2 = generateBaseMap(nyc)

# referenced https://justinmorganwilliams.medium.com/how-to-make-a-time-lapse-heat-map-with-folium-using-nyc-bike-sh
# create a more meaningful index for heat map with time
start = datetime(2020,1,1,0)
end = datetime(2020,1,1,23)
daterange = pd.date_range(start=start,end=end, periods=24) #use pandas daterange function to generate date range obj

time_index = [d.strftime("%I:%M %p") for d in daterange] # format time with AM/PM

# instantiate HeatMapWithTime
HeatMapWithTime(df_hour_list,radius=11,
                index=time_index,
                gradient={0.1: 'blue', 0.5: 'lime', 0.7: 'orange', 1: 'red'},
                min_opacity=0.4,
                max_opacity=0.8,
                use_local_extrema=True)\
                .add_to(base_map_2)

# save as html
base_map_2.save('./heatmapwithtime_bikeshare.html')

# call result
base_map_2
```

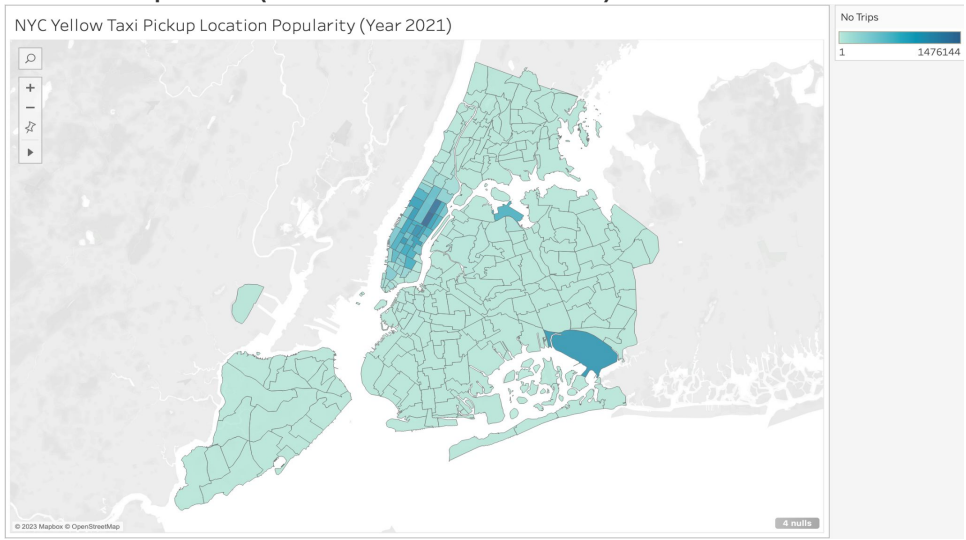
PART 02

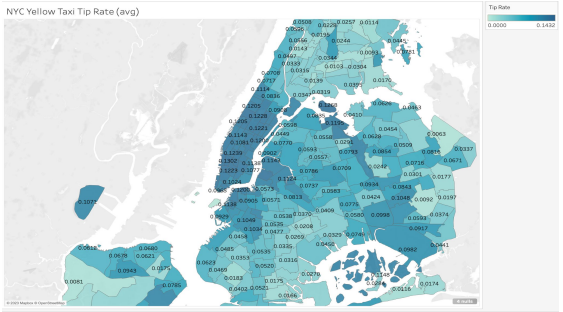
Data Analysis

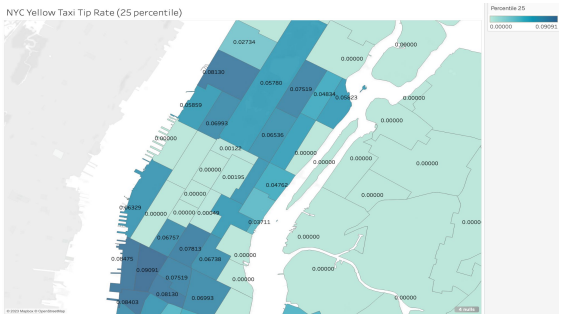
Q: What is the geographic pattern for using Taxi?

Pickup Location Popularity (year around)

- Mainly concentrated on Manhattan (Upper East / West Side especially)
- Airport (LaGuardia / JFK)



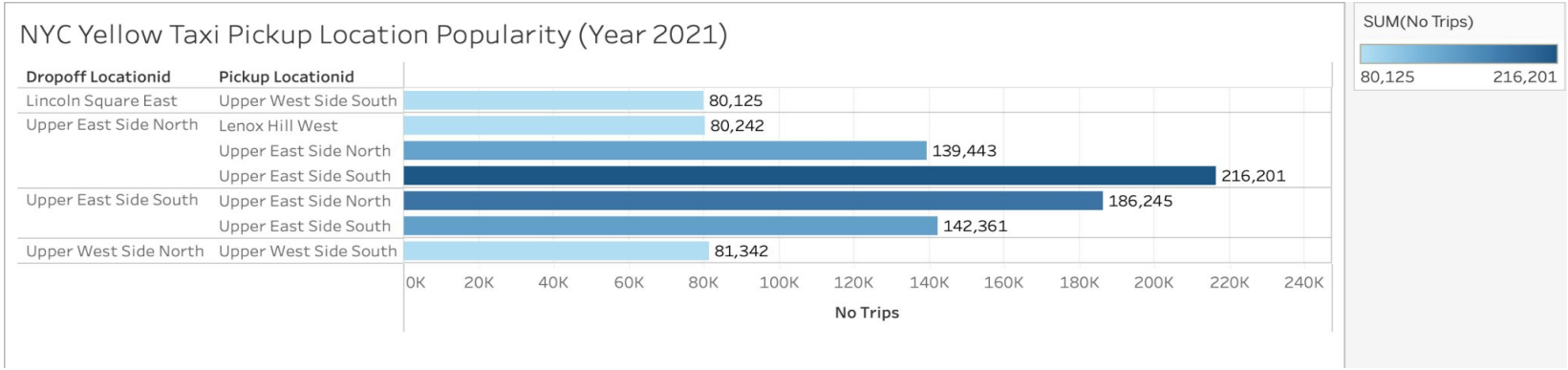




Q: What is the geographic pattern for using Taxi?

Route popularity

- The most popular route is between Upper East Side North and Upper East Side South.
  - Cultural preference / economic factor
  - Public transportation dynamics (Second Ave subway is opened in Jan 2017 before that the neighborhood's transit options were only buses and the Lexington Ave subway)
  - Parking challenges

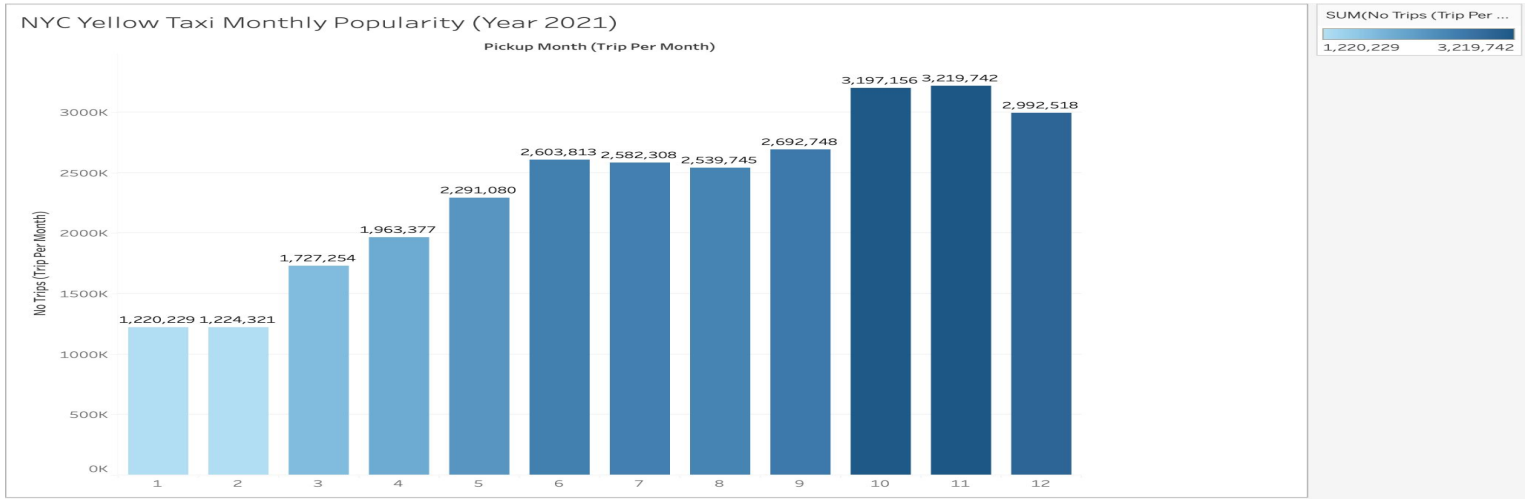


NYU

Q: What is the usage pattern per timeline?

Monthly trip records count

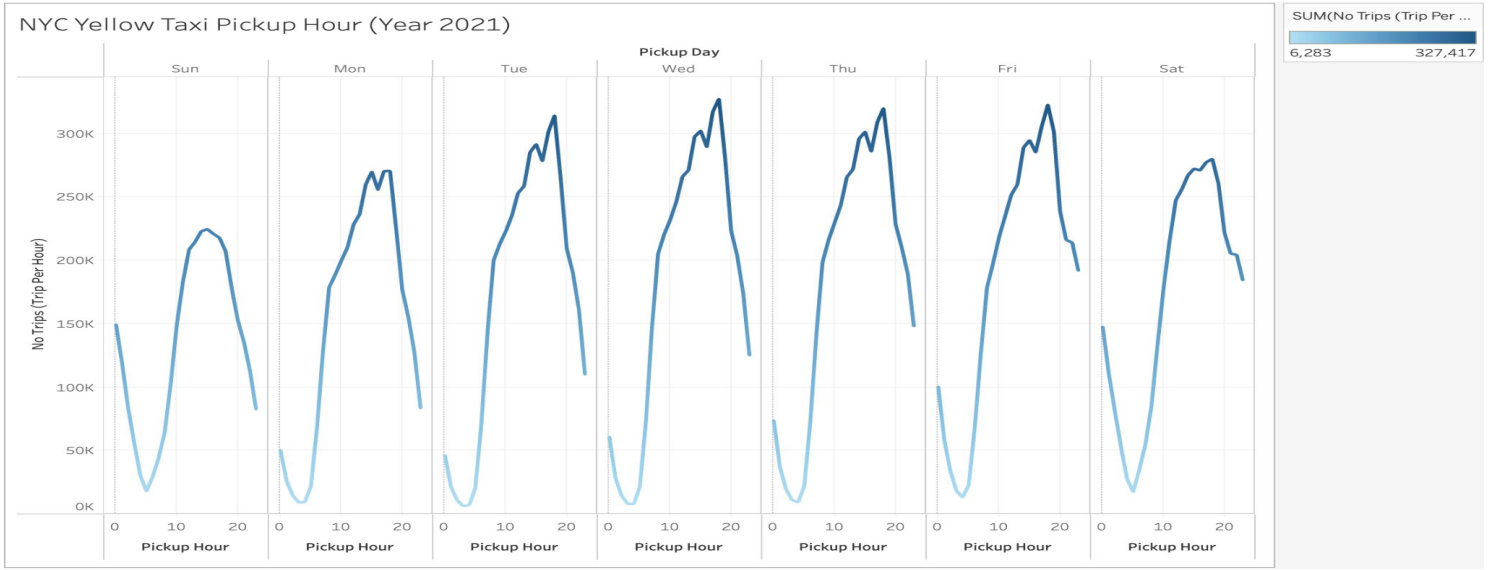
- Taxi usage peaks between October and December
  - Weather condition / Daylight saving time change
  - Holiday Shopping / Year-end activities / Tourism
- Relatively decreased usage in Jan and Feb, which aligns with the relatively higher average traffic speed in those periods



Q: What is the usage pattern per timeline?

Weekly trip records count

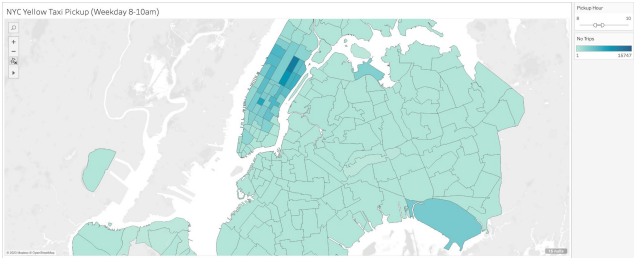
- Peak hour starts around 9am and ends around 8pm
- More trips on Tuesday to Friday
- Extra volume on Saturday and Sunday midnight



Q: Where to hail a taxi at different time?

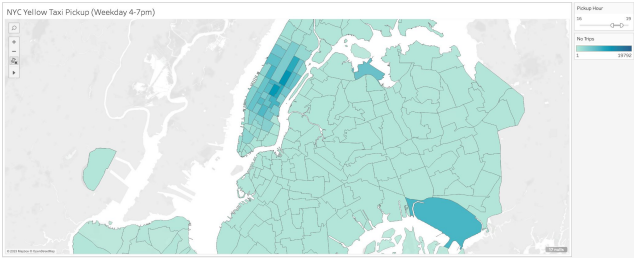
Weekday morning rush (8-10am):

- Most popular pickup location is Upper East Side North/South
- Followed by Penn Station and Grand Central area, the greater commuter gateway to the city major business district.



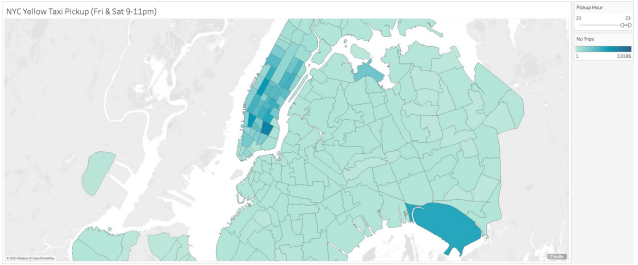
Weekday evening rush (4-7pm)

- Most popular pick up location is still Upper East Side South/North



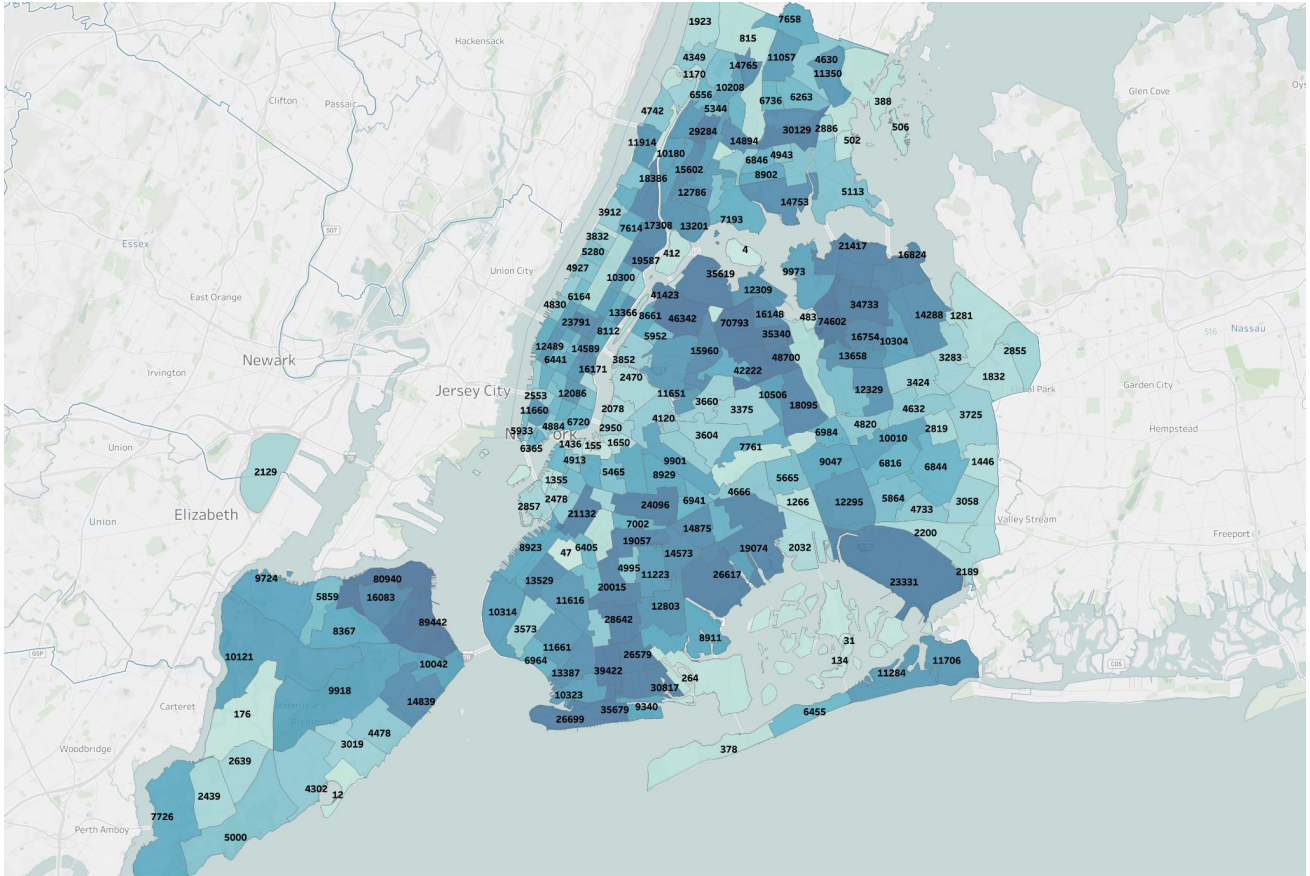
Weekend Schedule (Friday & Sat night 9-11 pm)

- It is much easier to hail a taxi in East Village or West Village area
- Also a more picks requirement from JFK airport

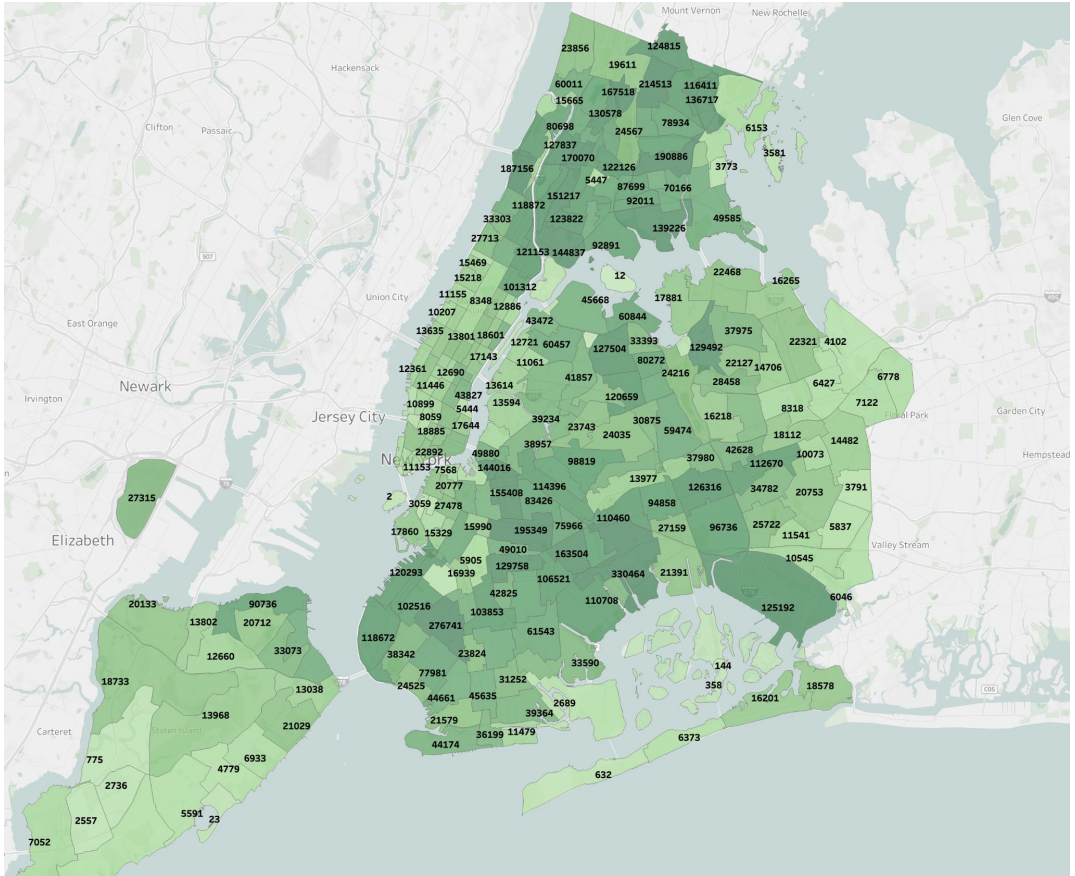


NYU

Q: What is the popularity of FHV Pick-Up location in 2021?



Q: What is the popularity of FHV Drop-Off location in 2021?



PART 02

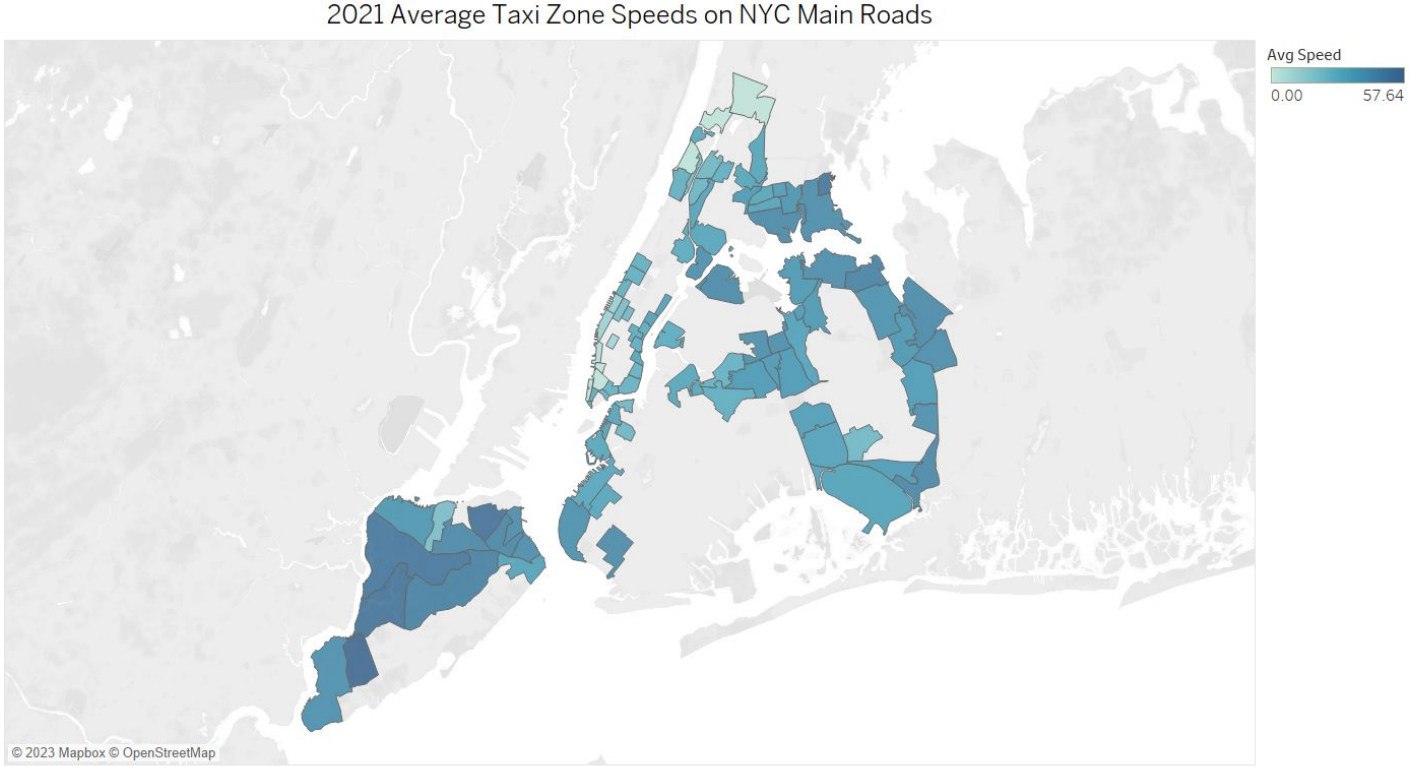
Data Limitation

- Speed data was only collected on main roads

Retrospective Data Profiling:
Location of the Sensor for Collecting Speed Data.

NYU

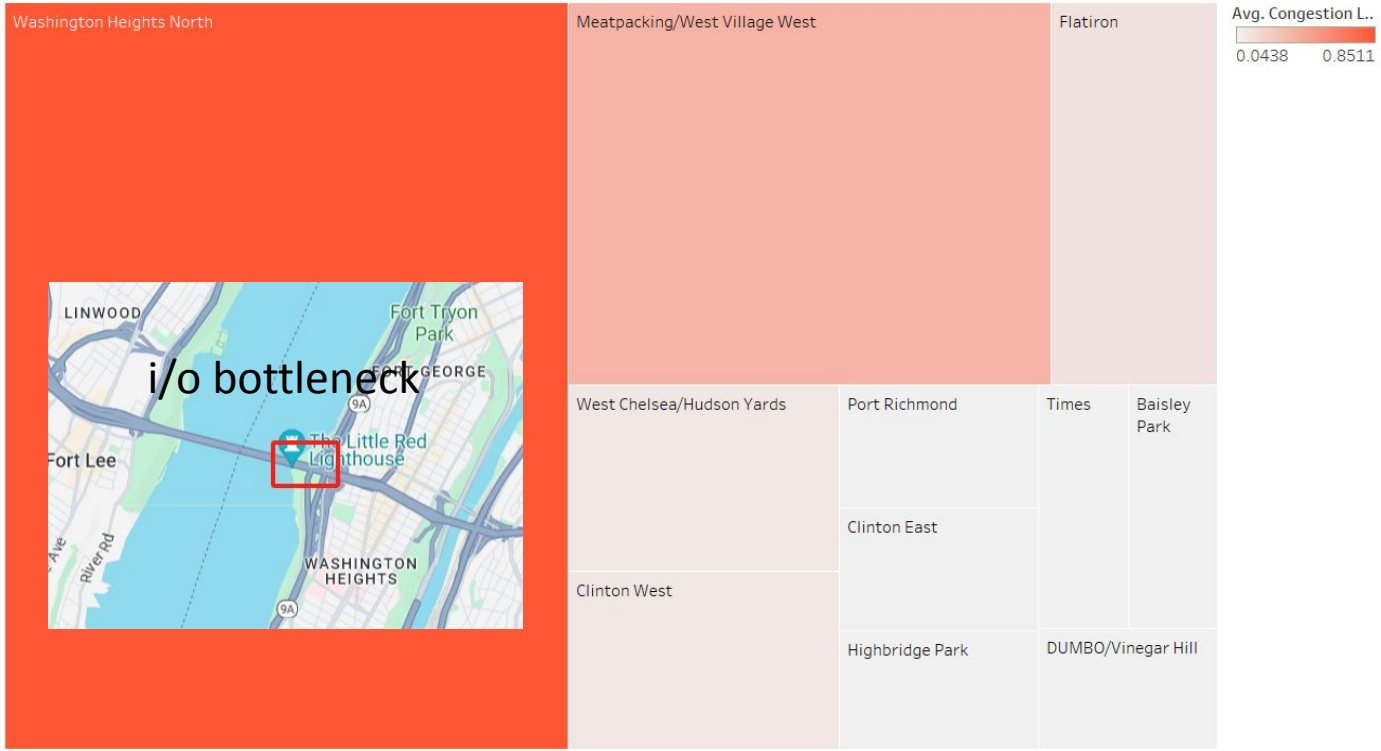
Q: What is the geographic pattern for speed on main roads?



Map based on Longitude (generated) and Latitude (generated). Colour shows details about Avg Speed. The view is filtered on Avg Speed, which keeps non-Null values only.

Q: What is the top congested zones by speed on main roads?

Top 11 Congested Taxi Zones in 2021 by Main Road Travel Times

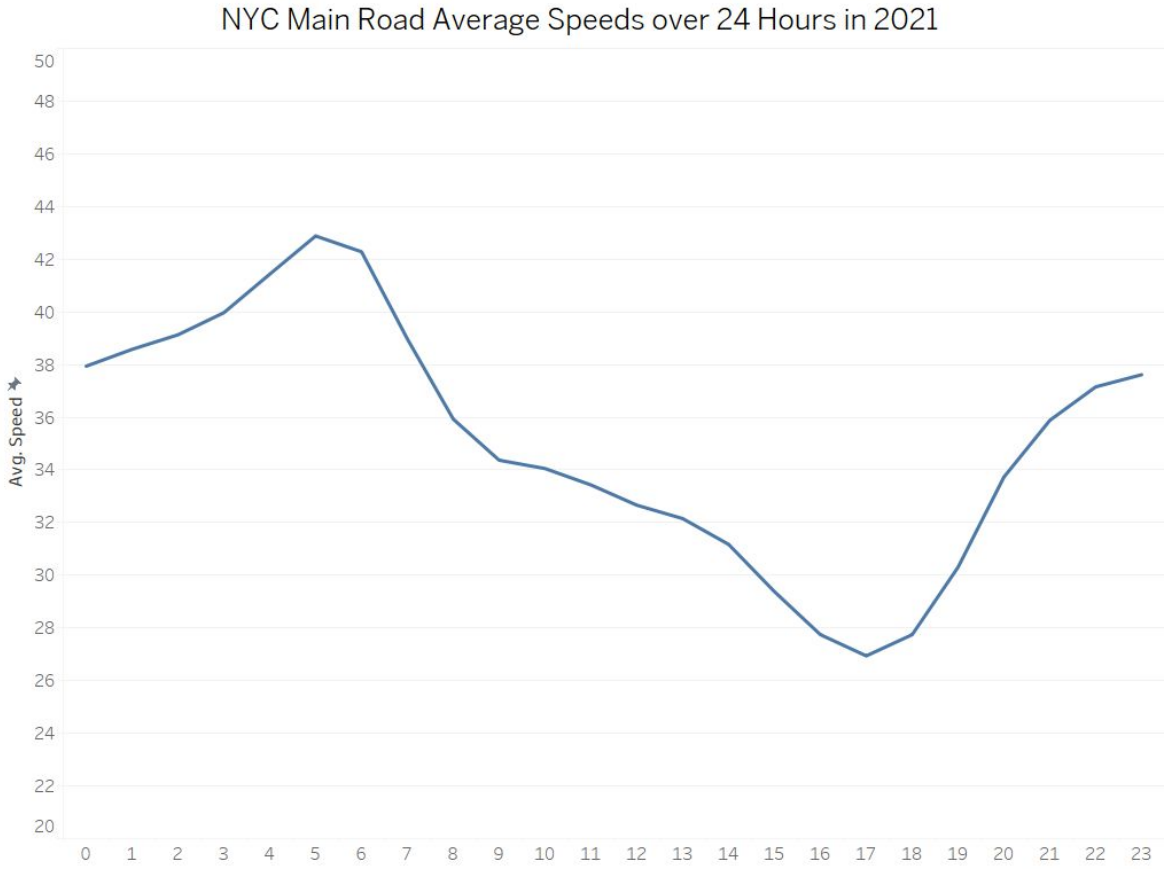


Zone. Colour shows average of Congestion Levels. Size shows average of Congestion Levels. The marks are labelled by Zone. The data is filtered on Congestion Levels, which includes values greater than or equal to 0.043.

NYU

PART 02

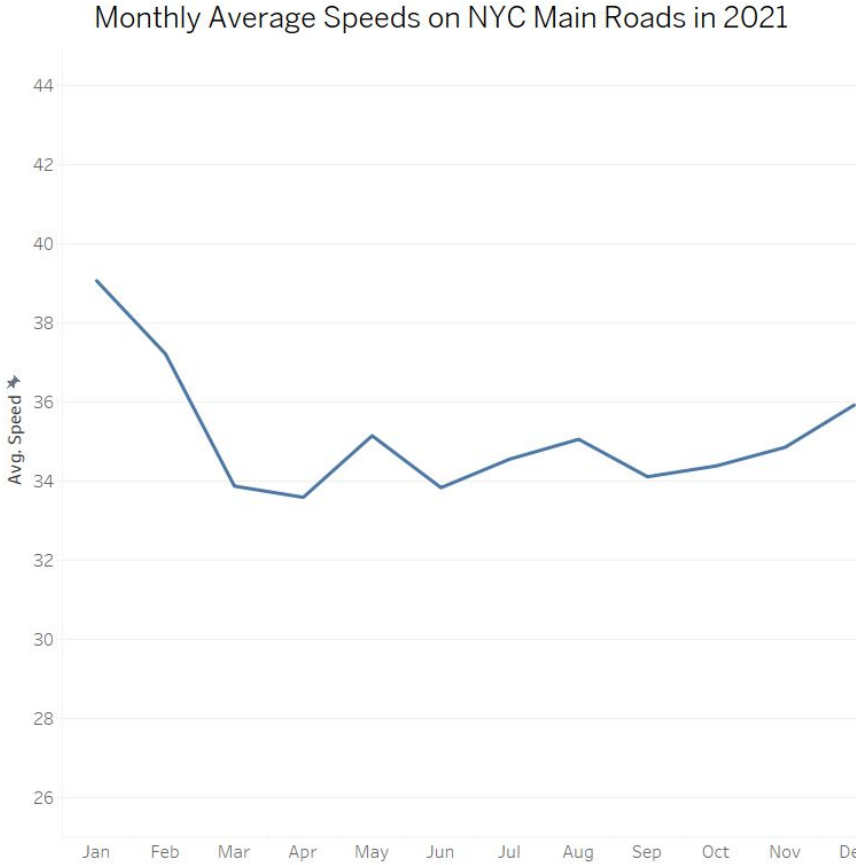
Identify peak and off-peak hours



NYU

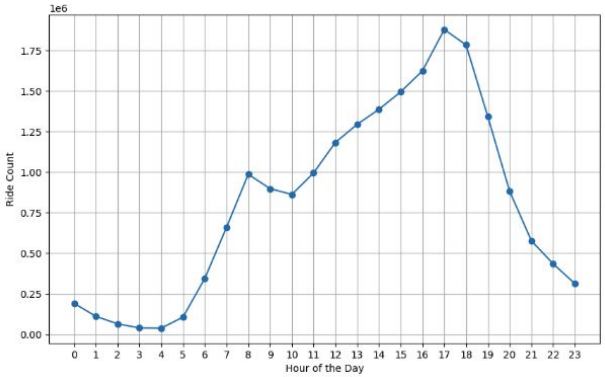
PART 02

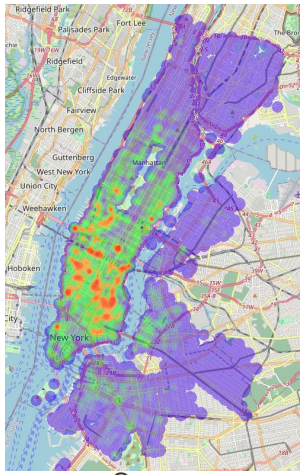
Compares traffic speeds across different months



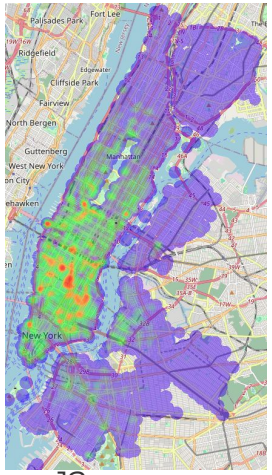
NYU

Compare start and end station at different times

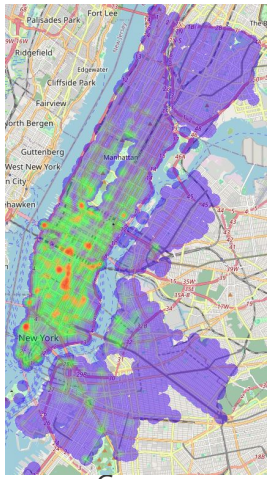




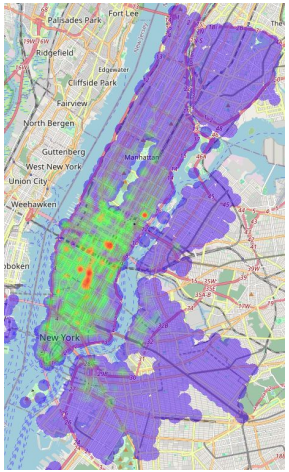
9am

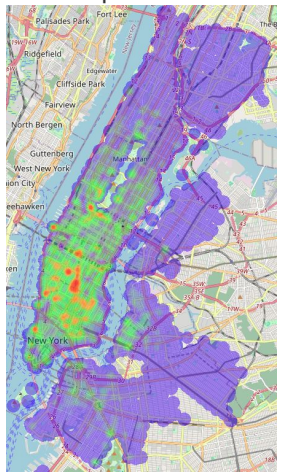


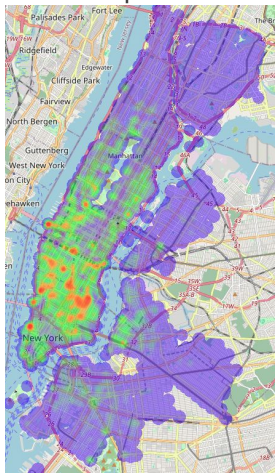
12pm



6pm







NYU

PART 02

Obstacles

- We need to join a GeoJSON file for data visualization. Since Tableau Desktop fetches data from cluster and **performs SQL queries locally for multi-source data that include local files**, we must **first compute the result table in Trino**. After that, we can carry out an Equi Join in Tableau Desktop
- In Tableau, there are problems with identifying the type of data columns. To solve this, we need to **edit data type while joining the datasets**
- In Tableau, times are **automatically shown in the local timezone**. Since our data source lacks timezone details, it defaults to UTC. To work around this, we'll need to **convert our original data from EST to UTC** by adding 5 hours before loading it into Tableau
- The Speed and CityBike dataset cannot cover all the taxi zones. For the latter, we **switch to a heatmap** for better visualization of our results.

PART 02

Analysis Limitation

Diverse Factors Shaping Traffic Patterns: The scope of our dataset is confined to traffic-related metrics. However, it is crucial to acknowledge that the dynamics of traffic patterns extend beyond these parameters. They are also significantly influenced by external variables, such as meteorological conditions and the economic profile of the community. This acknowledgment of external factors is essential for a comprehensive understanding of the data.

Neglecting Long-term Trends: For simplicity and focus, our study is confined to the examination of traffic-related data exclusively from the year 2021. This approach inherently limits our ability to conduct a longitudinal analysis that would consider historical trends and patterns over an extended period.

PART 02

Acknowledgements

- Special thanks to the NYU HPC team for their patient and thorough responses to our inquiries via email, greatly aiding our research
- Grateful acknowledgment to Tableau for the complimentary license
- Deepest appreciation to Professor Yang

PART 02

References

- **NYC Real-Time Traffic Speed Data:** https://www.kaggle.com/datasets/aadimator/nyc-realtime-traffic-speed-data/data
- **NYC Taxi Zones:** https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc
- Opencsv: https://opencsv.sourceforge.net/
- **The JTS Topology Suite (JTS):** https://locationtech.github.io/jts/
- **Algorithms for calculating variance:** https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance
- **NYC FHV (Uber/Lyft) Trip Data Simple (2015-2022):** https://www.kaggle.com/datasets/jeffsinsel/nyc-fhv-uberlyft-trip-data-simple-2015-2022
- **Upper East Side Taxis, For-Hire-Vehicles and the Second Avenue Subway**: https://wagner.nyu.edu/files/faculty/publications/SASFHVandTaxis.pdf

Thank you!