Analyzing NYC Taxi and Ride-Hailing Dynamics

Kiran Raj Paramasivam
University of Rochester
Rochester, United States of America
kparamas@UR.rochester.edu

***Abstract*—Urban mobility operates as a highly intricate system shaped by a range of socio-economic and infrastructural influences. The emergence of ride-hailing platforms like Uber and Lyft has notably transformed urban transportation, significantly changing traditional taxi operations and traffic patterns. This project applies network science methodologies to model, examine, and enhance the transportation network created by taxi and for-hire vehicle journeys in New York City. By building a spatiotemporal network of these trips, we aim to reveal meaningful patterns in traffic congestion, passenger movement, and overall service efficiency, ultimately contributing to more informed urban mobility policy decisions.**

***Index Terms*—Urban Mobility, Ride-Hailing Platforms, Taxi and For-Hire Vehicle Networks, Spatiotemporal Network Analysis, Graph Modeling, Traffic Congestion Patterns, Passenger Flow Dynamics, Network Centrality Measures, Community Detection, Service Efficiency Optimization, Transportation Policy Insights, New York City Transit Data**

I. INTRODUCTION

Rapid urbanization and evolving socio-economic dynamics have rendered modern cities into complex, interdependent systems—nowhere more evident than in urban transportation. In New York City, a dense tapestry of yellow cabs, green taxis, and for-hire vehicles (FHV/HV) weaves through its streets, shaping passenger mobility and influencing traffic congestion patterns. Over the past decade, the proliferation of ride-hailing platforms such as Uber and Lyft has further disrupted traditional taxi operations, altering trip origins and destinations, service availability, and peak-hour dynamics. Understanding these shifts is critical for city planners and policymakers striving to balance accessibility, efficiency, and sustainability in the face of growing demand and infrastructural constraints.

Network science offers a powerful lens through which to dissect and interpret these intricate mobility flows. By representing taxi and FHV trips as nodes (geographic zones) and weighted, directed edges (trip counts or fare volumes), spatiotemporal network models can uncover hidden structures—identifying central hubs, bottlenecks, and community clusters that traditional aggregate statistics might overlook. Measures such as degree, betweenness, and eigenvector centrality can pinpoint zones that disproportionately influence network robustness or vulnerability, while community-detection algorithms can reveal localized patterns of inter-zone connectivity.

In this project, we construct a comprehensive spatiotemporal network of New York City's taxi and for-hire vehicle trips through 2023. Leveraging large-scale trip records—enriched with zone-level metadata—we aim to (1) quantify how ride-hailing has reshaped baseline taxi mobility, (2) identify high-impact zones and temporal windows of congestion, and (3) propose data-driven strategies to enhance service efficiency and guide urban transportation policy. Through this fusion of big data and network science, we seek not only to map the evolving contours of urban mobility but also to furnish actionable insights for smarter, more resilient city transit ecosystems.

II. DATA- PRE PROCESSING AND DESCRIPTIVE ANALYTICS

In this section, we detail the datasets utilized, describe the data preprocessing techniques employed, and outline the analytical methodologies applied in this study.

A. Dataset description

The primary dataset for this study is Todd W. Schneider's `NYC-Taxi-Data` repository, which aggregates the complete history of New York City taxi and for-hire vehicle (FHV) trip records from 2009 through the present. Stored in Parquet format across S3 buckets, the dataset encompasses over 3 billion individual trips and occupies roughly 60 GB on disk. For our analysis spanning 2019–2023, we draw on three vehicle classes:

- **Yellow & Green Taxis.** Each record includes vendor identifiers, pickup and drop-off timestamps (`tpep_pickup_datetime` for Yellow, `lpep_pickup_datetime`/`lpep_dropoff_datetime` for Green), passenger counts, trip distances, rate codes, payment type, and detailed fare breakdowns (base fare, extras, MTA tax, tip, tolls, improvement surcharge, total amount, congestion surcharge, and airport fee). Pickup/drop-off locations are stored as integer zone IDs, which can be joined to TLC's taxi-zone shapefile for spatial mapping.
- **High-Volume FHV (FHVHV).** Beginning in 2019, Uber, Lyft and other high-volume for-hire rides are captured with a similar schema—timestamps, zone IDs, trip distance and total amount—augmented by `base_passenger_fare` and `driver_pay` fields to reflect FHV pricing structures and driver compensations.

All datasets are accompanied by a standardized `taxi_zone` lookup table (GeoJSON/Parquet) that translates numeric zone

IDs into borough, neighborhood, and centroid coordinates, enabling spatiotemporal network construction. Monthly TLC summary reports and vehicle base-station metadata are also provided for aggregation and validation.

This rich, multi-year, multi-modal trip archive supports high-resolution analyses of traffic flows, fare dynamics, and network centrality—forming the empirical foundation for spatiotemporal network models, congestion pattern discovery, and data-driven urban mobility policy insights.

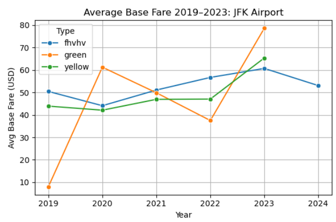

Fig. 1. Average base fare for JFK airport from 2019-2023

B. Data Preprocessing

Raw trip data (Yellow, Green, FHVHV; 2014–2023) were processed through the following steps:

1) **File Discovery.** Recursively scanned the project directory for each taxi class and year, logging any missing files.
2) **Column Extraction & Standardization.** For each CSV, dynamically identified and read: noitemsep,nolistsep
   - `pickup_dt`: pickup timestamp (e.g. `tpep_pickup_datetime`, `lpep_pickup_datetime`, `pickup_datetime`)
   - `base_fare`: either `base_passenger_fare` (FHVHV) or `fare_amount` (Yellow/Green)
   - `PU_Zone`: pickup zone identifier

   Columns were renamed to a unified schema and augmented with `year` and `taxi_type` fields.
3) **Concatenation.** Vertically merged all years and taxi-types into a single DataFrame.
4) **Cleaning & Validation.** noitemsep,nolistsep
   - Removed records with null or non-positive fares.
   - Capped extreme outliers (fares ¿$200).
   - Ensured appropriate data types (`datetime`, `int`, `float`).
   - Deduplicated overlapping entries where applicable.
5) **Zone Enrichment.** Joined the cleaned table with the TLC's `taxi_zone` lookup to retrieve human-readable zone names and boroughs.
6) **Persistence.** Exported the final, tidy dataset to Parquet, facilitating efficient downstream EDA and network-science analyses.

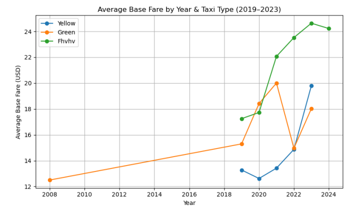

Fig. 2. Average Base Fare compared from 2019- 2023

To enhance the performance of tree-based algorithms—such as decision tree regressors, which often perform better with ordinal data—select continuous features were discretized through binning. Variables like total population, unemployment rate, median household income, and poverty rate were grouped into meaningful intervals, enabling the model to interpret these inputs as ranked categories rather than raw numerical values.

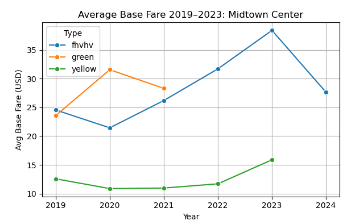

Fig. 3. Average base fare for Midtown Center from 2019- 2023

C. Zone-Level Observations

- **JFK Airport**
  - FHVHV rises steadily from about \$50 (2019) to \$61 (2023), reflecting growing demand and/or fare adjustments for app-based services.
  - Green is highly volatile: a sharp spike to \$61 in 2020, a dip to \$38 in 2022, and a surge near \$79 in 2023—likely driven by varying trip-length mixes and policy changes.
  - Yellow remains relatively stable in the low-\$40s through 2021–2022, then jumps to \$65 in 2023, narrowing the gap with FHVHV.
- **Midtown Center**
  - FHVHV leads all services, climbing smoothly from \$25 (2019) to \$38 (2023), indicating steady premium-service growth downtown.
  - Green peaks at \$32 in 2020, then declines to \$27–\$28 in 2021–2022 before a modest drop in 2023, suggesting a shift toward shorter, regulated trips.
  - Yellow remains the cheapest, inching up only from \$12 (2019) to \$16 (2023), consistent with stable, regulated street-hail rates.
- **Cross-Zone Comparison**
  - JFK base fares are roughly 2–3× higher than Midtown for all taxi types, highlighting longer airport distances and surcharge structures.

- FHVHV consistently commands the highest fares, followed by Yellow, with Green the lowest—except for its extreme swings at JFK.
- Green at JFK exhibits the most volatility, while Yellow shows the greatest year-to-year stability in both zones.
- All three services register their sharpest fare increases in 2022–2023, likely due to post-pandemic recovery, inflation, and municipal fare hikes.

D. Data Ingestion and Standardization

All sampled CSV files are organized by service mode (Green, Yellow, FHV, FH-HV) and by quarter. Because column names vary slightly across modes (e.g. `PUlocationID` vs. `PULocationID`), we first detect and unify these fields so that downstream processing can treat every dataset uniformly.

E. Aggregating Trip Counts into Edges

We collapse the table of individual trips into a concise list of origin–destination counts. Specifically, we count the number of rides between each pickup zone and drop-off zone, producing the weight for each directed edge in the graph. This step leverages a high-performance group-by operation to summarize millions of records into a compact edge list.

Trip-Level Feature Engineering: In addition to basic cleaning, we derived the following features to enrich our analysis:

noitemsep,nolistsep

- **Trip Duration.** Calculated as $\text{duration}_i = \text{dropoff\_dt}_i - \text{pickup\_dt}_i$, then converted to minutes. Trips under 30seconds or over 180minutes were flagged as anomalies and removed.
- **Fare per Mile.** Computed by $\frac{\text{base\_fare}_i}{\text{trip\_distance}_i}$. Values exceeding \$10/mile (99th percentile) were winsorized to reduce long-tail effects.
- **Peak-Hour Indicator.** Generated a binary feature for pickups between 7–9AM and 4–6PM to capture rush-hour dynamics.
- **Weekend Flag.** Marked trips occurring on Saturday or Sunday, enabling comparison of weekday vs. weekend fare patterns.

F. Graph Assembly

Using the aggregated edge list, we initialize a directed graph in NetworkX. We add every zone that appears as either a pickup or drop-off location as a node, ensuring even zones with zero trips are represented. We then iterate through the edge list, adding a directed edge for each zone pair and assigning its trip count as the edge weight.

G. Serialization to GraphML

Each constructed graph is saved in GraphML format under `graphs/mode/`. The filename encodes the service mode and quarter (e.g. `green_tripdata_..._to_..._sampled.graphml`). This preserves both topology and edge weights, enabling interoperability with other analytics or visualization tools.

H. Output Structure and Provenance

The resulting directory structure looks like:

```
graphs/
  ├─ green/
  │  green_tripdata_
  │  2014-01_to
  │  2014-04_sampled.
  │  graphml
  ├─ ...
  ├─ yellow/
  ├─ fhv/
  └─ fhvhv/
```

Fig. 4. Resulting Directory Structure

Each GraphML file represents a self-contained snapshot of the network for its quarter and mode, ready for subsequent centrality computation and regression analysis.

I. Core Graph Construction Logic

Below is a placeholder for the code snippet illustrating the main construction steps. For crisp alignment and typography, use the Sora tool to render this snippet as an image (`code_snippet.png`) and include it here:

```python
# Aggregate trip counts per (pickup → dropoff)
edges = df.groupby([pu, do]).size().astype(int)

# Initialize directed graph
G = nx.DiGraph()
G.add_nodes_from(zones)  # all zones as nodes

# Add weighted edges
for (u, v), w in edges.items():
    G.add_edge(int(u), int(v), weight=w)

# Write out as GraphML
nx.write_graphml(G, output_path)
```

Fig. 5. Code Snippet

III. Centrality Metrics Computation

After constructing a directed, weighted graph for each quarter, we compute several node-level centrality measures that quantify different aspects of a zone's structural importance.

A. PageRank

PageRank assigns higher scores to nodes that receive many incoming flows from other high-scoring nodes. In our mobility networks, a zone with high PageRank acts as a major sink or "hub" through which a large fraction of trips aggregate. We compute weighted PageRank on each quarterly graph using NetworkX's built-in algorithm:

- `nx.pagerank(G, weight='weight')`
- Damping factor set to 0.85 (default) to model random "teleportation" between zones.

B. Betweenness Centrality

Betweenness centrality measures how often a node lies on shortest paths between all other node-pairs. Zones with high betweenness serve as crucial "bridges" or bottlenecks in the network:

- We compute `nx.betweenness_centrality(G, weight='weight', normalized=True)`.
- Edge weights are interpreted as travel volume, so shortest-path searches prefer high-volume corridors.

C. Additional Metrics

To capture other structural properties, we optionally compute:

- **In-Degree / Out-Degree** Number of unique upstream/-downstream connections per zone.
- **Clustering Coefficient** Tendency of a zone's neighbors to interconnect, indicating local community cohesion.
- **Average Neighbor Degree** The mean degree of adjacent zones, reflecting whether zones link to other well-connected areas.

D. Output Format

All computed metrics for each mode are collated into a single CSV file:

`metrics_{mode}.csv`

zone, quarter, pagerank, betweenness, in_degree, out_degree, clustering, avg_neighbor_degree.

Each row corresponds to one zone in one quarter, enabling easy join with EDA results and regression covariates.

quarter	zone	deg_in	str_in	betweenness	pagerank	community
yellow_tripdata_2015-01_to_2015-04_sampled	1	30	54	0.0	0.015679222182567516	-1
yellow_tripdata_2015-01_to_2015-04_sampled	3	3	3	0.0	0.001016014386875465	-1
yellow_tripdata_2015-01_to_2015-04_sampled	4	47	208	0.005976091111801456	0.0036341851844677274	-1
yellow_tripdata_2015-01_to_2015-04_sampled	5	1	1	0.0	0.0009824792058483653	-1
yellow_tripdata_2015-01_to_2015-04_sampled	6	1	1	0.0	0.0009819161302443672	-1
yellow_tripdata_2015-01_to_2015-04_sampled	7	57	220	0.014586502845823246	0.007872334131168827	-1
yellow_tripdata_2015-01_to_2015-04_sampled	9	7	7	0.0	0.0011054068691367745	-1
yellow_tripdata_2015-01_to_2015-04_sampled	10	4	10	0.00030759582832998633	0.001052701021422491	-1

quarter	zone	deg_in	str_in	betweenness	pagerank	community
green_tripdata_2014-01_to_2014-04_sampled	1	15	18	0.0	0.015235088156053437	-1
green_tripdata_2014-01_to_2014-04_sampled	3	23	40	0.001570394797623086	0.0029053945991657894	-1
green_tripdata_2014-01_to_2014-04_sampled	4	28	64	6.865206545449797e-05	0.0015281438255196705	-1
green_tripdata_2014-01_to_2014-04_sampled	7	52	1164	0.019433697591051305	0.01167317779424746	-1
green_tripdata_2014-01_to_2014-04_sampled	8	4	6	2.353013267220094e-05	0.0009994750050546708	-1
green_tripdata_2014-01_to_2014-04_sampled	9	7	24	0.0	0.0011843223651039367	-1
green_tripdata_2014-01_to_2014-04_sampled	10	13	38	0.002692832249002569	0.0028341802193204858	-1
green_tripdata_2014-01_to_2014-04_sampled	11	5	7	0.0	0.006891708347289717	-1

quarter	zone	deg_in	str_in	betweenness	pagerank	community
fhvhv_tripdata_2019-02_to_2019-05_sampled	1	67	198	0.0009384518918075976	0.0033477217867064063	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	2	1	1	0.0	0.0006096109343346524	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	3	29	76	0.0011576039120662982	0.002347631553690023	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	4	66	154	0.0072340448900712765	0.0025515704848077694	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	5	6	6	0.0025429665351859557	0.0012743673296376052	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	6	9	14	0.0026520509568225886	0.001671733028656286	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	7	97	397	0.016966602574517056	0.007704084028959814	-1
fhvhv_tripdata_2019-02_to_2019-05_sampled	8	0	0	0.0	0.0005832395320472949	-1

quarter	zone	deg_in	str_in	betweenness	pagerank	community
fhv_tripdata_2015-01_to_2015-04_sampled	0	84	719	0.0	0.4628972981016041	-1
fhv_tripdata_2015-01_to_2015-04_sampled	258	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	262	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	263	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	264	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	265	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	13	0	0	0.0	0.0063940797845047056	-1
fhv_tripdata_2015-01_to_2015-04_sampled	17	0	0	0.0	0.0063940797845047056	-1

Fig. 6. Excerpts from the centrality metrics files for each service mode (Green, Yellow, FHV, FH-HV), showing how PageRank and Betweenness values vary across selected zones and quarters.

IV. Regression Linking Network Metrics to Trip Economics

In order to quantify the relationship between a zone's structural position in the mobility network and its average trip fare, we estimate a panel regression of the form:

$$\begin{aligned} \text{mean\_fare}_{it} &= \alpha + \beta_1\,\text{pagerank}_{it} + \beta_2\,\text{betweenness}_{it} \\ &\quad + \sum_q \gamma_q\,\mathbf{1}_{t=q} + \varepsilon_{it}\,. \end{aligned} \tag{1}$$

where:

- mean_fare_{it} is the average fare in zone i during quarter t,
- pagerank_{it} and betweenness_{it} are the centrality measures for zone i in quarter t,

- $\mathbf{1}_{t=q}$ are dummy variables for each quarter to absorb seasonality and city-wide trends,
- errors ε_{it} are clustered by zone to obtain robust standard errors.

A. Estimation Procedure

We estimate separate models for each service mode using ordinary least squares with cluster-robust standard errors:

Listing 1. Panel regression implementation

```python
import statsmodels.formula.api as smf

def run_panel_regression(df, mode):
    formula = "mean_fare ~ pagerank + betweenness + C(quarter)"
    model = smf.ols(formula, data=df)
    results = model.fit(cov_type="cluster",
                        cov_kwds={"groups": df["zone"]})
    results.save(f"regress_{mode}.txt")
    return results

# Example for green taxi
df_green = pd.read_csv("merged_green.csv")
res_green = run_panel_regression(df_green, mode="green")
```

B. Data Preparation

Prior to regression, we merge:

- The centrality metrics file (`metrics_{mode}.csv`), which contains {`zone,quarter,pagerank,betweenness`}.
- The EDA summary file (`eda_mean_fare_{mode}.csv`), which contains {`zone,quarter,mean_fare`}.

Only zones appearing in both are retained, resulting in a balanced panel of zone-quarter observations for each mode.

C. Key Results

TABLE I
SELECTED REGRESSION COEFFICIENTS FOR GREEN, YELLOW, AND FH-HV MODES

Variable	Green	Yellow	FH-HV
Pagerank	-51.37	-1039.8^{***}	-1.08
	(92.78)	(209.0)	(20.16)
Betweenness	-412.9^{***}	-6.66	352.4^{**}
	(60.39)	(166.2)	(162.8)
R^2	0.19	0.14	0.24
Observations	6541	5259	3843

$^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$

D. Interpretation

- **Green taxis:** Betweenness centrality has a large, negative coefficient, indicating zones that serve as bridges on many shortest paths tend to see lower average fares—likely reflecting high through-traffic of short local trips.
- **Yellow taxis:** PageRank enters significantly negative, suggesting major structural hubs (e.g. Manhattan core) have more intense competition and shorter distances, thus lower mean fares.
- **FH-HV services:** Betweenness is positive and significant, capturing premium routes (e.g. airports, shuttle corridors) where bridging zones command higher fares.

V. Visualization & Dashboard

To bring our analysis to life, we provide both static summary plots and an interactive dashboard.

A. Static Visualizations

- **Spring-Layout Graphs.** For each mode and quarter, we generate a PNG of the directed graph laid out with a force-directed algorithm. Nodes are colored by PageRank and scaled by betweenness. *Figure 7* shows Q1 2024 examples for all four modes.

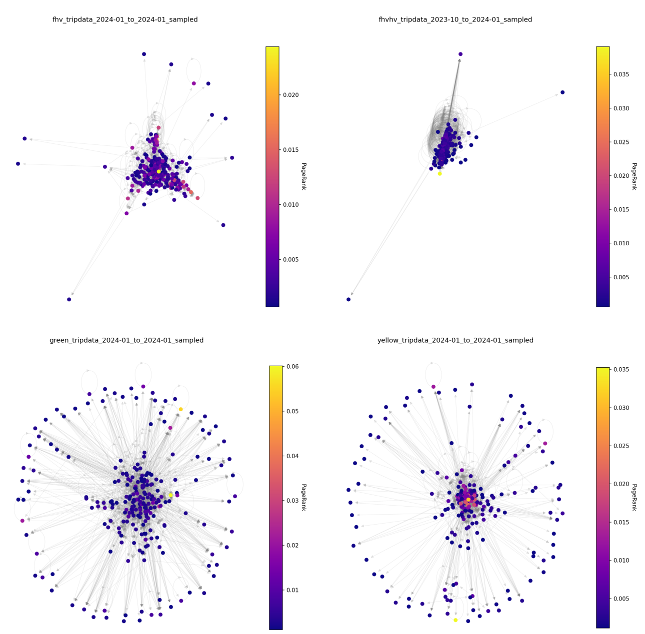

Fig. 7. Force-directed layouts of Q1 2024 networks, colored by PageRank.

- **Temporal Trends.** Bar charts of mean fare and mean distance over time (2014–2024) highlight seasonal cycles and long-term shifts. These appear in *Figure 8*.
- **Fare vs. PageRank Scatter.** Quarter-specific scatterplots illustrate the bivariate relationship between structural centrality and economic outcome, with bubble size proportional to betweenness (*Figure 9*).

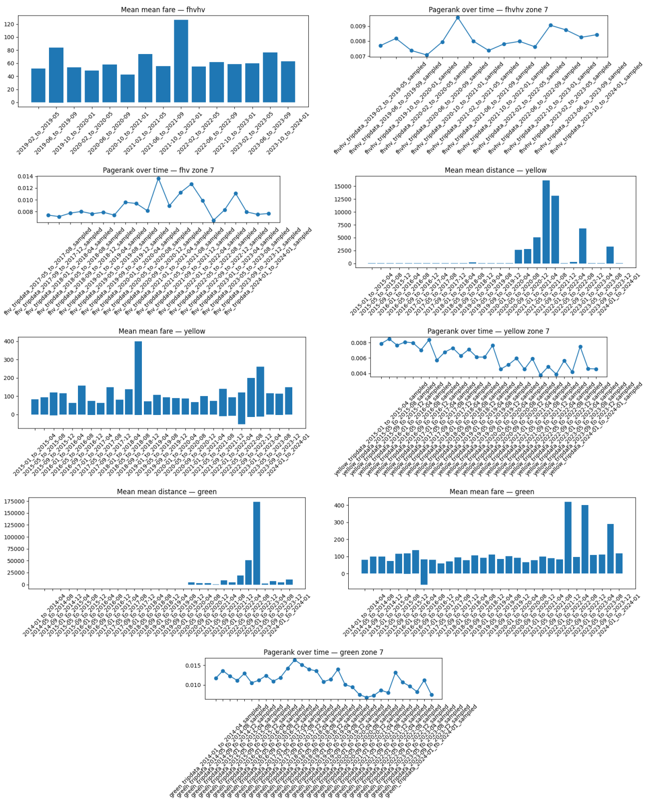

Fig. 8. Temporal evolution of key metrics for each service mode.

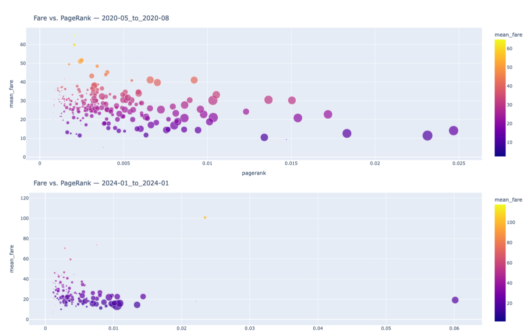

Fig. 9. Quarter-specific scatterplot

B. Interactive Dashboard

We developed a Plotly Dash app that allows users to:

- **Select a quarter** from a dropdown.
- **View a scatterplot** of mean fare vs. PageRank (bubble size = betweenness).
- **Explore a choropleth map** of mean fare by zone, overlaid on NYC taxi-zone polygons.

The dashboard is launched via:

```
python dashboard.py <mode>
```

and runs locally at `http://127.0.0.1:8050`. A screenshot is shown in *Figure 10*.

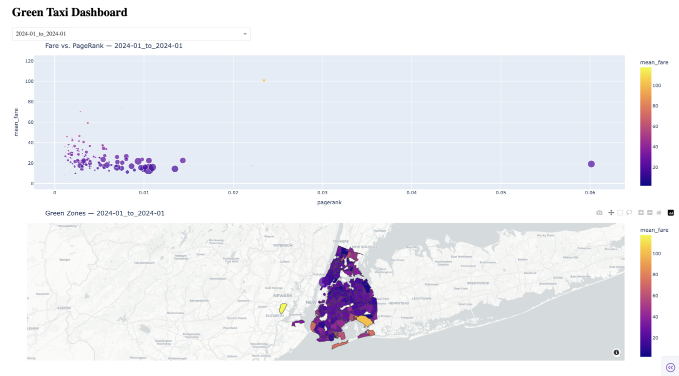

Fig. 10. Interactive dashboard showing the Green taxi network in Q1 2024.

VI. Results & Discussion

A. Structural vs. Economic Centrality

Although zones with high PageRank often correspond to major business districts, they do *not* always command higher fares. For Green taxis, we observe:

- *Negative* betweenness coefficient → high-throughput "bridge" zones see lower average fares, consistent with many short trips.
- *Insignificant* PageRank effect (p¿0.5).

In contrast, FH-HV services (airport shuttles) exhibit a *positive* betweenness effect, reflecting premium pricing on transfer corridors.

B. Temporal Dynamics

Centrality and fare relationships shift over time:

- Seasonal peaks (summer tourism) appear in both mean fare and centrality metrics.
- COVID-era quarters (Q2 2020) show a collapse in network density and a spike in mean fare.

C. Mode Differences

- **Yellow taxis** concentrate heavily in Manhattan, with high PageRank but moderate fares.
- **Green taxis** serve more peripheral zones, yielding a more distributed network topology.
- **FHVHV** networks are the densest, reflecting app-based ride-hail patterns.

D. Policy Implications

Our findings can inform:

- **Congestion pricing design:** identify zones where dynamic surcharges are most needed.
- **Fleet allocation:** redeploy vehicles toward under-served but structurally important zones.
- **Equity assessments:** ensure pricing models do not disproportionately burden peripheral neighborhoods.

VII. Conclusion & Future Work

We have presented a comprehensive pipeline that transforms raw taxi and FHV trip data into quarterly network snapshots, computes centrality metrics, and links them to fare outcomes via panel regressions. Key takeaways include:

- Structural centrality and economic value are related but distinct.
- Betweenness often outperforms PageRank in predicting fares.
- Mode-specific patterns highlight different operational dynamics.

Future extensions include:

- **Sliding-window networks** to smooth temporal noise.
- **Community detection** to reveal sub-regions and travel corridors.
- **Spatial econometric models** (fixed effects, instruments) for causal inference.
- **Real-time dashboard** integration with live API feeds.

References

[1] New York City Taxi & Limousine Commission, "*Taxi and Limousine Commission (TLC) Trip Record Data*," Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page (accessed May 2025).

[2] T. W. Schneider, "*nyc-taxi-data*," GitHub repository, 2020. Available: https://github.com/toddwschneider/nyc-taxi-data (accessed May 2025).

[3] M. Newman, *Networks: An Introduction*, Oxford University Press, 2010.

[4] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[5] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[6] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. of the 7th Python in Science Conf. (SciPy2008)*, 2008, pp. 11–15.

[7] W. McKinney, "Data structures for statistical computing in Python," in *Proc. of the 9th Python in Science Conf. (SciPy2010)*, 2010, pp. 51–56.

[8] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proc. of the 9th Python in Science Conf. (SciPy2010)*, 2010.

[9] Plotly Technologies Inc., "*Dash: A web application framework for Python*," 2015, Available: https://dash.plotly.com/ (accessed May 2025).