# UBER RIDES TRENDS IN NEW YORK CITY

Ricardo de los Rios
Chong Zhuang
Ezinne Obayagbona
Lina Orjuela

# AGENDA

**01**
Questions we found interesting & motivation

**02**
how and where we found the data

**03**
data exploration & clean up process

**04**
Analysis process

**05**
Numerical Summary & Visualization of Summary

**06**
Implication of Findings

# QUESTIONS WE FOUND INTERESTING & MOTIVATION

**01** **Which destination has the highest number of rides?**

**Motivation:** To understand what hours of the day are the peak requests and which city has the most Uber requests

What are the earnings of Uber drivers based on factors such as time of day, location, or type of vehicle used?
Is there a relation between public transportation lines and Uber pickup and dropoff location?

**02** **What is the average fare amount?**

**Motivation:** We were curious about how a driver can get data into which area has the highest fares. Basically, do Uber Fares depend on different times of the day and duration of the ride?

**03** **Does weather affect transportation and traffic?**

**Motivation:** To get a sense of distribution of rides by weather conditions and if there is a correlation between weather and rides

# HOW AND WHERE WE FOUND THE DATA

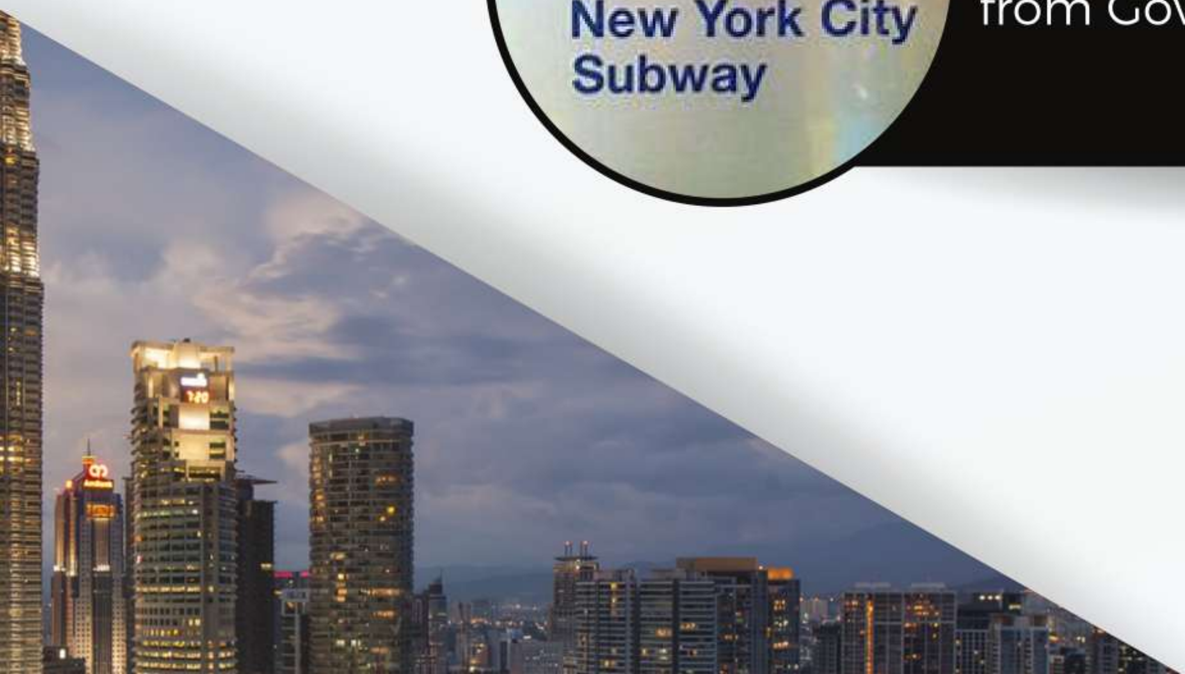
We use Uber ride data de Kaggle, for New York City, from 2009 to 2015.

**Openweather**

- We couldn't get API data directly due to historical data for analysis
- Used a Paid version and got an export for this

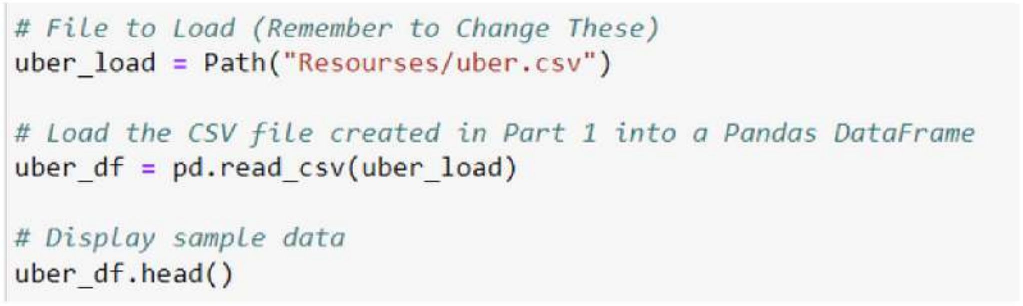
New York City Subway lines data from Government website

Geoapify.com - for visualization of subway destinations and drop off locations.



# DATA EXPLORATION & CLEAN UP PROCESS

## 1 Import libraries and load the database



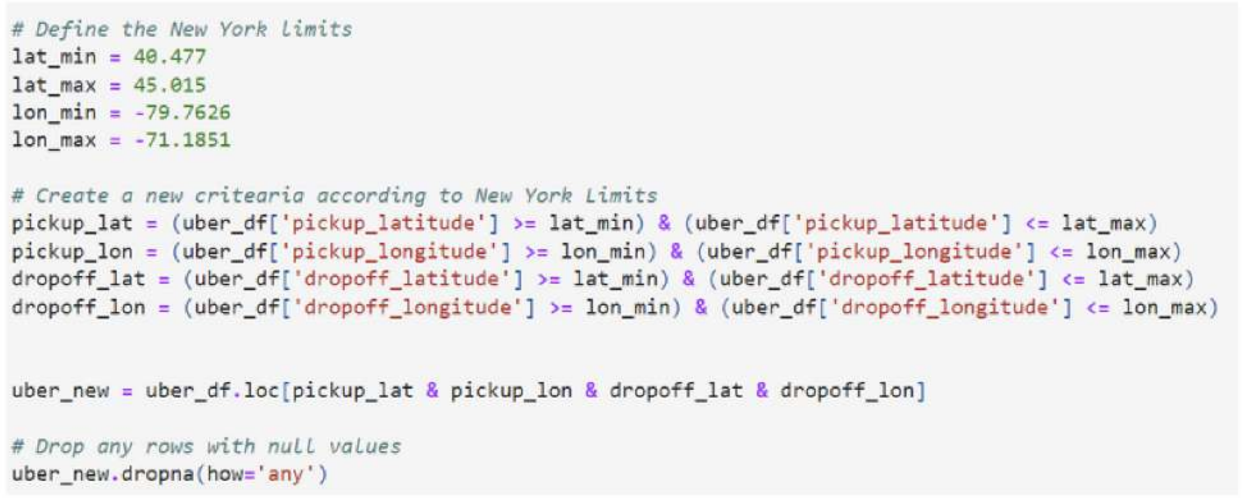
```
# File to Load (Remember to Change These)
uber_load = Path("Resourses/uber.csv")

# Load the CSV file created in Part 1 into a Pandas DataFrame
uber_df = pd.read_csv(uber_load)

# Display sample data
uber_df.head()
```

## 2 Data exploration

Obtain information about the structure and contents of the database

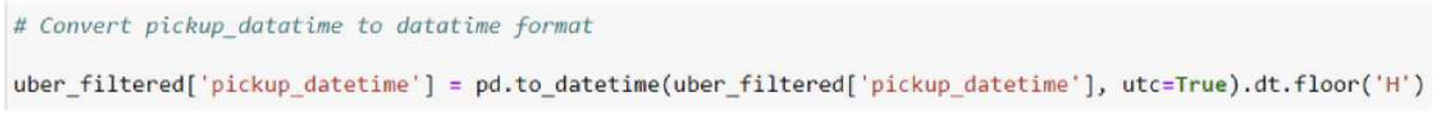
## 3 clean data up

**Original data:**
200.000 rows
**New data:**
195.795 rows

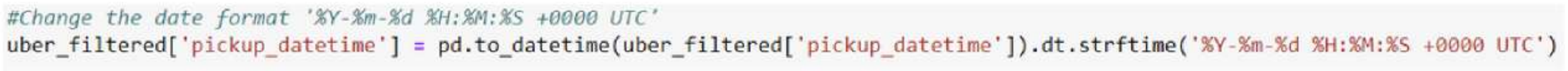


```
# Define the New York limits
lat_min = 40.477
lat_max = 45.015
lon_min = -79.7626
lon_max = -71.1851

# Create a new critearia according to New York Limits
pickup_lat = (uber_df['pickup_latitude'] >= lat_min) & (uber_df['pickup_latitude'] <= lat_max)
pickup_lon = (uber_df['pickup_longitude'] >= lon_min) & (uber_df['pickup_longitude'] <= lon_max)
dropoff_lat = (uber_df['dropoff_latitude'] >= lat_min) & (uber_df['dropoff_latitude'] <= lat_max)
dropoff_lon = (uber_df['dropoff_longitude'] >= lon_min) & (uber_df['dropoff_longitude'] <= lon_max)


uber_new = uber_df.loc[pickup_lat & pickup_lon & dropoff_lat & dropoff_lon]

# Drop any rows with null values
uber_new.dropna(how='any')
```

## 4 Standardize



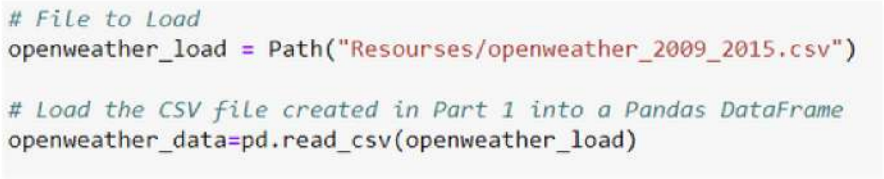
```
# Convert pickup_datatime to datatime format

uber_filtered['pickup_datetime'] = pd.to_datetime(uber_filtered['pickup_datetime'], utc=True).dt.floor('H')
```



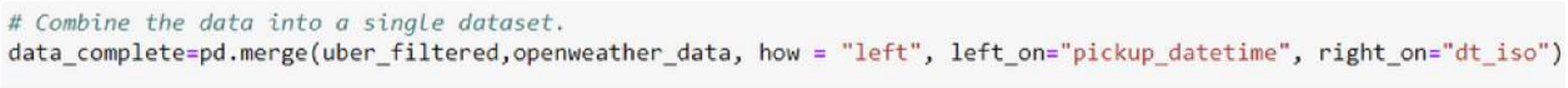
```
#Change the date format '%Y-%m-%d %H:%M:%S +0000 UTC'
uber_filtered['pickup_datetime'] = pd.to_datetime(uber_filtered['pickup_datetime']).dt.strftime('%Y-%m-%d %H:%M:%S +0000 UTC')
```

# DATA EXPLORATION & CLEAN UP PROCESS

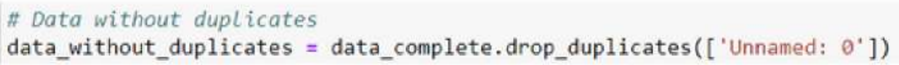
## 5 Load, merge and remove duplicates



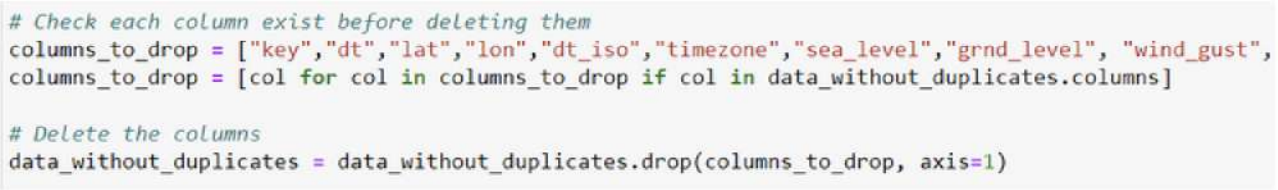
```
# File to Load
openweather_load = Path("Resourses/openweather_2009_2015.csv")

# Load the CSV file created in Part 1 into a Pandas DataFrame
openweather_data=pd.read_csv(openweather_load)
```



```
# Combine the data into a single dataset.
data_complete=pd.merge(uber_filtered,openweather_data, how = "left", left_on="pickup_datetime", right_on="dt_iso")
```

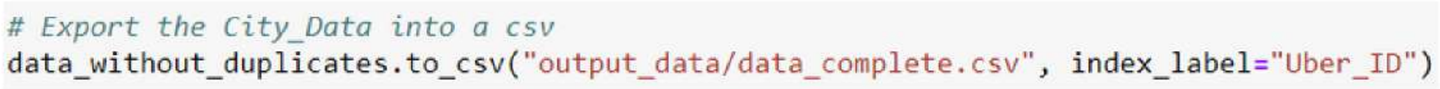


```
# Data without duplicates
data_without_duplicates = data_complete.drop_duplicates(['Unnamed: 0'])
```



```
# Check each column exist before deleting them
columns_to_drop = ["key","dt","lat","lon","dt_iso","timezone","sea_level","grnd_level", "wind_gust",
columns_to_drop = [col for col in columns_to_drop if col in data_without_duplicates.columns]

# Delete the columns
data_without_duplicates = data_without_duplicates.drop(columns_to_drop, axis=1)
```

## 6 Export a CSV

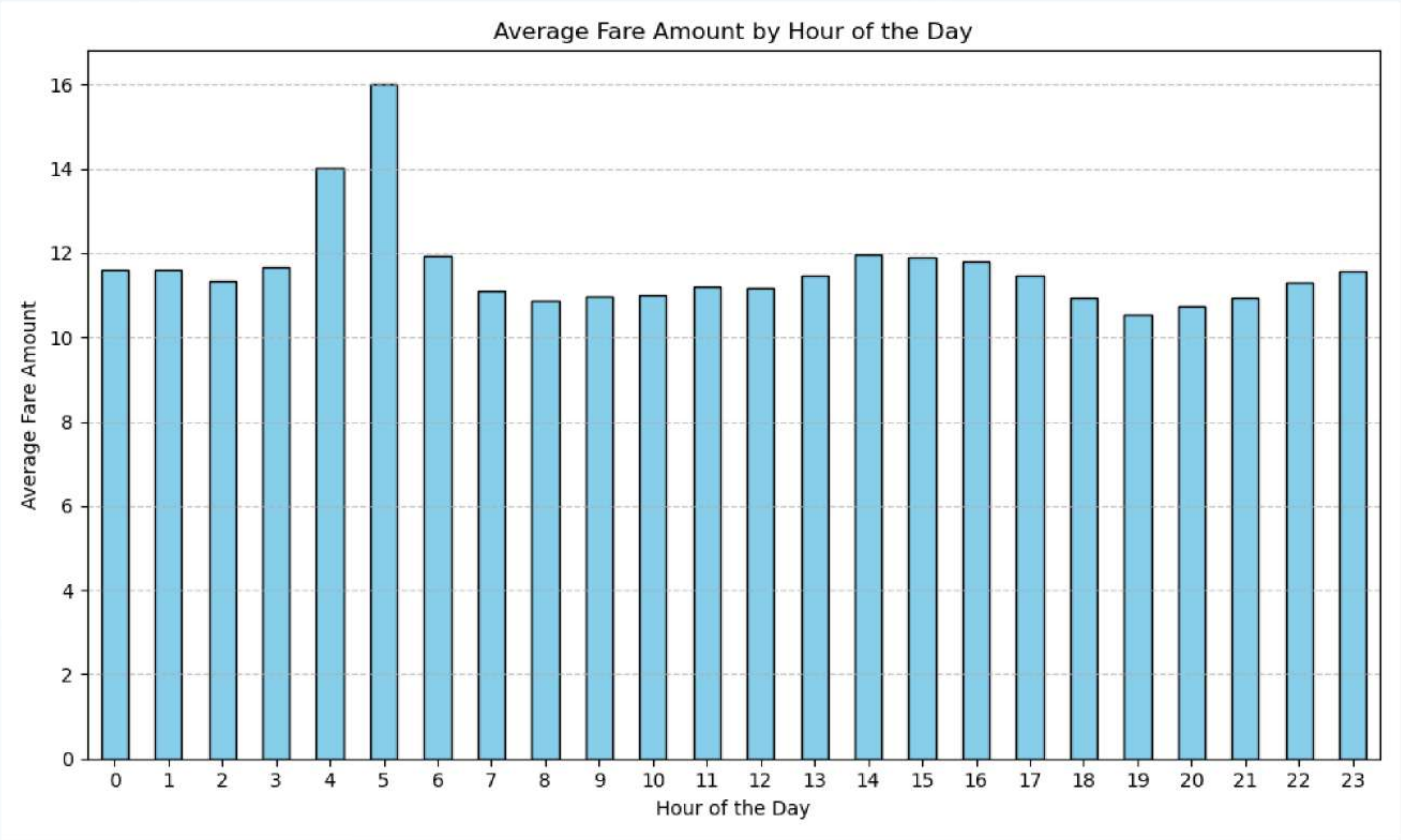


```
# Export the City_Data into a csv
data_without_duplicates.to_csv("output_data/data_complete.csv", index_label="Uber_ID")
```

# ANALYSIS PROCESS

FARE

# AVERAGE FARE AMOUNT BY HOUR OF THE DAY

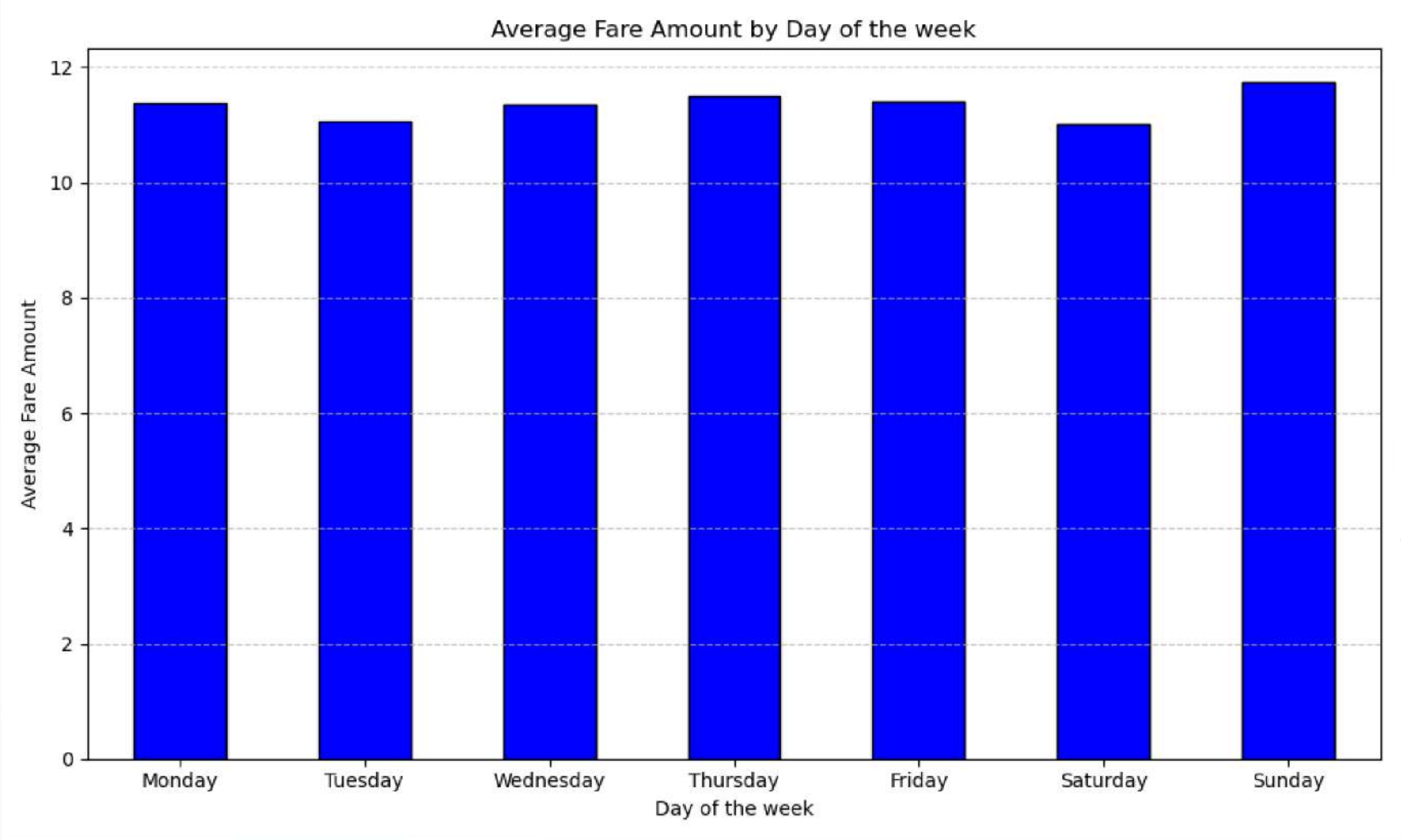


**Source:** Own elaboration.

5 am ET has the highest fare amount on average at $16

FARE

# AVERAGE FARE AMOUNT BY DAY OF THE WEEK

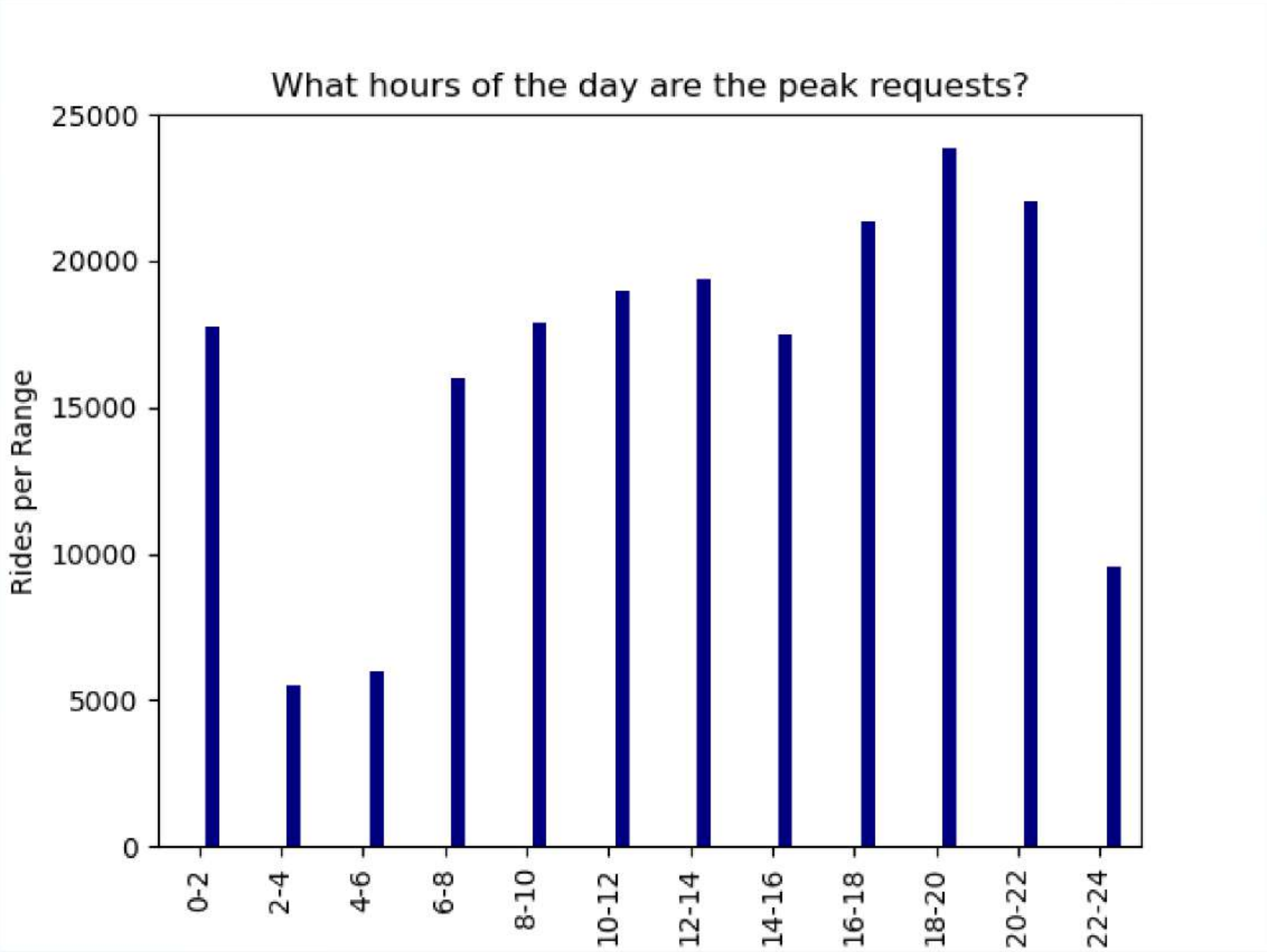


**Source:** Own elaboration.

Sundays are higher than the other bars.

Uber

# WHAT HOURS OF THE DAY ARE THE PEAK REQUESTS?



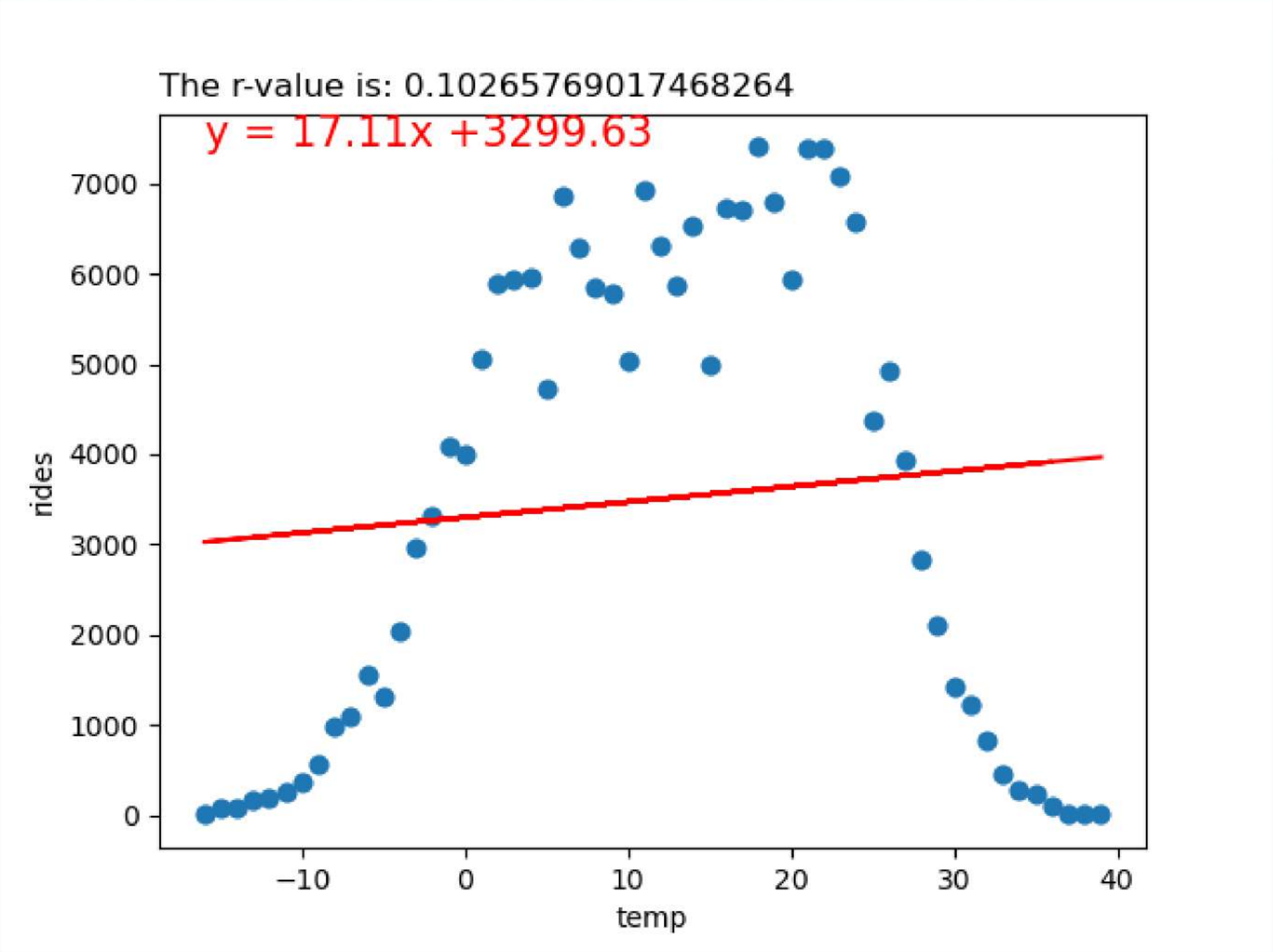
**Source:** Own elaboration.

We found that from 4 pm (16:00) to 11 pm (22:00) are the hours with the highest number of Uber ride requests. However, the time slot with the highest number is between 6 pm (18:00) and 8 pm (20:00)

Uber

WEATHER

# CORRELATION BETWEEN RIDES & TEMPERATURE

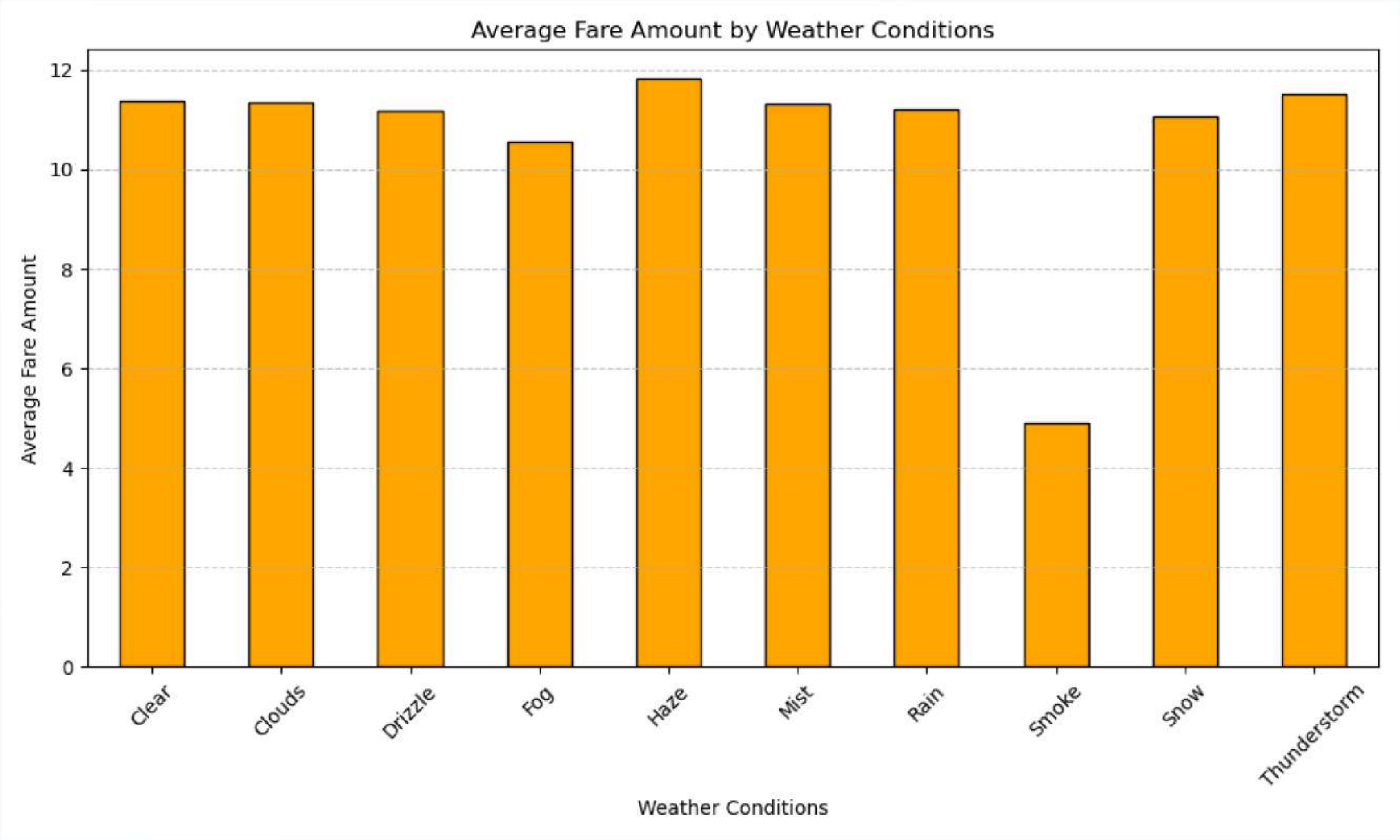


**Source:** Own elaboration.

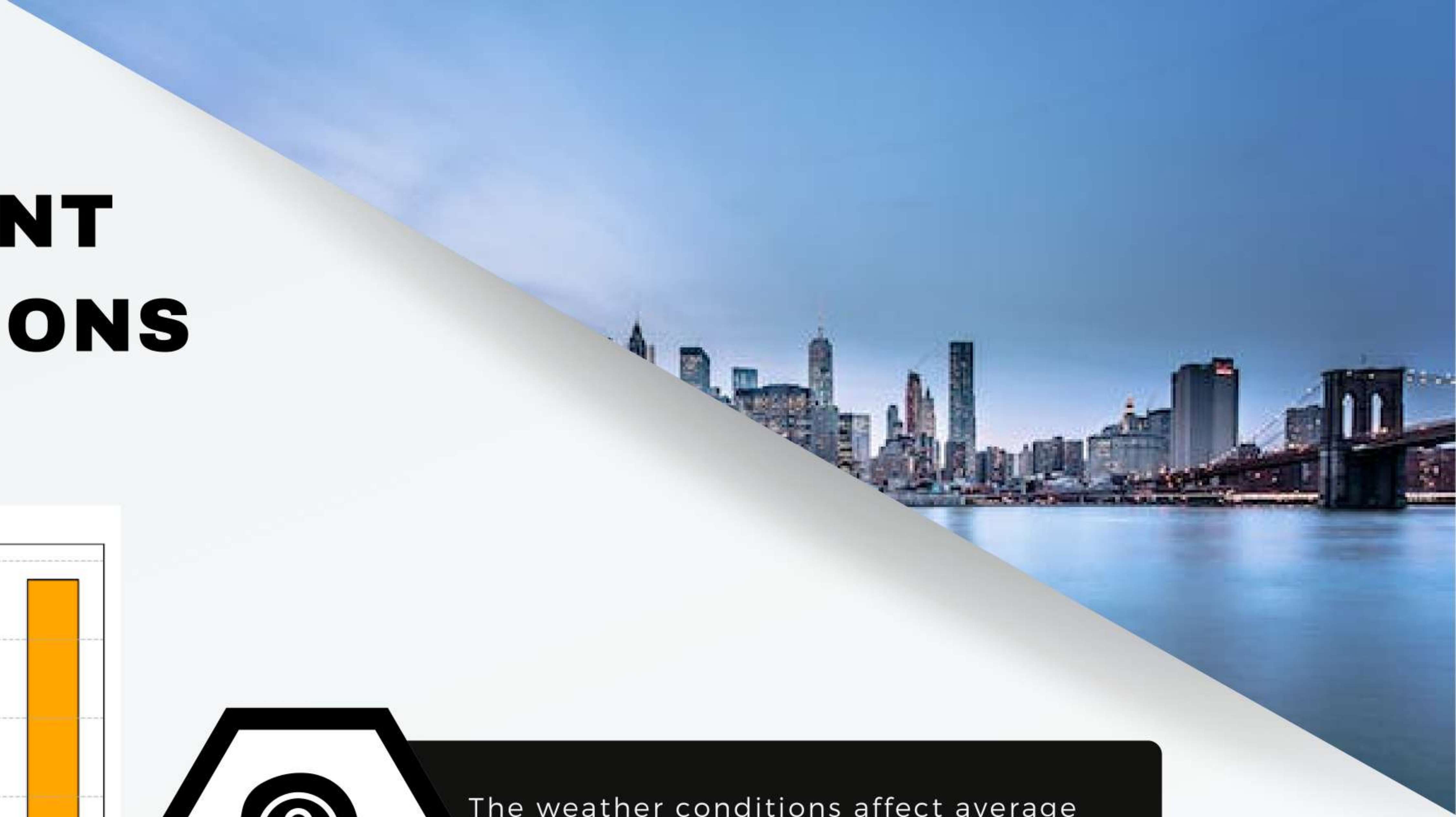
There is no significant linear relationship between rides and temperature

Uber

# AVERAGE FARE AMOUNT BY WEATHER CONDITIONS



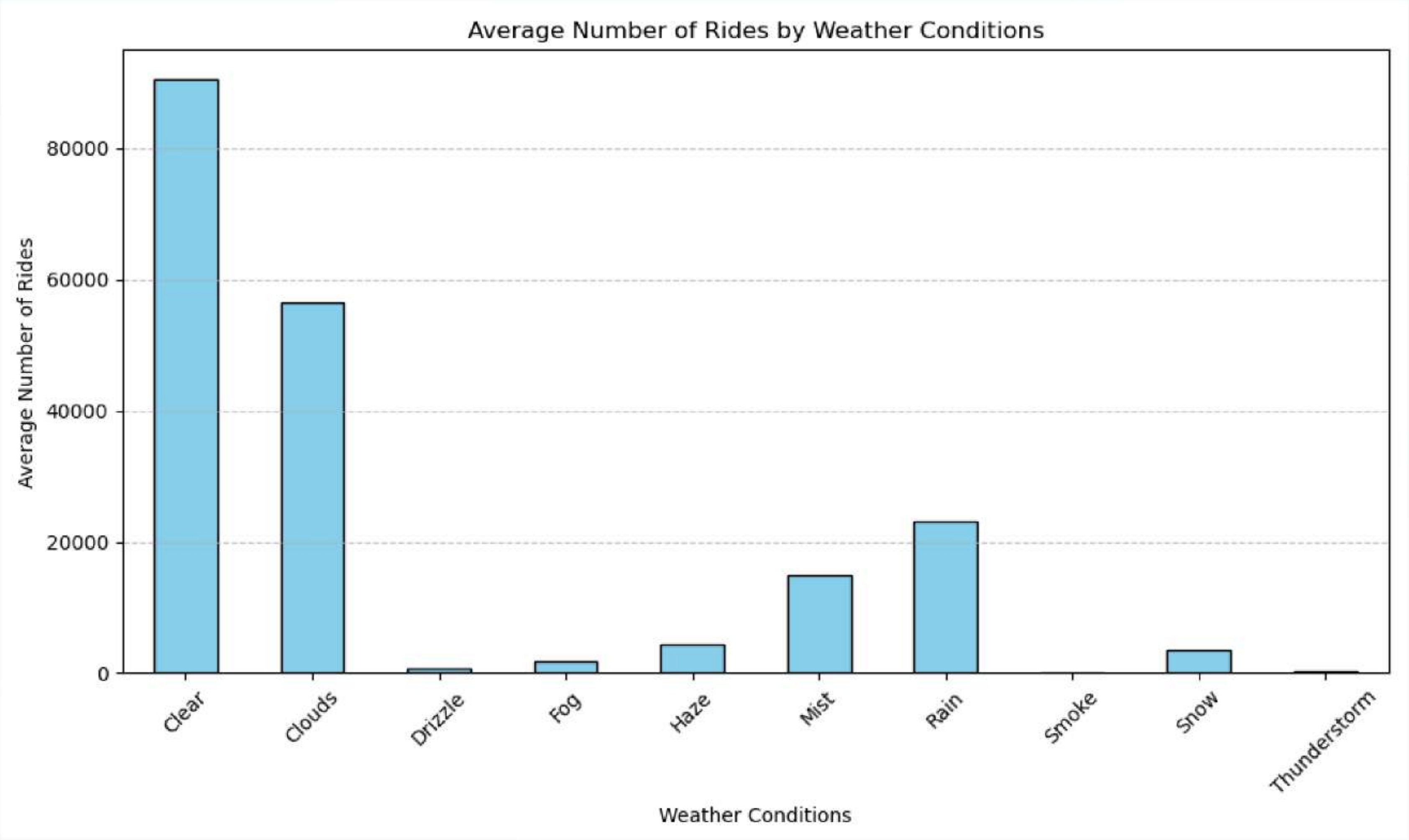
**Source:** Own elaboration.



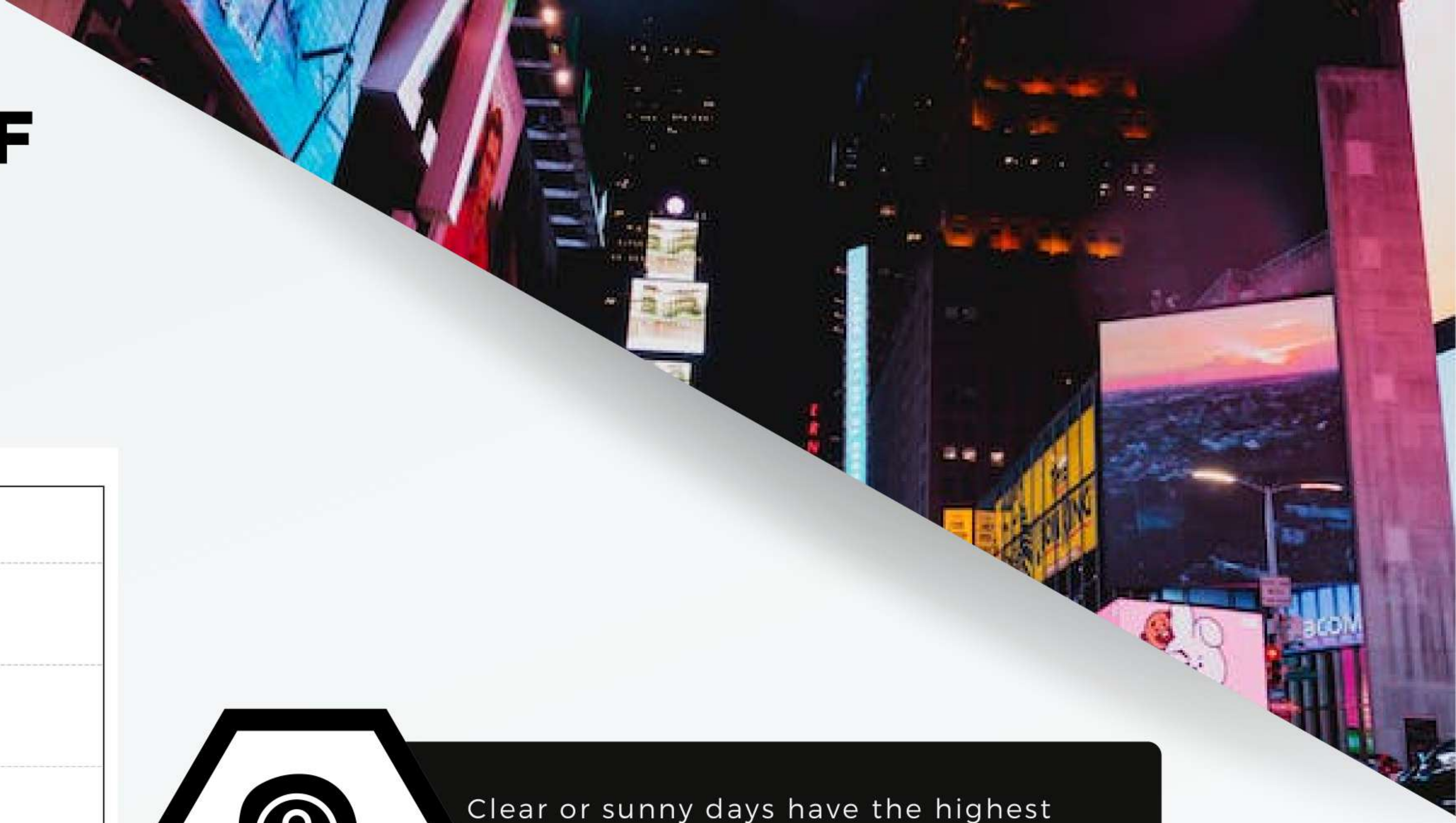
The weather conditions affect average fare amounts on each given day with Haze weather having the highest average fare amount

WEATHER

# AVERAGE NUMBER OF RIDES BY WEATHER CONDITIONS



**Source:** Own elaboration.

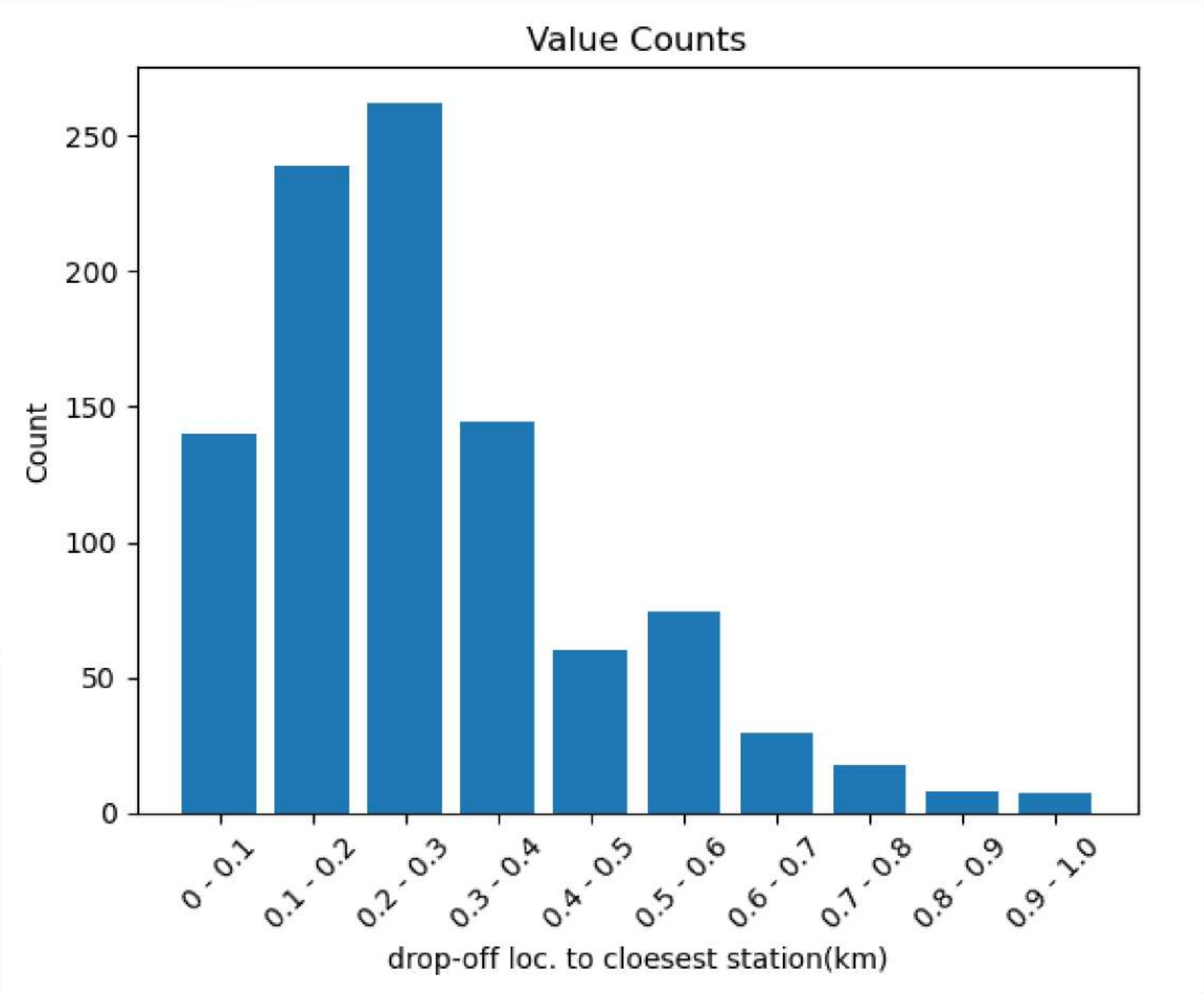


Clear or sunny days have the highest number of rides

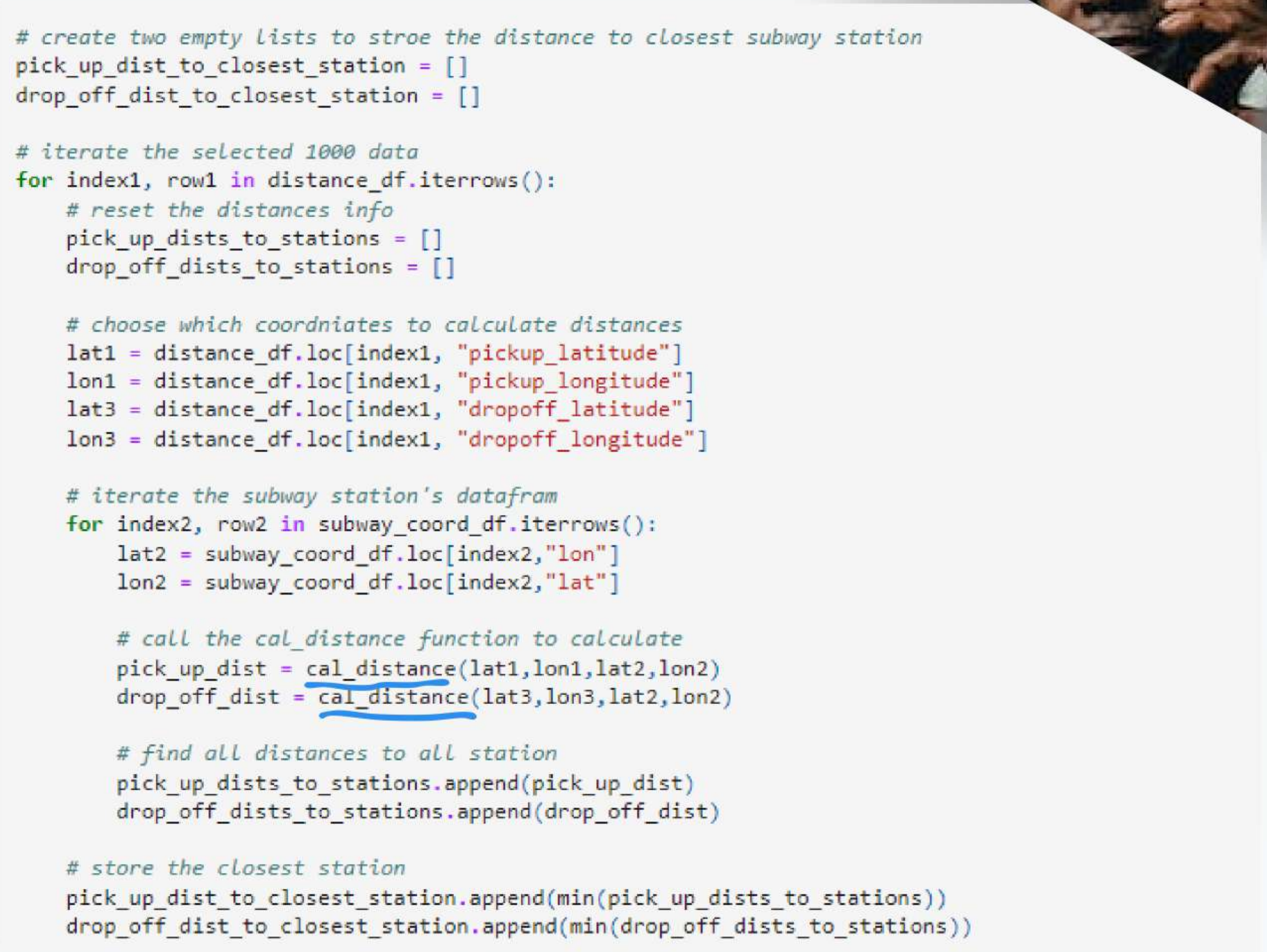
Uber

# DESTINATION

Relationship between disances to the closest subway station. Most people who take Uber go the places about 100-300 meters away from the closest the station.



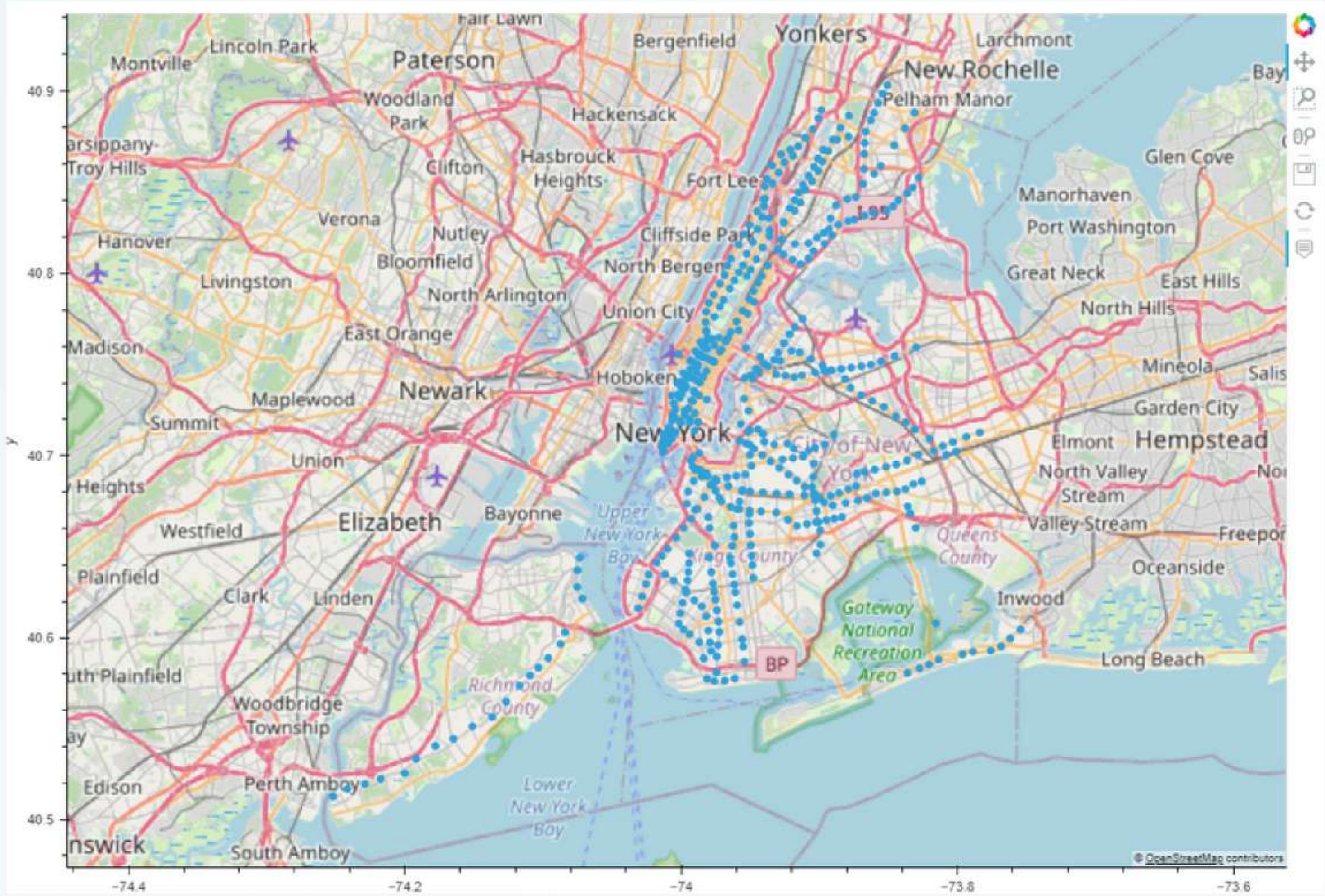
At peak hour: 18:00-20:00
Sample: 1000



```
# create two empty lists to stroe the distance to closest subway station
pick_up_dist_to_closest_station = []
drop_off_dist_to_closest_station = []

# iterate the selected 1000 data
for index1, row1 in distance_df.iterrows():
    # reset the distances info
    pick_up_dists_to_stations = []
    drop_off_dists_to_stations = []

    # choose which coordniates to calculate distances
    lat1 = distance_df.loc[index1, "pickup_latitude"]
    lon1 = distance_df.loc[index1, "pickup_longitude"]
    lat3 = distance_df.loc[index1, "dropoff_latitude"]
    lon3 = distance_df.loc[index1, "dropoff_longitude"]

    # iterate the subway station's datafram
    for index2, row2 in subway_coord_df.iterrows():
        lat2 = subway_coord_df.loc[index2,"lon"]
        lon2 = subway_coord_df.loc[index2,"lat"]

        # call the cal_distance function to calculate
        pick_up_dist = cal_distance(lat1,lon1,lat2,lon2)
        drop_off_dist = cal_distance(lat3,lon3,lat2,lon2)

        # find all distances to all station
        pick_up_dists_to_stations.append(pick_up_dist)
        drop_off_dists_to_stations.append(drop_off_dist)

    # store the closest station
    pick_up_dist_to_closest_station.append(min(pick_up_dists_to_stations))
    drop_off_dist_to_closest_station.append(min(drop_off_dists_to_stations))
```

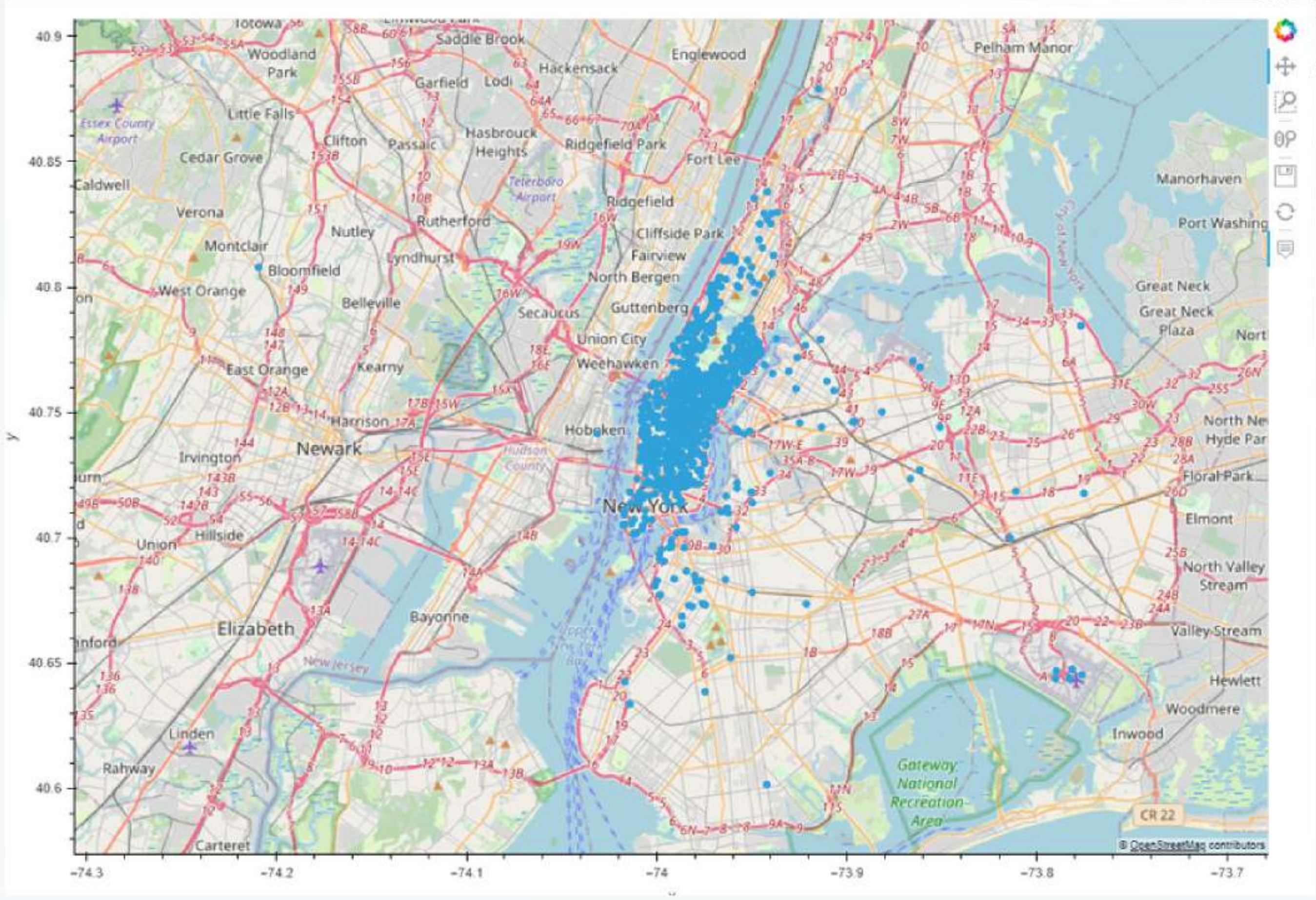
**Source:** Own elaboration.

# Maps in NYC



Subway Stations in NYC

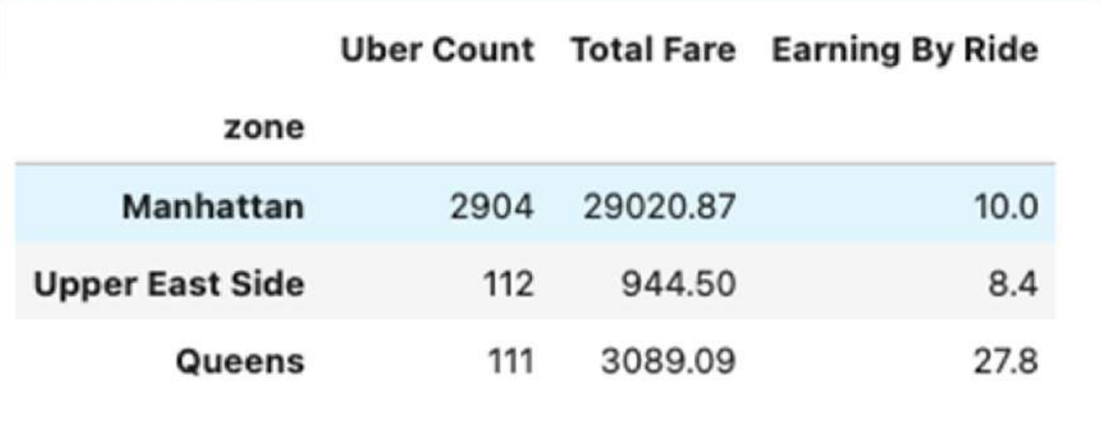
Selected 1000 Samples of Drop-off locations



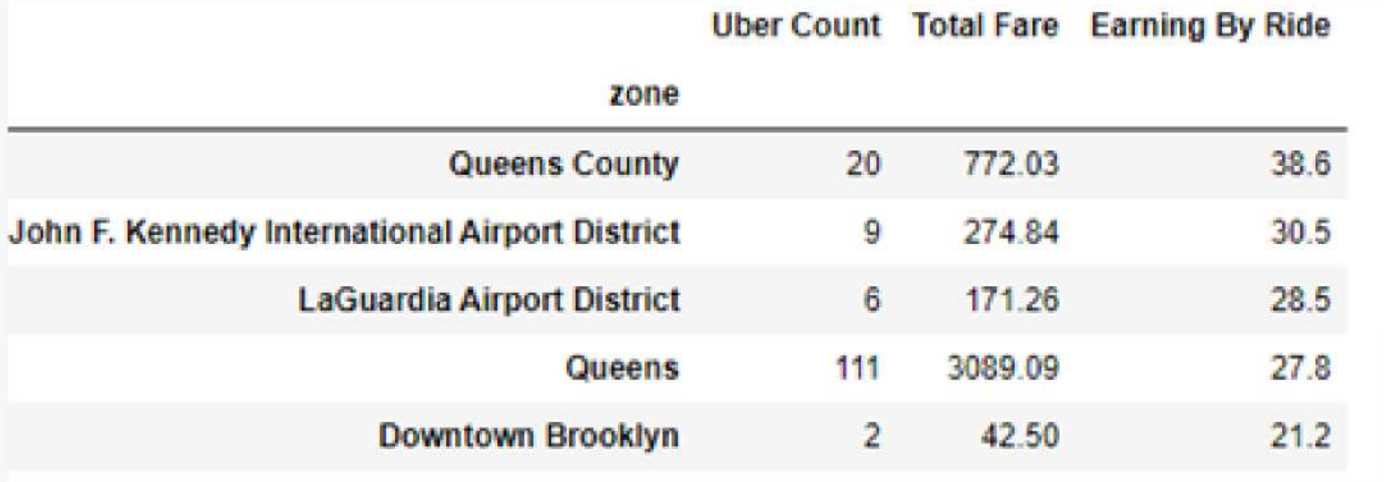
**Source:** Own elaboration.

# NUMERICAL SUMMARY & VISUALIZATION OF SUMMARY

The zone with the most ride requests is Manhattan; however, Queens county is the zone with the best fare per ride, close to $40 USD.

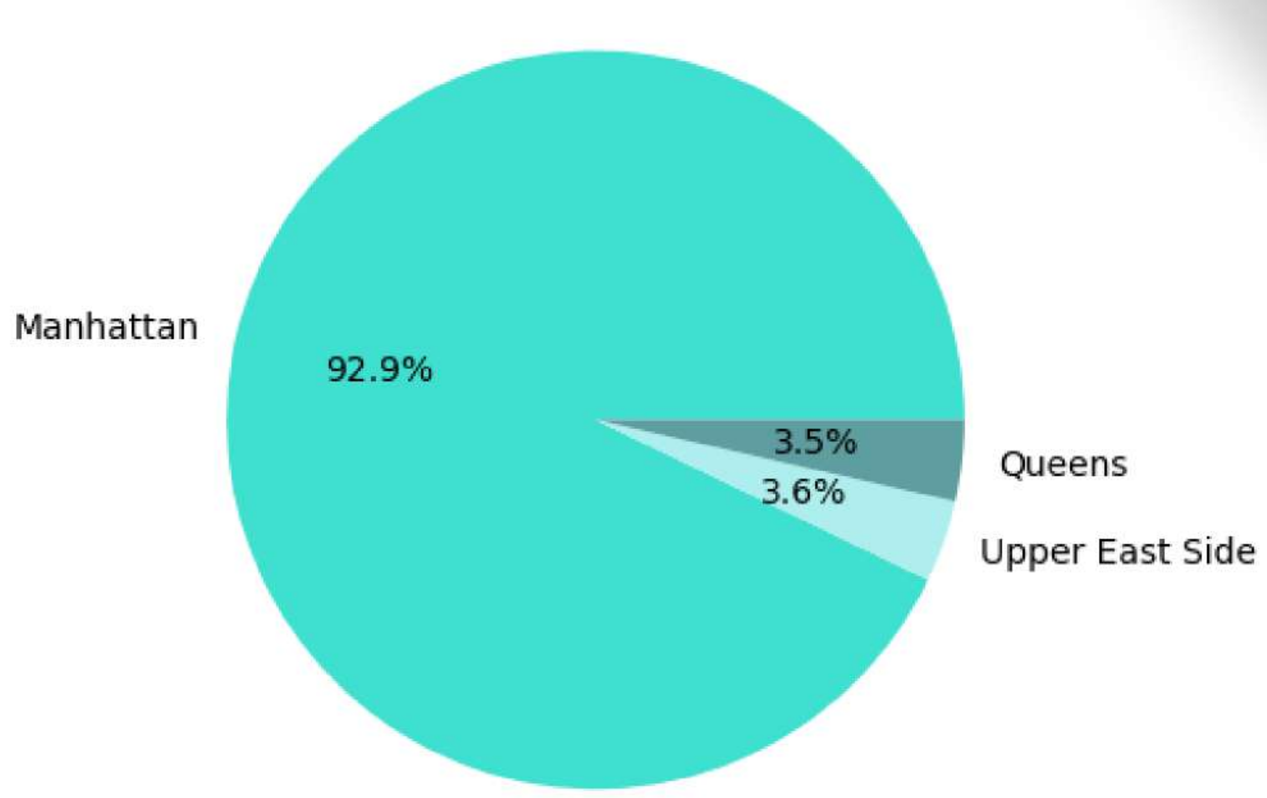


| zone | Uber Count | Total Fare | Earning By Ride |
|---|---|---|---|
| Manhattan | 2904 | 29020.87 | 10.0 |
| Upper East Side | 112 | 944.50 | 8.4 |
| Queens | 111 | 3089.09 | 27.8 |



| zone | Uber Count | Total Fare | Earning By Ride |
|---|---|---|---|
| Queens County | 20 | 772.03 | 38.6 |
| John F. Kennedy International Airport District | 9 | 274.84 | 30.5 |
| LaGuardia Airport District | 6 | 171.26 | 28.5 |
| Queens | 111 | 3089.09 | 27.8 |
| Downtown Brooklyn | 2 | 42.50 | 21.2 |

Zone with the highest requests in the request peak hour



**Source:** Own elaboration.

# IMPLICATION OF FINDINGS

- No linear regression between temperature and the rides
- On rainy days, people seem to take less rides
- Manhattan is the destination with the most rides
- Peak time for rides is between 6pm - 8pm ET which is rush hour

Uber