NYC TAXI TRIPS STUDY USING SPARK SQL

Gonzalo Lencina Lorenzón
Víctor Lavado Campos
Fernando de Elena Escaladas

High-performance computing for Big Data in companies
Master in Big Data Analytics

Contents

1. Introduction
2. Average speed of taxis in terms of the hour
3. Most common taxi trips
4. Average price per passenger in terms of the final rate code
5. Average tip per passenger in terms of the number of passengers
6. Price variation for similar trips to JFK airport
7. Data metrics

1. Introduction

For this project, Apache Spark SQL is the program used to extract and process useful information, to later draw conclusions in graphical formats on various aspects of the green and yellow taxi trip records, obtained from the NYC Taxi and Limousine Commission. The data is read from a CSV file where the main columns are pick-up and drop-off dates, times and locations, trip distances, itemized fares, rate types, payment types and driver-reported passenger counts.

Certain queries are applied on this data to extract interesting information. This information is analysed, processed and reflected in the form of a graph, so that it is visually easier to interpret. In addition, for each query performed, apart from the graph, an explanation is provided in which an analysis of the results is performed.

Moreover, for some of the queries, the Taxi Zone Lookup Table that relates the location ID and the actual zone name is used in the graphs, also provided by the NYC Taxi and Limousine Commission.

Finally, a table summarizing the execution time, amount of data processed, and processing speed of each query is presented at the end of the document.

2. Average speed of taxis in terms of the hour

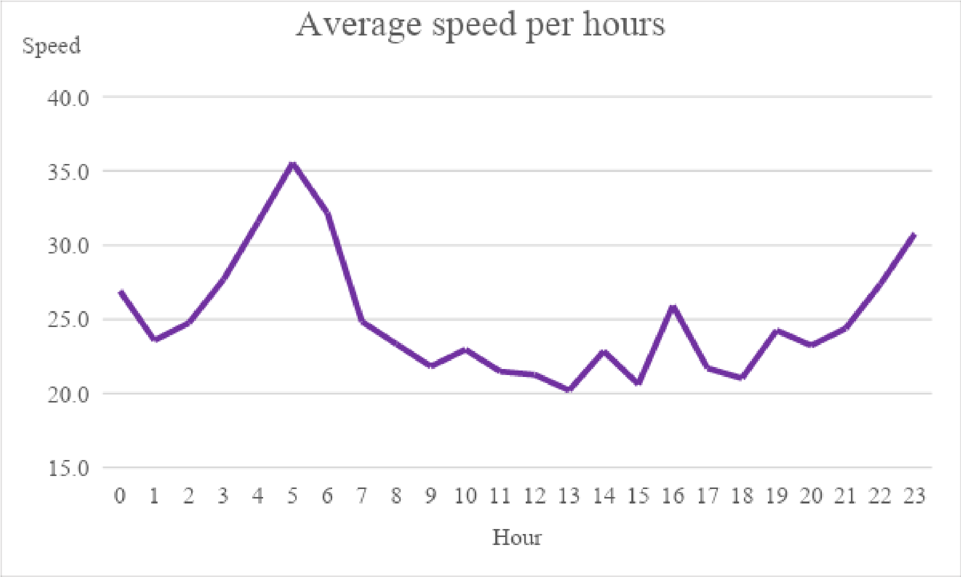

Figure 1: Average speed per hours.

```
val target1 = sql("select hour(tpep_dropoff_datetime) as hour,
avg(((trip_distance)*1.609)*3600/(unix_timestamp(tpep_dropoff_datetime) -
unix_timestamp(tpep_pickup_datetime))) as speed
from df1 where trip_distance is not null and tpep_pickup_datetime is not null and tpep_dropoff_datetime is not null
and tpep_pickup_datetime <> tpep_dropoff_datetime group by hour")
```

This first study shows the average speed of the taxis depending on the hours. Speed is simply calculated dividing distance by time. Moreover, time is calculated subtracting the drop off and pick up times, which gives the total time of the trip. The hours are extracted from the drop off time.

Results can be seen in *Figure 1*. During night time, average speed is considerably high, with the highest peaks in the very early morning (at 5am), coinciding with the time when there is less traffic on the roads, in contrast to the central hours of the day, especially between 9am and 6pm.

It is interesting to see that during rush hours (usually considered to be from 7-9am and 4-6pm), the average speed is not as low as expected, which may be a consequence of being more in a hurry than during the rest of the day.

3. Most common taxi trips

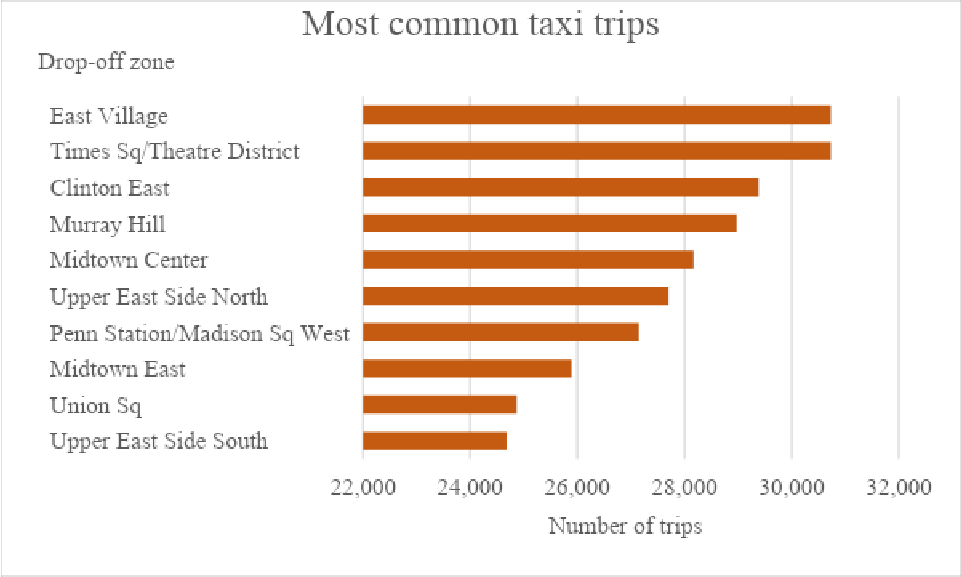

Figure 2: Most common taxi trips.

```
val target2 = sql("select DOLocationID, count(DOLocationID) from record group by DOLocationID")
```

The object of study in this part are the most common taxi trips, depending on the drop off zone. In this case, the number of trips ending in different zones are counted.

As it can be seen in *Figure 2*, most repeated taxi trips end in the heart of Manhattan. In fact, all top 10 shown in the graph are Manhattan zones. It really makes sense that East Village and Times Square/Theatre District are the two most common taxi trips, considering that the first zone includes the most known traditional clubs and pubs, as well as art galleries and famous restaurants, and the second includes Broadway theatres, Times Square and numerous giant shops.

4. Average price per passenger in terms of the final rate code

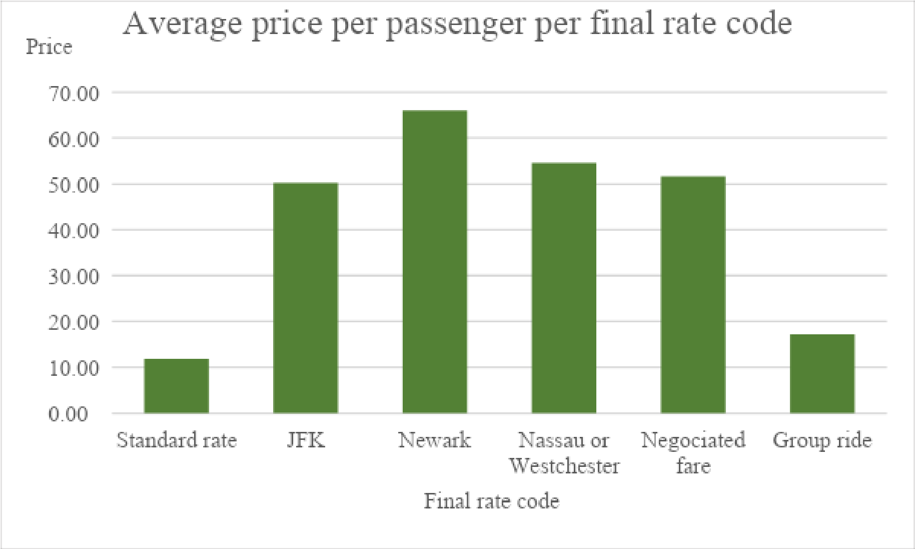

Figure 3: Average price per passenger per final rate code.

```
val target3 = sql("select RateCodeID as end, avg(Total_amount/Passenger_count) as individual_amount from df1
where RateCodeID is not null and Total_amount is not null and Passenger_count is not null group by end")
```

This study shows the average price paid per passenger depending on the final rate code. Average price per passenger is calculated dividing the total amount paid by the number of passengers in the taxi, grouping the results by the rate code.

Results show different things: firstly, going to the JFK airport or to any of the more distant areas in New York (Newark, in New Jersey, Nassau or Westchester) increases considerably the price per passenger.

Secondly, negotiated fares are made usually in long distances, that is why the price of this element is also one of the highest.

Lastly, both standard rates and group rides have the lowest average price per passenger, but for two different reasons. The reason for the first group is because distance is obviously much lower inside the different neighbourhoods in New York City than going to any of the other surrounding areas with a different final rate code. However, the second group reduces its average price per passenger because it is a group ride and the cost of the trip is divided.

5. Average tip per passenger in terms of the number of passengers

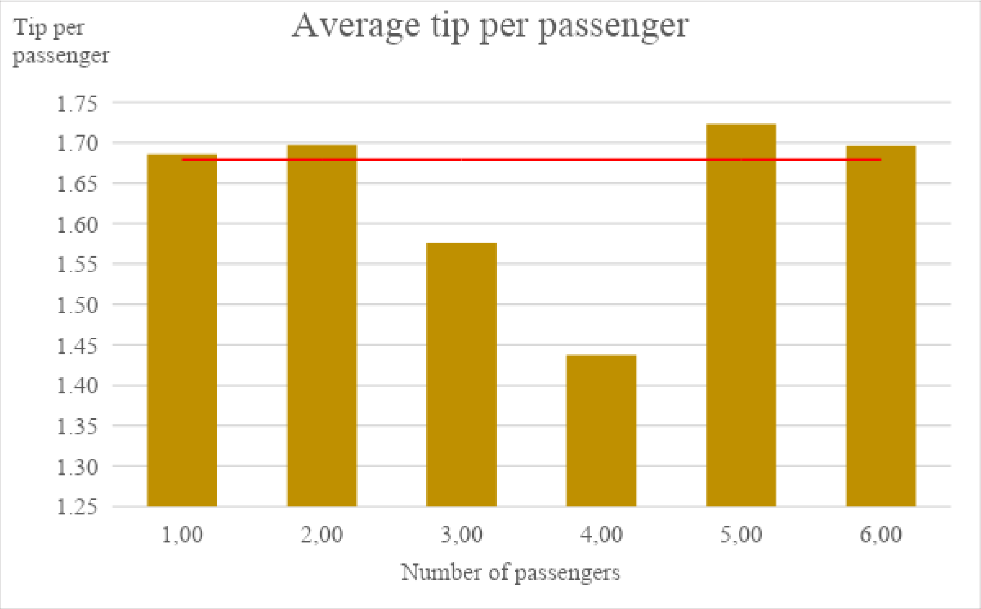

Figure 4: Average tip per passenger.

```
val target4 = sql("select Passenger_count as Passenger, avg(Tip_amount) as Avg_tip, count(Passenger_count) from
df1 where Passenger_count is not null and Tip_amount is not null and Passenger_count <> 0 group by Passenger")
```

This study shows interesting results related to human behaviour when the time of tipping comes. Average tip per passenger has been calculated using the tip amount divided by the number of passengers in the taxi.

The number of observations for each number of passengers is also included in the query. We observed that for 7 passengers and above the number of observations is so low that they may not be significant. Thus, they are not included in *Figure 4.*

From the bar graph we can extract several conclusions: average tip per passenger is slightly higher when there are two passengers rather than just one, maybe because of feeling the pressure of what the other person thinks.

The average tip decreases when there are 3 and 4 passengers, but then it goes above the average (the red line) again when having 5 and 6 passengers. The reason for these is because the jump between 4 and 5 passengers represents the difference between having a normal taxi or a bigger one. Individuals realise this distinction and tip more on average when using a bigger taxi.

6. Price variation for similar trips to JFK airport

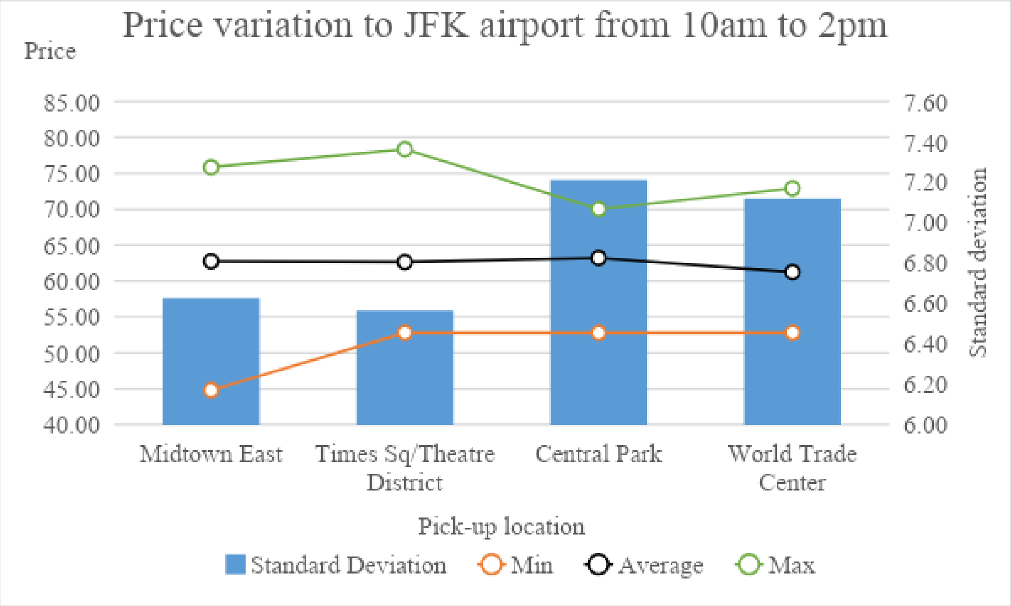

Figure 5: Price variation to JFK airport from 10am to 2pm.

```
val target5 = sql("select PULocationID, min(total_amount), max(total_amount), avg(total_amount),
stddev(total_amount) from df1 where DOLocationID = 132 and total_amount is not null and PULocationID in (43,
162, 230, 261) and hour(tpep_dropoff_datetime) between 10 and 13  group by PULocationID")
```

This last study examines the variation in price between trips to the JFK airport from four popular places in New York: Midtown East, where Grand Central Terminal is located, Times Square, Central Park and World Trade Center.

In order to observe this variation, several parameters are extracted from the data: minimum, average and maximum values for each of the trips to the JFK airport, as well as the standard deviation (in the right vertical axes in *Figure 5*). As these values have a broad range depending on the hours, we have chosen a short period of time, from 10am to 2pm, to avoid increasing the variance if considering rush hours (traffic increases the price) or night times (with night fares), among others.

The results in *Figure 5* show a surprising evidence: there is a $20-30 difference in price between similar trips to JFK airport, from any of the pickup locations. Among the four locations, Central Park is the starting location with a higher standard deviation, although the difference between maximum and minimum values is the lowest.

The conclusion that can be drawn is that taxis charge really varying prices for similar trips. However, there are other variables apart from the time that cannot be studied with this database and affect the trip price, such as the weather, accidents or any other event that influence the traffic.

7. Data metrics

Finally, here is a table with the data metrics obtained by accessing to the host at the Spark context Web UI, showing execution time, amount of data and processing speed for each one of the queries:

Task	**Execution time**	**Amount of data**	**Processing speed**
Reading	0,6 s	64.0 KB	106.7 KB/s
Query 1	17 s	81.9 MB	4.8 MB/s
Query 2	10 s	81.9 MB	8.2 MB/s
Query 3	5 s	81.9 MB	16.4 MB/s
Query 4	4 s	81.9 MB	20.5 MB/s
Query 5	6 s	81.9 MB	13.7 MB/s