The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Data @ Scale

COURSEWORK, 2019/20

TAHSINUR KHAN

Contents

1. KPI Definitions

KPI Description (in words): `gid` vs. `fare_amount`

KPI formula: sum of the fare amount per pickup zone

Steps to realize KPI:

```
select zone_id_pickup, sum(fare_amount) as "Total earnings in Zone"
from cabs_gid_cleaned
--where pickup_datetime >= '2015-10-01 00:00:00' AND  pickup_datetime <
'2015-12-31 23:59:59'
group by zone_id_pickup
order by "Total earnings in Zone" desc;
```

Additional Notes: have to clean the data table first and obtain cabs_gid_cleaned table (see das_cw-1.sql file)

KPI Description (in words): `borrocde` and `zone_id_pickup` vs. `passenger_count`

KPI formula: sum of passenger count per borrocode and pickup zone group

Steps to realize KPI:

```
select cabs_gid_cleaned.zone_id_pickup, nymc_gid.borocode,
sum(cabs_gid_cleaned.passenger_count) as "Total passenger count per
zone/borocode combination"
from cabs_gid_cleaned join nymc_gid
on cabs_gid_cleaned.zone_id_pickup = nymc_gid.gid
group by cabs_gid_cleaned.zone_id_pickup, nymc_gid.borocode
order by "Total passenger count per zone/borocode combination" desc;
```

Additional Notes: have to clean the data table first and obtain cabs_gid_cleaned table (see das_cw-1.sql file)

KPI Description (in words): `zone_id_dropoff` and `borocode` vs. `tip amount`

KPI formula: total tip amount per dropoff zone and borocode group

Steps to realize KPI:

```
select cabs_gid_cleaned.zone_id_dropoff, nymc_gid.borocode,
sum(cabs_gid_cleaned.tip_amount) as "Total tip per zone/borocode
combination"

from cabs_gid_cleaned join nymc_gid

on cabs_gid_cleaned.zone_id_dropoff = nymc_gid.gid

group by cabs_gid_cleaned.zone_id_dropoff, nymc_gid.borocode

order by "Total tip per zone/borocode combination" desc;
```

Additional Notes: have to clean the data table first and obtain cabs_gid_cleaned table (see das_cw-1.sql file)

KPI Description (in words): Most profitable routes (zones) based on total earnings and total number of trips available

KPI formula: total fare amount, total trip distance and total number of trips per pickup zone and drop off zone group

Steps to realize KPI:

```
create table KPI4 as

select zone_id_pickup, zone_id_dropoff, sum(fare_amount) as
"Total_earnings",

           sum(trip_distance) as "Total_distance_travelled",

           count(*) as "total_number_trips_in_route"

from cabs_gid_cleaned

group by zone_id_pickup, zone_id_dropoff

order by "Total_earnings" desc;


--drop table kpi4;

--select * from KPI4
```

```
-- Adding total_earnings_per_km column to kpi4:

ALTER TABLE KPI4 ADD COLUMN total_earnings_per_km numeric;

UPDATE kpi4

SET "total_earnings_per_km" = "Total_earnings" /
("Total_distance_travelled" / 1000);


-- Adding total_earnings_per_trip column to kpi4:

ALTER TABLE KPI4 ADD COLUMN total_earnings_per_trip numeric;

UPDATE kpi4

SET "total_earnings_per_trip" = ("Total_earnings" /
"total_number_trips_in_route");


--

select * from kpi4

order by "total_earnings_per_trip" desc;
```

Additional Notes: have to clean the data table first and obtain cabs_gid_cleaned table (see das_cw-1.sql file) and have to make an extra table: kpi4

KPI Description (in words): Most important time to drive based on total number of trips (and total earnings)

KPI formula: total number of trips and total fare amount per hour of the day

Steps to realize KPI:

```
select date_part('hour', pickup_datetime) as trip_hour,

          count(*) as total_trips,

          sum (fare_amount) as total_earnings

from cabs_gid_cleaned

where (zone_id_pickup = 1 AND (zone_id_dropoff = 1 OR zone_id_dropoff = 9
OR zone_id_dropoff = 14))

      OR (zone_id_pickup = 9 AND (zone_id_dropoff = 1 OR zone_id_dropoff
```

```
= 9 OR zone_id_dropoff = 14))
      OR (zone_id_pickup = 14 AND (zone_id_dropoff = 1 OR zone_id_dropoff
= 9 OR zone_id_dropoff = 14))
group by trip_hour
order by trip_hour;
```

Additional Notes: have to clean the data table first and obtain cabs_gid_cleaned table (see das_cw-1.sql file)

2. Executive Summary

For this study, I am interested in examining the most optimized routes for taxicab drivers in New York in order to maximize their income.

Key Performance Indicators:

After the initial data cleaning, we used the cabs_gid_cleaned and the nymc_gid data tables for implementing our KPIs. The five KPIs we implemented were:

- KPI 1: gid vs. fare_amount- Gives us the full earnings potential of each specific region
- KPI 2: borrocde and zone_id_pickup vs. passenger_count- Offers insight into which boroughs have the most customers
- KPI 3: zone_id_dropoff and borocode vs. tip amount- We are interested to see the zonal demographics that are the highest tippers
- KPI 4: Most profitable routes (zones) based on total earnings and total number of trips available
- KPI 5: Most important time to drive based on total number of trips (and total earnings)

Our primary indictor will include gid which we will use with fare_amount to get first look into which regions have the most available in terms of total earnings that can be realized and since we are restricted to a maximum of three zones. We also measured borrocode and match this with zode_id_pickup and passenger_count as this will indicate the busiest sections of the the major boroughs. Downtime is one of the significant causes for loss of earnings amongst taxi drivers. We will also need to look at significant regions with the busiest routes, therefore, we will match gid with zone_id_pickup to get a sense of the busiest routes on top of the highest grossing regions so as to be able to identify where our cabbies must be in order to maximize our output. The last indicator will be tip_amount which represents the total tip earned in cash by the cab drivers. We will juxtapose the tip_amount to the borrocode as customers in regions such as Manhattan and Brooklyn tend to tip considerably higher than other boroughs as they have significantly higher disposable income [1]. Lastly we look towards pickup_datetime which we will use to filter our busiest locations with the exact timing that he should be spending on the road [2]. For example, one of the prime earning hours are often weekday rush hours, we will look to identify the key times that are busy and match them to specific locations to get a good understanding of where the cabbie should be located.

Our results indicate that the highest earning zone is zone 9 with potential earnings of $69 million, followed by 14 then 1 and 3 comes in at last with less than a million in potential earnings. Moreover, we also find that routes 9 and 14 which serve borough 1 (Manhattan) has the highest number of customers at 13 and 4 and a half million customers. We also hypotehsized that passengers hailing rides from zones 9 and 14 which represent Manhattan tip the highest as well at over $8 million and $3 million respectively. Digging deeper we find that our highest grossing pick-up and drop off zones involve a combination of zones 9, 14 and 1 which are representative of Manhattan and Brooklyn where individuals have more disposable income to spend on transportation. Lastly, examining time stamps we find that the most profitable hours are between 12 noon and 8 pm indicating that the demand is highest for taxis between lunch hours and

evening rush hours when people return home from work with the lowest being around 5 am when there is much less foot traffic and far less transportation.

3. Comparative Analysis:

3.1. Data Exploration and Cleaning:

Before starting off writing the queries, we put the data tables under a microscope to see what sort of patterns they revealed.

We started off my counting the number of trips in the entire **`cabs`** table, which came out to be **85,142,944** rows. However, for our analysis we were only asked to cover regions **1, 3, 9 and 14**. Therefore, we filtered out the table to only contain records of taxi trips in those regions only. The `zone_id_pickup` and `zone_id_dropoff` were used for the purposes of filtering off the data based on regions. Therefore, we were left with records pertaining to 16 zone combinations in total –i.e. four destination zones for each of the pickup zones. This process reduced the number of rows to **11,049,322.**

After the initial filtering of data based on regions, we searched for any data discrepancy that might be present in the Cabs table. One area of inspection was to check the column values from `fare_amount` to `tolls_amount` and adding them up to see if they equaled to the `total_amount` column values. Ideally, there shouldn't be any difference between the values in the `total_amount` column and summation of the five columns from `fare_amount` to `tolls_amount`. Following this convention reduced the number of rows **10,950,836**.

Further checking was carried out to see any numeric or int datatype columns had any negative numbers or not. According to the data dictionary, any negative values in those columns would render the definition of those columns meaningless. E.g., there cannot be a negative value in the `total_amount` columns, as it would mean that a customer was charged negative value!

Moreover, the `TIMESTAMP` datatype columns – `pickup_datetime` and `dropoff_datetime` was also checked separately to make sure that `dropoff_time` >= `pickup_datetime`, because anything otherwise would not make any sense.

There was a further date constraint in our analysis. We were asked to analyze the **October to December** period. However, after filtering and cleaning the table, we found that the dates ranged from **2015-01-01** to **2015-06-30**. Hence, there were no instances in the specified range. As a result the whole 6 month period from Jan 1st to June 30th, 2015 were taken into consideration for our analysis.

Finally, after all the cleaning, the Cabs data table was checked for any null values and any rows containing any null values were removed. The final table came out to be **10,950,836** rows long and was named **`cabs_gid_cleaned.`**

Similar procedure was carried out on the **`nymc`** table and the processed table was named **`nymc_gid.`**

3.2. Key Performance Indicators:

After the initial data cleaning, we used the `cabs_gid_cleaned` and the `nymc_gid` data tables for implementing our KPIs. The five KPIs we implemented were:

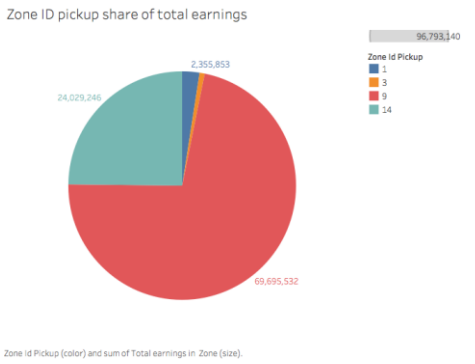
- KPI 1: `gid` vs. `fare_amount`
- KPI 2: `borrocde` and `zone_id_pickup` vs. `passenger_count`

- KPI 3: `zone_id_dropoff` and `borocode` vs. `tip amount`
- KPI 4: Most profitable routes (zones) based on total earnings and total number of trips available
- KPI 5: Most important time to drive based on total number of trips (and total earnings)

3.2.1. KPI 1: gid vs. fare_amount

We are interested in examining the most optimized routes for taxicab drivers in New York in order to maximize their income. Our primary indictor will include `gid` which we will use with `fare_amount` to get first look into which regions have the most available in terms of total earnings that can be realized, since we are restricted to a maximum of three zones.

Figure 1 shows a pie chart with the corresponding table of the result of grouping the total earnings based on the pickup zone of the passengers. From this we can get an idea that the most profitable region would be 9, followed by 14, 1 and finally 3.



KPI 1: Zone_id_pickup vs. Total earning from the zone	
zone_id_pickup	Total earnings in Zone
9	69695532.17
14	24029245.59
1	2355852.99
3	712509.55

Figure 1: Breakdown of total earnings based on pickup zones

3.2.2. KPI 2: borrocde and zone_id_pickup vs. passenger_count

One of the metrics of interest is `borrocode` and match this with `zode_id_pickup` and `passenger_count` as this will indicate the busiest combinations of major boroughs and regions by showing which boroughs have the most customers. In turn this will serve as an indicator of the busiest routes in the area and one of the best methods to maximize our earnings.

Figure 2 shows the total passenger count based on region and borough code. From here it is evident that routes 9 and 14, serving borough 1 (Manhattan) will provide the top two highest number of customers – 13,590,114 and 4,459,778 respectively; followed by region 1 which serves borough 3 (Brooklyn) with a total passenger count of 275,831 over the entire 6 month period.

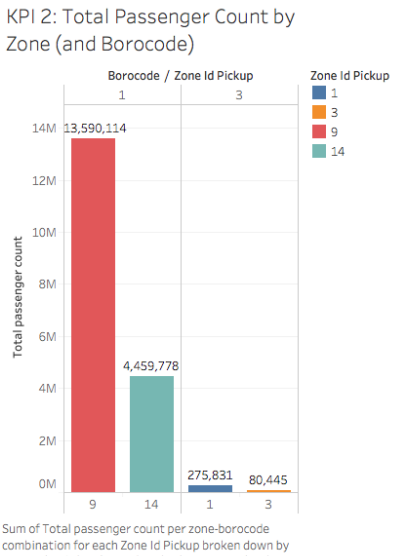


Figure 2: Total Passenger count based on zone and borocode

3.2.3. KPI 3: zone_id_dropoff and borocode vs. tip amount

Our third key performance indicator was based on the `tip_amount` which represents the total tip earned in cash by the cab drivers. We will juxtapose the `tip_amount` to the `borrocode` as customers in regions such as Manhattan and Brooklyn tend to tip considerably higher than other boroughs as they have significantly higher disposable income [3]. Figure 3 shows the total amount in tips based on zone and borocode over the given time period.

Here, clearly it can be seen that our hypothesis was correctly. Customers travelling to Manhattan (`borocode: 1`), which was served by zones 9 and 14, tipped the most at 8,656,648 and 3,136,036; followed by customers travelling to Brooklyn (`borocode: 3`) who tipped a total of 587,416.

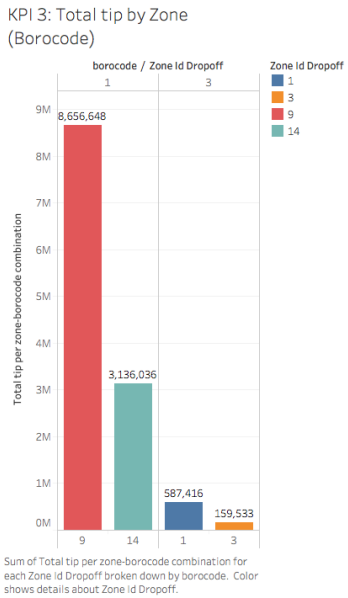


Figure 3: Total tip based on borocode and dropoff zone

3.2.4. KPI 4: Most profitable routes (zones) based on total earnings and total number of trips available

Our fourth performance metric involved finding out the most profitable route based on total earnings and total number of trips available. Based on these figures, we found that zone 9 was the most profitable route, followed by zone 14 and then 1. Intra-zone travel in zone 9 had the highest earnings, as well as the highest number of trips. This meant that zone 9 was quite a busy place of passenger transactions and that the downtime, which is one of the most significant causes for loss of earnings amongst taxi driver, was the lowest in zone 9, followed by zones 14 and 1.

Figure 4 shows the total earnings and total number of trips broken down per drop off and pickup zones. The table 1, below figure 4, shows the top seven highest grossing pick up and drop off zones combining regions 9, 14 and 1.

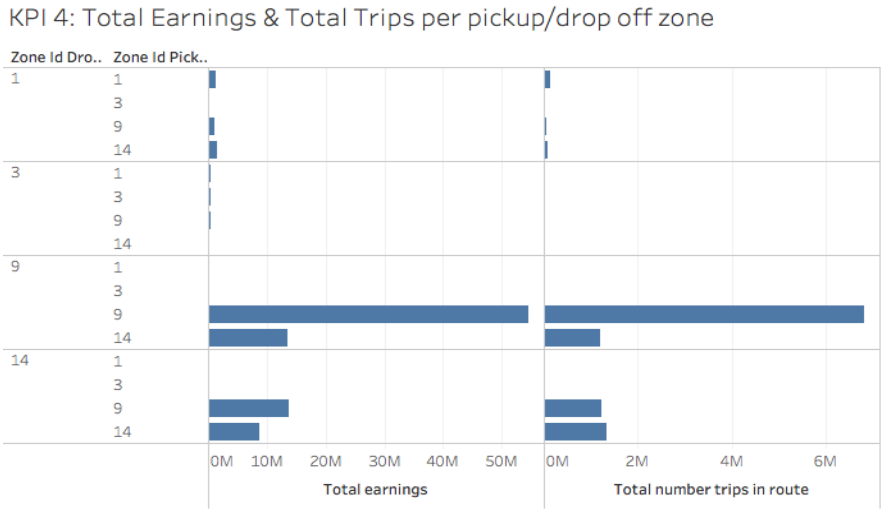


Figure 4: Total earnings and total number of trips according to dropoff and pickup zones

KPI 4: Most profitable route (zones) based on total earnings and total number of trips available						
zone_id_pickup	zone_id_dropoff	Total_earnings	Total_distance_travelled	Total_number_trips_in_route	total_earnings_per_km	total_earnings_per_trip
9	9	54412835.97	59962581.75	6801318	907.4465172	8.000336989
9	14	13810300.48	6739398.57	1214160	2049.188861	11.37436621
14	9	13543504.74	15172601.18	1197658	892.6290607	11.30832403
14	14	8672980.96	10072781.02	1324410	861.0314215	6.548561971
14	1	1551109.58	7771684.64	86684	199.5847299	17.89383946
1	1	1343915.84	175460.96	139103	7659.343936	9.661300188
9	1	1141126.31	311074.61	43494	3668.336384	26.23640755

Table 1: Table containing the top seven pickup and dropoff zones based on total_earnings and total number of trips. Notice, we also calculated total earnings per km and total earnings per trip. But these were ignored since they were found to b irrelevant for this

3.2.5. KPI 5: Most important time to drive based on total number of trips (and total earnings)

Our final key performance indicator consisted of finding out the most important times for driving taxis at regions 9, 14 and 1. Since one of the requirements of the project involved findings the most profitable route and also maintain a regular schedule, so we had to determine what would be the most optimal schedule for the driver. This KPI was implemented by splitting up the entire day into 24 subparts of 1 hour each. Then total number of trips at each of those hours were aggregated. Figure 5 displays a trend in number of trips at each hour of the day. As can be seen from the graph, the busiest period is around 7 pm (674,791 trips) and least busiest at 5 am (66,780). Looking closely into the graph we can see that the most optimal 8 hour schedule for the driver would be from 12 noon to 8 pm.

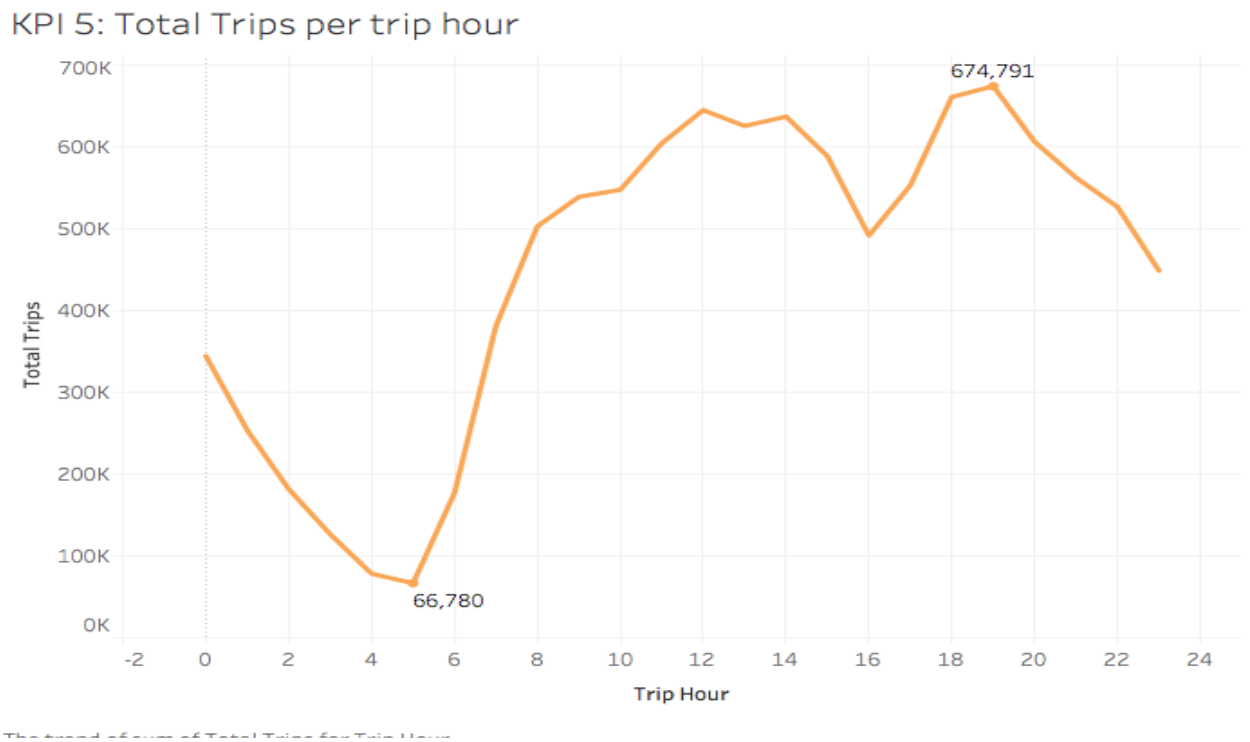


Figure 5: Trend in total number of trips across regions 9, 14 and 1 over a 24 hour period

4. References

[1] Chaudhari, Harshal A., John W. Byers, and Evimaria Terzi. "Putting data in the driver's seat: Optimizing earnings for on-demand ride-hailing." *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 2018

[2] Li, Bin, et al. "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset." *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2011.

[3] https://patch.com/new-york/new-york-city/see-how-much-nyc-households-make-year