BIG DATA SYSTEMS ASSIGNMENT 1

GROUP 22

Authored by:

1. **Chandra Sekhar Gupta – 2019ab04187**
2. **Mahesh Sai – 2019ab04135**
3. **Snigdha Tarua – 2019ab04171**

STEP I. Hadoop and Hive setup :

Setup using docker container : In the command prompt :

1. `$ git clone https://github.com/big-data-europe/docker-hive.git`
2. Navigate to folder with the compose.yml file
3. `$ docker-compose up -d`
4. `$ docker container ls`
5. We need to upload the taxi data to docker container :
   `$ docker cp <source path of data> <container id>:<destination path>`
6. `$ docker exec -it <container name> /bin/bash`
7. `$ /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000`

You hive system is ready to use . You can run a basic query to check if hive is working properly.

PART 1: For Green Taxi

STEP II. Uploading data :

Our data is stored in container path - `<container_id>`:./opt/data/green_taxi/`<all csv files>`

```
-- creating a database for assignment

Create database if not exists  bds ;


USE bds;


-- create the table

Create external table if not exists bds.green_taxi_2019(

VendorID int,

lpep_pickup_datetime timestamp,

lpep_dropoff_datetime timestamp,

store_and_fwd_flag string,

RatecodeID int,

PULocationID int,

DOLocationID int,

passenger_count int,
```

```
trip_distance decimal(10,2),
fare_amount decimal(10,2),
extra decimal(10,2),
mta_tax decimal(10,2),
tip_amount decimal(10,2),
tolls_amount decimal(10,2),
ehail_fee decimal(10,2) ,
improvement_surcharge decimal(10,2),
total_amount decimal(10,2),
payment_type int,
trip_type int,
congestion_surcharge int
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/opt/data/green_taxi/'
tblproperties ("skip.header.line.count"="1");

-- load the table with data
LOAD DATA LOCAL INPATH '/opt/data/green_taxi/' OVERWRITE INTO TABLE
green_taxi_2019;
ALTER TABLE green_taxi_2019 RENAME TO green_taxi_2019_to_dump;

-- Querying table for validation
select count(*) from green_taxi_2019_to_dump ;
+----------+
|   _c0    |
+----------+
| 6044050  |
```

STEP III. Sanity Checks :

Check vendors :

```
select vendorid,count(*) from green_taxi_2019_to_dump group by vendorid;


+-----------+----------+
| vendorid  |   _c1    |
+-----------+----------+
| NULL      | 414107   |
| 1         | 894041   |
| 2         | 4735902  |
+-----------+----------+


select 5629943/6044050;
```

Observations

- Vendor 2 has lager chuck of data
- 0.93% of data has vendor details
- 0.84% data belong to vendor2 , that leaves 0.16% of vendor 1

Count for nulls :

```
select sum(case when   VendorID   is null then 1 else 0 end)   VendorID  ,
sum(case when   lpep_pickup_datetime   is null then 1 else 0 end)
lpep_pickup_datetime  ,
sum(case when   lpep_dropoff_datetime    is null then 1 else 0 end)
lpep_dropoff_datetime   ,
sum(case when   store_and_fwd_flag   is null then 1 else 0 end)
store_and_fwd_flag  ,
sum(case when   RatecodeID   is null then 1 else 0 end)   RatecodeID  ,
sum(case when   PULocationID   is null then 1 else 0 end)   PULocationID
,
sum(case when   DOLocationID   is null then 1 else 0 end)   DOLocationID
,
```

```
sum(case when   passenger_count    is null then 1 else 0 end)
passenger_count   ,
sum(case when   trip_distance    is null then 1 else 0 end)
trip_distance   ,
sum(case when   fare_amount    is null then 1 else 0 end)   fare_amount
,
sum(case when   extra    is null then 1 else 0 end)   extra   ,
sum(case when   mta_tax    is null then 1 else 0 end)   mta_tax   ,
sum(case when   tip_amount   is null then 1 else 0 end)   tip_amount  ,
sum(case when   tolls_amount   is null then 1 else 0 end)   tolls_amount
,
sum(case when   ehail_fee   is null then 1 else 0 end)   ehail_fee  ,
sum(case when   improvement_surcharge    is null then 1 else 0 end)
improvement_surcharge   ,
sum(case when   total_amount   is null then 1 else 0 end)   total_amount
,
sum(case when   payment_type   is null then 1 else 0 end)   payment_type
,
sum(case when   trip_type   is null then 1 else 0 end)   trip_type,
sum(case when   congestion_surcharge   is null then 1 else 0 end)
congestion_surcharge
from bds.green_taxi_2019_to_dump ;


+-----------+----------------------+-----------------------+------------
---------+-------------+---------------+---------------+------------------
+----------------+--------------+--------+----------+-------------+-------
--------+------------+-----------------------+---------------+-----------
----+------------+-----------------------+

| vendorid  | lpep_pickup_datetime  | lpep_dropoff_datetime  |
store_and_fwd_flag  | ratecodeid  | pulocationid  | dolocationid  |
passenger_count  | trip_distance  | fare_amount  | extra  | mta_tax  |
tip_amount  | tolls_amount  | ehail_fee  | improvement_surcharge  |
total_amount  | payment_type  | trip_type  | congestion_surcharge  |
```

```
+-----------+----------------------+----------------------+------------
---------+-------------+--------------+--------------+-----------------
+----------------+--------------+--------+----------+-------------+-------
--------+------------+----------------------+--------------+-----------
----+------------+-----------------------+

| 414107    | 0                    | 0                    | 0
| 414107      | 0             | 0             | 414107           | 0
| 0            | 0      | 0        | 0           | 0             | 6043696
| 2                     | 0             | 414107        | 414466     |
960489               |

+-----------+----------------------+----------------------+------------
---------+-------------+--------------+--------------+-----------------
+----------------+--------------+--------+----------+-------------+-------
--------+------------+----------------------+--------------+-----------
----+---------+-----------------------+
```

- We need to remove the rows with null values.
- "congestion_surcharge" and "ehail_fee" can be removed as they have many null values.

Range of data :

```
select max(VendorID ) max_VendorID  , min(VendorID ) min_VendorID ,

max(lpep_pickup_datetime ) max_lpep_pickup_datetime ,
min(lpep_pickup_datetime ) min_lpep_pickup_datetime ,

max(lpep_dropoff_datetime ) max_lpep_dropoff_datetime ,
min(lpep_dropoff_datetime ) min_lpep_dropoff_datetime ,

max(passenger_count ) max_passenger_count , min(passenger_count )
min_passenger_count ,

max(trip_distance ) max_trip_distance , min(trip_distance )
min_trip_distance ,

max(RatecodeID ) max_RatecodeID , min(RatecodeID ) min_RatecodeID ,

max(store_and_fwd_flag ) max_store_and_fwd_flag , min(store_and_fwd_flag )
min_store_and_fwd_flag ,

max(PULocationID ) max_PULocationID , min(PULocationID ) min_PULocationID
,

max(DOLocationID ) max_DOLocationID , min(DOLocationID ) min_DOLocationID
,
```

```
max(payment_type ) max_payment_type , min(payment_type ) min_payment_type
,
max(fare_amount ) max_fare_amount , min(fare_amount ) min_fare_amount ,
max(extra ) max_extra , min(extra ) min_extra ,
max(mta_tax ) max_mta_tax , min(mta_tax ) min_mta_tax ,
max(tip_amount ) max_tip_amount , min(tip_amount ) min_tip_amount ,
max(tolls_amount ) max_tolls_amount , min(tolls_amount ) min_tolls_amount
,
max(improvement_surcharge ) max_improvement_surcharge ,
min(improvement_surcharge ) min_improvement_surcharge ,
max(total_amount ) max_total_amount , min(total_amount ) min_total_amount
,
max(trip_type ) max_trip_type , min(trip_type ) min_trip_type
from bds.green_taxi_2019_to_dump ;


+---------------+---------------+---------------------------+-------------
--------------+---------------------------+---------------------------+-
----------------------+----------------------+--------------------+--------
------------+-----------------+-----------------+-------------------------
+-------------------------+-------------------+-------------------+-------
------------+-------------------+-------------------+-------------------+-
----------------+-----------------+------------+------------+-----------
---+--------------+-----------------+-----------------+-------------------
+--------------------+---------------------------+-----------------------
----+--------------------+--------------------+----------------+-----------
----+
| max_vendorid  | min_vendorid  | max_lpep_pickup_datetime  |
min_lpep_pickup_datetime  | max_lpep_dropoff_datetime  |
min_lpep_dropoff_datetime  | max_passenger_count  | min_passenger_count  |
max_trip_distance  | min_trip_distance  | max_ratecodeid  | min_ratecodeid
| max_store_and_fwd_flag  | min_store_and_fwd_flag  | max_pulocationid  |
min_pulocationid  | max_dolocationid  | min_dolocationid  |
max_payment_type  | min_payment_type  | max_fare_amount  | min_fare_amount
| max_extra  | min_extra  | max_mta_tax  | min_mta_tax  | max_tip_amount
| min_tip_amount  | max_tolls_amount  | min_tolls_amount  |
```

```
max_improvement_surcharge  | min_improvement_surcharge  | max_total_amount
| min_total_amount  | max_trip_type  | min_trip_type  |
+---------------+---------------+---------------------------+-------------
--------------+---------------------------+---------------------------+-
--------------------+--------------------+-------------------+--------
-----------+-----------------+-----------------+------------------------
+------------------------+-------------------+-------------------+-------
-----------+-------------------+-------------------+-------------------+-
-----------------+------------------+-----------+-----------+-----------
---+--------------+-----------------+-----------------+-------------------
+-------------------+---------------------------+------------------------
----+-------------------+-------------------+----------------+------------
----+
| 2             | 1             | 2062-08-15 00:00:00.0     | 2008-10-21
15:52:05.0     | 2062-08-15 16:34:10.0      | 2008-10-21 15:54:26.0      |
9                    | 0                    | 666.60             | -23.88
| 99              | 1               | Y                        | N
| 265               | 1                 | 265               | 1
| 5                 | 1                 | 4011.50          | -890.00
| 11.58      | -4.50      | 17.33        | -0.50        | 441.00
| -90.50          | 935.50            | -21.00            | 0.47
| -0.30                       | 4012.30          | -890.30          | 2
| 1              |
+---------------+---------------+---------------------------+-------------
--------------+---------------------------+---------------------------+-
--------------------+--------------------+-------------------+--------
-----------+-----------------+-----------------+------------------------
+------------------------+-------------------+-------------------+-------
-----------+-------------------+-------------------+-------------------+-
-----------------+------------------+-----------+-----------+-----------
---+--------------+-----------------+-----------------+-------------------
+-------------------+---------------------------+------------------------
----+-------------------+-------------------+----------------+------------
----+
```

Column which seems to match the meta data description :

- vendorid is fine; values between two provider of 1 & 2
- pulocationid,dolocationid:- pickup and drop location are raning from 1 to 265
- payment_type in data is spread between 1-6 which is with provided values of 1-5
- store_and_fwd_flag has yes or no values
- trip_type is also between 1-2

Columns that are not matching the meta data description :

Checking for remaining individual columns based on their range and metadata provided :

tpep_pickup_datetime

- The data is for 2019
- Any day before or after 2019 is out of context

```
select count(*) from  bds.green_taxi_2019_to_dump
where lpep_pickup_datetime < '2019-01-01 00:00:00.0' or
lpep_pickup_datetime  >= '2020-01-01 00:00:00.0' ;
+------+
| _c0  |
+------+
| 273  |
+------+
```

- 273 records are not in the range

```
select  vendorid, count(*)from  bds.green_taxi_2019_to_dump
where lpep_pickup_datetime < '2019-01-01 00:00:00.0' or
lpep_pickup_datetime  >= '2020-01-01 00:00:00.0' group by vendorid;


+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| 2         | 273  |
+-----------+------+
```

- seems Vendor 2 is at fault.

tpep_dropoff_datetime

- The data is for 2019
- Any day before or after 2019 is out of context

```
select  vendorid, count(*) from  bds.green_taxi_2019_to_dump
where lpep_dropoff_datetime < '2019-01-01 00:00:00.0' or
lpep_dropoff_datetime >= '2020-01-01 00:00:00.0' group by vendorid;


+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| NULL      | 7    |
| 1         | 9    |
| 2         | 396  |
+-----------+------+
```

- vendor 2 has 396 records
- for vendor 1 :

```
select * from  bds.green_taxi_2019_to_dump BDM
where (lpep_dropoff_datetime < '2019-01-01 00:00:00.0' or
lpep_dropoff_datetime >= '2020-01-01 00:00:00.0')
and vendorid=1;
```

- seems like the data is corrupt with one in past(2018)
- drop of time can't be greater or equal too pick up time :

```
select count(*) from bds.green_taxi_2019_to_dump
where lpep_dropoff_datetime<=lpep_pickup_datetime;


+--------+
|  _c0   |
+--------+
```

```
| 13260  |
+--------+


select 13260/ 6044050 ;
```

- 0.002,a smaller set of records can be rejected

```
select vendorid, count(*) from bds.green_taxi_2019_to_dump
where lpep_dropoff_datetime<=lpep_pickup_datetime
group by vendorid;
+-----------+-------+
| vendorid  |  _c1  |
+-----------+-------+
| NULL      | 2313  |
| 1         | 5612  |
| 2         | 5335  |
+-----------+-------+
```

- Both vendors have faults
- We will reject these records

passenger_count

```
select passenger_count, count(*) from  bds.green_taxi_2019_to_dump  group
by passenger_count;
+------------------+----------+
| passenger_count  |   _c1    |
+------------------+----------+
| NULL             | 414107   |
| 0                | 11689    |
| 3                | 76983    |
| 6                | 88485    |
| 9                | 35       |
| 1                | 4843792  |
```

```
| 4                 | 28862    |
| 7                 | 87       |
| 2                 | 415566   |
| 5                 | 164320   |
| 8                 | 124      |
+-------------------+----------+
```

- Null value needs to be removed
- 0 again seems like an disinterested driver not putting in details, or an empty parcel being sent in the cab
- passenger_count > 6 seems strange , since the value is driver-entered it could be a error.

```
select vendorid,passenger_count, count(*)
from  bds.green_taxi_2019_to_dump
where passenger_count in  (0,7,8,9) group by vendorid,passenger_count
order by passenger_count,vendorid;
+-----------+------------------+--------+
| vendorid  | passenger_count  |  _c2   |
+-----------+------------------+--------+
| 1         | 0                | 10340  |
| 2         | 0                | 1349   |
| 2         | 7                | 87     |
| 2         | 8                | 124    |
| 2         | 9                | 35     |
+-----------+------------------+--------+
```

- We will keep 0 in passenger_count as the count or records could be for parcels.
- Records we will keep as is assuming that they are bigger cars
- Vendor 1 seems to be at fault w.r.t 0 passenger_count
- Vendor 2 has higher value of passenger_count

trip_distance

- The elapsed trip distance in miles reported by the taximeter.
- max_trip_distance,min_trip_distance
- 666.60 , -23.88

- Check negative distance :

```
select  count(*) from  bds.green_taxi_2019_to_dump  where
trip_distance<=0;
+----------+
|   _c0    |
+----------+
| 155085   |
+----------+
```

- we will reject this data with 0 or negative distance

```
select  vendorid,count(*) from  bds.green_taxi_2019_to_dump where
trip_distance<=0 group by vendorid;
+-----------+--------+
| vendorid  |  _c1   |
+-----------+--------+
| NULL      | 22367  |
| 1         | 56825  |
| 2         | 75893  |
+-----------+--------+
```

- Both vendors have error data

ratecodeid

```
select  ratecodeid,count(*) from   bds.green_taxi_2019_to_dump  BDM group
by ratecodeid;
+-------------+----------+
| ratecodeid  |   _c1    |
+-------------+----------+
| NULL        | 414107   |
| 3           | 2646     |
| 6           | 58       |
| 99          | 51       |
```

```
| 1           | 5357348  |
| 4           | 4167     |
| 2           | 11206    |
| 5           | 254467   |
+-------------+----------+
```

- 1-6 are valid id as per metadata, and 99 value is incorrect
- We will reject null and value 99.
- Vendorid wise analysis:

```
select vendorid , count(*)
from  bds.green_taxi_2019_to_dump BDM
where ratecodeid=99
group by vendorid;
+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| 1         | 5    |
| 2         | 46   |
+-----------+------+
```

- Vendor 2 is the mazor contributor towards this data discripency

store_and_fwd_flag

```
select  store_and_fwd_flag,count(*) from  bds.green_taxi_2019_to_dump
group by store_and_fwd_flag;
+---------------------+----------+
| store_and_fwd_flag  |   _c1    |
+---------------------+----------+
|                     | 414107   |
| N                   | 5615484  |
| Y                   | 14459    |
+---------------------+----------+
```

- The value of yes and no are fine
- null rows to be removed

fare_amount

- The time-and-distance fare calculated by the meter.
- max_fare_amount,min_fare_amount,
- 4011.50 | -890.00

```
select
percentile_approx(fare_amount,array(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.
99))
from   bds.green_taxi_2019_to_dump;


+----------------------------------------------------+
|                        _c0                         |
+----------------------------------------------------+
|
[4.998886635822649,6.0,7.491366973305637,8.5,10.47525966696858,12.49431261
1170402,15.49682227081285,20.5,29.11,59.99183053691275] |


+----------------------------------------------------+
```

- Most these values are within range
- Seems like the negative values and very high values are wrong data or outliner.
- We can easily reject negative values
- Max amount is also fine

```
select count(*),vendorid from  bds.green_taxi_2019_to_dump where
fare_amount<0 group by vendorid;
+--------+-----------+
|  _c0   | vendorid  |
+--------+-----------+
| 1033   | NULL      |
| 1      | 1         |
| 18635  | 2         |
+--------+-----------+
```

extra

- max_extra,min_extra
- 11.58 | -4.50
- But data disctionary says :-Miscellaneous extras and surcharges.
- Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
- Hence we will reject these values let's verify their count

```
select count(*) from  bds.green_taxi_2019_to_dump where extra not in
(0,0.5,1);
+---------+
|   _c0   |
+---------+
| 517065  |
+---------+
select 517065/6044050 ;
```

- 0.085 this data can be safely rejected

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump where extra not
in (0,0.5,1)
 group by vendorid;
+-----------+---------+
| vendorid  |   _c1   |
+-----------+---------+
| NULL      | 393927  |
| 1         | 111665  |
| 2         | 11473   |
+-----------+---------+
```

- vendor 1 has high number of error values

mta_tax

Data Dictionary :-$0.50 MTA tax that is automatically triggered based on the metered rate in use.

- max_mta_tax,min_mta_tax
- 17.33 | -0.50

```
select count(*) from  bds.green_taxi_2019_to_dump where mta_tax not in
(0,0.5);
+---------+
|  _c0    |
+---------+
| 18295   |
+---------+
select 18295/6044050 ;
```

- 0.003 smaller set, based on data disctionary we would reject these

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump
where mta_tax not in (0,0.5) group by vendorid;
+-----------+---------+
| vendorid  |  _c1    |
+-----------+---------+
| NULL      | 330     |
| 1         | 35      |
| 2         | 17930   |
+-----------+---------+
```

- Vendor 2 is mazorly at fault

tip_amount

- Tip amount – This field is automatically populated for credit card tips.
- Cash tips are not included.
- max_tip_amount,min_tip_amount
- 496, -218,

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump
where tip_amount <0 group by vendorid;
+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| 2         | 224  |
```

- 224 values are negative can be rejected
- all belong to vendor 2
- Check if their are non credit card based tips

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump where
Payment_type!=1 and tip_amount>0 group by vendorid;
+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| 1         | 112  |
| 2         | 1    |
+-----------+------+
```

- 113 records have payment mode other than credit and still have tip amount greate than 0
- Vendor 1 has most errors
- we will reject these records to sanity as well.

tolls_amount

- Data Dictionary:- Total amount of all tolls paid in trip.
- The value can't be negative
- max_tolls_amount,min_tolls_amount,
- 935.50 | -21.00

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump
where tolls_amount <0
group by vendorid;
+-----------+------+
| vendorid  | _c1  |
+-----------+------+
| 2         | 19   |
+-----------+------+
```

- Vendor 2 with all error value in tolls

improvement_surcharge

- Data Dictionary :- $0.30 improvement surcharge assessed trips at the flag drop.

- The improvement surcharge began being levied in 2015.
- max_improvement_surcharge,min_improvement_surcharge,
- 0.47 | -0.30

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump
where improvement_surcharge not in (0,0.3)
group by vendorid;
+-----------+--------+
| vendorid  |  _c1   |
+-----------+--------+
| NULL      | 6      |
| 2         | 17820  |
+-----------+--------+
```

- All records belong to vendor 2
- error values should be removed

total_amount

- Data Dictionary:- The total amount charged to passengers. Does not include cash tips
- cant be negative and has similar high value as fare_amount, we will check this with similar queries
- max_total_amount,min_total_amount
- 4012.30 | -890.30
- vendor wise :

```
select vendorid,count(*) from  bds.green_taxi_2019_to_dump
where total_amount<0
group by vendorid;
+-----------+--------+
| vendorid  |  _c1   |
+-----------+--------+
| NULL      | 1008   |
| 2         | 18635  |
+-----------+--------+
```

- Group 2 highly dominate in the corrupt data section

STEP IV . Preprocessing

From above analysis, we need to do the following :

- rows with null values to be removed
- "congestion_surcharge" and "ehail_fee" can be removed as they have many null values
- we are interested in trips made in 2019.
- records with lpep_dropoff_datetime<=lpep_pickup_datetime should be removed
- we will reject this data with 0 or negative trip_distance
- we will reject null and value 99 for ratecodeid
- negative fare_amount to be removed
- extra should be in (0,0.5,1)
- mta_tax should in (0,0.5)
- tip_amount < 0 should be removed
- tip_amount should be only with payment_type = 1
- tolls_amount < 0 should be removed
- improvement_surcharge in (0,0.3)
- total_amount < 0 should be removed

```
use bds;


CREATE TABLE green_taxi_2019 AS

SELECT vendorid , lpep_pickup_datetime  , lpep_dropoff_datetime  ,
store_and_fwd_flag  , ratecodeid  ,pulocationid  , dolocationid
,passenger_count , trip_distance  , fare_amount , extra  , mta_tax  ,
tip_amount  , tolls_amount  ,  improvement_surcharge  , total_amount  ,
payment_type  ,trip_type

FROM bds.green_taxi_2019_to_dump

WHERE

(lpep_pickup_datetime >= '2019-01-01 00:00:00.0' and lpep_pickup_datetime
< '2020-01-01 00:00:00.0') and

(lpep_dropoff_datetime >= '2019-01-01 00:00:00.0' and
lpep_dropoff_datetime < '2020-01-01 00:00:00.0') and

(lpep_dropoff_datetime>lpep_pickup_datetime) and

(passenger_count is not null) and

(trip_distance>0) and

(ratecodeid!=99) and
```

```
(fare_amount>0 ) and
 (extra in (0,0.5,1)) and
 (mta_tax  in (0,0.5)) and
((tip_amount >=0 and Payment_type=1) or (Payment_type!=1 and
tip_amount=0)) and
( tolls_amount >=0) and
( improvement_surcharge in (0,0.3)) and
(total_amount>0 ) ;



select count(*) from green_taxi_2019;
+----------+
|   _c0    |
+----------+
| 5359662  |
+----------+
select 6044050-5359662;
+---------+
|   _c0   |
+---------+
| 684388  |
+---------+
select 684388/6044050;
```

Observation :

- 310507 were removed
- amounting to 0.11% of data

STEP V. Execute queries :

Q1) Which vendor provides the most useful data?

```
select vendorid,count(*) from bds.green_taxi_2019 group by vendorid;
+-----------+----------+
| vendorid  |   _c1    |
+-----------+----------+
| 1         | 717262   |
| 2         | 4642400  |
+-----------+----------+
```

- Quantity wise : Vendor 2 provides large quantity of data .
- Quality wise (From analysis steps in sanity check) : Vendor 1 provides less erroneous data .
- Overall, Vendor 1 provides more useful data.

Q2) Find the month wise trip count, average distance and average passenger count from the trips completed by green taxis in 2019. Summary visualizations will be preferred for better analysis

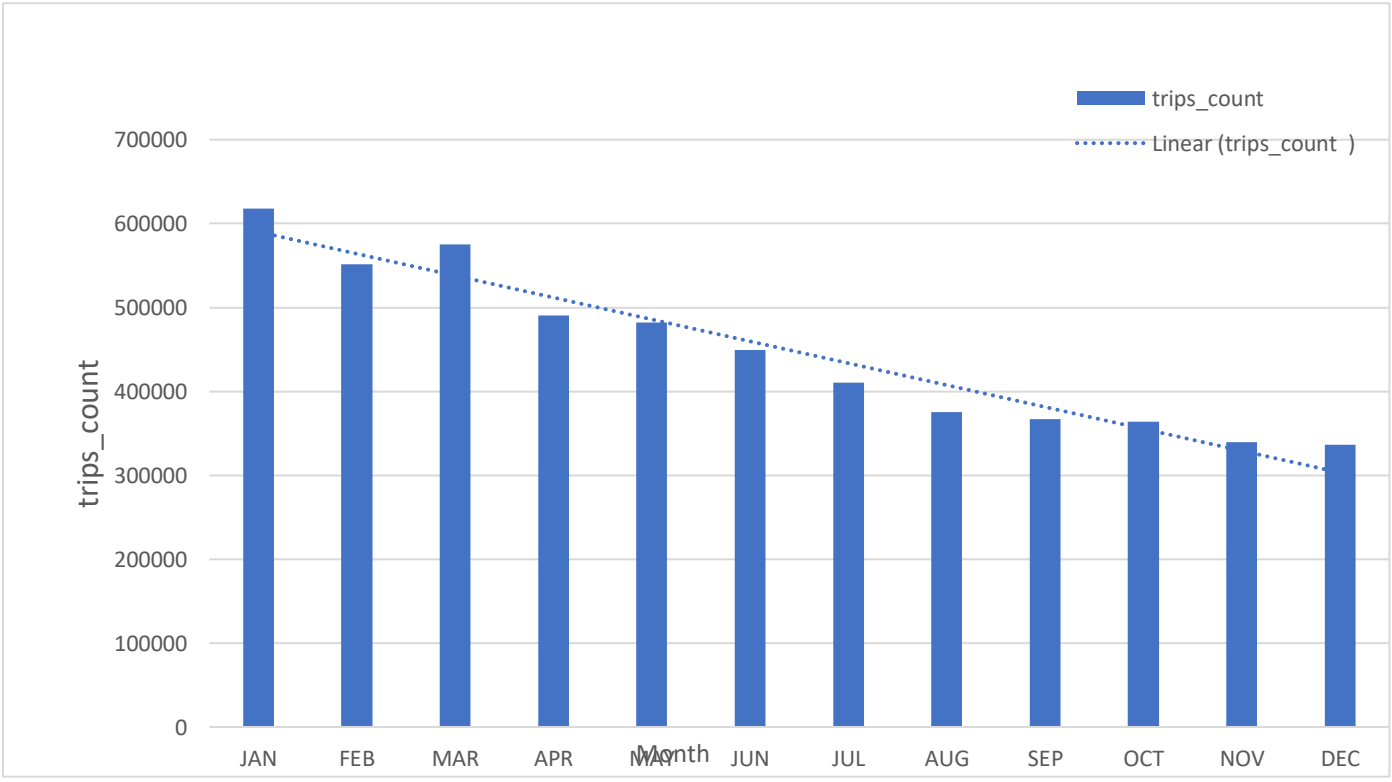
```
SELECT MONTH(lpep_pickup_datetime) Month_No,COUNT(*) Trips_Count,
ROUND(AVG(trip_distance),2)
Average_Distance,AVG(passenger_count) Average_Passengers from
bds.green_taxi_2019
group by MONTH(lpep_pickup_datetime)
order by  Month_No ;


+-----------+--------------+-------------------+----------------------+
| month_no  | trips_count  | average_distance  | average_passengers   |
+-----------+--------------+-------------------+----------------------+
| 1         | 617657       | 3.51              | 1.3175840312665443   |
| 2         | 551903       | 3.56              | 1.3097772615840102   |
| 3         | 575538       | 3.50              | 1.3038461404807329   |
| 4         | 490506       | 3.02              | 1.3184996717675217   |
| 5         | 481850       | 3.00              | 1.3110138009754073   |
```

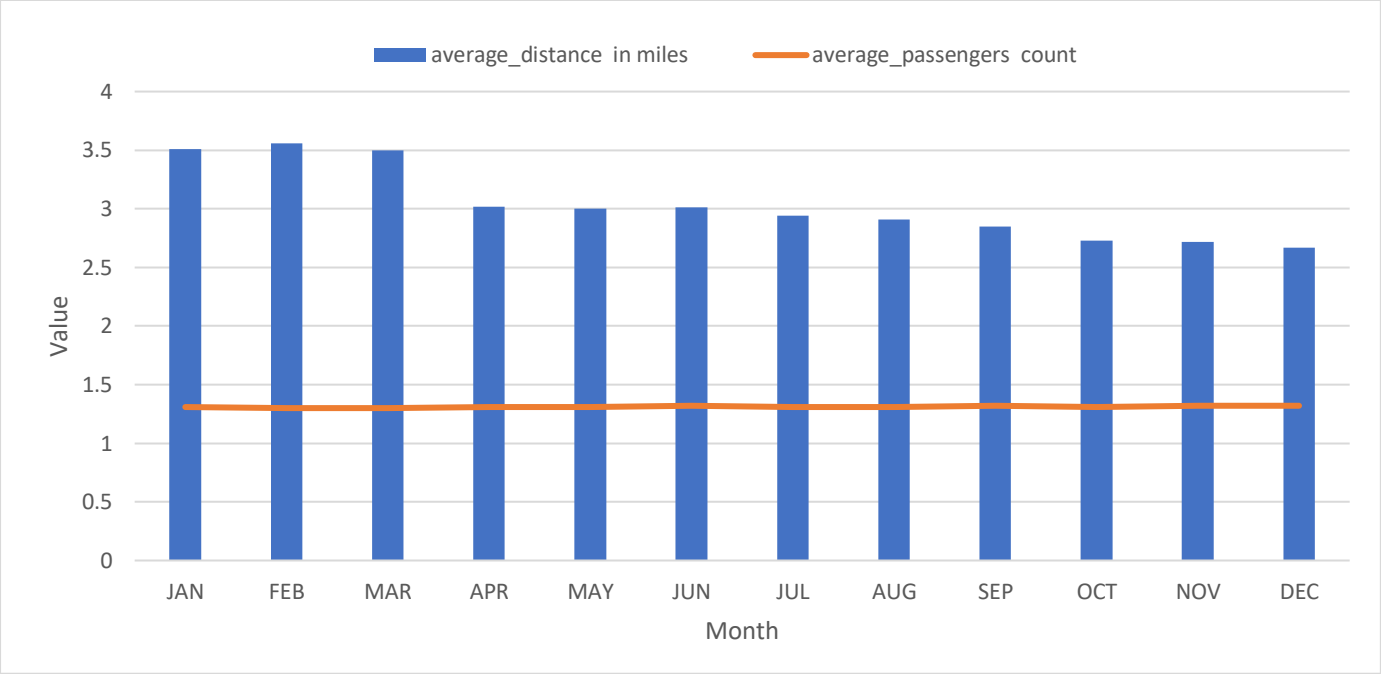
6	449592	3.01	1.3239982028149968
7	410602	2.94	1.3150569164300223
8	375052	2.91	1.3109382165673027
9	366737	2.85	1.3208430019332655
10	363883	2.73	1.316456113640923
11	339787	2.72	1.3270078019465135
12	336555	2.67	1.3211362184486934

Observation :

- January had the most number of taxi trip
- Trip count gradually decreases from Jan to Dec.
- Average Passenger Count remains almost same over the year
- Average Distance of a taxi trip is about 3 miles .

Summary visualizations:





Q3) Find out the five busiest routes served by the green taxis during 2019. The name of start and drop points to be provided.

```
select pulocationID,dolocationID,count(*) as trip_count from
bds.green_taxi_2019  group by pulocationID,dolocationID order by
trip_count desc LIMIT 5;


+---------------+---------------+-------------+
| pulocationid  | dolocationid  | trip_count  |
+---------------+---------------+-------------+
| 75            | 74            | 72293       |
| 7             | 7             | 66774       |
| 74            | 75            | 62027       |
| 41            | 42            | 56921       |
| 95            | 95            | 52612       |
+---------------+---------------+-------------+
```

Top 5 busiest routes :

1. 75 (Manhattan,East Harlem South,Boro Zone) - 74 (Manhattan ,East Harlem North,Boro Zone)
2. Within areacode 7 (Queens,Astoria,Boro Zone)
3. 74 (Manhattan,East Harlem North,Boro Zone) - 75 (Manhattan,East Harlem South,Boro Zone)
4. 41 (Manhattan,Central Harlem,Boro Zone)- 42 (Manhattan,Central Harlem North,Boro Zone)
5. Within areacode 95 (Queens,Forest Hills,Boro Zone)

Q4) What are the top 3 busiest hours of the day for the taxis?

Top 3 hours having maximum trip count.

```
select HOUR(lpep_pickup_datetime) hour , count(*) as trip_count from
bds.green_taxi_2019  group by HOUR(lpep_pickup_datetime) order by
trip_count desc LIMIT 3;
+------+-------------+
| hour  | trip_count  |
+------+-------------+
| 18   | 385175      |
| 17   | 374181      |
| 16   | 352605      |
+------+-------------+
```

- 4 pm to 7 pm looks busiest hours.

Q5) What is the most preferred way of payment used by the passengers? What are the weekly trends observed for the methods of payments?

```
select payment_type,count(*) cnt
from bds.green_taxi_2019 group by payment_type
order by cnt desc;
+---------------+----------+
| payment_type  |   cnt    |
+---------------+----------+
| 1             | 3046757  |
| 2             | 2294075  |
| 3             | 12491    |
| 4             | 6199     |
```

- Payment Type 1 stands for credit card . It seems to be the most preferred way for payment.
- Weekly trend :

```
-- count of payments made from each type in first 7 weeks of year 2019


select weekofyear(lpep_pickup_datetime) as weekno, payment_type ,count(*)
cnt from bds.green_taxi_2019

group by weekofyear(lpep_pickup_datetime), payment_type

order by weekno , payment_type LIMIT 35;


+---------+---------------+--------+

| weekno  | payment_type  |  cnt   |

+---------+---------------+--------+

| 1       | 1             | 75023  |

| 1       | 2             | 56558  |

| 1       | 3             | 301    |

| 1       | 4             | 148    |

| 1       | 5             | 2      |

| 2       | 1             | 89622  |

| 2       | 2             | 53490  |

| 2       | 3             | 289    |

| 2       | 4             | 183    |

| 2       | 5             | 5      |

| 3       | 1             | 87830  |

| 3       | 2             | 52132  |

| 3       | 3             | 269    |

| 3       | 4             | 166    |

| 3       | 5             | 6      |

| 4       | 1             | 87057  |
```

```
| 4         | 2              | 51910  |
| 4         | 3              | 290    |
| 4         | 4              | 115    |
| 4         | 5              | 4      |
| 5         | 1              | 92073  |
| 5         | 2              | 54876  |
| 5         | 3              | 315    |
| 5         | 4              | 147    |
| 5         | 5              | 5      |
| 6         | 1              | 90261  |
| 6         | 2              | 52911  |
| 6         | 3              | 289    |
| 6         | 4              | 158    |
| 6         | 5              | 6      |
| 7         | 1              | 84920  |
| 7         | 2              | 50896  |
| 7         | 3              | 278    |
| 7         | 4              | 150    |
| 7         | 5              | 4      |
+---------+---------------+--------+
-- check the count of payment types for Monday to Sunday .
select date_format(lpep_pickup_datetime ,'u') as day_of_week, payment_type
,count(*) cnt from bds.green_taxi_2019
group by date_format(lpep_pickup_datetime ,'u'), payment_type
order by day_of_week , payment_type ,cnt desc LIMIT 35;


+--------------+---------------+---------+
| day_of_week  | payment_type  |   cnt   |
+--------------+---------------+---------+
| 5            | 1             | 483796  |
```

4	1	476354
3	1	460756
2	1	440438
6	1	426105
1	1	397289
7	1	362019
6	2	385839
5	2	368452
4	2	326603
3	2	311709
2	2	306602
7	2	300909
1	2	293961
6	3	2053
5	3	1970
7	3	1728
4	3	1723
2	3	1716
3	3	1652
1	3	1649
6	4	1063
5	4	954
1	4	883
7	4	870
3	4	848
4	4	812
2	4	769
3	5	28
4	5	24
2	5	23

```
| 5             | 5              | 21       |
| 1             | 5              | 18       |
| 6             | 5              | 17       |
| 7             | 5              | 9        |
+---------------+----------------+----------+
```

About weekly trends :

- Credit card(1) and cash(2) are the preferred type of payment
- There are very few unknown mode of payment(5) and null voided trip(6)
- weeks with holidays have less trips than normal weeks
- Payment count increases from Monday to Fridays and starts dropping then.

PART 2: For Yellow Taxi

STEP II. Uploading data :

Our data is stored in container path - `<container_id>:./opt/data/yellow_taxi/<all csv files>`

```
create database bds;

show databases;

+----------------+

| database_name  |

+----------------+

| bds            |

| default        |

+----------------+

2 rows selected (0.231 seconds)

-- create a table for 2019 yellow taxi data

create external table if not exists yellow_taxi_2019

( vendor_id int,

tpep_pickup_datetime timestamp,

tpep_dropoff_datetime timestamp,

passenger_count int,

trip_distance decimal(10,2),

rate_code_id int,

store_and_fwd_flag string,

pulocationid int,

dolocationid int,

payment_type int,

fare_amount decimal(10,2),

extra decimal(10,2),
```

```
mta_tax decimal(10,2),
tip_amount decimal(10,2),
tolls_amount decimal(10,2),
improvement_surcharge decimal(10,2),
total_amount decimal(10,2),
congestion_surcharge int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/opt/data/yellow_taxi'
tblproperties ("skip.header.line.count"="1");


-- structure of the table
describe yellow_taxi_2019;
+------------------------+----------------+----------+
|        col_name        |   data_type    | comment  |
+------------------------+----------------+----------+
| vendor_id              | int            |          |
| tpep_pickup_datetime   | timestamp      |          |
| tpep_dropoff_datetime  | timestamp      |          |
| passenger_count        | int            |          |
| trip_distance          | decimal(10,2)  |          |
| rate_code_id           | int            |          |
| store_and_fwd_flag     | string         |          |
| pulocationid           | int            |          |
| dolocationid           | int            |          |
| payment_type           | int            |          |
| fare_amount            | decimal(10,2)  |          |
| extra                  | decimal(10,2)  |          |
```

```
| mta_tax                | decimal(10,2)  |          |
| tip_amount             | decimal(10,2)  |          |
| tolls_amount           | decimal(10,2)  |          |
| improvement_surcharge  | decimal(10,2)  |          |
| total_amount           | decimal(10,2)  |          |
| congestion_surcharge   | int            |          |
+------------------------+----------------+----------+
18 rows selected (0.133 seconds)


-- load the data into the hive table


LOAD DATA LOCAL INPATH '/opt/data/yellow_taxi' OVERWRITE INTO TABLE
yellow_taxi_2019;


time taken to load data into the table
No rows affected (39.063 seconds)
```

STEP III. Sanity Checks :

```
--  number of records in the table (sanity check)
select count(*) from yellow_taxi_2019;
+-----------+
|    _c0    |
+-----------+
| 84399019  |
+-----------+
1 row selected (48.204 seconds)
```

```
-- number of records per vendor

SELECT vendor_id, count(*) as COUNT from yellow_taxi_2019 group by vendor_id;

+------------+-----------+

| vendor_id  |   count   |

+------------+-----------+

| NULL       | 246601    |

| 2          | 53517181  |

| 4          | 267080    |

| 1          | 30368157  |

+------------+-----------+

4 rows selected (74.649 seconds)


-- checking null values in all the columns

select sum(case when   vendor_id   is null then 1 else 0 end)   vendor_id  ,

sum(case when   tpep_pickup_datetime   is null then 1 else 0 end)
tpep_pickup_datetime  ,

sum(case when   tpep_dropoff_datetime    is null then 1 else 0 end)
tpep_dropoff_datetime   ,

sum(case when   passenger_count    is null then 1 else 0 end)
passenger_count   ,

sum(case when   trip_distance    is null then 1 else 0 end)   trip_distance
,

sum(case when   rate_code_id    is null then 1 else 0 end)   rate_code_id   ,

sum(case when   store_and_fwd_flag   is null then 1 else 0 end)
store_and_fwd_flag  ,

sum(case when   pulocationid   is null then 1 else 0 end)   pulocationid  ,

sum(case when   dolocationid   is null then 1 else 0 end)   dolocationid  ,

sum(case when   payment_type    is null then 1 else 0 end)   payment_type   ,

sum(case when   fare_amount    is null then 1 else 0 end)   fare_amount   ,
```

```
sum(case when   extra    is null then 1 else 0 end)   extra   ,

sum(case when   mta_tax    is null then 1 else 0 end)   mta_tax   ,

sum(case when   tip_amount    is null then 1 else 0 end)   tip_amount   ,

sum(case when   tolls_amount    is null then 1 else 0 end)   tolls_amount   ,

sum(case when   improvement_surcharge    is null then 1 else 0 end)
imporvement_surcharge   ,

sum(case when   total_amount    is null then 1 else 0 end)   total_amount   ,

sum(case when   congestion_surcharge    is null then 1 else 0 end)
congestion_surcharge

from yellow_taxi_2019;



-- output

+------------+----------------------+-------------------------+--------------
----+----------------+--------------+---------------------+---------------+-
--------------+---------------+--------------+--------+----------+-----------
--+---------------+-----------------------+---------------+-----------------
------+

| vendor_id  | tpep_pickup_datetime  | tpep_dropoff_datetime  |
passenger_count  | trip_distance  | rate_code_id  | store_and_fwd_flag  |
pulocationid  | dolocationid  | payment_type  | fare_amount  | extra  |
mta_tax  | tip_amount  | tolls_amount  | imporvement_surcharge  |
total_amount  | congestion_surcharge  |

+------------+----------------------+-------------------------+--------------
----+----------------+--------------+---------------------+---------------+-
--------------+---------------+--------------+--------+----------+-----------
--+---------------+-----------------------+---------------+-----------------
------+

| 246601     | 0                    | 0                     | 246601
| 0              | 246601        | 0                   | 0            | 0
| 246601        | 0            | 0      | 0        | 0           | 0
| 0                    | 0             | 4855981              |

+------------+----------------------+-------------------------+--------------
----+----------------+--------------+---------------------+---------------+-
--------------+---------------+--------------+--------+----------+-----------
--+---------------+-----------------------+---------------+-----------------
------+

1 row selected (459.877 seconds)
```

The above data looks ok and let's start analysing data and preprocessing them each column.

VendorID

A code indicating the TPEP provider that provided the record.
1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.

```
-- query
select vendor_id, count(*), count(*) * 100 / 84399019 from yellow_taxi_2019
group by vendor_id;

-- output
+------------+-----------+----------------------+
| vendor_id  |    _c1    |         _c2          |
+------------+-----------+----------------------+
| NULL       | 246601    | 0.2921846757484231   |
| 2          | 53517181  | 63.40971925277947    |
| 4          | 267080    | 0.31644917578959064  |
| 1          | 30368157  | 35.981646895682516   |
+------------+-----------+----------------------+
4 rows selected (131.501 seconds)
```

Observation

- we have some data which doesn't have vendor id and some other whose id is 4.
- we can remove vendor 4 data and NULL vendor id data as it is only 0.3% and 0.29% respectively.
- vendor 2 provides 63.4 % of data and vendor 1 provides 35.9 % of data.

Preprocessing requirement

- removing data with null vendor id and vendor id = 4 (as it is not given in the data dictionary).

tpep_pickup_datetime, tpep_dropoff_datetime

The date and time when the meter was engaged, disengaged respectively

```
-- query to check min and max time period for pickup and dropoff
select min(tpep_pickup_datetime), max(tpep_pickup_datetime),
min(tpep_dropoff_datetime), max(tpep_dropoff_datetime) from
yellow_taxi_2019;
-- output

+-----------------------+-----------------------+----------------------
-+-------------------------+
|          _c0          |          _c1          |          _c2
|
```

```
+-------------------------+-------------------------+-----------------------
-+-------------------------+
| 2001-01-01 00:02:08.0  | 2090-12-31 06:41:26.0  | 2001-01-01 01:00:02.0
| 2090-12-31 07:18:49.0  |
+---------------------+---------------------+---------------------+---------------------+
1 row selected (230.301 seconds)
```

Observation

- We are trying to focus only on the rides in 2019.
- Looks like other data is also present and we can get rid of that.

```
-- query to find wrong data given by the vendor

select count(*) as num_of_wrong_date_trips , vendor_id from
yellow_taxi_2019
where (tpep_pickup_datetime < '2019-01-01 00:00:00.0' or
tpep_pickup_datetime >= '2020-01-01 00:00:00.0')
group by vendor_id;

-- output
+--------------------------+------------+
| num_of_wrong_date_trips  | vendor_id  |
+--------------------------+------------+
| 1442                     | 2          |
+--------------------------+------------+
1 row selected (155.463 seconds)

-- query to find wrong data given by the vendor

select count(*) as num_of_wrong_date_trips , vendor_id from
yellow_taxi_2019
where (tpep_dropoff_datetime < '2019-01-01 00:00:00.0' or
tpep_dropoff_datetime >= '2020-01-01 00:00:00.0')
group by vendor_id;

-- output
+--------------------------+------------+
| num_of_wrong_date_trips  | vendor_id  |
+--------------------------+------------+
| 9                        | NULL       |
| 2265                     | 2          |
| 233                      | 1          |
+--------------------------+------------+
3 rows selected (180.105 seconds)
```

Preprocessing requirement

- Trips which do not fall in 2019 time period can be removed.

```
-- query to find tpep_dropoff_datetime <= tpep_pickup_datetime

select count(*), vendor_id from yellow_taxi_2019
where tpep_dropoff_datetime<=tpep_pickup_datetime
group by vendor_id;

-- output

+--------+------------+
|  _c0   | vendor_id  |
+--------+------------+
| 1323   | NULL       |
| 8595   | 2          |
| 15     | 4          |
| 67478  | 1          |
+--------+------------+
4 rows selected (179.612 seconds)
```

Observations

- This is the faulty data and it has to be removed as pickup can't be after dropoff time.
- vendor 1 has given most number of wrong data points in this category.

Passenger_count

The number of passengers in the vehicle. This is a driver-entered value.

```
-- query
select passenger_count, count(*), count(*) * 100 / 84399019  from
yellow_taxi_2019  group by passenger_count;

-- output
+------------------+-----------+------------------------+
| passenger_count  |    _c1    |          _c2           |
+------------------+-----------+------------------------+
| NULL             | 246601    | 0.2921846757484231     |
| 0                | 1525798   | 1.8078385484551662     |
| 9                | 225       | 2.66590776369095E-4    |
| 7                | 416       | 4.928967243090823E-4   |
| 8                | 277       | 3.2820286690773027E-4  |
| 2                | 12785787  | 15.14921281253281      |
| 5                | 3398212   | 4.026364334874556      |
| 3                | 3583919   | 4.246398882906447      |
```

```
| 6                 | 2039148   | 2.4160802153399437     |
| 4                 | 1709802   | 2.025855300521917      |
| 1                 | 59108834  | 70.03497753925315      |
+-------------------+-----------+------------------------+
11 rows selected (107.236 seconds)

-- query to find min and max passengers

select min(passenger_count) as min_passenger_count,
max(passenger_count) as max_passenger_count
from yellow_taxi_2019;

-- output

+----------------------+----------------------+
| min_passenger_count  | max_passenger_count  |
+----------------------+----------------------+
| 0                    | 9                    |
+----------------------+----------------------+
1 row selected (87.029 seconds)
```

Observation

- 70% of trips has only 1 passenger.
- 15% of trips are with 2 passengers.
- as this is entered by driver 0.29% of time the value is not entered.

trip_distance

The elapsed trip distance in miles reported by the taximeter.

```
-- query

select  vendor_id, count(*) as num_trips_where_distance_lessthan_0 from
yellow_taxi_2019  where trip_distance<=0 group by vendor_id;

-- output
+------------+---------------------------------------+
| vendor_id  | num_trips_where_distance_lessthan_0   |
+------------+---------------------------------------+
| NULL       | 10668                                 |
| 2          | 369586                                |
| 4          | 1667                                  |
| 1          | 369003                                |
+------------+---------------------------------------+
4 rows selected (131.396 seconds)
```

```
-- query to find min and max trip_distance

select min(trip_distance) as min_trip_distance,
max(trip_distance) as max_trip_distance
from yellow_taxi_2019;

-- output

+--------------------+--------------------+
| min_trip_distance  | max_trip_distance  |
+--------------------+--------------------+
| -37264.53          | 45977.22           |
+--------------------+--------------------+
1 row selected (101.438 seconds)
```

Observations

- all of the records has to be removed.
- all the negative values and distance greater than 10000 miles can be removed.

RateCodeID

The final rate code in effect at the end of the trip.

1= Standard rate , 2=JFK

3=Newark , 4=Nassau or Westchester

5=Negotiated fare , 6=Group ride

```
-- query
select  rate_code_id, count(*) from  yellow_taxi_2019  group by
rate_code_id;
-- output
+---------------+-----------+
| rate_code_id  |    _c1    |
+---------------+-----------+
| NULL          | 246601    |
| 2             | 2235882   |
| 5             | 489049    |
| 3             | 190632    |
| 6             | 538       |
| 99            | 3897      |
| 4             | 66748     |
| 1             | 81165672  |
+---------------+-----------+
8 rows selected (106.423 seconds)
```

Observations

- Null and value 99 to be removed since they are note given the description.

Store_and_fwd_flag

This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server.
Y= store and forward trip
N= not a store and forward trip

```
-- query
select  store_and_fwd_flag, vendor_id, count(*) as num_records from
yellow_taxi_2019 group by store_and_fwd_flag, vendor_id;

-- output

+---------------------+------------+--------------+
| store_and_fwd_flag  | vendor_id  | num_records  |
+---------------------+------------+--------------+
|                     | NULL       | 246601       |
| Y                   | 1          | 670062       |
| Y                   | 2          | 21137        |
| N                   | 1          | 29698095     |
| N                   | 2          | 53496044     |
| N                   | 4          | 267080       |
+---------------------+------------+--------------+
6 rows selected (117.068 seconds)
```

Observation

- This seems to be correct .
- Nulls to be removed

Payment_type

A numeric code signifying how the passenger paid for the trip.
1= Credit card 2= Cash
3= No charge 4= Dispute
5= Unknown 6= Voided trip

```
--  query

select  payment_type, vendor_id, count(*) as num_records from
yellow_taxi_2019 group by payment_type, vendor_id;

-- output

+---------------+------------+--------------+
| payment_type  | vendor_id  | num_records  |
+---------------+------------+--------------+
| NULL          | NULL       | 246601       |
| 2             | 1          | 7983073      |
| 2             | 2          | 14831945     |
| 2             | 4          | 81294        |
| 5             | 1          | 33           |
| 3             | 1          | 370050       |
| 3             | 2          | 77191        |
| 4             | 1          | 115171       |
| 4             | 2          | 71392        |
| 1             | 1          | 21899830     |
| 1             | 2          | 38536653     |
| 1             | 4          | 185786       |
+---------------+------------+--------------+
12 rows selected (99.438 seconds)
```

Observation

- Values are within the range (1-6)

Fare_amount

The time-and-distance fare calculated by the meter.

```
-- query
select count(*) as num_records, vendor_id from yellow_taxi_2019 where
fare_amount<0 group by vendor_id;

-- output
+--------------+------------+
| num_records  | vendor_id  |
+--------------+------------+
| 367          | NULL       |
| 169433       | 2          |
| 25           | 1          |
+--------------+------------+
3 rows selected (86.211 seconds)
```

```
-- query to find min and max fare_amount

select min(fare_amount) as min_fare_amount,
max(fare_amount) as max_fare_amount
from yellow_taxi_2019;

-- output

+------------------+------------------+
| min_fare_amount  | max_fare_amount  |
+------------------+------------------+
| -1856.00         | 943274.80        |
+------------------+------------------+
1 row selected (111.006 seconds)
```

Observation

- Fare cannot be negative
- They need to be removed

Extra

Miscellaneous extras and surcharges.
Currently, this only includes the $0.50 and $1 rush hour and overnight charges.

```
-- query
select vendor_id, count(*) as num_records from yellow_taxi_2019 where
extra not in (0,0.5,1) group by vendor_id;

-- output

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 234542       |
| 2          | 421892       |
| 4          | 1466         |
| 1          | 25155461     |
+------------+--------------+
4 rows selected (109.676 seconds)

-- query to find min and max extra

select min(extra) as min_extra,
max(extra) as max_extra
from yellow_taxi_2019;
```

```
-- output

+------------+------------+
| min_extra  | max_extra  |
+------------+------------+
| -60.00     | 535.38     |
+------------+------------+
1 row selected (396.64 seconds)
```

Observation

- all of the above are fault records according to the dictionary and has to be removed.

mta_tax

$0.50 MTA tax that is automatically triggered based on the metered rate in use.

```
-- query

select count(*) from  yellow_taxi_2019 where mta_tax not in (0,0.5);

-- output

+---------+
|   _c0   |
+---------+
| 166074  |
+---------+
1 row selected (103.098 seconds)

-- query to find min and max mta_tax

select min(mta_tax) as min_mta_tax,
max(mta_tax) as max_mta_tax
from yellow_taxi_2019;

-- output

+--------------+--------------+
| min_mta_tax  | max_mta_tax  |
+--------------+--------------+
| -0.50        | 212.42       |
+--------------+--------------+
1 row selected (124.761 seconds)
```

Observation

- This is the fault records according to the data and has to be removed.

Improvement_surcharge

$0.30 improvement surcharge assessed trips at the flag drop.
The improvement surcharge began being levied in 2015.

```
-- query

select vendor_id, count(*) as num_records from yellow_taxi_2019
where improvement_surcharge not in (0,0.3)
group by vendor_id;

-- output

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 60           |
| 2          | 169410       |
| 4          | 1            |
+------------+--------------+
3 rows selected (106.845 seconds)

-- query to find min and max improvement_surcharge

select min(improvement_surcharge) as min_improvement_surcharge,
max(improvement_surcharge) as max_improvement_surcharge
from yellow_taxi_2019;

-- output
+----------------------------+----------------------------+
| min_improvement_surcharge  | max_improvement_surcharge  |
+----------------------------+----------------------------+
| -0.30                      | 1.00                       |
+----------------------------+----------------------------+
1 row selected (112.987 seconds)
```

Observations

- more number of faulty records in improvement_surcharge is given by vendor id 2.

Tip_amount

Tip amount – This field is automatically populated for credit card tips.
Cash tips are not included.

```
-- query

select vendor_id, count(*) as num_records from yellow_taxi_2019
where tip_amount < 0 group by vendor_id;

-- output

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 2            |
| 2          | 1825         |
+------------+--------------+
2 rows selected (107.783 seconds)

-- query to find min and max tip_amount

select min(tip_amount) as min_tip_amount,
max(tip_amount) as max_tip_amount
from yellow_taxi_2019;

-- output

+-----------------+-----------------+
| min_tip_amount  | max_tip_amount  |
+-----------------+-----------------+
| -221.00         | 141492.02       |
+-----------------+-----------------+
1 row selected (114.799 seconds)
```

Observations

- faulty data in tip_amount category is given by vendor id 2

Tolls_amount

Total amount of all tolls paid in trip.

```
-- query

select vendor_id,count(*) as num_records from yellow_taxi_2019
where tolls_amount < 0
group by vendor_id;

-- output

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| 2          | 3645         |
+------------+--------------+
1 row selected (102.447 seconds)

-- query to find min and max tolls_amount

select min(tolls_amount) as min_tolls_amount,
max(tolls_amount) as max_tolls_amount
from yellow_taxi_2019;

-- output

+--------------------+--------------------+
| min_tolls_amount   | max_tolls_amount   |
+--------------------+--------------------+
| -70.00             | 3288.00            |
+--------------------+--------------------+
1 row selected (168.942 seconds)
```

Observations

- Faulty data in the category of tolls_amount is given by vendor id 2.

Total_amount

The total amount charged to passengers. Does not include cash tips.

```
-- query

select vendor_id, count(*) as num_records from yellow_taxi_2019
where total_amount<0
group by vendor_id;
```

```
-- output

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 343          |
| 2          | 169436       |
+------------+--------------+
2 rows selected (86.917 seconds)

-- query to find min and max total_amount

select min(total_amount) as min_total_amount,
max(total_amount) as max_total_amount
from yellow_taxi_2019;

-- output

+-------------------+-------------------+
| min_total_amount  | max_total_amount  |
+-------------------+-------------------+
| -1871.80          | 1084772.17        |
+-------------------+-------------------+
1 row selected (163.446 seconds)
```

Observations

- Faulty records in this category is provided by vendor id 2.

STEP IV . Preprocessing

From above analysis, we need to do the following :

- rows with null values to be removed
- congestion_surcharge column can be removed as it having lot of null values.
- removing all of the trips which are having pickup and dropoff times outside 2019.
- remove all the rows where vendor id is 4 and NULL (as it is not specified in data dictionary),
- removing all of the trips which are having pickup and dropoff times outside 2019.
- records with tpep_dropoff_datetime<=tpep_pickup_datetime should be removed.
- when passenger count can be zero it can be treated as if driver is carrying a parcel or incorrect value. so retaining these values.
- we will reject this data with 0 or negative trip_distance and trip distance greater than 10000 miles as it doesn't make any sense.
- we will reject null and value 99 for rate_code_id
- negative fare_amount to be removed and also fare amount greater than 1000 $ (as it is very abnormal).
- extra should be in (0,0.5,1) .. removing all other values
- mta_tax should in (0,0.5) .. removing all other values
- tip_amount < 0 should be removed and tip amount greater than 100 $ seems very abnormal and can be removed.
- tip_amount should be only with payment_type = 1
- tolls_amount < 0 should be removed
- improvement_surcharge in (0,0.3)
- total_amount < 0 should be removed and total_amount greater than 10000$ can be removed.

```
+-------------------------+----------------+----------+
|        col_name         |   data_type    | comment  |
+-------------------------+----------------+----------+
| vendor_id               | int            |          |
| tpep_pickup_datetime    | timestamp      |          |
| tpep_dropoff_datetime   | timestamp      |          |
| passenger_count         | int            |          |
| trip_distance           | decimal(10,2)  |          |
| rate_code_id            | int            |          |
| store_and_fwd_flag      | string         |          |
| pulocationid            | int            |          |
| dolocationid            | int            |          |
| payment_type            | int            |          |
| fare_amount             | decimal(10,2)  |          |
| extra                   | decimal(10,2)  |          |
| mta_tax                 | decimal(10,2)  |          |
```

```
| tip_amount             | decimal(10,2)  |          |
| tolls_amount           | decimal(10,2)  |          |
| improvement_surcharge  | decimal(10,2)  |          |
| total_amount           | decimal(10,2)  |          |
| congestion_surcharge   | int            |          |
+------------------------+----------------+----------+
18 rows selected (0.133 seconds)


-- query to do preprocessing

create table yellow_taxi_2019_processed as
select vendor_id , tpep_pickup_datetime  , tpep_dropoff_datetime  ,
passenger_count,
trip_distance, rate_code_id, store_and_fwd_flag, pulocationid,
dolocationid,
payment_type, fare_amount , extra  , mta_tax  , tip_amount  , tolls_amount
,
improvement_surcharge, total_amount
from yellow_taxi_2019
where
(vendor_id == 1 or vendor_id == 2) and
(tpep_pickup_datetime >= '2019-01-01 00:00:00.0' and tpep_pickup_datetime
< '2020-01-01 00:00:00.0') and
(tpep_dropoff_datetime >= '2019-01-01 00:00:00.0' and
tpep_dropoff_datetime < '2020-01-01 00:00:00.0') and
(tpep_dropoff_datetime > tpep_pickup_datetime) and
(passenger_count is not null) and
(trip_distance > 0 and trip_distance < 10000) and
(rate_code_id != 99) and
(fare_amount > 0 and fare_amount < 10000) and
 (extra in (0,0.5,1)) and
 (mta_tax  in (0,0.5)) and
((tip_amount >=0 and Payment_type=1) or (Payment_type!=1 and
tip_amount=0)) and
(tolls_amount >=0) and
(tip_amount < 10000) and
(improvement_surcharge in (0,0.3)) and
(total_amount > 0 and total_amount < 10000) ;

-- time taken to run
No rows affected (698.887 seconds)

-- new table is created now
```

```
+------------------------+----------------+----------+
|        col_name        |   data_type    | comment  |
+------------------------+----------------+----------+
| vendor_id              | int            |          |
| tpep_pickup_datetime   | timestamp      |          |
| tpep_dropoff_datetime  | timestamp      |          |
| store_and_fwd_flag     | string         |          |
| rate_code_id           | int            |          |
| pulocationid           | int            |          |
| dolocationid           | int            |          |
| passenger_count        | int            |          |
| trip_distance          | decimal(10,2)  |          |
| fare_amount            | decimal(10,2)  |          |
| extra                  | decimal(10,2)  |          |
| mta_tax                | decimal(10,2)  |          |
| tip_amount             | decimal(10,2)  |          |
| tolls_amount           | decimal(10,2)  |          |
| improvement_surcharge  | decimal(10,2)  |          |
| total_amount           | decimal(10,2)  |          |
| payment_type           | int            |          |
+------------------------+----------------+----------+
17 rows selected (0.073 seconds)

-- query

select count(*) num_total_records from yellow_taxi_2019_processed;

-- output
+--------------------+
| num_total_records  |
+--------------------+
| 57610485           |
+--------------------+
1 row selected (0.115 seconds)
```

- 57610485 / 84399019 --- retained rows after preprocessing / initial number of rows
- 68.25% of rows are retained after preprocessing.

STEP V. Execute queries :

Q1) Which vendor provides the most useful data?

```
-- query

-- before preprocessing

select vendor_id, count(*) as num_records, count(*) * 100 / 84399019 as
rows_percentage
from yellow_taxi_2019
group by vendor_id;

+------------+-----------+----------------------+
| vendor_id  |    _c1    |         _c2          |
+------------+-----------+----------------------+
| NULL       | 246601    | 0.2921846757484231   |
| 2          | 53517181  | 63.40971925277947    |
| 4          | 267080    | 0.31644917578959064  |
| 1          | 30368157  | 35.981646895682516   |
+------------+-----------+----------------------+
4 rows selected (131.501 seconds)


-- after preprocessing

select vendor_id, count(*) as num_records, count(*) * 100 / 57610485
from yellow_taxi_2019_processed
group by vendor_id;

+------------+-------------+--------------------+
| vendor_id  | num_records |        _c2         |
+------------+-------------+--------------------+
| 2          | 52662509    | 91.41132729571709  |
| 1          | 4947976     | 8.58867270428291   |
+------------+-------------+--------------------+
2 rows selected (43.913 seconds)
```

Observations

- Removed rows provided by vendor 1 : (30368157 - 4947976) / 30368157 : 83% rows are removed
- Removed rows provided by vendor 2 : (53517181 - 52662509) / 53517181 : 15% rows are removed.

```
-- tpep_pickup_datetime

+--------------------------+------------+
| num_of_wrong_date_trips  | vendor_id  |
+--------------------------+------------+
| 1442                     | 2          |
+--------------------------+------------+
1 row selected (155.463 seconds)

-- tpep_dropoff_datetime

+--------------------------+------------+
| num_of_wrong_date_trips  | vendor_id  |
+--------------------------+------------+
| 9                        | NULL       |
| 2265                     | 2          |
| 233                      | 1          |
+--------------------------+------------+
3 rows selected (180.105 seconds)

-- tpep_dropoff_datetime <= tpep_pickup_datetime

+--------+------------+
|  _c0   | vendor_id  |
+--------+------------+
| 1323   | NULL       |
| 8595   | 2          |
| 15     | 4          |
| 67478  | 1          |
+--------+------------+
4 rows selected (179.612 seconds)
```

- lot of faulty data by vendor id 1 when dropoff time is before pick up time

```
-- trip_distance

+------------+---------------------------------------+
| vendor_id  | num_trips_where_distance_lessthan_0   |
+------------+---------------------------------------+
| NULL       | 10668                                 |
| 2          | 369586                                |
| 4          | 1667                                  |
| 1          | 369003                                |
+------------+---------------------------------------+
4 rows selected (131.396 seconds)
```

- lot of faulty data provided by vendor id 1
- even though we have same amount of faulty rows from vendor id 1 and 2 as data rows is very less provided by vendor id 1 compared to vendor id 2.

```
-- fare_amount < 0

+--------------+------------+
| num_records  | vendor_id  |
+--------------+------------+
| 367          | NULL       |
| 169433       | 2          |
| 25           | 1          |
+--------------+------------+
3 rows selected (86.211 seconds)
```

- in the aspect of fare_amount vendor id 2 data is very faulty.

```
-- extra not in (0,0.5,1)

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 234542       |
| 2          | 421892       |
| 4          | 1466         |
| 1          | 25155461     |
+------------+--------------+
4 rows selected (109.676 seconds)
```

- lot of faulty data provided by vendor id 1.
- we are losing a lot of data from vendor id 1.

```
-- improvement_surcharge not in (0,0.3)

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 60           |
| 2          | 169410       |
| 4          | 1            |
+------------+--------------+
3 rows selected (106.845 seconds)
```

- faulty data provided by vendor id 2 in this aspect.

```
-- tip_amount < 0

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 2            |
| 2          | 1825         |
+------------+--------------+
2 rows selected (107.783 seconds)
```

- faulty data provided by vendor id 2 wrt tip_amount

```
-- tolls_amount < 0

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| 2          | 3645         |
+------------+--------------+
1 row selected (102.447 seconds)
```

- faulty data provided by vendor id 2 wrt tolls_amount

```
-- total_amount < 0

+------------+--------------+
| vendor_id  | num_records  |
+------------+--------------+
| NULL       | 343          |
| 2          | 169436       |
+------------+--------------+
2 rows selected (86.917 seconds)
```

- faulty data provided by vendor 2 wrt total_amount

Conclusion

- If we look into quality wrt to overall categories then vendor 1 provides better data.
- If we look into number of rows that is removed then vendor 2 provides more and better data.

Overall, we can say that vendor 2 provides better data (even though this violates in all categories more number of data points are provided).

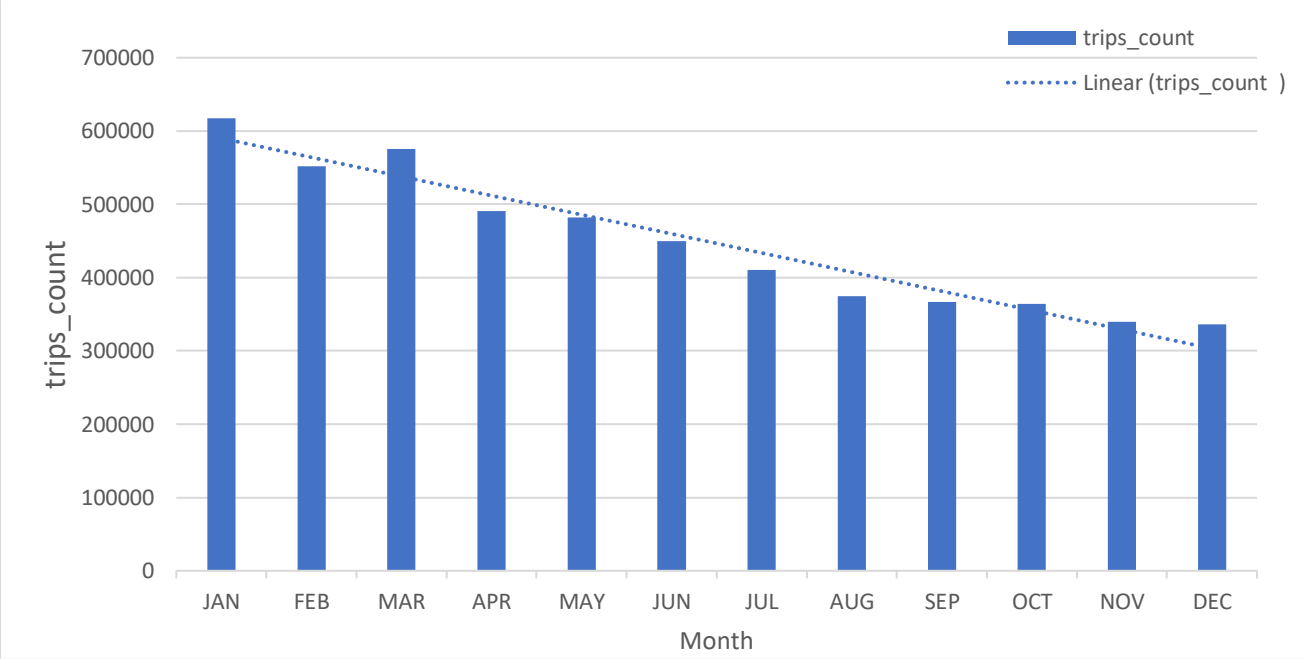
Q2) Find the month wise trip count, average distance and average passenger count from the trips completed by yellow taxis in 2019. Summary visualizations will be preferred for better analysis.

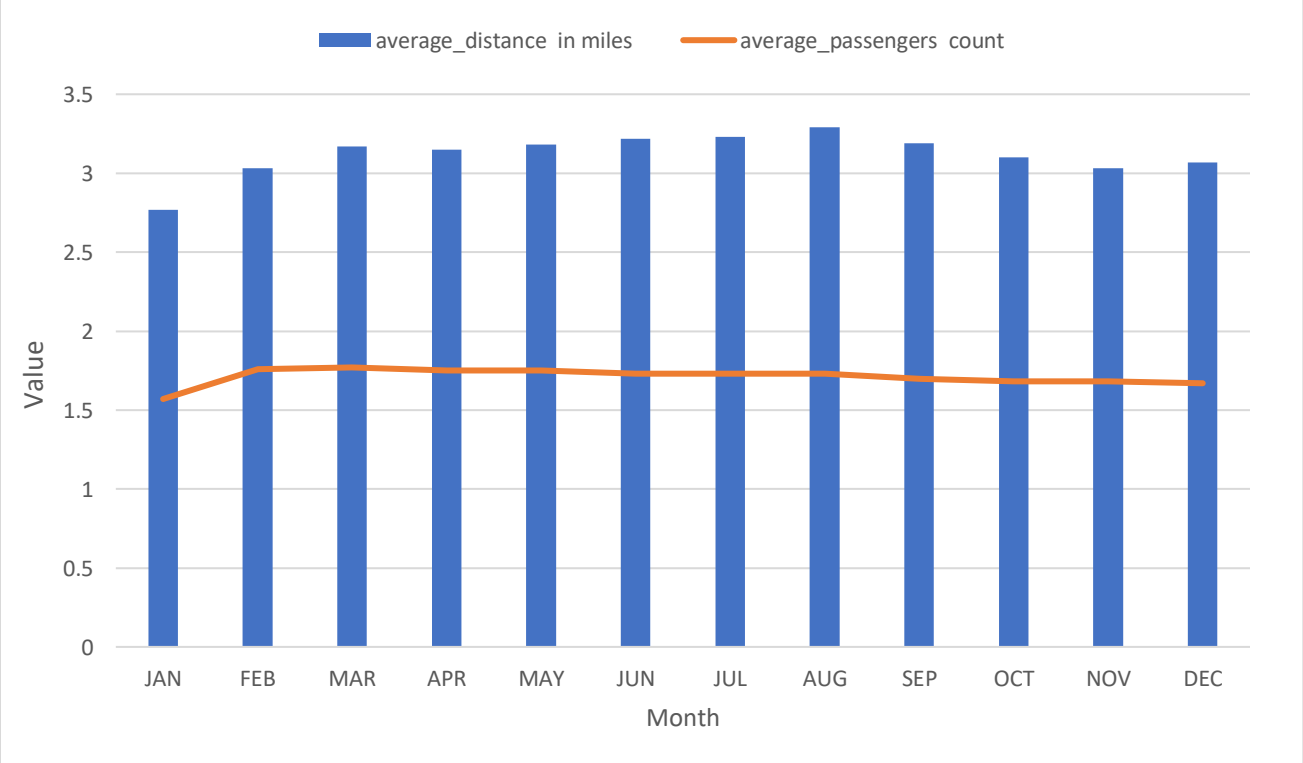
```
-- query
SELECT MONTH(tpep_pickup_datetime) month, COUNT(*) trips_Count,
ROUND(AVG(trip_distance),2)
average_distance ,AVG(passenger_count) average_passengers from
yellow_taxi_2019_processed
group by MONTH(tpep_pickup_datetime)
order by month;

-- output
+--------+--------------+--------------------+----------------------+
| month  | trips_count  | average_distance   | average_passengers   |
+--------+--------------+--------------------+----------------------+
| 1      | 7499009      | 2.77               | 1.5740138463628994   |
| 2      | 4575650      | 3.03               | 1.7623528897533685   |
| 3      | 4988243      | 3.17               | 1.7727510468114724   |
| 4      | 4752865      | 3.15               | 1.7598700994032022   |
| 5      | 4886482      | 3.18               | 1.7512210215856725   |
| 6      | 4506873      | 3.22               | 1.7366981496927028   |
| 7      | 4126897      | 3.23               | 1.7351065461531994   |
| 8      | 3980329      | 3.29               | 1.7304222841880659   |
| 9      | 4320090      | 3.19               | 1.7089257399730098   |
| 10     | 4760863      | 3.10               | 1.689583170110125    |
| 11     | 4579514      | 3.03               | 1.683790463354845    |
| 12     | 4633670      | 3.07               | 1.6749291598236387   |
+--------+--------------+--------------------+----------------------+
12 rows selected (107.53 seconds)
```

Observation :

- January had the most number of taxi trip
- Trip count gradually decreases from Jan to Dec.
- Average Passenger Count remains almost same over the year
- Average Distance of a taxi trip is about 3 miles .

Summary visualizations:





Q3) Find out the five busiest routes served by the yellow taxis during 2019. The name of start and drop points to be provided.

```
-- query

select pulocationid, dolocationid ,count(*) as trip_count
from  yellow_taxi_2019_processed
group by pulocationid, dolocationid
order by trip_count desc LIMIT 5;

-- output

+---------------+---------------+-------------+
| pulocationid  | dolocationid  | trip_count  |
+---------------+---------------+-------------+
| 264           | 264           | 526281      |
| 237           | 236           | 351673      |
| 236           | 237           | 296713      |
| 236           | 236           | 295942      |
| 237           | 237           | 284339      |
+---------------+---------------+-------------+
5 rows selected (63.367 seconds)
```

Top 5 busiest routes fall in area :

1. 264 (Area name not specified in lookup table)
2. 236 (Manhattan,Upper East Side North ,Yellow Zone)
3. 237 (Manhattan,Upper East Side South ,Yellow Zone)

Q4) What are the top 3 busiest hours of the day for the taxis? Top 3 hours having maximum trip count.

```
-- query

select HOUR(tpep_pickup_datetime) hour , count(*) as trip_count
from yellow_taxi_2019_processed
group by HOUR(tpep_pickup_datetime)
order by trip_count desc LIMIT 3;
```

```
-- output

+-------+-------------+
| hour  | trip_count  |
+-------+-------------+
| 18    | 3728121     |
| 19    | 3545876     |
| 17    | 3327857     |
+-------+-------------+
3 rows selected (72.002 seconds)
```

Q5) What is the most preferred way of payment used by the passengers? What are the weekly trends observed for the methods of payments?

```
-- query

select payment_type, count(*) as num_count
from bds.yellow_taxi_2019_processed
group by payment_type
order by num_count desc;

-- output

+---------------+------------+
| payment_type  | num_count  |
+---------------+------------+
| 1             | 41371307   |
| 2             | 16140685   |
| 3             | 74598      |
| 4             | 23895      |
+---------------+------------+
                   4  rows selected (41.326 seconds)
```

- Looks like credit card is mostly used form of payment followed by cash.

```
-- query

select weekofyear(tpep_pickup_datetime) as week_no, payment_type ,count(*)
num_records
from yellow_taxi_2019_processed
```

```
group by weekofyear(tpep_pickup_datetime), payment_type
order by week_no , payment_type LIMIT 35;

-- output

+----------+---------------+--------------+
| week_no  | payment_type  | num_records  |
+----------+---------------+--------------+
| 1        | 1             | 983613       |
| 1        | 2             | 499768       |
| 1        | 3             | 4616         |
| 1        | 4             | 1669         |
| 2        | 1             | 1261979      |
| 2        | 2             | 472417       |
| 2        | 3             | 5395         |
| 2        | 4             | 1696         |
| 3        | 1             | 1255511      |
| 3        | 2             | 468104       |
| 3        | 3             | 5375         |
| 3        | 4             | 1655         |
| 4        | 1             | 1260237      |
| 4        | 2             | 465572       |
| 4        | 3             | 5619         |
| 4        | 4             | 1638         |
| 5        | 1             | 1240715      |
| 5        | 2             | 429398       |
| 5        | 3             | 4657         |
| 5        | 4             | 1347         |
| 6        | 1             | 851285       |
| 6        | 2             | 291348       |
| 6        | 3             | 932          |
| 6        | 4             | 270          |
| 7        | 1             | 794814       |
| 7        | 2             | 286920       |
| 7        | 3             | 873          |
| 7        | 4             | 302          |
| 8        | 1             | 766436       |
| 8        | 2             | 278182       |
| 8        | 3             | 799          |
| 8        | 4             | 278          |
| 9        | 1             | 851709       |
| 9        | 2             | 298498       |
| 9        | 3             | 994          |
+----------+---------------+--------------+
35 rows selected (74.366 seconds)
```

- This shows the trend of the payment type based on week numbers.

```
-- query

select date_format(tpep_pickup_datetime ,'u') as day_of_week, payment_type
,count(*) num_records
from yellow_taxi_2019_processed
group by date_format(tpep_pickup_datetime ,'u'), payment_type
order by day_of_week , payment_type ,num_records desc LIMIT 35;

-- output

+--------------+---------------+--------------+
| day_of_week  | payment_type  | num_records  |
+--------------+---------------+--------------+
| 1            | 1             | 5279210      |
| 1            | 2             | 2070422      |
| 1            | 3             | 9760         |
| 1            | 4             | 2995         |
| 2            | 1             | 6124795      |
| 2            | 2             | 2239739      |
| 2            | 3             | 10415        |
| 2            | 4             | 3224         |
| 3            | 1             | 6310206      |
| 3            | 2             | 2244812      |
| 3            | 3             | 10544        |
| 3            | 4             | 3190         |
| 4            | 1             | 6548102      |
| 4            | 2             | 2319327      |
| 4            | 3             | 11086        |
| 4            | 4             | 3534         |
| 5            | 1             | 6346023      |
| 5            | 2             | 2453753      |
| 5            | 3             | 11851        |
| 5            | 4             | 3814         |
| 6            | 1             | 5749316      |
| 6            | 2             | 2594959      |
| 6            | 3             | 10725        |
| 6            | 4             | 3547         |
| 7            | 1             | 5013655      |
| 7            | 2             | 2217673      |
| 7            | 3             | 10217        |
| 7            | 4             | 3591         |
+--------------+---------------+--------------+
28 rows selected (90.137 seconds)
```

- This shows the trend of the payment type based on day of the week.

About weekly trends :

- Credit card(1) and cash(2) are the preferred type of payment
- There are no unknown mode of payment(5) and null voided trip(6)
- weeks with holidays have less trips than normal weeks
- Payment count increases from Monday to Fridays and starts dropping then.