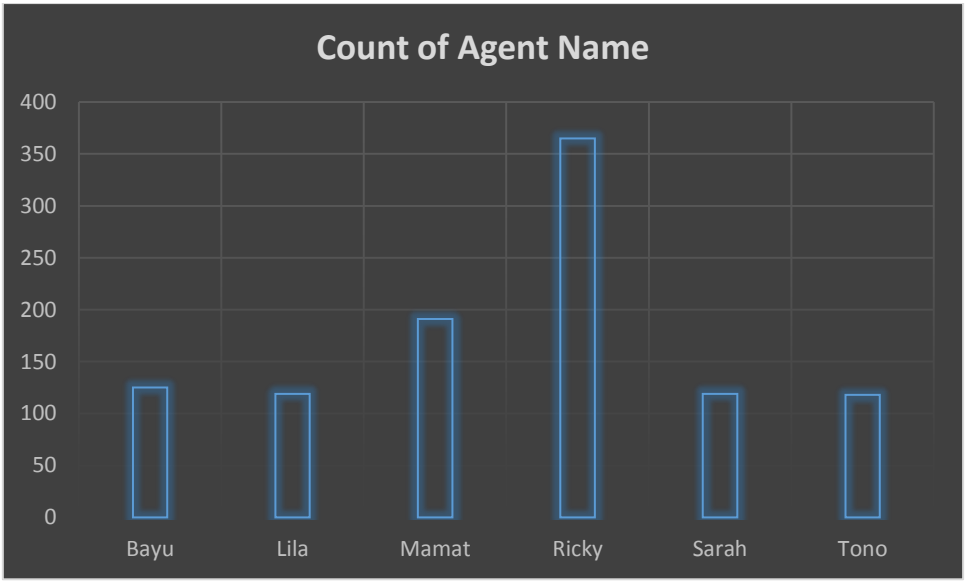
GO-JEK FRAUD ANALYST ASSIGNMENT

Data Analysis:

Problem A:



Number of drivers registered by each agent :

Ricky : 361

Mamat : 191

Tono : 117

Bayu : 125

Lila : 119

Sarah : 118

Average number of drivers registered by the agents (8-14 May 2017) : 173

Average number of drivers registered by the agents – Excluding Ricky(8-14 May 2017) : 134

We can clearly see that the numbers generated by Ricky is twice the group average and almost thrice if we calculate the average excluding Ricky's contribution.

The hypothesis that we can derive from the above analysis can be two:

i) Apart from Ricky, to an extent Mamat, others are not fast enough or efficient in registering the drivers

ii) Otherwise, the numbers generated by Ricky and Mamat are anomalous and requires further probe into the case.

Digging deeper into the data, we found the following insights:

i) All agents apart from Ricky and Mamat , work between 10 am – 5pm on most of the days with the maximum time being 7pm on the last day of the week (Friday) . Where as , Ricky and Mamat's logs shows that lot of their registrations are post 8 Pm and some even stretching until midnight 12.

ii) All agents apart from Ricky and Mamat, have worked between 8th may 2017 (Monday) to 12th May (2017). Where as Ricky and Mamat's logs show that these both have shown records registering drivers over the weekend.

iii) All the records after the timestamp **2017-05-12 19:55:11.466+07**, belongs to Ricky or Mamat.

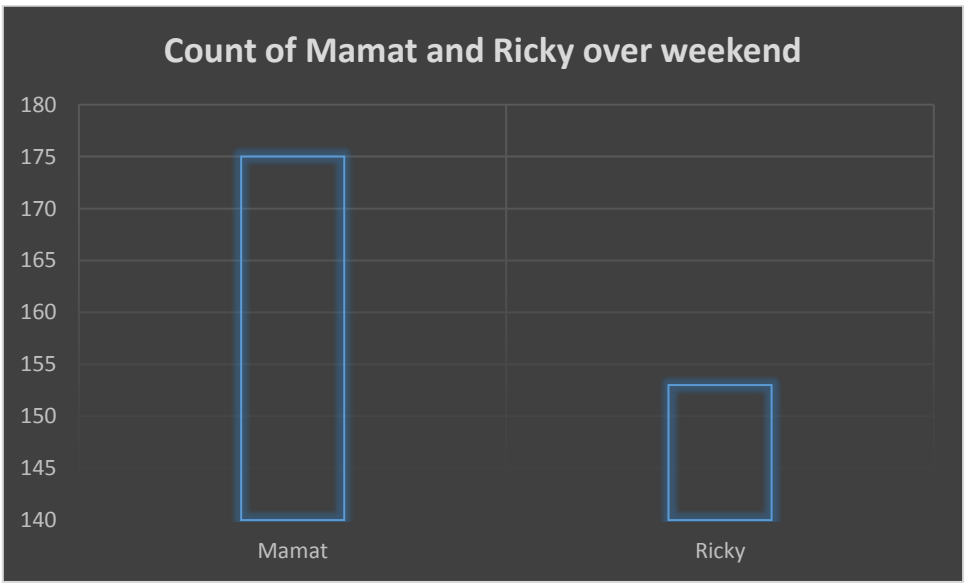
iv) Interestingly, Both Ricky and Mahat never had any over lapse of timestamp. Meaning when one was present the other wasn't. This might be a plan between Mamat and Ricky to generate more numbers over the weekend.

v) If we dig deeper into Ricky's number we can see that at times he has registered drivers in millisecond speeds which is impossible for a human. This might be because of some bot programming or activity that has generated such anomalous numbers.

1. **Ricky's sample numbers:**

2017-05-13 19:59:51.061+07
2017-05-13 19:59:51.079+07
2017-05-13 19:59:51.12+07
2017-05-13 19:59:51.154+07
2017-05-13 19:59:51.25+07
2017-05-13 19:59:51.268+07
2017-05-13 19:59:51.306+07
2017-05-13 19:59:51.387+07

Count of Ricky and Mamat over weekend :



From this we can see that most of the numbers generated by Ricky and Mamat were during the weekend which demands more probe.

Especially Mamat, **91%** of Mamat's total numbers were generated during the weekend implying Mamat was not active during the weekdays where as **42%** of Ricky's numbers were generated during the weekend.

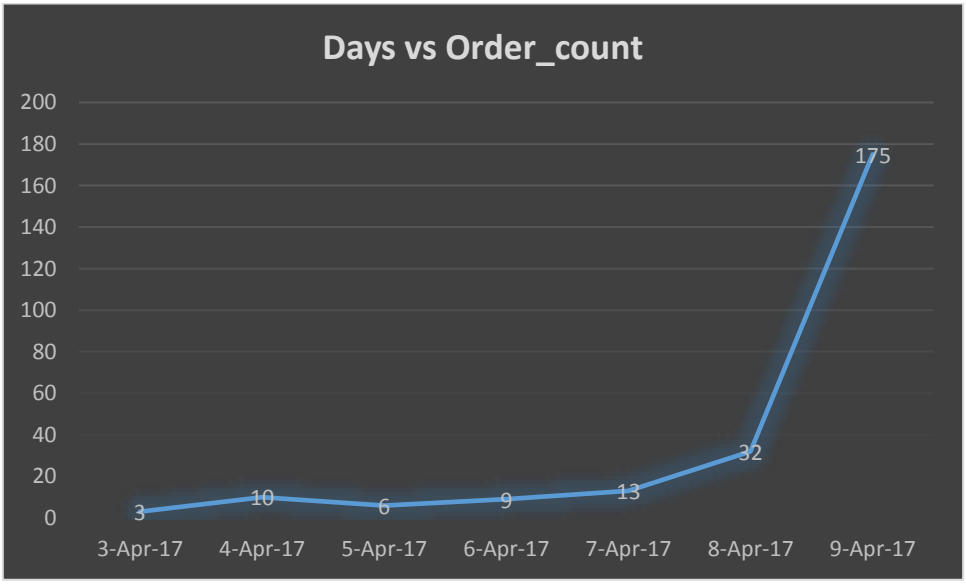
So from the above analysis , its clear that all other agents apart from Ricky and Mamat had performed their duties and task actively over the week where as Ricky and Mamat has done some illegal activities over the weekend and late evenings to boost up their number which is very evident from the data.

Therefore more probing has to be done on Ricky and Mamat to understand this Anamoly.

Problem B:

First let us see how the food ordering data is distributed among the dates given:

Date	Order_Count
3-Apr-2017	3
4-Apr-2017	10
5-Apr-2017	6
6-Apr-2017	9
7-Apr-2017	13
8-Apr-2017	32
9-Apr-2017	175



From the above data and graph it is clearly evident that there is a sudden spike in the order count on 9th april 2017.

Average order value per day at Warkop ABC : 35

Average order value per day at Warkop ABC excluding 9th apr : 12

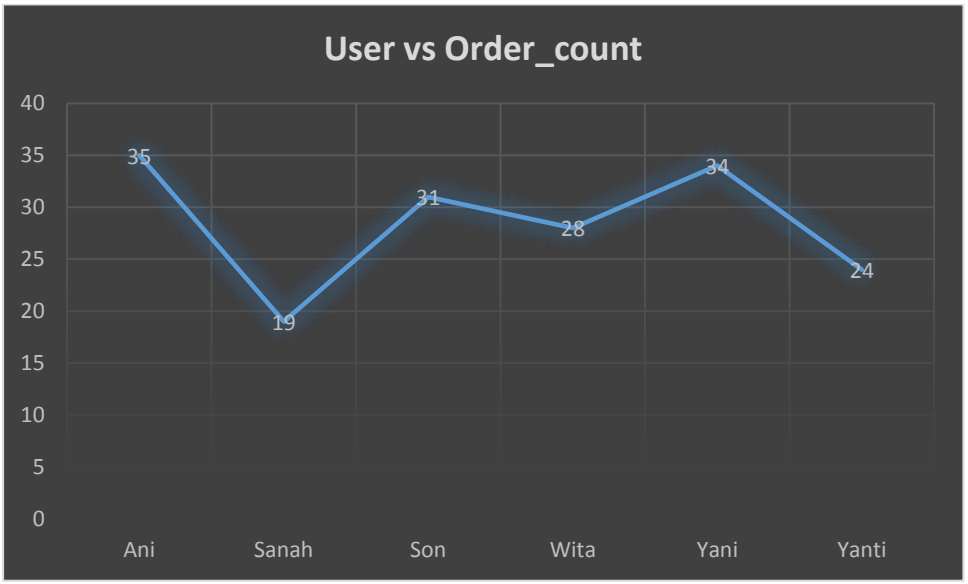
So therefore it is clearly evident that something fraudulent has happened on 9th of april which has contributed to this huge spike.

To analyse further , we drill down deeper into the transactions that has happened on 9th of april, 2017.

By analysing the data further, we can obtain the following insights:

i) The following six users have been responsible for the outrageous order activities on 9th of april

Customer_id	Customer_name	Order_count
249894626	Ani	35
249908000	Sanah	19
249903719	Son	31
249894289	Wita	28
249895739	Yani	34
249747620	Yanti	24



These six users have contributed to 97% of the total order count on 9th april, 2017.

ii) Another interesting point to note is, the driver associated with all these delivery is the same.

Driver name: HARRY

Driver id: 364640292

iii) Another interesting pattern to note is the change in the location of order. All these above mentioned users order few orders from one place and then go to a nearby place to order again. Even though they change their location of ordering a bit, still it comes under the place Kota Depok

iv) All these orders started by the midnight (12 AM) of 9th april and ended after two hours and 18 min (2.18 AM) of 9th april.

Which means about **171 orders in 138 min -> 1.31 orders per minute** delivered by the same driver to 6 different people around the same location. It would be humanly impossible to deliver 1.31 orders per minute to 6 different people.This stat clearly shows that there has been some fraudulent activities that has happened on 9th april with respect to the customer-ids and the driver_id listed above.

To analyse further , We see the second highest order count on 8th of april. As expected , the same set of people (Except Ani) has created the small hike in the order and once again the same driver is associated with all the orders.

To understand and rectify this situation, further probe has to be taken involving the driver and the users listed above to prevent any such irregular activities from taking place.

SQL:

I have used **Postgress SQL** for performing the SQL querying

Problem C:

// Selecting order_no between booking and cancellation for a user

```
with T1 as (
select t1.order_no
from Bookings as t1,Bookings as t2
where t1.booking_time between t2.booking_time and t2.cancel_time
group by t1.user_id),
```

// Selecting order_no between booking and completion for a user

```
T2 as (select t1.order_no
from Bookings as t1,Bookings as t2
where t1.booking_time between t2.booking_time and t2.complete_time
group by t1.user_id)
```

// Selecting unique order numbers from both tables

```
select order_no from T1
union
select order_no from T2
order by order_no asc
```

Problem D:

// Create a temp table to filter out records of completed GO-CAR rides in the last 30 days

```
with t1 as (
select b.customer_id,b.driver_id,d.first_name,d.last_name
from Bookings as b join Drivers as d
on b.driver_id = d.id
and b.service_type='GO-CAR'
and b.status='completed'
and DATE_PART('day',current_timestamp-b.booking_time)<=30 ),
```

// Create a temp table to store the count of each customer

```
t2 as (
select customer_id, count(customer_id) as c_count
from t1
group by 1 ),
```

// Create a temp table to store the count of how many times the driver has served a particular customer

```
t3 as (
select customer_id,driver_id,count(driver_id) as d_count
from t1
group by customer_id,driver_id),
```

// Create a temp table to store the driver to customer ratio percentage

```
t4 as (
select t2.customer_id,driver_id,(d_count*100/c_count) as percentage_served
from t2 join t3
on t2.customer_id = t3.customer_id
)
```

// Select the distinct driver id that has over 60% percentage_served and the associated name with the ids

```
select d.first_name,d.last_name
from DRIVERS as d
where d.id in (select distinct(driver_id) from t4 where t4.percentage_served >=60)
order by 1 asc
```

Fraud Domain Knowledge:

Problem E:

(a) What kind of fraudulent activities could be done with multiple accounts?

With respect to customers following irregularities may happen when they have multiple accounts:

i) Excessive use of the promos and offer. One of the main reasons for any customer to hold multiple accounts is to multiply the promos and offers.

For ex. If an use has two accounts and if the merchant is providing a signup promo ,say 10$ he/she can make use of the promo twice. This is one of the most common misuse of accounts that happen.

ii) Another potential misuse is account blocking. Imagine a scenario where a user account is blocked for some irregularities. In such cases, if the user using the account has multiple accounts , then only the old account gets blocked where as the user still remains active via new account and might create the same trouble again.

iii) Fraudulent referral.
   a. If an user holds multiple accounts , then he/she can refer thyself and therefore earn offers and rewards on both the accounts.

With respect to drivers following irregularities may happen when they have multiple accounts:

i) With multiple driver accounts, a driver can try to get more rides than other drivers by using two or more accounts.
ii) If the driver ratings are less on one account then he can still continue driving using the other account even though the driver remains the same
iii) In some ride-hailing apps, if the driver has less acceptance rate then he will be timed out for a while. In such scenarios, he can use the other account to be on road and hence surpass the timeout effect
iv) If a driver has multiple customer accounts, he/she can try to artificially surge up the price by requesting for cab in the user account and then increasing the price in the near by neighbourhood.

What kind of datapoints would you collect to identify these duplicate accounts, and what methods or thresholds would you set?

Even though it is nearly impossible to prevent user from multiple accounts, we can reduce the number of accounts an user can hold by using the following data points.

i) Email
ii) Phone Number
iii) Credit Card details (For Payment)
iv) ip-address/ Mobile device details
v) Set a tracking cookie and log it value while login and look for multiple logins from the same cookie value
vi) Some unique ID for KYC
vii) Similar first name/last name and Password for multiple accounts

If we can efficiently set up a signup mechanism using the above datapoints, it should help us reduce the multiple accounts problem.

For accounts post signup, login patterns and device features can be monitored for identifying multiple accounts.

(c) From an operational and also a data perspective, what recommendations would you make to prevent them?

When an user signup for an account, along with name, Email, Phone number and Credit Card details some unique ID can be asked for ID proof and that can be used for signup. Since the unique ID holds

some unique identifier value , it would be difficult for the user to create multiple accounts as his/her ID would already be registered.

Similarly, for already existing accounts, we can also create user account clusters using the ip address and user-agent the user logs in. If we tend to find the same ip-address/user-agent for multiple accounts, then the user can be asked to prove his unique identity in order to sustain the usage of his/her account.

Try to find behavioural patterns in accounts and if different accounts have similar patterns then it might also lead to preventing multiple user accounts.