

# **FINAL PROJECT**

## **NYC CITI BIKE**

**Northeastern University**

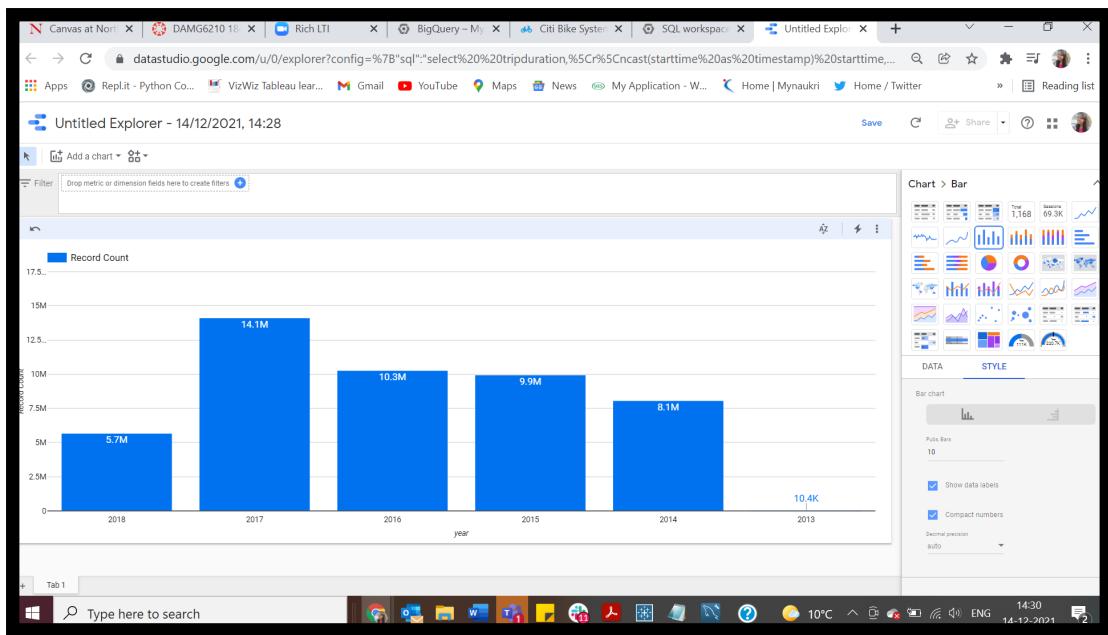
Data Management and Data Design

Semester-I (Sept 2021-Dec 2021)

**Target:** Analysis of Business Requirement Questions on public NYC Citi Bike Dataset

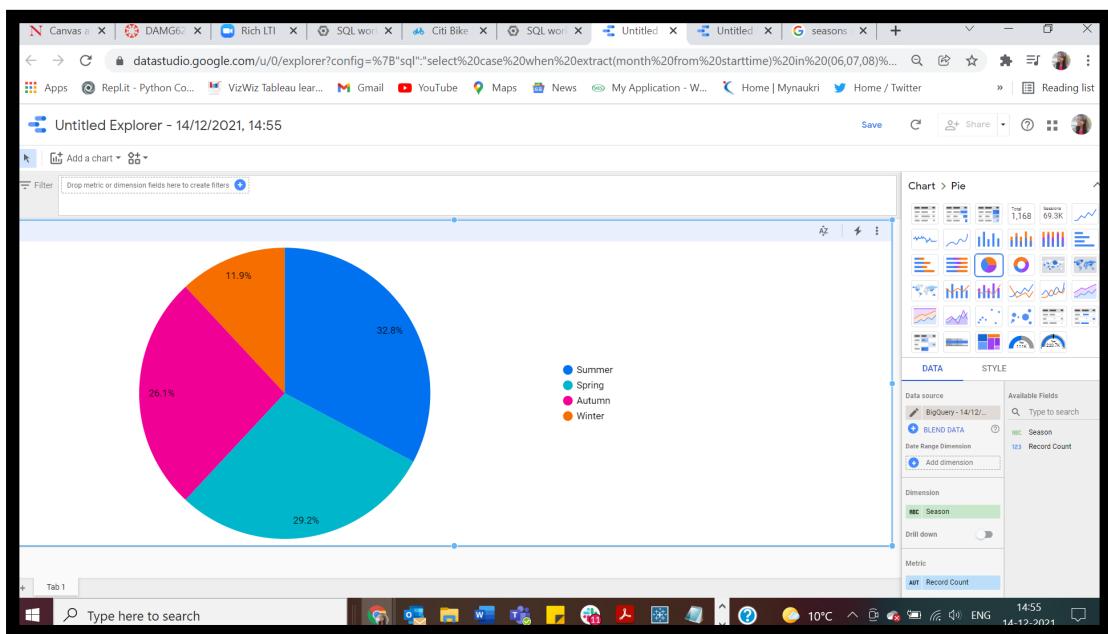
### **1. Trend year over year**

```
select tripduration,
       cast(starttime as timestamp) starttime,
       cast(stoptime as timestamp) stoptime,
       extract(year from starttime) as year,
       start_station_id,start_station_name, start_station_latitude,start_station_longitude,
       end_station_id, end_station_name, end_station_latitude,end_station_longitude
       ,
       bikeid, usertype, birth_year, gender, customer_plan
from `bigquery-public-data.new_york_citibike.citibike_trips`
where starttime > '2013-12-31'
order by starttime desc
```



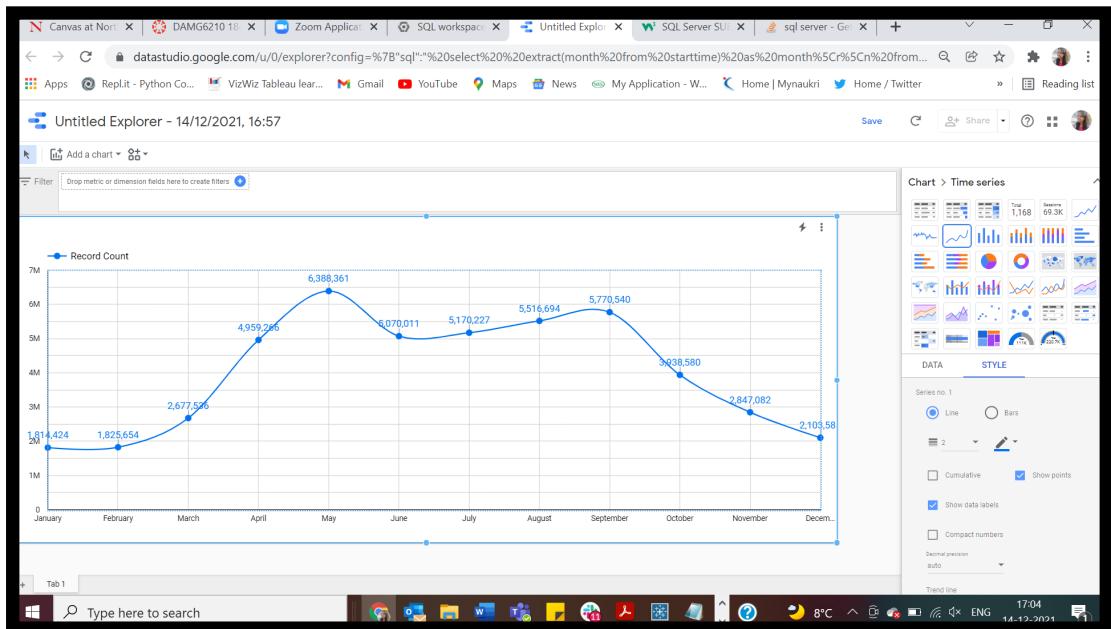
## 2. Seasonality

```
select case when extract(month from starttime) in (06,07,08) then 'Summer'
when extract(month from starttime) in (09,10,11) then 'Autumn'
when extract (month from starttime) in (12,01,02) then 'Winter'
else 'Spring' end as Season from `healthy-
genre-330203.damg_dataset.citibike_trips`
```



### 3. Month-wise trend

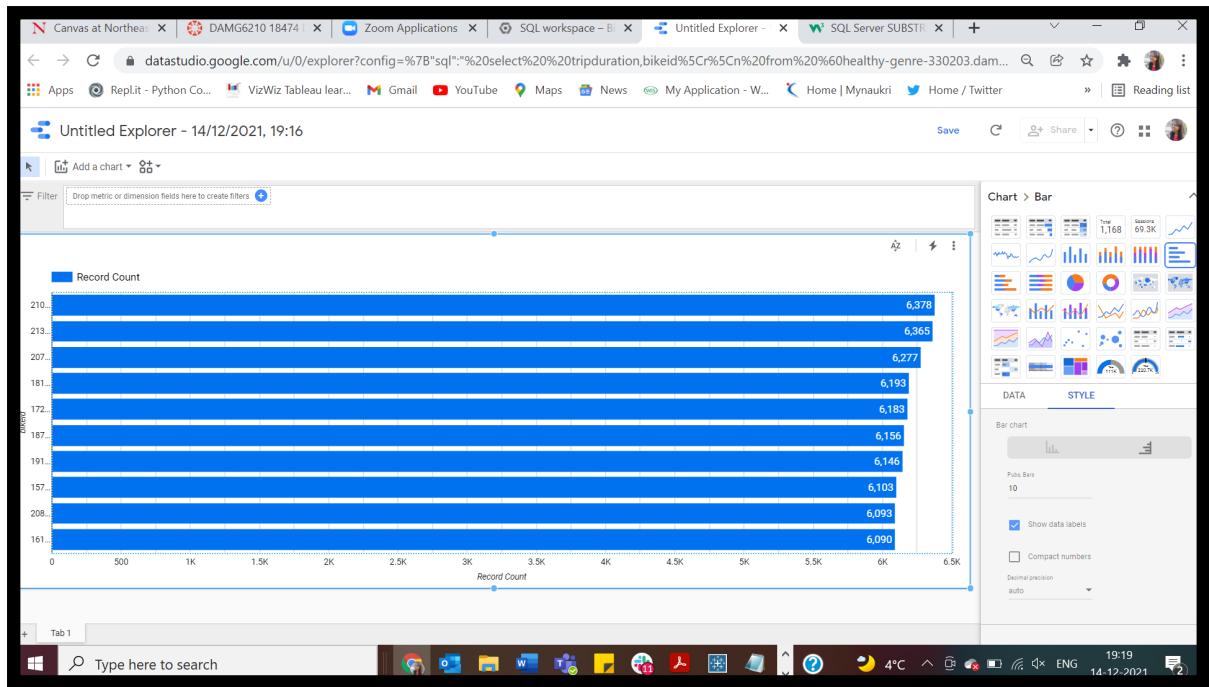
```
select extract(month from starttime) as month  
from `healthy-genre-330203.damg_datset.citibike_trips`
```



## Bike Analysis:

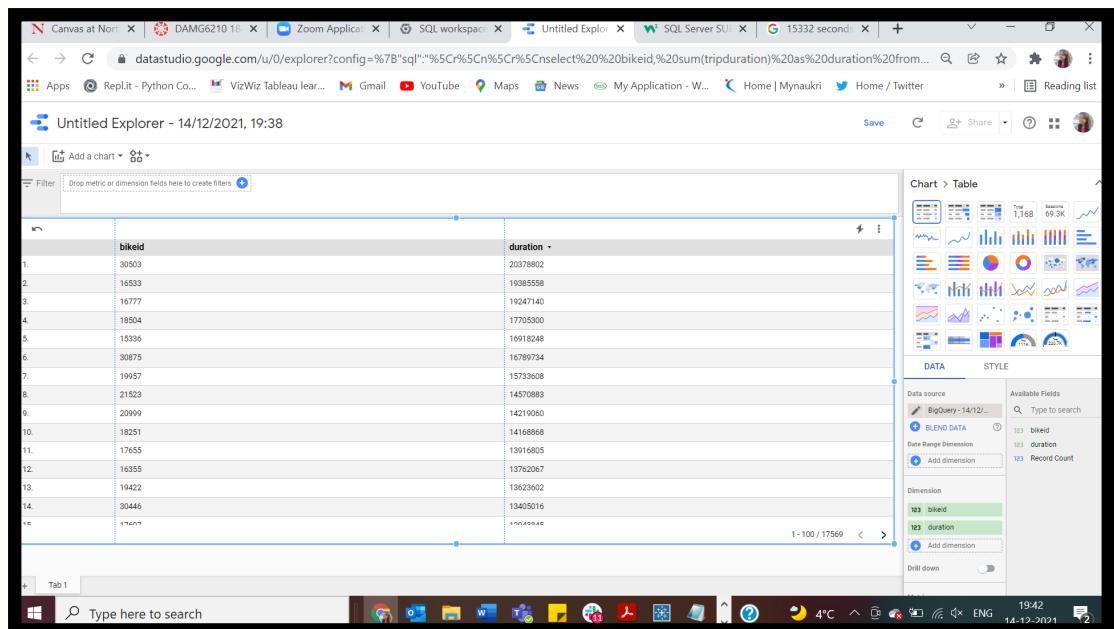
### 1. Top 5 bike ID by trips

```
select tripduration,bikeid  
from `healthy-genre-330203.damg_datset.citibike_trips`
```



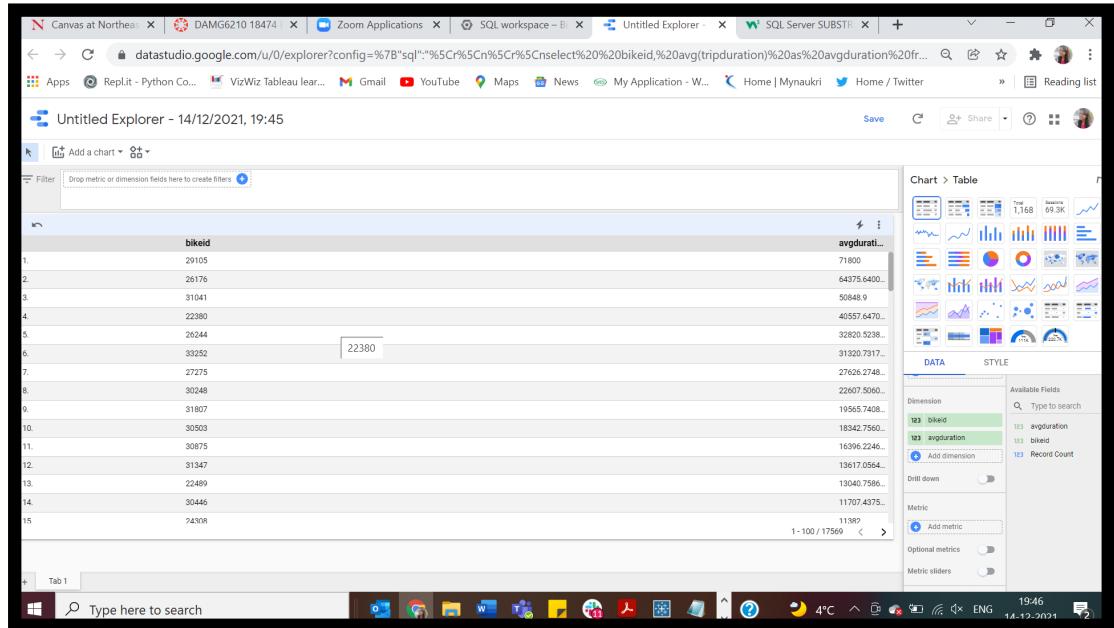
## 2. Top 5 bikes by time

```
select bikeid, sum(tripduration) as duration from `healthy-
genre-330203.damg_dataset.citibike_trips` 
group by 1
```



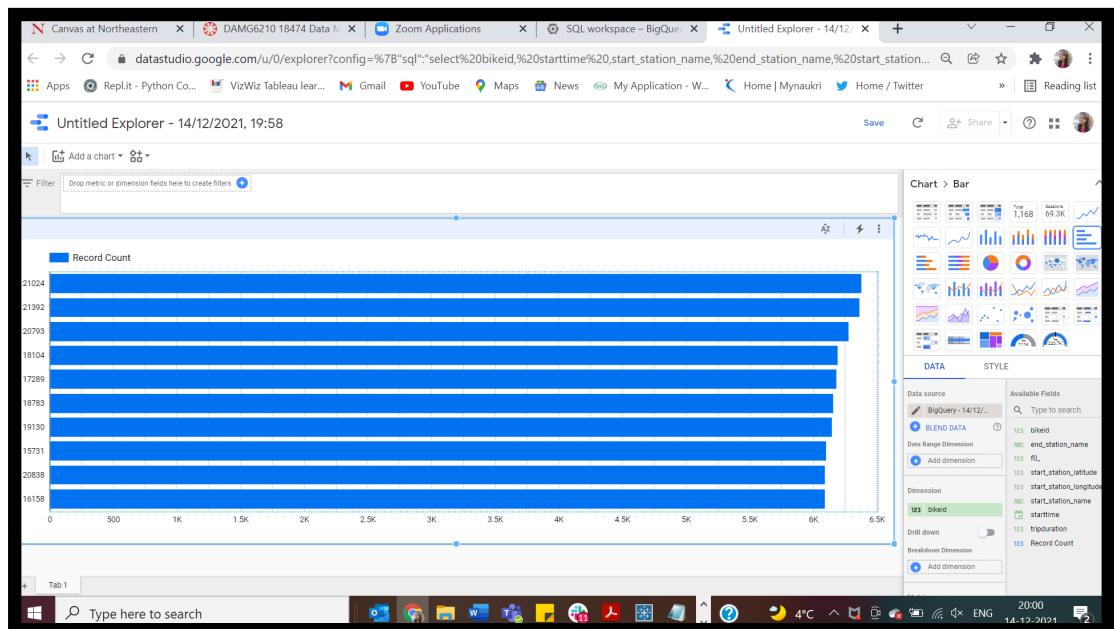
### 3. Average trip time

```
select bikeid, avg(tripduration) as avgduration from `healthy-genre-330203.damg_datset.citibike_trips`  
group by 1
```

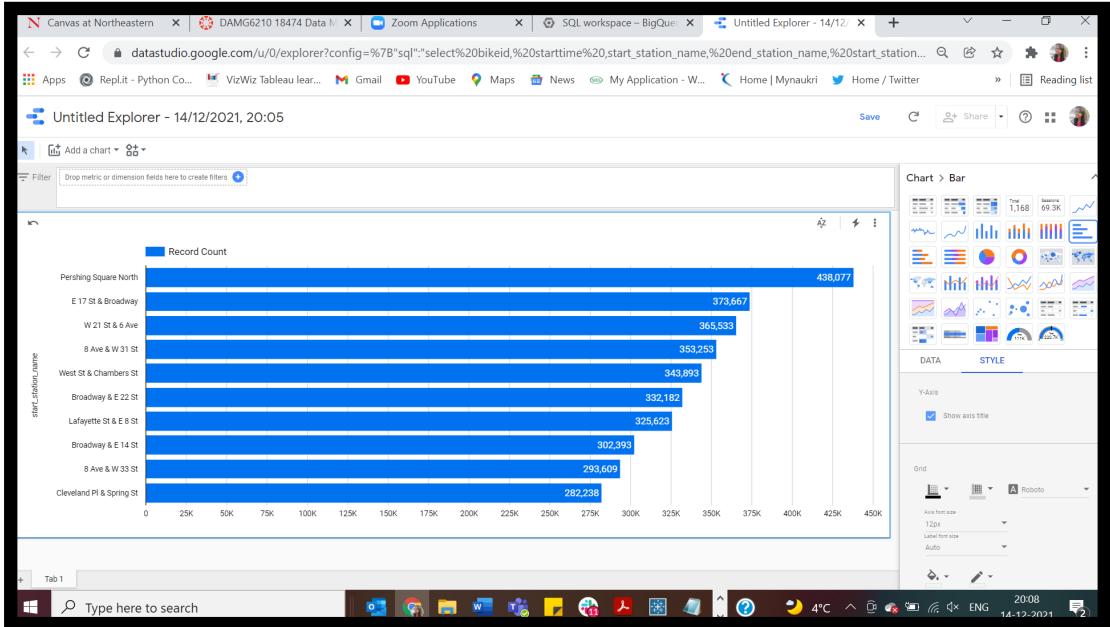


### 4. Top Bike

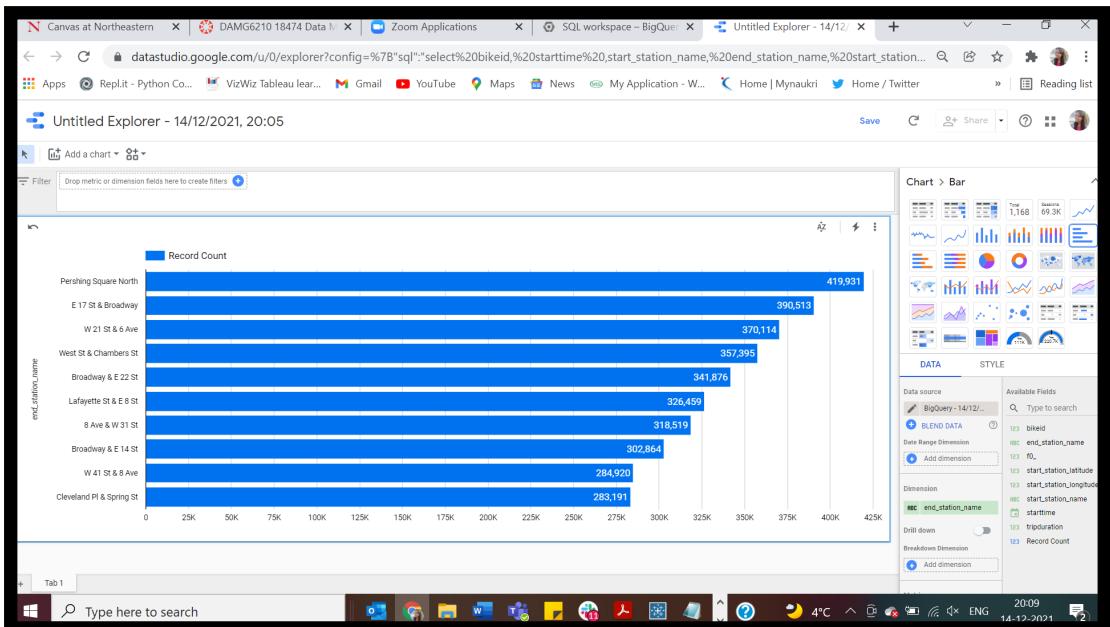
```
select bikeid, starttime ,start_station_name, end_station_name, start_statio  
n_latitude, start_station_longitude, tripduration, count(tripduration)  
from `healthy-genre-330203.damg_datset.citibike_trips`  
group by 1,2,3,4,5,6,7
```



## 5. Route- start location



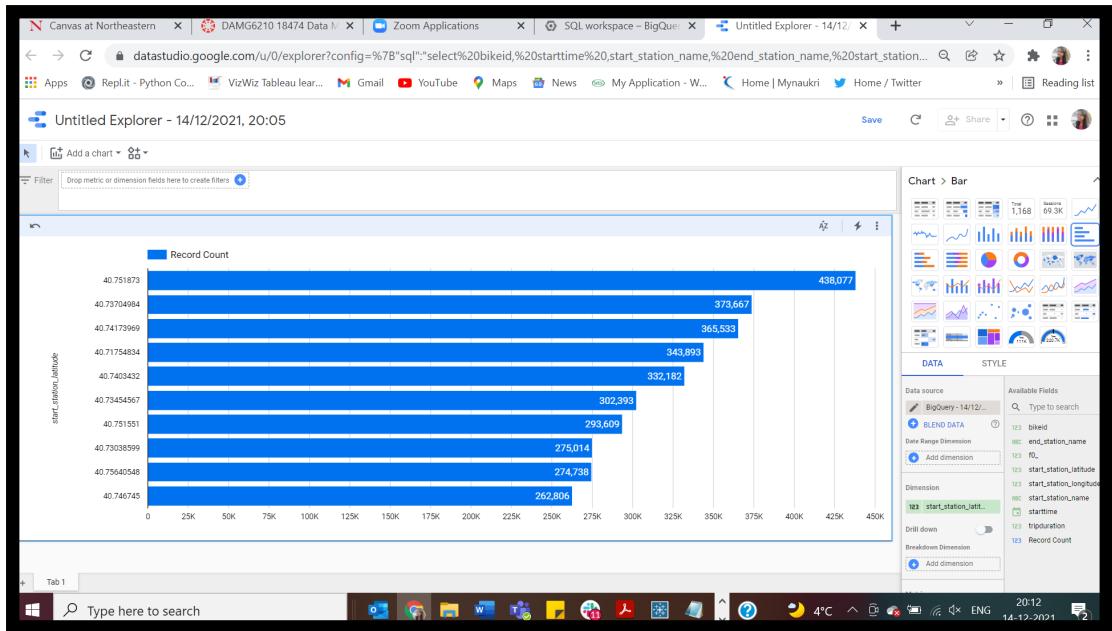
## 6. Route- End location



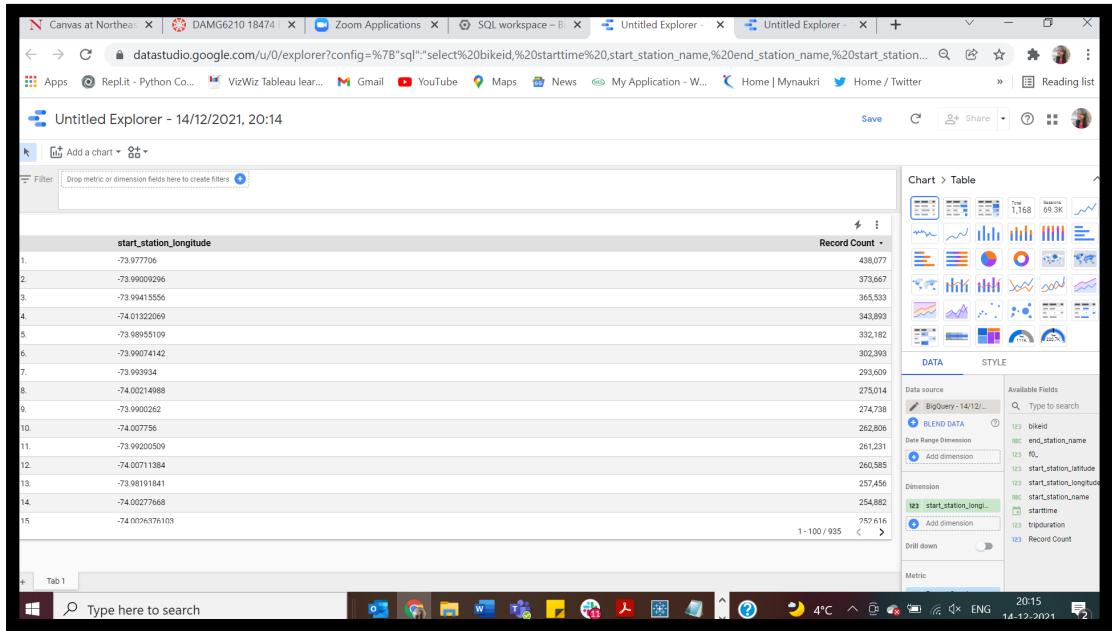
## 7. Top latitude

```
select bikeid, starttime ,start_station_name, end_station_name, start_statio
n_latitude, start_station_longitude, tripduration, count(bikeid)
from `healthy-genre-330203.damg_datset.citibike_trips`
```

group by 1, 2, 3, 4, 5, 6, 7



## 8. Top longitude



## 9. Duration

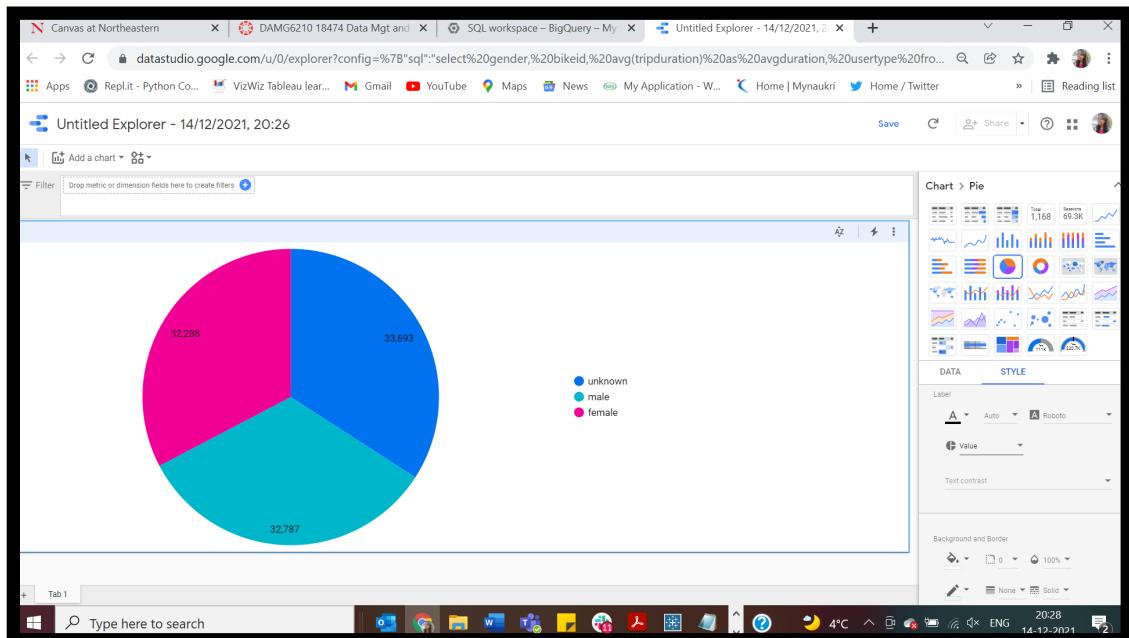
A screenshot of a Google Data Studio dashboard titled "Untitled Explorer - 14/12/2021, 20:19". The main area displays a table with two columns: "bikeid" and "sumduration". The table contains 14 rows of data. To the right of the table is a sidebar titled "Chart > Table" which includes a chart preview, data summary, and a list of available fields.

bikeid	sumduration
30503	19510049
30875	15962256
16777	15020934
16533	13931824
18504	13586276
30446	12479323
33252	11749576
19957	11699746
15336	11138807
24939	10283682
17655	10172645
27076	9735948
21523	9582723
19422	9452993
10841	839441

## Gender Analysis:

### 1. Male Vs Female

```
select gender, bikeid, avg(tripduration) as avgduration, usertype from `healthy-genre-330203.damg_dataset.citibike_trips`  
group by 1,2,4
```



## 2. Average trip male vs female

```
select gender, avg(tripduration) as avgduration from `healthy-
genre-330203.damg_datset.citibike_trips`
where gender in ('male','female')
group by 1
```

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the 'Explorer' sidebar lists datasets like 'healthy-genre-330203' and 'new\_york\_citibike'. In the main area, a query editor window is open with the following SQL code:

```
5 --select bikeid, starttime, start_station_name, end_station_name, start_station.latitude, start_station.longitude, tripduration, c
6 --from healthy-genre-330203.damg_datset.citibike_trips
7 --group by 1,2,3,4,5,6,7
8
9 --select gender, bikeid, avg(tripduration) as avgduration, usertype from healthy-genre-330203.damg_datset.citibike_trips
10 --group by 1,2,4
11
12
13 select gender, avg(tripduration) as avgduration from healthy-genre-330203.damg_datset.citibike_trips
14 where gender in ('male','female')
15 group by 1
```

The 'Query results' section shows the output of the query:

Row	gender	avgduration
1	male	810.65686693732
2	female	964.999797839278

## 3. Average trip duration by gender

```
select gender,usertype, avg(tripduration) as avgduration from `healthy-
genre-330203.damg_datset.citibike_trips`
where gender in ('male','female')
group by 1,2
```

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the 'Explorer' sidebar lists datasets and tables, including 'healthy-genre-330203.damg\_dataset.citibike\_trips'. In the main area, a query editor window displays the following SQL code:

```

5 --From `healthy-genre-330203.damg_dataset.citibike_trips`
6 --group by 1,2,3,4,5,6,7
7
8 --select gender, bikeid, avg(tripduration) as avgduration, usertype from `healthy-genre-330203.damg_dataset.citibike_trips`
9
10
11
12
13 select gender,usertype, avg(tripduration) as avgduration from `healthy-genre-330203.damg_dataset.citibike_trips`
14 where gender in ('male','female')
15 GROUP BY 1,2

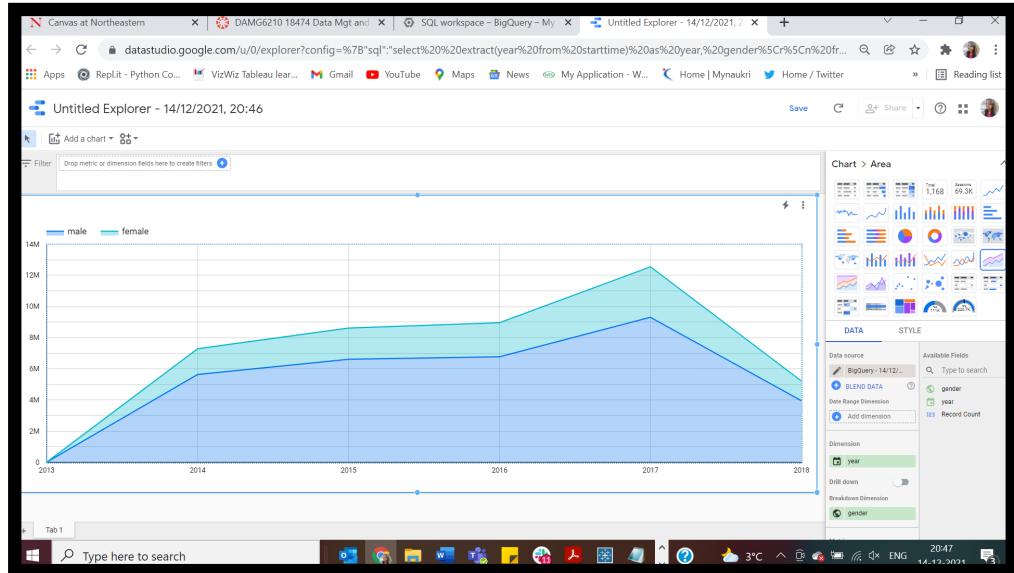
```

The 'Processing location: US' section indicates 'Query complete (0.9 sec elapsed, 1.2 GB processed)'. The 'Job information' tab shows 'Results' selected, and the 'Execution details' tab shows 'Row gender usertype avgduration'. The results table contains the following data:

Row	gender	usertype	avgduration
1	female	Subscriber	910.0489036529694
2	male	Customer	4676.803500405631
3	female	Customer	3938.303931060737
4	male	Subscriber	773.3736305773575

## 4. Year over year trend

```
select extract(year from starttime) as year, gender
from `healthy-genre-330203.damg_dataset.citibike_trips`
where gender in ('male','female')
```



## Age Analysis

### 1. SELECT

```

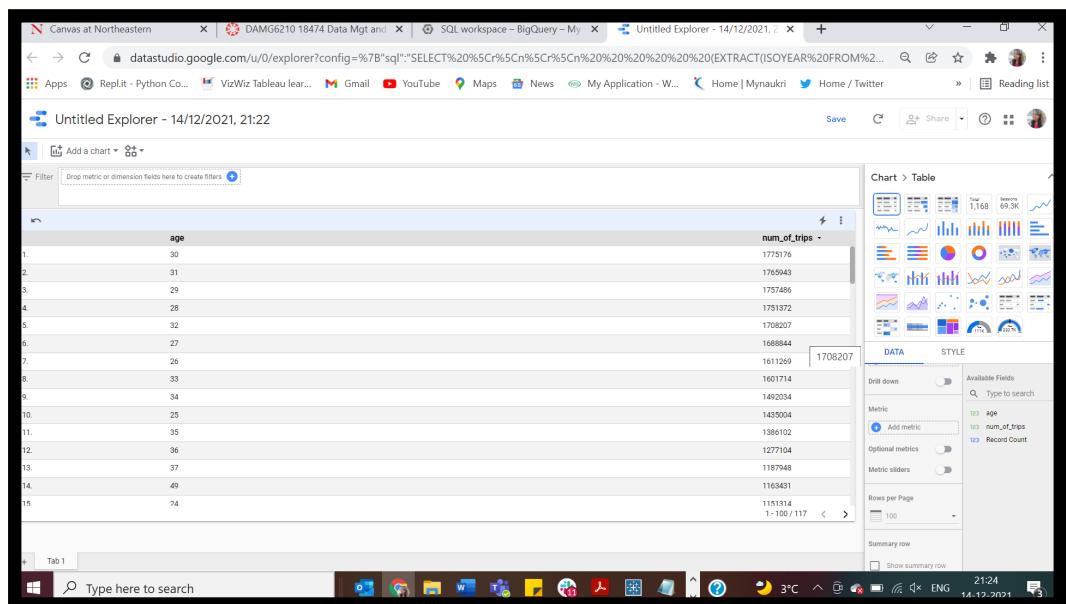
        (EXTRACT(ISOYEAR FROM starttime) -
(cast(birth_year as INT64))) as age,
        COUNT(bikeid) num_of_trips, FROM `healthy-
genre-330203.damg_datset.citibike_trips` 

WHERE CAST(birth_year as string) != "null"

GROUP BY age

ORDER BY age

```



## 2. SELECT

```

        (EXTRACT(ISOYEAR FROM starttime) -
(cast(birth_year as INT64))) as age,
        COUNT(bikeid) num_of_trips,avg(tripduration) as avgduration

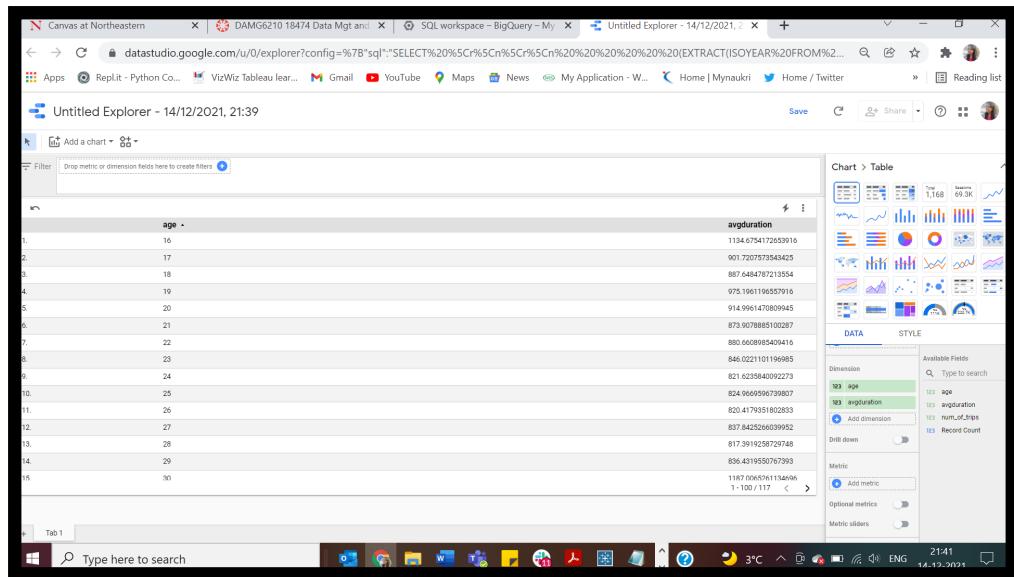
FROM `healthy-genre-330203.damg_datset.citibike_trips` 

WHERE CAST(birth_year as string) != "null"

GROUP BY age

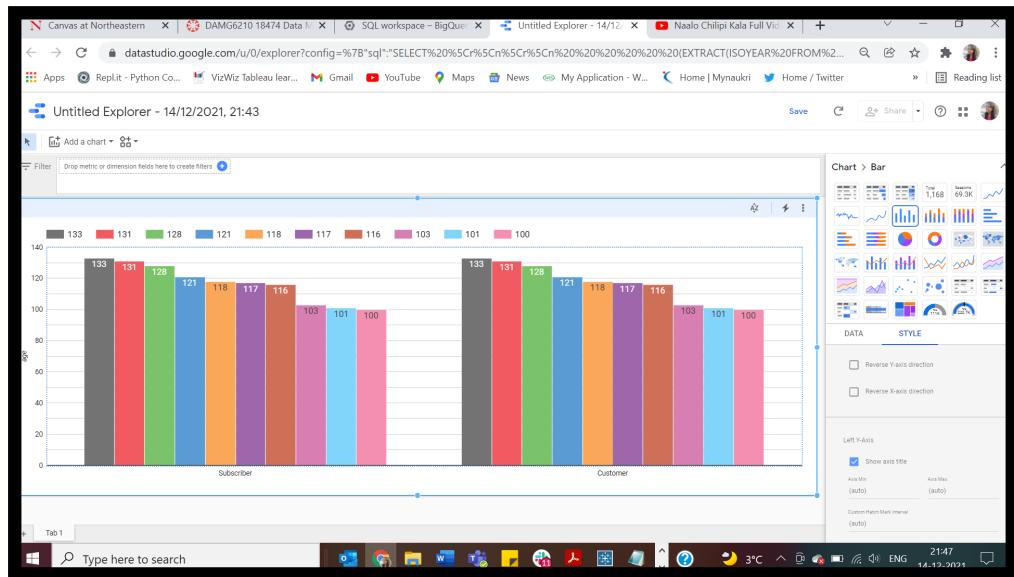
ORDER BY age

```



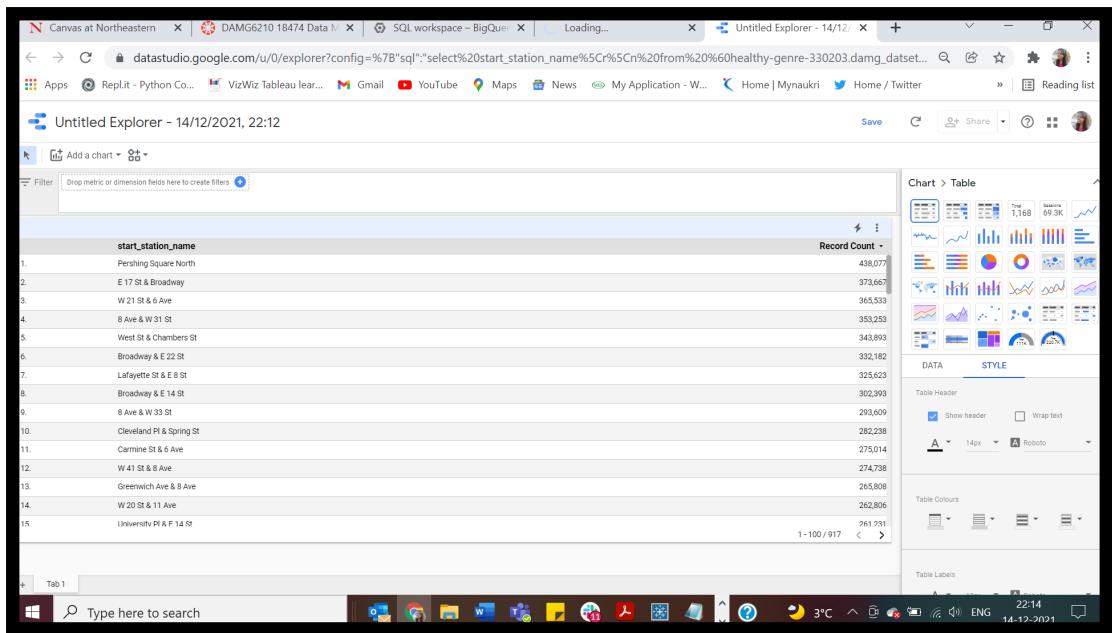
```

3. SELECT (EXTRACT(ISOYEAR FROM starttime) -
    (cast(birth_year as INT64))) as
    age, COUNT(bikeid) num_of_trips, usertype
    FROM `healthy-genre-330203.damg_datset.citibike_trips`
    WHERE CAST(birth_year as string) != "null"
    GROUP BY age, usertype ORDER BY age
  
```

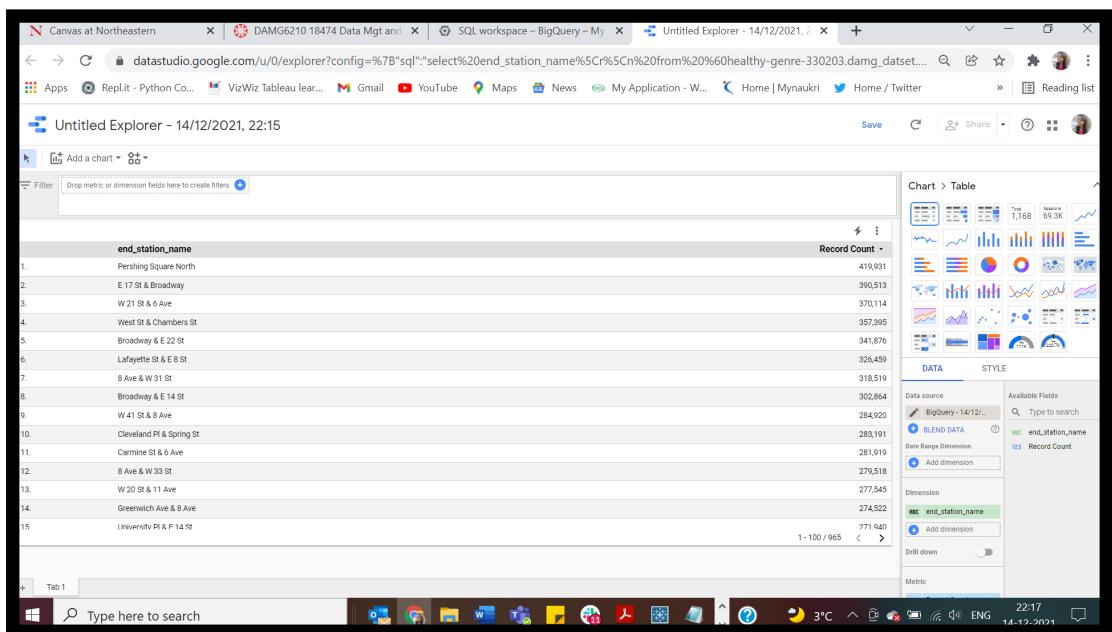


## Most Popular Stations:

1. select start\_station\_name
 from `healthy-genre-330203.damg\_datset.citibike\_trips`

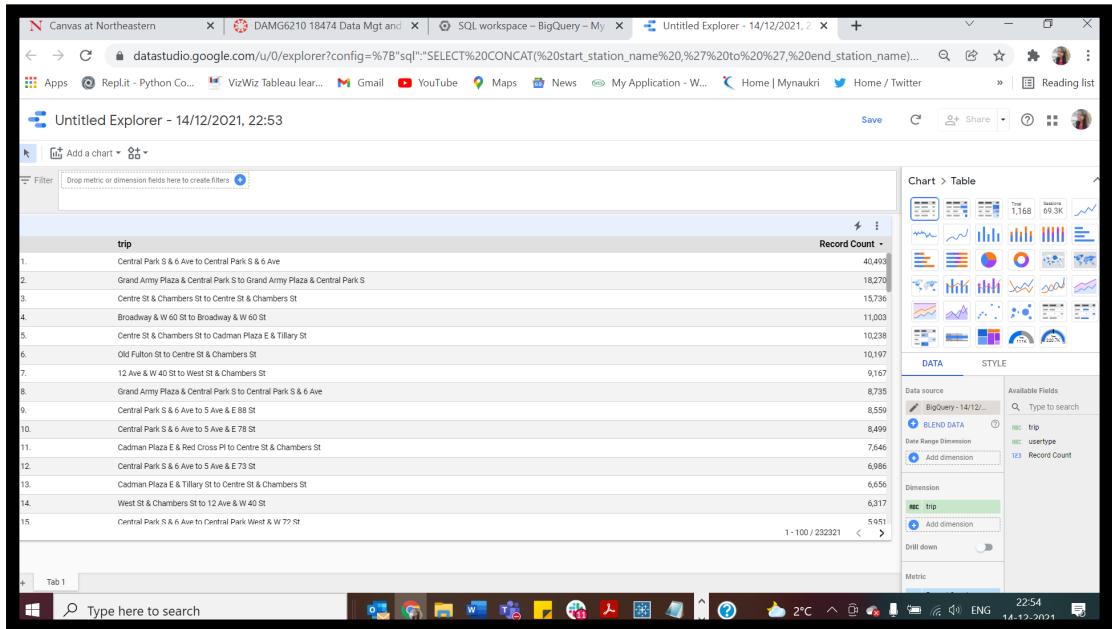
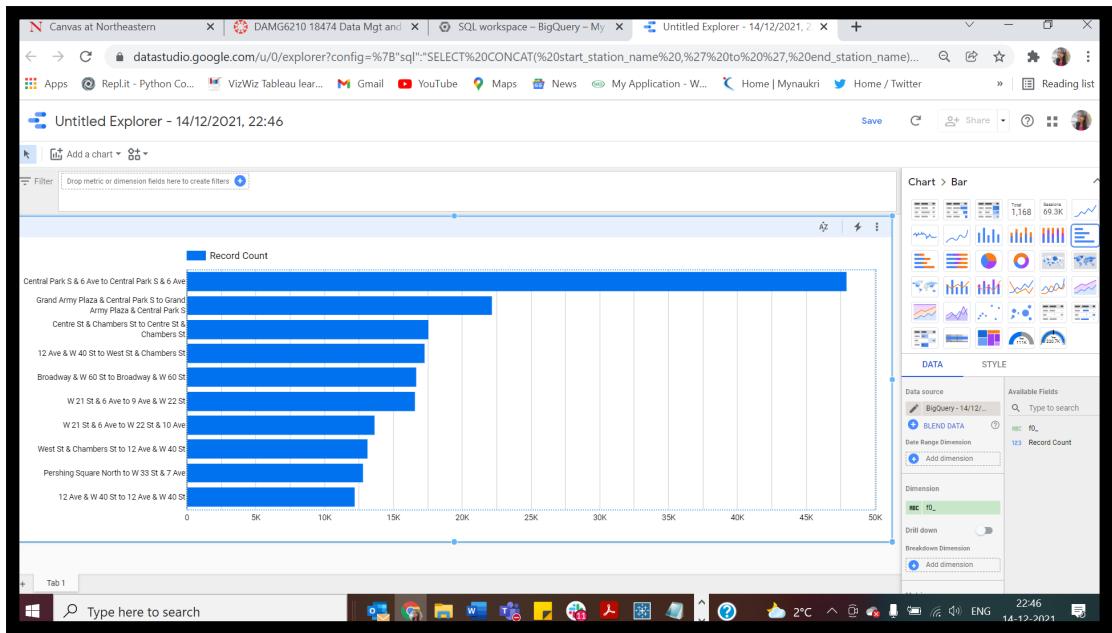


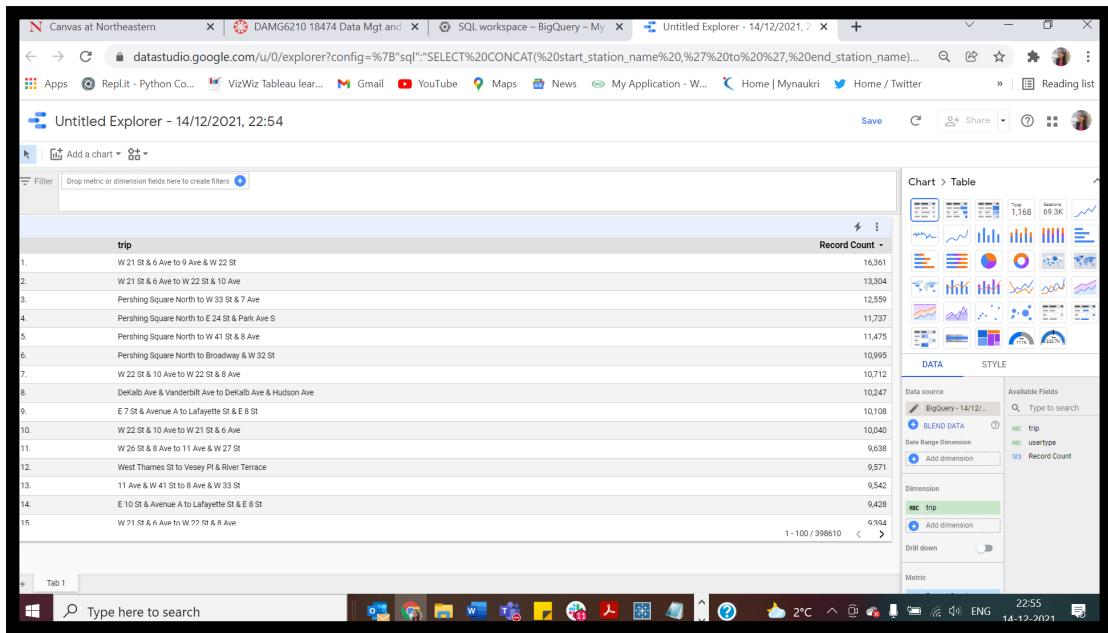
## 2. End station highest



## 3. Both start and end trip

```
SELECT CONCAT( start_station_name , ' to ' , end_station_name) from `healthy-genre-330203.damg_datset.citibike_trips` ;
```

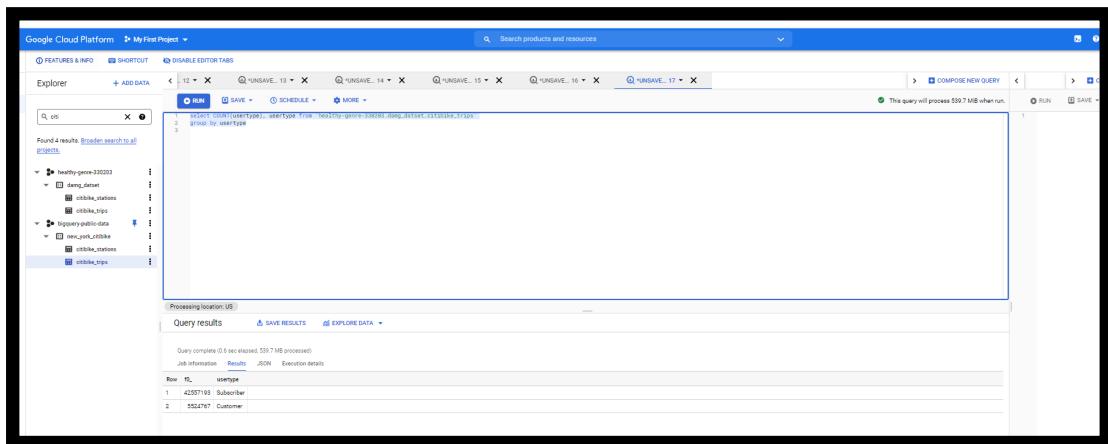


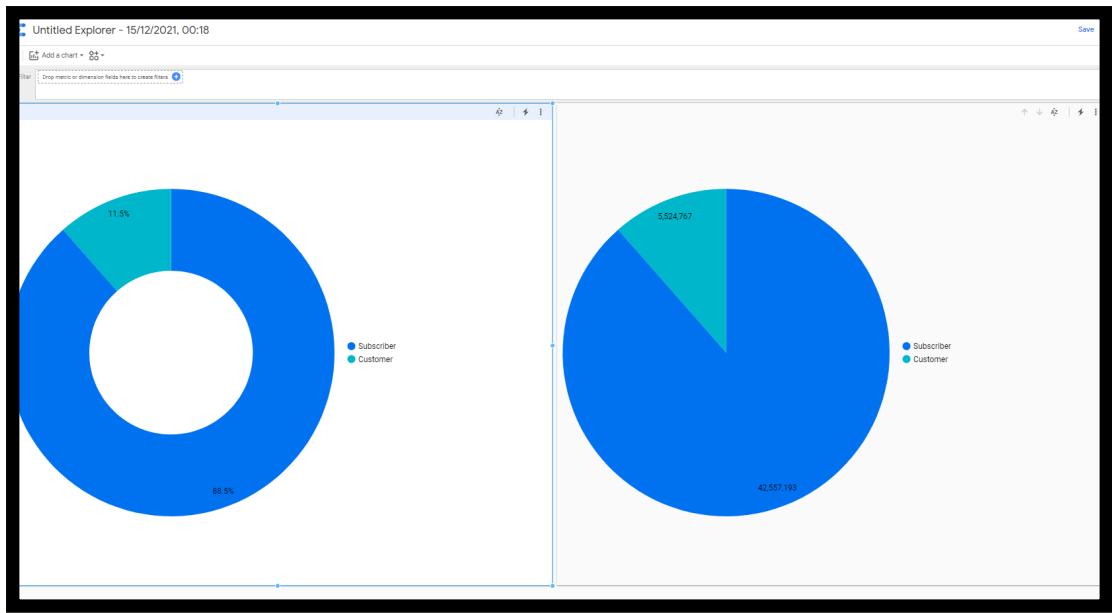


## Trip Analysis:

### 1. Subscriber Vs Customer

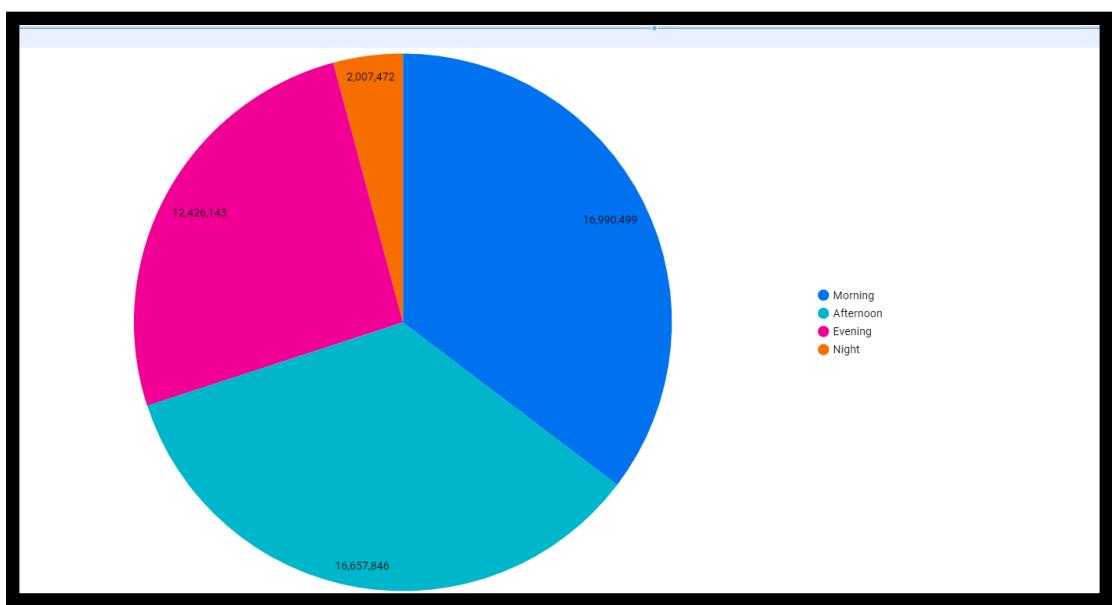
```
select COUNT(usertype), usertype from `healthy-genre-330203.damg_dataset.citibike_trips`  
group by usertype
```





## 2. Bike trips during what period of the day

```
select case when extract(hour from starttime) in (06,07,08, 09, 10, 11, 12) then 'Morning'
when extract(hour from starttime) in (13,14,15, 16, 17) then 'Afternoon'
when extract (hour from starttime) in (18, 19, 20, 21, 22) then 'Evening'
else 'Night' end as timeday from `healthy-genre-330203.damg_datset.citibike_trips`
```



## 3. Bike trips for what hour

```

select count(bikeid) as number_of_trips, extract(hour from starttime) as hour_of_day
from `healthy-genre-330203.damg_datset.citibike_trips`
group by bikeid, 2

```

A screenshot of a data viewer interface showing a table of data. The table has two columns: 'hour\_of\_day' and 'Record Count'. The data shows the number of trips for each hour of the day, from 07 to 17. The record count for each hour is listed to the right. The table is paginated at the bottom, showing pages 1-24 of 24.

hour_of_day	Record Count
1.	17,407
2.	17,404
3.	17,393
4.	17,387
5.	17,375
6.	17,375
7.	17,362
8.	17,362
9.	17,355
10.	17,354
11.	17,354
	1 - 24 / 24

#### 4. Round trips

A screenshot of the Google Cloud Platform BigQuery interface. The left sidebar shows a project structure with datasets like 'damg\_datset' and tables like 'citibike\_trips'. The main area shows a query editor with a single query run. The results show a single row with the value '1016995' for 'num\_of\_round\_trips'.

```

SELECT COUNT(bikeid) as num_of_round_trips,
FROM `healthy-genre-330203.damg_datset.citibike_trips`
WHERE start_station_name = end_station_name

```

Row	num_of_round_trips
1	1016995

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the project structure under 'My First Project'. The main area shows a query editor with the following SQL code:

```

1 SELECT COUNT(bikeid) AS num_of_non_round_trips
2 FROM `healthy-genre-330203.damg_datset.citibike_trips`
3 WHERE start_station_name != end_station_name
4
5
6
7

```

The results pane shows the query completed successfully with 47054955 rows.

## 5. Round trips customer vs subscriber

```

SELECT COUNT(bikeid) AS num_of_round_trips, usertype
FROM `healthy-genre-330203.damg_datset.citibike_trips`
WHERE start_station_name = end_station_name
GROUP BY usertype

```

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the project structure under 'My First Project'. The main area shows a query editor with the following SQL code:

```

1 SELECT COUNT(bikeid) AS num_of_round_trips, usertype
2 FROM `healthy-genre-330203.damg_datset.citibike_trips`
3 WHERE start_station_name = end_station_name
4 GROUP BY usertype
5
6
7
8
9

```

The results pane shows the query completed successfully with 376513 rows for Customer and 640482 rows for Subscriber.

## 6.

```

SELECT COUNT(bikeid) AS num_of_non_round_trips, usertype
FROM `healthy-genre-330203.damg_datset.citibike_trips`
WHERE start_station_name != end_station_name
GROUP BY usertype

```

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays datasets: `healthy-genre-330203` (selected), `damg\_dataset`, and `bqquery-public-data` (with `new\_york\_citibike` selected). The main area contains a query editor with the following SQL code:

```

SELECT count(bikeid) as num_of_non_round_trips, user type
FROM `healthy-genre-330203.damg_dataset.citibike_trips`
WHERE start_station_name != end_station_name
group by user type

```

The status bar at the bottom indicates "Processing location: US". Below the query editor, the results section shows the following table:

Row	num_of_non_round_trips	user type
1	5162534	Customer
2	41916711	Subscriber

## 7. Round trips average length of trip

```

SELECT avg(tripduration) as average_round
FROM `healthy-genre-330203.damg_dataset.citibike_trips`
WHERE start_station_name = end_station_name

```

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays datasets: `healthy-genre-330203` (selected), `damg\_dataset`, and `bqquery-public-data` (with `new\_york\_citibike` selected). The main area contains a query editor with the following SQL code:

```

SELECT avg(tripduration) as average_round
FROM `healthy-genre-330203.damg_dataset.citibike_trips`
WHERE start_station_name = end_station_name

```

The status bar at the bottom indicates "Processing location: US". Below the query editor, the results section shows the following table:

Row	average_round
1	1802.5595819055168

## 8.

```

SELECT avg(tripduration) as average_non_round
FROM `healthy-genre-330203.damg_dataset.citibike_trips`
WHERE start_station_name != end_station_name

```

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays datasets and tables under the project 'My First Project'. The main area contains a query editor with the following SQL code:

```

1 SELECT avg(tripduration) as average_non_round
2 FROM `healthy-genre-330203.damg_dataset.citibike_trips`
3 WHERE start_station_name != end_station_name
4
5
6
7
8
9

```

Below the query, the results pane shows the output of the query:

Query results

Row	average_non_round
1	952.9124597882937

## Analysis of length of trip

### 1. Average length of trip

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays datasets and tables under the project 'My First Project'. The main area contains a query editor with the following SQL code:

```

1 select avg(tripduration) as avgtrip from `healthy-genre-330203.damg_dataset.citibike_trips`

```

Below the query, the results pane shows the output of the query:

Query results

Row	avgtrip
1	970.2952747556897

### 2. Average length of trip by customer vs Subscriber

```

select avg(tripduration) as avgtrip, usertype from `healthy-
genre-330203.damg_dataset.citibike_trips`
group by 2

```

```

    select avg(tripduration) as avgtrip, usertype from `healthy-genre-330203.damg_dataset.citibike_trips`
    group by 2
  
```

Row	avgtrip	usertype
1	808.2168847694237	Subscriber
2	2218.791257622693	Customer

### 3. Trips less than 15 minutes

```

select Count(tripduration) as tripslessthan15 from `healthy-
genre-330203.damg_datset.citibike_trips`
where tripduration <900
  
```

```

    select Count(tripduration) as tripslessthan15 from `healthy-
genre-330203.damg_datset.citibike_trips`
    where tripduration <900
  
```

Row	tripslessthan15
1	32625320

### 4. Trip duration less than an hour

```

select Count(tripduration) as tripduration from `healthy-
genre-330203.damg_datset.citibike_trips`
where tripduration >900 and tripduration <3600
  
```

The screenshot shows the Google Cloud Platform BigQuery interface. The query window contains the following SQL code:

```

1 select count(tripduration) as tripduration from `healthy-genre-330203.damg_dataset.citibike_trips`
2 where tripduration <3600

```

The results pane shows one row of data:

Row	tripduration
1	14023239

## 5. Trip duration more than 1 hour

```

select Count(tripduration) as tripdurationmorethan1 from `healthy-
genre-330203.damg_dataset.citibike_trips`
where tripduration >3600

```

The screenshot shows the Google Cloud Platform BigQuery interface. The query window contains the following SQL code:

```

1 select count(tripduration) as tripdurationmorethan1 from `healthy-genre-330203.damg_dataset.citibike_trips`
2 where tripduration >3600

```

The results pane shows one row of data:

Row	tripdurationmorethan1
1	508960