Cyclistic Project Final Report

Case study: How does a bike-share navigate speedy success?
(All querys)

Data Cleaning and Manipulation

I initially attempted to use Google Sheets to handle the data. However, due to the large size of the files and the numerous rows, the program kept freezing when trying to merge them into a single sheet. Given these limitations, I ultimately decided to switch to SQL in BigQuery

The first step in reviewing the data for errors was to open BigQuery and execute the following query:

```
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL;
```

This resulted in 1,626,328 rows with null values in the columns: start_station_name, start_station_id, end_station_name, and end_station_id.

Preliminarily, we observed that the rows with null values belong to both casual and annual members, and that the rideable_type column contains electric_bike and electric_scooter. We then performed a query to determine what percentage of the total dataset contained null values to assess whether it would impact the analysis.

```
SELECT
  (SUM(CASE WHEN start_station_id IS NULL OR end_station_id IS NULL THEN 1 ELSE 0 END) * 100.0) / COUNT(*) AS porcentaje_nulos
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
```

Executing the query, we found that 27.77% of the data contained null values.

To determine which values appeared most frequently in this 27.77%, we conducted another query to identify which type of member had the most null values and what type of equipment they were using.

```
SELECT
  member_casual,
  rideable_type,
  COUNT(*) AS num_registros,
  ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`), 2) AS porcentaje_total
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL
GROUP BY member_casual, rideable_type
ORDER BY num_registros DESC;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN

Fila	member_casual	rideable_type	num_registros	porcentaje_total
1	member	electric_bike	960976	16.41
2	casual	electric_bike	560921	9.58
3	casual	electric_scooter	59475	1.02
4	member	electric_scooter	37035	0.63
5	casual	classic_bike	6194	0.11
6	member	classic_bike	1727	0.03

Considering that the top three results constitute a significant percentage of electric bikes (from both member types) and electric scooters used by casual members, we decided to retain this data for analysis, as removing it could introduce biases.

We then performed another query on the month and station columns to check if null values were more frequent in certain months or seasons.

```
SELECT
  EXTRACT(MONTH FROM started_at) AS mes,
  COUNT(*) AS num_nulos,
   ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`), 2) AS porcentaje_nulos
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL
GROUP BY mes
ORDER BY mes;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO

Fila	mes	num_nulos	porcentaje_nulos
1	1	31065	0.53
2	2	38428	0.66
3	3	71409	1.22
4	4	117227	2.0
5	5	167415	2.86
6	6	216405	3.7
7	7	208063	3.55
8	8	214481	3.66
9	9	283853	4.85
10	10	133332	2.28
11	11	87720	1.5
12	12	56930	0.97

The results indicated an increase in the percentage of null values from June to September, with the highest being 4.85%. These months coincide with the summer period, which could indicate a seasonal change in data recording or an increase in usage by tourists and casual riders.

Next, we conducted a query to determine whether null values were more common in short or long trips.

```
SELECT
  member_casual,
  rideable_type,
      AVG(TIMESTAMP_DIFF(ended_at, started_at, MINUTE)) AS duracion_promedio_minutos,
   ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`), 2) AS porcentaje_nulos
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL
GROUP BY member_casual, rideable_type;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | D

Fila	member_casual	rideable_type	duracion_promedio_	porcentaje_nulos
1	casual	electric_bike	13.14848258489...	9.58
2	member	electric_bike	11.20367105942...	16.41
3	member	electric_scooter	7.958849736735...	0.63
4	casual	electric_scooter	11.72102564102...	1.02
5	casual	classic_bike	1478.239263803...	0.11
6	member	classic_bike	1337.153445280...	0.03

We found a higher percentage of null values for both member types using electric bikes. However, an unusual result appeared for classic bike users, with trip durations of 1,337 and 1,478 minutes. We investigated further for potential registration errors or missing values.

To address this, we ran a query to identify trips longer than three hours.

```
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE TIMESTAMP_DIFF(ended_at, started_at, MINUTE) > 180
```

Resultados de la consulta | GUARDAR LOS RESULTADOS | EXPL

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN | GRÁFICO DE EJECUCIÓN

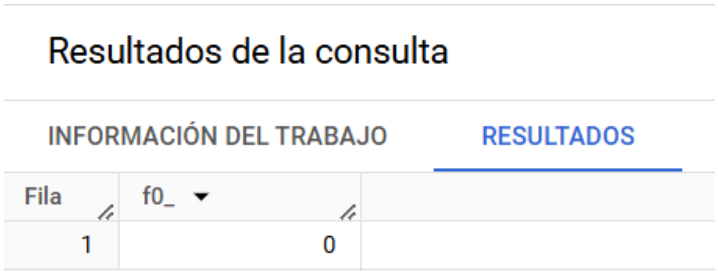
Fila	ride_id	rideable_type	started_at	ended_at	start_station_name
1	334541814CF42BCA	electric_bike	2024-06-05 17:57:01.383000 U...	2024-06-05 21:48:55.593000 U...	null
2	4E1DFCDAC968A44C	electric_bike	2024-04-11 16:28:38 UTC	2024-04-11 21:12:05 UTC	null
3	1DFC4CE21FCB2A56	electric_bike	2024-04-08 18:55:10 UTC	2024-04-08 22:07:39 UTC	null
4	20D40D6D12E75F75	electric_bike	2024-06-19 10:33:29.422000 U...	2024-06-19 15:43:34.997000 U...	null
5	608C29C465EADA44	electric_bike	2024-07-09 12:20:11.939000 U...	2024-07-09 15:39:20.679000 U...	null
6	AA420585AA920AF6	electric_bike	2024-03-31 18:35:30 UTC	2024-04-01 02:35:26 UTC	null
7	506C9D4F2BB39497	electric_bike	2023-11-08 16:49:18 UTC	2023-11-08 20:55:31 UTC	null
8	148D5D5BFE940CCE	electric_scooter	2024-08-31 21:07:36.220000 U...	2024-09-01 00:29:21.965000 U...	null
9	DCB52A3D3C0B449A	electric_scooter	2024-09-20 17:23:52.717000 U...	2024-09-20 21:07:24.234000 U...	null

Resultados por página: 50 | 1 – 50 de 21604

A total of 21,604 rows were found, which, considering the total dataset (5,854,544 rows), represents only 0.37%. We decided to remove these records, as such prolonged trip durations may be due to errors such as users forgetting to end their trips or technical issues. Removing these anomalies ensures that our dataset reflects typical vehicle usage.

```
DELETE FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE TIMESTAMP_DIFF(ended_at, started_at, MINUTE) > 180;
```

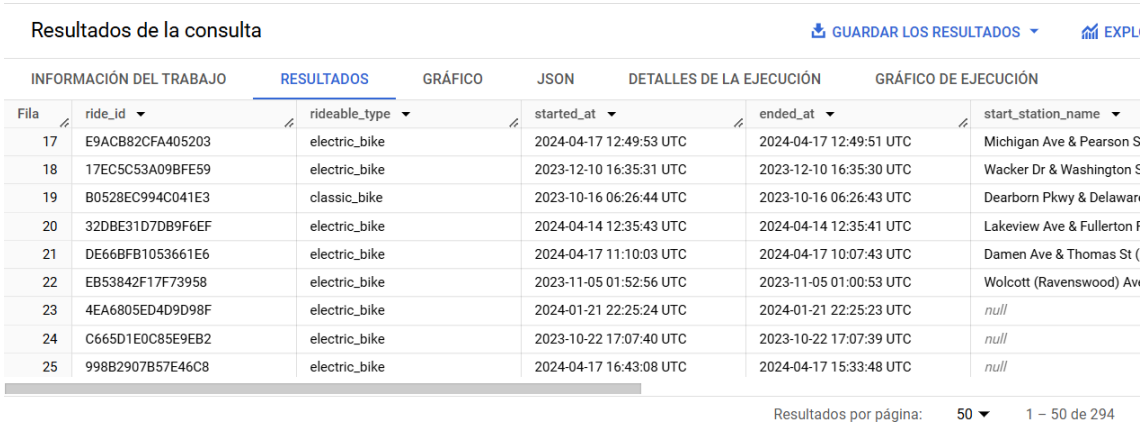
```
SELECT COUNT(*)
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE TIMESTAMP_DIFF(ended_at, started_at, MINUTE) > 180;
```



We executed the first query to delete trips longer than three hours, followed by a second query to confirm the deletion.

Additionally, we ran a query to identify trips where the end time was earlier than the start time.

```
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE ended_at < started_at;
```



This resulted in 294 records. Since these were clearly errors, we removed them, as they represented a minimal percentage of the total dataset.

```
DELETE FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE ended_at < started_at;
```

```
SELECT COUNT (*)
```

```
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE ended_at < started_at;
```

Correctly removed.

Resultados de la consulta

INFORMACIÓN DEL TRABAJO

Fila	f0_
1	0

Next, we examined the start_lat and start_lng columns to identify locations with the most null values and determine the average location with the highest missing data.

```
SELECT
  ROUND(start_lat, 2) AS lat_area,
  ROUND(start_lng, 2) AS lng_area,
  COUNT(*) AS num_nulos
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL
GROUP BY lat_area, lng_area
ORDER BY num_nulos DESC
LIMIT 10;
```

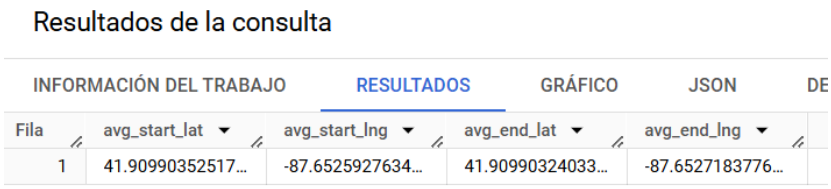
Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO

Fila	lat_area	lng_area	num_nulos
1	41.89	-87.63	53357
2	41.88	-87.64	46022
3	41.9	-87.63	44760
4	41.94	-87.65	42085
5	41.88	-87.63	41797
6	41.91	-87.63	41599
7	41.95	-87.65	40620
8	41.89	-87.62	39326
9	41.93	-87.64	35582
10	41.89	-87.64	34933

```
SELECT
  AVG(start_lat) AS avg_start_lat,
  AVG(start_lng) AS avg_start_lng,
  AVG(end_lat) AS avg_end_lat,
  AVG(end_lng) AS avg_end_lng
```

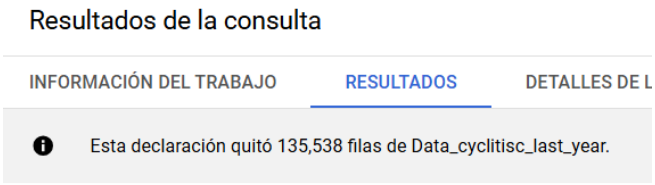
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE start_station_id IS NULL OR end_station_id IS NULL;



After checking the coordinates, we confirmed that they correspond to Chicago, Illinois.

The difference between one coordinate and another is a decimal, suggesting that null values may be due to technical errors or specific station registration issues. We decided to keep this data while considering this discrepancy in the analysis.

DELETE FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`
WHERE TIMESTAMP_DIFF(ended_at, started_at, SECOND) < 60;



To finalize our data cleaning process, we removed trips shorter than one minute, as these were likely user errors and provided little analytical value.

We then standardized the format of the start_lat, start_lng, end_lat, and end_lng columns, limiting them to six decimal places to improve readability.

SELECT
ride_id,
rideable_type,
started_at,
ended_at,
start_station_name,
start_station_id,
end_station_name,
end_station_id,
ROUND(start_lat, 6) AS start_lat,
ROUND(start_lng, 6) AS start_lng,
ROUND(end_lat, 6) AS end_lat,
ROUND(end_lng, 6) AS end_lng,
member_casual
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_last_year`;

In the following query, we will identify anomalous trips that started or ended outside common coordinates. With this format change, we have renamed the table to *decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned.*

```
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
WHERE start_lat NOT BETWEEN 41.5 AND 42.5
   OR start_lng NOT BETWEEN -88 AND -87
   OR end_lat NOT BETWEEN 41.5 AND 42.5
   OR end_lng NOT BETWEEN -88 AND -87;
```

Resultados de la consulta

GUARDAR LOS RESULTADOS | EXPLORA

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN | GRÁFICO DE EJECUCIÓN

Fila	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_cas
43	null	null	41.88	-87.64	42.53	-87.8	member
44	null	null	41.857926	-87.624336	42.14	-88.94	casual
45	null	null	41.880958	-87.616743	35.92	-82.94	casual
46	null	null	41.861267	-87.656625	40.85	-89.39	member
47	null	null	41.9	-87.63	21.79	-92.62	member
48	null	null	41.92	-87.7	42.51	-87.5	casual
49	null	null	41.81153	-87.70885	41.23	-88.28	member

Resultados por página: 50 | 1 – 49 de 49

This result gives us 49 rows of information, which represents a minimal percentage of the total table. We will proceed to delete them, but first, we will run another query to group the member type with the highest number of incidents.

```
SELECT
  rideable_type,
  member_casual,
  COUNT(*) AS num_registros
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
WHERE start_lat NOT BETWEEN 41.5 AND 42.5
   OR start_lng NOT BETWEEN -88 AND -87
   OR end_lat NOT BETWEEN 41.5 AND 42.5
   OR end_lng NOT BETWEEN -88 AND -87
GROUP BY rideable_type, member_casual
ORDER BY num_registros DESC;
```

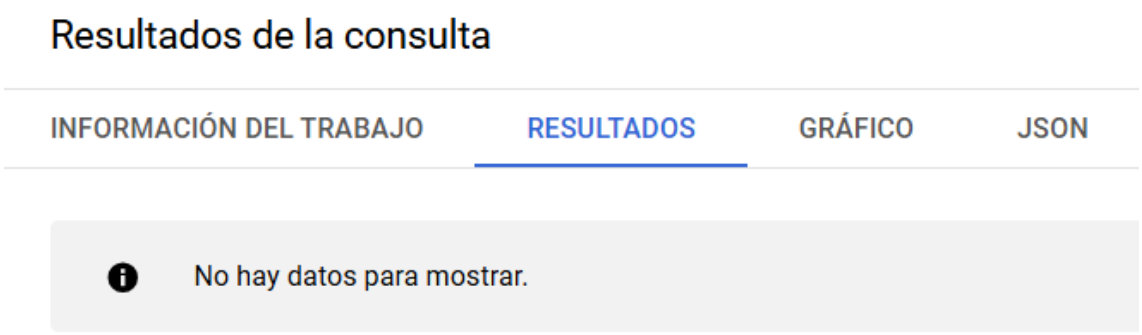
Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES I

Fila	rideable_type	member_casual	num_registros
1	electric_bike	member	30
2	electric_bike	casual	19

In the result, we can see that the type of vehicle used in all cases was the electric bike, and that 30 of the incidents were recorded by annual members, while the other 19 were casual members. Now, we proceed with the query to delete those 49 anomalous records.

```
DELETE FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
WHERE start_lat NOT BETWEEN 41.5 AND 42.5
  OR start_lng NOT BETWEEN -88 AND -87
  OR end_lat NOT BETWEEN 41.5 AND 42.5
  OR end_lng NOT BETWEEN -88 AND -87;
```



I ran the query again that grouped by member type and vehicle type, and we see that there are no data to display, confirming that the 49 rows were successfully deleted.

Following this, we identified and eliminated duplicate records by running queries on rider_id.

```
SELECT ride_id, COUNT(*) AS occurrences
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
GROUP BY ride_id
HAVING occurrences > 1;
```

Fila	ride_id	occurrences
142	171D1ADC1F6AB687	2
143	BE96DFD7724F59C5	2
144	43CD52984AD22D99	2
145	D115C403314536C4	2
146	1D8856396862BE62	2
147	57D54C76FF580E3C	2
148	8A58BE5ACD18DA0B	2
149	5322E55C9310A9D2	2

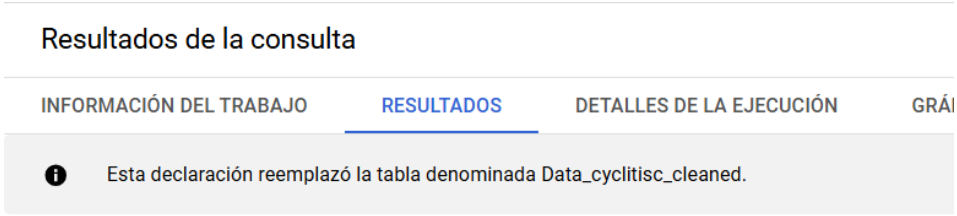
We found 149 duplicates, mostly occurring twice.

```
SELECT
  ride_id,
  COUNT(*) AS num_ocurrencias,
  ARRAY_AGG(STRUCT(started_at, ended_at, start_station_id, end_station_id,
rideable_type, member_casual)) AS detalles
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
GROUP BY ride_id
```

HAVING num_ocurrencias > 1;

A verification query confirmed that these duplicate rows contained the same information, except for variations in the started_at and ended_at columns due to double registration. We proceeded to delete these duplicates.

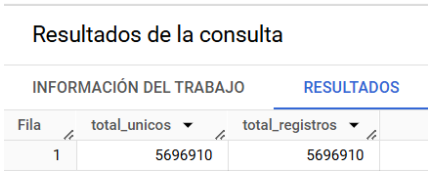
```
CREATE OR REPLACE TABLE `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned` AS
SELECT *
FROM (
  SELECT
    *,
    ROW_NUMBER() OVER (
      PARTITION BY ride_id  -- Group records with the same ride_id
      ORDER BY started_at ASC  -- Prioritize the record with the earliest start date
    ) AS row_num
  FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
)
WHERE row_num = 1;  -- Keep only one row per ride_id
```



Data modified.

We will run one last query to verify the total number of records remaining in the table.

```
SELECT COUNT(DISTINCT ride_id) AS total_unicos, COUNT(*) AS total_registros FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```



This data is correct since, prior to the data deletions, we had 5,697,108 records.

Finally, we added two new columns:

- ride_length: Calculated as the difference between started_at and ended_at, formatted as HH:MM:SS.

- day_of_week: Derived from started_at, indicating the day of the week for each trip.

```
SELECT
  *,
  CONCAT(
    LPAD(CAST(FLOOR(TIMESTAMP_DIFF(ended_at, started_at, SECOND) / 3600) AS STRING), 2, '0'), ':',
    LPAD(CAST(FLOOR((TIMESTAMP_DIFF(ended_at, started_at, SECOND) - (FLOOR(TIMESTAMP_DIFF(ended_at, started_at, SECOND) / 3600) * 3600)) / 60) AS STRING), 2, '0'), ':',
        LPAD(CAST((TIMESTAMP_DIFF(ended_at, started_at, SECOND) - (FLOOR(TIMESTAMP_DIFF(ended_at, started_at, SECOND) / 3600) * 3600) - (FLOOR((TIMESTAMP_DIFF(ended_at, started_at, SECOND) - (FLOOR(TIMESTAMP_DIFF(ended_at, started_at, SECOND) / 3600) * 3600)) / 60) * 60)) AS STRING), 2, '0')
  ) AS ride_length
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```

Resultados de la consulta

GUARDAR LOS RESULTADOS | EXPLORAR DATOS

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN | GRÁFICO DE EJECUCIÓN

Fila	started_at	ended_at	start_statio	start_stat	end_statio	end_sta	start_lat	start_lng	end_lat	end_lng	member_ca	row_	ride_length
1	2024-01-29 1...	2024-01-29 13...	null	null	null	null	42.0	-87.66	42.0	-87.66	member	1	00:03:14
2	2024-08-08 1...	2024-08-08 15...	null	null	null	null	42.0	-87.66	42.0	-87.66	member	1	00:10:50
3	2024-09-13 1...	2024-09-13 15...	null	null	null	null	41.98	-87.67	42.0	-87.67	member	1	00:09:17
4	2023-10-12 1...	2023-10-12 17...	null	null	null	null	42.01	-87.67	42.0	-87.66	member	1	00:04:59
5	er	2024-09-21 1...	2024-09-21 11...	null	null	null	null	42.0	-87.68	42.0	-87.68	casual	1	00:03:42
6		2024-07-15 1...	2024-07-15 17...	null	null	null	null	42.03	-87.71	42.0	-87.67	casual	1	00:15:41
7	er	2024-09-26 2...	2024-09-27 00...	null	null	null	null	42.01	-87.7	42.0	-87.69	casual	1	00:11:11
8		2024-06-25 1...	2024-06-25 12...	null	null	null	null	42.0	-87.67	42.0	-87.68	casual	1	00:03:24

```
SELECT
  *,
  FORMAT_TIMESTAMP('%A', started_at) AS day_of_week
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```

Resultados de la consulta

GUARDAR LOS RESULTADOS | EXPLORAR DATOS

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN | GRÁFICO DE EJECUCIÓN

Fila	start_lng	end_lat	end_lng	member_casual	row_num	ride_length	day_of_week
1	-87.65	41.936253	-87.652662	casual	1	00:16:29	Sunday
2	-87.65	41.936253	-87.652662	member	1	00:02:34	Wednesday
3	-87.69	41.883043	-87.649931	member	1	00:22:30	Wednesday
4	-87.64	41.885837	-87.6355	member	1	00:01:05	Wednesday
5	-87.68	41.91461	-87.667968	member	1	00:09:51	Wednesday

We then sorted the dataset by ascending trip dates and saved it as a new table in BigQuery.

```
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
ORDER BY started_at ASC
LIMIT 5;
```

Resultados de la consulta

GUARDAR LOS RESU

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA EJECUCIÓN | GRÁFICO DE E

Fila	ride_id	rideable_type	started_at	ended_at
1	B541441FAF64B31C	classic_bike	2023-10-01 00:00:05 UTC	2023-10-01 00:32:19 UTC
2	163D8093FDDEBBC8	classic_bike	2023-10-01 00:00:06 UTC	2023-10-01 00:17:53 UTC
3	46803FC8C419FBB0	classic_bike	2023-10-01 00:00:12 UTC	2023-10-01 00:02:02 UTC
4	0ED9A33B7B80912D	classic_bike	2023-10-01 00:00:20 UTC	2023-10-01 00:05:32 UTC
5	0FBFC8BCF712A9B1	classic_bike	2023-10-01 00:00:24 UTC	2023-10-01 00:39:35 UTC

We checked the last records.

```
SELECT started_at, ended_at, ride_length
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
ORDER BY started_at DESC
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETALLES DE LA E

Fila	started_at	ended_at	ride_length
1	2024-09-30 23:54:05.552000 U...	2024-09-30 23:55:47.884000 U...	00:01:42
2	2024-09-30 23:53:46.550000 U...	2024-09-30 23:57:13.730000 U...	00:03:27
3	2024-09-30 23:53:31.833000 U...	2024-09-30 23:59:45.854000 U...	00:06:14
4	2024-09-30 23:52:58.172000 U...	2024-09-30 23:56:49.705000 U...	00:03:51
5	2024-09-30 23:52:36.941000 U...	2024-09-30 23:59:52.562000 U...	00:07:15

And we replace de table with the new one.

```
CREATE OR REPLACE TABLE `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned` AS
SELECT *
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
ORDER BY started_at ASC;
```

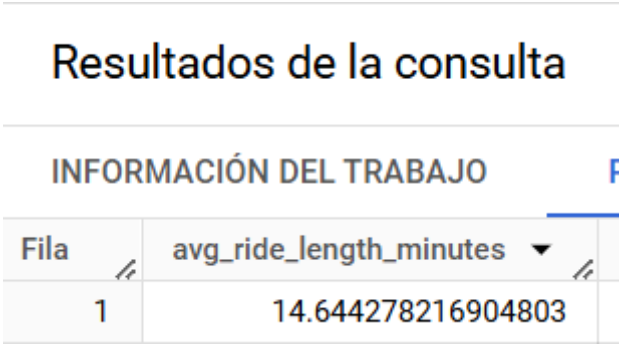
Fila	ride_id	rideable_type	started_at	ended_at
1	B541441FAF64B31C	classic_bike	2023-10-01 00:00:05 UTC	2023-10-01 00:32:19 UTC
2	163D8093FDDEBBC8	classic_bike	2023-10-01 00:00:06 UTC	2023-10-01 00:17:53 UTC
3	46803FC8C419FBB0	classic_bike	2023-10-01 00:00:12 UTC	2023-10-01 00:02:02 UTC
4	0ED9A33B7B80912D	classic_bike	2023-10-01 00:00:20 UTC	2023-10-01 00:05:32 UTC
5	0FBFC8BCF712A9B1	classic_bike	2023-10-01 00:00:24 UTC	2023-10-01 00:39:35 UTC
6	5F08B2CEF05DAF9F	classic_bike	2023-10-01 00:00:25 UTC	2023-10-01 00:04:56 UTC
7	D9A4DDCEAC712087	classic_bike	2023-10-01 00:00:29 UTC	2023-10-01 00:07:33 UTC
8	EF6EB5AA908EAAC6	electric_bike	2023-10-01 00:00:34 UTC	2023-10-01 00:07:52 UTC
9	43829211A58A614C	electric_bike	2023-10-01 00:00:34 UTC	2023-10-01 00:21:06 UTC
10	7E36C633B8D1B8D6	electric_bike	2023-10-01 00:00:38 UTC	2023-10-01 00:09:15 UTC

Analyze Phase

As recommended in the exercise, we will perform some calculations to gain a better understanding of the data we are working with. We will start by calculating the average ride_length using the following query:

```
SELECT
    AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60 AS avg_ride_length_minutes
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```

This gives us an average of 14.64 minutes.


Resultados de la consulta

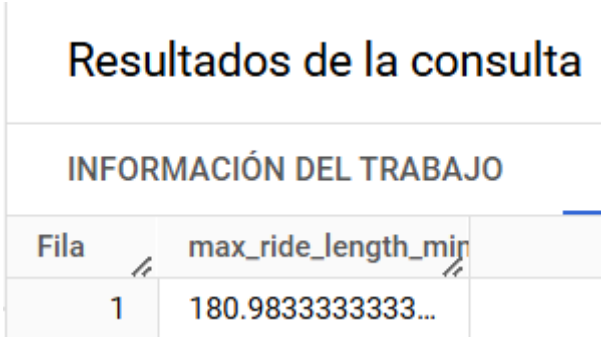
INFORMACIÓN DEL TRABAJO

Fila	avg_ride_length_minutes
1	14.644278216904803

Next, we will calculate the maximum ride_length.

```
SELECT
    MAX(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60 AS max_ride_length_minutes
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```

Executing this query results in a maximum ride duration of 180.98 minutes.


Resultados de la consulta

INFORMACIÓN DEL TRABAJO

Fila	max_ride_length_min
1	180.9833333333...

Now, we will determine the minimum value in the ride_length column.

```
SELECT
    MIN(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60 AS min_ride_length_minutes
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO

Fila	min_ride_length_min
1	1.0

The query results indicate that the shortest trip duration was 1 minute. Previously, during the data cleaning phase, all trips shorter than 1 minute were removed, as they were likely caused by recording errors and did not provide relevant insights for the analysis.

We will now calculate the mode of day_of_week.

```
SELECT
  day_of_week,
  COUNT(*) AS occurrences
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
GROUP BY day_of_week
ORDER BY occurrences DESC
LIMIT 7;
```

The initial results show that Saturdays have the highest number of recorded trips, followed by the other days in descending order.

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁF

Fila	day_of_week	occurrences
1	Saturday	874808
2	Wednesday	861873
3	Thursday	830610
4	Friday	803776
5	Tuesday	786032
6	Monday	775379
7	Sunday	764432

Next, we will calculate the average trip duration for both types of members.

```
SELECT
  member_casual,
    AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60 AS
avg_ride_length_minutes
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
GROUP BY member_casual;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GF

Fila	member_casual	avg_ride_length_min
1	casual	19.50650051227...
2	member	11.90840833917...

We observe that casual members have an average trip duration of 19.50 minutes, whereas annual members have an average of 11.90 minutes.

Now, we will execute a query to determine the average number of trips per member type per day of the week.

```
SELECT
  day_of_week,
  member_casual,
    AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60 AS avg_ride_length_minutes
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
GROUP BY day_of_week, member_casual
ORDER BY day_of_week, member_casual;
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁFICO | JSON | DETA

Fila	day_of_week	member_casual	avg_ride_length_min
1	Friday	casual	18.74380465507...
2	Friday	member	11.60151581818...
3	Monday	casual	18.99110079606...
4	Monday	member	11.38020190382...
5	Saturday	casual	22.18803278012...
6	Saturday	member	13.22810439207...
7	Sunday	casual	22.58872783381...
8	Sunday	member	13.23406540692...

We can see that casual users have a higher average usage than annual members on all days of the week.

Finally, we will analyze the total number of trips per user type by day of the week.

```
SELECT
  day_of_week,
  member_casual,
  COUNT(ride_id) AS total_trips
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
```

GROUP BY day_of_week, member_casual
ORDER BY day_of_week, member_casual;

Resultados de la consulta

INFORMACIÓN DEL TRABAJO | RESULTADOS | GRÁ

Fila	day_of_week	member_casual	total_trips
1	Friday	casual	294689
2	Friday	member	509087
3	Monday	casual	245919
4	Monday	member	529460
5	Saturday	casual	412506
6	Saturday	member	462302
7	Sunday	casual	352019
8	Sunday	member	412413

We observe that annual members have more recorded trips per weekday than casual members.

These are the general data insights for the entire year. Next, we will conduct the same analysis on a monthly basis, paying close attention to seasonal trends, holidays, and patterns.

We will execute queries to separate the dataset month by month, starting from October 2023 to September 2024.

CREATE TABLE `decoded-tesla-418214.Data_cyclistic_cleaned.2024_09` AS
*SELECT **
FROM `decoded-tesla-418214.Data_cyclitisc.Data_cyclitisc_cleaned`
WHERE DATE(started_at) BETWEEN '2024-09-01' AND '2024-09-30'
ORDER BY started_at ASC;

We used these queries to create a new folder within "decoded-tesla-418214" called "Data_cyclistic_cleaned," which corrects a typo in the word "Cyclistic" and moves our database into a new folder containing all our cleaned data. This includes the general dataset from October 2023 to September 2024 as well as the monthly datasets. We used the previous query while modifying the month and year parameters accordingly and adjusting the table name.

We will now execute a query that consolidates all previous commands, allowing us to view the required results in a single output. Additionally, we will include a command to add the rideable_type column, which will provide insights into users' vehicle preferences.

SELECT
day_of_week,
member_casual,
rideable_type,
ROUND(AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS avg_ride_length_minutes,

```
    ROUND(MAX(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2)
AS max_ride_length_minutes,
   ROUND(MIN(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS
min_ride_length_minutes,
   COUNT(ride_id) AS total_trips
FROM
   `decoded-tesla-418214.Data_cyclistic_cleaned.2023_10`
GROUP BY
   day_of_week, member_casual, rideable_type
ORDER BY
   day_of_week ASC, member_casual ASC, rideable_type ASC;
```

Fila	day_of_week	member_casual	rideable_type	avg_ride_length_mi	max_ride_length_mi	min_ride_length_mi	total_trips
1	Friday	casual	classic_bike	20.56	179.65	1.0	8971
2	Friday	casual	electric_bike	11.98	135.73	1.0	10972
3	Friday	member	classic_bike	11.21	169.42	1.0	20978
4	Friday	member	electric_bike	10.33	178.82	1.0	21360
5	Monday	casual	classic_bike	22.09	180.9	1.0	10815
6	Monday	casual	electric_bike	12.67	172.37	1.0	12501
7	Monday	member	classic_bike	11.19	172.22	1.0	31096
8	Monday	member	electric_bike	10.03	178.73	1.0	26820
9	Saturday	casual	classic_bike	23.22	178.85	1.0	12314

We need to adjust the query since, although the results are correct, saving it as a table in BigQuery separates data by member type, and the days of the week are not in a logical order. To address this, we will use the CASE command to assign numerical values to the days in our day_of_week column so that they can be logically ordered from Monday to Sunday.

```
SELECT
   day_of_week,
   member_casual,
  rideable_type
    ROUND(AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2)
AS avg_ride_length_minutes,
    ROUND(MAX(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2)
AS max_ride_length_minutes,
   ROUND(MIN(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS
min_ride_length_minutes,
   COUNT(ride_id) AS total_trips,
   CASE
      WHEN day_of_week = 'Monday' THEN 1
      WHEN day_of_week = 'Tuesday' THEN 2
      WHEN day_of_week = 'Wednesday' THEN 3
      WHEN day_of_week = 'Thursday' THEN 4
      WHEN day_of_week = 'Friday' THEN 5
      WHEN day_of_week = 'Saturday' THEN 6
      WHEN day_of_week = 'Sunday' THEN 7
   END AS day_order
FROM
```

```
    `decoded-tesla-418214.Data_cyclistic_cleaned.2023_10`
GROUP BY
    day_of_week, day_order, member_casual, rideable_type
ORDER BY
    day_order ASC, member_casual ASC;
```

Fila	day_of_week	member_casual	avg_ride	max_ride_l	min_ri	total_trip	day_o
1	Monday	casual	17.04	180.9	1.0	23316	1
2	Monday	member	10.65	178.73	1.0	57916	1
3	Tuesday	casual	16.45	180.48	1.0	26098	2
4	Tuesday	member	11.33	180.68	1.0	67116	2
5	Wednesday	casual	15.87	180.2	1.0	22363	3
6	Wednesday	member	11.2	179.9	1.0	54686	3

Now, we can see the table organized by day and member type results. We will repeat this process for the other months, using the same query while adjusting the year and month parameters for each table to ensure easy access to results when comparing different months.

With the dataset cleaned and containing the required data, we will run a query to view the results for a specific vehicle type initially, and later, for the other types. If a comparison is needed, we can refer directly to the cleaned monthly table, which contains all this information.

```
SELECT
*
FROM
    `decoded-tesla-418214.Data_cyclistic_cleaned.2023_10_results`
WHERE
    rideable_type IN ('classic_bike')
ORDER BY
    day_order ASC, member_casual ASC, rideable_type ASC;
```

day_of_week	member_casu	rideable_type	avg_ride	max_ride	min_ri	total_trips	day_o
Monday	casual	classic_bike	22.09	180.9	1.0	10815	1
Monday	member	classic_bike	11.19	172.22	1.0	31096	1
Tuesday	casual	classic_bike	21.0	180.48	1.0	11935	2
Tuesday	member	classic_bike	11.84	180.15	1.0	34997	2
Wednesday	casual	classic_bike	20.5	180.2	1.0	9696	3
Wednesday	member	classic_bike	11.58	179.9	1.0	27944	3
Thursday	casual	classic_bike	18.19	180.27	1.0	8149	4
Thursday	member	classic_bike	11.22	176.28	1.0	25032	4

This method allows us to visualize the data. We will apply this to all months, modifying the month names in the table accordingly.

```
SELECT
*
FROM
   `decoded-tesla-418214.Data_cyclistic_cleaned.2023_12_results`
WHERE
   rideable_type IN ('electric_bike')
ORDER BY
   day_order ASC, member_casual ASC, rideable_type ASC;
```

For the next step, we will use the CREATE OR REPLACE TABLE function to combine these 12 tables into a single file for future reference.

```
CREATE OR REPLACE TABLE `decoded-tesla-418214.Data_cyclistic_cleaned.all_months` AS
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2023_10`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2023_11`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2023_12`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_01`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_02`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_03`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_04`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_05`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_06`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_07`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_08`
UNION ALL
SELECT * FROM `decoded-tesla-418214.Data_cyclistic_cleaned.2024_09`;
```

Using the following query, we will generate a complete dataset with the most relevant data for our analysis, including month, user type, vehicle type, and the calculated averages. We will save this table as all.months_cleaned.

```
   ROUND(AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS avg_ride_length_minutes,
   ROUND(MAX(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS max_ride_length_minutes,
```

```
    ROUND(MIN(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) / 60, 2) AS
min_ride_length_minutes,
    COUNT(ride_id) AS total_trips
FROM
    `decoded-tesla-418214.Data_cyclistic_cleaned.all_months`
GROUP BY
    month, day_of_week, member_casual, rideable_type
ORDER BY
    month ASC, day_of_week ASC, member_casual ASC, rideable_type ASC;
```

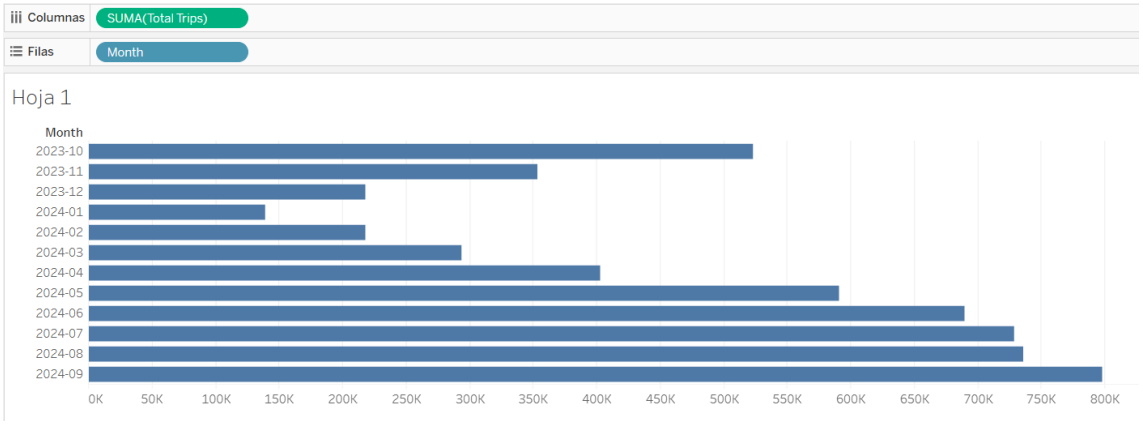
Fila	month	day_of_week	member_casual	rideable_type	avg_ride_length	max_ride_length	min_ride_length	total_trips
1	2023-10	Friday	casual	classic_bike	20.56	179.65	1.0	8971
2	2023-10	Monday	casual	classic_bike	22.09	180.9	1.0	10815
3	2023-10	Saturday	casual	classic_bike	23.22	178.85	1.0	12314
4	2023-10	Sunday	casual	classic_bike	25.81	180.63	1.0	19026
5	2023-10	Thursday	casual	classic_bike	18.19	180.27	1.0	8149
6	2023-10	Tuesday	casual	classic_bike	21.0	180.48	1.0	11935
7	2023-10	Wednesday	casual	classic_bike	20.5	180.2	1.0	9696
8	2023-11	Friday	casual	classic_bike	18.02	178.83	1.0	5318
9	2023-11	Monday	casual	classic_bike	18.65	180.53	1.12	4409
10	2023-11	Saturday	casual	classic_bike	22.02	180.93	1.02	8738
11	2023-11	Sunday	casual	classic_bike	22.92	177.63	1.02	6499
12	2023-11	Thursday	casual	classic_bike	17.8	179.15	1.0	7294
13	2023-11	Tuesday	casual	classic_bike	16.06	179.47	1.02	4057
14	2023-11	Wednesday	casual	classic_bike	16.33	180.42	1.0	5181
15	2023-12	Friday	casual	classic_bike	18.45	179.4	1.08	3553
16	2023-12	Monday	casual	classic_bike	18.4	180.58	1.1	1920
17	2023-12	Saturday	casual	classic_bike	18.68	176.38	1.02	4078
18	2023-12	Sunday	casual	classic_bike	18.8	178.9	1.02	3016

Although we have saved the tables in BigQuery, they may not always appear in the desired order. To ensure proper sorting, we will use the following query so that the table always appears in order by the day of the week.

```
SELECT
*
FROM
    `decoded-tesla-418214.Data_cyclistic_cleaned.2023_12_results`
WHERE
    rideable_type IN ('electric_bike')
ORDER BY
    day_order ASC, member_casual ASC, rideable_type ASC;
```

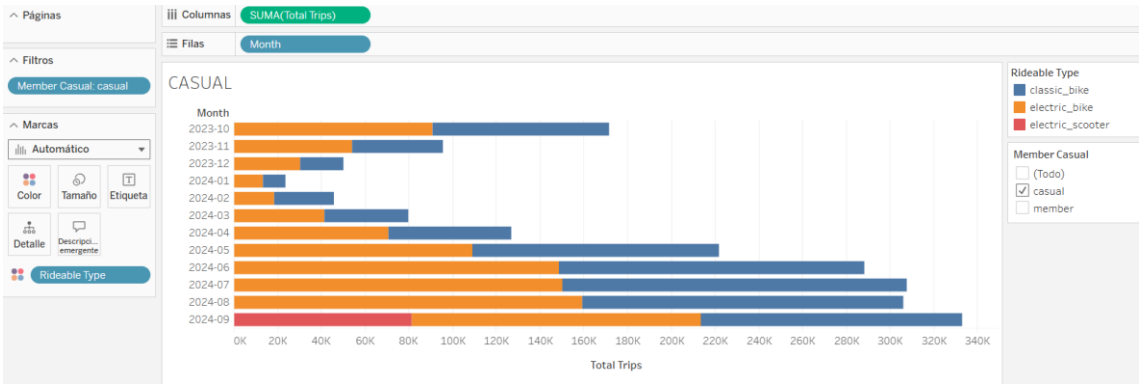
At this point, we switch to Tableau to visualize these data more graphically and intuitively.

After downloading the data from BigQuery as a CSV file, we import it into Tableau Public. For the first general visualization, we place total_trips in the columns and Month in the rows. The result is as follows:



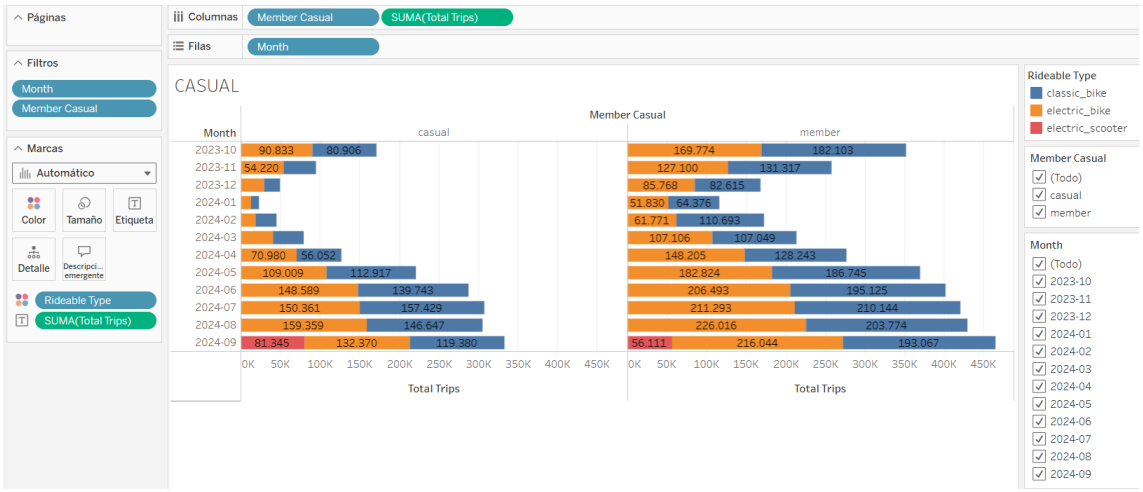
We observe that the total number of trips begins to decline in November (354,133) compared to October (523,616). This downward trend continues until January, which has the lowest number of trips (139,753). From February onward, the number of trips begins to increase, reaching 218,282 in February and continuing to rise until September. This decline may be due to seasonal factors, with usage increasing as temperatures rise and seasons change.

We experimented with different column, row, filter, and color configurations and created detailed bar charts displaying total trips per month, vehicle type, and filtering options by member type.



By hovering over the bars, we can see the total trips for each vehicle type. On the right, a color legend is displayed, with blue representing classic_bike, orange for electric_bike, and red for electric_scooter. Additionally, there is a filter to select casual members, annual members, or both. Adding total_trips to the label section returns numerical trip totals for each bar.

For a comparative analysis, we placed member_casual and avg_ride_length in the columns, month and day_of_week in the rows, and applied member_casual and month as filters. The show option was selected to make them visible on the right side of the screen. Finally, rideable_type was assigned to color, and total_trips was set as detail. This setup provides the following visualization. It allows for a comparative analysis of users, displaying vehicle usage averages, total trips by day of the week, and monthly trends. The filter options allow us to focus on specific months for further insights.



When analyzing the general data, we observe that **annual members** complete a **higher number of total trips** than casual users, suggesting that annual members use Cyclistic as **their primary mode of transportation**.

However, during the **launch month of electric scooters**, **casual users** recorded a significantly higher number of trips with this vehicle type, **surpassing annual members** by **30,000 trips**. This indicates a **rapid adoption** of scooters among casual users, likely driven by their interest in trying new transportation options.

Additionally, casual members exhibit longer average usage times compared to annual members, particularly for classic_bikes. A pattern is also observed on Saturdays and Sundays.

We will change the filter from member_casual to day_of_week to focus on weekday trends and examine the higher weekend usage pattern.

This graph highlights a higher average usage of classic_bikes among casual users, as well as a greater, though less pronounced, usage of electric_bikes. This trend increases during the summer months. We will compare three winter months against three summer months.

We can observe that casual members increase their average usage during the summer months, particularly with classic_bikes rather than electric_bikes. We also note higher usage of electric_scooters.

This graph displays complete data for August and September. We clearly see that casual members have a higher average usage time compared to annual members when using electric_scooters. Combined with previous insights, we conclude that during the first month of the electric_scooter launch, casual members took more trips and had a longer average usage compared to annual members.

We can conclude that casual members use the Cyclistic service primarily for recreational purposes, tourism, or infrequent travel, whereas annual members use it for short and frequent commutes.

In summary, we performed various analyses, including calculating the average trip duration, maximum and minimum trip durations, and ride frequency per day of the week for both member types.

Key findings:

1. **User trends:**

Annual members completed a higher number of trips throughout the year, totaling 3,645,605, which accounts for **63.99%** of the total. This suggests a more frequent and consistent use of the service, likely as a **primary mode of transportation**.

Casual members recorded a significantly longer average trip duration, exceeding that of annual members by **54.45%**. This indicates a more **recreational or tourism-oriented use of the service**.

2. **Seasonal Decline During Winter Months:**

A drop in service usage was observed during the winter months. Between December and February, **casual members** completed a total of 119,623 trips, representing only **2.10%** of the annual trips. In contrast, **annual members completed** 457,053 trips in the same period, accounting for **8.02%** of **total trips for the year.**

3. **Successful Launch of the Electric Scooter:**

In its launch month, the **electric scooter** received an **excellent response**, particularly among casual members, who completed 81,406 trips, representing **22.43%** of their **total trips for the month**. In comparison, **annual members** recorded 56,130 trips, making up **12.07%** of **their total trips for the month**.

4. **Increased Summer Usage by Casual Members:**

During the summer, **casual members** significantly **increased** both the **total number of trips** and their **average ride duration**, particularly on weekends. Compared to the rest of the year, the number of trips taken by casual members rose by **85.25% for classic bikes** and **38.48% for electric bikes**. Additionally, the **average trip duration increased by 22.52% for classic bikes** and **29.06% for electric bikes** on Saturdays and Sundays, reinforcing the seasonal recreational trend.