REPORT

Case study: How does a bike-share navigate speedy success?

March 2024
Author Maria Orlova

Introduction

The data analytics case study was prepared by the learning platform Coursera. This is an end-of-certificate project or capstone which allows the opportunity to practice all new knowledge and skills. To answer the business questions, I have to follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act. The Case Study Roadmap tables will help me stay on the right path.

I have a scenario where I am a junior data analyst working on the marketing analyst team at Cyclistic, a bike-share company in Chicago. Cyclistic has two user types: casual riders and members. The user type depends on pricing plans. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members. Our team wants to understand how casual riders and annual members use Cyclistic bikes differently. The team will design a new marketing strategy from these insights to convert casual riders into annual members.

Lily Moreno (the director of marketing and my manager) has assigned me the question to answer: How do annual members and casual riders use Cyclistic bikes differently?

I have to produce a report with the following deliverables:

1. A statement of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of analysis
5. Supporting visualizations and key findings
6. Top three recommendations based on my analysis

1. A statement of the business task

Identify differences in Cyclistic bicycle use by casual riders and members. Developing three recommendations to help convert casual riders into members.

2. A description of all data sources used

The data that I am going to work with is first-party data. The data has been made available by Motivate International Inc. This is public data that I can use to explore how different customer types are using Cyclistic bikes. There aren’t any issues with bias or credibility in this data. Our data is reliable, original, comprehensive, current and cited. The data privacy issues prohibit me from using riders’ personally identifiable information.

The data is organized in 12 .csv files, each file contains the data about one month of 2023 year. The sizes of these files are from 37551 KB to 151472 KB.

3. Documentation of any cleaning or manipulation of data

At first, I decided to explore the data with Google Sheets. I imported file 202301-divvy-tripdata.csv (data for January 2023). I named the file “Cyclistic_2023_01”. This table consisted of 12 columns:

A	ride_id	H	end_station_id
B	rideable_type	I	start_lat
C	started_at	J	start_lng
D	ended_at	K	end_lat
E	start_station_name	L	end_lng
F	start_station_id	M	member_casual
G	end_station_name		

The table had 190302 rows.

I have done some steps in Google Sheets to get more familiar with the data (Table 1).

Table 1. An exploration of the data with Google Sheets

<table>
<tr><th>№</th><th>Step</th><th>Result</th></tr>
<tr><td>1</td><td>Remove duplicates</td><td>now duplicate rows were found, 190301 unique rows remain</td></tr>
<tr><td>2</td><td>Trim whitespase</td><td>trimmed whitespase from 16 selected sells</td></tr>
<tr><td colspan="3">I found some empty cells in column E and I wanted to determine the number of them. I decided to count cells with values, not empty.</td></tr>
<tr><td>3</td><td>= COUNTA(E1:E190302)
= COUNTA(F1:F190302)
= COUNTA(G2:G190303)
= COUNTA(H1:H190302)</td><td>163581
163581
162462
163581</td></tr>
<tr><td colspan="3">I calculated the percentage of cells with a value out of the total number of cells in column E. The same calculation I made for columns F, G, and H.</td></tr>
<tr><td>4</td><td>= 163581 / 190301 * 100
= 162462 / 190301 * 100</td><td>85,96 %
85,37%</td></tr>
<tr><td colspan="3">So, table “Cyclistic_2023_01” had about 85 % cells with values and 15 % empty cells.</td></tr>
<tr><td colspan="3">Then, I wanted to check column M.</td></tr>
<tr><td>5</td><td>= COUNTA(M1:M190302)</td><td>190302 (column M is without empty cells)</td></tr>
</table>

6	conditional formatting “text is exactly "casual"”; “text is exactly "member"”	I didn’t see cells without assigned colors.
7	=COUNTIF(M1:M190302, "casual") =COUNTIF(M1:M190302, "member")	40008 + 150293=190302 There aren’t any values except "casual" and "member".
I added a new column N ride_duration. After that, I calculated the duration of each ride by subtracting the column started_at from the column ended_at. I formatted the values in this column as HH:MM:SS using Format > Cells > Time.		
8	= D2-C2	A new column was created. I sorted the sheet by column N (Z to A). The max value was 560:03:44 and the min value was 0:00:00.
I created a new column O day_of_week and found a day of the week where the trip started.		
9	=WEEKDAY(C2)	The day of the week showed like number from 1 to 7. Days were counted from Sunday and the value of Sunday is 1, therefore the value of Saturday is 7.
10	I scrolled down the columns I (start_lat), J (start_lng), K (end_lat) and L (end_lng).	The values in cells had different amount of numbers after the coma. Most of the values had eight numbers after coma, but some had only two. If there were only two numbers after coma in the cells with coordinates, the cells with the name of the station were empty in this row.

Then, I imported file 202302-divvy-tripdata.csv (data for February 2023). I named the file “Cyclistic_2023_02”. This file had the same structure as “Cyclistic_2023_01”. There were 190446 rows in February’s file. I have done all the previous steps which are described in “*Table 1. An exploration of the data with Google Sheets*” in this file. I found the same problems with data on file “Cyclistic_2023_02”.

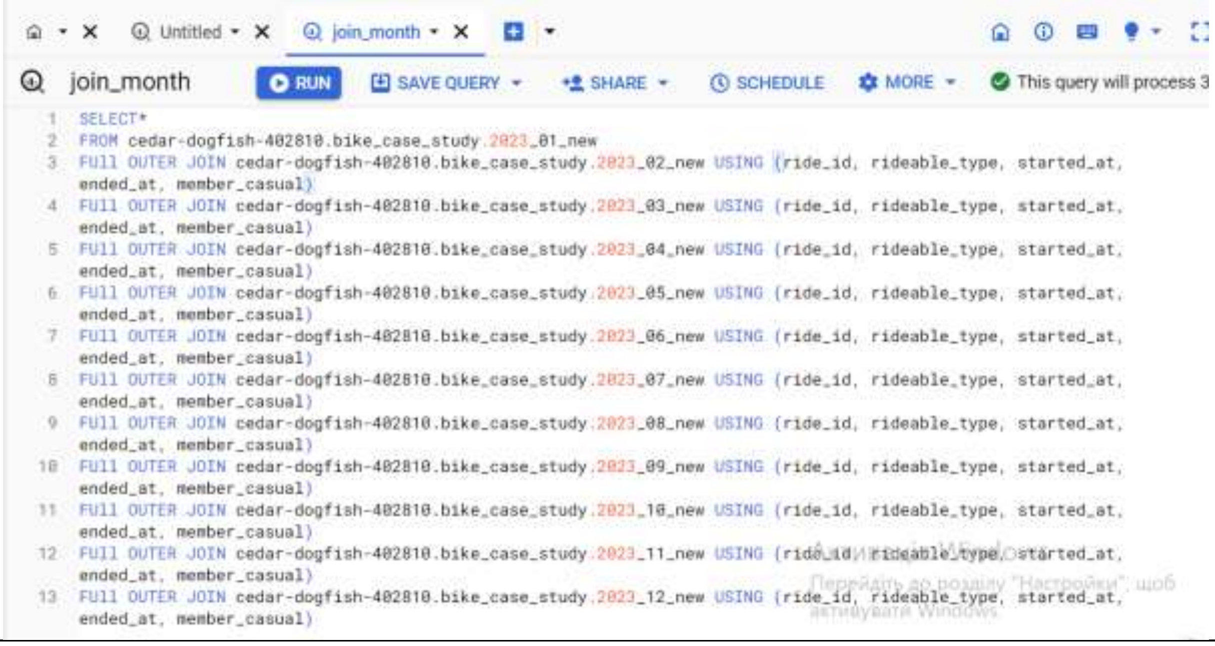
The columns E (start_station_name), F (start_station_id), G (end_station_name), and H (end_station_id) had many empty cells. Each of these columns has about 15 % empty cells. I couldn’t recognize the name of the station with its id because if I had missing data in the station name, I had missing data in the station id too. The data about the start and end stations wasn’t complete. I had an inconsistent format of the data in columns I (start_lat), J (start_lng), K (end_lat), and L (end_lng).

I couldn’t import the next file 202303-divvy-tripdata.csv (data for March 2023) because it was a larger size than the previous ones. The April-Desember files were larger too, so I decided to move to BigQuery (Google Cloud) and continue to work with SQL.

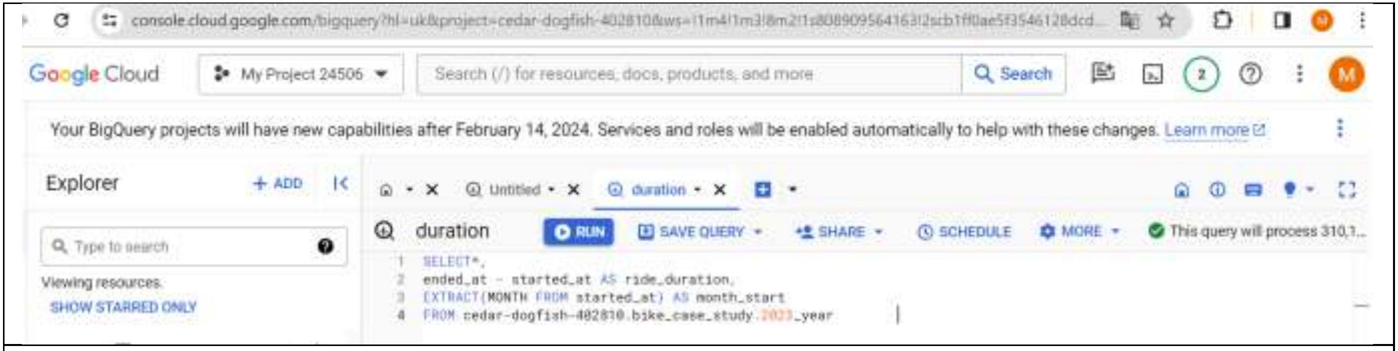
At first, I had to import all the monthly files in Big Query. I created a dataset “bike_case_study”. Then I wanted to create 12 new tables in this dataset and upload 12 monthly files. I easily created and uploaded the first four files, but when I was creating a fifth table Big Query reported that it was impossible. It is recommended to use Google Cloud Storage for files larger than 100 MB. I had to create a bucket in Google Cloud Storage and put their files larger than 100 MB. After that, I created tables and uploaded files without problems.

All my steps with SQL in BigQuery I described in Table 2.

Table 2. An exploration, manipulation, and analysis of the data with SQL

<table>
<tr><th>№</th><th>Step</th><th>Result</th></tr>
<tr><td colspan="3">I wanted to explore the columns which had missing data.</td></tr>
<tr><td>1</td><td>SELECT count(ride_id) AS num_ride
FROM cedar-dogfish-402810.bike_case_study.2023_05
WHERE start_station_id IS null</td><td>89240 (the whole number of rows is 604827, so about 14,75 % of values are missing)</td></tr>
<tr><td>2</td><td>SELECT count(ride_id) AS num_ride
FROM cedar-dogfish-402810.bike_case_study.2023_06
WHERE start_station_id is null</td><td>116259 (the whole number of rows is 719618, so about 16,15 % of values are missing)</td></tr>
<tr><td>3</td><td>SELECT count(ride_id) AS num_ride
FROM cedar-dogfish-402810.bike_case_study.2023_07
WHERE start_station_id is null</td><td>122943 (the whole number of rows is 767650, so about 16,02 % of values are missing)</td></tr>
<tr><td colspan="3">I decided not to use columns with missing data. I selected columns ride_id, rideable_type, started_at, ended_at, member_casual and created the new tables with names 2023_01_new, 2023_02_new, etc.</td></tr>
<tr><td colspan="3">3. I joined all monthly tables in one and named it 2023_year.</td></tr>
<tr><td colspan="3">

<pre>SELECT*
FROM cedar-dogfish-402810.bike_case_study.2023_01_new
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_02_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_03_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_04_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_05_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_06_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_07_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_08_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_09_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_10_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_11_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)
Full OUTER JOIN cedar-dogfish-402810.bike_case_study.2023_12_new USING (ride_id, rideable_type, started_at, ended_at, member_casual)</pre></td></tr>
<tr><td colspan="3">4. I decided to add new columns to my table 2023_year. There were columns ride_duration and month_start.</td></tr>
</table>



console.cloud.google.com/bigquery?hl=uk&project=cedar-dogfish-402810&ws=!1m4!1m3!8m2!1s808909564163!2scb1ff0ae5f3546128dcd...
Google Cloud
My Project 24506
Search (/) for resources, docs, products, and more
Search
Your BigQuery projects will have new capabilities after February 14, 2024. Services and roles will be enabled automatically to help with these changes. Learn more
Explorer
+ ADD
Type to search
Viewing resources.
SHOW STARRED ONLY
Untitled
duration
RUN
SAVE QUERY
SHARE
SCHEDULE
MORE
This query will process 310,1...

```
SELECT*,
ended_at - started_at AS ride_duration,
EXTRACT(MONTH FROM started_at) AS month_start
FROM cedar-dogfish-402810.bike_case_study.2023_year
```

I saved the result as table 2023_year_new. I ran the query to calculate min, max, and average values of ride duration.

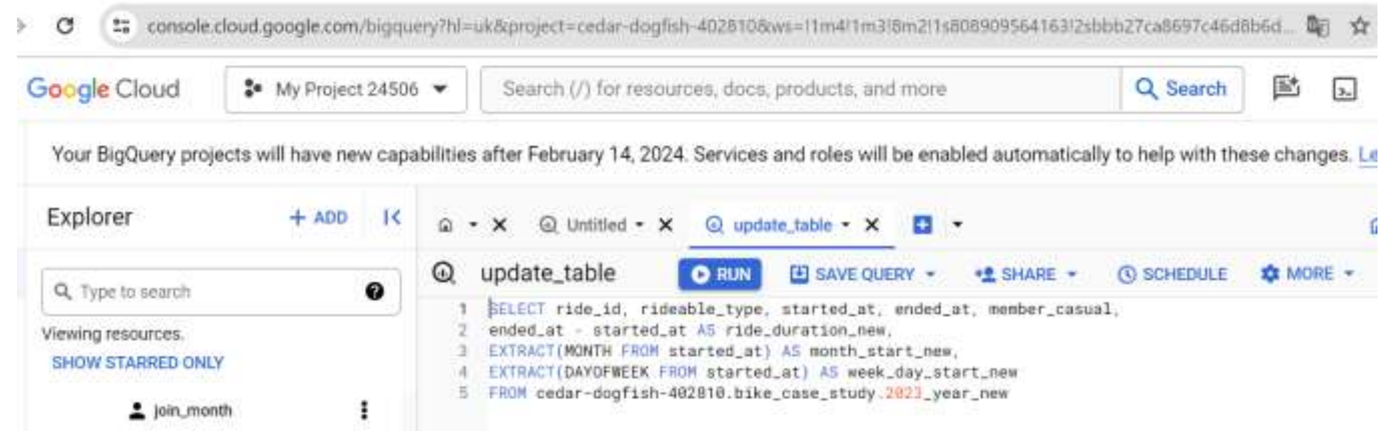
5	SELECT month_start, MAX(ride_duration) AS max_duration,
	MIN(ride_duration) AS min_duration,
	AVG(ride_duration) AS avg_duration
	FROM cedar-dogfish-402810.bike_case_study.2023_year_new
	GROUP BY month_start
		The result of the min value at column ride_duration was a negative number.

6	SELECT*
	FROM cedar-dogfish-402810.bike_case_study.2023_year_new
	ORDER BY ride_duration
		231 rows in column ride_duration have negative values because the values in the started_at and ended_at columns have been swapped.

I had to fix this error in the data.

7	UPDATE cedar-dogfish-402810.bike_case_study.2023_year_new
	set started_at = ended_at,
	ended_at = started_at
	WHERE ended_at < started_at
		This statement modified 231 rows in 2023_year_new.

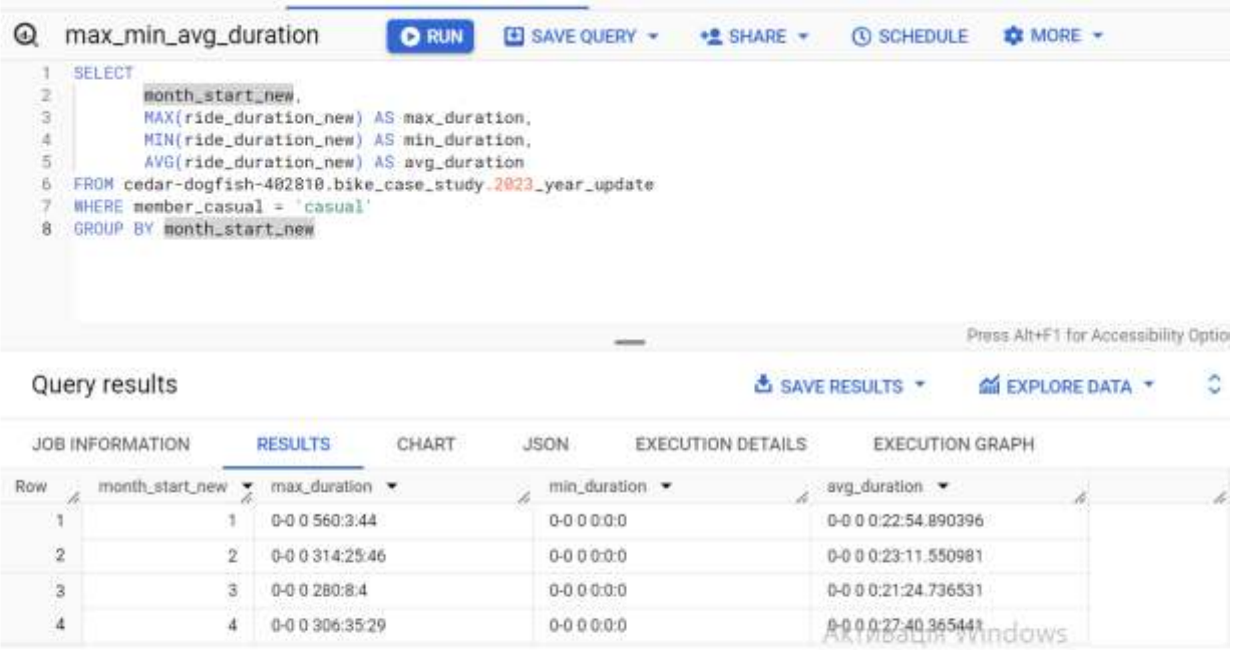
8. After that, I needed to update data in the columns ride_duration and month_start. At the same time, I added a new column week_day_start_new. I saved the result as table 2023_year_update.



console.cloud.google.com/bigquery?hl=uk&project=cedar-dogfish-402810&ws=!1m4!1m3!8m2!1s808909564163!2sbbb27ca8697c46d8b6d...
Google Cloud
My Project 24506
Search (/) for resources, docs, products, and more
Search
Your BigQuery projects will have new capabilities after February 14, 2024. Services and roles will be enabled automatically to help with these changes.
Explorer
+ ADD
Type to search
Viewing resources.
SHOW STARRED ONLY
join_month
Untitled
update_table
RUN
SAVE QUERY
SHARE
SCHEDULE
MORE

```
SELECT ride_id, rideable_type, started_at, ended_at, member_casual,
ended_at - started_at AS ride_duration_new,
EXTRACT(MONTH FROM started_at) AS month_start_new,
EXTRACT(DAYOFWEEK FROM started_at) AS week_day_start_new
FROM cedar-dogfish-402810.bike_case_study.2023_year_new
```

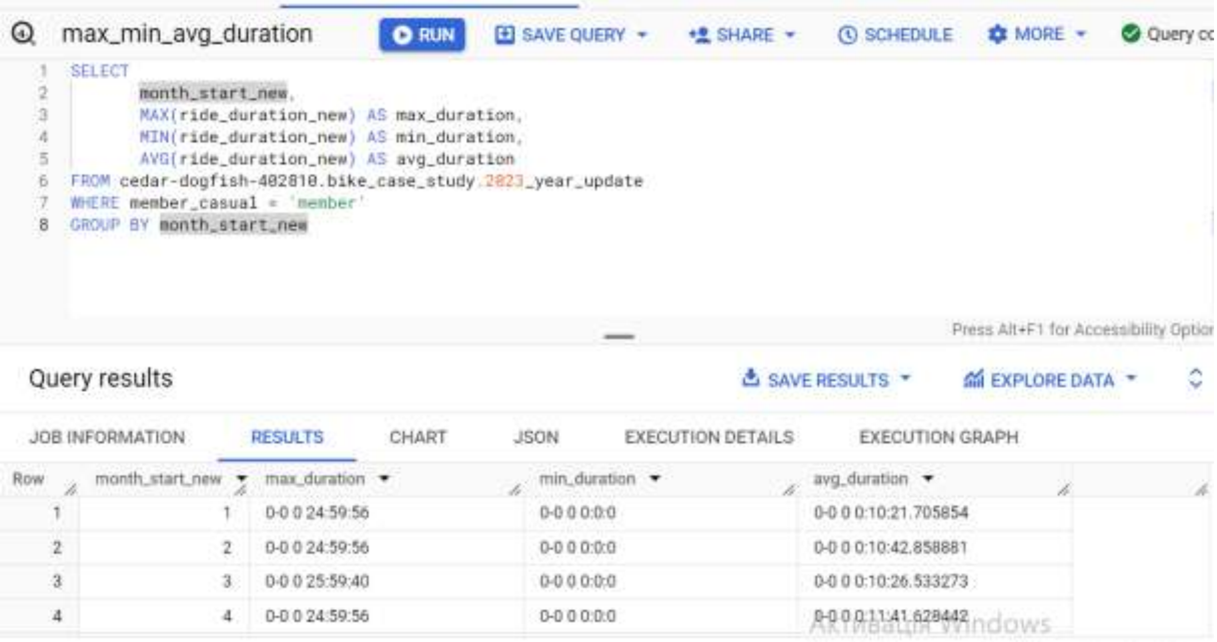
9. I had to run the query to calculate the min, max, and average values of the ride duration again in the updated table 2023_year_update. At first, I did it for the casual riders. I saved the result as a Google Sheets.



```
SELECT
       month_start_new,
       MAX(ride_duration_new) AS max_duration,
       MIN(ride_duration_new) AS min_duration,
       AVG(ride_duration_new) AS avg_duration
FROM cedar-dogfish-402810.bike_case_study.2023_year_update
WHERE member_casual = 'casual'
GROUP BY month_start_new
```

Row	month_start_new	max_duration	min_duration	avg_duration
1	1	0-0 0 560:3:44	0-0 0 0:0:0	0-0 0 0:22:54.890396
2	2	0-0 0 314:25:46	0-0 0 0:0:0	0-0 0 0:23:11.550981
3	3	0-0 0 280:8:4	0-0 0 0:0:0	0-0 0 0:21:24.736531
4	4	0-0 0 306:35:29	0-0 0 0:0:0	0-0 0 0:27:40.365441

10. After that, I run the same query for the member riders. I saved the result as a Google Sheets.



```
SELECT
       month_start_new,
       MAX(ride_duration_new) AS max_duration,
       MIN(ride_duration_new) AS min_duration,
       AVG(ride_duration_new) AS avg_duration
FROM cedar-dogfish-402810.bike_case_study.2023_year_update
WHERE member_casual = 'member'
GROUP BY month_start_new
```

Row	month_start_new	max_duration	min_duration	avg_duration
1	1	0-0 0 24:59:56	0-0 0 0:0:0	0-0 0 0:10:21.705854
2	2	0-0 0 24:59:56	0-0 0 0:0:0	0-0 0 0:10:42.858881
3	3	0-0 0 25:59:40	0-0 0 0:0:0	0-0 0 0:10:26.533273
4	4	0-0 0 24:59:56	0-0 0 0:0:0	0-0 0 0:11:41.628442

I calculated how many hours rides by casual riders were been in sum. And how many hours were rides by member riders?

11	SELECT SUM(ride_duration_new) AS sum_duration
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'
		0-0 0 998718:10:54

12	SELECT SUM(ride_duration_new) AS sum_duration
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
		0-0 0 763945:53:6

Then, I counted the total number of rides by two types of riders.

13	SELECT COUNT(ride_id) AS num_id	2094241

	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'	

14	SELECT COUNT(ride_id) AS num_id
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
		3660604

Next, I counted the number of rides by the month of the year for two types of riders.		

15	SELECT month_start_new,
	COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'
	GROUP BY month_start_new
		I saved the result as a Google Sheets.

16	SELECT month_start_new,
	COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
	GROUP BY month_start_new
		I saved the result as a Google Sheets.

Then, I calculated the number of rides by the days of the week for two types of riders.		

17	SELECT week_day_start_new,
	COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'
	GROUP BY week_day_start_new
		I saved the result as a Google Sheets.

18	SELECT week_day_start_new,
	COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
	GROUP BY week_day_start_new
		I saved the result as a Google Sheets.

I wanted to find max and average values of ride duration by type of bike for casual and member riders.		

19	SELECT rideable_type,
	MAX(ride_duration_new) AS max_duration,
	AVG(ride_duration_new) AS avg_duration
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'
	GROUP BY rideable_type
		I saved the result as a Google Sheets.

20	SELECT rideable_type, MAX(ride_duration_new) AS max_duration,	I saved the result as a Google Sheets.

	AVG(ride_duration_new) AS avg_duration
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
	GROUP BY rideable_type	

After that, I calculated the ride number by type of bike.		

21	SELECT rideable_type, COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'casual'
	GROUP BY rideable_type
		I saved the result as a Google Sheets.

22	SELECT rideable_type, COUNT(ride_id) AS num_ride
	FROM cedar-dogfish-402810.bike_case_study.2023_year_update
	WHERE member_casual = 'member'
	GROUP BY rideable_type
		I saved the result as a Google Sheets.

When I was processing the data with SQL in BigQuery (Google Cloud) I recognized one more problem with data. I calculated the duration of each ride by subtracting the column started_at from the column ended_at and created a column ride_duration. Then I counted the min value at column ride_duration and the result was a negative number. I explored my table 2023_year (which contains the data from all monthly tables) and found that 231 rows in column ride_duration have negative values because the values in the started_at and ended_at columns have been swapped. I had to fix this error in the data and I used the update function for that. This statement modified 231 rows.

I created the final table 2023_year_update for analysis by updating the data in the columns ride_duration and month_start and adding a new column week_day_start_new. After that, I calculated the minimum, maximum, and average values of the ride duration for casual and member riders. I counted the number of rides by the month of the year and by the days of the week for two types of riders. I found the max and average values of ride duration and the ride number by type of bike.

4. A summary of analysis

The analysis results with SQL were saved and organized in Google Sheets tables.

Table 3. The total number of rides and total duration of rides

Type of riders	Sun_num_ride	Sun_num_ride, %	Sum_duration_ride	Sum_duration_ride, %
casual	2094241	36.39%	998718:10:54	56.66%
member	3660604	63.61%	763945:53:06	43.34%
Total	5754845	100.00%	1762664:04:00	100.00%

Table 4. The ride duration and the number of rides by the month of the year for two types of riders

trip_month_start	casual_trip_max_duration	member_trip_max_duration	casual_trip_avg_duration	member_trip_avg_duration	casual_trip_number	member_trip_number
January	560:03:44	24:59:56	0:22:55	0:10:22	40008	150293
February	314:25:46	24:59:56	0:23:12	0:10:43	43016	147429
March	280:08:04	25:59:40	0:21:25	0:10:27	62201	196477
April	306:35:29	24:59:56	0:27:40	0:11:42	147285	279305
May	486:50:31	25:00:31	0:28:31	0:13:02	234181	370646
June	491:05:58	24:59:56	0:29:24	0:13:12	301230	418388
July	857:41:24	24:59:57	0:32:20	0:13:41	331358	436292
August	1641:29:04	24:59:57	0:35:15	0:13:46	311130	460563
September	461:37:34	172:33:21	0:27:59	0:13:00	296697	404642
October	277:36:31	277:15:13	0:23:07	0:12:16	177074	360045
November	25:00:25	24:59:56	0:19:54	0:11:35	98389	264123
December	24:59:57	24:59:56	0:19:56	0:11:27	51672	172401

Table 5. The number of rides by the season of the year for two types of riders

Season	casual_trip_number	casual_trip_number, %	member_trip_number	member_trip_number,%
Winter	134696	6.43%	470123	12.84%
Spring	443667	21.19%	846428	23.12%
Summer	943718	45.06%	1315243	35.93%
Autumn	572160	27.32%	1028810	28.10%

Table 6. The number of rides by the days of the week for two types of riders

trip_day_start	casual_trip_number	casual_trip_number, %	member_trip_number	member_trip_number, %
Sunday	327574	15.64%	399294	10.91%
Monday	239764	11.45%	498453	13.62%
Tuesday	252940	12.08%	580721	15.86%
Wednesday	256491	12.25%	587151	16.04%
Thursday	288275	13.77%	604213	16.51%
Friday	322822	15.41%	533287	14.57%
Saturday	406375	19.40%	457485	12.50%

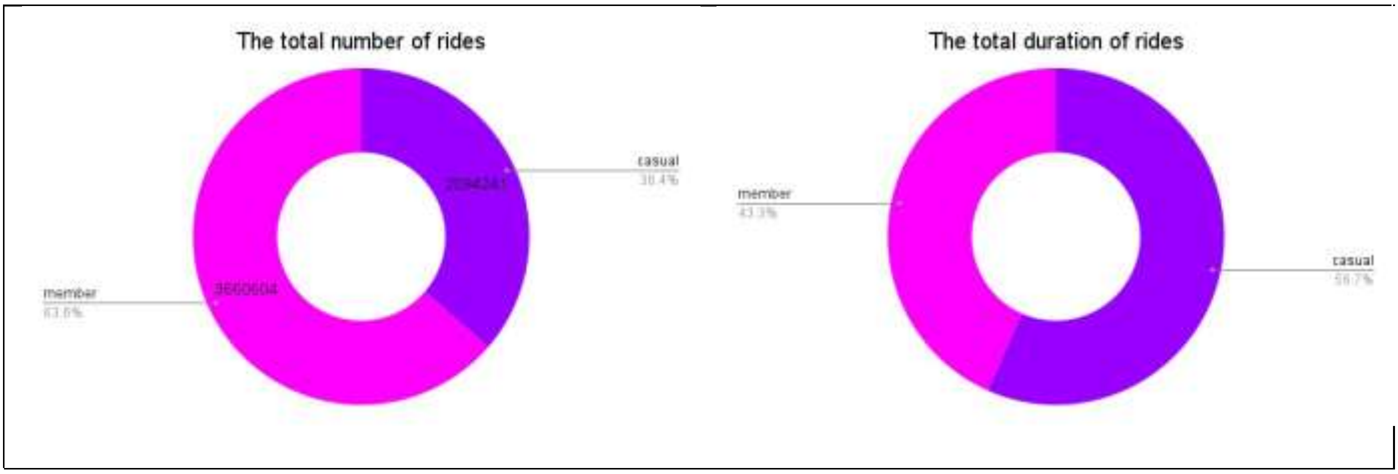
Table 7. The maximum and average ride duration by type of bike for casual and member riders

rideable_type	trip_max_duration_casual	trip_max_duration_member	trip_avg_duration_casual	trip_avg_duratiom_member
electric_bike	0-0 0 8:0:27	0-0 0 172:33:21	0-0 0 0:14:20.086534	0-0 0 0:11:8.212396
classic_bike	0-0 0 277:36:31	0-0 0 277:15:13	0-0 0 0:30:58.686881	0-0 0 0:13:56.735660
docked_bike	0-0 0 1641:29:4		0-0 0 2:54:55.167541	

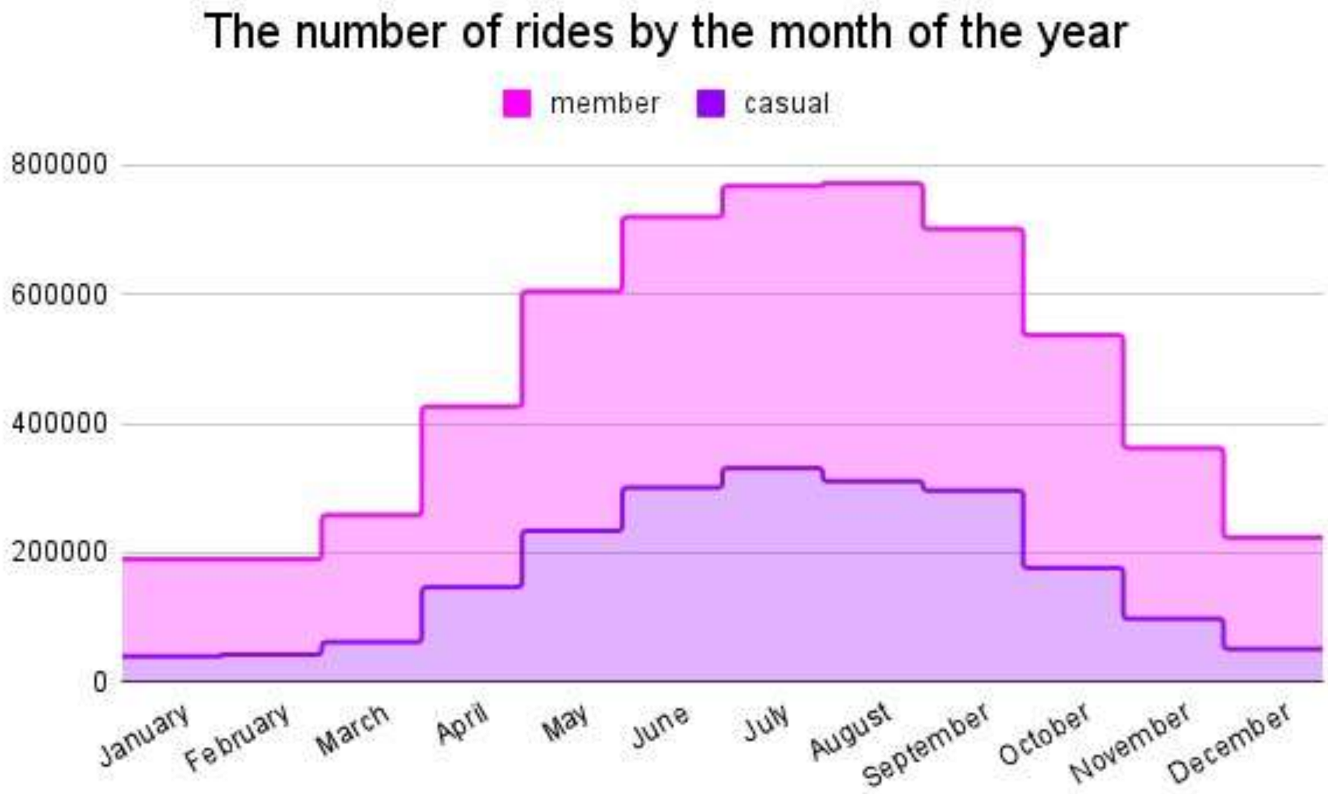
Table 8. The number of rides by type of bike for casual and member riders

rideable_type	trip_number_casual	trip_number_casual, %	trip_number_member	trip_number_member, %
electric_bike	1146108	54.73%	1855843	50.70%
docked_bike	98113	4.68%	0	0.00%
classic_bike	850020	40.59%	1804761	49.30%

5. Supporting visualizations and key findings



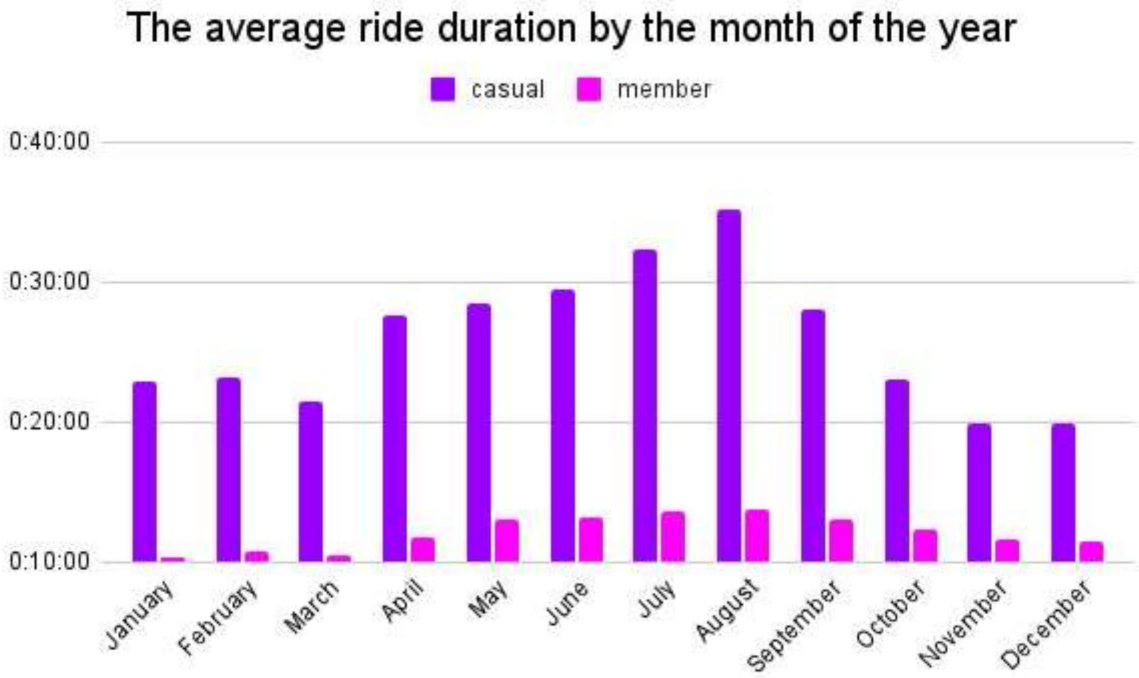
The casual riders did less number of rides, but the total duration of rides was more than member riders had. More than 63% of the total number of rides was done by members, but their duration was only about 43 % of the total ride duration.



In general, the distribution of the number of trips by month for both types of riders is similar. Starting from January, the number of trips increases and reaches a maximum in July for casual riders and in August for member riders. From these peaks to December, the number of rides gradually decreases.

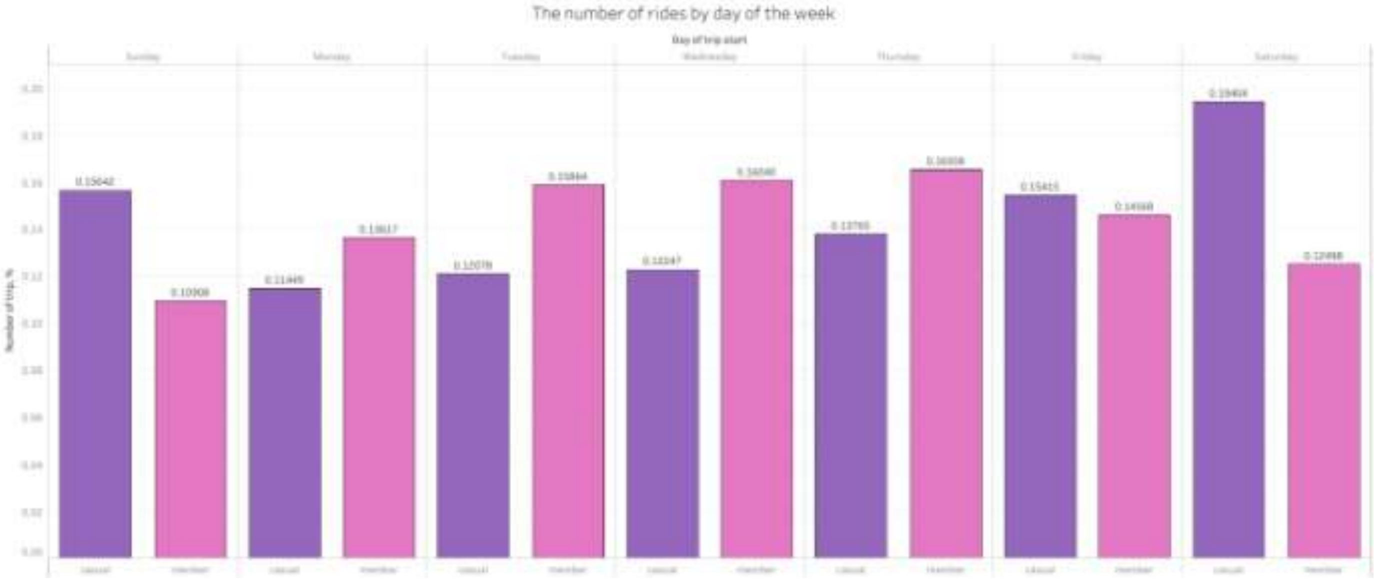
Table 9. The max ride duration by the month of the year

trip_month_start	casual_trip_max_duration	member_trip_max_duration
January	560:03:44	24:59:56
February	314:25:46	24:59:56
March	280:08:04	25:59:40
April	306:35:29	24:59:56
May	486:50:31	25:00:31
June	491:05:58	24:59:56
July	857:41:24	24:59:57
August	1641:29:04	24:59:57
September	461:37:34	172:33:21
October	277:36:31	277:15:13
November	25:00:25	24:59:56
December	24:59:57	24:59:56

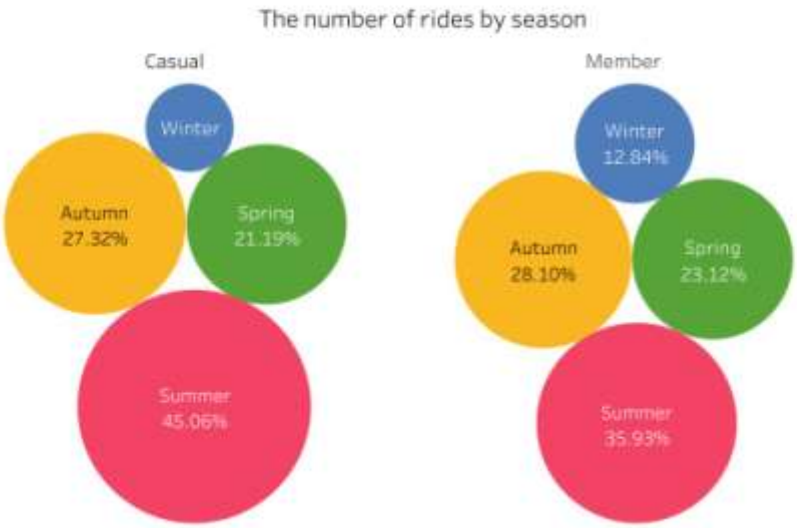


The max ride duration on all months is higher for casual riders. Especially long rides were done by casual riders in August (more than 68 days), January (more than 23 days), September (more than 19 days), and October (more than 11 days). The rides with max duration by members were done in October (more than 11 days) and September (more than 7 days).

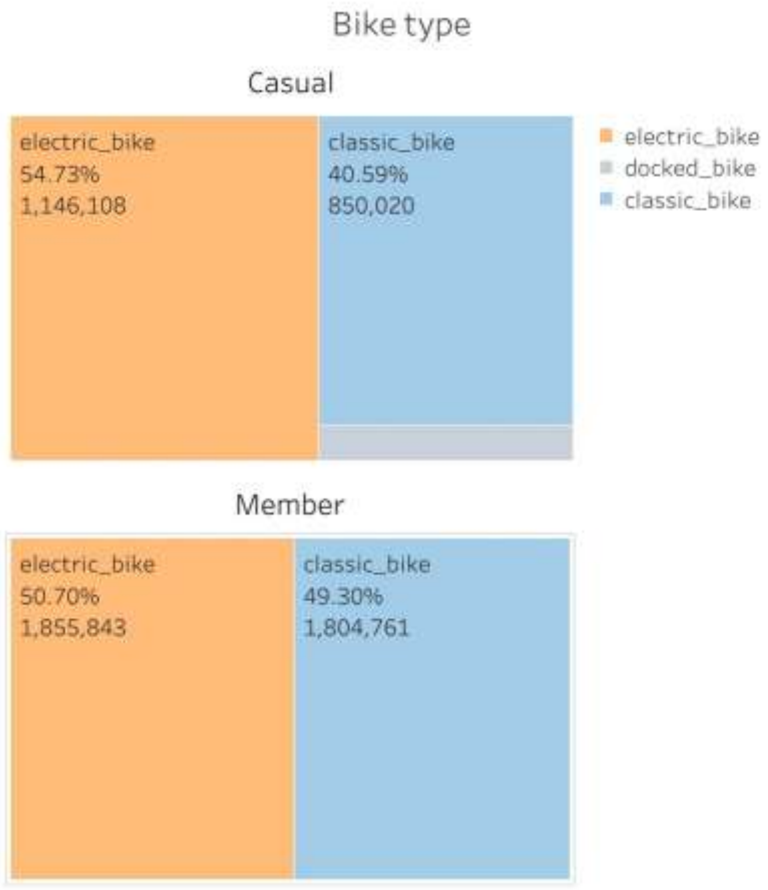
In all months, the average ride duration of casual riders is higher than that of member riders. The highest values of average trip duration were recorded in the period from April to September for casual riders (more than 27 minutes) and from May to September for members (13 or more minutes).



The casual riders do more rides on Fridays and weekends. The member riders more often use bikes on workdays.



The most popular season for rides by casual and member riders is summer. The least popular season for both types of riders is winter, but the percentage of winter rides is higher by member riders two times.



Casual riders use three types of bikes: electric, classic, and docked. More than half of rides by casual riders are done on electric bikes and nearly 5 % of rides are done on docked bikes. The member riders use only electric and classic bikes. The number of rides by electric and classic bikes is approximately equal.

6. Top three recommendations based on my analysis

1. Casual riders more often used bicycles for recreation. We need to emphasize the benefits of using bicycles for commuting to a job.

2. Some of the casual riders' trips were very long. Maybe, it was tourism trips. It is recommended to tell about the opportunities of using bikes for tourism. For such long trips, it would be more profitable to purchase annual memberships.

3. Casual riders more used bikes on weekends. There is a high probability that they want to ride with their family. Should offer a loyalty system for families.