CYCLISTIC BIKE-SHARE DATA ANALYSIS

- **About the company:**
- In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.
- Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One 2 approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.
- Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.
- Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

Question to answer:

How do annual members and casual riders use Cyclistic bikes differently?

Goal:

Converting casual riders into annual members

Data source:

The data has been made available by Motivate International Inc. and downloaded from
https://divvy-tripdata.s3.amazonaws.com/index.html
The chosen data range is January – December 2023. Each month data is stored in a separate zip folder containing relevant CSV files.

About the data :

- **Reliable** - it is a complete internal showing of all bike users. That excludes sample selection bias
- **Orginal** – direct access to the original data source of e Cyclistic's historical trip data
- **Comprehensive** – showing enough information to answer the question
- **Current** – data comes from 2023 and is relevant
- **Cited** – known reliable location and ownership

Joining separate data tables into a single one:

```
SELECT*
FROM `project-1-asia.Rides.Jan`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Feb`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.March`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.April`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.May`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.June`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.July`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Aug`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Sep`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Oct`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Nov`
UNION ALL
SELECT*
FROM `project-1-asia.Rides.Dec`
```

The data has 5,719,877 rows. Due to the high volume of data, and the need for aggregation and handling ease, SQL was used to check for potential issues and to pull summarized information.

Table columns and data overview:

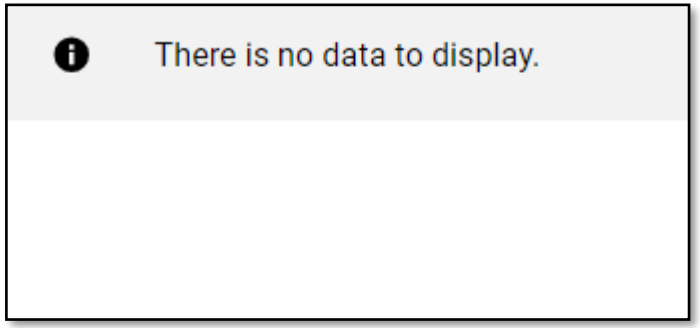
```
SELECT*
FROM  `project-1-asia.Rides.a_Full_Year`
```

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	member_casual
171EC2DFD542D2A4	electric_bike	2023-01-03 17:10:00 UTC	2023-01-03 17:11:00 UTC	*null*	*null*	*null*	*null*	member
9A63D36A3A19862F	electric_bike	2023-01-10 06:41:00 UTC	2023-01-10 06:48:00 UTC	*null*	*null*	*null*	*null*	casual
027A95333F29A515	electric_bike	2023-01-09 17:08:00 UTC	2023-01-09 17:32:00 UTC	*null*	*null*	*null*	*null*	casual

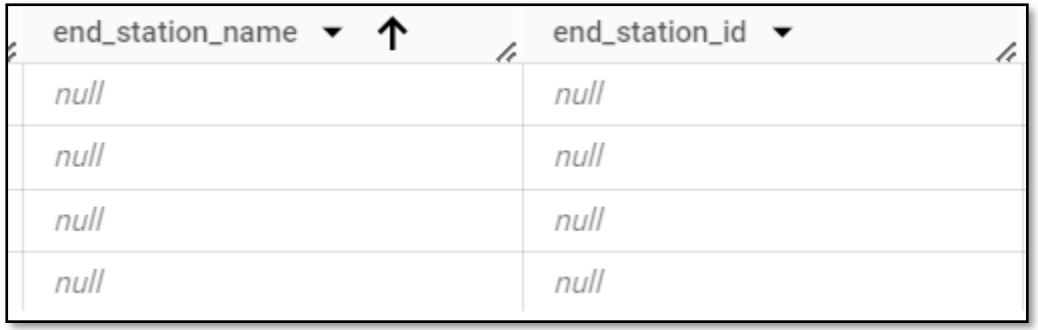
Results per page: 50 1 – 50 of 5719877

Checking for NULL and 0 using query

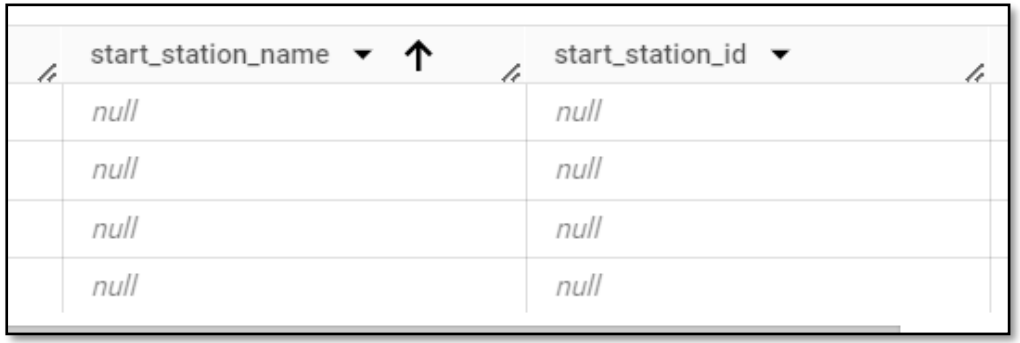
```
SELECT *
FROM `project-1-asia.Rides.Full_Year_sorted`
WHERE ride_id = '0' OR ride_id IS NULL
```



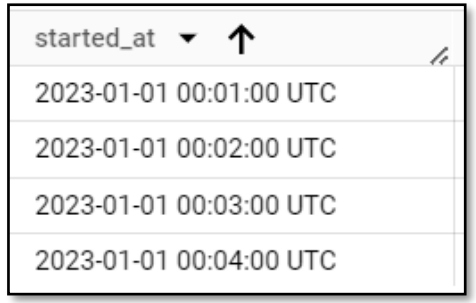
Ascending and descending sorting to show issues:



end_station_name	end_station_id
null	null
null	null
null	null
null	null



start_station_name	start_station_id
null	null
null	null
null	null
null	null



started_at
2023-01-01 00:01:00 UTC
2023-01-01 00:02:00 UTC
2023-01-01 00:03:00 UTC
2023-01-01 00:04:00 UTC

Exploring data: Checking counts of stations and rides per start station (3rd column):

```
SELECT
COUNT (start_station_id) as
start_stations_count
FROM `project-1-
asia.Rides.a_Full_Year`
```

Result:

Row	start_stations_count
1	4844029

```
SELECT
COUNT (end_station_id) as
end_stations_count
FROM `project-1-
asia.Rides.a_Full_Year`
```

Result:

Row	end_stations_count
1	4790534

```
SELECT
start_station_id,
COUNT
(ride_id) as rides_count,
FROM `project-1-
asia.Rides.a_Full_Year`
GROUP BY
start_station_id
```

Result:

Row	start_station_id	rides_count
1	*null*	875848
2	20	2869
3	308	65
4	310	72

Since start/end station IDs and names have a significant nubmer of missing (null) values, are not crucial to analysis and can't be provided by Cyclistic, they will be not included in analysis

Checking for duplicates:

```
SELECT
DISTINCT(rideable_type)
FROM `project-1-
asia.Rides.a_Full_Year`
```

Result: no duplicates

Row	rideable_type
1	electric_bike
2	docked_bike
3	classic_bike

```
SELECT
DISTINCT(member_casual)
FROM `project-1-
asia.Rides.a_Full_Year`
```

Result: no duplicates

Row	member_casual
1	member
2	casual

```
SELECT ride_id,
COUNT (ride_id) as duplicates_count
FROM `project-1-
asia.Rides.a_Full_Year`
GROUP BY ride_id
HAVING COUNT (ride_id)>1
```

Result: duplicates found

Row	ride_id	duplicates_count
1	5.05094E+15	2
2	2.70E+15	2
3	3.88E+15	2
4	1.71E+15	2

Further investigation of duplicates in the *ride_id column* :

```
WITH duplicates as (SELECT
    *,
    COUNT(ride_id) OVER (PARTITION BY ride_id) AS
duplicate_count
  FROM
    `project-1-asia.Rides.a_Full_Year`)
  SELECT*
  FROM duplicates
  WHERE duplicate_count >1
```

Result:

Although IDs are duplicated, the rest of data related to them is unique - * verified using conditional formatting, duplicates option in the started_at column

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	member_casua	duplicate_count
4260000000000000	classic_bike	2023-10-04 13:52:00.000000 UTC	2023-10-04 15:05:00.	DuSable Lake Shore Dr	13300	Streeter Dr & Grand A	13022	casual	2
4260000000000000	electric_bike	2023-10-12 13:25:00.000000 UTC	2023-10-12 13:30:00.	Southport Ave & Irving	TA1309000043	Southport Ave & Rosc	13071	casual	2
1710000000000000	electric_bike	2023-10-30 13:19:00.000000 UTC	2023-10-30 13:27:00.	Canal St & Monroe St	13056	Cityfront Plaza Dr & Pi	13427	member	2
1710000000000000	classic_bike	2023-06-27 16:04:00.000000 UTC	2023-06-27 16:05:00.	Wabash Ave & Adams S	KA1503000015	Wabash Ave & Adams	KA1503000015	member	2
5710000000000000	classic_bike	2023-10-25 17:10:00.000000 UTC	2023-10-25 17:35:00.	Wells St & Huron St	TA1306000012	Seeley Ave & Roscoe S	13144	member	3
5710000000000000	classic_bike	2023-10-05 22:00:00.000000 UTC	2023-10-05 22:09:00.	Elston Ave & Cortland S	TA1305000039	Lincoln Ave & Fullerto	TA1309000058	casual	3
5710000000000000	electric_bike	2023-10-03 20:30:00.000000 UTC	2023-10-03 20:58:00.	Wabash Ave & Grand A	TA1307000117	New St & Illinois St	TA1306000013	member	3
1410000000000000	electric_bike	2023-10-18 14:25:00.000000 UTC	2023-10-18 14:34:00.000000 UTC			Orleans St & Merchan	TA1305000022	member	2
1410000000000000	classic_bike	2023-10-07 15:42:00.000000 UTC	2023-10-07 16:05:00.	Milwaukee Ave & Gran	13033	Laflin St & Cullerton S	13307	member	2
7020000000000000	electric_bike	2023-10-07 22:52:00.000000 UTC	2023-10-07 22:59:00.	Sheffield Ave & Fullerto	TA1306000016			casual	2
7020000000000000	electric_bike	2023-10-31 08:27:00.000000 UTC	2023-10-31 08:34:00.	Wells St & Concord Ln	TA1308000050			member	2
1830000000000000	electric_bike	2023-03-27 12:25:00.000000 UTC	2023-03-27 13:26:00.	Milwaukee Ave & Fulle	428			casual	3
1830000000000000	electric_bike	2023-10-21 16:47:00.000000 UTC	2023-10-21 16:49:00.	Rush St & Cedar St	KA1504000133	Clark St & Elm St	TA1307000039	member	3
1830000000000000	classic_bike	2023-10-08 19:54:00.000000 UTC	2023-10-08 20:01:00.	Canal St & Jackson Blv	13138	Halsted St & Polk St	TA1307000121	member	3

Ride durations statistics

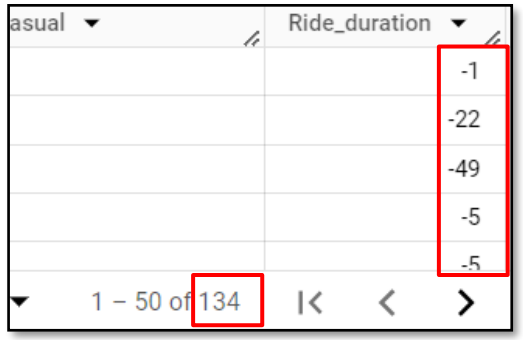
```
SELECT
  member_casual,
  AVG (TIMESTAMP_DIFF(ended_at, started_at, HOUR)) as Avg_Ride_duration,
  SUM (TIMESTAMP_DIFF(ended_at, started_at, HOUR)) as SUM_Ride_duration,
  MIN (TIMESTAMP_DIFF(ended_at, started_at, HOUR)) as MIN_Ride_duration,
  MAX (TIMESTAMP_DIFF(ended_at, started_at, HOUR)) as MAX_Ride_duration,
FROM `project-1-asia.Rides.a_Full_Year`
GROUP BY member_casual
```

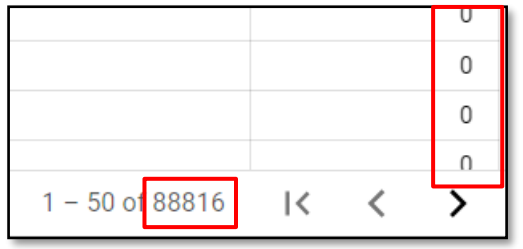
Result:
Some ride durations are below or 0 – which is incorrect

Row	member_casual	Avg_Ride_duration	SUM_Ride_duration	MIN_Ride_duration	MAX_Ride_duration
1	member	0.022677369179...	83015	-277	26
2	casual	0.218448711840...	449825	-277	1641

Query exploring the numer of rides below and equal to 0:

```
WITH Ride_duration_table AS (
  SELECT ride_id, rideable_type, started_at, ended_at,member_casual,
TIMESTAMP_DIFF(ended_at, started_at, MINUTE) as Ride_duration
FROM `project-1-asia.Rides.a_Full_Year`
)
SELECT ride_id, rideable_type, started_at, ended_at,member_casual,
Ride_duration
FROM Ride_duration_table
WHERE Ride_duration <0
```





The above query with a changed WHERE clause:

```
WHERE Ride_duration =0
```

Result:

134 rows show < 0 ride duration

88816 rows of ride durations are = 0
They can be filtered out as it's not a significant percentage of the total data of Ride_durations

Total rows	Ride duration<0	Ride duration=0
5719877	134	88816
100%	0.002%	1.553%

New table with new columns and information filtered out:

- Ride_duration > 0 only
- Stations' names and IDs columns are dropped
- Month and day of week extracted from the date

```
WITH Ride_duration_table AS (
  SELECT
ride_id, rideable_type, started_at, ended_at,member_casual,
TIMESTAMP_DIFF(ended_at, started_at, MINUTE) as Ride_duration
FROM `project-1-asia.Rides.a_Full_Year`
)
SELECT
ride_id, rideable_type, started_at, ended_at,member_casual, Ride_duration,
EXTRACT(MONTH from started_at) as MONTH_start,
EXTRACT(MONTH from ended_at) as MONTH_end,
FORMAT_DATE('%A', started_at) as day_of_week
FROM Ride_duration_table
WHERE Ride_duration >0
```

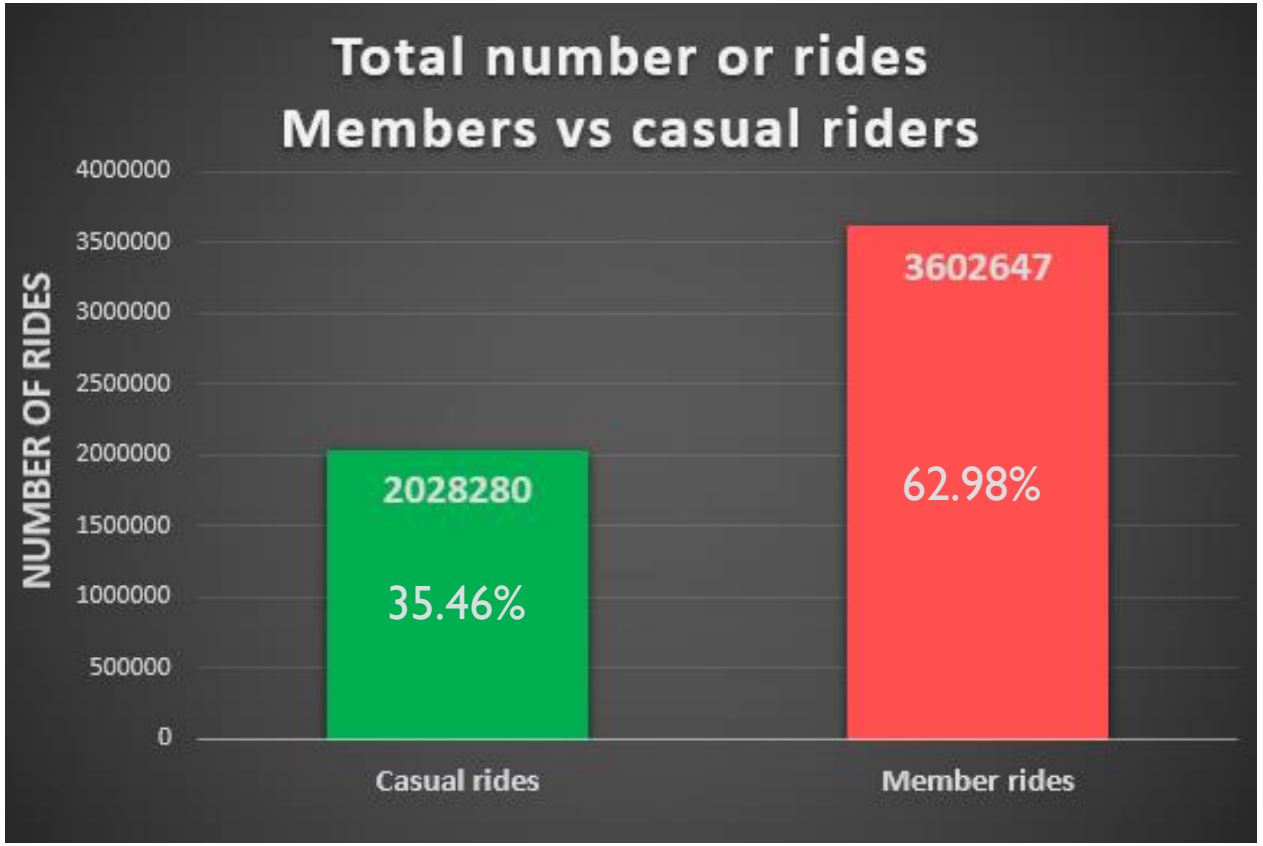
Row	ride_id	rideable_type	started_at	ended_at	member_casual	Ride_duration	MONTH_start	MONTH_end	day_of_week
1	FA8B08EE6C4DEA...	electric_bi...	2023-01-06 07:21:00 U...	2023-01-06 07:27:00 UTC	member	6	1	1	Friday
2	38C5DFCFFFBA36...	electric_bi...	2023-01-06 22:43:00 U...	2023-01-06 22:50:00 UTC	casual	7	1	1	Friday
3	F3A34B3A0B577546	electric_bi...	2023-01-27 13:22:00 U...	2023-01-27 13:42:00 UTC	casual	20	1	1	Friday

```sql
SELECT
member_casual,
COUNT
(member_casual) as user_count
  FROM `project-1-
asia.Rides.Full_Year_sorted`
GROUP BY
  member_casual
```

Result:

Row	member_casual	user_count
1	member	3602647
2	casual	2028280

Excel chart:

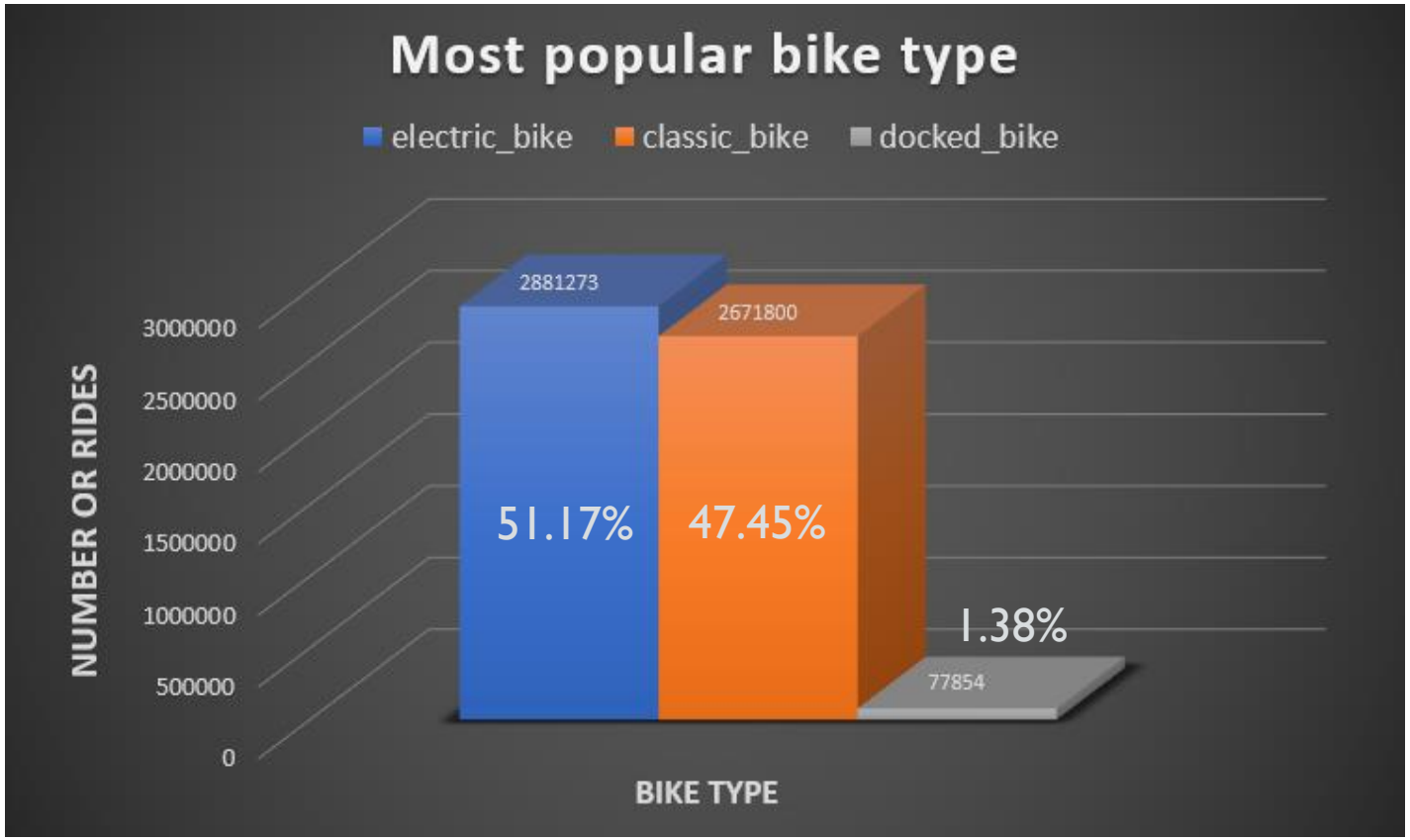


```sql
SELECT
rideable_type,
COUNT(member_casual) as
User_count_per_bike_type
FROM `project-1-
asia.Rides.Full_Year_sorted`
GROUP BY
rideable_type
```

Result:

rideable_type	User_count_per_bike
electric_bike	2881273
classic_bike	2671800
docked_bike	77854

Excel chart:

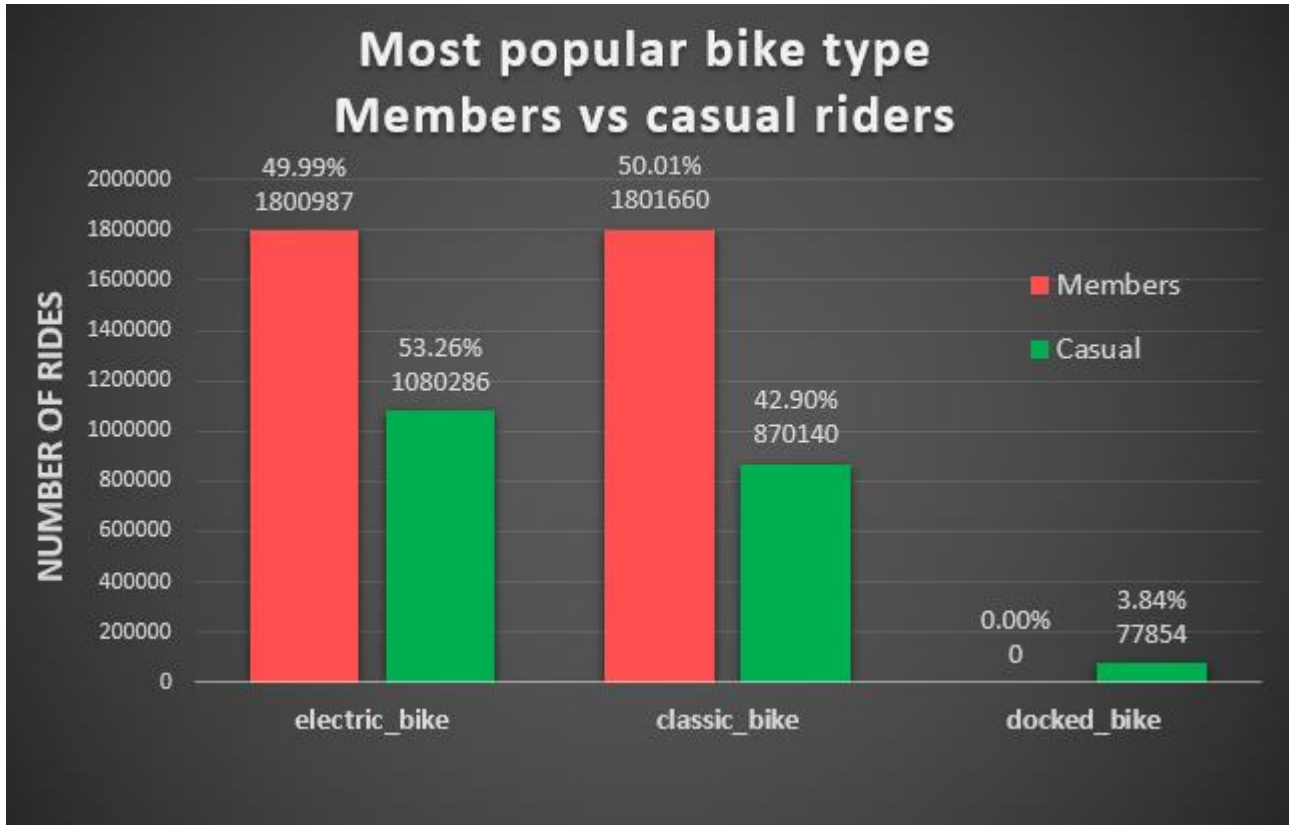


```sql
SELECT
  rideable_type,
  COUNTIF(member_casual = 'member')
as Member_count,
  COUNTIF(member_casual = 'casual')
as Casual_count
FROM
  `project-1-
asia.Rides.Full_Year_sorted`
GROUP BY
  rideable_type;
```

Result:

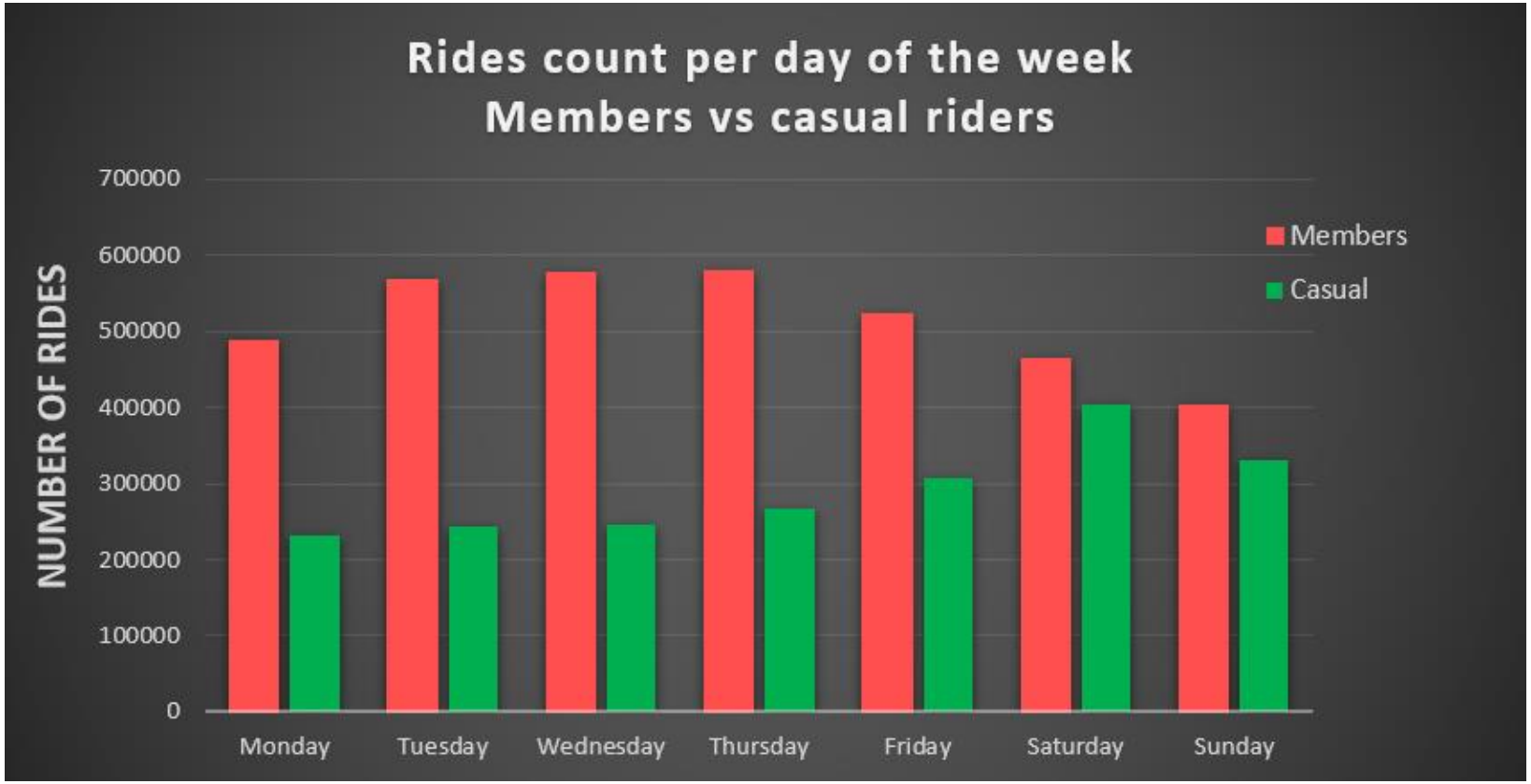
rideable_type	Member_count	Casual_count
electric_bike	1800987	1080286
classic_bike	1801660	870140
docked_bike	0	77854

Excel chart:



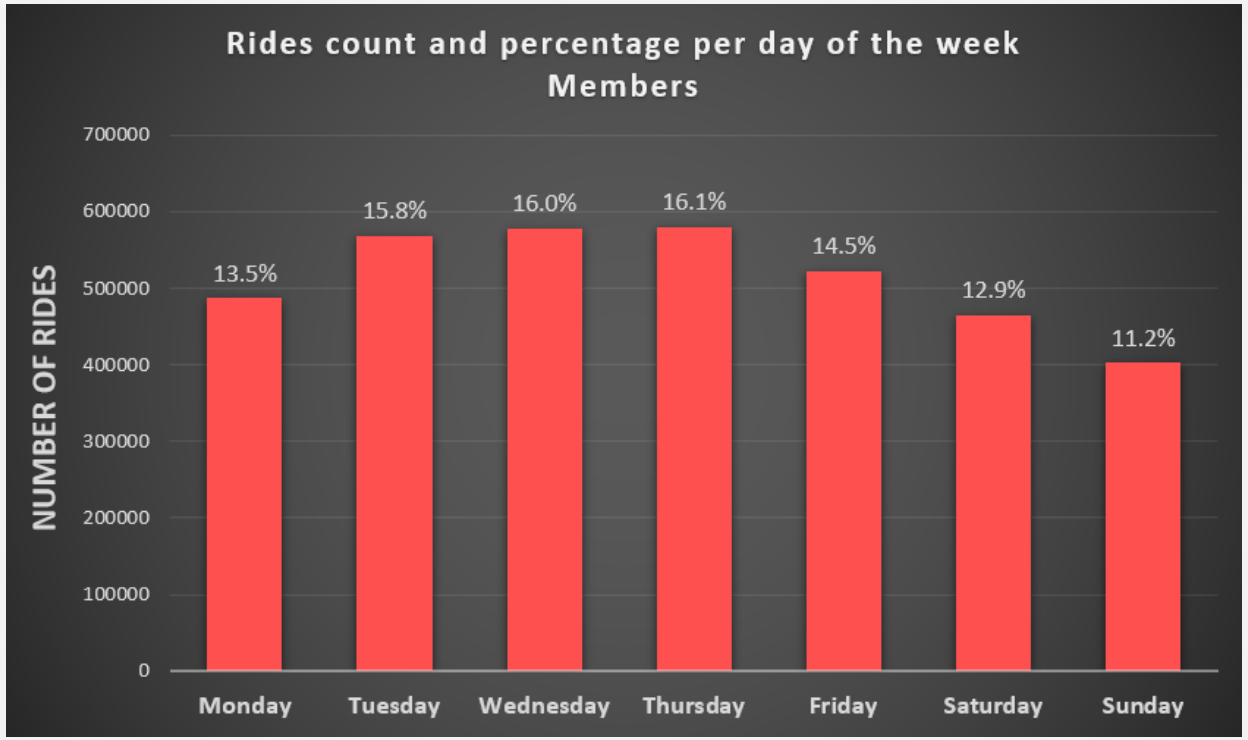
```sql
SELECT
  day_of_week,
  COUNTIF(member_casual = 'member')
as Member_count,
  COUNTIF(member_casual = 'casual')
as Casual_count
FROM
  `project-1-
asia.Rides.Full_Year_sorted`
GROUP BY
  day_of_week
ORDER BY
  CASE day_of_week
    WHEN 'Monday' THEN 1
    WHEN 'Tuesday' THEN 2
    WHEN 'Wednesday' THEN 3
    WHEN 'Thursday' THEN 4
    WHEN 'Friday' THEN 5
    WHEN 'Saturday' THEN 6
    WHEN 'Sunday' THEN 7
    ELSE 999  -- Handle potential
unexpected values
  END;
```

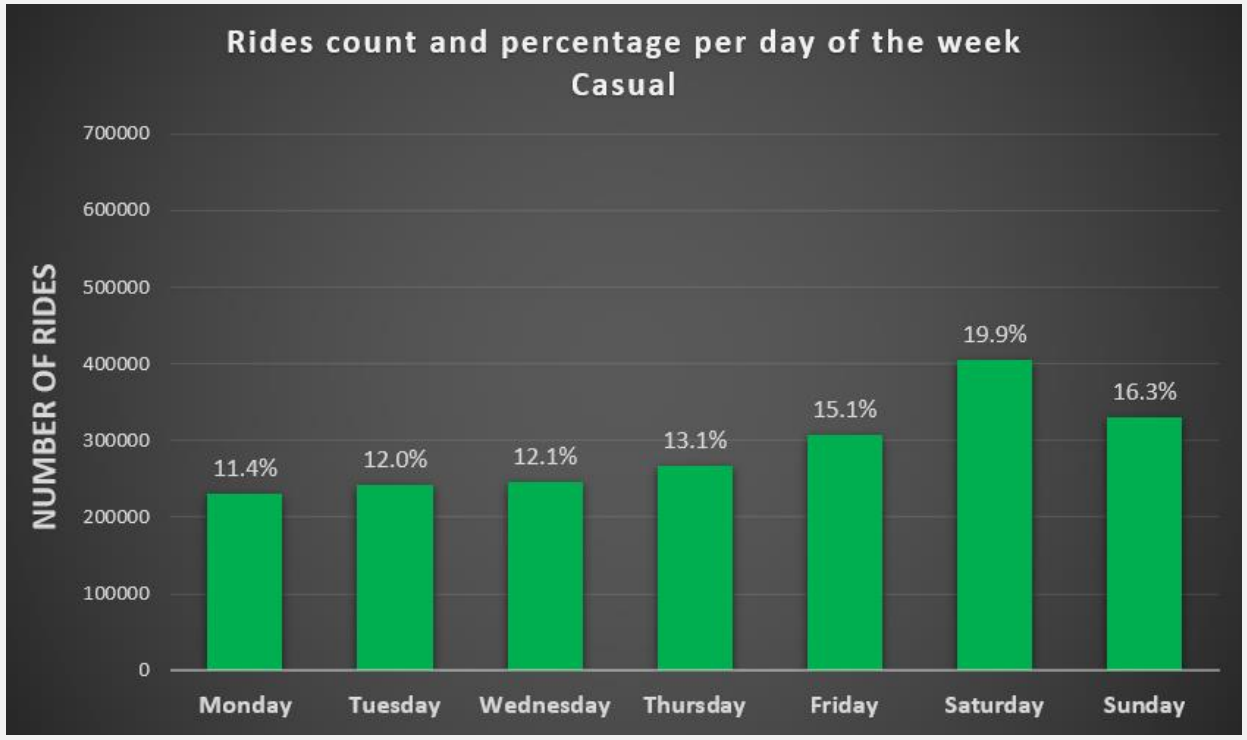
Excel chart:



Weekdays are more popular amongst members and weekend amongst casual riders

Previous chart split between Members and Casual riders,
and showing percentages comparison





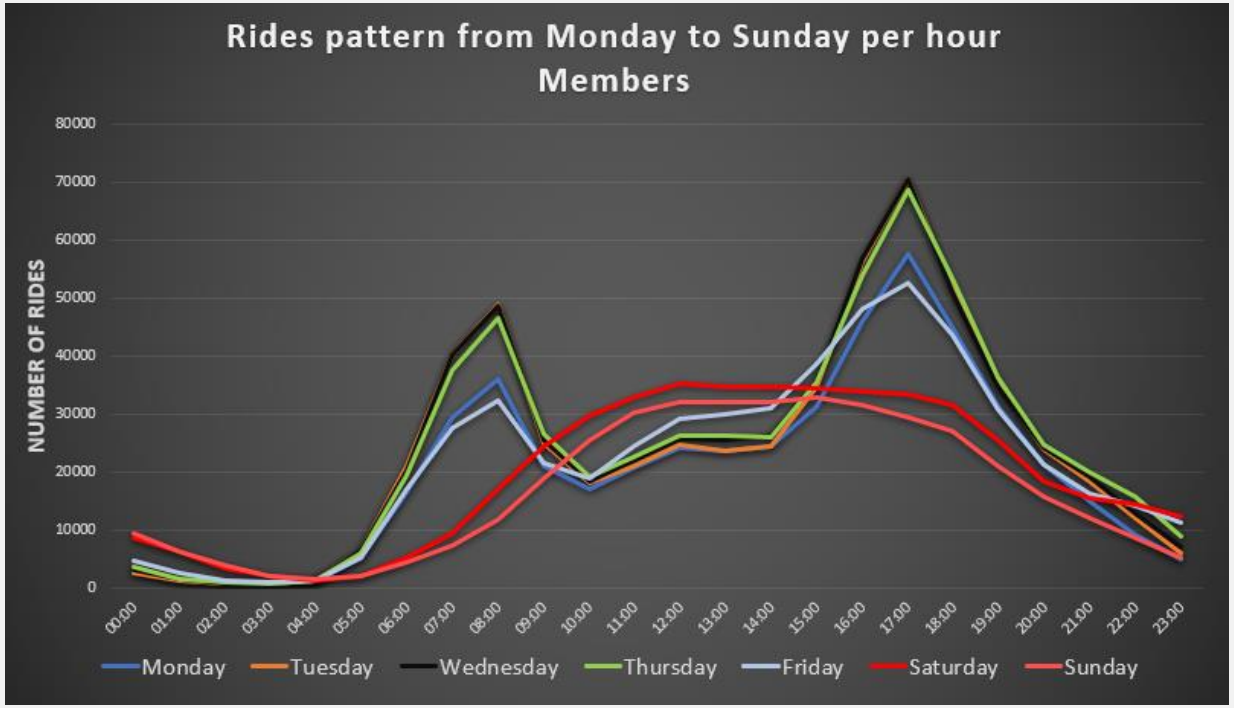
Hourly use per day of the week

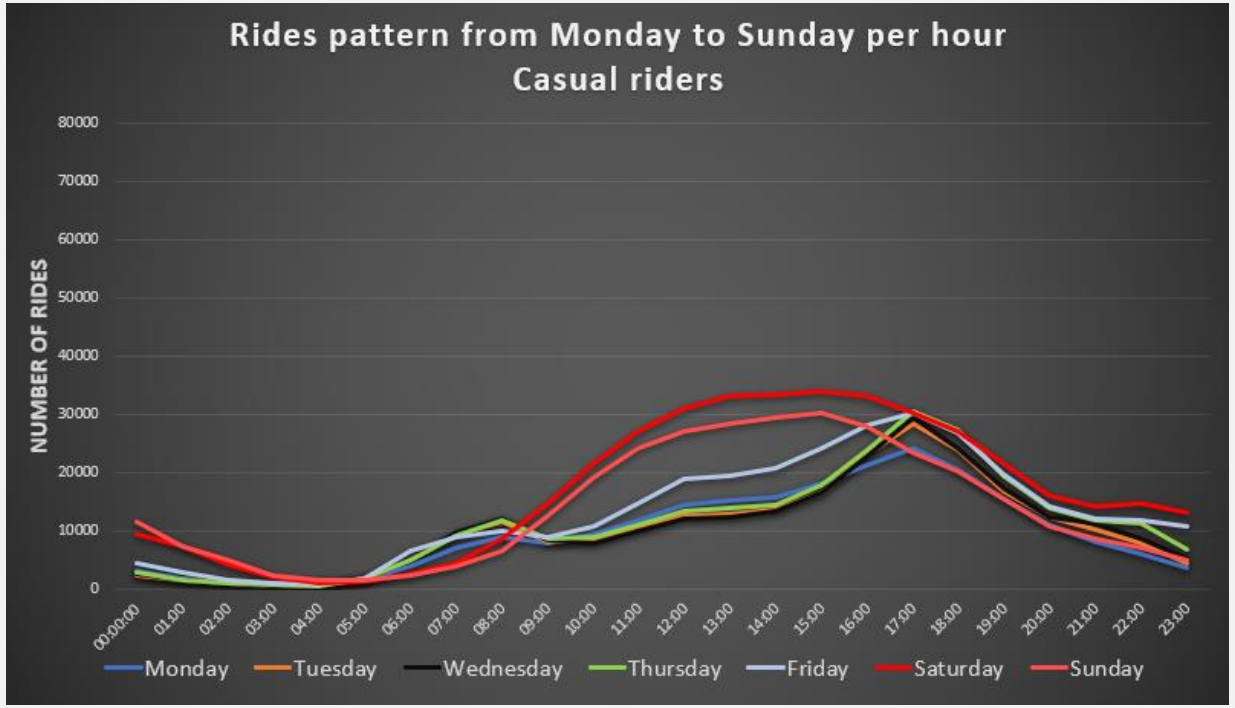
```
WITH truncated as( SELECT
  started_at,member_casual,day_of_week,
  DATE_TRUNC(started_at, HOUR) AS
truncated_hour
  FROM
  `project-1-asia.Rides.Full_Year_sorted` )
SELECT
  EXTRACT(TIME FROM truncated_hour) AS
Hourly_range,
  day_of_week,member_casual,
  COUNT(*) AS Count_of_Members_Casual
FROM
 truncated
GROUP BY
  day_of_week,
  Hourly_range,
  member_casual
ORDER BY
CASE day_of_week
    WHEN 'Monday' THEN 1
    WHEN 'Tuesday' THEN 2
    WHEN 'Wednesday' THEN 3
    WHEN 'Thursday' THEN 4
    WHEN 'Friday' THEN 5
    WHEN 'Saturday' THEN 6
    WHEN 'Sunday' THEN 7
    ELSE 999
  END,
  Hourly_range,
  member_casual
```

Row	Hourly_range	day_of_week	member_casual	Count_of_Members
1	00:00:00	Monday	casual	3212
2	00:00:00	Monday	member	3058
3	01:00:00	Monday	casual	1874
4	01:00:00	Monday	member	1524
5	02:00:00	Monday	casual	1052

Excel table

Time rounde	Colum	user type	Monday	Tuesday	Wednesd	Thursday	Friday	Saturday	Sunday
00:00:00		casual	3212	2408	2612	2841	4293	9396	11497
01:00		casual	1874	1400	1450	1581	2702	7190	7276
02:00		casual	1052	856	910	1042	1405	4092	4831
03:00		casual	769	528	582	592	880	2097	2365
04:00		casual	725	604	632	571	716	1064	1561
05:00		casual	1632	1720	1772	1657	1737	1322	1431
06:00		casual	3889	4903	4806	4801	6551	2441	2342
07:00		casual	6975	9344	9579	9095	8937	4426	3911
08:00		casual	8946	11609	12145	11868	9807	8725	6637
09:00		casual	7919	8162	8225	8596	8952	14737	12436
10:00		casual	9426	8225	7774	8811	10717	21460	19036
11:00		casual	11850	10539	10017	10933	14611	26945	24116
12:00		casual	14422	12537	12000	13234	18736	30889	27145
13:00		casual	15148	12688	12349	13775	19313	33049	28310
14:00		casual	15640	13674	13401	14449	20643	33244	29497
15:00		casual	18033	16593	16421	17838	24137	33780	30116
16:00		casual	21158	23098	22680	23961	27990	33048	27838
17:00		casual	24251	28285	29467	30447	30064	30284	23442
18:00		casual	20394	23805	24036	27331	26673	27058	20272
19:00		casual	15457	16747	17263	19494	19549	21425	15335
20:00		casual	11104	12255	12376	13863	14106	16098	10717
21:00		casual	7979	10273	11112	11805	12002	14141	8712
22:00		casual	5934	7923	8592	11269	11880	14617	7061
23:00		casual	3611	4336	5160	6666	10829	13026	4819





Both casual riders and members use bikes differently on weekdays than on the weekends.
Members use bikes for work hence different use patterns during weekdays between members and casual riders.

Bike type popularity data

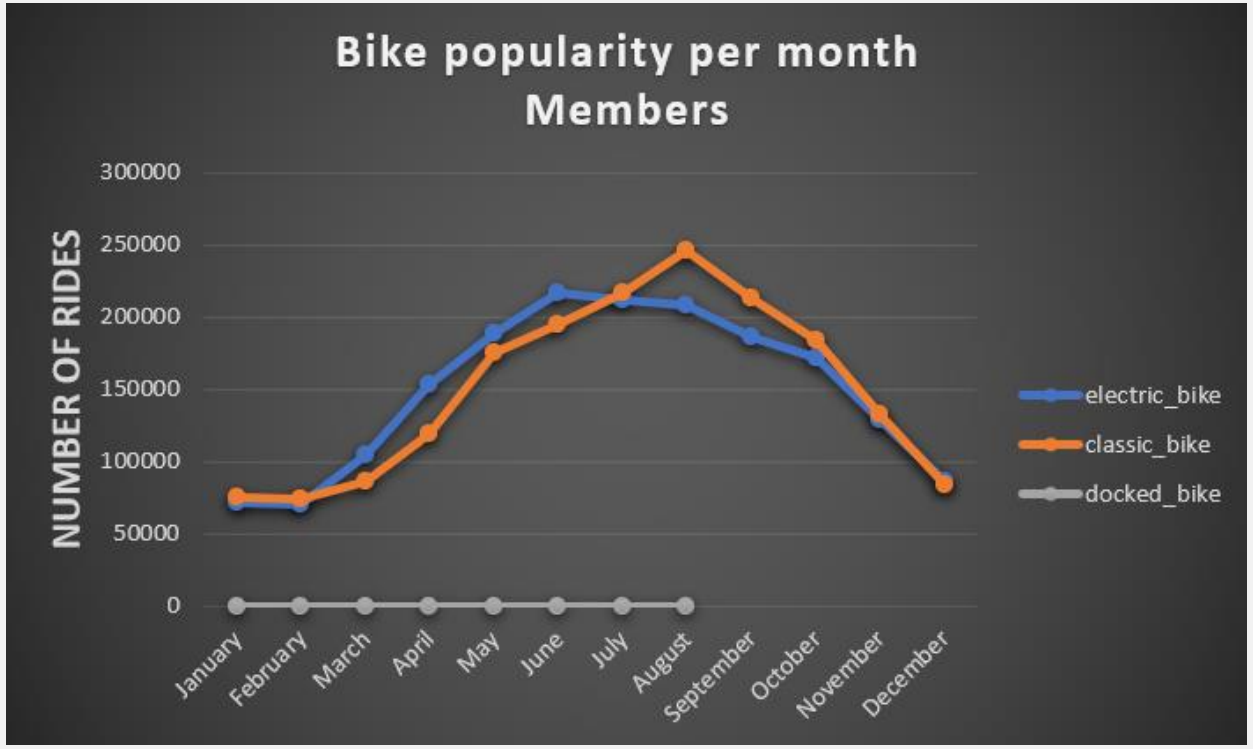
```sql
SELECT
  day_of_week, rideable_type,MONTH_start,
  COUNTIF(member_casual = 'member') as
Member_count,
  COUNTIF(member_casual = 'casual') as
Casual_count
FROM
  `project-1-asia.Rides.Full_Year_sorted`

GROUP BY
  day_of_week,
  rideable_type,
  MONTH_start

ORDER BY
  MONTH_start,  -- Order by month ascending

  CASE day_of_week
    WHEN 'Monday' THEN 1
    WHEN 'Tuesday' THEN 2
    WHEN 'Wednesday' THEN 3
    WHEN 'Thursday' THEN 4
    WHEN 'Friday' THEN 5
    WHEN 'Saturday' THEN 6
    WHEN 'Sunday' THEN 7
    ELSE 999  -- Handle potential unexpected
values
  END,
  rideable_type;  -- Order by rideable_type
```

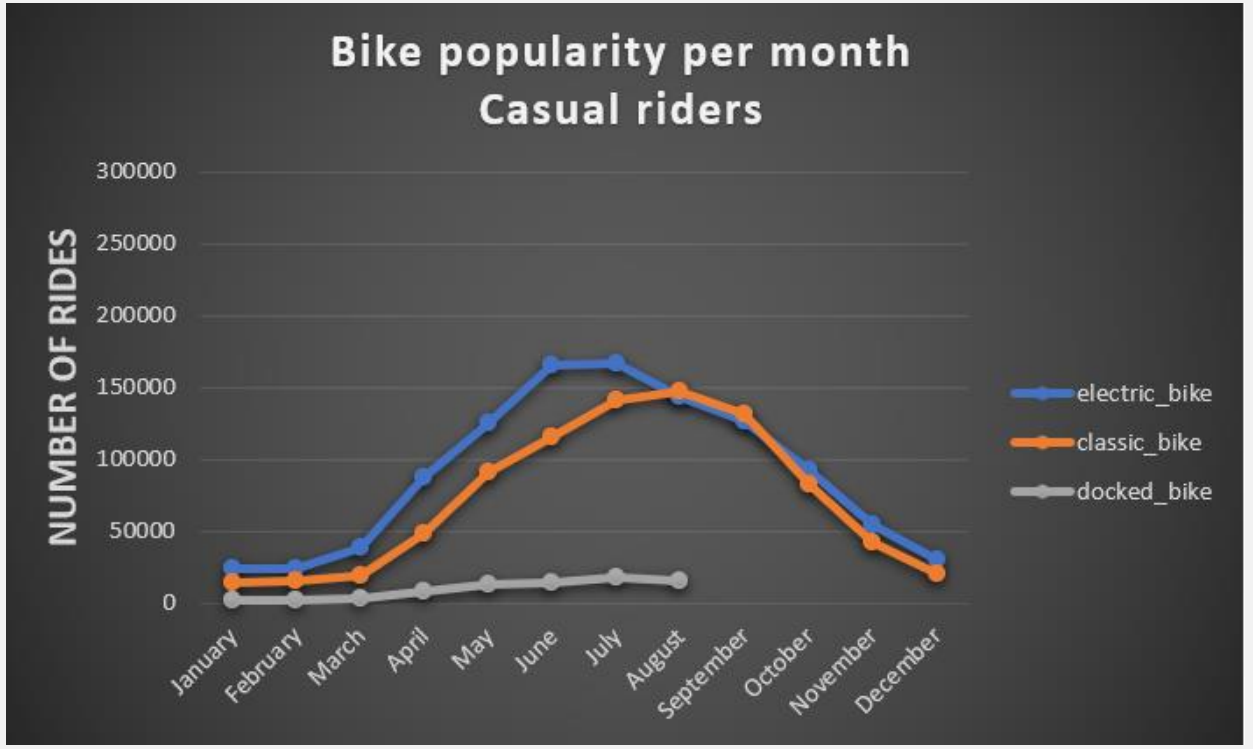
Row	day_of_week	rideable_type	MONTH_start	Member_count	Casual_count
35	Friday	docked_bike	2	0	164
36	Friday	electric_bike	2	8129	2510

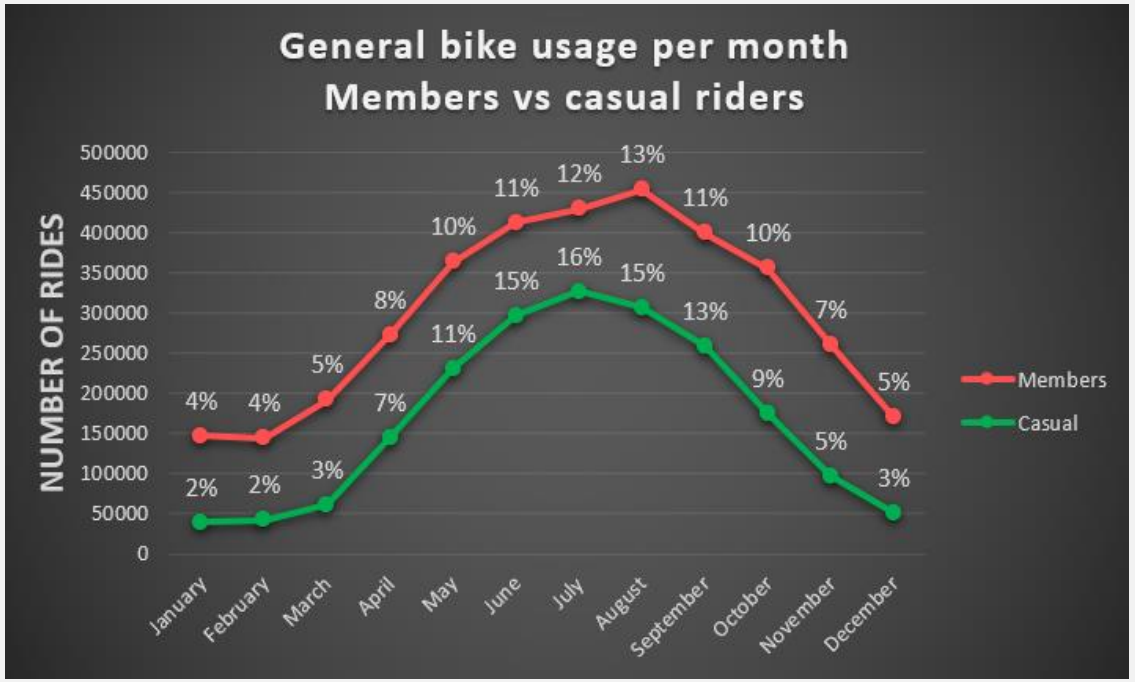
Excel pivot table

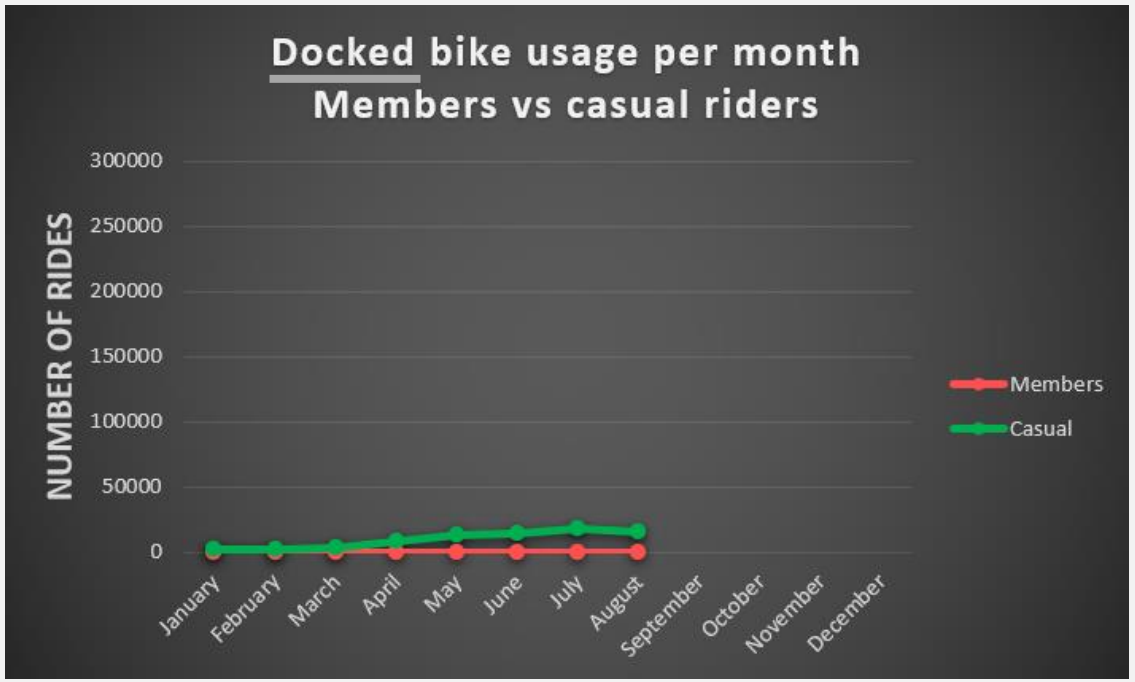
Sum of User_count_per_bike_type	rideable_type		
MONTH_start	electric_bike	classic_bike	docked_bike
January	95267	89492	1731
February	95361	89119	2183
March	143725	106182	3008
April	240352	168167	8834
May	313715	267366	13023
June	383188	310436	14870
July	378981	358645	18329
August	351627	393468	15876
September	313558	344366	
October	264320	266072	
November	183540	174688	
December	117639	103799	

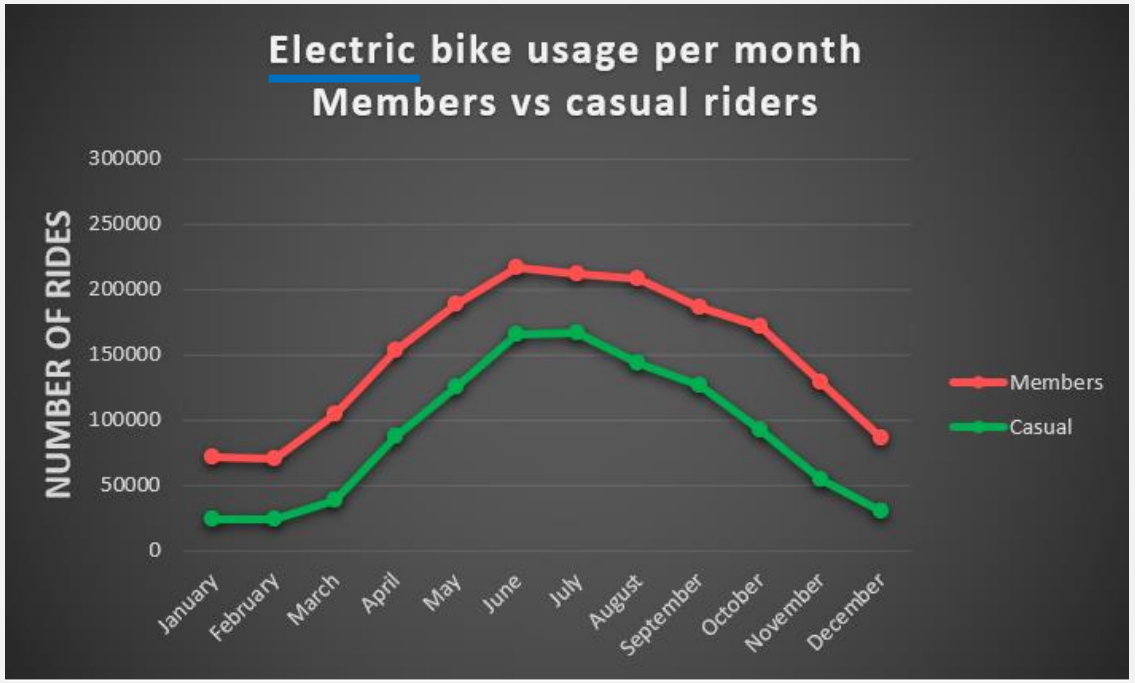
Excel chart (docked bike has no data for the last quarter possibly discontinued)

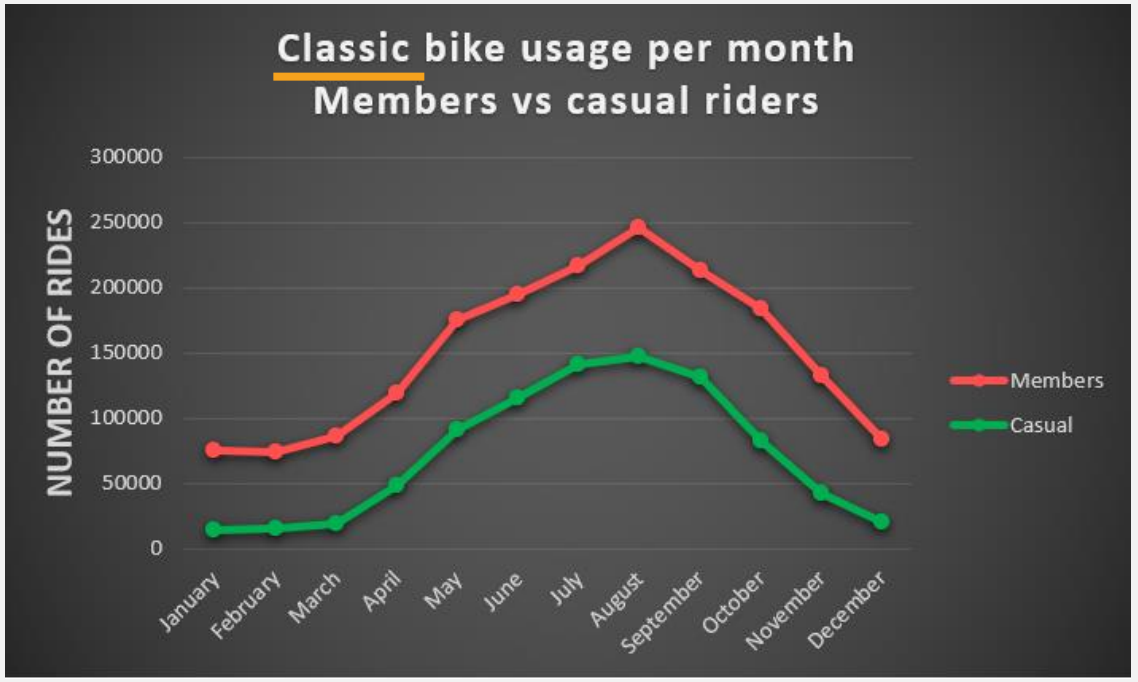





General bike usage per month
Members vs casual riders
NUMBER OF RIDES
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
4% 4% 5% 8% 10% 11% 12% 13% 11% 10% 7% 5%
2% 2% 3% 7% 11% 15% 16% 15% 13% 9% 5% 3%
January February March April May June July August September October November December
Members
Casual


Docked bike usage per month
Members vs casual riders
NUMBER OF RIDES
300000
250000
200000
150000
100000
50000
0
January February March April May June July August September October November December
Members
Casual


Electric bike usage per month
Members vs casual riders
NUMBER OF RIDES
300000
250000
200000
150000
100000
50000
0
January February March April May June July August September October November December
Members
Casual

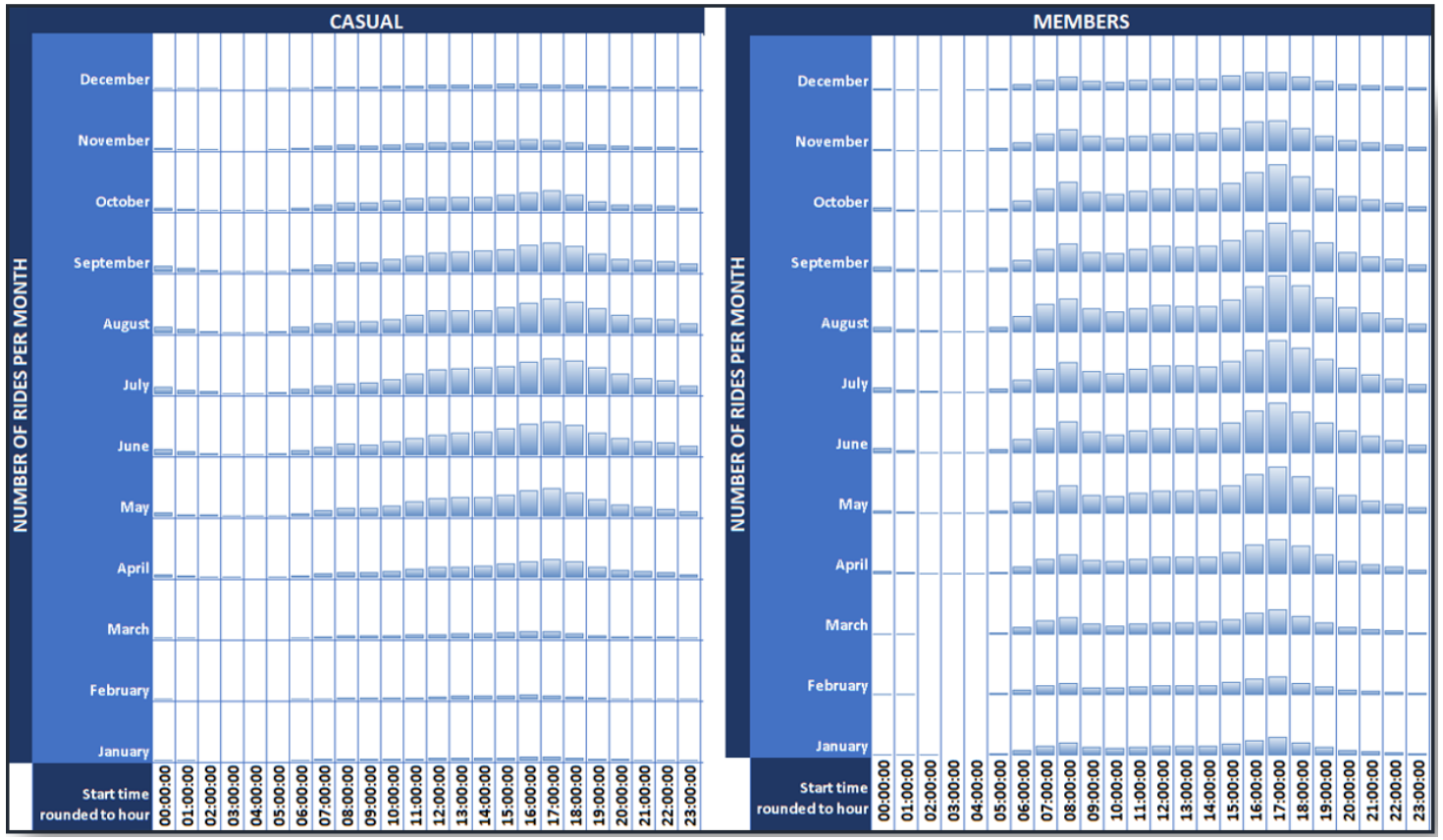

Classic bike usage per month
Members vs casual riders
NUMBER OF RIDES
300000
250000
200000
150000
100000
50000
0
January February March April May June July August September October November December
Members
Casual

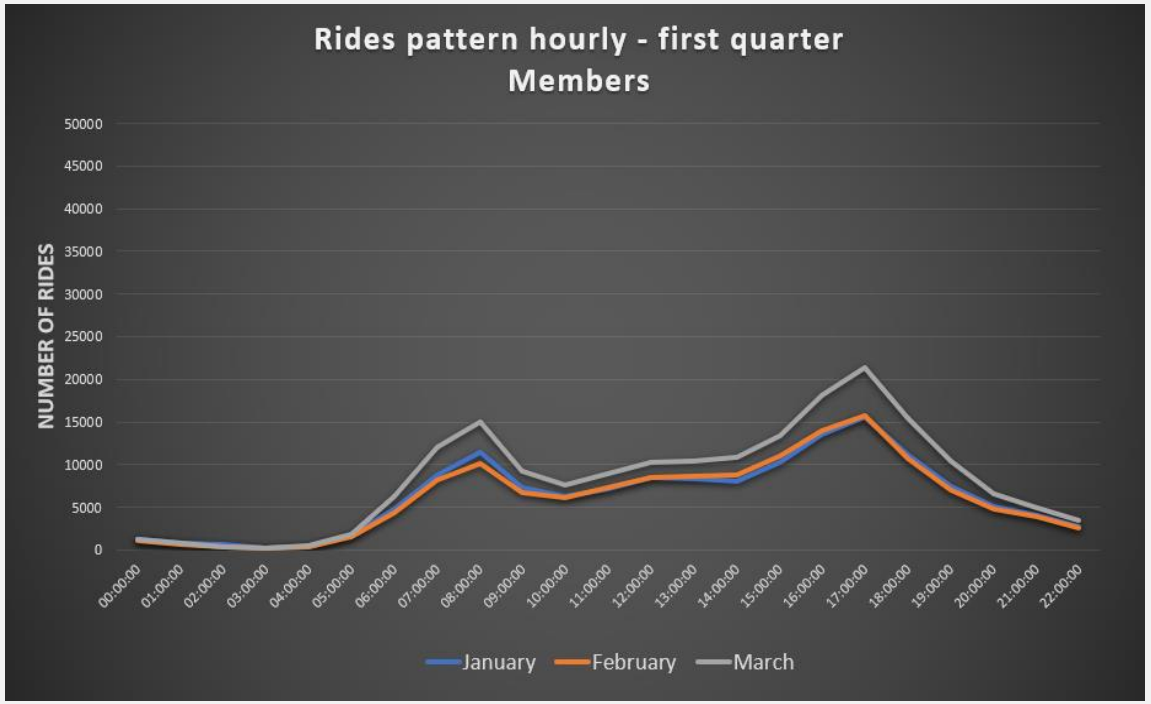
Most common ride times

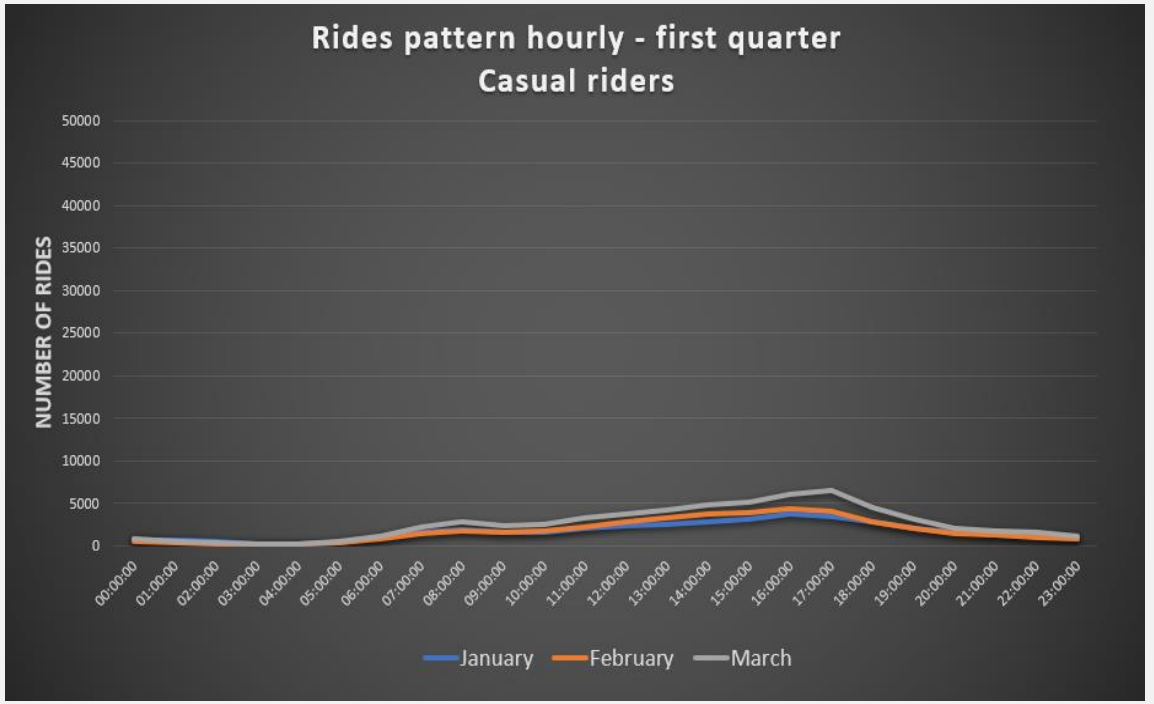
```sql
WITH truncated as( SELECT
  started_at,member_casual,
  DATE_TRUNC(started_at, HOUR) AS
truncated_hour
  FROM
  `project-1-asia.Rides.Full_Year_sorted` )

SELECT
  EXTRACT(TIME FROM truncated_hour) AS
Hourly_range,
  member_casual,
  COUNT(*) AS Count_of_Members_Casual
FROM
 truncated
GROUP BY
  Hourly_range,
  member_casual;
```

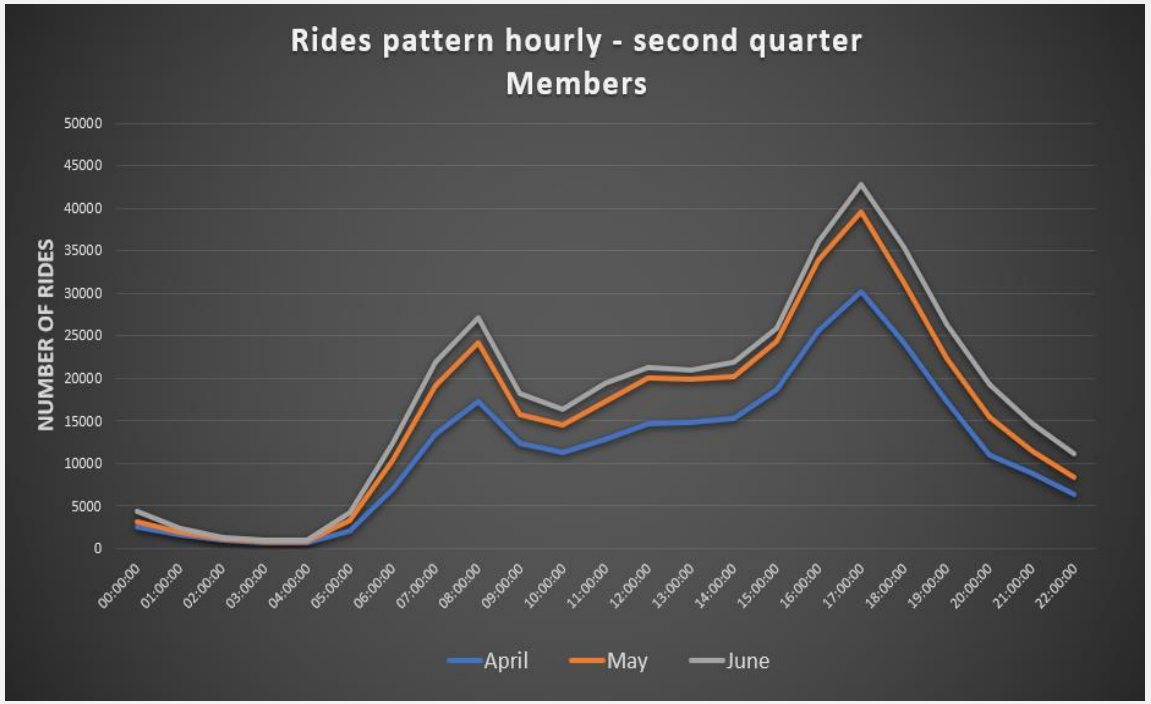
Row	Hourly_range	member_casual	Count_of_Members
1	00:00:00	member	34899
2	00:00:00	casual	36259
3	01:00:00	casual	23473
4	01:00:00	member	20779
5	02:00:00	member	12032
6	02:00:00	casual	14188
7	03:00:00	member	7824

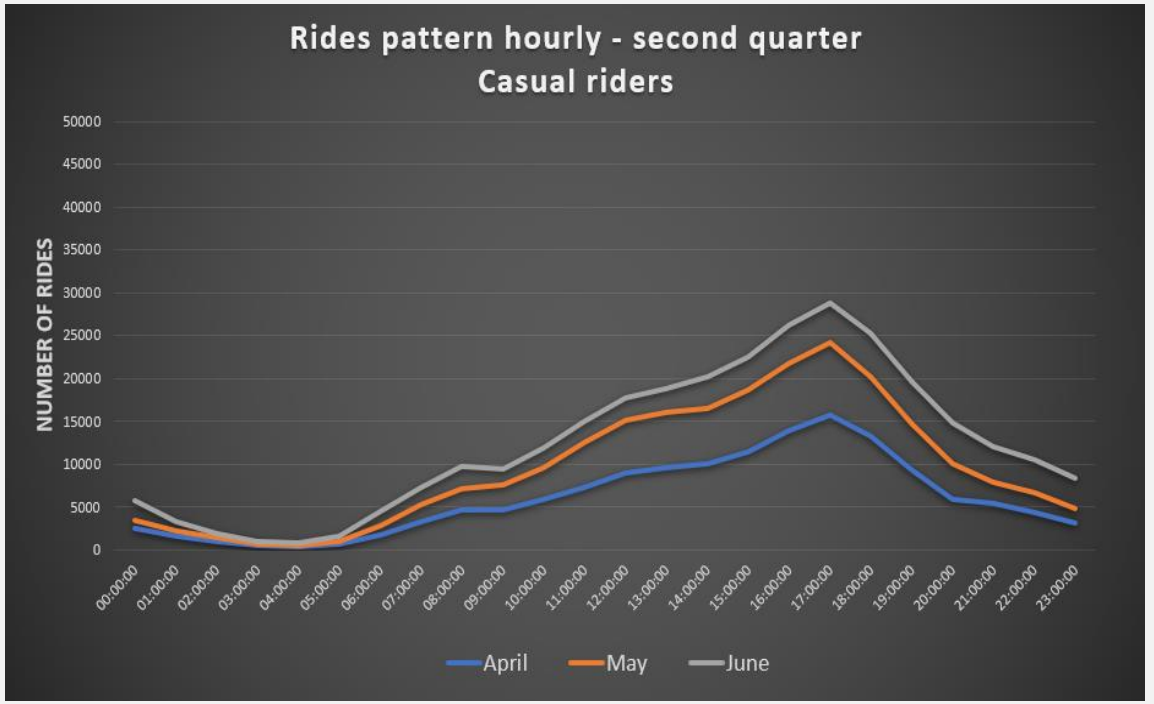
Excel table with conditional formatting data bars for an overview

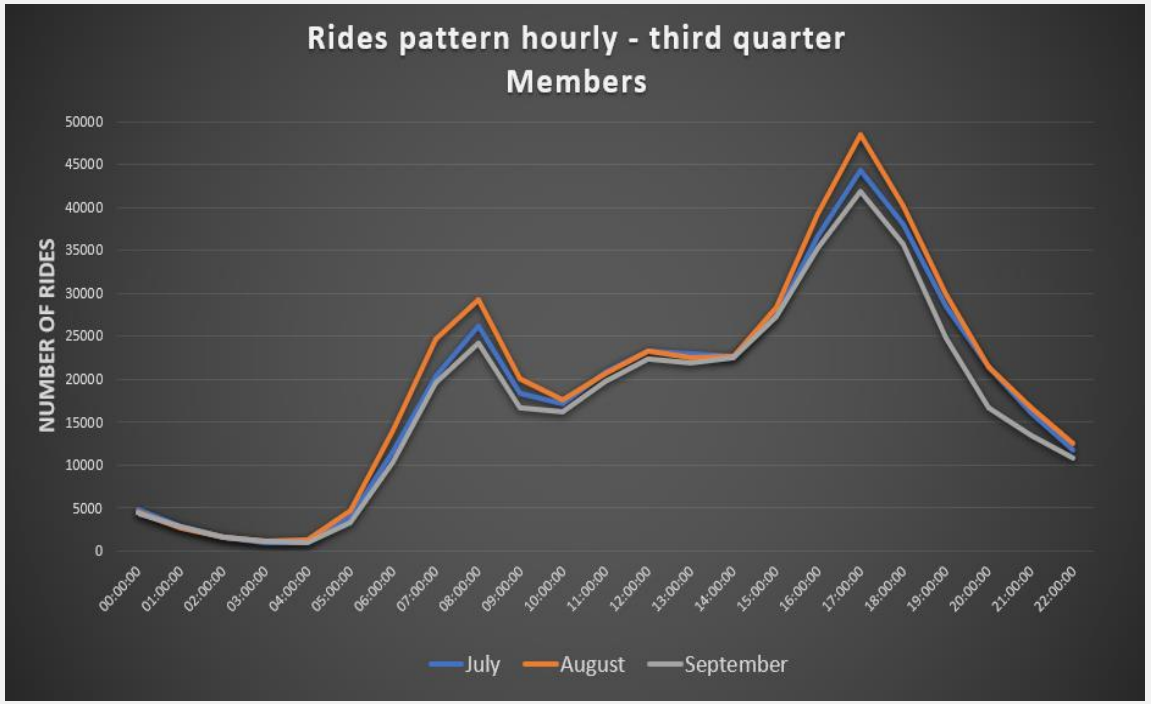


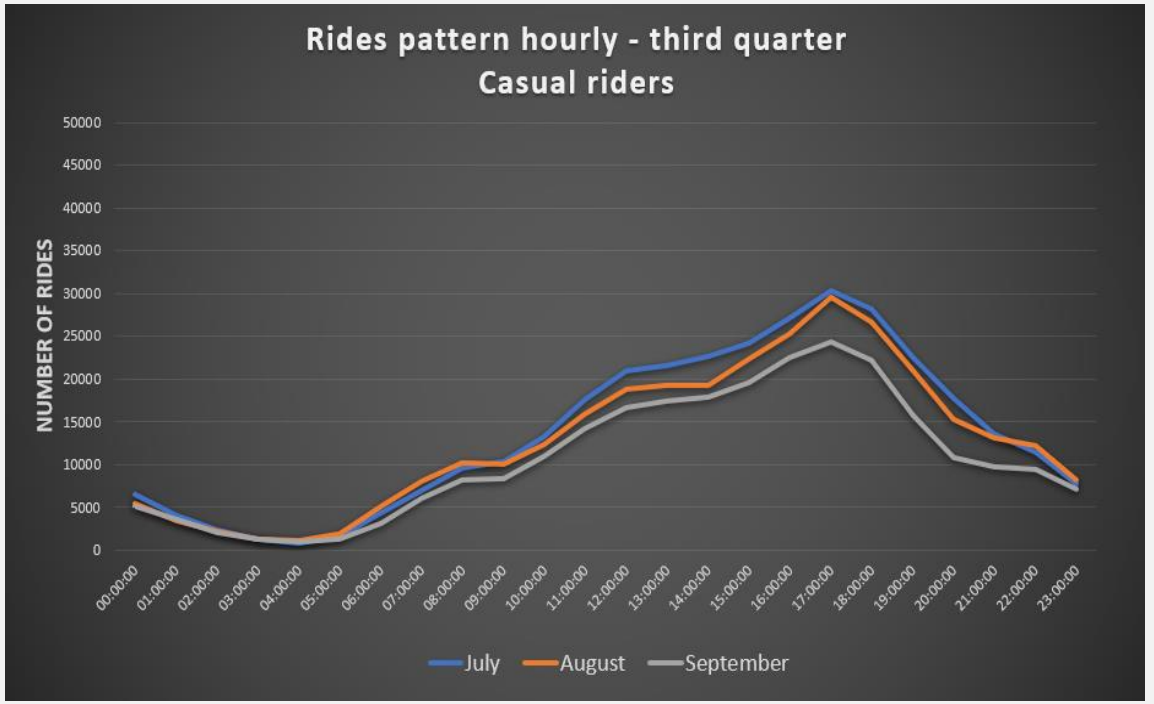

Rides pattern hourly - first quarter
Members
NUMBER OF RIDES
January
February
March

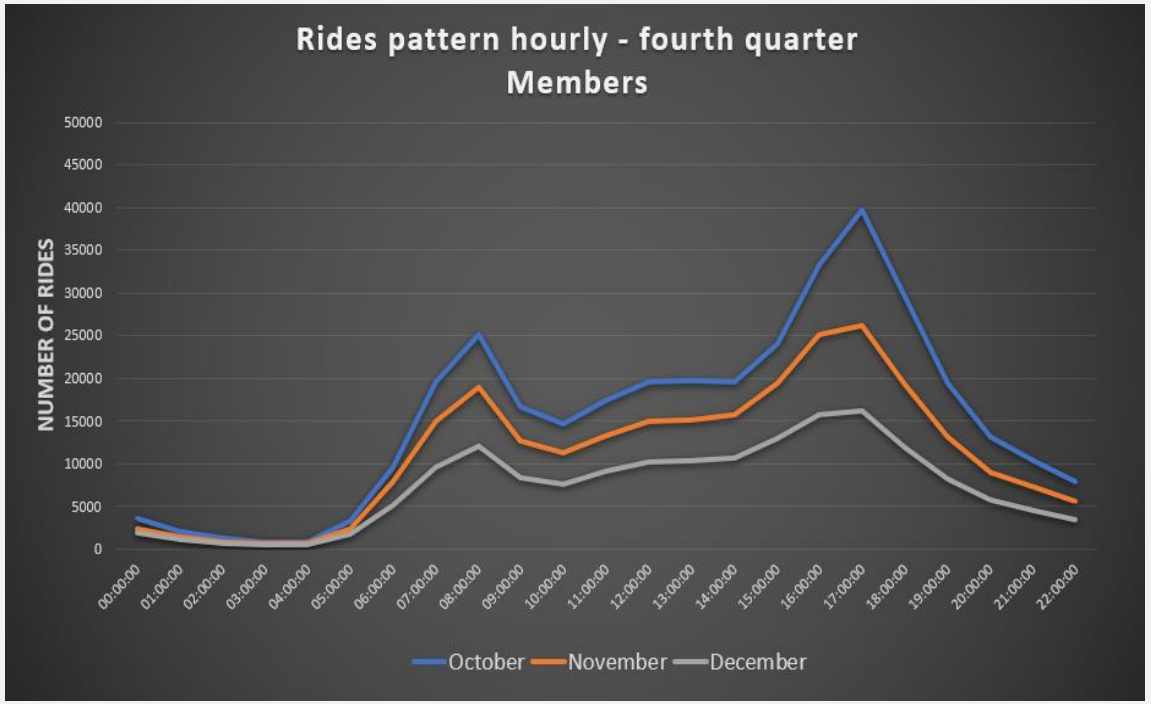

Rides pattern hourly - first quarter
Casual riders
NUMBER OF RIDES
January
February
March

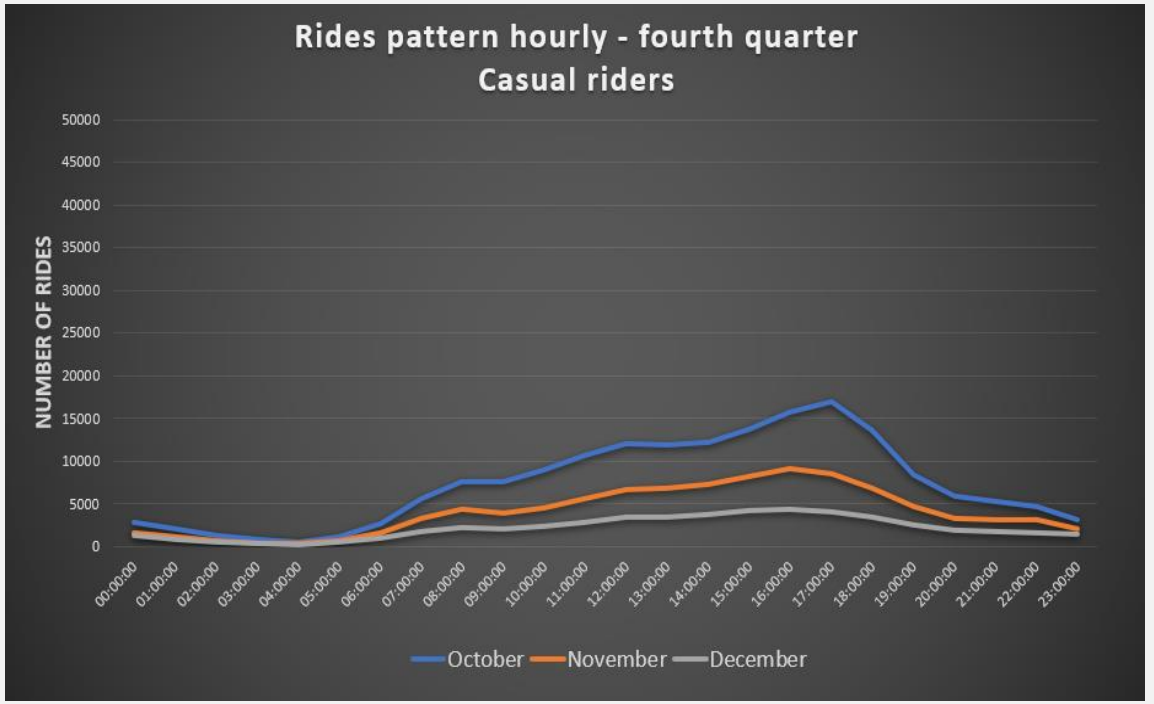

Rides pattern hourly - second quarter
Members
NUMBER OF RIDES
April
May
June


Rides pattern hourly - second quarter
Casual riders
NUMBER OF RIDES
April
May
June


Rides pattern hourly - third quarter
Members
NUMBER OF RIDES
July
August
September


Rides pattern hourly - third quarter
Casual riders
NUMBER OF RIDES
July
August
September

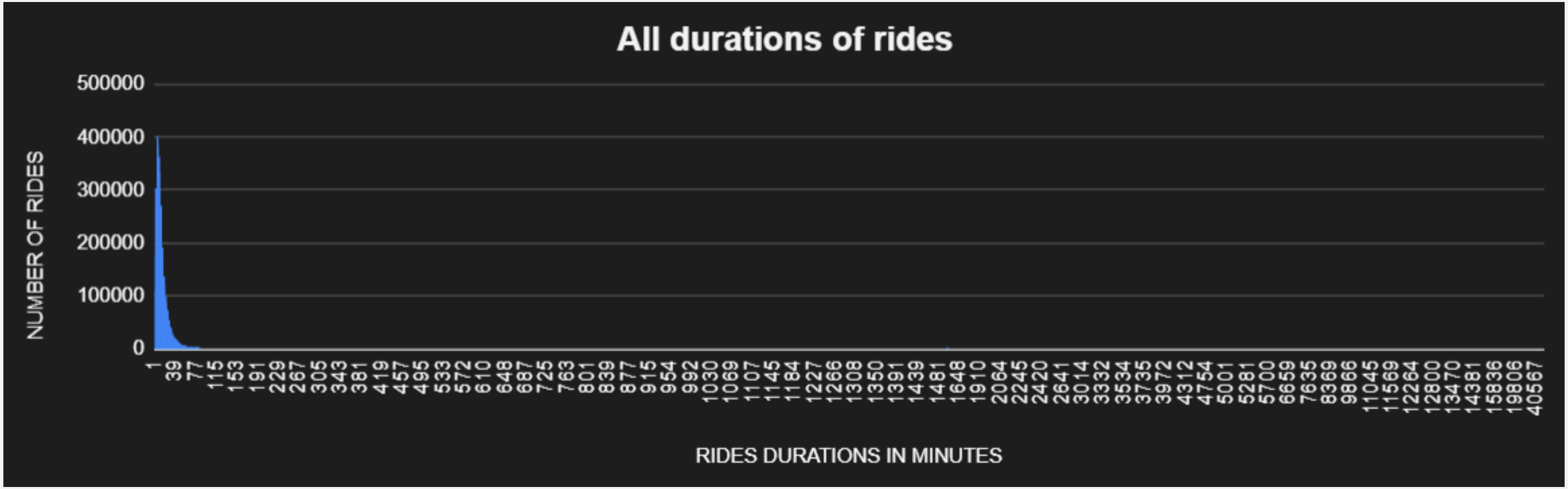

Rides pattern hourly - fourth quarter
Members
NUMBER OF RIDES
October
November
December


Rides pattern hourly - fourth quarter
Casual riders
NUMBER OF RIDES
October
November
December

Most popular ride durations

```
SELECT ride_duration,
FROM `project-1-asia.Rides.Full_Year_sorted`
```

Data exploration from BigQuery directly in Google Sheets



Number of rides longer than 26 hours

Number of rides longer than 26 hours:

```
WITH  Count_long_rides as (SELECT
 ride_duration, member_casual,
 COUNT(Ride_duration) as number_of_rides
FROM
`project-1-asia.Rides.Full_Year_sorted`
WHERE ride_duration>1560
GROUP BY Ride_duration, member_casual
ORDER BY
ride_duration ASC,
member_casual ASC )

SELECT member_casual,
COUNT(number_of_rides) as
Number_of_rides_over26h
FROM  Count_long_rides
GROUP BY
member_casual
```

There are 1091 very long-duration rides, which is uncommon for a bike share. All of these are only casual rides. The data suggest that users keep bikes for days or even weeks. It would have to be investigated further why there are such long-duration rides in the data set. Since the number of these is only a small percentage out of 5mln row data filtering them out will not cause a significant data loss.

member_casual	Number_of_rides_over26h
casual	1091

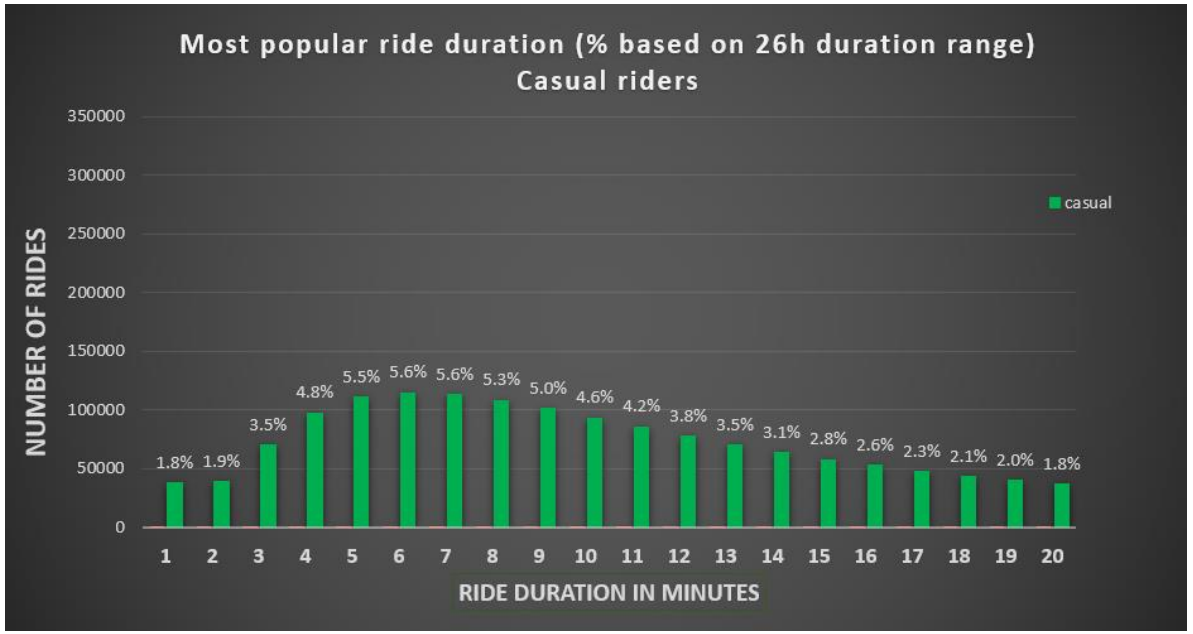
Longest ride duration minutes	98489
Longest ride duration hours	1641.483
Number of rides longer than 26 hours	1091

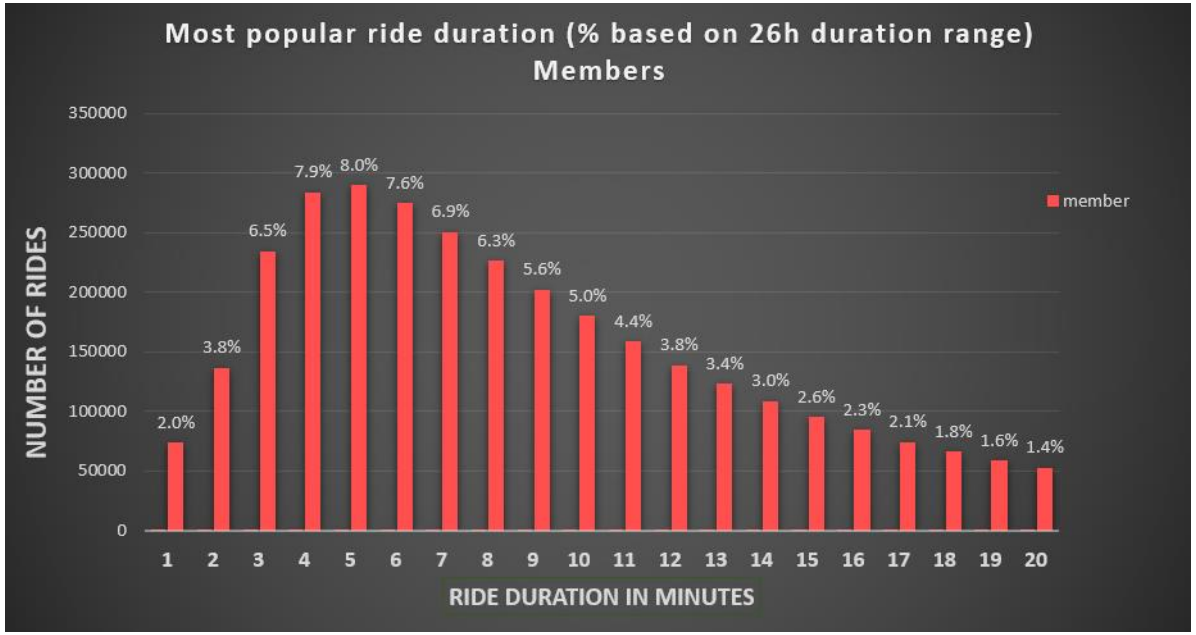
More detailed visualisation of most common ride durations, Excel chart

```
SELECT
 ride_duration, member_casual,
 COUNT(Ride_duration) as number_of_rides
FROM
`project-1-asia.Rides.Full_Year_sorted`
GROUP BY Ride_duration, member_casual
ORDER BY
ride_duration ASC,
member_casual ASC
```

Row	ride_duration	member_casual	number_of_rides
1	1	casual	37398
2	1	member	73489
3	2	casual	39087
4	2	member	136185
5	3	casual	70106

Excel calculation of % based on up to 26h duration of rides.





Conclusions:

Rider Demographics:

Casual riders constitute 35% of all rides.

They prefer electric bikes (53%) over classic ones (43%), while docked bikes have low usage (4%).

Members use electric and classic bikes equally but rarely opt for docked bikes.

Usage Patterns:

Casual riders prefer weekends, with the peak day being Saturday.

Members predominantly use bikes during weekdays, peaking from Tuesday to Thursday. Saturday also sees significant usage.

Seasonal Trends:

Both casual and member bike usage increases during the summer.

Casual riders show a faster increase in summer ride count compared to members.

Classic bike usage by members peaks from May to August, while casual riders shift to classic bikes after docked bikes end in August.

Monthly Peaks:

Classic bike usage peaks for both user types in July.

Members' peak usage falls in August.

Daily Commute Trends:

Members use bikes for commuting at 8:00 and 17:00 and around lunchtime (12:00).

A portion of casual riders also exhibit similar behavior. Casual riders' overall ride count increases steadily until 17:00.

Both user types use bikes around midnight in the summer.

Common Ride Durations and times:

The most common ride durations for members are 3-10 minutes (54% of rides), peaking at 4-6 minutes.

The range of similar duration times is wider for casual users:

Casual riders prefer 3-14 minute rides (54% of rides), with a peak from 5-8 minutes.

Weekend hourly usage patterns are similar for both, with members starting rides earlier. Peak times are 11:00-17:00.

Ride Duration Anomalies:

Rides with durations below zero or exceeding 26 hours represent under 2% of all rides.

The reason behind these anomalies needs further investigation.

Station ID/Name Data Missing:

There is missing station ID/name data that requires exploration to understand the root cause.

Summary:

Members use bikes to commute to work and around lunchtime and therefore use them more during weekdays than on weekends. Favourite duration ride range is more narrow than casual riders'.

Casual riders prefer weekends and summer, in general, the usage during the day increases steadily towards around 17:00. Electric bikes are preferred slightly over classic bikes.

Three top recommendations:

- Offer a limited-time free trial membership for casual riders, allowing them to experience the benefits of membership without an immediate commitment.
- Introduce weekend memberships, designed for rides exclusively on weekends (Friday – Sunday) throughout the year, and offer quarterly memberships with reduced pricing during low-usage months.
- Launch an advertising campaign showcasing scenic and nature destinations accessible by bikes, along with guidance on biking to a location and returning via public transport.