# ‘Cyclistic’ Bike-Share Analysis: How Does A Bike-Share Navigate Speedy Success?

For this case study, I assume the role of a junior data analyst working in the marketing analyst team at *Cyclistic*, a bike-share company in Chicago. The director of marketing believes that the company’s future success depends on maximizing the number of annual memberships. Therefore, my team is tasked to design a new marketing strategy to convert casual riders into annual members. To do that, we need to better understand (1) how annual members and casual riders differ, (2) why casual riders would buy a membership, and (3) how digital media could affect their marketing tactics. The director of marketing specifically assigned me to solve the first problem.

To accomplish my task, I will follow the steps of the data analysis process: **Ask**, **Prepare**, **Process**, **Analyze**, **Share**, and **Act**.

**Tools**: Spreadsheet (Microsoft Excel), SQL (BigQuery), Tableau Public

**Documentation**: Visit my[**GitHub account**](https://github.com/jhermienpaul)**.**

**Visualization**: Go to my [**Tableau account**](https://public.tableau.com/app/profile/jhermienpaul)**.**

## Step 1: Ask

**Context**

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geo-tracked and locked into a network of 692 stations across Chicago.

Cyclistic offers three pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase the first two are called casual riders, and the rest are considered annual members.

**Task**

Analyze how casual riders and annual members use Cyclistic bikes differently. Specifically, determine the following:

1. Proportion of **casual riders** and **annual members**

2. Number of rides by **bike type**

3. Frequency of rides by **month** (season)

4. Frequency of rides by **day** of the week (weekday vs weekend)

5. Frequency of rides by **hour** within a day

6. Average **ride duration** by **day** of the week (weekday vs weekend)

## Step 2: Prepare

**Access the data**

To accomplish my task, I will use Cyclistic’s historical trip [data](https://divvy-tripdata.s3.amazonaws.com/index.html) for the year 2022. The dataset has been made available by Motivate International Inc. under this [license](https://www.divvybikes.com/data-license-agreement).

**Collect the data**

The dataset is composed of CSV files containing monthly data. I downloaded 12 files (1 GB in total) corresponding to the months of January-December 2022.

Table

Description automatically generated

**Examine the data**

Upon inspection, each file has the following attributes or headers:

|  |  |
| --- | --- |
| Header | Description |
| ride\_id | unique id of the ride |
| rideable\_type | type of bike ridden |
| started\_at | date and time the ride started |
| ended\_at | date and time the ride ended |
| start\_station\_name | name of the ride’s starting station |
| start\_station\_id | id of the ride’s starting station |
| end\_station\_name | name of the ride’s ending station |
| end\_station\_id | id of the ride’s ending station |
| start\_lat | latitude coordinate of the ride’s starting station |
| start\_lng | longitude coordinate of the ride’s starting station |
| end\_lat | latitude coordinate of the ride’s ending station |
| end\_lng | longitude coordinate of the ride’s ending station |
| member\_casual | membership type: **member** for annual membership riders or **casual** for casual riders |

The files have the following number of observations or rows:

|  |  |  |
| --- | --- | --- |
| **Table** | | **Rows** |
| 1 | January | 103,771 |
| 2 | February | 115,610 |
| 3 | March | 284,043 |
| 4 | April | 371,250 |
| 5 | May | 634,859 |
| 6 | June | 769,205 |
| 7 | July | 823,489 |
| 8 | August | 785,933 |
| 9 | September | 701,340 |
| 10 | October | 558,686 |
| 11 | November | 337,736 |
| 12 | December | 181,807 |
| **Total** | | **5,667,729** |

## Step 3: Process

**Clean the data with spreadsheet**

I initially cleaned each file in Microsoft Excel to reduce its size before uploading it to BigQuery (limited to 100 MB).

1. I checked for incomplete values in all columns using the COUNTBLANK function. I found a significant number of missing values in start\_station\_name, start\_station\_id, end\_station\_name, and end\_station\_id, making them unreliable for data analysis. Thus, I decided to delete them from each table.

Graphical user interface, application, table, Excel

Description automatically generated

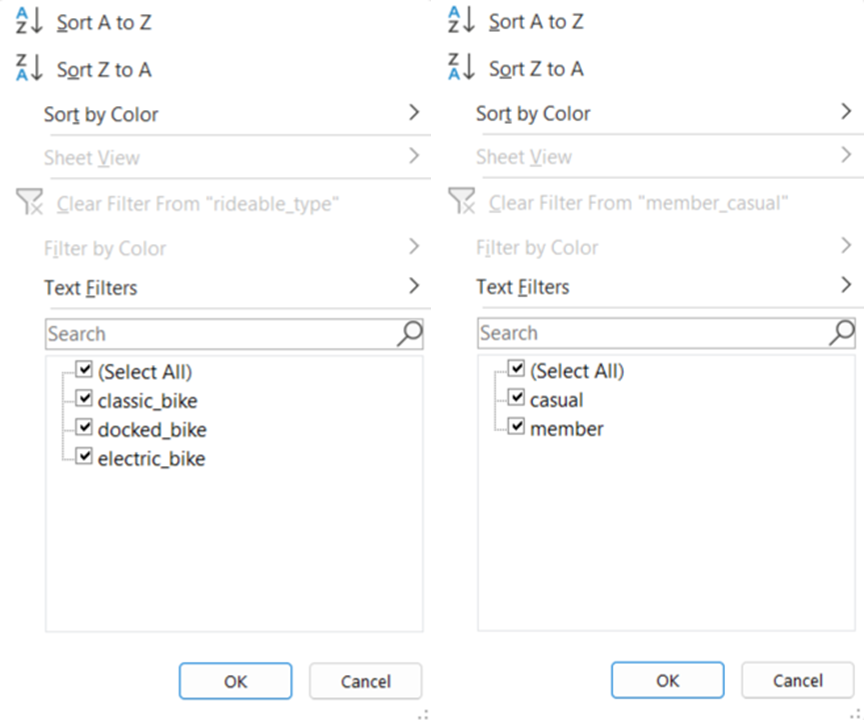
2. I checked for irrelevant values in the remaining columns and noticed that start\_lat, start\_lng, end\_lat, and end\_lng are unlikely to be useful in addressing my specific tasks. Hence, I decided to also delete them.

3. I checked for duplicate values in ride\_id using the Remove Duplicates feature. No duplicate values were found in each file.

Graphical user interface, text

Description automatically generated

4. I checked for inaccurate values in rideable\_type and member\_casual. No inaccurate values were found in each file: bike type only includes classic\_bike, docked\_bike, and electric\_bike, and membership type only includes casual and member.



5. Since I intend to compare the ride duration of casual riders and annual members, I inserted a column named ride\_length computed as ended\_at minus started\_at. Then, I converted the column into TIME (HH:MM:SS) format.

6. I checked for outliers in ride\_length using the Custom Filter feature. I retrieved values less than 0:01:00 (one minute) or greater than 24:00:00 (24 hours) and deleted the rows containing those values.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | | **Deleted Rows** | **Percentage** |
| 1 | January | 10,326 | 9.95% |
| 2 | February | 15,924 | 13.77% |
| 3 | March | 32,304 | 11.37% |
| 4 | April | 45,822 | 12.34% |
| 5 | May | 78,630 | 12.39% |
| 6 | June | 98,490 | 12.80% |
| 7 | July | 112,308 | 13.64% |
| 8 | August | 110,058 | 14.00% |
| 9 | September | 97,254 | 13.87% |
| 10 | October | 81,456 | 14.58% |
| 11 | November | 47,562 | 14.08% |
| 12 | December | 29,478 | 16.21% |
| **Total** | | **647,304** | **13.40%** |

7. Finally, I sorted the table according to started\_at in ascending order and saved it.

**Transform the data with SQL**

In BigQuery, I created a dataset named **cyclistic\_2022** and uploaded the 12 cleaned tables. Then, I executed the following procedures:

1. I created a table named **all\_months** to combine the rows from the 12 tables using the UNION ALL command.

CREATE TABLE awesome-tempo-374012.cyclistic\_2022.all\_months AS

  SELECT \*

  FROM (

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.January`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.February`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.March`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.April`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.May`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.June`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.July`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.August`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.September`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.October`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.November`

    UNION ALL

    SELECT \*

    FROM `awesome-tempo-374012.cyclistic\_2022.December`

  );

2. From that table, I created a new table named **main\_table** to add other columns necessary to accomplish my tasks. Since I intend to compare the frequency of rides by month, day, and hour, I inserted columns for such attributes using the CASE statement and the EXTRACT function. Since I also want to get the average ride duration and realized ride\_length (TIME format) cannot be used for the AVG function, I also added a new column named duration\_mins (INTEGER format) computed as the minute interval between ended\_at and started\_at.

CREATE TABLE awesome-tempo-374012.cyclistic\_2022.main\_table AS

  SELECT

    ride\_id,

    rideable\_type,

    started\_at,

    ended\_at,

ride\_length,

    member\_casual,

    CASE

      WHEN EXTRACT(MONTH FROM started\_at) = 1 THEN "January"

      WHEN EXTRACT(MONTH FROM started\_at) = 2 THEN "February"

      WHEN EXTRACT(MONTH FROM started\_at) = 3 THEN "March"

      WHEN EXTRACT(MONTH FROM started\_at) = 4 THEN "April"

      WHEN EXTRACT(MONTH FROM started\_at) = 5 THEN "May"

      WHEN EXTRACT(MONTH FROM started\_at) = 6 THEN "June"

      WHEN EXTRACT(MONTH FROM started\_at) = 7 THEN "July"

      WHEN EXTRACT(MONTH FROM started\_at) = 8 THEN "August"

      WHEN EXTRACT(MONTH FROM started\_at) = 9 THEN "September"

      WHEN EXTRACT(MONTH FROM started\_at) = 10 THEN "October"

      WHEN EXTRACT(MONTH FROM started\_at) = 11 THEN "November"

      ELSE "December"

    END AS month,

    CASE

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN "Sunday"

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN "Monday"

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN "Tuesday"

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN "Wednesday"

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN "Thursday"

      WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN "Friday"

      ELSE "Saturday"

    END AS day,

    EXTRACT(HOUR FROM started\_at) AS hour,

    DATE\_DIFF(ended\_at, started\_at, minute) AS duration\_mins,

  FROM `awesome-tempo-374012.cyclistic\_2022.all\_months`

  ORDER BY started\_at ASC;

Here’s a preview of the latest table:

Table, Excel

Description automatically generated

## Step 4: Analyze

Once again, I used BigQuery to analyze the data and extract the values I need to accomplish my tasks.

**Proportion of casual riders and annual members**

SELECT

  member\_casual as membership\_type,

  COUNT(member\_casual) AS total\_rides,

  ROUND(COUNT(member\_casual) / SUM(COUNT(member\_casual)) OVER() \* 100, 2) AS percentage

FROM

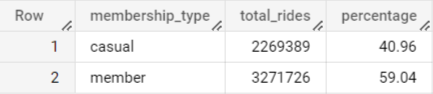
  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  member\_casual

ORDER BY

 member\_casual;



**Number of rides by bike type**

SELECT

  rideable\_type AS bike\_type,

  member\_casual AS membership\_type,

  COUNT(rideable\_type) AS total\_rides,

  ROUND(COUNT(rideable\_type) / SUM(COUNT(rideable\_type)) OVER(PARTITION BY rideable\_type) \* 100, 2) AS percentage\_per\_bike\_type

FROM

  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  rideable\_type, member\_casual

ORDER BY

  rideable\_type, member\_casual;

Table

Description automatically generated

**Frequency of rides by month (season)**

SELECT

  month,

  member\_casual,

  COUNT(month) AS total\_rides,

  ROUND(COUNT(month) / SUM(COUNT(month)) OVER(PARTITION BY month) \* 100, 2) AS percentage\_per\_month

FROM

  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  month, member\_casual

ORDER BY

  month, member\_casual;

Table

Description automatically generated

**Frequency of rides by day (weekday vs weekend)**

SELECT

  day,

  member\_casual,

  COUNT(day) AS total\_rides

FROM

  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  day, member\_casual

ORDER BY

  day, member\_casual;

Table

Description automatically generated

**Frequency of rides by hour**

SELECT

  member\_casual,

  hour,

  COUNT(hour) AS total\_rides

FROM

  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  member\_casual, hour

ORDER BY

  member\_casual, hour;

Table

Description automatically generated

**Average ride duration by day (weekday vs weekend)**

SELECT

  day,

  member\_casual,

  ROUND(AVG(duration\_mins), 2) AS avg\_duration\_mins

FROM

  `awesome-tempo-374012.cyclistic\_2022.main\_table`

GROUP BY

  day, member\_casual

ORDER BY

  day, member\_casual;

Table

Description automatically generated

## Step 5: Share

I used Tableau Public to visualize the results of my analysis. Visit my [**Tableau account**](https://public.tableau.com/app/profile/jhermienpaul) to check the data viz for this project.

**Proportion of casual riders and annual members**

A majority of the 5.54 million total rides taken in 2022 were by annual members (59.04%).

Diagram

Description automatically generated

**Number of rides by bike type**

Classic bikes are preferred by annual members, while electric bikes are preferred by casual riders.

Chart, bar chart

Description automatically generated

**Frequency of rides by month (season)**

Cyclistic bikes are rented the most by both annual members and casual riders during the summer months (June to August), and the least during the winter months (December to January).

Chart, bar chart

Description automatically generated

**Frequency of rides by day (weekday vs weekend)**

Annual members tend to rent bikes more during weekdays, while casual riders rent bikes more during weekends.

Chart, bar chart

Description automatically generated

**Frequency of rides by hour**

Annual members typically rent bikes during rush hours (6-10 AM and 3-7 PM), while casual riders gradually rent bikes from 6 AM until 5 PM.

Chart, line chart

Description automatically generated

**Average ride duration by day (weekday vs weekend)**

On average, casual riders rent bikes for longer periods than annual members, particularly on weekends.

Chart, bar chart

Description automatically generated

## Step 6: Act

Based on these findings, I would recommend the following strategies to help Cyclistic convert more casual riders into annual members:

**Expand the electric bike fleet**

As casual riders tend to prefer electric bikes, Cyclistic could consider expanding its electric bike fleet to attract more casual riders. This could include offering discounts or promotions for new members who choose the electric bike option.

**Promote summer riding**

Since both annual members and casual riders rent bikes the most during the summer, Cyclistic could develop a summer promotion campaign that offers discounts or other incentives to encourage casual riders to become annual members during this peak season.

**Offer flexible pricing plans**

Since casual riders tend to rent bikes more during weekends, Cyclistic could consider offering flexible pricing plans that allow weekend-only usage at a discounted rate. This could appeal to casual riders who may not want to commit to a full annual membership.

**Host events during peak hours**

Cyclistic could host events during peak hour periods that cater to the needs of casual riders and promote the benefits of becoming an annual member. This could include commuter-focused events, such as group rides or bike safety classes, that help riders feel more comfortable using bikes for commuting purposes.

**Offer bike storage solutions**

Since casual riders tend to rent bikes for longer periods, Cyclistic could consider offering bike storage solutions at select locations to make it easier for casual riders to store their bikes safely and conveniently during longer rides or breaks.

Graphical user interface, application

Description automatically generated

*This project was completed as part of the requirements for the* [***Google Data Analytics Professional Certificate***](https://www.coursera.org/professional-certificates/google-data-analytics)*program.*