Bike-Share Case Study

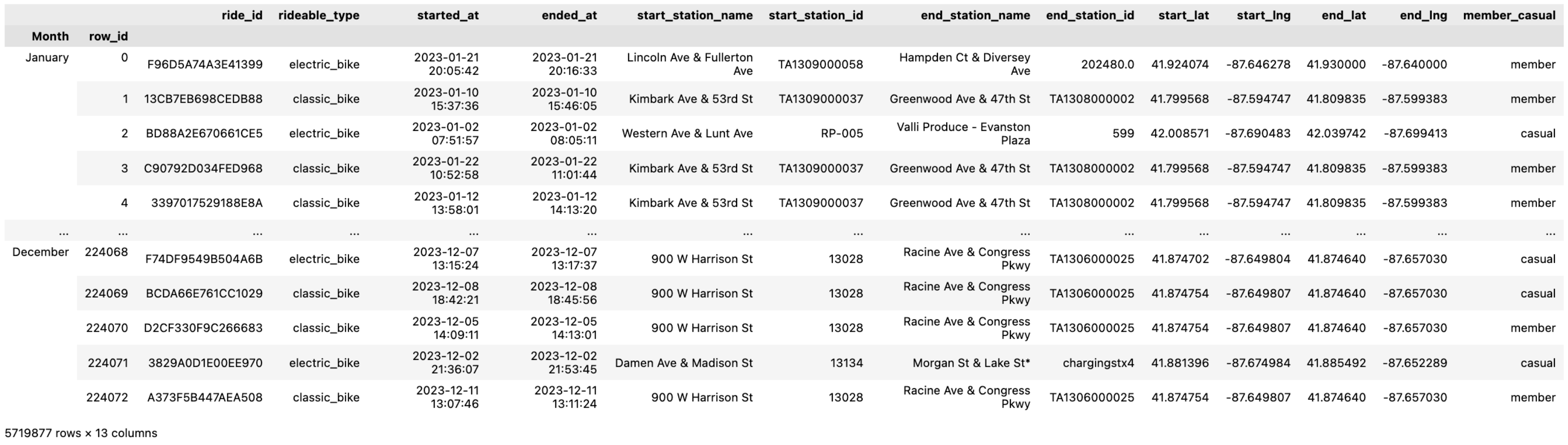
This report provides the results and step-by-step explanation of the data analysis performed for a bike-sharing case study. The data belongs to a company that has two kinds of users: annual members and casual riders. The goal of the study was to identify how annual members and casual riders use the bikes differently in order to help the stake-holders decide whether or not to target converting casual riders into annual members in the next marketing campaign. The data on which the analysis was carried out is from January-December 2023 and was downloaded from https://divvy-tripdata.s3.amazonaws.com/index.html.

The library Pandas from Python was used to perform the analysis, and Matplotlib was used to plot the results. The code can be found in the Jupyter Notebook bike_share_analysis.ipynb.

In the following sections the methods that were used throughout the analysis are explained.

Exploration:

- *read_data*:
  This method reads the bike rides from each .csv file (month), stores each month into a DataFrame (DF), and then concatenates the data into a single multi-index DF. Using a multi-index DF allows the distinction between the different months to still be maintained, while also facilitating the aggregation of values across the entire year when needed. In Figure (1) we can see the first and last 5 entries of bike rides from the concatenated multi-index DF:



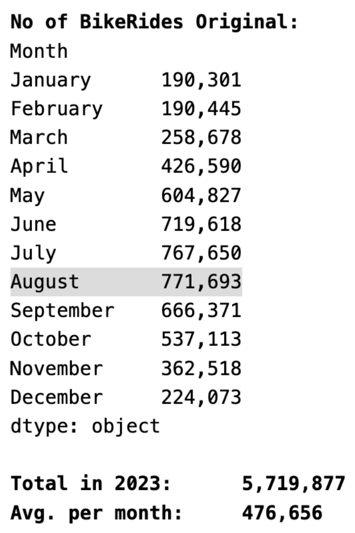
		ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
Month	**row_id**													
January	0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	Lincoln Ave & Fullerton Ave	TA1309000058	Hampden Ct & Diversey Ave	202480.0	41.924074	-87.646278	41.930000	-87.640000	member
	1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	Kimbark Ave & 53rd St	TA1309000037	Greenwood Ave & 47th St	TA1308000002	41.799568	-87.594747	41.809835	-87.599383	member
	2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	Western Ave & Lunt Ave	RP-005	Valli Produce - Evanston Plaza	599	42.008571	-87.690483	42.039742	-87.699413	casual
	3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	Kimbark Ave & 53rd St	TA1309000037	Greenwood Ave & 47th St	TA1308000002	41.799568	-87.594747	41.809835	-87.599383	member
	4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	Kimbark Ave & 53rd St	TA1309000037	Greenwood Ave & 47th St	TA1308000002	41.799568	-87.594747	41.809835	-87.599383	member
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
December	224068	F74DF9549B504A6B	electric_bike	2023-12-07 13:15:24	2023-12-07 13:17:37	900 W Harrison St	13028	Racine Ave & Congress Pkwy	TA1306000025	41.874702	-87.649804	41.874640	-87.657030	casual
	224069	BCDA66E761CC1029	classic_bike	2023-12-08 18:42:21	2023-12-08 18:45:56	900 W Harrison St	13028	Racine Ave & Congress Pkwy	TA1306000025	41.874754	-87.649807	41.874640	-87.657030	casual
	224070	D2CF330F9C266683	classic_bike	2023-12-05 14:09:11	2023-12-05 14:13:01	900 W Harrison St	13028	Racine Ave & Congress Pkwy	TA1306000025	41.874754	-87.649807	41.874640	-87.657030	member
	224071	3829A0D1E00EE970	electric_bike	2023-12-02 21:36:07	2023-12-02 21:53:45	Damen Ave & Madison St	13134	Morgan St & Lake St*	chargingstx4	41.881396	-87.674984	41.885492	-87.652289	casual
	224072	A373F5B447AEA508	classic_bike	2023-12-11 13:07:46	2023-12-11 13:11:24	900 W Harrison St	13028	Racine Ave & Congress Pkwy	TA1306000025	41.874754	-87.649807	41.874640	-87.657030	member

5719877 rows × 13 columns

Figure 1: First and last 5 entries of bike rides from the original data

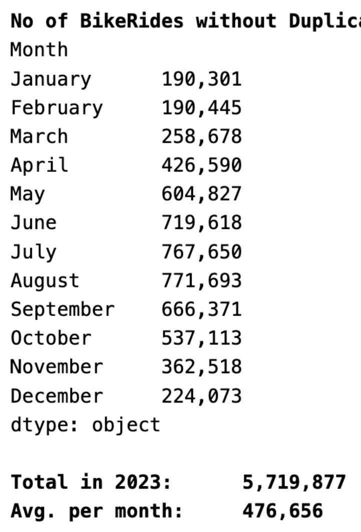
In Figure (1) the multi-index of the DF is shown in the first two columns (month, row_id). Then looking at the entries themselves we can see that the data consists of 13 columns: 1) ride id, 2) type of bike, 3-4) date and time for the start and end of the ride, 5-12) the name, id, latitude and longitude of the start and end stations, and 13) whether the rider was a casual rider or a member.

- *count_entries*:
  This method shows the number of bike rides in each month, the total number of bike rides in the dataset (year 2023), and the average per month. The result is shown in Figure (2a). The dataset contains in total almost 5.7 Million entries, with an average of approximately 480,000 rides per month. From Decemeber to March the number of rides is relatively lower than the average, which is expected as these are cold months. This is confirmed by the peak highlighted in August. After retrieving this information for the original dataset, the duplicates are dropped, and the method is called again. The result of running the method after dropping the duplicates is shown in Figure (2b). The number of entries before and after is identical, therefore the original dataset did not have any duplicates.



```
No of BikeRides Original:
Month
January      190,301
February     190,445
March        258,678
April        426,590
May          604,827
June         719,618
July         767,650
August       771,693
September    666,371
October      537,113
November     362,518
December     224,073
dtype: object

Total in 2023:      5,719,877
Avg. per month:     476,656
```

(a)



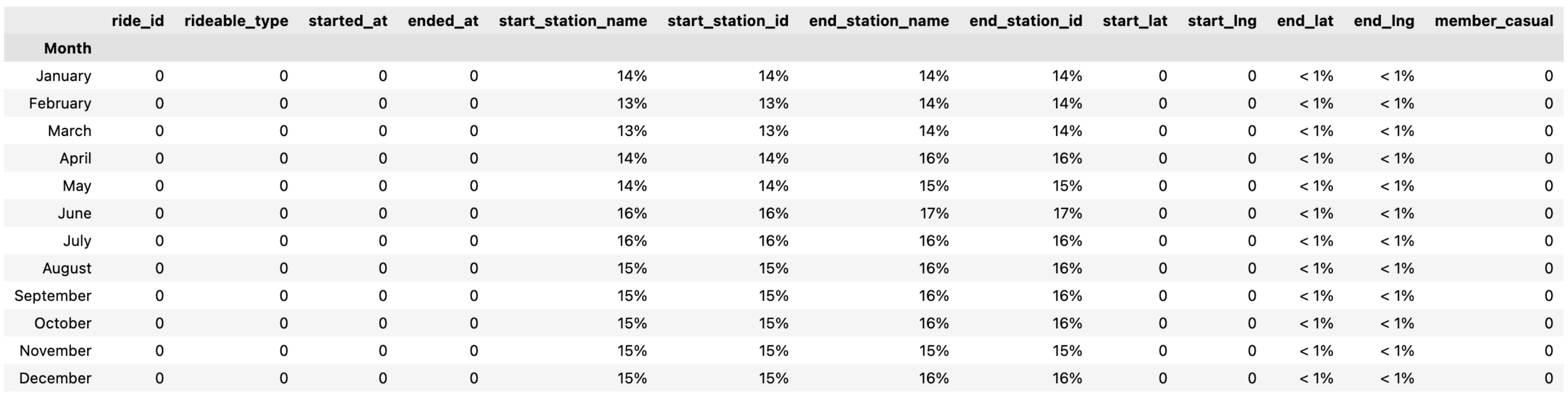
```
No of BikeRides without Duplicates:
Month
January      190,301
February     190,445
March        258,678
April        426,590
May          604,827
June         719,618
July         767,650
August       771,693
September    666,371
October      537,113
November     362,518
December     224,073
dtype: object

Total in 2023:      5,719,877
Avg. per month:     476,656
```

(b)

Figure 2: Total no of entries and average over the year, before and after removing duplicates

- *get_null_percentage*:
  This method calculates the percentage of null values for each column. The results are shown in Figure (3). As we can see the columns *start_station_name*, *start_station_id*, *end_station_name*, *end_station_id* in every month have 13-17% null values. The columns *end_lat* and *end_long* have less than 1% null values.

- *get_unique*:
  This method gets the number of unique (distinct) values for each column in the month of May. The result is shown in Figure (4). The choice of the month of May is random and should not make a difference.
  - *ride_id_unique* = 604,827: as expected, *ride_id* has as many unique values as the number of entries in the dataframe.
  - *rideable_type_unique* = 3: these 3 values are the types of bikes that are offered by the company, which are: $\big($electric_bike, classic_bike, docked_bike$\big)$.



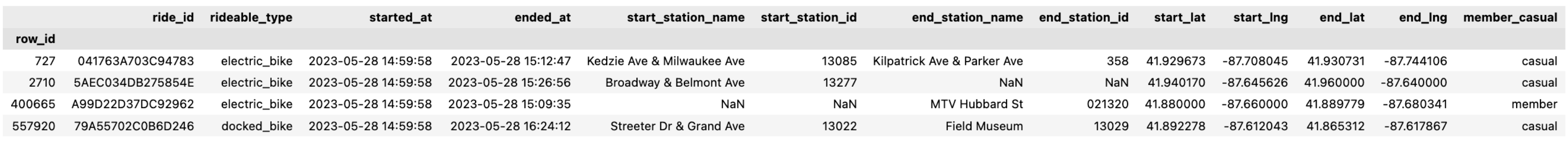
	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
Month													
January	0	0	0	0	14%	14%	14%	14%	0	0	< 1%	< 1%	0
February	0	0	0	0	13%	13%	14%	14%	0	0	< 1%	< 1%	0
March	0	0	0	0	13%	13%	14%	14%	0	0	< 1%	< 1%	0
April	0	0	0	0	14%	14%	16%	16%	0	0	< 1%	< 1%	0
May	0	0	0	0	14%	14%	15%	15%	0	0	< 1%	< 1%	0
June	0	0	0	0	16%	16%	17%	17%	0	0	< 1%	< 1%	0
July	0	0	0	0	16%	16%	16%	16%	0	0	< 1%	< 1%	0
August	0	0	0	0	15%	15%	16%	16%	0	0	< 1%	< 1%	0
September	0	0	0	0	15%	15%	16%	16%	0	0	< 1%	< 1%	0
October	0	0	0	0	15%	15%	16%	16%	0	0	< 1%	< 1%	0
November	0	0	0	0	15%	15%	15%	15%	0	0	< 1%	< 1%	0
December	0	0	0	0	15%	15%	16%	16%	0	0	< 1%	< 1%	0

Figure 3: Percentage of null values for each column

```
Column Name          NUnique Values
ride_id                604,827
rideable_type                3
started_at             503,683
ended_at               505,259
start_station_name       1,287
start_station_id         1,250
end_station_name         1,254
end_station_id           1,210
start_lat              188,591
start_lng              185,410
end_lat                  4,759
end_lng                  4,762
member_casual                2
```

Figure 4: No. of unique values for every column for the month of May

– *started(ended)_at_unique* = 503,683, 505,259: since these are datetimes (yy-mm-dd hh:mm:ss), one may have expected them to have as many distinct values as the *ride_id_unique* = 604,827. Because, at a first glance, it seems unlikely for more than one rider to have rented a bike at the exact same time as another rides down to the second. However, the number of unique values in these columns is less than the number of entries by 16%. Which means that 16% of the bikes rides have the same *started_at* as other rides. To explore this a bit further, one of these duplicates has been retrieved and is shown in Figure (5). By looking at the rides, it is clear that they are indeed different entries but with the exact same start time.



	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
row_id													
727	041763A703C94783	electric_bike	2023-05-28 14:59:58	2023-05-28 15:12:47	Kedzie Ave & Milwaukee Ave	13085	Kilpatrick Ave & Parker Ave	358	41.929673	-87.708045	41.930731	-87.744106	casual
2710	5AEC034DB275854E	electric_bike	2023-05-28 14:59:58	2023-05-28 15:26:56	Broadway & Belmont Ave	13277	NaN	NaN	41.940170	-87.645626	41.960000	-87.640000	casual
400665	A99D22D37DC92962	electric_bike	2023-05-28 14:59:58	2023-05-28 15:09:35	NaN	NaN	MTV Hubbard St	021320	41.880000	-87.660000	41.889779	-87.680341	member
557920	79A55702C0B6D246	docked_bike	2023-05-28 14:59:58	2023-05-28 16:24:12	Streeter Dr & Grand Ave	13022	Field Museum	13029	41.892278	-87.612043	41.865312	-87.617867	casual

Figure 5: Entries from May that have the same *started_at* date and time

– *start(end)_station_name(id)_unique* = 1287, 1250, 1254, 1210: since there is a limited number of stations, it is expected that these columns have a smaller number of unique values than the number of entries. However, one would have expected the number of station names and ids to be the same, whereas the ids are less than the names by a small fraction. Which could either by accounted for by the null values (Fig. 4) or could mean that there are stations that have the different names but the same id.

  - *start(end)_lat(long)_unique* = 188,591, 185,419, 4759, 4762: the start latitude and longitude numbers seem to be as expected, which is less than the total number of entries, but more than the number of stations (this is based on the assumption that the exact location where a bike is parked can vary within the station especially that the values are given to the 6^{th} decimal place). However, there is a large difference between the number of values in the start (∼188,000) and the numbers in the end (∼4800). This difference cannot be accounted for by the null values in the end columns, since these were less than 1%. If these values are true, that would mean that users rode there bikes from many start locations, but mostly ended up in a much smaller set of locations. Which cannot be the case since that would have been reflected in a similar difference between the number of start and end stations.
  - *member_casual_unique* = 2: these 2 values are the types of riders, which are: $\big($member, casual$\big)$.

Cleaning:

After exploring the dataset, we can see that the extractable information can be divided into information about the:

1. rider (casual/member)
2. bike (electric/classic/docked)
3. ride (start-end: time, date, location)

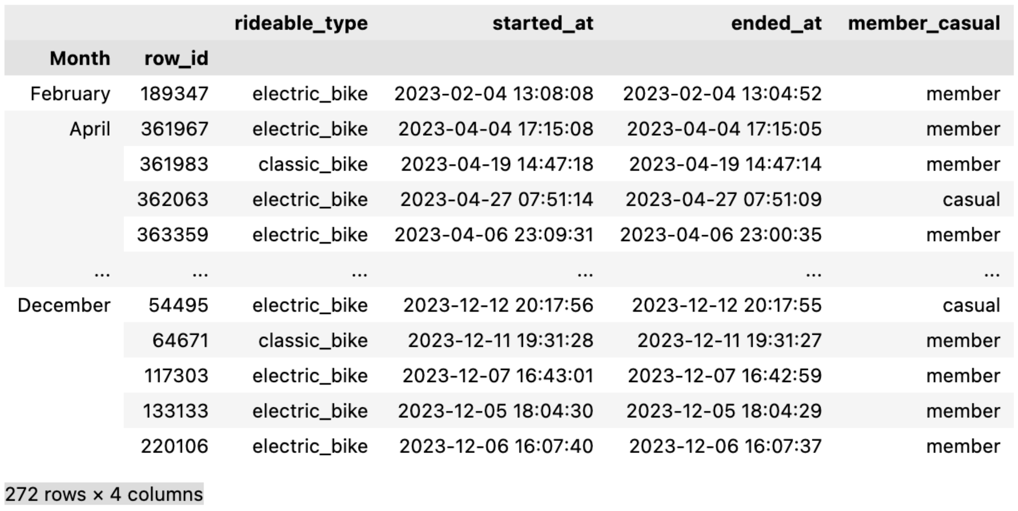
Since, the goal of the analysis is to find out whether to target converting casual riders into members or not, it seems that all the information is relevant to the analysis, with the exception of the ride location. This is based on the assumption that the amount paid by the rider is based on how long the bike was rented, regardless of where. Furthermore, the geographical location would have been important if for example the goal of the analysis was to find out whether more stations should be added and where to do so. Therefore, since the locations seem to be irrelevant, contain null values and discrepancies, these columns will be dropped in the cleaning process. The column *ride_id* also does not provide any valuable information for the current analysis.

Another task to be performed in the cleaning phase is data formatting. We have already looked at the columns *rideable_type* and *member_casual*, and ensured that they have only the expected values and no nulls. As for the columns *started_at* and *ended_at*, these do not have null values, but still need to be checked to ensure that the *ended_at* time always comes after *started_at* time. In order to compare the values in these columns, they first need to converted from strings of characters to a numerical date-time format.

- *drop_and_change_format*:
  This method performs three tasks. First, it drops all the columns related to geographical location and ride id. Second, it converts the columns *started_at* and *ended_at* from strings of characters to a numerical datetime format. Finally, the method returns a Series of flags which indicate whether or not the *ended_at* time is before the *started_at* time.

- *clean_data*:
  By using the flags returned by the method *drop_and_change_format*, the rides where *ended_at* time is before the *started_at* time can be filtered out as shown in Figure (6). Looking at these entries, we

can see that there are cases when the *ended_at* time is before the *started_at* time by just a seconds. It can be that in these incidents the start and end time were switched due to some glitch, perhaps the bike rental time being very short (shorter than the server response time). But there are also cases when the time difference is a few minutes. In the entire dataset of approximately 5.7 Million entries, there is a total of 272 entries that have this issue. Since, the dataset is large, the method *clean_data* simply drops these entries.



		rideable_type	started_at	ended_at	member_casual
Month	**row_id**				
February	189347	electric_bike	2023-02-04 13:08:08	2023-02-04 13:04:52	member
April	361967	electric_bike	2023-04-04 17:15:08	2023-04-04 17:15:05	member
	361983	classic_bike	2023-04-19 14:47:18	2023-04-19 14:47:14	member
	362063	electric_bike	2023-04-27 07:51:14	2023-04-27 07:51:09	casual
	363359	electric_bike	2023-04-06 23:09:31	2023-04-06 23:00:35	member
...	...	...	...	...	...
December	54495	electric_bike	2023-12-12 20:17:56	2023-12-12 20:17:55	casual
	64671	classic_bike	2023-12-11 19:31:28	2023-12-11 19:31:27	member
	117303	electric_bike	2023-12-07 16:43:01	2023-12-07 16:42:59	member
	133133	electric_bike	2023-12-05 18:04:30	2023-12-05 18:04:29	member
	220106	electric_bike	2023-12-06 16:07:40	2023-12-06 16:07:37	member

272 rows × 4 columns

Figure 6: Entries where the ended_at time is before the started_at time

Preparation:

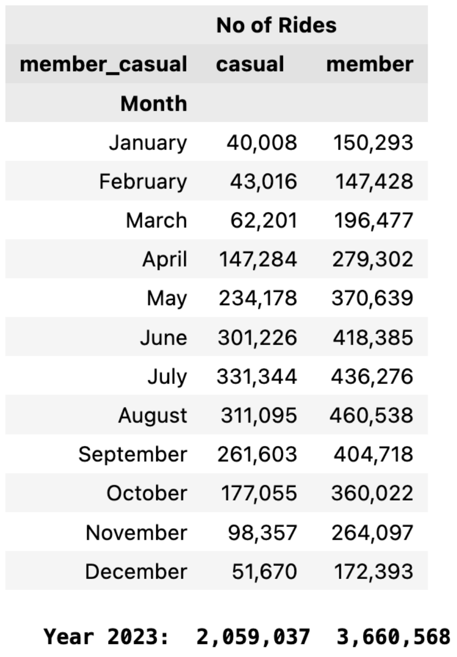
Now that the dataset has been explored and cleaned, the next step is to prepare the data needed for analysis. Based on the assumption that the rides are charged according to how long they were rented, a column *ride_length* is added. Another column *day_of_week* is also added to explore the behaviour of different riders a bit further.

- *prepare_data*:
  This method adds the two new columns: *ride_length*: the difference between the columns *ended_at* and *started_at* times, and *day_of_week*: extracted from the date in *started_at*. The method then drops the columns *started_at* and *ended_at*, since the new columns make them redundant.

Analysis:

Finally, the data can be analysed in order to determine how casual riders and members differ.

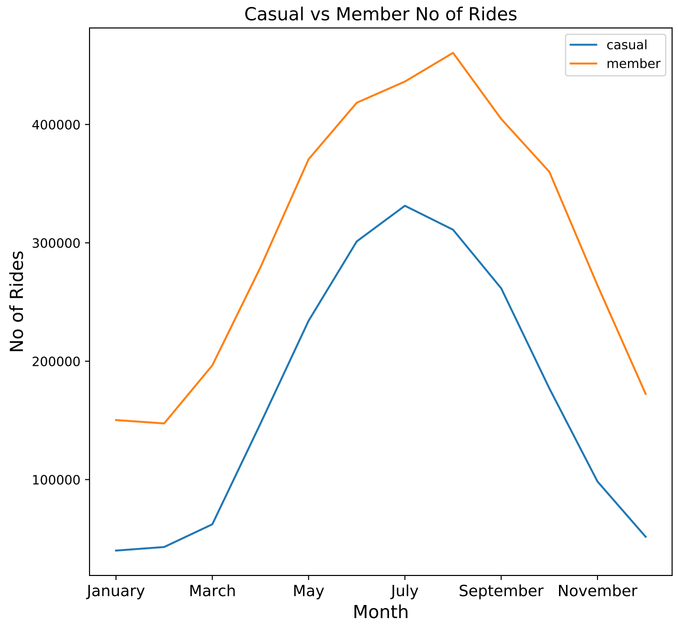
- *count_rides_per_rider*:
  This method produces a pivot table whose index is the month, and the columns are the number of rides for each rider type. The method also returns the total number of rides for the entire year. The resulting pivot table is shown in Figure (7a), and is visualised using the plot shown on the right Figure (7b). From the plot we observe that, per month there is approximately 100,000 more rides by members than by casual riders. Next, we see that rides by members and casual riders follow the same pattern of peaking during the summer months, and dropping during the winter months.



	No of Rides	
member_casual	**casual**	**member**
Month		
January	40,008	150,293
February	43,016	147,428
March	62,201	196,477
April	147,284	279,302
May	234,178	370,639
June	301,226	418,385
July	331,344	436,276
August	311,095	460,538
September	261,603	404,718
October	177,055	360,022
November	98,357	264,097
December	51,670	172,393

Year 2023: 2,059,037 3,660,568

(a)



(b)

Figure 7: Total no of rides per rider type

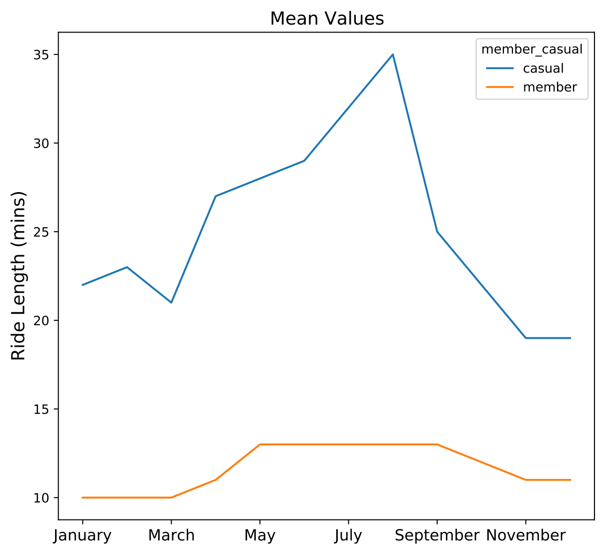
- *mean_max_by_month*:
  This method produces a pivot table whose index is the month, and the columns are the mean and maximum ride length for each rider type. The method also calculates the mean and max values across the entire year. The pivot table is shown in Figure (8a), and the corresponding visualisations are shown below. From the plot of the mean values in Figure (8b) we observe that casual riders have longer rides, than members across the entire year. Also, the mean ride length for casual riders varies across the year from approximately 20 minutes in the colder months to a peak of 35 minutes in August. As for members, their mean ride length changes only from 10 to 13 minutes. When looking at the maximum ride length plot in Figure (8c) we see a similar pattern: casual riders have longer rides, and their maximum ride lengths changes from 1 day in the cold months to 68 days in August. As opposed to, members whom have a maximum ride length of 1 day all throughout the year.

  However, it is important to note here that such longer rides, as seen in the maximum ride length plot, are outliers that occur quite rarely in the dataset. This can be seen when plotting the entire dataset distribution of ride lengths using a box plot as shown in Figure (9).

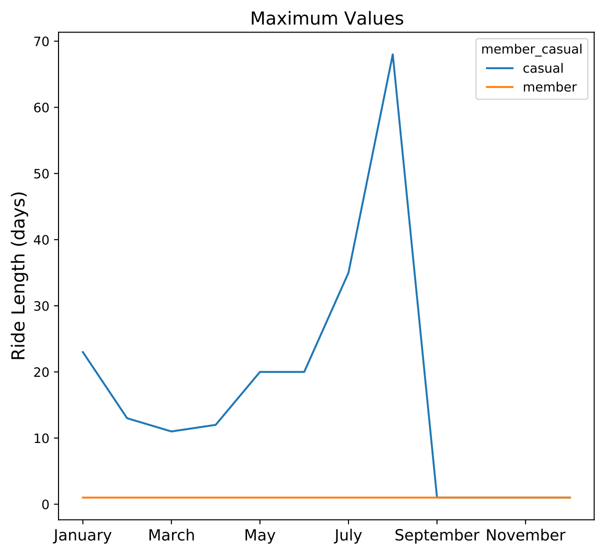
	ride_length			
	mean		max	
member_casual	member	casual	member	casual
Month				
January	00:10:21	00:22:54	01 days 00:59:56	23 days 08:03:44
February	00:10:42	00:23:11	01 days 00:59:56	13 days 02:25:46
March	00:10:26	00:21:24	01 days 01:59:40	11 days 16:08:04
April	00:11:41	00:27:40	01 days 00:59:56	12 days 18:35:29
May	00:13:02	00:28:31	01 days 01:00:31	20 days 06:50:31
June	00:13:12	00:29:24	01 days 00:59:56	20 days 11:05:58
July	00:13:41	00:32:20	01 days 00:59:57	35 days 17:41:24
August	00:13:46	00:35:14	01 days 00:59:57	68 days 09:29:04
September	00:13:08	00:25:11	01 days 00:59:57	01 days 01:07:46
October	00:12:09	00:22:52	01 days 00:59:56	01 days 00:59:57
November	00:11:34	00:19:54	01 days 00:59:56	01 days 01:00:25
December	00:11:26	00:19:56	01 days 00:59:56	01 days 00:59:57

`Year 2023:  00:12:06  00:25:42  01 days 01:59:40 68 days 09:29:04`

(a)



(b)



(c)

Figure 8: Mean and max ride length divided by rider type

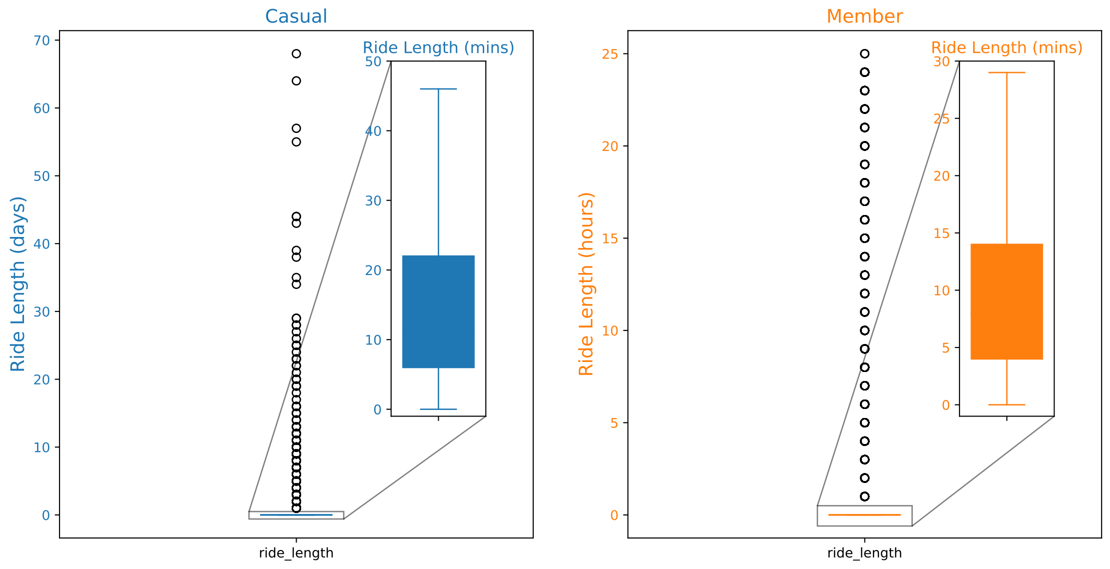


Figure 9: Box plots of the distribution of values in the ride length for casual riders and members

A box plot typically divides the values in a distribution into 4 quartiles (0-25%, 25-50%, 50-75%, 75-100%), and displays them as follows: the middle 2 quartiles (25-75% of the distribution) are shown inside a box, the lower and upper 25% are shown by whiskers, and circles are used to represent outliers. From Figure 9 (left) we can see that, for casual riders, the outlier points are the ones that occupy the range [1 - 70] days, whereas the distribution itself can only be seen in the [0 - 50] minute range. As for the members (Fig. (9) right), the distribution of ride lengths is in the [0 - 30] minute range, and its outliers are in the [1- 25] hour range. Therefore, in order to see the comparison between the ride lengths for casual riders and members more clearly, the outliers are dropped, and the box plots, are shown side by side in Figure (10).

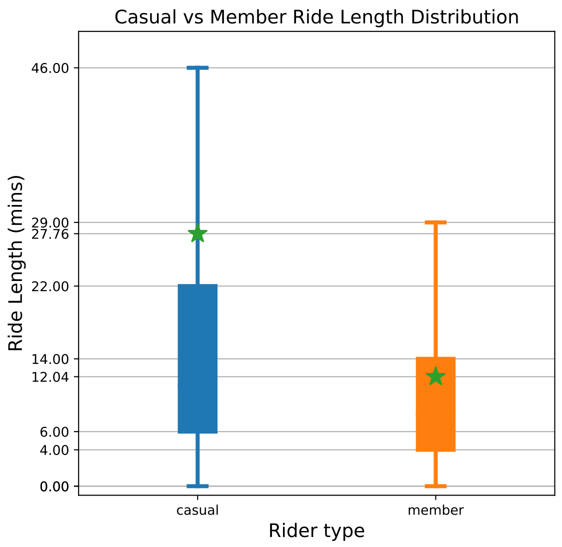
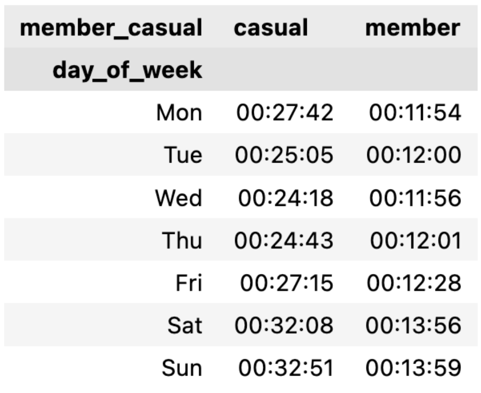


Figure 10: Box plot comparison between the ride lengths of casual and member riders without outliers

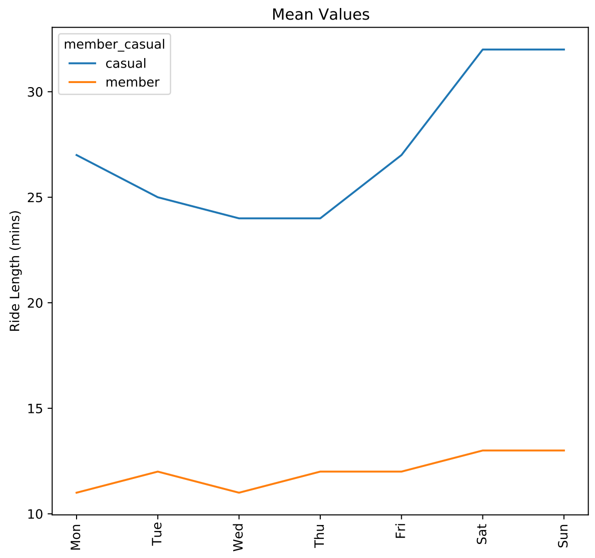
Figure (10) shows that the main distribution (i.e. without outliers) of ride lengths for casual riders varies between [0 - 46] minutes, and that the middle 50% of the values are between [6 - 22] minutes. As opposed to members whose ride length vary between [0 - 29] minutes, and the middle 50% are between [4 - 14] minutes. The mean values calculated from the entire distribution are marked on the plot using the green stars. For casual riders the mean is 27.7 minutes (slightly higher than the mean calculated first by month and then for the year = 25.7 Figure (8a)), and for members it is 12 minutes.

- *mean_by_day_of_week:*
  So far we have compared the rider types either by first grouping the data using the month (Fig. 8b) or by looking at the entire distribution (Fig. 9). In order to get a different perspective, we will now look at the mean value of the ride length after the data has been grouped by the day of the week. The resulting pivot table as well as its visualisation are shown in Figure (11). Here we see the same higher average for the casual riders when compared to members, with a peak for casual riders during the weekend.



member_casual	casual	member
day_of_week		
Mon	00:27:42	00:11:54
Tue	00:25:05	00:12:00
Wed	00:24:18	00:11:56
Thu	00:24:43	00:12:01
Fri	00:27:15	00:12:28
Sat	00:32:08	00:13:56
Sun	00:32:51	00:13:59

(a)



(b)

Figure 11: Mean ride length per rider type grouped by day of the week

- *count_rideable_type*:
  This method produces a pivot table with the rider type (casual/member) as an index, and the columns are rideable type, which is the type of bike (classic/docked/electric). The values are the number of entries in each category. The result is visualised in Figure (12). From the histogram we can see that for classic and electric bikes, there are more member rides than casual rides. Which is to be expected, since the overall number of member rides is larger than the casual riders (Fig. 7a). However, the interesting finding is that when it comes to docked bikes, only casual riders have used those in 2023.

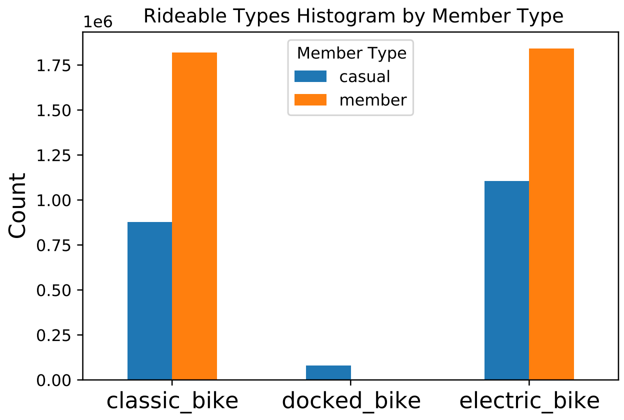


Figure 12: No of Rides for each type of bike per rider type

Conclusion:

In this report, the analysis of a bike-sharing case-study was presented. First, the dataset was explored by looking at what the original data consisted of, what were the unique values in each column, as well as the percentage of null values. For the year 2023, on average, there were ∼480,000 rides per month. During the cold months December-March the rides dropped to ∼200,00, and in the summer months they went up to ∼770,000 rides. No duplicates were found in the dataset, but the columns *start(end)_station_name(id)* had 13-17% null values, and the columns *end_lat(long)* had less than 1% null values. When looking at the number of unique values for each column, some discrepancies were found, for example there were more station names than there were station ids. Next, the data was cleaned by dropping the columns that had inconsistencies, null values or were irrelevant to the analysis. The data was also formatted by converting the start and end times into date-time numerical values and the entries where the end time was before the start were dropped. In order to prepare the data for analysis, the columns ride length and day of the week were added.

The data was then analysed by first looking at the number of rides performed by members versus casual riders. The entire dataset for the year 2023 consists of 5.7 Million rides, 64% of which were made by members, and 36% were by casual riders. The dataset was then grouped by month, and the average and maximum ride length were calculated per rider type. For members the mean ride length in 2023 was ∼12 minutes, whereas for casual riders it was ∼25 minutes. When looking at the maximum ride length it was shown that members never rented the bike for more than one day, whereas casual riders had a maximum ride length ranging from 1 - 68 days. However, by plotting the entire distribution of ride length values using box plots, it was shown that such longer rides were outliers that occurred quite rarely. The box plots showed that the main distribution of ride lengths after ignoring the outliers can be summarised by the values shown in Table (1):

	0 - 25%	25 - 70%	75 - 100%	mean (mins)
casual	0 - 6	6 - 22	22 - 46	27
member	0 - 4	4 - 14	14 - 29	12

Table 1: Distribution of ride lengths (minutes) for casual riders and members without outliers

By looking at the results shown in Table (1) we can confidently conclude that casual riders have longer ride lengths, whether we compare the mean across the whole dataset, or if we compare each quartile of the distribution. Therefore, this analysis supports the recommendation to target converting casual riders into members. Lastly, it was found that in the year 2023, docked bikes were used only by casual riders. So it might be beneficial to use that information within the marketing campaign.