Bike Share Company - Case study

Author: Dimitrios Lavdogiannis
Case study source: Google Analytics Professional Certificate
Duration: 1 week

Table of contents

1. Case study scenario and objectives

1.1 Scenario

You are a junior data analyst working on the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. **From these insights, your team will design a new marketing strategy to convert casual riders into annual members.** But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

1.2 Data Analysis Objectives

The company leaders are determined to make data driven decisions regarding their future strategy. The future marketing program will be guided by the following questions.

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

We are assigned to provide data-driven insights in order to answer the first question. We need to analyze the data, in order to identify trends that show how annual members and casual users use the company's services differently.Analyzing these differences correctly will be the key for answering the other two questions and achieving the company's objective.

2. Data Manipulation and cleaning

2.1 Data Sources

The main data source we used in our analysis is provided by the company. Cyclistic have gathered data on how annual members and casual users have used their services for the past 12 months. The raw data is organized in a **CSV file named “trip data”**. This data set is compiled by the company itself, so it is considered to be highly reliable.

2.2 Data Cleaning

Our first step was to clean our data and make sure that the data set was organized in a meaningful and useful way. The data cleaning process took place using **Google Sheets** and **WPS sheets**. We opened the raw data using **Google Sheets** the. The raw data set had the form that is shown in the following picture.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_nan	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casua
2	A847FADBBC63	docked_bike	4/26/2020 17:45	4/26/2020 18:12	Eckhart Park	86	Lincoln Ave & Di	152	418.964	-87.661	419.322	-876.586	member
3	5405B80E996FF	docked_bike	4/17/2020 17:08	4/17/2020 17:17	Drake Ave & Fuller	503	Kosciuszko Park	499	419.244	-877.154	419.306	-877.238	member
4	5DD24A79A4E0	docked_bike	4/1/2020 17:54	4/1/2020 18:08	McClurg Ct & Erie	142	Indiana Ave & R	255	418.945	-876.179	418.679	-87.623	member
5	2A59BBDF5CDE	docked_bike	4/7/2020 12:50	4/7/2020 13:02	California Ave & D	216	Wood St & Augu	657	41.903	-876.975	418.992	-876.722	member
6	27AD306C119C	docked_bike	4/18/2020 10:22	4/18/2020 11:15	Rush St & Hubbar	125	Sheridan Rd & L	323	418.902	-876.262	419.695	-876.547	casual
7	356216E875132	docked_bike	4/30/2020 17:55	4/30/2020 18:01	Mies van der Rohe	173	Streeter Dr & Gr	35	418.969	-876.217	418.923	-87.612	member
8	A2759CB06A81	docked_bike	4/2/2020 14:47	4/2/2020 14:52	Streeter Dr & Grar	35	Fairbanks St & S	635	418.923	-87.612	418.957	-876.201	member
9	FC8BC2E2D54F	docked_bike	4/7/2020 12:22	4/7/2020 13:38	Ogden Ave & Roo	434	Western Ave & (	382	418.665	-876.847	418.747	-876.864	casual
10	9EC5648678DE	docked_bike	4/15/2020 10:30	4/15/2020 10:35	LaSalle Dr & Huro	627	Larrabee St & Di	359	418.949	-876.323	419.035	-876.434	casual
11	A8FFF89140C3	docked_bike	4/4/2020 15:02	4/4/2020 15:19	Kedzie Ave & Lake	377	Central Park Ave	508	418.846	-877.063	419.097	-877.166	member
12	788B1BB8A749	docked_bike	4/4/2020 15:22	4/4/2020 15:46	Central Park Ave &	508	Western Ave & V	374	419.097	-877.166	418.984	-876.866	member
13	C83C113858BA	docked_bike	4/25/2020 15:43	4/25/2020 15:48	Western Ave & Wa	374	Damen Ave & Cl	128	418.984	-876.866	418.958	-876.772	member
14	D2038D92195B	docked_bike	4/24/2020 18:09	4/24/2020 18:18	Western Ave & Wa	374	Ashland Ave & D	210	418.984	-876.866	419.035	-876.677	member
15	C554B4E072B0	docked_bike	4/11/2020 17:15	4/11/2020 17:19	Western Ave & Wa	374	Damen Ave & Cl	128	418.984	-876.866	418.958	-876.772	member
16	F962D972BC1E	docked_bike	4/20/2020 17:18	4/20/2020 17:42	Wabash Ave & 9th	321	Michigan Ave &	623	418.708	-876.257	418.728	-87.624	member
17	1DDBC1F4D20	docked_bike	4/18/2020 15:49	4/18/2020 16:24	Wabash Ave & 9th	321	Sheffield Ave & I	20	418.708	-876.257	419.105	-876.531	member
18	AA1C8D93190B	docked_bike	4/19/2020 13:39	4/19/2020 14:01	Wabash Ave & 9th	321	Clark St & Lincol	141	418.708	-876.257	419.157	-876.346	member
19	B0EEA4FBCF6E	docked_bike	4/18/2020 2:59	4/18/2020 3:07	Leavitt St & Archer	9	Leavitt St & Arch	9	418.288	-876.806	418.288	-876.806	casual
20	5F2A5CC2510F	docked_bike	4/4/2020 10:52	4/4/2020 11:08	Clark St & Lincoln	141	Southport Ave &	190	419.157	-876.346	419.288	-876.639	casual
21	BEF186AA6B3D	docked_bike	4/25/2020 12:32	4/25/2020 12:38	900 W Harrison St	109	Aberdeen St & F	621	418.748	-876.498	418.841	-876.543	member
22	042511EE70500	docked_bike	4/5/2020 15:39	4/5/2020 16:15	Museum of Scienc	424	Cornell Dr & Hay	653	417.917	-875.839	417.806	-875.848	casual
23	0FF35889739F3	docked_bike	4/10/2020 16:00	4/10/2020 16:32	Ashland Ave & Gr	347	Southport Ave &	227	419.507	-876.687	419.482	-876.639	casual
24	0ECBACECBAC	docked_bike	4/18/2020 12:07	4/18/2020 12:12	Green St & Madis	198	900 W Harrison	109	418.819	-876.488	418.748	-876.498	member
25	B5BBCA72F5A8	docked_bike	4/5/2020 14:29	4/5/2020 14:44	Ravenswood Ave	344	Southport Ave &	227	419.691	-876.742	419.482	-876.639	member
26	F65BB1A0FBF1	docked_bike	4/5/2020 13:44	4/5/2020 13:58	Southport Ave & W	227	Ravenswood Av	344	419.482	-876.639	419.691	-876.742	member

Image 1: raw dataset

The raw data set contains information about the bike trips for the past 12 months, including the trip’s id, the start and end date-time, the start and end station and the user type (annual member or casual user).

2.2.1 Date-time stamp correction

The first problem we encountered was present in the **started_at** and **ended_at** columns. These columns contain date time information but, as we can see below, some of the cells had a wrong format.

	A	B	C	D	E	
1	ride_id	rideable_type	started_at	ended_at	start_station_name	sta
2	A847FADBBC638E45	docked_bike	4/26/2020 17:45	4/26/2020 18:12	Eckhart Park	
3	5405B80E996FF60D	docked_bike	4/17/2020 17:08	4/17/2020 17:17	Drake Ave & Fullerton Ave	
4	5DD24A79A4E006F4	docked_bike	4/1/2020 17:54	4/1/2020 18:08	McClurg Ct & Erie St	
5	2A59BBDF5CDBA725	docked_bike	4/7/2020 12:50	4/7/2020 13:02	California Ave & Division St	
6	27AD306C119C6158	docked_bike	4/18/2020 10:22	4/18/2020 11:15	Rush St & Hubbard St	
7	356216E875132F61	docked_bike	4/30/2020 17:55	4/30/2020 18:01	Mies van der Rohe Way & Ch	
8	A2759CB06A81F2BC	docked_bike	4/2/2020 14:47	4/2/2020 14:52	Streeter Dr & Grand Ave	
9	FC8BC2E2D54F35ED	docked_bike	4/7/2020 12:22	4/7/2020 13:38	Ogden Ave & Roosevelt Rd	
10	9EC5648678DE06E6	docked_bike	4/15/2020 10:30	4/15/2020 10:35	LaSalle Dr & Huron St	
11	A8FFF89140C33017	docked_bike	4/4/2020 15:02	4/4/2020 15:19	Kedzie Ave & Lake St	
12	788B1BB8A7491EBD	docked_bike	4/4/2020 15:22	4/4/2020 15:46	Central Park Ave & North Ave	
13	C83C113858BA06DA	docked_bike	4/25/2020 15:43	4/25/2020 15:48	Western Ave & Walton St	
14	D2038D92195BDD67	docked_bike	4/24/2020 18:09	4/24/2020 18:18	Western Ave & Walton St	
15	C554B4E072B077F8	docked_bike	4/11/2020 17:15	4/11/2020 17:19	Western Ave & Walton St	
16	F962D972BC1EF3F0	docked_bike	4/20/2020 17:18	4/20/2020 17:42	Wabash Ave & 9th St	
17	1DDBC1F4D208C2B3	docked_bike	4/18/2020 15:49	4/18/2020 16:24	Wabash Ave & 9th St	
18	AA1C8D93190BB6A9	docked_bike	4/19/2020 13:39	4/19/2020 14:01	Wabash Ave & 9th St	
19	B0EEA4FBCF6E26A3	docked_bike	4/18/2020 2:59	4/18/2020 3:07	Leavitt St & Archer Ave	
20	5F2A5CC2510F0396	docked_bike	4/4/2020 10:52	4/4/2020 11:08	Clark St & Lincoln Ave	
21	BEF186AA6B3DD4CC	docked_bike	4/25/2020 12:32	4/25/2020 12:38	900 W Harrison St	
22	042511EE70500A4A	docked_bike	4/5/2020 15:39	4/5/2020 16:15	Museum of Science and Indu	
23	0FF35889739F390A	docked_bike	4/10/2020 16:00	4/10/2020 16:32	Ashland Ave & Grace St	
24	0ECBACECBACC97A1	docked_bike	4/18/2020 12:07	4/18/2020 12:12	Green St & Madison St	
25	B5BBCA72F5A8BE3C	docked_bike	4/5/2020 14:29	4/5/2020 14:44	Ravenswood Ave & Lawrence	
26	F65BB1A0FBE1A613	docked_bike	4/5/2020 13:44	4/5/2020 13:58	Southport Ave & Waveland A	

Image 2:Wrong date-time format

We edited the cells using **google sheets,** in order for all the cells to have a correct date-time format. The changes are shown below.

1	ride_id	rideable_type	started_at	ended_at	start_station_name
2	A847FADBBC638E45	docked_bike	4/26/2020 17:45:00	4/26/2020 18:12:00	Eckhart Park
3	5405B80E996FF60D	docked_bike	4/17/2020 17:08:00	4/17/2020 17:17:00	Drake Ave & Fullerton Ave
4	5DD24A79A4E006F4	docked_bike	1/4/2020 17:54:00	1/4/2020 18:08:00	McClurg Ct & Erie St
5	2A59BBDF5CDBA725	docked_bike	7/4/2020 12:50:00	7/4/2020 13:02:00	California Ave & Division St
6	27AD306C119C6158	docked_bike	4/18/2020 10:22:00	4/18/2020 11:15:00	Rush St & Hubbard St
7	356216E875132F61	docked_bike	4/30/2020 17:55:00	4/30/2020 18:01:00	Mies van der Rohe Way & C
8	A2759CB06A81F2BC	docked_bike	2/4/2020 14:47:00	2/4/2020 14:52:00	Streeter Dr & Grand Ave
9	FC8BC2E2D54F35ED	docked_bike	7/4/2020 12:22:00	7/4/2020 13:38:00	Ogden Ave & Roosevelt Rd
10	9EC5648678DE06E6	docked_bike	4/15/2020 10:30:00	4/15/2020 10:35:00	LaSalle Dr & Huron St
11	A8FFF89140C33017	docked_bike	4/4/2020 15:02:00	4/4/2020 15:19:00	Kedzie Ave & Lake St
12	788B1BB8A7491EBD	docked_bike	4/4/2020 15:22:00	4/4/2020 15:46:00	Central Park Ave & North Av
13	C83C113858BA06DA	docked_bike	4/25/2020 15:43:00	4/25/2020 15:48:00	Western Ave & Walton St
14	D2038D92195BDD67	docked_bike	4/24/2020 18:09:00	4/24/2020 18:18:00	Western Ave & Walton St
15	C554B4E072B077F8	docked_bike	11/4/2020 17:15:00	11/4/2020 17:19:00	Western Ave & Walton St
16	F962D972BC1EF3F0	docked_bike	4/20/2020 17:18:00	4/20/2020 17:42:00	Wabash Ave & 9th St
17	1DDBC1F4D208C2B3	docked_bike	4/18/2020 15:49:00	4/18/2020 16:24:00	Wabash Ave & 9th St
18	AA1C8D93190BB6A9	docked_bike	4/19/2020 13:39:00	4/19/2020 14:01:00	Wabash Ave & 9th St
19	B0EEA4FBCF6E26A3	docked_bike	4/18/2020 2:59:00	4/18/2020 3:07:00	Leavitt St & Archer Ave
20	5F2A5CC2510F0396	docked_bike	4/4/2020 10:52:00	4/4/2020 11:08:00	Clark St & Lincoln Ave
21	BEF186AA6B3DD4CC	docked_bike	4/25/2020 12:32:00	4/25/2020 12:38:00	900 W Harrison St
22	042511EE70500A4A	docked_bike	5/4/2020 15:39:00	5/4/2020 16:15:00	Museum of Science and Ind
23	0FF35889739F390A	docked_bike	10/4/2020 16:00:00	10/4/2020 16:32:00	Ashland Ave & Grace St
24	0ECBACECBACC97A1	docked_bike	4/18/2020 12:07:00	4/18/2020 12:12:00	Green St & Madison St
25	B5BBCA72F5A8BE3C	docked_bike	5/4/2020 14:29:00	5/4/2020 14:44:00	Ravenswood Ave & Lawren
26	F65BB1A0FBE1A613	docked_bike	5/4/2020 13:44:00	5/4/2020 13:58:00	Southport Ave & Waveland

Image 3: Corrected date-time format

2.2.2 Managing missing data

Our dataset consists of **84777 rows**. As a next step, we had to examine whether our dataset had empty cells. We used the **google sheets** filtering tool and we found out that the **start_station_name ,start_station_id, end_lat** and **end_lng** columns had some empty cells. The missing cells are shown below.

	E	F	G	H	I	J	K	L
1	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng
1003	Wells St & Concord Ln	289			419,121	-876,347		
1866	Racine Ave & Wrightwood Ave	343			419,289	-87,659		
2169	Racine Ave & 18th St	15			418,582	-876,565		
2460	Morgan Ave & 14th Pl	137			418,624	-876,511		
3836	Lake Shore Dr & Wellington Ave	157			419,367	-876,368		
5102	Ashland Ave & Chicago Ave	350			41,896	-876,677		
5796	Wells St & Huron St	53			418,947	-876,344		
7253	State St & Pearson St	106			418,974	-876,287		
7406	Bissell St & Armitage Ave	113			419,184	-876,522		
9036	Central Park Ave & North Ave	508			419,097	-877,166		
9553	Phillips Ave & 79th St	579			417,518	-875,652		
9580	Wells St & Evergreen Ave	291			419,067	-876,348		
9781	Lincoln Ave & Waveland Ave	257			419,488	-876,753		
10127	Ashland Ave & Wrightwood Ave	166			419,288	-876,685		
10616	Lincoln Ave & Belmont Ave	131			419,394	-876,684		
11460	Clarendon Ave & Gordon Ter	312			419,579	-876,495		
11571	Sheffield Ave & Waveland Ave	114			419,494	-876,545		
13855	McClurg Ct & Erie St	142			418,945	-876,179		
13957	Clarendon Ave & Gordon Ter	312			419,579	-876,495		
14329	McClurg Ct & Erie St	142			418,945	-876,179		
15917	Sedgwick St & North Ave	118			419,114	-876,387		
16635	Clark St & Schiller St	301			41,908	-876,315		
16699	Museum of Science and Industry	424			417,917	-875,839		
17102	Sedgwick St & Huron St	111			418,947	-876,384		
17278	Dearborn Pkwy & Delaware Pl	140			41,899	-876,299		

Image 4: Missing Values

We had to decide how to manage the rows that contained the empty cells. One option was to completely discard them. These rows are only 101 of the total 84777 in the dataset. As we can see, they are only a tiny percentage. In addition, the missing information is not critical for the analysis that will follow. For the above reasons, we decided to keep these rows in our data set.

2.2.3 Formatting the latitude and longitude columns

Another major problem we encountered was at the **start_lat**, **start_lng**, **end_lat**, **end_lng** columns. First of all, we used the **search and replace tool to replace all the "," symbols with the "."** symbol in order to have decimal numbers. After this intervention the columns had the form that is shown in the photo below.

I	J	K	L
418.708	-876.257	418.944	-876.227
419.902	-876.934	419.824	-877.089
419.106	-876.494	41.968	-87.65
419.617	-876.546	419.401	-876.455
419.579	-876.495	419.942	-876.894
419.183	-876.363	419.402	-87.653
418.916	-876.484	418.834	-876.412
418.946	-876.534	418.946	-876.534
418.946	-876.534	418.946	-876.534
418.604	-876.258	418.969	-876.217
419.691	-876.742	419.522	-876.981
419.437	-87.649	41.968	-87.65
419.579	-876.495	419.695	-876.547
41.884	-876.247	418.949	-876.323
41.903	-876.313	419.295	-876.431
418.722	-876.615	419.102	-876.823
41.969	-87.696	41.969	-87.696
419.069	-876.262	419.069	-876.262
417.917	-875.839	417.917	-875.839
419.184	-876.522	419.201	-876.779
419.253	-876.658	418.834	-876.412
418.576	-876.615	418.576	-876.615
418.958	-876.259	419.007	-876.626
41.903	-876.313	418.822	-876.411
417.952	-875.807	417.993	-87.601
419.324	-876.527	419.126	-876.814

Image 5: Wrong latitude-longitude format

In order to validate the integrity of these numbers, we compared the values with the typical longitude and latitude values of Chicago City. These typical values are shown in the following table.

Chicago City Typical Coordinates In Decimal Number

Table 1: Chicago City Coordinates

Latitude	41.881832
Longitude	-87.623177

Every value in the latitude and longitude columns must be a close variation of the numbers above. As we can see, this was not the case. Although many cells had correct numbers, most of them had numbers that were one hundred times larger by absolute number value. Our dataset is 84777 rows long, so it was impossible to make the necessary corrections manually or by using Google Sheets. So **we downloaded the dataset as a CSV file called tripdata_cleaning_phase1 and wrote the following Python Script:**

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jan  8 23:30:33 2024

@author: jim47
"""
import pandas as pd
import numpy as np

""" Import the semi-cleaned dataset """
tripdata = pd.read_csv(r"D:\data analysis_2\CASE_STUDY\tripdata_cleaning_phase1.csv")

""" fix the wrong lat-long values"""
for i in range(84776):
    if tripdata.start_lat[i]>100 :
        tripdata.start_lat[i] = tripdata.start_lat[i]*0.1
    else:
        tripdata.start_lat[i] = tripdata.start_lat[i]*1

for i in range(84776):
    if tripdata.end_lat[i]>100 :
        tripdata.end_lat[i] = tripdata.end_lat[i]*0.1
    else:
        tripdata.end_lat[i] = tripdata.end_lat[i]*1

for i in range(84776):
   if tripdata.start_lng[i]<-100 :
      tripdata.start_lng[i] = tripdata.start_lng[i]*0.1
   else:
      tripdata.start_lng[i] = tripdata.start_lng[i]*1

for i in range(84776):
    if tripdata.end_lng[i]<-100 :
        tripdata.end_lng[i] = tripdata.end_lng[i]*0.1
    else:
        tripdata.end_lng[i] = tripdata.end_lng[i]*1

cleaned_tripdata = tripdata

cleaned_tripdata.to_csv(r"D:\data analysis_2\CASE_STUDY\tripdata_cleaning_phase2.csv")
```

Code chunk 1 (Python)

We then opened the new CSV file we produced, named tripdata**_cleaning_phase2.csv,** using **WPS Sheets.** Now every latitude and longitude format has the correct format.

H	I	J	K	L	M	N
end_station	start_lat	start_lng	end_lat	end_lng	member_casual	
152	41.8964	-87.661	41.9322	-87.6586	member	
499	41.9244	-87.7154	41.9306	-87.7238	member	
255	41.8945	-87.6179	41.8679	-87.623	member	
657	41.903	-87.6975	41.8992	-87.6722	member	
323	41.8902	-87.6262	41.9695	-87.6547	casual	
35	41.8969	-87.6217	41.8923	-87.612	member	
635	41.8923	-87.612	41.8957	-87.6201	member	
382	41.8665	-87.6847	41.8747	-87.6864	casual	
359	41.8949	-87.6323	41.9035	-87.6434	casual	
508	41.8846	-87.7063	41.9097	-87.7166	member	
374	41.9097	-87.7166	41.8984	-87.6866	member	
128	41.8984	-87.6866	41.8958	-87.6772	member	
210	41.8984	-87.6866	41.9035	-87.6677	member	
128	41.8984	-87.6866	41.8958	-87.6772	member	
623	41.8708	-87.6257	41.8728	-87.624	member	
20	41.8708	-87.6257	41.9105	-87.6531	member	
141	41.8708	-87.6257	41.9157	-87.6346	member	

Image 6: Corrected latitude - longitude format

2.3 Data manipulation

In order to make our following analysis more productive we added 2 more columns to our data set.

1. ride_length: We found the difference between the ended_at and started_at columns in minutes.
2. day_of_week: We used the Google Sheets' **WEEKDAY command** to find the day of the week that each one of the trips started. The days are represented by the following numbers **[1.Sunday, 2.Monday, 3.Tuesday, 4.Wensday, 5.Thursday, 6.Friday, 7. Saturday.]**

O	P
ride_length	day_of_week
0:27:00	1
0:09:00	6
0:14:00	7
0:12:00	7
0:53:00	7
0:06:00	5
0:05:00	3
1:16:00	7
0:05:00	4
0:17:00	7
0:24:00	7
0:05:00	7
0:09:00	6
0:04:00	4
0:24:00	2
0:35:00	7
0:22:00	1
0:08:00	7
0:16:00	7
0:06:00	7
0:36:00	2
0:32:00	1
0:05:00	7
0:15:00	2

Image 7 : New columns

As a last step, we found 99 rows, where the ride_length was not a valid positive numbers in minutes, due to errors at the started_at and ended_at date-time stamps. 99 rows represents just a tiny percentage of the complete dataset and we had not credible means to get the correct date-time stamps. For the above reasons, we decided to exclude these rows from our analysis.

After the data cleaning and manipulation process, we saved our primed dataset as a new CSV file called tripdata_CLEANED.This is the dataset on which we based our analysis.

3. Analysis

The biggest part of our analysis was made using **SQL programming and BigQuery Sandbox.** We imported the tripdata_CLEANED.csv file as TripData and created a table named trip_data

3.1 Analytical Process

3.1.1 Total trips

The first thing we wanted to see, was the total trips that had been made the annual members and the casual users the past 12 months. We used the query below to achieve this goal.

```
SELECT
  member_casual,
  COUNT (*) AS total_rides
FROM
  `bike-share-case-study-410714.TripData.trip_data`
GROUP BY
  member_casual
```

Code chunk 2 (SQL)

We had the following results:

Table 2: Total Trips per member type

User type	Total Trips
Member	61128
Casual	23589

We can easily see that the trips made by **Annual Members are over 2.5 times more than the trips made by Casual Users.**

3.1.2 Average Trip Duration

With this analysis, we aim to find differences on how the company's services are used, between the Annual members and the Casual Users. A critical way to establish differences between the two groups is to examine the average trip duration of each group. We examined the dataset using the following query:

```sql
WITH
  trip_duration_in_minutes AS (
  SELECT
    ride_id,
    started_at,
    ended_at,
    member_casual,
    ride_length,
    TIMESTAMP_DIFF(ended_at, started_at,MINUTE) AS minutes
  FROM
    `bike-share-case-study-410714.TripData.trip_data`)
SELECT
  member_casual,
  AVG(minutes) AS average_trip_duration,
FROM
  trip_duration_in_minutes
GROUP BY
  member_casual
```

Code chunk 3 (SQL)

By running this query we got the following results:

Table 3: Average Trip Duration per user type

User type	Average Trip Duration in minutes
Member	~61
Casual	~309

We can see that, even thought the trips made by Casual Users are by far less, their average duration is 5 times bigger than the average duration of the trips that are made by Annual Members.

3.1.3 Trips per day

In our pursuit to identify trends in how the two groups use the company's services, we decided to examine how many trips were made each day of the week, by the two user groups. We also wanted to identify the percentage of the trips that are made by the Annual Members and the Casual Users, for each day of the week. We used the following query.

```sql
WITH
 rides_in_specific_day_and_memebertype AS (
 SELECT
  member_casual,
  COUNT (*) AS total_rides_per_day_and_type,
  day_of_week
```

```sql
 FROM
  `bike-share-case-study-410714.TripData.trip_data`
 GROUP BY
  member_casual,
  day_of_week),
 rides_per_specific_day AS (
 SELECT
  COUNT(*) AS total_rides_per_day,
  day_of_week
 FROM
  `bike-share-case-study-410714.TripData.trip_data`
 GROUP BY
  day_of_week)
SELECT
 rides_in_specific_day_and_memebertype.member_casual,
 rides_in_specific_day_and_memebertype.day_of_week,
 rides_in_specific_day_and_memebertype.total_rides_per_day_and_type,
 rides_per_specific_day.total_rides_per_day,
 (rides_in_specific_day_and_memebertype.total_rides_per_day_and_type /
rides_per_specific_day.total_rides_per_day)*100 AS ride_percentage
FROM
 rides_in_specific_day_and_memebertype
LEFT JOIN
 rides_per_specific_day
ON
 rides_in_specific_day_and_memebertype.day_of_week =
rides_per_specific_day.day_of_week
ORDER BY
 rides_in_specific_day_and_memebertype.day_of_week
```

Code chunk 4 (SQL)

This query, after the right formatting gave us the following results:

Member Type	Day of Week	Total Trips per User Type and Day	Total Trips Per Day	Trip Percentage (%)
member	Mon	8110	11203	72
casual	Mon	3093	11203	28
casual	Tue	2991	12595	24
member	Tue	9604	12595	76
casual	Wen	3258	12040	27
member	Wen	8782	12040	73
casual	Tue	1967	9539	21
member	Tue	7572	9539	79
casual	Fri	2594	9336	28
member	Fri	6742	9336	72
casual	Sat	4772	15926	30
member	Sat	11154	15926	70
casual	Sun	4914	14078	35
member	Sun	9164	14078	65

Image 8: Total trips and trips percentage throughout the week

These results lead to some very interesting insights that we will examine later in these report.

3.1.4 Top Start Stations

Another key step of our analysis was to examine the most popular start stations that are selected by the two individual user groups. Our goal was to identify different areas of interest in the city, that may show different trends for the two groups. We choose to examine the 50 top start stations for the the two groups and find their location in the city. We found the start stations with the following query.

```
WITH
 start_count_top10 AS (
 SELECT
  start_station_name,
  COUNT(*) AS start_count
 FROM
  `bike-share-case-study-410714.TripData.trip_data`
 WHERE
  member_casual = "member"
 GROUP BY
  start_station_name
 ORDER BY
  start_count DESC
 LIMIT
  50),
 start_lat_lng AS (
 SELECT
  DISTINCT start_station_name,
  start_lat,
  start_lng
```

```sql
  FROM
   `bike-share-case-study-410714.TripData.trip_data`)
SELECT
  start_count_top10.start_station_name,
  start_count_top10.start_count,
  start_lat_lng.start_lat,
  start_lat_lng.start_lng
FROM
  start_count_top10
JOIN
  start_lat_lng
ON
  start_count_top10.start_station_name = start_lat_lng.start_station_name
```

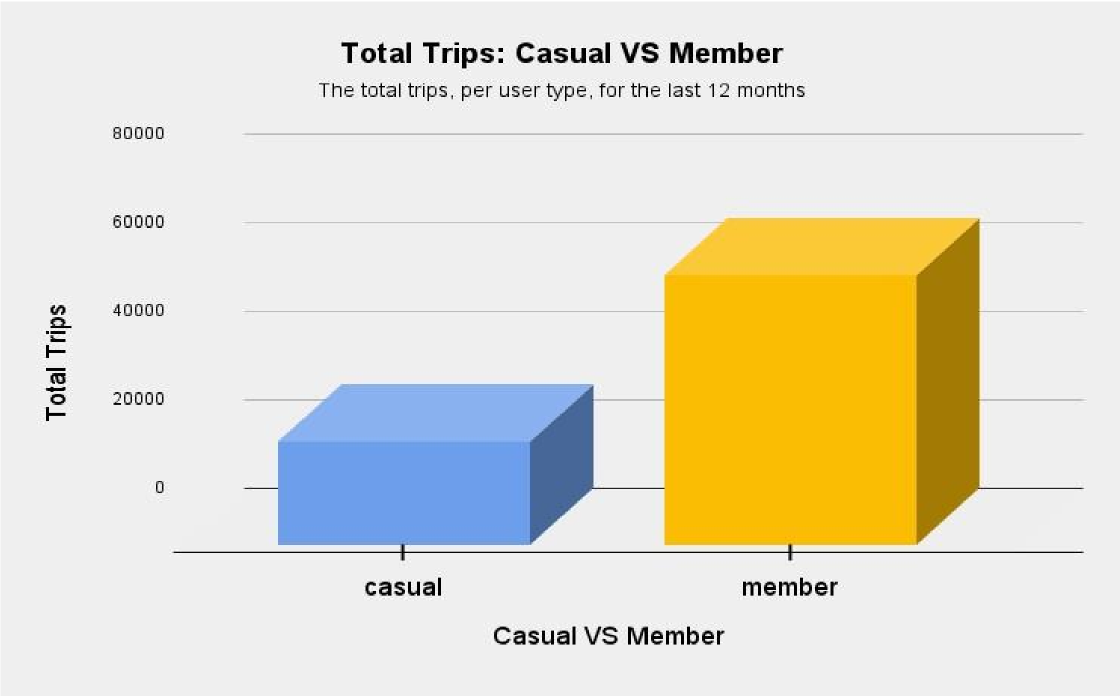
Code chunk 5(SQL)

The results will be shown in the following sections of this report using an interesting visualization.

4.Key Insights

In this part of our report we will present some key insights of our analysis backed with some comprehensive visuals.

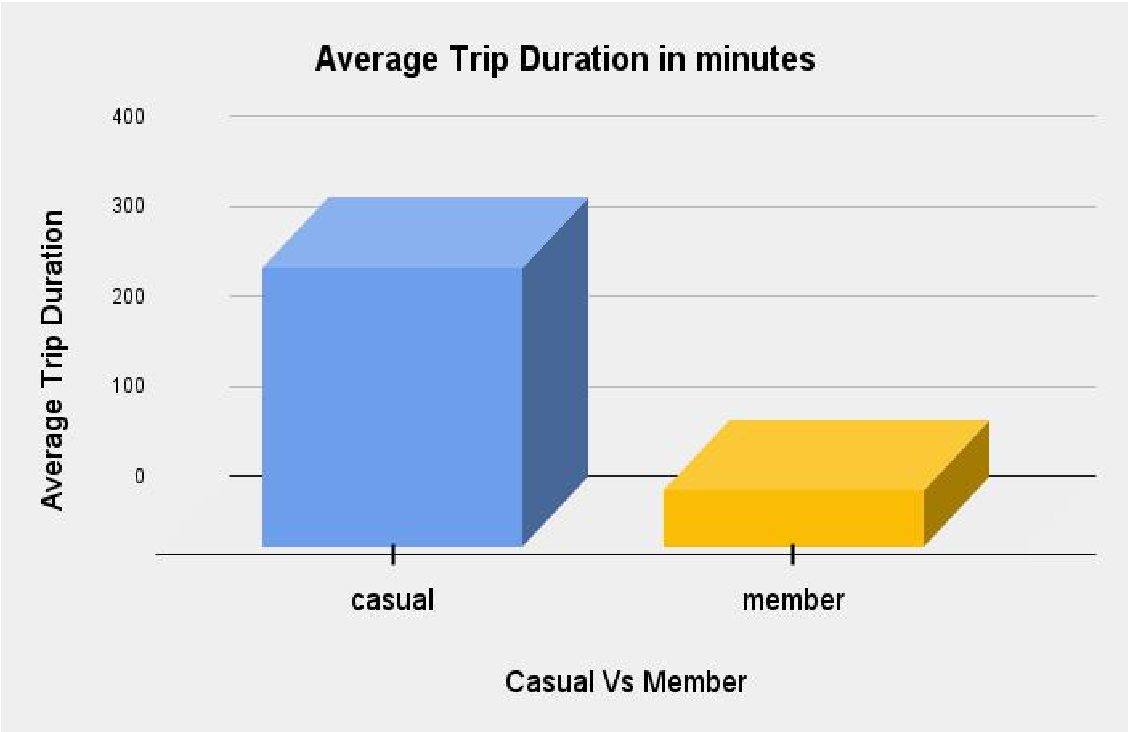
The first useful insight is about the total trips that are made by the each of the the two user groups. This difference is clearly displayed in the graph below.



Graph 1: Total Trips: Casual VS Member

<u>Insight 1:</u> The annual members have made over 2.5 times more trips than the casual users. These numbers may indicate that annual members use the company's bikes,primarily for their everyday transports.

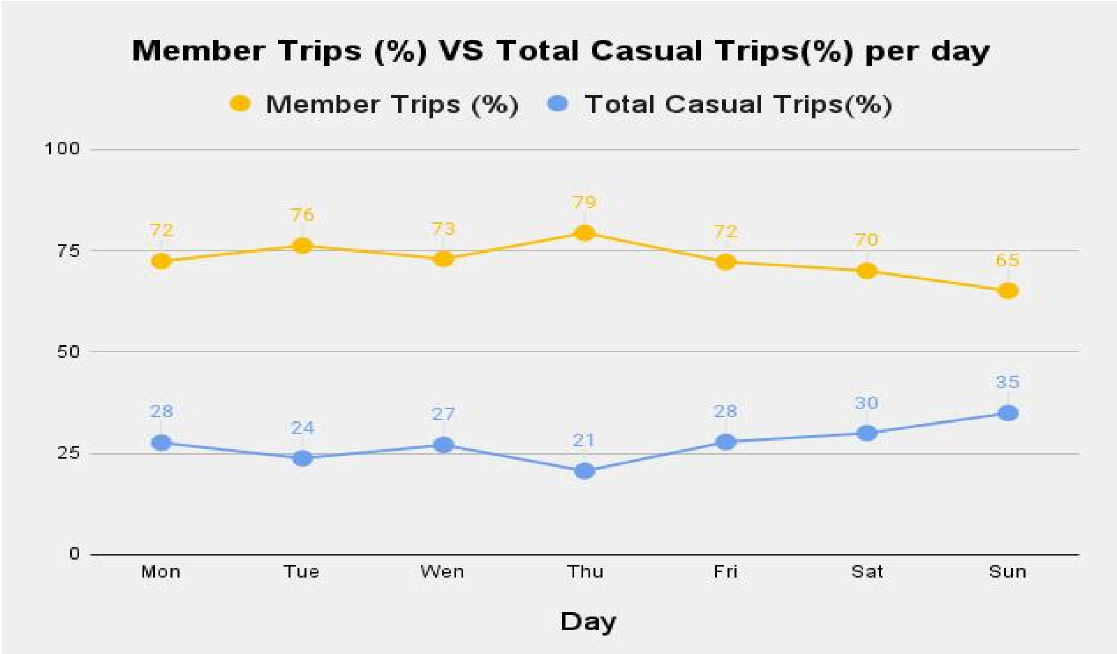
We then moved on to examining the average trip duration for the two groups, for the past 12 months. This analysis gave us a completely different reality. As we can see bellow, the average trip duration for the casual users is much larger than the average trip duration for the annual members.



Graph 2: Average Trip Duration in minutes

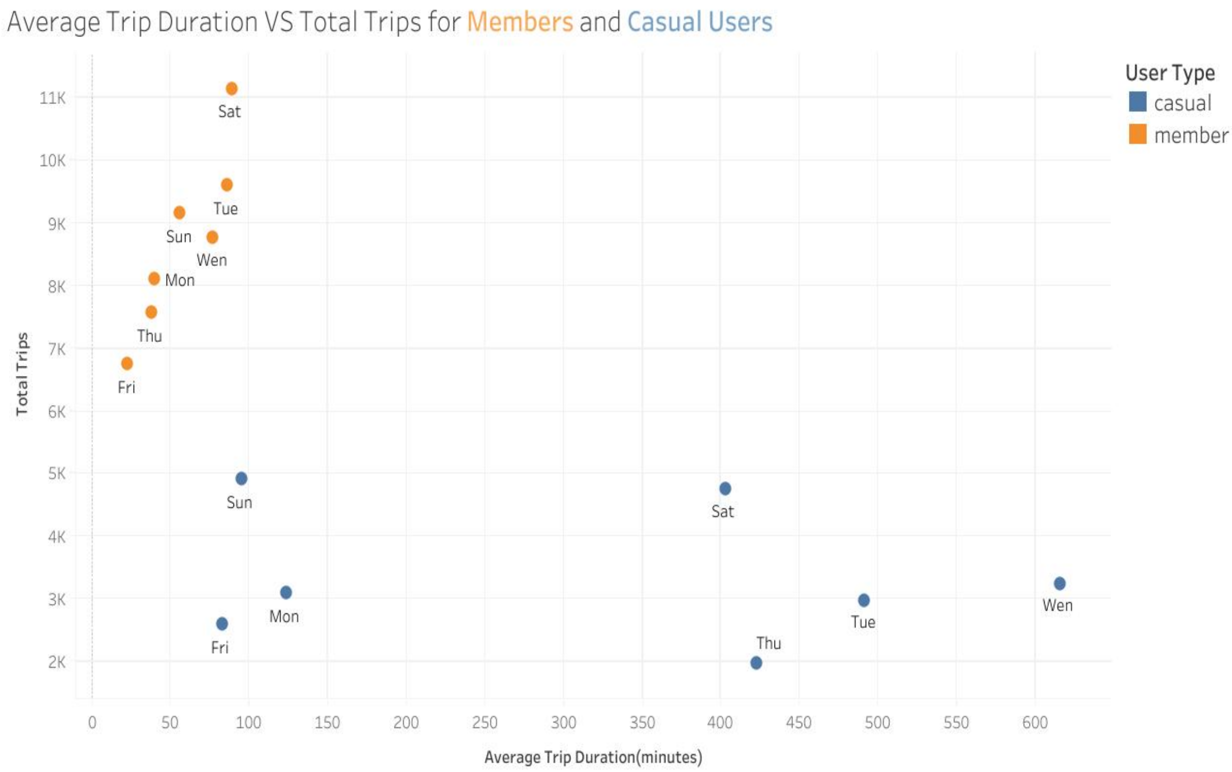
<u>Insight 2:</u>The average trip of the casual user lasts 5 times longer than the one of the annual member. This is a strong indication that casual members use the bike share service for long rides, for leisure and entertainment purposes.

We also wanted to know, how the percentage composition of trips change throughout the week.



Graph 3: Member Trips percentage VS Total Casual Trips percentage per day

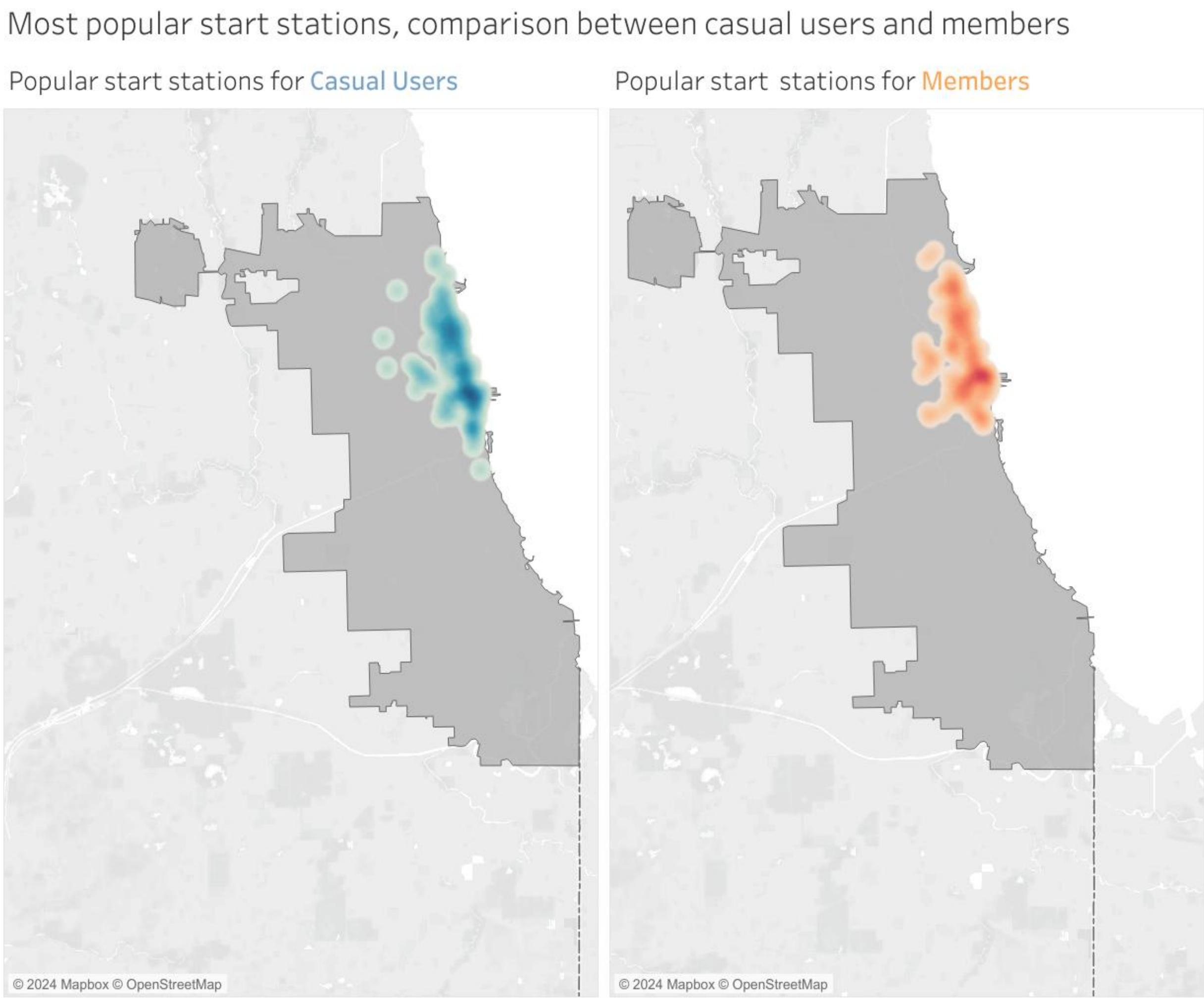
<u>Insight 3:</u> On weekends and especially on Sunday, the casual users' trips represent a much larger percentage of the total trips in contrast with what happens midweek. The above information can be summarized in the following graph, that enable us to extract some useful trends that govern the habits of the two user groups.



Graph 4: Average Trip Duration VS Total Trips for Members and casual Users

<u>Summarized insight</u>: As we can see in the the graph above, the two user groups have major differences in their habits. <u>The annual members make many trips that last a relatively small amount of time.</u> These data indicate that annual members use the company's bikes,mainly, for everyday transports such as commuting to work or going for their groceries. This is also backed by the fact that we do not see considerable differences in the average trip duration during the week. <u>On the other hand, casual users made a lot less trips, but for the better part of an average week, their trips last a lot longer.</u> A longer bike trip has more probabilities to be a leisure activity than a way to complete an everyday task. This observation is also supported by the fact, that we see large differences in trip duration throughout the week.

As a last step, we investigated the start stations that each of the user groups prefer to use. We extracted the top 50 start stations for each group and created the following heat map, based on the city of Chicago.



Graph 5: Most popular start stations,comparison between casual users and members

<u>Insight 4:</u>We can see that the two user groups select a similar general location to start their bike trips. A close look at the heat maps reveals, that the points of interest for the casual users are not that much concentrated. <u>On the contrary, they are spread over a wider area in the City of Chicago. This observation may show a tendency for city exploration. Yet another fact that supports the theory that the casual users use the bikes for entertainment purposes .</u>

5.Data Driven Recommendations

The main business goal is to convert casual users to annual members. A potential effective way to achieve this goal is to offer annual membership perks and bonuses that are tailor made for the habits of the average casual user. Some of our top recommendations are the following.

1. Bonus and perks that rewards longer trips: As we saw, casual users tend to make longer bike trips. The company can offer bonuses when the user makes trips that exceed a specified period of time. For example, if a member makes many long trips, gets a discount in the next year membership.

2.Rewarding city exploration: We established that the majority of the casual users select a larger variety of areas to start their bike trips. The company can offer bonuses for each user that visits over a specified number of bike stations. This idea could be also be supported with a social media page, where users could upload content of their bike exploration of Chicago.

3.Focus on specific days: If a casual member can not be persuaded to buy the standard annual membership, the company could offer them slightly cheaper annual memberships, that grants unlimited rides only for specific days. The company can focus on days such as Saturday and Sunday, when the percentage of the rides made by casual users is higher.