BIKE-SHARE WITH DATA SCIENCE

JHOSEP A. BAILON VELASQUEZ

10-31-2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data Collection with BigQuery
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Analytics conclusions

Introduction

Project background and 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 geotracked and locked into a network of 692 stations across Chicago. 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

Desire Outcomes

- Understand how annual members and casual riders' behavior
- Why casual riders would buy a membership?

Problem:

Maximize the number of annual members

Solution:

Help to design a **marketing strategy** to convert casual riders to annual members

Methodology

Section 1

Methodology

- Executive Summary
- Data collection methodology:
  - Data is public and was extracted from Index of Bucket "divvy-tripdata“
- Perform data wrangling
  - Data was pre-processing using SQL in BigQuery, adding new columns in the process
- Perform exploratory data analysis (EDA) using visualization and SQL

Data Collection

- The data was extracted from an online repository (https://divvy-tripdata.s3.amazonaws.com/index.html)
- Google Cloud and Vertex AI were used to manage data and create a notebook

Index of bucket "divvy-tripdata"

Name	Date Modified	Size	Type
202004-divvy-tripdata.zip	Jun 1st 2020, 09:50:06 am	3.32 MB	ZIP file
202005-divvy-tripdata.zip	Jun 1st 2020, 09:50:09 am	7.99 MB	ZIP file
202006-divvy-tripdata.zip	Jul 5th 2020, 07:31:49 pm	14.73 MB	ZIP file
202007-divvy-tripdata.zip	Aug 11th 2020, 09:10:49 pm	23.62 MB	ZIP file
202008-divvy-tripdata.zip	Sep 4th 2020, 10:11:40 am	27.86 MB	ZIP file
202009-divvy-tripdata.zip	Oct 13th 2020, 03:06:37 pm	24.34 MB	ZIP file
202010-divvy-tripdata.zip	Nov 4th 2020, 08:17:21 am	17.86 MB	ZIP file
202011-divvy-tripdata.zip	Dec 4th 2020, 05:32:44 pm	11.67 MB	ZIP file
202012-divvy-tripdata.zip	Jan 5th 2021, 08:56:54 am	4.84 MB	ZIP file
202101-divvy-tripdata.zip	Feb 4th 2021, 04:52:59 pm	3.66 MB	ZIP file
202102-divvy-tripdata.zip	Mar 9th 2021, 07:03:24 pm	1.91 MB	ZIP file
202103-divvy-tripdata.zip	Apr 8th 2021, 09:28:53 am	8.02 MB	ZIP file
202104-divvy-tripdata.zip	May 7th 2021, 09:52:05 am	11.78 MB	ZIP file
202105-divvy-tripdata.zip	Jun 11th 2021, 12:10:18 pm	18.89 MB	ZIP file
202106-divvy-tripdata.zip	Jul 15th 2021, 06:22:05 pm	26.52 MB	ZIP file

Data Collection

- Union function was used to join all tables from the last year
- The link to the notebook: https://github.com/Jhosep14/Google_Capstone_Share/blob/main/Cyclistic%20data.ipynb

RUN SAVE SHARE SCHEDULE MORE

```sql
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2021_08
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2021_09
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2021_10
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2021_11
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2021_12
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_01
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_02
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_03
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_04
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_05
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_06
UNION ALL
SELECT * FROM personal-projects-351015.Cyclistic_data.tripdata_2022_07
```

Data Wrangling

- Data was processed for checking null and invalid values and drop unnecessary columns
- The link to the notebook: https://github.com/Jhosep14/Google_Capstone_Share/blob/main/Cyclistic%20data.ipynb

```
df.info(verbose=False)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5901463 entries, 0 to 5901462
Columns: 15 entries, ride_id to ride_length
dtypes: datetime64[ns, UTC](2), float64(4), int64(2), object(7)
memory usage: 675.4+ MB
```

[7]:
```
#Dropping unnecessary columns for analysis
df.drop(['start_lat', 'start_lng', 'end_lat', 'end_lng', 'start_station_name','start_station_id','end_station_name','end_station_id'], axis=1, inplace=True
df.head()
```

[7]:

	ride_id	rideable_type	started_at	ended_at	member_casual	day_of_week	ride_length
0	5D66026D9DCAA2B6	classic_bike	2022-07-17 09:03:00+00:00	2022-07-17 09:10:00+00:00	casual	1	7
1	A19A2D6794E7324B	classic_bike	2022-07-24 09:30:00+00:00	2022-07-24 09:31:00+00:00	member	1	1
2	BB3FF2254B168ADA	classic_bike	2022-07-10 01:25:00+00:00	2022-07-10 01:42:00+00:00	member	1	17
3	60DEC72D587DDF27	classic_bike	2022-07-10 15:11:00+00:00	2022-07-10 15:30:00+00:00	casual	1	19
4	3A46BD2E72254D25	classic_bike	2022-07-10 18:58:00+00:00	2022-07-10 18:59:00+00:00	casual	1	1

[8]:
```
df.isnull().sum()
```

[8]:
```
ride_id          0
rideable_type    0
started_at       0
ended_at         0
member_casual    0
day_of_week      0
ride_length      0
```

EDA with Data Visualization

- The ride length of each type of rider was calculated using SQL
- Day and Month was extracted from date columns to see the distribution of riders
- Rider's type amount was compared to contrast with rides behavior
- Notebook: https://github.com/Jhosep14/Google_Capstone_Share/blob/main/Cyclistic%20data.ipynb

EDA Results

- The ride length average in casual riders is greater than annual member
- The number of casual riders per day is greater than annual members
- Notebook: https://github.com/Jhosep14/Google_Capstone_Share/blob/main/Cyclistic%20data.ipynb

```
%%bigquery
SELECT member_casual,AVG(ride_length) AS ride_average
FROM `personal-projects-351015.Cyclistic_data.full_annual_tripdata`
GROUP BY member_casual
ORDER BY ride_average DESC
```

```
Query complete after 0.01s: 100%|██████████| 1/1 [00:00<00:00, 770.45query/s]
Downloading: 100%|██████████| 2/2 [00:01<00:00,  1.43rows/s]
```

	member_casual	ride_average
0	casual	29.211727
1	member	12.932331

```
df.groupby(['day_of_week'])['member_casual'].value_counts(ascending=False)
```

```
day_of_week  member_casual
1            casual           380729
             member           328498
2            member           377494
             casual           229314
3            member           419981
             casual           204581
4            member           418796
             casual           210696
5            member           414455
             casual           237586
6            member           365123
             casual           263751
7            casual           422715
             member           355511
Name: member_casual, dtype: int64
```

EDA Results

- The number of rides by annual member were greater than casual rider
- A new column was added to find out the number of rides per month

```
riders = """
SELECT member_casual, COUNT(member_casual) AS TOTAL
FROM `personal-projects-351015.Cyclistic_data.full_annual_tripdata`
GROUP BY member_casual
ORDER BY TOTAL DESC"""
total = client.query(riders).to_dataframe()
total_riders = total.set_index('member_casual')
total_riders
```

	TOTAL
member_casual	
member	3379237
casual	2522226

```
df['month'] = df['started_at'].dt.strftime('%m')
df.month
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: Setting
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
  """Entry point for launching an IPython kernel.
```

```
0          07
1          07
2          07
3          07
4          07
           ..
5901182    05
5901229    05
5901263    07
5901264    12
5901280    05
Name: month, Length: 4629230, dtype: object
```

EDA Results

- The average of length by casual riders were greater than annual members throughout the weekdays

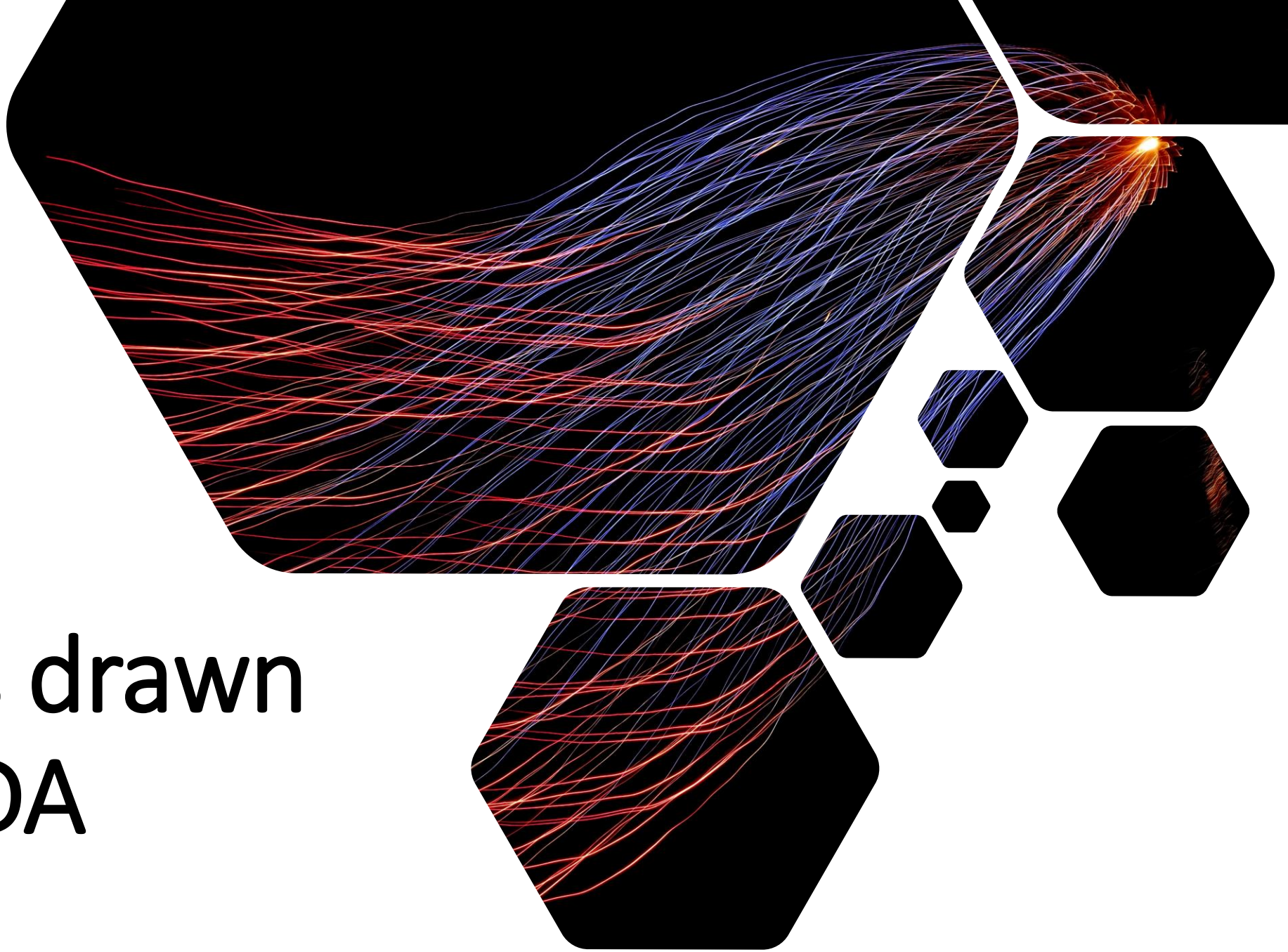
```
df['length'] = df['ride_length'].abs()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWar
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  """Entry point for launching an IPython kernel.
```

```
ride_length = df.groupby(['day_of_week','member_casual'])['ride_length'].mean()
ride_length
```

```
day_of_week  member_casual
1            casual          30.621261
             member          14.340273
2            casual          27.663706
             member          12.216255
3            casual          23.200023
             member          11.751639
4            casual          22.807134
             member          11.888798
5            casual          23.410542
             member          12.063486
6            casual          24.661431
             member          12.254840
7            casual          29.039464
             member          14.223841
Name: ride_length, dtype: float64
```

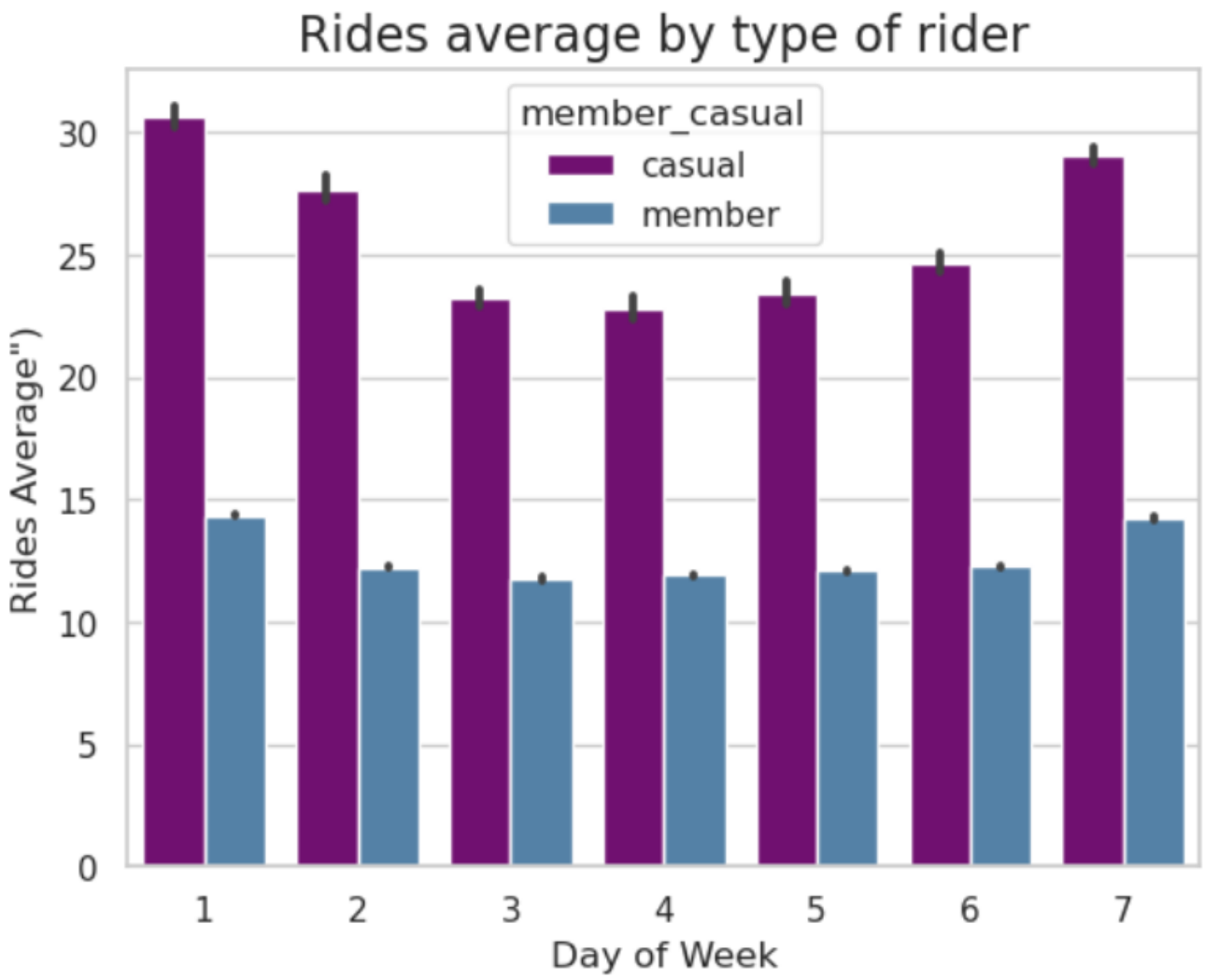


Section 2

Insights drawn from EDA

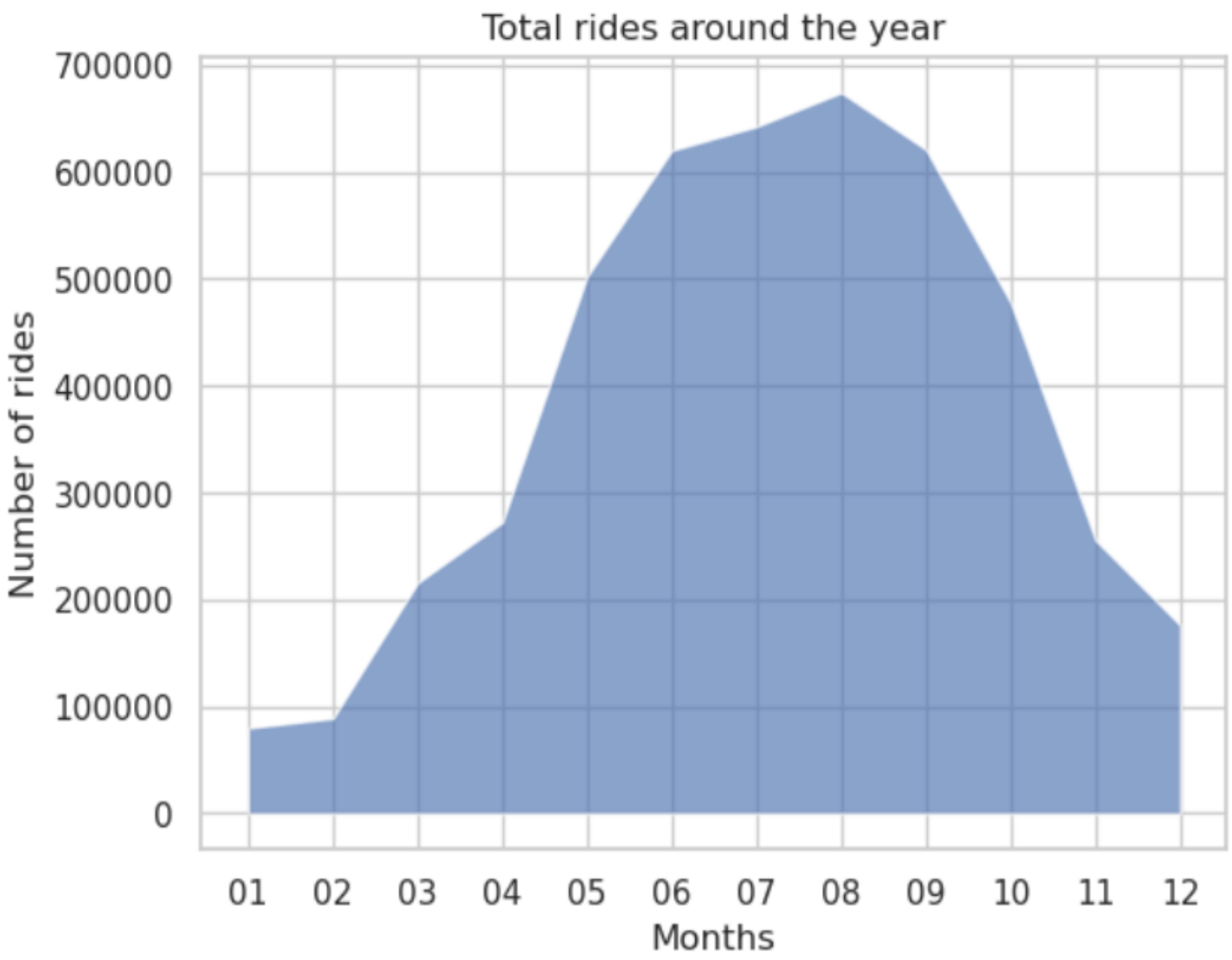
Rides Average for each Day of Week

- Casual riders tend to use bikes share services almost twice time more compared with member riders



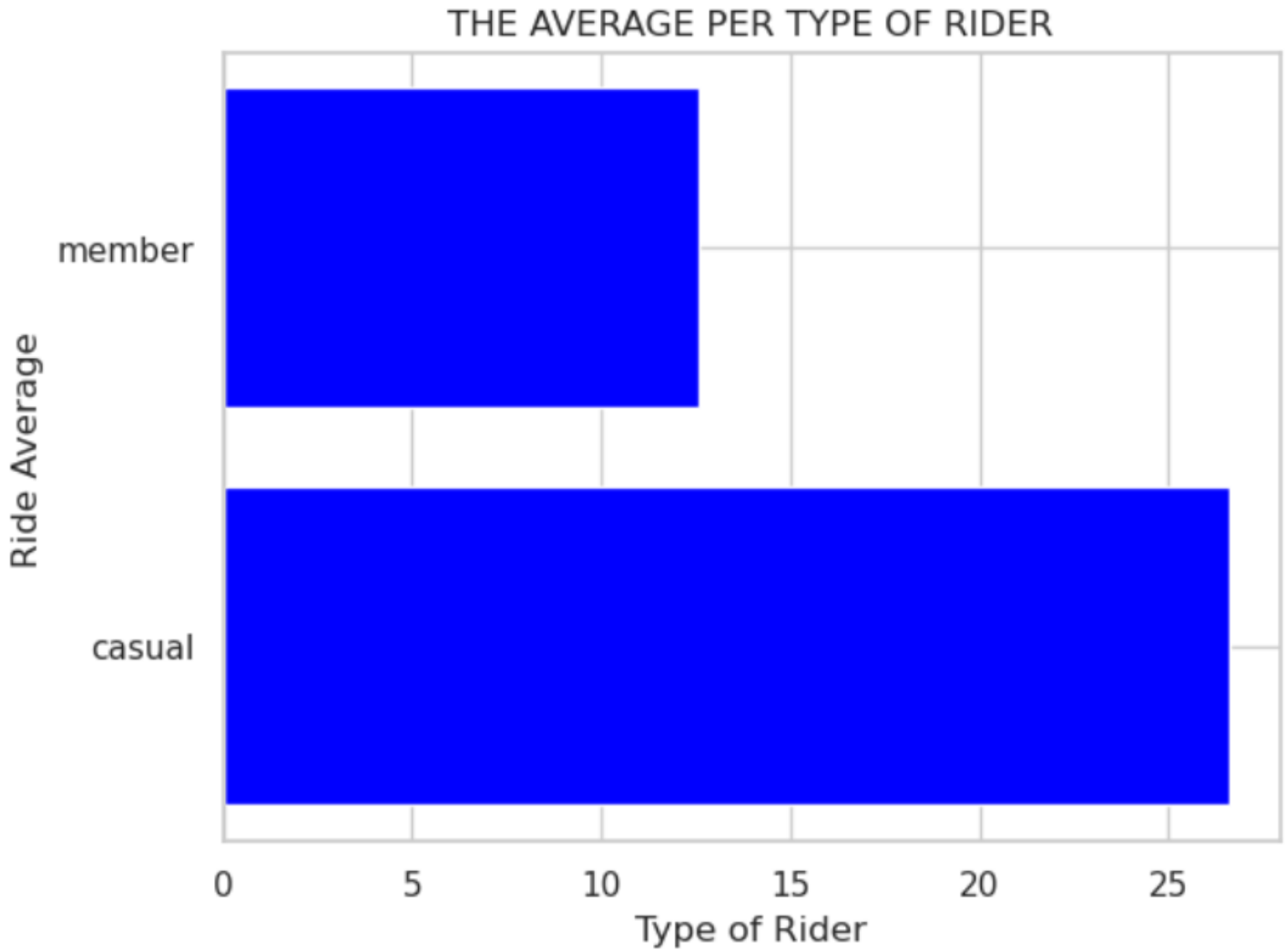
Number of Rides in The Last Year

- There is a huge increase of the number of rides passing the middle of the year



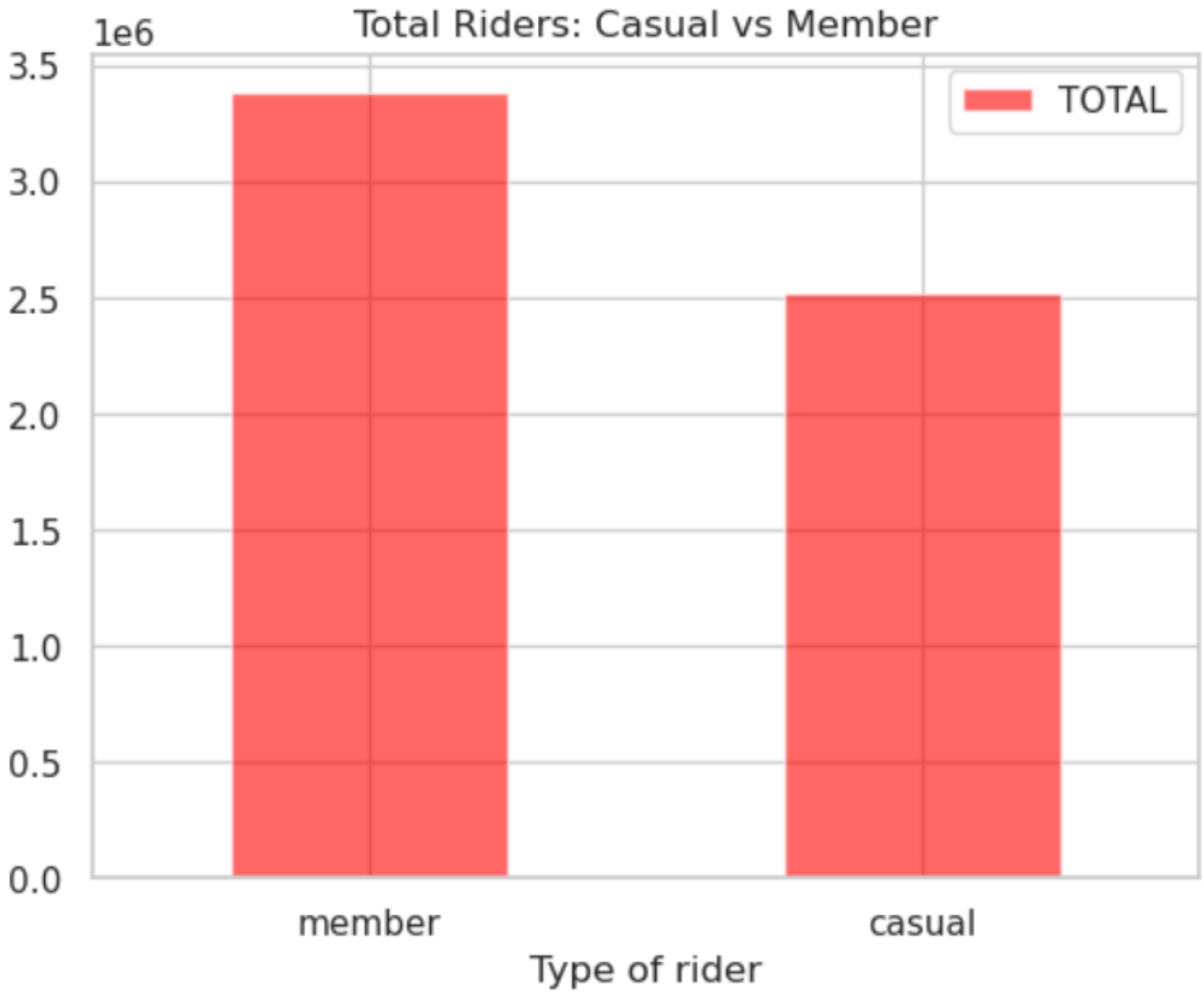
Ride Average vs. Type of Rider

- Casual riders tend to use bikes more time than annual members



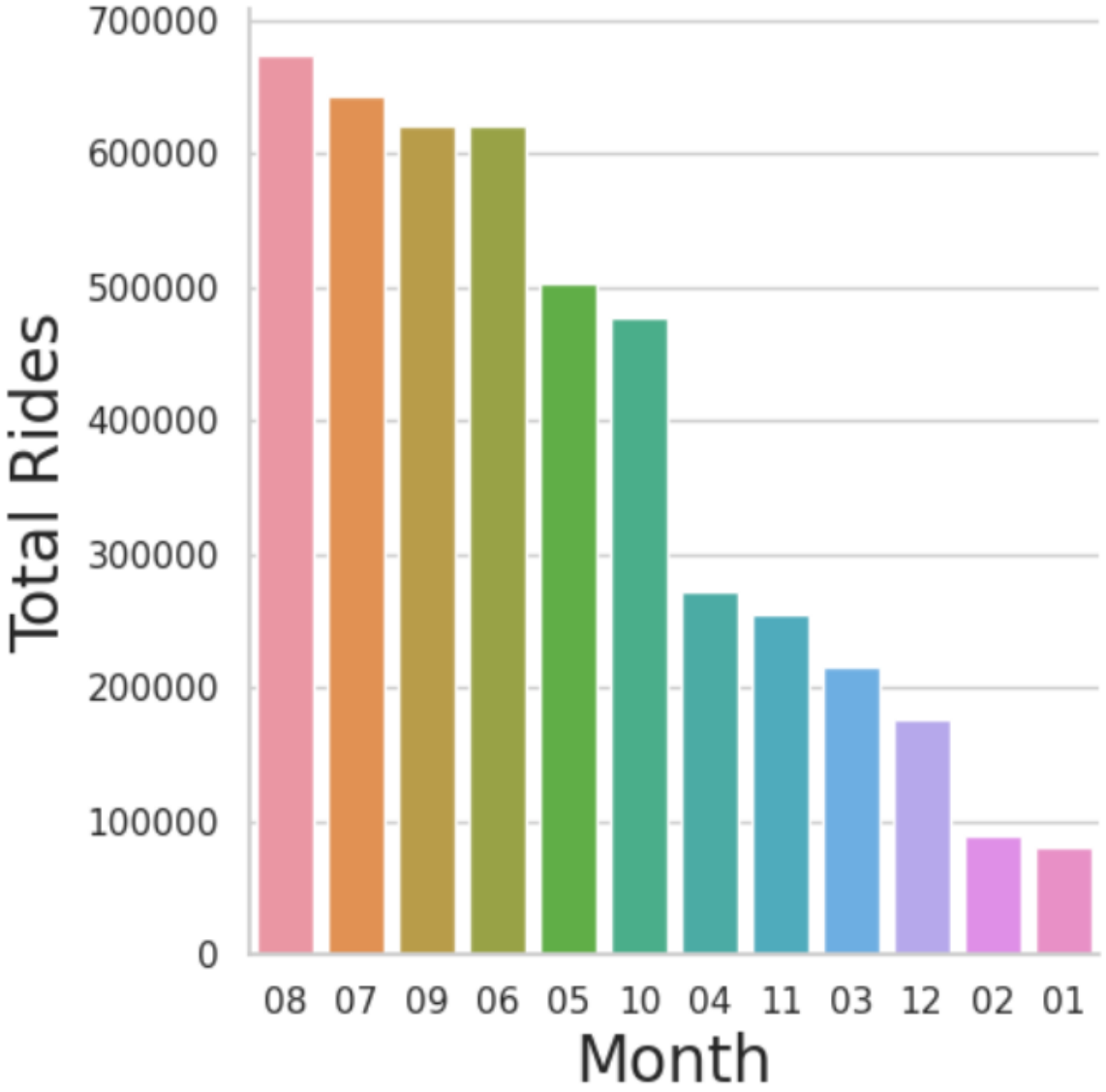
Members vs. Casual Riders

- The number of casual riders are lower than annual members
- There are a great range where to convert these casual riders to annual members



Monthly Rides

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



Section 3

Recommendations

Results:

- Casual riders represent a huge percentage of total riders
- The marketing strategy has to focus on months with more rides (July - October)
- Rides length throughout days and months is always higher in casual riders than in annual members

Thank you!