I am a General Accountant and just switched to Data Analyst. I want to upload small project that is case study of Ultimate course in Google Data Analytics.

# Scenario

In this case, I am a junior data analyst working in 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, my team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, my 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. I will present my analysis to the company's executive team and give recommendations for the marketing strategy.

**Products**

Bike-share offering: The bikes can be unlocked from one station and returned to any other station in the system anytime.
Types of bike : Classic, Electric, Docked
Flexibles pricing : single-ride passes, full-day passes, and annual memberships.

**Customers**

Casual rider : purchase single-ride or full-day passes
Annual membership: subcription

# *Phase 1. Ask*

Goal of Step: Define the problem and business task of this Case Study and the expectation of stakeholders.

Three questions will guide the future marketing program:
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?

The first question to answer: How do annual members and casual riders use Cyclistic bikes differently?

I will produce a report with the following deliverables:

1. A clear statement of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of your analysis
5. Supporting visualizations and key findings
6. Your top recommendations based on your analysis

## *Phase 2. Prepare*

Data sourse: https://divvy-tripdata.s3.amazonaws.com/index.html

Start with: Think what data need, read dataframe what data had, decide which data will be used, how the data will be organized, and what limitations in the data exist in our efforts to answer the business task and questions.

Have 43 zip files in online source. However, the data tables store heterogeneous attributes, so the analysis will be based on data previous 12 months.

Data tools: Jupyter with pandas and pyspark.

Installed & Loaded Packages

In [1]:
```
import findspark
from pyspark.sql import SparkSession
import pandas as pd
import pyspark.sql.functions as f
from pyspark.sql.types import DateType
# from pyspark.sql.functions import col, substring, dayofweek, date_format
from pyspark.sql.types import IntegerType
import dask.dataframe as dd
import os
import glob
from pyspark.sql.functions import *
```

In [2]:
```
#Create SparkSession
spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()
```

```
22/05/30 15:26:52 WARN Utils: Your hostname, Mis-MacBook-Pro.local resolves to a loopback address: 127.0.0.1; using 1
22/05/30 15:26:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/05/30 15:26:53 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
22/05/30 15:26:53 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
```

Import data

In [3]:
```
# load data spark
df_trip = spark.read.csv("tripdata_*.csv",header=True)
```

In [4]:
```
# load data pandas
pd_trip = dd.read_csv("/Users/minguyen/tripdata*.csv")
```

# *Phase 3. Process*

Verify Data: Make sure data was imported correctly and look for errors

In [7]:
```
df_trip.show()
```

```
+----------------+-------------+-------------------+-------------------+------------------+----------------+---------
|         ride_id|rideable_type|         started_at|           ended_at|start_station_name|start_station_id|    end_s
+----------------+-------------+-------------------+-------------------+------------------+----------------+---------
|99FEC93BA843FB20|electric_bike|2021-06-13 14:31:28|2021-06-13 14:34:11|              null|            null|
|06048DCFC8520CAF|electric_bike|2021-06-04 11:18:02|2021-06-04 11:24:19|              null|            null|
|9598066F68045DF2|electric_bike|2021-06-04 09:49:35|2021-06-04 09:55:34|              null|            null|
|B03C0FE48C412214|electric_bike|2021-06-03 19:56:05|2021-06-03 20:21:55|              null|            null|
|B9EEA89F8FEE73B7|electric_bike|2021-06-04 14:05:51|2021-06-04 14:09:59|              null|            null|
|62B943CEAAA420BA|electric_bike|2021-06-03 19:32:01|2021-06-03 19:38:46|              null|            null|
|7E2546FBA79C46EE|electric_bike|2021-06-10 16:30:10|2021-06-10 16:36:21|              null|            null|
|3DDF3BBF6C4C3C89|electric_bike|2021-06-10 17:00:30|2021-06-10 17:06:48|              null|            null|
|2608805637155AB6|electric_bike|2021-06-10 12:46:16|2021-06-10 12:55:02|              null|            null|
|AF529C946F28ED42|electric_bike|2021-06-23 17:57:29|2021-06-23 18:06:40|              null|            null|Michigan
|E6010941FB92E4A6|electric_bike|2021-06-22 19:28:02|2021-06-22 19:39:48|              null|            null|
|1149C0723F7AFFD5|electric_bike|2021-06-29 17:35:49|2021-06-29 17:55:11|              null|            null|
|8762DB62099E6011|electric_bike|2021-06-05 14:55:05|2021-06-05 15:13:29|              null|            null|
|BE3AC77CBFF17E6A|electric_bike|2021-06-05 14:05:00|2021-06-05 14:09:01|              null|            null|
|8E9F2CB0893B96A0|electric_bike|2021-06-05 13:39:04|2021-06-05 13:57:21|              null|            null|
|6344B71B7BB6E09E|electric_bike|2021-06-22 18:52:53|2021-06-22 18:59:13|              null|            null|
|59CE9444E2ED2530|electric_bike|2021-06-02 10:30:11|2021-06-02 10:37:03|              null|            null|
|2D6929277855EBE5|electric_bike|2021-06-08 13:49:03|2021-06-08 13:53:01|              null|            null|
|F7107122A837A50B|electric_bike|2021-06-08 18:31:31|2021-06-08 18:38:25|              null|            null|
|45ABF9231CC02E3C|electric_bike|2021-06-07 22:24:08|2021-06-07 22:35:25|              null|            null|
+----------------+-------------+-------------------+-------------------+------------------+----------------+---------
only showing top 20 rows
```

In [8]:
```
df_trip.printSchema()
```

```
root
 |-- ride_id: string (nullable = true)
 |-- rideable_type: string (nullable = true)
 |-- started_at: string (nullable = true)
 |-- ended_at: string (nullable = true)
 |-- start_station_name: string (nullable = true)
 |-- start_station_id: string (nullable = true)
 |-- end_station_name: string (nullable = true)
 |-- end_station_id: string (nullable = true)
 |-- start_lat: string (nullable = true)
 |-- start_lng: string (nullable = true)
 |-- end_lat: string (nullable = true)
 |-- end_lng: string (nullable = true)
 |-- member_casual: string (nullable = true)
```

Issues: When I check datatypes of each column, all of both are string; moreover, some column have Null value.

- String: I change datatypes of two column that are started_time and ended_time to calculate

- Null value: null values appear in start_station_name, start_station_id, end_station_name, end_station_id column. However, I don't remove these rows. Firstly, these rows still have other data needed for analysis, deleting it will cause data to be biased. Secondly, columns have null values related to location information, the reason for this problem may be unstable GPS navigation system.

In [9]:
```
df_trip_1 = (
    df_trip
    .withColumn("start_time",to_timestamp("started_at").cast("long"))
    .withColumn("end_time",to_timestamp("ended_at").cast("long"))
    .withColumn("ride_length",(col("end_time")-col("start_time"))/60)
    .withColumn("day_of_week",f.date_format("started_at","E").cast("string"))
    .withColumn("day_week",(((f.dayofweek("started_at")+5)%7)+1))
)
```

In [10]:
```
df_trip_1.show(5)
```

```
+----+---------+---------+-------+-------+-------------+----------+----------+------------------+-----------+--------+
›n_id|start_lat|start_lng|end_lat|end_lng|member_casual|start_time|  end_time|       ride_length|day_of_week|day_week|
+----+---------+---------+-------+-------+-------------+----------+----------+------------------+-----------+--------+
null|     41.8|   -87.59|   41.8|  -87.6|       member|1623569488|1623569651| 2.716666666666667|        Sun|       7|
null|    41.79|   -87.59|   41.8|  -87.6|       member|1622780282|1622780659| 6.283333333333333|        Fri|       5|
null|     41.8|    -87.6|  41.79| -87.59|       member|1622774975|1622775334| 5.983333333333333|        Fri|       5|
null|    41.78|   -87.58|   41.8|  -87.6|       member|1622724965|1622726515|25.833333333333332|        Thu|       4|
null|     41.8|   -87.59|  41.79| -87.59|       member|1622790351|1622790599| 4.133333333333334|        Fri|       5|
+----+---------+---------+-------+-------+-------------+----------+----------+------------------+-----------+--------+
```

# *Phase 4. Analyse*

I want to show some different between casual and member, dataframe included time start and end in each trip, type of bike.

So, I started summary:

1. Average time customers spend for each trip

In [25]:
```
(df_trip_1
.groupBy("member_casual")
.avg("ride_length")
).show()
```

```
[Stage 72:=================================================>     (9 + 1) / 10]
```

```
+-------------+------------------+
|member_casual|  avg(ride_length)|
+-------------+------------------+
|       casual| 31.29077266826416|
|       member|13.144280819766637|
+-------------+------------------+
```

2. Demand for bikes

```
In [13]: (df_trip_1
          .groupBy("member_casual")
          .pivot("rideable_type")
          .agg(f.countDistinct("ride_id").alias("num_ride_trip"))
         ).show()
```

```
+-------------+------------+-----------+-------------+
|member_casual|classic_bike|docked_bike|electric_bike|
+-------------+------------+-----------+-------------+
|       casual|     1234410|     291391|      1010557|
|       member|     1968374|       null|      1252819|
+-------------+------------+-----------+-------------+
```

3. Frequency of bike share use in weekday

```
In [26]: (df_trip_1
          .groupBy("member_casual","day_of_week")
          .agg(f.countDistinct("ride_id").alias("num_ride_trip"))
         .sort(f.col("member_casual").asc(),f.asc("num_ride_trip"))
         ).show()
```

```
+-------------+-----------+-------------+
|member_casual|day_of_week|num_ride_trip|
+-------------+-----------+-------------+
|       casual|        Tue|       270548|
|       casual|        Wed|       284868|
|       casual|        Mon|       289029|
|       casual|        Thu|       298061|
|       casual|        Fri|       358203|
|       casual|        Sun|       477032|
|       casual|        Sat|       558617|
|       member|        Sun|       388042|
|       member|        Sat|       442741|
|       member|        Mon|       445635|
|       member|        Fri|       453281|
|       member|        Thu|       485843|
|       member|        Tue|       498682|
|       member|        Wed|       506969|
+-------------+-----------+-------------+
```

Analysis of Trends

- Member's average time spent on each trip is lower than casual. On an average, casual spends 31 minutes on a trip, in while members only spend about 13 minutes.
- Bike-Share provide customer three types of bike. Casual group uses all 3 types, members only use electric and classic.
- Summary of bike usage frequency on weekdays of two customer groups sorted by number of trips. In this data sheet, it can be seen that the casual group uses the bike a lot on weekends, but the members do the opposite.

## *Phase 5. Share*

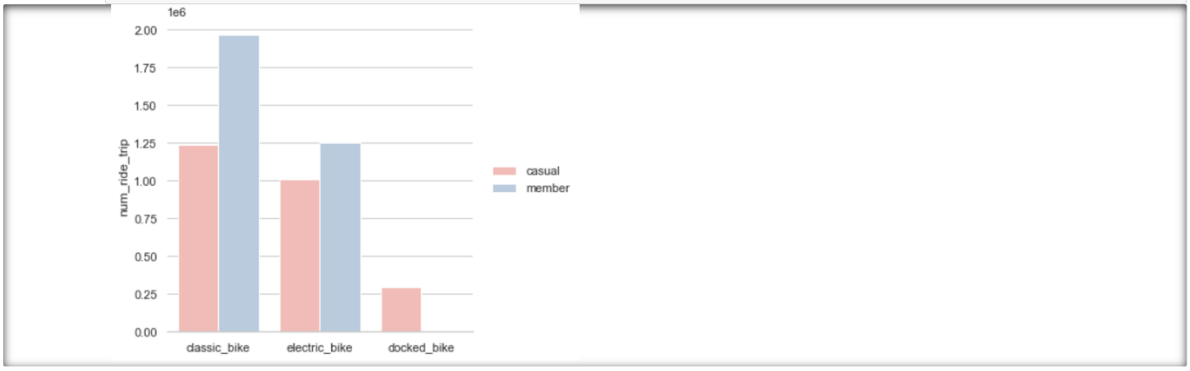
I will create Visualizations of my data, so my findings and analysis can be presented effectively and efficiently

The first visualization, I want to show and compare usage demand between two groups of customers with each type of bicycle that the company offers.

In [15]:
```
import seaborn as sns
sns.set_theme(style="whitegrid")

df_TripType_viz=(df_trip_1
 .groupBy("member_casual","rideable_type")
 .agg(f.countDistinct("ride_id").alias("num_ride_trip"))
 .toPandas()
)

g = sns.catplot(
    data=df_TripType_viz, kind="bar",
    x="rideable_type", y="num_ride_trip", hue="member_casual",
    ci="sd", palette="Pastel1", alpha=1, height=5
)
g.despine(left=True)
g.set_axis_labels("", "num_ride_trip")
g.legend.set_title("")
```




This column chart, only one-fifth of casual use a docked-bike and none of the members choose this type of bike. Both groups of customers mainly choose the remaining two types of bike. In addition, the classic bike is chosen to be used the most in both groups. Docked-bikes are bicycles managed by the station system, riders who want to use this type of bike must go to the correct stations to unlock and use and must return the bike to the specified stations. While electric and classic cars are vehicles that have integrated the unlocking system right on the car (dockless), the rider could use it and end his trip anywhere. Dockless is convenient for customers who have an usage demand stable, with certain route and destination. Therefore, members mostly use dockless. Whereas casuals use both docked-bike and dockless, since their intended use that is not binding, any location can be their starting point.

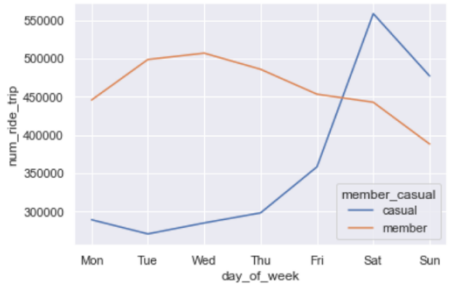
The second visualization, I want to show usage frequency on weekdays of two rider group.

In [17]:
```
sns.set_theme(style="darkgrid")

df_day_week=(df_trip_1
 .groupBy("member_casual","day_of_week","day_week")
 .agg(f.countDistinct("ride_id").alias("num_ride_trip"))
 .sort(f.asc("day_week"))
            ).toPandas()

sns.lineplot(x="day_of_week", y="num_ride_trip",
             hue="member_casual",
             data=df_day_week)
```

```
Out[17]: <AxesSubplot:xlabel='day_of_week', ylabel='num_ride_trip'>
```




This line chart show the difference between casual rider and member. While members have a relatively stable frequency of bicycle use and decrease slightly on weekends, casual riders have unstable demand, the number of trips skyrockets on weekends.

A main characteristic of annual members is their frequent and stable use of the bike-sharing service, in terms of trip number and trip duration. Which means bike is their preferred transportation in certain works, or it serves their needs of environmental issues, cost efficiency, convenience, etc.

On the other hand, casual riders don't often use bike as frequently or necessary as annual members (that's why they don't purchase annual membership). But when they do, they like to take weekends bike trip a lot, for leisure or relaxation. Their longer and volatile trip time may tells that their bike time does not tie to a fixed routine

## *Phase 6. Act*

Having the necessary insights (from my personal perspective), I would then give recommendations to help marketing team design new strategy to convert casual riders into annual members:

The marketing strategy should clarify and emphasize the benefit of using bikes instead of usual means of transportation. Particularly, casual riders need to be persuaded that using bikes frequently will have positive impacts on their own self-interest such as health, convenience, cost, etc.

The annual membership might include more benefits such as coupons, discounts, exclusive features, better customer service or access to online or offline communities of bikers around the city.

Social media is also a crucial tool that needs to be taken into great care too. Since people spend most of their time on social sites such as Facebook, Tiktok, etc. we could build and grow our community there, along with offline events or meetings exclusively for annual members. We can even use KOL or influencers to advertise our service efficiently.

## Recommendations

Since this is a fictitious case, the characteristics or metrics in the data source are quite limited. In fact, to be able to analyze more deeply, I personally think it is necessary to have other data such as user id, age group, gender. Because these factors have an influence on the rider's purpose and behavior. For example, young people use bicycles for exercise while middle-aged people use bicycles mainly to go to work.

In addition, this dataset show bicycle usage location information, but it is incomplete and not segmented into regions. This characteristics is also important because location will influence user behavior.