Report

Divvy shared bike

By TAO MENGJIE MB955521
LI WANSHAN MB955352
CHANG LAI NGA MB95536
GROUP F

Introduction of the data

Introduce the data briefly

The data we used in this project is about bike sharing (Divvy is Chicago's bike share system, with over 600 stations and 6,000+ bikes across Chicagoland. It's a fun, affordable and convenient way to get around). We got the data from https://www.divvybikes.com/system-data. This website offers historical trip data in Chicago. The trip data is released twice a year: once following the end of calendar Q2 and once following the end of calendar Q4. This data is provided according to the Divvy Data License Agreement.
Each file is anonymized and includes trip start day and time, trip end day and time, trip start station, trip end station and rider type (member, single and day pass) and there are 12 columns in total. If a member trip, it will also include member's self-reported gender and year of birth.

The reason why we chose the data

We all know that bike sharing is very popular in China. However, from time to time we can see some negative information about it in newspapers or from the Internet. Due to this reason, we want to see how bike sharing is going aboard. Therefore, we chose to explore the historical trip data of Divvy which is Chicago's bike share system very similar to bike sharing system in China.

Collection of the data

In this project, the method we used to collect the data was to write some codes to download the materials from the website automatically. We divided this part into two sections. The one is downloading some pictures, which we think these pictures can better show what we are going to explore. The other is downloading the data. For downloading the pictures, we used two libraries: IPython.display and IPython.core. display. The most important step of this section is to get the URL of the picture. In order to get the URL of the picture, we need to take advantage of patterns in the webpage underlying code and find the source code for this page known as HTML. If we want to get the URL of the picture, we can right click the picture and select Inspect Element. Then the URL of the picture is highlighted. For downloading the data, we used Requests. Getting the URL of the data is also the most essential step. Running the codes we wrote can only get the ZIP files, so we need to unzip them afterwards.

Data processing

We divided this part into three sections. First of all, we merged four csv files into one. Secondly, we tried to check out whether columns have missing values. We found that the missing percentage was not very large, so we used some functions to fill the missing values.
In order to merge four files into one, we the concat founction in Pandas. And Seaborn library can help us draw a chart and clearly show whether there are missing values and where they are.

Data cleaning

The datasets we collect are already structured, but it doesn't mean that they are good enough for us to analyze. We still have to clean out the invalid data to ensure that there won't be fuzzy mistakes in our analysis results.

Firstly, we found that not all record stands for a significant trip. Due to the nature of Shared bikes, there are a considerable number of records that come from faulty operations. If we use the attributes in datasets to describe those faulty operation records, then the characteristics of them are as follows: i) short duration; ii) the location hasn't changed after the trip. So we drop the records which have less than 100 seconds trip duration and the same starting and ending stations. The number of those invalid records are 6016.

Secondly, we find that there are some missing values in our datasets. The missing values are concentrated in "gender" column and "birthyear" column. Since we are going to analyze the users of bike-sharing, the missing values must be filled up in some reasonable ways. As for the "gender" attribute, we find that the percentage of the null values is not large, about 15%, so we used the fillna() function in Pandas and propagate the non-null values forward in "gender" column. Turning now to the "birthyear" attribute, we also propagated the non-null values forward to fill in the missing values and compared the generation distribution of users before and after. These two distributions are almost the same, except for a proportional increase in the number of users in each age group. So the result is that the way we used to fill up the columns will not significantly influence the characteristics of the datasets.

New libraries

In our project, we use several new libraries to analyze and visualize our data. Here are the three ones that we used a lot.

Seaborn

The first one is seaborn which is a graphical visualization python libraty based on matplotlib. It provides a highly interactive interface that allows users to make a variety of attractive statistical charts. And it is highly compatible with numpy and pandas data structures. In the most of the time, seaborn is convenient since it can directly process dataframe and dataframes are the main structures we used to analyze our data. But there are some flaws with seaborn,

the biggest one is that it isn't as powerful as matplotlib. Seaborn can just be seen as a complement to matplotlib, not a substitute. There are still a lot of types of charts that seaborn cannot draw, such as a bar chart showing the percentage of different components and multiple data line graph. We'd better use seaborn to make attractive plots and use matplotlib to produce more featured illustration.

Collections

The second one is collections library. We introduced this library primarily to use its Counter() function. This function is powerful and easy to use. It can traverse all the elements in data structure and record the number of times the values of each column appear. Although there is a function named value_count() in pandas which has the similar effects, Counter() is much easier to use and can make the code concise. So our code can be easy to read.

Requests

The third one is requests library which is a very useful Python HTTP client library. It is often used when writing crawlers and testing server response data. Our datasets are from divvybikes website and we used get() function in requests library to access those data. The function is very useful, all we need to do is to get the URLs of certain websites as the arguments, then we can collect the data we need.

Data analyzing and data visualization

The finding in this dataset:

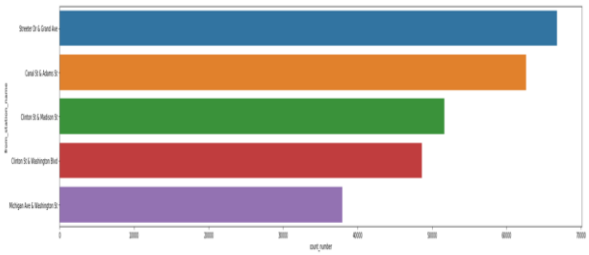

Fig. 1 – Top 5 most frequent starting points

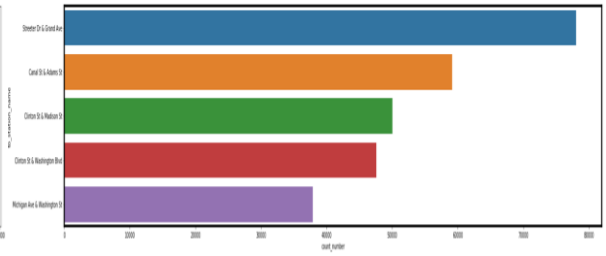

Fig. 2 – Top 5 most frequent ending points

These two charts are both drawn by using seaborn library, we first created new dataframes to describe the number of records of each starting and ending station and then directly passed this dataframe to the barplot() function to generate the charts. Fig. 1 In this horizontal bar plot we have plotted 5 most frequent starting points of the shared bike. They are Streeter Dr & Grand Ave which has 66786 records, Canal St & Adams St, Clinton St & Madison St, Clinton St & Washington Blvd, Michigan Ave & Washington St. Fig. 2 We can find the same 5 stations and the same order in this chart which has plotted 5 most frequent ending points of

the shared bike. The above 5 ending point, as far as we can guess, they would be near to some business community or business street.

Business Insight: Divvy should make sure that bikes are always available to pick up on these points. So our analysis and be used as an inspiration for dispatching bike distribution.

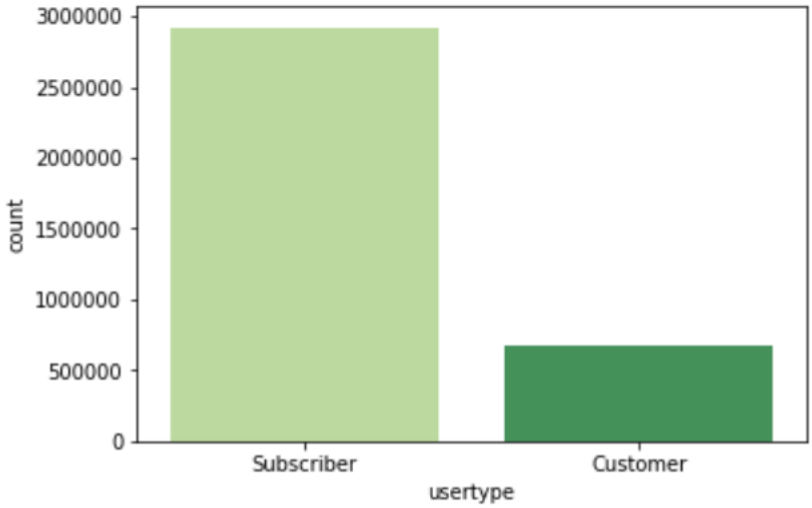

Fig. 3 – the distribution of users' type

To draw this chart, we used a new function of seaborn named countplot(). This function can directly count the number of records of each subclass in one column of a dataframe and visualize them. Fig. 3 There are two user type of shared bike, which is Subscriber and Customer. This chart is showing number of Subscriber and the number of Customer, the most popular user type of shared bike is subscriber. The reason why there are more number of subscribers than the customers is that more number of Divvy Bike users are working population who rents the bike on a daily basis.

Business suggestion: A subscriber is the one who has an ongoing commitment for renting the Divvy shared bike. Customers do not commit for this product. They may or may not use the service on this product in the future. Therefore, Divvy should keep focusing on maintaining its subscribers.

After we analyzed the percentage of the missing values of gender column, we find that it is small and we can propagate the non-null values forward to fill up this column. Fig. 4 This chart is showing the male users of shared bike are more than the female users of shared bike.

Business Value: The above histogram seems very important for the business purpose. As we can see most of the users are lying in the male users. Divvy stakeholders can make various strategy to focus on the female customers. They should focus on to promote female to rent the Divvy Bike for some strategies, for example: taking a bike is good for their health etc.

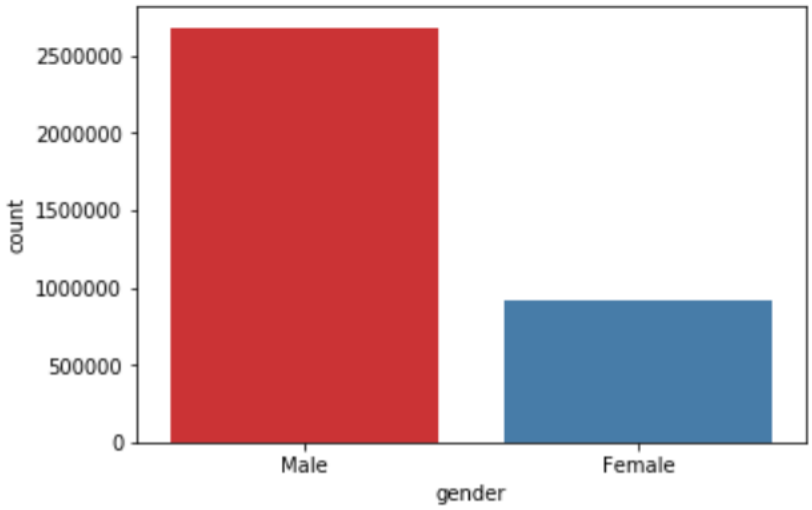

Fig. 4 – The gender distribution of users

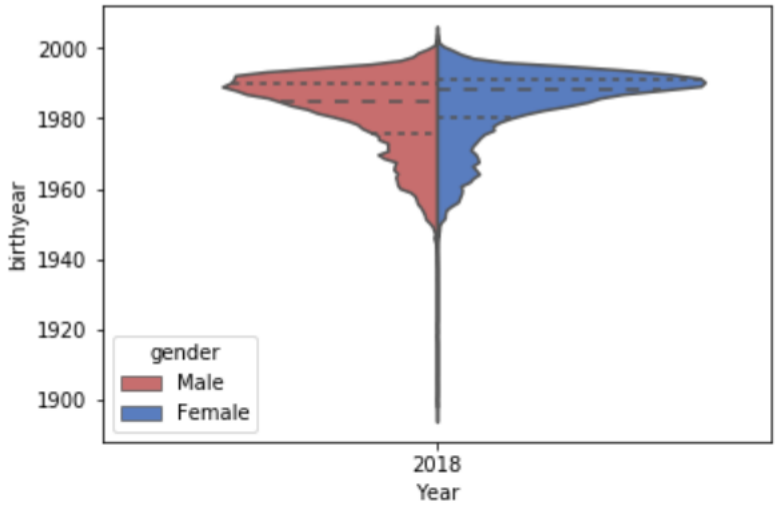

Fig. 5 – the generation distribution of users

As for the age of users, we used the ‘birthyear’ column to calculate a ‘generation’ column, since we want to get a more general result. Fig. 5 The chart is also drawn by seaborn and it is using a violinplot() function. This chart is showing the distribution of the users' age of shared bike, the post-1990 generation is the mainly group to use shared bike.

Business Value: The above chart seems very important for the business purpose. As we can see most of the users are lying in the post-1990 generation. Divvy stakeholders can make various strategy to focus on the generation of 2000 to rent the Divvy Bike, for example: taking a shared bike is a good way for saving their money rather than other transportation .

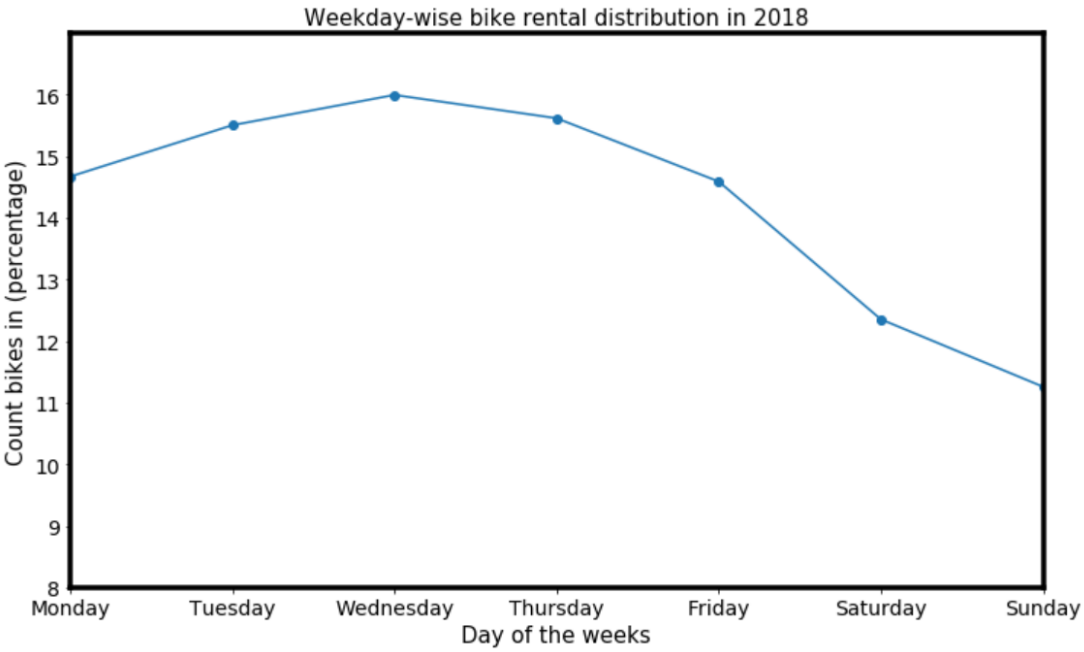

Fig. 6 – Frequency distribution for each day of the week

Fig. 6 The line plot drawn by matplotlib is showing the distribution of renting of shared bike in the week days. To draw this chart we converted the time data to datetime type and created a new column to describe day of week of each trip record. As we and see, Wednesday is the highest bar which indicates Wednesday was the peak day when people rented Divvy Bike. The plot also clearly indicates that on Saturday and Sunday, there are less number of shared bike users rented the bike in comparison to the week days.

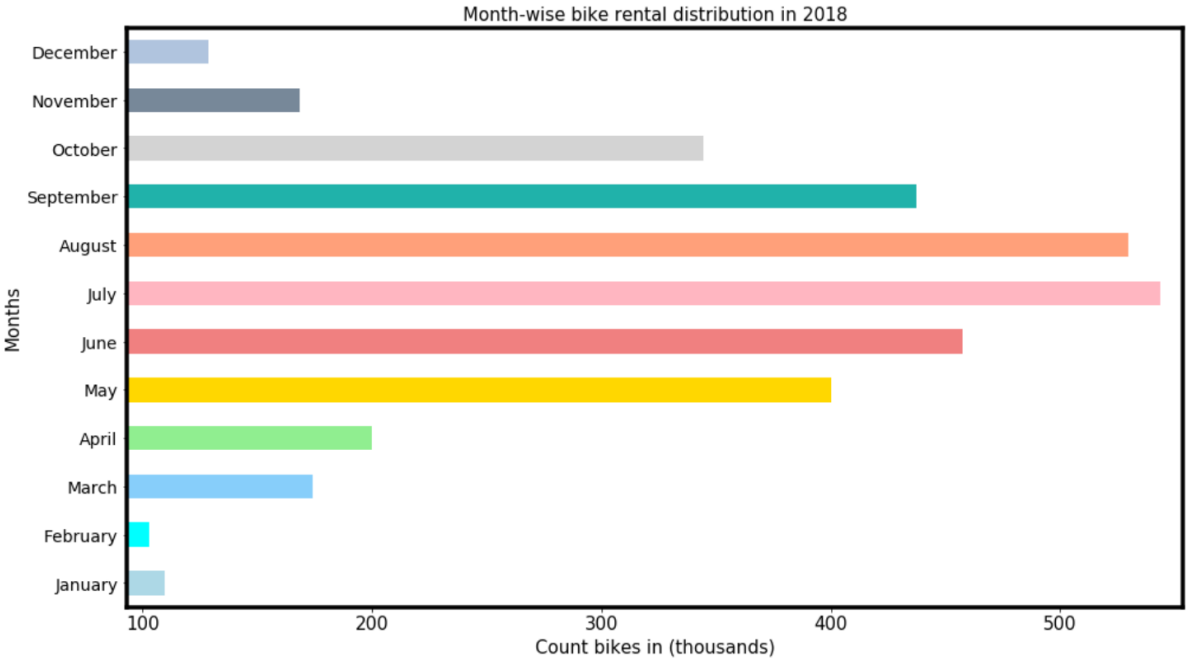

Fig. 7 – Frequency distribution of each month

Fig. 7 The bar plot is showing the breakdown of shared bike renting in months, which is giving a clear indication of the season in Chicago. The bar of July and August is the highest frequency of shared bike renting and the bar of January and February is the lowest frequency of shared bike renting. There are more people to use shared bike in summer rather than a cold winter.

In a summary, to better understand the use of divvy shared-bike in Chicago, we analyzed the popular stations, the main types of users and the using frequency during different time periods of divvy shared-bike. The result is a good inspiration for running the business.