Impacts of COVID-19 and "shelter-in-place" order on the Lyft biking user behavior

by Duy Pham

Preliminary Wrangling

This data set includes information about individual rides made in the Lyft bike-sharing system covering the greater San Francisco Bay area. The link to the dataset can be found here (https://s3.amazonaws.com/baywheels-data/index.html)

Noting that the aim of the project is to find out the impact of COVID-19 in the dataset. However, the format of the dataset in April and May 2020 are different from different dataset. Thus, the scope of the project will dive into the dataset in March 2020 only, where the dataset formats are similar.

The changing of the dataset format can be observed in the exploration task below

In [1]:
```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

%matplotlib inline
```

> Load in your dataset and describe its properties through the questions below. Try and motivate your exploration goals through this section.

In [2]:
```
df_10 = pd.read_csv ('201910-baywheels-tripdata.csv')
df_11= pd.read_csv ('201911-baywheels-tripdata.csv')
df_12 = pd.read_csv ('201912-baywheels-tripdata.csv')
df_1 = pd.read_csv ('202001-baywheels-tripdata.csv')
df_2 = pd.read_csv ('202002-baywheels-tripdata.csv')
df_3 = pd.read_csv ('202003-baywheels-tripdata.csv')
df_4 = pd.read_csv ('202004-baywheels-tripdata.csv')
```

```
D:\Anaconda\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarni
ng: Columns (14) have mixed types. Specify dtype option on import or set low_me
mory=False.
  interactivity=interactivity, compiler=compiler, result=result)
D:\Anaconda\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarni
ng: Columns (13) have mixed types. Specify dtype option on import or set low_me
mory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [3]:
```
1 df_11.head(1)
```

Out[3]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	707	2019-11-30 23:54:47.2970	2019-12-01 00:06:34.3780	30.0	San Francisco Caltrain (Townsend St at 4th St)	37.7765

In [4]:
```
1 df_1.head(1)
```

Out[4]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	83118	2020-01-31 15:23:47.7330	2020-02-01 14:29:06.2630	400.0	Buchanan St at North Point St	37.8042

In [5]:
```
1 df_2.head(1)
```

Out[5]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	62083	2020-02-29 18:32:30.5750	2020-03-01 11:47:14.0850	176.0	MacArthur BART Station	37.828

In [6]:
```
1 df_3.head(1)
```

Out[6]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	35187	2020-03-31 20:42:10.0790	2020-04-01 06:28:37.8440	462.0	Cruise Terminal at Pier 27	37.8046

In [7]:
```
1 #Starting from April, the dataset format changes
2 df_4.head(1)
```

Out[7]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end
0	5A1FF31692371859	electric_bike	2020-04-04 08:28:20	2020-04-04 08:33:34	NaN	NaN	

In [8]:
```
1 df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 295854 entries, 0 to 295853
Data columns (total 14 columns):
duration_sec               295854 non-null int64
start_time                 295854 non-null object
end_time                   295854 non-null object
start_station_id           146286 non-null float64
start_station_name         146866 non-null object
start_station_latitude     295854 non-null float64
start_station_longitude    295854 non-null float64
end_station_id             145934 non-null float64
end_station_name           146511 non-null object
end_station_latitude       295854 non-null float64
end_station_longitude      295854 non-null float64
bike_id                    295854 non-null int64
user_type                  295854 non-null object
rental_access_method       185746 non-null object
dtypes: float64(6), int64(2), object(6)
memory usage: 31.6+ MB
```

There are some null values, but it seems like it is not an error.

- For example, there are null values in the start_station_id, start_station_name; but not start_station_latitude and start_station_longtitude. It means that the Lyft bike session is over, but the bike is not returned to the Lyft station. Thus, we have the location of the bike at the beginning and end of the session.
- However, the most important point that we would want to pay attention to is the user_id, bike_id, location before and after. Thus, the data seems to be clean.

In [9]:

```
#Let's concatenate the dataset and observe, using dataset with similar data
#They are Lyft biking dataset from October 2019 to March 2020
frames = [df_10, df_11, df_12, df_1, df_2, df_3]
df = pd.concat(frames)
df.head()
```

```
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: FutureWarning: Sorting b
ecause non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  after removing the cwd from sys.path.
```

Out[9]:

	bike_id	bike_share_for_all_trip	duration_sec	end_station_id	end_station_latitude	end_station_lo
0	12222	No	62337	385.0	37.850578	-122
1	282	No	72610	30.0	37.776598	-122
2	10940	No	56636	453.0	37.777934	-122
3	12623	No	42250	163.0	37.797320	-122
4	2601	No	40076	237.0	37.775232	-122

The error is made because the data format of the last column are different, but the rest of the dataset is the same. Thus, this is the final data that I will be using to explore the impact of COVID-19

In [10]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1480500 entries, 0 to 176798
Data columns (total 15 columns):
bike_id                    1480500 non-null int64
bike_share_for_all_trip    423981 non-null object
duration_sec               1480500 non-null int64
end_station_id             972399 non-null float64
end_station_latitude       1480500 non-null float64
end_station_longitude      1480500 non-null float64
end_station_name           973675 non-null object
end_time                   1480500 non-null object
rental_access_method       646949 non-null object
start_station_id           972256 non-null float64
start_station_latitude     1480500 non-null float64
start_station_longitude    1480500 non-null float64
start_station_name         973494 non-null object
start_time                 1480500 non-null object
user_type                  1480500 non-null object
dtypes: float64(6), int64(2), object(7)
memory usage: 180.7+ MB
```

In [11]:
```
1 #Check the number of rows and columns
2 df.shape
```

Out[11]:
```
(1480500, 15)
```

In [12]:
```
1 df.describe()
```

Out[12]:

	bike_id	duration_sec	end_station_id	end_station_latitude	end_station_longitude	star
count	1.480500e+06	1.480500e+06	972399.000000	1.480500e+06	1.480500e+06	972
mean	2.239660e+05	8.077897e+02	160.186824	3.774980e+01	-1.223459e+02	
std	2.930290e+05	1.905540e+03	139.685620	3.314746e-01	1.013874e+00	
min	4.000000e+00	6.000000e+01	3.000000	0.000000e+00	-1.225758e+02	
25%	1.002300e+04	3.700000e+02	43.000000	3.776657e+01	-1.224174e+02	
50%	1.254500e+04	5.910000e+02	112.000000	3.777779e+01	-1.224009e+02	
75%	4.299800e+05	9.190000e+02	254.000000	3.779202e+01	-1.223910e+02	
max	9.999600e+05	9.121100e+05	521.000000	3.989257e+01	0.000000e+00	

In [13]:
```
1 df.duplicated().sum()
```

Out[13]:
```
7512
```

In [14]:
```
df.user_type.value_counts()
```

Out[14]:
```
Subscriber    951451
Customer      529049
Name: user_type, dtype: int64
```

In [15]:
```
df.rental_access_method.value_counts()
```

Out[15]:
```
app        595649
clipper     51300
Name: rental_access_method, dtype: int64
```

In [16]:
```
#Checking for missing values using isnull()
df.isnull().sum()
```

Out[16]:
```
bike_id                       0
bike_share_for_all_trip    1056519
duration_sec                  0
end_station_id               508101
end_station_latitude             0
end_station_longitude            0
end_station_name             506825
end_time                      0
rental_access_method         833551
start_station_id             508244
start_station_latitude           0
start_station_longitude          0
start_station_name           507006
start_time                       0
user_type                        0
dtype: int64
```

In [17]:
```
#Recognize that the bike_share_for_all_trip and rental_access_method have a
#or unimportant for research question
df.drop(['bike_share_for_all_trip', 'rental_access_method'], axis = 1, inpla
```

DATA ASSESSMENT

What is the structure of your dataset?

bike_id
duration_sec
end_station_id
end_station_latitude
end_station_longitude
end_station_name
end_time
start_station_id
start_station_latitude

start_station_longitude
start_station_name
start_time
user_type

What is/are the main feature(s) of interest in your dataset?

1. Was there a sudden stop in the Lyft bike order in March, compared to previous months due to COVID-19? For more specific, the "shelter-in-place" order took effect in March 17. Was there a sudden drop in the later half of March compared to many other days?
2. What are the frequent riding behavior of people after the "shelter-in-place order" took effect? Did riding time reduce? Did people bike early in the morning more often? Did people reduce the frequency of riding the bike in midday because they didn't have to go to work at the office (but work from home)
3. What are the changes of user_type in March, compared to previous months?

What features in the dataset do you think will help support your investigation into your feature(s) of interest?

1. duration_sec
2. start time (month and hour)
3. user type
4. (count of) orders

Make a copy

In [18]:
```
1 df_clean = df.copy()
```

Data cleaning

In [19]:

```
#DEFINE: start_time into datetime

#CODE
# (1) Timestamp to datetime format
df_clean['start_time'] = pd.to_datetime(df_clean['start_time'], format='%Y-%

#Test
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1480500 entries, 0 to 176798
Data columns (total 13 columns):
bike_id                    1480500 non-null int64
duration_sec               1480500 non-null int64
end_station_id             972399 non-null float64
end_station_latitude       1480500 non-null float64
end_station_longitude      1480500 non-null float64
end_station_name           973675 non-null object
end_time                   1480500 non-null object
start_station_id           972256 non-null float64
start_station_latitude     1480500 non-null float64
start_station_longitude    1480500 non-null float64
start_station_name         973494 non-null object
start_time                 1480500 non-null datetime64[ns]
user_type                  1480500 non-null object
dtypes: datetime64[ns](1), float64(6), int64(2), object(4)
memory usage: 158.1+ MB
```

In [20]:

```
## DEFINE: Convert start time into smaller component

## CODE
df_clean['duration_min'] = df_clean['duration_sec']/60
df_clean['start_month_number'] = df_clean.start_time.dt.strftime('%m')
df_clean['start_day'] = df_clean.start_time.dt.strftime('%d')
df_clean['start_hour'] = df_clean.start_time.dt.strftime('%H')
df_clean['start_day_of_week'] = df_clean.start_time.dt.strftime('%A')
df_clean['start_month'] = df_clean.start_time.dt.strftime('%B')

## TEST
df_clean.head()
```

Out[20]:

ke_id	duration_sec	end_station_id	end_station_latitude	end_station_longitude	end_station_name	
2222	62337	385.0	37.850578	-122.278175	Woolsey St at Sacramento St	09
282	72610	30.0	37.776598	-122.395282	San Francisco Caltrain (Townsend St at 4th St)	09
0940	56636	453.0	37.777934	-122.396973	Brannan St at 4th St	09
2623	42250	163.0	37.797320	-122.265320	Lake Merritt BART Station	07
2601	40076	237.0	37.775232	-122.224498	Fruitvale BART Station	05

In [21]:

```
#Still have some April transaction, need to remove it
df_clean.start_month.value_counts()
```

Out[21]:

```
February    424789
January     295854
October     239895
November    185496
March       182632
December    150102
April         1732
Name: start_month, dtype: int64
```

In [22]:
```
## DEFINE: remove April transactions

## CODE
df_clean = df_clean[df_clean['start_month']!='April']

## TEST
df_clean.start_month.value_counts()
```

Out[22]:
```
February    424789
January     295854
October     239895
November    185496
March       182632
December    150102
Name: start_month, dtype: int64
```

In [23]:
```
#DEFINE: upper half - lower half of the month

##CODE
df_clean['start_day'] = df_clean.start_day.astype(np.int64)
df_clean['month_upper'] = df_clean.start_day.apply(lambda x: ' - second' if
df_clean['half-month'] = df_clean.start_month + df_clean.month_upper
```

In [24]:
```
df_clean.head()
```

Out[24]:

	bike_id	duration_sec	end_station_id	end_station_latitude	end_station_longitude	end_station_na
0	12222	62337	385.0	37.850578	-122.278175	Woolsey S Sacramento
1	282	72610	30.0	37.776598	-122.395282	San Franci Caltrain (Towns St at 4th
2	10940	56636	453.0	37.777934	-122.396973	Brannan St at
3	12623	42250	163.0	37.797320	-122.265320	Lake Merritt BA Sta
4	2601	40076	237.0	37.775232	-122.224498	Fruitvale BA Sta

5 rows × 21 columns

Save the dataset

In [25]:
```
# save the cleaned data to csv file
df_clean.to_csv('data.csv',index=None)
```

Open the dataset

In [26]:
```
df = pd.read_csv ('data.csv')
```

In [27]:
```
df.head(3)
```

Out[27]:

	bike_id	duration_sec	end_station_id	end_station_latitude	end_station_longitude	end_station_na
0	12222	62337	385.0	37.850578	-122.278175	Woolsey S Sacramento
1	282	72610	30.0	37.776598	-122.395282	San Franci Caltrain (Towns St at 4th
2	10940	56636	453.0	37.777934	-122.396973	Brannan St at

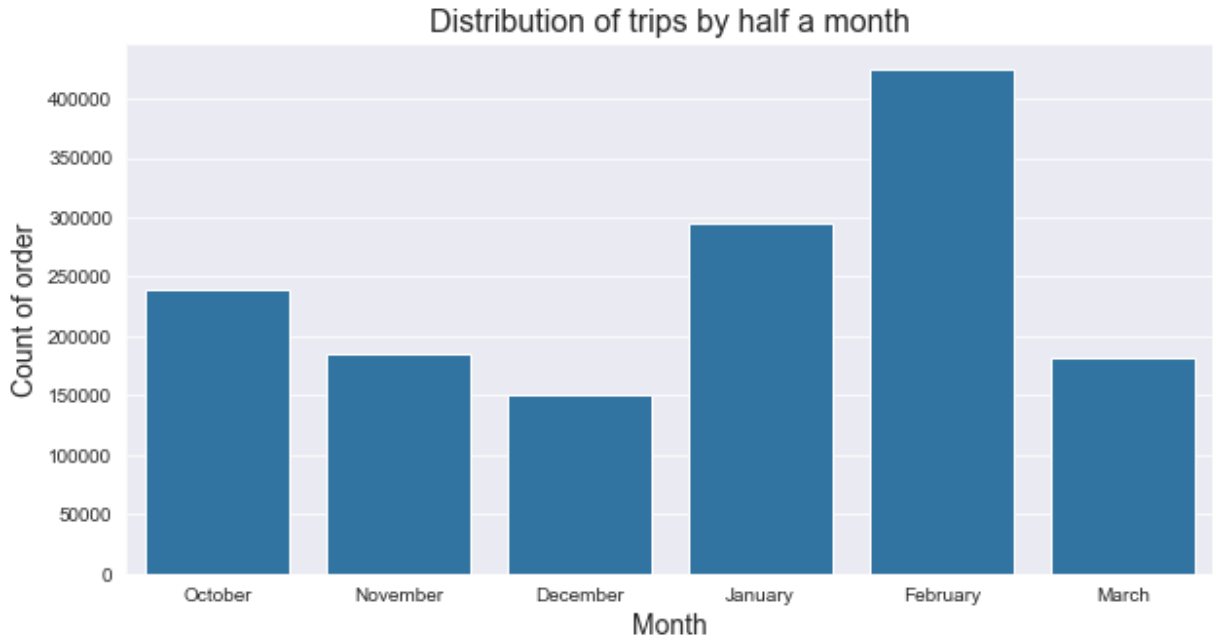
3 rows × 21 columns

Univariate Exploration

Number of biking order per month

In [28]:

```
plt.figure (figsize = (10,5))
base_color = sb.color_palette('Paired')[1]
sb.set_style('darkgrid')
sb.countplot(data=df, x='start_month', color=base_color)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Count of order', fontsize=14)
plt.title("Distribution of trips by half a month", fontsize=16);
```



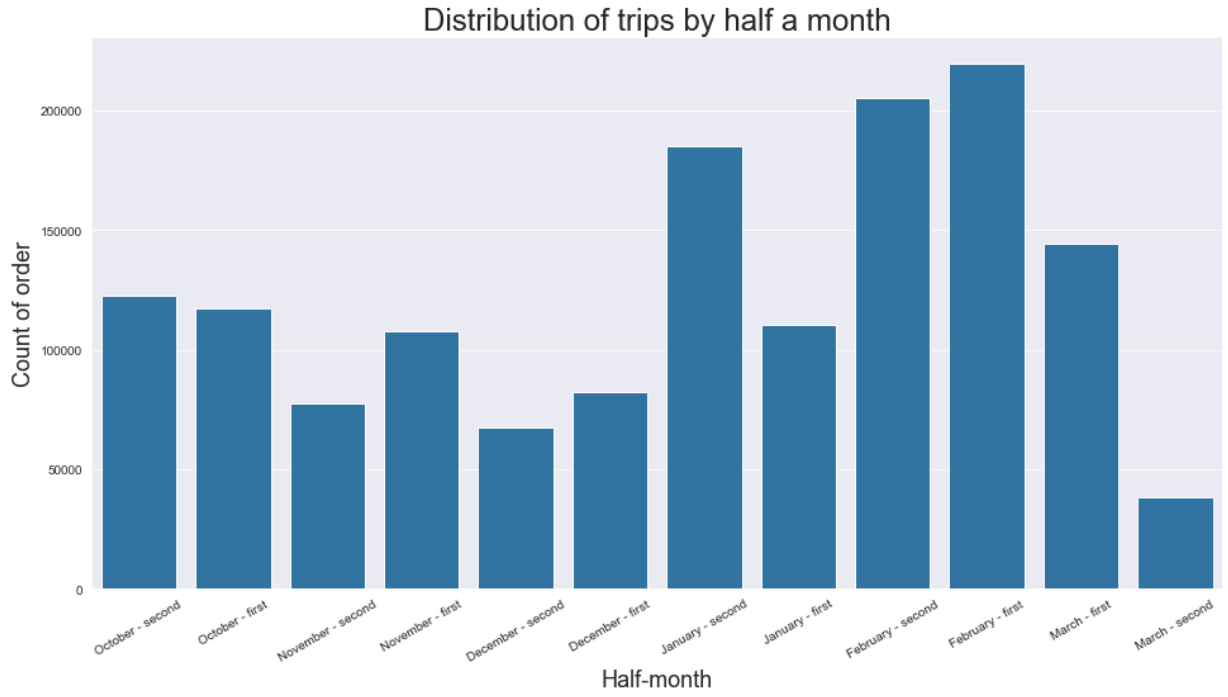
Let's look at this!! The number of bike order in March is significantly lower than January and February, but it was not that different from October, November and December last year. Thus, we might not be able to conclude that the drop in March was due to the COVID-19, but it might also come from the order cycle of the year (just like economic cycle).

The surge in the number of January might come from the LyftUp strategy, announced by Lyft, which can be found here (https://www.lyft.com/blog/posts/lyftup-bikes). There has been an upward trend since then, except for March 2020

Number of biking order per half a month

In [29]:

```
plt.figure (figsize = (16,8))
sb.countplot(data=df, x='half-month', color=base_color)
plt.xlabel('Half-month', fontsize=18)
plt.xticks (rotation = 30)
plt.ylabel('Count of order', fontsize=18)
plt.title("Distribution of trips by half a month", fontsize=24);
```

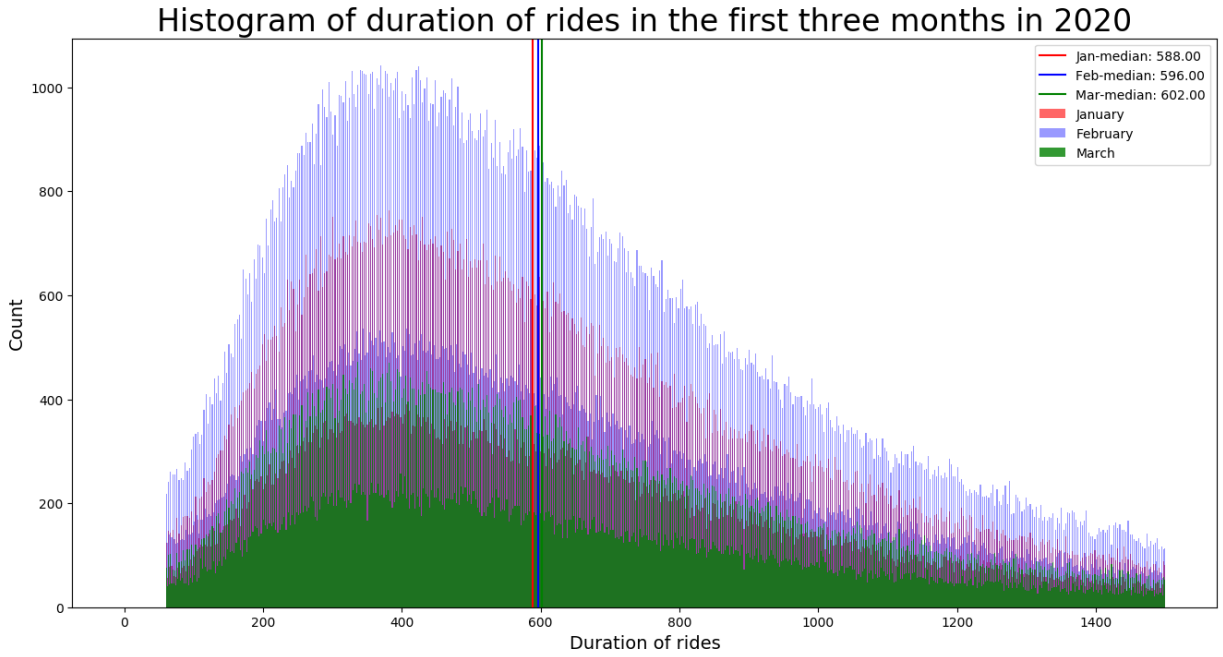


From this analysis, it is clear that the COVID-19 actually struck Lyft bike business significantly. The later half of the March witnesses a significant drop in the number of order, compared to any other previous half a month. For more specific, the news of COVID-19 spread around the Bay Area since the beginning of March, which witnesses a number of drop in bike sharing. But the "shelter-in-place" in March 17 order put a great damage into this business.

Histogram of riding duration in the first three months of 2020

In [30]:

```
import matplotlib as mpl
mpl.rcParams.update(mpl.rcParamsDefault)

plt.figure (figsize = (16,8))

plt.hist(df[df['start_month']=='January'].duration_sec, color = 'r', alpha =
plt.axvline (np.median(df[df['start_month']=='January'].duration_sec), color

plt.hist(df[df['start_month']=='February'].duration_sec, color = 'b', alpha
plt.axvline (np.median(df[df['start_month']=='February'].duration_sec), colo

plt.hist(df[df['start_month']=='March'].duration_sec, color = 'g', alpha = 0
plt.axvline (np.median(df[df['start_month']=='March'].duration_sec), color =

plt.xlabel('Duration of rides', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.legend()
plt.title("Histogram of duration of rides in the first three months in 2020"
```

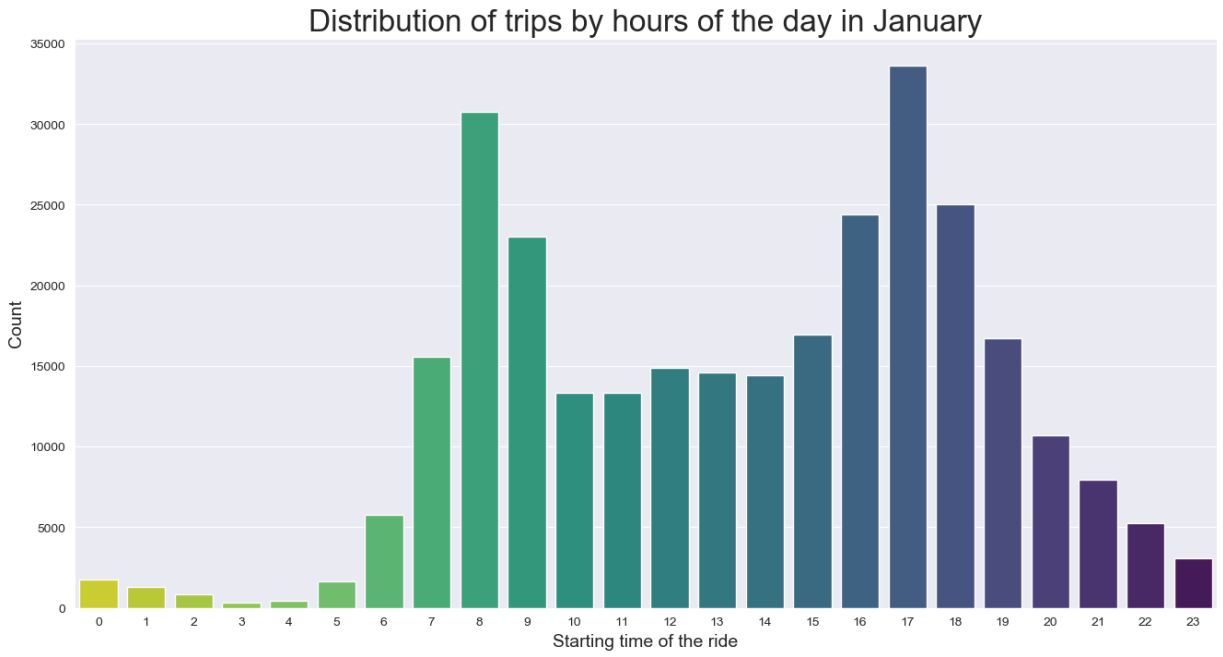


Based on the histogram, it is clear that the median duration of rides does not change. In that case, the revenue generated from each ride is unlikely to reduce. Lyft bike business is likely to be impacted by the reduction in the number of rides only.

What is the popular starting time of Lyft ride

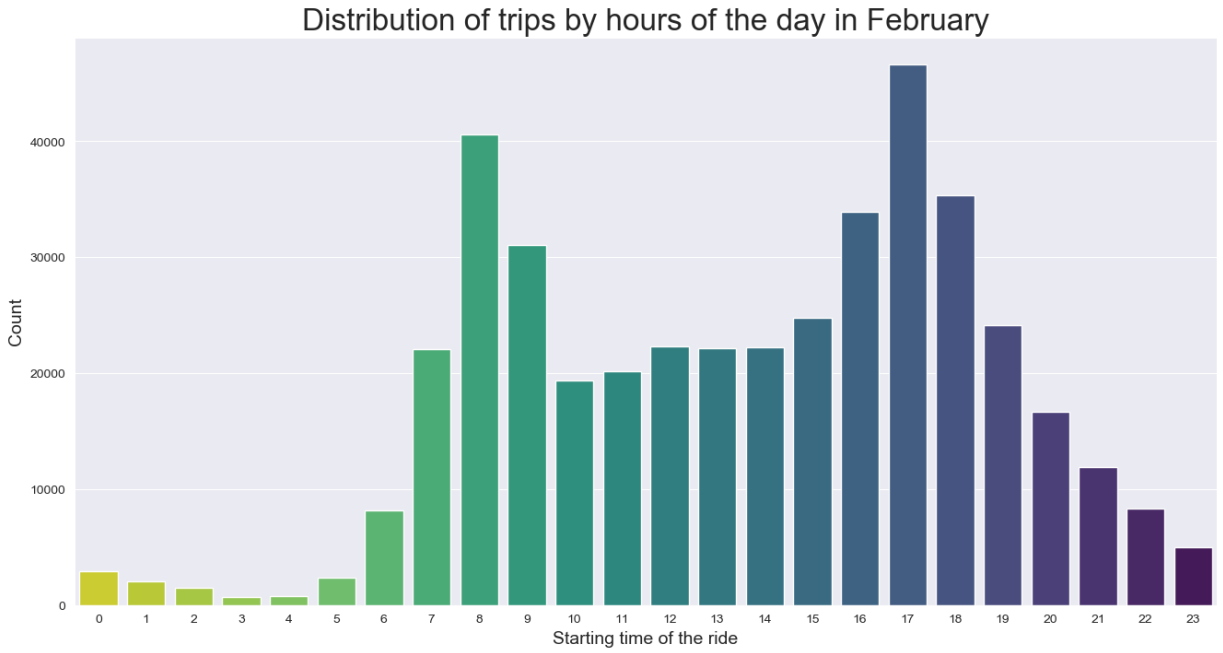
In [31]:

```
# Distribution of trips by hours
base_color = sb.color_palette('Paired')[1]
sb.set_style('darkgrid')

plt.figure (figsize = (16,8))
sb.countplot(data=df[df['start_month']=='January'], x='start_hour', palette
plt.xlabel('Starting time of the ride', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("Distribution of trips by hours of the day in January", fontsize=2
```



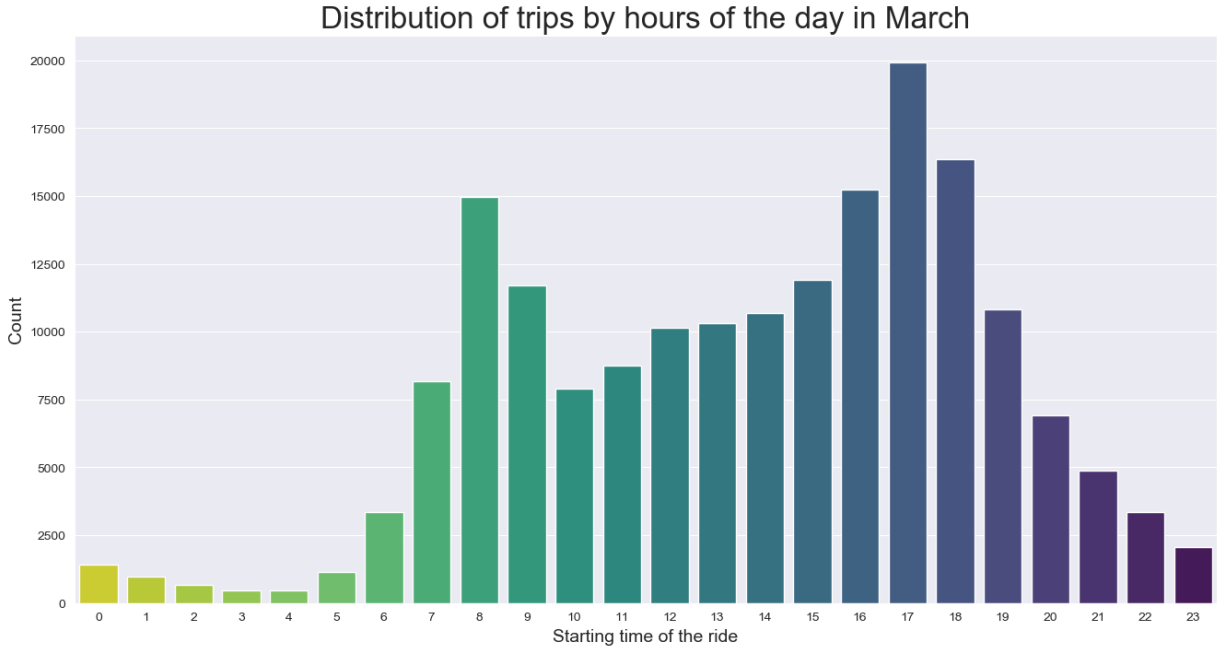
In [32]:

```
# Visualizing distribution of trips by hours
plt.figure (figsize = (16,8))
sb.countplot(data=df[df['start_month']=='February'], x='start_hour', palette
plt.xlabel('Starting time of the ride', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("Distribution of trips by hours of the day in February", fontsize=
```



In [34]:

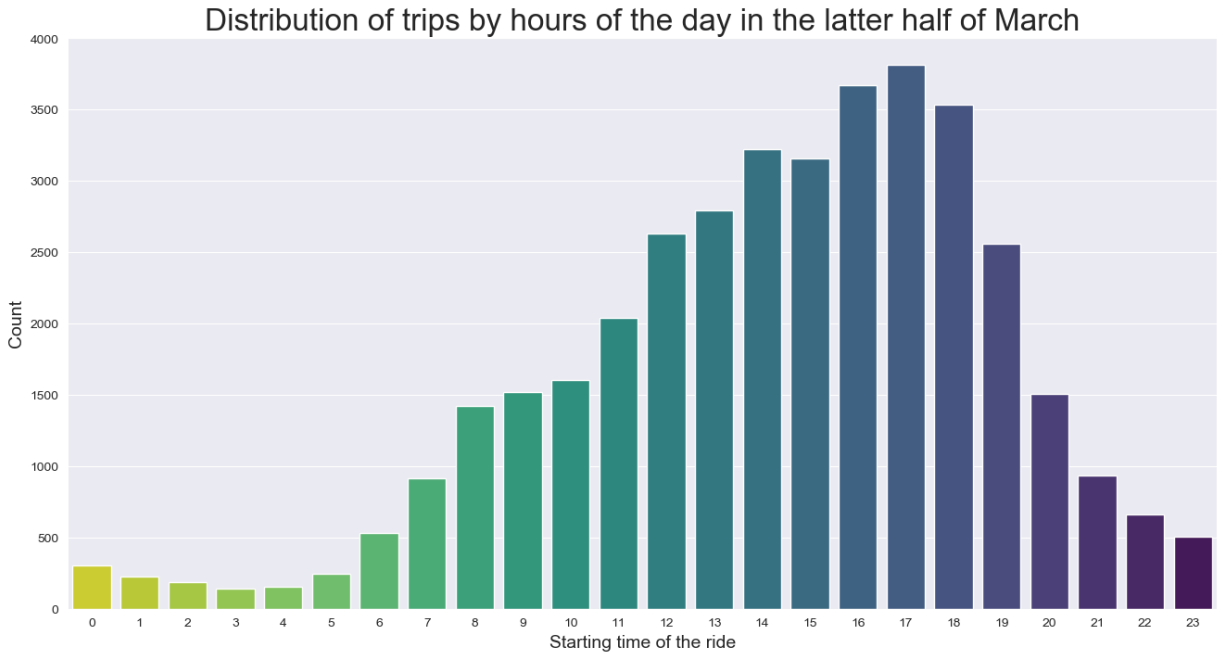
```
# Visualizing distribution of trips by hours
plt.figure (figsize = (16,8))
sb.countplot(data=df[df['start_month']=='March'], x='start_hour', palette =
plt.xlabel('Starting time of the ride', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("Distribution of trips by hours of the day in March", fontsize=24)
```



Looking at the graph, morning (7am - 9am) and afternoon (4pm-6pm) are the busiest time of the

day in terms of bike riding. In January and February, there was only a slight increase in the afternoon bike sharing compared to the morning. However, in March, there was a significantly visible drop in the distribution of people riding bike in the morning. It seems that fewer people going to work led to the drop in the number of riding bike. Let's look at the latter half of March with the data visualization below.

In [35]:
```
# Visualizing distribution of trips by hours
plt.figure (figsize = (16,8))
sb.countplot(data=df[df['half-month']=='March - second'], x='start_hour', pa
plt.xlabel('Starting time of the ride', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("Distribution of trips by hours of the day in the latter half of M
```



As expected, at the latter half of March when the "shelter-in-place" took effect, there is a significant drop in the proportion of people riding a bike in the morning. The drop in the morning is highly correlated with the working time. Below are two hypotheses for further user study:

1. After the "shelter-in-place" took effect, people do not need to go to work. The assumption is that people often go to work by bike in the past. As they don't have to work in the office, they don't have to take Lyft bike.
2. The biking is for physical exercise purposes. When people can work from home, people basically have a more flexible schedule. For that reason, instead of waking up super early and biking in the morning so that they can go to the office on time, they can basically exercise any time of the day. Thus, people decide to exercise equally over the day.

Bivariate Exploration

Relationship of pairs of variables:

1. User type and month
2. User type and time of the day

3. User type and day of the week

In [36]:
```
1 df.head(2)
```

Out[36]:

	bike_id	duration_sec	end_station_id	end_station_latitude	end_station_longitude	end_station_na
0	12222	62337	385.0	37.850578	-122.278175	Woolsey S Sacramento
1	282	72610	30.0	37.776598	-122.395282	San Franci Caltrain (Towns St at 4th

2 rows × 21 columns

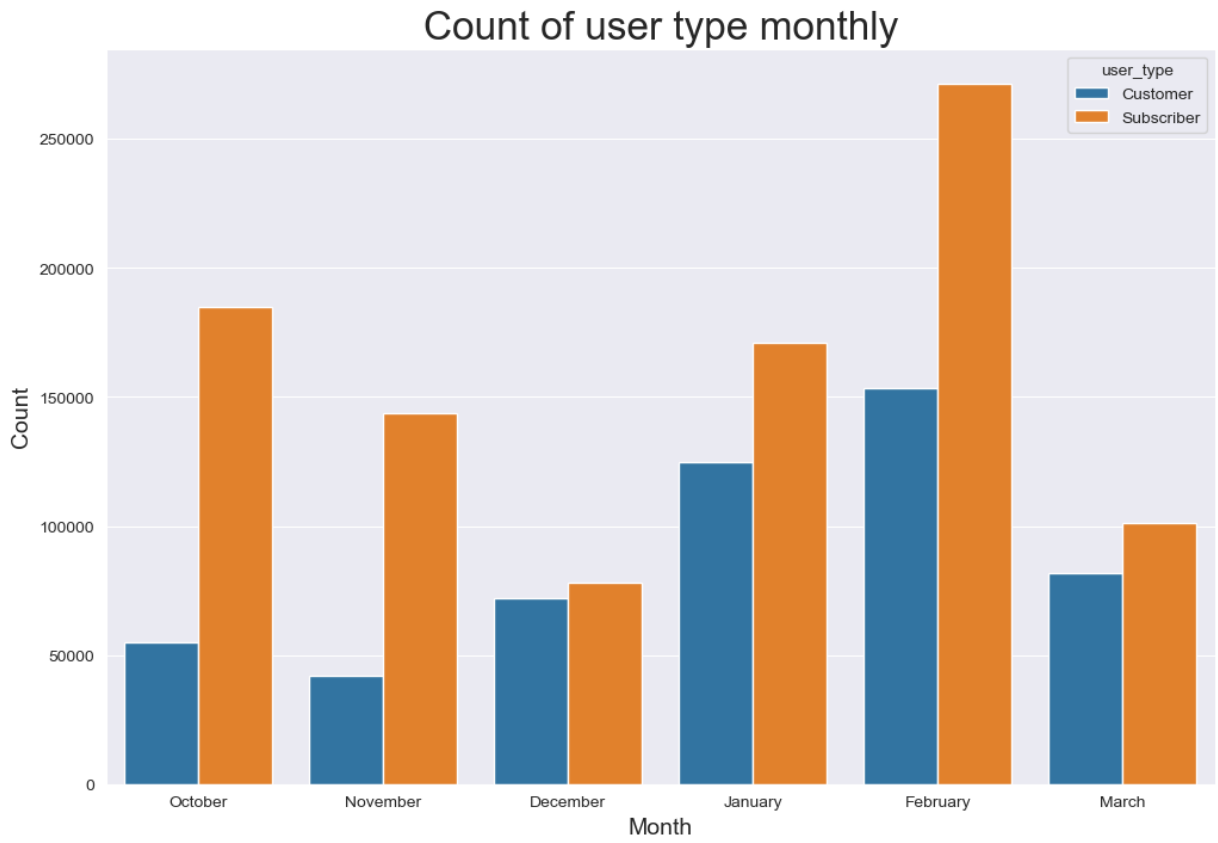
In [37]:
```
1 df.columns
```

Out[37]:
```
Index(['bike_id', 'duration_sec', 'end_station_id', 'end_station_latitude',
       'end_station_longitude', 'end_station_name', 'end_time',
       'start_station_id', 'start_station_latitude', 'start_station_longitude',
       'start_station_name', 'start_time', 'user_type', 'duration_min',
       'start_month_number', 'start_day', 'start_hour', 'start_day_of_week',
       'start_month', 'month_upper', 'half-month'],
      dtype='object')
```

Monthly usage by user types

In [38]:

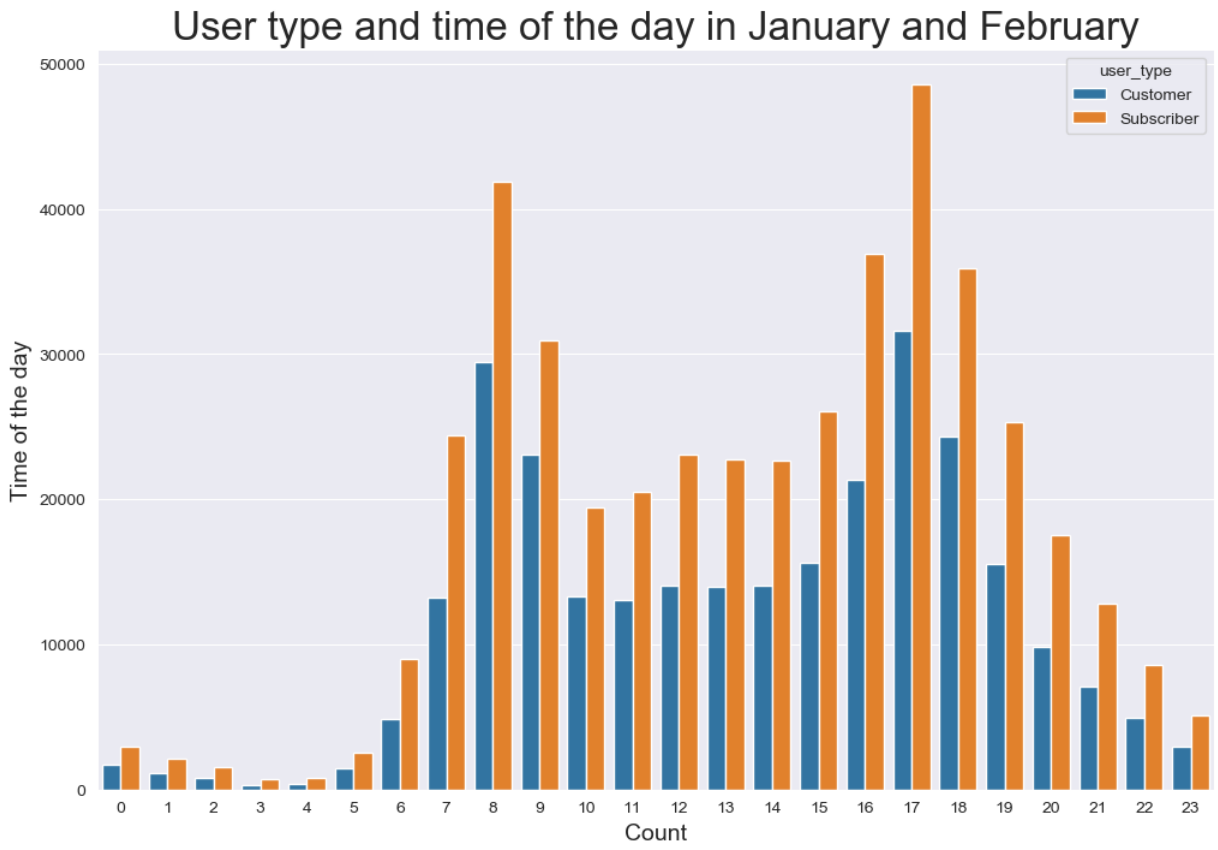
```
# Monthly usage by user types
plt.figure (figsize = (12,8))
sb.countplot(data=df, x='start_month', hue='user_type');
plt.xlabel('Month', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("Count of user type monthly", fontsize=24);
```



From the chart above, I can observe that the number of "subscriber" should be more than the number of "customer" throughout all year. However, there is a significant drop in the surplus of "subscribers" compared to the "customer" segment in December and March. By having a quick Google about Lyft scandal, I can easily find out the [sexual harassment scandal (https://www.cbsnews.com/news/lyft-sexual-assault-lawsuit-passengers-sue-lyft-claiming-they-were-sexually-assaulted-by-drivers-today-2019-12-04/)](https://www.cbsnews.com/news/lyft-sexual-assault-lawsuit-passengers-sue-lyft-claiming-they-were-sexually-assaulted-by-drivers-today-2019-12-04/) of Lyft in early December, which I believe create a significant drop in the number of users. However, the drop in March implies that people are less likely to use Lyft regularly, which make it less attractive for subscribers. With this evidence, I have a stronger belief on my first hypothesis stated above, such that people use Lyft bike to go to work in the morning before COVID-19 happened.

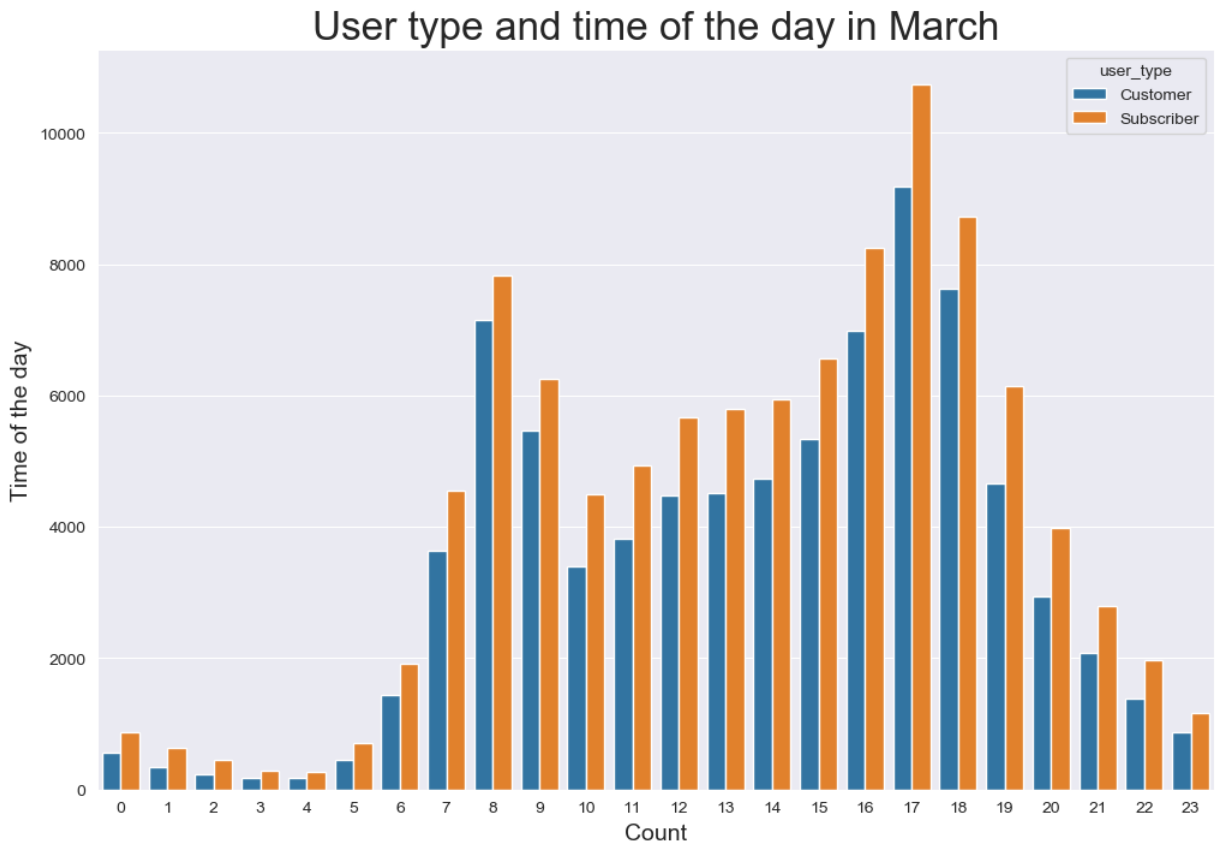
Time of the day and user types

In [39]:
```
#In January and February
plt.figure (figsize = (12,8))
sb.countplot(data=df[(df['start_month']=='January')|(df['start_month']=='Feb
plt.xlabel('Count', fontsize=14)
plt.ylabel('Time of the day', fontsize=14)
plt.title("User type and time of the day in January and February", fontsize=
```



In [40]:

```
#In March
plt.figure (figsize = (12,8))
sb.countplot(data=df[(df['start_month']=='March')], x='start_hour', hue='use
plt.xlabel('Count', fontsize=14)
plt.ylabel('Time of the day', fontsize=14)
plt.title("User type and time of the day in March", fontsize=24);
```

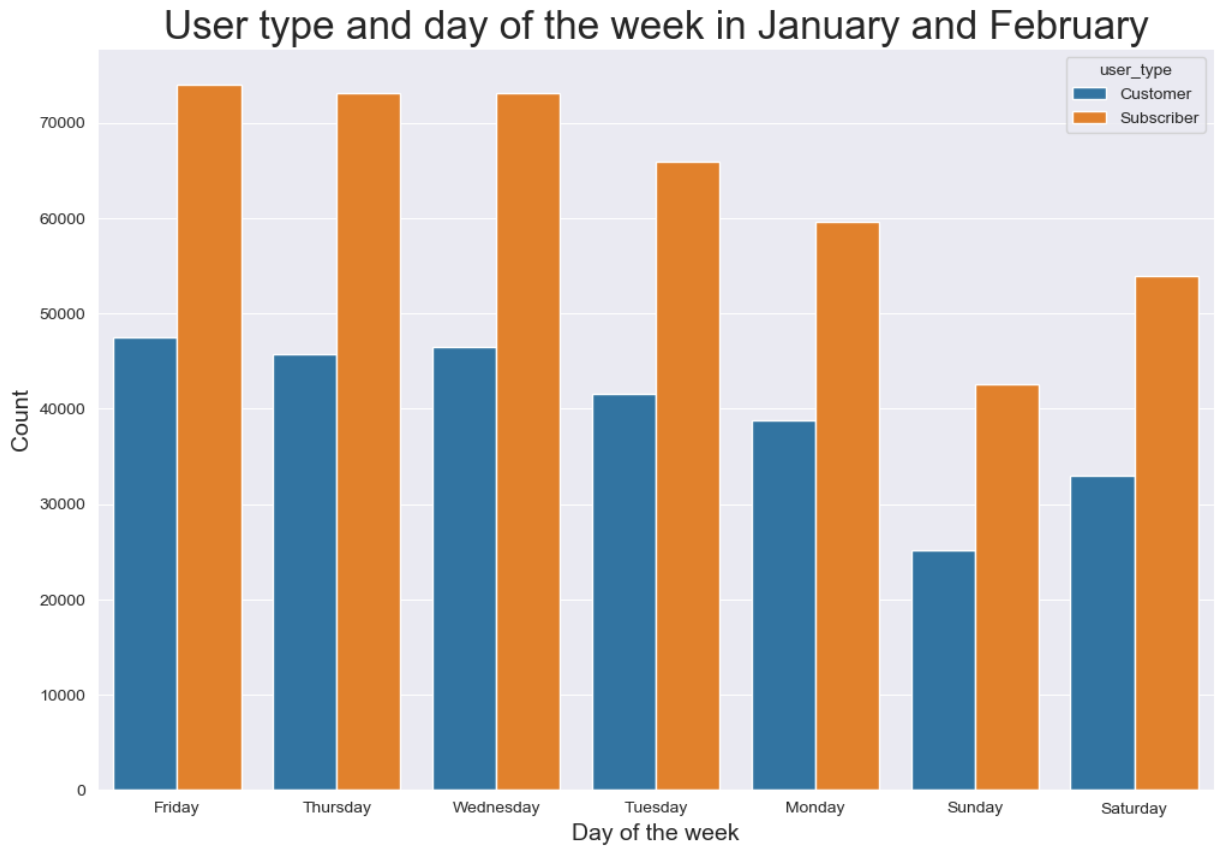


It seems that both type of customers reduce the frequency of using Lyft ride sharing in March uniformly throughout the day.

User type and day of the week

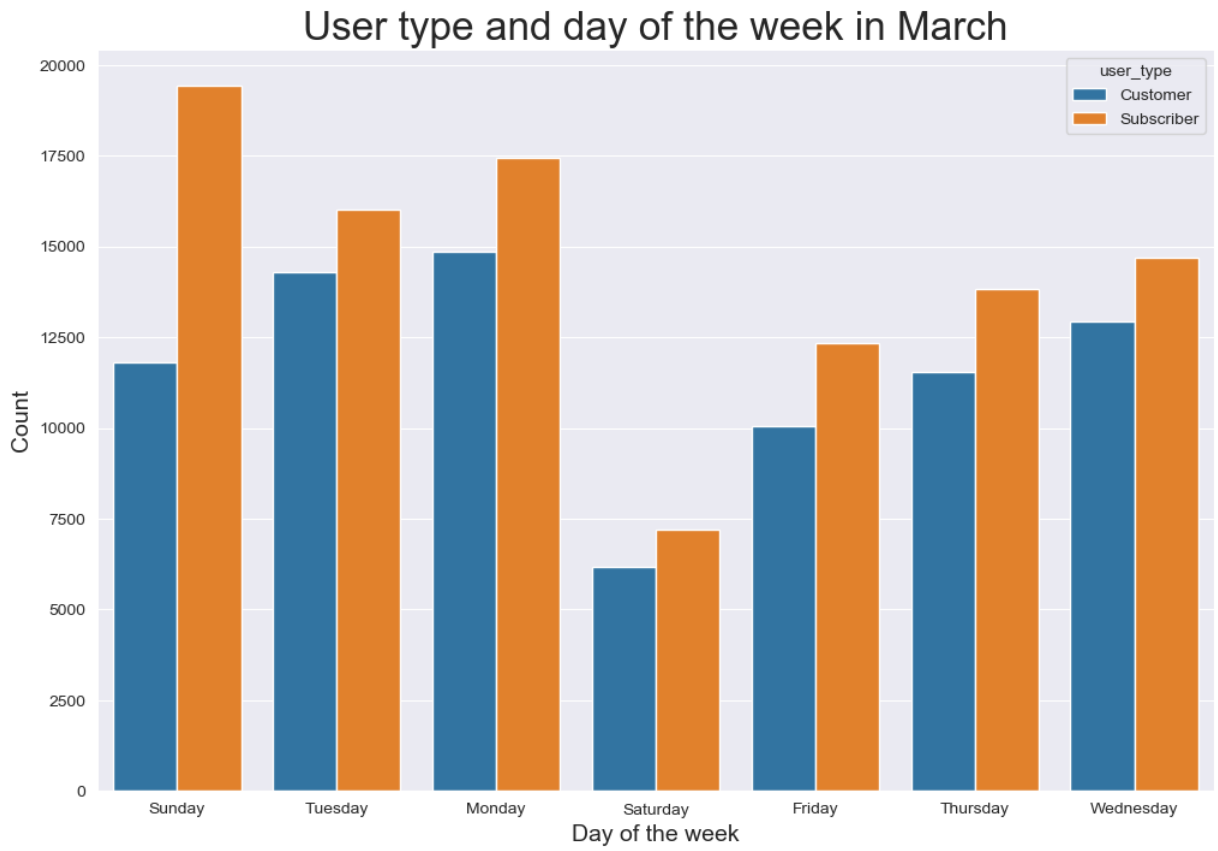
In [46]:

```
#In January and February
plt.figure (figsize = (12,8))
sb.countplot(data=df[(df['start_month']=='January')|(df['start_month']=='Feb
plt.xlabel('Day of the week', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("User type and day of the week in January and February", fontsize=
```



In [45]:

```
#In March
plt.figure (figsize = (12,8))
sb.countplot(data=df[(df['start_month']=='March')], x='start_day_of_week', h
plt.xlabel('Day of the week', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title("User type and day of the week in March", fontsize=24);
```

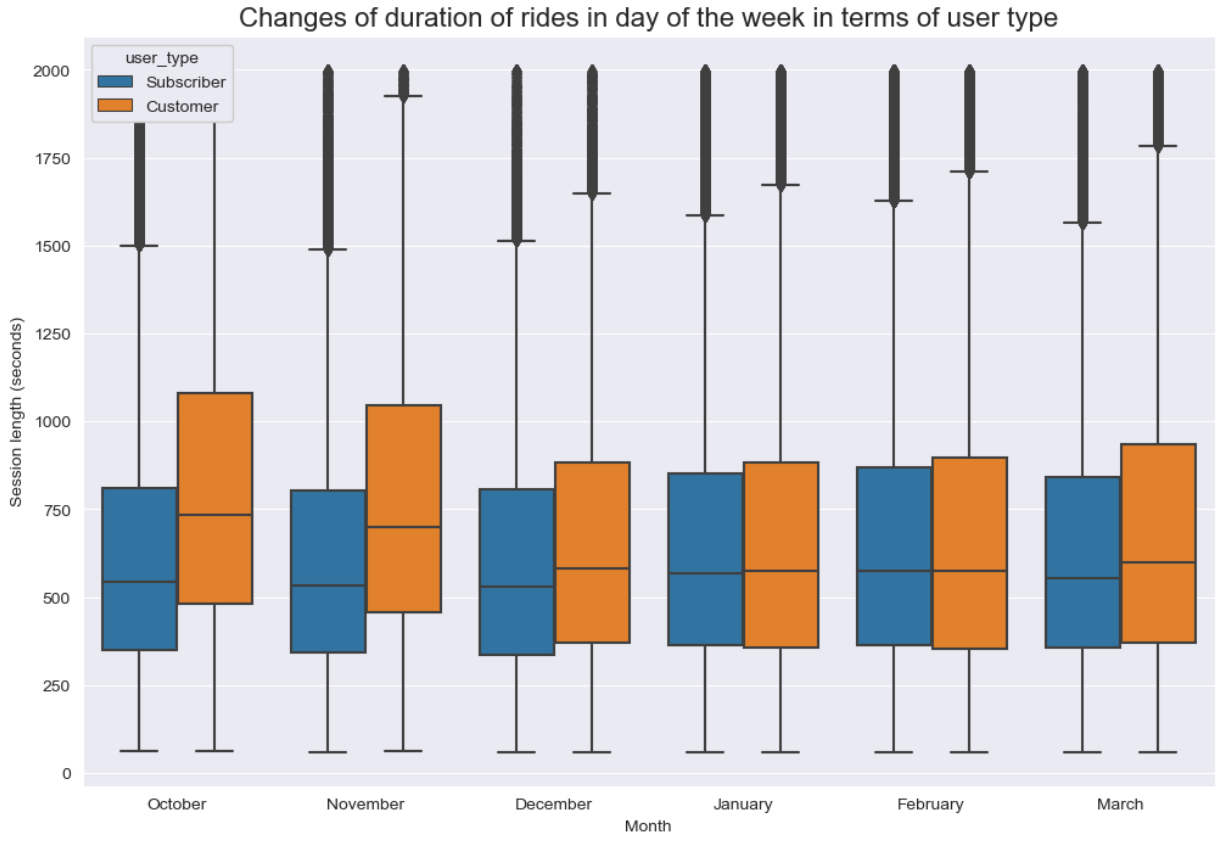


Interestingly, the number of rides in total witnesses a significant change.

1. Before March, weekdays have the highest amount of ride requests. In March, Sunday, Tuesday and Monday have the highest amount of ride requests. For more specifically, there is a much significant drop in the number of subscribers compared to customers. In other words, subscribers do not have to go to the office to work on weekdays, but they still go for exercise on Sunday (or because they feel sad for their subscription fee so they ride on Sunday).

Multivariate Exploration

In [47]:
```
plt.figure (figsize = (12,8))
sb.boxplot(data = df[df['duration_sec']<2000], x = 'start_month', y = 'durat
plt.legend(loc = 2, framealpha = 1, title = 'user_type')
plt.title('Changes of duration of rides in day of the week in terms of user
plt.ylabel('Session length (seconds)')
plt.xlabel('Month')
plt.show()
```

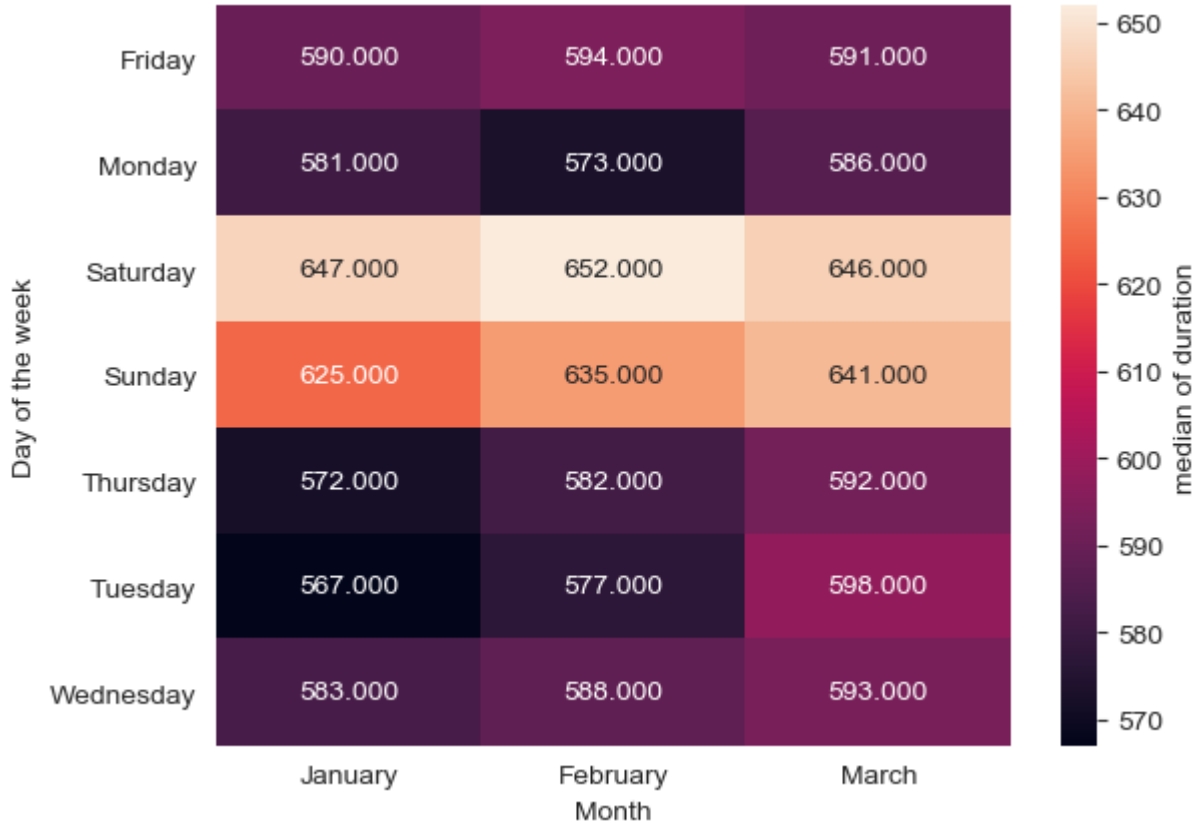


There is no significant changes in the duration of rides in terms of month and user_type, if considering the COVID-19 as a benchmark

Observe the changes of the median duration of day of the week over time

In [44]:

```
cat_median = df[df['start_month_number']<4].groupby(['start_month_number', '
cat_median = cat_median.reset_index(name = 'duration_sec_median')
cat_median = cat_median.pivot(index = 'start_day_of_week', columns = 'start_
                              values = 'duration_sec_median')
sb.heatmap(cat_median, annot = True, fmt = '.3f',
           cbar_kws = {'label' : 'median of duration'}, xticklabels=np.array
plt.xlabel ('Month')
plt.ylabel ('Day of the week')
plt.show()
```



As the heatmap describes, it seems that people in March has a slightly longer session compared to January and February. Further research needs to be done to understand this finding. One possible reason might come from the fact that "work-from-home" order allows people to have a flexible time, which allows people to ride more than usual. However, when looking up at the boxplot above, I recognize that this assumption is not true, especially when I compare with October, November and December of the previous year.

CONCLUSION

1. The COVID-19 definitely impacted the general number of bike rides in terms of session count.
2. The COVID-19 and shelter-in-place order have changed the way people use Lyft bike. For example, significantly lower number of people Lyft biking session in the morning, and more people are biking on Sunday (proportionally).
3. The COVID-19 and shelter-in-place order do not seem to change the session length.
4. The COVID-19 and shelter-in-place order changes the proportion of customer_type. A significantly lower number of subscribers do not use Lyft bike session compared to the traditional customers.