```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import missingno as msno
import seaborn as sns
```

```
july1_df = pd.read_csv("/content/202306-citibike-tripdata_1.csv", parse_dates=True)
july2_df = pd.read_csv("/content/202306-citibike-tripdata_2.csv", parse_dates=True)
july3_df = pd.read_csv("/content/202306-citibike-tripdata_3.csv", parse_dates=True)
july4_df = pd.read_csv("/content/202306-citibike-tripdata_4.csv", parse_dates=True)
```

```
<ipython-input-2-c5105525da06>:1: DtypeWarning: Columns (5,7) have mixed types. Specify dtype option on import or set low_me
  july1_df = pd.read_csv("/content/202306-citibike-tripdata_1.csv", parse_dates=True)
<ipython-input-2-c5105525da06>:2: DtypeWarning: Columns (5,7) have mixed types. Specify dtype option on import or set low_me
  july2_df = pd.read_csv("/content/202306-citibike-tripdata_2.csv", parse_dates=True)
<ipython-input-2-c5105525da06>:3: DtypeWarning: Columns (5,7) have mixed types. Specify dtype option on import or set low_me
  july3_df = pd.read_csv("/content/202306-citibike-tripdata_3.csv", parse_dates=True)
<ipython-input-2-c5105525da06>:4: DtypeWarning: Columns (5,7) have mixed types. Specify dtype option on import or set low_me
  july4_df = pd.read_csv("/content/202306-citibike-tripdata_4.csv", parse_dates=True)
```

```
dataset = pd.concat([july1_df, july2_df, july3_df, july4_df], axis = 0)
```

```
citibike_df = dataset.copy()
```

```
citibike_df.head()
```

	ride_id	rideable_type	started_at	ended_at	start_station_name	sta
0	984F50BCBC76DD9A	classic_bike	2023-06-11 06:54:21	2023-06-11 07:12:28	W 84 St & Columbus Ave	
1	03E3D62E7FB76B05	classic_bike	2023-06-19 15:23:11	2023-06-19 16:00:05	E 89 St & York Ave	
2	8E7EE421A0B8BBF3	classic_bike	2023-06-06 16:07:05	2023-06-06 16:15:14	E 51 St & 2 Ave	
3	24D66A0C46493CB1	classic_bike	2023-06-26 19:52:23	2023-06-26 19:55:47	India St Pier	
4	E944882A074B8F61	classic_bike	2023-06-05 08:57:57	2023-06-05 09:13:36	E 47 St & 2 Ave	

```
citibike_df.tail()
```

	ride_id	rideable_type	started_at	ended_at	start_station_name
560254	E6B04D2FD398493E	classic_bike	2023-06-22 18:06:15	2023-06-22 18:19:31	University Pl & E 8 St
560255	A9C0665FC912A63B	classic_bike	2023-06-12 09:16:47	2023-06-12 09:42:18	N 7 St & Driggs Ave
560256	C6B241D76A0137FD	classic_bike	2023-06-11 17:23:58	2023-06-11 18:02:36	12 Ave & W 125 St
560257	B218070FEA49C6B3	classic_bike	2023-06-27 18:16:21	2023-06-27 18:33:12	W 34 St & 11 Ave
560258	69FB0F587F4022B6	classic_bike	2023-06-02 14:25:05	2023-06-02 14:32:29	Sullivan St & Washington Sq

```
citibike_df.columns
```

```
Index(['ride_id', 'rideable_type', 'started_at', 'ended_at',
       'start_station_name', 'start_station_id', 'end_station_name',
       'end_station_id', 'start_lat', 'start_lng', 'end_lat', 'end_lng',
       'member_casual'],
      dtype='object')
```

```
citibike_df.describe()
```

	start_lat	start_lng	end_lat	end_lng
count	3.560259e+06	3.560259e+06	3.557533e+06	3.557533e+06
mean	4.074148e+01	-7.397230e+01	4.074132e+01	-7.397227e+01
std	4.193651e-02	2.837256e-02	5.621432e-02	7.362634e-02
min	4.057541e+01	-7.403721e+01	0.000000e+00	-7.407196e+01
25%	4.071512e+01	-7.399266e+01	4.071512e+01	-7.399280e+01
50%	4.073957e+01	-7.397994e+01	4.073936e+01	-7.397990e+01
75%	4.076413e+01	-7.395484e+01	4.076413e+01	-7.395485e+01
max	4.088559e+01	-7.384572e+01	4.093000e+01	0.000000e+00

```
citibike_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3560259 entries, 0 to 560258
Data columns (total 13 columns):
 #   Column              Dtype  
---  ------              -----  
 0   ride_id             object 
 1   rideable_type       object 
 2   started_at          object 
 3   ended_at            object 
 4   start_station_name  object 
 5   start_station_id    object 
 6   end_station_name    object 
 7   end_station_id      object 
 8   start_lat           float64
 9   start_lng           float64
 10  end_lat             float64
 11  end_lng             float64
 12  member_casual       object 
dtypes: float64(4), object(9)
memory usage: 380.3+ MB
```

```
citibike_df['started_at']= pd.to_datetime(citibike_df['started_at'])
citibike_df['ended_at']= pd.to_datetime(citibike_df['ended_at'])
```

```
citibike_df.isnull().sum()
```

```
ride_id                  0
rideable_type            0
started_at               0
ended_at                 0
start_station_name    1972
start_station_id      1972
end_station_name      7808
end_station_id        7808
start_lat                0
start_lng                0
end_lat               2726
end_lng               2726
member_casual            0
dtype: int64
```

```
#bc theres over 3,000,000 rows of data even 7808 of unknown rows won't effect the analysis and only 0% of the data is considered
data_loss = round((citibike_df['end_station_name'].isna().sum()/citibike_df.shape[0])*100)
data_loss
```

```
0
```

```
df_null = citibike_df[citibike_df.isnull().any(axis=1)]
print(df_null)
```

```
                 ride_id  rideable_type          started_at  \
267     68A6F87CE026DCC2   classic_bike 2023-06-13 16:43:24
273     DA6274B855F9804C   classic_bike 2023-06-02 23:51:19
288     8BDB421532501E7A   classic_bike 2023-06-09 14:29:04
291     59BFB2C2ACA54903   classic_bike 2023-06-23 15:22:28
295     6146781811F307E9   classic_bike 2023-06-16 23:52:15
...                  ...           ...                  ...
136574  E16B1135CFBA3B0E   classic_bike 2023-06-17 18:38:48
136590  6CAA335D23BF33C0   classic_bike 2023-06-22 01:51:51
136592  A297BF798A4598C4   classic_bike 2023-06-02 09:03:48
136596  D904A340A480898E   classic_bike 2023-06-25 21:01:15
136617  1CD65115DC4C5329  electric_bike 2023-06-17 18:07:18

                  ended_at             start_station_name start_station_id  \
267    2023-06-13 18:16:09                E 81 St & 1 Ave          7121.13
273    2023-06-03 01:27:04               3 Ave & E 169 St          8103.05
288    2023-06-09 15:31:56                7 Ave & Park Pl          4125.07
291    2023-06-23 16:41:27      South End Ave & Albany St          5114.08
295    2023-06-17 02:12:44              E 40 St & Park Ave          6432.11
...                    ...                           ...              ...
136574 2023-06-18 19:38:27            N 7 St & Driggs Ave          5340.01
136590 2023-06-22 03:57:33  Linden St & Knickerbocker Ave          4743.04
136592 2023-06-02 10:16:23               W 37 St & 10 Ave          6611.02
136596 2023-06-26 22:01:09    Bronx Shore Comfort Station          7618.01
136617 2023-06-18 19:07:02            N 7 St & Driggs Ave          5340.01

       end_station_name end_station_id  start_lat  start_lng  end_lat  \
267                 NaN            NaN  40.773428 -73.952337    40.76
273                 NaN            NaN  40.832635 -73.905223    40.84
288                 NaN            NaN  40.677515 -73.973371    40.66
291                 NaN            NaN  40.709901 -74.016592    40.74
295                 NaN            NaN  40.750832 -73.978598    40.77
...                 ...            ...        ...        ...      ...
136574              NaN            NaN  40.716967 -73.956388      NaN
136590              NaN            NaN  40.697160 -73.915709    40.65
136592              NaN            NaN  40.756580 -73.998186    40.76
136596              NaN            NaN  40.800460 -73.924040      NaN
136617              NaN            NaN  40.716967 -73.956388      NaN

        end_lng member_casual
267      -73.96        member
273      -73.90        casual
288      -73.96        member
291      -74.00        member
295      -73.95        member
...         ...           ...
136574      NaN        casual
136590   -73.98        casual
136592   -73.97        member
136596      NaN        casual
136617      NaN        member

[7859 rows x 13 columns]
```

```
citibike_df.dropna(axis=0, inplace=True)
citibike_df.isna().sum()
```

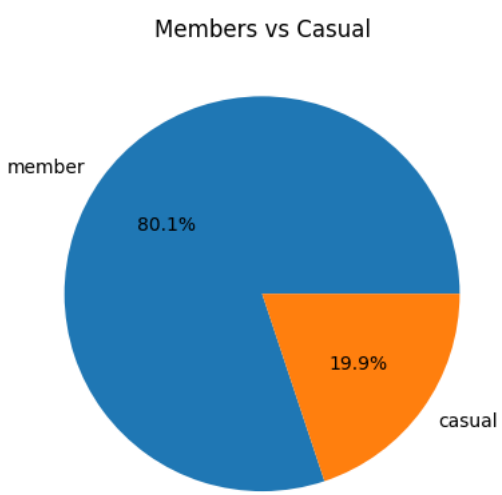
```
ride_id              0
rideable_type        0
started_at           0
ended_at             0
start_station_name   0
start_station_id     0
end_station_name     0
end_station_id       0
start_lat            0
start_lng            0
end_lat              0
end_lng              0
member_casual        0
dtype: int64
```

```
citibike_df.sum()
```

```
citibike_df.nunique()
```

Members vs casual

```
user_type_count = citibike_df['member_casual'].value_counts()
plt.pie(user_type_count.values, labels= user_type_count.index,autopct='%1.1f%%')
plt.title('Members vs Casual')
plt.show()
```



Majority of users are members

Duration

```
#calculate duration
duration = citibike_df['ended_at'] - citibike_df['started_at']

#create new column called duration
citibike_df['duration'] = duration
```

```
citibike_df['hour'] = citibike_df['started_at'].dt.hour

citibike_df['durationInSeconds'] = citibike_df['duration'].dt.total_seconds()
citibike_df['durationInMinutes'] = round((citibike_df['durationInSeconds']/60), 0)

citibike_df
```

	ride_id	rideable_type	started_at	ended_at	start_station_name
0	984F50BCBC76DD9A	classic_bike	2023-06-11 06:54:21	2023-06-11 07:12:28	W 84 St & Columbus Ave
1	03E3D62E7FB76B05	classic_bike	2023-06-19 15:23:11	2023-06-19 16:00:05	E 89 St & York Ave
2	8E7EE421A0B8BBF3	classic_bike	2023-06-06 16:07:05	2023-06-06 16:15:14	E 51 St & 2 Ave
3	24D66A0C46493CB1	classic_bike	2023-06-26 19:52:23	2023-06-26 19:55:47	India St Pier
4	E944882A074B8F61	classic_bike	2023-06-05 08:57:57	2023-06-05 09:13:36	E 47 St & 2 Ave
...	...	...	...	...	...
560254	E6B04D2FD398493E	classic_bike	2023-06-22 18:06:15	2023-06-22 18:19:31	University Pl & E 8 St
560255	A9C0665FC912A63B	classic_bike	2023-06-12 09:16:47	2023-06-12 09:42:18	N 7 St & Driggs Ave
560256	C6B241D76A0137FD	classic_bike	2023-06-11 17:23:58	2023-06-11 18:02:36	12 Ave & W 125 St
560257	B218070FEA49C6B3	classic_bike	2023-06-27 18:16:21	2023-06-27 18:33:12	W 34 St & 11 Ave
560258	69FB0F587F4022B6	classic_bike	2023-06-02 14:25:05	2023-06-02 14:32:29	Sullivan St & Washington Sq

3552400 rows × 17 columns

```
citibike_df['hour'].value_counts()
```

```
hour
17    328898
18    312354
16    266106
19    242383
15    235036
14    214192
13    200869
8     199299
12    192632
20    183122
9     171248
11    169199
10    150274
21    137418
7     130560
22    112281
23     82009
6      68677
0      51456
1      32183
5      25546
2      21179
3      13889
4      11590
Name: count, dtype: int64
```

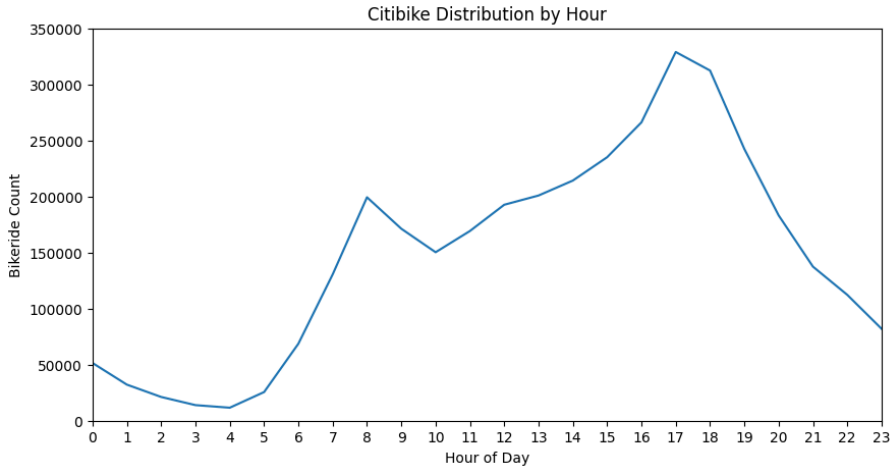
When do users ride?

Rides throughout the day (including weekends and weekdays)

Visualizing when users bike the most can help us understand why they're using them. For example, whether they're using Citibikes to commute to work or to see NYC.

```
rentals_per_hour = pd.value_counts(citibike_df['hour'])
rph_df = rentals_per_hour.to_frame()
rph_df.reset_index(inplace=True)

plt.figure(figsize=(10, 5))

ax = sns.lineplot(x = rentals_per_hour.index, y = rentals_per_hour, data=rph_df);
ax.set_xticks(range(24))
ax.set_xticklabels([i for i in range(0,24)])

plt.title("Citibike Distribution by Hour")
plt.xlabel("Hour of Day")
plt.ylabel("Bikeride Count")

ax.set(xlim=(0, 23))
ax.set(ylim=(0, 350000))
```

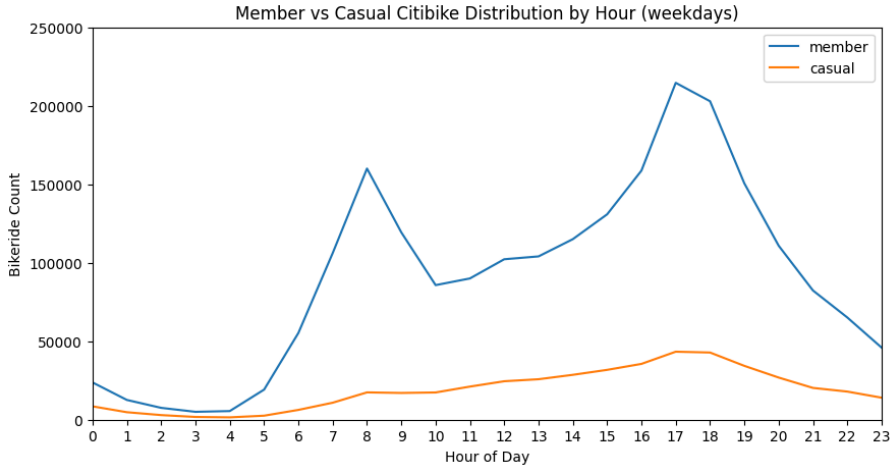
```
[(0.0, 350000.0)]
```



What times do users ride their bike during weekdays?

```
#fitering for only data on weekdays
#citibike_df['hour'] = citibike_df['started_at'].dt.hour
citibike_df['WEEKDAY'] = citibike_df['started_at'].dt.dayofweek

bins = [-1, 4, 7, np.inf]
weekday_weekend = ['Weekday','Weekend', '']

citibike_df['weekday_weekend'] = pd.cut(citibike_df['WEEKDAY'], bins, labels=weekday_weekend)

df2_member = citibike_df[(citibike_df.WEEKDAY.isin([5,6]) == False) & (citibike_df["member_casual"] == "member")]
time_member = pd.value_counts(df2_member['hour'])
time_member_df = time_member.to_frame()
time_member_df.reset_index(inplace = True)

df2_casual = citibike_df[(citibike_df.WEEKDAY.isin([5,6]) == False) & (citibike_df["member_casual"] == "casual")]
time_casual = pd.value_counts(df2_casual['hour'])
time_casual_df = time_casual.to_frame()
time_casual_df.reset_index(inplace = True)

plt.figure(figsize=(10,5))

ax = sns.lineplot(x = time_member.index, y = time_member, data=time_member_df, label= 'member')
ax = sns.lineplot(x = time_casual.index, y = time_casual, data=time_casual_df, label = 'casual')
ax.set(xlim=(0, 23))
ax.set(ylim=(0, 250000))

plt.title("Member vs Casual Citibike Distribution by Hour (weekdays)")
plt.xlabel("Hour of Day")
plt.ylabel("Bikeride Count")

ax.set_xticks(range(24))
ax.set_xticklabels([i for i in range(0,24)])
```

```
[Text(0, 0, '0'),
 Text(1, 0, '1'),
 Text(2, 0, '2'),
 Text(3, 0, '3'),
 Text(4, 0, '4'),
 Text(5, 0, '5'),
 Text(6, 0, '6'),
 Text(7, 0, '7'),
 Text(8, 0, '8'),
 Text(9, 0, '9'),
 Text(10, 0, '10'),
 Text(11, 0, '11'),
 Text(12, 0, '12'),
 Text(13, 0, '13'),
 Text(14, 0, '14'),
 Text(15, 0, '15'),
 Text(16, 0, '16'),
 Text(17, 0, '17'),
 Text(18, 0, '18'),
 Text(19, 0, '19'),
 Text(20, 0, '20'),
 Text(21, 0, '21'),
 Text(22, 0, '22'),
 Text(23, 0, '23')]
```

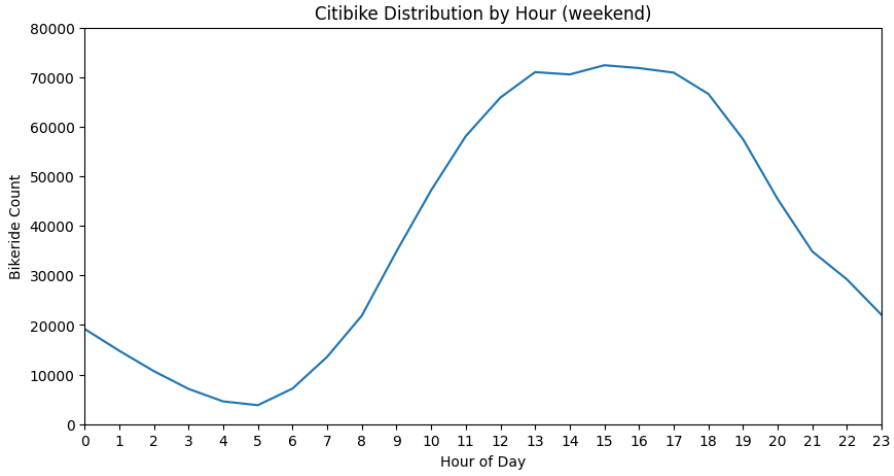


On weekdays, users that are members increased between 6-10 am and 4-11 pm due to work hours and commuting, while casual users remained consistent without as much of a significant increase. People who are commuting are members because when they use Citibikes almost everyday it's more cost effective to pay 219.99 a year than paying $ 4.79 for 30 minutes each trip. This shows that members who are commuting get more exercise as they use citibikes nearly everyday to get to work.

What times do users ride their bike during weekends?

```
df3 = citibike_df[citibike_df.WEEKDAY.isin([0,1,2,3,4]) == False]

timeWeekends = pd.value_counts(df3['hour'])
timeWeekends_df = timeWeekends.to_frame()
timeWeekends_df.reset_index(inplace = True)

plt.figure(figsize=(10,5))

ax = sns.lineplot(x = timeWeekends.index, y = timeWeekends, data=timeWeekends_df)
ax.set(xlim=(0, 23))
ax.set(ylim=(0, 80000))

plt.title("Citibike Distribution by Hour (weekend)")
plt.xlabel("Hour of Day")
plt.ylabel("Bikeride Count")

ax.set_xticks(range(24))
ax.set_xticklabels([i for i in range(0,24)])
```

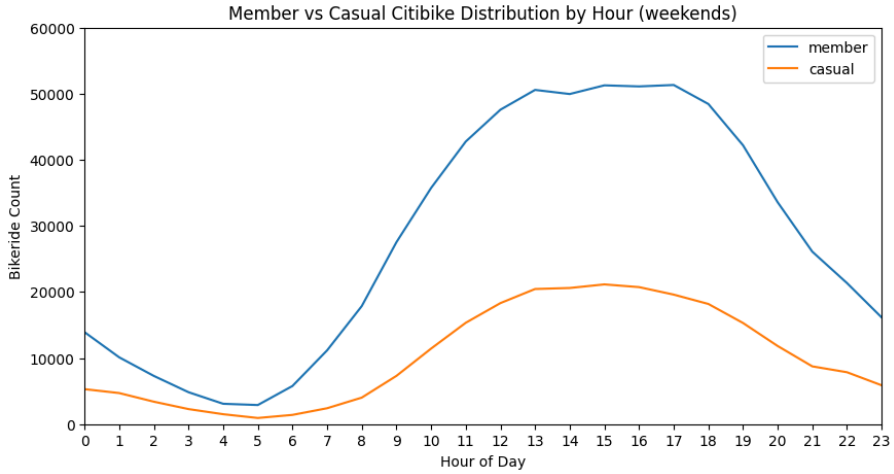
```
[Text(0, 0, '0'),
 Text(1, 0, '1'),
 Text(2, 0, '2'),
 Text(3, 0, '3'),
 Text(4, 0, '4'),
 Text(5, 0, '5'),
 Text(6, 0, '6'),
 Text(7, 0, '7'),
 Text(8, 0, '8'),
 Text(9, 0, '9'),
 Text(10, 0, '10'),
 Text(11, 0, '11'),
 Text(12, 0, '12'),
 Text(13, 0, '13'),
 Text(14, 0, '14'),
 Text(15, 0, '15'),
 Text(16, 0, '16'),
 Text(17, 0, '17'),
 Text(18, 0, '18'),
 Text(19, 0, '19'),
 Text(20, 0, '20'),
 Text(21, 0, '21'),
 Text(22, 0, '22'),
 Text(23, 0, '23')]
```



During the weekends, there's a major decrease in the overall number of citibikes used from an overall max of 328,898 during the weekdays to only a max of 72,419 during weekends.

```
df3_member = citibike_df[(citibike_df.WEEKDAY.isin([0,1,2,3,4]) == False) & (citibike_df['member_casual'] == 'member')]

timeWeekends_member = pd.value_counts(df3_member['hour'])
timeWeekends_member_df = timeWeekends_member.to_frame()
timeWeekends_member_df.reset_index(inplace = True)

df3_casual = citibike_df[(citibike_df.WEEKDAY.isin([0,1,2,3,4]) == False) & (citibike_df['member_casual'] == 'casual')]
timeWeekends_casual = pd.value_counts(df3_casual['hour'])
timeWeekends_casual_df = timeWeekends_casual.to_frame()
timeWeekends_casual_df.reset_index(inplace = True)

plt.figure(figsize=(10,5))

ax = sns.lineplot(x = timeWeekends_member.index, y = timeWeekends_member, data=timeWeekends_member_df, label = 'member')
ax = sns.lineplot(x = timeWeekends_casual.index, y = timeWeekends_casual, data=timeWeekends_casual_df, label = 'casual')

ax.set(xlim=(0, 23))
ax.set(ylim=(0, 60000))

plt.title("Member vs Casual Citibike Distribution by Hour (weekends)")
plt.xlabel("Hour of Day")
plt.ylabel("Bikeride Count")


ax.set_xticks(range(24))
ax.set_xticklabels([i for i in range(0,24)])
```

```
[Text(0, 0, '0'),
 Text(1, 0, '1'),
 Text(2, 0, '2'),
 Text(3, 0, '3'),
 Text(4, 0, '4'),
 Text(5, 0, '5'),
 Text(6, 0, '6'),
 Text(7, 0, '7'),
 Text(8, 0, '8'),
 Text(9, 0, '9'),
 Text(10, 0, '10'),
 Text(11, 0, '11'),
 Text(12, 0, '12'),
 Text(13, 0, '13'),
 Text(14, 0, '14'),
 Text(15, 0, '15'),
 Text(16, 0, '16'),
 Text(17, 0, '17'),
 Text(18, 0, '18'),
 Text(19, 0, '19'),
 Text(20, 0, '20'),
 Text(21, 0, '21'),
 Text(22, 0, '22'),
 Text(23, 0, '23')]
```



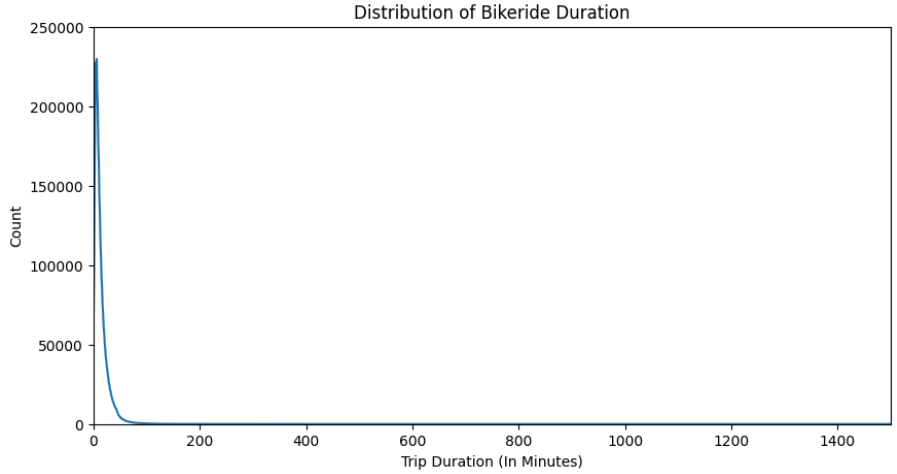
Unlike weekdays, during weekends customers and users have a similar distribution where most biked between 8am and 10pm. In addition, there are many more members than casual users riding citibikes. As users have more free time during the weekend, Citibike could provide more of an incentive by offering a lower price or allowing them to use bikes for a longer time on weekends to encourage being more active

Bike Ride Duration

Analyzing the duration helps understand how long users are riding bikes and how much exercise they're getting. The higher the duration, the more exercise the users get.

```
duration_per_hour = pd.value_counts(citibike_df['durationInMinutes'])
dph_df = duration_per_hour.to_frame()
dph_df.reset_index(inplace=True)

plt.figure(figsize=(10,5))

ax = sns.lineplot(x = duration_per_hour.index, y = duration_per_hour, data=dph_df)
ax.set(xlim=(0, 1500))
ax.set(ylim=(0, 250000))

plt.title("Distribution of Bikeride Duration")
plt.xlabel("Trip Duration (In Minutes)")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

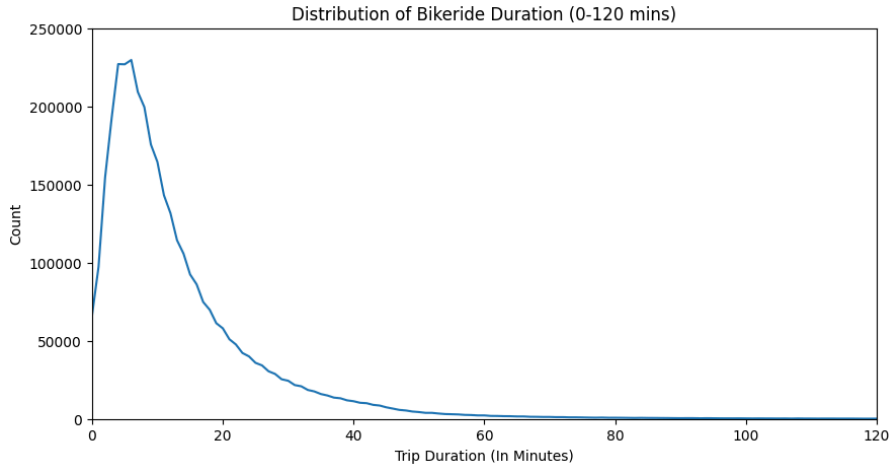


Majority of bikes were ridden for less than 45 minutes. The bikes taken for 120+ minutes were most likely stolen, lost, or mistakes of ppl forgetting to return bikes as after 30 minutes for casual users and 45 minutes for members it costs $4 for every extra 15 minutes which is a high price to pay. So we'll drop these values.

```
citibike_df = citibike_df.loc[~(citibike_df['durationInMinutes'] > 120)]
```

```
duration_per_hour = pd.value_counts(citibike_df['durationInMinutes'])
dph_df = duration_per_hour.to_frame()
dph_df.reset_index(inplace=True)

plt.figure(figsize=(10,5))

duration_per_hour
ax = sns.lineplot(x = duration_per_hour.index, y = duration_per_hour, data=dph_df)

ax.set(xlim=(0, 120))
ax.set(ylim=(0, 250000))

plt.title("Distribution of Bikeride Duration (0-120 mins)")
plt.xlabel("Trip Duration (In Minutes)")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```



Most bike rides are between 0 and 20 minutes which isn't very long. In addition, there are 66,167 bikes that were used for 0 minutes and 97,379 bikes that were used for 1 minute, which isn't enough time to bike anywhere. This is most likely due to users taking out a bike and putting it back due to a mistake or getting a broken bike so we'll remove these values if the start and end station are the same.

```
duration_per_hour = pd.value_counts(citibike_df['durationInMinutes'])
duration_per_hour.head(n=20)
```

```
durationInMinutes
6.0     229671
4.0     227044
5.0     226900
7.0     209234
8.0     199529
3.0     192414
9.0     175583
10.0    164387
2.0     154514
11.0    143097
12.0    131615
13.0    114339
14.0    105710
1.0      97379
15.0     92460
16.0     86107
17.0     74812
18.0     69839
0.0      66167
19.0     61264
Name: count, dtype: int64
```

```
#drop duration in minutes that are is only 0 or 1 minute long if start and end station are the same
citibike_df = citibike_df.loc[~(((citibike_df['durationInMinutes'] == 0) & (citibike_df['durationInMinutes'] == 1)) & (citibike_
```

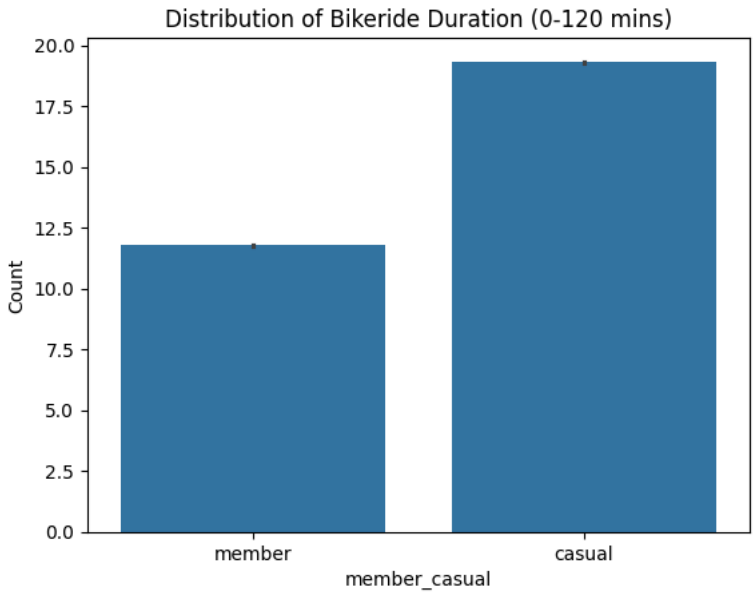
Member vs Casual user Trip Duration

```
citibike_df.groupby('member_casual').durationInMinutes.mean()
```

```
member_casual
casual    19.300538
member    11.773360
Name: durationInMinutes, dtype: float64
```

```
#plt.figure(figsize=(10,5))
ax = sns.barplot(x = 'member_casual', y = 'durationInMinutes', data=citibike_df)
plt.title("Distribution of Bikeride Duration (0-120 mins)")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

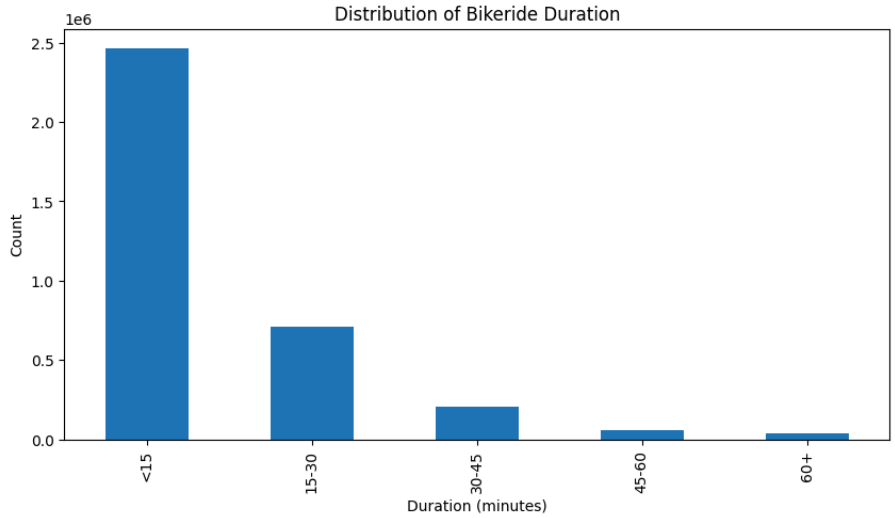


Casual users biked for a longer duration than members which was surprising but it does make sense as casual users use citibikes less often than members and pay more for each trip so want to ride for a longer time to get the most benefit. As can see above the average for casual users is 19 minutes when the time limit is 30 minutes. So unless there's a problem with not having enough bikes for everyone, Citibike could consider having a higher time limit which further encourages being active.

```
#Dividing duration into categories
bins = [0, 15, 30, 45, 60, np.inf]
minutesCategory = ['<15', '15-30', '30-45', '45-60', '60+']

citibike_df['minutesCat'] = pd.cut(citibike_df['durationInMinutes'], bins, labels=minutesCategory)
```

```
plt.figure(figsize=(10,5))

citibike_df['minutesCat'].value_counts().plot(kind='bar')
plt.title("Distribution of Bikeride Duration")
plt.xlabel("Duration (minutes)")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

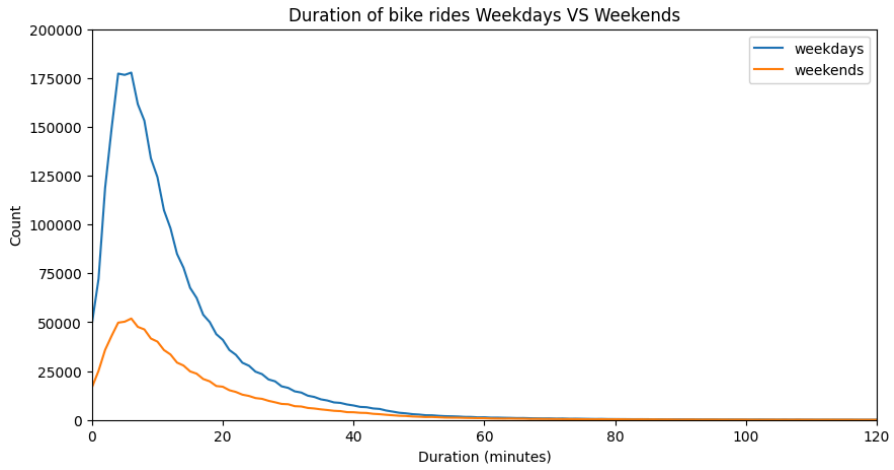


Majority of bikers ride for less than 15 minutes, followed by 15-30 and it gets lower as we go further after 45. This does make sense as if the user is a member they can bike for free upto 45 minutes and if they're a casual user they get up to 30 minutes for free, so there are much less users that ride for over 45 minutes. But due to users having to put their bikes back within the specified time this can discourge users from riding bikes as long as they want to and users may put the bike back and get another one to continue riding and avoid overtime fee of $4 for 15 minutes

Duration of bike rides Weekdays VS Weekends

```
df3_weekends = citibike_df[citibike_df.WEEKDAY.isin([0,1,2,3,4]) == False]
durationWeekends = pd.value_counts(df3_weekends['durationInMinutes'])
timeWeekends_df = durationWeekends.to_frame()
timeWeekends_df.reset_index(inplace = True)

df3_weekday = citibike_df[citibike_df.WEEKDAY.isin([5,6]) == False]
durationWeekday = pd.value_counts(df3_weekday['durationInMinutes'])
timeWeekday_df = durationWeekday.to_frame()
timeWeekday_df.reset_index(inplace = True)

plt.figure(figsize=(10,5))

ax = sns.lineplot(x = durationWeekday.index, y = durationWeekday, data=timeWeekday_df, label = 'weekdays')
ax = sns.lineplot(x = durationWeekends.index, y = durationWeekends, data=timeWeekends_df, label = 'weekends')

ax.set(xlim=(0, 120))
ax.set(ylim=(0, 200000))

plt.title("Duration of bike rides Weekdays VS Weekends")
plt.xlabel("Duration (minutes)")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

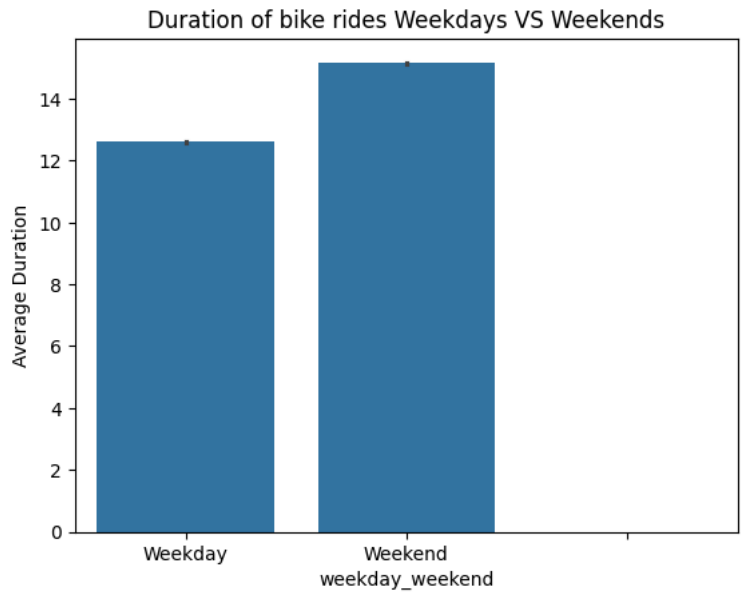



```
citibike_df.groupby('weekday_weekend').durationInMinutes.mean()
```

```
weekday_weekend
Weekday    12.599312
Weekend    15.149218
               NaN
Name: durationInMinutes, dtype: float64
```

```
ax = sns.barplot(x = 'weekday_weekend', y = 'durationInMinutes', data=citibike_df)

plt.title("Duration of bike rides Weekdays VS Weekends")
plt.ylabel("Average Duration")
```

```
Text(0, 0.5, 'Average Duration')
```



The average duration biked during weekends is slightly higher than weekdays which makes sense as people have more free time to bike longer and farther

What type of Bikes are people using?

```
rideable_type = citibike_df['rideable_type'].value_counts()
fig,ax=plt.subplots(figsize=(10,5))
ax.bar(x=rideable_type.index, height=rideable_type)
ax.set_ylabel("Count",fontsize=10)

plt.title("Distribution of Bike Type")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```