Analysis of FordGo Bike trends

(January 2018 - April 2019)

Inroduction

FordGo bike now Bay Wheels is a regional public bicycle sharing system in the San Francisco Bay Area, California operated by Motivate (a company based in New York City that operates bicycle sharing systems in the United States), in a partnership with the Metropolitan Transportation Commission and the Bay Area Air Quality Management District. Beginning operation in August 2013 as Bay Area Bike Share, the Bay Wheels system currently has over 2,600 bicycles in 262 stations across San Francisco, East Bay and San Jose. On June 28, 2017, the system officially re-launched as Ford GoBike in a partnership with Ford Motor Company. After Motivate's acquisition by Lyft, the system was subsequently renamed to Bay Wheels in June 2019. The system is expected to expand to 7,000 bicycles around 540 stations in San Francisco, Oakland, Berkeley, Emeryville, and San Jose. Bay Wheels is the first regional and large-scale bicycle sharing system deployed in California and on the West Coast of the United States.

As of January 2018, the system had seen nearly 500,000 rides since the launch in 2017 and had about 10,000 annual subscribers.

Data Wrangling

The data set contains ride of each user over a time period of January 2019 to April 2019.

In [1]:
```
# importing necessary libraries
import os
import glob
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
from math import radians, sin, cos, acos
import requests
import zipfile

%matplotlib inline
```

Data Collection

We gather the data. The code below will download, unzip and merge the data together to a final *.csv file from January 2018 to April 2019.

In [2]:
```
#defining filenames to be download
year_data = [x for x in range(201801, 201813)] + [x for x in range(201901, 201
905)]

for year in year_data:

    url = f"https://s3.amazonaws.com/fordgobike-data/{year}-fordgobike-tripdat
a.csv.zip"
    response = requests.get(url)

    #saving file
    with open(f"./Data/{year}-fordgobike-tripdata.csv.zip", mode = "wb") as fi
le:
        file.write(response.content)
```

In [3]:
```
#defining file names

files = [x for x in os.walk("./Data/")][0][2]

#loop over file names
for x in files:
    if ".zip" in x:
        with zipfile.ZipFile(f"./Data/{x}",'r') as zip_ref:
            zip_ref.extractall("./Data/")
```

In [4]:
```
#saving each file to a single csv

path = r'./Data/'
file1 = glob.glob(os.path.join(path, "*.csv"))

df = pd.concat((pd.read_csv(a) for a in file1), ignore_index = True)

df.to_csv('fordgo_master.csv', index = False)
```

In [5]:
```
#checking data

df = pd.read_csv('fordgo_master.csv')
df.info(null_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2734625 entries, 0 to 2734624
Data columns (total 16 columns):
duration_sec               2734625 non-null int64
start_time                 2734625 non-null object
end_time                   2734625 non-null object
start_station_id           2722124 non-null float64
start_station_name         2722124 non-null object
start_station_latitude     2734625 non-null float64
start_station_longitude    2734625 non-null float64
end_station_id             2722124 non-null float64
end_station_name           2722124 non-null object
end_station_latitude       2734625 non-null float64
end_station_longitude      2734625 non-null float64
bike_id                    2734625 non-null int64
user_type                  2734625 non-null object
member_birth_year          2583000 non-null float64
member_gender              2583354 non-null object
bike_share_for_all_trip    2734625 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 333.8+ MB
```

In [6]:

```
#sampling the data for checking for the required changes to be made.

df.sample(20)
```

Out[6]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_sta
752252	237	2018-06-11 20:14:40.0170	2018-06-11 20:18:37.1700	176.0	MacArthur BART Station	
2165042	273	2019-02-12 19:37:43.8080	2019-02-12 19:42:17.7540	157.0	65th St at Hollis St	
834776	683	2018-07-29 16:52:16.7180	2018-07-29 17:03:40.6570	88.0	11th St at Bryant St	
1868341	2835	2019-01-31 14:47:14.4930	2019-01-31 15:34:30.1810	198.0	Snow Park	
1610254	660	2018-11-28 16:19:07.5410	2018-11-28 16:30:08.3430	7.0	Frank H Ogawa Plaza	
1719651	703	2018-11-02 17:48:15.9540	2018-11-02 17:59:59.3660	81.0	Berry St at 4th St	
1937637	302	2019-01-23 08:31:52.1280	2019-01-23 08:36:54.4740	5.0	Powell St BART Station (Market St at 5th St)	
974755	406	2018-07-09 08:21:00.1790	2018-07-09 08:27:46.7770	17.0	Embarcadero BART Station (Beale St at Market St)	
2283229	402	2019-03-26 20:08:39.0270	2019-03-26 20:15:21.1820	90.0	Townsend St at 7th St	
1194526	289	2018-08-03 09:10:03.8160	2018-08-03 09:14:53.1640	15.0	San Francisco Ferry Building (Harry Bridges Pl...	
358352	767	2018-04-21 19:29:02.7870	2018-04-21 19:41:50.6440	15.0	San Francisco Ferry Building (Harry Bridges Pl...	
1275957	892	2018-09-20 15:18:43.2330	2018-09-20 15:33:35.4990	19.0	Post St at Kearny St	
505647	446	2018-05-21 11:29:57.1320	2018-05-21 11:37:23.5630	47.0	4th St at Harrison St	
714558	738	2018-06-17 17:34:45.5700	2018-06-17 17:47:03.6360	39.0	Scott St at Golden Gate Ave	
2670697	615	2019-04-07 21:06:26.9620	2019-04-07 21:16:42.4570	109.0	17th St at Valencia St	
69929	756	2018-01-11 08:48:49.8220	2018-01-11 09:01:26.2440	44.0	Civic Center/UN Plaza BART Station (Market St ...	
2317647	1569	2019-03-22 08:40:18.0210	2019-03-22 09:06:27.7940	324.0	Union Square (Powell St at Post St)	
898875	344	2018-07-19 17:41:16.5950	2018-07-19 17:47:00.9600	84.0	Duboce Park	
405932	3678	2018-04-10 16:39:24.2530	2018-04-10 17:40:42.4550	196.0	Grand Ave at Perkins St	
1970747	378	2019-01-17 17:25:37.1580	2019-01-17 17:31:56.0930	6.0	The Embarcadero at Sansome St	

In [7]:
```
#checking for in-depth description for the data

df.describe()
```

Out[7]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id
count	2.734625e+06	2.722124e+06	2.734625e+06	2.734625e+06	2.722124e+06
mean	8.316217e+02	1.258610e+02	3.776825e+01	-1.223510e+02	1.243537e+02
std	2.232948e+03	1.052229e+02	1.057828e-01	1.684623e-01	1.052322e+02
min	6.100000e+01	3.000000e+00	0.000000e+00	-1.224737e+02	3.000000e+00
25%	3.460000e+02	3.700000e+01	3.777041e+01	-1.224117e+02	3.300000e+01
50%	5.500000e+02	9.200000e+01	3.778107e+01	-1.223974e+02	9.000000e+01
75%	8.610000e+02	1.960000e+02	3.779728e+01	-1.222894e+02	1.960000e+02
max	8.636600e+04	4.200000e+02	4.551000e+01	0.000000e+00	4.200000e+02

In [8]:
```
#checking for duplicate values

df.duplicated().sum()
```

Out[8]: 0

In [9]:
```
#checking for NaN values

df.isna().sum()
```

Out[9]:
```
duration_sec                    0
start_time                      0
end_time                        0
start_station_id            12501
start_station_name          12501
start_station_latitude          0
start_station_longitude         0
end_station_id              12501
end_station_name            12501
end_station_latitude            0
end_station_longitude           0
bike_id                         0
user_type                       0
member_birth_year          151625
member_gender              151271
bike_share_for_all_trip         0
dtype: int64
```

Quality issues

1. start time and end time datatype needs to be changed to timestamps
2. bike id, start_station_id, end_station_id can be changed to object
3. user type, gender and bike_share_for_all_trip can be changed to category
4. age column can be added by calculating it by year of birth
5. we can calculate the details like month, day, hour

Data Cleaning

In [10]:
```
#creating copy of data

df_clean = df.copy()
```

Define

Changing the datatypes of the columns as mentioned in the Quality issues.

Code

In [11]:
```
#changing the datatype of start time and end time to timestamps

df_clean.start_time = pd.to_datetime(df_clean.start_time)
df_clean.end_time = pd.to_datetime(df_clean.end_time)
```

In [12]:
```
# changing the datatype of bike id, start_station_id, end_station_id to object

df_clean.bike_id = df_clean.bike_id.astype(str)
df_clean.start_station_id = df_clean.bike_id.astype(str)
df_clean.end_station_id = df_clean.bike_id.astype(str)
```

In [38]:
```
#changing the datatype of user type, gender, bike_share_for_all_trip to catego
ry

df_clean.user_type = df_clean.user_type.astype('category')
df_clean.member_gender = df_clean.member_gender.astype('category')
df_clean.bike_share_for_all_trip = df_clean.bike_share_for_all_trip.astype('ca
tegory')
df_clean.member_age = df_clean.member_age.astype('category')
```

Test

In [15]:
```
df_clean.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2734625 entries, 0 to 2734624
Data columns (total 16 columns):
duration_sec               2734625 non-null int64
start_time                 2734625 non-null datetime64[ns]
end_time                   2734625 non-null datetime64[ns]
start_station_id           2734625 non-null object
start_station_name         2722124 non-null object
start_station_latitude     2734625 non-null float64
start_station_longitude    2734625 non-null float64
end_station_id             2734625 non-null object
end_station_name           2722124 non-null object
end_station_latitude       2734625 non-null float64
end_station_longitude      2734625 non-null float64
bike_id                    2734625 non-null object
user_type                  2734625 non-null category
member_birth_year          2583000 non-null float64
member_gender              2583354 non-null category
bike_share_for_all_trip    2734625 non-null category
dtypes: category(3), datetime64[ns](2), float64(5), int64(1), object(5)
memory usage: 279.1+ MB
```

Define

Adding a new column member_age by calculating it with member_birth_year

Code

In [16]:
```
#subtracting by current year

df_clean['member_age'] = 2019-df_clean['member_birth_year']
```

Test

In [17]:
```
df_clean.sample(20)
```

Out[17]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
1336287	409	2018-09-11 17:28:58.770	2018-09-11 17:35:47.820	3234	San Pablo Park	
1828278	207	2018-12-08 16:36:59.034	2018-12-08 16:40:26.679	2664	Market St at Dolores St	
910002	478	2018-07-18 09:46:03.595	2018-07-18 09:54:02.304	167	7th St at Brannan St	
801607	537	2018-06-04 08:48:22.610	2018-06-04 08:57:20.315	3414	Derby St at College Ave	
1612528	291	2018-11-28 08:36:26.620	2018-11-28 08:41:17.815	1363	Milvia St at Derby St	
1773980	222	2018-12-19 09:36:20.011	2018-12-19 09:40:02.626	3796	Fountain Alley at S 2nd St	
514225	543	2018-05-19 12:36:33.299	2018-05-19 12:45:36.601	557	Davis St at Jackson St	
1569732	736	2018-10-04 18:05:21.128	2018-10-04 18:17:37.967	3676	Bryant St at 15th St	
2321161	2411	2019-03-21 18:10:35.337	2019-03-21 18:50:46.924	3139	Harrison St at 17th St	
510422	802	2018-05-20 12:22:42.013	2018-05-20 12:36:04.613	627	11th St at Natoma St	
533130	284	2018-05-16 10:54:36.336	2018-05-16 10:59:21.201	1238	MacArthur Blvd at Telegraph Ave	
2280643	270	2019-03-27 08:55:05.493	2019-03-27 08:59:35.685	6094	Montgomery St BART Station (Market St at 2nd St)	
1175328	1322	2018-08-06 19:48:04.064	2018-08-06 20:10:06.341	1394	Harmon St at Adeline St	
2241097	424	2019-03-31 17:26:33.625	2019-03-31 17:33:37.970	4315	29th St at Church St	
838676	566	2018-07-28 16:57:31.850	2018-07-28 17:06:57.899	2130	Adeline St at 40th St	
1594143	842	2018-10-01 15:47:43.376	2018-10-01 16:01:46.133	1313	14th St at Mission St	
1267229	201	2018-09-21 17:18:40.619	2018-09-21 17:22:01.659	1686	Market St at 10th St	
1826681	426	2018-12-09 11:52:38.723	2018-12-09 11:59:45.110	3635	Telegraph Ave at 27th St	
744004	783	2018-06-12 22:01:16.877	2018-06-12 22:14:20.072	82	Powell St BART Station (Market St at 4th St)	
585508	1270	2018-05-07 17:33:14.652	2018-05-07 17:54:24.984	3856	The Embarcadero at Sansome St	

Define

Adding detailed columns for month, day and hour.

Code

In [18]:
```
# extracting month name to new column start_time_month_name

df_clean['start_time_month_name'] = df_clean['start_time'].dt.strftime('%B')
```

In [19]:
```
# extracting start time month number to new column start_time_month

df_clean['start_time_month'] = df_clean['start_time'].dt.month.astype(int)
```

In [20]:
```
# extracting start time day to new column start_time_day

df_clean['start_time_day'] = df_clean['start_time'].dt.strftime('%a')
```

In [21]:
```
# extracting start date to new column start_date

df_clean['start_date'] = df_clean['start_time'].dt.day.astype(int)
```

In [22]:
```
# extracting start time hour to new column start_time_hour

df_clean['start_time_hour'] = df_clean['start_time'].dt.hour
```

Test

In [23]:

```
#Sampling the data

df_clean.sample(20)
```

Out[23]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
1772434	466	2018-12-19 14:37:47.383	2018-12-19 14:45:34.249	2165	Yerba Buena Center for the Arts (Howard St at ...	
932094	1389	2018-07-14 22:06:11.328	2018-07-14 22:29:21.208	2414	Telegraph Ave at 27th St	
165484	394	2018-02-09 10:03:00.095	2018-02-09 10:09:35.035	647	16th St Mission BART Station 2	
89853	472	2018-01-03 07:56:51.054	2018-01-03 08:04:43.141	558	Market St at 10th St	
185458	137	2018-02-05 10:47:00.849	2018-02-05 10:49:18.392	378	14th St at Mandela Pkwy	
522356	807	2018-05-17 20:11:59.155	2018-05-17 20:25:26.564	3928	San Francisco Caltrain (Townsend St at 4th St)	
70718	1423	2018-01-11 06:29:35.852	2018-01-11 06:53:19.835	2634	The Embarcadero at Sansome St	
1153698	295	2018-08-09 18:59:34.166	2018-08-09 19:04:30.067	2405	Oregon St at Adeline St	
2267065	662	2019-03-28 16:15:21.778	2019-03-28 16:26:24.040	6076	Beale St at Harrison St	
2541771	710	2019-04-23 19:31:40.315	2019-04-23 19:43:30.470	2291	Koshland Park	
1213746	451	2018-09-30 11:21:32.154	2018-09-30 11:29:03.278	977	Valencia St at Clinton Park	
60046	295	2018-01-14 13:47:51.881	2018-01-14 13:52:47.374	887	Grand Ave at Perkins St	
649540	382	2018-06-27 08:31:01.858	2018-06-27 08:37:24.083	2644	18th St at Noe St	
730868	277	2018-06-14 17:22:55.061	2018-06-14 17:27:32.877	3909	San Francisco Caltrain (Townsend St at 4th St)	
2498287	328	2019-04-30 17:09:57.938	2019-04-30 17:15:26.586	238	Hyde St at Post St	
2375627	520	2019-03-16 08:48:21.381	2019-03-16 08:57:01.820	4040	Morrison Ave at Julian St	
2192110	396	2019-02-08 10:09:22.808	2019-02-08 10:15:59.003	4974	Montgomery St BART Station (Market St at 2nd St)	
1738524	288	2018-12-29 19:02:04.285	2018-12-29 19:06:52.619	5511	59th St at Horton St	
769639	1947	2018-06-08 14:02:26.271	2018-06-08 14:34:53.741	1107	Laguna St at Hayes St	
372149	1303	2018-04-18 18:07:52.905	2018-04-18 18:29:36.300	2043	Howard St at 2nd St	

20 rows × 22 columns

In [25]:
```
df_clean.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2734625 entries, 0 to 2734624
Data columns (total 22 columns):
duration_sec                2734625 non-null int64
start_time                  2734625 non-null datetime64[ns]
end_time                    2734625 non-null datetime64[ns]
start_station_id            2734625 non-null object
start_station_name          2722124 non-null object
start_station_latitude      2734625 non-null float64
start_station_longitude     2734625 non-null float64
end_station_id              2734625 non-null object
end_station_name            2722124 non-null object
end_station_latitude        2734625 non-null float64
end_station_longitude       2734625 non-null float64
bike_id                     2734625 non-null object
user_type                   2734625 non-null category
member_birth_year           2583000 non-null float64
member_gender               2583354 non-null category
bike_share_for_all_trip     2734625 non-null category
member_age                  2583000 non-null float64
start_time_month_name       2734625 non-null object
start_time_month            2734625 non-null int32
start_time_day              2734625 non-null object
start_date                  2734625 non-null int32
start_time_hour             2734625 non-null int64
dtypes: category(3), datetime64[ns](2), float64(6), int32(2), int64(2), objec
t(7)
memory usage: 383.4+ MB
```

Define

Changing the datatype for member_age and start_time_day to integer

Code

In [41]:
```
# changing member_age datatype to integer

df_clean.member_age = df_clean.member_age.astype(int)
```

Test

In [42]:
```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2734625 entries, 0 to 2734624
Data columns (total 22 columns):
duration_sec               int64
start_time                 datetime64[ns]
end_time                   datetime64[ns]
start_station_id           object
start_station_name         object
start_station_latitude     float64
start_station_longitude    float64
end_station_id             object
end_station_name           object
end_station_latitude       float64
end_station_longitude      float64
bike_id                    object
user_type                  category
member_birth_year          float64
member_gender              category
bike_share_for_all_trip    category
member_age                 int32
start_time_month_name      object
start_time_month           int32
start_time_day             object
start_date                 int32
start_time_hour            int64
dtypes: category(3), datetime64[ns](2), float64(5), int32(3), int64(2), objec
t(7)
memory usage: 372.9+ MB
```

In [46]:

```
# sampling the data

df_clean.sample(50)
```

Out[46]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
1106450	1486	2018-08-17 10:56:28.280	2018-08-17 11:21:14.493	3658	Harrison St at 17th St	
94877	834	2018-02-28 21:33:36.793	2018-02-28 21:47:31.231	2757	Market St at Franklin St	
1248491	375	2018-09-25 08:32:35.322	2018-09-25 08:38:51.027	1500	Embarcadero BART Station (Beale St at Market St)	
1085233	399	2018-08-21 12:16:29.855	2018-08-21 12:23:09.228	1306	College Ave at Harwood Ave	
232648	783	2018-03-24 16:38:28.345	2018-03-24 16:51:31.742	1498	Market St at Franklin St	
478784	277	2018-05-25 15:01:29.883	2018-05-25 15:06:07.008	523	Montgomery St BART Station (Market St at 2nd St)	
2439081	77	2019-03-08 17:50:16.031	2019-03-08 17:51:33.164	1494	Powell St BART Station (Market St at 4th St)	
2200277	517	2019-02-07 12:07:51.011	2019-02-07 12:16:28.144	4453	Steuart St at Market St	
1656220	180	2018-11-14 09:57:34.104	2018-11-14 10:00:35.047	3587	Division St at Potrero Ave	
2668655	685	2019-04-08 08:23:47.481	2019-04-08 08:35:12.857	1285	Market St at 10th St	
2047843	648	2019-01-02 20:20:49.847	2019-01-02 20:31:38.083	4953	Washington St at Kearny St	
1465957	262	2018-10-20 22:26:32.443	2018-10-20 22:30:54.742	1590	SAP Center	
476014	239	2018-05-25 22:06:06.472	2018-05-25 22:10:06.162	3497	Beale St at Harrison St	
1687394	368	2018-11-08 07:31:07.574	2018-11-08 07:37:16.246	1123	Bay Pl at Vernon St	
1295720	312	2018-09-17 21:18:52.983	2018-09-17 21:24:05.623	4065	Valencia St at 16th St	
171295	477	2018-02-08 08:45:18.561	2018-02-08 08:53:16.065	2594	7th St at Brannan St	
2215039	460	2019-02-05 18:10:06.481	2019-02-05 18:17:46.915	5051	Howard St at 8th St	
2344929	604	2019-03-19 15:47:17.689	2019-03-19 15:57:21.725	6628	San Fernando St at 7th St	
1281769	528	2018-09-19 18:20:32.497	2018-09-19 18:29:20.589	64	Lake Merritt BART Station	
2477990	378	2019-03-04 08:45:13.189	2019-03-04 08:51:31.266	6570	Folsom St at 3rd St	

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_statio
14759	2133	2018-01-28 14:15:31.969	2018-01-28 14:51:05.155	3160	Jackson Playground	
129231	700	2018-02-19 08:42:54.049	2018-02-19 08:54:34.592	2128	Sanchez St at 15th St	
2636424	805	2019-04-10 19:35:56.365	2019-04-10 19:49:22.033	1522	The Embarcadero at Steuart St	
1721067	736	2018-11-02 15:33:14.876	2018-11-02 15:45:31.530	3589	11th St at Natoma St	
798420	968	2018-06-04 17:00:27.669	2018-06-04 17:16:35.906	4047	14th St at Mission St	
2129375	498	2019-02-19 13:18:44.389	2019-02-19 13:27:03.134	5442	8th St at Ringold St	
1130948	823	2018-08-13 19:21:34.021	2018-08-13 19:35:17.234	1552	Howard St at Mary St	
2285055	963	2019-03-26 17:53:38.831	2019-03-26 18:09:42.038	5338	Broadway at Kearny	
41933	256	2018-01-19 19:49:30.901	2018-01-19 19:53:46.999	561	Civic Center/UN Plaza BART Station (Market St ...	
189495	1013	2018-02-03 21:38:25.554	2018-02-03 21:55:19.126	714	Garfield Square (25th St at Harrison St)	
414033	254	2018-04-09 08:41:33.250	2018-04-09 08:45:48.011	3491	4th St at Harrison St	
602864	228	2018-05-04 07:50:20.080	2018-05-04 07:54:08.835	1590	McCoppin St at Valencia St	
817074	235	2018-06-01 10:16:50.277	2018-06-01 10:20:45.877	11	Telegraph Ave at 23rd St	
834831	880	2018-07-29 16:40:09.267	2018-07-29 16:54:50.140	781	The Embarcadero at Bryant St	
741030	15873	2018-06-13 06:57:34.887	2018-06-13 11:22:07.896	1119	Union Square (Powell St at Post St)	
527081	1423	2018-05-17 08:40:52.144	2018-05-17 09:04:35.873	1648	45th St at Manila	
2031062	1081	2019-01-07 08:11:35.935	2019-01-07 08:29:37.052	708	Valencia St at 21st St	
502653	238	2018-05-21 18:42:56.516	2018-05-21 18:46:55.047	3178	Yerba Buena Center for the Arts (Howard St at ...	
2479427	436	2019-03-03 22:10:41.675	2019-03-03 22:17:58.276	5130	San Francisco City Hall (Polk St at Grove St)	
877891	688	2018-07-23 09:42:17.639	2018-07-23 09:53:45.946	2813	Berry St at 4th St	

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
1134297	875	2018-08-13 11:31:03.640	2018-08-13 11:45:39.203	534	Koshland Park	
1016822	962	2018-07-01 15:07:03.012	2018-07-01 15:23:05.333	1039	Golden Gate Ave at Polk St	
2015366	555	2019-01-09 10:11:56.035	2019-01-09 10:21:11.178	4951	San Francisco Caltrain Station 2 (Townsend St...	
1687759	104	2018-11-08 05:59:51.854	2018-11-08 06:01:36.031	2231	Market St at 10th St	
560590	780	2018-05-11 10:17:36.043	2018-05-11 10:30:36.486	3982	Yerba Buena Center for the Arts (Howard St at ...	
1817572	762	2018-12-11 09:24:25.468	2018-12-11 09:37:08.462	1282	Potrero Ave at 15th St (Temporary Location)	
2615820	950	2019-04-12 16:07:09.179	2019-04-12 16:22:59.974	6745	Mechanics Monument Plaza (Market St at Bush St)	
2259811	815	2019-03-29 10:15:08.292	2019-03-29 10:28:43.737	4533	Church St at Duboce Ave	
2511478	1522	2019-04-28 16:27:30.987	2019-04-28 16:52:52.987	1926	Bancroft Way at College Ave	
1665204	2551	2018-11-12 18:29:19.629	2018-11-12 19:11:51.257	4373	Fell St at Stanyan St	

50 rows × 22 columns

Define

Removing the invalid values from member_age column

Code

In [49]:
```
df_clean.member_age.describe(percentiles = [ .1 , .2 , .3 , .4 , .5 , .6 , .7
, .8 , .95])
```

Out[49]:
```
count    2.734625e+06
mean    -1.190701e+08
std      4.914505e+08
min     -2.147484e+09
10%      2.300000e+01
20%      2.600000e+01
30%      2.800000e+01
40%      3.000000e+01
50%      3.200000e+01
60%      3.500000e+01
70%      3.800000e+01
80%      4.200000e+01
95%      5.600000e+01
max      1.410000e+02
Name: member_age, dtype: float64
```

In [50]:
```
# 95% of the people are under 56 and there are negative values, so we can set
 age limit 16 to 60
# and remove rest of the negative values.

df_clean = df_clean.query('member_age <=60' and 'member_age >= 16')
```

Test

In [51]:
```
df_clean.member_age.describe()
```

Out[51]:
```
count    2.583000e+06
mean     3.538395e+01
std      1.035014e+01
min      1.800000e+01
25%      2.800000e+01
50%      3.300000e+01
75%      4.000000e+01
max      1.410000e+02
Name: member_age, dtype: float64
```

In [52]:
```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2583000 entries, 0 to 2734624
Data columns (total 22 columns):
duration_sec                 int64
start_time                   datetime64[ns]
end_time                     datetime64[ns]
start_station_id             object
start_station_name           object
start_station_latitude       float64
start_station_longitude      float64
end_station_id               object
end_station_name             object
end_station_latitude         float64
end_station_longitude        float64
bike_id                      object
user_type                    category
member_birth_year            float64
member_gender                category
bike_share_for_all_trip      category
member_age                   int32
start_time_month_name        object
start_time_month             int32
start_time_day               object
start_date                   int32
start_time_hour              int64
dtypes: category(3), datetime64[ns](2), float64(5), int32(3), int64(2), objec
t(7)
memory usage: 372.0+ MB
```

Define

Adding column ride_distance for ride between stations.

Code

In [54]:
```
#Calculations are derived from the 'haversine' formula which is used to calcul
ate the great-circle distance between tow points,
#i.e. the shortest distance over the earth's surface.

def distance(origin, destination):

    lat1, long1 = origin
    lat2, long2 = destination
    radius = 6371

    dlat = math.radians(lat2 - lat1)
    dlong = math.radians(long2 - long1)

    a = (math.sin(dlat / 2) * math.sin(dlat / 2) + math.cos(math.radians(lat1
)) * math.cos(math.radians(lat2)) * math.sin(dlong / 2) * math.sin(dlong / 2))
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    d = radius * c

    return d
```

In [55]:
```
# Using the calcuated math on columns for lat and long

df_clean['ride_distance'] = df_clean.apply(lambda x: distance((x['start_statio
n_latitude'], x['start_station_longitude']),
                                                                  (x['end_station_
latitude'], x['end_station_longitude'])), axis=1)
```

Test

In [56]:
```
# data set info

df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2583000 entries, 0 to 2734624
Data columns (total 23 columns):
duration_sec                int64
start_time                  datetime64[ns]
end_time                    datetime64[ns]
start_station_id            object
start_station_name          object
start_station_latitude      float64
start_station_longitude     float64
end_station_id              object
end_station_name            object
end_station_latitude        float64
end_station_longitude       float64
bike_id                     object
user_type                   category
member_birth_year           float64
member_gender               category
bike_share_for_all_trip     category
member_age                  int32
start_time_month_name       object
start_time_month            int32
start_time_day              object
start_date                  int32
start_time_hour             int64
ride_distance               float64
dtypes: category(3), datetime64[ns](2), float64(6), int32(3), int64(2), objec
t(7)
memory usage: 391.7+ MB
```

In [57]:
```
# data sampling

df_clean.sample(50)
```

Out[57]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
873529	405	2018-07-23 19:37:06.249	2018-07-23 19:43:51.783	611	Jersey St at Castro St	
831888	695	2018-07-30 09:01:53.085	2018-07-30 09:13:28.444	1730	Spear St at Folsom St	
711452	505	2018-06-18 09:24:07.611	2018-06-18 09:32:32.668	4064	Mission Playground	
362597	970	2018-04-20 16:59:28.851	2018-04-20 17:15:38.901	3168	5th St at Howard St	
2327868	603	2019-03-21 08:59:27.859	2019-03-21 09:09:30.987	1852	14th St at Mission St	
2419987	731	2019-03-11 19:05:26.933	2019-03-11 19:17:38.831	3975	NaN	
2497861	804	2019-04-30 17:25:13.717	2019-04-30 17:38:38.536	1520	Hearst Ave at Euclid Ave	
2524164	679	2019-04-26 08:50:41.181	2019-04-26 09:02:00.580	3160	7th St at Brannan St	
2408859	322	2019-03-12 19:52:19.883	2019-03-12 19:57:42.604	5376	Parker Ave at McAllister St	
1576312	467	2018-10-03 20:35:26.131	2018-10-03 20:43:13.304	1206	22nd St Caltrain Station	
2230484	282	2019-02-02 17:06:04.165	2019-02-02 17:10:46.758	5113	16th St Mission BART Station 2	
1381321	698	2018-09-04 17:51:05.095	2018-09-04 18:02:43.390	2974	San Francisco Ferry Building (Harry Bridges Pl...	
520720	639	2018-05-18 08:46:44.115	2018-05-18 08:57:23.869	3823	Berry St at 4th St	
607886	565	2018-05-03 09:42:38.436	2018-05-03 09:52:03.791	1977	Steuart St at Market St	
1253024	782	2018-09-24 16:13:34.381	2018-09-24 16:26:36.861	1788	4th St at Mission Bay Blvd S	
2245010	132	2019-03-31 10:39:11.750	2019-03-31 10:41:23.962	6755	20th St at Bryant St	
2557449	1409	2019-04-21 15:58:33.639	2019-04-21 16:22:03.303	3147	San Francisco Ferry Building (Harry Bridges Pl...	
1522018	470	2018-10-12 07:58:59.443	2018-10-12 08:06:49.493	262	Vine St at Shattuck Ave	
1051404	330	2018-08-27 08:33:41.573	2018-08-27 08:39:11.852	3056	San Francisco Caltrain Station 2 (Townsend St...	
1433743	291	2018-10-25 19:48:47.430	2018-10-25 19:53:38.922	1300	Bancroft Way at College Ave	

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_static
2355933	742	2019-03-18 16:30:04.287	2019-03-18 16:42:26.671	5907	Montgomery St BART Station (Market St at 2nd St)	
1781943	1200	2018-12-18 08:00:46.833	2018-12-18 08:20:47.663	3241	Raymond Kimbell Playground	
2699714	263	2019-04-04 14:31:31.815	2019-04-04 14:35:55.508	5970	Duboce Park	
373769	291	2018-04-18 13:44:08.683	2018-04-18 13:49:00.234	637	Division St at Potrero Ave	
656020	340	2018-06-26 11:07:29.456	2018-06-26 11:13:09.488	3648	Mechanics Monument Plaza (Market St at Bush St)	
1844663	322	2018-12-05 15:56:22.398	2018-12-05 16:01:44.906	4451	Valencia St at 24th St	
2572195	526	2019-04-18 18:02:52.417	2019-04-18 18:11:38.712	4230	San Salvador St at 9th St	
439012	111	2018-04-02 16:09:45.701	2018-04-02 16:11:36.728	2599	Yerba Buena Center for the Arts (Howard St at ...	
1409734	596	2018-10-30 09:03:59.071	2018-10-30 09:13:55.323	3412	Webster St at MacArthur Blvd (Temporary Location)	
55224	390	2018-01-16 11:06:34.739	2018-01-16 11:13:05.180	419	Grand Ave at Santa Clara Ave	
642850	523	2018-06-28 06:40:14.144	2018-06-28 06:48:57.852	3872	Market St at Dolores St	
1471112	1249	2018-10-19 18:16:27.750	2018-10-19 18:37:17.330	2563	El Embarcadero at Grand Ave	
2436151	591	2019-03-09 11:22:38.261	2019-03-09 11:32:29.825	172	Union Square (Powell St at Post St)	
1109616	1274	2018-08-16 19:26:40.633	2018-08-16 19:47:55.254	526	Howard St at 2nd St	
2432689	7291	2019-03-09 23:38:13.004	2019-03-10 01:39:44.532	5800	Webster St at 2nd St	
89440	382	2018-01-03 08:49:46.595	2018-01-03 08:56:09.408	1624	Victoria Manalo Draves Park	
942448	934	2018-07-13 08:41:50.673	2018-07-13 08:57:25.115	34	Market St at 10th St	
1375097	205	2018-09-05 16:44:17.078	2018-09-05 16:47:42.688	297	2nd St at Townsend St	
892302	1742	2018-07-20 15:47:44.587	2018-07-20 16:16:46.636	69	S Park St at 3rd St	
2357012	1216	2019-03-18 14:16:30.509	2019-03-18 14:36:47.134	4970	Market St at Dolores St	

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_statio
1113276	591	2018-08-16 11:41:51.948	2018-08-16 11:51:42.972	1954	Howard St at 8th St	
1701346	243	2018-11-06 08:40:18.649	2018-11-06 08:44:22.272	757	8th St at Ringold St	
1830655	380	2018-12-07 21:22:04.915	2018-12-07 21:28:25.190	1396	Duboce Park	
2368635	348	2019-03-17 07:46:26.961	2019-03-17 07:52:15.349	5765	Mosswood Park	
1242296	611	2018-09-25 22:38:39.052	2018-09-25 22:48:50.424	3158	Valencia St at 16th St	
539870	337	2018-05-15 10:53:08.234	2018-05-15 10:58:45.651	592	Montgomery St BART Station (Market St at 2nd St)	
1808263	337	2018-12-12 18:23:03.457	2018-12-12 18:28:40.683	554	19th Street BART Station	
1996760	649	2019-01-12 09:56:00.205	2019-01-12 10:06:49.511	1751	20th St at Bryant St	
449863	655	2018-05-31 08:36:20.679	2018-05-31 08:47:16.088	2248	San Francisco Caltrain (Townsend St at 4th St)	
226560	529	2018-03-26 17:24:18.344	2018-03-26 17:33:08.181	820	Berry St at 4th St	

50 rows × 23 columns

In [58]:
```
df_clean.to_csv('fordgo_master_clean.csv', index = False)
```

What is the structure of your dataset?

Previously there were 2,734,625 bike rides but after cleaning the data it is 2,583,000 rides happended during Jan 2018 to April 2019. The structure of the datatset:

```
trip duration : total ride duration in seconds

start time and end time : detailed timestamp

station id : unique station id

start station and end station name : characters

latitude and longitude for start station and end station : coordinates

customer user type : customer or subscriber

customer gender

customer birth date

bike id : unique bike id
```

Additional columns were added in order to gain in-depth insight of the dataset:

```
member age

additional month, day , date , hour column were created

ride distance : in km
```

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

Interest is getting the insight out and understanding the user behavior with relationship to their attributes like:

Distribution of riders on a monthly and daily basis

Average ride duration

Average ride distance

Age groups of users

Gender distribution

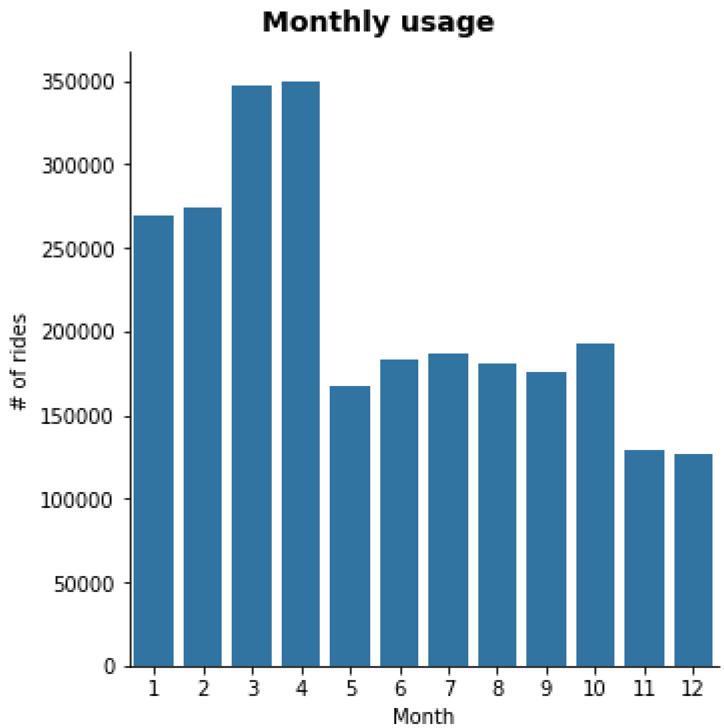
What features in the dataset do you think will help support your investigation into your feature(s) of interest?

I feel that age group ,usage type , start and end time will make a impact in the analysis. It should provide some insight on the user's behaviour.

Univariate Exploration

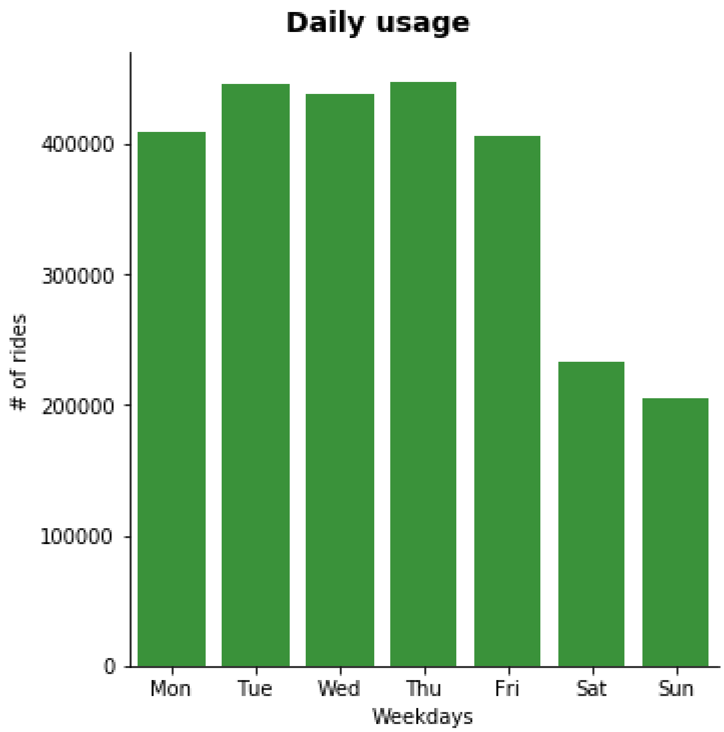
We begin with the monthly trend of number of bike rentals and distribution of weekdays and hours of the day.

In [69]:
```
# monthly usage

g = sns.catplot(data=df_clean, x='start_time_month', kind='count', color = sns
.color_palette()[0] )
g.set_axis_labels("Month", "# of rides")
g.fig.suptitle('Monthly usage', y=1.03, fontsize=14, fontweight='semibold');
```



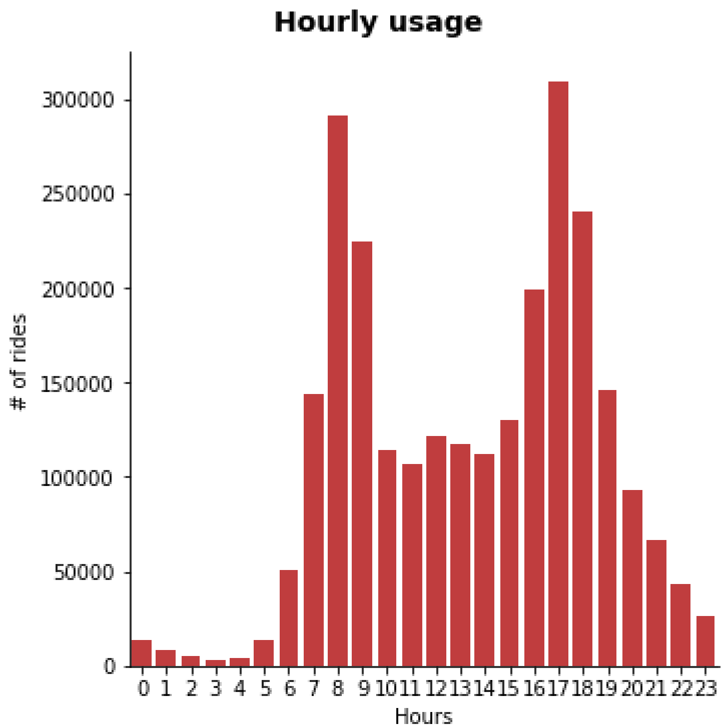
From the above analysis we can clearly noticed that usage was high during first quater and in the month of April and then there is a sudden downfall in usage. This may be due to the climatic conditions.

In [73]:
```
# daily usage

days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
g = sns.catplot(data=df_clean, x='start_time_day', kind='count', color = sns.c
olor_palette()[2], order = days)
g.set_axis_labels("Weekdays", "# of rides")
g.fig.suptitle('Daily usage', y=1.03, fontsize=14, fontweight='semibold');
```



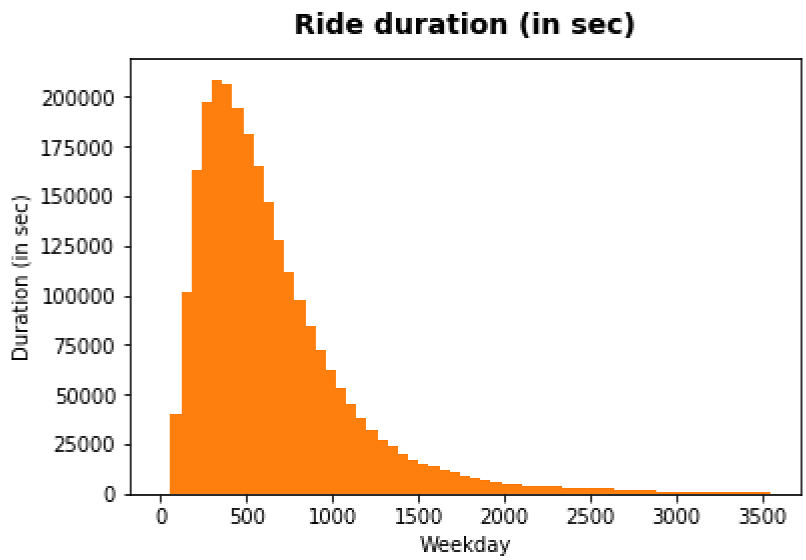
From the above analysis we can clearly see that people tend to rent a bike on the weekdays and on weekends user's prefer private means.

In [74]:
```
# hourly usage

g = sns.catplot(data=df_clean, x='start_time_hour', kind='count', color = sns.
color_palette()[3])
g.set_axis_labels("Hours", "# of rides")
g.fig.suptitle('Hourly usage', y=1.03, fontsize=14, fontweight='semibold');
```



Users' mainly rent a bike during the office hours to commute, 7-9am to 4-6pm are rush hours.

In [76]:
```
# proportion duration (sec)

bin_edges = np.arange(0, 3600,60)

plt.hist(data = df_clean, x = 'duration_sec', bins = bin_edges, color = sns.co
lor_palette()[1])

plt.title("Ride duration (in sec)", y=1.03, fontsize=14, fontweight='semibold'
)
plt.xlabel('Weekday')
plt.ylabel('Duration (in sec)');
```

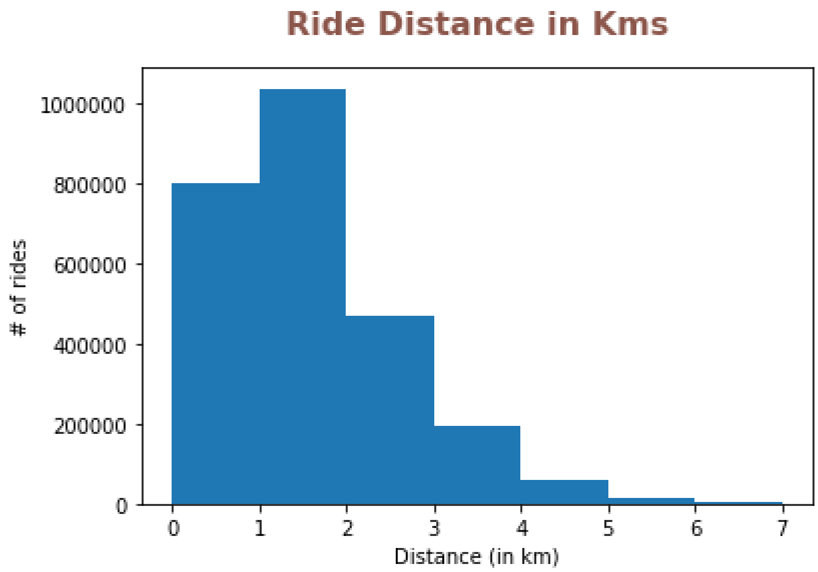


In [77]:
```
df_clean.duration_sec.describe()
```

Out[77]:
```
count    2.583000e+06
mean     7.679708e+02
std      1.910007e+03
min      6.100000e+01
25%      3.410000e+02
50%      5.390000e+02
75%      8.370000e+02
max      8.628100e+04
Name: duration_sec, dtype: float64
```

The average trip is just under 12.7 minutes, with 75% of trips being under 14 minutes. Observing the histogram, most rides are between the 3 - 11 minute range. Thus it means that rides are booked for short distances.

In [79]:
```
# Ride distance (in km)
bin_edges = np.arange(0, 8, 1)

plt.hist(data = df_clean, x = 'ride_distance', bins = bin_edges);

plt.title("Ride Distance in Kms", y=1.05, fontsize=16, fontweight='bold', colo
r = sns.color_palette()[5])
plt.xlabel('Distance (in km)')
plt.ylabel('# of rides');
```

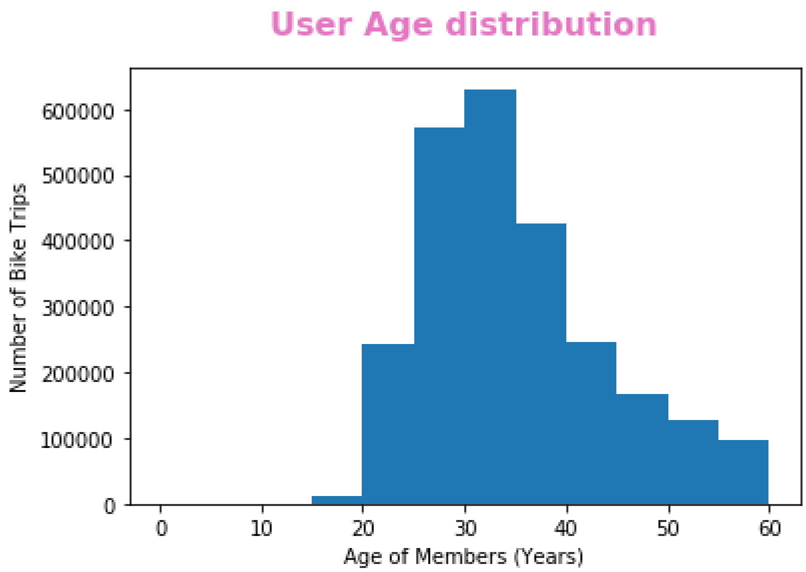


In [80]:
```
df_clean.ride_distance.describe()
```

Out[80]:
```
count    2.583000e+06
mean     1.727871e+00
std      3.471370e+01
min      0.000000e+00
25%      8.852018e-01
50%      1.400244e+00
75%      2.140739e+00
max      1.279835e+04
Name: ride_distance, dtype: float64
```

From the above observation we can see that bikes are booked for short distances with average distance of 1.7 kms and 75% of the users go around 2.2 kms.

In [82]:
```
# Age group distribution

bin_edges = np.arange(0, 65, 5)

plt.hist(data = df_clean, x = 'member_age', bins = bin_edges);

plt.title("User Age distribution", y=1.05, fontsize=16, fontweight='bold', col
or = sns.color_palette()[6])
plt.xlabel('Age of Members (Years)')
plt.ylabel('Number of Bike Trips');
```



In [83]:
```
df_clean.member_age.describe()
```

Out[83]:
```
count    2.583000e+06
mean     3.538395e+01
std      1.035014e+01
min      1.800000e+01
25%      2.800000e+01
50%      3.300000e+01
75%      4.000000e+01
max      1.410000e+02
Name: member_age, dtype: float64
```

We can see that the average user's age is 35 and generally 75% of the users are under 40 years of age.

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

Unusal points came for the duration, where sometimes the value was more than 24 hours. So i had to set the histogram accordingly, max range to 3600 sec = 60 min.

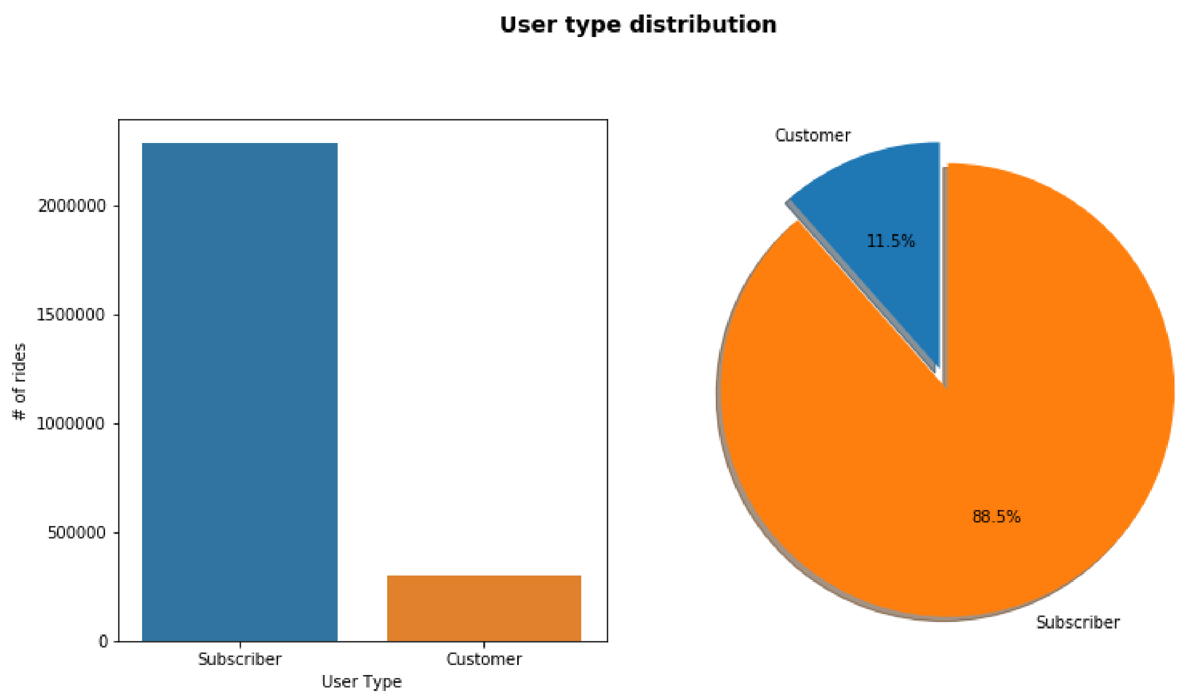
Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

Unusal distribution occured for the member birth year, in which some values were dated before 1900. Since 95% of the members were between 18 and 56 years, I removed users older than 60.

Bivariate Exploration

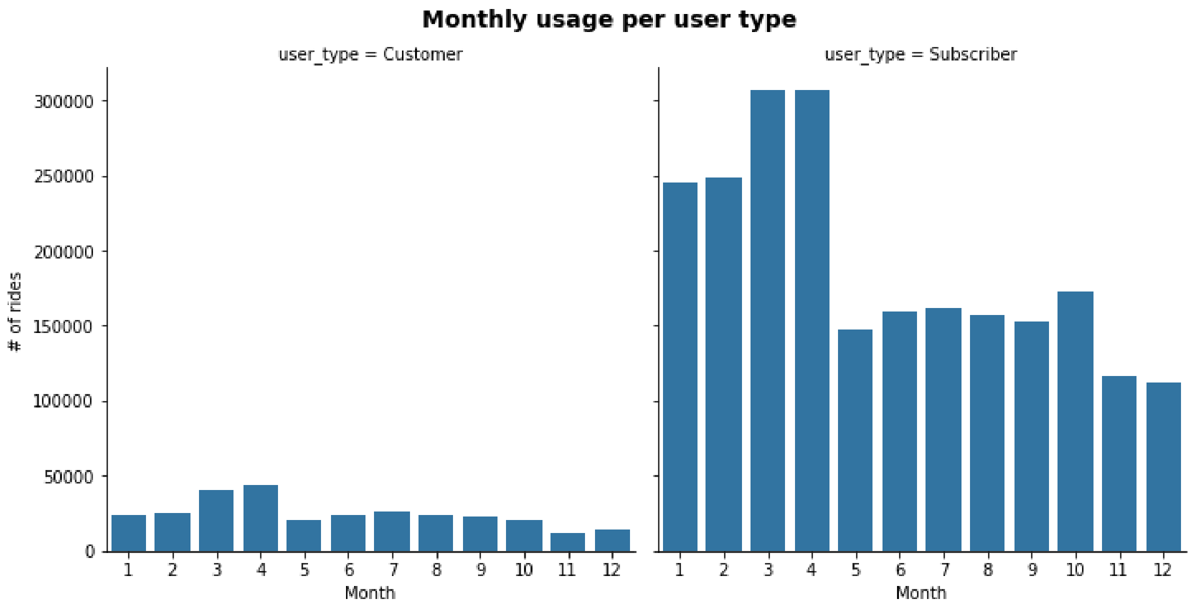
In [84]:
```
# distribution of user types

customer = df_clean.query('user_type == "Customer"')['bike_id'].count()
subscriber = df_clean.query('user_type == "Subscriber"')['bike_id'].count()

customer_distribution = customer / df_clean['bike_id'].count()
subscriber_distribution = subscriber / df_clean['bike_id'].count()
```

In [88]:
```
plt.figure(figsize = [12, 6])

# bar chart
plt.subplot(1, 2, 1)

g = sns.countplot(data=df_clean, x="user_type", order=df_clean.user_type.value
_counts().index)
g.set_xlabel('User Type')
g.set_ylabel('# of rides')

# pie chart
plt.subplot(1, 2, 2)

labels = ['Customer', 'Subscriber']
sizes = [customer_distribution, subscriber_distribution]
explode = (0, 0.1)

plt.pie(sizes, explode=explode, labels=labels,autopct='%1.1f%%', shadow=True,
startangle=90)
plt.axis('equal')

plt.suptitle('User type distribution', y=1.03, fontsize=14, fontweight='semibo
ld');
```



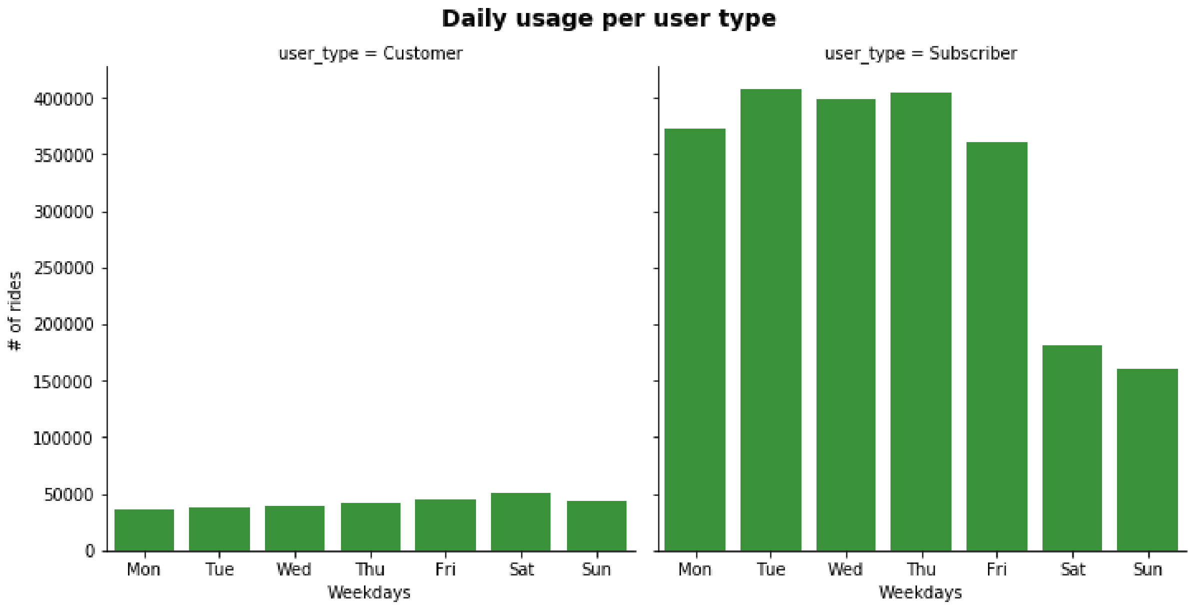
The bike sharing system is mainly used by subscribers with 88% proportion and than ocassional, customer with 12%.

In [90]:
```
# monthly usage per user type

g = sns.catplot(data=df_clean, x='start_time_month',col = 'user_type', kind='c
ount', color = sns.color_palette()[0] )
g.set_axis_labels("Month", "# of rides")
g.fig.suptitle('Monthly usage per user type', y=1.03, fontsize=14, fontweight=
'semibold');
```



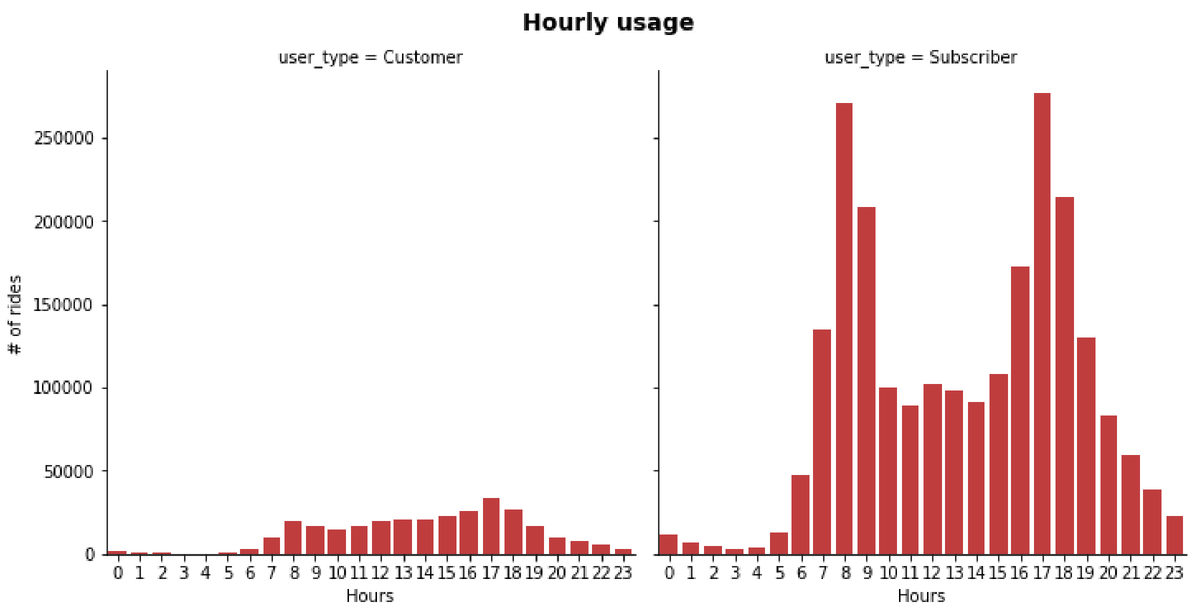
The trend is similar for both customer and subscriber first quater and april has high usage.

In [91]:
```
# daily usage per user type

days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
g = sns.catplot(data=df_clean, x='start_time_day',col = 'user_type', kind='count', color = sns.color_palette()[2], order = days)
g.set_axis_labels("Weekdays", "# of rides")
g.fig.suptitle('Daily usage per user type', y=1.03, fontsize=14, fontweight='semibold');
```



For subscriber we can see the trend with weekdays whereas for customers its almost same for each day.

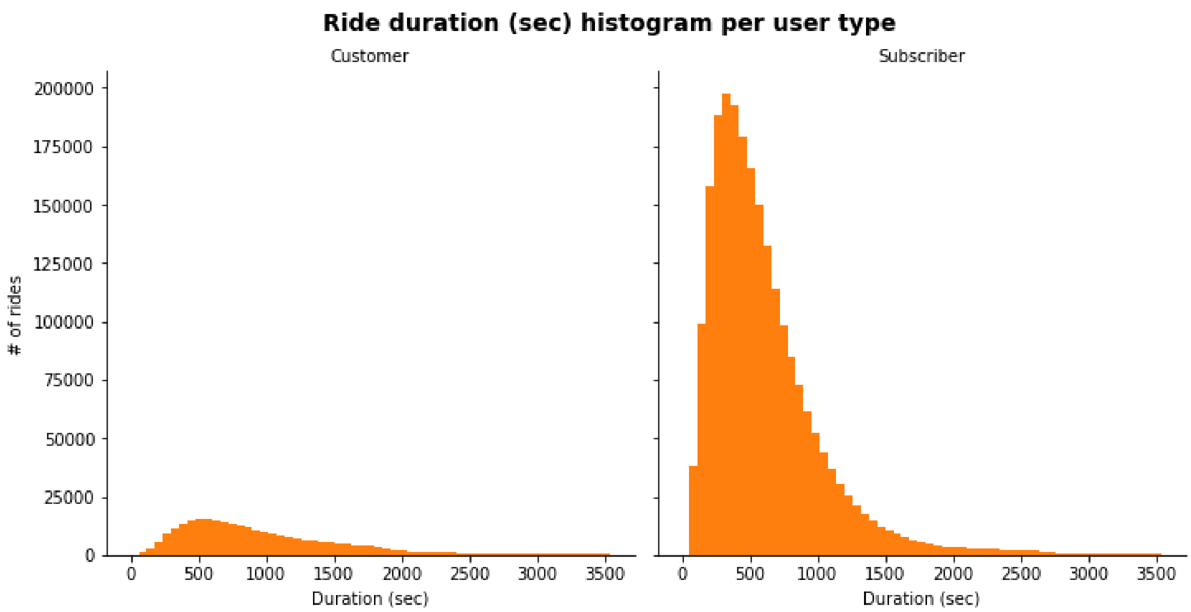
In [99]:
```
# hourly usage per user type

g = sns.catplot(data=df_clean, x='start_time_hour',col = 'user_type', kind='co
unt', color = sns.color_palette()[3])
g.set_axis_labels("Hours", "# of rides")
g.fig.suptitle('Hourly usage', y=1.03, fontsize=14, fontweight='semibold');
```



Both customer and subscriber has high usage during office hours.

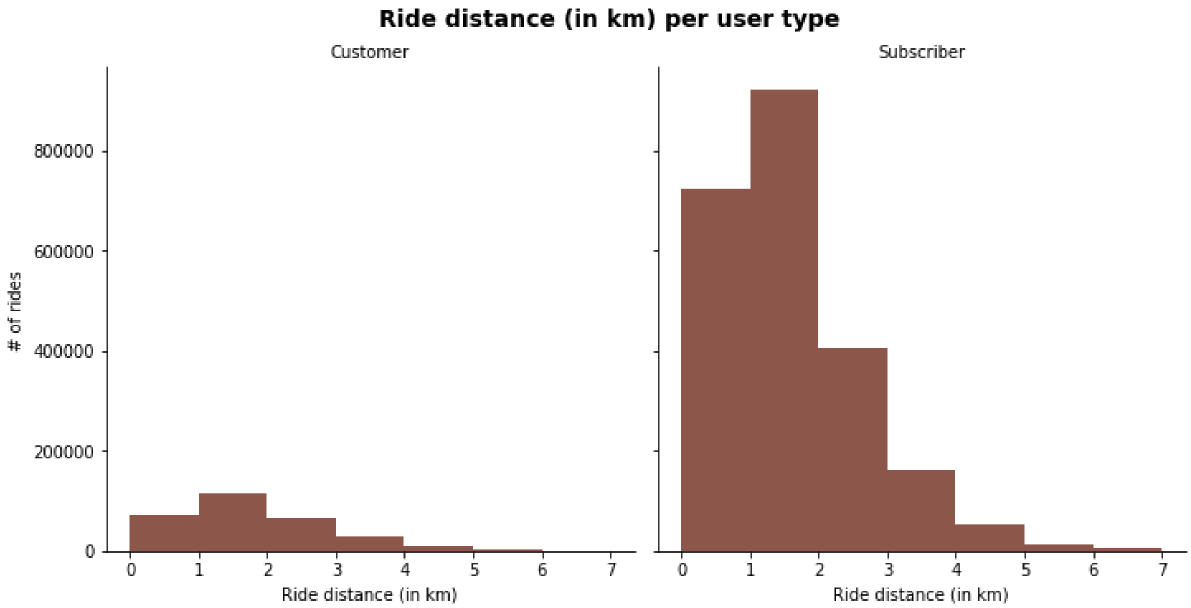
In [100]:
```
#duration (sec) distribution per user type

g = sns.FacetGrid(df_clean, col="user_type", margin_titles=True, size=5)
bin_edges = np.arange(0, 3600,60)
g.map(plt.hist, "duration_sec", color=sns.color_palette()[1], bins=bin_edges)
g.set_axis_labels("Duration (sec)", "# of rides")
g.set_titles(col_template = '{col_name}')
g.fig.suptitle('Ride duration (sec) histogram per user type', y=1.03, fontsize
=14, fontweight='semibold');
```

```
c:\users\nilad\appdata\local\programs\python\python37\lib\site-packages\seabo
rn\axisgrid.py:230: UserWarning: The `size` paramter has been renamed to `hei
ght`; please update your code.
  warnings.warn(msg, UserWarning)
```



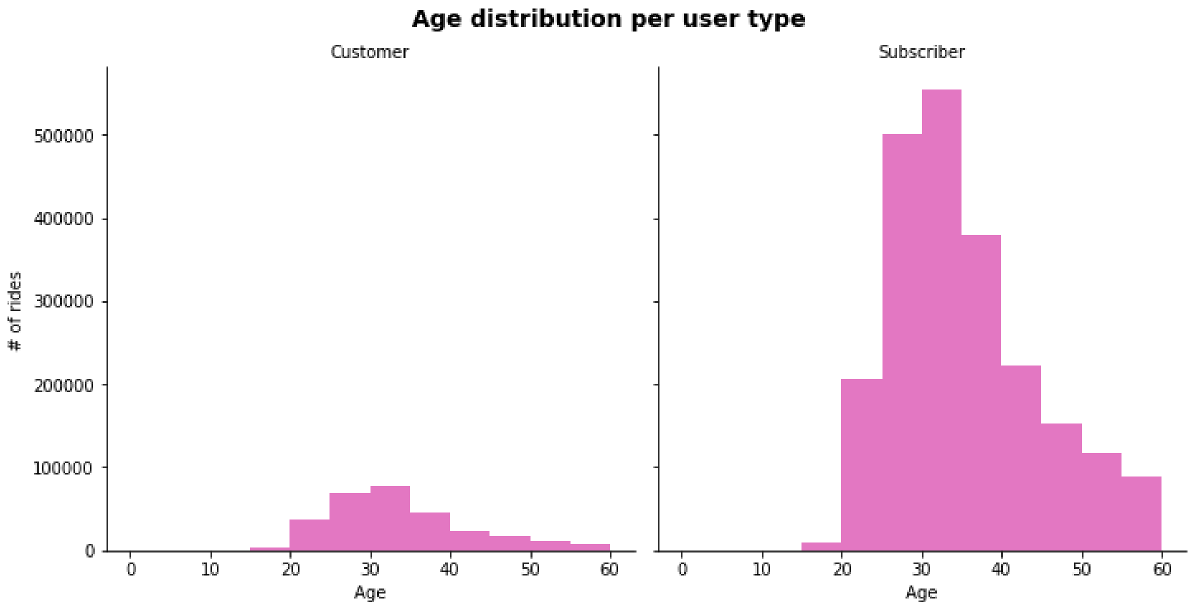
We can observe that trip durations are longer for customers around 8 to 23 minutes than for subscribers 7 to 12 minutes.

In [101]:
```
#ride distance (in km) distribution per user type

g = sns.FacetGrid(df_clean, col="user_type", margin_titles=True, size=5)
bin_edges = np.arange(0, 8, 1)
g.map(plt.hist, "ride_distance", color=sns.color_palette()[5], bins=bin_edges)
g.set_axis_labels("Ride distance (in km)", "# of rides")
g.set_titles(col_template = '{col_name}')
g.fig.suptitle('Ride distance (in km) per user type', y=1.03, fontsize=14, fon
tweight='semibold');
```



Both customer and subscriber travel for short distances, the number of rides of subscribers are much greater than customers.

In [102]:
```
#age group distribution per user type

g = sns.FacetGrid(df_clean, col="user_type", margin_titles=True, size=5)
bin_edges = np.arange(0, 65, 5)
g.map(plt.hist, "member_age", color=sns.color_palette()[6], bins=bin_edges)
g.set_axis_labels(" Age ", "# of rides")
g.set_titles(col_template = '{col_name}')
g.fig.suptitle('Age distribution per user type', y=1.03, fontsize=14, fontweig
ht='semibold');
```



The age distribution is same for both customer and subscriber with 18 to 40 years group users rent more.

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

Grouping the data in user type provided much more insigh of the data. People who rent bike are generally casual riders like tourists, or students residing nearby and is mainly rented during first quater and april month. Customers tend to increase during weekends. Bikes are mainly rented during 7-9 am and 5-7pm to commute to office or educational institute.

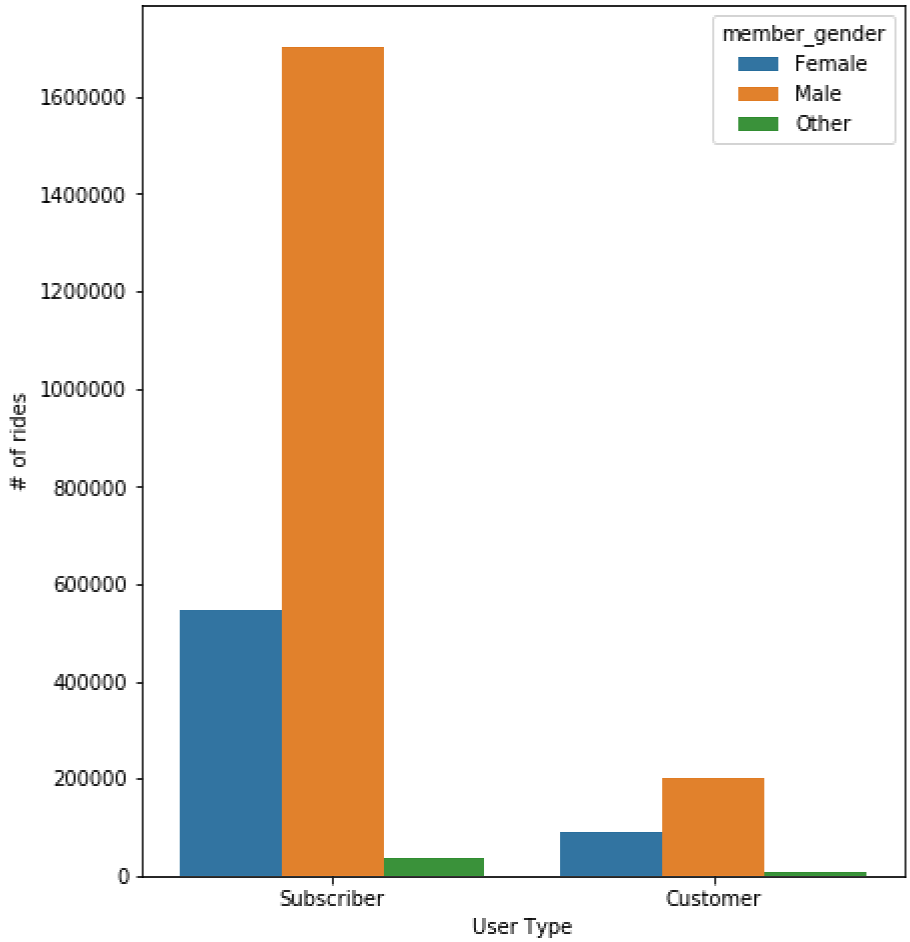
Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

Subscribers most frequently rent, around 7-9am and 4-6pm. Customers rent at weekend around 10am-5pm and weekday 5-6pm. Customers rent during weekend for casual purpose.

Multivariate Exploration

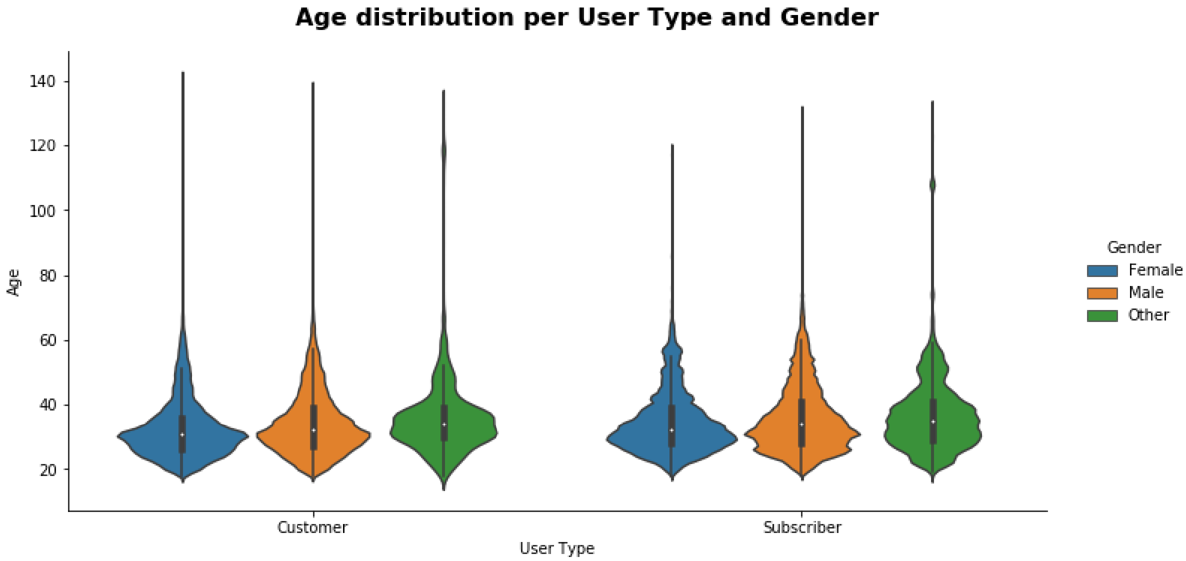
In [125]:

```
# no of bike trips vs user type with category filters as gender

plt.figure(figsize = [15, 8])

plt.subplot(1, 2, 1)

g = sns.countplot(data=df_clean, x="user_type", hue="member_gender", order=df_
clean.user_type.value_counts().index)
g.set_xlabel('User Type')
g.set_ylabel('# of rides');
```



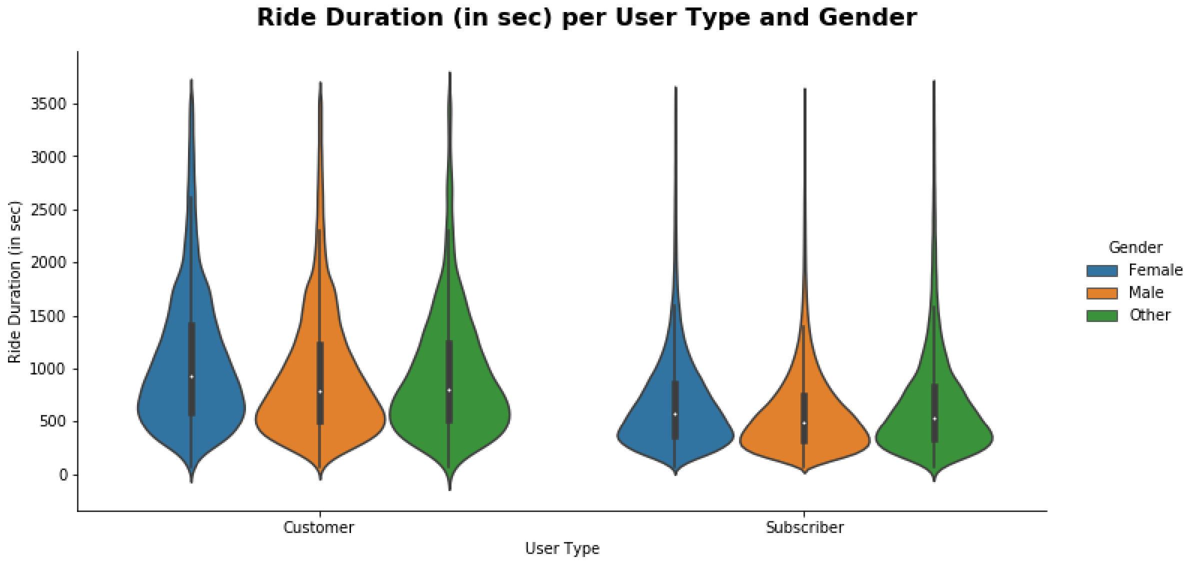
Male in subscriber user type tend to have more rides than the male in customer, female and other have few rides. We can predict that customers are mainly casual visiters.

In [121]:
```
#age distribution per user type and gender

graph = sns.catplot(data=df_clean, x='user_type', y="member_age", hue="member_
gender", kind="violin", height=5, aspect=2);

graph.set_axis_labels("User Type", "Age")
graph._legend.set_title('Gender')
graph.fig.suptitle('Age distribution per User Type and Gender', y=1.05, fontsi
ze=16, fontweight='bold');
```



Its good to see that all genders have equal age distribution also for user types. But subscribers also have slighly aged persons 40 to 50 years age which is very encouraging.

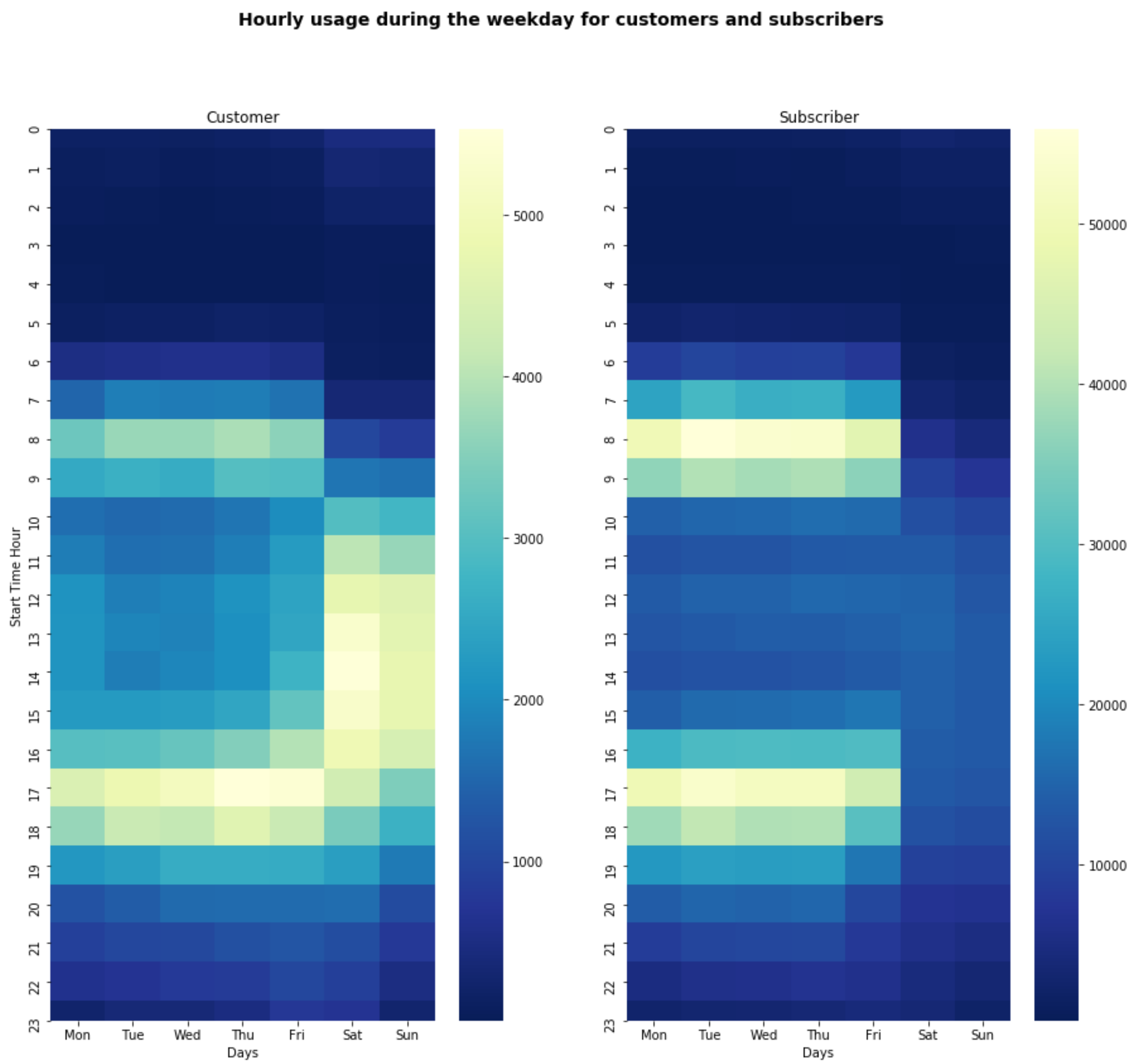
In [120]:
```
#ride duration per user type and gender

graph = sns.catplot(data=df_clean.query('duration_sec < 3600'), x='user_type',
y="duration_sec", hue="member_gender", kind="violin", height=5, aspect=2);

graph.set_axis_labels("User Type", "Ride Duration (in sec)")
graph._legend.set_title('Gender')
graph.fig.suptitle('Ride Duration (in sec) per User Type and Gender', y=1.05,
fontsize=16, fontweight='bold');
```



Subscriber tend to have less ride hour as they mainly commute to office or educational institute, so they have a fixed distance. While customers have rather more ride durations as compared to customers beacuse they are mainly tourists or casual travellers.

In [126]:

```
# weekday order

df_clean['start_time_day'] = pd.Categorical(df_clean['start_time_day'],categor
ies=['Mon','Tue','Wed','Thu','Fri','Sat', 'Sun'],
                                              ordered=True)
plt.figure(figsize=(15,13))
plt.suptitle('Hourly usage during the weekday for customers and subscribers',
fontsize=14, fontweight='semibold')

# heatmap for customers
plt.subplot(1, 2, 1)
df_customer = df_clean.query('user_type == "Customer"').groupby(["start_time_h
our", "start_time_day"])["bike_id"].size().reset_index()
df_customer = df_customer.pivot("start_time_hour", "start_time_day", "bike_id"
)
sns.heatmap(df_customer, cmap='YlGnBu_r')

plt.title("Customer", y=1.015)
plt.xlabel('Days')
plt.ylabel('Start Time Hour')

# heatmap for subscribers
plt.subplot(1, 2, 2)
df_subscriber = df_clean.query('user_type == "Subscriber"').groupby(["start_ti
me_hour", "start_time_day"])["bike_id"].size().reset_index()
df_subscriber = df_subscriber.pivot("start_time_hour", "start_time_day", "bike
_id")
sns.heatmap(df_subscriber, cmap='YlGnBu_r')

plt.title("Subscriber", y=1.015)
plt.xlabel('Days')
plt.ylabel('');
```



Customers rent more often on weekends, while Subscribers primarily use the bikes on weekdays.

Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

Customers: During weekdays, most bike rides occur between 4-6pm, peaking on Fridays around 5pm. During weekends, most bike rides occur between 11am and 6pm, peaking on Saturdays around 2pm.

Subscribers: During weekdays, most bike rides occur around 8-9am and 4-6pm.

Were there any interesting or surprising interactions between features?

It was interesting and also surprising to see 40-50 years old group active.

Sources

1. FordGoBike Data Set
2. Haversine formula used to calculate distances using latitude and longitude
3. Stackoverflow
4. Google