Ford GoBike

April 22, 2019

1 Analyzing Ford GoBike

Ahmad Abu Saida

2 Introduction

Ford GoBike is a regional public bicycle sharing system in the San Francisco Bay Area, California. Beginning operation in August 2013 as Bay Area Bike Share, the Ford GoBike 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 launched as Ford GoBike in a partnership with Ford Motor Company.

Ford GoBike, like other bike share systems, consists of a fleet of specially designed, sturdy and durable bikes that are locked into a network of docking stations throughout the city. The bikes can be unlocked from one station and returned to any other station in the system, making them ideal for one-way trips. The bikes are available for use 24 hours/day, 7 days/week, 365 days/year and riders have access to all bikes in the network when they become a member or purchase a pass.

3 Preliminary Wrangling

This document explores the Ford GoBike's trip data for public containing approximately 1,850,000 bike rides from FY2018.

Part I - Gathering Data

```
In [320]: # import all packages and set plots to be embedded inline
          from requests import get
          from os import path, getcwd, makedirs, listdir
          from io import BytesIO
          from zipfile import ZipFile
          import pandas as pd
          import numpy as np
          import matplotlib
          from matplotlib import pyplot as plt
          import matplotlib.ticker as tick
          import seaborn as sns
          import datetime
          import math
          import calendar
```

```python
import warnings
warnings.filterwarnings('ignore')
from IPython.display import Image
%matplotlib inline
```

In [321]:
```python
# download the dataset with pandas
folder_name_of_csvs = 'trip_data_files'
```

In [322]:
```python
# Combine All Locally Saved CSVs into One DataFrame
list_csvs = []
for file_name in listdir(folder_name_of_csvs):
    list_csvs.append(pd.read_csv(folder_name_of_csvs+'/'+file_name))
df = pd.concat(list_csvs)
```

In [323]:
```python
df.to_csv('data.csv')
```

Part II - Assessing Data

In [324]:
```python
# Visually check first 5 records
df.head()
```

Out[324]:
```
   Unnamed: 0  bike_id bike_share_for_all_trip  duration_sec  end_station_id  \
0         NaN     1035                      No           598           114.0
1         NaN     1673                      No           943           324.0
2         NaN     3498                      No         18587            15.0
3         NaN     3129                      No         18558            15.0
4         NaN     1839                     Yes           885           297.0

   end_station_latitude  end_station_longitude  \
0             37.764478            -122.402570
1             37.788300            -122.408531
2             37.795392            -122.394203
3             37.795392            -122.394203
4             37.322980            -121.887931

                                    end_station_name  \
0                          Rhode Island St at 17th St
1                 Union Square (Powell St at Post St)
2  San Francisco Ferry Building (Harry Bridges Pl...
3  San Francisco Ferry Building (Harry Bridges Pl...
4                             Locust St at Grant St

                  end_time  member_birth_year member_gender  \
0  2018-03-01 00:09:45.1870             1988.0          Male
1  2018-02-28 23:36:59.9740             1987.0          Male
2  2018-02-28 23:30:42.9250             1986.0        Female
3  2018-02-28 23:30:12.4500             1981.0          Male
4  2018-02-28 23:29:58.6080             1976.0        Female
```

```
   start_station_id  start_station_latitude  start_station_longitude  \
0             284.0               37.784872              -122.400876   
1               6.0               37.804770              -122.403234   
2              93.0               37.770407              -122.391198   
3              93.0               37.770407              -122.391198   
4             308.0               37.336802              -121.894090   

                                  start_station_name  \
0  Yerba Buena Center for the Arts (Howard St at ...   
1                     The Embarcadero at Sansome St   
2                      4th St at Mission Bay Blvd S   
3                      4th St at Mission Bay Blvd S   
4                                   San Pedro Square   

                start_time   user_type  
0  2018-02-28 23:59:47.0970  Subscriber  
1  2018-02-28 23:21:16.4950    Customer  
2  2018-02-28 18:20:55.1900    Customer  
3  2018-02-28 18:20:53.6210    Customer  
4  2018-02-28 23:15:12.8580  Subscriber  
```

In [325]:
```python
# Visually check 50 random records
df.sample(50)
```

Out[325]:
```
        Unnamed: 0  bike_id bike_share_for_all_trip  duration_sec  \
11921          NaN      907                      No          3421   
138542         NaN     2941                      No           320   
102301         NaN     3756                      No          9814   
89151          NaN     3415                      No           740   
51194      51194.0     1129                     NaN           435   
83547          NaN     4041                      No           311   
17651          NaN     3034                      No           126   
129185         NaN     1121                      No          7710   
5811           NaN     4387                      No           975   
71881          NaN      176                      No           630   
51902          NaN     3210                      No          1398   
125783         NaN      746                      No          7808   
401956    401956.0     2150                     NaN           304   
28099          NaN      217                      No           735   
11388          NaN     3478                      No           648   
357439    357439.0     1776                     NaN          1360   
24095          NaN      212                     Yes           362   
60479          NaN     3264                      No           391   
104690         NaN     3957                     Yes            78   
75566          NaN      781                     Yes           776   
496502    496502.0     1967                     NaN           572   
156145         NaN     3554                      No           633   
148139         NaN     3931                      No           254   
```

117536	NaN	304	No	320
293811	293811.0	67	NaN	166
77378	NaN	1779	No	1043
118074	NaN	3304	No	597
76456	NaN	269	Yes	744
121566	121566.0	2631	NaN	925
88858	NaN	1220	No	1809
32078	NaN	4012	No	436
105554	NaN	2370	No	447
22853	NaN	2967	No	540
285185	285185.0	2534	NaN	369
44800	NaN	3235	No	1133
27204	NaN	3327	Yes	541
373224	373224.0	2019	NaN	921
138994	NaN	2419	No	750
292	NaN	626	Yes	201
15418	NaN	3491	No	136
170469	NaN	4276	No	520
5202	NaN	2324	No	408
52710	NaN	411	No	294
109255	NaN	2881	No	470
3177	NaN	411	No	417
168856	NaN	4456	No	1120
58266	NaN	3033	No	1127
106144	NaN	1333	No	806
104366	NaN	343	No	558
150497	NaN	3874	No	393

	end_station_id	end_station_latitude	end_station_longitude	\
11921	317.0	37.333955	-121.877349	
138542	89.0	37.769218	-122.407646	
102301	148.0	37.829705	-122.287610	
89151	200.0	37.800214	-122.253810	
51194	180.0	37.812678	-122.268773	
83547	80.0	37.775306	-122.397380	
17651	90.0	37.771058	-122.402717	
129185	163.0	37.797320	-122.265320	
5811	26.0	37.787290	-122.394380	
71881	58.0	37.776619	-122.417385	
51902	36.0	37.783830	-122.398870	
125783	70.0	37.773311	-122.444293	
401956	14.0	37.795001	-122.399970	
28099	67.0	37.776639	-122.395526	
11388	240.0	37.866043	-122.258804	
357439	125.0	37.759200	-122.409851	
24095	99.0	37.767037	-122.415443	
60479	222.0	37.792714	-122.248780	
104690	89.0	37.769218	-122.407646	

75566	279.0	37.339146	-121.884105
496502	19.0	37.788975	-122.403452
156145	67.0	37.776639	-122.395526
148139	13.0	37.794231	-122.402923
117536	231.0	37.808750	-122.283282
293811	16.0	37.794130	-122.394430
77378	19.0	37.788975	-122.403452
118074	203.0	37.795195	-122.273970
76456	217.0	37.817015	-122.271761
121566	74.0	37.776435	-122.426244
88858	74.0	37.776435	-122.426244
32078	110.0	37.763708	-122.415204
105554	8.0	37.799953	-122.398525
22853	67.0	37.776639	-122.395526
285185	196.0	37.808894	-122.256460
44800	80.0	37.775306	-122.397380
27204	222.0	37.792714	-122.248780
373224	218.0	37.812331	-122.285171
138994	239.0	37.868813	-122.258764
292	349.0	37.781010	-122.405666
15418	145.0	37.743684	-122.426806
170469	NaN	37.410000	-121.930000
5202	21.0	37.789625	-122.400811
52710	276.0	37.332233	-121.912517
109255	61.0	37.776513	-122.411306
3177	15.0	37.795392	-122.394203
168856	39.0	37.778999	-122.436861
58266	73.0	37.771793	-122.433708
106144	60.0	37.774520	-122.409449
104366	67.0	37.776639	-122.395526
150497	21.0	37.789625	-122.400811

	end_station_name \
11921	San Salvador St at 9th St
138542	Division St at Potrero Ave
102301	Horton St at 40th St
89151	2nd Ave at E 18th St
51194	Telegraph Ave at 23rd St
83547	Townsend St at 5th St
17651	Townsend St at 7th St
129185	Lake Merritt BART Station
5811	1st St at Folsom St
71881	Market St at 10th St
51902	Folsom St at 3rd St
125783	Central Ave at Fell St
401956	Clay St at Battery St
28099	San Francisco Caltrain Station 2 (Townsend St...
11388	Haste St at Telegraph Ave

```
357439                                  20th St at Bryant St
24095                                   Folsom St at 15th St
60479                                  10th Ave at E 15th St
104690                             Division St at Potrero Ave
75566                                Santa Clara St at 7th St
496502                                  Post St at Kearny St
156145  San Francisco Caltrain Station 2  (Townsend St...
148139                         Commercial St at Montgomery St
117536                                 14th St at Filbert St
293811                                Steuart St at Market St
77378                                   Post St at Kearny St
118074                                  Webster St at 2nd St
76456                                  27th St at MLK Jr Way
121566                                 Laguna St at Hayes St
88858                                  Laguna St at Hayes St
32078    17th & Folsom Street Park (17th St at Folsom St)
105554                          The Embarcadero at Vallejo St
22853   San Francisco Caltrain Station 2  (Townsend St...
285185                                Grand Ave at Perkins St
44800                                  Townsend St at 5th St
27204                                  10th Ave at E 15th St
373224                                         DeFremery Park
138994                          Bancroft Way at Telegraph Ave
292                                     Howard St at Mary St
15418                                   29th St at Church St
170469                                                    NaN
5202     Montgomery St BART Station (Market St at 2nd St)
52710                                Julian St at The Alameda
109255                                   Howard St at 8th St
3177    San Francisco Ferry Building (Harry Bridges Pl...
168856                            Scott St at Golden Gate Ave
58266                                 Pierce St at Haight St
106144                                  8th St at Ringold St
104366  San Francisco Caltrain Station 2  (Townsend St...
150497   Montgomery St BART Station (Market St at 2nd St)

                        end_time  member_birth_year member_gender  \
11921   2018-06-28 22:17:07.5270             2000.0        Female
138542  2018-10-10 09:45:16.8280             1985.0          Male
102301  2018-04-08 19:37:33.9080                NaN           NaN
89151   2018-02-05 17:36:47.7490             1975.0        Female
51194   2017-12-11 13:43:56.6250             1988.0          Male
83547   2018-07-19 09:05:05.5880             1983.0        Female
17651   2018-06-28 09:39:12.8650             1976.0          Male
129185  2018-04-01 20:18:16.1770                NaN           NaN
5811    2018-09-29 16:28:25.3080             1967.0          Male
71881   2018-06-20 14:04:37.6760             1994.0        Female
51902   2018-04-20 09:18:33.4150             1980.0        Female
```

125783	2018-07-12 19:34:02.9810	1988.0	Male
401956	2017-08-28 13:30:19.9500	1984.0	Female
28099	2018-02-21 07:57:30.9540	1988.0	Male
11388	2018-04-28 14:40:37.1680	1969.0	Female
357439	2017-09-11 21:51:23.7340	1992.0	Male
24095	2018-06-27 13:40:23.9250	1963.0	Male
60479	2018-03-15 21:28:38.6040	1987.0	Male
104690	2018-05-14 04:37:39.5190	1953.0	Male
75566	2018-11-10 13:01:36.8220	1995.0	Male
496502	2017-07-19 09:04:07.7420	1988.0	Female
156145	2018-09-06 17:47:45.2710	1993.0	Male
148139	2018-06-08 09:29:03.3200	1993.0	Male
117536	2018-07-13 21:34:46.5920	1989.0	Male
293811	2017-09-29 19:33:48.9250	1958.0	Male
77378	2018-01-09 08:35:25.4590	1978.0	Male
118074	2018-09-12 17:17:27.7910	1975.0	Male
76456	2018-10-19 15:30:25.6800	1966.0	Female
121566	2017-11-18 16:41:10.8690	1965.0	Female
88858	2018-09-17 14:03:22.4720	1996.0	Male
32078	2018-05-25 21:12:39.7090	1984.0	Male
105554	2018-07-16 12:28:05.2120	1984.0	Male
22853	2018-01-25 18:25:29.0640	1984.0	Male
285185	2017-10-02 19:31:30.8480	1978.0	Male
44800	2018-07-25 08:04:07.1790	NaN	NaN
27204	2018-11-25 10:53:56.2530	1974.0	Male
373224	2017-09-06 20:52:23.2480	1990.0	Male
138994	2018-10-10 09:13:34.4140	1984.0	Male
292	2018-08-31 21:19:14.4110	1985.0	Female
15418	2018-07-29 17:46:08.4180	1984.0	Male
170469	2018-07-06 09:28:35.6470	1998.0	Male
5202	2018-06-29 19:32:14.2610	1991.0	Male
52710	2018-03-18 21:15:57.1580	1972.0	Male
109255	2018-09-13 18:04:02.0210	1994.0	Male
3177	2018-10-31 16:13:23.2210	1982.0	Male
168856	2018-10-05 10:18:51.7600	1972.0	Male
58266	2018-04-18 22:40:44.9820	1987.0	Male
106144	2018-09-14 08:21:23.9150	1982.0	Female
104366	2018-11-06 06:54:27.0120	1996.0	Female
150497	2018-09-07 14:32:49.7670	1984.0	Female

	start_station_id	start_station_latitude	start_station_longitude	\
11921	317.0	37.333955	-121.877349	
138542	80.0	37.775306	-122.397380	
102301	148.0	37.829705	-122.287610	
89151	195.0	37.812314	-122.260779	
51194	197.0	37.808848	-122.249680	
83547	116.0	37.764802	-122.394771	
17651	80.0	37.775306	-122.397380	

129185	163.0	37.797320	-122.265320
5811	15.0	37.795392	-122.394203
71881	122.0	37.760299	-122.418892
51902	71.0	37.773063	-122.439078
125783	70.0	37.773311	-122.444293
401956	21.0	37.789625	-122.400811
28099	122.0	37.760299	-122.418892
11388	245.0	37.870348	-122.267764
357439	66.0	37.778742	-122.392741
24095	58.0	37.776619	-122.417385
60479	163.0	37.797320	-122.265320
104690	89.0	37.769218	-122.407646
75566	299.0	37.323678	-121.874119
496502	30.0	37.776598	-122.395282
156145	61.0	37.776513	-122.411306
148139	19.0	37.788975	-122.403452
117536	7.0	37.804562	-122.271738
293811	8.0	37.799953	-122.398525
77378	134.0	37.752428	-122.420628
118074	198.0	37.807813	-122.264496
76456	7.0	37.804562	-122.271738
121566	132.0	37.751819	-122.426614
88858	13.0	37.794231	-122.402923
32078	119.0	37.761047	-122.432642
105554	24.0	37.789677	-122.390428
22853	113.0	37.764555	-122.410345
285185	182.0	37.809013	-122.268247
44800	15.0	37.795392	-122.394203
27204	201.0	37.797673	-122.262997
373224	149.0	37.831275	-122.285633
138994	18.0	37.850222	-122.260172
292	5.0	37.783899	-122.408445
15418	147.0	37.744067	-122.421472
170469	NaN	37.400000	-121.920000
5202	343.0	37.783172	-122.393572
52710	307.0	37.332692	-121.900084
109255	30.0	37.776598	-122.395282
3177	36.0	37.783830	-122.398870
168856	37.0	37.785000	-122.395936
58266	34.0	37.783988	-122.412408
106144	52.0	37.777416	-122.441838
104366	97.0	37.768265	-122.420110
150497	50.0	37.780526	-122.390288

	start_station_name \
11921	San Salvador St at 9th St
138542	Townsend St at 5th St
102301	Horton St at 40th St

89151	Bay Pl at Vernon St
51194	El Embarcadero at Grand Ave
83547	Mississippi St at 17th St
17651	Townsend St at 5th St
129185	Lake Merritt BART Station
5811	San Francisco Ferry Building (Harry Bridges Pl...
71881	19th St at Mission St
51902	Broderick St at Oak St
125783	Central Ave at Fell St
401956	Montgomery St BART Station (Market St at 2nd St)
28099	19th St at Mission St
11388	Downtown Berkeley BART
357439	3rd St at Townsend St
24095	Market St at 10th St
60479	Lake Merritt BART Station
104690	Division St at Potrero Ave
75566	Bestor Art Park
496502	San Francisco Caltrain (Townsend St at 4th St)
156145	Howard St at 8th St
148139	Post St at Kearny St
117536	Frank H Ogawa Plaza
293811	The Embarcadero at Vallejo St
77378	Valencia St at 24th St
118074	Snow Park
76456	Frank H Ogawa Plaza
121566	24th St at Chattanooga St
88858	Commercial St at Montgomery St
32078	18th St at Noe St
105554	Spear St at Folsom St
22853	Franklin Square
285185	19th Street BART Station
44800	San Francisco Ferry Building (Harry Bridges Pl...
27204	10th St at Fallon St
373224	Emeryville Town Hall
138994	Telegraph Ave at Alcatraz Ave
292	Powell St BART Station (Market St at 5th St)
15418	29th St at Tiffany Ave
170469	NaN
5202	Bryant St at 2nd St
52710	SAP Center
109255	San Francisco Caltrain (Townsend St at 4th St)
3177	Folsom St at 3rd St
168856	2nd St at Folsom St
58266	Father Alfred E Boeddeker Park
106144	McAllister St at Baker St
104366	14th St at Mission St
150497	2nd St at Townsend St

	start_time	user_type
11921	2018-06-28 21:20:06.5010	Customer
138542	2018-10-10 09:39:56.3450	Subscriber
102301	2018-04-08 16:53:59.0630	Customer
89151	2018-02-05 17:24:27.6090	Subscriber
51194	2017-12-11 13:36:41.0530	Subscriber
83547	2018-07-19 08:59:54.0430	Subscriber
17651	2018-06-28 09:37:06.3020	Subscriber
129185	2018-04-01 18:09:45.2070	Customer
5811	2018-09-29 16:12:09.4130	Subscriber
71881	2018-06-20 13:54:07.1020	Customer
51902	2018-04-20 08:55:14.8930	Subscriber
125783	2018-07-12 17:23:54.2580	Subscriber
401956	2017-08-28 13:25:15.0760	Subscriber
28099	2018-02-21 07:45:15.3760	Subscriber
11388	2018-04-28 14:29:48.2010	Subscriber
357439	2017-09-11 21:28:43.0040	Customer
24095	2018-06-27 13:34:21.0780	Subscriber
60479	2018-03-15 21:22:07.0690	Subscriber
104690	2018-05-14 04:36:21.3220	Subscriber
75566	2018-11-10 12:48:39.9510	Subscriber
496502	2017-07-19 08:54:34.9910	Subscriber
156145	2018-09-06 17:37:11.7820	Subscriber
148139	2018-06-08 09:24:49.1610	Subscriber
117536	2018-07-13 21:29:25.6550	Subscriber
293811	2017-09-29 19:31:02.3850	Subscriber
77378	2018-01-09 08:18:02.2440	Subscriber
118074	2018-09-12 17:07:30.3020	Subscriber
76456	2018-10-19 15:18:00.9180	Subscriber
121566	2017-11-18 16:25:45.4940	Subscriber
88858	2018-09-17 13:33:12.9180	Customer
32078	2018-05-25 21:05:23.1420	Subscriber
105554	2018-07-16 12:20:37.6210	Subscriber
22853	2018-01-25 18:16:28.0650	Subscriber
285185	2017-10-02 19:25:21.5900	Subscriber
44800	2018-07-25 07:45:13.7330	Subscriber
27204	2018-11-25 10:44:55.0470	Subscriber
373224	2017-09-06 20:37:02.2190	Subscriber
138994	2018-10-10 09:01:03.7510	Subscriber
292	2018-08-31 21:15:52.5000	Subscriber
15418	2018-07-29 17:43:52.1500	Subscriber
170469	2018-07-06 09:19:55.4350	Subscriber
5202	2018-06-29 19:25:26.1370	Customer
52710	2018-03-18 21:11:02.1950	Subscriber
109255	2018-09-13 17:56:11.3450	Subscriber
3177	2018-10-31 16:06:26.1450	Subscriber
168856	2018-10-05 10:00:11.3590	Subscriber
58266	2018-04-18 22:21:57.1040	Subscriber

```
106144  2018-09-14 08:07:57.0680  Subscriber
104366  2018-11-06 06:45:08.0260  Subscriber
150497  2018-09-07 14:26:16.3020  Subscriber
```

In [326]:
```python
# View info of the dataframe
df.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2252058 entries, 0 to 201457
Data columns (total 17 columns):
Unnamed: 0                 519700 non-null float64
bike_id                    2252058 non-null int64
bike_share_for_all_trip    1732358 non-null object
duration_sec               2252058 non-null int64
end_station_id             2240479 non-null float64
end_station_latitude       2252058 non-null float64
end_station_longitude      2252058 non-null float64
end_station_name           2240479 non-null object
end_time                   2252058 non-null object
member_birth_year          2079810 non-null float64
member_gender              2080240 non-null object
start_station_id           2240479 non-null float64
start_station_latitude     2252058 non-null float64
start_station_longitude    2252058 non-null float64
start_station_name         2240479 non-null object
start_time                 2252058 non-null object
user_type                  2252058 non-null object
dtypes: float64(8), int64(2), object(7)
memory usage: 309.3+ MB
```

In [327]:
```python
# Check if duplicates exist
df.duplicated().sum()
```

Out[327]: 0

In [328]:
```python
# View descriptive statistics of the dataframe
df.describe()
```

Out[328]:

	Unnamed: 0	bike_id	duration_sec	end_station_id	\
count	519700.000000	2.252058e+06	2.252058e+06	2.240479e+06	
mean	259849.500000	2.101589e+03	9.181335e+02	1.114495e+02	
std	150024.611786	1.195229e+03	2.686599e+03	9.702559e+01	
min	0.000000	1.000000e+01	6.100000e+01	3.000000e+00	
25%	129924.750000	1.098000e+03	3.580000e+02	2.800000e+01	
50%	259849.500000	2.131000e+03	5.660000e+02	8.100000e+01	
75%	389774.250000	3.059000e+03	8.880000e+02	1.790000e+02	
max	519699.000000	4.466000e+03	8.636900e+04	3.810000e+02	

```
       end_station_latitude  end_station_longitude  member_birth_year  \
count          2.252058e+06           2.252058e+06       2.079810e+06
mean           3.776810e+01          -1.223520e+02       1.982467e+03
std            1.014484e-01           1.556892e-01       1.051074e+01
min            3.726331e+01          -1.224737e+02       1.881000e+03
25%            3.777166e+01          -1.224094e+02       1.977000e+03
50%            3.778175e+01          -1.223971e+02       1.985000e+03
75%            3.779539e+01          -1.222948e+02       1.990000e+03
max            4.551000e+01          -7.357000e+01       2.000000e+03

       start_station_id  start_station_latitude  start_station_longitude
count      2.240479e+06            2.252058e+06             2.252058e+06
mean       1.132275e+02            3.776797e+01            -1.223525e+02
std        9.713899e+01            1.015587e-01             1.560933e-01
min        3.000000e+00            3.726331e+01            -1.224737e+02
25%        3.000000e+01            3.777143e+01            -1.224114e+02
50%        8.100000e+01            3.778127e+01            -1.223974e+02
75%        1.800000e+02            3.779539e+01            -1.222948e+02
max        3.810000e+02            4.551000e+01            -7.357000e+01
```

Quality issues

```
-start time and end time are objects not a timestamps
-user type, gender and bike_share_for_all_trip can be set to category
-bike id, start_station_id, end_station_id can be set to object
-member birth year has dates prior to 1900
-we can calculate the age of the user
-we can further enhance the dataset with more details about the time like month, day, hour, week
-we can calculate the distance for rides between stations
```

4 Part III - Cleaning Data

In [329]:
```python
# Create copies of original DataFrames
df_clean = df.copy()
```

Define
Set appropriate data types for fields mentioned in the Quality issues
Code

In [330]:
```python
# set dates 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 [331]:
```python
# set user type, gender and bike_share_for_all_trip to category
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('category')
```

```
In [332]: # set 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)
```

Test

```
In [333]: df_clean.info(verbose=True, null_counts=True)

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2252058 entries, 0 to 201457
Data columns (total 17 columns):
Unnamed: 0                 519700 non-null float64
bike_id                    2252058 non-null object
bike_share_for_all_trip    1732358 non-null category
duration_sec               2252058 non-null int64
end_station_id             2252058 non-null object
end_station_latitude       2252058 non-null float64
end_station_longitude      2252058 non-null float64
end_station_name           2240479 non-null object
end_time                   2252058 non-null datetime64[ns]
member_birth_year          2079810 non-null float64
member_gender              2080240 non-null category
start_station_id           2252058 non-null object
start_station_latitude     2252058 non-null float64
start_station_longitude    2252058 non-null float64
start_station_name         2240479 non-null object
start_time                 2252058 non-null datetime64[ns]
user_type                  2252058 non-null category
dtypes: category(3), datetime64[ns](2), float64(6), int64(1), object(5)
memory usage: 264.2+ MB
```

Define
Calculate the age of the member
Code

```
In [334]: # substract the birth year from the current year
          df_clean['member_age'] = 2019-df_clean['member_birth_year']
```

Test

```
In [335]: df_clean.head(20)

Out[335]:    Unnamed: 0 bike_id bike_share_for_all_trip  duration_sec end_station_id  \
          0         NaN    1035                      No           598           1035
          1         NaN    1673                      No           943           1673
          2         NaN    3498                      No         18587           3498
          3         NaN    3129                      No         18558           3129
```

```
4         NaN   1839                    Yes         885         1839
5         NaN   2656                     No         921         2656
6         NaN   1616                     No         277         1616
7         NaN    144                     No         285          144
8         NaN   3351                     No         363         3351
9         NaN   1699                    Yes         226         1699
10        NaN    908                    Yes         219          908
11        NaN   2807                     No         261         2807
12        NaN     48                     No         530           48
13        NaN   3276                     No         762         3276
14        NaN   1450                    Yes         637         1450
15        NaN   1859                     No         789         1859
16        NaN    413                    Yes         144          413
17        NaN   2011                     No         258         2011
18        NaN     54                     No         280           54
19        NaN    439                     No        1983          439

    end_station_latitude  end_station_longitude  \
0              37.764478            -122.402570
1              37.788300            -122.408531
2              37.795392            -122.394203
3              37.795392            -122.394203
4              37.322980            -121.887931
5              37.350964            -121.902016
6              37.335885            -121.885660
7              37.808894            -122.256460
8              37.839649            -122.271756
9              37.332039            -121.881766
10             37.332039            -121.881766
11             37.333658            -121.908586
12             37.329732            -121.901782
13             37.842630            -122.267738
14             37.332039            -121.881766
15             37.332039            -121.881766
16             37.338395            -121.880797
17             37.763281            -122.407377
18             37.823321            -122.275732
19             37.797280            -122.398436

                                     end_station_name                end_time  \
0                        Rhode Island St at 17th St 2018-03-01 00:09:45.187
1                 Union Square (Powell St at Post St) 2018-02-28 23:36:59.974
2   San Francisco Ferry Building (Harry Bridges Pl... 2018-02-28 23:30:42.925
3   San Francisco Ferry Building (Harry Bridges Pl... 2018-02-28 23:30:12.450
4                              Locust St at Grant St 2018-02-28 23:29:58.608
5                              Mission St at 1st St 2018-02-28 23:29:40.437
6                          San Fernando St at 4th St 2018-02-28 23:26:27.222
7                            Grand Ave at Perkins St 2018-02-28 23:26:05.405
```

```
8                              Genoa St at 55th St 2018-02-28 23:25:22.274
9                        5th St at San Salvador St 2018-02-28 23:19:06.620
10                       5th St at San Salvador St 2018-02-28 23:19:03.068
11                       Morrison Ave at Julian St 2018-02-28 23:18:31.281
12                        San Jose Diridon Station 2018-02-28 23:18:16.868
13                             Dover St at 57th St 2018-02-28 23:15:51.269
14                       5th St at San Salvador St 2018-02-28 23:15:45.778
15                       5th St at San Salvador St 2018-02-28 23:13:03.377
16                          9th St at San Fernando 2018-02-28 23:12:08.821
17                     Potrero Ave and Mariposa St 2018-02-28 23:06:21.498
18                      Market St at Brockhurst St 2018-02-28 23:05:16.208
19                          Davis St at Jackson St 2018-02-28 23:02:59.697

    member_birth_year member_gender start_station_id  start_station_latitude  \
0              1988.0          Male             1035               37.784872
1              1987.0          Male             1673               37.804770
2              1986.0        Female             3498               37.770407
3              1981.0          Male             3129               37.770407
4              1976.0        Female             1839               37.336802
5              1997.0          Male             2656               37.329732
6              1957.0        Female             1616               37.330165
7              1990.0        Female              144               37.807813
8              1975.0          Male             3351               37.828410
9              1996.0          Male             1699               37.332794
10             1995.0          Male              908               37.332794
11             1972.0          Male             2807               37.332692
12             1985.0          Male               48               37.326730
13             1988.0        Female             3276               37.868813
14             1998.0          Male             1450               37.335388
15             1997.0        Female             1859               37.332692
16             1990.0          Male              413               37.335885
17             1989.0          Male             2011               37.770030
18             1984.0          Male               54               37.828410
19             1990.0          Male              439               37.759210

    start_station_longitude  \
0               -122.400876
1               -122.403234
2               -122.391198
3               -122.391198
4               -121.894090
5               -121.901782
6               -121.885831
7               -122.264496
8               -122.266315
9               -121.875926
10              -121.875926
11              -121.900084
```

```
12              -121.889273
13              -122.258764
14              -121.897921
15              -121.900084
16              -121.885660
17              -122.411726
18              -122.266315
19              -122.421339

                                 start_station_name              start_time  \
0   Yerba Buena Center for the Arts (Howard St at ... 2018-02-28 23:59:47.097
1                    The Embarcadero at Sansome St 2018-02-28 23:21:16.495
2                     4th St at Mission Bay Blvd S 2018-02-28 18:20:55.190
3                     4th St at Mission Bay Blvd S 2018-02-28 18:20:53.621
4                                 San Pedro Square 2018-02-28 23:15:12.858
5                         San Jose Diridon Station 2018-02-28 23:14:19.170
6                        San Salvador St at 1st St 2018-02-28 23:21:49.274
7                                        Snow Park 2018-02-28 23:21:19.631
8                            MacArthur BART Station 2018-02-28 23:19:18.606
9                             William St at 10th St 2018-02-28 23:15:20.033
10                            William St at 10th St 2018-02-28 23:15:23.480
11                                       SAP Center 2018-02-28 23:14:09.368
12                      Almaden Blvd at Balbach St 2018-02-28 23:09:26.795
13                   Bancroft Way at Telegraph Ave 2018-02-28 23:03:08.627
14           W St John St at Guadalupe River Trail 2018-02-28 23:05:08.754
15                                       SAP Center 2018-02-28 22:59:54.088
16                       San Fernando St at 4th St 2018-02-28 23:09:44.738
17                            11th St at Bryant St 2018-02-28 23:02:02.525
18                           MacArthur BART Station 2018-02-28 23:00:35.761
19                              Mission Playground 2018-02-28 22:29:56.631

     user_type  member_age
0   Subscriber        31.0
1     Customer        32.0
2     Customer        33.0
3     Customer        38.0
4   Subscriber        43.0
5     Customer        22.0
6   Subscriber        62.0
7   Subscriber        29.0
8   Subscriber        44.0
9   Subscriber        23.0
10  Subscriber        24.0
11  Subscriber        47.0
12  Subscriber        34.0
13  Subscriber        31.0
14  Subscriber        21.0
15  Subscriber        22.0
```

```
16   Subscriber        29.0
17   Subscriber        30.0
18   Subscriber        35.0
19   Subscriber        29.0
```

Define
Enhance dataset with new date related fields
Code

In [336]:
```
# extract start time month name
df_clean['start_time_month_name']=df_clean['start_time'].dt.strftime('%B')
```

In [337]:
```
# extract start time month number
df_clean['start_time_month']=df_clean['start_time'].dt.month.astype(int)
```

In [338]:
```
# extract start time weekdays
df_clean['start_time_weekday']=df_clean['start_time'].dt.strftime('%a')
```

In [339]:
```
# extract start time day
df_clean['start_time_day']=df_clean['start_time'].dt.day.astype(int)
```

In [340]:
```
# extract start time hour
df_clean['start_time_hour']=df_clean['start_time'].dt.hour
```

Test

In [341]:
```
df_clean.head()
```

Out[341]:
```
   Unnamed: 0 bike_id bike_share_for_all_trip  duration_sec end_station_id  \
0         NaN     1035                      No           598           1035
1         NaN     1673                      No           943           1673
2         NaN     3498                      No         18587           3498
3         NaN     3129                      No         18558           3129
4         NaN     1839                     Yes           885           1839

   end_station_latitude  end_station_longitude  \
0             37.764478            -122.402570
1             37.788300            -122.408531
2             37.795392            -122.394203
3             37.795392            -122.394203
4             37.322980            -121.887931

                                  end_station_name                end_time  \
0                           Rhode Island St at 17th St 2018-03-01 00:09:45.187
1                  Union Square (Powell St at Post St) 2018-02-28 23:36:59.974
2  San Francisco Ferry Building (Harry Bridges Pl... 2018-02-28 23:30:42.925
3  San Francisco Ferry Building (Harry Bridges Pl... 2018-02-28 23:30:12.450
4                                Locust St at Grant St 2018-02-28 23:29:58.608
```

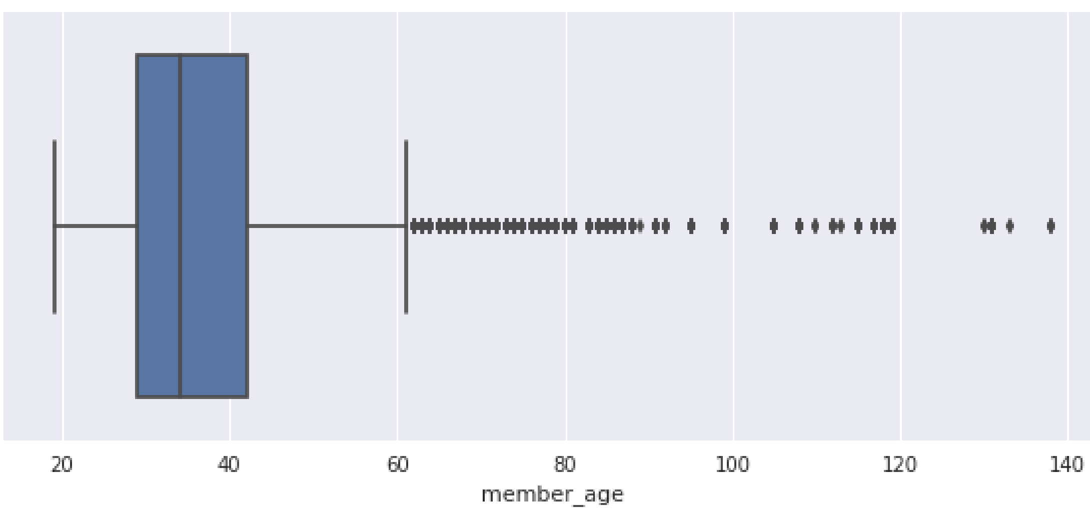
```
   member_birth_year       ...       start_station_longitude  \
0             1988.0       ...                   -122.400876
1             1987.0       ...                   -122.403234
2             1986.0       ...                   -122.391198
3             1981.0       ...                   -122.391198
4             1976.0       ...                   -121.894090

                                start_station_name              start_time  \
0  Yerba Buena Center for the Arts (Howard St at ... 2018-02-28 23:59:47.097
1                     The Embarcadero at Sansome St 2018-02-28 23:21:16.495
2                      4th St at Mission Bay Blvd S 2018-02-28 18:20:55.190
3                      4th St at Mission Bay Blvd S 2018-02-28 18:20:53.621
4                                  San Pedro Square 2018-02-28 23:15:12.858

    user_type member_age start_time_month_name start_time_month  \
0  Subscriber       31.0              February                2
1    Customer       32.0              February                2
2    Customer       33.0              February                2
3    Customer       38.0              February                2
4  Subscriber       43.0              February                2

   start_time_weekday start_time_day  start_time_hour
0                 Wed             28               23
1                 Wed             28               23
2                 Wed             28               18
3                 Wed             28               18
4                 Wed             28               23

[5 rows x 23 columns]
```

In [342]:
```
df_clean.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2252058 entries, 0 to 201457
Data columns (total 23 columns):
Unnamed: 0                 519700 non-null float64
bike_id                    2252058 non-null object
bike_share_for_all_trip    1732358 non-null category
duration_sec               2252058 non-null int64
end_station_id             2252058 non-null object
end_station_latitude       2252058 non-null float64
end_station_longitude      2252058 non-null float64
end_station_name           2240479 non-null object
end_time                   2252058 non-null datetime64[ns]
member_birth_year          2079810 non-null float64
member_gender              2080240 non-null category
start_station_id           2252058 non-null object
start_station_latitude     2252058 non-null float64
```

```
start_station_longitude    2252058 non-null float64
start_station_name         2240479 non-null object
start_time                 2252058 non-null datetime64[ns]
user_type                  2252058 non-null category
member_age                 2079810 non-null float64
start_time_month_name      2252058 non-null object
start_time_month           2252058 non-null int64
start_time_weekday         2252058 non-null object
start_time_day             2252058 non-null int64
start_time_hour            2252058 non-null int64
dtypes: category(3), datetime64[ns](2), float64(7), int64(4), object(7)
memory usage: 367.3+ MB
```

In [343]:
```
# code for the age boxplot

plt.figure(figsize = [10, 4])
base_color = sns.color_palette()[0]

sns.boxplot(data=df_clean, x='member_age', color=base_color);
```



In [344]:
```
df_clean.member_age.mean()
```

Out[344]: 36.53289483173944

In [345]:
```
df_clean.member_age.describe(percentiles = [ .95])
```

Out[345]:
```
count     2.079810e+06
mean      3.653289e+01
std       1.051074e+01
```

```
min        1.900000e+01
50%        3.400000e+01
95%        5.700000e+01
max        1.380000e+02
Name: member_age, dtype: float64
```

Define

Remove age outliers. As mentioned in the Quality issues, there are customers with the birth year before 1900 thus customers with age above 100 years. As 95% of the users are below 58 , I am going to keep users below 60.

Code

```
In [346]: # Keep records below 60, it automatically removes null values
          df_clean = df_clean.query('member_age <=60')
```

```
In [347]: # change age and birth year to integer
          df_clean.member_age = df_clean.member_age.astype(int)
          df_clean.member_birth_year = df_clean.member_birth_year.astype(int)
```

Test

```
In [348]: df_clean.describe()
```

```
Out[348]:           Unnamed: 0  duration_sec  end_station_latitude  \
          count  436822.000000  2.021694e+06          2.021694e+06
          mean   254931.294974  7.915458e+02          3.776762e+01
          std    148988.491497  2.138149e+03          1.024279e-01
          min         0.000000  6.100000e+01          3.726331e+01
          25%    125832.250000  3.500000e+02          3.777143e+01
          50%    253015.500000  5.460000e+02          3.778127e+01
          75%    381832.750000  8.400000e+02          3.779539e+01
          max    519699.000000  8.628100e+04          4.551000e+01

                 end_station_longitude  member_birth_year  start_station_latitude  \
          count           2.021694e+06       2.021694e+06            2.021694e+06
          mean           -1.223510e+02       1.983347e+03            3.776752e+01
          std             1.599688e-01       9.127963e+00            1.025529e-01
          min            -1.224737e+02       1.959000e+03            3.726331e+01
          25%            -1.224094e+02       1.978000e+03            3.777106e+01
          50%            -1.223971e+02       1.985000e+03            3.778107e+01
          75%            -1.222914e+02       1.990000e+03            3.779539e+01
          max            -7.357000e+01       2.000000e+03            4.551000e+01

                 start_station_longitude    member_age  start_time_month  \
          count             2.021694e+06  2.021694e+06      2.021694e+06
          mean             -1.223516e+02  3.565346e+01      7.283402e+00
          std               1.604003e-01  9.127963e+00      2.962403e+00
          min              -1.224737e+02  1.900000e+01      1.000000e+00
          25%              -1.224114e+02  2.900000e+01      5.000000e+00
```

```
50%              -1.223974e+02  3.400000e+01      8.000000e+00
75%              -1.222914e+02  4.100000e+01      1.000000e+01
max              -7.357000e+01  6.000000e+01      1.200000e+01

       start_time_day  start_time_hour
count    2.021694e+06     2.021694e+06
mean     1.582123e+01     1.349412e+01
std      8.819759e+00     4.748167e+00
min      1.000000e+00     0.000000e+00
25%      8.000000e+00     9.000000e+00
50%      1.600000e+01     1.400000e+01
75%      2.400000e+01     1.700000e+01
max      3.100000e+01     2.300000e+01
```

```
In [349]: df_clean.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2021694 entries, 0 to 201457
Data columns (total 23 columns):
Unnamed: 0                 436822 non-null float64
bike_id                    2021694 non-null object
bike_share_for_all_trip    1584872 non-null category
duration_sec               2021694 non-null int64
end_station_id             2021694 non-null object
end_station_latitude       2021694 non-null float64
end_station_longitude      2021694 non-null float64
end_station_name           2010488 non-null object
end_time                   2021694 non-null datetime64[ns]
member_birth_year          2021694 non-null int64
member_gender              2021694 non-null category
start_station_id           2021694 non-null object
start_station_latitude     2021694 non-null float64
start_station_longitude    2021694 non-null float64
start_station_name         2010488 non-null object
start_time                 2021694 non-null datetime64[ns]
user_type                  2021694 non-null category
member_age                 2021694 non-null int64
start_time_month_name      2021694 non-null object
start_time_month           2021694 non-null int64
start_time_weekday         2021694 non-null object
start_time_day             2021694 non-null int64
start_time_hour            2021694 non-null int64
dtypes: category(3), datetime64[ns](2), float64(5), int64(6), object(7)
memory usage: 329.7+ MB
```

What is the structure of your dataset?
Originally there were approx. 185,000 bike rides that happen in 2018 in the San Francisco Bay Area. The dataset contained features about:

```
-trip duration: start/end time, how long the trip took in seconds
-stations: start/end station, name, geolocation (latitude/longitude)
-anonymized customer data: gender, birth date and user type
-rented bikes: bike id
```

The dataset was further enhanced with features that I may find neccessary to perform interesting analysis:

```
-rental time: month, day, hour of the day, weekday (both for start and end date)
-customer: age
```

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

I'm most interested in figuring out when and where bikes are high in demand (during the day/weekday/month). Moreover which age range and gender uses the service the most and if the service is mostly used by members or casual riders.

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

I expect that the start time will be most exploited in my analysis as well as customer related data. I expect that location and datetime will have the strongest effect on bike demand.

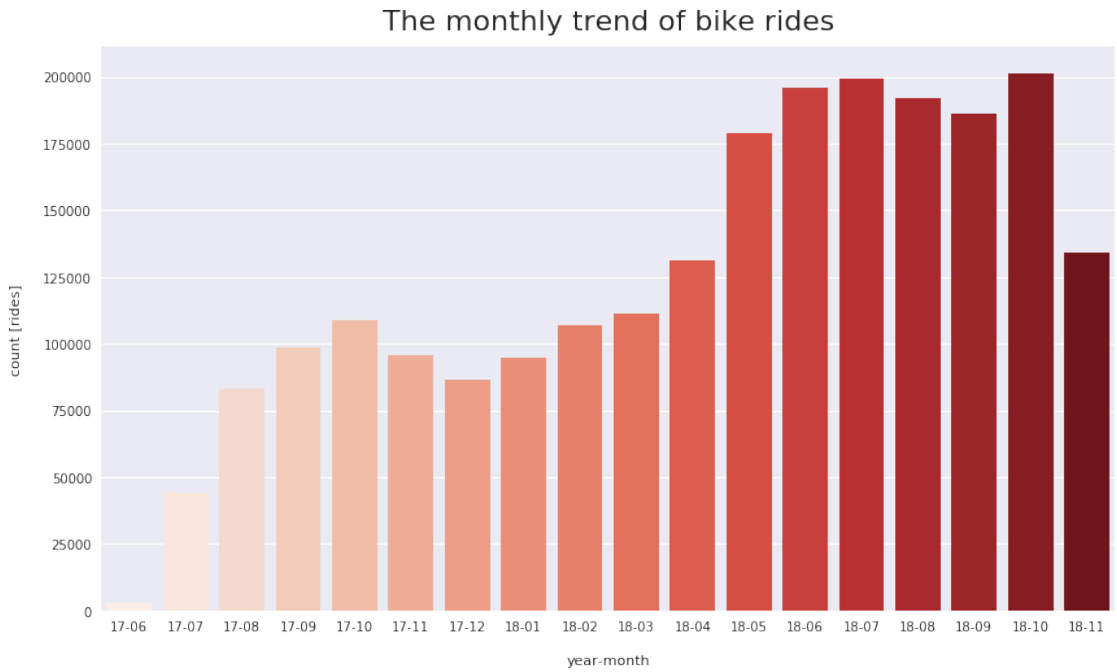
Part IV - Univariate Exploration

I'll start by determine start time and end time, then looking at the monthly trend of bike rides

In [350]:
```
#Generate new fields for date from start_time and end_time
df['start_time']=pd.to_datetime(df['start_time'])
df['end_time']=pd.to_datetime(df['end_time'])
df['start_time_date']=df['start_time'].dt.date
df['end_time_date']=df['end_time'].dt.date
df['start_time_year_month']=df['start_time'].map(lambda x: x.strftime('%Y-%m'))
df['end_time_year_month']=df['end_time'].map(lambda x: x.strftime('%Y-%m'))
df['start_time_year_month_renamed'] = df['start_time'].dt.strftime('%y' + '-' + '%m')
df['start_time_year']=df['start_time'].dt.year.astype(int)
df['end_time_year']=df['end_time'].dt.year.astype(int)
df['start_time_month']=df['start_time'].dt.month.astype(int)
df['end_time_month']=df['end_time'].dt.month.astype(int)
df['start_time_hour_minute']=df['start_time'].map(lambda x: x.strftime('%H-%m'))
df['end_time_hour_minute']=df['end_time'].map(lambda x: x.strftime('%H-%m'))
df['start_time_hour']=df['start_time'].dt.hour
df['end_time_hour']=df['end_time'].dt.hour
df['start_time_weekday']=df['start_time'].dt.weekday_name
df['end_time_weekday']=df['end_time'].dt.weekday_name
df['start_time_weekday_abbr']=df['start_time'].dt.weekday.apply(lambda x: calendar.day
df['end_time_weekday_abbr']=df['end_time'].dt.weekday.apply(lambda x: calendar.day_abb
```

In [351]:
```
# monthly usege of the bike sharing system
plt.figure(figsize=(14,8))
sns.countplot(x='start_time_year_month_renamed', palette="Reds", data=df.sort_values(b
plt.title('The monthly trend of bike rides', fontsize=22, y=1.015)
plt.xlabel('year-month', labelpad=16)
plt.ylabel('count [rides]', labelpad=16)
```

```
ax = plt.gca()
plt.savefig('image03.png')
```



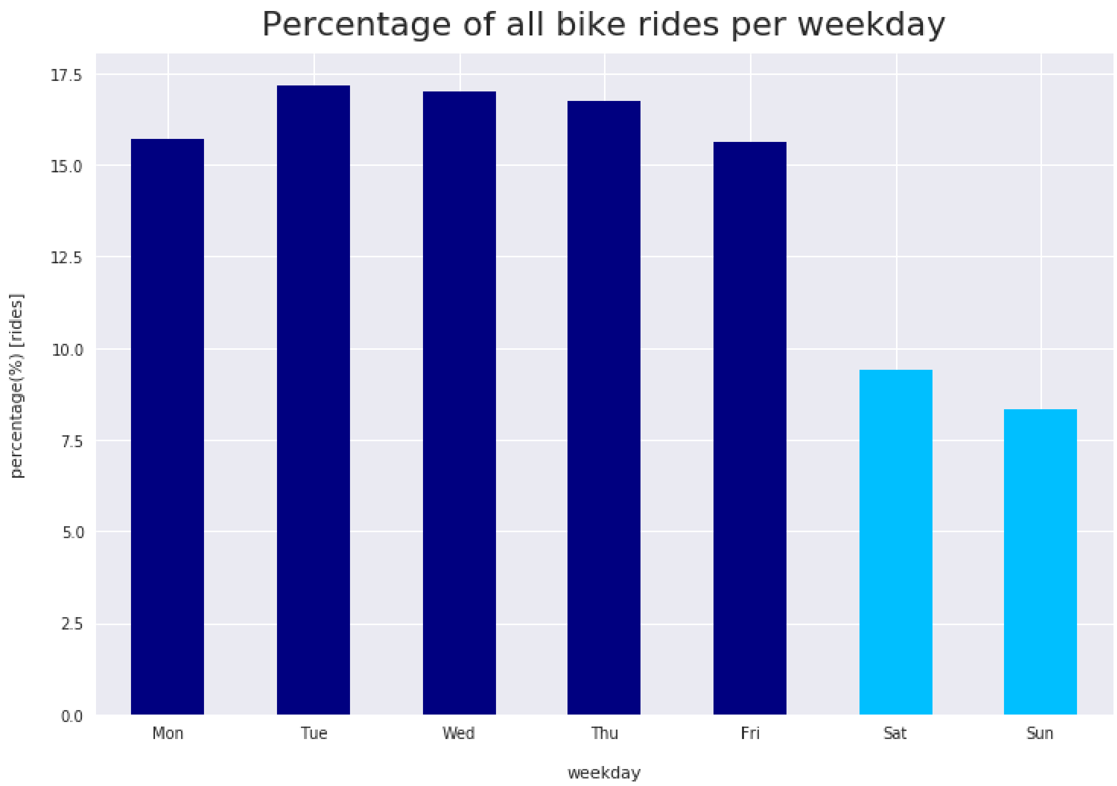
There is seasonality when the season is winter because it is cold. However, bike rides of July 2017 and 2018 increased more than 5 times.

Winter months are the worst for the bike sharing system most probably due to the weather conditions. The bike renting is high in demand between May and October, reaching its peak in October, followed by July.

Bike rides per weekday

Determine precentage trend of bike rides per weekday

```
In [352]: # weekday usege of the bike

          trip_by_weekday_df = df.groupby('start_time_weekday_abbr').agg({'bike_id':'count'})

In [353]: trip_by_weekday_df['perc'] = (trip_by_weekday_df['bike_id']/trip_by_weekday_df['bike_i

In [354]: weekday_index = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']

In [355]: new_color = ['navy', 'navy', 'navy', 'navy', 'navy', 'deepskyblue', 'deepskyblue']
          trip_by_weekday_df.reindex(weekday_index)['perc'].plot(kind='bar', color=new_color, fi
          plt.title('Percentage of all bike rides per weekday', fontsize=22, y=1.015)
          plt.xlabel('weekday', labelpad=16)
          plt.ylabel('percentage(%) [rides]', labelpad=16)
          plt.xticks(rotation=360)
          plt.savefig('image07.png');
```



The bike share system is mainly used during weekdays, with Tuesday - Thursday as the most popular days for bike rides. The system is most probably used as a daily work/school commute.

People use this service on weekdays more than weekends.

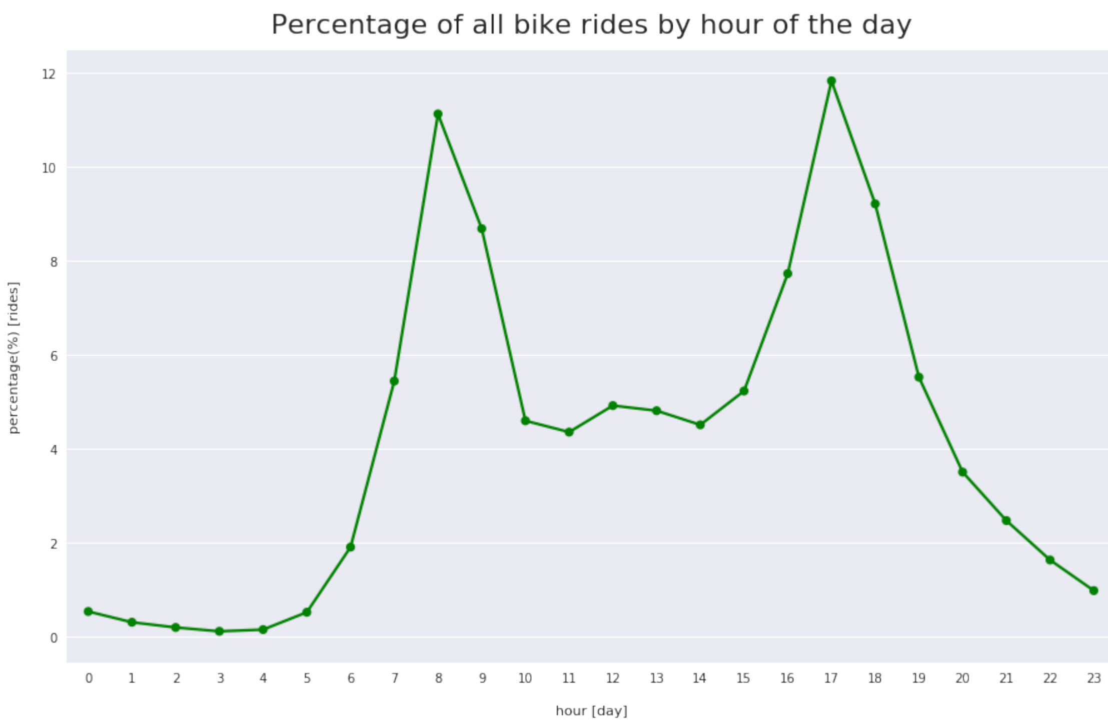
Bike rides hourly

Determine precentage trend of bike rides per hour of the day

In [356]:
```
# hourly usege of the bike sharing system

trip_by_hour_df = df.groupby('start_time_hour').agg({'bike_id':'count'}).reset_index()
```

In [357]:
```
trip_by_hour_df['bike_id'] = (trip_by_hour_df['bike_id']/trip_by_hour_df['bike_id'].su
```

In [358]:
```
plt.figure(figsize=(15,9))
sns.pointplot(x='start_time_hour', y='bike_id', scale=.7, color='green', data=trip_by_
plt.title('Percentage of all bike rides by hour of the day', fontsize=22, y=1.015)
plt.xlabel('hour [day]', labelpad=16)
plt.ylabel('percentage(%) [rides]', labelpad=16)
plt.savefig('image08.png');
```



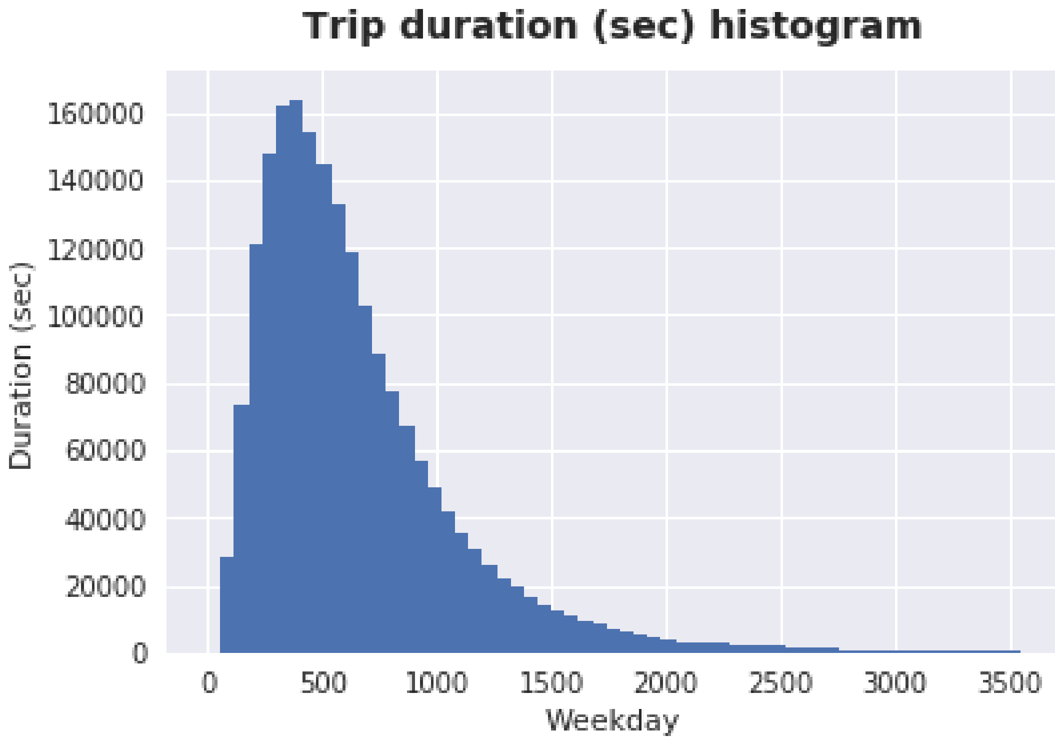
The hourly distribution is bimodal, the system is used mainly around 8-9am and 5-6pm when people get to and gat back from work.

8am and 5pm are the peak hours for this service. Also, people use this service when they are in lunch time as well.

Trip duration

Determine trip duration by second

In [359]:
```
# code for the (histogram) duration (sec) distribution per user type

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

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

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



Looking at the histogram, we can see that trip durations are no longer than 30 min (1800 sec) and usually last 6 to 15 min. This can be explained by two facts:

1.The way the system works: single trips and 24h or 72h access pass are free of additional charge for trips up to 30 min, otherwise you pay extra $3 for additional 15 min. Only the monthly pass offers free of charge 45 min rides.

2.The way the system is used: as is looks like people use the system for commuting, they trips are usually short in time probably due to the closeness of their homes to workplace/school.

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

There was one unusal points for the duration (sec), which in some cases lasted more than 24h. For the histogram I set the 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?

There was one unusal distribution for the member birth year, which in some cases was dated before 1900. Since 95% of the members are between 17 and 57 years, I removed users older than 60.

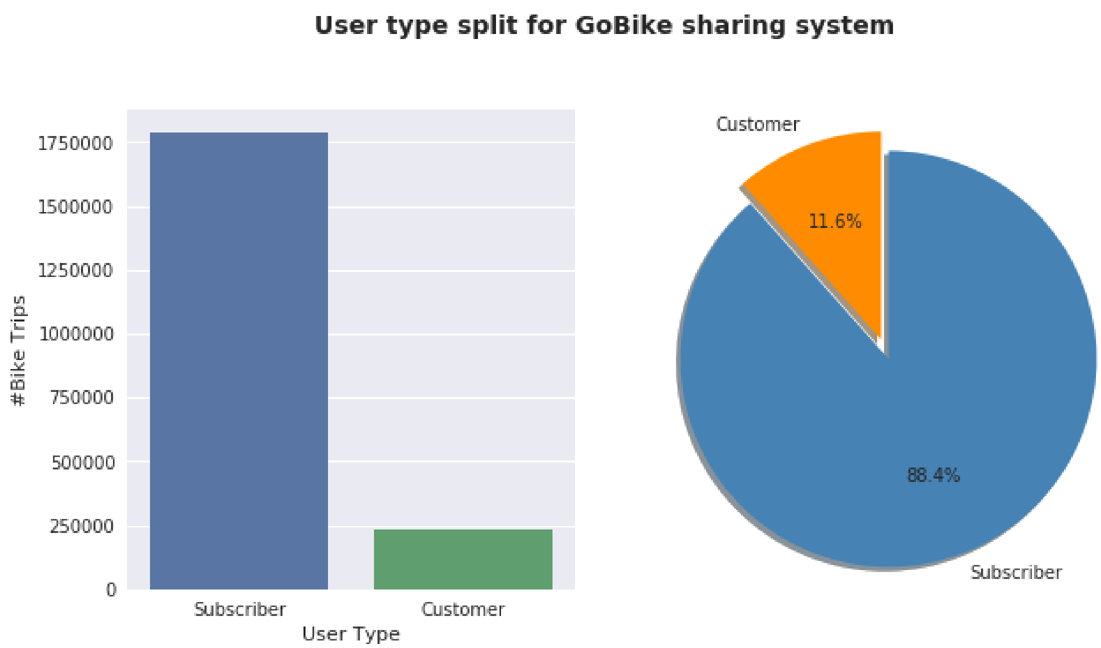
5 Part V - Bivariate Exploration

In this section I will further explore the dataset by adding the customer type to the analysis.

```
In [360]: # calculating % split for the user type
          customer = df_clean.query('user_type == "Customer"')['bike_id'].count()
```

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

customer_proportion = customer / df_clean['bike_id'].count()
subscriber_proportion = subscriber / df_clean['bike_id'].count()
```

In [361]:
```
plt.figure(figsize = [10, 5])

# code for the bar chart
plt.subplot(1, 2, 1)

g = sns.countplot(data=df_clean, x="user_type", order=df_clean.user_type.value_counts(
g.set_xlabel('User Type')
g.set_ylabel('#Bike Trips')

# code for the pie chart
plt.subplot(1, 2, 2)

labels = ['Customer', 'Subscriber']
sizes = [customer_proportion, subscriber_proportion]
colors = ['darkorange', 'steelblue']
explode = (0, 0.1)

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

plt.suptitle('User type split for GoBike sharing system', y=1.03, fontsize=14, fontwei
```



The bike sharing system is mainly used by subscribers (88%) than ocassional riders (12%). Next I will see the monthly trend of bike rides

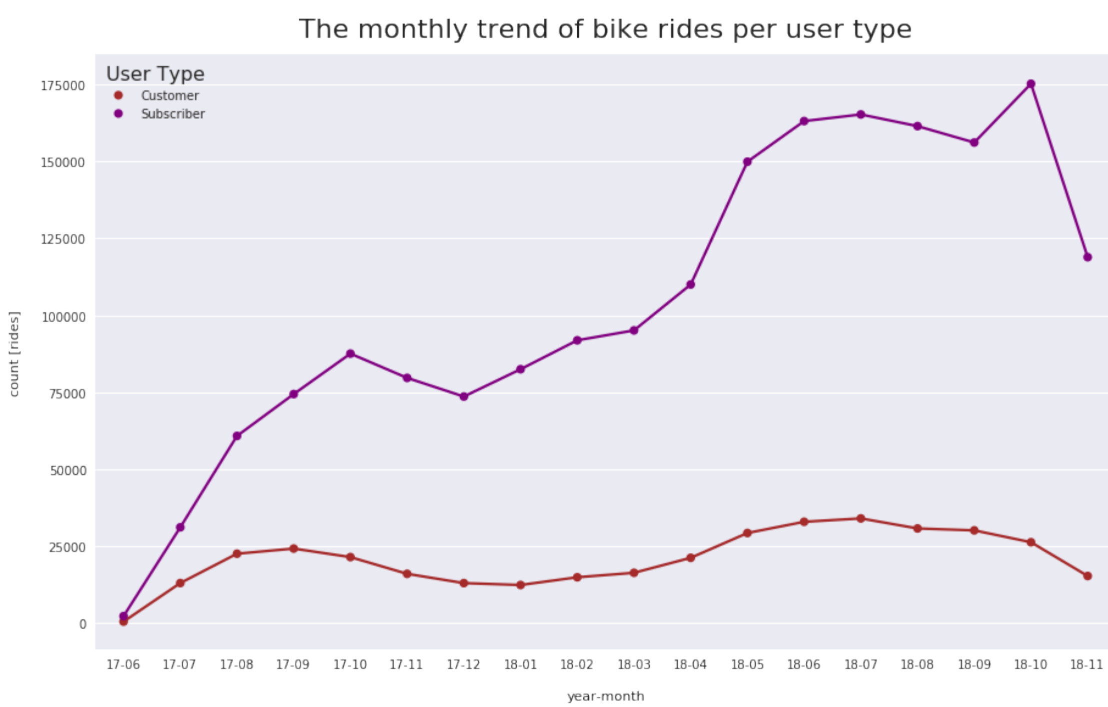
In [362]:
```
# monthly usege of the bike sharing system per user type

user_type_count_per_year_df = df.groupby(["start_time_year_month_renamed", "user_type"
```

In [363]:
```
# weekday usege of the bike sharing system per user type

count_of_rides_per_user_type = df.groupby('user_type').size().reset_index(name='count'
count_of_rides_per_user_type['count']/len(df)*100
```

Out[363]:
```
0    16.588649
1    83.411351
Name: count, dtype: float64
```

Percentage of subscribers is almost %88.15.
Percentage of customers is almost %11.85.

In [364]:
```
plt.figure(figsize=(15,9))
my_palette = {'Subscriber':'purple', 'Customer':'brown'}
ax = sns.pointplot(x='start_time_year_month_renamed', y=0, hue='user_type', palette=my
plt.title('The monthly trend of bike rides per user type', fontsize=22, y=1.015)
plt.xlabel('year-month', labelpad=16)
plt.ylabel('count [rides]', labelpad=16)
leg = ax.legend()
leg.set_title('User Type',prop={'size':16})
ax = plt.gca()
plt.savefig('image09.png');
```



Customers' rides seems increasing slightly. There is a decrease on November 2018 for subscribers but it seems like it is related with winter season.

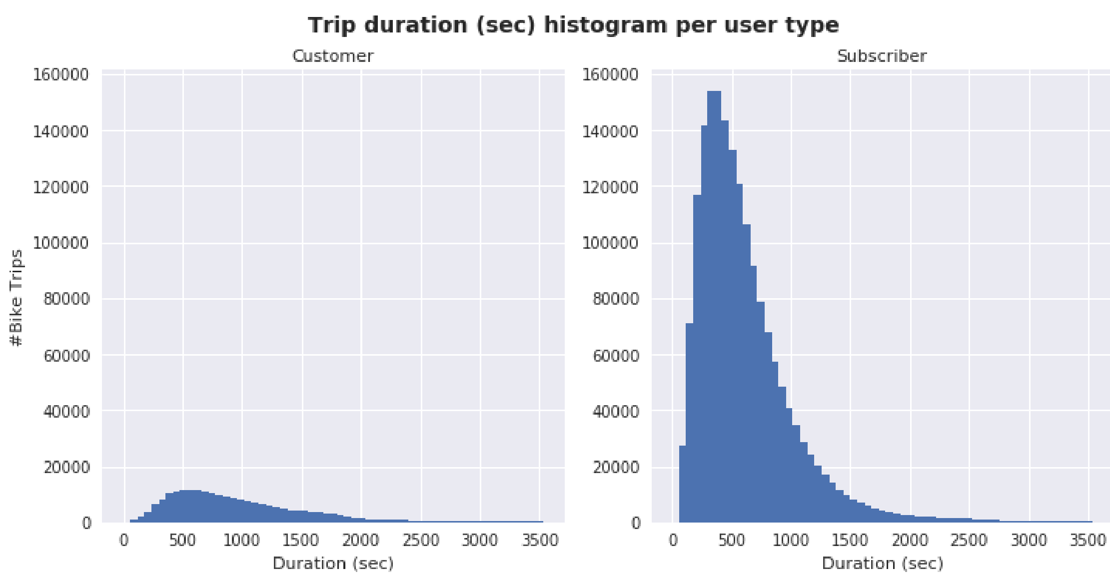
Winter months are the worst for the bike sharing system for both groups what can be determined by the harsher weather.

For Customers, the bike renting is high in demand around summertime, reaching its peak in July. Customers are most probably occasional reiders or tourist coming to visit the Bay Area. For Subscribers, the highest demand is from May till October, reaching it's peak in October. Customers are most probably regular riders using bikes for a daily commute.

There is also a different trend of when during the day bikes are rented most often. Customers use bikes mainly between 8 am - 7 pm, reaching the renting peak around 5pm. Subscribers on the other side use the system at around 8-9am and 5-6pm when they go and come back from work.

Next, I am going to check how the trip duration varies between customers and subscribers.

```
In [365]: # code for the (histogram) 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=base_color, bins=bin_edges)
          g.set_axis_labels("Duration (sec)", "#Bike Trips")
          g.set_titles(col_template = '{col_name}')
          g.fig.suptitle('Trip duration (sec) histogram per user type', y=1.03, fontsize=14, fon
```



Looking at both charts (histograms and box plots), we can see that trip durations are longer for customers (9 to 23 minutes) than for subscribers (7 to 13 minutes). This can probably be explained by the fact that subscribers are mainly commuters who take short trips to work/school rather than longer trips around the Bay Area.

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?

Adding the user type to the analysis depicted different usage behaviours between customers and subscribers. As mentioned above customers are casual riders, most probably tourists who rent bikes mainly in summertime (the peak in July), more often during weekends than weekdays and they rent bikes more often within the day rather than around commute hours (8-9am and 5-6pm). Subscribers are daily commuters, who also use the system around summertime, May-October (with the peak in October). They rent bikes more often during weekdays than weekends and mainly around the time they go and go back from work or school (8-9am and 5-6pm).

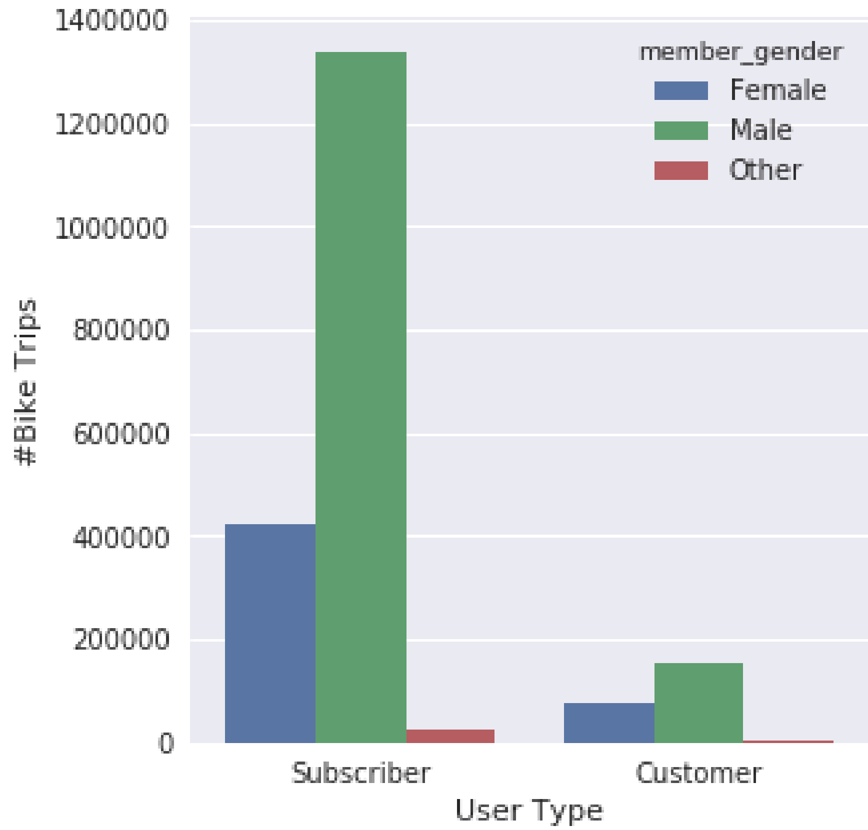
Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

There is a difference in the trip duration between customers and subscribers. Customers trips are usually longer than for subscribers, most probably due to the fact they prefer bike rides around weekends in summertime, what encourages longer trips around the area. Subscribers on the other hand use the system mainly for commute purposes so they rather prefer quick rides to and from work/school.

6 Part VI - Multivariate Exploration

In this section I will further explore the dataset by adding gender to the customer type and check the hourly distribution of bike rides during weekdays for customers and subscribers.

```
In [366]: plt.figure(figsize = [10, 5])

          # code for the bar chart
          plt.subplot(1, 2, 1)

          g = sns.countplot(data=df_clean, x="user_type", hue="member_gender", order=df_clean.us
          g.set_xlabel('User Type')
          g.set_ylabel('#Bike Trips');
```



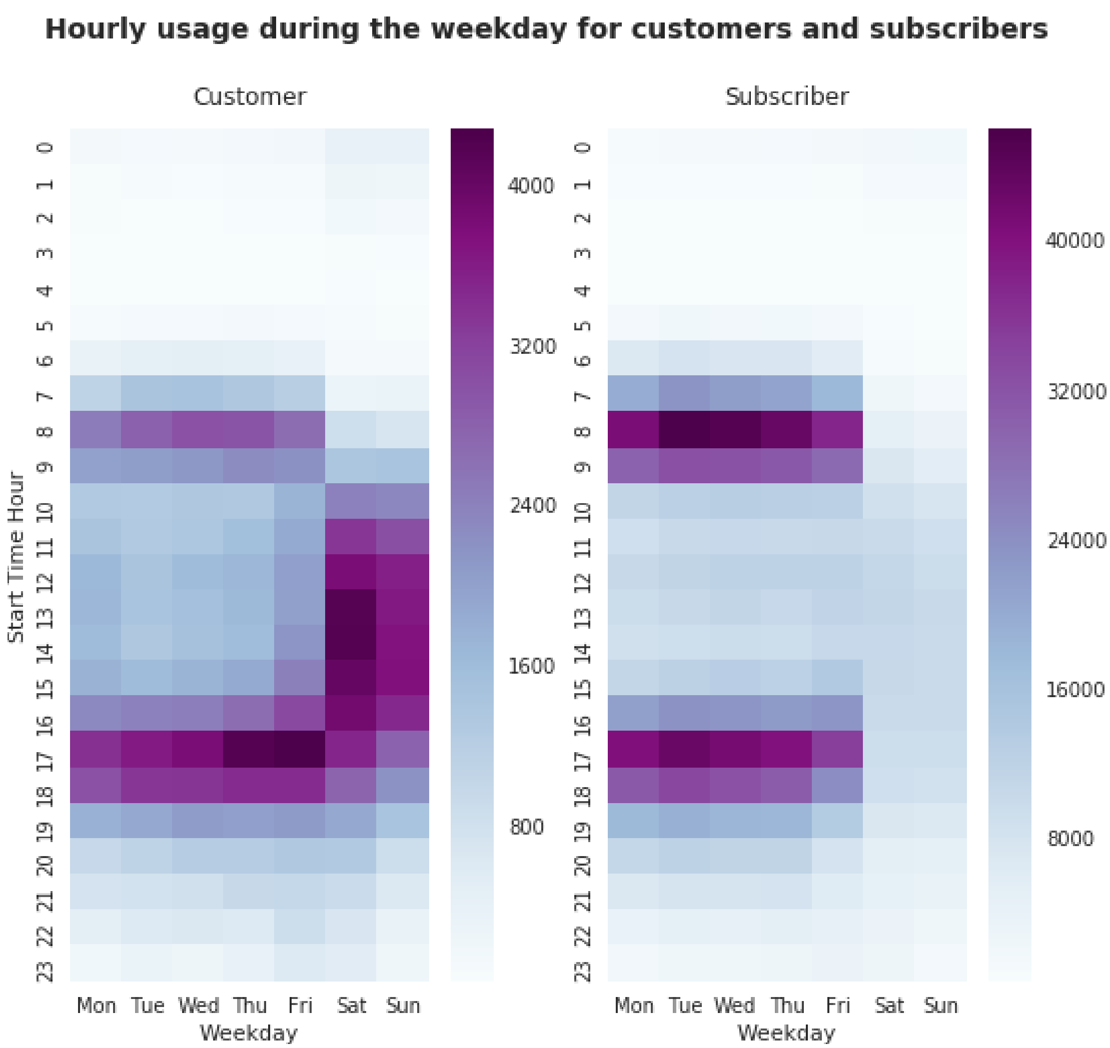
In general, males are using the system more often than females and others (the registration system allows you to choose 'Other' as a gender). However, the ratio is much smaller between males and females for customers (more ore less 2:1) than for subscribers (3:1).

Let's explore if gender affects the way the bike system is used within a year, weekdays and hours of the day.

Here we can observe that in both cases, females take longer trips (measured in time) than males and other. The difference is more visible for customers (~13 min for males and other vs ~15 for females) than for subscribers (the difference is quite small).

```
In [367]: # Setting the weekday order
          df_clean['start_time_weekday'] = pd.Categorical(df_clean['start_time_weekday'],
                                                           categories=['Mon','Tue','Wed','Thu','F
                                                           ordered=True)
          plt.figure(figsize=(9,8))
          plt.suptitle('Hourly usage during the weekday for customers and subscribers', fontsize

          # heatmap for customers
          plt.subplot(1, 2, 1)
          df_customer = df_clean.query('user_type == "Customer"').groupby(["start_time_hour", "s
          df_customer = df_customer.pivot("start_time_hour", "start_time_weekday", "bike_id")
          sns.heatmap(df_customer, cmap="BuPu")
```

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

# heatmap for subscribers
plt.subplot(1, 2, 2)
df_subscriber = df_clean.query('user_type == "Subscriber"').groupby(["start_time_hour"
df_subscriber = df_subscriber.pivot("start_time_hour", "start_time_weekday", "bike_id"
sns.heatmap(df_subscriber, cmap="BuPu")

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



The plot perfectly summarizes in one place the diffrent trends for customers and subscribers I was writing up before.

Customers use the bike sharing system more often on weekends:

```
weekdays: most bike rides hapen around 8-9am and 5-6pm with the peak on Fridays around 5pm
weekends: most bike rides happen between 10am - 8pm with the peak on Saturdays around 2pm
```

Subscribers use the bike sharing system mainly on weekdays:

```
weekdays: most bike rides hapen around 8-9am and 5-6pm with the peak on Tuesdays around 8am
weekends: bikes are still rented but there is a significant drop in numbers of rented bikes thro
```

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?

Plotting a heatmap of when bikes are high in demand throughout the day on each weekday shed a new light on the customers behaviour. Plotting #bike trips throughout the day and #bike trips within the weekdays separately gave the impression that the demand for bikes is quite high throughout the day with a peak around 5pm which is not entirely true. The trend within weekdays for customers follows (although customers are rather not early birds) the one for subscribers who rent bikes mainly around commute hours (8-9am and 5-6pm). For customers, as depicted in univariate explorations, most of the trips happen on weekends but mainly between 10am - 8pm with the peak on Saturdays around 2pm, what was previously not visible.

Were there any interesting or surprising interactions between features?

I have also checked if there is a trend difference for genders for each user group. There are not much of the differences in trends but surprisingly there are quite a lot of females using the system between January and March in comparison to males - the ratio (male:female) is much smaller than for the rest of the year. Moreover females take longer trips (measured in time) than males and others.

```
In [ ]:
```