Cyclista EDA

April 5, 2023

1 Step 0: Import and Reading Data

[1]:
```
#Import the most common library for EDA.

import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
import glob
from datetime import datetime
plt.style.use('ggplot')
```

2 Check current directory

[2]:
```
pwd
```

[2]:
```
'C:\\Users\\Ana\\Python Exploratory Data Analysis'
```

3 Retrive CSV file from the address

[3]:
```
df = pd.read_csv('C:/Users/Ana/Python Exploratory Data Analysis/CSV Files/DAX.
↪csv', low_memory=False)
```

4 Step 1: Data Understanding

- Dataframe shape
- head and tail
- dtypes
- describe

[4]:
```
df.shape
```

[4]:
```
(5667717, 18)
```

[5]:
```
df.head()
```

[5]:
```
   rideable_type              started_at                ended_at  \
0  electric_bike  2022-04-28 10:36:10.000  2022-04-28 10:38:20.000
1  electric_bike  2022-04-28 13:00:52.000  2022-04-28 13:03:23.000
2  electric_bike  2022-04-28 13:12:51.000  2022-04-28 13:15:28.000
3  electric_bike  2022-04-28 10:49:18.000  2022-04-28 10:52:00.000
4  electric_bike  2022-04-28 15:40:20.000  2022-04-28 15:43:11.000

  start_station_name end_station_name  start_lat  start_lng  end_lat  end_lng  \
0                NaN              NaN      41.79      -87.6    41.79    -87.6
1                NaN              NaN      41.79      -87.6    41.79    -87.6
2                NaN              NaN      41.79      -87.6    41.79    -87.6
3                NaN              NaN      41.79      -87.6    41.79    -87.6
4                NaN              NaN      41.79      -87.6    41.79    -87.6

  member_casual day_of_week  day_of_week_no              start_date  \
0        member    Thursday               4  2022-04-28 00:00:00.000
1        member    Thursday               4  2022-04-28 00:00:00.000
2        member    Thursday               4  2022-04-28 00:00:00.000
3        member    Thursday               4  2022-04-28 00:00:00.000
4        member    Thursday               4  2022-04-28 00:00:00.000

                start_hour                  end_date                end_hour  \
0  1899-12-30 10:36:10.000  2022-04-28 00:00:00.000  1899-12-30 10:38:20.000
1  1899-12-30 13:00:52.000  2022-04-28 00:00:00.000  1899-12-30 13:03:23.000
2  1899-12-30 13:12:51.000  2022-04-28 00:00:00.000  1899-12-30 13:15:28.000
3  1899-12-30 10:49:18.000  2022-04-28 00:00:00.000  1899-12-30 10:52:00.000
4  1899-12-30 15:40:20.000  2022-04-28 00:00:00.000  1899-12-30 15:43:11.000

   trip_duration_minutes  distance_KM
0                      2          0.0
1                      2          0.0
2                      2          0.0
3                      2          0.0
4                      2          0.0
```

[6]:
```
df.tail()
```

[6]:
```
         rideable_type              started_at                ended_at  \
5667712  classic_bike  2022-11-24 18:02:15.000  2022-11-24 18:07:50.000
5667713  classic_bike  2022-11-24 17:59:55.000  2022-11-24 18:05:01.000
5667714  classic_bike  2022-11-24 15:40:51.000  2022-11-24 15:46:33.000
5667715  classic_bike  2022-11-24 17:02:01.000  2022-11-24 17:07:41.000
5667716  classic_bike  2022-11-24 14:36:09.000  2022-11-24 14:41:34.000

                           start_station_name             end_station_name  \
5667712  Damen Ave & Thomas St (Augusta Blvd)     Leavitt St & Chicago Ave
5667713           Cottage Grove Ave & 47th St  Cottage Grove Ave & 43rd St
```

```
5667714            Larrabee St & Kingsbury St      Larrabee St & North Ave
5667715                 Shields Ave & 31st St           State St & 33rd St
5667716                 Kimbark Ave & 53rd St          Ellis Ave & 53rd St

         start_lat  start_lng    end_lat    end_lng member_casual day_of_week  \
5667712  41.901315 -87.677409  41.895501 -87.682017        member    Thursday
5667713  41.809855 -87.606755  41.816499 -87.606582        member    Thursday
5667714  41.897764 -87.642884  41.910210 -87.643500        member    Thursday
5667715  41.838464 -87.635406  41.834734 -87.625813        member    Thursday
5667716  41.799568 -87.594747  41.799336 -87.600958        member    Thursday

         day_of_week_no               start_date               start_hour  \
5667712               4  2022-11-24 00:00:00.000  1899-12-30 18:02:15.000
5667713               4  2022-11-24 00:00:00.000  1899-12-30 17:59:55.000
5667714               4  2022-11-24 00:00:00.000  1899-12-30 15:40:51.000
5667715               4  2022-11-24 00:00:00.000  1899-12-30 17:02:01.000
5667716               4  2022-11-24 00:00:00.000  1899-12-30 14:36:09.000

                       end_date                 end_hour  \
5667712  2022-11-24 00:00:00.000  1899-12-30 18:07:50.000
5667713  2022-11-24 00:00:00.000  1899-12-30 18:05:01.000
5667714  2022-11-24 00:00:00.000  1899-12-30 15:46:33.000
5667715  2022-11-24 00:00:00.000  1899-12-30 17:07:41.000
5667716  2022-11-24 00:00:00.000  1899-12-30 14:41:34.000

         trip_duration_minutes  distance_KM
5667712                      5     0.750600
5667713                      5     0.738918
5667714                      5     1.384871
5667715                      5     0.896457
5667716                      5     0.515508
```

[7]:
```
df.columns
```

[7]:
```
Index(['rideable_type', 'started_at', 'ended_at', 'start_station_name',
       'end_station_name', 'start_lat', 'start_lng', 'end_lat', 'end_lng',
       'member_casual', 'day_of_week', 'day_of_week_no', 'start_date',
       'start_hour', 'end_date', 'end_hour', 'trip_duration_minutes',
       'distance_KM'],
      dtype='object')
```

[8]:
```
df.dtypes
```

[8]:
```
rideable_type            object
started_at               object
ended_at                 object
start_station_name       object
```

```
end_station_name          object
start_lat                float64
start_lng                float64
end_lat                  float64
end_lng                  float64
member_casual             object
day_of_week               object
day_of_week_no             int64
start_date                object
start_hour                object
end_date                  object
end_hour                  object
trip_duration_minutes      int64
distance_KM              float64
dtype: object
```

[9]:
```
df.describe()
```

[9]:
```
          start_lat     start_lng       end_lat       end_lng  day_of_week_no  \
count  5.667717e+06  5.667717e+06  5.661859e+06  5.661859e+06    5.667717e+06
mean   4.190222e+01 -8.764783e+01  4.190242e+01 -8.764790e+01    3.102581e+00
std    4.626109e-02  2.999925e-02  6.805821e-02  1.082985e-01    2.014908e+00
min    4.164000e+01 -8.784000e+01  0.000000e+00 -8.814000e+01    0.000000e+00
25%    4.188103e+01 -8.766154e+01  4.188103e+01 -8.766260e+01    1.000000e+00
50%    4.190000e+01 -8.764410e+01  4.190000e+01 -8.764414e+01    3.000000e+00
75%    4.193000e+01 -8.762957e+01  4.193000e+01 -8.762963e+01    5.000000e+00
max    4.563503e+01 -7.379648e+01  4.237000e+01  0.000000e+00    6.000000e+00

       trip_duration_minutes   distance_KM
count           5.667717e+06  5.657815e+06
mean            1.301255e+01  2.140977e+00
std             1.110012e+01  1.183433e+01
min            -5.700000e+01  0.000000e+00
25%             5.000000e+00  8.738332e-01
50%             1.000000e+01  1.576027e+00
75%             1.700000e+01  2.780769e+00
max             5.900000e+01  9.814069e+03
```

5 Step 2: Data Preparation

- Dropping irrelevant columns and rows
- Identifying duplicated columns
- Renaming columns
- Feature creation

[10]:
```
df.shape
```

[10]: (5667717, 18)

[11]:
```
df.dtypes
```

[11]:
```
rideable_type              object
started_at                 object
ended_at                   object
start_station_name         object
end_station_name           object
start_lat                 float64
start_lng                 float64
end_lat                   float64
end_lng                   float64
member_casual              object
day_of_week                object
day_of_week_no              int64
start_date                 object
start_hour                 object
end_date                   object
end_hour                   object
trip_duration_minutes       int64
distance_KM               float64
dtype: object
```

[12]:
```
# Change the data type of started_at to datetime data type

df['started_at'] = pd.to_datetime(df['started_at'])
```

[13]:
```
# Change the data type of ended_at to datetime data type

df['ended_at'] = pd.to_datetime(df['ended_at'])
```

[14]:
```
# Change the data type of start_date to date time data type

df['start_date'] = pd.to_datetime(df['start_date'])
```

[15]:
```
# Change the data type of end_date to date time data type

df['end_date'] = pd.to_datetime(df['end_date'])
```

[16]:
```
# Change the data type of start_hour to date time data type

df['start_hour'] = pd.to_datetime(df['start_hour'])
```

[17]:
```
# Change the data type of end_hour to date time data type

df['end_hour'] = pd.to_datetime(df['end_hour'])
```

[18]:
```
df.dtypes
```

[18]:
```
rideable_type                 object
started_at                datetime64[ns]
ended_at                  datetime64[ns]
start_station_name            object
end_station_name              object
start_lat                    float64
start_lng                    float64
end_lat                      float64
end_lng                      float64
member_casual                 object
day_of_week                   object
day_of_week_no                 int64
start_date                datetime64[ns]
start_hour                datetime64[ns]
end_date                  datetime64[ns]
end_hour                  datetime64[ns]
trip_duration_minutes          int64
distance_KM                  float64
dtype: object
```

[19]:
```
df.head()
```

[19]:

	rideable_type	started_at	ended_at	start_station_name	\
0	electric_bike	2022-04-28 10:36:10	2022-04-28 10:38:20	NaN	
1	electric_bike	2022-04-28 13:00:52	2022-04-28 13:03:23	NaN	
2	electric_bike	2022-04-28 13:12:51	2022-04-28 13:15:28	NaN	
3	electric_bike	2022-04-28 10:49:18	2022-04-28 10:52:00	NaN	
4	electric_bike	2022-04-28 15:40:20	2022-04-28 15:43:11	NaN	

	end_station_name	start_lat	start_lng	end_lat	end_lng	member_casual	\
0	NaN	41.79	-87.6	41.79	-87.6	member	
1	NaN	41.79	-87.6	41.79	-87.6	member	
2	NaN	41.79	-87.6	41.79	-87.6	member	
3	NaN	41.79	-87.6	41.79	-87.6	member	
4	NaN	41.79	-87.6	41.79	-87.6	member	

	day_of_week	day_of_week_no	start_date	start_hour	end_date	\
0	Thursday	4	2022-04-28	1899-12-30 10:36:10	2022-04-28	
1	Thursday	4	2022-04-28	1899-12-30 13:00:52	2022-04-28	
2	Thursday	4	2022-04-28	1899-12-30 13:12:51	2022-04-28	
3	Thursday	4	2022-04-28	1899-12-30 10:49:18	2022-04-28	
4	Thursday	4	2022-04-28	1899-12-30 15:40:20	2022-04-28	

	end_hour	trip_duration_minutes	distance_KM
0	1899-12-30 10:38:20	2	0.0

```
 1 1899-12-30 13:03:23                    2          0.0
 2 1899-12-30 13:15:28                    2          0.0
 3 1899-12-30 10:52:00                    2          0.0
 4 1899-12-30 15:43:11                    2          0.0
```

[20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5667717 entries, 0 to 5667716
Data columns (total 18 columns):
 #   Column                 Dtype
---  ------                 -----
 0   rideable_type          object
 1   started_at             datetime64[ns]
 2   ended_at               datetime64[ns]
 3   start_station_name     object
 4   end_station_name       object
 5   start_lat              float64
 6   start_lng              float64
 7   end_lat                float64
 8   end_lng                float64
 9   member_casual          object
 10  day_of_week            object
 11  day_of_week_no         int64
 12  start_date             datetime64[ns]
 13  start_hour             datetime64[ns]
 14  end_date               datetime64[ns]
 15  end_hour               datetime64[ns]
 16  trip_duration_minutes  int64
 17  distance_KM            float64
dtypes: datetime64[ns](6), float64(5), int64(2), object(5)
memory usage: 778.3+ MB
```

[21]:
```
# Check for missing values

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

[21]:
```
rideable_type               0
started_at                  0
ended_at                    0
start_station_name     833064
end_station_name       892742
start_lat                   0
start_lng                   0
end_lat                  5858
end_lng                  5858
member_casual               0
day_of_week                 0
```

```
day_of_week_no                0
start_date                    0
start_hour                    0
end_date                      0
end_hour                      0
trip_duration_minutes         0
distance_KM                9902
dtype: int64
```

[22]:
```
# Check for any duplicated data

df.loc[df.duplicated()]
```

[22]:
```
         rideable_type          started_at            ended_at  \
52092    electric_bike 2022-04-20 17:24:16 2022-04-20 17:42:06
197564     docked_bike 2022-02-22 19:55:20 2022-02-22 20:14:29
199666     docked_bike 2022-02-16 15:11:15 2022-02-16 15:39:45
621578    classic_bike 2022-03-05 15:27:23 2022-03-05 15:39:15
830939    classic_bike 2022-03-20 12:39:34 2022-03-20 12:48:46
981771    classic_bike 2022-01-13 06:50:42 2022-01-13 06:56:05
1622834   classic_bike 2022-05-30 16:15:35 2022-05-30 16:27:35
1716019   classic_bike 2022-06-18 13:18:24 2022-06-18 13:57:26
1785982   classic_bike 2022-05-30 16:14:38 2022-05-30 16:39:15
2719650   classic_bike 2022-07-06 20:10:39 2022-07-06 21:43:13
2740769   classic_bike 2022-07-21 18:07:58 2022-07-21 18:22:25
2762496   classic_bike 2022-07-08 10:39:55 2022-07-08 11:02:05
2782184   classic_bike 2022-07-31 13:14:49 2022-07-31 13:23:17
2851830   classic_bike 2022-07-09 16:26:59 2022-07-09 17:13:09
3084907   classic_bike 2022-07-23 23:58:28 2022-07-24 00:06:34
3144077   classic_bike 2022-07-10 13:15:25 2022-07-10 13:15:52
3305362    docked_bike 2022-08-12 22:24:52 2022-08-12 22:59:31
3898200   classic_bike 2022-09-01 18:12:58 2022-09-01 18:26:59
4049726   classic_bike 2022-08-05 21:54:25 2022-08-05 22:11:13
4142375   classic_bike 2022-08-17 19:55:36 2022-08-17 20:15:15
4165370   classic_bike 2022-09-28 16:57:11 2022-09-28 17:17:56
4376253    docked_bike 2022-10-07 19:08:10 2022-10-07 19:19:52
4969900   classic_bike 2022-09-02 18:23:01 2022-09-02 18:25:40
5241784    docked_bike 2022-09-10 01:12:02 2022-09-15 04:55:01

                            start_station_name  \
52092                                      NaN
197564                Streeter Dr & Grand Ave
199666           California Ave & Division St
621578                   Federal St & Polk St
830939            Sheffield Ave & Webster Ave
981771               Loomis St & Lexington St
1622834         Lakeview Ave & Fullerton Pkwy
```

```
1716019                      Adler Planetarium
1785982                  Clark St & Berwyn Ave
2719650              Bissell St & Armitage Ave
2740769            Southport Ave & Belmont Ave
2762496               Halsted St & Dickens Ave
2782184             Damen Ave & Wellington Ave
2851830                        Millennium Park
3084907                 Halsted St & Roscoe St
3144077                Green St & Randolph St*
3305362               Wells St & Evergreen Ave
3898200             Racine Ave & Fullerton Ave
4049726         California Ave & Milwaukee Ave
4142375                     Theater on the Lake
4165370  DuSable Lake Shore Dr & Diversey Pkwy
4376253                 Aberdeen St & Monroe St
4969900             Desplaines St & Randolph St
5241784             Prairie Ave & Garfield Blvd

                          end_station_name  start_lat  start_lng    end_lat  \
52092                                  NaN  41.950000 -87.740000  41.930000
197564   Mies van der Rohe Way & Chicago Ave  41.892278 -87.612043  41.896945
199666          California Ave & Division St  41.903029 -87.697474  41.903029
621578                Michigan Ave & 18th St  41.872078 -87.629544  41.857813
830939          Sheffield Ave & Kingsbury St  41.921540 -87.653818  41.910522
981771             Halsted St & Roosevelt Rd  41.872187 -87.661501  41.867324
1622834          Bissell St & Armitage Ave*  41.925858 -87.638973  41.918296
1716019                McClurg Ct & Ohio St  41.866095 -87.607267  41.892592
1785982            Lincoln Ave & Roscoe St*  41.978031 -87.668565  41.943350
2719650          Bissell St & Armitage Ave*  41.918018 -87.652182  41.918296
2740769          Bissell St & Armitage Ave*  41.939478 -87.663748  41.918296
2762496          Bissell St & Armitage Ave*  41.919936 -87.648830  41.918296
2782184            Lincoln Ave & Roscoe St*  41.935880 -87.678420  41.943350
2851830          Bissell St & Armitage Ave*  41.881032 -87.624084  41.918296
3084907            Lincoln Ave & Roscoe St*  41.943670 -87.648950  41.943350
3144077              Green St & Randolph St*  41.883602 -87.648627  41.883602
3305362              Green St & Randolph St*  41.906724 -87.634830  41.883602
3898200            Lincoln Ave & Roscoe St*  41.925563 -87.658404  41.943350
4049726          Bissell St & Armitage Ave*  41.922695 -87.697153  41.918296
4142375            Lincoln Ave & Roscoe St*  41.926277 -87.630834  41.943350
4165370          Bissell St & Armitage Ave*  41.932588 -87.636427  41.918296
4376253             Green St & Randolph St*  41.880419 -87.655519  41.883602
4969900             Green St & Randolph St*  41.884616 -87.644571  41.883602
5241784                                  NaN  41.794853 -87.618691        NaN

           end_lng member_casual day_of_week  day_of_week_no start_date  \
52092   -87.770000        member   Wednesday               3 2022-04-20
197564  -87.621758        casual     Tuesday               2 2022-02-22
```

```
199666  -87.697474        casual   Wednesday              3 2022-02-16
621578  -87.624550        casual    Saturday              6 2022-03-05
830939  -87.653106        member      Sunday              0 2022-03-20
981771  -87.648625        member    Thursday              4 2022-01-13
1622834 -87.652183        casual      Monday              1 2022-05-30
1716019 -87.617289        casual    Saturday              6 2022-06-18
1785982 -87.670668        member      Monday              1 2022-05-30
2719650 -87.652183        casual   Wednesday              3 2022-07-06
2740769 -87.652183        casual    Thursday              4 2022-07-21
2762496 -87.652183        casual      Friday              5 2022-07-08
2782184 -87.670668        casual      Sunday              0 2022-07-31
2851830 -87.652183        casual    Saturday              6 2022-07-09
3084907 -87.670668        member    Saturday              6 2022-07-23
3144077 -87.648627        member      Sunday              0 2022-07-10
3305362 -87.648627        casual      Friday              5 2022-08-12
3898200 -87.670668        member    Thursday              4 2022-09-01
4049726 -87.652183        member      Friday              5 2022-08-05
4142375 -87.670668        member   Wednesday              3 2022-08-17
4165370 -87.652183        member   Wednesday              3 2022-09-28
4376253 -87.648627        casual      Friday              5 2022-10-07
4969900 -87.648627        member      Friday              5 2022-09-02
5241784        NaN        casual    Saturday              6 2022-09-10

                start_hour   end_date            end_hour  \
52092   1899-12-30 17:24:16 2022-04-20 1899-12-30 17:42:06
197564  1899-12-30 19:55:20 2022-02-22 1899-12-30 20:14:29
199666  1899-12-30 15:11:15 2022-02-16 1899-12-30 15:39:45
621578  1899-12-30 15:27:23 2022-03-05 1899-12-30 15:39:15
830939  1899-12-30 12:39:34 2022-03-20 1899-12-30 12:48:46
981771  1899-12-30 06:50:42 2022-01-13 1899-12-30 06:56:05
1622834 1899-12-30 16:15:35 2022-05-30 1899-12-30 16:27:35
1716019 1899-12-30 13:18:24 2022-06-18 1899-12-30 13:57:26
1785982 1899-12-30 16:14:38 2022-05-30 1899-12-30 16:39:15
2719650 1899-12-30 20:10:39 2022-07-06 1899-12-30 21:43:13
2740769 1899-12-30 18:07:58 2022-07-21 1899-12-30 18:22:25
2762496 1899-12-30 10:39:55 2022-07-08 1899-12-30 11:02:05
2782184 1899-12-30 13:14:49 2022-07-31 1899-12-30 13:23:17
2851830 1899-12-30 16:26:59 2022-07-09 1899-12-30 17:13:09
3084907 1899-12-30 23:58:28 2022-07-24 1899-12-30 00:06:34
3144077 1899-12-30 13:15:25 2022-07-10 1899-12-30 13:15:52
3305362 1899-12-30 22:24:52 2022-08-12 1899-12-30 22:59:31
3898200 1899-12-30 18:12:58 2022-09-01 1899-12-30 18:26:59
4049726 1899-12-30 21:54:25 2022-08-05 1899-12-30 22:11:13
4142375 1899-12-30 19:55:36 2022-08-17 1899-12-30 20:15:15
4165370 1899-12-30 16:57:11 2022-09-28 1899-12-30 17:17:56
4376253 1899-12-30 19:08:10 2022-10-07 1899-12-30 19:19:52
4969900 1899-12-30 18:23:01 2022-09-02 1899-12-30 18:25:40
```

```
5241784 1899-12-30 01:12:02 2022-09-15 1899-12-30 04:55:01

         trip_duration_minutes  distance_KM
52092                       17     3.332093
197564                      19     0.957032
199666                      28     0.000000
621578                      11     1.639175
830939                       9     1.226561
981771                       5     1.195457
1622834                     12     1.379012
1716019                     39     3.060950
1785982                     24     3.860200
2719650                     32     0.030848
2740769                     14     2.542225
2762496                     22     0.332041
2782184                      8     1.049347
2851830                     46     4.751567
3084907                      8     1.796569
3144077                      0     0.000000
3305362                     34     2.813255
3898200                     14     2.222867
4049726                     16     3.752683
4142375                     19     3.802776
4165370                     20     2.055456
4376253                     11     0.671429
4969900                      2     0.354227
5241784                     42          NaN
```

[23]:
```
df.duplicated()
```

[23]:
```
0          False
1          False
2          False
3          False
4          False
           …
5667712    False
5667713    False
5667714    False
5667715    False
5667716    False
Length: 5667717, dtype: bool
```

[24]:
```
# Removing any missing values

df = df.dropna()
```

[25]:
```
# Check for missing values

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

[25]:
```
rideable_type            0
started_at               0
ended_at                 0
start_station_name       0
end_station_name         0
start_lat                0
start_lng                0
end_lat                  0
end_lng                  0
member_casual            0
day_of_week              0
day_of_week_no           0
start_date               0
start_hour               0
end_date                 0
end_hour                 0
trip_duration_minutes    0
distance_KM              0
dtype: int64
```

[26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4365316 entries, 172090 to 5667716
Data columns (total 18 columns):
 #   Column                Dtype
---  ------                -----
 0   rideable_type         object
 1   started_at            datetime64[ns]
 2   ended_at              datetime64[ns]
 3   start_station_name    object
 4   end_station_name      object
 5   start_lat             float64
 6   start_lng             float64
 7   end_lat               float64
 8   end_lng               float64
 9   member_casual         object
 10  day_of_week           object
 11  day_of_week_no        int64
 12  start_date            datetime64[ns]
 13  start_hour            datetime64[ns]
 14  end_date              datetime64[ns]
 15  end_hour              datetime64[ns]
 16  trip_duration_minutes  int64
```

```
 17  distance_KM            float64
dtypes: datetime64[ns](6), float64(5), int64(2), object(5)
memory usage: 632.8+ MB
```

[27]:
```
df.duplicated()
```

[27]:
```
172090     False
172091     False
172092     False
172093     False
172094     False
           …
5667712    False
5667713    False
5667714    False
5667715    False
5667716    False
Length: 4365316, dtype: bool
```

6 Step 3: Feature Understanding

(Univariate Analysis)

- Plot Feature Distribution
  - Histogram
  - KDE
  - Boxplot

[28]:
```
# Check the value count of member_casual

df['member_casual'].value_counts()
```

[28]:
```
member    2610129
casual    1755187
Name: member_casual, dtype: int64
```

[29]:
```
# Check the value count of rideable_type

df['rideable_type'].value_counts()
```

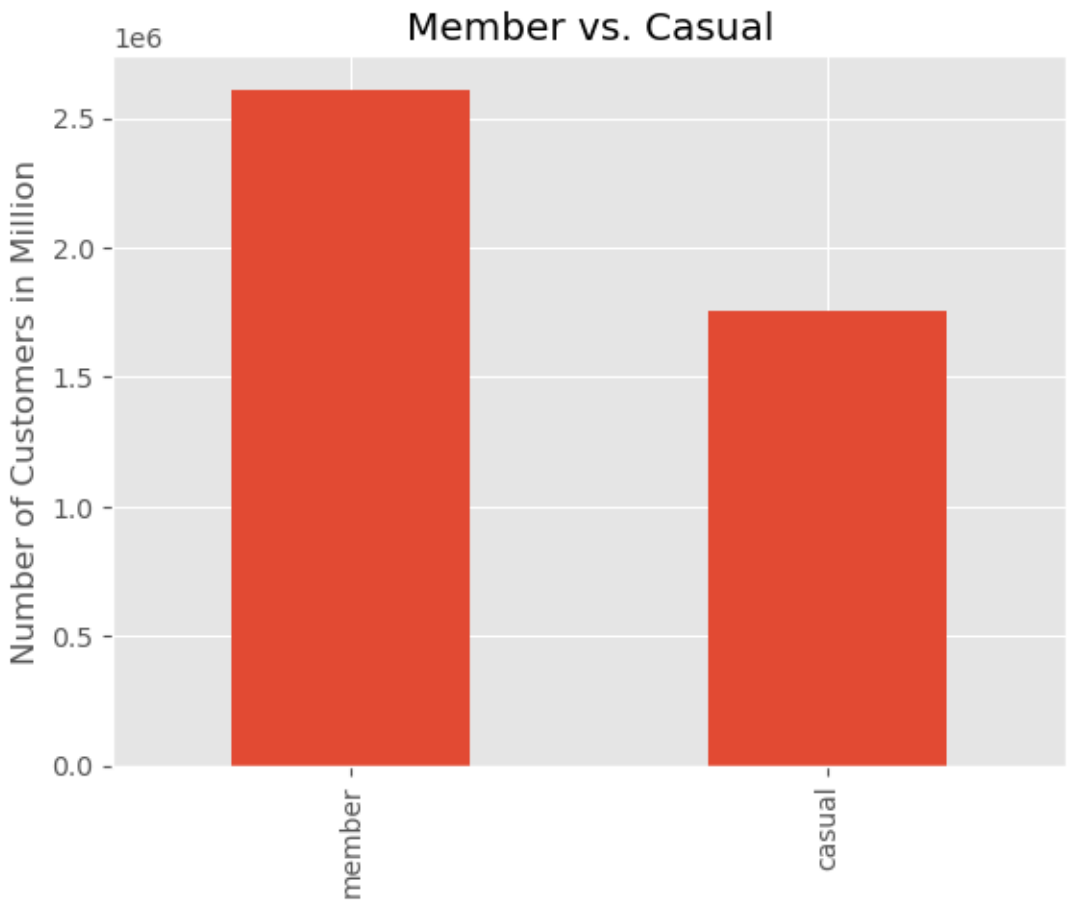
[29]:
```
classic_bike     2593745
electric_bike    1597076
docked_bike       174495
Name: rideable_type, dtype: int64
```

[30]:
```
# Check the value count of day_of_week

df['day_of_week'].value_counts()
```

[30]:
```
Saturday     704770
Thursday     645412
Wednesday    615989
Friday       608257
Tuesday      607227
Sunday       598266
Monday       585395
Name: day_of_week, dtype: int64
```

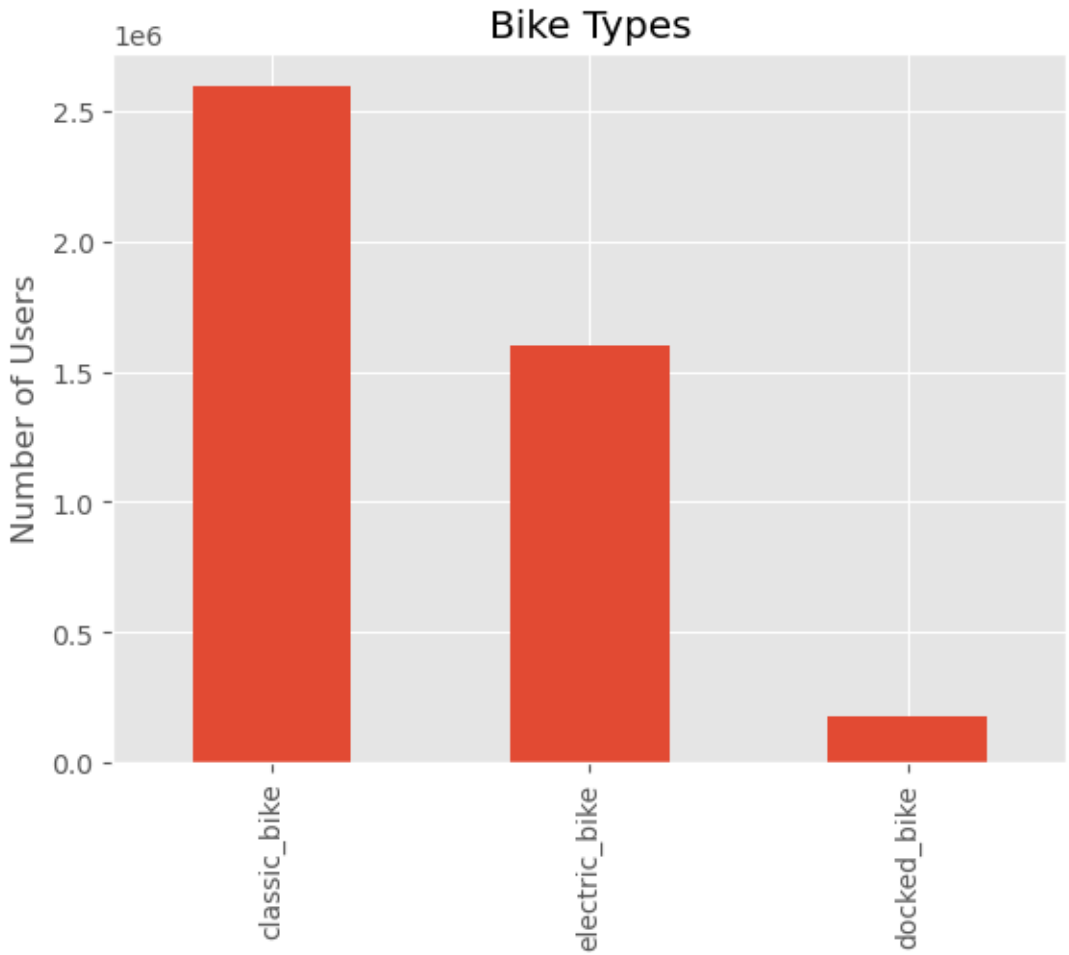
[31]:
```
# Creating bar graph for data member_casual

ax = df['member_casual'].value_counts() \
    .head() \
    .plot(kind = 'bar', title = 'Member vs. Casual')

ax.set_ylabel('Number of Customers in Million')
```

[31]:
```
Text(0, 0.5, 'Number of Customers in Million')
```



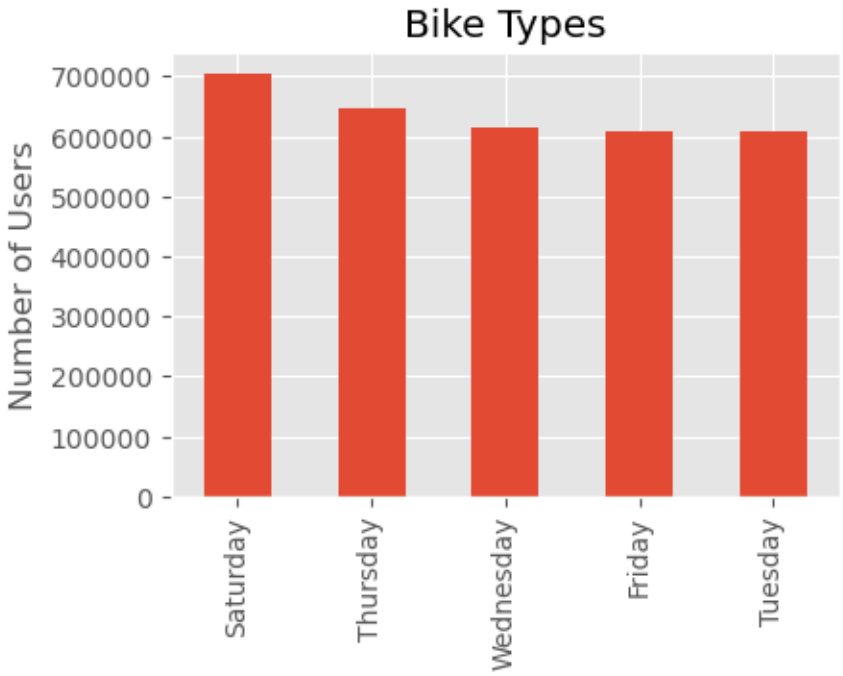
[32]:
```
# Creating bar graph for data rideable_type

ax = df['rideable_type'].value_counts() \
    .head() \
    .plot(kind = 'bar', title = 'Bike Types')

ax.set_ylabel('Number of Users')
```

[32]:
```
Text(0, 0.5, 'Number of Users')
```



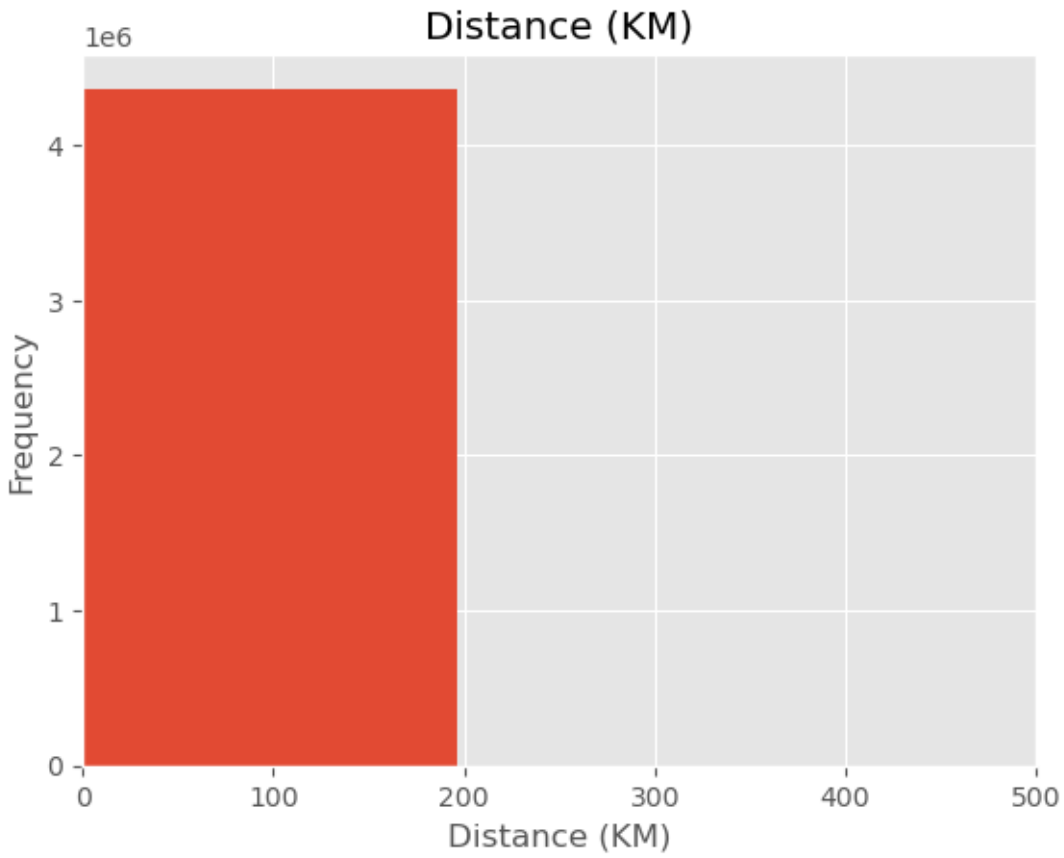
[57]:
```
# Creating bar graph for day_of_week

fig, ax = plt.subplots(figsize=(4.5, 3))

ax = df['day_of_week'].value_counts() \
    .head() \
    .plot(kind = 'bar', ax=ax, title = 'Bike Types')
```

```
ax.set_ylabel('Number of Users')

plt.show()
```

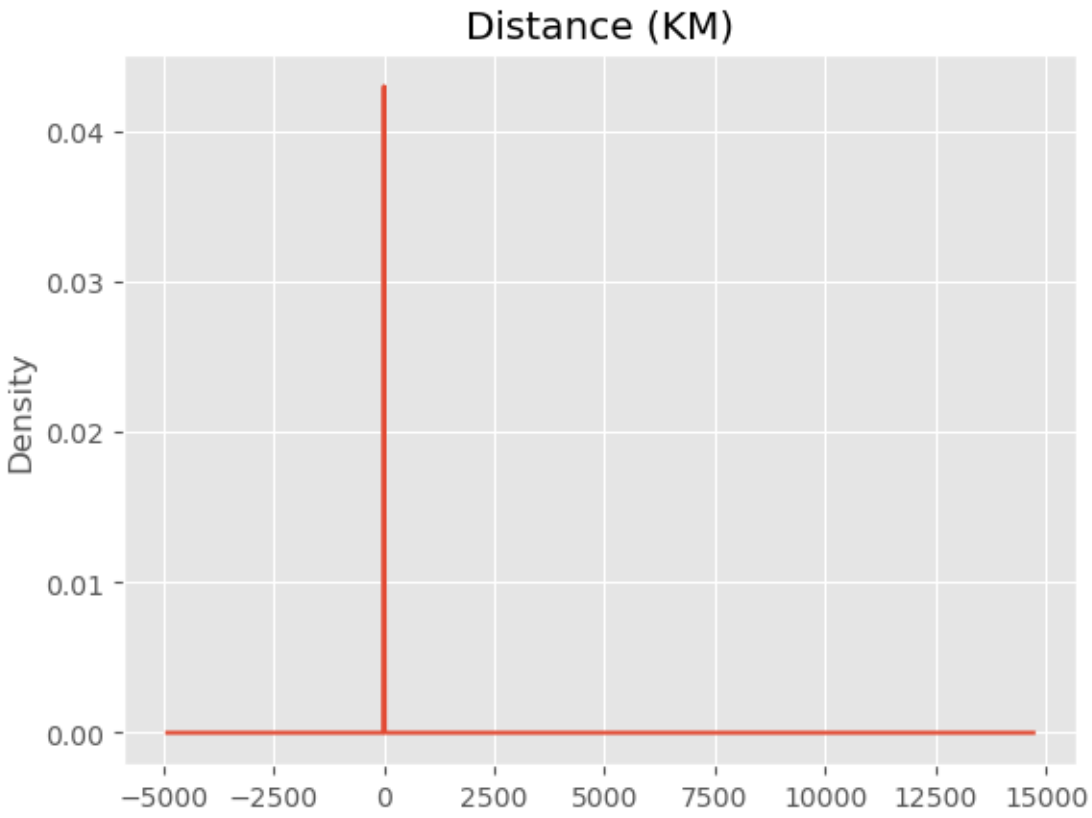


[34]:
```
# Creating histogram for distance_KM

ax = df['distance_KM'].plot(kind='hist',
                        bins=50,
                        title='Distance (KM)')

ax.set_xlabel('Distance (KM)')
ax.set_xlim(0, 500)
```

[34]: (0.0, 500.0)



[35]:
```
# Creating KDE for distance_KM

ax = df['distance_KM'].plot(kind='kde',
                            title='Distance (KM)')
```



7 Step 4: Feature Relationships

- Scatterplot
- Heatmap Correlation
- Pairplot
- Groupby comparisons

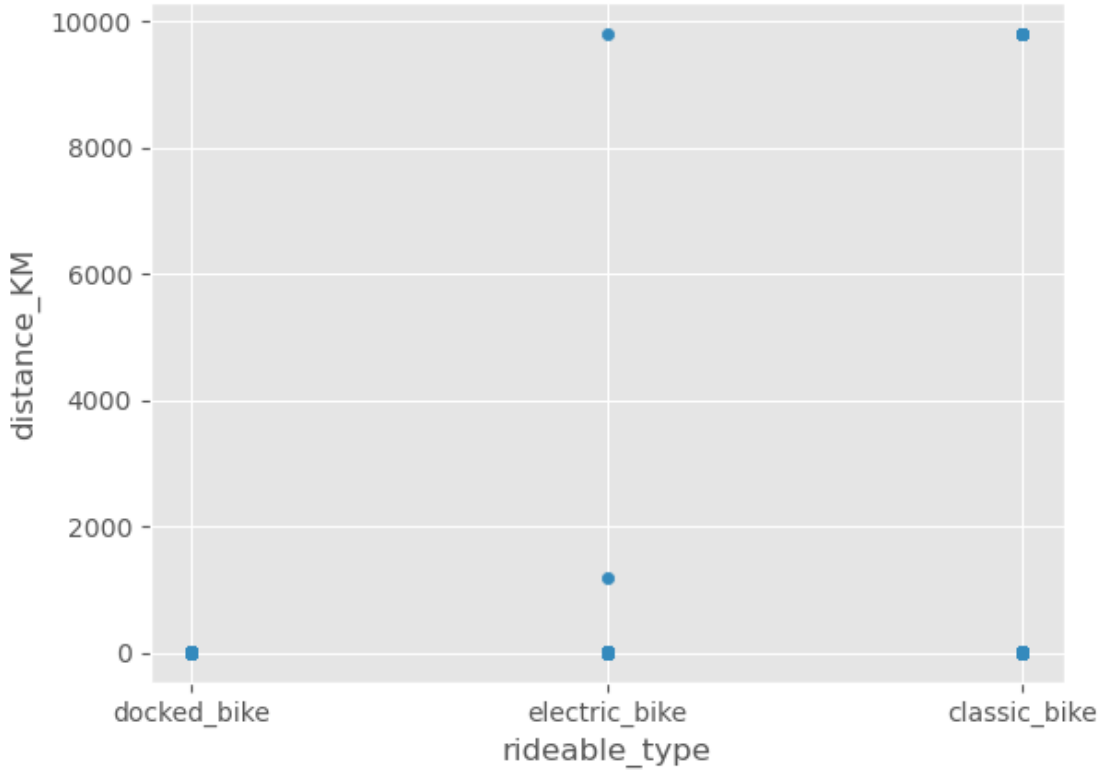
[36]:
```
df
```

[36]:
```
         rideable_type          started_at            ended_at  \
172090     docked_bike 2022-01-19 02:58:15 2022-01-19 03:19:14
172091     docked_bike 2022-01-19 14:40:18 2022-01-19 14:50:22
172092     docked_bike 2022-01-05 00:22:40 2022-01-05 00:45:47
172093     docked_bike 2022-01-26 13:23:58 2022-01-26 15:02:49
172094     docked_bike 2022-01-05 02:34:50 2022-01-05 02:38:57
…                  …                   …                   …
5667712  classic_bike 2022-11-24 18:02:15 2022-11-24 18:07:50
5667713  classic_bike 2022-11-24 17:59:55 2022-11-24 18:05:01
5667714  classic_bike 2022-11-24 15:40:51 2022-11-24 15:46:33
5667715  classic_bike 2022-11-24 17:02:01 2022-11-24 17:07:41
```

```
5667716  classic_bike 2022-11-24 14:36:09 2022-11-24 14:41:34

                          start_station_name             end_station_name  \
172090              Clinton St & Roosevelt Rd         Halsted St & 35th St
172091              Sedgwick St & Webster Ave            Clark St & Elm St
172092              Ashland Ave & Division St            Clark St & Elm St
172093              Halsted St & Roosevelt Rd   Wood St & Taylor St (Temp)
172094                  Broadway & Berwyn Ave         Broadway & Argyle St
…                                         …                            …
5667712  Damen Ave & Thomas St (Augusta Blvd)     Leavitt St & Chicago Ave
5667713           Cottage Grove Ave & 47th St  Cottage Grove Ave & 43rd St
5667714            Larrabee St & Kingsbury St      Larrabee St & North Ave
5667715                 Shields Ave & 31st St           State St & 33rd St
5667716                 Kimbark Ave & 53rd St          Ellis Ave & 53rd St

         start_lat  start_lng    end_lat    end_lng member_casual day_of_week  \
172090   41.867118 -87.641088  41.830661 -87.647172        casual   Wednesday
172091   41.922167 -87.638888  41.902973 -87.631280        casual   Wednesday
172092   41.903450 -87.667747  41.902973 -87.631280        casual   Wednesday
172093   41.867324 -87.648625  41.869265 -87.673731        casual   Wednesday
172094   41.978353 -87.659753  41.973815 -87.659660        casual   Wednesday
…              …         …         …         …           …          …
5667712  41.901315 -87.677409  41.895501 -87.682017        member    Thursday
5667713  41.809855 -87.606755  41.816499 -87.606582        member    Thursday
5667714  41.897764 -87.642884  41.910210 -87.643500        member    Thursday
5667715  41.838464 -87.635406  41.834734 -87.625813        member    Thursday
5667716  41.799568 -87.594747  41.799336 -87.600958        member    Thursday

         day_of_week_no start_date          start_hour   end_date  \
172090                3 2022-01-19 1899-12-30 02:58:15 2022-01-19
172091                3 2022-01-19 1899-12-30 14:40:18 2022-01-19
172092                3 2022-01-05 1899-12-30 00:22:40 2022-01-05
172093                3 2022-01-26 1899-12-30 13:23:58 2022-01-26
172094                3 2022-01-05 1899-12-30 02:34:50 2022-01-05
…                   …         …                   …          …
5667712               4 2022-11-24 1899-12-30 18:02:15 2022-11-24
5667713               4 2022-11-24 1899-12-30 17:59:55 2022-11-24
5667714               4 2022-11-24 1899-12-30 15:40:51 2022-11-24
5667715               4 2022-11-24 1899-12-30 17:02:01 2022-11-24
5667716               4 2022-11-24 1899-12-30 14:36:09 2022-11-24

                   end_hour  trip_duration_minutes  distance_KM
172090  1899-12-30 03:19:14                     20     4.085036
172091  1899-12-30 14:50:22                     10     2.225186
172092  1899-12-30 00:45:47                     23     3.018457
172093  1899-12-30 15:02:49                     38     2.090070
172094  1899-12-30 02:38:57                      4     0.504661
```

```
…                    …                    …          …
5667712 1899-12-30 18:07:50                    5     0.750600
5667713 1899-12-30 18:05:01                    5     0.738918
5667714 1899-12-30 15:46:33                    5     1.384871
5667715 1899-12-30 17:07:41                    5     0.896457
5667716 1899-12-30 14:41:34                    5     0.515508

[4365316 rows x 18 columns]
```

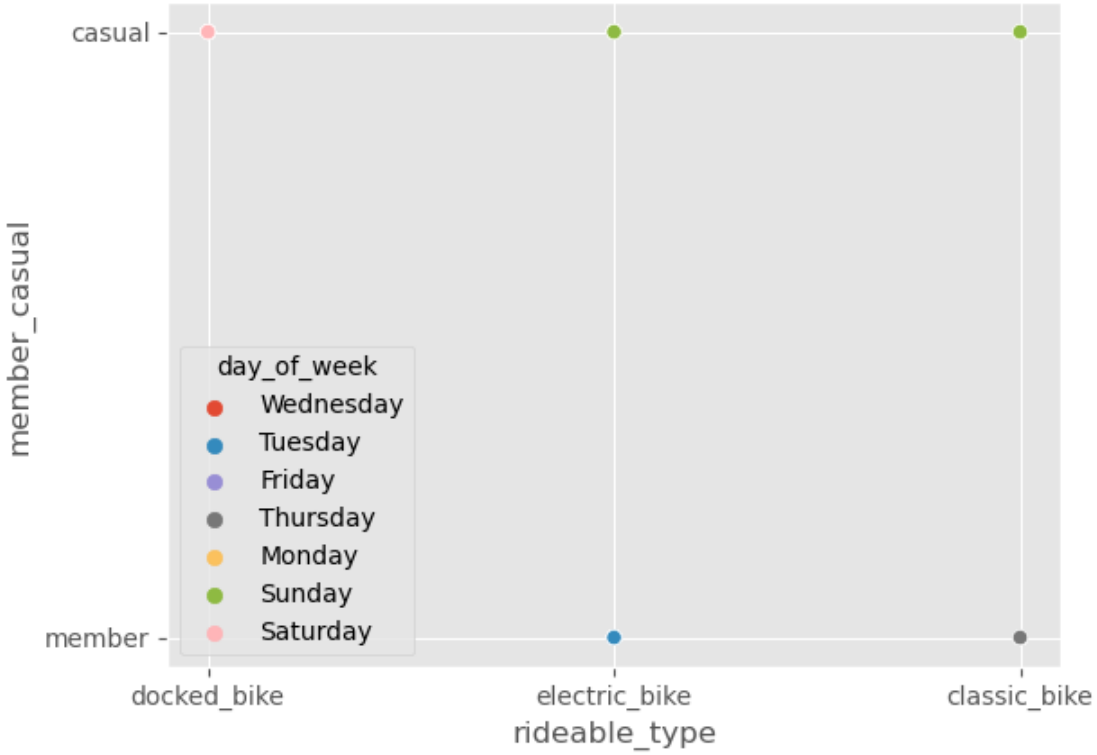
[56]:
```
# creating scatter plot

df.plot(kind='scatter',
        x='rideable_type',
       y='distance_KM')

plt.show()
```



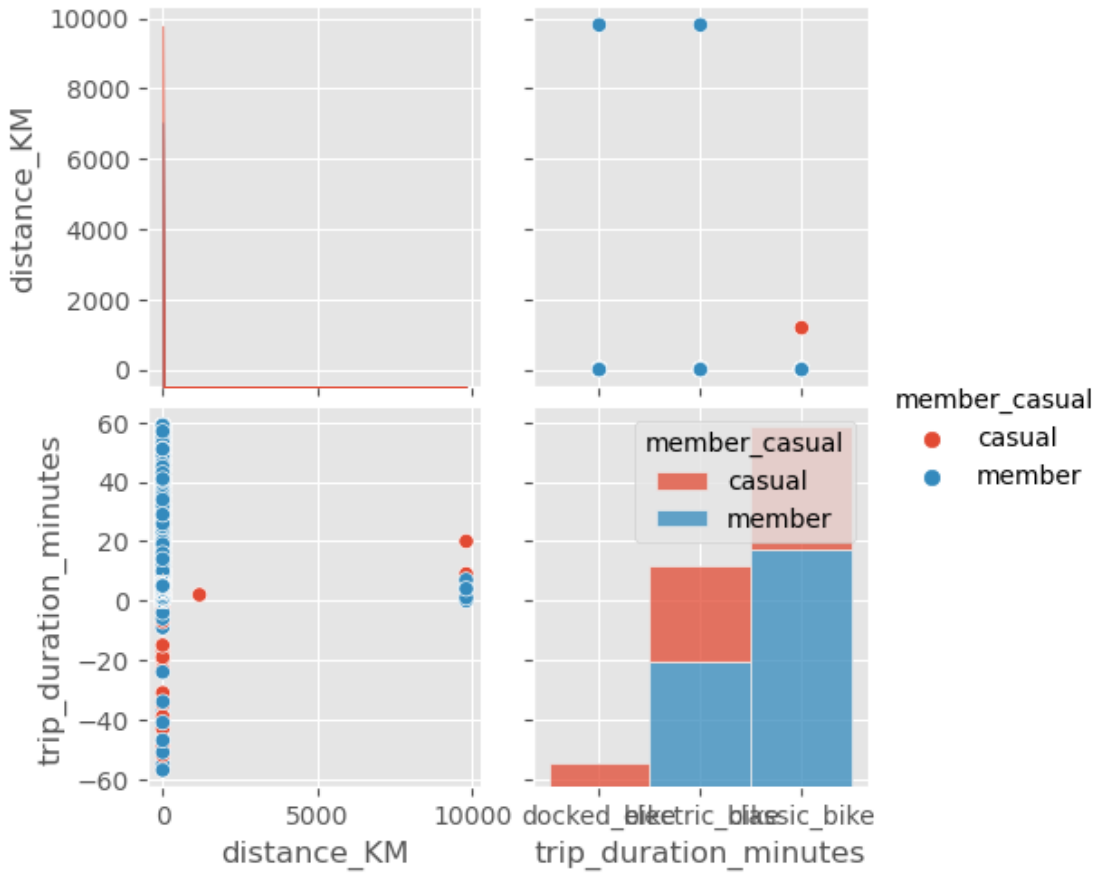
[59]:
```
# creating graphs using seaborn

sns.scatterplot(x='rideable_type',
                y='member_casual',
                hue='day_of_week',
```

```
                data= df)
```

[59]: <AxesSubplot:xlabel='rideable_type', ylabel='member_casual'>



[63]:
```
sns.pairplot(df,
             vars=['distance_KM','trip_duration_minutes'],
             hue='member_casual')

sns.histplot(df, x='rideable_type', hue='member_casual', multiple='stack')

plt.show()
```



8 Step 5: Ask questions about the data

- Try to answer a question you have about the data using a plot or statistics.
  - What is the relationship between the casual and member in terms of bike preference?

[78]:
```
df['start_station_name'].value_counts()
```

[78]:
```
Streeter Dr & Grand Ave                                  71269
DuSable Lake Shore Dr & Monroe St                        39251
DuSable Lake Shore Dr & North Blvd                       37698
Michigan Ave & Oak St                                    37208
Wells St & Concord Ln                                    34508
                                                         ...
Public Rack - Union Ave & 111th St                           1
Public Rack - Western Ave & 111th St - NW                    1
Public Rack - Michele Clark Magnet High School               1
10101 S Stony Island Ave                                     1
Public Rack - 63rd & Western Ave S                           1
Name: start_station_name, Length: 1556, dtype: int64
```

[ ]:
```
df['rideable_type'].value_counts()
```

[70]:
```
# relationship with classic_bike

df[df['rideable_type'] == 'classic_bike'] \
    .groupby('member_casual')['trip_duration_minutes'] \
    .agg(['mean', 'count'])
```

[70]:
```
                    mean    count
member_casual
casual          16.93369   886141
member          11.92073  1707604
```

[71]:
```
# relationship with electric_bike

df[df['rideable_type'] == 'electric_bike'] \
    .groupby('member_casual')['trip_duration_minutes'] \
    .agg(['mean', 'count'])
```

[71]:
```
                    mean   count
member_casual
casual          13.937384  694551
member          10.016914  902525
```

[72]:
```
# relationship with docked_bike

df[df['rideable_type'] == 'docked_bike'] \
    .groupby('member_casual')['trip_duration_minutes'] \
    .agg(['mean', 'count'])
```

[72]:
```
                    mean   count
member_casual
casual          23.943912  174495
```

[77]:
```
# relationship with member

df[df['member_casual'] == 'member'] \
    .groupby('rideable_type')['distance_KM'] \
    .agg(['mean', 'count'])
```

[77]:
```
                    mean    count
rideable_type
classic_bike    1.964682  1707604
electric_bike   2.267771   902525
```

[74]:
```
# relationship with casual

df[df['member_casual'] == 'casual'] \
```

```
    .groupby('rideable_type')['distance_KM'] \
    .agg(['mean', 'count'])
```

[74]:
```
                    mean   count
rideable_type
classic_bike    2.091261  886141
docked_bike     2.178154  174495
electric_bike   2.254534  694551
```

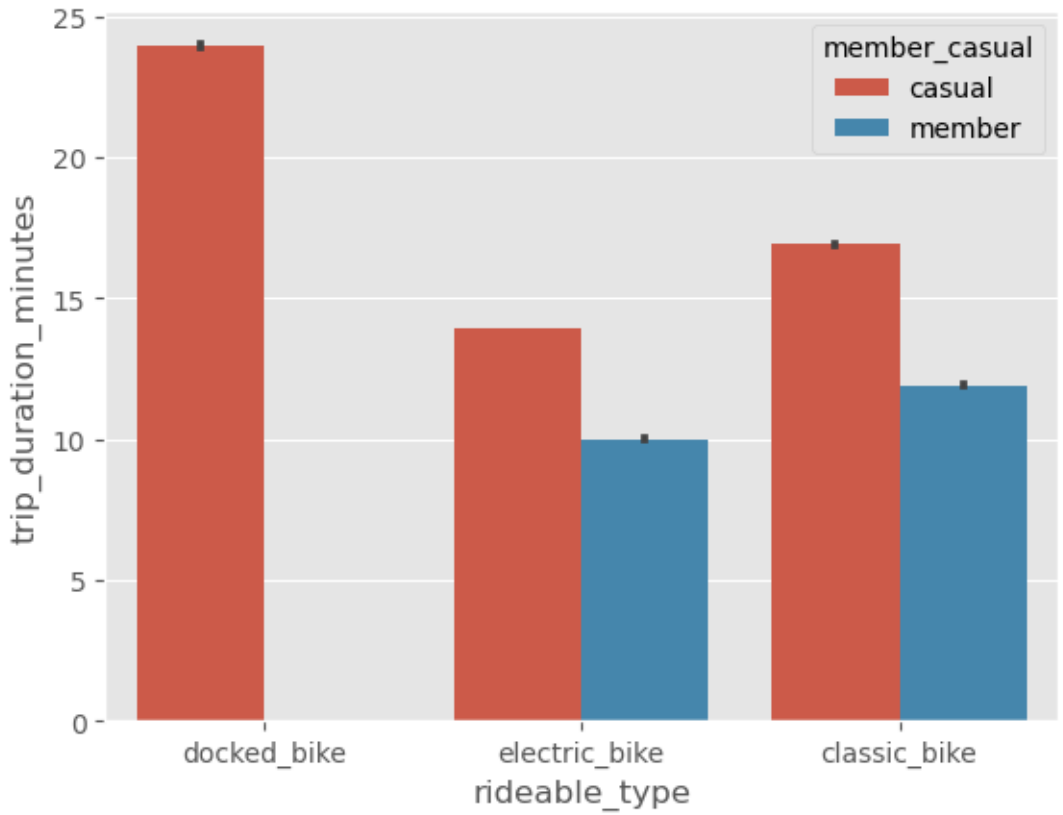
[65]:
```
df.head()
```

[65]:
```
        rideable_type          started_at            ended_at  \
172090   docked_bike 2022-01-19 02:58:15 2022-01-19 03:19:14
172091   docked_bike 2022-01-19 14:40:18 2022-01-19 14:50:22
172092   docked_bike 2022-01-05 00:22:40 2022-01-05 00:45:47
172093   docked_bike 2022-01-26 13:23:58 2022-01-26 15:02:49
172094   docked_bike 2022-01-05 02:34:50 2022-01-05 02:38:57

                start_station_name            end_station_name  start_lat  \
172090  Clinton St & Roosevelt Rd        Halsted St & 35th St  41.867118
172091  Sedgwick St & Webster Ave           Clark St & Elm St  41.922167
172092  Ashland Ave & Division St           Clark St & Elm St  41.903450
172093  Halsted St & Roosevelt Rd  Wood St & Taylor St (Temp)  41.867324
172094      Broadway & Berwyn Ave        Broadway & Argyle St  41.978353

         start_lng    end_lat    end_lng member_casual day_of_week  \
172090 -87.641088  41.830661 -87.647172        casual   Wednesday
172091 -87.638888  41.902973 -87.631280        casual   Wednesday
172092 -87.667747  41.902973 -87.631280        casual   Wednesday
172093 -87.648625  41.869265 -87.673731        casual   Wednesday
172094 -87.659753  41.973815 -87.659660        casual   Wednesday

        day_of_week_no start_date          start_hour   end_date  \
172090               3 2022-01-19 1899-12-30 02:58:15 2022-01-19
172091               3 2022-01-19 1899-12-30 14:40:18 2022-01-19
172092               3 2022-01-05 1899-12-30 00:22:40 2022-01-05
172093               3 2022-01-26 1899-12-30 13:23:58 2022-01-26
172094               3 2022-01-05 1899-12-30 02:34:50 2022-01-05

                  end_hour  trip_duration_minutes  distance_KM
172090 1899-12-30 03:19:14                     20     4.085036
172091 1899-12-30 14:50:22                     10     2.225186
172092 1899-12-30 00:45:47                     23     3.018457
172093 1899-12-30 15:02:49                     38     2.090070
172094 1899-12-30 02:38:57                      4     0.504661
```

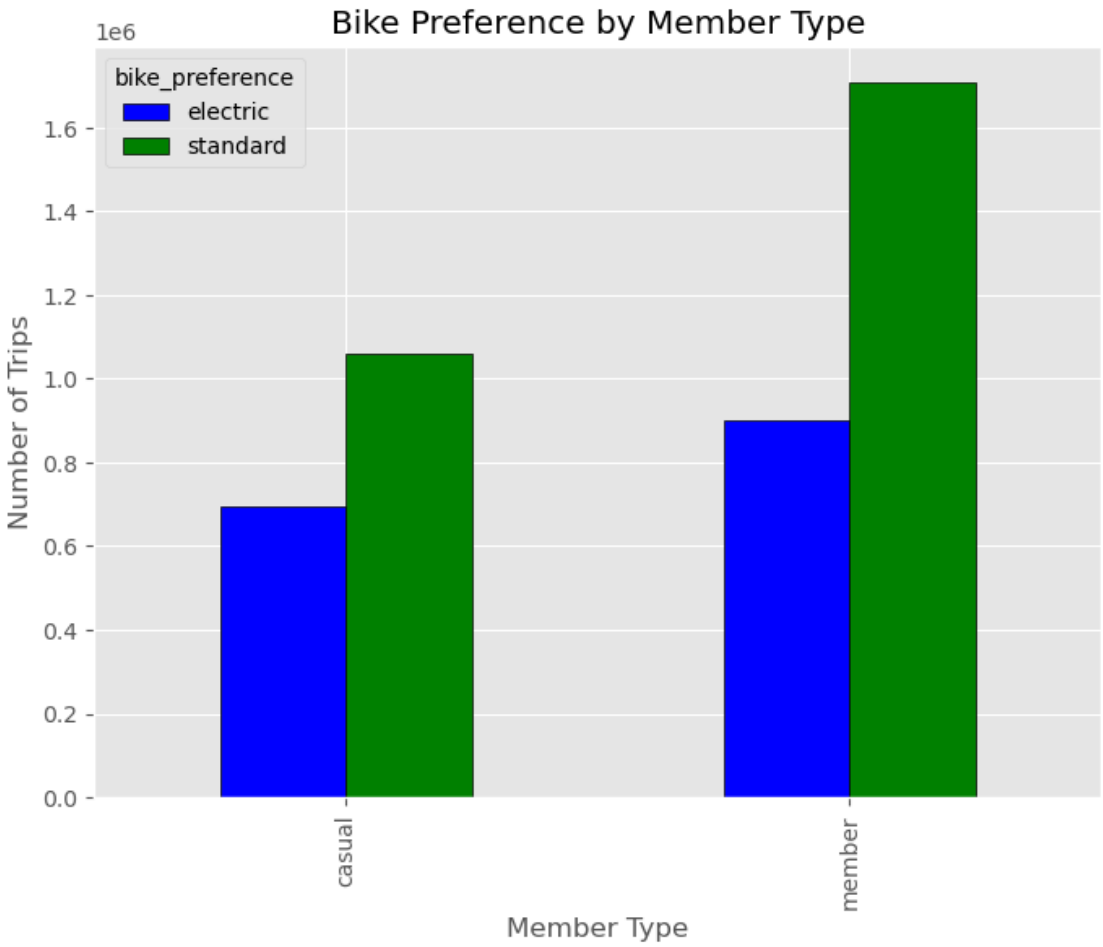
[79]:
```
# Group the data by member type and bike preference
bike_pref_counts = df.groupby(["member_casual", "rideable_type"]).size().
 ↪unstack(fill_value=0)

# Normalize the counts by the total number of trips for each member type
bike_pref_perc = bike_pref_counts.div(bike_pref_counts.sum(axis=1), axis=0)

# Print the results
print(bike_pref_perc)
```

```
rideable_type  classic_bike  docked_bike  electric_bike
member_casual
casual             0.504870     0.099417       0.395713
member             0.654222     0.000000       0.345778
```

[85]:
```
sns.barplot(data=df, x='rideable_type', y='trip_duration_minutes',␣
 ↪hue='member_casual')

plt.show()
```



[86]:
```
# create a new column for bike preference
df['bike_preference'] = df['rideable_type'].apply(lambda x: 'electric' if␣
 ↪'electric' in x else 'standard')

# group by member type and bike preference
grouped = df.groupby(['member_casual', 'bike_preference']).size().unstack()

# plot the bar chart
ax = grouped.plot(kind='bar', figsize=(8, 6), color=['b', 'g'],␣
 ↪edgecolor='black')
ax.set_xlabel('Member Type')
ax.set_ylabel('Number of Trips')
ax.set_title('Bike Preference by Member Type')

plt.show()
```

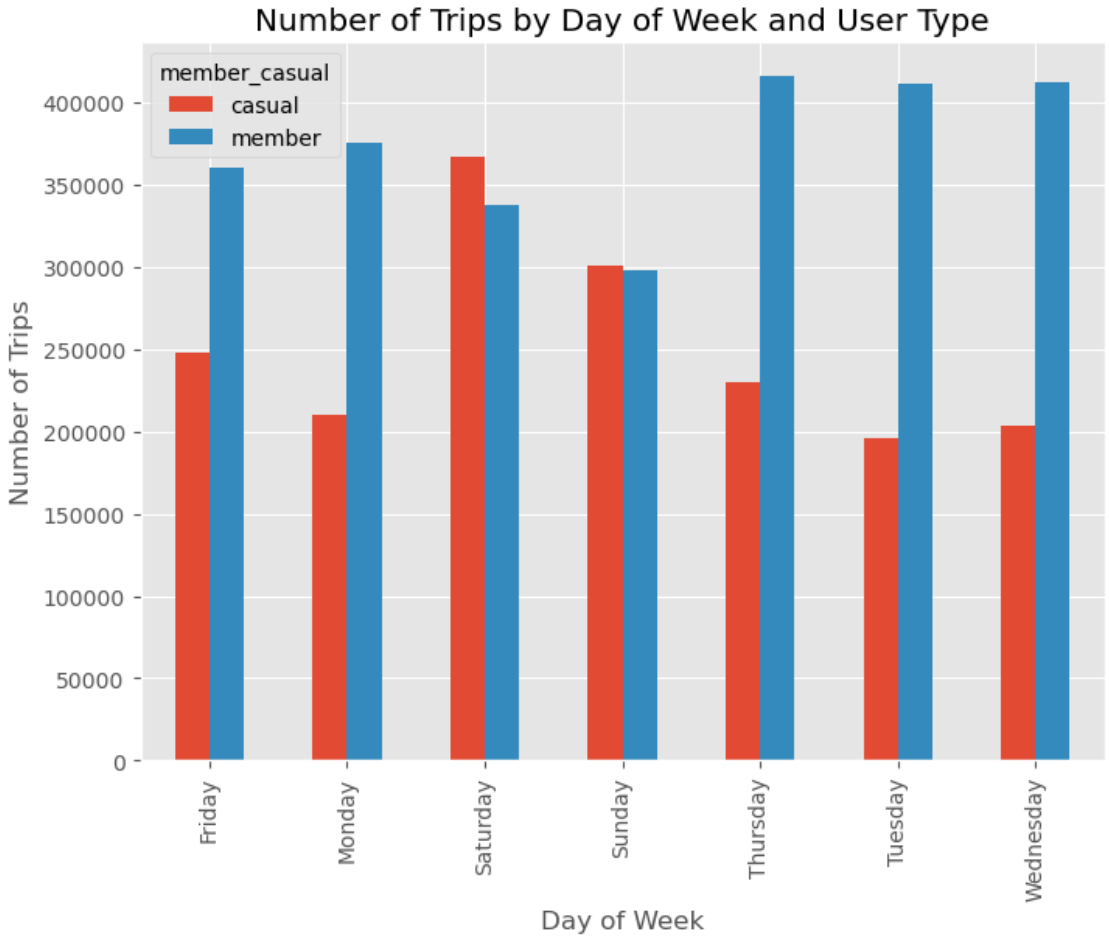


[87]:
```
df.head()
```

[87]:
```
        rideable_type          started_at            ended_at  \
172090   docked_bike 2022-01-19 02:58:15 2022-01-19 03:19:14
172091   docked_bike 2022-01-19 14:40:18 2022-01-19 14:50:22
172092   docked_bike 2022-01-05 00:22:40 2022-01-05 00:45:47
172093   docked_bike 2022-01-26 13:23:58 2022-01-26 15:02:49
172094   docked_bike 2022-01-05 02:34:50 2022-01-05 02:38:57

                start_station_name            end_station_name  start_lat  \
172090  Clinton St & Roosevelt Rd        Halsted St & 35th St  41.867118
172091  Sedgwick St & Webster Ave           Clark St & Elm St  41.922167
172092  Ashland Ave & Division St           Clark St & Elm St  41.903450
172093  Halsted St & Roosevelt Rd  Wood St & Taylor St (Temp)  41.867324
172094      Broadway & Berwyn Ave        Broadway & Argyle St  41.978353

         start_lng    end_lat    end_lng member_casual day_of_week  \
172090 -87.641088  41.830661 -87.647172        casual   Wednesday
172091 -87.638888  41.902973 -87.631280        casual   Wednesday
172092 -87.667747  41.902973 -87.631280        casual   Wednesday
172093 -87.648625  41.869265 -87.673731        casual   Wednesday
172094 -87.659753  41.973815 -87.659660        casual   Wednesday

        day_of_week_no start_date          start_hour   end_date  \
172090               3 2022-01-19 1899-12-30 02:58:15 2022-01-19
172091               3 2022-01-19 1899-12-30 14:40:18 2022-01-19
172092               3 2022-01-05 1899-12-30 00:22:40 2022-01-05
172093               3 2022-01-26 1899-12-30 13:23:58 2022-01-26
172094               3 2022-01-05 1899-12-30 02:34:50 2022-01-05

                  end_hour  trip_duration_minutes  distance_KM bike_preference
172090 1899-12-30 03:19:14                     20     4.085036        standard
172091 1899-12-30 14:50:22                     10     2.225186        standard
172092 1899-12-30 00:45:47                     23     3.018457        standard
172093 1899-12-30 15:02:49                     38     2.090070        standard
172094 1899-12-30 02:38:57                      4     0.504661        standard
```

[88]:
```
# group by day of the week and user type, and count the number of trips
counts = df.groupby(['day_of_week', 'member_casual']).size().
 ↪reset_index(name='num_trips')

# pivot the data to make it easier to plot
pivoted_counts = counts.pivot(index='day_of_week', columns='member_casual',␣
 ↪values='num_trips')

# plot the data
pivoted_counts.plot(kind='bar', figsize=(8, 6))
plt.title('Number of Trips by Day of Week and User Type')
plt.xlabel('Day of Week')
```

```
plt.ylabel('Number of Trips')
plt.show()

# find the maximum number of trips for each user type
max_member_trips = pivoted_counts['member'].max()
max_casual_trips = pivoted_counts['casual'].max()

print(f"The most number of trips for members in a single day is␣
 ↪{max_member_trips}")
print(f"The most number of trips for casual users in a single day is␣
 ↪{max_casual_trips}")
```

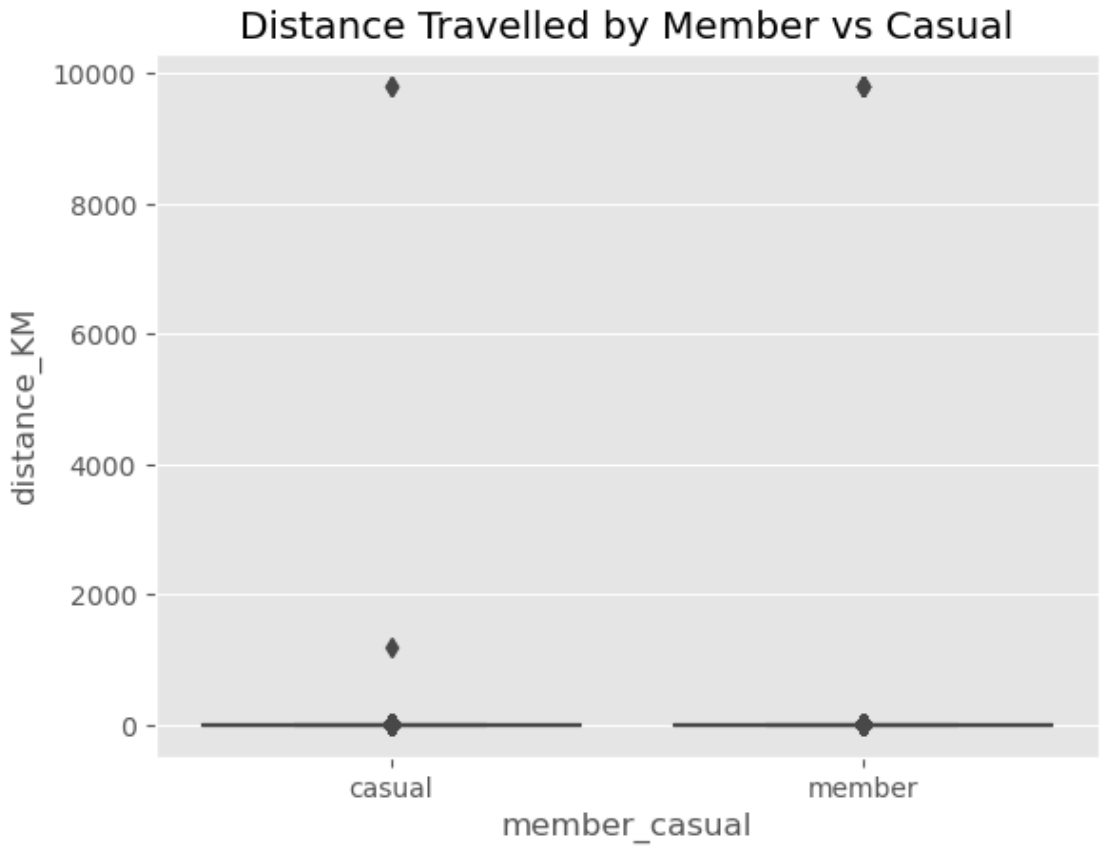



```
The most number of trips for members in a single day is 415739
The most number of trips for casual users in a single day is 366659
```

[90]:
```
df.query("member_casual=='member'") \
  .sort_values("distance_KM", ascending=False) \
  .head(1)
```

[90]:
```
         rideable_type          started_at            ended_at  \
5645109  classic_bike 2022-11-09 12:21:55 2022-11-09 12:26:18

                start_station_name         end_station_name  start_lat  \
5645109  Aberdeen St & Randolph St  Green St & Madison Ave*  41.884114

          start_lng  end_lat  end_lng member_casual day_of_week  \
5645109 -87.654264      0.0      0.0        member   Wednesday

          day_of_week_no start_date          start_hour   end_date  \
5645109                3 2022-11-09 1899-12-30 12:21:55 2022-11-09

                    end_hour  trip_duration_minutes  distance_KM  \
5645109 1899-12-30 12:26:18                      4  9813.377581

         bike_preference
5645109        standard
```

[91]:
```
sns.boxplot(x='member_casual', y='distance_KM', data=df)
plt.title('Distance Travelled by Member vs Casual')
plt.show()
```



[ ]: