1- Importing the datatrip and combining it as one dataset

In [1]:
```
#import library

import pandas as pd
from zipfile import ZipFile,Path
import glob
import fnmatch
from io import BytesIO, StringIO
import numpy as np
import datetime
```

In [2]:
```
#df_master : it is the combined data set

path = r"C:\Users\G84183771\Downloads\Learn\tripdata\tripcsv\*"
#load all zip files in folder
all_files = glob.glob(path)


df_master = pd.DataFrame()
#flog = False

for filename in all_files:
    df=pd.read_csv(filename)
    #print(df.head())
    df_master=pd.concat([df_master, df])
```

dataset information

In [3]:
```
''' Having columns of the dataset'''
df_master.columns
```

Out[3]:
```
Index(['ride_id', 'rideable_type', 'started_at', 'ended_at',
       'start_station_name', 'start_station_id', 'end_station_name',
       'end_station_id', 'start_lat', 'start_lng', 'end_lat', 'end_lng',
       'member_casual'],
      dtype='object')
```

In [4]:
```
'''converting to column to good data type'''
df_master['started_at']= pd.to_datetime(df_master['started_at'])
df_master['ended_at']=pd.to_datetime(df_master['ended_at'])
```

In [6]:
```
df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5883043 entries, 0 to 785931
Data columns (total 13 columns):
 #   Column              Dtype  
---  ------              -----  
 0   ride_id             object 
 1   rideable_type       object 
 2   started_at          datetime64[ns]
 3   ended_at            datetime64[ns]
 4   start_station_name  object 
 5   start_station_id    object 
 6   end_station_name    object 
 7   end_station_id      object 
 8   start_lat           float64
 9   start_lng           float64
 10  end_lat             float64
 11  end_lng             float64
 12  member_casual       object 
dtypes: datetime64[ns](2), float64(4), object(7)
memory usage: 628.4+ MB
```

2- Cleaning the dataset

adding new columns

In [7]:
```
''' calculating the ride length'''

## ride_length is in seconds

df_master['ride_length'] = df_master.ended_at-df_master.started_at
df_master['ride_length']=df_master['ride_length'].astype('timedelta64[s]') #converting it to seconds

df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5883043 entries, 0 to 785931
Data columns (total 14 columns):
 #   Column              Dtype  
---  ------              -----  
 0   ride_id             object 
 1   rideable_type       object 
 2   started_at          datetime64[ns]
 3   ended_at            datetime64[ns]
 4   start_station_name  object 
 5   start_station_id    object 
 6   end_station_name    object 
 7   end_station_id      object 
 8   start_lat           float64
 9   start_lng           float64
 10  end_lat             float64
 11  end_lng             float64
 12  member_casual       object 
 13  ride_length         float64
dtypes: datetime64[ns](2), float64(5), object(7)
memory usage: 673.3+ MB
```

In [9]:
```
''' Getting the week day 0=monday and 6=sunday'''

df_master['day_number']=df_master['started_at'].dt.day_of_week
df_master['day_name']=df_master['started_at'].dt.day_name()

df_master['month_name']= pd.to_datetime(df_master['started_at']).dt.to_period('M')
```

Verify if ride_length is negative

In [11]:
```
df_master[df_master.ride_length < 0]
```

Out[11]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_l
8950	BE9371BDC9182ED6	classic_bike	2021-09-29 17:04:38	2021-09-29 17:04:27	Shields Ave & 28th Pl	15443	Shields Ave & 28th Pl	15443	41.8427
49311	6E5FD2F624ACB7D3	classic_bike	2021-09-01 17:49:37	2021-09-01 17:49:31	Clybourn Ave & Division St	TA1307000115	Clybourn Ave & Division St	TA1307000115	41.9046
69949	FA4DC99A39C36D54	classic_bike	2021-09-29 16:53:34	2021-09-29 16:53:29	Financial Pl & Ida B Wells Dr	SL-010	Financial Pl & Ida B Wells Dr	SL-010	41.8750
82802	858C495341A82F18	electric_bike	2021-09-01 18:45:38	2021-09-01 18:45:24	Halsted St & Dickens Ave	13192	Halsted St & Dickens Ave	13192	41.9198
139417	4A68473D329D45C9	classic_bike	2021-09-29 18:42:50	2021-09-29 18:36:24	Ashland Ave & Division St	13061	Ashland Ave & Division St	13061	41.9034
...	...	...	...	...	...	...	...	...	...
677683	A2991D490436A806	electric_bike	2022-08-27 13:18:54	2022-08-27 13:15:58	Lincoln Ave & Roscoe St*	chargingstx5	Lincoln Ave & Roscoe St*	chargingstx5	41.9433
677684	E2F6294C6E6E07AA	electric_bike	2022-08-27 13:17:51	2022-08-27 13:15:58	Lincoln Ave & Roscoe St*	chargingstx5	Lincoln Ave & Roscoe St*	chargingstx5	41.9433
677685	ECE5401B617CC3AE7	electric_bike	2022-08-27 13:22:25	2022-08-27 13:15:58	Lincoln Ave & Roscoe St*	chargingstx5	Lincoln Ave & Roscoe St*	chargingstx5	41.9432
682325	0D8781397E228787	electric_bike	2022-08-27 13:16:39	2022-08-27 13:15:58	NaN	NaN	Lincoln Ave & Roscoe St*	chargingstx5	41.9400
682421	F33D11F3AF0F522E	electric_bike	2022-08-25 00:39:40	2022-08-25 00:39:34	NaN	NaN	Lincoln Ave & Roscoe St*	chargingstx5	41.9400

135 rows × 17 columns

From that we can identify that the start_at and end_at are not accurate, we cannot use this values for our analysis. This will be excluded to continue

In [12]:
```
''' The new data frame with the accurate ride_length'''
df_master=df_master[df_master.ride_length > 0]
```

Checking for duplicate rows

In [13]:
```
df_master[df_master.duplicated()]
```

Out[13]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat

Finding Missing values in every columns

In [14]:
```
df_master.isnull().sum()
```

Out[14]:
```
ride_id                0
rideable_type          0
started_at             0
ended_at               0
start_station_name    884333
start_station_id      884331
end_station_name      946004
end_station_id        946004
start_lat                0
start_lng                0
end_lat               5727
end_lng               5727
member_casual            0
ride_length              0
day_number               0
day_name                 0
month_name               0
dtype: int64
```

3- Analysis

calculations

In [15]:
```
# calculation of the mean of ride_length

mean_value=df_master.ride_length.mean()
print('The mean of ride_length is : {} seconds'.format(mean_value))
```

```
The mean of ride_length is : 1185.366370604564 seconds
```

In [16]:
```
# calculation of the max of ride_length

max_value=df_master.ride_length.max()
print('The max of ride_length is : {} seconds'.format(max_value))
```

```
The max of ride_length is : 2442301.0 seconds
```

In [17]:
```
# calculation of the mode of the day of the week

mode_week_day=df_master.day_name.mode()

print('Mode of the day of the week is : '+str(mode_week_day))
```

```
Mode of the day of the week is : 0    Saturday
Name: day_name, dtype: object
```

average ride_length for members and casual riders.

In [18]:
```
df_master.groupby(['member_casual'])['ride_length'].mean()
```

Out[18]:
```
member_casual
casual    1758.072523
member     771.358566
Name: ride_length, dtype: float64
```

The average ride length of casual user is greater than that of members

average ride_length for users by day_of_week.

In [19]:
```
draw2=df_master.groupby(['member_casual','day_name'])['ride_length'].mean().unstack()
draw2
```

Out[19]:

day_name	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
member_casual							
casual	1673.518862	1790.722222	1922.618228	2051.404179	1561.735976	1555.067563	1502.584880
member	755.722793	747.054532	858.159594	865.520164	742.521129	730.728589	731.830212

number of rides for users by day_of_week by adding Count of trip_id to Values.

In [20]:
```
df_master.groupby(['member_casual','day_name'])['ride_id'].count().unstack()
```

Out[20]:

day_name	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
member_casual							
casual	345988	292158	509996	437368	311457	277979	293221
member	472149	474789	453984	404467	525451	536453	546977

In [21]:
```
df_master.to_csv('final_trip.csv',index=False)
```

In [22]:
```
df_master.groupby(['day_name'])['day_name'].count()
```

Out[22]:
```
day_name
Friday       818137
Monday       766947
Saturday     963980
Sunday       841835
Thursday     836908
Tuesday      814432
Wednesday    840198
Name: day_name, dtype: int64
```

In [ ]:
```

```