Task 1.

Overview of the user behaviour

Task 1.1 - Your employer wants to have an overview of the users' behaviour on those applications. • Aggregate per user the following information in the column • number of xDR sessions • Session duration • the total download (DL) and upload (UL) data • the total data volume (in Bytes) during this session for each applicatio

In [1]:
```
#import libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
data=pd.read_csv("telcom_data1.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	Bearer Id	Start	Start ms	End	End ms	Dur. (ms)	IMSI	MSISDN/Number	IMEI	Last Location Name	...	Youtube DL (Bytes)	UL
0	1.310000e+19	04-04-19 12:01	770.0	4/25/2019 14:35	662.0	1823652.0	2.082010e+14	3.366496e+10	3.552120e+13	9.16E+15	...	15854611.0	25
1	1.310000e+19	04-09-19 13:04	235.0	4/25/2019 8:15	606.0	1365104.0	2.082020e+14	3.368185e+10	3.579400e+13	L77566A	...	20247395.0	19
2	1.310000e+19	04-09-19 17:42	1.0	4/25/2019 11:58	652.0	1361762.0	2.082000e+14	3.376063e+10	3.528150e+13	D42335A	...	19725661.0	146
3	1.310000e+19	04-10-19 0:31	486.0	4/25/2019 7:36	171.0	1321509.0	2.082010e+14	3.375034e+10	3.535660e+13	T21824A	...	21388122.0	151
4	1.310000e+19	04-12-19 20:10	565.0	4/25/2019 10:40	954.0	1089009.0	2.082010e+14	3.369980e+10	3.540700e+13	D88865A	...	15259380.0	189

5 rows × 55 columns

In [4]:
```
data.shape
```

Out[4]:
```
(150001, 55)
```

'Bearer Id','Dur. (ms)','Activity Duration DL (ms)','Activity Duration UL (ms)','Social Media DL (Bytes)', 'Social Media UL (Bytes)', 'Google DL (Bytes)', 'Google UL (Bytes)', 'Email DL (Bytes)', 'Email UL (Bytes)', 'Youtube DL (Bytes)', 'Youtube UL (Bytes)', 'Netflix DL (Bytes)', 'Netflix UL (Bytes)', 'Gaming DL (Bytes)', 'Gaming UL (Bytes)', 'Other DL (Bytes)', 'Other UL (Bytes)', 'Total UL (Bytes)', 'Total DL (Bytes)'

In [5]:
```
data.columns
```

Out[5]:
```
Index(['Bearer Id', 'Start', 'Start ms', 'End', 'End ms', 'Dur. (ms)', 'IMSI',
       'MSISDN/Number', 'IMEI', 'Last Location Name', 'Avg RTT DL (ms)',
       'Avg RTT UL (ms)', 'Avg Bearer TP DL (kbps)', 'Avg Bearer TP UL (kbps)',
       'TCP DL Retrans. Vol (Bytes)', 'TCP UL Retrans. Vol (Bytes)',
       'DL TP < 50 Kbps (%)', '50 Kbps < DL TP < 250 Kbps (%)',
       '250 Kbps < DL TP < 1 Mbps (%)', 'DL TP > 1 Mbps (%)',
       'UL TP < 10 Kbps (%)', '10 Kbps < UL TP < 50 Kbps (%)',
       '50 Kbps < UL TP < 300 Kbps (%)', 'UL TP > 300 Kbps (%)',
       'HTTP DL (Bytes)', 'HTTP UL (Bytes)', 'Activity Duration DL (ms)',
       'Activity Duration UL (ms)', 'Dur. (ms).1', 'Handset Manufacturer',
       'Handset Type', 'Nb of sec with 125000B < Vol DL',
       'Nb of sec with 1250B < Vol UL < 6250B',
       'Nb of sec with 31250B < Vol DL < 125000B',
       'Nb of sec with 37500B < Vol UL',
       'Nb of sec with 6250B < Vol DL < 31250B',
       'Nb of sec with 6250B < Vol UL < 37500B',
       'Nb of sec with Vol DL < 6250B', 'Nb of sec with Vol UL < 1250B',
       'Social Media DL (Bytes)', 'Social Media UL (Bytes)',
       'Google DL (Bytes)', 'Google UL (Bytes)', 'Email DL (Bytes)',
       'Email UL (Bytes)', 'Youtube DL (Bytes)', 'Youtube UL (Bytes)',
       'Netflix DL (Bytes)', 'Netflix UL (Bytes)', 'Gaming DL (Bytes)',
       'Gaming UL (Bytes)', 'Other DL (Bytes)', 'Other UL (Bytes)',
       'Total UL (Bytes)', 'Total DL (Bytes)'],
      dtype='object')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150001 entries, 0 to 150000
Data columns (total 55 columns):
 #   Column                                    Non-Null Count   Dtype  
---  ------                                    --------------   -----  
 0   Bearer Id                                 149010 non-null  float64
 1   Start                                     150000 non-null  object 
 2   Start ms                                  150000 non-null  float64
 3   End                                       150000 non-null  object 
 4   End ms                                    150000 non-null  float64
 5   Dur. (ms)                                 150000 non-null  float64
 6   IMSI                                      149431 non-null  float64
 7   MSISDN/Number                             148935 non-null  float64
 8   IMEI                                      149429 non-null  float64
 9   Last Location Name                        148848 non-null  object 
 10  Avg RTT DL (ms)                           122172 non-null  float64
 11  Avg RTT UL (ms)                           122189 non-null  float64
 12  Avg Bearer TP DL (kbps)                   150000 non-null  float64
 13  Avg Bearer TP UL (kbps)                   150000 non-null  float64
 14  TCP DL Retrans. Vol (Bytes)               61855 non-null   float64
 15  TCP UL Retrans. Vol (Bytes)               53352 non-null   float64
 16  DL TP < 50 Kbps (%)                       149247 non-null  float64
 17  50 Kbps < DL TP < 250 Kbps (%)            149247 non-null  float64
 18  250 Kbps < DL TP < 1 Mbps (%)             149247 non-null  float64
 19  DL TP > 1 Mbps (%)                        149247 non-null  float64
 20  UL TP < 10 Kbps (%)                       149209 non-null  float64
 21  10 Kbps < UL TP < 50 Kbps (%)             149209 non-null  float64
 22  50 Kbps < UL TP < 300 Kbps (%)            149209 non-null  float64
 23  UL TP > 300 Kbps (%)                      149209 non-null  float64
 24  HTTP DL (Bytes)                           68527 non-null   float64
 25  HTTP UL (Bytes)                           68191 non-null   float64
 26  Activity Duration DL (ms)                 150000 non-null  float64
 27  Activity Duration UL (ms)                 150000 non-null  float64
 28  Dur. (ms).1                               150000 non-null  float64
 29  Handset Manufacturer                      149429 non-null  object 
 30  Handset Type                              149429 non-null  object 
 31  Nb of sec with 125000B < Vol DL           52463 non-null   float64
 32  Nb of sec with 1250B < Vol UL < 6250B     57107 non-null   float64
 33  Nb of sec with 31250B < Vol DL < 125000B  56415 non-null   float64
 34  Nb of sec with 37500B < Vol UL            19747 non-null   float64
 35  Nb of sec with 6250B < Vol DL < 31250B    61684 non-null   float64
 36  Nb of sec with 6250B < Vol UL < 37500B    38158 non-null   float64
 37  Nb of sec with Vol DL < 6250B             149246 non-null  float64
```

```
 38  Nb of sec with Vol UL < 1250B            149208 non-null  float64
 39  Social Media DL (Bytes)                  150001 non-null  float64
 40  Social Media UL (Bytes)                  150001 non-null  float64
 41  Google DL (Bytes)                        150001 non-null  float64
 42  Google UL (Bytes)                        150001 non-null  float64
 43  Email DL (Bytes)                         150001 non-null  float64
 44  Email UL (Bytes)                         150001 non-null  float64
 45  Youtube DL (Bytes)                       150001 non-null  float64
 46  Youtube UL (Bytes)                       150001 non-null  float64
 47  Netflix DL (Bytes)                       150001 non-null  float64
 48  Netflix UL (Bytes)                       150001 non-null  float64
 49  Gaming DL (Bytes)                        150001 non-null  float64
 50  Gaming UL (Bytes)                        150001 non-null  float64
 51  Other DL (Bytes)                         150001 non-null  float64
 52  Other UL (Bytes)                         150001 non-null  float64
 53  Total UL (Bytes)                         150000 non-null  float64
 54  Total DL (Bytes)                         150000 non-null  float64
dtypes: float64(50), object(5)
memory usage: 62.9+ MB
```

In [7]:
```
null_pct= data.isnull().sum() / len(data) * 100
null_pct
```

Out[7]:

Bearer Id	0.660662
Start	0.000667
Start ms	0.000667
End	0.000667
End ms	0.000667
Dur. (ms)	0.000667
IMSI	0.379997
MSISDN/Number	0.710662
IMEI	0.381331
Last Location Name	0.768662
Avg RTT DL (ms)	18.552543
Avg RTT UL (ms)	18.541210
Avg Bearer TP DL (kbps)	0.000667
Avg Bearer TP UL (kbps)	0.000667
TCP DL Retrans. Vol (Bytes)	58.763608
TCP UL Retrans. Vol (Bytes)	64.432237
DL TP < 50 Kbps (%)	0.502663
50 Kbps < DL TP < 250 Kbps (%)	0.502663
250 Kbps < DL TP < 1 Mbps (%)	0.502663
DL TP > 1 Mbps (%)	0.502663
UL TP < 10 Kbps (%)	0.527996
10 Kbps < UL TP < 50 Kbps (%)	0.527996
50 Kbps < UL TP < 300 Kbps (%)	0.527996
UL TP > 300 Kbps (%)	0.527996
HTTP DL (Bytes)	54.315638
HTTP UL (Bytes)	54.539636
Activity Duration DL (ms)	0.000667
Activity Duration UL (ms)	0.000667
Dur. (ms).1	0.000667
Handset Manufacturer	0.381331
Handset Type	0.381331
Nb of sec with 125000B < Vol DL	65.024900
Nb of sec with 1250B < Vol UL < 6250B	61.928920
Nb of sec with 31250B < Vol DL < 125000B	62.390251
Nb of sec with 37500B < Vol UL	86.835421
Nb of sec with 6250B < Vol DL < 31250B	58.877607
Nb of sec with 6250B < Vol UL < 37500B	74.561503
Nb of sec with Vol DL < 6250B	0.503330
Nb of sec with Vol UL < 1250B	0.528663
Social Media DL (Bytes)	0.000000
Social Media UL (Bytes)	0.000000
Google DL (Bytes)	0.000000
Google UL (Bytes)	0.000000

```
Email DL (Bytes)                          0.000000
Email UL (Bytes)                          0.000000
Youtube DL (Bytes)                        0.000000
Youtube UL (Bytes)                        0.000000
Netflix DL (Bytes)                        0.000000
Netflix UL (Bytes)                        0.000000
Gaming DL (Bytes)                         0.000000
Gaming UL (Bytes)                         0.000000
Other DL (Bytes)                          0.000000
Other UL (Bytes)                          0.000000
Total UL (Bytes)                          0.000667
Total DL (Bytes)                          0.000667
dtype: float64
```

In [8]:
```
# top 10 Handset Type use by customers
Handset_type=data["Handset Type"].value_counts(ascending=False).head(10)
Handset_type
```

Out[8]:
```
Huawei B528S-23A                19752
Apple iPhone 6S (A1688)          9419
Apple iPhone 6 (A1586)           9023
undefined                        8987
Apple iPhone 7 (A1778)           6326
Apple iPhone Se (A1723)          5187
Apple iPhone 8 (A1905)           4993
Apple iPhone Xr (A2105)          4568
Samsung Galaxy S8 (Sm-G950F)     4520
Apple iPhone X (A1901)           3813
Name: Handset Type, dtype: int64
```

In [9]:
```
# top 3 handset manufacturer
top_3_manufacturer=data["Handset Manufacturer"].value_counts(ascending=False).head(3)
top_3_manufacturer
```

Out[9]:
```
Apple      59565
Samsung    40839
Huawei     34423
Name: Handset Manufacturer, dtype: int64
```

recommendation to marketing team

- Huawei,Apple and,Samsung are the top 3 manufacturer in the current market
- From top 10 handset type we conclude,these are the popular devices in current market
- so the marketing team should focus on promoting these brands and collaborate with apple samsung and Huewai manufacturer to achieve target

In [11]:
```
# separate categorical and numerical variable
categorical_feature=data.select_dtypes(include=[np.object])
categorical_feature.columns
```

Out[11]:
```
Index(['Start', 'End', 'Last Location Name', 'Handset Manufacturer',
       'Handset Type'],
      dtype='object')
```

we can see there is data inconsistancy as some colums are numerical bit define as object type

In [10]:
```
# select numerical
numeric_features = data.select_dtypes(include=[np.number])

numeric_features.columns
```

Out[10]:
```
Index(['Bearer Id', 'Start ms', 'End ms', 'Dur. (ms)', 'IMSI', 'MSISDN/Number',
       'IMEI', 'Avg RTT DL (ms)', 'Avg RTT UL (ms)', 'Avg Bearer TP DL (kbps)',
       'Avg Bearer TP UL (kbps)', 'TCP DL Retrans. Vol (Bytes)',
       'TCP UL Retrans. Vol (Bytes)', 'DL TP < 50 Kbps (%)',
       '50 Kbps < DL TP < 250 Kbps (%)', '250 Kbps < DL TP < 1 Mbps (%)',
       'DL TP > 1 Mbps (%)', 'UL TP < 10 Kbps (%)',
       '10 Kbps < UL TP < 50 Kbps (%)', '50 Kbps < UL TP < 300 Kbps (%)',
       'UL TP > 300 Kbps (%)', 'HTTP DL (Bytes)', 'HTTP UL (Bytes)',
       'Activity Duration DL (ms)', 'Activity Duration UL (ms)', 'Dur. (ms).1',
       'Nb of sec with 125000B < Vol DL',
       'Nb of sec with 1250B < Vol UL < 6250B',
       'Nb of sec with 31250B < Vol DL < 125000B',
       'Nb of sec with 37500B < Vol UL',
       'Nb of sec with 6250B < Vol DL < 31250B',
       'Nb of sec with 6250B < Vol UL < 37500B',
       'Nb of sec with Vol DL < 6250B', 'Nb of sec with Vol UL < 1250B',
       'Social Media DL (Bytes)', 'Social Media UL (Bytes)',
       'Google DL (Bytes)', 'Google UL (Bytes)', 'Email DL (Bytes)',
       'Email UL (Bytes)', 'Youtube DL (Bytes)', 'Youtube UL (Bytes)',
       'Netflix DL (Bytes)', 'Netflix UL (Bytes)', 'Gaming DL (Bytes)',
       'Gaming UL (Bytes)', 'Other DL (Bytes)', 'Other UL (Bytes)',
       'Total UL (Bytes)', 'Total DL (Bytes)'],
      dtype='object')
```

In [11]:
```
##task 1.1
###overview of the user behaviour
```

In [12]:
```
'Dur. (ms)', 'Start ms','End ms','Total DL (Bytes)','Total UL (Bytes)'
```

Out[12]:
```
('Dur. (ms)', 'Start ms', 'End ms', 'Total DL (Bytes)', 'Total UL (Bytes)')
```

In [24]:
```
# lets consider user as Bearer Id
user_df=data.groupby('MSISDN/Number').agg({
                                            'Dur. (ms)':'sum',
                                            'Start ms':'sum',
                                            'End ms':'sum',
                                            'Social Media DL (Bytes)':'sum',
                                            'Social Media UL (Bytes)':'sum',
                                            'Google DL (Bytes)':'sum',
                                            'Google UL (Bytes)':'sum',
                                            'Email DL (Bytes)':'sum',
                                            'Email UL (Bytes)':'sum',
                                            'Youtube DL (Bytes)':'sum',
                                            'Youtube UL (Bytes)':'sum',

                                             'Netflix DL (Bytes)':'sum',
                                              'Netflix UL (Bytes)':'sum',
                                              'Gaming DL (Bytes)':'sum',
                                              'Gaming UL (Bytes)':'sum',
                                              'Other DL (Bytes)':'sum',
                                              'Other UL (Bytes)':'sum',
                                              'Total UL (Bytes)':'sum',
                                              'Total DL (Bytes)':'sum'}).reset_index()
```

In [25]:
```
# the user behaviour dataframe (if we consider user as bearer Id)
print(user_df)
```

```
        MSISDN/Number  Dur. (ms)  Start ms  End ms  Social Media DL (Bytes)  \
0        3.360100e+10   116720.0     138.0   278.0                2206504.0
1        3.360100e+10   181230.0     422.0   385.0                2598548.0
2        3.360100e+10   134969.0      80.0   454.0                3148004.0
3        3.360101e+10    49878.0     158.0   182.0                 251469.0
4        3.360101e+10    37104.0    1297.0  1750.0                2861230.0
...               ...        ...       ...     ...                      ...
106851   3.379000e+10     8810.0     829.0   517.0                 234320.0
106852   3.379000e+10   140988.0     846.0   505.0                 442214.0
106853   3.197020e+12   877385.0     345.0   670.0                 668596.0
106854   3.370000e+14   253030.0     811.0    12.0                 496337.0
106855   8.823970e+14   869844.0     259.0   459.0                1500145.0

        Social Media UL (Bytes)  Google DL (Bytes)  Google UL (Bytes)  \
0                       25631.0          3337123.0          1051882.0
1                       62017.0          4197697.0          1137166.0
2                       47619.0          3343483.0            99643.0
3                       28825.0          5937765.0          3740728.0
4                       51312.0         13728668.0          4770948.0
```

In [26]:
```
user_df
```

Out[26]:

	MSISDN/Number	Dur. (ms)	Start ms	End ms	Social Media DL (Bytes)	Social Media UL (Bytes)	Google DL (Bytes)	Google UL (Bytes)	Email DL (Bytes)	Email UL (Bytes)	Youtube DL (Bytes)	Yo UL (E
0	3.360100e+10	116720.0	138.0	278.0	2206504.0	25631.0	3337123.0	1051882.0	837400.0	493962.0	14900201.0	6724
1	3.360100e+10	181230.0	422.0	385.0	2598548.0	62017.0	4197697.0	1137166.0	2828821.0	478960.0	5324251.0	7107
2	3.360100e+10	134969.0	80.0	454.0	3148004.0	47619.0	3343483.0	99643.0	2436500.0	768880.0	2137272.0	19196
3	3.360101e+10	49878.0	158.0	182.0	251469.0	28825.0	5937765.0	3740728.0	2178618.0	106052.0	4393123.0	2584
4	3.360101e+10	37104.0	1297.0	1750.0	2861230.0	51312.0	13728668.0	4770948.0	2247808.0	1057661.0	10339971.0	31193
...	...	...	...	...	...	...	...	...	...	...	...	
106851	3.379000e+10	8810.0	829.0	517.0	234320.0	65863.0	6834178.0	697091.0	480946.0	525969.0	8294310.0	18353
106852	3.379000e+10	140988.0	846.0	505.0	442214.0	56355.0	1472406.0	3957299.0	2513433.0	664.0	5596862.0	14254
106853	3.197020e+12	877385.0	345.0	670.0	668596.0	46628.0	8572779.0	1865881.0	842279.0	678492.0	9839889.0	2120
106854	3.370000e+14	253030.0	811.0	12.0	496337.0	25229.0	8215537.0	1615080.0	2989663.0	328919.0	16690728.0	20044
106855	8.823970e+14	869844.0	259.0	459.0	1500145.0	45943.0	5985089.0	3233558.0	2518425.0	812549.0	18980320.0	21960

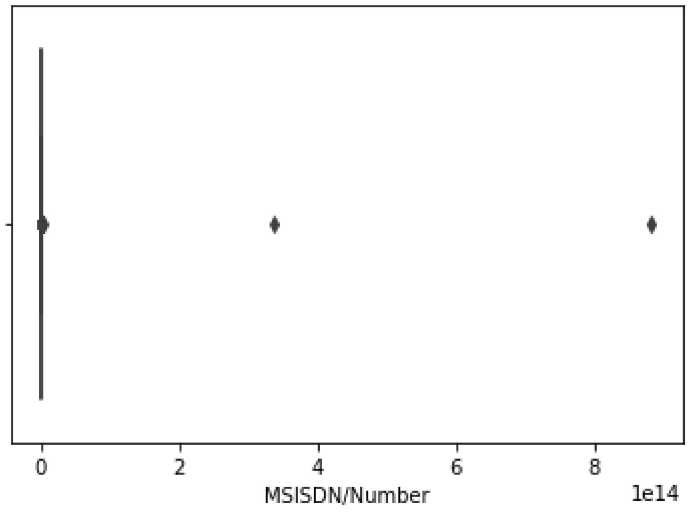
106856 rows × 20 columns

In [27]:
```
df=pd.DataFrame(user_df)
```

In [28]:
```
df.shape
```

Out[28]:
```
(106856, 20)
```

In [29]:
```
for i in df.columns:
    sns.boxplot(x=i,data=df)
    plt.show()
```



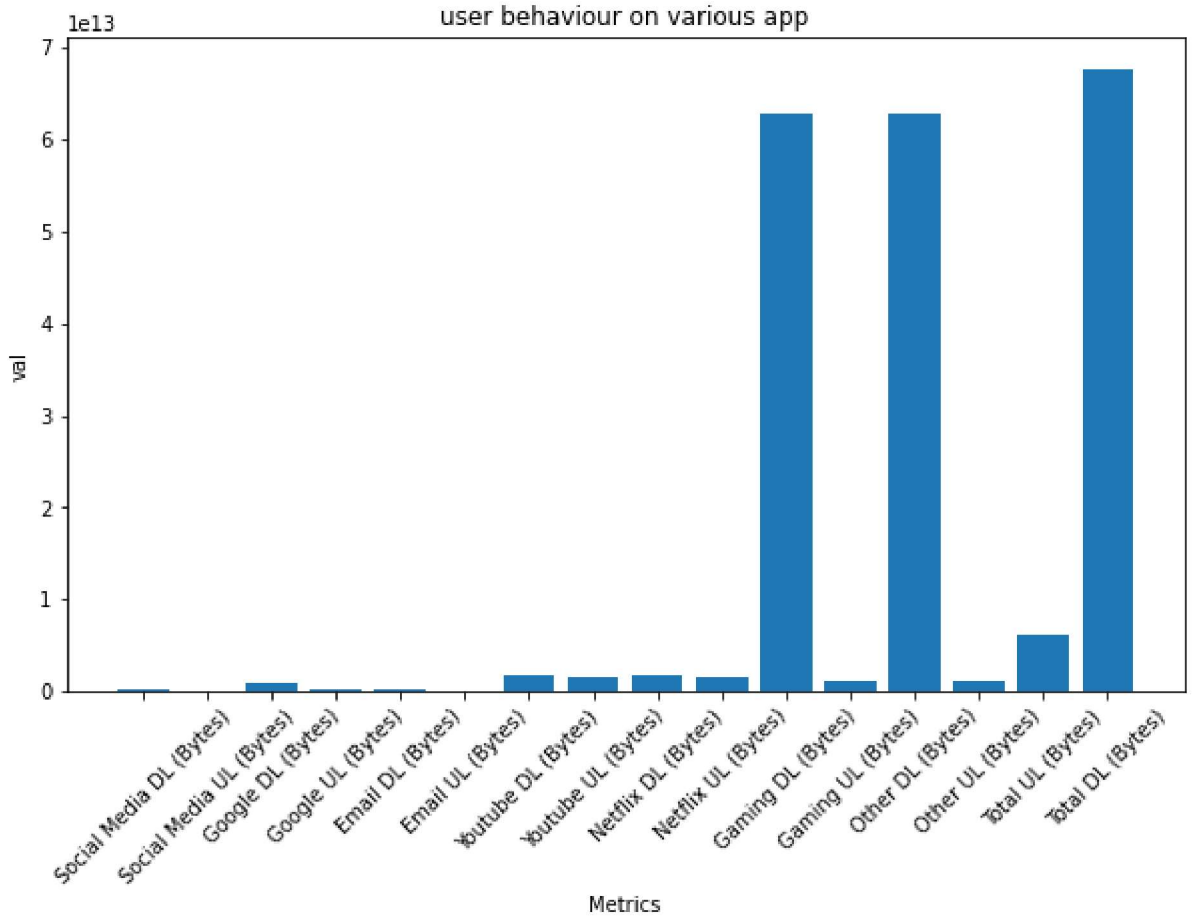
In [31]:
```
Metrics=['Social Media DL (Bytes)', 'Social Media UL (Bytes)',
       'Google DL (Bytes)', 'Google UL (Bytes)', 'Email DL (Bytes)',
       'Email UL (Bytes)', 'Youtube DL (Bytes)', 'Youtube UL (Bytes)',
       'Netflix DL (Bytes)', 'Netflix UL (Bytes)', 'Gaming DL (Bytes)',
       'Gaming UL (Bytes)', 'Other DL (Bytes)', 'Other UL (Bytes)',
       'Total UL (Bytes)', 'Total DL (Bytes)']
```

In [32]:
```
val=df[Metrics].sum()
```

In [33]:
```
val
```

Out[33]:
```
Social Media DL (Bytes)    2.673623e+11
Social Media UL (Bytes)    4.903196e+09
Google DL (Bytes)          8.566107e+11
Google UL (Bytes)          3.062427e+11
Email DL (Bytes)           2.668571e+11
Email UL (Bytes)           6.961065e+10
Youtube DL (Bytes)         1.732628e+12
Youtube UL (Bytes)         1.639576e+12
Netflix DL (Bytes)         1.731750e+12
Netflix UL (Bytes)         1.638310e+12
Gaming DL (Bytes)          6.285438e+13
Gaming UL (Bytes)          1.234535e+12
Other DL (Bytes)           6.272357e+13
Other UL (Bytes)           1.230678e+12
Total UL (Bytes)           6.123856e+12
Total DL (Bytes)           6.770959e+13
dtype: float64
```

In [34]:
```
# plot the bar graph to view user behaviour
plt.figure(figsize=(10,6))
plt.bar(Metrics,val)
plt.title('user behaviour on various app') # Set Title
plt.xlabel('Metrics')
plt.ylabel('val')
plt.xticks(rotation=45) # rotate x label for readability
plt.show()
```



from above data we can conclude that users are spending much of data on watching Netflix,Gaming, and downloading data

In [35]:
```
user_df.skew()
```

Out[35]:
```
MSISDN/Number              281.347079
Dur. (ms)                   20.539551
Start ms                     2.026216
End ms                       2.345960
Social Media DL (Bytes)      2.064133
Social Media UL (Bytes)      2.015622
Google DL (Bytes)            2.064540
Google UL (Bytes)            2.032406
Email DL (Bytes)             2.099718
Email UL (Bytes)             2.056152
Youtube DL (Bytes)           2.072224
Youtube UL (Bytes)           2.069953
Netflix DL (Bytes)           2.021664
Netflix UL (Bytes)           2.023251
Gaming DL (Bytes)            2.040510
Gaming UL (Bytes)            2.008822
Other DL (Bytes)             1.978933
Other UL (Bytes)             2.030694
Total UL (Bytes)             2.895198
Total DL (Bytes)             2.149144
dtype: float64
```

if skewness is betn -0.5 and 0.5 the data is nearly symmetrical. And skewness range betn -1 and -0.5 are negatively skewed., and if range is betn 0.5 to 1 data is positively skewed and if range greater than 1 data is extremly skewed