Data Analysis and Insights for different page Optimization & How to get more user install & Engagement from the App & Website

Introduction

This dataset contains data of XYZ Inc ,containing comprehensive user data spanning various regions, customer demographics, product details, and marketing campaigns . This analysis, findings and recommendations will serve as a compass guiding XYZ Inc. toward enhanced user satisfaction, increased installations, and heightened engagement.

In [ ]:

```python
# Loading the dataset and importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Read multiple sheets from the same Excel file
xls = pd.ExcelFile(r'C:/Users/expon/Downloads/Data set for BA.xlsx')
```

In [ ]:

```python
xls.sheet_names
```

Out[ ]:

```
['Report Snapshot',
 'User Acquisition',
 'Traffic Aquisition',
 'Event Report',
 'Conversion Report',
 'Pages & Screens Report',
 'Retention Overview',
 'User Engagement Overview',
 'Demographics Report',
 'Citiwise Report',
 'Gender Report',
 'User By Interest',
 'User by Language',
 'User By Age',
 'Google Ads Report']
```

In [ ]:

```python
print(f'This dataset has  {len(xls.sheet_names)} sheets')
```

```
This dataset has  15 sheets
```

User Installation & Engagement Performance Analysis:

Analysis on Event Report

In [ ]:

```python
er = pd.read_excel(xls, 'Event Report')
er.head(20)
```

Out[ ]:

	Event name	Event count	Total users	Event count per user	Total revenue
0	screen_view	694729	23254	30.865870	0
1	notification_receive	125146	1700	138.896800	0
2	user_engagement	124836	22699	5.622230	0
3	notification_dismiss	70128	1369	144.000000	0
4	session_start	61163	23226	3.121357	0
5	first_open	22872	23059	0.991890	0
6	app_remove	18080	18030	4.037517	0
7	Promilo113_login	12719	6174	2.068803	0
8	Promilo1116_feeds	7942	1837	4.323353	0
9	Promilo111_login	7449	3517	2.130721	0
10	Promilo111_feeds	6240	1045	5.988484	0
11	Promilo1116_my_rewards_screen	5863	884	6.632353	0
12	Promilo106_feeds	5629	1146	4.911867	0
13	Promilo113_feeds	5495	1376	3.996364	0
14	Promilo1116_login	5315	2263	2.351770	0
15	Promilo106_login	5258	2232	2.365272	0
16	Promilo1114_feeds	5062	1187	4.264532	0
17	Promilo1114_my_rewards_screen	4774	509	9.379175	0
18	Promilo111_my_rewards_screen	4056	488	8.311475	0
19	Promilo1116_storyboard	3914	2464	1.598856	0

In [ ]:

```
er.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 379 entries, 0 to 378
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Event name            378 non-null    object 
 1   Event count           379 non-null    int64  
 2   Total users           379 non-null    int64  
 3   Event count per user  379 non-null    float64
 4   Total revenue         379 non-null    int64  
dtypes: float64(1), int64(3), object(1)
memory usage: 14.9+ KB
```

In [ ]:

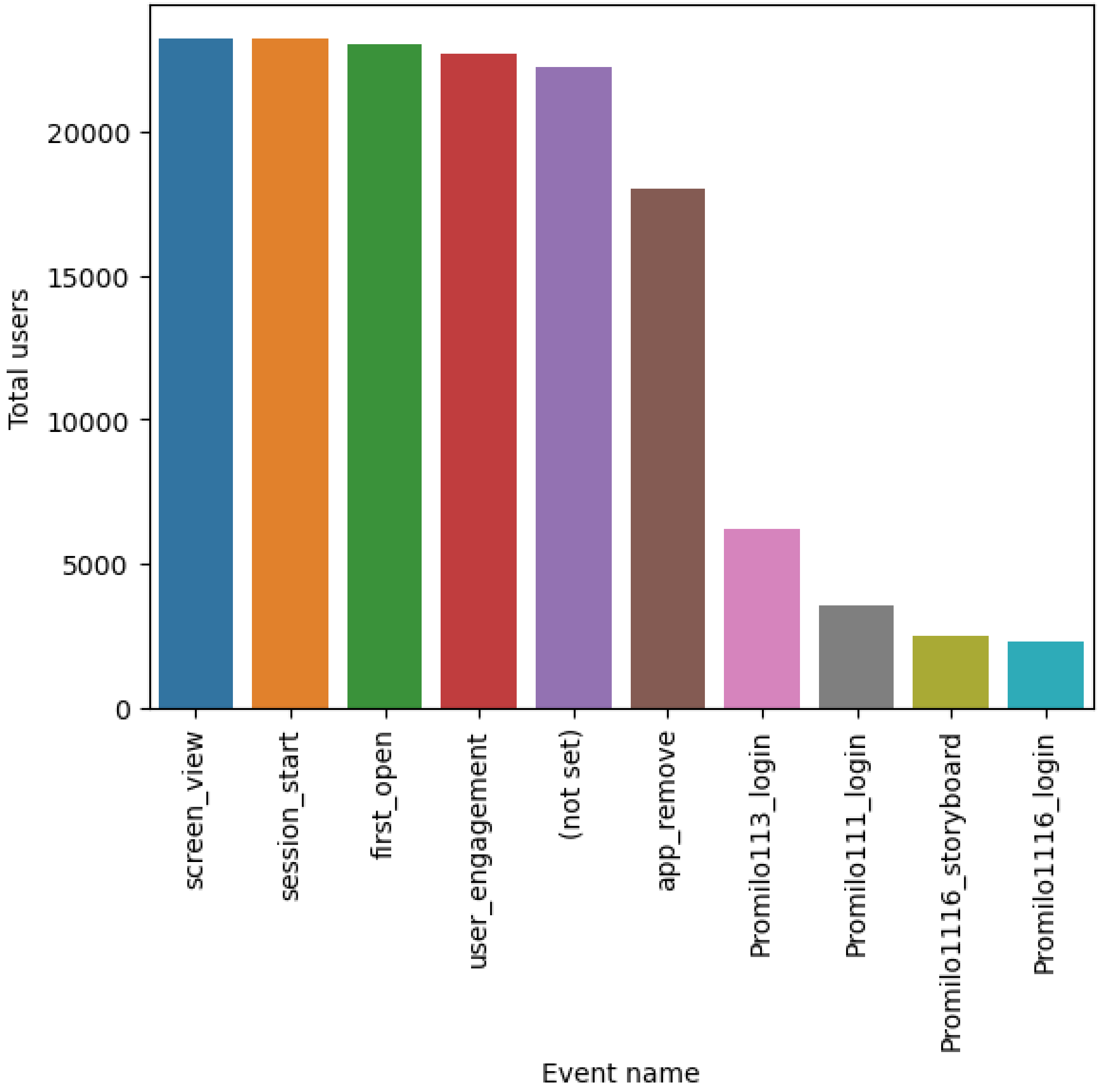
```
#sorting the values on Total users
er=er.sort_values(by=['Total users'],ascending=False)
```

In [ ]:

```
er1=er.head(10)
```

In [ ]:

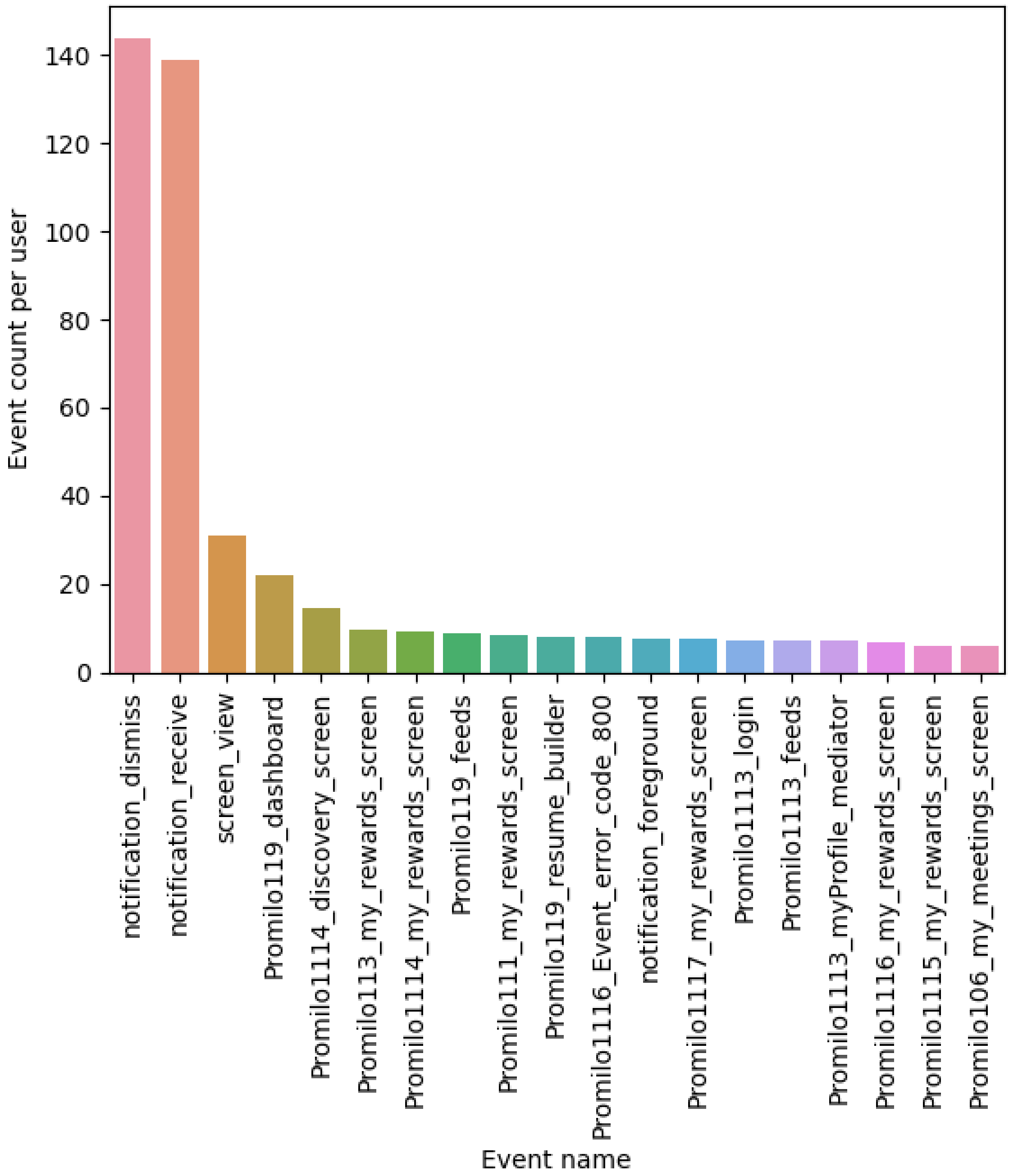
```
sns.barplot(data=er1,x='Event name',y='Total users')
plt.xticks(rotation=90)
plt.show()
```



- this plot shows that most of the users who install the app atleast start a session
- There are 23254 users and 18030 users removed app which is a high number
- only 1700 users reveive notification

```
In [ ]: # Event count per user is a parameter,where we can see which app feature is being u

        er2=er.sort_values(by=['Event count per user'],ascending=False)
        er2=er2.head(20)
        sns.barplot(data=er2,x='Event name',y='Event count per user')
        plt.xticks(rotation=90)
        plt.show()
```



Analysis On User Demographics

Analysis on Gender Report

In [ ]:
```
gr= pd.read_excel(xls, 'Gender Report')
gr
```

Out[ ]:

	Gender	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions	r
0	unknown	13142	12691	23161	0.564077	1.762365	439.5776	761771	93180	
1	male	7218	5877	10467	0.543091	1.450125	128.2319	282504	65651	
2	female	4944	4304	7877	0.637710	1.593244	208.7407	274254	35083	

In [ ]:
```
gr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Gender                       3 non-null      object 
 1   Users                        3 non-null      int64  
 2   New users                    3 non-null      int64  
 3   Engaged sessions             3 non-null      int64  
 4   Engagement rate              3 non-null      float64
 5   Engaged sessions per user    3 non-null      float64
 6   Average engagement time      3 non-null      float64
 7   Event count                  3 non-null      int64  
 8   Conversions                  3 non-null      int64  
 9   Total revenue                3 non-null      int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 368.0+ bytes
```

In [ ]:

```
px.pie(gr,values='Users',names='Gender')
```

In [ ]:

```
px.pie(gr,values='Average engagement time',names='Gender')
```

- male customers are more in number than female customer
- Despite that we can see that Average engagement time for female are more

Analysis on Age Data

In [ ]:

```
uba= pd.read_excel(xls, 'User By Age')
uba1=uba.drop(index=0)
uba1
```

Out[ ]:

	Age	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions	Tot revenu
1	18-24	4282	3678	7291	0.695308	1.702709	251.16300	309328	53661	
2	25-34	2920	2161	3749	0.504780	1.283904	97.24144	90074	20172	
3	65+	1422	1081	1640	0.539829	1.153305	52.30661	24780	4891	
4	55-64	1403	979	1552	0.519411	1.106201	55.37063	25169	4823	
5	35-44	1202	785	1420	0.510424	1.181364	96.08236	33016	8111	
6	45-54	810	552	881	0.561862	1.087654	84.54321	18661	2946	

In [ ]:

```
uba.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Age                          7 non-null      object 
 1   Users                        7 non-null      int64  
 2   New users                    7 non-null      int64  
 3   Engaged sessions             7 non-null      int64  
 4   Engagement rate              7 non-null      float64
 5   Engaged sessions per user    7 non-null      float64
 6   Average engagement time      7 non-null      float64
 7   Event count                  7 non-null      int64  
 8   Conversions                  7 non-null      int64  
 9   Total revenue                7 non-null      int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 688.0+ bytes
```

In [ ]:

```
px.pie(uba1,values='Users',names='Age')
```

- most of the users belongs to 18-24 category
- 18-35 age group constitutes 60% of the total users

In [ ]:

```
px.pie(uba1,values='Average engagement time',names='Age')
```

- however 18-24 age group has highest average engagement time

Analysis on User Interest report

In [ ]:

```
ui= pd.read_excel(xls, 'User By Interest')
ui=ui.sort_values(by=['Users'],ascending=False)
ui1=ui.head(30)
ui1.head(3)
```

Out[ ]:

	Interests	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Cor
0	Shoppers	10950	9256	15652	0.581534	1.429406	162.8347	490664	
1	Media & Entertainment/Comics & Animation Fans	10946	9247	15680	0.583008	1.432487	165.1772	491025	
2	Technology/Mobile Enthusiasts	10934	9239	15619	0.582451	1.428480	162.6945	489353	

In [ ]:

```
ui.info()
```

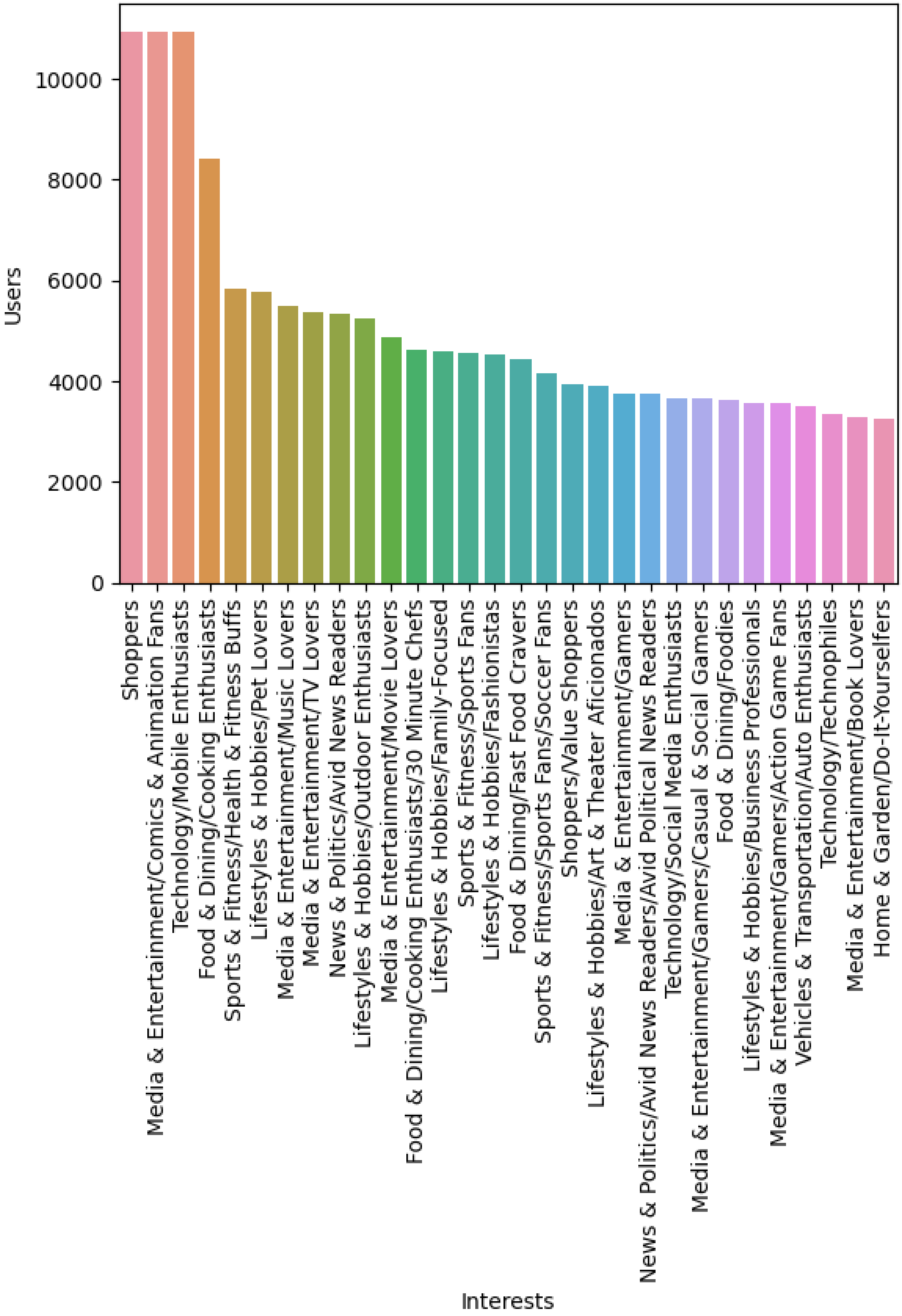
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 89 entries, 0 to 88
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Interests                  89 non-null     object 
 1   Users                      89 non-null     int64  
 2   New users                  89 non-null     int64  
 3   Engaged sessions           89 non-null     int64  
 4   Engagement rate            89 non-null     float64
 5   Engaged sessions per user  89 non-null     float64
 6   Average engagement time    89 non-null     float64
 7   Event count                89 non-null     int64  
 8   Conversions                89 non-null     int64  
 9   Total revenue              89 non-null     int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 7.6+ KB
```

In [ ]:

```python
sns.barplot(data=ui1,x='Interests',y='Users')
plt.xticks(rotation=90)
plt.figure(figsize=(30,60))
```

Out[ ]:

```
<Figure size 3000x6000 with 0 Axes>
```



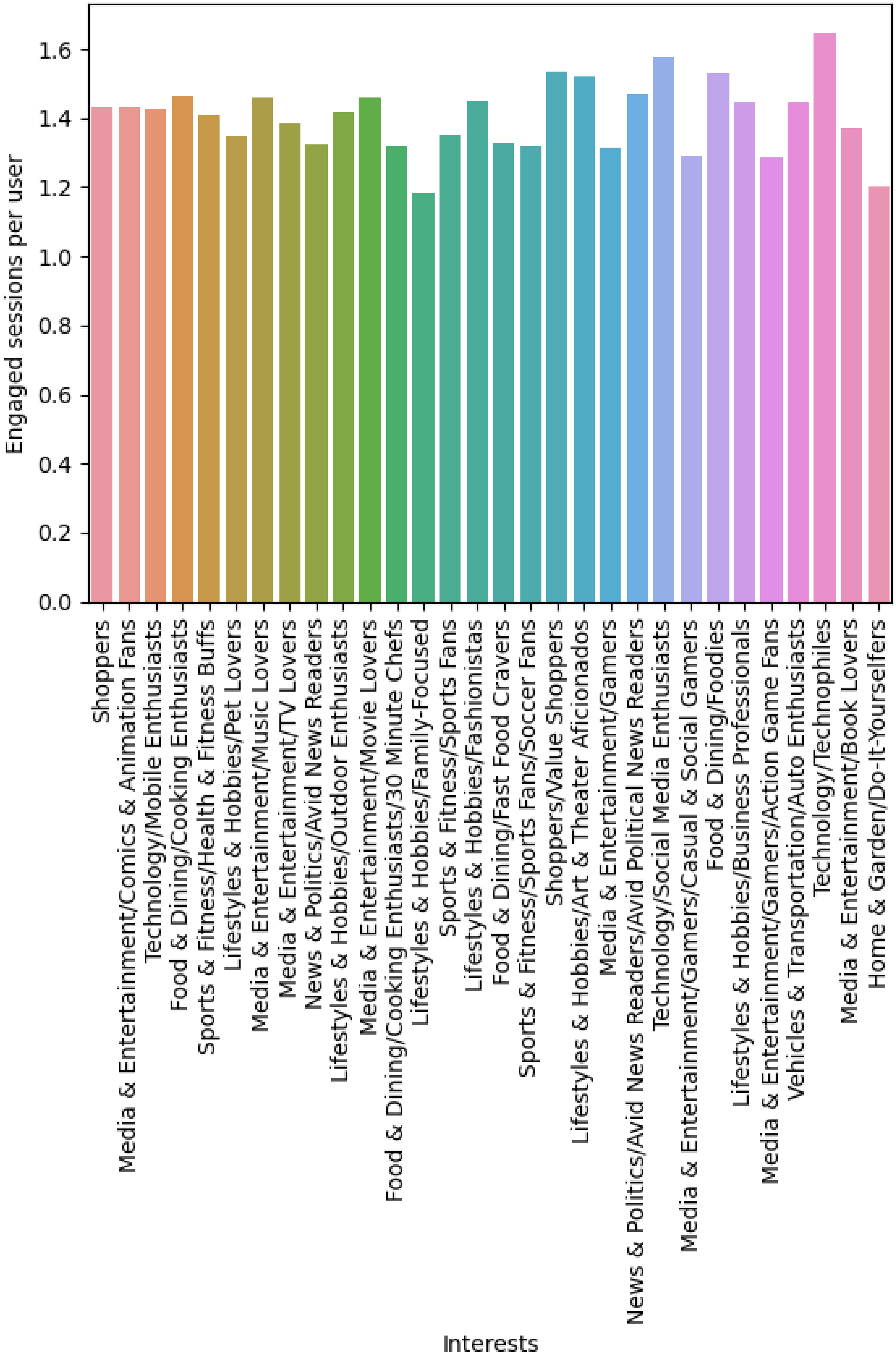
```
<Figure size 3000x6000 with 0 Axes>
```

In [ ]:

```
sns.barplot(data=ui1,x='Interests',y='Engaged sessions per user')
plt.xticks(rotation=90)
plt.figure(figsize=(30,60))
```

Out[ ]:

```
<Figure size 3000x6000 with 0 Axes>
```



```
<Figure size 3000x6000 with 0 Axes>
```

- Interest of user has negligible effect on Engaged Sessions per user
- Shoppers, Media & Entertainment/Comics & Animation Fans, Technology/Mobile Enthusiast, Food & Dining/Cooking Enthusiasts are the top most interest of the users
- This can be used to get more conversions from targeted ads

Analysis on Demographics Report

In [ ]:

```
dr = pd.read_excel(xls, 'Demographics Report')
dr.head(5)
```

Out[ ]:

	Country	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions
0	India	23024	22528	41479	0.593626	1.801555	334.81660	1312097	192766
1	United States	272	213	197	0.491272	0.724265	50.96324	3157	643
2	Canada	37	18	25	0.416667	0.675676	43.21622	410	121
3	(not set)	36	36	17	0.459459	0.472222	24.80556	241	54
4	United Kingdom	20	8	13	0.371429	0.650000	61.85000	289	43

In [ ]:

```
dr.info()
```

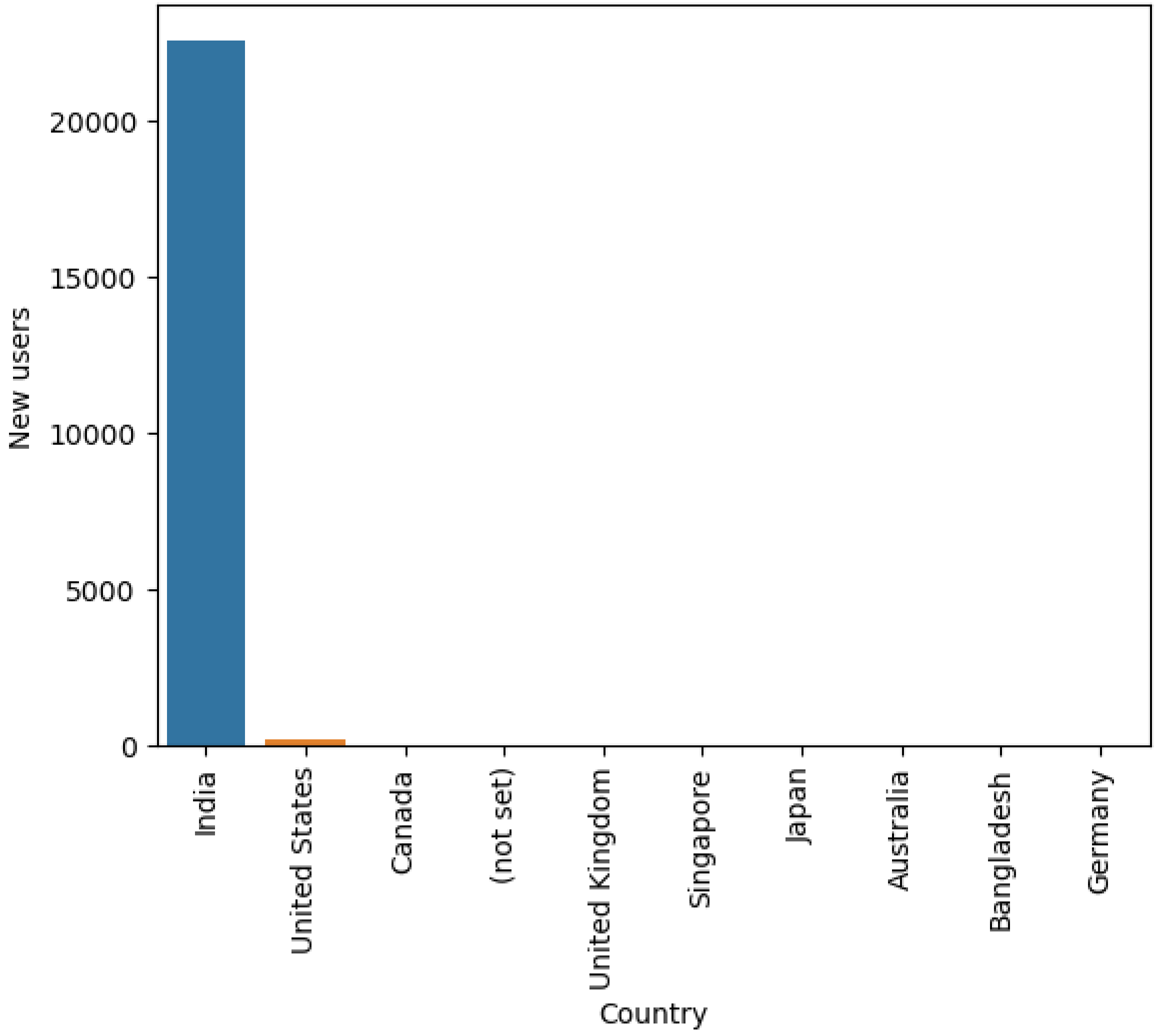
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47 entries, 0 to 46
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Country                    47 non-null     object 
 1   Users                      47 non-null     int64  
 2   New users                  47 non-null     int64  
 3   Engaged sessions           47 non-null     int64  
 4   Engagement rate            47 non-null     float64
 5   Engaged sessions per user  47 non-null     float64
 6   Average engagement time    47 non-null     float64
 7   Event count                47 non-null     int64  
 8   Conversions                47 non-null     int64  
 9   Total revenue              47 non-null     int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 3.8+ KB
```

In [ ]:

```
ss=dr.head(10)
sns.barplot(data=ss,x='Country',y='New users')
plt.xticks(rotation=90)
px.pie(ss,values='Users',names='Country')
plt.figure(figsize=(30,50))
```

Out[ ]:

```
<Figure size 3000x5000 with 0 Axes>
```



```
<Figure size 3000x5000 with 0 Axes>
```

In [ ]:

```
px.pie(ss,values='Conversions',names='Country')
```

- most of the users are from India
- And some from United States
- Other countries has almost negligible users

Analysis on Citiwise Report

In [ ]:

```
cr= pd.read_excel(xls, 'Citiwise Report')
cr.head(5)
```

Out[ ]:

	Town/City	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions
0	Bengaluru	6097	5685	15013	0.769385	2.462359	762.20550	607200	62939
1	Patna	1594	1467	2127	0.440646	1.334379	98.22208	38830	6980
2	Hyderabad	1038	920	1578	0.569264	1.520231	243.69080	96826	34103
3	Indore	983	915	1241	0.426460	1.262462	67.89115	21383	4121
4	Lucknow	897	839	1125	0.450180	1.254181	83.40580	21041	3650

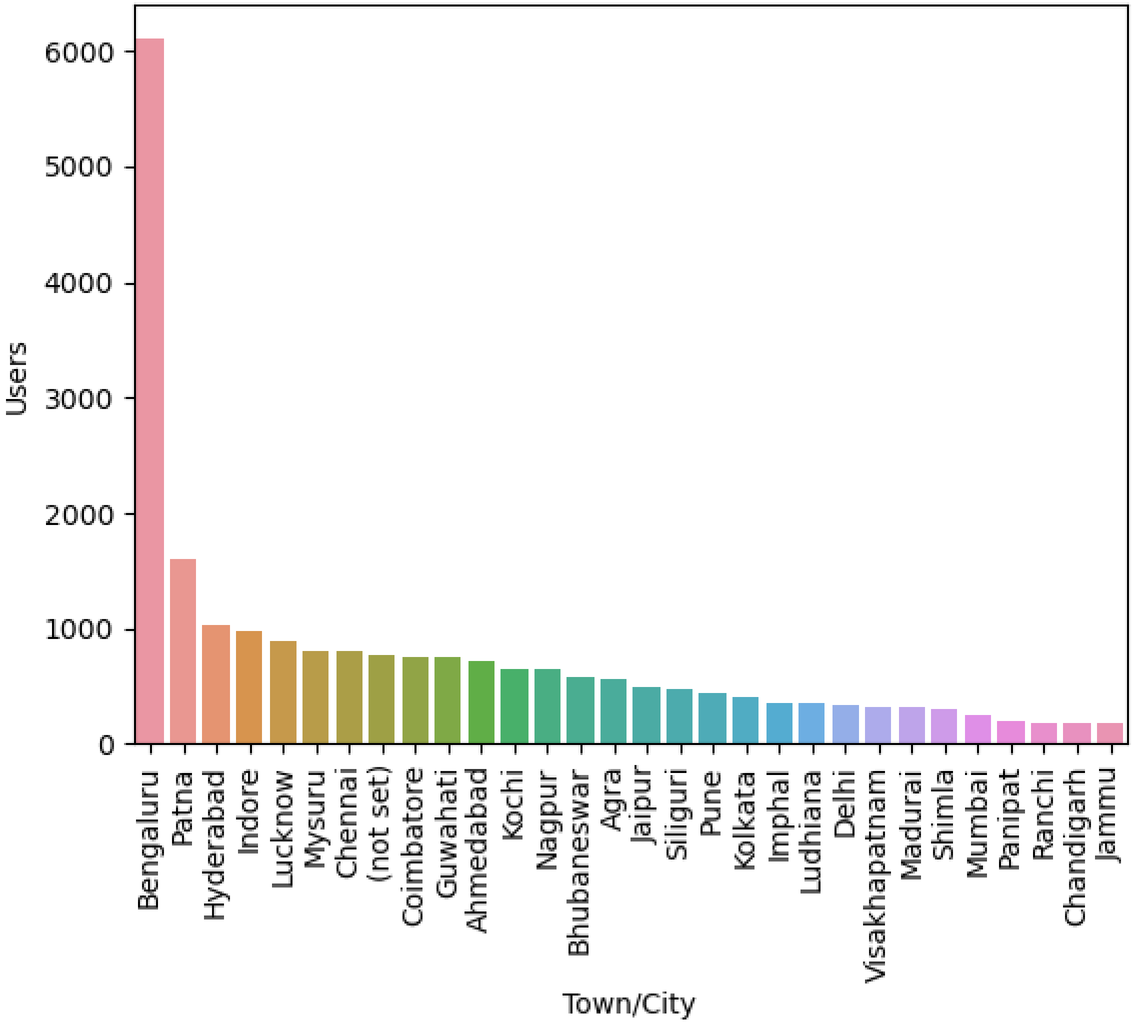
In [ ]:

```
cr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574 entries, 0 to 573
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Town/City                   574 non-null    object 
 1   Users                       574 non-null    int64  
 2   New users                   574 non-null    int64  
 3   Engaged sessions            574 non-null    int64  
 4   Engagement rate             574 non-null    float64
 5   Engaged sessions per user   574 non-null    float64
 6   Average engagement time     574 non-null    float64
 7   Event count                 574 non-null    int64  
 8   Conversions                 574 non-null    int64  
 9   Total revenue               574 non-null    int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 45.0+ KB
```

In [ ]:

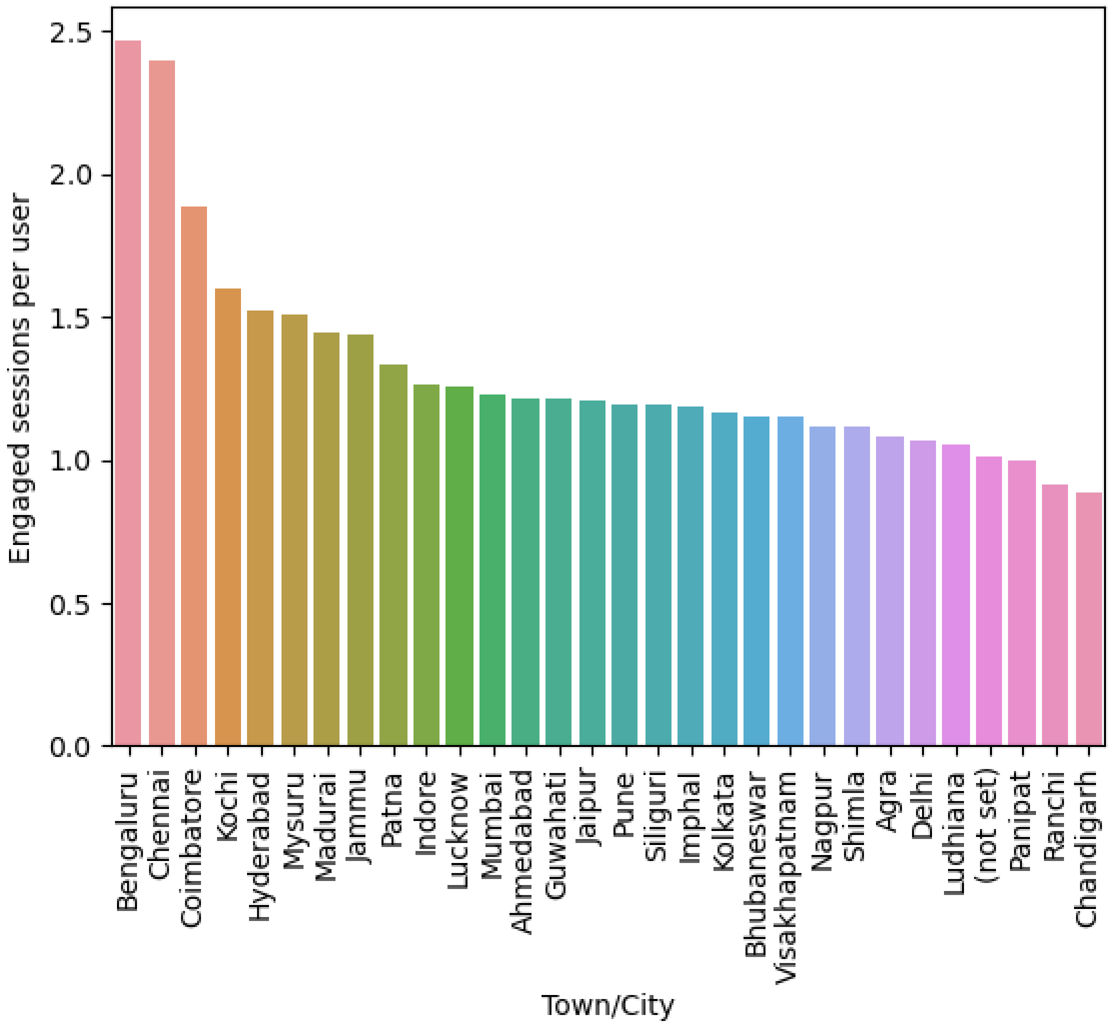
```
cr1=cr.head(30)
cr1=cr1.sort_values(by=['Users'],ascending=False)
sns.barplot(data=cr1,x='Town/City',y='Users')
plt.xticks(rotation=90)
plt.show()
```



In [ ]:

```
cr2=cr.head(30)
cr2=cr2.sort_values(by=['Engaged sessions per user'],ascending=False)
sns.barplot(data=cr2,x='Town/City',y='Engaged sessions per user')
```

```
plt.xticks(rotation=90)
plt.show()
```



- Bangaluru has highst Engaged sessions per user
- Bangaluru ,Chennai ,Coimbatore , Kochi has Engaged sessions per user is more than 1.5 which we can consider high

- Most of the users are from Bangalore

Analysis on User by Language report

In [ ]:
```
ul= pd.read_excel(xls, 'User by Language')
ul.head(5)
```

Out[ ]:

	Language	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions
0	English	22495	21990	40639	0.595147	1.806579	341.36350	1297970	189946
1	Hindi	586	552	798	0.406314	1.361775	60.03413	13523	2699
2	Marathi	85	84	98	0.426087	1.152941	38.48235	1589	323
3	Gujarati	78	77	100	0.448430	1.282051	46.53846	1794	327
4	Telugu	43	42	56	0.455285	1.302326	36.65116	812	170

In [ ]:

```
ul.info()
```

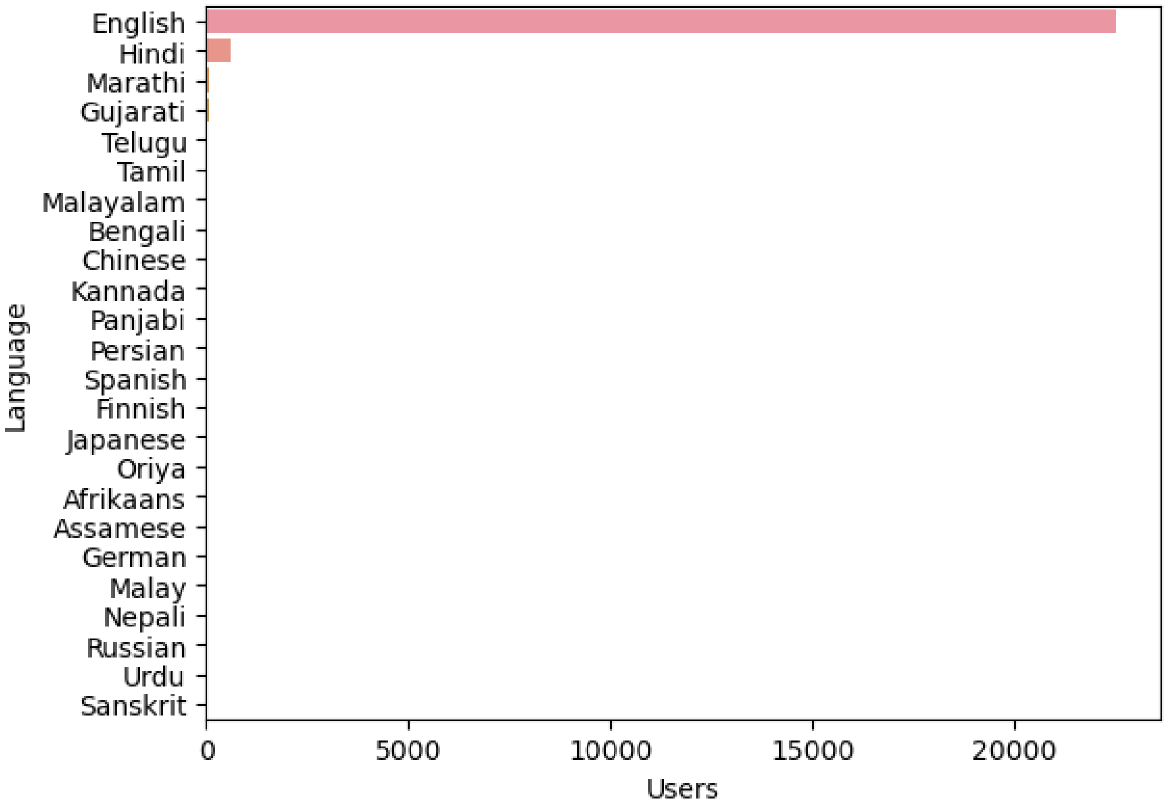
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Language                   24 non-null     object 
 1   Users                      24 non-null     int64  
 2   New users                  24 non-null     int64  
 3   Engaged sessions           24 non-null     int64  
 4   Engagement rate            24 non-null     float64
 5   Engaged sessions per user  24 non-null     float64
 6   Average engagement time    24 non-null     float64
 7   Event count                24 non-null     int64  
 8   Conversions                24 non-null     int64  
 9   Total revenue              24 non-null     int64  
dtypes: float64(3), int64(6), object(1)
memory usage: 2.0+ KB
```

In [ ]:

```
sns.barplot(data=ul,x='Users',y='Language')
```

Out[ ]:

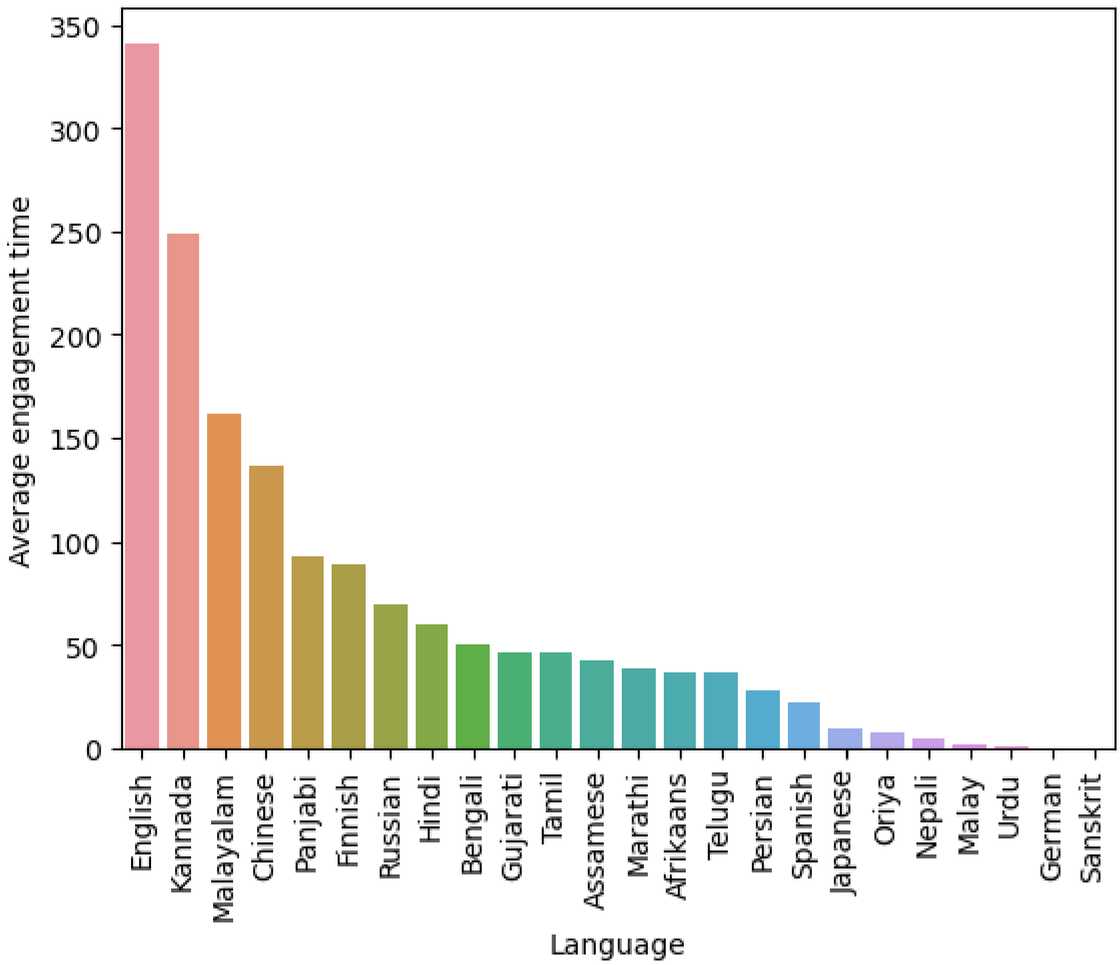
```
<AxesSubplot:xlabel='Users', ylabel='Language'>
```



```
In [ ]: # users distribution for top 5 languages
        ul1=ul.head(5)
        px.pie(ul1,values='Users',names='Language')
```

- Most of the users use English 96.6%
- 2.52% users speak Hindi

```
In [ ]: ul2=ul.sort_values(by=['Average engagement time'],ascending=False)
        sns.barplot(data=ul2,x='Language',y='Average engagement time')
        plt.xticks(rotation=90)
        plt.show()
```



- Average engagement time is highest for English
- language like Kannada, Malayalam ,Chinese has high Average engagement time

Analysis on User Interest report

```
In [ ]: ui= pd.read_excel(xls, 'User By Interest')
        ui=ui.sort_values(by=['Users'],ascending=False)
```

```
In [ ]: ui1=ui.head(30)
```

```
In [ ]: ui1.head(5)
```

Out[ ]:

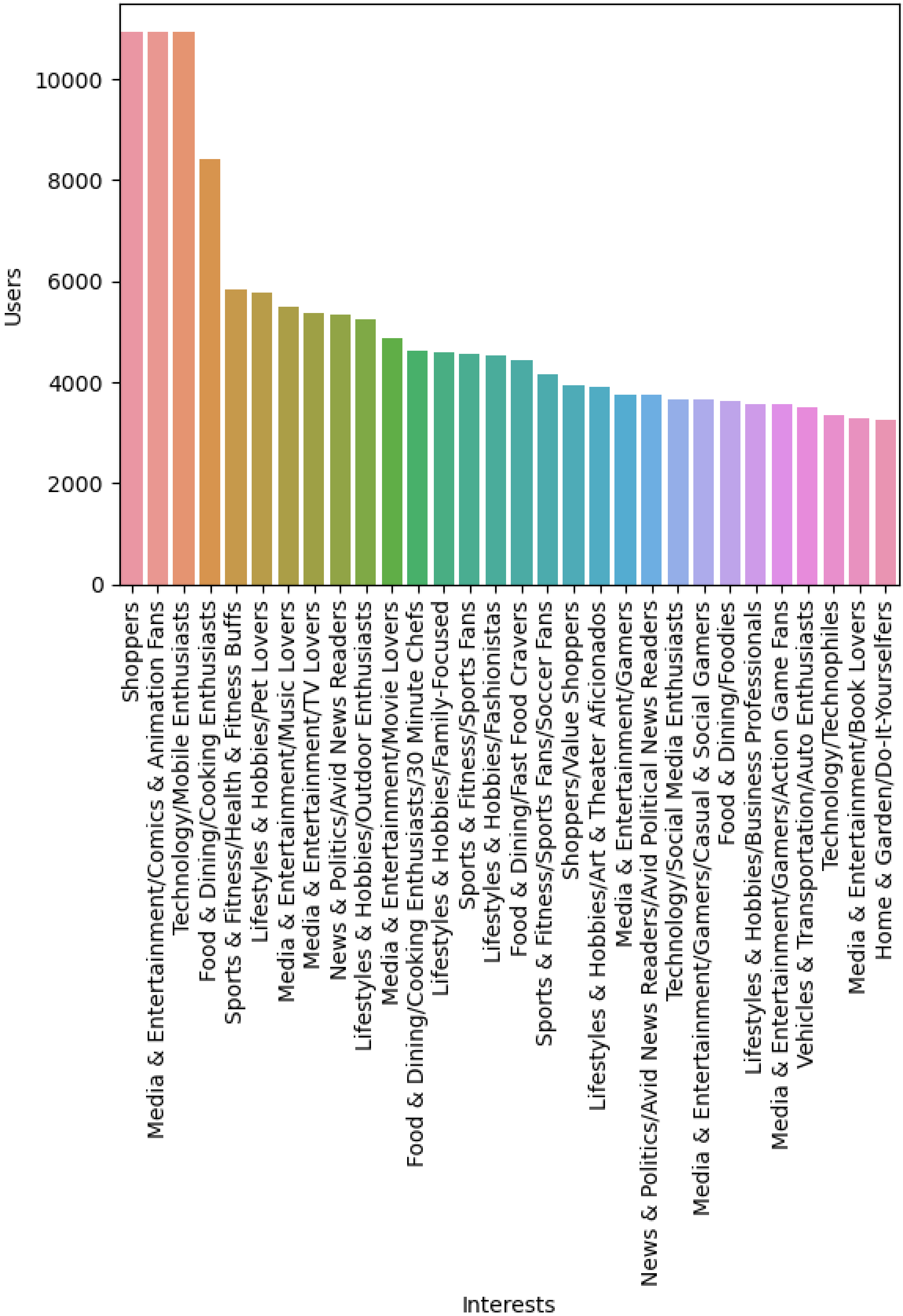
	Interests	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Cor
0	Shoppers	10950	9256	15652	0.581534	1.429406	162.8347	490664	
1	Media & Entertainment/Comics & Animation Fans	10946	9247	15680	0.583008	1.432487	165.1772	491025	
2	Technology/Mobile Enthusiasts	10934	9239	15619	0.582451	1.428480	162.6945	489353	
3	Food & Dining/Cooking Enthusiasts	8410	6970	12332	0.602325	1.466350	176.9567	409713	
4	Sports & Fitness/Health & Fitness Buffs	5844	4580	8226	0.588328	1.407598	155.1451	257831	

In [ ]:

```
sns.barplot(data=ui1,x='Interests',y='Users')
plt.xticks(rotation=90)
plt.figure(figsize=(30,60))
```

Out[ ]:

```
<Figure size 3000x6000 with 0 Axes>
```



```
<Figure size 3000x6000 with 0 Axes>
```

- Shoppers, Media & Entertainment/Comics & Animation Fans, Technology/Mobile Enthusiasts , Food & Dining/Cooking Enthusiasts ,these are the most common interest of the users . This can be more useful targeted ads and Dynamic Landing Pages and Strategic Partnerships and Collaborations

Marketing Campaign Analysis:

Analysis on Google Ads Report

In [ ]:

```
ga= pd.read_excel(xls, 'Google Ads Report')
ga.head(5)
```

Out[ ]:

	Session Google Ads campaign	Users	Sessions	Engaged sessions	Google Ads clicks	Google Ads cost	Google Ads cost per click	Conversions	Cost per conversion
0	App Installation for May -- Shahid	5429	10936	6276	147100	179175.000	1.218049	12257	14.618180
1	App Install-States-A200Inst-20Jun22	842	1655	968	28742	24309.130	0.845770	1794	13.550240
2	App Install-States-B100Installs-22Jun22	742	1332	780	17809	22374.580	1.256363	1422	15.734580
3	App Install for April -- Shahid	473	976	546	19302	20525.180	1.063370	1115	18.408230
4	Video-AppInstall-PS-Internships-11Jul22	510	966	515	9831	6377.833	0.648747	1032	6.180071

In [ ]:

```
ga.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Session Google Ads campaign  15 non-null     object 
 1   Users                        15 non-null     int64  
 2   Sessions                     15 non-null     int64  
 3   Engaged sessions             15 non-null     int64  
 4   Google Ads clicks            15 non-null     int64  
 5   Google Ads cost              15 non-null     float64
 6   Google Ads cost per click    15 non-null     float64
 7   Conversions                  15 non-null     int64  
 8   Cost per conversion          15 non-null     float64
 9   Event count                  15 non-null     int64  
 10  Total revenue                15 non-null     int64  
 11  Return on ad spend           15 non-null     int64  
dtypes: float64(3), int64(8), object(1)
memory usage: 1.5+ KB
```

In [ ]:

```
# creating a columns for Money Spent in each a campaign
ga['money_spent']=ga['Conversions']*ga['Cost per conversion']
ga.head(5)
```

Out[ ]:

	Session Google Ads campaign	Users	Sessions	Engaged sessions	Google Ads clicks	Google Ads cost	Google Ads cost per click	Conversions	Cost per conversion
0	App Installation for May -- Shahid	5429	10936	6276	147100	179175.000	1.218049	12257	14.618180
1	App Install-States-A200Inst-20Jun22	842	1655	968	28742	24309.130	0.845770	1794	13.550240
2	App Install-States-B100Installs-22Jun22	742	1332	780	17809	22374.580	1.256363	1422	15.734580
3	App Install for April -- Shahid	473	976	546	19302	20525.180	1.063370	1115	18.408230
4	Video-AppInstall-PS-Internships-11Jul22	510	966	515	9831	6377.833	0.648747	1032	6.180071

In [ ]:

```python
# Creating a column to judge cost effectiveness of Campign
ga['cost_effictiveness']=ga['Cost per conversion']-sum(ga['money_spent'])/sum(ga['C
```

In [ ]:

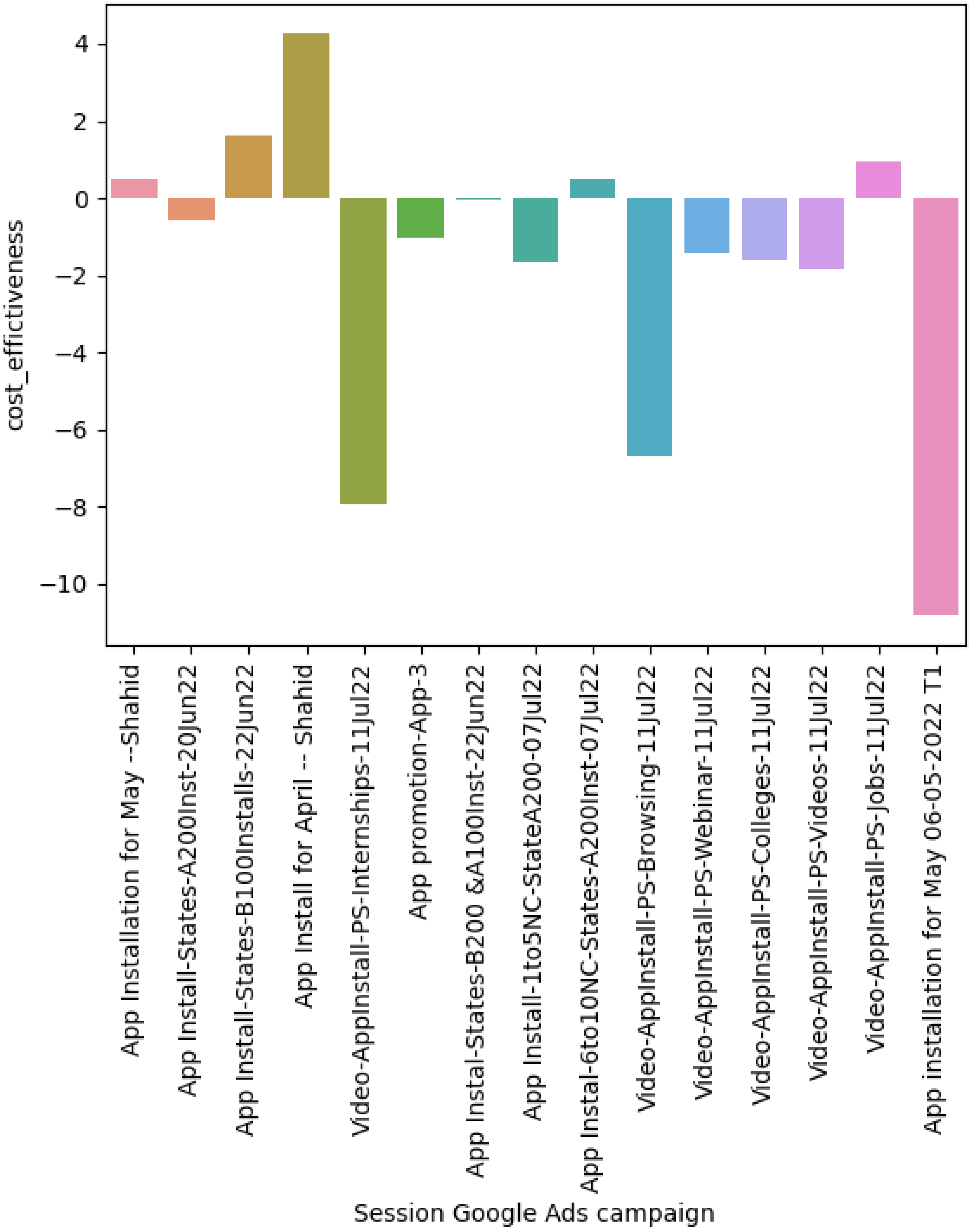
```python
ga.head(5)
```

Out[ ]:

	Session Google Ads campaign	Users	Sessions	Engaged sessions	Google Ads clicks	Google Ads cost	Google Ads cost per click	Conversions	Cost per conversion
0	App Installation for May -- Shahid	5429	10936	6276	147100	179175.000	1.218049	12257	14.618180
1	App Install-States-A200Inst-20Jun22	842	1655	968	28742	24309.130	0.845770	1794	13.550240
2	App Install-States-B100Installs-22Jun22	742	1332	780	17809	22374.580	1.256363	1422	15.734580
3	App Install for April -- Shahid	473	976	546	19302	20525.180	1.063370	1115	18.408230
4	Video-AppInstall-PS-Internships-11Jul22	510	966	515	9831	6377.833	0.648747	1032	6.180071

In [ ]:

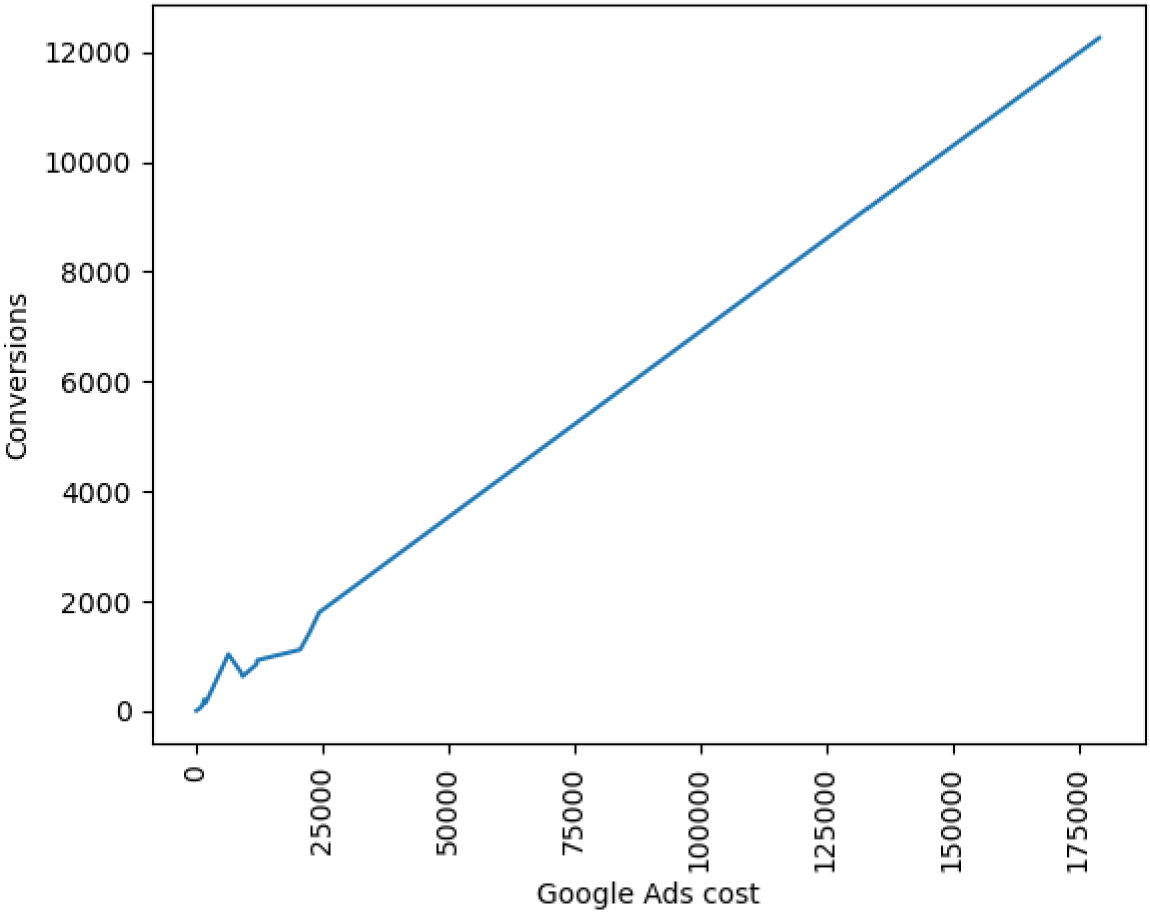
```
sns.barplot(data=ga,x='Session Google Ads campaign',y='cost_effictiveness')
plt.xticks(rotation=90)
plt.show()
```



- The most successful campaign was >- 'App Install for April -- Shahid' in terms of cost effectiveness.
- App installation for May 06-05-2022 T1 as the least successful
- App Instal-6to10NC-States-A200Inst-07Jul22 , Video-AppInstall-PS-Browsing-11Jul22 was significantly less successful in terms of cost effectiveness

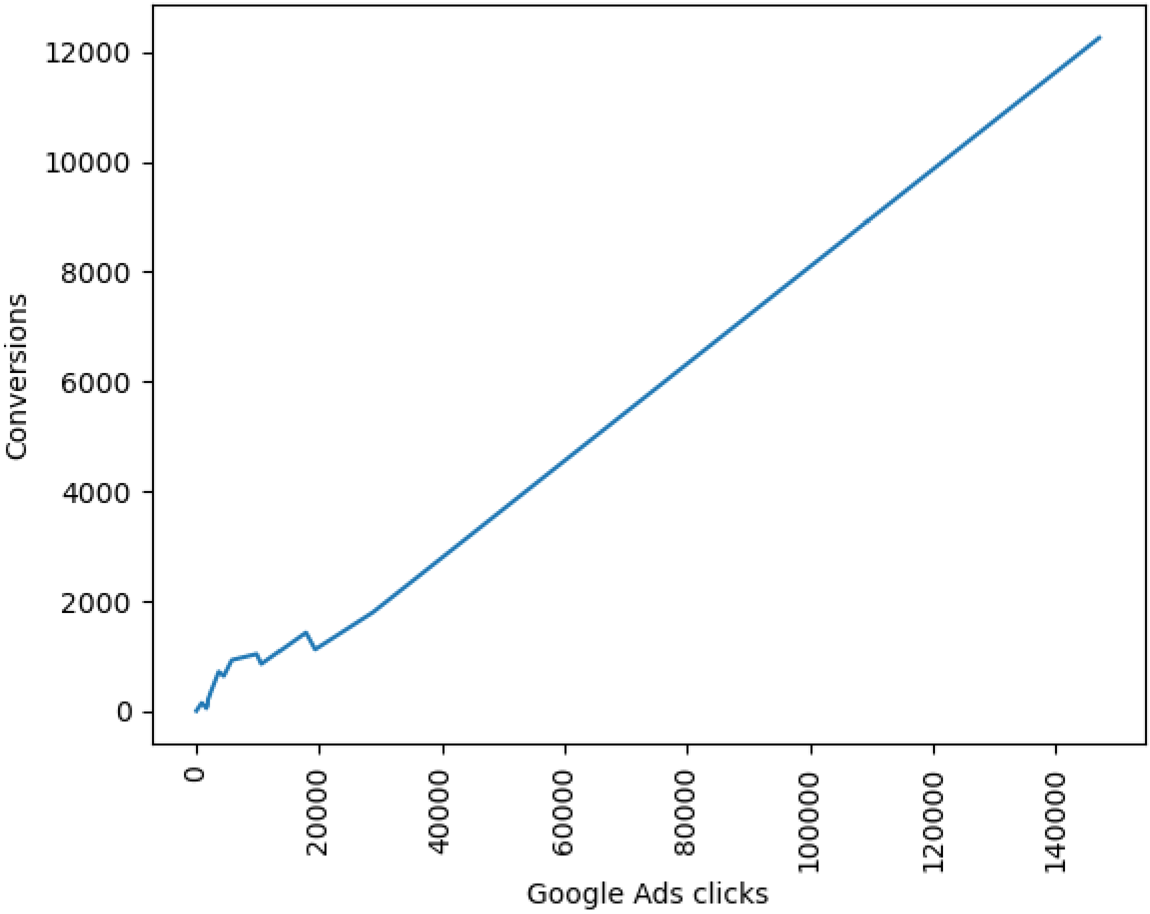
In [ ]:

```
sns.lineplot(data=ga,x='Google Ads cost',y='Conversions')
plt.xticks(rotation=90)
plt.show()
```



In [ ]:

```
sns.lineplot(data=ga,x='Google Ads clicks',y='Conversions')
plt.xticks(rotation=90)
plt.show()
```



- the ad clicks and conversions plot is linier ,that means most of the cunsumer, who clicks ad gets converted
- same with the amount spent on campaign

Analysis on Traffic Aquisition

In [ ]:

```
ta= pd.read_excel(xls, 'Traffic Aquisition')
ta.head(5)
```

Out[ ]:

	Session default channel group	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count
0	Unassigned	20263	13448	1481	34.11704	0.073089	18.023130	0.110128	242375
1	Display	9613	18292	10613	28.52198	1.104026	9.069320	0.580199	165896
2	Organic Search	7689	21241	17814	195.94340	2.316816	29.302290	0.838661	622410
3	Direct	4042	13220	7649	177.17060	1.892380	17.135850	0.578593	226536
4	Paid Search	2909	6788	3452	36.65321	1.186662	8.989982	0.508544	61024

In [ ]:

```
ta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 11 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   Session default channel group        6 non-null      object 
 1   Users                                6 non-null      int64  
 2   Sessions                             6 non-null      int64  
 3   Engaged sessions                     6 non-null      int64  
 4   Average engagement time per session  6 non-null      float64
 5   Engaged sessions per user            6 non-null      float64
 6   Events per session                   6 non-null      float64
 7   Engagement rate                      6 non-null      float64
 8   Event count                          6 non-null      int64  
 9   Conversions                          6 non-null      int64  
 10  Total revenue                        6 non-null      int64  
dtypes: float64(4), int64(6), object(1)
memory usage: 656.0+ bytes
```

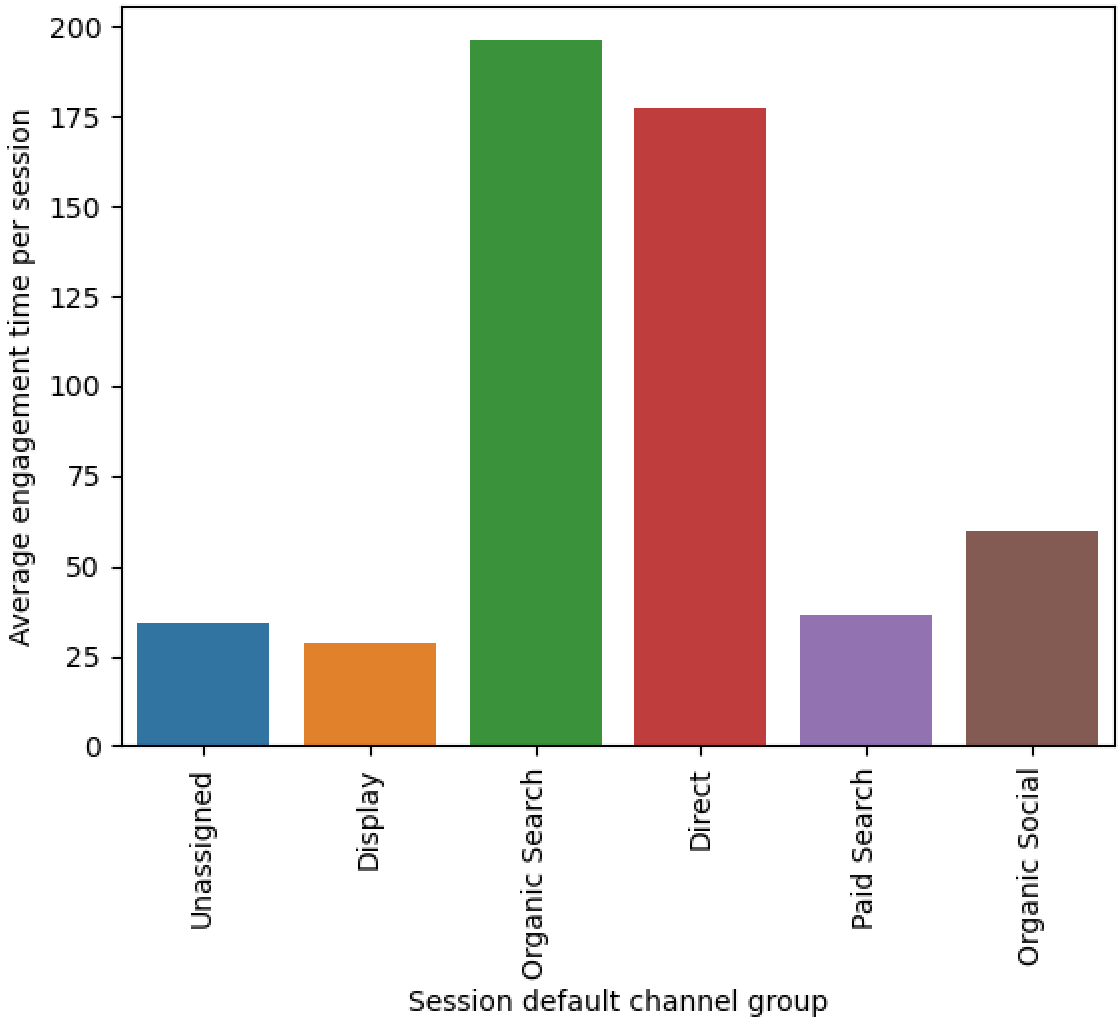
In [ ]:

```
px.pie(ta,values='Users',names='Session default channel group')
```

- Display is the highest contributor for traffic aquisition followed bu organic search
- Paid search contributes only 6.53% of total traffic
- Paid search is good for aquiring new users but not for traffic aquisition

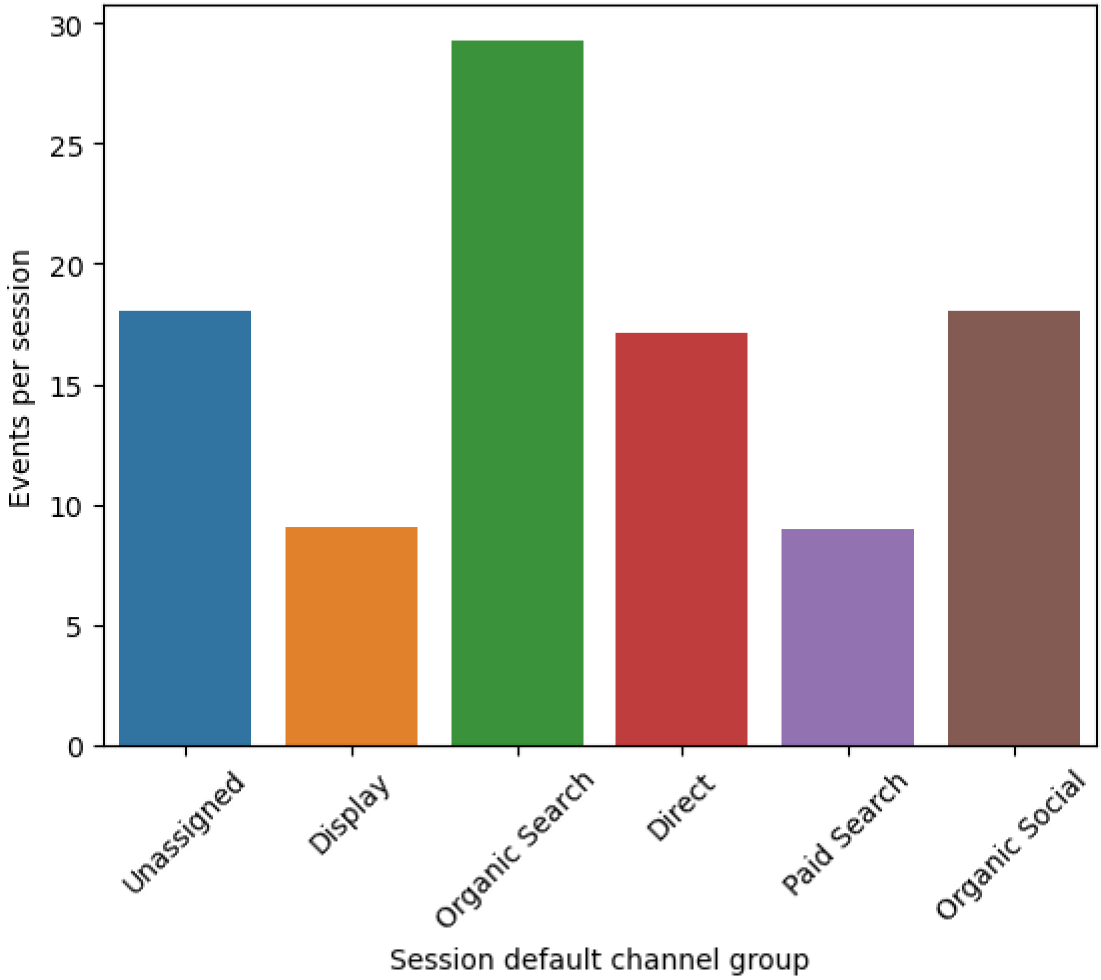
In [ ]:

```
sns.barplot(data=ta,x='Session default channel group',y='Average engagement time pe
plt.xticks(rotation=90)
plt.show()
```



- Organic Search and Direct has more Average engagement time per session even though they have 17% and 9% traffic contribution

```
In [ ]: sns.barplot(data=ta,y='Events per session',x='Session default channel group')
        plt.xticks(rotation=45)
        plt.show()
```



- when people organic search they explore the app more
- in case of paid and display type channel group the Events per session ,they browse less which has less possibilty for fequent user

Business Insights And Key Findings :

User Installation & Engagement Performance Analysis:

- Users who install the app at least start a session, which means they at least try the app.
- There are 23,254 users, and 18,030 users removed the app, which is a high number.
- Only 1,700 users receive notifications.

Analysis on Gender Report:

- Male customers are more in number than female customers.
- Despite that, the average engagement time for females is more.

Analysis on Age Data:

- Most of the users belong to the 18-24 category.
- The 18-35 age group constitutes 60% of the total users; however, the 18-24 age group has the highest average engagement time.

Analysis on User Interest Report:

- Interest of users has a negligible effect on Engaged Sessions per user.
- Shoppers, Media & Entertainment/Comics & Animation Fans, Technology/Mobile Enthusiasts, Food & Dining/Cooking Enthusiasts are the top interests of the users.

Analysis on Demographics Report:

- Most of the users are from India, and some are from the United States.
- Other countries have almost negligible users.
- Bangalore has the highest Engaged sessions per user.
- Bangalore, Chennai, Coimbatore, Kochi have Engaged sessions per user more than 1.5, which we can consider high.
- Most of the users are from Bangalore.
- Most of the users use English (96.6%).
- 2.52% of users speak Hindi.
- Average engagement time is highest for English.
- Languages like Kannada, Malayalam, Chinese have high Average engagement time.
- Shoppers, Media & Entertainment/Comics & Animation Fans, Technology/Mobile Enthusiasts, Food & Dining/Cooking Enthusiasts, these are the most common interests of the users. This can be more useful for targeted ads, Dynamic Landing Pages, and Strategic Partnerships and Collaborations.

Marketing Campaign Analysis:

- The most successful campaign was 'App Install for April -- Shahid' in terms of cost-effectiveness.
- App installation for May 06-05-2022 T1 was the least successful.
- App Install-6to10NC-States-A200Inst-07Jul22, Video-AppInstall-PS-Browsing-11Jul22 was significantly less successful in terms of cost-effectiveness.
- The ad clicks and conversions plot is linear, that means most of the consumers who click ads get converted.
- Same with the amount spent on the campaign.
- Display is the highest contributor to traffic acquisition followed by organic search.
- Paid search contributes only 6.53% of total traffic.
- Paid search is good for acquiring new users but not for traffic acquisition.
- Organic Search and Direct have more Average engagement time per session even though they have 17% and 9% traffic contribution.

Business Recommendations

1. User Engagement Enhancement Strategy

- Develop a notification strategy to engage users after installation. Since only a small percentage currently receives notifications, explore ways to increase this number.
- Investigate the reasons behind the high number of app removals. Conduct user feedback surveys to understand pain points and areas for improvement.
- Develop targeted retention campaigns to encourage users to revisit the app. Consider offering incentives or exclusive content to re-engage dormant users. ##### Gender

Based Engagement Strategy

- Despite a higher number of male users, focus on creating content that resonates with female users. Leverage insights from the higher average engagement time for females to tailor content and features ##### Age-Group Targeting: Recognize the importance of the 18-24 age group in terms of both population and engagement time. Develop targeted engagement strategies, exclusive content, and features to cater to this demographic. ##### Targeted Advertising
- Leverage the identified top interests (Shoppers, Media & Entertainment/Comics & Animation Fans, Technology/Mobile Enthusiasts, Food & Dining/Cooking Enthusiasts) for more effective targeted advertising.

- Implement dynamic landing pages and collaborate with partners aligned with these interests to enhance user engagement. ##### Region Based Strategy

- Given the concentration of users in Bangalore, Chennai, Coimbatore, and Kochi, consider tailoring marketing efforts and content for these regions.
- Utilize the language insights to optimize content delivery and possibly introduce localized features.

2. Marketing Campaign Optimization

- Analyze the successful 'App Install for April -- Shahid' campaign to understand what worked well. Apply similar strategies to future campaigns.

- Reevaluate and optimize less successful campaigns, such as 'App Instal-6to10NC-States-A200Inst-07Jul22' and 'Video-AppInstall-PS-Browsing-11Jul22,' to improve cost-effectiveness.

- Acknowledge the effectiveness of Display and Organic Search for traffic acquisition. Allocate resources accordingly and consider increasing investment in these channels.

- Assess the performance of Paid Search and explore ways to improve its effectiveness for both user acquisition and engagement.