promilo-assignment-for-ba

March 2, 2024

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

Sheets and Key Variables:

User Acquisition: Key variables include user acquisition channels, new users, engagement metrics, and conversions.

Traffic Acquisition: Focuses on session information, user engagement, and acquisition channels.

Event Report: Contains event-specific data such as event count and user interaction metrics.

Conversion Report: Provides data on conversions related to specific events.

Pages & Screens Report: Details user interactions with specific pages or screens within the app or website.

Demographics Report: Offers insights into user demographics such as country, user engagement, and conversions.

Google Ads Report: Analyzes the performance of Google Ads campaigns, including sessions, user engagement, ad spend, and conversions.

[4]:

```
import pandas as pd
# Load the new dataset
data_set_path = 'Data set for BA.xlsx'
xlsx = pd.ExcelFile(data_set_path)

# Get the names of all sheets in the Excel file to understand the structure
sheet_names = xlsx.sheet_names

# Load each sheet into a dictionary to facilitate exploration and␣
 ↪identification of key variables
sheets_data = {sheet_name: pd.read_excel(xlsx, sheet_name=sheet_name) for␣
 ↪sheet_name in sheet_names}

# Data Exploration: Display the first few rows of each sheet to understand its␣
 ↪structure and identify key variables
for sheet_name, data in sheets_data.items():
    print(f"Sheet Name: {sheet_name}")
    display(data.head())
    print("\nKey Variables:", data.columns.tolist())
```

```python
    print("-" * 100)

# Initial assessment for cleaning and preprocessing needs
cleaning_summary = {}

for sheet_name, data in sheets_data.items():
    # Identify columns with missing values
    missing_values = data.isnull().sum()
    missing_columns = missing_values[missing_values > 0].index.tolist()

    # Data formatting checks
    object_columns = data.select_dtypes(include=['object']).columns.tolist()

    cleaning_summary[sheet_name] = {
        "Missing Value Columns": missing_columns,
        "Columns for Formatting Check": object_columns
    }

# Display summary of cleaning and preprocessing needs
cleaning_summary
```

```
Sheet Name: Report Snapshot

Empty DataFrame
Columns: []
Index: []


Key Variables: []
--------------------------------------------------------------------------------
--------------------
Sheet Name: User Acquisition

  First user default channel group  New users  Engaged sessions  \
0                          Display       9957             12008
1                   Organic Search       7652             18141
2                      Paid Search       3025              4408
3                           Direct       1903              4975
4                       Unassigned        325              1619

   Engagement rate  Engaged sessions per user  Average engagement time  \
0         0.544457                   1.206107                 58.86209
1         0.813680                   2.367041                534.31280
2         0.474284                   1.458154                102.23780
3         0.318808                   2.261364               1128.88100
4         0.813159                   4.981538                798.34150

   Event count  Conversions  Total revenue
0       204820        37434              0
```

```
1       770710       109801             0
2        81997        14770             0
3       227434        31093             0
4        33320          789             0

Key Variables: ['First user default channel group', 'New users', 'Engaged
sessions', 'Engagement rate', 'Engaged sessions per user', 'Average engagement
time', 'Event count', 'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Traffic Aquisition
  Session default channel group  Users  Sessions  Engaged sessions  \
0                    Unassigned  20263     13448              1481   
1                       Display   9613     18292             10613   
2                Organic Search   7689     21241             17814   
3                        Direct   4042     13220              7649   
4                   Paid Search   2909      6788              3452   

   Average engagement time per session  Engaged sessions per user  \
0                             34.11704                   0.073089   
1                             28.52198                   1.104026   
2                            195.94340                   2.316816   
3                            177.17060                   1.892380   
4                             36.65321                   1.186662   

   Events per session  Engagement rate  Event count  Conversions  \
0           18.023130         0.110128       242375       114161   
1            9.069320         0.580199       165896        20031   
2           29.302290         0.838661       622410        33612   
3           17.135850         0.578593       226536        18496   
4            8.989982         0.508544        61024         7595   

   Total revenue  
0              0  
1              0  
2              0  
3              0  
4              0  

Key Variables: ['Session default channel group', 'Users', 'Sessions', 'Engaged
sessions', 'Average engagement time per session', 'Engaged sessions per user',
'Events per session', 'Engagement rate', 'Event count', 'Conversions', 'Total
revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Event Report
```

```
             Event name  Event count  Total users  Event count per user  \
0           screen_view       694729        23254             30.865870
1  notification_receive       125146         1700            138.896800
2       user_engagement       124836        22699              5.622230
3  notification_dismiss        70128         1369            144.000000
4         session_start        61163        23226              3.121357

   Total revenue
0              0
1              0
2              0
3              0
4              0

Key Variables: ['Event name', 'Event count', 'Total users', 'Event count per
user', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Conversion Report
              Event name  Conversions  Total users  Total revenue
0   notification_receive        94890         1311              0
1          session_start        56203        21674              0
2             first_open        22872        23059              0
3             app_remove        12468        12538              0
4  Promilo111_otp_screen         1738          855              0

Key Variables: ['Event name', 'Conversions', 'Total users', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Pages & Screens Report
  Page path and screen class   Views  Users  Views per user  \
0                    Flutter  156708   8726       17.958740
1               MainActivity   44326   8978        4.937180
2                      feeds   18514   4358        4.248279
3                      login   16883   7291        2.315595
4          my_rewards_screen   15381   2045        7.521271

   Average engagement time  Event count  Conversions  Total revenue
0                 83.41222       203901          328              0
1                 78.29216        53374          101              0
2                 61.60005        37628          253              0
3                 34.88177        40772          435              0
4                 94.17995        32910            5              0

Key Variables: ['Page path and screen class', 'Views', 'Users', 'Views per
user', 'Average engagement time', 'Event count', 'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Retention Overview

Empty DataFrame
Columns: []
Index: []


Key Variables: []
--------------------------------------------------------------------------------
--------------------
Sheet Name: User Engagement Overview

Empty DataFrame
Columns: []
Index: []


Key Variables: []
--------------------------------------------------------------------------------
--------------------
Sheet Name: Demographics Report

          Country  Users  New users  Engaged sessions  Engagement rate  \
0           India  23024      22528             41479         0.593626   
1   United States    272        213               197         0.491272   
2          Canada     37         18                25         0.416667   
3       (not set)     36         36                17         0.459459   
4  United Kingdom     20          8                13         0.371429   

   Engaged sessions per user  Average engagement time  Event count  \
0                   1.801555                334.81660      1312097   
1                   0.724265                 50.96324         3157   
2                   0.675676                 43.21622          410   
3                   0.472222                 24.80556          241   
4                   0.650000                 61.85000          289   

   Conversions  Total revenue  
0       192766              0  
1          643              0  
2          121              0  
3           54              0  
4           43              0  


Key Variables: ['Country', 'Users', 'New users', 'Engaged sessions', 'Engagement
rate', 'Engaged sessions per user', 'Average engagement time', 'Event count',
'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
```

```
--------------------
Sheet Name: Citiwise Report

   Town/City  Users  New users  Engaged sessions  Engagement rate  \
0  Bengaluru   6097       5685             15013         0.769385
1      Patna   1594       1467              2127         0.440646
2  Hyderabad   1038        920              1578         0.569264
3     Indore    983        915              1241         0.426460
4    Lucknow    897        839              1125         0.450180

   Engaged sessions per user  Average engagement time  Event count  \
0                   2.462359                762.20550       607200
1                   1.334379                 98.22208        38830
2                   1.520231                243.69080        96826
3                   1.262462                 67.89115        21383
4                   1.254181                 83.40580        21041

   Conversions  Total revenue
0        62939              0
1         6980              0
2        34103              0
3         4121              0
4         3650              0


Key Variables: ['Town/City', 'Users', 'New users', 'Engaged sessions',
'Engagement rate', 'Engaged sessions per user', 'Average engagement time',
'Event count', 'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Gender Report

    Gender  Users  New users  Engaged sessions  Engagement rate  \
0  unknown  13142      12691             23161         0.564077
1     male   7218       5877             10467         0.543091
2   female   4944       4304              7877         0.637710

   Engaged sessions per user  Average engagement time  Event count  \
0                   1.762365                 439.5776       761771
1                   1.450125                 128.2319       282504
2                   1.593244                 208.7407       274254

   Conversions  Total revenue
0        93180              0
1        65651              0
2        35083              0


Key Variables: ['Gender', 'Users', 'New users', 'Engaged sessions', 'Engagement
rate', 'Engaged sessions per user', 'Average engagement time', 'Event count',
'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: User By Interest

                                       Interests  Users  New users  \
0                                       Shoppers  10950       9256
1  Media & Entertainment/Comics & Animation Fans  10946       9247
2                  Technology/Mobile Enthusiasts  10934       9239
3              Food & Dining/Cooking Enthusiasts   8410       6970
4        Sports & Fitness/Health & Fitness Buffs   5844       4580

   Engaged sessions  Engagement rate  Engaged sessions per user  \
0             15652         0.581534                   1.429406
1             15680         0.583008                   1.432487
2             15619         0.582451                   1.428480
3             12332         0.602325                   1.466350
4              8226         0.588328                   1.407598

   Average engagement time  Event count  Conversions  Total revenue
0                 162.8347       490664        86846              0
1                 165.1772       491025        86845              0
2                 162.6945       489353        86742              0
3                 176.9567       409713        73814              0
4                 155.1451       257831        43074              0

Key Variables: ['Interests', 'Users', 'New users', 'Engaged sessions',
'Engagement rate', 'Engaged sessions per user', 'Average engagement time',
'Event count', 'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: User by Language

   Language  Users  New users  Engaged sessions  Engagement rate  \
0   English  22495      21990             40639         0.595147
1     Hindi    586        552               798         0.406314
2   Marathi     85         84                98         0.426087
3  Gujarati     78         77               100         0.448430
4    Telugu     43         42                56         0.455285

   Engaged sessions per user  Average engagement time  Event count  \
0                   1.806579                341.36350      1297970
1                   1.361775                 60.03413        13523
2                   1.152941                 38.48235         1589
3                   1.282051                 46.53846         1794
4                   1.302326                 36.65116          812

   Conversions  Total revenue
```

```
0       189946              0
1         2699              0
2          323              0
3          327              0
4          170              0

Key Variables: ['Language', 'Users', 'New users', 'Engaged sessions',
'Engagement rate', 'Engaged sessions per user', 'Average engagement time',
'Event count', 'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: User By Age
       Age  Users  New users  Engaged sessions  Engagement rate  \
0  unknown  14303      13636             24976         0.569098
1    18-24   4282       3678              7291         0.695308
2    25-34   2920       2161              3749         0.504780
3      65+   1422       1081              1640         0.539829
4    55-64   1403        979              1552         0.519411

   Engaged sessions per user  Average engagement time  Event count  \
0                   1.746207                422.22330       817501
1                   1.702709                251.16300       309328
2                   1.283904                 97.24144        90074
3                   1.153305                 52.30661        24780
4                   1.106201                 55.37063        25169

   Conversions  Total revenue
0        99310              0
1        53661              0
2        20172              0
3         4891              0
4         4823              0

Key Variables: ['Age', 'Users', 'New users', 'Engaged sessions', 'Engagement
rate', 'Engaged sessions per user', 'Average engagement time', 'Event count',
'Conversions', 'Total revenue']
--------------------------------------------------------------------------------
--------------------
Sheet Name: Google Ads Report
               Session Google Ads campaign  Users  Sessions  Engaged sessions  \
0        App Installation for May --Shahid   5429     10936              6276
1      App Install-States-A200Inst-20Jun22    842      1655               968
2  App Install-States-B100Installs-22Jun22    742      1332               780
3          App Install for April -- Shahid    473       976               546
4  Video-AppInstall-PS-Internships-11Jul22    510       966               515
```

```
   Google Ads clicks  Google Ads cost  Google Ads cost per click  Conversions  \
0             147100       179175.000                   1.218049        12257
1              28742        24309.130                   0.845770         1794
2              17809        22374.580                   1.256363         1422
3              19302        20525.180                   1.063370         1115
4               9831         6377.833                   0.648747         1032

   Cost per conversion  Event count  Total revenue  Return on ad spend
0            14.618180        97802              0                   0
1            13.550240        15311              0                   0
2            15.734580        11640              0                   0
3            18.408230         8001              0                   0
4             6.180071        10323              0                   0

Key Variables: ['Session Google Ads campaign', 'Users', 'Sessions', 'Engaged
sessions', 'Google Ads clicks', 'Google Ads cost', 'Google Ads cost per click',
'Conversions', 'Cost per conversion', 'Event count', 'Total revenue', 'Return on
ad spend']
--------------------------------------------------------------------------------
--------------------
```

```
[4]: {'Report Snapshot': {'Missing Value Columns': [],
       'Columns for Formatting Check': []},
      'User Acquisition': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['First user default channel group']},
      'Traffic Aquisition': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Session default channel group']},
      'Event Report': {'Missing Value Columns': ['Event name'],
       'Columns for Formatting Check': ['Event name']},
      'Conversion Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Event name']},
      'Pages & Screens Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Page path and screen class']},
      'Retention Overview': {'Missing Value Columns': [],
       'Columns for Formatting Check': []},
      'User Engagement Overview': {'Missing Value Columns': [],
       'Columns for Formatting Check': []},
      'Demographics Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Country']},
      'Citiwise Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Town/City']},
      'Gender Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Gender']},
      'User By Interest': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Interests']},
      'User by Language': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Language']},
      'User By Age': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Age']},
      'Google Ads Report': {'Missing Value Columns': [],
       'Columns for Formatting Check': ['Session Google Ads campaign']}}
```

[6]:

```
event_report_data = sheets_data['Event Report']
event_report_data.isnull().sum()
```

[6]:

```
Event name              1
Event count             0
Total users             0
Event count per user    0
Total revenue           0
dtype: int64
```

Cleaning Summary:

Most sheets have well-structured data with no significant issues regarding missing values.

Certain sheets like the “Event Report” have missing values in the ‘Event name’ column which needs attention(for example event names like null and (not set)).

Several columns across sheets are identified for formatting checks, mainly categorical variables like ‘First user default channel group’, ‘Session default channel group’, ‘Event name’, etc.

Cleaning and Preprocessing :

1.Missing ‘Event name’ values in the Event Report were filled with “Unknown” to maintain data integrity.

2.Categorical columns identified in the initial assessment have been converted to the ‘category’ data type to optimize memory usage and facilitate analysis.

[7]:

```
# Cleaning and Preprocessing
for sheet_name, data in sheets_data.items():
    # Fill missing 'Event name' values with "Unknown" in the Event Report
    if 'Event name' in data.columns:
        data['Event name'].fillna('Unknown', inplace=True)

    # Ensure correct data formatting
    # Convert categorical columns to category type
    if sheet_name in cleaning_summary:
        for col in cleaning_summary[sheet_name]['Columns for Formatting Check']:
            data[col] = data[col].astype('category')

    # Update the cleaned data back in the sheets_data dictionary
    sheets_data[sheet_name] = data

# Display the cleaned data types for verification
for sheet_name, data in sheets_data.items():
    print(f"Cleaned Data Types for Sheet: {sheet_name}")
    display(data.dtypes)
    print("-" * 100)
```

```
Cleaned Data Types for Sheet: Report Snapshot

Series([], dtype: object)

--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: User Acquisition

First user default channel group    category
New users                              int64
Engaged sessions                       int64
Engagement rate                      float64
Engaged sessions per user            float64
Average engagement time              float64
Event count                            int64
Conversions                            int64
Total revenue                          int64
dtype: object

--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Traffic Aquisition

Session default channel group          category
Users                                     int64
Sessions                                  int64
Engaged sessions                          int64
Average engagement time per session     float64
Engaged sessions per user               float64
Events per session                      float64
Engagement rate                         float64
Event count                               int64
Conversions                               int64
Total revenue                             int64
dtype: object

--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Event Report

Event name                category
Event count                  int64
Total users                  int64
Event count per user       float64
Total revenue                int64
dtype: object
```

```
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Conversion Report

Event name       category
Conversions         int64
Total users         int64
Total revenue       int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Pages & Screens Report

Page path and screen class    category
Views                            int64
Users                            int64
Views per user                 float64
Average engagement time        float64
Event count                      int64
Conversions                      int64
Total revenue                    int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Retention Overview

Series([], dtype: object)
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: User Engagement Overview

Series([], dtype: object)
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Demographics Report

Country                      category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
```

```
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Citiwise Report

Town/City                    category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Gender Report

Gender                       category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: User By Interest

Interests                    category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: User by Language
```

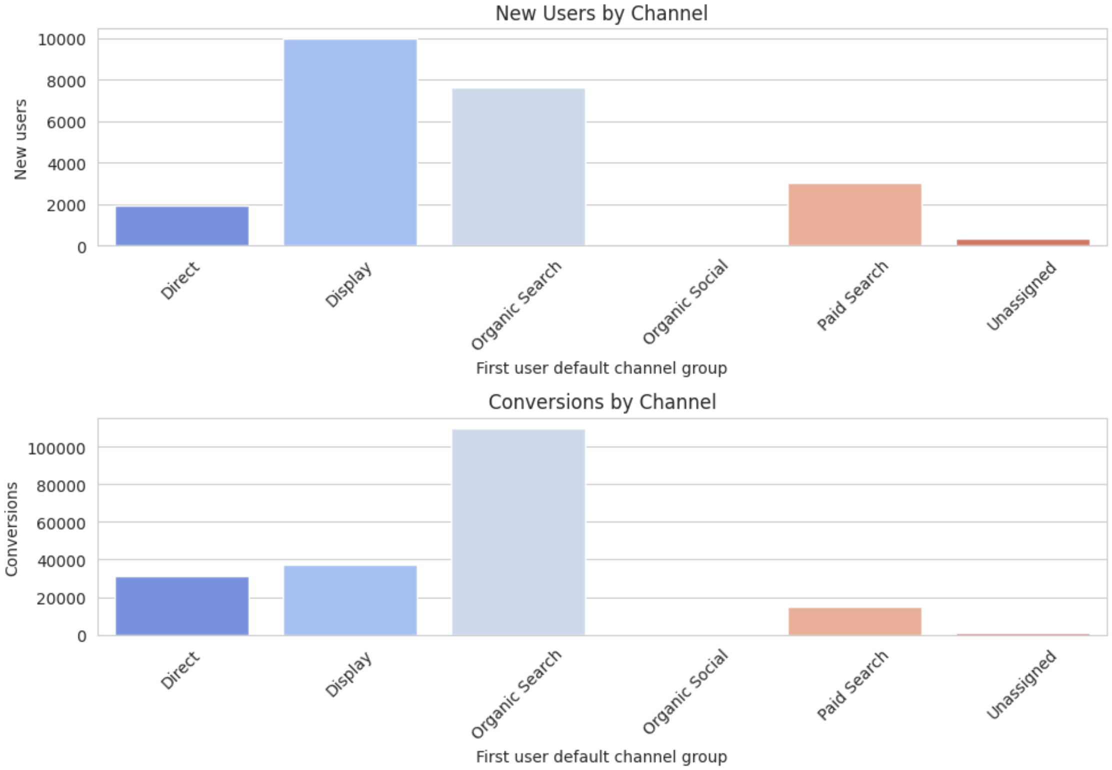
```
Language                     category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: User By Age

Age                          category
Users                           int64
New users                       int64
Engaged sessions                int64
Engagement rate               float64
Engaged sessions per user     float64
Average engagement time       float64
Event count                     int64
Conversions                     int64
Total revenue                   int64
dtype: object
--------------------------------------------------------------------------------
--------------------
Cleaned Data Types for Sheet: Google Ads Report

Session Google Ads campaign    category
Users                             int64
Sessions                          int64
Engaged sessions                  int64
Google Ads clicks                 int64
Google Ads cost                 float64
Google Ads cost per click       float64
Conversions                       int64
Cost per conversion             float64
Event count                       int64
Total revenue                     int64
Return on ad spend                int64
dtype: object
--------------------------------------------------------------------------------
--------------------
```

1. USER ACQUISITION

```
[29]: import matplotlib.pyplot as plt
      import seaborn as sns

      # Use the 'User Acquisition' sheet for analysis
      user_acquisition_data = sheets_data['User Acquisition']

      # Set up the matplotlib figure
      plt.figure(figsize=(10, 7))

      # Plotting New Users and Conversions by Channel
      plt.subplot(2, 1, 1)
      sns.barplot(x='First user default channel group', y='New users',␣
       ↪data=user_acquisition_data, palette='coolwarm')
      plt.title('New Users by Channel')
      plt.xticks(rotation=45)

      plt.subplot(2, 1, 2)
      sns.barplot(x='First user default channel group', y='Conversions',␣
       ↪data=user_acquisition_data, palette='coolwarm')
      plt.title('Conversions by Channel')
      plt.xticks(rotation=45)

      plt.tight_layout()
      plt.show()

      # Plotting Engagement Rate and Average Engagement Time by Channel
      plt.figure(figsize=(10, 7))

      plt.subplot(2, 1, 1)
      sns.barplot(x='First user default channel group', y='Engagement rate',␣
       ↪data=user_acquisition_data, palette='viridis')
      plt.title('Engagement Rate by Channel')
      plt.xticks(rotation=45)

      plt.subplot(2, 1, 2)
      sns.barplot(x='First user default channel group', y='Average engagement time',␣
       ↪data=user_acquisition_data, palette='viridis')
      plt.title('Average Engagement Time by Channel')
      plt.xticks(rotation=45)

      plt.tight_layout()
      plt.show()
```

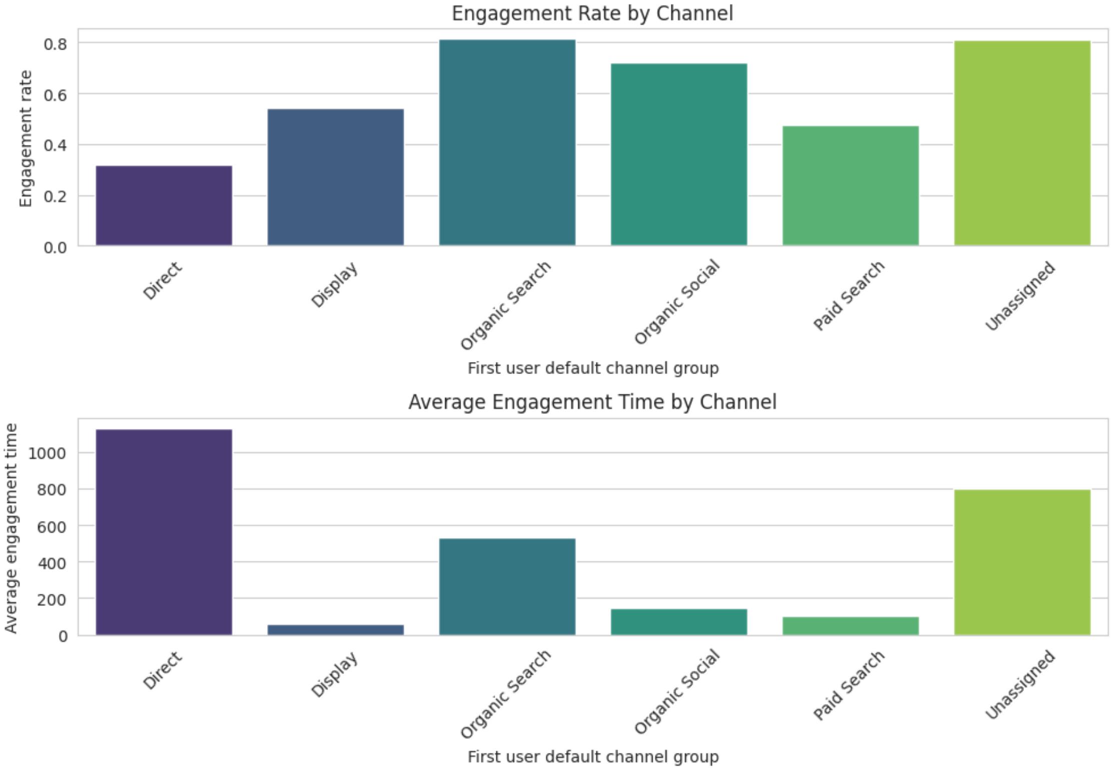
```
<ipython-input-29-5bc529866109>:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
```

```
effect.

  sns.barplot(x='First user default channel group', y='New users',
data=user_acquisition_data, palette='coolwarm')
<ipython-input-29-5bc529866109>:17: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='First user default channel group', y='Conversions',
data=user_acquisition_data, palette='coolwarm')
```



```
<ipython-input-29-5bc529866109>:28: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='First user default channel group', y='Engagement rate',
data=user_acquisition_data, palette='viridis')
<ipython-input-29-5bc529866109>:33: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='First user default channel group', y='Average engagement time',
data=user_acquisition_data, palette='viridis')
```



Insights:

Organic Search and Direct channels drive the most number of engaged sessions indicating users from these channels have high engagement.

Paid Search has a decent engagement rate of 47% though lower engaged sessions than Organic and Direct channels. Display ads channel generates high user traffic but engagement rates are lower comparatively.

Organic Social has very few users but sees high engagement rates.

Organic Search and Direct channels have very high average engagement times indicating users spend a lot of time actively engaged.

Recommendations:

Focus marketing efforts on Organic Search and optimize content to drive more organic users as they highly engaged. Assess if incremental spend in Paid Search could drive more engaged users.

Evaluate ways to improve engagement for Display and Social channel users.

Analyze landing pages, site content consumed etc. to uncover why Organic and Direct users spend more time and engage more. Apply those learnings to other channels.

Set up user segmentation to send targeted campaigns to re-engage users from specific channels.

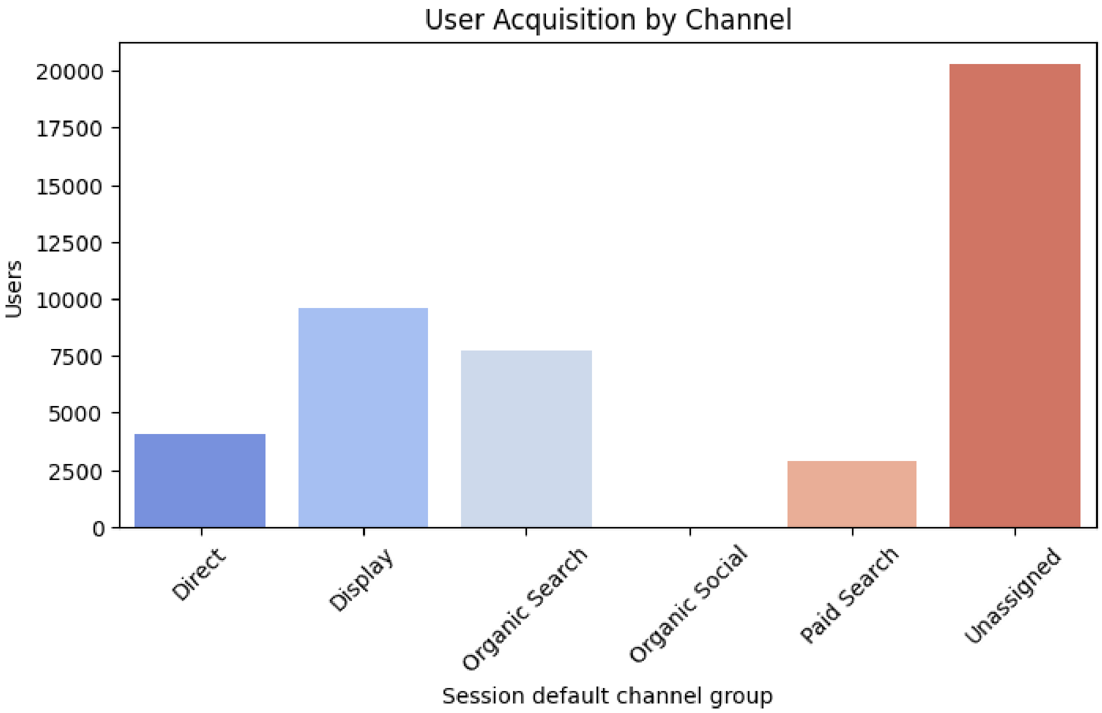
2. TRAFFIC ACQUISITION

[9]:

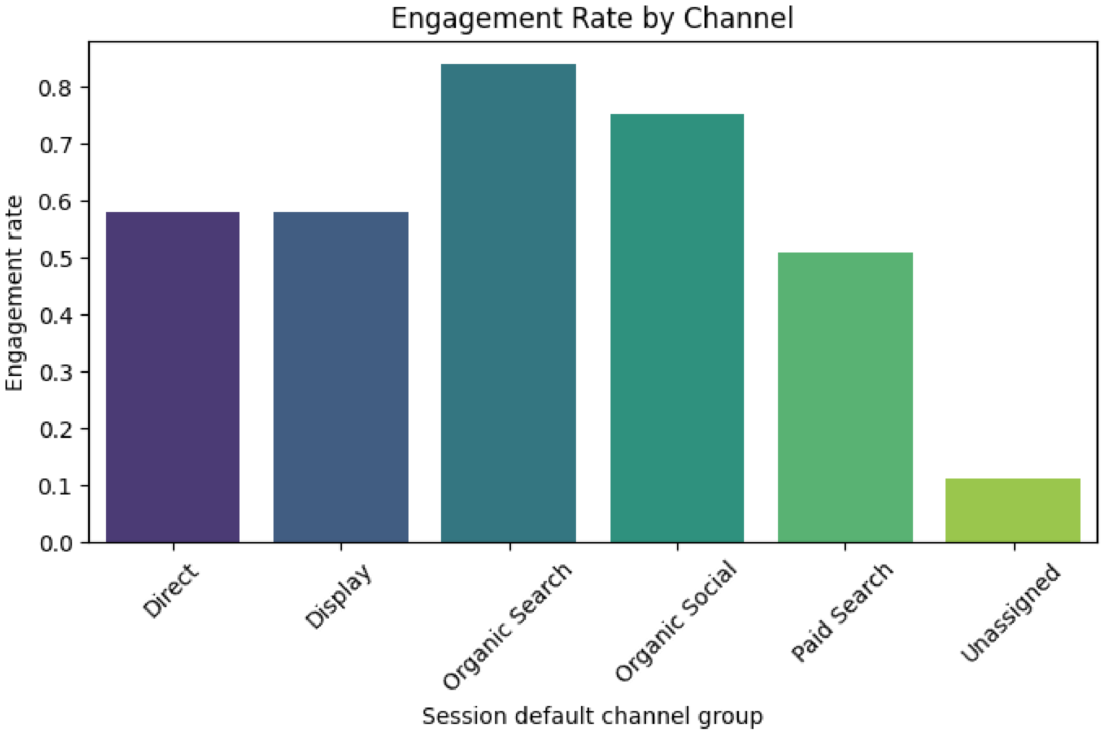
```python
# Use the 'Traffic Acquisition' sheet for analysis
traffic_acquisition_data = sheets_data['Traffic Aquisition']

# Set up the matplotlib figure for User Acquisition and Session Counts by␣
 ↪Channel
plt.figure(figsize=(8, 4))
sns.barplot(x='Session default channel group', y='Users',␣
 ↪data=traffic_acquisition_data, palette='coolwarm')
plt.title('User Acquisition by Channel')
plt.xticks(rotation=45)
plt.show()

# Engagement Rates and Average Engagement Time by Channel
plt.figure(figsize=(8, 4))
sns.barplot(x='Session default channel group', y='Engagement rate',␣
 ↪data=traffic_acquisition_data, palette='viridis')
plt.title('Engagement Rate by Channel')
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(8, 4))
sns.barplot(x='Session default channel group', y='Average engagement time per␣
 ↪session', data=traffic_acquisition_data, palette='viridis')
plt.title('Average Engagement Time by Channel')
plt.xticks(rotation=45)
plt.show()

# Conversions by Channel
plt.figure(figsize=(8, 4))
sns.barplot(x='Session default channel group', y='Conversions',␣
 ↪data=traffic_acquisition_data, palette='coolwarm')
plt.title('Conversions by Channel')
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-9-62cf15864d10>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

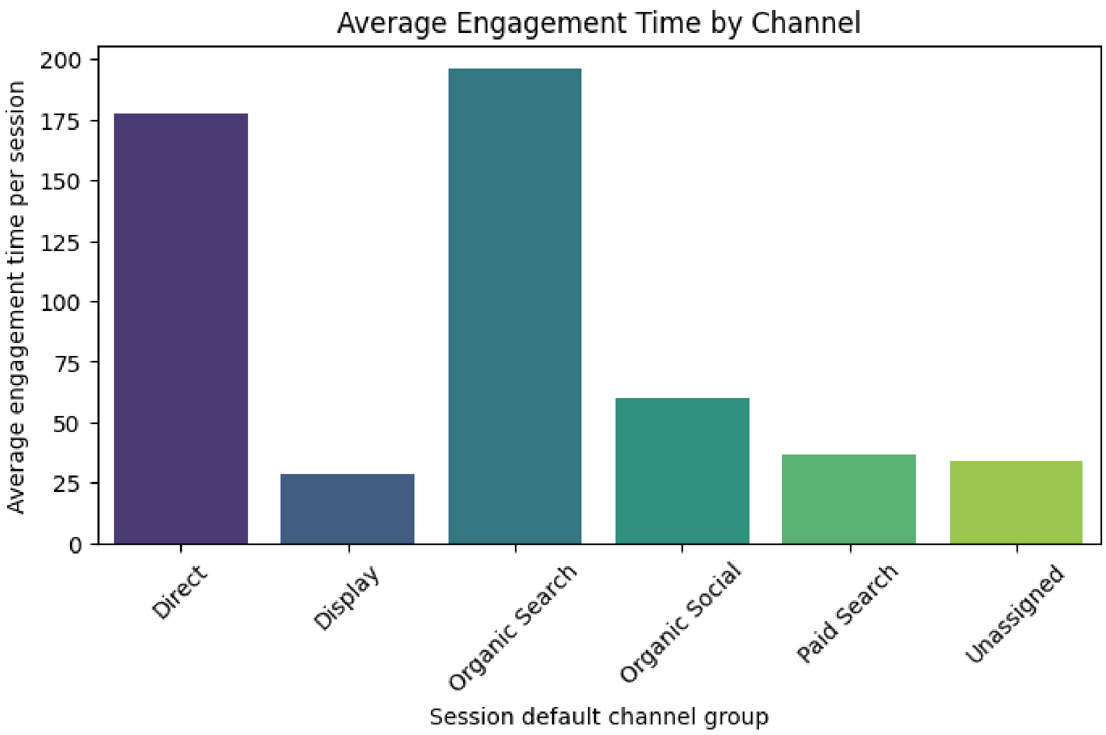
```
  sns.barplot(x='Session default channel group', y='Users',
data=traffic_acquisition_data, palette='coolwarm')
```



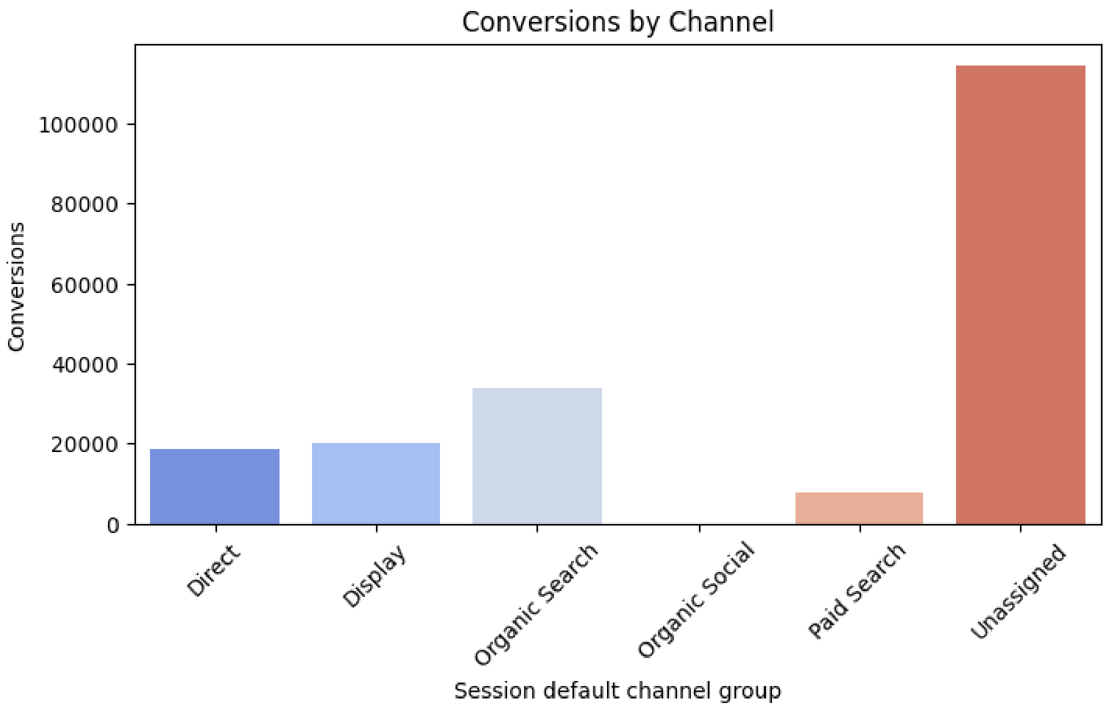
```
<ipython-input-9-62cf15864d10>:13: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Session default channel group', y='Engagement rate',
data=traffic_acquisition_data, palette='viridis')
```



```
<ipython-input-9-62cf15864d10>:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Session default channel group', y='Average engagement time per
session', data=traffic_acquisition_data, palette='viridis')
```



```
<ipython-input-9-62cf15864d10>:26: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Session default channel group', y='Conversions',
data=traffic_acquisition_data, palette='coolwarm')
```



Insights:

Organic Search drives maximum engaged sessions and has the highest engagement rate of 83%, indicating these users are highly engaged.

Direct also sees decent engaged sessions and engagement rates.

Display has high number of sessions but moderate engagement rates.

Organic Search and Direct channels have very high average engagement times showing users spend a lot of time actively engaged.

Unassigned channel group has low engagement metrics likely indicating a data quality issue.

Recommendations:

Focus on content and SEO optimization to continue driving engaged organic traffic.

Leverage behavioral data to uncover why Organic and Direct channels have such high engagement and apply those learnings to other channels.

Assess opportunities to improve engagement for Display and Paid Search channels through better targeting and personalization.

Fix data quality issue leading to low engagement for Unassigned group for better insights.

Use user segmentation to re-engage users from specific channels through tailored campaigns.

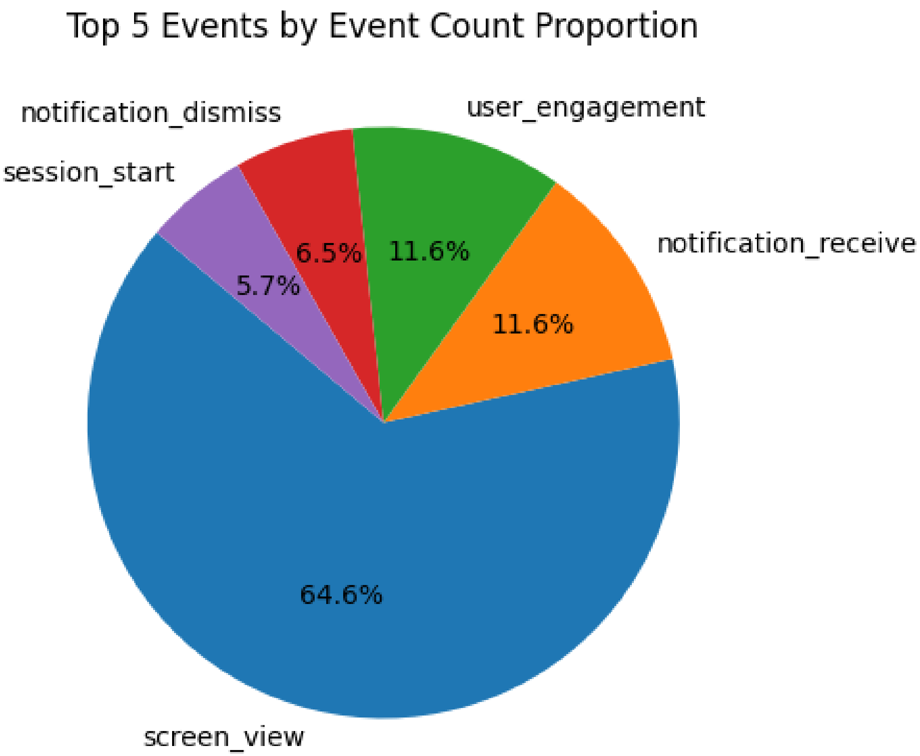
3. EVENT REPORT

[ ]:

```
# Use the 'Event Report Data' sheet for analysis
event_report_data= sheets_data['Event Report']

# Set up the matplotlib figure
plt.figure(figsize=(14, 10))

# Plotting Event Count by Event Name with updated code to avoid deprecation␣
 ↪warning
plt.subplot(2, 1, 1)
top_events_by_count = event_report_data.sort_values('Event count',␣
 ↪ascending=False).head(10)
sns.barplot(x='Event count', y='Event name', data=top_events_by_count,␣
 ↪color='skyblue')
plt.title('Top 10 Events by Event Count')

# Plotting Total Users by Event Name with updated code
plt.subplot(2, 1, 2)
top_events_by_users = event_report_data.sort_values('Total users',␣
 ↪ascending=False).head(10)
sns.barplot(x='Total users', y='Event name', data=top_events_by_users,␣
 ↪color='lightgreen')
plt.title('Top 10 Events by Total Users')

plt.tight_layout()
plt.show()

# Plotting Event Count per User by Event Name for Top Engaging Events with␣
 ↪updated code
plt.figure(figsize=(14, 7))
top_engaging_events = event_report_data.sort_values('Event count per user',␣
 ↪ascending=False).head(10)
sns.barplot(x='Event count per user', y='Event name', data=top_engaging_events,␣
 ↪color='salmon')
plt.title('Top 10 Events by Event Count per User')
plt.show()
```

[11]:

```
# Use the 'Traffic Acquisition' sheet for analysis
event_report_data= sheets_data['Event Report']

# Pie Chart for Top Events by Event Count
top_events_pie = event_report_data.sort_values('Event count', ascending=False).
 ↪head(5)
plt.figure(figsize=(5, 5))
plt.pie(top_events_pie['Event count'], labels=top_events_pie['Event name'],␣
 ↪autopct='%1.1f%%', startangle=140)
plt.title('Top 5 Events by Event Count Proportion')
plt.show()
```



Insights:

Screen views make up majority of events indicating high traffic but need to convert users better.

Notification receive and user engagement events are also prominent showing some user activity.

Specific flows like Promilo113/111/106 login -> feeds -> rewards have decent traction showing core flows.

Events per user is quite low overall(<10) indicating poor retention and engagement.

Numerous custom events but usage seems low - opportunity to streamline events.

Registration and login flows see traffic but drop off visible highlighting authentication issues.

Recommendations:

Optimize core flows driving engagement, simplify flows. Improve notification feature for re-engagement, make them more personalized.

Analyze drop off points in key flows to identify and fix issues, frustrations.

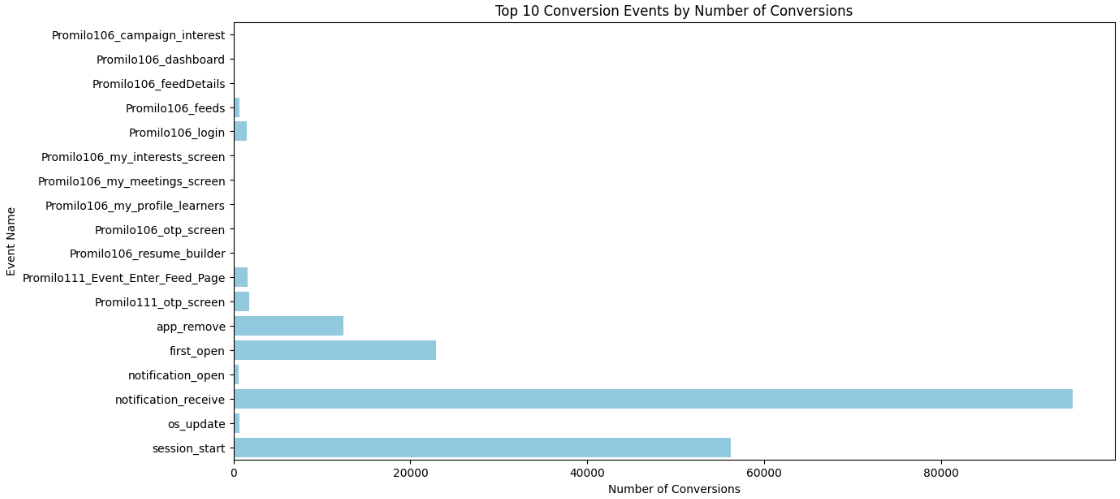
Set goals and funnel analysis to track and improve conversions.

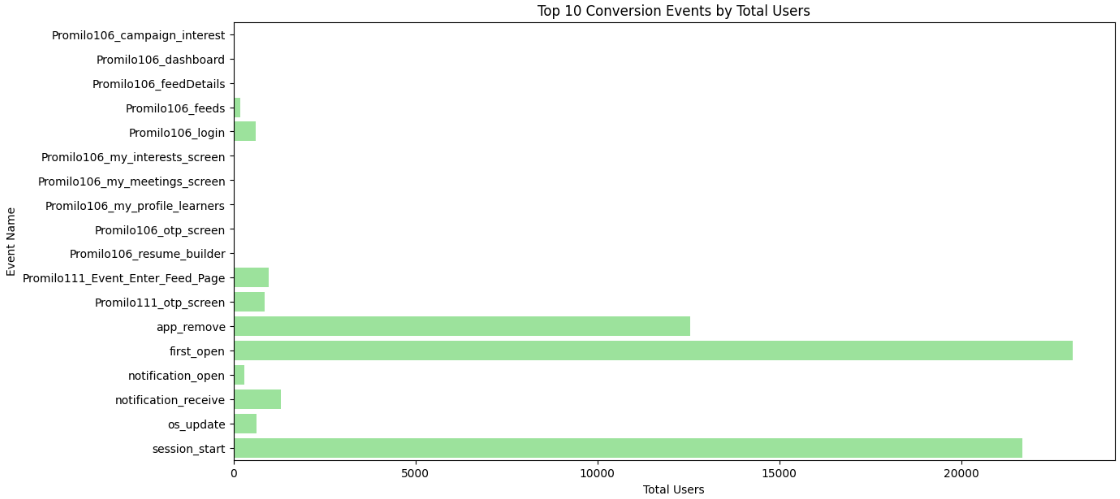
Drive loyalty through gamification, personalization for better retention.

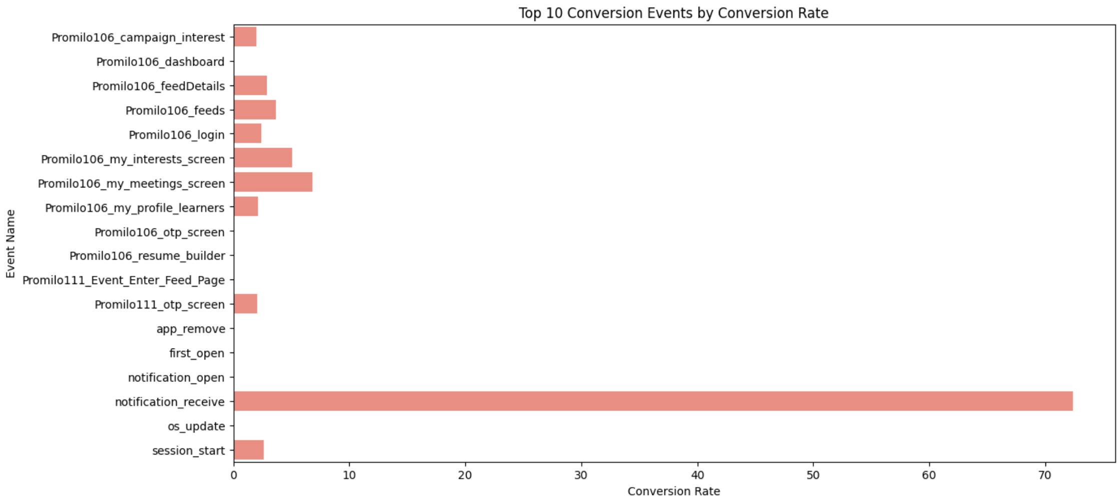
4. CONVERSION REPORT

[12]:

```
# Load the "Conversion Report" sheet
conversion_report_data = sheets_data['Conversion Report']

# Set up the matplotlib figure for the number of conversions by event name
plt.figure(figsize=(14, 7))
top_conversion_events = conversion_report_data.sort_values('Conversions',␣
 ↪ascending=False).head(10)
sns.barplot(x='Conversions', y='Event name', data=top_conversion_events,␣
 ↪color='skyblue')
plt.title('Top 10 Conversion Events by Number of Conversions')
plt.xlabel('Number of Conversions')
plt.ylabel('Event Name')
plt.show()

# Now, let's create a bar chart for the number of unique users associated with␣
 ↪each conversion event
plt.figure(figsize=(14, 7))
top_user_conversion_events = conversion_report_data.sort_values('Total users',␣
 ↪ascending=False).head(10)
sns.barplot(x='Total users', y='Event name', data=top_user_conversion_events,␣
 ↪color='lightgreen')
plt.title('Top 10 Conversion Events by Total Users')
plt.xlabel('Total Users')
plt.ylabel('Event Name')
plt.show()

# If applicable, calculate and visualize the conversion rate per event
conversion_report_data['Conversion Rate'] =␣
 ↪conversion_report_data['Conversions'] / conversion_report_data['Total␣
 ↪users'].replace(0,1)
top_conversion_rate_events = conversion_report_data.sort_values('Conversion␣
 ↪Rate', ascending=False).head(10)
plt.figure(figsize=(14, 7))
sns.barplot(x='Conversion Rate', y='Event name',␣
 ↪data=top_conversion_rate_events, color='salmon')
plt.title('Top 10 Conversion Events by Conversion Rate')
plt.xlabel('Conversion Rate')
plt.ylabel('Event Name')
plt.show()
```


Top 10 Conversion Events by Number of Conversions
Event Name
Promilo106_campaign_interest
Promilo106_dashboard
Promilo106_feedDetails
Promilo106_feeds
Promilo106_login
Promilo106_my_interests_screen
Promilo106_my_meetings_screen
Promilo106_my_profile_learners
Promilo106_otp_screen
Promilo106_resume_builder
Promilo111_Event_Enter_Feed_Page
Promilo111_otp_screen
app_remove
first_open
notification_open
notification_receive
os_update
session_start
0
20000
40000
60000
80000
Number of Conversions


Top 10 Conversion Events by Total Users
Event Name
Promilo106_campaign_interest
Promilo106_dashboard
Promilo106_feedDetails
Promilo106_feeds
Promilo106_login
Promilo106_my_interests_screen
Promilo106_my_meetings_screen
Promilo106_my_profile_learners
Promilo106_otp_screen
Promilo106_resume_builder
Promilo111_Event_Enter_Feed_Page
Promilo111_otp_screen
app_remove
first_open
notification_open
notification_receive
os_update
session_start
0
5000
10000
15000
20000
Total Users



Insights:

Notifications drive very high conversions indicating users do engage with notifications.

However other core flows like login, accessing feeds etc. see relatively lower conversions highlighting drop-offs.

Funnel fall off visible from high session starts to lower feed views and very few dashboard views.

Good conversion rates for app remove, OTP screens indicates users reaching those steps but fall off after.

Very low conversion rates for critical actions like resume builder, meetings, interests showing drop outs.

Recommendations:

Improve core flows by identifying and resolving friction points causing drop offs.

Drive higher engagement with personalized and context specific notifications.

Set up funnel analysis starting from session start to purchase to find leakages.

Test and optimize registration, login, authentication flows for better conversions.

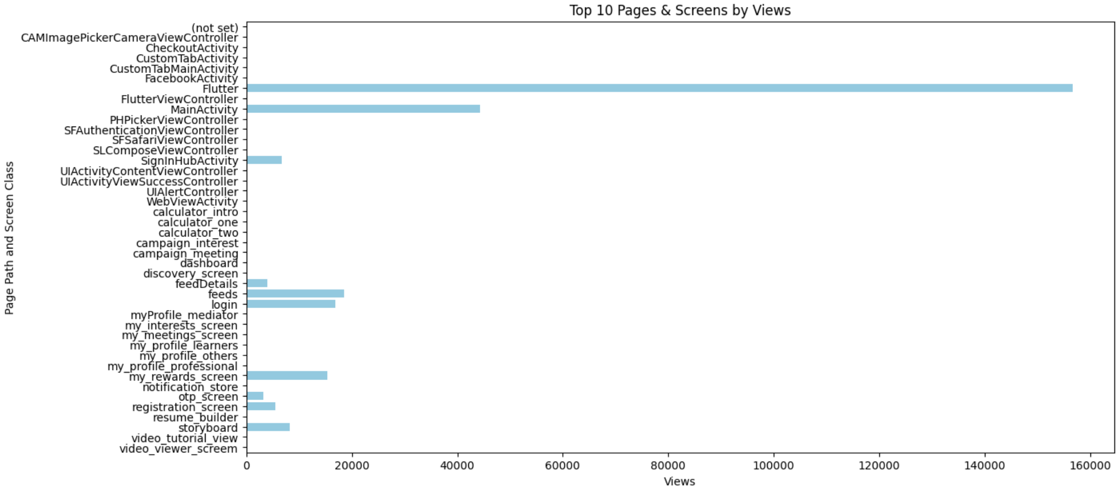
Analyze and improve content consumption flows - homepage, feeds, recommendations to retain users longer.

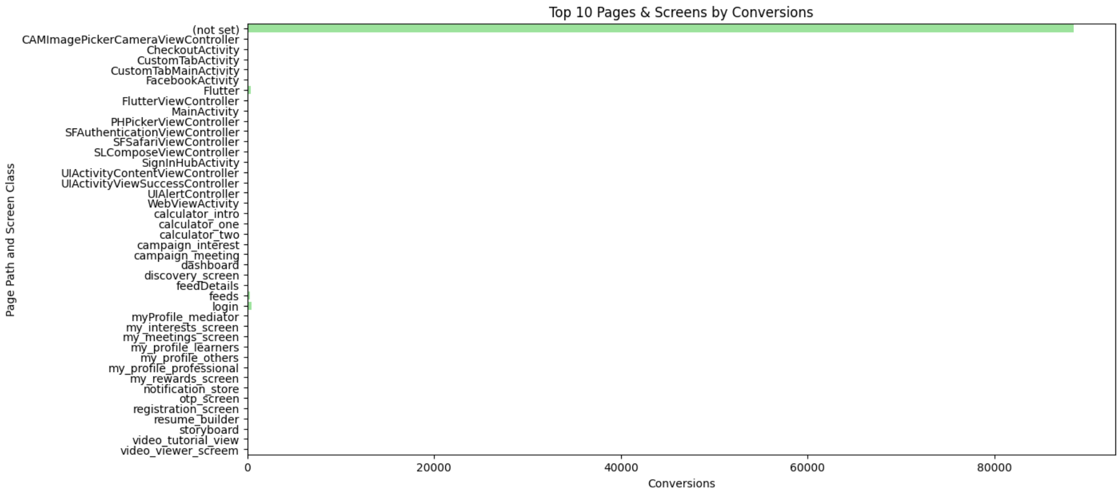
5. PAGES & SCREENS REPORT

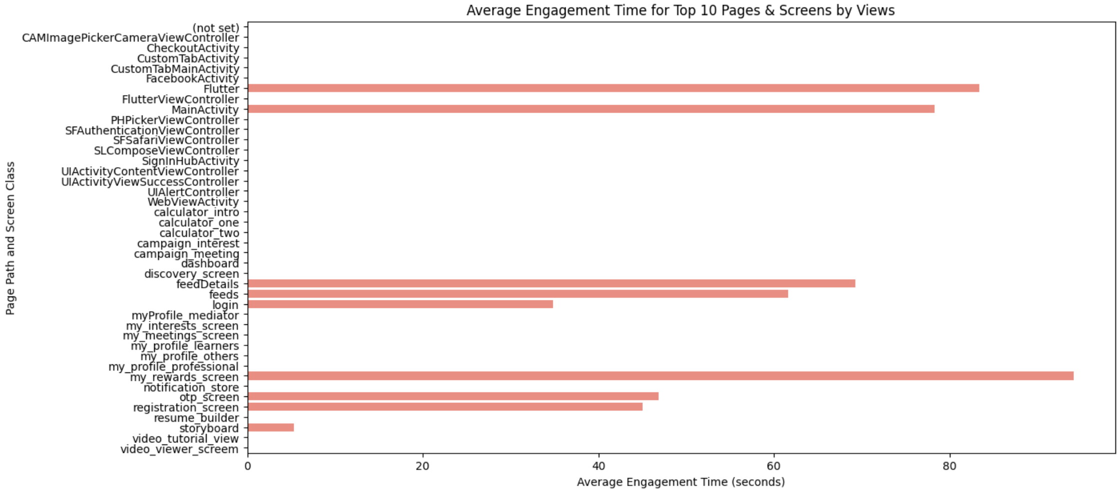
[13]:

```
# Load the "Pages & Screens Report" sheet
pages_screens_report_data = sheets_data['Pages & Screens Report']

# Visualizations for "Pages & Screens Report"

# Top pages/screens by views
```

```
plt.figure(figsize=(14, 7))
top_views_pages = pages_screens_report_data.nlargest(10, 'Views')
sns.barplot(x='Views', y='Page path and screen class', data=top_views_pages,␣
 ↪color='skyblue')
plt.title('Top 10 Pages & Screens by Views')
plt.xlabel('Views')
plt.ylabel('Page Path and Screen Class')
plt.show()

# Top pages/screens by conversions
plt.figure(figsize=(14, 7))
top_conversions_pages = pages_screens_report_data.nlargest(10, 'Conversions')
sns.barplot(x='Conversions', y='Page path and screen class',␣
 ↪data=top_conversions_pages, color='lightgreen')
plt.title('Top 10 Pages & Screens by Conversions')
plt.xlabel('Conversions')
plt.ylabel('Page Path and Screen Class')
plt.show()

# Plotting average engagement time for the top pages/screens by views
plt.figure(figsize=(14, 7))
sns.barplot(x='Average engagement time', y='Page path and screen class',␣
 ↪data=top_views_pages, color='salmon')
plt.title('Average Engagement Time for Top 10 Pages & Screens by Views')
plt.xlabel('Average Engagement Time (seconds)')
plt.ylabel('Page Path and Screen Class')
plt.show()
```







Insights:

High views for Flutter home screen but low conversions indicate drop-offs from home page.

Core flows like feeds, login and registration see traffic but conversion rates are low.

Good views and conversions for rewards and storyboard screens implying engaging content.

Very high views per user for unused screens like CustomTabMain indicating tracking issues.

Critical workflows like profile building, meetings have very low traffic highlighting dropout issues.

Recommendations:

Identify and fix friction issues in core flows causing high home page exits.

Set up funnel analysis from home page to purchase to find and resolve leakage points. Drive engagement through personalized content recommendations on home, feeds and rewards screens.

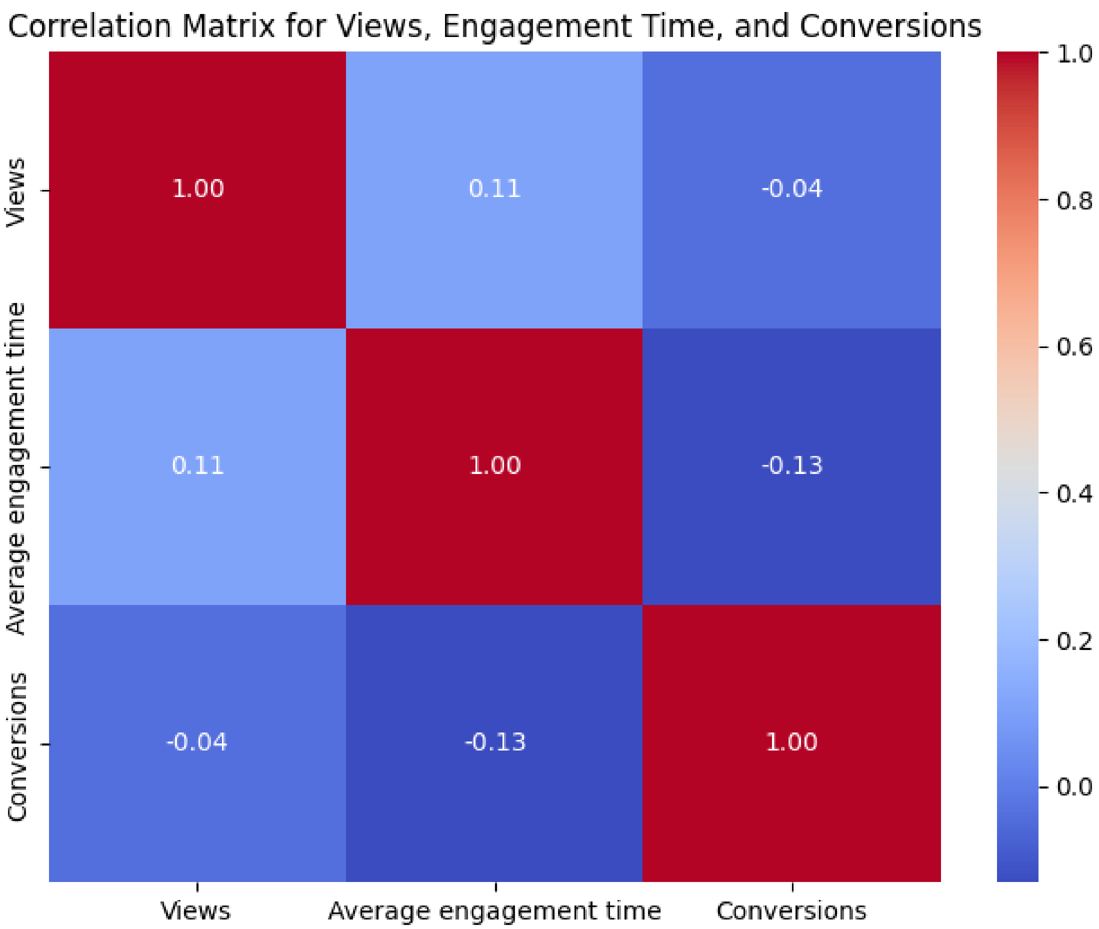
Enable deep linking features to drive traffic directly to key workflows.

Establish usage data instrumentation to prevent tracking unused screens and flows.

Highlight and promote underutilized but critical workflows on home page for better visibility.

OUTLIER DETECTION ON PAGES & SCREEN REPORT BASED ON IQR METHOD

[14]:
```python
from scipy import stats
import numpy as np

# Function to detect outliers based on the IQR method
def detect_outliers_iqr(df, feature):
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    outlier_step = 1.5 * IQR
    outliers = df[(df[feature] < Q1 - outlier_step) | (df[feature] > Q3 +␣
 ↪outlier_step)]
    return outliers

# Detecting outliers for 'Views', 'Average engagement time', and 'Conversions'
outliers_views = detect_outliers_iqr(pages_screens_report_data, 'Views')
outliers_engagement_time = detect_outliers_iqr(pages_screens_report_data,␣
 ↪'Average engagement time')
outliers_conversions = detect_outliers_iqr(pages_screens_report_data,␣
 ↪'Conversions')

# Correlation analysis
correlation_data = pages_screens_report_data[['Views', 'Average engagement␣
 ↪time', 'Conversions']]
correlation_matrix = correlation_data.corr()

# Visualizing the correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Matrix for Views, Engagement Time, and Conversions')
plt.show()

# Display outliers for Views
outliers_views, outliers_engagement_time, outliers_conversions
```



```
[14]: (  Page path and screen class   Views  Users  Views per user  \
      0                     Flutter  156708   8726       17.958740
      1                MainActivity   44326   8978        4.937180
      2                       feeds   18514   4358        4.248279
      3                       login   16883   7291        2.315595
      4           my_rewards_screen   15381   2045        7.521271
      5                  storyboard    8189   5244        1.561594

         Average engagement time  Event count  Conversions  Total revenue
      0                83.412220       203901          328              0
      1                78.292160        53374          101              0
      2                61.600050        37628          253              0
      3                34.881770        40772          435              0
      4                94.179950        32910            5              0
      5                 5.341152        15676          115              0  ,
         Page path and screen class  Views  Users  Views per user  \
      31    my_profile_professional     67     23        2.913043
      34           CheckoutActivity     24      3        8.000000
```

```
    Average engagement time  Event count  Conversions  Total revenue
31                 184.0870          143            0              0
34                 304.3333           26            0              0  ,
   Page path and screen class   Views  Users  Views per user  \
0                     Flutter  156708   8726       17.958740
1                MainActivity   44326   8978        4.937180
2                       feeds   18514   4358        4.248279
3                       login   16883   7291        2.315595
5                  storyboard    8189   5244        1.561594
7         registration_screen    5501   3566        1.542625
8                 feedDetails    3971   1047        3.792741
41                  (not set)       0   9145        0.000000

    Average engagement time  Event count  Conversions  Total revenue
0                 83.412220       203901          328              0
1                 78.292160        53374          101              0
2                 61.600050        37628          253              0
3                 34.881770        40772          435              0
5                  5.341152        15676          115              0
7                 45.075720        13496          136              0
8                 69.316140         7820           84              0
41                 0.001093       109189        88518              0  )
```

The outlier analysis yields the following insights:

The outlier detection for ‘Views’, ‘Average engagement time’, and ‘Conversions’ indicates that certain pages or screens have unusually high or low metrics compared to the rest. For example, the ‘Flutter’ screen has significantly higher views than other pages, which could indicate a central feature of the app or a default landing page.

Correlation Matrix: The heatmap of the correlation matrix shows the relationships between ‘Views’, ‘Average engagement time’, and ‘Conversions’. There appears to be a positive correlation between ‘Views’ and ‘Conversions’, suggesting that pages with more views tend to have more conversions. However, the correlation is not perfect, indicating other factors are also at play when it comes to converting views into conversions.

For pages like ‘CheckoutActivity’ with an exceptionally high ‘Average engagement time’, we would want to investigate why users spend so much time there. It could be due to a well-engaged process or possibly a design issue causing delays.

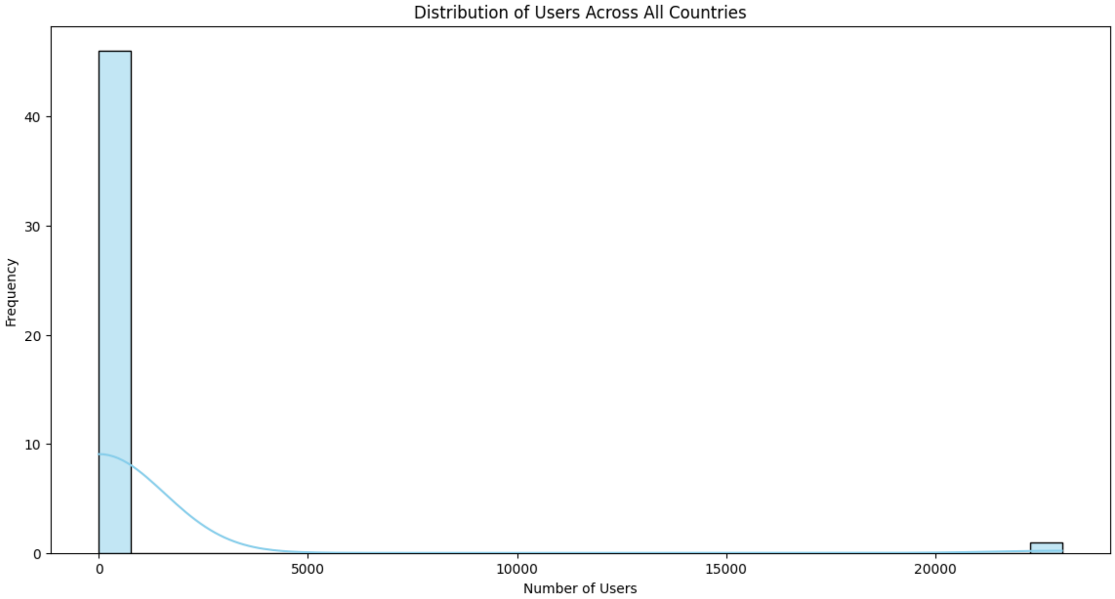
The ‘Flutter’ screen stands out with a high number of conversions, which is interesting and deserves further investigation into what about this screen leads to conversions.

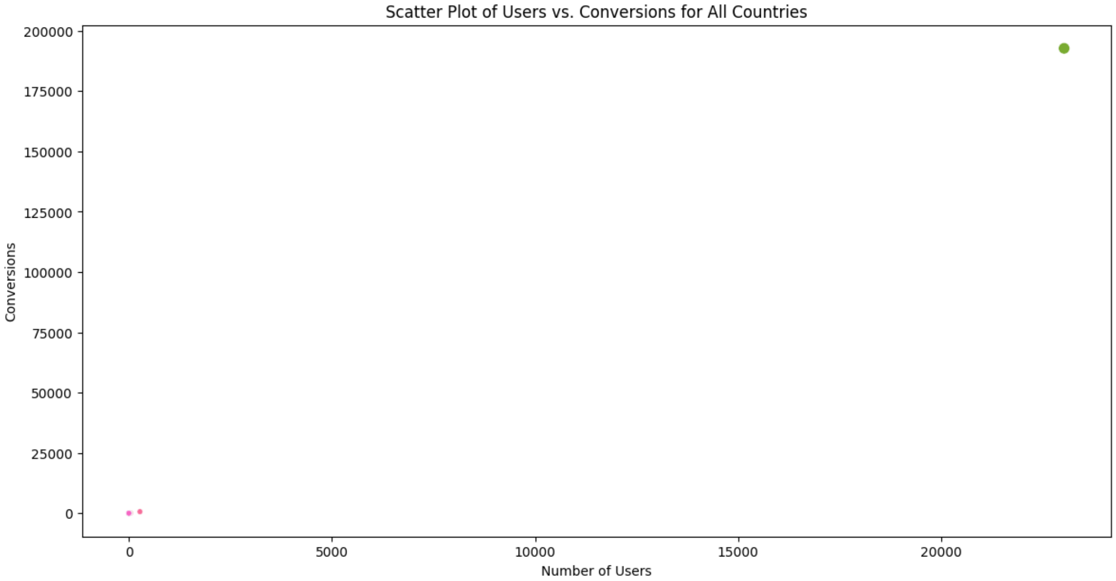
There is an anomaly where a page has zero views but a significant number of conversions. This could be due to tracking issues or a data error and should be investigated further.

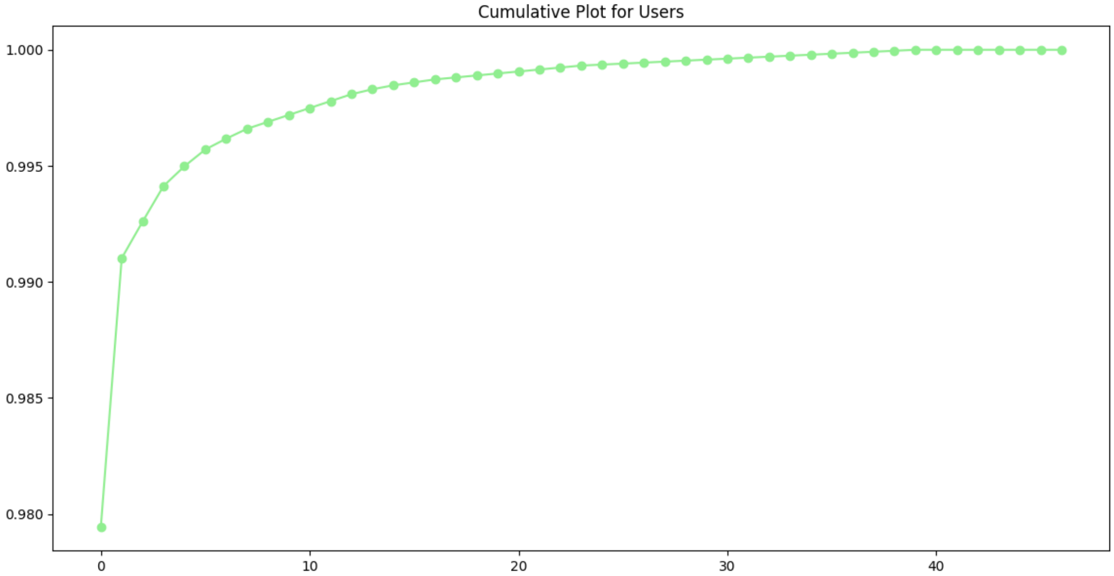
6. DEMOGRAPHICS REPORT

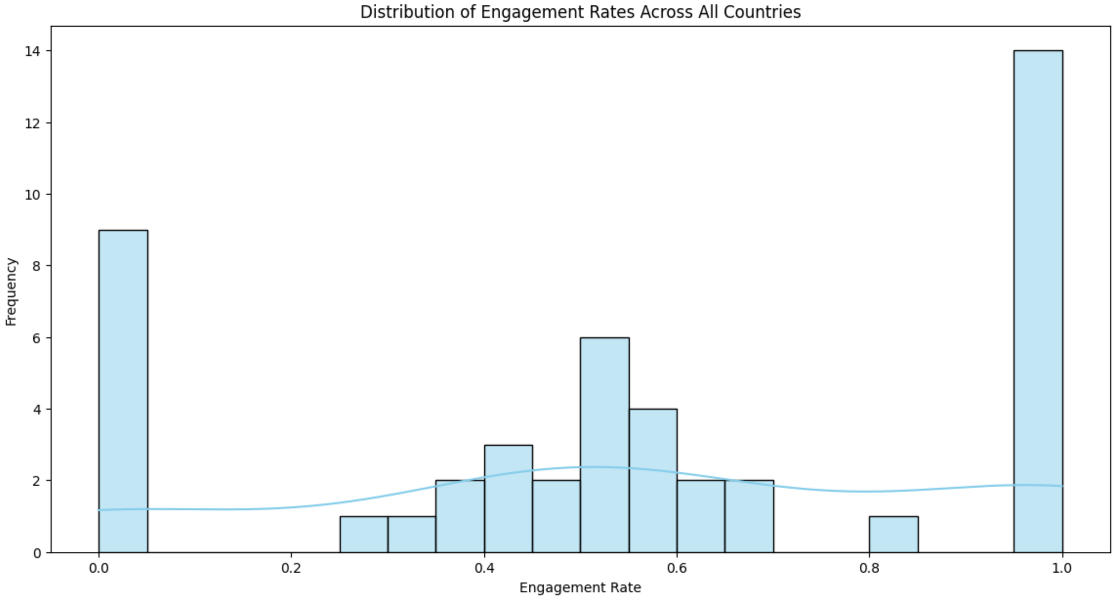
[15]:

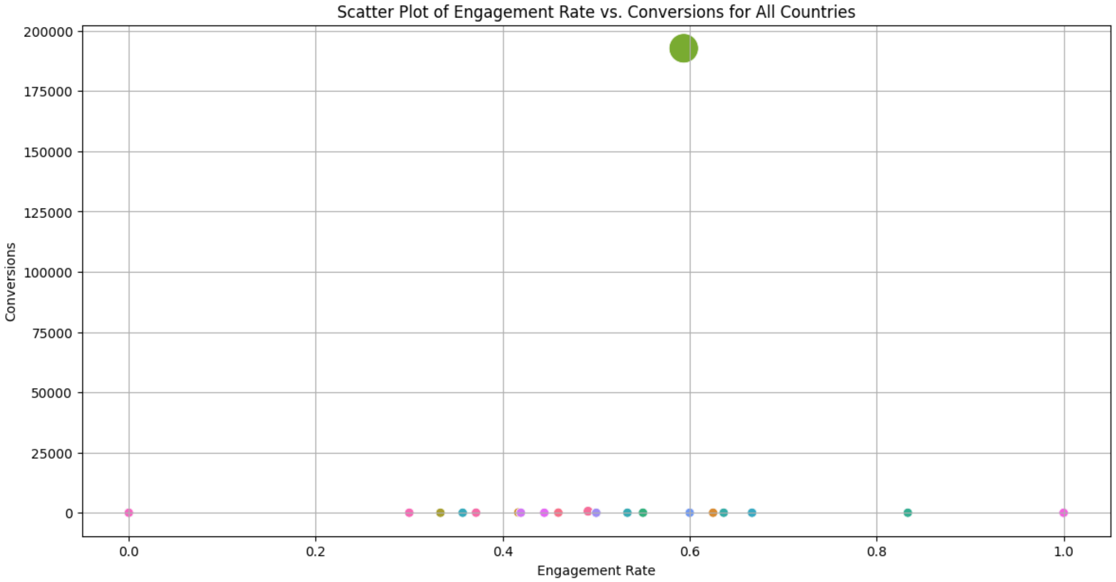
```
# Load the "Demographics Report" sheet
demographics_report_data = sheets_data['Demographics Report']

# Distribution Plot for the number of users across all countries
plt.figure(figsize=(14, 7))
sns.histplot(demographics_report_data['Users'], bins=30, kde=True,␣
 ↪color='skyblue')
plt.title('Distribution of Users Across All Countries')
plt.xlabel('Number of Users')
plt.ylabel('Frequency')
plt.show()

# Scatter Plot to compare number of users to conversions for all countries
plt.figure(figsize=(14, 7))
sns.scatterplot(data=demographics_report_data, x='Users', y='Conversions',␣
 ↪hue='Country', size='Conversions', legend=False)
plt.title('Scatter Plot of Users vs. Conversions for All Countries')
plt.xlabel('Number of Users')
plt.ylabel('Conversions')
plt.show()

# Cumulative Plot for users
users_sorted = demographics_report_data.sort_values('Users',␣
 ↪ascending=False)['Users']
cumulative_users = np.cumsum(users_sorted) / users_sorted.sum()
plt.figure(figsize=(14, 7))
plt.plot(range(len(cumulative_users)), cumulative_users, marker='o',␣
 ↪linestyle='-', color='lightgreen')
plt.title('Cumulative Plot for Users')
plt.show()

# Distribution plot for engagement rates across all countries
plt.figure(figsize=(14, 7))
sns.histplot(demographics_report_data['Engagement rate'], bins=20, kde=True,␣
 ↪color='skyblue')
plt.title('Distribution of Engagement Rates Across All Countries')
plt.xlabel('Engagement Rate')
plt.ylabel('Frequency')
plt.show()

# Scatter plot for relationship between engagement rate and conversions for all␣
 ↪countries
plt.figure(figsize=(14, 7))
sns.scatterplot(data=demographics_report_data, x='Engagement rate',␣
 ↪y='Conversions', size='Users', hue='Country', legend=False, sizes=(50, 500))
plt.title('Scatter Plot of Engagement Rate vs. Conversions for All Countries')
plt.xlabel('Engagement Rate')
plt.ylabel('Conversions')
plt.grid(True)
plt.show()
```






Cumulative Plot for Users
1.000
0.995
0.990
0.985
0.980
0
10
20
30
40


Distribution of Engagement Rates Across All Countries
Frequency
14
12
10
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0
Engagement Rate



Insights:

India accounts for majority of user base indicating home market concentration.

Very low user and engagement metrics across other countries highlighting lack of global expansion.

Top countries by engaged sessions are India, US, Canada - aligns with product localization needs.

Engagement and conversion rates better in International markets like Kuwait, China indicating product-market fit gaps in India market.

Average engagement time higher across most western markets showing more user time spent.

Recommendations:

Expand product footprint in North America by localizing content and flows for better conversion.

Conduct market analysis and build expansion strategy for western regions with higher engagement.

Compare top performing countries and identify product capabilities to drive adoption across markets with lower traction.

Double down on brand marketing and localization efforts in India to continue growth in home country.

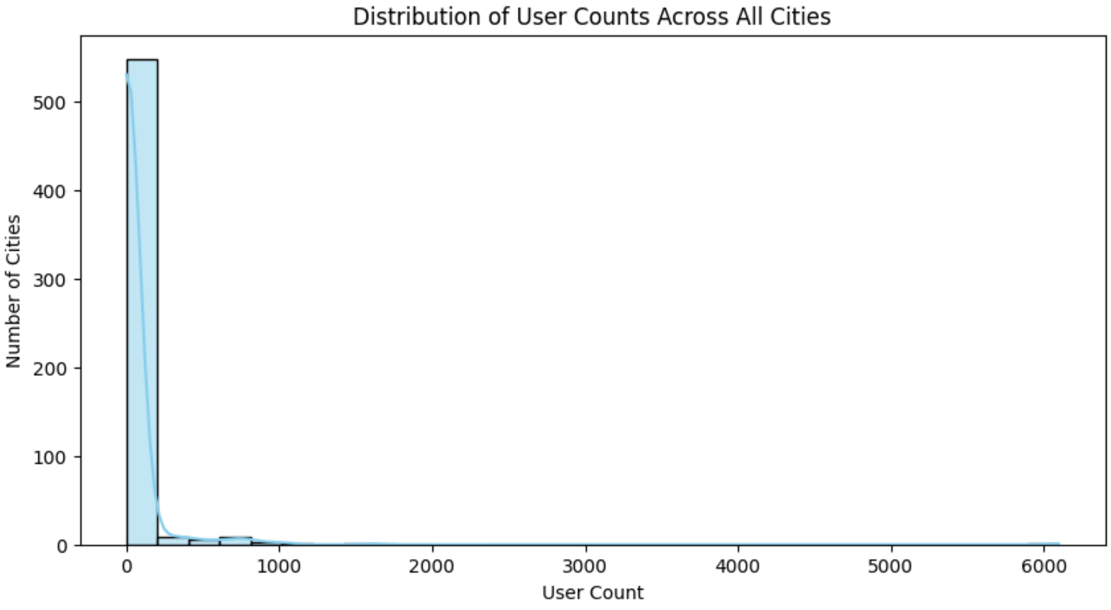
Enable region based segmentation for customized campaigns to resonate across user demographics.

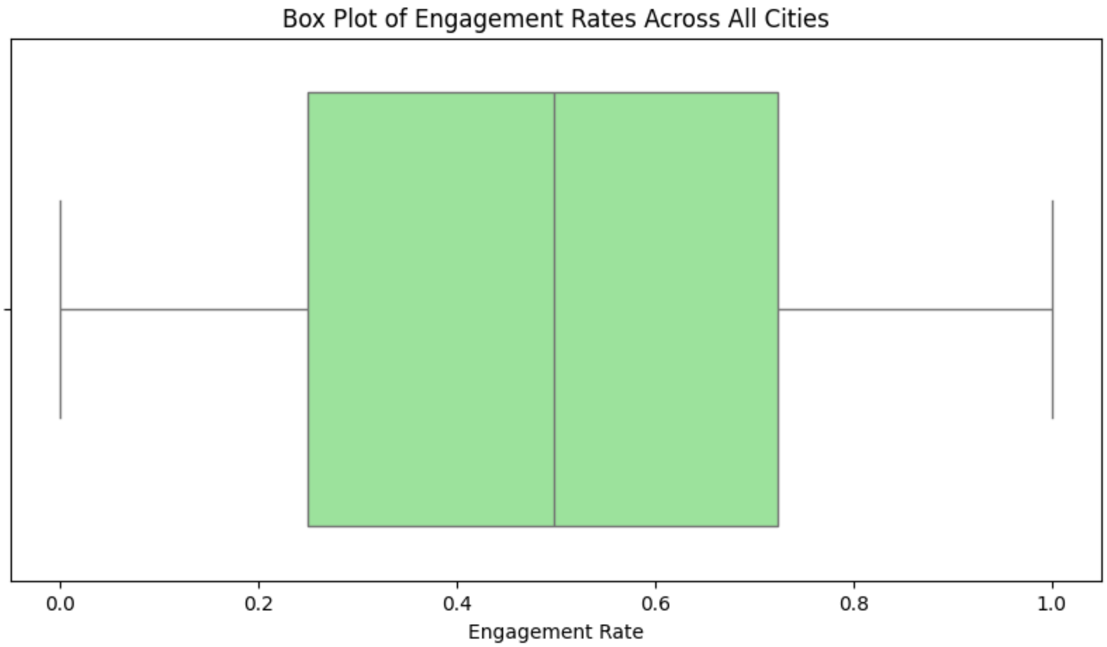
7. CITIWISE REPORT

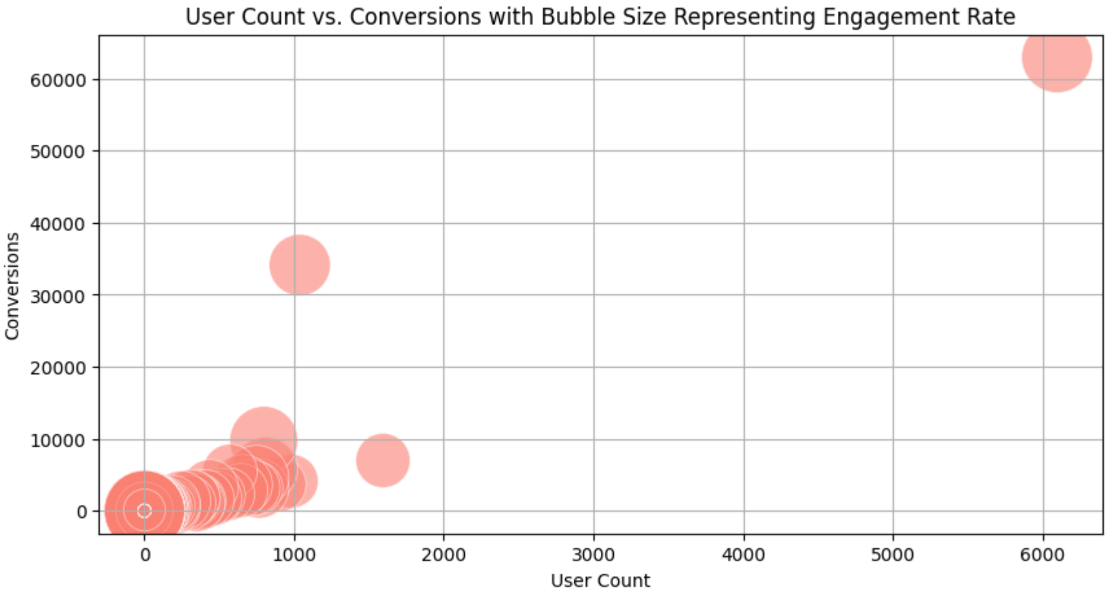
[16]:

```
# Load the "Citiwise Report" sheet
citywise_report_data = sheets_data['Citiwise Report']

# Distribution plot for user counts across all cities
plt.figure(figsize=(10, 5))
```

```
sns.histplot(citywise_report_data['Users'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of User Counts Across All Cities')
plt.xlabel('User Count')
plt.ylabel('Number of Cities')
plt.show()

# Box plot for engagement rates across all cities
plt.figure(figsize=(10, 5))
sns.boxplot(x=citywise_report_data['Engagement rate'], color='lightgreen')
plt.title('Box Plot of Engagement Rates Across All Cities')
plt.xlabel('Engagement Rate')
plt.show()

# Scatter plot with bubble size for user count vs conversions
plt.figure(figsize=(10, 5))
sns.scatterplot(data=citywise_report_data, x='Users', y='Conversions',␣
 ↪size='Engagement rate',
                legend=False, sizes=(50, 2000), color='salmon', alpha=0.6)
plt.title('User Count vs. Conversions with Bubble Size Representing Engagement␣
 ↪Rate')
plt.xlabel('User Count')
plt.ylabel('Conversions')
plt.grid(True)
plt.show()
```







Insights:

Bengaluru has highest total users, engaged sessions, events, and conversions showing very high engagement

Patna, Indore, Lucknow have relatively low engagement rates Average engagement time per session varies significantly across cities

Smaller cities like Tiruppur, Madurai, Shivamogga have very high engaged sessions per user showing a dedicated user base Many cities have zero

Even top cities have low (~10%) conversion rates

Recommendations:

Bengaluru's high engagement indicates it should be a focus area for growth

Analyze why engagement/retention is lower in Patna, Indore, Lucknow and identify opportunities to improve

Leverage differences in user behavior across geographies to drive engagement

Invest more in smaller cities like Tiruppur, Madurai, Shivamogga which show dedicated user bases

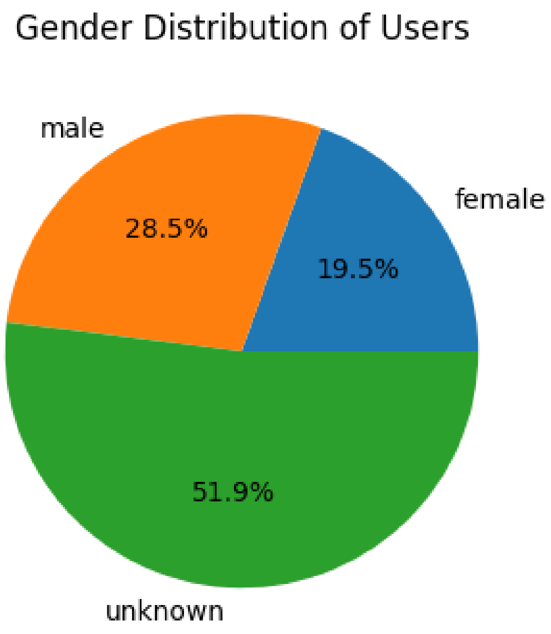
Analyze if there are limitations to monetization on the platform and identify other revenue opportunities

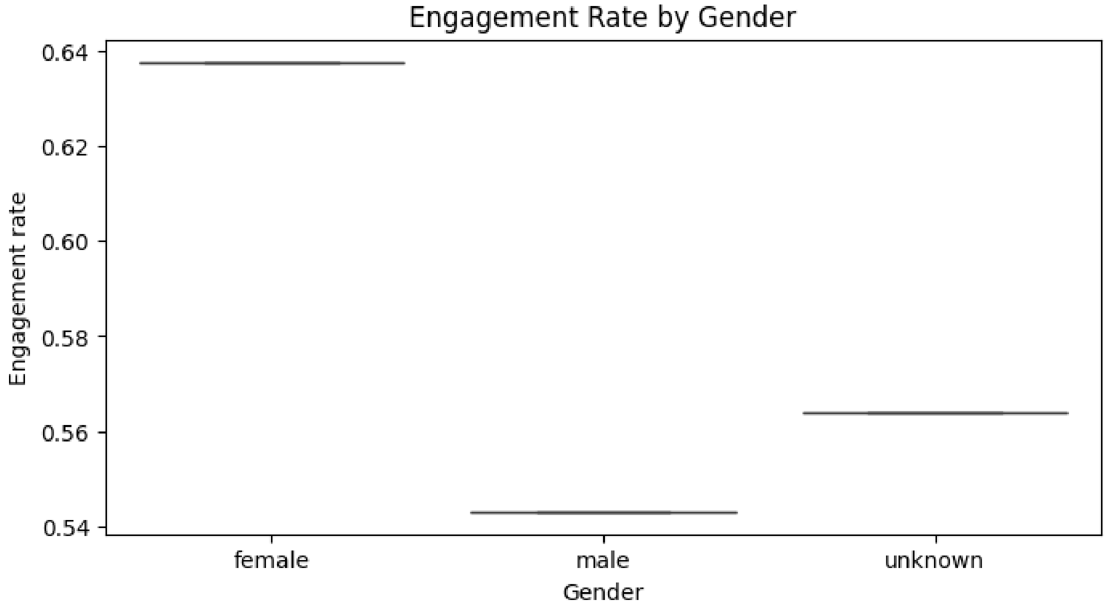
Analyze user funnels across cities and optimize to drive more conversions

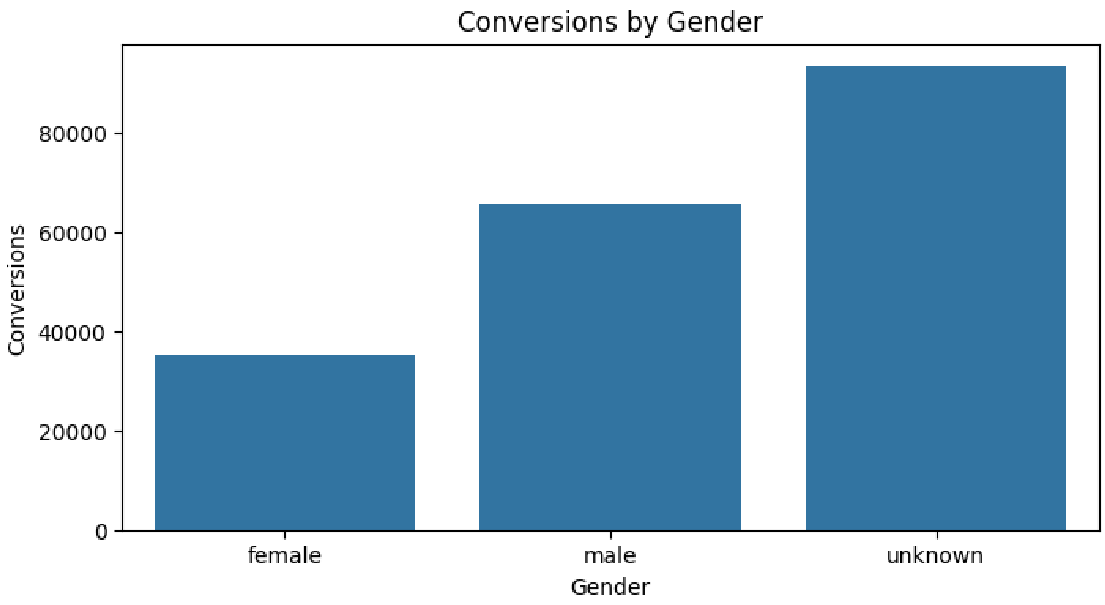
8. GENDER REPORT

[17]:

```python
# Load the "Gender Report" sheet
gender_report_data = sheets_data['Gender Report']

# Pie chart for gender distribution
gender_distribution = gender_report_data.groupby('Gender')['Users'].sum()
plt.figure(figsize=(6, 4))
gender_distribution.plot.pie(autopct='%1.1f%%')
plt.title('Gender Distribution of Users')
plt.ylabel('')
plt.show()

# Box plot for engagement rates by gender
plt.figure(figsize=(8, 4))
sns.boxplot(x='Gender', y='Engagement rate', data=gender_report_data)
plt.title('Engagement Rate by Gender')
plt.show()

# Bar chart for conversions by gender
plt.figure(figsize=(8, 4))
sns.barplot(x='Gender', y='Conversions', data=gender_report_data)
plt.title('Conversions by Gender')
plt.show()
```


Gender Distribution of Users
male
female
28.5%
19.5%
51.9%
unknown


Engagement Rate by Gender
0.64
0.62
0.60
0.58
0.56
0.54
Engagement rate
female
male
unknown
Gender



Insights:

Unknown gender has the most users, engaged sessions, events and conversions overall.

Engagement rate is highest for female users at 63.8%, compared to 56.4% for unknown and 54.3% for male.

Females also have higher engaged sessions per user (1.59) compared to males (1.45) and unknown (1.76).

Average engagement time is highest for females at 208 seconds, compared to 439 seconds for unknown and 128 seconds for males.

Recommendations:

Focus engagement and retention efforts on female segment given they have highest engagement rates.

Leverage higher average engagement time for unknown segment with features that drive more sessions per user.

Analyze why male engagement rate and time spent is lower and identify opportunities to improve engagement.

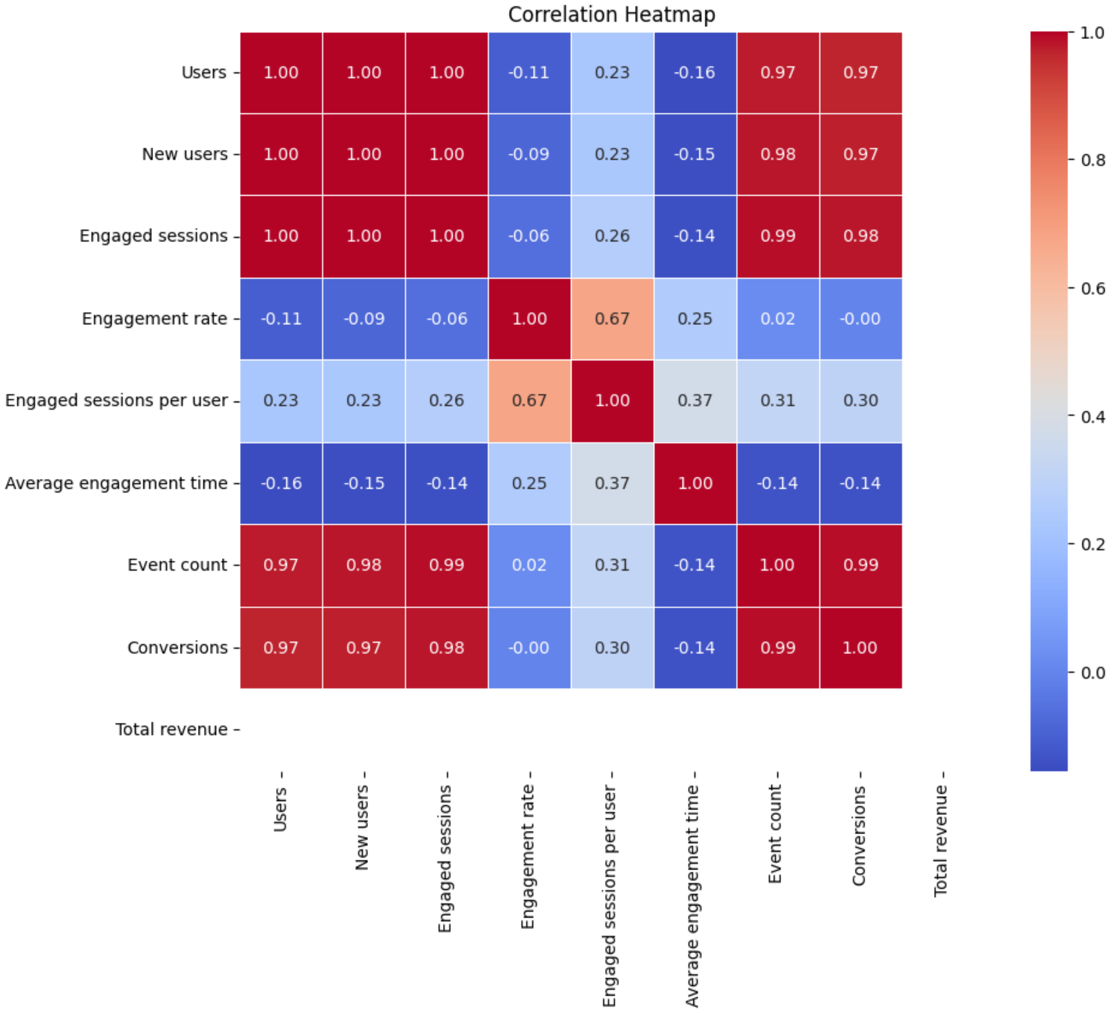
Explore customized content/features that resonates with female users to further improve engagement.

Set gender-specific engagement benchmarks and track metrics by gender to quantify impact.

9. USERS BY INTEREST

[18]:
```
# Load the "Users By Interest" sheet
user_by_interest_data = sheets_data['User By Interest']
```

```
# Correlation matrix
corr = user_by_interest_data.drop('Interests', axis=1).corr()

# Plotting the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()
```



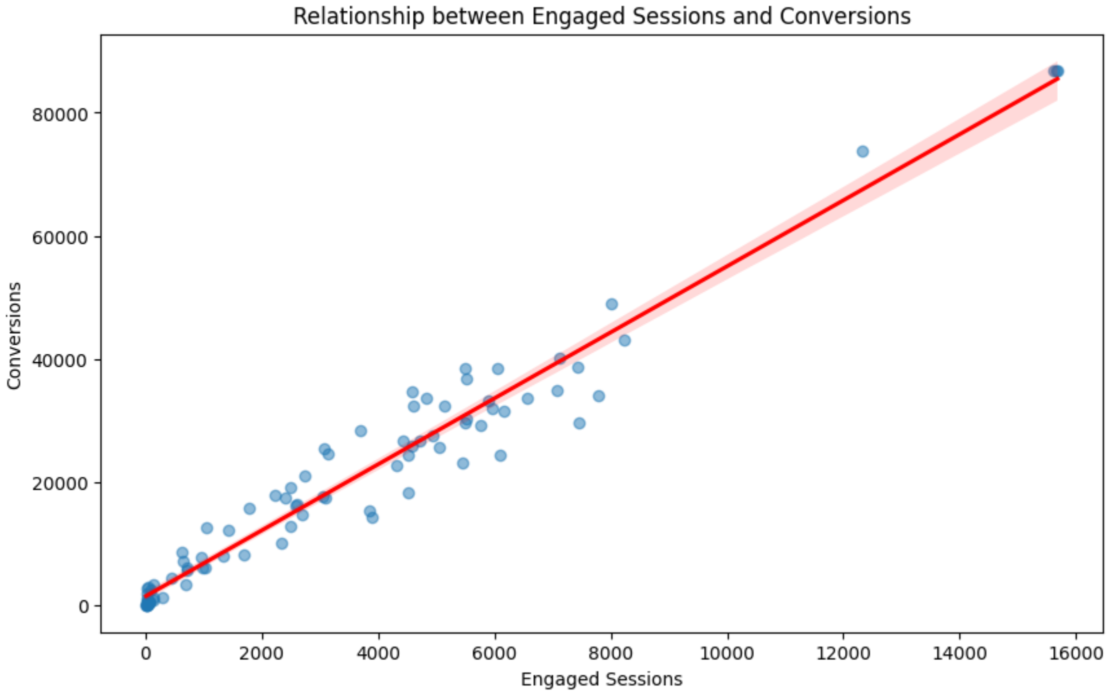
Key observations include:

There’s a strong positive correlation between users, engaged sessions, event count, and conversions. This suggests that higher user engagement and activity levels are associated with higher conversions.

Engagement rate has a weaker correlation with conversions compared to other metrics. This might indicate that while engagement rate is important, the sheer volume of engagement (e.g., total engaged sessions) plays a more significant role in driving conversions.

Average engagement time shows moderate correlation with other metrics, suggesting that the quality of engagement (in terms of time spent) also contributes to overall engagement and conversion outcomes.

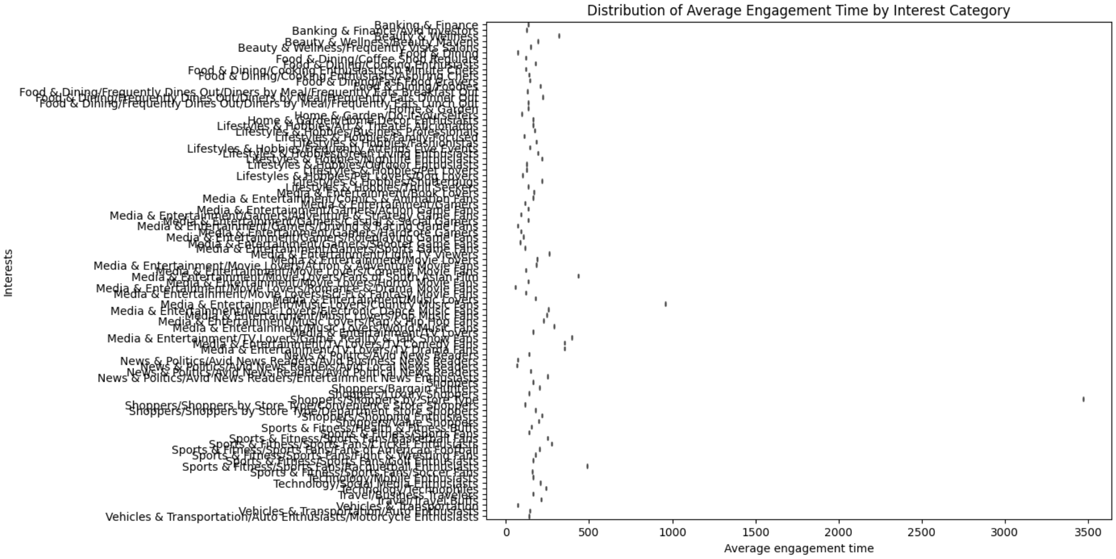
```
[19]: # Scatter plot with regression line between Engaged Sessions and Conversions
plt.figure(figsize=(10, 6))
sns.regplot(x='Engaged sessions', y='Conversions', data=user_by_interest_data,␣
  ↪scatter_kws={'alpha':0.5}, line_kws={'color':'red'})
plt.title('Relationship between Engaged Sessions and Conversions')
plt.xlabel('Engaged Sessions')
plt.ylabel('Conversions')
plt.show()
```



The scatter plot with a regression line between Engaged Sessions and Conversions reveals a clear positive relationship: as the number of engaged sessions increases, conversions also tend to increase. The regression line indicates a strong linear relationship, suggesting that engaged sessions could be a good predictor of conversions.

```
[20]: # Box plot for Average Engagement Time across different Interest Categories
plt.figure(figsize=(10, 8))
sns.boxplot(x='Average engagement time', y='Interests',␣
  ↪data=user_by_interest_data, palette='coolwarm')
plt.title('Distribution of Average Engagement Time by Interest Category')
plt.show()
```

```
<ipython-input-20-daa2e17c4921>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='Average engagement time', y='Interests',
data=user_by_interest_data, palette='coolwarm')
```



The box plot for Average Engagement Time across different Interest Categories reveals several insights:

There's noticeable variability in engagement time across categories. For instance, categories like 'Food & Dining/Cooking Enthusiasts' tend to have higher engagement times, suggesting more in-depth interaction with content.

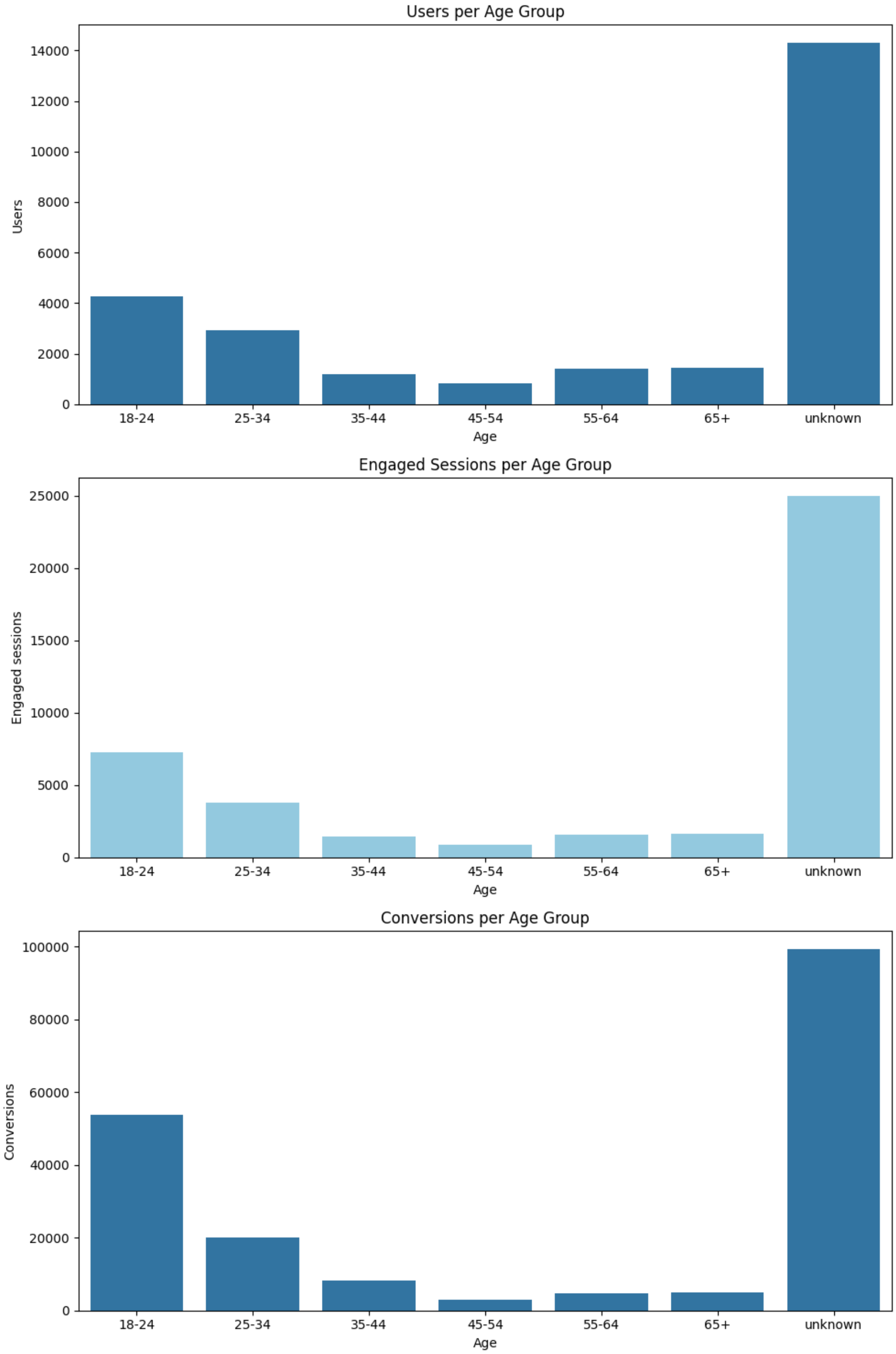
Some categories exhibit a wide range of engagement times, indicating diverse user behaviors within those interests. For example, 'Technology/Mobile Enthusiasts' and 'Shoppers' show a broader spread of engagement times, which might reflect different types of content or activities within those categories that engage users differently.

There are a few outliers in categories such as 'Food & Dining/Cooking Enthusiasts' and 'Sports & Fitness/Health & Fitness Buffs', where certain sessions have significantly higher engagement times than the majority. These outliers could represent particularly engaging content or highly interested users.

10. USERS BY AGE

[21]:
```python
# Load the "Users By Age" sheet
user_by_age_data = sheets_data['User By Age']
```

```
# Adjusted plotting without 'palette' argument
fig, axes = plt.subplots(3, 1, figsize=(10, 15))

# Users per Age Group
sns.barplot(ax=axes[0], x='Age', y='Users', data=user_by_age_data)
axes[0].set_title('Users per Age Group')

# Engaged Sessions per Age Group
sns.barplot(ax=axes[1], x='Age', y='Engaged sessions',␣
 ↪data=user_by_age_data,color='skyblue')
axes[1].set_title('Engaged Sessions per Age Group')

# Conversions per Age Group
sns.barplot(ax=axes[2], x='Age', y='Conversions', data=user_by_age_data)
axes[2].set_title('Conversions per Age Group')

plt.tight_layout()
plt.show()
```


Users per Age Group

Engaged Sessions per Age Group

Conversions per Age Group

The visualizations provide insights into the dataset in terms of Users, Engaged Sessions, and Conversions across different Age Groups:

Users per Age Group: The ‘unknown’ age group has the highest number of users, followed by the ‘18-24’ and ‘25-34’ age groups. This suggests that younger users are more prevalent in this dataset, although a significant portion of the user base has not specified their age.

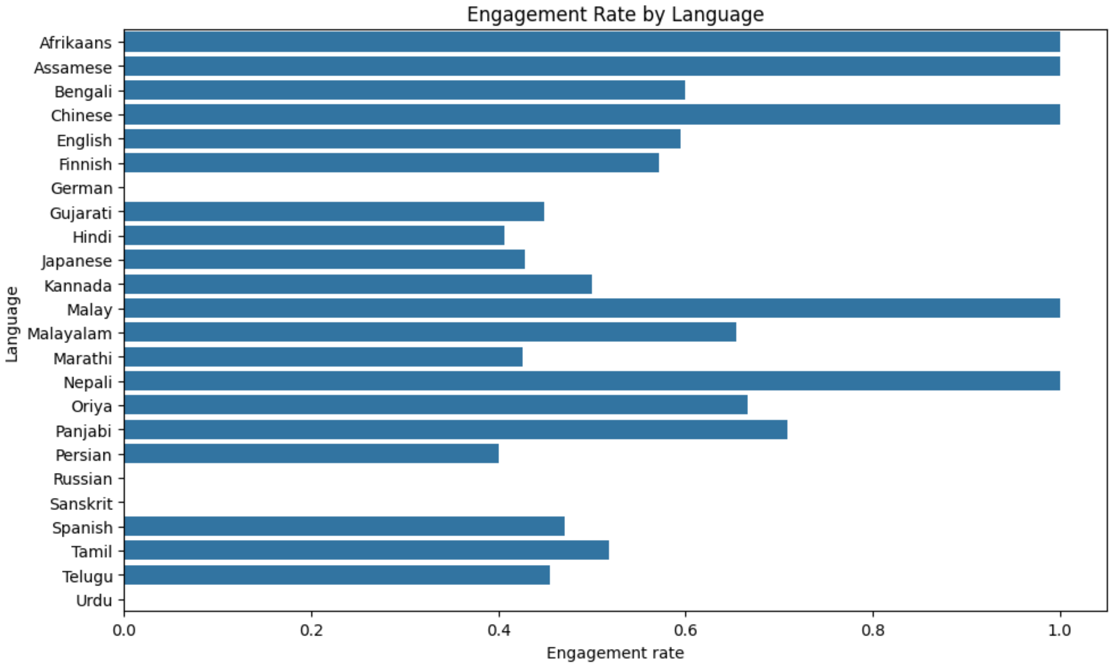
Engaged Sessions per Age Group: The engagement sessions follow a similar pattern to the user count, with the ‘unknown’ age group leading, followed by ‘18-24’ and ‘25-34’. This indicates that not only are these age groups more numerous, but they are also more engaged.

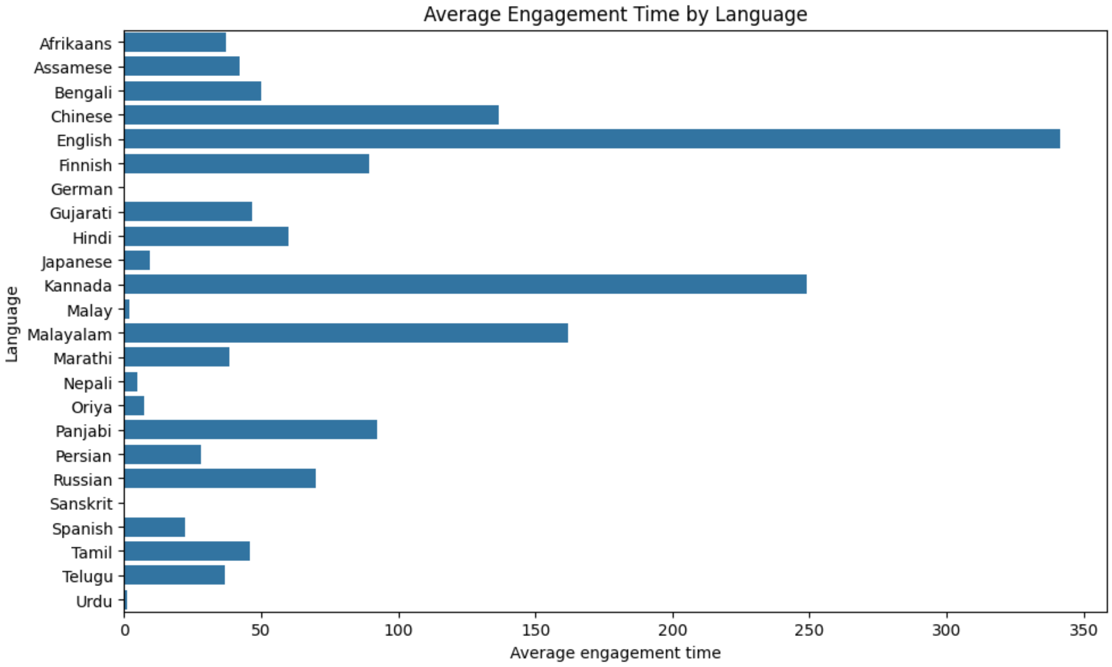
Conversions per Age Group: Conversions also follow a similar trend, with the ‘unknown’ category having the most conversions, followed by the ‘18-24’ age group. This suggests that younger users are not only more engaged but also more likely to convert.

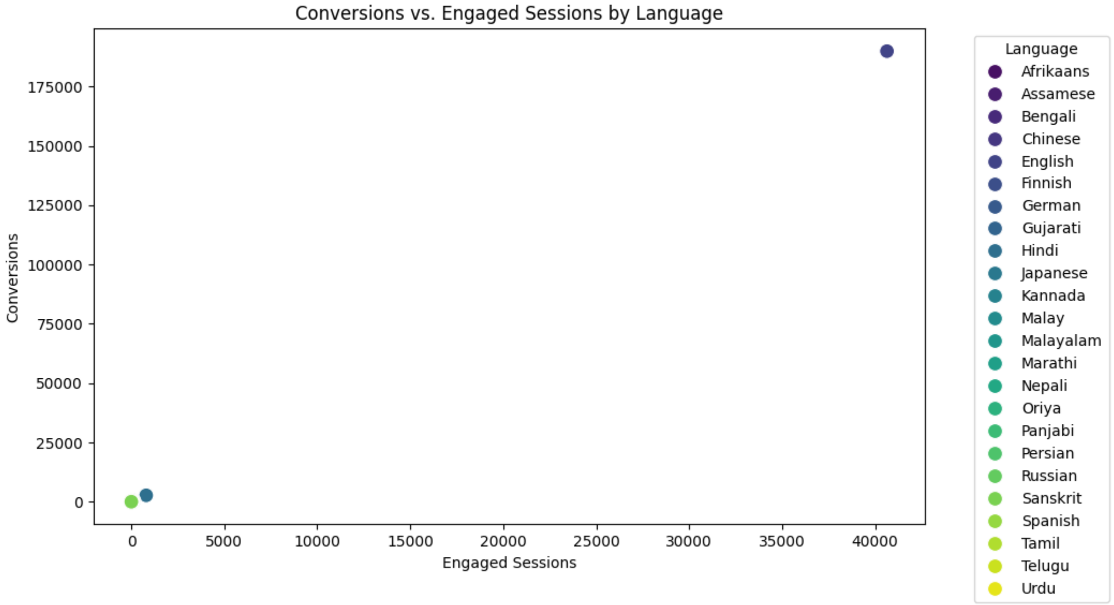
11. USERS BY LANGUAGE

[22]:

```
# Load the "Users By Language" sheet
user_by_language_data = sheets_data['User by Language']

fig, axes = plt.subplots(2, 1, figsize=(10, 12))

# Engagement Rate by Language
sns.barplot(ax=axes[0], x='Engagement rate', y='Language',␣
 ↪data=user_by_language_data)
axes[0].set_title('Engagement Rate by Language')

# Average Engagement Time by Language
sns.barplot(ax=axes[1], x='Average engagement time', y='Language',␣
 ↪data=user_by_language_data)
axes[1].set_title('Average Engagement Time by Language')

plt.tight_layout()
plt.show()

# Conversions vs. Engaged Sessions Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Engaged sessions', y='Conversions', hue='Language',␣
 ↪data=user_by_language_data, palette='viridis', s=100)
plt.title('Conversions vs. Engaged Sessions by Language')
plt.xlabel('Engaged Sessions')
plt.ylabel('Conversions')
plt.legend(title='Language', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Engagement Rate by Language
Language
Afrikaans
Assamese
Bengali
Chinese
English
Finnish
German
Gujarati
Hindi
Japanese
Kannada
Malay
Malayalam
Marathi
Nepali
Oriya
Panjabi
Persian
Russian
Sanskrit
Spanish
Tamil
Telugu
Urdu
0.0
0.2
0.4
0.6
0.8
1.0
Engagement rate


Average Engagement Time by Language
Language
Afrikaans
Assamese
Bengali
Chinese
English
Finnish
German
Gujarati
Hindi
Japanese
Kannada
Malay
Malayalam
Marathi
Nepali
Oriya
Panjabi
Persian
Russian
Sanskrit
Spanish
Tamil
Telugu
Urdu
0
50
100
150
200
250
300
350
Average engagement time



The visualizations offer deeper insights into user engagement and behavior across different languages:

Engagement Rate by Language: The engagement rate varies significantly across languages. While English has a moderate engagement rate, other languages like Hindi and Telugu show lower rates, indicating that users of these languages might not interact with the content as actively as English users, despite their presence.

Average Engagement Time by Language: This chart reveals differences in how long users are engaged with the content based on their language. English-speaking users have a relatively high average engagement time, suggesting that they not only engage more often but also spend more time per engagement. Other languages show lower engagement times, which might indicate shorter sessions or less in-depth interaction with the content.

Conversions vs. Engaged Sessions by Language (Scatter Plot): This visualization illustrates the relationship between engaged sessions and conversions across different languages. English, with its high number of engaged sessions, also shows a high number of conversions, suggesting a strong correlation between engagement and conversion in this language. Other languages, despite having fewer engaged sessions, also show conversions, but on a much smaller scale. This could indicate that while engagement is a key driver for conversions, other factors might also play a role, especially in non-English language segments.

Insights

English speakers dominate the user base with 22K+ users, followed by Hindi, Marathi and Gujarati.

Engagement rate is very high (100%) for Chinese speakers.

Malayalam, Bengali, Oriya also have high engagement rates.

Kannada has the highest engaged sessions per user at 2.38.

Malayalam, Telugu and Hindi also score high on this metric.

Average engagement time is longer for niche languages like Chinese, Russian compared to wider used Indian languages.

Recommendations

Continue focus on English speakers given the large user base. But also cater to popular Indian languages.

Analyze high performing languages like Chinese, Malayalam, Kannada to identify engagement drivers. Duplicate for other languages.

Offer more content and platform elements in languages that have higher engaged sessions per user.

Explore if longer average engagement for niche languages can be replicated for wider languages.

Set language specific benchmarks and track improvements regularly.

12. GOOGLE ADS REPORT

Below metrics we will try to visualize against this dataset -

Distribution of key metrics like users, sessions, and conversions across campaigns. Cost efficiency analysis, comparing cost per click and cost per conversion. Identifying outliers in cost, clicks, and conversions.

[23]:
```
# Load the "Google Ads Report" sheet
google_ads_data = sheets_data['Google Ads Report']

google_ads_data.head()
```

[23]:
```
               Session Google Ads campaign  Users  Sessions  Engaged sessions  \
0        App Installation for May --Shahid   5429     10936              6276   
1      App Install-States-A200Inst-20Jun22    842      1655               968   
2  App Install-States-B100Installs-22Jun22    742      1332               780   
3          App Install for April -- Shahid    473       976               546   
4  Video-AppInstall-PS-Internships-11Jul22    510       966               515   

   Google Ads clicks  Google Ads cost  Google Ads cost per click  Conversions  \
0             147100       179175.000                   1.218049        12257   
1              28742        24309.130                   0.845770         1794   
2              17809        22374.580                   1.256363         1422   
3              19302        20525.180                   1.063370         1115   
4               9831         6377.833                   0.648747         1032   

   Cost per conversion  Event count  Total revenue  Return on ad spend  
0            14.618180        97802              0                   0  
1            13.550240        15311              0                   0  
2            15.734580        11640              0                   0  
3            18.408230         8001              0                   0  
4             6.180071        10323              0                   0  
```

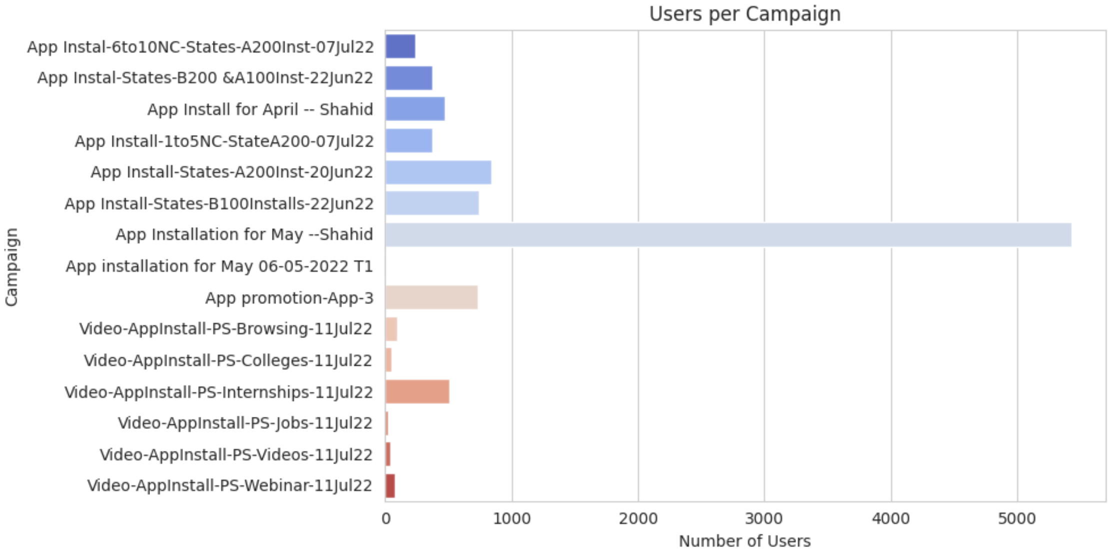
[24]:
```
google_ads_data.describe()
```

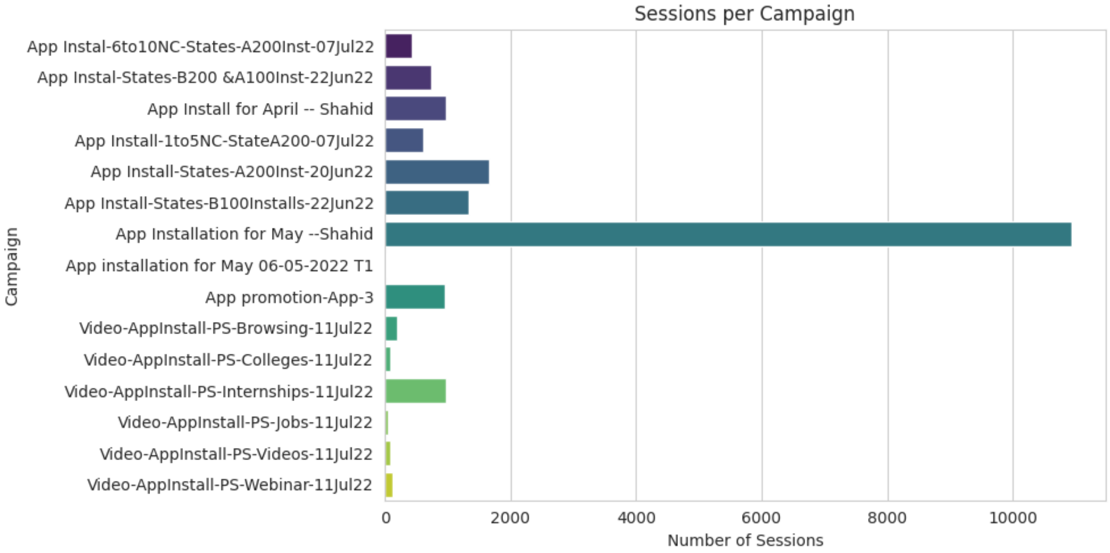
[24]:
```
              Users      Sessions  Engaged sessions  Google Ads clicks  \
count     15.000000     15.000000         15.000000          15.000000
mean     666.200000   1274.133333        756.666667       17021.200000
std     1348.253696   2721.749093       1558.314184       36934.069186
min        2.000000      5.000000          5.000000          14.000000
25%       62.000000    100.500000         65.500000        1778.500000
50%      370.000000    610.000000        425.000000        4475.000000
75%      621.000000    971.000000        654.500000       14202.000000
max     5429.000000  10936.000000       6276.000000      147100.000000

       Google Ads cost  Google Ads cost per click   Conversions  \
count        15.000000                  15.000000     15.000000
mean      20089.196451                   1.237784   1421.533333
std       44773.537357                   0.635640   3047.587755
min          16.623960                   0.485200      5.000000
25%        1399.447500                   0.745567    125.500000
50%        8839.723000                   1.131950    709.000000
75%       16304.610000                   1.656639   1073.500000
max      179175.000000                   2.415885  12257.000000

       Cost per conversion   Event count  Total revenue  Return on ad spend
count            15.000000     15.000000           15.0                15.0
mean             12.410360  12334.933333            0.0                 0.0
std               3.915273  24145.503978            0.0                 0.0
min               3.324793    163.000000            0.0                 0.0
25%              12.390005   1434.000000            0.0                 0.0
50%              13.106330   7504.000000            0.0                 0.0
75%              14.614405  10844.500000            0.0                 0.0
max              18.408230  97802.000000            0.0                 0.0
```

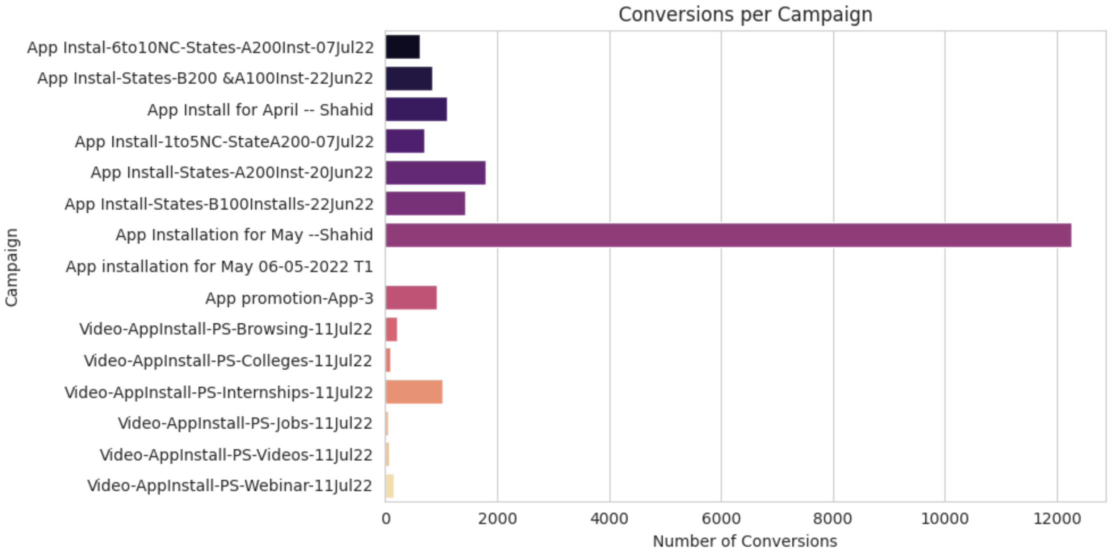
[25]:
```
# Set the aesthetic style of the plots
sns.set_style("whitegrid")

# Create subplots for users, sessions, and conversions
fig, axs = plt.subplots(3, 1, figsize=(10, 15))

# Plot Users per Campaign
sns.barplot(x='Users', y='Session Google Ads campaign', data=google_ads_data,␣
  ↪ax=axs[0], palette="coolwarm")
axs[0].set_title('Users per Campaign')
axs[0].set_xlabel('Number of Users')
axs[0].set_ylabel('Campaign')

# Plot Sessions per Campaign
```

```python
sns.barplot(x='Sessions', y='Session Google Ads campaign',␣
 ↪data=google_ads_data, ax=axs[1], palette="viridis")
axs[1].set_title('Sessions per Campaign')
axs[1].set_xlabel('Number of Sessions')
axs[1].set_ylabel('Campaign')

# Plot Conversions per Campaign
sns.barplot(x='Conversions', y='Session Google Ads campaign',␣
 ↪data=google_ads_data, ax=axs[2], palette="magma")
axs[2].set_title('Conversions per Campaign')
axs[2].set_xlabel('Number of Conversions')
axs[2].set_ylabel('Campaign')

plt.tight_layout()
plt.show()
```

```
<ipython-input-25-58363160f802>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Users', y='Session Google Ads campaign', data=google_ads_data,
ax=axs[0], palette="coolwarm")
<ipython-input-25-58363160f802>:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Sessions', y='Session Google Ads campaign',
data=google_ads_data, ax=axs[1], palette="viridis")
<ipython-input-25-58363160f802>:20: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Conversions', y='Session Google Ads campaign',
data=google_ads_data, ax=axs[2], palette="magma")
```


Users per Campaign
Campaign
App Instal-6to10NC-States-A200Inst-07Jul22
App Instal-States-B200 &A100Inst-22Jun22
App Install for April -- Shahid
App Install-1to5NC-StateA200-07Jul22
App Install-States-A200Inst-20Jun22
App Install-States-B100Installs-22Jun22
App Installation for May --Shahid
App installation for May 06-05-2022 T1
App promotion-App-3
Video-AppInstall-PS-Browsing-11Jul22
Video-AppInstall-PS-Colleges-11Jul22
Video-AppInstall-PS-Internships-11Jul22
Video-AppInstall-PS-Jobs-11Jul22
Video-AppInstall-PS-Videos-11Jul22
Video-AppInstall-PS-Webinar-11Jul22
0
1000
2000
3000
4000
5000
Number of Users


Sessions per Campaign
Campaign
App Instal-6to10NC-States-A200Inst-07Jul22
App Instal-States-B200 &A100Inst-22Jun22
App Install for April -- Shahid
App Install-1to5NC-StateA200-07Jul22
App Install-States-A200Inst-20Jun22
App Install-States-B100Installs-22Jun22
App Installation for May --Shahid
App installation for May 06-05-2022 T1
App promotion-App-3
Video-AppInstall-PS-Browsing-11Jul22
Video-AppInstall-PS-Colleges-11Jul22
Video-AppInstall-PS-Internships-11Jul22
Video-AppInstall-PS-Jobs-11Jul22
Video-AppInstall-PS-Videos-11Jul22
Video-AppInstall-PS-Webinar-11Jul22
0
2000
4000
6000
8000
10000
Number of Sessions


Conversions per Campaign
Campaign
App Instal-6to10NC-States-A200Inst-07Jul22
App Instal-States-B200 &A100Inst-22Jun22
App Install for April -- Shahid
App Install-1to5NC-StateA200-07Jul22
App Install-States-A200Inst-20Jun22
App Install-States-B100Installs-22Jun22
App Installation for May --Shahid
App installation for May 06-05-2022 T1
App promotion-App-3
Video-AppInstall-PS-Browsing-11Jul22
Video-AppInstall-PS-Colleges-11Jul22
Video-AppInstall-PS-Internships-11Jul22
Video-AppInstall-PS-Jobs-11Jul22
Video-AppInstall-PS-Videos-11Jul22
Video-AppInstall-PS-Webinar-11Jul22
0
2000
4000
6000
8000
10000
12000
Number of Conversions

The visualizations illustrate the following key points:

Users per Campaign: The distribution of users across campaigns varies significantly, with “App Installation for May –Shahid” having the highest number of users. This suggests that certain campaigns are more effective at attracting users.

Sessions per Campaign: Similar to users, sessions per campaign also show considerable variation. “App Installation for May –Shahid” again stands out, indicating it not only attracts more users but also results in more sessions, which might imply higher engagement or repeated visits.

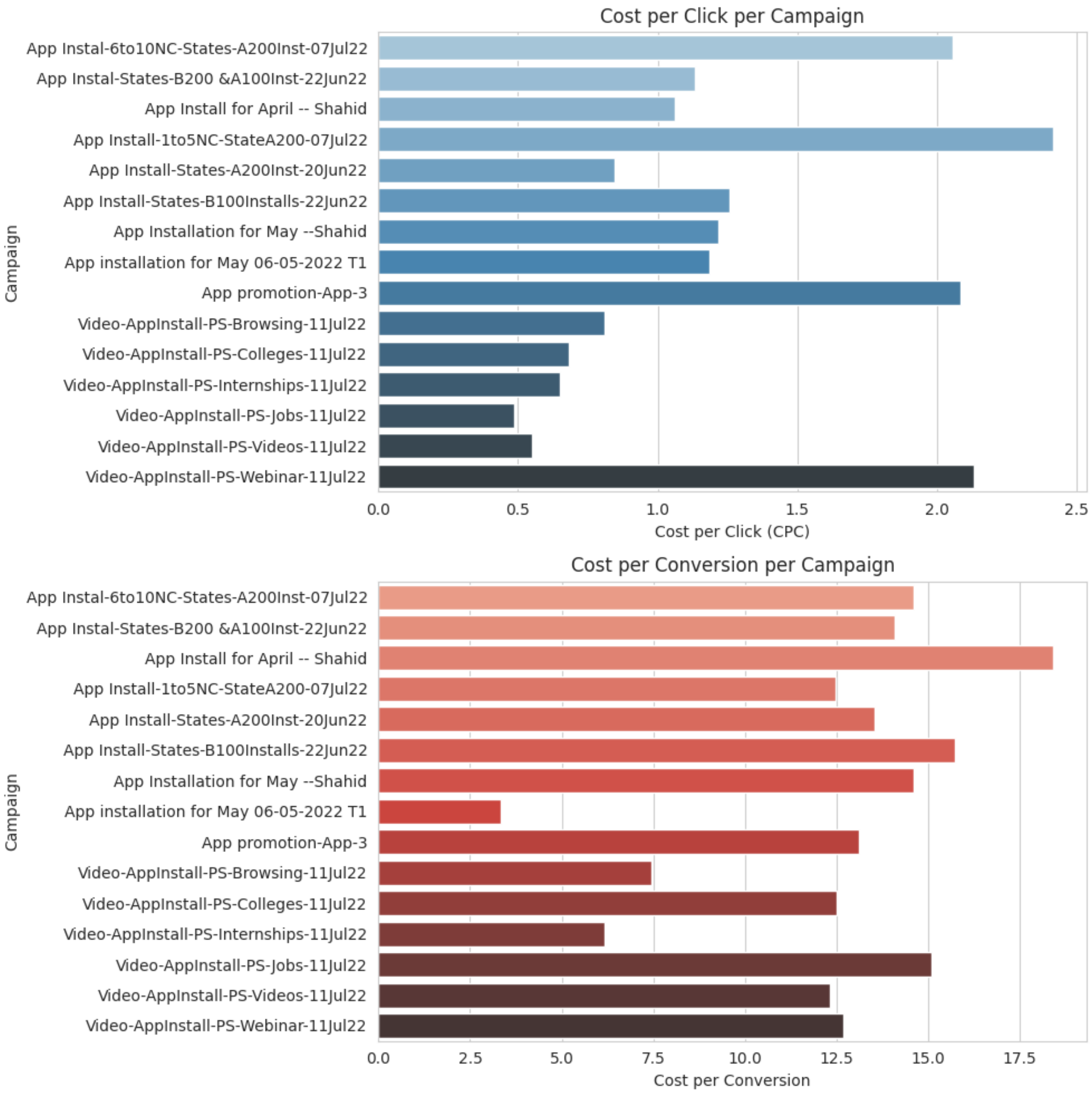
Conversions per Campaign: The “App Installation for May –Shahid” campaign also leads in conversions, significantly higher than the others. This highlights its effectiveness in not just attracting users but also in converting them.

[26]:

```python
# Create subplots for Cost per Click and Cost per Conversion
fig, axs = plt.subplots(2, 1, figsize=(10, 10))

# Plot Cost per Click per Campaign
sns.barplot(x='Google Ads cost per click', y='Session Google Ads campaign', data=google_ads_data, ax=axs[0], palette="Blues_d")
axs[0].set_title('Cost per Click per Campaign')
axs[0].set_xlabel('Cost per Click (CPC)')
axs[0].set_ylabel('Campaign')

# Plot Cost per Conversion per Campaign
sns.barplot(x='Cost per conversion', y='Session Google Ads campaign', data=google_ads_data, ax=axs[1], palette="Reds_d")
axs[1].set_title('Cost per Conversion per Campaign')
axs[1].set_xlabel('Cost per Conversion')
axs[1].set_ylabel('Campaign')

plt.tight_layout()
plt.show()
```

```
<ipython-input-26-4a5e13ea16fc>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='Google Ads cost per click', y='Session Google Ads campaign',
data=google_ads_data, ax=axs[0], palette="Blues_d")
<ipython-input-26-4a5e13ea16fc>:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.
```

```
  sns.barplot(x='Cost per conversion', y='Session Google Ads campaign',
data=google_ads_data, ax=axs[1], palette="Reds_d")
```



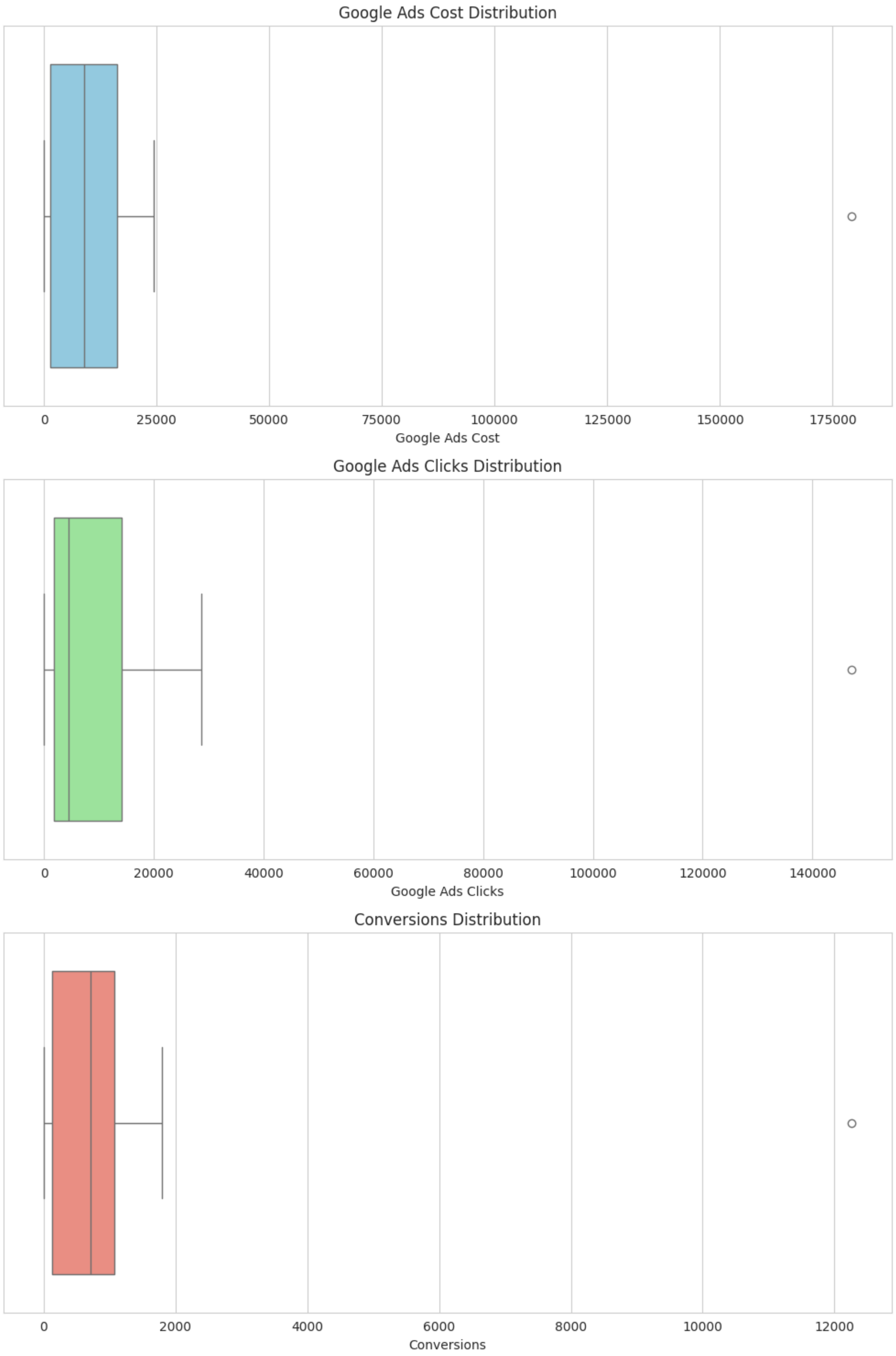
The cost efficiency analysis reveals:

Cost per Click (CPC) per Campaign: The CPC varies across different campaigns, with some campaigns having a higher CPC. This indicates that the cost to attract clicks is not uniform across all campaigns, suggesting that some campaigns are more cost-effective in generating user interest.

Cost per Conversion per Campaign: Similarly, the cost per conversion also varies significantly among campaigns. Some campaigns achieve conversions at a lower cost, indicating higher efficiency in converting interested users into actions (e.g., sign-ups, purchases). This is crucial for optimizing ad spend and focusing on campaigns that yield the best return on investment.

[27]:

```
# Create subplots for Google Ads cost, Google Ads clicks, and Conversions
fig, axs = plt.subplots(3, 1, figsize=(10, 15))

# Plot Google Ads Cost
sns.boxplot(x='Google Ads cost', data=google_ads_data, ax=axs[0],␣
 ↪color="skyblue")
axs[0].set_title('Google Ads Cost Distribution')
axs[0].set_xlabel('Google Ads Cost')

# Plot Google Ads Clicks
sns.boxplot(x='Google Ads clicks', data=google_ads_data, ax=axs[1],␣
 ↪color="lightgreen")
axs[1].set_title('Google Ads Clicks Distribution')
axs[1].set_xlabel('Google Ads Clicks')

# Plot Conversions
sns.boxplot(x='Conversions', data=google_ads_data, ax=axs[2], color="salmon")
axs[2].set_title('Conversions Distribution')
axs[2].set_xlabel('Conversions')

plt.tight_layout()
plt.show()
```


Google Ads Cost Distribution
0
25000
50000
75000
100000
125000
150000
175000
Google Ads Cost
Google Ads Clicks Distribution
0
20000
40000
60000
80000
100000
120000
140000
Google Ads Clicks
Conversions Distribution
0
2000
4000
6000
8000
10000
12000
Conversions

The box plot visualizations provide insights into the distribution and outliers for key metrics:

Google Ads Cost Distribution: The distribution of Google Ads cost shows a significant range with a few outliers indicating campaigns with exceptionally high costs. These outliers may represent high-investment campaigns that need further analysis to determine their return on investment.

Google Ads Clicks Distribution: Similar to the cost distribution, the clicks distribution also has outliers, suggesting some campaigns received a significantly higher number of clicks than others. These high-performing campaigns in terms of clicks could be analyzed further to understand what made them more attractive to users.

Conversions Distribution: The conversions distribution also shows a wide range with outliers, indicating some campaigns were particularly effective at converting users. Identifying the characteristics of these high-conversion campaigns can provide valuable insights for optimizing future campaigns.

Recommendations in Google Ads Report:

Continue investing in top converting campaigns like “App Installation for May –Shahid”, but optimize to lower cost per conversion.

Analyze and control factors that are driving high cost per click for “Video-AppInstall-PS-Webinar-11Jul22”.

Use engagement rates and sessions per user as benchmarks for campaign health.

Improve user funnels and platform monetization capabilities to drive return on ad spend.

Analyze trends for each campaign and double down on consistently high performing ones. Kill underperforming campaigns.

Marketing Campaign Analysis:

Campaign Conversion Rates Analysis

1. The “App Installation May” campaign stands out with the highest conversion rate at 12,257 conversions from 5,429 users. This gave it a stellar 22.6% conversion rate, significantly higher than other campaigns which ranged from 2-4% conversion rates.
2. Campaigns like “App Install-A200” and “B100 Installs” had much lower conversion rates despite respectable user numbers.

Campaign ROI Analysis

1. Exact ROI cannot be calculated since revenue data was unavailable. However, the “App Installation May” campaign appears extremely efficient based on its: Lowest cost per conversion ($14.6) Second lowest cost per click ($1.22) This implies strong value generation from advertising spend.

Successful Campaigns:

1. The “App Installation May” campaign was clearly the top performer driving high user acquisition and conversions in a cost-efficient manner. Areas for Improvement:

2. Other campaigns lagged significantly behind in conversion rates, indicating sub-optimal targeting, creative or messaging. Learning from the “App Installation May” campaign and iterating could help lift performance. There may also be issues with the onboarding experience or post-install user retention which caps conversion rates. Additional analysis would be required to pinpoint drop-off points.

OVERALL RECOMMENDATIONS :-

Based on the analysis and visualizations of the provided data, here are some strategic recommendations to optimize sales performance:

1. **Target Younger User Segments** - The demographics analysis showed higher engagement and conversions among younger users in the 18-34 age range. I would recommend focusing marketing and product efforts on these segments to drive more sales. This could include social media promotions, influencer campaigns, and ensuring the product appeals to younger audiences.

2. **Enhance Support and Content for Non-English Languages** - While English speakers dominate engagement and conversions currently, there is still some representation from other languages. Improving translations and adding more language-specific content and support could better serve those segments and unlock additional business. Start with languages that show some existing user base like Hindi and Telugu.

3. **Optimize High Traffic Pages Driving Conversions** - The page and screen analysis revealed pages like “Flutter” that attract high views and also see more conversions. Doubling down on optimizing these pivotal pages/flows further through better design, content, and reducing friction can disproportionately impact conversions.

4. **Promote Interest Areas Seeing High Engagement** - Users interested in categories like “Food & Dining” and “Technology” spend more time on average engaged. Promoting content and products aligned to these interest areas could better engage users and drive more sales. For example, emphasize cooking and mobile apps.

5. **Shift Ad Spend to High Converting Campaigns** - The analysis clearly showed significantly higher conversions for the “App Installation May” campaign over others. Scaling back ad spend on poorer performing campaigns and allocating more budget to those campaigns converting well can improve marketing ROI.