Portfolio Traffic Analysis using Google Analytics and Snowflake

PROJECT LINK

This project involves extracting traffic data from Google Analytics for my portfolio website, transforming the data, and loading it into Snowflake for analysis and visualization.

In [1]:
```
# Import necessary libraries
import os
import pandas as pd
import numpy as np
import snowflake.connector
from snowflake.connector import DictCursor
from snowflake.connector.pandas_tools import write_pandas
from google.analytics.data_v1beta import BetaAnalyticsDataClient
from google.analytics.data_v1beta.types import DateRange, Dimension, Metric, RunR
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Set environment variables for Google Analytics credentials
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'Credentials_Path'
```

In [ ]:
```
# Google Analytics API setup [Update them with your Credentials]
property_id = "*********"
client = BetaAnalyticsDataClient()
```

Part 1: Data Extraction from Google Analytics

In [4]:
```
# Function to run report and convert to DataFrame
def run_report_to_df(property_id, dimensions, metrics, start_date="2023-01-01", e
    request = RunReportRequest(
        property=f"properties/{property_id}",
        dimensions=[Dimension(name=dim) for dim in dimensions],
        metrics=[Metric(name=metric) for metric in metrics],
        date_ranges=[DateRange(start_date=start_date, end_date=end_date)],
    )
    response = client.run_report(request)

    if not response.rows:
        print("No data returned from the API.")
        return pd.DataFrame()
```

```
    rows = []
    for row in response.rows:
        dimension_values = [dim.value for dim in row.dimension_values]
        metric_values = [metric.value for metric in row.metric_values]
        rows.append(dimension_values + metric_values)

    return pd.DataFrame(rows, columns=dimensions + metrics)
```

In [5]:
```
# Attributes to extract
dimensions = [
    "date",
    "country",
    "city",
    "newvsreturning",
    "firstUserSource",
    "firstUserMedium",
    "deviceCategory",
    "percentscrolled",
    "dayofweekname"
]
```

In [6]:
```
metrics = [
    "activeUsers",
    "newUsers",
    "screenPageViews",
    "sessions",
    "averageSessionDuration",
    "bounceRate",
    "EngagedSessions",
    "eventCount",
    "eventCountPerUser",
    "UserEngagementDuration"
]
```

In [7]:
```
# Extract data
df_all_data = run_report_to_df(
    property_id,
    dimensions=dimensions,
    metrics=metrics,
    start_date="2023-01-01",
    end_date="today"
)
```

Part 2: Data Transformation and Loading into Snowflake

In [8]:
```
# Snowflake connection parameters [Update them with your Credentials]
snowflake_account = '*****'
snowflake_username = '*****'
snowflake_password = '*****'
snowflake_role = '*****'
snowflake_warehouse = '*****'
```

In [9]:
```
# Connect to Snowflake
conn = snowflake.connector.connect(
    user=snowflake_username,
    password=snowflake_password,
    account=snowflake_account,
    role=snowflake_role,
    warehouse=snowflake_warehouse
)
```

In [10]:
```
# Create a cursor object
cur = conn.cursor(DictCursor)
```

In [11]:
```
# Create database, schema, and table in Snowflake
cur.execute("CREATE DATABASE IF NOT EXISTS Portfolio_Traffic")
cur.execute("USE DATABASE Portfolio_Traffic")
cur.execute("CREATE SCHEMA IF NOT EXISTS User")
cur.execute("USE SCHEMA User")
```

Out[11]:
```
<snowflake.connector.cursor.DictCursor at 0x7f98714bd3a0>
```

In [14]:
```
# Define and execute table creation query
create_table_query = """
CREATE OR REPLACE TABLE portfolio_analysis (
    date DATE,
    country STRING,
    city STRING,
    newvsreturning STRING,
    firstUserSource STRING,
    firstUserMedium STRING,
    deviceCategory STRING,
    percentscrolled FLOAT,
    dayofweekname STRING,
    activeUsers NUMBER,
    newUsers NUMBER,
    screenPageViews NUMBER,
    sessions NUMBER,
    averageSessionDuration FLOAT,
    bounceRate FLOAT,
    EngagedSessions NUMBER,
    eventCount NUMBER,
    eventCountPerUser FLOAT,
    UserEngagementDuration FLOAT
)
"""
```

In [15]:
```
cur.execute(create_table_query)
```

Out[15]:
```
<snowflake.connector.cursor.DictCursor at 0x7f98714bd3a0>
```

In [16]:
```
# Data preparation and type conversion for Snowflake
df_all_data.columns = map(str.upper, df_all_data.columns)
```

In [17]:
```
# List of columns that should be numeric
numeric_columns = [
    'PERCENTSCROLLED',
    'SESSIONS',
    'ACTIVEUSERS',
    'NEWUSERS',
    'AVERAGESESSIONDURATION',
    'BOUNCERATE',
    'ENGAGEDSESSIONS',
    'EVENTCOUNT',
    'EVENTCOUNTPERUSER',
    'USERENGAGEMENTDURATION'
]


# Convert empty strings to NaN for numeric columns
for column in numeric_columns:
    df_all_data[column] = pd.to_numeric(df_all_data[column], errors='coerce')

# Replace NaN with NULL (for Snowflake)
df_all_data = df_all_data.replace({np.nan: None})

df_all_data['DATE'] = pd.to_datetime(df_all_data['DATE'], format='%Y%m%d').dt.dat
```

In [18]:
```
# Load data into Snowflake
success, nchunks, nrows, _ = write_pandas(conn, df_all_data, 'PORTFOLIO_ANALYSIS'
```

In [19]:
```
# Close the cursor and the connection
cur.close()
conn.close()
```

In [20]:
```
# Data Verification in Snowflake
```

In [21]:
```
# Re-establish Snowflake connection
ctx = snowflake.connector.connect(
    user=snowflake_username,
    password=snowflake_password,
    account=snowflake_account,
    role=snowflake_role,
    warehouse=snowflake_warehouse,
    database='PORTFOLIO_TRAFFIC',
    schema='User'
)

# Create a cursor object
cur = ctx.cursor()
```

In [22]:
```
# Simple query to display a few rows for verification
query = "SELECT * FROM PORTFOLIO_ANALYSIS LIMIT 3;"
cur.execute(query)
df_results = cur.fetch_pandas_all()
print("Sample data from Snowflake:")
print(df_results)
```

```
Sample data from Snowflake:
         DATE        COUNTRY       CITY NEWVSRETURNING FIRSTUSERSOURCE  \
0  2023-12-17        Algeria  (not set)            new         lnkd.in
1  2023-12-13  United States  (not set)            new        (direct)
2  2023-11-06          India     Mumbai            new         lnkd.in

  FIRSTUSERMEDIUM DEVICECATEGORY  PERCENTSCROLLED DAYOFWEEKNAME  ACTIVEUSERS  \
0        referral         mobile              NaN        Sunday            9
1          (none)        desktop              NaN     Wednesday            7
2        referral         mobile              NaN        Monday            6

   NEWUSERS  SCREENPAGEVIEWS  SESSIONS  AVERAGESESSIONDURATION  BOUNCERATE  \
0         9                9         9                5.859284    0.666667
1         7                7         7                1.140085    0.857143
2         6                6         6               11.301259    0.500000

   ENGAGEDSESSIONS  EVENTCOUNT  EVENTCOUNTPERUSER  USERENGAGEMENTDURATION
0                3          31           3.444444                    11.0
1                1          22           3.142857                    12.0
2                3          23           3.833333                    19.0
```

In [44]:
```
df_results
```

Out[44]:

	DATE	COUNTRY	CITY	NEWVSRETURNING	FIRSTUSERSOURCE	FIRSTUSERMEDIUM	DEVICE
0	2023-12-17	Algeria	(not set)	new	lnkd.in	referral	
1	2023-12-13	United States	(not set)	new	(direct)	(none)	
2	2023-11-06	India	Mumbai	new	lnkd.in	referral	

In [23]:
```
# Query to count the total number of rows for verification
query_count = "SELECT COUNT(*) FROM PORTFOLIO_ANALYSIS;"
```

```
cur.execute(query_count)
count_result = cur.fetchone()
print(f"Total number of rows in PORTFOLIO_ANALYSIS TABLE: {count_result[0]}")
```

```
Total number of rows in PORTFOLIO_ANALYSIS TABLE: 8184
```

In [24]:
```
# Close the cursor and the connection
cur.close()
conn.close()
```

Part 3: In-depth Data Analysis and Visualization

This section delves into the analysis of four distinct datasets extracted from Google Analytics. Due to the rate limit constraints of the Google Analytics API, the data is segmented into four sets, each focusing on different dimensions and metrics to provide a comprehensive understanding of user behavior and site performance. Visualization techniques are employed to represent the data effectively, providing clear insights.

In [25]:
```
# Set 1 - User Session and Geographic Information Analysis
```

In [26]:
```
dimensions_set1 = [
    "date",
    "country",
    "region",
    "city",
    "continent",
    "firstUserSource",
    "firstUserMedium",
    "deviceCategory",
    "FirstUserDefaultChannelGroup"
]

metrics_set1 = [
    "sessions",
    "activeUsers",
    "newUsers",
    "averageSessionDuration",
    "bounceRate",
    "EngagedSessions",
    "UserEngagementDuration"
]
```

In [27]:
```
df_set1 = run_report_to_df(
    property_id,
    dimensions=dimensions_set1,
    metrics=metrics_set1,
    start_date="2023-01-01",
    end_date="today"
)
print(df_set1.head())
```

```
       date        country            region        city continent  \
0  20231217        Algeria  Algiers Province   (not set)    Africa
1  20231213  United States         (not set)   (not set)  Americas
2  20231106          India       Maharashtra      Mumbai      Asia
3  20231106  United States        Washington     Seattle  Americas
4  20231113  United States              Iowa  Des Moines  Americas

  firstUserSource firstUserMedium deviceCategory FirstUserDefaultChannelGroup  \
0         lnkd.in        referral         mobile                Organic Social
1        (direct)          (none)        desktop                        Direct
2         lnkd.in        referral         mobile                Organic Social
3        (direct)          (none)         mobile                        Direct
4        (direct)          (none)        desktop                        Direct

  sessions activeUsers newUsers averageSessionDuration           bounceRate  \
0        9           9        9     7.3189108888888894  0.66666666666666663
1        7           7        7     1.9624251428571429   0.8571428571428571
2        6           6        6     12.945288833333331                  0.5
3        6           5        5     30.610651166666667  0.83333333333333337
4        6           6        6     7.7935759999999989                  0.5

  EngagedSessions UserEngagementDuration
0               3                     11
1               1                     12
2               3                     19
3               1                      0
4               3                     43
```

In [46]:
```
df_set1.sample(2)
```

Out[46]:

	date	country	region	city	continent	firstUserSource	firstUserMedium	deviceCat
1579	2023-11-07	Germany	Bavaria	Munich	Europe	lnkd.in	referral	
689	2023-04-24	United States	Wisconsin	Marinette	Americas	(direct)	(none)	

In [47]:
```
df_set1.firstUserSource.value_counts()
```

Out[47]:
```
(direct)                        2779
linkedin.com                    1047
lnkd.in                          625
github.com                        27
m.facebook.com                     2
evernote.com                       1
google                             1
akshayakalpa.kapturecrm.com        1
trello.com                         1
(not set)                          1
Name: firstUserSource, dtype: int64
```

In [29]:
```
# Set 2 - Page Information and User Behavior Analysis
```

In [30]:
```
dimensions_set2 = [
    "date",
    "pageTitle",
    "pagePath",
    "devicemodel",
    "browser",
    "newvsreturning",
    "percentscrolled",
    "linktext",
    "linkurl"
]

metrics_set2 = [
```

```
    "screenPageViews",
    "sessions",
    "eventCount",
    "eventCountPerUser",
    "ItemListCLickEvents",
    "ItemListViewEvents",
    "UserEngagementDuration"
]
```

In [31]:
```
df_set2 = run_report_to_df(
    property_id,
    dimensions=dimensions_set2,
    metrics=metrics_set2,
    start_date="2023-01-01",
    end_date="today"
)
print(df_set2.head())
```

```
       date                       pageTitle                          pagePath  \
0  20231117  Abhishek Chandragiri Portfolio  /Abhishek-Chandragiri-Portfolio/
1  20231105  Abhishek Chandragiri Portfolio  /Abhishek-Chandragiri-Portfolio/
2  20240110  Abhishek Chandragiri Portfolio  /Abhishek-Chandragiri-Portfolio/
3  20231113  Abhishek Chandragiri Portfolio  /Abhishek-Chandragiri-Portfolio/
4  20240122  Abhishek Chandragiri Portfolio  /Abhishek-Chandragiri-Portfolio/

  devicemodel browser newvsreturning percentscrolled linktext linkurl  \
0   (not set)  Chrome            new
1   (not set)  Chrome            new
2   (not set)  Chrome            new
3   (not set)  Chrome            new
4   (not set)  Chrome            new

  screenPageViews sessions eventCount   eventCountPerUser ItemListCLickEvents  \
0             134      125        458              3.664                   0
1              87       77        283  3.6753246753246751                   0
2              78       66        255  3.8636363636363638                   0
3              74       64        250             3.90625                   0
4              64       54        205  3.7962962962962963                   0

  ItemListViewEvents UserEngagementDuration
0                  0                   1186
1                  0                    734
2                  0                    945
3                  0                    752
4                  0                    538
```

In [49]:
```
df_set2.sample(2)
```

Out[49]:

	date	pageTitle	pagePath	devicemodel	browser	newvsreturning	percentscrolled	I
2251	20231129	Abhishek Chandragiri Portfolio	/Abhishek-Chandragiri-Portfolio/	(not set)	Safari	new	90	
2108	20231122	Abhishek Chandragiri Portfolio	/Abhishek-Chandragiri-Portfolio/	(not set)	Chrome	new		

In [48]:
```
df_set2['percentscrolled'].value_counts()
```

Out[48]:
```
      2483
90     844
Name: percentscrolled, dtype: int64
```

In [50]:
```
df_set2.linktext.value_counts()
```

Out[50]:
```
          3309
Resume      18
Name: linktext, dtype: int64
```

In [51]:
```
r = df_set2[df_set2['linktext'] == 'Resume']
```

In [52]:
```
r.linkurl
```

Out[52]:
```
1031    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1053    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1056    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1084    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1109    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1155    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1156    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1169    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1232    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1305    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1328    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1343    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1365    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1366    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1402    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1437    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1486    https://github.com/Abhi0323/RESUME_LATEX/blob/...
1531    https://github.com/Abhi0323/RESUME_LATEX/blob/...
Name: linkurl, dtype: object
```

In [33]:
```
# Set 3 - Detailed User Session and Time Analysis
```

In [34]:
```
dimensions_set3 = [
    "date",
    "datehourminute",
    "hour",
    "day",
    "dayofweek",
    "dayofweekname",
    "FirstSessionDate",
    "firstusersourcemedium",
    "appVersion"
]

metrics_set3 = [
    "sessions",
    "activeUsers",
    "EngagedSessions",
    "averageSessionDuration",
    "UserEngagementDuration",
    "newUsers"
]
```

In [35]:
```
df_set3 = run_report_to_df(
    property_id,
    dimensions=dimensions_set3,
    metrics=metrics_set3,
    start_date="2023-01-01",
    end_date="today"
)
print(df_set3.head())
```

```
       date datehourminute hour day dayofweek dayofweekname FirstSessionDate  \
0  20231213   202312132042   20  13         3     Wednesday         20231213
1  20231217   202312170348    3  17         0        Sunday         20231217
2  20240125   202401250133    1  25         4      Thursday         20240125
3  20231201   202312012032   20  01         5        Friday         20231201
4  20231213   202312130941    9  13         3     Wednesday         20231213

  firstusersourcemedium appVersion sessions activeUsers EngagedSessions  \
0     (direct) / (none)  (not set)        5           5               1
1    lnkd.in / referral  (not set)        4           4               1
2     (direct) / (none)  (not set)        4           4               1
3     (direct) / (none)  (not set)        3           3               2
4     (direct) / (none)  (not set)        3           3               0

  averageSessionDuration UserEngagementDuration newUsers
0             11.7911136                     52        4
1     4.7804505000000006                      4        4
2     7.9615934999999993                     17        4
3     150.93052733333334                     11        2
4     1.3088330000000001                      0        3
```

In [55]:
```
df_set3.sample(2)
```

Out[55]:

	date	datehourminute	hour	day	dayofweek	dayofweekname	FirstSessionDate	firstus
3815	20240128	202401281431	14	28	0	Sunday	20240128	
1762	20240101	202401010257	2	01	1	Monday	20240101	

In [36]:
```
# Set 4 - Language and Regional Preferences Analysis
```

In [37]:
```
dimensions_set4 = [
    "date",
    "country",
    "city",
    "languageCode",
    "browser",
    "deviceCategory",
    "FirstUserMedium",
    "pagereferrer",
    "region"
]

metrics_set4 = [
    "sessions",
    "activeUsers",
    "newUsers",
    "averageSessionDuration",
    "bounceRate",
    "TotalUsers"
]
```

In [38]:
```
df_set4 = run_report_to_df(
    property_id,
    dimensions=dimensions_set4,
    metrics=metrics_set4,
    start_date="2023-01-01",
    end_date="today"
)
print(df_set4.head())
```

```
       date        country        city languageCode          browser  \
0  20231217        Algeria   (not set)        en-us  Android Webview
1  20231106          India      Mumbai        en-in  Android Webview
2  20231106  United States     Seattle        en-us           Safari
3  20231113  United States  Des Moines        en-us           Chrome
4  20231206  United States     Raleigh        en-us           Chrome

  deviceCategory FirstUserMedium     pagereferrer            region sessions  \
0         mobile        referral  http://lnkd.in/  Algiers Province        9
1         mobile        referral  http://lnkd.in/       Maharashtra        6
2         mobile          (none)                        Washington        6
3        desktop          (none)                              Iowa        6
4        desktop          (none)                    North Carolina        6

  activeUsers newUsers averageSessionDuration          bounceRate TotalUsers
0           9        9     7.3189108888888894  0.66666666666666663          9
1           6        6     12.945288833333331                  0.5          6
2           5        5     30.610651166666667  0.83333333333333337          5
3           6        6     7.7935759999999989                  0.5          6
4           2        1     270.93300016666666                  0.5          2
```

In [54]:
```
df_set4.sample(2)
```

Out[54]:

	date	country	city	languageCode	browser	deviceCategory	FirstUserMedium	
2899	20231204	Egypt	New Cairo City	en-us	Edge	desktop	(none)	
3360	20231220	India	Hyderabad	en-us	Chrome	desktop	referral	h

In [53]:
```
df_set4.pagereferrer.value_counts()
```

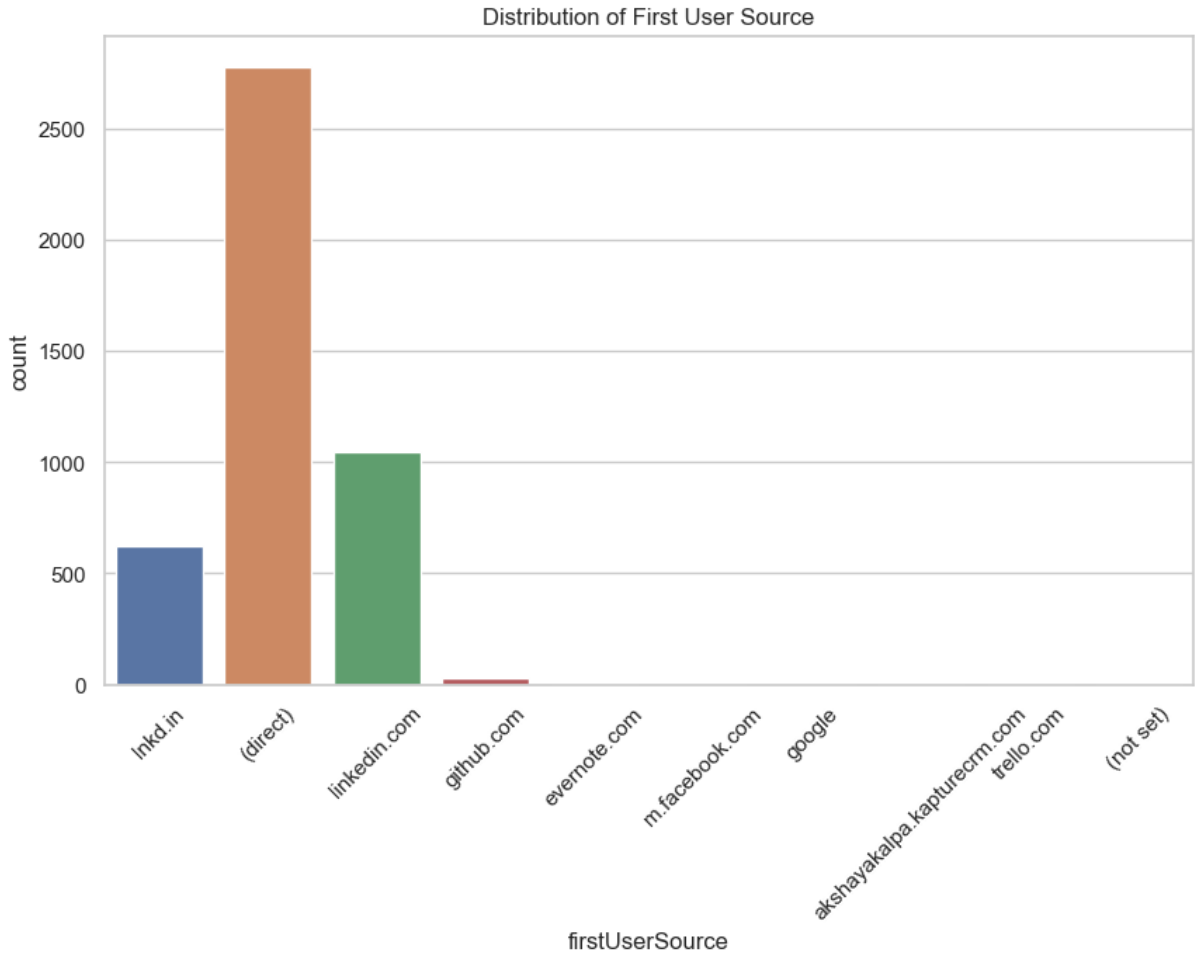
Out[53]:

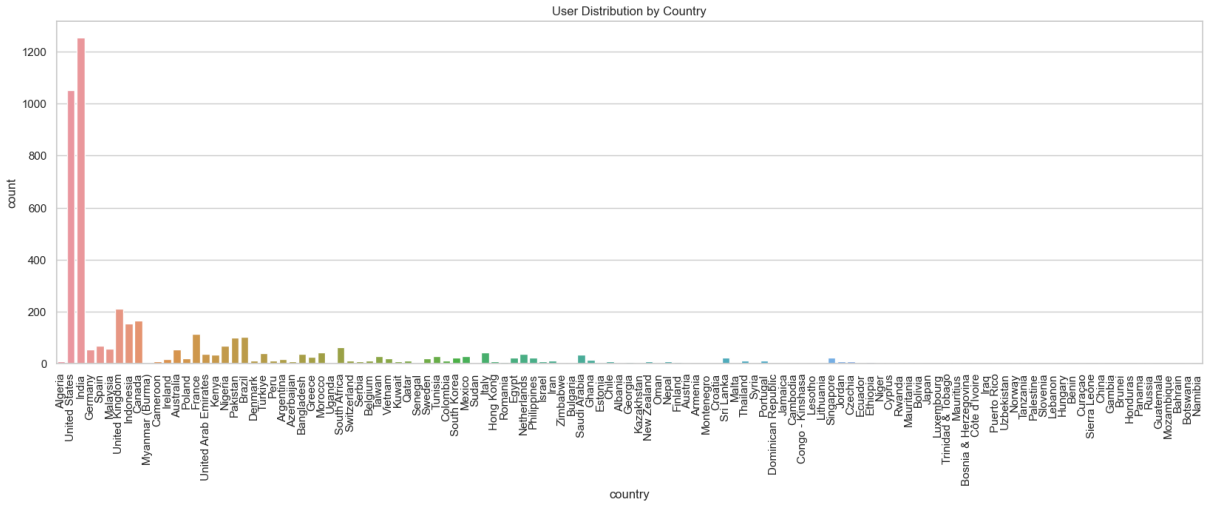
```
1786
android-app://com.linkedin.android/
1150
https://www.linkedin.com/
1092
http://lnkd.in/
655
https://github.com/
36
https://github.com/Abhi0323
12
https://github.com/Abhi0323/Abhishek-Chandragiri-Portfolio
10
android-app://com.linkedin.android
3
https://www.google.com/
3
https://www.overleaf.com/
3
https://github.com/Abhi0323/Abhi0323/blob/main/README.md
2
http://m.facebook.com/
2
https://abhi0323.github.io/Abhishek-Chandragiri-Portfolio/
2
https://www.linkedin.com/redir/redirect?url=https%3A%2F%2Fabhi0323%2Egithub%2Eio%
2FAbhishek-Chandragiri-Portfolio%2F&urlhash=wUzK&trk=contact-info       2
https://www.evernote.com/
1
http://127.0.0.1:5500/index.html
1
https://github.com/Abhi0323/Abhishek-Chandragiri-Portfolio/tree/main
1
https://github.com/Abhi0323?tab=overview&from=2023-11-01&to=2023-11-04
1
https://l.facebook.com/
1
https://akshayakalpa.kapturecrm.com/
1
https://github.com/Abhi0323/Abhi0323
1
https://github.com/Abhi0323/Abhishek-Chandragiri-Portfolio/blob/main/README.md
1
https://linkedin.com/
1
https://trello.com/
1
https://github.com/Abhi0323/Abhishek-Chandragiri-Portfolio/deployments/github-pag
es                                                                      1
Name: pagereferrer, dtype: int64
```

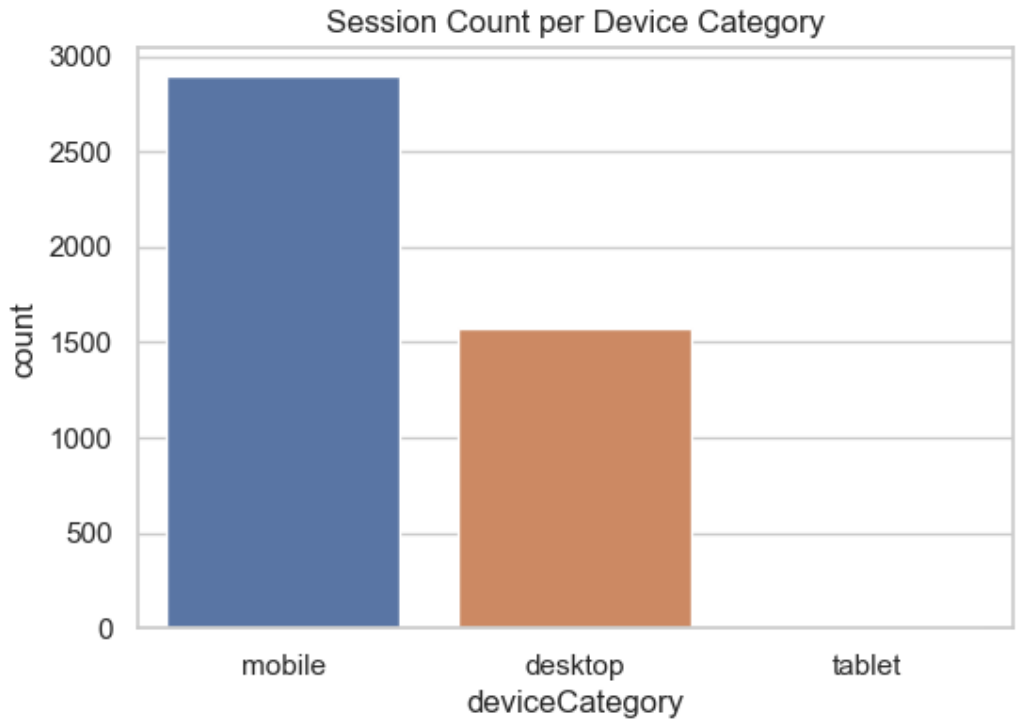
In [43]:

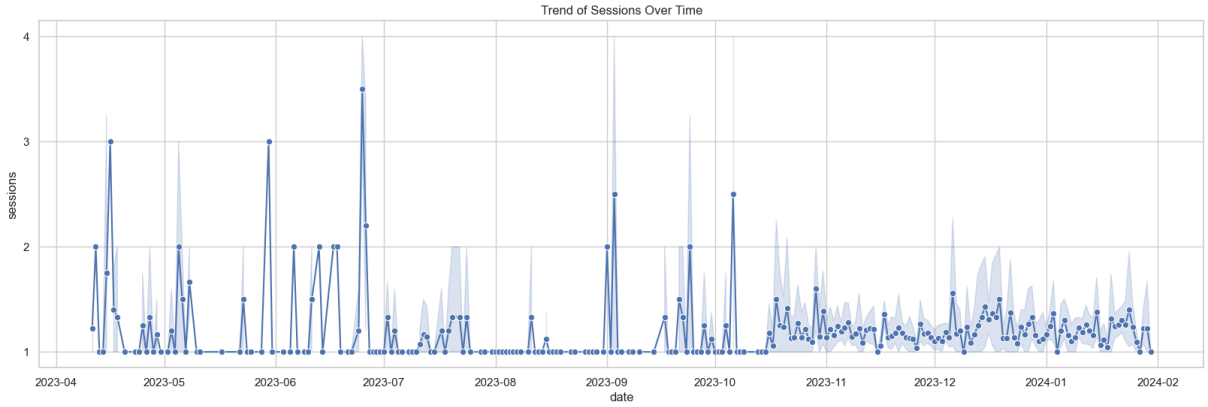
```
# Visualization settings for aesthetics
sns.set(style="whitegrid")  # Setting the seaborn style for consistent aesthetics

"""
Visualization for Set 1: User Session and Geographic Information
These visualizations focus on understanding the sources of traffic, the geographi
and the devices used to access the portfolio site.
"""

# a. Channels and Sources Visualization
# This chart shows the distribution of the first user source, providing insights
plt.figure(figsize=(10, 6))
sns.countplot(data=df_set1, x='firstUserSource')
plt.title('Distribution of First User Source')
plt.xticks(rotation=45)
plt.show()
```

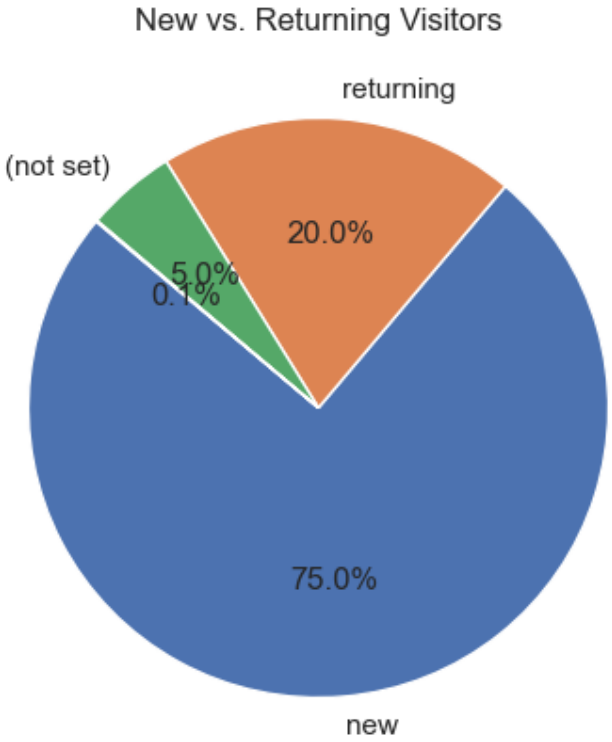
```python
# b. Geographic Distribution Visualization (example for countries)
# This chart presents the user distribution by country, highlighting where the si
plt.figure(figsize=(20, 6))
sns.countplot(data=df_set1, x='country')
plt.title('User Distribution by Country')
plt.xticks(rotation=90)
plt.show()

# c. Device Usage Visualization
# This chart depicts the session count per device category, showing the device pr
plt.figure(figsize=(6, 4))
sns.countplot(data=df_set1, x='deviceCategory')
plt.title('Session Count per Device Category')
plt.show()

# d. Engagement Metrics Analysis (example for sessions trend)
# This line chart shows the trend of sessions over time, providing insights into
df_set1['date'] = pd.to_datetime(df_set1['date'], format='%Y%m%d')  # Ensure date
plt.figure(figsize=(20, 6))
sns.lineplot(data=df_set1, x='date', y='sessions', marker='o')
plt.title('Trend of Sessions Over Time')
plt.show()

"""
Visualization for Set 2: Page Information and User Behavior
These visualizations aim to understand user behavior on the site, including the n
"""

# e. New vs. Returning Visitors
# This pie chart compares the ratio of new vs. returning visitors, providing insig
new_vs_returning_counts = df_set2['newvsreturning'].value_counts()
plt.figure(figsize=(5, 5))
new_vs_returning_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('New vs. Returning Visitors')
plt.ylabel('')  # Hide the y-label
plt.show()

# f. Browser Usage
# This chart illustrates the browser usage among visitors, indicating which brows
plt.figure(figsize=(10, 6))
sns.countplot(data=df_set2, y='browser', order=df_set2['browser'].value_counts().
plt.title('Browser Usage')
plt.xlabel('Count')
plt.ylabel('Browser')
plt.show()

"""
Visualization for Set 4: Language and Regional Preferences
This visualization focuses on device category distribution, providing insights in
"""

# g. Device Category Distribution
# This pie chart shows the distribution of sessions across different device categ
device_counts = df_set4['deviceCategory'].value_counts()
plt.figure(figsize=(4, 4))
device_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Device Category Distribution')
plt.ylabel('')  # Hide the y-label
plt.show()
```

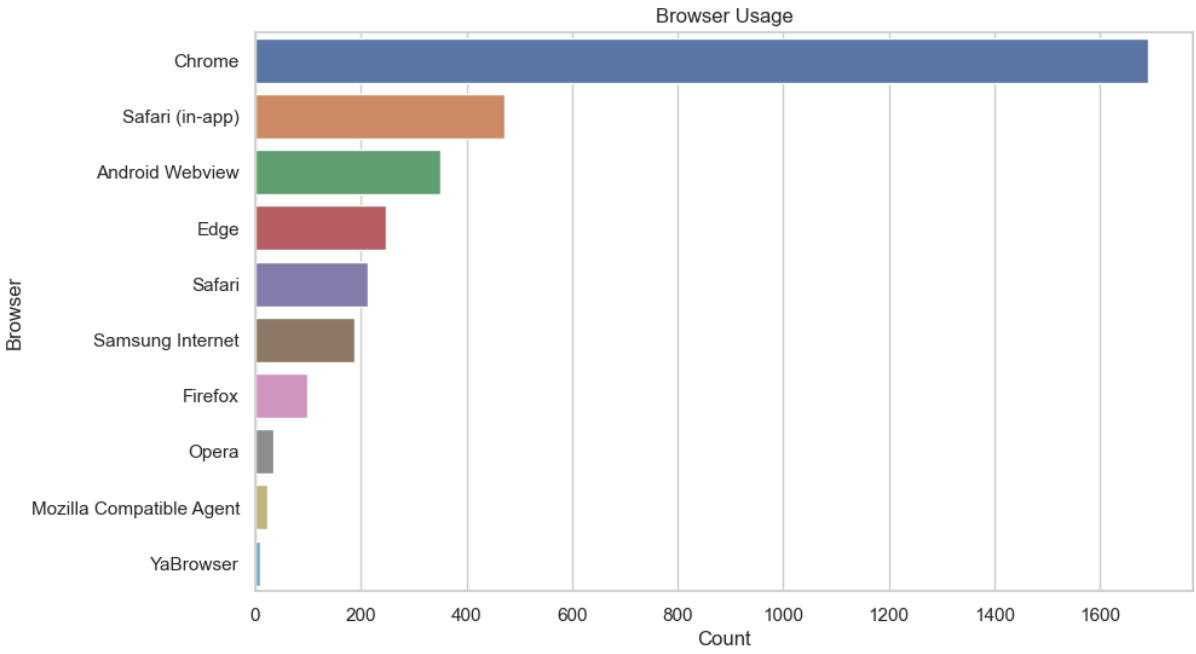

Distribution of First User Source
count
2500
2000
1500
1000
500
0
lnkd.in
(direct)
linkedin.com
github.com
evernote.com
m.facebook.com
google
akshayakalpa.kapturecrm.com
trello.com
(not set)
firstUserSource

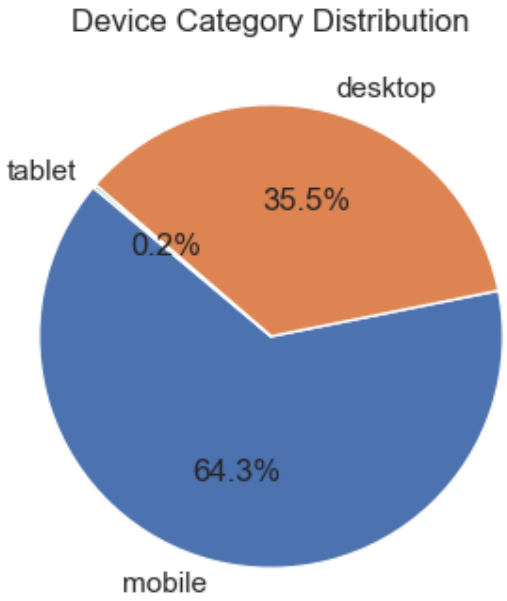

User Distribution by Country
count
1200
1000
800
600
400
200
0
country


Session Count per Device Category
3000
2500
2000
1500
1000
500
0
count
mobile
desktop
tablet
deviceCategory


Trend of Sessions Over Time
4
3
2
1
sessions
2023-04
2023-05
2023-06
2023-07
2023-08
2023-09
2023-10
2023-11
2023-12
2024-01
2024-02
date


New vs. Returning Visitors
returning
(not set)
20.0%
5.0%
0.1%
75.0%
new





SNOWFLAKE DASHBOARD