In [1]:

```
#import numpy, pandas library and data
import pandas as pd
import numpy as np
events =  pd.read_csv('c:\\users\\maagalu\\Desktop\\traffic.csv')
```

In [2]:

```
events
```

Out[2]:

	event	date	country	city	artist	album	track	isrc	
0	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	9 1c5
1	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	9 1c5
2	click	8/21/2021	India	Ludhiana	Reyanna Maria	So Pretty	So Pretty	USUM72100871	349
3	click	8/21/2021	France	Unknown	Simone & Simaria, Sebastian Yatra	No Llores Más	No Llores Más	BRUM72003904	4 08a
4	click	8/21/2021	Maldives	Malé	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	9 1c5
...	...	...	...	...	...	...	...	...	
226273	pageview	8/24/2021	Kuwait	Kuwait City	Sean Paul	The Trinity	Temperature	USAT20505520	3cdb
226274	pageview	8/24/2021	India	Chennai	Miscél	when you left	when you left	QM42K1907890	8 5d9
226275	pageview	8/24/2021	India	Jaipur	Trippie Redd, Lil Uzi Vert	Holy Smokes (feat. Lil Uzi Vert)	Holy Smokes	QZJ842001118	6 26
226276	pageview	8/24/2021	France	Unknown	Young Thug	Tick Tock	Tick Tock	USAT22104514	a2c

	event	date	country	city	artist	album	track	isrc	
226277	pageview	8/24/2021	Iraq	Duhok	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	9 1c5

226278 rows × 9 columns

In [3]:

```
#number of rows
len(events)
```

Out[3]:

```
226278
```

In [5]:

```
#number of columns
len(events.columns)
```

Out[5]:

```
9
```

In [6]:

```
#column headers
events.columns
```

Out[6]:

```
Index(['event', 'date', 'country', 'city', 'artist', 'album', 'track', 'isrc',
       'linkid'],
      dtype='object')
```

In [7]:

```
#top rows
events.head()
```

Out[7]:

	event	date	country	city	artist	album	track	isrc	linkid
0	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8
1	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8
2	click	8/21/2021	India	Ludhiana	Reyanna Maria	So Pretty	So Pretty	USUM72100871	23199824-9cf5-4b98-942a-34965c3b0cc2
3	click	8/21/2021	France	Unknown	Simone & Simaria, Sebastian Yatra	No Llores Más	No Llores Más	BRUM72003904	35573248-4e49-47c7-af80-08a960fa74cd

	event	date	country	city	artist	album	track	isrc	linkid
4	click	8/21/2021	Maldives	Malé	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8

In [8]:

```
#bottom rows
events.tail()
```

Out[8]:

	event	date	country	city	artist	album	track	isrc	
226273	pageview	8/24/2021	Kuwait	Kuwait City	Sean Paul	The Trinity	Temperature	USAT20505520	04b 105 3cdb0d
226274	pageview	8/24/2021	India	Chennai	Miscél	when you left	when you left	QM42K1907890	2fc 83aa 5d96c6
226275	pageview	8/24/2021	India	Jaipur	Trippie Redd, Lil Uzi Vert	Holy Smokes (feat. Lil Uzi Vert)	Holy Smokes	QZJ842001118	eec 6bd2 260c3
226276	pageview	8/24/2021	France	Unknown	Young Thug	Tick Tock	Tick Tock	USAT22104514	e0a 7cc a2c55c
226277	pageview	8/24/2021	Iraq	Duhok	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d8 97b6 1c5fb9

In [9]:

```
#datatype
events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 226278 entries, 0 to 226277
Data columns (total 9 columns):
 #   Column   Non-Null Count   Dtype 
---  ------   --------------   ----- 
 0   event    226278 non-null  object
 1   date     226278 non-null  object
 2   country  226267 non-null  object
 3   city     226267 non-null  object
 4   artist   226241 non-null  object
 5   album    226273 non-null  object
 6   track    226273 non-null  object
 7   isrc     219157 non-null  object
 8   linkid   226278 non-null  object
dtypes: object(9)
memory usage: 15.5+ MB
```

In [10]:

```python
#data summary
events.describe()
```

Out[10]:

	event	date	country	city	artist	album	track	isrc	linkid
count	226278	226278	226267	226267	226241	226273	226273	219157	226278
unique	3	7	211	11993	2419	3253	3562	709	3839
top	pageview	8/19/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8
freq	142015	35361	47334	22791	40841	40841	40841	40841	40841

Data Analysis - Method 1

```
(without manipulating the original data)
```

In [12]:

```python
#unique event
events.event.drop_duplicates()
```

Out[12]:

```
0            click
53605      preview
84043     pageview
Name: event, dtype: object
```

In [14]:

```python
# 1.a) How many total pageview events did the links receive in the full period

events.linkid[events.event=='pageview'].count()
```

Out[14]:

```
142015
```

In [18]:

```python
# 1.b) Total pageview event received per day

events.groupby('date')['event'].apply(lambda x: (x=='pageview').sum()).reset_index(name
```

Out[18]:

	date	pageviews
0	8/19/2021	22366
1	8/20/2021	21382
2	8/21/2021	21349
3	8/22/2021	20430
4	8/23/2021	18646
5	8/24/2021	18693
6	8/25/2021	19149

In [15]:

```
# 2.a) Other recorded events i.e click event for full period

events.linkid[events.event=='click'].count()
```

Out[15]:

```
55732
```

In [19]:

```
#clcik event per day
events.groupby('date')['event'].apply(lambda x: (x=='click').sum()).reset_index(name='c
```

Out[19]:

	date	click
0	8/19/2021	9207
1	8/20/2021	8508
2	8/21/2021	8071
3	8/22/2021	7854
4	8/23/2021	7315
5	8/24/2021	7301
6	8/25/2021	7476

In [16]:

```
# 2.b) Other recorded events i.e preview event for full period

events.linkid[events.event=='preview'].count()
```

Out[16]:

```
28531
```

In [20]:

```
#preview event per day
events.groupby('date')['event'].apply(lambda x: (x=='preview').sum()).reset_index(name=
```

Out[20]:

	date	preview
0	8/19/2021	3788
1	8/20/2021	4222
2	8/21/2021	4663
3	8/22/2021	4349
4	8/23/2021	3847
5	8/24/2021	3840
6	8/25/2021	3822

In [21]:

```
#overall events received
events.groupby('date')['event'].count()
```

Out[21]:

```
date
8/19/2021    35361
8/20/2021    34112
8/21/2021    34083
8/22/2021    32633
8/23/2021    29808
8/24/2021    29834
8/25/2021    30447
Name: event, dtype: int64
```

In [23]:

```
#each events received per day
events.groupby(['date','event'])['event'].count()
```

Out[23]:

```
date       event
8/19/2021  click        9207
           pageview    22366
           preview      3788
8/20/2021  click        8508
           pageview    21382
           preview      4222
8/21/2021  click        8071
           pageview    21349
           preview      4663
8/22/2021  click        7854
           pageview    20430
           preview      4349
8/23/2021  click        7315
           pageview    18646
           preview      3847
8/24/2021  click        7301
           pageview    18693
           preview      3840
8/25/2021  click        7476
           pageview    19149
           preview      3822
Name: event, dtype: int64
```

In [24]:

```
# 3) Which countries did the pageviews come from

events.loc[events['event'] == 'pageview','country'].drop_duplicates()
```

Out[24]:

```
84043                 Saudi Arabia
84044                United States
84046                      Ireland
84047               United Kingdom
84051                       France
                    ...
165434                 Afghanistan
176541    Central African Republic
200553                    Guernsey
216014                Sint Maarten
223904                Saint Martin
Name: country, Length: 212, dtype: object
```

In [25]:

```
# 4) Overall click rate (clicks/pageviews)

clickrate=events.linkid[events.event=='click'].count()/events.linkid[events.event=='pag
```

In [26]:

```
clickrate
```

Out[26]:

```
0.3924374185825441
```

In [27]:

```
# 5) how does the clickrate distributed across the link

#taking out sum of pageview and click event separately
pageviews=events.groupby('linkid')['event'].apply(lambda x: (x=='pageview').sum()).rese
```

In [32]:

```
clicks=events.groupby('linkid')['event'].apply(lambda x: (x=='click').sum()).reset_inde
```

In [37]:

```
#merging seoarated pageview and click event
pc= pd.merge(pageviews,clicks, on='linkid')
```

In [41]:

```
#sorting in descending order
Sorted_df = pc.sort_values("pageviews", ascending=False)
```

In [42]:

```
#then adding new column clickrate
sorted_df['clickrate']=sorted_df['clicks']/sorted_df['pageviews']
```

In [43]:

```
sorted_df
```

Out[43]:

	linkid	pageviews	clicks	clickrate
709	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	25175	9692	0.384985
1250	522da5cc-8177-4140-97a7-a84fdb4caf1c	6600	2109	0.319545
3477	e849515b-929d-44c8-a505-e7622f1827e9	5981	2198	0.367497
2951	c2c876ab-b093-4750-9449-6b4913da6af3	4303	1429	0.332094
537	23199824-9cf5-4b98-942a-34965c3b0cc2	3532	1187	0.336070
...	...	...	...	...
2159	8c71ba08-d449-521e-8092-5d4f7e14d759	1	0	0.000000
2160	8c7849a7-cb1f-5482-ae81-043546086f2e	1	0	0.000000
653	2a20c79c-7578-5247-878b-a6b71fba3769	1	1	1.000000
1280	54166799-1895-4f35-9b2f-b249c2f7a351	0	1	inf
2669	aee2b83d-5f50-4309-9e62-200c404d4751	0	1	inf

3839 rows × 4 columns

In [45]:

```
#removed infinite values
df = sorted_df.replace([np.inf, -np.inf], np.nan).dropna(axis=0)
```

In [46]:

```
# clickrate across the links
df
```

Out[46]:

	linkid	pageviews	clicks	clickrate
709	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	25175	9692	0.384985
1250	522da5cc-8177-4140-97a7-a84fdb4caf1c	6600	2109	0.319545
3477	e849515b-929d-44c8-a505-e7622f1827e9	5981	2198	0.367497
2951	c2c876ab-b093-4750-9449-6b4913da6af3	4303	1429	0.332094
537	23199824-9cf5-4b98-942a-34965c3b0cc2	3532	1187	0.336070
...	...	...	...	...
836	3653d3aa-474e-59a5-ac34-28a7df269a01	1	1	1.000000
2158	8c646478-fdc9-5410-89cd-15385794cf84	1	1	1.000000
2159	8c71ba08-d449-521e-8092-5d4f7e14d759	1	0	0.000000
2160	8c7849a7-cb1f-5482-ae81-043546086f2e	1	0	0.000000
653	2a20c79c-7578-5247-878b-a6b71fba3769	1	1	1.000000

3837 rows × 4 columns

In [59]:

```
# 6.a) Correlation between clicks and previews

import scipy as sp
import scipy.stats
```

In [60]:

```
click=events.groupby('linkid')['event'].apply(lambda x: (x=='click').sum()).reset_index
```

In [61]:

```
preview=events.groupby('linkid')['event'].apply(lambda x: (x=='preview').sum()).reset_i
```

In [62]:

```
df1 = pd.merge(click,preview, on='linkid')
```

In [63]:

```
#correlation between clicks and previews
df1.corr()
```

Out[63]:

	click	preview
click	1.000000	0.988659
preview	0.988659	1.000000

--

In [64]:

```
# 6.b) Significance and effect
```

```
from scipy.stats import pearsonr
```

In [65]:

```
#correlation coefficient and P-value
pearsonr(df1['click'], df1['preview'])
```

Out[65]:

```
(0.9886586274883709, 0.0)
```

--

In [68]:

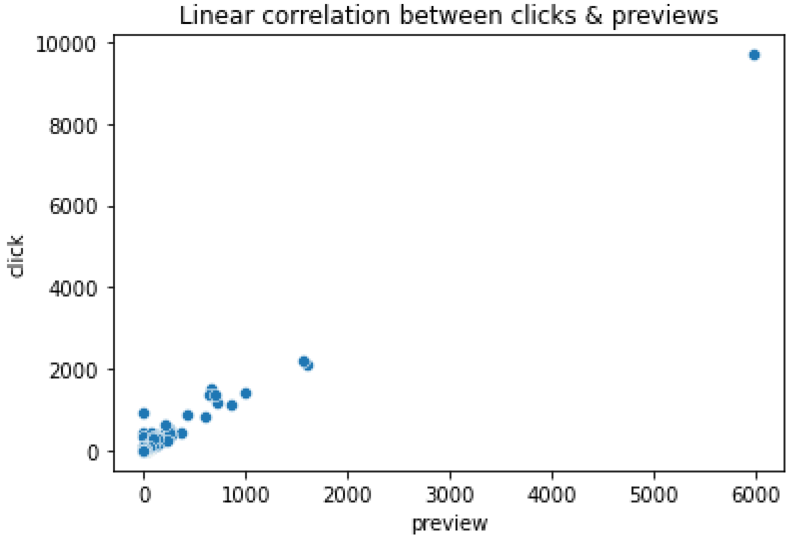
```
# 6.c) Linear and categorical relationships between both variables

import matplotlib
import matplotlib.pyplot as pp

import pandas.plotting

from IPython import display

%matplotlib inline
```

In [69]:

```
import seaborn as sns
pc=sns.scatterplot(x="preview", y="click", data=df1);
pc.set_title("Linear correlation between clicks & previews")
```

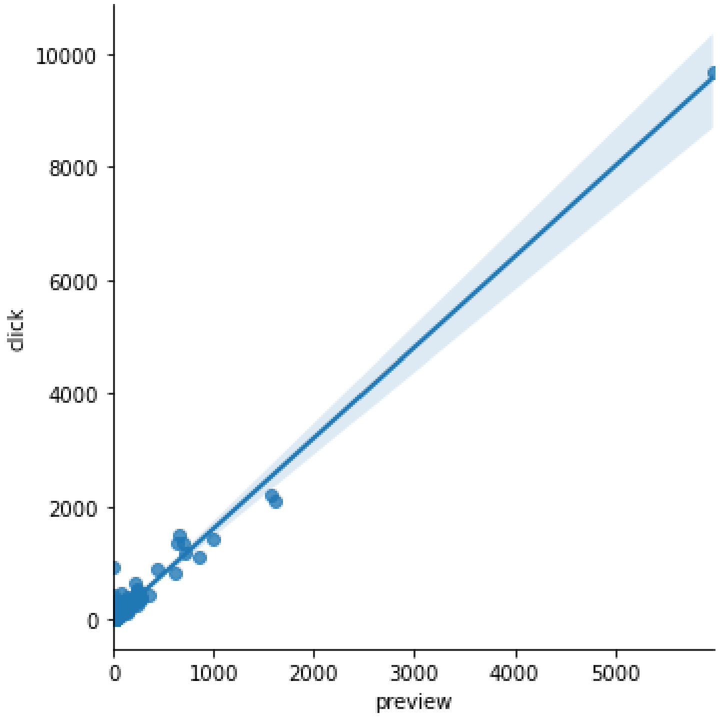
Out[69]:

```
Text(0.5, 1.0, 'Linear correlation between clicks & previews')
```



In [70]:

```
sns.lmplot(x="preview", y="click", data=df1);
```

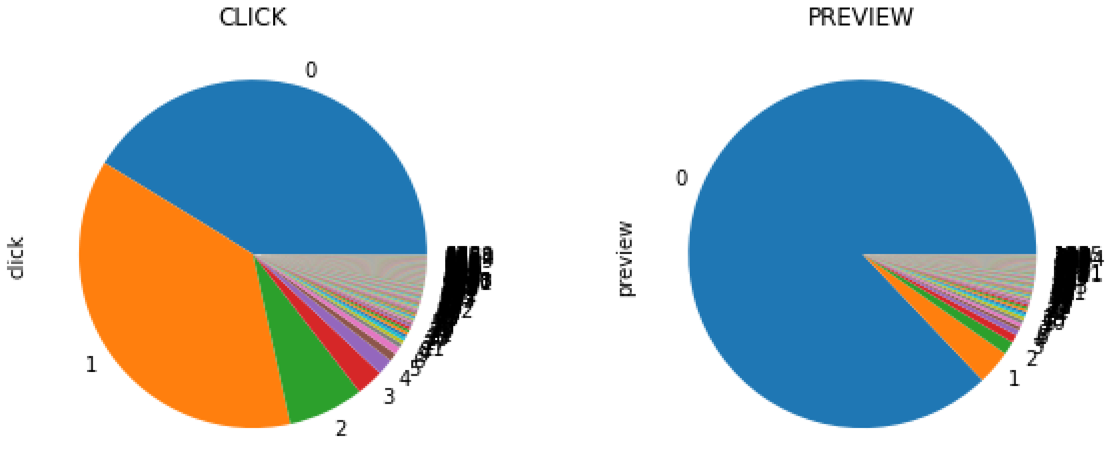


In [71]:

```
pp.figure(figsize=(10,4))

pp.subplot(1,2,1);df1.click.value_counts().plot(kind='pie');pp.title('CLICK')
pp.subplot(1,2,2);df1.preview.value_counts().plot(kind='pie');pp.title('PREVIEW')
```

Out[71]:

```
Text(0.5, 1.0, 'PREVIEW')
```



Data Analysis - Method 2

```
                     (by manipulating the original data and taking sample
    data for analysis )
```

In [72]:

```
events.head()
```

Out[72]:

	event	date	country	city	artist	album	track	isrc	linkid
0	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8
1	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8
2	click	8/21/2021	India	Ludhiana	Reyanna Maria	So Pretty	So Pretty	USUM72100871	23199824-9cf5-4b98-942a-34965c3b0cc2
3	click	8/21/2021	France	Unknown	Simone & Simaria, Sebastian Yatra	No Llores Más	No Llores Más	BRUM72003904	35573248-4e49-47c7-af80-08a960fa74cd
4	click	8/21/2021	Maldives	Malé	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8

In [86]:

```
#adding new columns by assigning values to events dataset
events['pageviews']=0
events['click']=0
events['preview']=0
```

In [87]:

```
#replacing 0 by 1 for corresponding events
events.loc[events.event == "pageview", "pageviews"] = 1
events.loc[events.event == "click", "click"] = 1
events.loc[events.event == "preview", "preview"] = 1
```

In [89]:

```
events.head()
events.drop('pageview', axis=1, inplace=True)
```

In [92]:

```
events.head()
```

Out[92]:

	event	date	country	city	artist	album	track	isrc	linkid	click
0	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	1
1	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	1

	event	date	country	city	artist	album	track	isrc	linkid	click
2	click	8/21/2021	India	Ludhiana	Reyanna Maria	So Pretty	So Pretty	USUM72100871	23199824-9cf5-4b98-942a-34965c3b0cc2	1
3	click	8/21/2021	France	Unknown	Simone & Simaria, Sebastian Yatra	No Llores Más	No Llores Más	BRUM72003904	35573248-4e49-47c7-af80-08a960fa74cd	1
4	click	8/21/2021	Maldives	Malé	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	1

In [94]:

```
events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 226278 entries, 0 to 226277
Data columns (total 12 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   event      226278 non-null  object
 1   date       226278 non-null  object
 2   country    226267 non-null  object
 3   city       226267 non-null  object
 4   artist     226241 non-null  object
 5   album      226273 non-null  object
 6   track      226273 non-null  object
 7   isrc       219157 non-null  object
 8   linkid     226278 non-null  object
 9   click      226278 non-null  int64 
 10  preview    226278 non-null  int64 
 11  pageviews  226278 non-null  int64 
dtypes: int64(3), object(9)
memory usage: 20.7+ MB
```

--

In [95]:

```
# 1.a) How many total pageview events did the links receive in the full period

events.pageviews.sum()
```

Out[95]:

```
142015
```

In [96]:

```
# 1.b) Total pageview event received per day

events.groupby('date')['pageviews'].sum()
```

Out[96]:

```
date
8/19/2021    22366
8/20/2021    21382
8/21/2021    21349
8/22/2021    20430
```

```
8/23/2021    18646
8/24/2021    18693
8/25/2021    19149
Name: pageviews, dtype: int64
```

In [97]:

```
# 2.a) Other recorded events i.e click event for full period

events.click.sum()
```

Out[97]:

```
55732
```

In [100…

```
#clcik event per day
events.groupby('date')['click'].sum()
```

Out[100…

```
date
8/19/2021    9207
8/20/2021    8508
8/21/2021    8071
8/22/2021    7854
8/23/2021    7315
8/24/2021    7301
8/25/2021    7476
Name: click, dtype: int64
```

In [101…

```
# 2.b) Other recorded events i.e preview event for full period
events.preview.sum()
```

Out[101…

```
28531
```

In [102…

```
#preview event per day
events.groupby('date')['preview'].sum()
```

Out[102…

```
date
8/19/2021    3788
8/20/2021    4222
8/21/2021    4663
8/22/2021    4349
8/23/2021    3847
8/24/2021    3840
8/25/2021    3822
Name: preview, dtype: int64
```

In [103…

```
# 3) Which countries did the pageviews come from

events.loc[events['pageviews'] == 1 ,'country'].drop_duplicates()
```

Out[103…

```
84043                 Saudi Arabia
84044                United States
84046                      Ireland
84047               United Kingdom
84051                       France
                         ...
165434                 Afghanistan
176541    Central African Republic
```

```
200553                      Guernsey
216014                  Sint Maarten
223904                  Saint Martin
Name: country, Length: 212, dtype: object
```

In [104…

```
# 4) Overall click rate (clicks/pageviews)

cr=events.click.sum()/events.pageviews.sum()
```

In [105…

```
cr
```

Out[105…

```
0.3924374185825441
```

In [106…

```
# 5) how does the clickrate distributed across the link

q=events.groupby('linkid')[['pageviews','click','preview']].sum()
```

In [107…

```
sorted_df2 = q.sort_values("pageviews", ascending=False)
```

In [108…

```
sorted_df2['clickrate']=sorted_df2['click']/sorted_df2['pageviews']
print(sorted_df2)
```

```
                                      pageviews  click  preview  clickrate
linkid
2d896d31-97b6-4869-967b-1c5fb9cd4bb8      25175   9692     5974   0.384985
522da5cc-8177-4140-97a7-a84fdb4caf1c       6600   2109     1605   0.319545
e849515b-929d-44c8-a505-e7622f1827e9       5981   2198     1571   0.367497
c2c876ab-b093-4750-9449-6b4913da6af3       4303   1429     1001   0.332094
23199824-9cf5-4b98-942a-34965c3b0cc2       3532   1187      718   0.336070
...                                         ...    ...      ...        ...
8c71ba08-d449-521e-8092-5d4f7e14d759          1      0        0   0.000000
8c7849a7-cb1f-5482-ae81-043546086f2e          1      0        0   0.000000
2a20c79c-7578-5247-878b-a6b71fba3769          1      1        0   1.000000
54166799-1895-4f35-9b2f-b249c2f7a351          0      1        0        inf
aee2b83d-5f50-4309-9e62-200c404d4751          0      1        0        inf

[3839 rows x 4 columns]
```

In [109…

```
#removed infinite values
df2 = sorted_df2.replace([np.inf, -np.inf], np.nan).dropna(axis=0)
```

In [110…

```
# Clickrate across the links
df2
```

Out[110…

	pageviews	click	preview	clickrate
linkid				
2d896d31-97b6-4869-967b-1c5fb9cd4bb8	25175	9692	5974	0.384985
522da5cc-8177-4140-97a7-a84fdb4caf1c	6600	2109	1605	0.319545

linkid	pageviews	click	preview	clickrate
e849515b-929d-44c8-a505-e7622f1827e9	5981	2198	1571	0.367497
c2c876ab-b093-4750-9449-6b4913da6af3	4303	1429	1001	0.332094
23199824-9cf5-4b98-942a-34965c3b0cc2	3532	1187	718	0.336070
...	...	...	...	...
3653d3aa-474e-59a5-ac34-28a7df269a01	1	1	0	1.000000
8c646478-fdc9-5410-89cd-15385794cf84	1	1	0	1.000000
8c71ba08-d449-521e-8092-5d4f7e14d759	1	0	0	0.000000
8c7849a7-cb1f-5482-ae81-043546086f2e	1	0	0	0.000000
2a20c79c-7578-5247-878b-a6b71fba3769	1	1	0	1.000000

3837 rows × 4 columns

In [124…

```
# 6.a) Correlation


df2.corr()
```

Out[124…

	pageviews	click	preview	clickrate
pageviews	1.000000	0.994001	0.996691	-0.004248
click	0.994001	1.000000	0.988659	0.076821
preview	0.996691	0.988659	1.000000	-0.004378
clickrate	-0.004248	0.076821	-0.004378	1.000000

In [111…

```
# 6.b) Significance and effects
#calculating overall mean and sample mean for data analysis

df2.click.mean()
```

Out[111…

```
14.524367995830076
```

In [112…

```
df2.preview.mean()
```

Out[112…

```
7.435757101902528
```

In [115…

```
#lets take click sample

sample_size=40
click_sample=np.random.choice(df2.click,sample_size)
```

In [116…

```
click_sample
```

Out[116…

```
array([  1,   0,   0,   0,   1,   0,   1, 117,  69,   0,   1,   1,   1,
         0,   3,   1,   1,   1,   6,   1,   1,   1,   1,   0,   1,   3,
       434,   1,   1,  11,   5,   0,   0,   0,   0,   0,   0,   1,   0,
         0], dtype=int64)
```

In [117…

```
#lets take preview sample
sample_size=40
preview_sample=np.random.choice(df2.preview,sample_size)
```

In [118…

```
preview_sample
```

Out[118…

```
array([  0,   0,   5,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
       150,   0,   0,   1,   0,   0,   0,   1,   0,   0,   3, 117,   0,
       612], dtype=int64)
```

In [119…

```
from scipy.stats import ttest_1samp
```

In [120…

```
ttest,p_value=ttest_1samp(click_sample,15)
```

In [121…

```
print(p_value)
```

```
0.8854692372744519
```

In [122…

```
ttest,p_value=ttest_1samp(preview_sample,7)
```

In [123…

```
##found out p-value is greater than 0.05 and null hypothesis is true(there is no differ
# where t-test is used to determine if there is a significant difference between the me
print(p_value)
```

```
0.3421412918387813
```

In [4]:

```
import pandas as pd
import numpy as np
```

In [6]:

```
events = pd.read_csv('c:\\users\\maagalu\\Desktop\\traffic.csv')
```

In [7]:

```
#adding binary values for the respective event
events['pageviews']=0
events['click']=0
events['preview']=0
```

In [8]:

```
#replacing 0 by 1 for corresponding events
events.loc[events.event == "pageview", "pageviews"] = 1
events.loc[events.event == "click", "click"] = 1
events.loc[events.event == "preview", "preview"] = 1
```

In [9]:

```
events.head()
```

Out[9]:

	event	date	country	city	artist	album	track	isrc	linkid	page
0	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	
1	click	8/21/2021	Saudi Arabia	Jeddah	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	
2	click	8/21/2021	India	Ludhiana	Reyanna Maria	So Pretty	So Pretty	USUM72100871	23199824-9cf5-4b98-942a-34965c3b0cc2	
3	click	8/21/2021	France	Unknown	Simone & Simaria, Sebastian Yatra	No Llores Más	No Llores Más	BRUM72003904	35573248-4e49-47c7-af80-08a960fa74cd	
4	click	8/21/2021	Maldives	Malé	Tesher	Jalebi Baby	Jalebi Baby	QZNWQ2070741	2d896d31-97b6-4869-967b-1c5fb9cd4bb8	

In [10]:

```
x=events.groupby('linkid')[['pageviews','click','preview']].sum()
```

In [12]:

```
df = x.sort_values("pageviews", ascending=False)
```

In [13]:

```
df
```

Out[13]:

	pageviews	click	preview
linkid			
2d896d31-97b6-4869-967b-1c5fb9cd4bb8	25175	9692	5974
522da5cc-8177-4140-97a7-a84fdb4caf1c	6600	2109	1605
e849515b-929d-44c8-a505-e7622f1827e9	5981	2198	1571
c2c876ab-b093-4750-9449-6b4913da6af3	4303	1429	1001
23199824-9cf5-4b98-942a-34965c3b0cc2	3532	1187	718
...	...	...	...

	pageviews	click	preview
linkid			
8c71ba08-d449-521e-8092-5d4f7e14d759	1	0	0
8c7849a7-cb1f-5482-ae81-043546086f2e	1	0	0
2a20c79c-7578-5247-878b-a6b71fba3769	1	1	0
54166799-1895-4f35-9b2f-b249c2f7a351	0	1	0
aee2b83d-5f50-4309-9e62-200c404d4751	0	1	0

3839 rows × 3 columns

In [14]:

```
df.corr()
```

Out[14]:

	pageviews	click	preview
pageviews	1.000000	0.994001	0.996691
click	0.994001	1.000000	0.988659
preview	0.996691	0.988659	1.000000

In [17]:

```
del df['pageviews']
```

In [18]:

```
df
```

Out[18]:

	click	preview
linkid		
2d896d31-97b6-4869-967b-1c5fb9cd4bb8	9692	5974
522da5cc-8177-4140-97a7-a84fdb4caf1c	2109	1605
e849515b-929d-44c8-a505-e7622f1827e9	2198	1571
c2c876ab-b093-4750-9449-6b4913da6af3	1429	1001
23199824-9cf5-4b98-942a-34965c3b0cc2	1187	718
...	...	...
8c71ba08-d449-521e-8092-5d4f7e14d759	0	0
8c7849a7-cb1f-5482-ae81-043546086f2e	0	0
2a20c79c-7578-5247-878b-a6b71fba3769	1	0
54166799-1895-4f35-9b2f-b249c2f7a351	1	0
aee2b83d-5f50-4309-9e62-200c404d4751	1	0

3839 rows × 2 columns

In [19]:

```
df.corr()
```

Out[19]:

	click	preview
click	1.000000	0.988659
preview	0.988659	1.000000

In [24]:

```
stats.pearsonr(df['preview'], df['click'])
```

Out[24]:

```
(0.9886586274883703, 0.0)
```

In [20]:

```
import scipy as sp
import scipy.stats
from scipy import stats
```

In [21]:

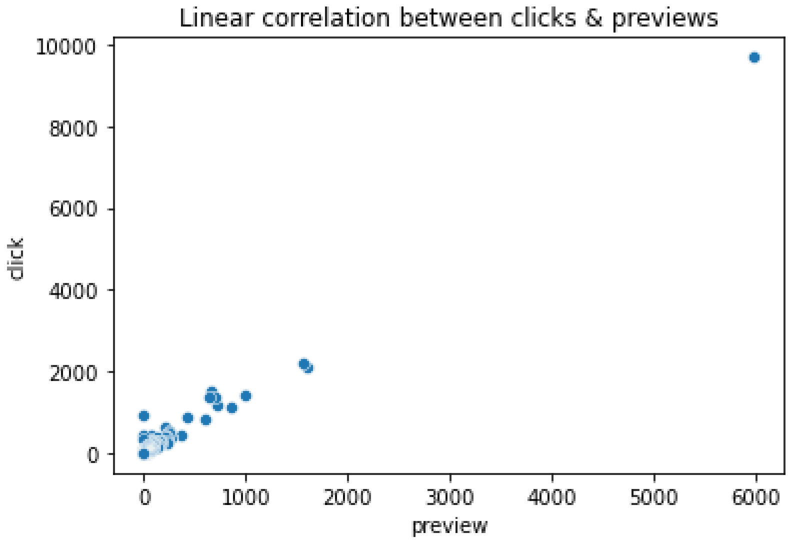
```
import matplotlib
import matplotlib.pyplot as pp

import pandas.plotting

from IPython import display

%matplotlib inline
```

In [22]:

```
import seaborn as sns
pc=sns.scatterplot(x="preview", y="click", data=df);
pc.set_title("Linear correlation between clicks & previews")
```

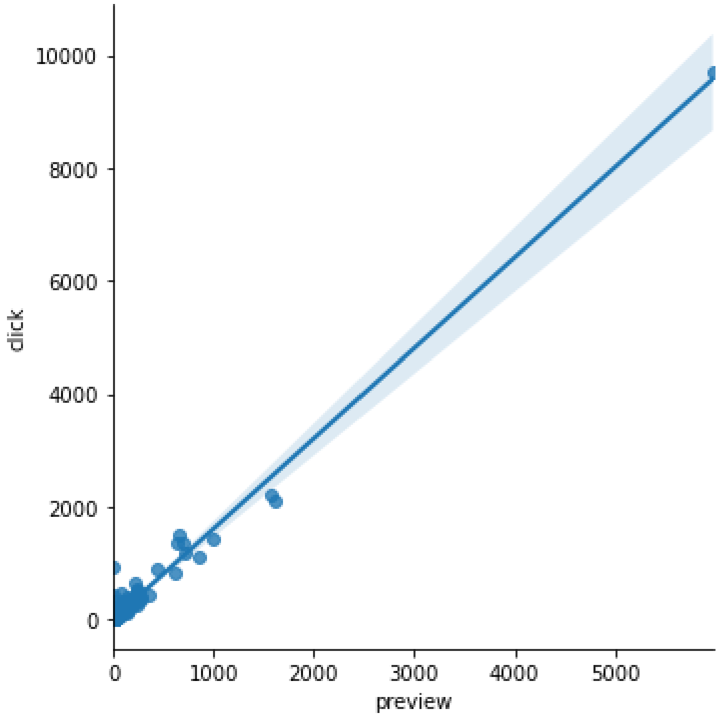
Out[22]:

```
Text(0.5, 1.0, 'Linear correlation between clicks & previews')
```



In [23]:

```
sns.lmplot(x="preview", y="click", data=df);
```



In [30]:

```
from scipy.stats import ttest_rel
```

In [34]:

```
_,p_value=stats.ttest_rel(a=df.preview,b=df.click)
```

In [35]:

```
print(p_value)
```

```
8.719233780308909e-10
```

In [36]:

```
if p_value < 0.05: #considering alpha value is 0.05 or 5%
    print("we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
we are rejecting null hypothesis
```

In [37]:

```
#Probability distribution
df
```

Out[37]:

	click	preview
linkid		
2d896d31-97b6-4869-967b-1c5fb9cd4bb8	9692	5974
522da5cc-8177-4140-97a7-a84fdb4caf1c	2109	1605
e849515b-929d-44c8-a505-e7622f1827e9	2198	1571
c2c876ab-b093-4750-9449-6b4913da6af3	1429	1001
23199824-9cf5-4b98-942a-34965c3b0cc2	1187	718

	click	preview
linkid		
...	...	...
8c71ba08-d449-521e-8092-5d4f7e14d759	0	0
8c7849a7-cb1f-5482-ae81-043546086f2e	0	0
2a20c79c-7578-5247-878b-a6b71fba3769	1	0
54166799-1895-4f35-9b2f-b249c2f7a351	1	0
aee2b83d-5f50-4309-9e62-200c404d4751	1	0

3839 rows × 2 columns

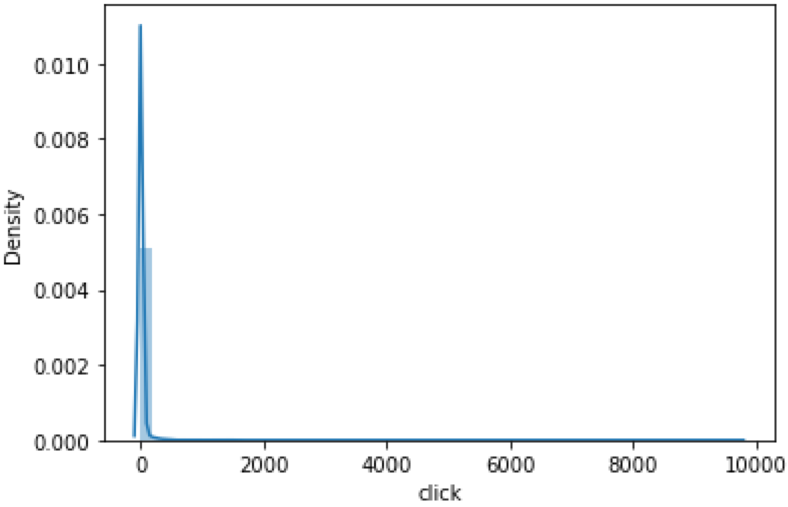
In [39]:

```
sns.distplot(df['click'])
```

```
C:\Users\maagalu\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please a
dapt your code to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[39]:

```
<AxesSubplot:xlabel='click', ylabel='Density'>
```



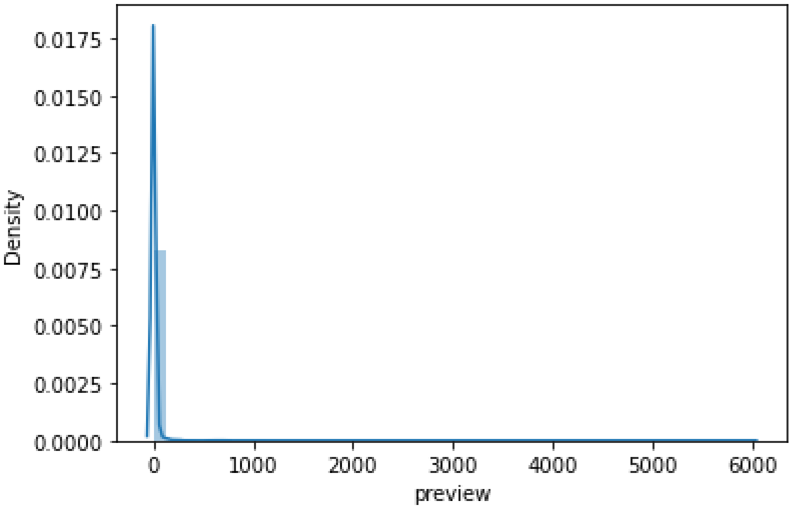
In [40]:

```
sns.distplot(df['preview'])
```

```
C:\Users\maagalu\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please a
dapt your code to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[40]:

```
<AxesSubplot:xlabel='preview', ylabel='Density'>
```



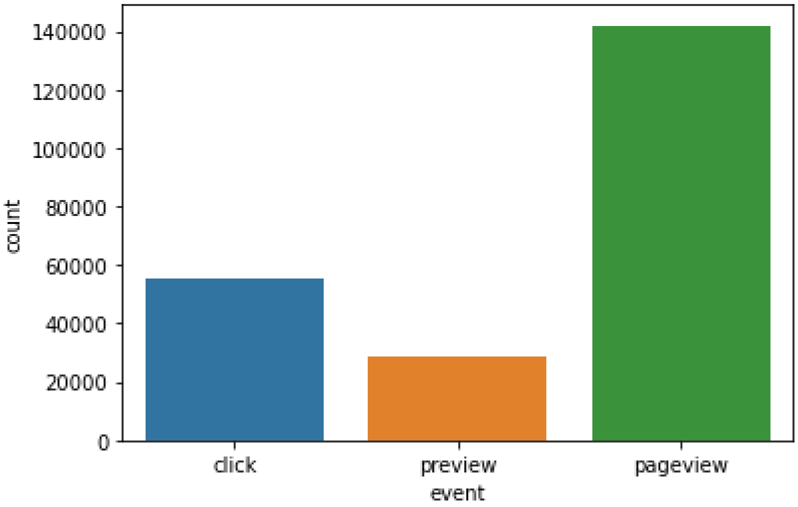
In [47]:

```
sns.countplot(events['event'])
```

```
C:\Users\maagalu\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variable as a keyword arg: x. From version 0.12, the only valid positi
onal argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
```

Out[47]:

```
<AxesSubplot:xlabel='event', ylabel='count'>
```



In [ ]: