In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```python
posts = pd.read_json("C:\\Users\\vez\\OneDrive\\Desktop\\Tumblr_Test\\posts.jsonl\\bu-posts.json", lines=True)
```

In [7]:
```python
posts.head()
```

Out[7]:

	comment_count	content	author	title	like_count	author_login	blog_id	date_gmt	author_id	post_id	lang	url	liker_ids	commenter_ids
0	0	The Snap! Jamaal Jackson tore knee ligaments i...	jasontromm	Biggest Challenge for the Eagles This Week?	0	jasontromm	753	2009-12-31 16:27:39	762	969	en	jasontromm.wordpress.com/2009/12/31/biggest-ch...	NaN	NaN
1	0	\r\n\r\n\r\n\r\nPenguins!\r\n\r\n\r\n\r\n	jasontromm	My Newest Cube Toy	0	jasontromm	753	2010-01-06 14:48:55	762	970	en	jasontromm.wordpress.com/2010/01/06/my-newest-...	NaN	NaN
2	0	\r\n\r\n\r\nDonovan McNabb's performance Satur...	jasontromm	Time to put McNabb on the Bench	0	jasontromm	753	2010-01-11 16:48:34	762	971	en	jasontromm.wordpress.com/2010/01/11/time-to-pu...	NaN	NaN
3	0	\r\nIf you think the Republicans are against b...	jasontromm	Republican Hypocrites in the Senate	0	jasontromm	753	2010-01-20 17:07:21	762	977	en	jasontromm.wordpress.com/2010/01/20/republican...	NaN	NaN
4	0	\r\nIf I was a Democrat, I wouldn't want Barac...	jasontromm	Obama's Campaign Failures	0	jasontromm	753	2010-01-20 19:42:44	762	978	en	jasontromm.wordpress.com/2010/01/20/obamas-cam...	NaN	NaN

In [9]:
```python
posts.describe()
```

Out[9]:

	comment_count	like_count	blog_id	author_id	post_id
count	1.809199e+06	1.809199e+06	1.809199e+06	1.809199e+06	1.809199e+06
mean	3.451310e+00	4.238735e+00	4.016731e+07	3.169361e+07	2.033201e+06
std	7.094134e+01	1.079453e+01	2.467020e+07	2.404092e+07	4.526585e+07
min	0.000000e+00	0.000000e+00	7.530000e+02	0.000000e+00	1.000000e+00
25%	0.000000e+00	0.000000e+00	1.707398e+07	1.073806e+07	5.730000e+02
50%	0.000000e+00	0.000000e+00	3.940051e+07	2.690976e+07	2.078000e+03
75%	3.000000e+00	4.000000e+00	6.226436e+07	5.415462e+07	7.215000e+03
max	9.018200e+04	1.063000e+03	9.093847e+07	8.742489e+07	1.110245e+09

In [41]:
```python
#out of bound dates
posts[posts['post_date'].isna()]
```

Out[41]:

	comment_count	content	author	title	like_count	author_login	blog_id	date_gmt	author_id	post_id	lang	url	liker_ids	co
206045	0	Welcome to NELTA CHOUTARI blogzine, which star...	SamSarma	About	1	sharmashyam	8393722	0209-07-10 17:28:44	7239500	2	en	neltachoutari.wordpress.com/?page_id=2	[43264509]	
306297	0	MUSEO KARURA ART CENTRE\r\n(MKAC)\r\nHISTÓRICO...	inakarura	HISTORICO GENERAL, POR AÑOS, DE EXPOSICIONES E...	0	inakarura	12115919	0201-12-30 10:45:55	13206429	6364	es	mkac.wordpress.com/0201/12/30/6364	NaN	
722087	15	It's my wedding anniversary. \r\n\r\nIt's com...	Stephanie	"Every Love Story Is a Ghost Story"	10	stephaniemartinglennon	31414728	0201-07-16 11:16:59	30840407	2287	en	stephaniemartinglennon.com/0201/07/16/every-lo...	[11761767, 19051174, 20185121, 23781177, 24196...	

In [11]:
```python
posts['post_date']=pd.to_datetime(posts.date_gmt, errors='coerce').dt.date #3 out of bound dates updated to NAT
```

In [17]:
```python
posts['post_year']=pd.to_datetime(posts['post_date']).dt.year
```

In [ ]:
```python
#posts['month']=pd.to_datetime(posts['post_date']).dt.month
```

In [175…
```python
posts['post_date'].value_counts()
```

Out[175]:
```
2015-05-14    5018
2015-05-17    4491
2015-01-05    4070
2015-05-13    3945
2015-05-18    3452
              ...
2004-01-25       1
2004-01-17       1
2004-01-11       1
2003-12-30       1
1995-10-02       1
Name: post_date, Length: 4619, dtype: int64
```

In [29]:
```python
posts['post_year'].value_counts()
```

Out[29]:
```
2014.0    619471
2015.0    344670
2013.0    337899
2012.0    214773
2011.0    128292
2010.0     69125
2009.0     40110
2008.0     25507
2007.0     13790
2006.0      9041
2005.0      4422
2004.0       949
2003.0       555
2002.0       398
2001.0        66
2000.0        38
1970.0        30
1990.0        25
1969.0         9
1999.0         6
1997.0         4
1994.0         4
1915.0         2
1977.0         2
1975.0         1
1996.0         1
1978.0         1
1985.0         1
1914.0         1
1965.0         1
1989.0         1
1995.0         1
Name: post_year, dtype: int64
```

In [178…
```python
posts.sort_values(by=['date_gmt'],inplace=True)
```

In [180…
```python
#median of full dataset
posts.like_count.median()
```

Out[180]:
```
0.0
```

In [181…
```python
#median over last 5 years
posts.loc[posts.post_year >= 2010].like_count.median()
```

Out[181]:
```
0.0
```

In [190…
```python
posts.loc[posts.post_year >= 2010].like_count.mean()
```

Out[190]:
```
4.469142997147407
```

In [191…
```python
#mean of full dataset
posts.like_count.mean()
```

Out[191]:
```
4.238734931867639
```

In [40]:
```python
#median and mean per post
posts.groupby(['post_id'])['like_count'].agg(median_per_post='median', mean_per_post = 'mean').reset_index()
```

Out[40]:

	post_id	median_per_post	mean_per_post
0	1	0.0	4.489196
1	2	1.0	6.572674
2	3	0.0	1.509804
3	4	0.0	1.252487
4	5	0.0	1.250687
...	...	...	...
121022	1110244488	26.0	26.000000
121023	1110244493	45.0	45.000000
121024	1110244507	20.0	20.000000
121025	1110244512	42.0	42.000000
121026	1110244524	38.0	38.000000

121027 rows × 3 columns

In [145…
```python
#daily likes and comments
#
#visualize data over the period September 7,2012 to May 28, 2015
post_daily_like_comment_count = posts.loc[posts.post_date>= pd.to_datetime('09/07/2012')].groupby("post_date").agg(
total_daily_like_count = pd.NamedAgg(column='like_count', aggfunc="sum"),
total_daily_comment_count = pd.NamedAgg(column='comment_count', aggfunc="sum")

).reset_index()
```

```
C:\Users\vez\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\ops\array_ops.py:73: FutureWarning: Comparison of Timestamp with datetime.date is deprecated in o
rder to match the standard library behavior. In a future version these will be considered non-comparable. Use 'ts == pd.Timestamp(date)' or 'ts.date() == date' instead.
  result = libops.scalar_compare(x.ravel(), y, op)
```

In [137…
```python
post_daily_like_comment_count
```

Out[137]:

	post_date	total_daily_like_count	total_daily_comment_count
0	2012-09-07	2300	2083
1	2012-09-08	2278	2040
2	2012-09-09	2506	2158
3	2012-09-10	2500	2515
4	2012-09-11	2665	2452
...	...	...	...
989	2015-05-24	9402	3719
990	2015-05-25	10169	4152
991	2015-05-26	9387	3887
992	2015-05-27	8111	2654
993	2015-05-28	622	192

994 rows × 3 columns

In [138…
```python
post_daily_like_comment_count.describe()
```

Out[138]:

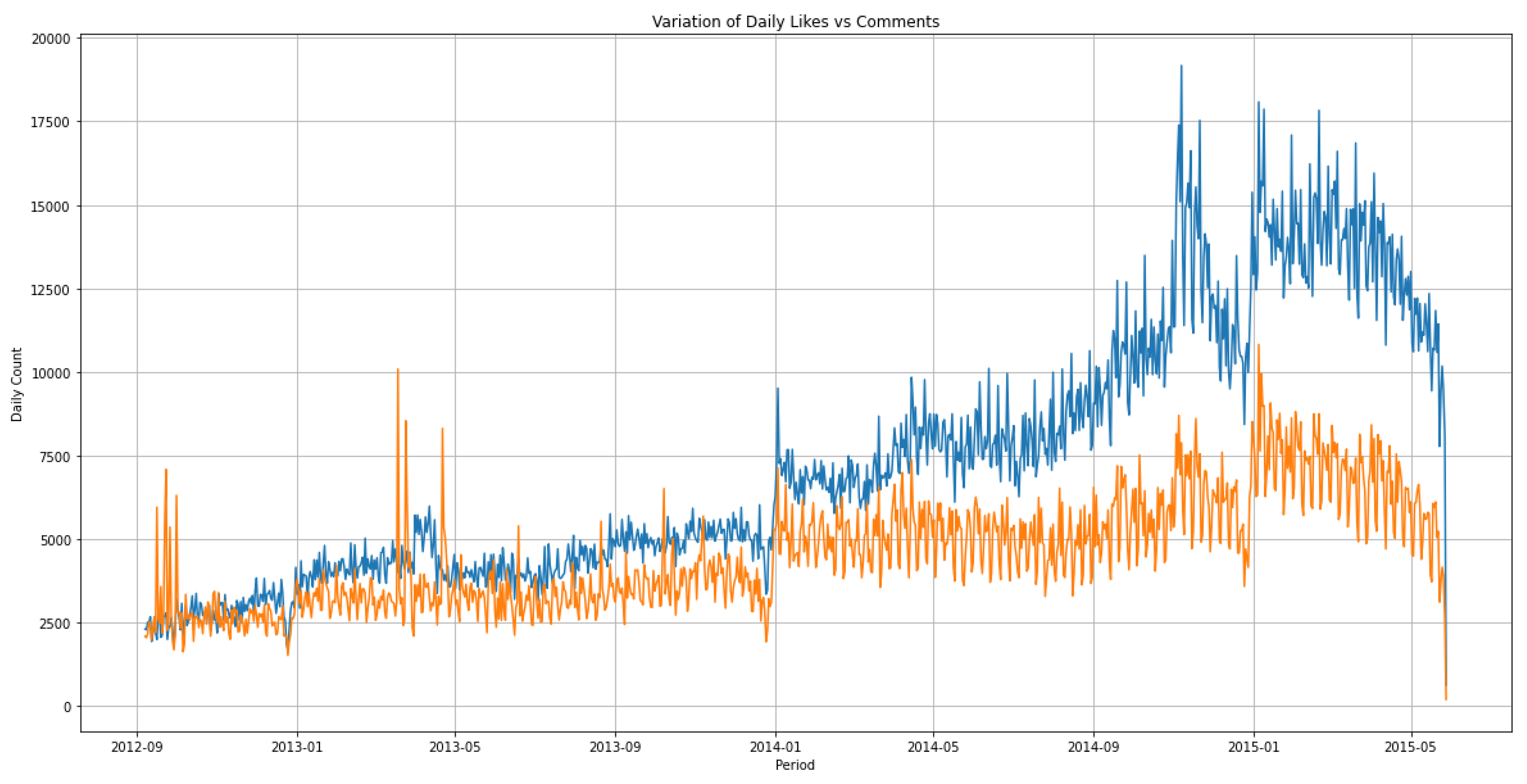
	total_daily_like_count	total_daily_comment_count
count	994.000000	994.000000
mean	7225.926559	4527.734406
std	3796.686107	1651.835937
min	622.000000	192.000000
25%	4168.000000	3191.250000
50%	6255.500000	4165.500000
75%	9791.500000	5700.000000
max	19174.000000	10815.000000

In [146…
```python
fig = plt.figure(figsize=(20,10))

plt.plot(post_daily_like_comment_count.post_date,post_daily_like_comment_count.total_daily_like_count)
plt.plot(post_daily_like_comment_count.post_date,post_daily_like_comment_count.total_daily_comment_count)

plt.title('Variation of Daily Likes vs Comments')
plt.xlabel('Period')
plt.ylabel('Daily Count')
plt.grid(True)


#plt.show()
```



In [166…
```python
#to find most popular author, we could aggregate total like and comment activity for each author
#posts.loc[posts.post_date>= pd.to_datetime('09/07/2012')].
most_popular_authors=posts.groupby(["author_id"]).agg(
total_daily_like_count = pd.NamedAgg(column='like_count', aggfunc="sum"),
total_daily_comment_count = pd.NamedAgg(column='comment_count', aggfunc="sum")

).reset_index()
```

In [167…
```python
#then sum total likes and comments together and called this calculation total_post_interaction
most_popular_authors['total_post_interaction'] = most_popular_authors.total_daily_like_count+most_popular_authors.total_daily_comment_count
```

In [168…
```python
#top 10 most popular authors
most_popular_authors.sort_values(by=['total_post_interaction'], ascending=False).head(10)
```

Out[168]:

	author_id	total_daily_like_count	total_daily_comment_count	total_post_interaction
1937	17432410	305270	31407	336677
0	0	2080	310039	312119
4125	33424361	90354	58177	148531
3967	31743217	87402	48023	135425
497	4695191	0	132194	132194
6397	49069439	109436	16350	125786
2265	19811010	47511	63925	111436
2726	23477253	2595	106232	108827
3806	31173849	62714	42323	105037
5132	40001150	76717	21959	98676