

Case Study- Titok Data Analysis

Lets explore this data analyse and try to find the answer of following questions....

Ques. What is the maximum video durations, like, comment, share count and play count of the verified authore and the un-verified author?

Ques. which duration video is highly created in tittok?

Ques. What is the unique video duration of titok video?

Ques. What is the highest sum/maximum comments and video likes got in tittok video?

Ques. Which video duration got the highest sum of shares and play count in tittok dataset?

Ques. Which video duration got the maximum shares and play count in tittok dataset?

Data Set

Tiktok dataset contains 17 columns namely: Id, Text, Create_Time, AuthorMeta/Id, Author_Name, Author_Nick_Name, Author_Verified, Author_Signature, Music_Id, Music_Name, Music_Author, Music_Original, Video_Duration, Video_Like, Share_Count, Play_Count and Comment_Countand and the 1000 rows.

Here we have a information about:-

Author tiktok id, author name and the nick name, author account veriication status, author signature, music id prefered by the author, name of the prefered songs, music orginiality used to make the video, duration of tittok video prefered by the author, no. of likes and the comments contains in the particular duration videos, shares video and the how many times a particular duration video is play

Import Library

```
In [53]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

Read the data

```
In [54]: d=pd.read_csv(r'C:\Users\lxm\Desktop\trending.csv')
d
```

Out[54]:

	id	text	createTime	authorMeta/id	authorname	authornickName	author
0	6.907230e+18	Confidence went ✅	1608214517	6.825540e+18	ninakleij		Nina
1	6.875470e+18	Quiet Zone... follow me on insta: joeysfo. Co...	1600819763	6.729290e+18	joeysfo		JoeySofo
2	6.898700e+18	Iphone bend test 😊 #tiktok #viral #fyp #iphone	1606228625	6.791900e+18	jackyephone		JackJacko
		...					
3	6.902820e+18	NaN	1607187987	6.574080e+18	naomivaneeren		Naomi van eeren ✨
		...					
4	6.905640e+18	小技です 🌸 #tiktok教室 #tutorial	1607843600	6.586850e+18	io.dreamer_mk		io. Dreamer
		...					
995	6.877190e+18	#foryou #foryoupage	1601220970	6.788450e+18	artistmiranda		ArtistMiranda
		Stop eating 🍔 #gttfg #gotothegym #swolefam #nu...					
996	6.908070e+18	#fy #foryoupage #foryou	1608410366	6.718790e+18	papaswolio		Papa Swolio
		...					
997	6.883480e+18	regretss ✅ #fyp #foryou #curls	1602686079	6.792310e+18	sanaelfarah		Sana El Farah
		The collab you didn't know you needed, myself ...					
998	6.898720e+18	#fyp #foryoupage #curls	1606233872	6.957010e+16	safae.kx		Safae
		...					
999	6.899120e+18	Erin Williams	1606325682	6.798140e+18	erinwilliams_1		Erin Williams

1000 rows × 17 columns

Data Cleaning

Understanding the shape and size of the

data

```
In [55]: d.shape
```

```
Out[55]: (1000, 17)
```

Checking the null values in the dataset

```
In [56]: d.isnull().sum()
```

```
Out[56]: id          0  
text         38  
createTime    0  
authorMeta/id  0  
authorname     0  
authornickName 0  
authorverified 0  
authorsignature 86  
musicId        0  
musicName       0  
musicAuthor      0  
musicoriginal    0  
videoduration    0  
videolike        0  
shareCount       0  
playCount        0  
commentCount      0  
dtype: int64
```

Deleting unnecessary columns

```
In [57]: d.drop(['authorsignature', 'authorMeta/id', 'authorname', 'authornickName', 'authorverified', 'videoduration', 'shareCount', 'commentCount'], axis=1, inplace=True)
```

```
In [58]: d.drop(['musicName'], axis=1, inplace=True)
```

In [59]: d

Out[59]:

	id	createTime	authorverified	musicoriginal	videoduration	videolike	shareCount
0	6.907230e+18	1608214517	False	False	15	3710	50
1	6.875470e+18	1600819763	False	True	11	55700	1817
2	6.898700e+18	1606228625	False	False	19	936200	21100
3	6.902820e+18	1607187987	False	True	13	12900	197
4	6.905640e+18	1607843600	False	True	22	8805	198
...
995	6.877190e+18	1601220970	False	True	13	13300	152
996	6.908070e+18	1608410366	False	True	20	12200	223
997	6.883480e+18	1602686079	False	True	13	26600	3392
998	6.898720e+18	1606233872	False	True	12	10000	111
999	6.899120e+18	1606325682	False	False	14	58900	235

1000 rows × 9 columns

◀	▶
---	---

Statistical details of the dataset

In [60]: d.describe(include='all')

Out[60]:

	id	createTime	authorverified	musicoriginal	videoduration	videolike	
count	1.000000e+03	1.000000e+03		1000	1000	1000.000000	1.000000e+03
unique	NaN	NaN		2	2	NaN	NaN
top	NaN	NaN	False	True	NaN	NaN	NaN
freq	NaN	NaN		944	746	NaN	NaN
mean	6.893099e+18	1.604925e+09		NaN	NaN	19.055000	9.679788e+04
std	9.947072e+15	2.315981e+06		NaN	NaN	13.889813	1.043487e+06
min	6.875320e+18	1.600785e+09		NaN	NaN	4.000000	1.300000e+01
25%	6.884270e+18	1.602868e+09		NaN	NaN	10.000000	2.864500e+03
50%	6.893780e+18	1.605084e+09		NaN	NaN	14.000000	7.412500e+03
75%	6.902058e+18	1.607011e+09		NaN	NaN	21.000000	2.285000e+04
max	6.908540e+18	1.608521e+09		NaN	NaN	60.000000	3.100000e+07

◀	▶
---	---

Maximum durations,likes,comment,play_unique

via author account verification

In [61]:

```
d[d['authorverified']==True].max()
```

Out[61]:

```
id           69082900000000000000.0
createTime      1608462215
authorverified      True
musicoriginal      True
videoduration        59
videolike          31000000
shareCount         220100
playCount          250800000
commentCount        625700
dtype: object
```

In [62]:

```
d[d['authorverified']==False].max()
```

Out[62]:

```
id           69085400000000000000.0
createTime      1608520716
authorverified      False
musicoriginal      True
videoduration        60
videolike          5000000
shareCount         164700
playCount          68700000
commentCount        61200
dtype: object
```

Statistical information of the tiktok video duration

In [63]:

```
d['videoduration'].describe()
```

Out[63]:

```
count    1000.000000
mean     19.055000
std      13.889813
min      4.000000
25%     10.000000
50%     14.000000
75%     21.000000
max      60.000000
Name: videoduration, dtype: float64
```

Unique counts of tiktok video durations

```
In [64]: d['videoduration'].value_counts()
```

```
Out[64]: 15      152
13       80
10       69
8        66
9        63
14       55
12       48
11       46
7        39
59       29
6         21
21       20
16       19
17       19
20       18
18       16
29       15
30       14
5         13
19       13
25       12
58       12
22       10
23       10
53        8
33        8
26        8
57        8
41        7
28        7
24        7
31        6
34        6
54        6
42        6
52        5
35        5
37        5
39        4
36        4
27        4
46        4
55        4
51        3
32        3
47        3
45        3
44        3
43        3
40        3
50        2
56        2
4         1
49        1
48        1
```

```
60      1  
Name: videotoduration, dtype: int64
```

Checking the correlations in the dataset

In [65]:

```
d1=pd.DataFrame(d.index=['videoduration','videolike','shareCount','playCount','commentCount'])  
  
File "<ipython-input-65-ac28a13d1970>", line 1  
    d1=pd.DataFrame(d.index=['videoduration','videolike','shareCount','playCount','commentCount'])  
          ^  
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

Unique video duration in the dataset

In [66]:

```
d.videoduration.unique()
```

```
Out[66]: array([15, 11, 19, 13, 22, 29, 9, 8, 10, 7, 5, 12, 14, 52, 20, 43, 25,  
44, 21, 54, 35, 32, 33, 17, 59, 37, 24, 53, 60, 58, 18, 16, 40, 6,  
30, 23, 57, 55, 26, 45, 31, 34, 41, 39, 27, 51, 48, 56, 28, 50, 46,  
42, 47, 36, 49, 4], dtype=int64)
```

Groupby the video duration column in the dataset

```
In [67]: # Groupby of video duration wise sum of sharecount, playcount, videolike and commentCount

a=d.groupby(['videoduration'])['videolike','shareCount','playCount','commentCount'].sum()
a
```

Out[67]:

	videolike	shareCount	playCount	commentCount
videoduration				
4	2232	19	47900	27
5	358956	14373	4401200	5059
6	351873	10363	5165437	3323
7	1759822	39337	30958700	10408
8	8200236	233460	71201728	78118
9	32892102	266400	268700000	650100
10	5386648	83169	62634884	39297
11	1383615	19104	21549770	16654
12	10782905	237928	137499046	91286
13	2126698	86300	40486700	18306
14	10355945	156147	129932684	48142
15	5670581	298668	71977635	92818
16	948361	14803	16140500	15170
17	324909	6298	4433800	6177
18	457164	20928	4604493	9449
19	1589210	58595	20772300	41457
20	788211	19340	9960989	7129
21	1481964	83477	18544600	13886
22	479189	7135	6365725	4308
23	663994	10005	3330600	8091
24	459346	8095	5812500	2332
25	601559	3872	6039600	3631
26	293046	5037	2044400	3432
27	665836	7479	4663000	5840
28	105356	1460	783200	1719
29	1507068	42760	19090100	8474
30	266197	5196	2653100	6201
31	65560	633	773600	1254
32	56500	1952	247800	1223
33	139681	3495	1166600	1502
34	51977	1447	278300	1415
35	52402	1943	268199	965

	videolike	shareCount	playCount	commentCount
videoduration				
36	910482	39364	8327600	5263
37	62747	1239	1147800	908
39	173314	479	1020700	1672
40	2508513	26067	25232400	12274
41	52282	1896	699547	509
42	66229	392	510800	1618
43	37023	306	284700	822
44	36434	1279	276000	934
45	74678	220	614800	928
46	34166	103	272200	549
47	84173	2898	1032503	3571
48	38200	139	317500	525
49	16400	25	95000	520
50	9514	816	107600	225
51	36561	505	259300	1001
52	267856	3505	1031200	2543
53	381794	2177	2328000	6744
54	130048	1762	1438537	1638
55	57191	1037	447100	1541
56	54175	833	806800	1455
57	96468	2880	789500	2374
58	185682	5646	1554600	3385
59	1214095	62570	8083200	23319
60	712	5	6458	24

In [68]: # Groupby of video duration wise maximum of sharecount, playcount, videolike and
d1=d.groupby(['videoduration'])['videolike','shareCount','playCount','commentCount'].
d1

Out[68]:

	videolike	shareCount	playCount	commentCount
videoduration				
4	2232	19	47900	27
5	171700	11600	2000000	3770
6	171500	5230	2400000	1051
7	1200000	21300	24800000	4454
8	3900000	164700	29900000	61200
9	31000000	220100	250800000	625700
10	4000000	45900	37900000	18600
11	330000	3712	11200000	3767
12	5200000	102700	64500000	32000
13	500800	22800	21500000	2397
14	5000000	110900	68700000	12500
15	619000	60800	9200000	11100
16	179600	5317	7400000	6389
17	83700	2315	1200000	1501
18	229800	10900	2400000	5260
19	936200	34100	15300000	27100
20	214700	9077	3500000	2694
21	759200	51500	7900000	4100
22	305300	2421	3600000	1777
23	426100	6805	1500000	5574
24	340300	5077	4400000	864
25	243400	2386	3100000	1432
26	103900	1972	659700	1578
27	558800	3311	4100000	4522
28	46500	364	270600	902
29	822200	30600	12900000	2732
30	84500	1811	493900	3735
31	32100	268	393800	538
32	25000	1257	99400	659
33	34100	1809	327900	535
34	24700	1133	76900	640
35	45900	1799	170700	713

	videolike	shareCount	playCount	commentCount
videoduration				
36	590400	34400	6700000	2976
37	20600	758	558900	539
39	157700	386	821200	1314
40	2500000	26000	25100000	12000
41	21700	1103	292500	123
42	39200	190	230700	721
43	29200	147	197300	656
44	21300	991	139900	513
45	63100	107	525900	392
46	19000	51	84200	318
47	76600	2596	974900	3254
48	38200	139	317500	525
49	16400	25	95000	520
50	6447	719	81000	147
51	31100	305	163100	833
52	179100	2305	607700	1941
53	299500	763	1800000	2637
54	102500	1634	1200000	1243
55	32500	900	198300	730
56	48300	559	747400	1252
57	23000	2114	252600	1112
58	46500	2232	427300	886
59	692300	48200	3500000	10700
60	712	5	6458	24

Sort the dataset by video_duration wise sum of video_comments

In [69]:

```
a3=a.sort_values('commentCount', ascending=False)
a3
```

Out[69]:

	videolike	shareCount	playCount	commentCount
videoduration				
9	32892102	266400	268700000	650100
15	5670581	298668	71977635	92818
12	10782905	237928	137499046	91286
8	8200236	233460	71201728	78118
14	10355945	156147	129932684	48142
19	1589210	58595	20772300	41457
10	5386648	83169	62634884	39297
59	1214095	62570	8083200	23319
13	2126698	86300	40486700	18306
11	1383615	19104	21549770	16654
16	948361	14803	16140500	15170
21	1481964	83477	18544600	13886
40	2508513	26067	25232400	12274
7	1759822	39337	30958700	10408
18	457164	20928	4604493	9449
29	1507068	42760	19090100	8474
23	663994	10005	3330600	8091
20	788211	19340	9960989	7129
53	381794	2177	2328000	6744
30	266197	5196	2653100	6201
17	324909	6298	4433800	6177
27	665836	7479	4663000	5840
36	910482	39364	8327600	5263
5	358956	14373	4401200	5059
22	479189	7135	6365725	4308
25	601559	3872	6039600	3631
47	84173	2898	1032503	3571
26	293046	5037	2044400	3432
58	185682	5646	1554600	3385
6	351873	10363	5165437	3323
52	267856	3505	1031200	2543
57	96468	2880	789500	2374
24	459346	8095	5812500	2332

	videolike	shareCount	playCount	commentCount
videoduration				
28	105356	1460	783200	1719
39	173314	479	1020700	1672
54	130048	1762	1438537	1638
42	66229	392	510800	1618
55	57191	1037	447100	1541
33	139681	3495	1166600	1502
56	54175	833	806800	1455
34	51977	1447	278300	1415
31	65560	633	773600	1254
32	56500	1952	247800	1223
51	36561	505	259300	1001
35	52402	1943	268199	965
44	36434	1279	276000	934
45	74678	220	614800	928
37	62747	1239	1147800	908
43	37023	306	284700	822
46	34166	103	272200	549
48	38200	139	317500	525
49	16400	25	95000	520
41	52282	1896	699547	509
50	9514	816	107600	225
4	2232	19	47900	27
60	712	5	6458	24

Sort the dataset by video_duration wise maximum of video_comments

In [70]:

```
a4=d1.sort_values('commentCount', ascending=False)
a4
```

Out[70]:

	videolike	shareCount	playCount	commentCount
videoduration				
9	31000000	220100	250800000	625700
8	3900000	164700	29900000	61200
12	5200000	102700	64500000	32000
19	936200	34100	15300000	27100
10	4000000	45900	37900000	18600
14	5000000	110900	68700000	12500
40	2500000	26000	25100000	12000
15	619000	60800	9200000	11100
59	692300	48200	3500000	10700
16	179600	5317	7400000	6389
23	426100	6805	1500000	5574
18	229800	10900	2400000	5260
27	558800	3311	4100000	4522
7	1200000	21300	24800000	4454
21	759200	51500	7900000	4100
5	171700	11600	2000000	3770
11	330000	3712	11200000	3767
30	84500	1811	493900	3735
47	76600	2596	974900	3254
36	590400	34400	6700000	2976
29	822200	30600	12900000	2732
20	214700	9077	3500000	2694
53	299500	763	1800000	2637
13	500800	22800	21500000	2397
52	179100	2305	607700	1941
22	305300	2421	3600000	1777
26	103900	1972	659700	1578
17	83700	2315	1200000	1501
25	243400	2386	3100000	1432
39	157700	386	821200	1314
56	48300	559	747400	1252
54	102500	1634	1200000	1243
57	23000	2114	252600	1112

	videolike	shareCount	playCount	commentCount
videoduration				
6	171500	5230	2400000	1051
28	46500	364	270600	902
58	46500	2232	427300	886
24	340300	5077	4400000	864
51	31100	305	163100	833
55	32500	900	198300	730
42	39200	190	230700	721
35	45900	1799	170700	713
32	25000	1257	99400	659
43	29200	147	197300	656
34	24700	1133	76900	640
37	20600	758	558900	539
31	32100	268	393800	538
33	34100	1809	327900	535
48	38200	139	317500	525
49	16400	25	95000	520
44	21300	991	139900	513
45	63100	107	525900	392
46	19000	51	84200	318
50	6447	719	81000	147
41	21700	1103	292500	123
4	2232	19	47900	27
60	712	5	6458	24

Sort the dataset by video_duration wise sum of video likes

```
In [71]: a.sort_values('videolike', ascending=False)
```

```
Out[71]:      videolike  shareCount  playCount  commentCount
```

videoduration	videolike	shareCount	playCount	commentCount
9	32892102	266400	268700000	650100
12	10782905	237928	137499046	91286
14	10355945	156147	129932684	48142
8	8200236	233460	71201728	78118
15	5670581	298668	71977635	92818
10	5386648	83169	62634884	39297
40	2508513	26067	25232400	12274
13	2126698	86300	40486700	18306
7	1759822	39337	30958700	10408
19	1589210	58595	20772300	41457
29	1507068	42760	19090100	8474
21	1481964	83477	18544600	13886
11	1383615	19104	21549770	16654
59	1214095	62570	8083200	23319
16	948361	14803	16140500	15170
36	910482	39364	8327600	5263
20	788211	19340	9960989	7129
27	665836	7479	4663000	5840
23	663994	10005	3330600	8091
25	601559	3872	6039600	3631
22	479189	7135	6365725	4308
24	459346	8095	5812500	2332
18	457164	20928	4604493	9449
53	381794	2177	2328000	6744
5	358956	14373	4401200	5059
6	351873	10363	5165437	3323
17	324909	6298	4433800	6177
26	293046	5037	2044400	3432
52	267856	3505	1031200	2543
30	266197	5196	2653100	6201
58	185682	5646	1554600	3385
39	173314	479	1020700	1672
33	139681	3495	1166600	1502
54	130048	1762	1438537	1638

	videolike	shareCount	playCount	commentCount
videoduration				
28	105356	1460	783200	1719
57	96468	2880	789500	2374
47	84173	2898	1032503	3571
45	74678	220	614800	928
42	66229	392	510800	1618
31	65560	633	773600	1254
37	62747	1239	1147800	908
55	57191	1037	447100	1541
32	56500	1952	247800	1223
56	54175	833	806800	1455
35	52402	1943	268199	965
41	52282	1896	699547	509
34	51977	1447	278300	1415
48	38200	139	317500	525
43	37023	306	284700	822
51	36561	505	259300	1001
44	36434	1279	276000	934
46	34166	103	272200	549
49	16400	25	95000	520
50	9514	816	107600	225
4	2232	19	47900	27
60	712	5	6458	24

**Sort the dataset by video_duration wise
maximum of video_like**

In [72]: `d1.sort_values('videolike', ascending=False)`

Out[72]:

videoduration	videolike	shareCount	playCount	commentCount
9	31000000	220100	250800000	625700
12	5200000	102700	64500000	32000
14	5000000	110900	68700000	12500
10	4000000	45900	37900000	18600
8	3900000	164700	29900000	61200
40	2500000	26000	25100000	12000
7	1200000	21300	24800000	4454
19	936200	34100	15300000	27100
29	822200	30600	12900000	2732
21	759200	51500	7900000	4100
59	692300	48200	3500000	10700
15	619000	60800	9200000	11100
36	590400	34400	6700000	2976
27	558800	3311	4100000	4522
13	500800	22800	21500000	2397
23	426100	6805	1500000	5574
24	340300	5077	4400000	864
11	330000	3712	11200000	3767
22	305300	2421	3600000	1777
53	299500	763	1800000	2637
25	243400	2386	3100000	1432
18	229800	10900	2400000	5260
20	214700	9077	3500000	2694
16	179600	5317	7400000	6389
52	179100	2305	607700	1941
5	171700	11600	2000000	3770
6	171500	5230	2400000	1051
39	157700	386	821200	1314
26	103900	1972	659700	1578
54	102500	1634	1200000	1243
30	84500	1811	493900	3735
17	83700	2315	1200000	1501
47	76600	2596	974900	3254
45	63100	107	525900	392

videoduration	videolike	shareCount	playCount	commentCount
56	48300	559	747400	1252
58	46500	2232	427300	886
28	46500	364	270600	902
35	45900	1799	170700	713
42	39200	190	230700	721
48	38200	139	317500	525
33	34100	1809	327900	535
55	32500	900	198300	730
31	32100	268	393800	538
51	31100	305	163100	833
43	29200	147	197300	656
32	25000	1257	99400	659
34	24700	1133	76900	640
57	23000	2114	252600	1112
41	21700	1103	292500	123
44	21300	991	139900	513
37	20600	758	558900	539
46	19000	51	84200	318
49	16400	25	95000	520
50	6447	719	81000	147
4	2232	19	47900	27
60	712	5	6458	24

Sort the dataset by video_duration wise sum of video share

In [76]: `a.sort_values('shareCount', ascending=False)`

Out[76]:

	videolike	shareCount	playCount	commentCount
videoduration				
15	5670581	298668	71977635	92818
9	32892102	266400	268700000	650100
12	10782905	237928	137499046	91286
8	8200236	233460	71201728	78118
14	10355945	156147	129932684	48142
13	2126698	86300	40486700	18306
21	1481964	83477	18544600	13886
10	5386648	83169	62634884	39297
59	1214095	62570	8083200	23319
19	1589210	58595	20772300	41457
29	1507068	42760	19090100	8474
36	910482	39364	8327600	5263
7	1759822	39337	30958700	10408
40	2508513	26067	25232400	12274
18	457164	20928	4604493	9449
20	788211	19340	9960989	7129
11	1383615	19104	21549770	16654
16	948361	14803	16140500	15170
5	358956	14373	4401200	5059
6	351873	10363	5165437	3323
23	663994	10005	3330600	8091
24	459346	8095	5812500	2332
27	665836	7479	4663000	5840
22	479189	7135	6365725	4308
17	324909	6298	4433800	6177
58	185682	5646	1554600	3385
30	266197	5196	2653100	6201
26	293046	5037	2044400	3432
25	601559	3872	6039600	3631
52	267856	3505	1031200	2543
33	139681	3495	1166600	1502
47	84173	2898	1032503	3571
57	96468	2880	789500	2374
53	381794	2177	2328000	6744

	videolike	shareCount	playCount	commentCount
videoduration				
32	56500	1952	247800	1223
35	52402	1943	268199	965
41	52282	1896	699547	509
54	130048	1762	1438537	1638
28	105356	1460	783200	1719
34	51977	1447	278300	1415
44	36434	1279	276000	934
37	62747	1239	1147800	908
55	57191	1037	447100	1541
56	54175	833	806800	1455
50	9514	816	107600	225
31	65560	633	773600	1254
51	36561	505	259300	1001
39	173314	479	1020700	1672
42	66229	392	510800	1618
43	37023	306	284700	822
45	74678	220	614800	928
48	38200	139	317500	525
46	34166	103	272200	549
49	16400	25	95000	520
4	2232	19	47900	27
60	712	5	6458	24

**Sort the dataset by video_duration wise
maximum of video share**

In [79]: `d1.sort_values('shareCount', ascending=False)`

Out[79]:

	videolike	shareCount	playCount	commentCount
videoduration				
9	31000000	220100	250800000	625700
8	3900000	164700	29900000	61200
14	5000000	110900	68700000	12500
12	5200000	102700	64500000	32000
15	619000	60800	9200000	11100
21	759200	51500	7900000	4100
59	692300	48200	3500000	10700
10	4000000	45900	37900000	18600
36	590400	34400	6700000	2976
19	936200	34100	15300000	27100
29	822200	30600	12900000	2732
40	2500000	26000	25100000	12000
13	500800	22800	21500000	2397
7	1200000	21300	24800000	4454
5	171700	11600	2000000	3770
18	229800	10900	2400000	5260
20	214700	9077	3500000	2694
23	426100	6805	1500000	5574
16	179600	5317	7400000	6389
6	171500	5230	2400000	1051
24	340300	5077	4400000	864
11	330000	3712	11200000	3767
27	558800	3311	4100000	4522
47	76600	2596	974900	3254
22	305300	2421	3600000	1777
25	243400	2386	3100000	1432
17	83700	2315	1200000	1501
52	179100	2305	607700	1941
58	46500	2232	427300	886
57	23000	2114	252600	1112
26	103900	1972	659700	1578
30	84500	1811	493900	3735
33	34100	1809	327900	535
35	45900	1799	170700	713

	videolike	shareCount	playCount	commentCount
videoduration				
54	102500	1634	1200000	1243
32	25000	1257	99400	659
34	24700	1133	76900	640
41	21700	1103	292500	123
44	21300	991	139900	513
55	32500	900	198300	730
53	299500	763	1800000	2637
37	20600	758	558900	539
50	6447	719	81000	147
56	48300	559	747400	1252
39	157700	386	821200	1314
28	46500	364	270600	902
51	31100	305	163100	833
31	32100	268	393800	538
42	39200	190	230700	721
43	29200	147	197300	656
48	38200	139	317500	525
45	63100	107	525900	392
46	19000	51	84200	318
49	16400	25	95000	520
4	2232	19	47900	27
60	712	5	6458	24

Sort the dataset by video_duration wise sum of video play

```
In [74]: a.sort_values('playCount', ascending=False)
```

```
Out[74]:      videolike  shareCount  playCount  commentCount
```

videoduration	videolike	shareCount	playCount	commentCount
9	32892102	266400	268700000	650100
12	10782905	237928	137499046	91286
14	10355945	156147	129932684	48142
15	5670581	298668	71977635	92818
8	8200236	233460	71201728	78118
10	5386648	83169	62634884	39297
13	2126698	86300	40486700	18306
7	1759822	39337	30958700	10408
40	2508513	26067	25232400	12274
11	1383615	19104	21549770	16654
19	1589210	58595	20772300	41457
29	1507068	42760	19090100	8474
21	1481964	83477	18544600	13886
16	948361	14803	16140500	15170
20	788211	19340	9960989	7129
36	910482	39364	8327600	5263
59	1214095	62570	8083200	23319
22	479189	7135	6365725	4308
25	601559	3872	6039600	3631
24	459346	8095	5812500	2332
6	351873	10363	5165437	3323
27	665836	7479	4663000	5840
18	457164	20928	4604493	9449
17	324909	6298	4433800	6177
5	358956	14373	4401200	5059
23	663994	10005	3330600	8091
30	266197	5196	2653100	6201
53	381794	2177	2328000	6744
26	293046	5037	2044400	3432
58	185682	5646	1554600	3385
54	130048	1762	1438537	1638
33	139681	3495	1166600	1502
37	62747	1239	1147800	908
47	84173	2898	1032503	3571

	videolike	shareCount	playCount	commentCount
videoduration				
52	267856	3505	1031200	2543
39	173314	479	1020700	1672
56	54175	833	806800	1455
57	96468	2880	789500	2374
28	105356	1460	783200	1719
31	65560	633	773600	1254
41	52282	1896	699547	509
45	74678	220	614800	928
42	66229	392	510800	1618
55	57191	1037	447100	1541
48	38200	139	317500	525
43	37023	306	284700	822
34	51977	1447	278300	1415
44	36434	1279	276000	934
46	34166	103	272200	549
35	52402	1943	268199	965
51	36561	505	259300	1001
32	56500	1952	247800	1223
50	9514	816	107600	225
49	16400	25	95000	520
4	2232	19	47900	27
60	712	5	6458	24

```
In [80]: # Sort the dataset by video_duration wise maximum of video play
```

In [81]: `d1.sort_values('playCount', ascending=False)`

Out[81]:

	videolike	shareCount	playCount	commentCount
videoduration				
9	31000000	220100	250800000	625700
14	5000000	110900	68700000	12500
12	5200000	102700	64500000	32000
10	4000000	45900	37900000	18600
8	3900000	164700	29900000	61200
40	2500000	26000	25100000	12000
7	1200000	21300	24800000	4454
13	500800	22800	21500000	2397
19	936200	34100	15300000	27100
29	822200	30600	12900000	2732
11	330000	3712	11200000	3767
15	619000	60800	9200000	11100
21	759200	51500	7900000	4100
16	179600	5317	7400000	6389
36	590400	34400	6700000	2976
24	340300	5077	4400000	864
27	558800	3311	4100000	4522
22	305300	2421	3600000	1777
59	692300	48200	3500000	10700
20	214700	9077	3500000	2694
25	243400	2386	3100000	1432
6	171500	5230	2400000	1051
18	229800	10900	2400000	5260
5	171700	11600	2000000	3770
53	299500	763	1800000	2637
23	426100	6805	1500000	5574
17	83700	2315	1200000	1501
54	102500	1634	1200000	1243
47	76600	2596	974900	3254
39	157700	386	821200	1314
56	48300	559	747400	1252
26	103900	1972	659700	1578
52	179100	2305	607700	1941
37	20600	758	558900	539

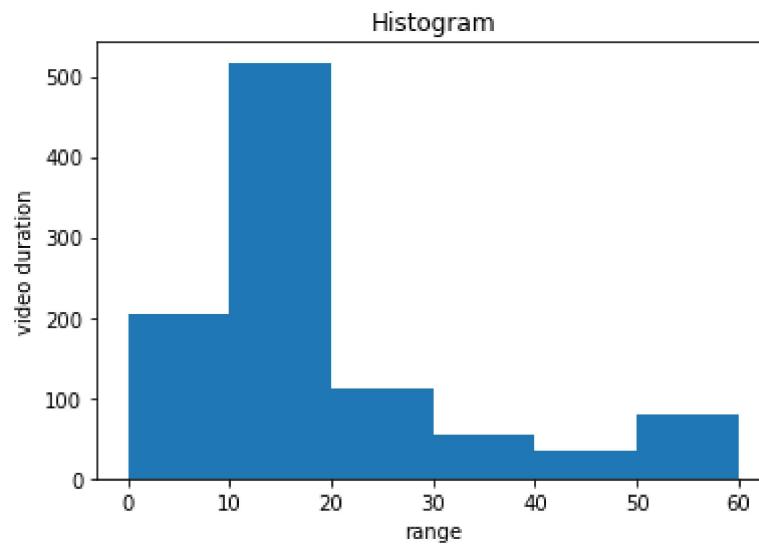
	videolike	shareCount	playCount	commentCount
videoduration				
45	63100	107	525900	392
30	84500	1811	493900	3735
58	46500	2232	427300	886
31	32100	268	393800	538
33	34100	1809	327900	535
48	38200	139	317500	525
41	21700	1103	292500	123
28	46500	364	270600	902
57	23000	2114	252600	1112
42	39200	190	230700	721
55	32500	900	198300	730
43	29200	147	197300	656
35	45900	1799	170700	713
51	31100	305	163100	833
44	21300	991	139900	513
32	25000	1257	99400	659
49	16400	25	95000	520
46	19000	51	84200	318
50	6447	719	81000	147
34	24700	1133	76900	640
4	2232	19	47900	27
60	712	5	6458	24

Data Visualization

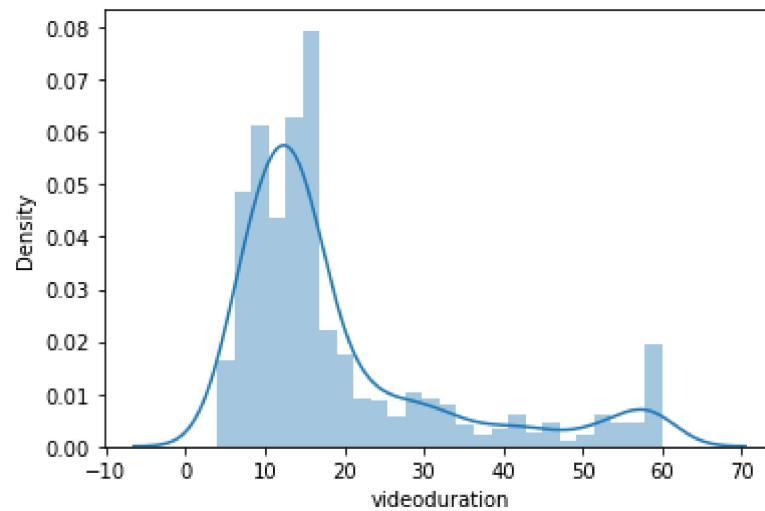
```
In [75]: import warnings  
warnings.filterwarnings('ignore')
```

Histogram

```
In [24]: age = d.videoduration  
bins = [0,10,20,30,40,50,60]  
plt.hist(age,bins,histtype='stepfilled',rwidth=0.6)  
plt.title("Histogram")  
plt.xlabel('range')  
plt.ylabel('video duration')  
plt.show()
```

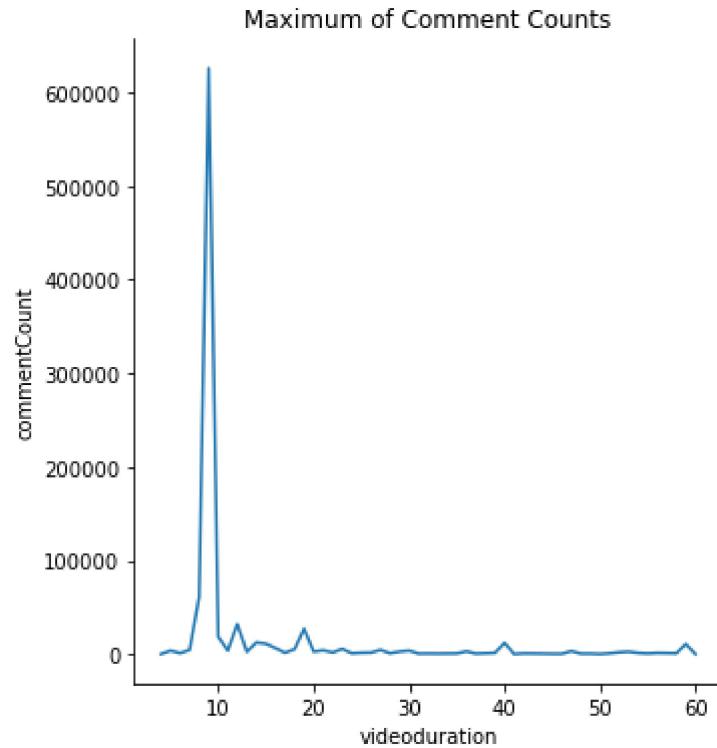


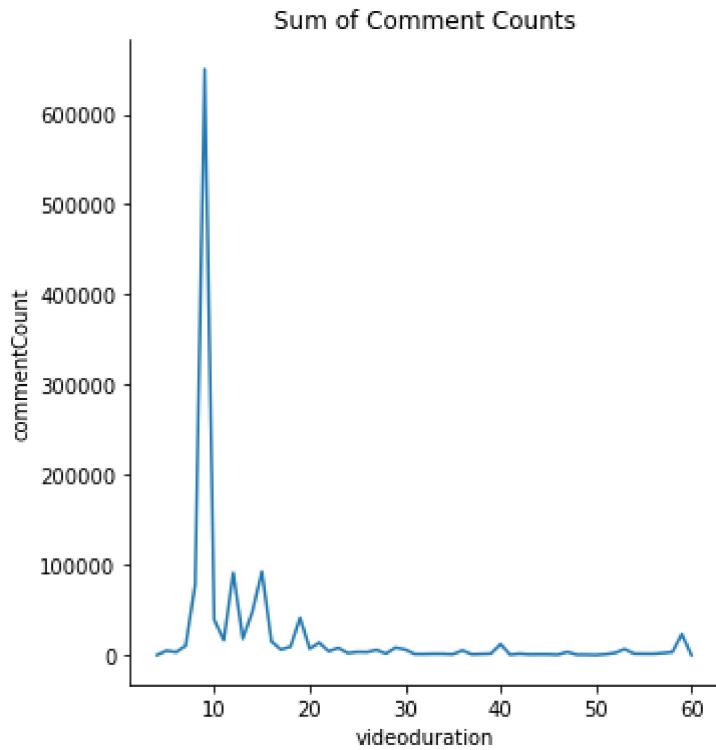
```
In [25]: sns.distplot(d['videoduration'])  
plt.show()
```



```
In [90]: sns.relplot(x='videoduration',y='commentCount',kind='line',data=d1).set(title='Maximum of Comment Counts')
plt.show()

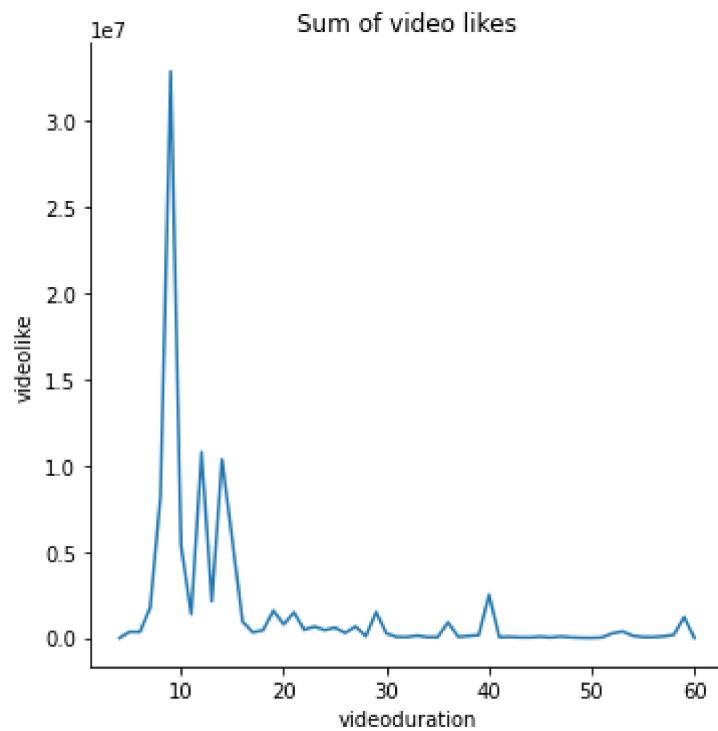
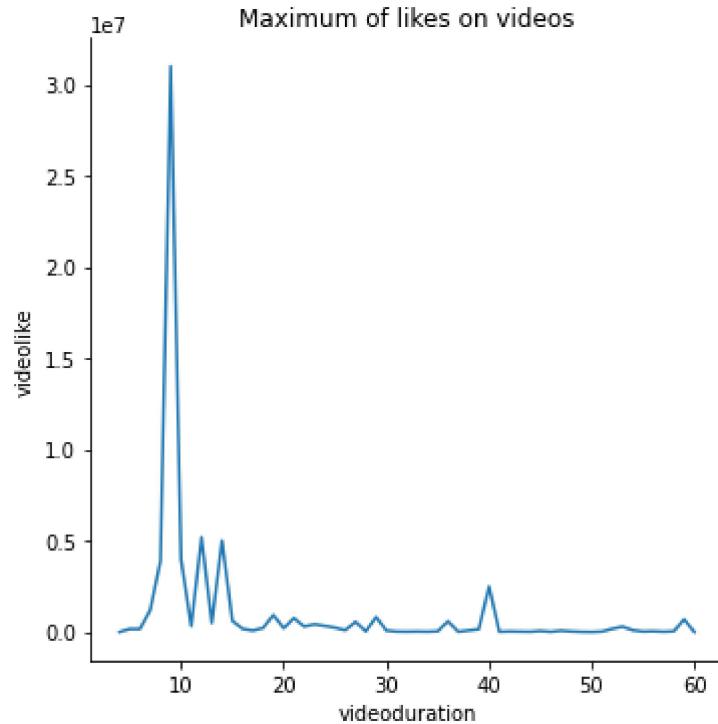
sns.relplot(x='videoduration',y='commentCount',kind='line',data=a).set(title='Summary of Comment Counts')
plt.show()
```





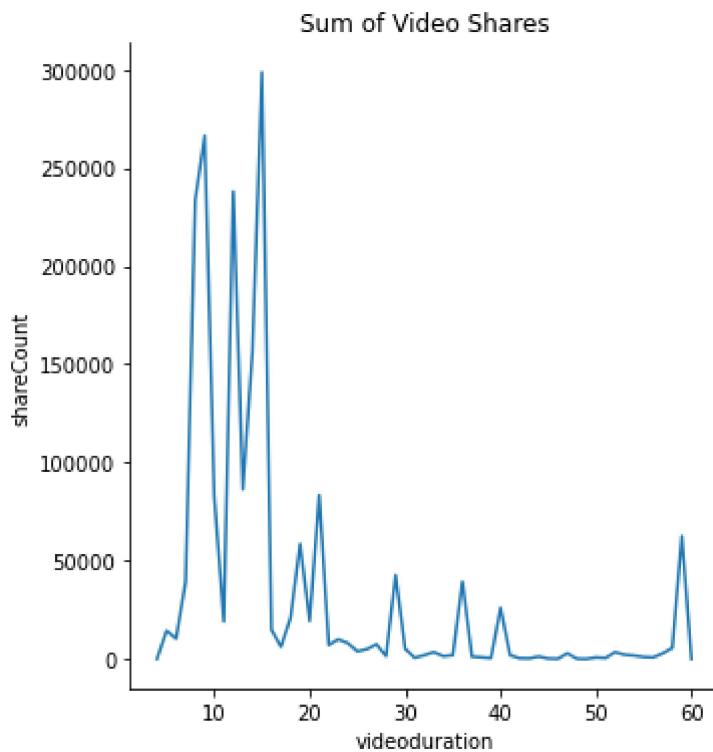
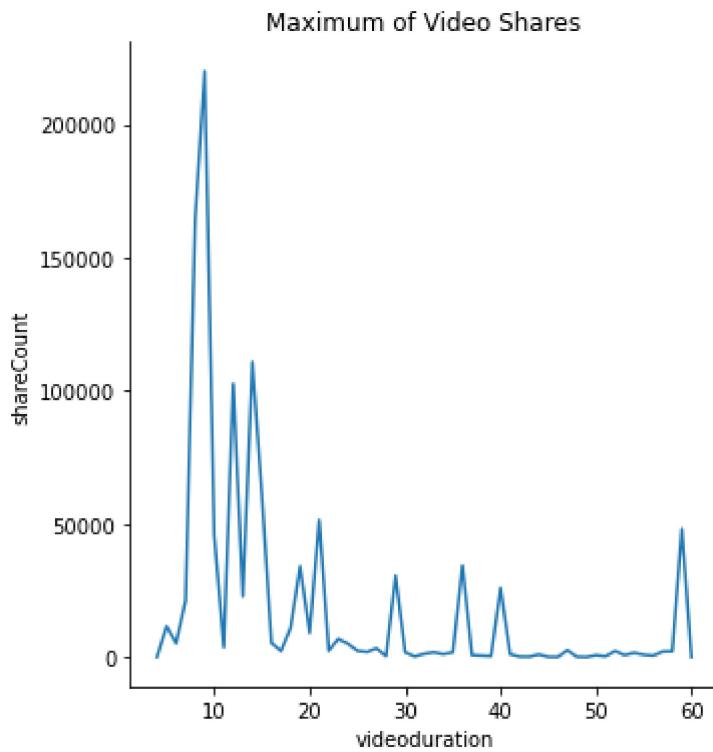
```
In [89]: sns.relplot(x='videoduration',y='videolike',kind='line',data=d1).set(title='Maximum of likes on videos')
plt.show()

sns.relplot(x='videoduration',y='videolike',kind='line',data=a).set(title='Sum of video likes')
plt.show()
```

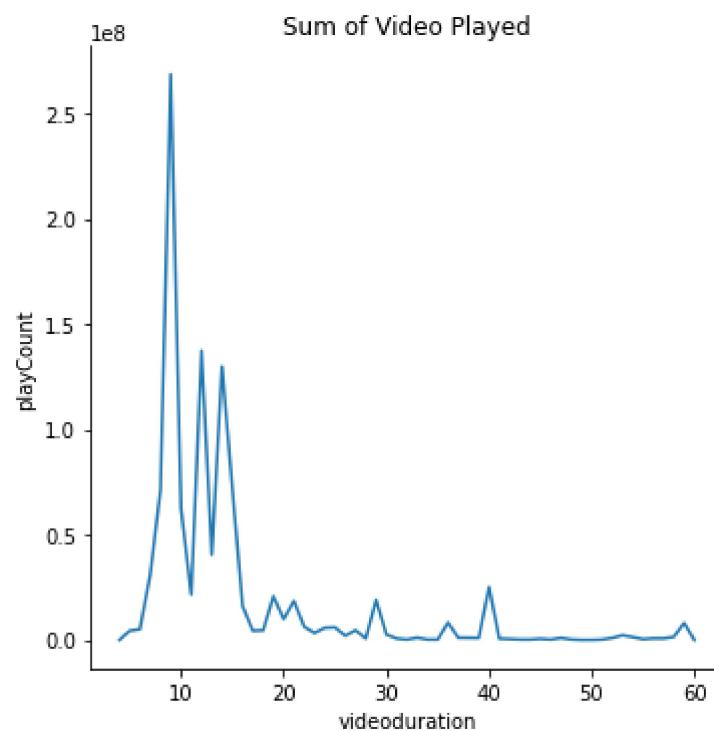
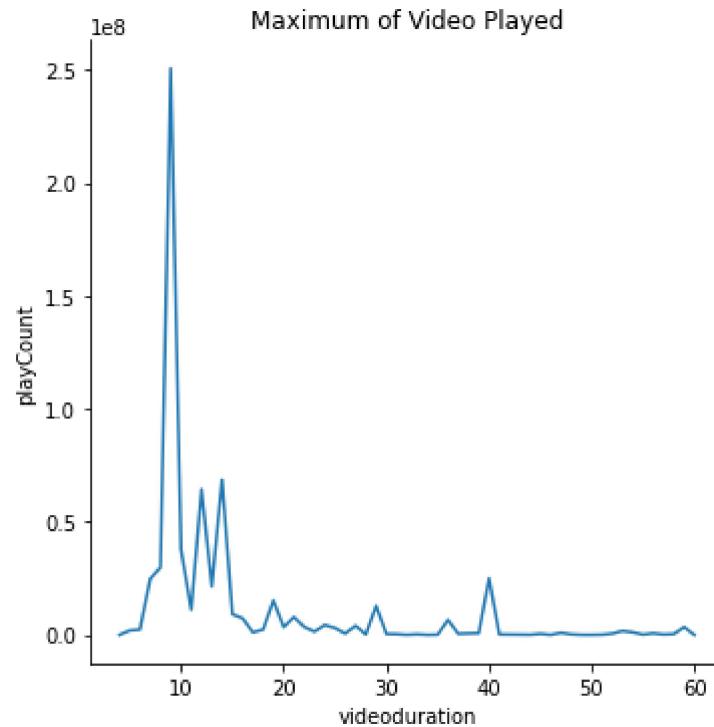


```
In [91]: sns.relplot(x='videoduration',y='shareCount',kind='line',data=d1).set(title='Maximum of Video Shares')
plt.show()

sns.relplot(x='videoduration',y='shareCount',kind='line',data=a).set(title='Sum of Video Shares')
plt.show()
```



```
In [92]: sns.relplot(x='videoduration',y='playCount',kind='line',data=d1).set(title='Maximum of Video Played')
sns.relplot(x='videoduration',y='playCount',kind='line',data=a).set(title='Sum of Video Played')
plt.show()
```



Conclusion

We are storing the tiktok data into the d variable which contains the 1000 rows and 17 columns: id, text, CreateTime, AuthorMeta/Id, Author Name, AuthorNick Name, Authorverified, Authorsignature, Music-Id, Music Name, Music Author, Music Original, Video Duration, Video like, Share Count, Play Count, and Comment Count.

We droped the un-necessary columns i.e. Music Name, Author Signature, AuthorMeta/id, Author Name, Author Nick Name, Author Signature, Text, MusicId, and Music Author.

**** Now, We are try to answer of the above questions****

People usually makes 15 duration video which is showing the highest count in the dataset i.e. 152.

The maximum duration of video in tiktok is 60 seconds and minimum duration 4 seconds.

The highest sum of video like in the dataset is 32892102 and the maximum video like is 31000000 of the video duration 9 seconds.

The highest sum of comments on the video in the dataset is 650100 and maximum comments is 625700 of the duration 9 seconds.

The highest sum of video share is 298668 of the video duration 15 and maximum video shaew is

220100 of the video duration 9 seconds.

The highest sum of video played in tiktok is 268700000 and maximum video play is 250800000 of the duration 9 sec.

As per the data author who verified their account has able to achieve the highest likes, share counts, play counts, and comment counts on the video as compare to those authore who haven't verified their accounts. ¶

As per the data, there is a positive relation between comments and likes, comments and share counts, comments and play count. And the negative relations is between share counts and the video duration, video duration and the comments, likes, and the play counts.