

Sheet

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

DATASET

```
df = pd.read_csv('Dataset1A.csv')
df.head()
```

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	v
0	2kyS6SvSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANtell martin	7
1	1ZAPwrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonight trump presidency "last week ...	2
2	5qpjK5DgCt4	17.14.11	Racist Superman Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman "rudy "mancuso "king "bach"...	3
3	puqaWrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13T11:00:04.000Z	rhett and link "gmm " "good mythical morning" "...	3
4	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	24	2017-11-13T11:00:04.000Z	ryan "higa " "higatv " "nigahiga " "i dare ...	2

```
new = df.loc[(df['ratings_disabled'] == False)]
```

```
new.shape[0]
```

```
40780
```

```
stats = new[['video_id', "title", 'views', 'likes', 'dislikes']]
stats.head().sort_values(by="views", ascending=False)
```

	video_id	title	views	likes	dislikes
2	5qpjK5DgCt4	Racist Superman Rudy Mancuso, King Bach & Le...	3191434	146033	5339
1	1ZAPwrtAFY	The Trump Presidency: Last Week Tonight with J...	2418783	97185	6146
4	d380meD0W0M	I Dare You: GOING BALD!?	2095731	132235	1989
0	2kyS6SvSYSE	WE WANT TO TALK ABOUT OUR MARRIAGE	748374	57527	2966
3	puqaWrEC7tY	Nickelback Lyrics: Real or Fake?	343168	10172	666

```
# stats["%like"] = stats["likes"] / stats["views"] * 100
# stats["%dislike"] = stats["dislikes"] / stats["views"] * 100
```

Ukuran Pusat

```
mean_views = stats['views'].mean()
mean_likes = stats['likes'].mean()
mean_dislikes = stats['dislikes'].mean()

median_views = stats['views'].median()
median_likes = stats['likes'].median()
median_dislikes = stats['dislikes'].median()

mode_views = stats['views'].mode()[0]
mode_likes = stats['likes'].mode()[0]
mode_dislikes = stats['dislikes'].mode()[0]
```

```
print("Mean of \nViews: {}\nLikes: {}\nDislikes: {}".format(mean_views, mean_likes, mean_dislikes))
print()
print("Median of \nViews: {}\nLikes: {}\nDislikes: {}".format(median_views, median_likes, median_dislikes))
print()
print("Mode of \nViews: {}\nLikes: {}\nDislikes: {}".format(mode_views, mode_likes, mode_dislikes))
```

```
Mean of
Views: 2353475.8211378125
Likes: 74574.47763609613
Dislikes: 3726.781633153507
```

```
Median of
Views: 682796.0
Likes: 18261.5
Dislikes: 3726.781633153507
```

```
Mode of
Views: 2078
Likes: 2
Dislikes: 0
```

Ukuran Variasi

```
std_views = stats['views'].std(skipna = True)
std_likes = stats['likes'].std(skipna = True)
std_dislikes = stats['dislikes'].std(skipna = True)
```

```
print("Standard Deviation of \nViews: {}\nLikes: {}\nDislikes: {}".format(std_views, std_likes, std_dislikes))
```

```
Standard Deviation of
Views: 7372046.893439033
Likes: 229309.08930018847
Dislikes: 29088.812231905446
```

Ukuran Lokasi

```
loc_views = stats['views'].quantile([0.25,0.5,0.75])
loc_likes = stats['likes'].quantile([0.25,0.5,0.75])
loc_dislikes = stats['dislikes'].quantile([0.25,0.5,0.75])
```

```
print("Quartiles of \nViews:\n{}\nLikes:\n{}\nDislikes:\n{}".format(loc_views, loc_likes, loc_dislikes))
```

```

Quartiles of
Views:
0.25      242924.25
0.50      682796.00
0.75      1825707.75
Name: views, dtype: float64
Likes:
0.25       5522.5
0.50      18261.5
0.75      55711.5
Name: likes, dtype: float64
Dislikes:
0.25       206.0
0.50       636.0
0.75      1949.0
Name: dislikes, dtype: float64

```

```

q1views, q3views = np.percentile(stats['views'], 25), np.percentile(stats['views'], 75)
q1likes, q3likes = np.percentile(stats['likes'], 25), np.percentile(stats['likes'], 75)
q1dislikes, q3dislikes = np.percentile(stats['dislikes'], 25), np.percentile(stats['dislikes'], 75)

```

```

iqr_views = q3views - q1views
iqr_likes = q3likes - q1likes
iqr_dislikes = q3dislikes - q1dislikes

```

```

print("IQR of \nViews: {}\nLikes: {}\nDislikes: {}".format(iqr_views, iqr_likes, iqr_dislikes))

```

```

IQR of
Views: 1582783.5
Likes: 50189.0
Dislikes: 1743.0

```

```

outliers_val_views = 1.5 * iqr_views
outliers_val_likes = 1.5 * iqr_likes
outliers_val_dislikes = 1.5 * iqr_dislikes

```

```

high_view = q3views+outliers_val_views
low_view = q1views-outliers_val_views

high_likes = q3likes+outliers_val_likes
low_likes = q1likes-outliers_val_likes

high_dislikes = q3dislikes+outliers_val_dislikes
low_dislikes = q1dislikes-outliers_val_dislikes

```

```

print("Outliers for Views is higher than {} and lower than {}".format(high_view, low_view))
print("Outliers for Likes is higher than {} and lower than {}".format(high_likes, low_likes))
print("Outliers for Dislikes is higher than {} and lower than {}".format(high_dislikes, low_dislikes))

```

```

Outliers for Views is higher than 4199883.0 and lower than -2131251.0
Outliers for Likes is higher than 130995.0 and lower than -69761.0
Outliers for Dislikes is higher than 4563.5 and lower than -2408.5

```

Visualisasi

Histogram

```

stats.describe()

```

	views	likes	dislikes
count	4.078000e+04	4.078000e+04	4.078000e+04
mean	2.353476e+06	7.457448e+04	3.726782e+03
std	7.372047e+06	2.293091e+05	2.908881e+04
min	5.490000e+02	0.000000e+00	0.000000e+00
25%	2.429242e+05	5.522500e+03	2.060000e+02
50%	6.827960e+05	1.826150e+04	6.360000e+02
75%	1.825708e+06	5.571150e+04	1.949000e+03
max	2.353476e+06	7.457448e+04	3.726782e+03

```

n = 40949
num_inter_views = (2*iqr_views)/n**(1/3)
num_inter_likes = (2*iqr_likes)/n**(1/3)
num_inter_dislikes = (2*iqr_dislikes)/n**(1/3)

range_views = stats['views'].max() - stats['views'].min()
bins_views = range_views / num_inter_views

range_likes = stats['likes'].max() - stats['likes'].min()
bins_likes = range_likes / num_inter_likes

range_dislikes = stats['dislikes'].max() - stats['dislikes'].min()
bins_dislikes = range_dislikes / num_inter_dislikes

```

```
sns.set(rc={'figure.figsize':(15,5)})
```

Views

```

mydatav = stats["views"]
mydatav2 = stats[stats["views"] < high_view]
mydatav2 = mydatav2["views"]

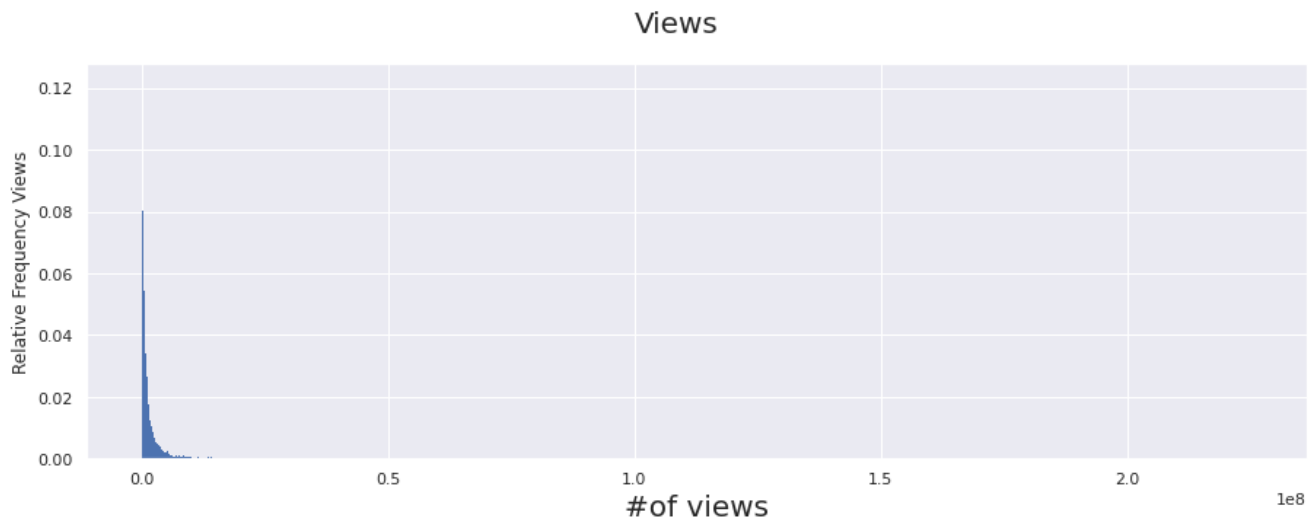
```

```

#histogram views dengan bins rumus
fig = plt.figure()
ax = fig.add_subplot(111)
n, bins, patches = ax.hist(mydatav, weights=np.zeros_like(mydatav) + 1. / mydatav.size, bins = round(
fig.suptitle('Views', fontsize=20)
ax.set_xlabel('#of views', size=20)
ax.set_ylabel('Relative Frequency Views')
ax.legend

plt.show()

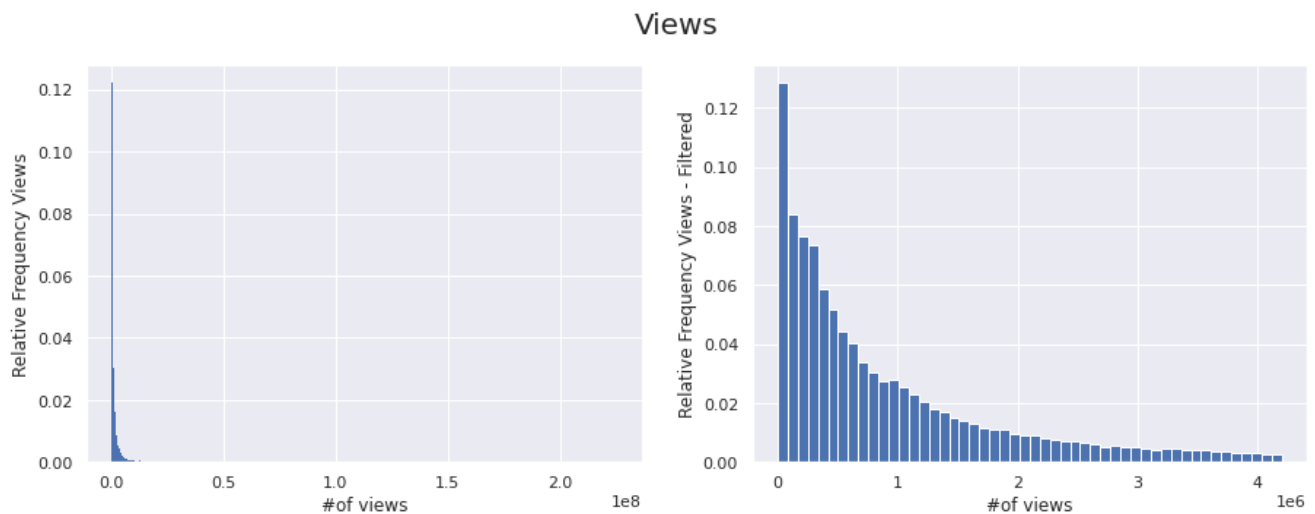
```



```
fig, ax = plt.subplots(1, 2)
n, bins, patches = ax[0].hist(mydatav, weights=np.zeros_like(mydatav) + 1. / mydatav.size, bins = round(n2, bins2), patches2 = ax[1].hist(mydatav2, weights=np.zeros_like(mydatav2) + 1. / mydatav2.size, bins = round(n2, bins2))
fig.suptitle('Views', fontsize=20)
ax[0].set_xlabel('#of views')
ax[0].set_ylabel('Relative Frequency Views')

ax[1].set_xlabel("#of views")
ax[1].set_ylabel('Relative Frequency Views - Filtered')
# ax.legend

plt.show()
```

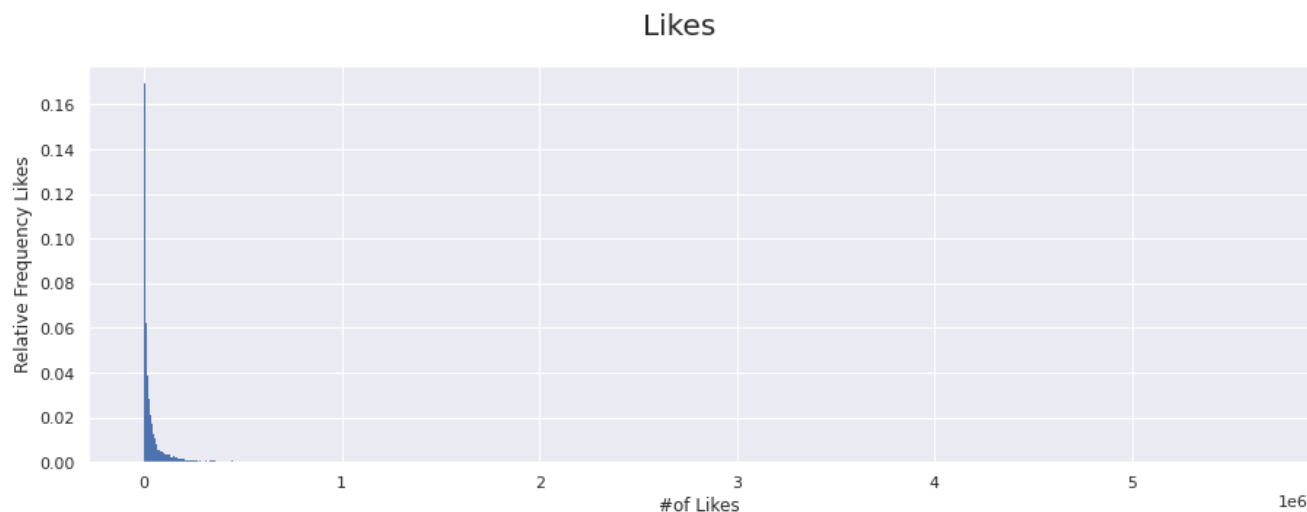


Likes

```
mydata1 = stats["likes"]
mydata12 = stats[stats["likes"] < high_likes]
mydata12 = mydata12["likes"]
```

```
#histogram likes dengan bin custom
fig = plt.figure()
ax = fig.add_subplot(111)
n, bins, patches = ax.hist(mydata1, weights=np.zeros_like(mydata1) + 1. / mydata1.size, bins = round(
fig.suptitle('Likes', fontsize=20)
ax.set_xlabel('#of Likes')
ax.set_ylabel('Relative Frequency Likes')
ax.legend

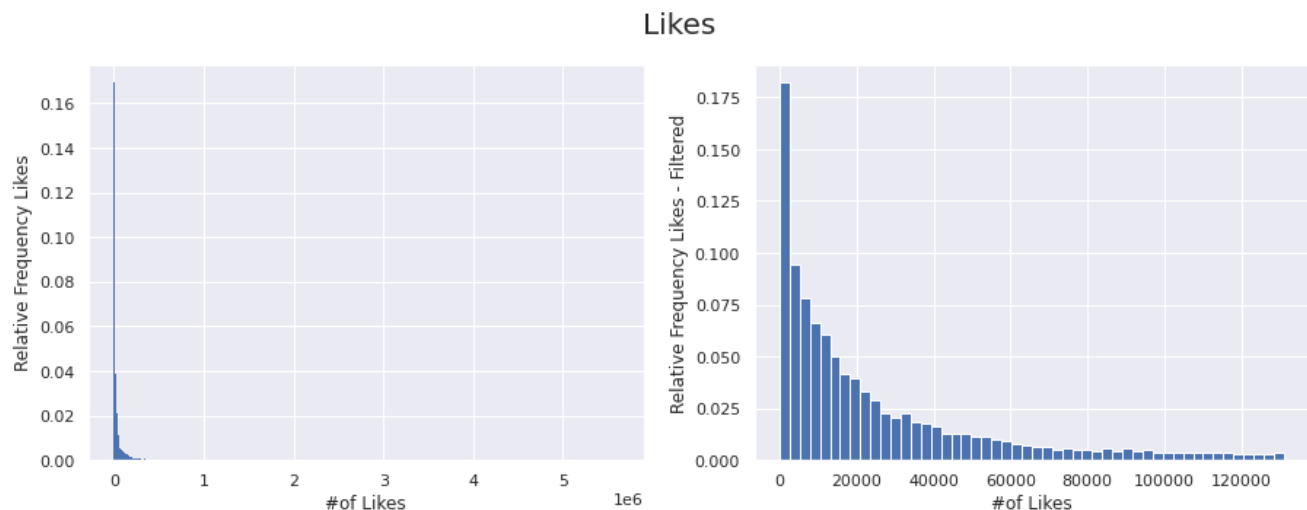
plt.show()
```



```
fig, ax = plt.subplots(1, 2)
n, bins, patches = ax[0].hist(mydata1, weights=np.zeros_like(mydata1) + 1. / mydata1.size, bins = round(
n2, bins2, patches2 = ax[1].hist(mydata12, weights=np.zeros_like(mydata12) + 1. / mydata12.size, bins = round(
fig.suptitle('Likes', fontsize=20)
ax[0].set_xlabel('#of Likes')
ax[0].set_ylabel('Relative Frequency Likes')

ax[1].set_xlabel("#of Likes")
ax[1].set_ylabel('Relative Frequency Likes - Filtered')
# ax.legend

plt.show()
```

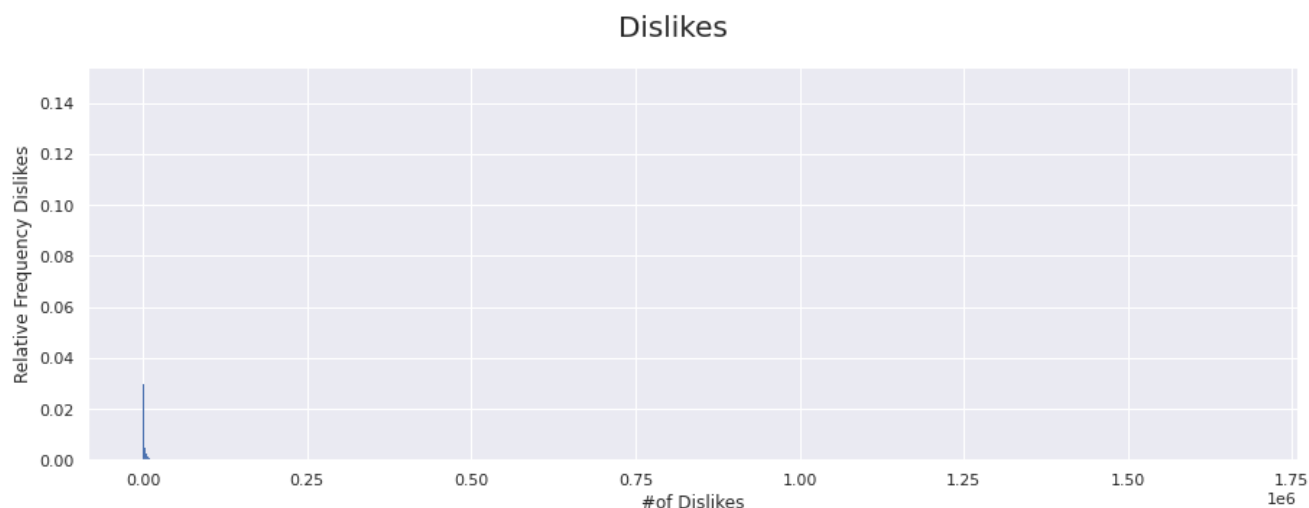


Dislikes

```
mydatad = stats["dislikes"]
mydatad2 = stats[stats["dislikes"] < high_dislikes]
mydatad2 = mydatad2["dislikes"]
```

```
fig = plt.figure()
ax = fig.add_subplot(111)
n, bins, patches = ax.hist(mydatad, weights=np.zeros_like(mydatad) + 1. / mydatad.size, bins = round(
fig.suptitle('Dislikes', fontsize=20)
ax.set_xlabel('#of Dislikes')
ax.set_ylabel('Relative Frequency Dislikes')
ax.legend

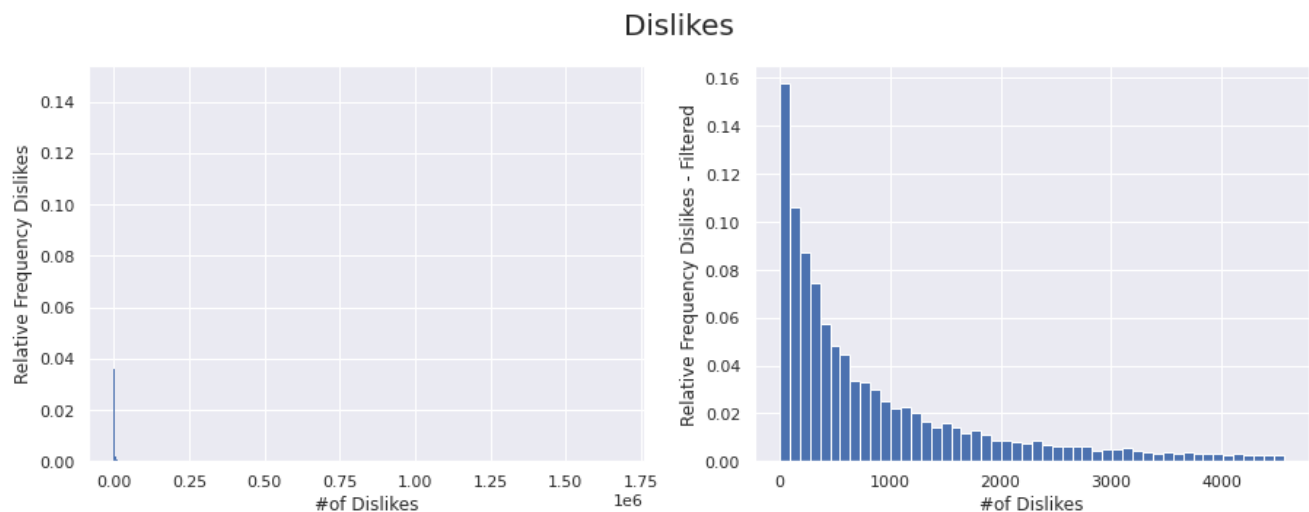
plt.show()
```



```
fig, ax = plt.subplots(1, 2)
n, bins, patches = ax[0].hist(mydatad, weights=np.zeros_like(mydatad) + 1. / mydatad.size, bins = round(
n2, bins2, patches2 = ax[1].hist(mydatad2, weights=np.zeros_like(mydatad2) + 1. / mydatad2.size, bins = round(
fig.suptitle('Dislikes', fontsize=20)
ax[0].set_xlabel('#of Dislikes')
ax[0].set_ylabel('Relative Frequency Dislikes')

ax[1].set_xlabel("#of Dislikes")
ax[1].set_ylabel('Relative Frequency Dislikes - Filtered')
# ax.legend

plt.show()
```



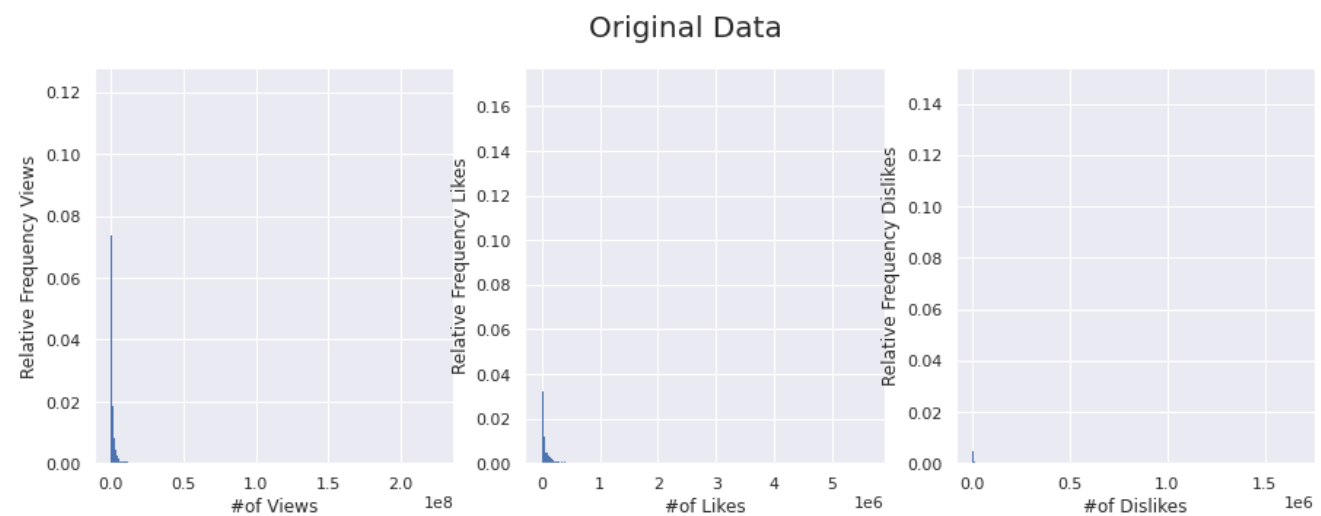
Original Data

```
fig, ax = plt.subplots(1, 3)
n, bins, patches = ax[0].hist(mydatav, weights=np.zeros_like(mydatav) + 1. / mydatav.size, bins = rou
n2, bins2, patches2 = ax[1].hist(mydata1, weights=np.zeros_like(mydata1) + 1. / mydata1.size, bins =
n3, bins3, patches3 = ax[2].hist(mydatad, weights=np.zeros_like(mydatad) + 1. / mydatad.size, bins =

fig.suptitle('Original Data', fontsize=20)

ax[0].set_xlabel('#of Views')
ax[0].set_ylabel('Relative Frequency Views')
ax[1].set_xlabel('#of Likes')
ax[1].set_ylabel('Relative Frequency Likes')
ax[2].set_xlabel('#of Dislikes')
ax[2].set_ylabel('Relative Frequency Dislikes')

plt.show()
```



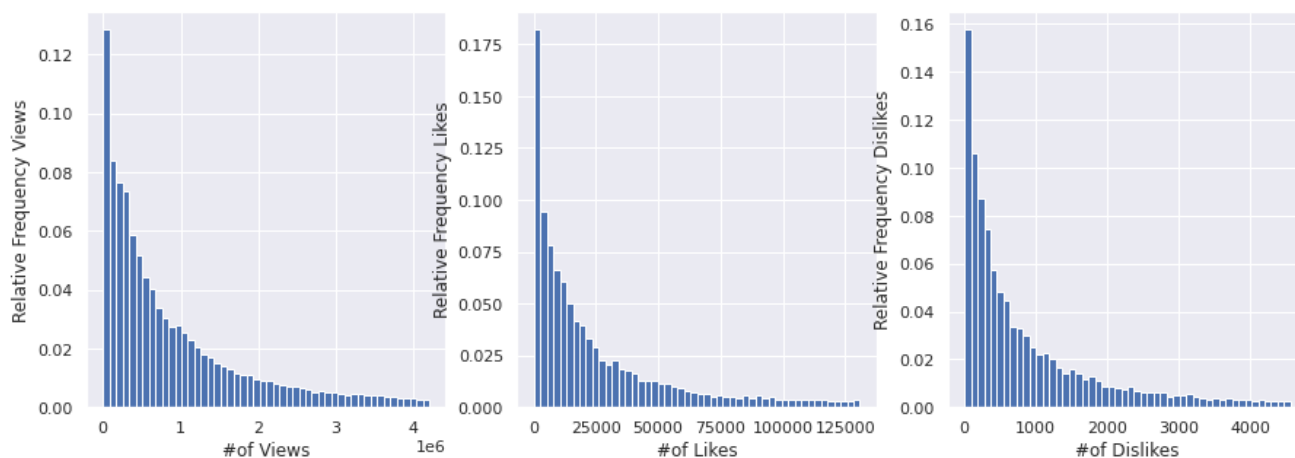
Filtered Data

```
fig, ax = plt.subplots(1, 3)
n, bins, patches = ax[0].hist(mydatav2, weights=np.zeros_like(mydatav2) + 1. / mydatav2.size, bins =
n2, bins2, patches2 = ax[1].hist(mydata12, weights=np.zeros_like(mydata12) + 1. / mydata12.size, bins =
n3, bins3, patches3 = ax[2].hist(mydatad2, weights=np.zeros_like(mydatad2) + 1. / mydatad2.size, bins =
fig.suptitle('Filtered Data', fontsize=20)

ax[0].set_xlabel('#of Views')
ax[0].set_ylabel('Relative Frequency Views')
ax[1].set_xlabel('#of Likes')
ax[1].set_ylabel('Relative Frequency Likes')
ax[2].set_xlabel('#of Dislikes')
ax[2].set_ylabel('Relative Frequency Dislikes')
# ax.legend

plt.show()
```


Filtered Data



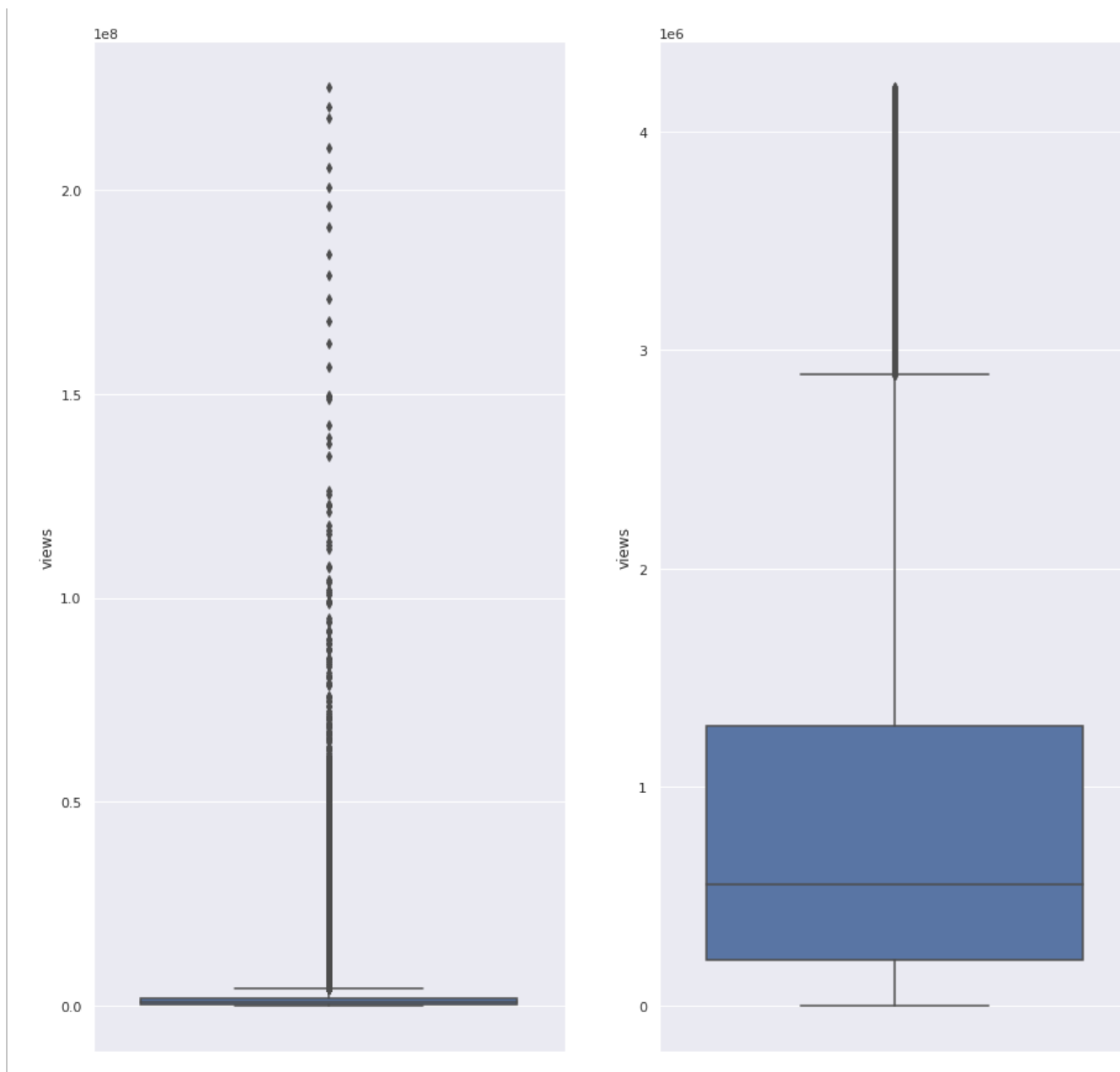
Box and Plot

```
sns.set(rc={'figure.figsize':(15,15)})
```

Views

```
fig, ax = plt.subplots(1, 2)
sns.boxplot(y=mydatav, ax=ax[0])
sns.boxplot(y=mydatav2, ax=ax[1])

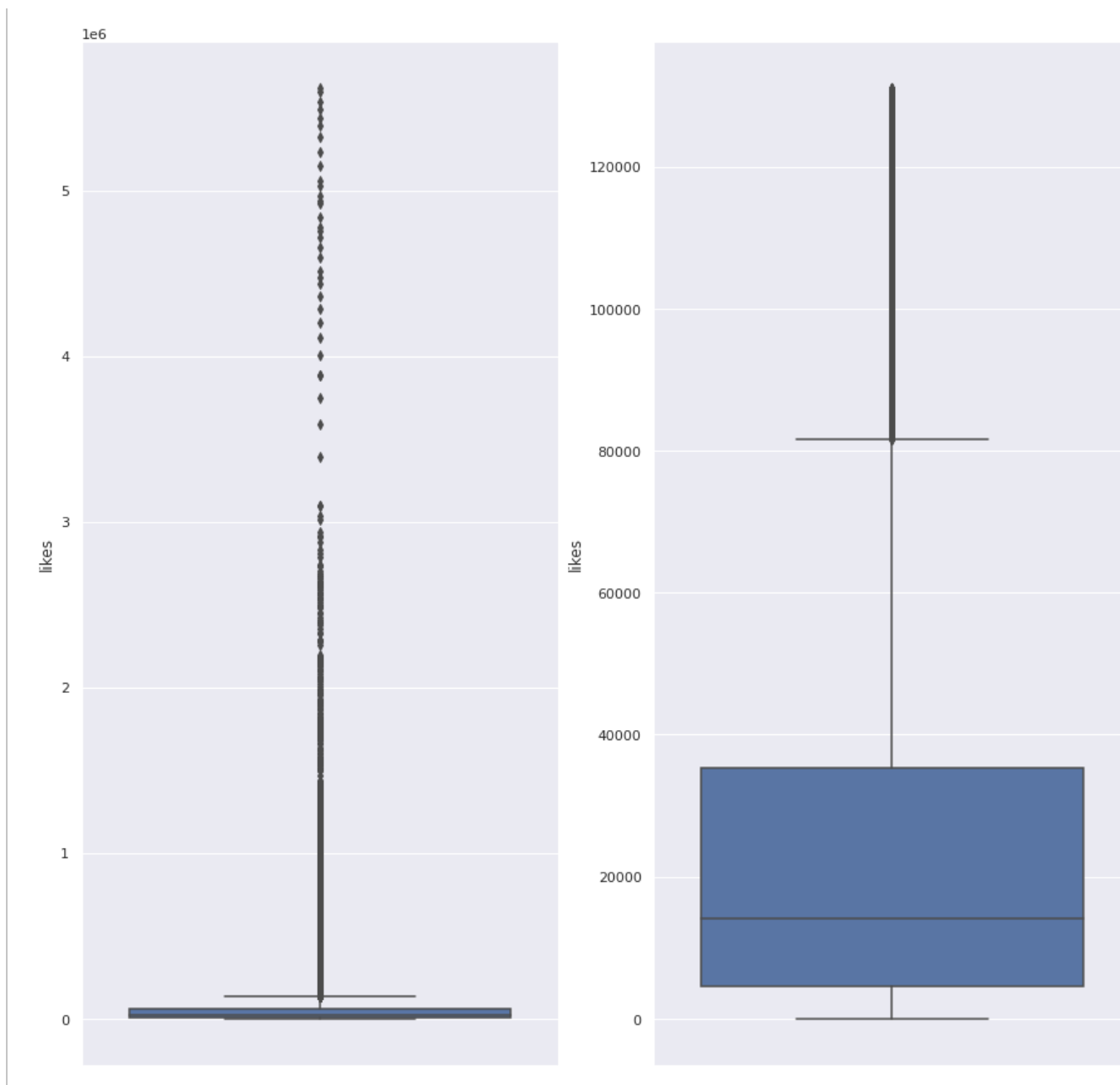
plt.show()
```



Likes

```
fig, ax = plt.subplots(1, 2)
sns.boxplot(y=mydata1, ax=ax[0])
sns.boxplot(y=mydata2, ax=ax[1])

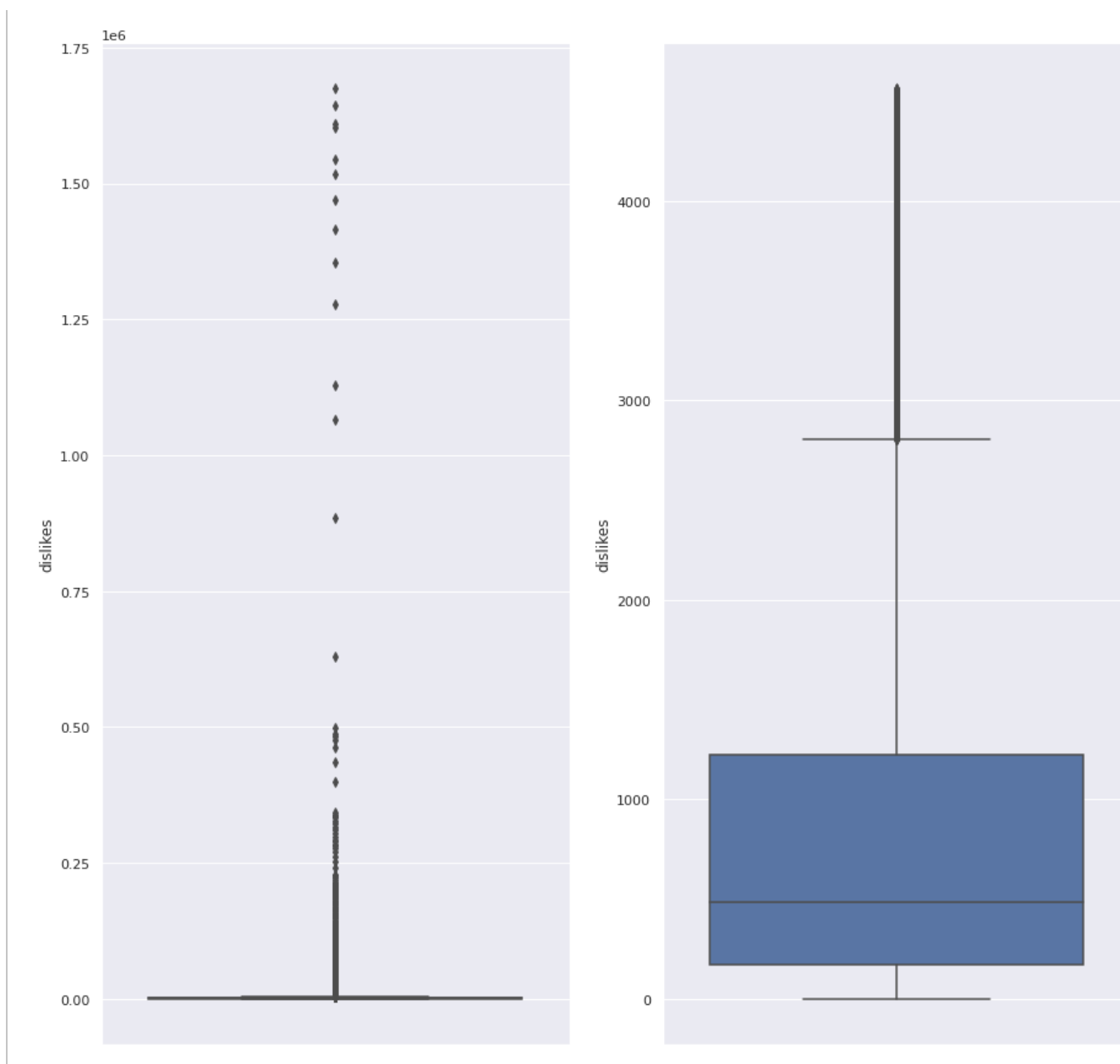
plt.show()
```



Dislikes

```
fig, ax = plt.subplots(1, 2)
sns.boxplot(y=mydatad, ax=ax[0])
sns.boxplot(y=mydatad2, ax=ax[1])

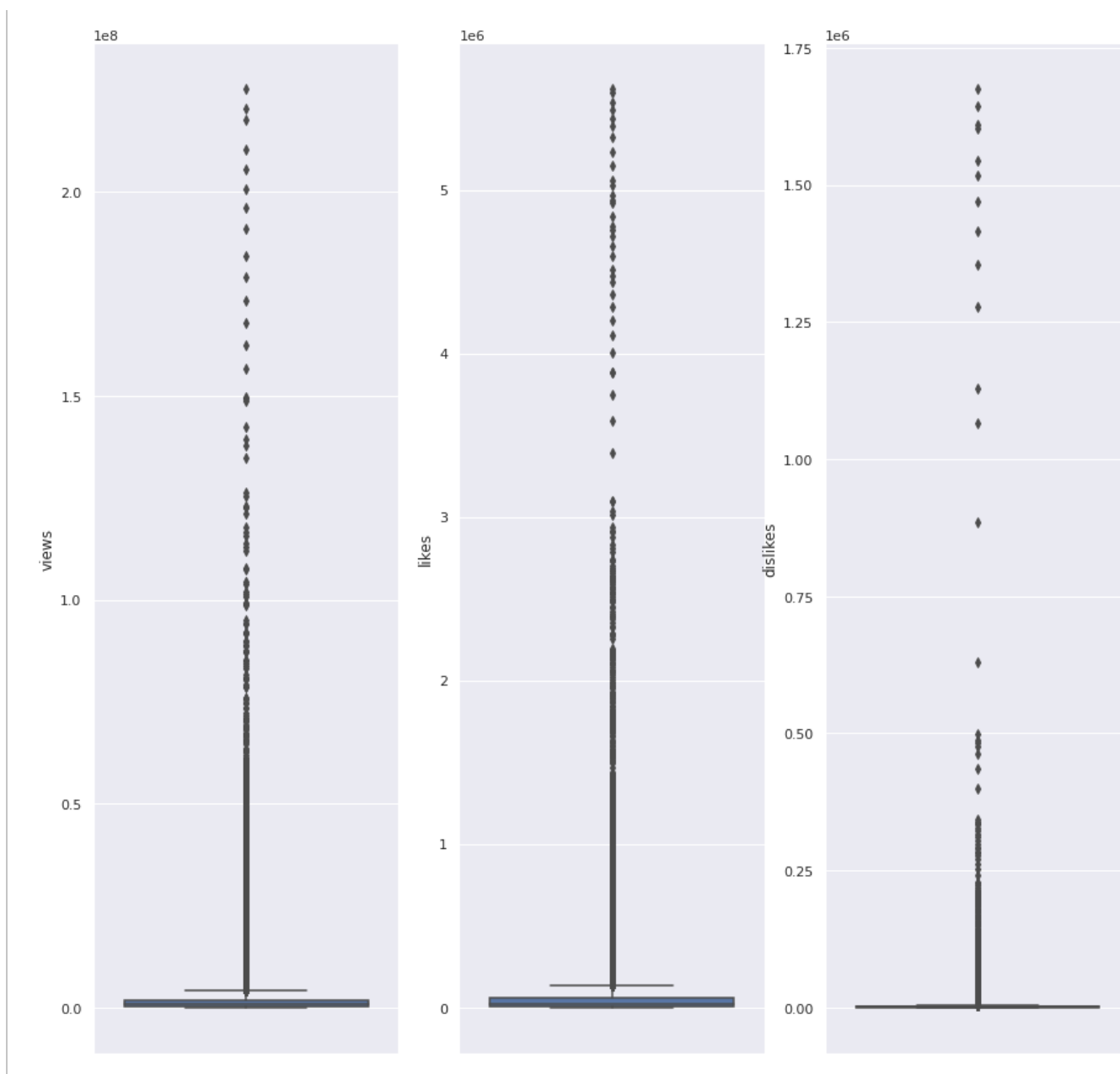
plt.show()
```



Original Data

```
fig, ax = plt.subplots(1, 3)
sns.boxplot(y=mydatav, ax=ax[0])
sns.boxplot(y=mydata1, ax=ax[1])
sns.boxplot(y=mydatad, ax=ax[2])

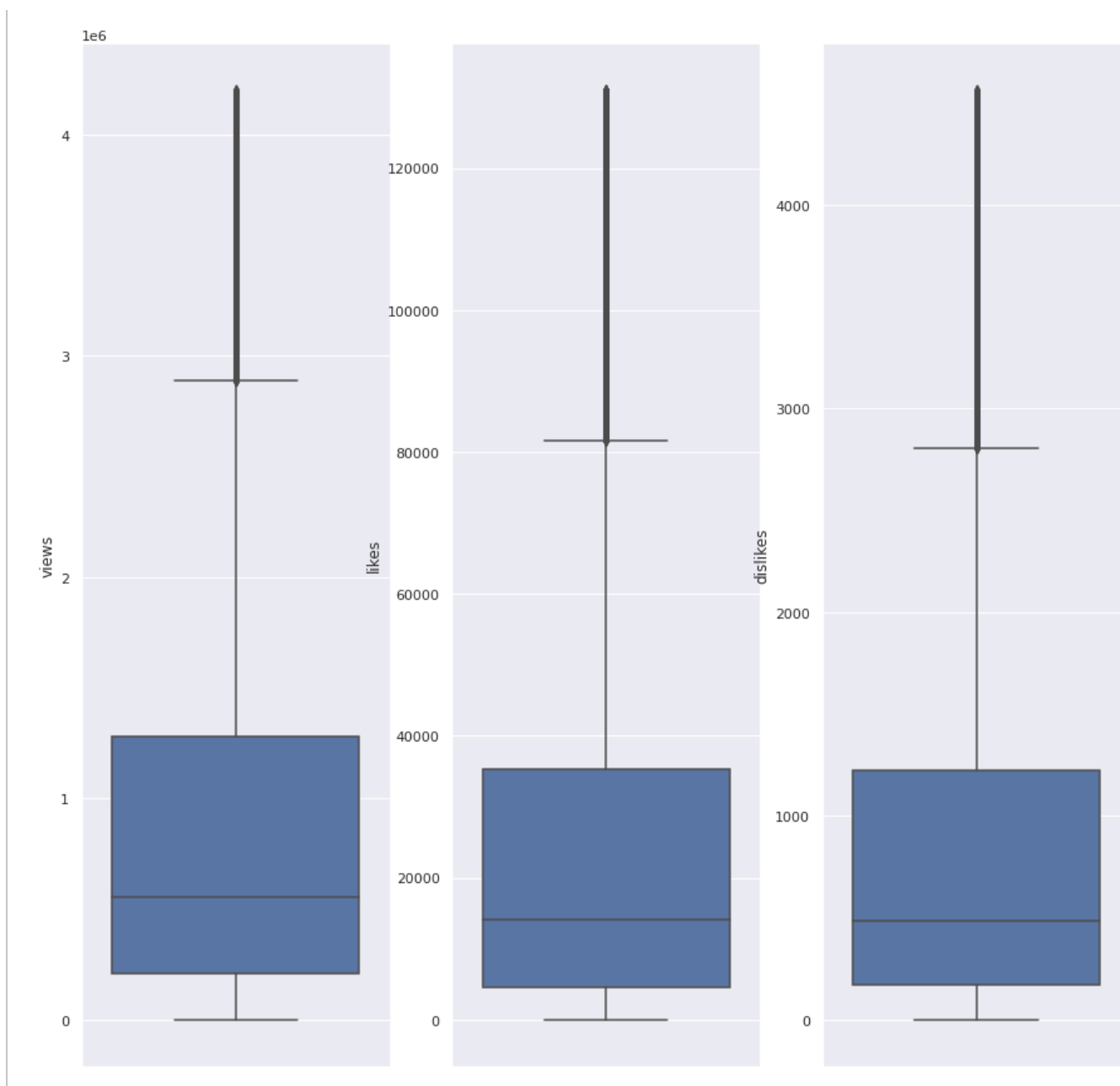
plt.show()
```



Filtered Data

```
fig, ax = plt.subplots(1, 3)
sns.boxplot(y=mydatav2, ax=ax[0])
sns.boxplot(y=mydata12, ax=ax[1])
sns.boxplot(y=mydatad2, ax=ax[2])

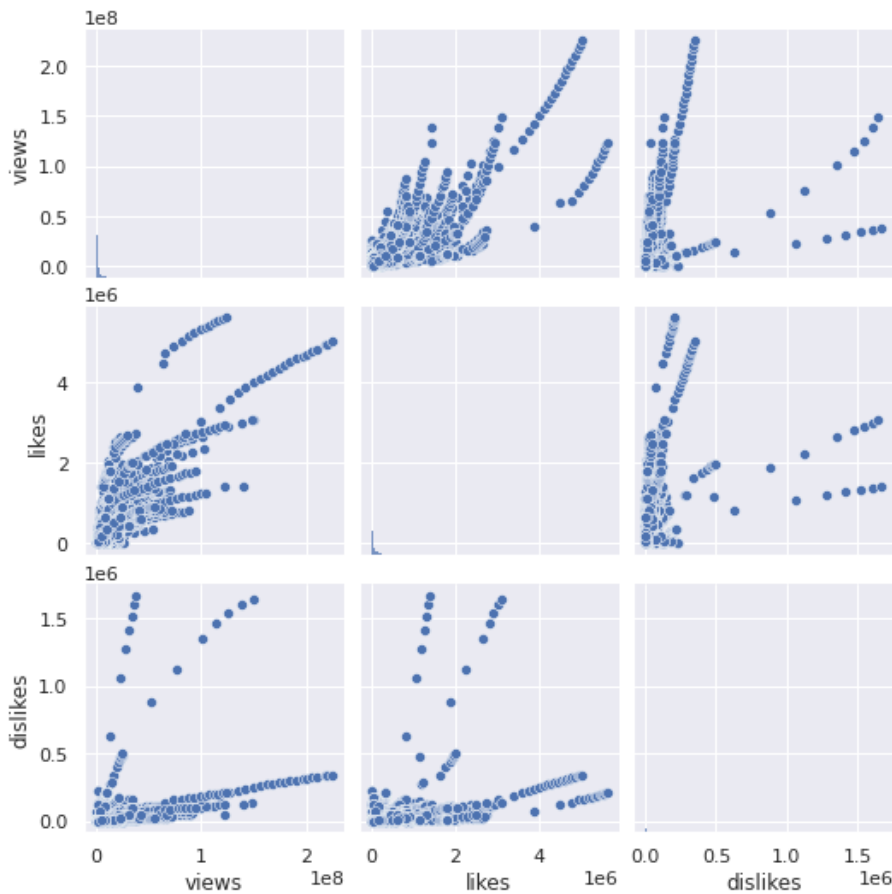
plt.show()
```



Relationship between Likes, Dislikes, Views

```
reldf = stats[["views", "likes", "dislikes"]]
```

```
sns.pairplot(reldf, height=2.5);
```



Pertanyaan diskusi

A

```
views = stats["views"]
```

```
val = views[views > 100000000].count()
```

```
n = views.size
prob_of_large_view = val / n
```

```
prob_of_large_view
```

```
0.000980872976949485
```

B

Untuk mendapatkan video favorit dengan percentil 0.1%, maka kita harus memilih persentil ke 99.9

```
res = np.percentile(stats['likes'], 99.9)
```

```
print("Minimal like yang diperlukan untuk mencapai video favorit adalah: {}".format(res))
```

Minimal like yang diperlukan untuk mencapai video favorit adalah: 2843022.2710002055

C

```
column = stats["dislikes"].copy()
column_df = column.copy()
column_df.replace(0, 1, inplace = True)
stats["L/D Ratio"] = stats["likes"].copy() / column_df
stats["(L/D)/Views Ratio"] = stats["L/D Ratio"].copy() / stats["views"].copy()
#STANDAR VIDEO VIRAL LIKES/DISLIKE = LIKES LEBIH GEDE%
#LIKES LEBIH GEDE% / VIEWS....
```

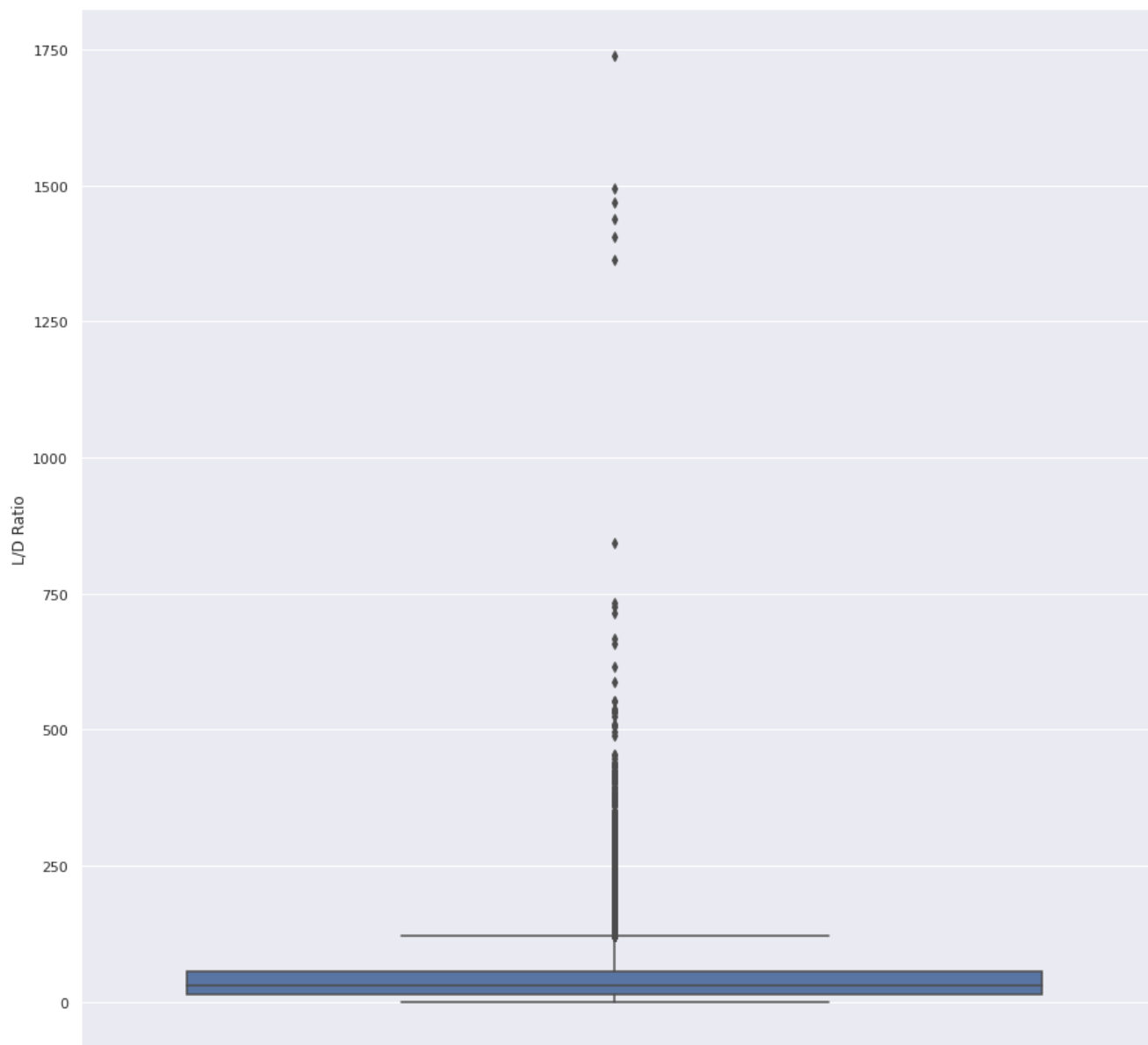
<ipython-input-166-2dda6f737815>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing
stats["L/D Ratio"] = stats["likes"].copy() / column_df

<ipython-input-166-2dda6f737815>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing
stats["(L/D)/Views Ratio"] = stats["L/D Ratio"].copy() / stats["views"].copy()

```
sns.boxplot(y=stats["L/D Ratio"])
plt.show()
```

```
stats.sort_values(by="(L/D)/Views Ratio", ascending = False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio
1126	fdpYWltgYMk	Tegan and Sara present The Con X: Covers – Flo...	5258	433	1	433.000000	8.235070e-02
12299	jXuAqchqTs8	Why are Koreans so obsessed with Bitcoin/Crypt...	6179	407	1	407.000000	6.586826e-02
16350	y00fPzC-xiA	ULTIMATE DIY electric motor using ONLY A WIRE	3240	179	1	179.000000	5.524691e-02
7933	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	33211	5212	3	1737.333333	5.231198e-02
10572	_LnZrXrdC00	Cypress FX3 as a Possible Logic Analyzer	5224	522	2	261.000000	4.996172e-02
...
5935	FIsCjmMhFmw	YouTube Rewind: The Shape of 2017 #YouTubeRe...	137843120	3014471	1602383	1.881242	1.364771e-08
6181	FIsCjmMhFmw	YouTube Rewind: The Shape of 2017 #YouTubeRe...	149376127	3093544	1643059	1.882795	1.260439e-08
1490	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2197	0	2	0.000000	0.000000e+00
1741	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2447	0	3	0.000000	0.000000e+00
192	wRGldR_SQAA	Apple Clips sample	2259	0	0	0.000000	0.000000e+00

40780 rows x 7 columns

Menurut L/D ratio

```
stats["L/D Ratio"].describe()
```

```
count    40780.000000
mean      43.850544
std       51.327325
min        0.000000
25%       13.187137
50%       29.021405
75%       56.220080
max      1737.333333
Name: L/D Ratio, dtype: float64
```

```
stats["views"].describe()
```

```
count    4.078000e+04
mean     2.353476e+06
std      7.372047e+06
min     5.490000e+02
25%     2.429242e+05
50%     6.827960e+05
75%     1.825708e+06
max     2.252119e+08
Name: views, dtype: float64
```

```
stats[stats['L/D Ratio'] < 53].sort_values(by="L/D Ratio", ascending=False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio
33525	uovntV3ZMDc	Janelle Monáe - I Like That [Official Video]	2070559	63699	1202	52.994176	2.559414e-05
27600	j1f04Jg7laA	TOP BEAUTY ESSENTIALS I CAN'T LIVE WITHOUT D...	505290	23263	439	52.990888	1.048722e-04
2373	hLglghY-FGc	I Tried to Bathe All My Pets	718343	40960	773	52.988357	7.376470e-05
19441	xYtsL9znopl	Khalid & Normani - Love Lies (Official Video)	3589995	196931	3717	52.981168	1.475801e-05
35941	D52qnC7dJcQ	Snow Patrol - What If This Is All The Love You...	1419174	28556	539	52.979592	3.733129e-05
...
7357	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1179072	9368	224152	0.041793	3.544573e-08
7116	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1142585	9100	218841	0.041583	3.639353e-08
1490	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2197	0	2	0.000000	0.000000e+00
192	wRGldR_SQAA	Apple Clips sample	2259	0	0	0.000000	0.000000e+00
1741	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2447	0	3	0.000000	0.000000e+00

```
30678 rows x 7 columns
```

```
stats.sort_values("L/D Ratio", ascending=False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio
7933	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	33211	5212	3	1737.333333	5.231198e-02
8985	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	43597	5975	4	1493.750000	3.426268e-02
8762	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	42211	5875	4	1468.750000	3.479543e-02
8552	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	40534	5755	4	1438.750000	3.549489e-02
8347	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	38629	5618	4	1404.500000	3.635869e-02
...
7357	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1179072	9368	224152	0.041793	3.544573e-08
7116	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1142585	9100	218841	0.041583	3.639353e-08
192	wRGldR_SQAA	Apple Clips sample	2259	0	0	0.000000	0.000000e+00
1490	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2197	0	2	0.000000	0.000000e+00
1741	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2447	0	3	0.000000	0.000000e+00

40780 rows x 7 columns

```
stats.sort_values(by="L/D Ratio", ascending= True)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio	L/V Ratio
1490	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2197	0	2	0.000000	0.000000e+00	0.000000
1741	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2447	0	3	0.000000	0.000000e+00	0.000000
192	wRGldR_SQAA	Apple Clips sample	2259	0	0	0.000000	0.000000e+00	0.000000
7116	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1142585	9100	218841	0.041583	3.639353e-08	0.007964
7357	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1179072	9368	224152	0.041793	3.544573e-08	0.007945
...
8347	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	38629	5618	4	1404.500000	3.635869e-02	0.145435
8552	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	40534	5755	4	1438.750000	3.549489e-02	0.141980
8762	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	42211	5875	4	1468.750000	3.479543e-02	0.139182
8985	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	43597	5975	4	1493.750000	3.426268e-02	0.137051
7933	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	33211	5212	3	1737.333333	5.231198e-02	0.156936

40780 rows x 9 columns

```
stats["L/V Ratio"] = stats["likes"] / stats["views"]
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio	L/V Ratio
0	2kyS6SvSYSE	WE WANT TO TALK ABOUT OUR MARRIAGE	748374	57527	2966	19.395482	0.000026	0.076869
1	1ZAPwfrtAFY	The Trump Presidency: Last Week Tonight with J...	2418783	97185	6146	15.812724	0.000007	0.040179
2	5qpjK5DgCt4	Racist Superman Rudy Mancuso, King Bach & Le...	3191434	146033	5339	27.352126	0.000009	0.045758
3	puqaWrEC7tY	Nickelback Lyrics: Real or Fake?	343168	10172	666	15.273273	0.000045	0.029641
4	d380meD0W0M	I Dare You: GOING BALD!?	2095731	132235	1989	66.483157	0.000032	0.063097

```
<ipython-input-182-2621eb84045f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
stats["L/V Ratio"] = stats["likes"] / stats["views"]
```

```
meanLV = stats["L/V Ratio"].mean()
print(meanLV)
```

```
0.03455559058736942
```

```
meanLD = stats["L/D Ratio"].mean()
print(meanLD)
```

```
43.850544004644185
```

```
finalstats = stats[stats["L/V Ratio"] > meanLV]
```

```
finalstats = stats[stats["L/D Ratio"] > meanLD]
```

```
finalstats.drop_duplicates(subset = "title", keep = "first", inplace = True)
```

```
<ipython-input-216-f20a65602d23>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexir
finalstats.drop_duplicates(subset = "title", keep = "first", inplace = True)
```

```
#FINAL DATAFRAME FOR QUALITY VIDEOS
finalstats.sort_values(by="L/D Ratio", ascending = False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio	L/V Ratio
7933	p1af9PKM8Eo	Jonghyun Lonely (Feat. 태연) - Piano Cover	33211	5212	3	1737.333333	0.052312	0.156936
11667	ONI_06wGbsQ	THINGS THAT ARE LOWER	109642	21080	25	843.200000	0.007690	0.192262
5147	8Jmd7-1quDM	JUST GO SHOOT. A PHOTOGRAPHY VLOG 137	60072	4671	7	667.285714	0.011108	0.077757
11577	CFwXUarN-wg	GOALS GOALS GOALS	83200	15262	26	587.000000	0.007055	0.183438
6141	w1zwGLBpULs	Interview with Clarice Lispector - São Paulo, ...	51474	552	1	552.000000	0.010724	0.010724
...
78	cOc3tsFWoRs	Jason Momoa & Lisa Bonet: Love at First Sight	1497519	15504	353	43.920680	0.000029	0.010353
29960	fNVUTgd4pio	Sugarland - Babe (Static Video) ft. Taylor Swift	559040	27423	625	43.876800	0.000078	0.049054
11171	getVCtn-4Zg	Ready To Fly!	167952	3554	81	43.876543	0.000261	0.021161
23633	PXAZ3KphJz4	Adam Rippon Savagely Ranks Olympic Sports Co...	265559	12895	294	43.860544	0.000165	0.048558
15908	WwyqQ-o6zHA	Pop Culture Typography	94248	3245	74	43.851351	0.000465	0.034430

```
lowq = stats[stats["L/V Ratio"] < meanLV]
```

```
lowq = stats[stats["L/D Ratio"] < meanLD]
```

```
lowq.drop_duplicates(subset = "title", keep = "first", inplace = True)
```

```
<ipython-input-219-f985c63103a6>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexir
lowq.drop_duplicates(subset = "title", keep = "first", inplace = True)
```

```
#dibawah rata-rata L/V Ratio
lowq.sort_values(by="L/D Ratio", ascending = False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio	L/V Ratio
39980	IdmDBJRARB4	Guillermo at 2018 NBA Media Day	1869710	40954	934	43.847966	2.345175e-05	0.021904
20591	C3_spbo8eVc	Ellie Goulding - Vincent (Audio)	150729	13242	302	43.847682	2.909041e-04	0.087853
8087	IWHZKP6xXHU	4 Lies That Ruined Christmas	1222600	57483	1311	43.846682	3.586347e-05	0.047017
15791	ipWfcK1HVdA	Jennifer Aniston Drops By to Wish Ellen a Happ...	1556517	31437	717	43.845188	2.816878e-05	0.020197
27574	zTY7dvFjBY	Don Diablo ft. Ansel Elgort - Believe Lyric ...	270593	20076	458	43.834061	1.619926e-04	0.074193
...
15347	DQDj074iEuM	LuLaRoe Consultant Mocks Those With Special Needs	212858	27	591	0.045685	2.146280e-07	0.000127
6201	8d_202I55LU	The FCC repeals its net neutrality rules	985179	4870	110707	0.043990	4.465178e-08	0.004943
7116	LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	1142585	9100	218841	0.041583	3.639353e-08	0.007964
192	wRGldR_SQAA	Apple Clips sample	2259	0	0	0.000000	0.000000e+00	0.000000
1490	A_mlvG_nRsg	Kelly Oubre Punches John Wall in the Lead duri...	2197	0	2	0.000000	0.000000e+00	0.000000

```
lowq["views"].max()
```

```
75969469
```

```
df.loc[df['views'] == 75969469]
```

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes
5020	FIsCjmMhFmw	17.09.12	YouTube Rewind: The Shape of 2017 #YouTubeRe...	YouTube Spotlight	24	2017-12-06T17:58:51.000Z	Rewind "Rewind 2017" "youtube rewind 2017" "#Y...	75969469	2251797	112779

```
#list of Videos below our filter (< L/D Ratio and < L/V Ratio)
lowq.sort_values(by="views", ascending = False)
```

	video_id	title	views	likes	dislikes	L/D Ratio	(L/D)/Views Ratio	L/V Ratio
5020	FlsCjmMhFmw	YouTube Rewind: The Shape of 2017 #YouTubeRe...	75969469	2251797	1127798	1.996631	2.628203e-08	0.029641
3818	6ZfuNTqbHE8	Marvel Studios' Avengers: Infinity War Officia...	74789251	2444952	46172	52.953132	7.080313e-07	0.032691
31877	ffxKSjUwKdU	Ariana Grande - No Tears Left To Cry	74523616	2562936	96851	26.462669	3.550911e-07	0.034391
33983	VYOjWnS4cMY	Childish Gambino - This Is America (Official V...	73432600	2478904	124290	19.944517	2.716030e-07	0.033758
11585	LsoLEjrDogU	Bruno Mars - Finesse (Remix) [Feat. Cardi B] [...]	57951412	1919583	73239	26.209847	4.522728e-07	0.033124
...
160	qg0GdM60syl	Huffy Metaloid Bicycle Commercial 1997	773	2	0	2.000000	2.587322e-03	0.002587
12716	zeQaJGkFyqQ	Raw: 3 South Carolina Deputies, 1 Officer Shot	748	9	0	9.000000	1.203209e-02	0.012032
546	-JVITToppE0	Coach Taggart Monday Presser Ahead of Arizona	687	10	2	5.000000	7.278020e-03	0.014556
14531	dQMZLXaa1L8	Artwork Forge	658	1	0	1.000000	1.519757e-03	0.001520
14335	y6KYFcta4SE	1 dead, others injured after Ky. school shooting	549	9	0	9.000000	1.639344e-02	0.016393

PCA and K-Means Cluster

```
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

```
mL_df = stats[["views", "likes", "dislikes"]]
mL_df.head()
```

	views	likes	dislikes
0	748374	57527	2966
1	2418783	97185	6146
2	3191434	146033	5339
3	343168	10172	666
4	2095731	132235	1989

```
X_std = StandardScaler().fit_transform(mL_df)
```

```
pca = PCA(n_components=1, whiten=True).fit()
```

```
sum(pca.explained_variance_ratio_)
```

```
0.7354339479666323
```

```
data = pca.transform(X_std)
```

```
model = KMeans(n_clusters=5)
model = model.fit(data)
# print(model.labels_)
plt.figure(figsize=(8, 6))
plt.scatter(data[:, 0], stats["L/D Ratio"], c=model.labels_.astype(float))
plt.show()
```

