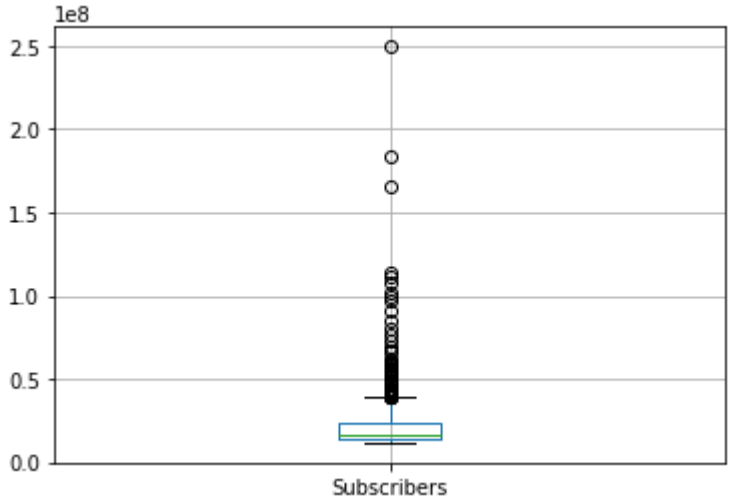
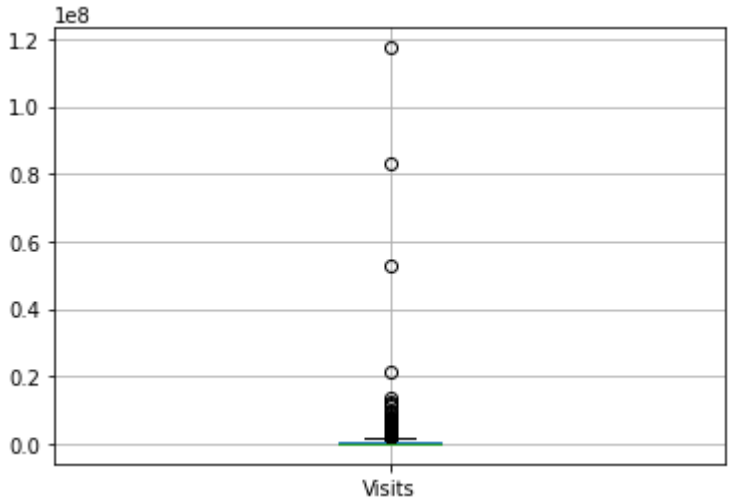
```
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: #import Libraries

In [2]: import numpy as np

In [3]: import pandas as pd

In [4]: import matplotlib.pyplot as plt

In [5]: #Read csv file

In [6]: df=pd.read_csv("C:/Users/Dell/Desktop/Excel/youtubers_df.csv")

In [7]: #Data Understanding

In [8]: print(df.shape)
(1000, 9)

In [9]: print(df.size)
9000

In [10]: df.describe()
Out[10]:
              Rank   Subscribers        Visits         Likes       Comments
count  1000.000000  1.000000e+03  1.000000e+03  1.000000e+03    1000.000000
mean    500.500000  2.189440e+07  1.209446e+06  5.363259e+04    1288.768000
std     288.819436  1.682775e+07  5.229942e+06  2.580457e+05    6778.188308
min       1.000000  1.170000e+07  0.000000e+00  0.000000e+00       0.000000
25%     250.750000  1.380000e+07  3.197500e+04  4.717500e+02       2.000000
50%     500.500000  1.675000e+07  1.744500e+05  3.500000e+03      67.000000
75%     750.250000  2.370000e+07  8.654750e+05  2.865000e+04     472.000000
max    1000.000000  2.495000e+08  1.174000e+08  5.300000e+06  154000.000000

In [11]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Rank         1000 non-null   int64
 1   Username     1000 non-null   object
 2   Categories   694 non-null    object
 3   Subscribers  1000 non-null   int64
 4   Country      1000 non-null   object
 5   Visits       1000 non-null   float64
 6   Likes        1000 non-null   int64
 7   Comments     1000 non-null   int64
 8   Links        1000 non-null   object
dtypes: float64(1), int64(4), object(4)
memory usage: 70.4+ KB

In [12]: df.head()
Out[12]:
   Rank  ...                                              Links
0     1  ...  http://youtube.com/channel/UCq-Fj5jknLsUf-MWSy...
1     2  ...  http://youtube.com/channel/UCX6OQ3DkcsbYNE6H8u...
2     3  ...  http://youtube.com/channel/UCbCmjCuTUZos6Inko4...
3     4  ...  http://youtube.com/channel/UCpEhnqL0y41EpW2TvW...
4     5  ...  http://youtube.com/channel/UCk8GzjMOrta8yxDcKf...

[5 rows x 9 columns]

In [13]: df.tail()
Out[13]:
```

```
     Rank  ...                                              Links
995   996  ...  http://youtube.com/channel/UCPKNKldggioffXPkSm...
996   997  ...  http://youtube.com/channel/UCk3fFpqI5kDMf__mUP...
997   998  ...  http://youtube.com/channel/UCdrHrQf0o0TO8YDntX...
998   999  ...  http://youtube.com/channel/UCObyBrdrtQ20BU9PxH...
999  1000  ...  http://youtube.com/channel/UCOjgc1p2hJ4GZi6pQQ...

[5 rows x 9 columns]

In [14]: df["Categories"].info()
<class 'pandas.core.series.Series'>
RangeIndex: 1000 entries, 0 to 999
Series name: Categories
Non-Null Count  Dtype
--------------  -----
694 non-null    object
dtypes: object(1)
memory usage: 7.9+ KB

In [15]: df["Country"].info()
<class 'pandas.core.series.Series'>
RangeIndex: 1000 entries, 0 to 999
Series name: Country
Non-Null Count  Dtype
--------------  -----
1000 non-null   object
dtypes: object(1)
memory usage: 7.9+ KB

In [16]: df["Likes"].info()
<class 'pandas.core.series.Series'>
RangeIndex: 1000 entries, 0 to 999
Series name: Likes
Non-Null Count  Dtype
--------------  -----
1000 non-null   int64
dtypes: int64(1)
memory usage: 7.9 KB

In [17]: df["Comments"].info()
<class 'pandas.core.series.Series'>
RangeIndex: 1000 entries, 0 to 999
Series name: Comments
Non-Null Count  Dtype
--------------  -----
1000 non-null   int64
dtypes: int64(1)
memory usage: 7.9 KB

In [18]: #Data preprocessing

In [19]: #Missing value

In [20]: df.dropna(subset='Categories',inplace=True)
    ...: df.describe()
Out[20]:
              Rank   Subscribers        Visits         Likes       Comments
count   694.000000  6.940000e+02  6.940000e+02  6.940000e+02     694.000000
mean    495.298271  2.241556e+07  1.210730e+06  5.347360e+04    1558.793948
std     289.222212  1.824123e+07  6.038274e+06  2.979711e+05    7967.470234
min       1.000000  1.170000e+07  0.000000e+00  0.000000e+00       0.000000
25%     244.250000  1.380000e+07  3.692500e+04  5.685000e+02       2.000000
50%     492.500000  1.680000e+07  1.587000e+05  3.550000e+03      78.000000
75%     746.750000  2.390000e+07  8.339000e+05  2.377500e+04     499.750000
max    1000.000000  2.495000e+08  1.174000e+08  5.300000e+06  154000.000000

In [21]: #Outlier detection

In [22]: df.boxplot(['Subscribers'])
```

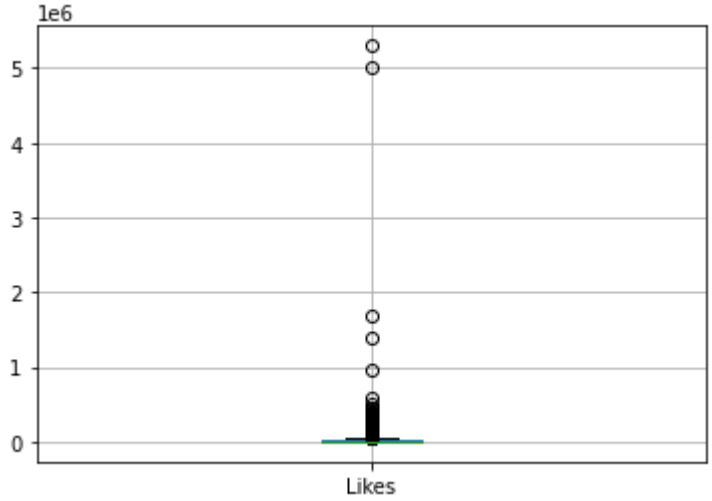
```
Out[22]: <Axes: >
```



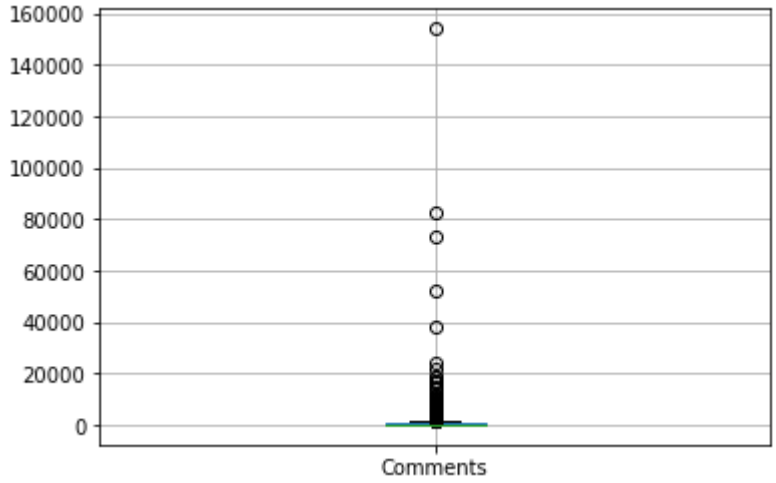

```
In [23]: Q1=df['Subscribers'].quantile(.25)
    ...: Q3=df['Subscribers'].quantile(.75)
    ...: print(Q1)
    ...: print(Q3)
13800000.0
23900000.0

In [24]: Quartile_Deviation=(Q3-Q1)/2
    ...: print(Quartile_Deviation)
5050000.0

In [25]: df.boxplot(['Visits'])
Out[25]: <Axes: >
```




```
In [26]: Q1=df['Visits'].quantile(.25)
    ...: Q3=df['Visits'].quantile(.75)
    ...: print(Q1)
    ...: print(Q3)
36925.0
833900.0

In [27]: Quartile_Deviation=(Q3-Q1)/2
    ...: print(Quartile_Deviation)
398487.5

In [28]: df.boxplot(['Likes'])
Out[28]: <Axes: >
```

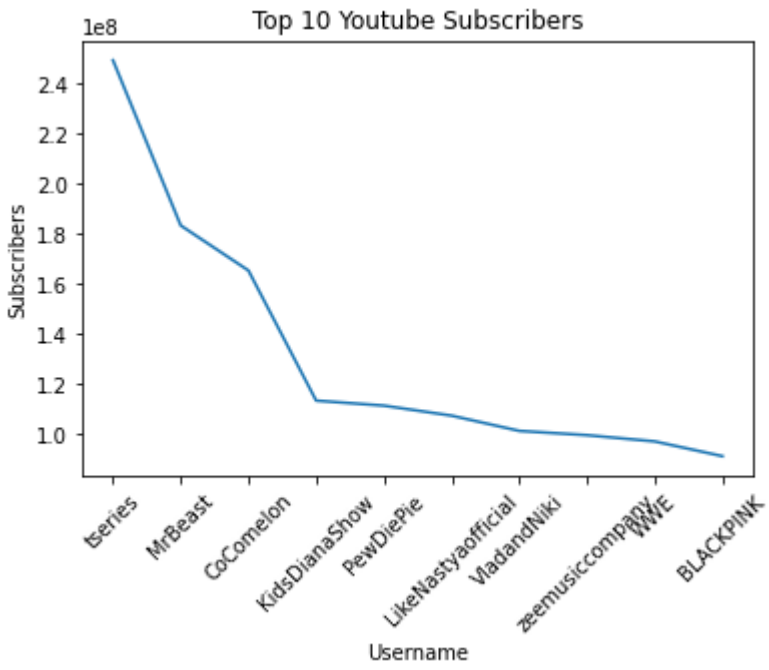



```
In [29]: Q1=df['Likes'].quantile(.25)
    ...: Q3=df['Likes'].quantile(.75)
    ...: print(Q1)
    ...: print(Q3)
568.5
23775.0

In [30]: Quartile_Deviation=(Q3-Q1)/2
    ...: print(Quartile_Deviation)
11603.25

In [31]: df.boxplot(['Comments'])
Out[31]: <Axes: >
```

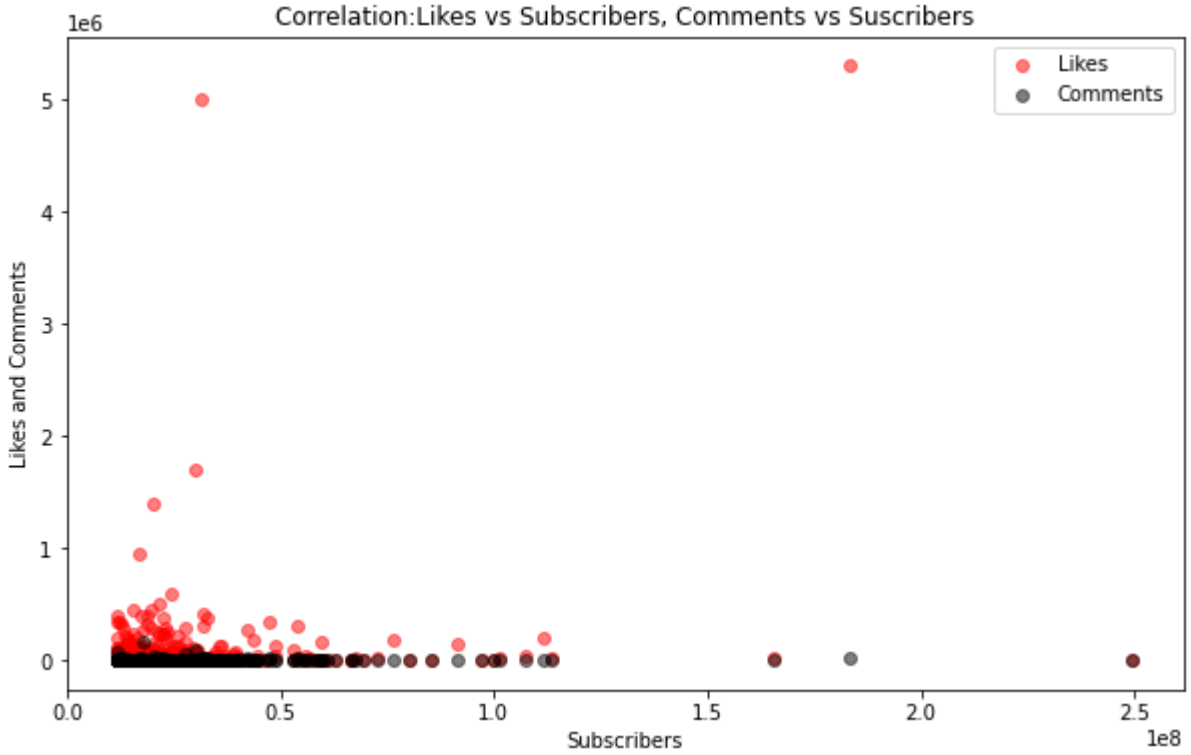



```
In [32]: Q1=df['Comments'].quantile(.25)
    ...: Q3=df['Comments'].quantile(.75)
    ...: print(Q1)
    ...: print(Q3)
2.0
499.75

In [33]: Quartile_Deviation=(Q3-Q1)/2
    ...: print(Quartile_Deviation)
248.875

In [34]: #Trend Analysis

In [35]: username=df["Username"].head(10)
    ...: subscribers=df["Subscribers"].head(10)
    ...: plt.plot(username,subscribers)
    ...: plt.xlabel("Username")
    ...: plt.ylabel("Subscribers")
    ...: plt.title('Top 10 Youtube Subscribers')
    ...: plt.xticks(rotation=45)
```

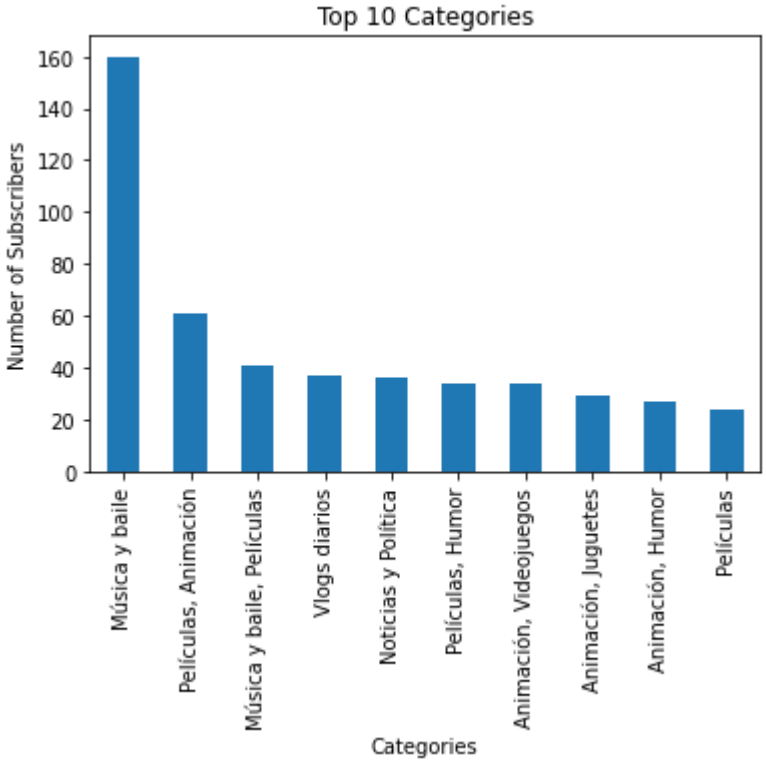
```
    ...: plt.show()
```



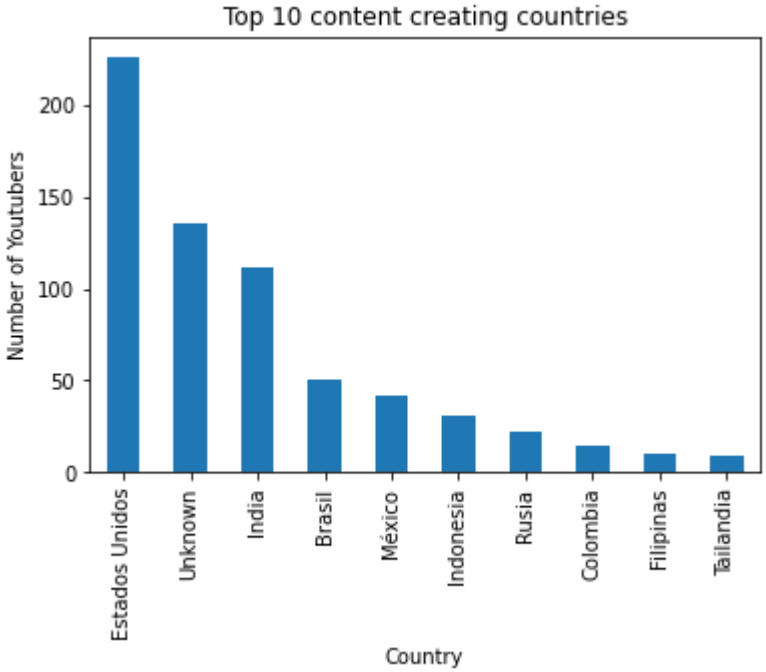

```
In [36]: correlation_likes=df['Subscribers'].corr(df['Likes'])
    ...: correlation_likes
Out[36]: 0.24838887155409808

In [37]: correlation_visits=df['Subscribers'].corr(df['Visits'])
    ...: correlation_visits
Out[37]: 0.2860387866913464

In [38]: correlation_Comments=df['Subscribers'].corr(df['Comments'])
    ...: correlation_Comments
Out[38]: 0.03729270392381571

In [39]: plt.figure(figsize=(10,6))
    ...: plt.scatter(df['Subscribers'],df['Likes'],alpha=0.5,label='Likes',color='Red')
    ...: plt.scatter(df['Subscribers'],df['Comments'],alpha=0.5,label='Comments',color='Black')
    ...: plt.xlabel('Subscribers')
    ...: plt.ylabel('Likes and Comments')
    ...: plt.legend()
    ...: plt.title('Correlation:Likes vs Subscribers, Comments vs Suscribers')
    ...: plt.show()
```




```
In [40]: #Performance Metrics

In [41]: average_suscribers=df['Subscribers'].mean()
    ...: print(average_suscribers)
22415561.95965418

In [42]: average_likes=df['Likes'].mean()
    ...: print(average_likes)
53473.59798270893

In [43]: average_visits=df['Visits'].mean()
    ...: print(average_visits)
1210729.6829971182

In [44]: average_comments=df['Comments'].mean()
    ...: print(average_comments)
1558.793948126801

In [45]: #Content Categories

In [46]: Top_categories=df["Categories"].value_counts().head(10)
    ...: Top_categories
    ...: Top_categories.plot(kind='bar')
    ...: plt.xlabel("Categories")
    ...: plt.ylabel("Number of Subscribers")
    ...: plt.title("Top 10 Categories")
    ...: plt.show()
```




```
In [47]: #Top 10 Content creating countries

In [48]: country_youtubers=df["Country"].value_counts().head(10)
    ...: country_youtubers
Out[48]:
Country
Estados Unidos    226
Unknown           136
India             112
Brasil             51
México             42
Indonesia          31
Rusia              22
Colombia           14
Filipinas          10
Tailandia           9
Name: count, dtype: int64

In [49]: country_youtubers.plot(kind="bar")
    ...: plt.xlabel("Country")
    ...: plt.ylabel("Number of Youtubers")
    ...: plt.title("Top 10 content creating countries")
    ...: plt.show()
```


Top 10 content creating countries
Number of Youtubers
0
50
100
150
200
Estados Unidos
Unknown
India
Brasil
México
Indonesia
Rusia
Colombia
Filipinas
Tailandia
Country

In [50]: