Sardar Patel Institute of Technology,Mumbai
Department of Electronics and Telecommunication Engineering
B.E. Sem-VII (2021-2022)
Data Analytics

Experiment: Exploratory Data Analysis (EDA)

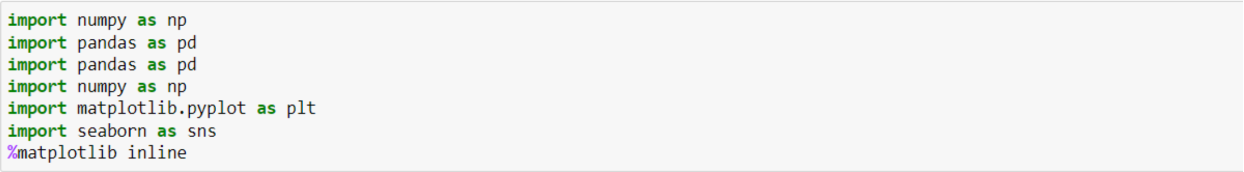
Name: Sneha Ghuge **UID:** 2019110015 **BE ETRX** **DA LAB 1**

Aim: Perform Exploratory Data Analysis (EDA) on 100 Top Youtube Channels data.

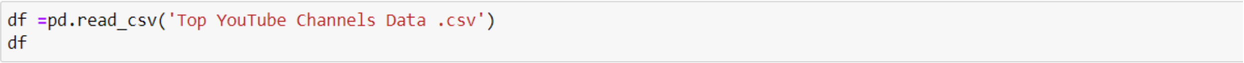
Dataset Overview

The dataset 'Top100 youtube channel' contains 7 columns :

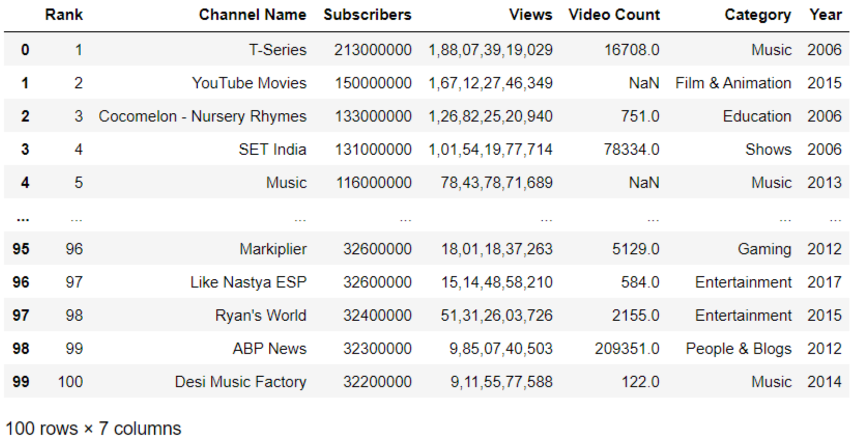
- Rank : Rank of the channel as per number of subscribers they have
- Channel Name : Channel official name or name of the YouTuber
- Subscribers : Number of subscribers
- Views : Total views of video
- Video Count : Number of videos channel has uploaded so far
- Category : Category (genre) of the channel
- Year : Year when the channel was started



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



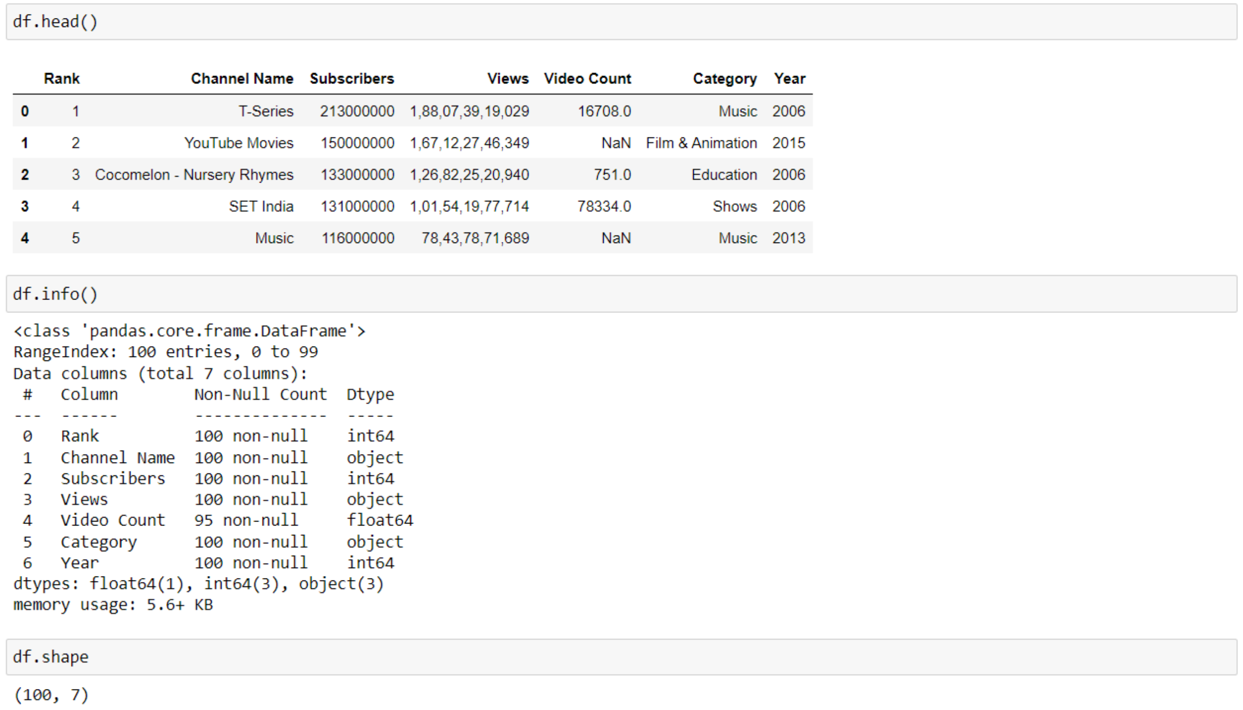
```
df =pd.read_csv('Top YouTube Channels Data .csv')
df
```



	Rank	Channel Name	Subscribers	Views	Video Count	Category	Year
0	1	T-Series	213000000	1,88,07,39,19,029	16708.0	Music	2006
1	2	YouTube Movies	150000000	1,67,12,27,46,349	NaN	Film & Animation	2015
2	3	Cocomelon - Nursery Rhymes	133000000	1,26,82,25,20,940	751.0	Education	2006
3	4	SET India	131000000	1,01,54,19,77,714	78334.0	Shows	2006
4	5	Music	116000000	78,43,78,71,689	NaN	Music	2013
...	...	...	...	...	...	...	...
95	96	Markiplier	32600000	18,01,18,37,263	5129.0	Gaming	2012
96	97	Like Nastya ESP	32600000	15,14,48,58,210	584.0	Entertainment	2017
97	98	Ryan's World	32400000	51,31,26,03,726	2155.0	Entertainment	2015
98	99	ABP News	32300000	9,85,07,40,503	209351.0	People & Blogs	2012
99	100	Desi Music Factory	32200000	9,11,55,77,588	122.0	Music	2014

100 rows × 7 columns

Successfully imported the necessary libraries and the dataset into the notebook



```
df.head()
```

	Rank	Channel Name	Subscribers	Views	Video Count	Category	Year
0	1	T-Series	213000000	1,88,07,39,19,029	16708.0	Music	2006
1	2	YouTube Movies	150000000	1,67,12,27,46,349	NaN	Film & Animation	2015
2	3	Cocomelon - Nursery Rhymes	133000000	1,26,82,25,20,940	751.0	Education	2006
3	4	SET India	131000000	1,01,54,19,77,714	78334.0	Shows	2006
4	5	Music	116000000	78,43,78,71,689	NaN	Music	2013

```
df.info()
```

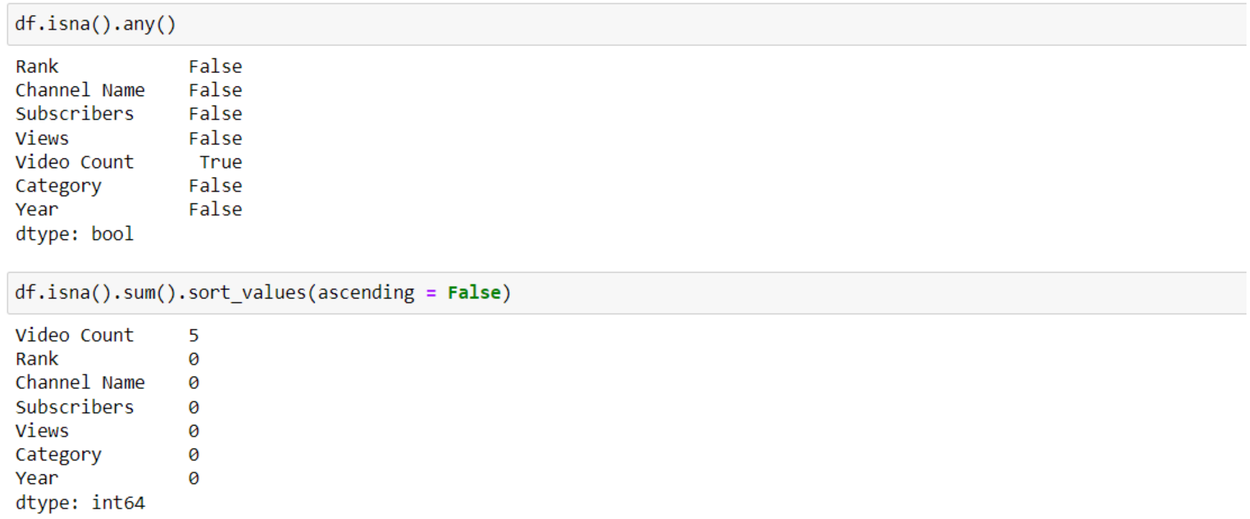
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Rank           100 non-null    int64  
 1   Channel Name   100 non-null    object 
 2   Subscribers    100 non-null    int64  
 3   Views          100 non-null    object 
 4   Video Count    95 non-null     float64
 5   Category       100 non-null    object 
 6   Year           100 non-null    int64  
dtypes: float64(1), int64(3), object(3)
memory usage: 5.6+ KB
```

```
df.shape
```

```
(100, 7)
```

The dataset has 100 rows and 7 columns to work with for EDA.

Next, We will explore numbers of NULL values or missing values the dataset has.



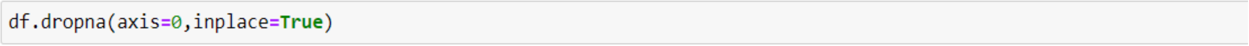
```
df.isna().any()
```

```
Rank            False
Channel Name    False
Subscribers     False
Views           False
Video Count      True
Category        False
Year            False
dtype: bool
```

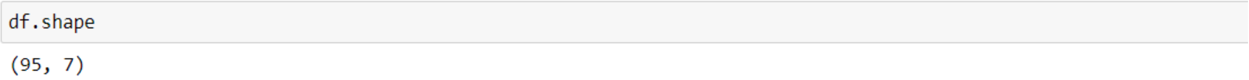
```
df.isna().sum().sort_values(ascending = False)
```

```
Video Count     5
Rank            0
Channel Name    0
Subscribers     0
Views           0
Category        0
Year            0
dtype: int64
```

We see that the column Video Count has 5 null values lets drop those values.



```
df.dropna(axis=0,inplace=True)
```

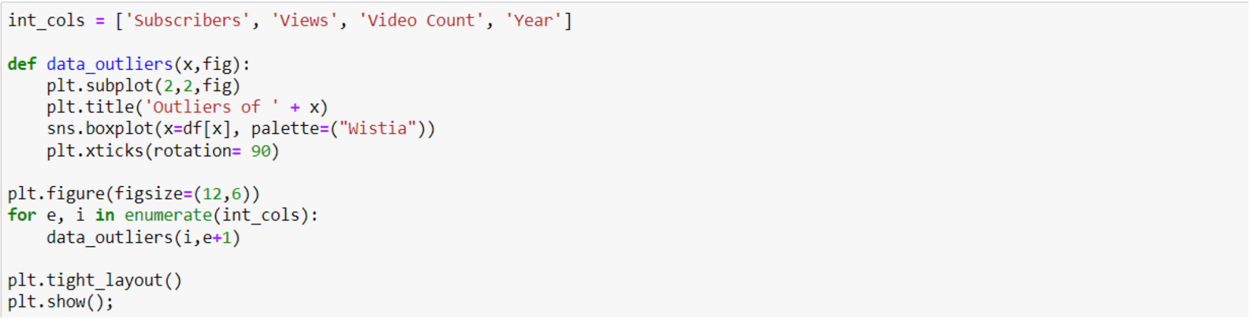


```
df.shape
```

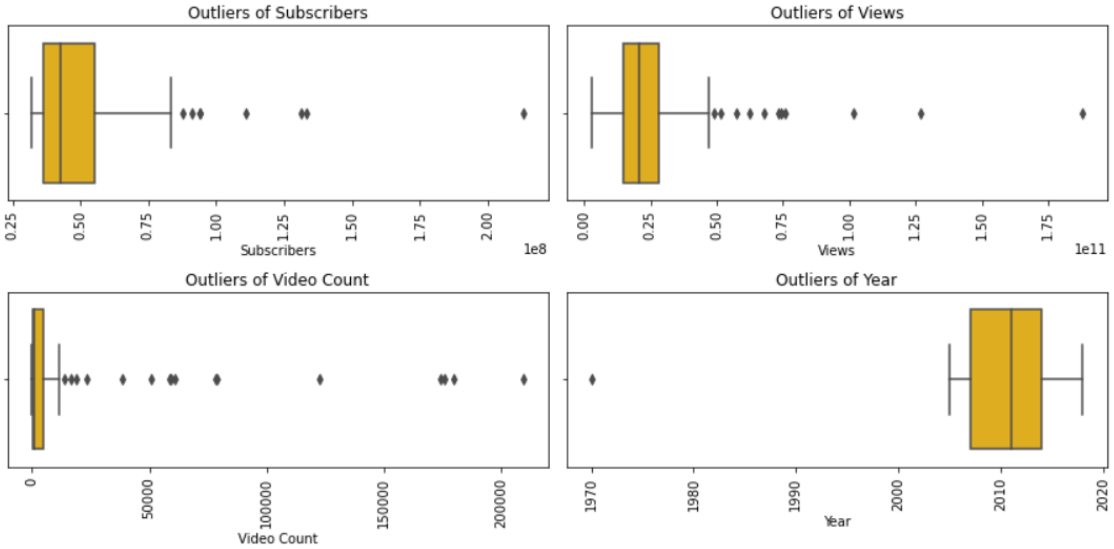
```
(95, 7)
```

After dropping the rows with missing values , the dataset has 95 rows and 7 columns to work upon.

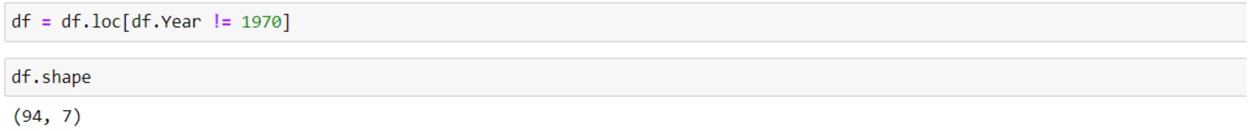
Now, Let us check for outliers



```
int_cols = ['Subscribers', 'Views', 'Video Count', 'Year']

def data_outliers(x,fig):
    plt.subplot(2,2,fig)
    plt.title('Outliers of ' + x)
    sns.boxplot(x=df[x], palette=("Wistia"))
    plt.xticks(rotation= 90)

plt.figure(figsize=(12,6))
for e, i in enumerate(int_cols):
    data_outliers(i,e+1)

plt.tight_layout()
plt.show();
```



Here a channel was started in 1970 . We will drop that value .



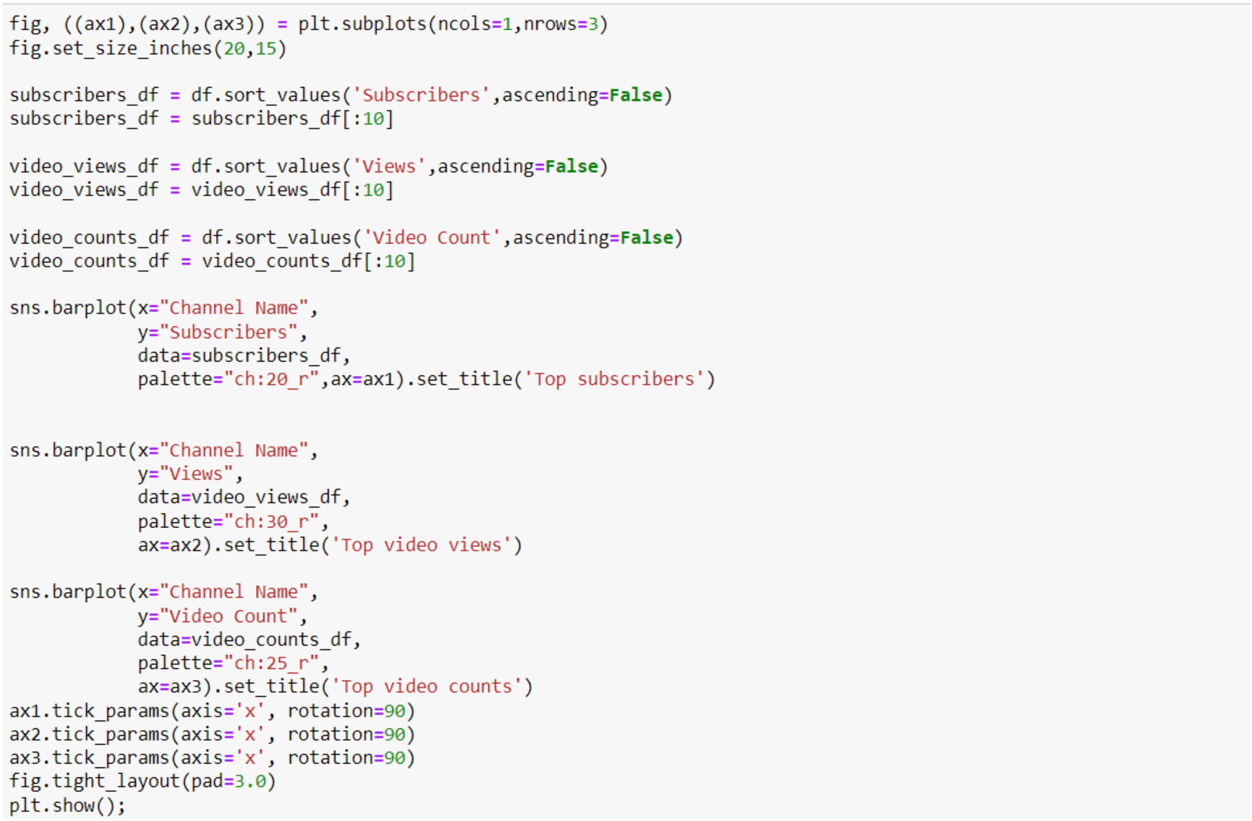
```
df = df.loc[df.Year != 1970]
```

```
df.shape
```

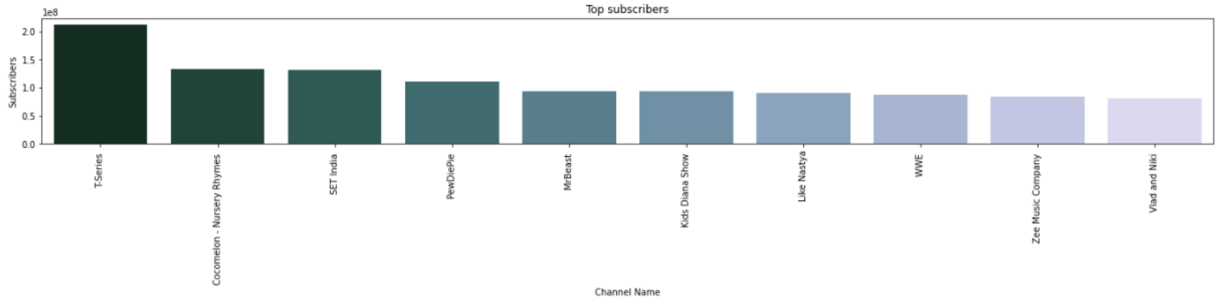
```
(94, 7)
```

Now lets plot Top 10 YouTubers with respect to the following data columns:

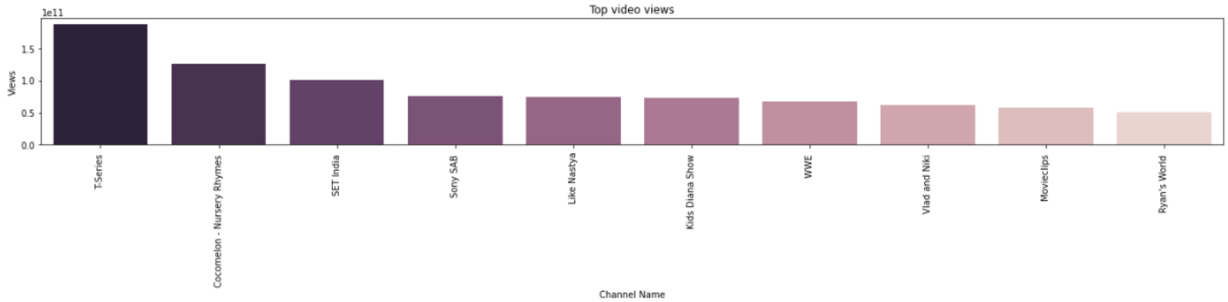
- Subscriber
- Video Views
- Video Counts



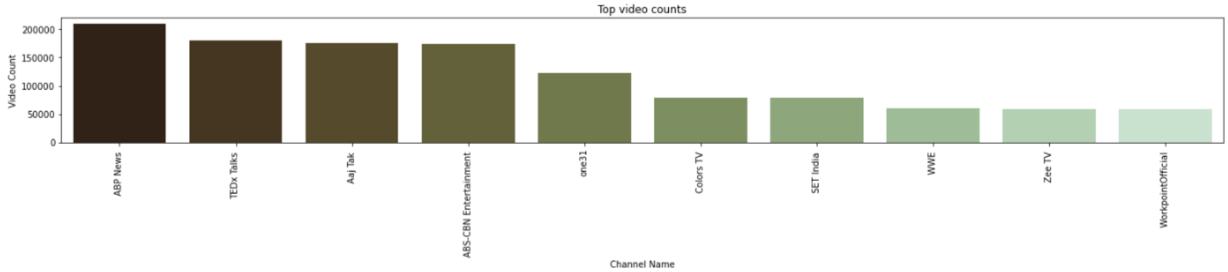
```
fig, ((ax1),(ax2),(ax3)) = plt.subplots(ncols=1,nrows=3)
fig.set_size_inches(20,15)

subscribers_df = df.sort_values('Subscribers',ascending=False)
subscribers_df = subscribers_df[:10]

video_views_df = df.sort_values('Views',ascending=False)
video_views_df = video_views_df[:10]

video_counts_df = df.sort_values('Video Count',ascending=False)
video_counts_df = video_counts_df[:10]

sns.barplot(x="Channel Name",
            y="Subscribers",
            data=subscribers_df,
            palette="ch:20_r",ax=ax1).set_title('Top subscribers')


sns.barplot(x="Channel Name",
            y="Views",
            data=video_views_df,
            palette="ch:30_r",
            ax=ax2).set_title('Top video views')

sns.barplot(x="Channel Name",
            y="Video Count",
            data=video_counts_df,
            palette="ch:25_r",
            ax=ax3).set_title('Top video counts')
ax1.tick_params(axis='x', rotation=90)
ax2.tick_params(axis='x', rotation=90)
ax3.tick_params(axis='x', rotation=90)
fig.tight_layout(pad=3.0)
plt.show();
```



- T-Series having highest Subscriber

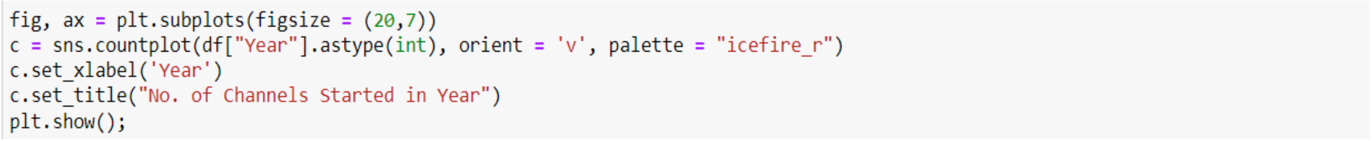


- T-Series having maximum View Views

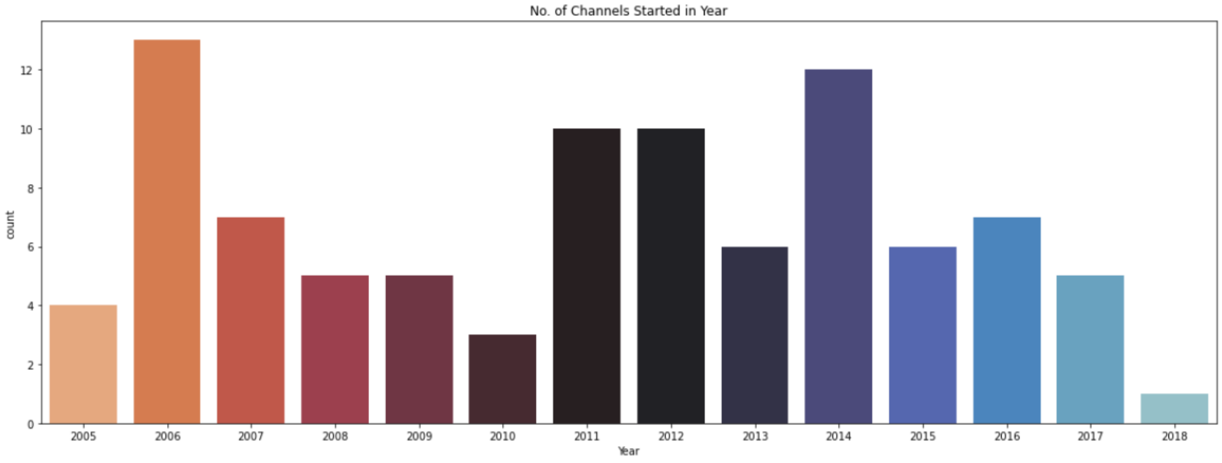


- ABP News having maximum Video Count

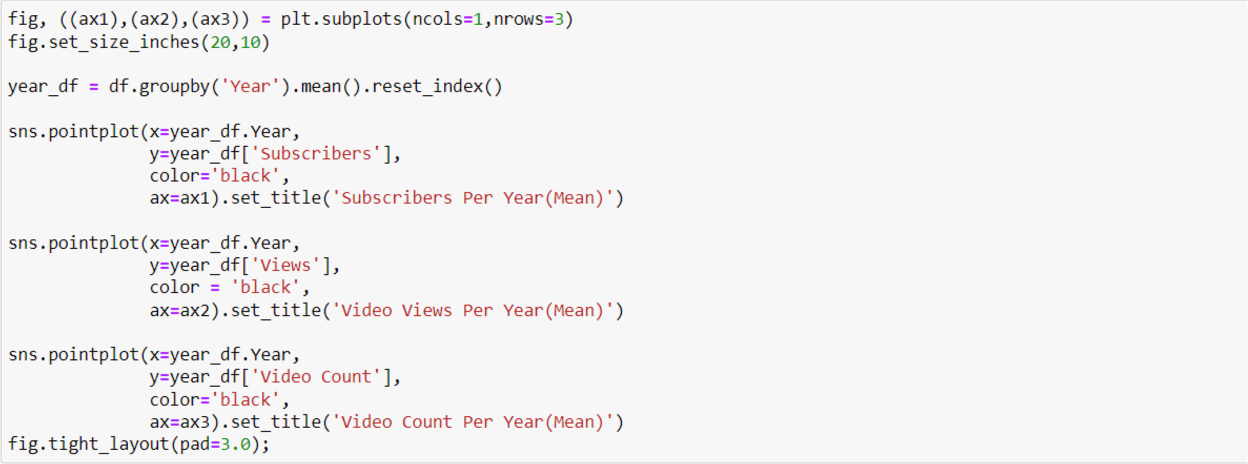
Started Year Analysis:



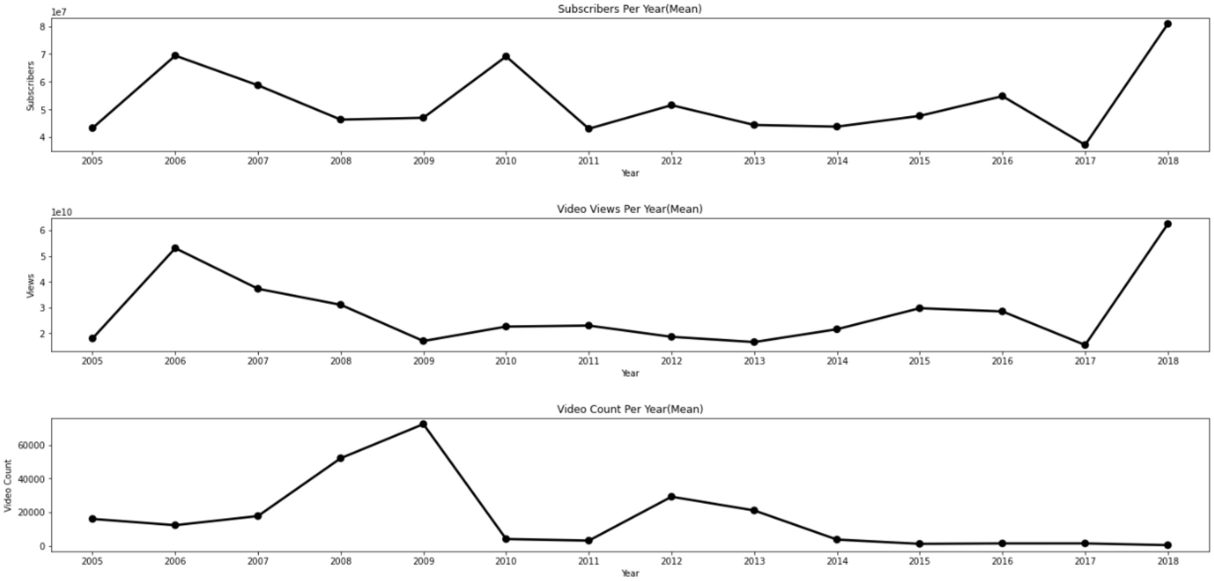
```
fig, ax = plt.subplots(figsize = (20,7))
c = sns.countplot(df["Year"].astype(int), orient = 'v', palette = "icefire_r")
c.set_xlabel('Year')
c.set_title("No. of Channels Started in Year")
plt.show();
```



- Most of the channels started in the years 2006 and 2014



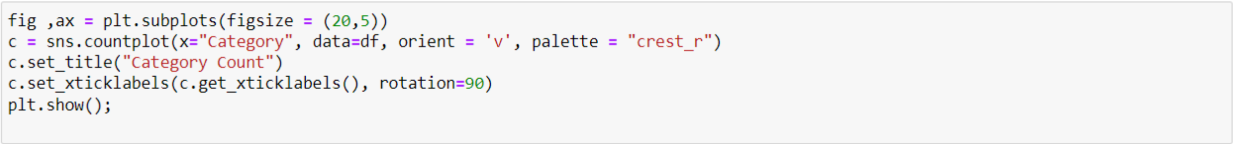
```
fig, ((ax1),(ax2),(ax3)) = plt.subplots(ncols=1,nrows=3)
fig.set_size_inches(20,10)

year_df = df.groupby('Year').mean().reset_index()

sns.pointplot(x=year_df.Year,
              y=year_df['Subscribers'],
              color='black',
              ax=ax1).set_title('Subscribers Per Year(Mean)')

sns.pointplot(x=year_df.Year,
              y=year_df['Views'],
              color = 'black',
              ax=ax2).set_title('Video Views Per Year(Mean)')

sns.pointplot(x=year_df.Year,
              y=year_df['Video Count'],
              color='black',
              ax=ax3).set_title('Video Count Per Year(Mean)')
fig.tight_layout(pad=3.0);
```



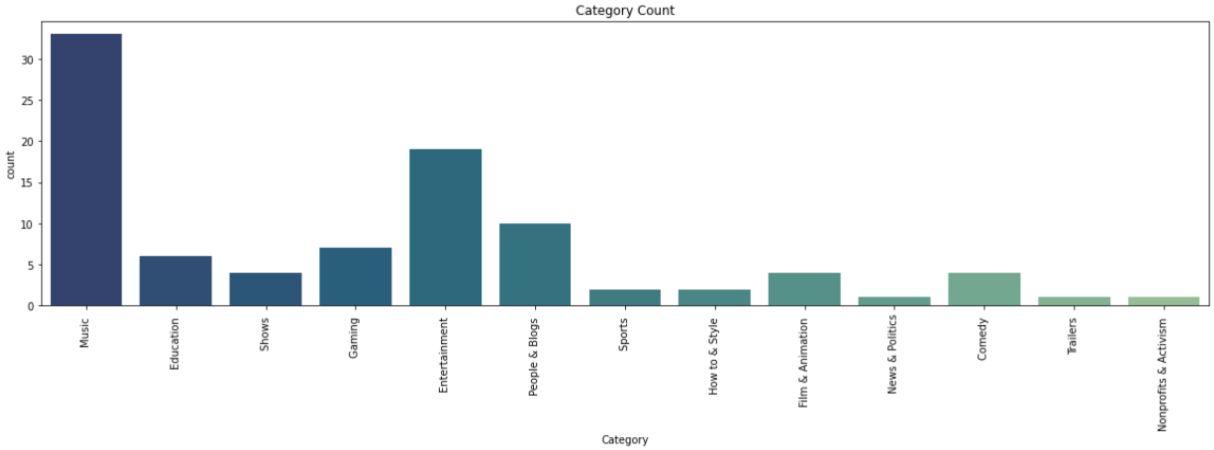
Above we see the mean Values of Subscribers, Views and View Counts with respect to Year.

Now, Let's try to sort the top channels based on the category the channel falls under.

Category Analysis



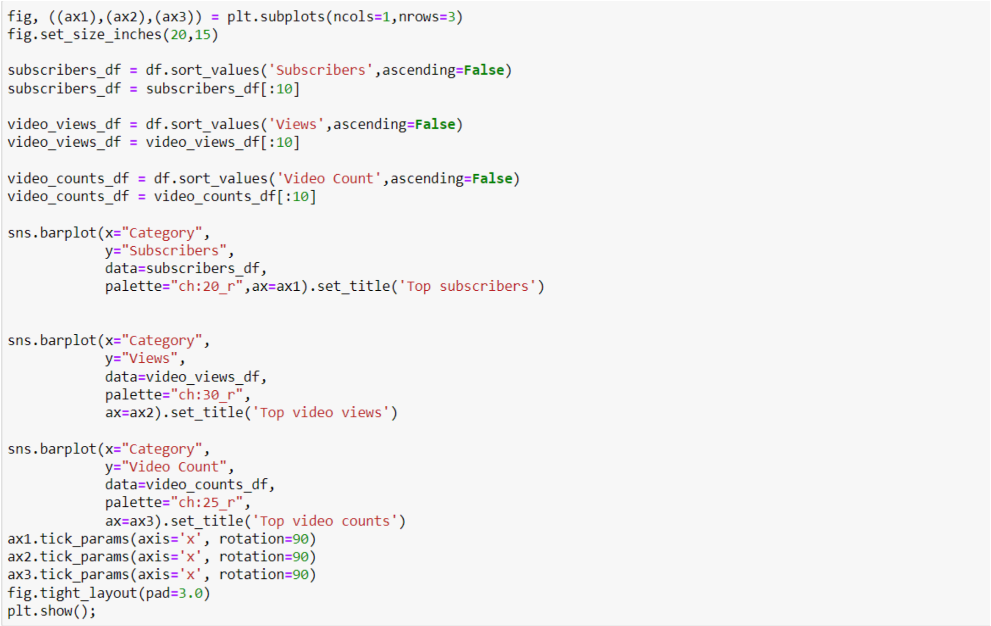
```
fig ,ax = plt.subplots(figsize = (20,5))
c = sns.countplot(x="Category", data=df, orient = 'v', palette = "crest_r")
c.set_title("Category Count")
c.set_xticklabels(c.get_xticklabels(), rotation=90)
plt.show();
```



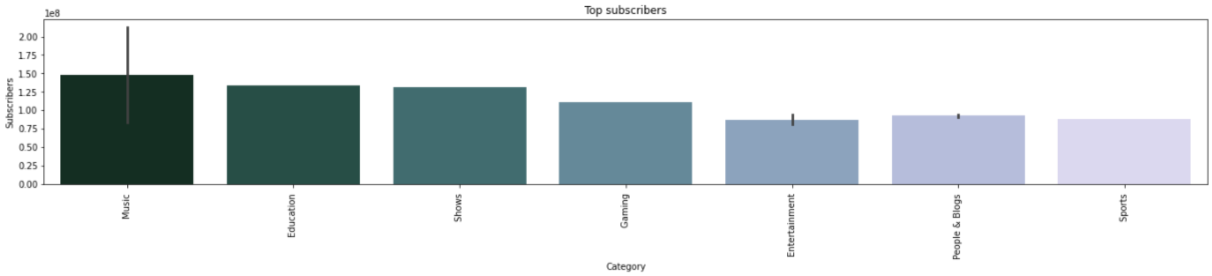
From the above figure we see that Music is the most popular category.

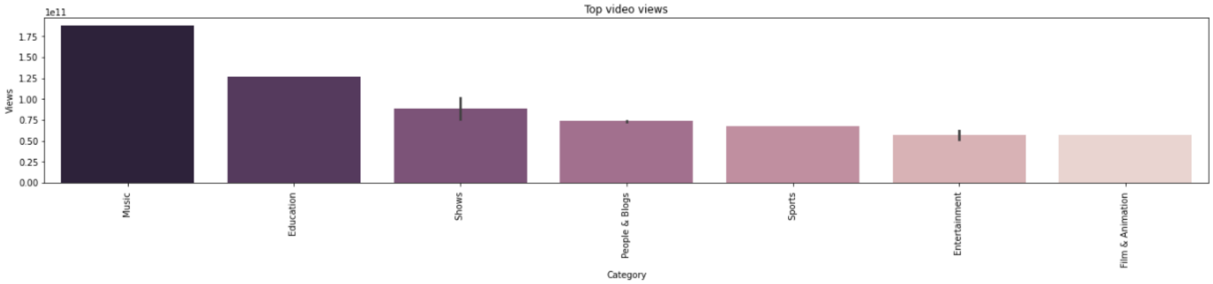
Top 10 Categories with respect to:

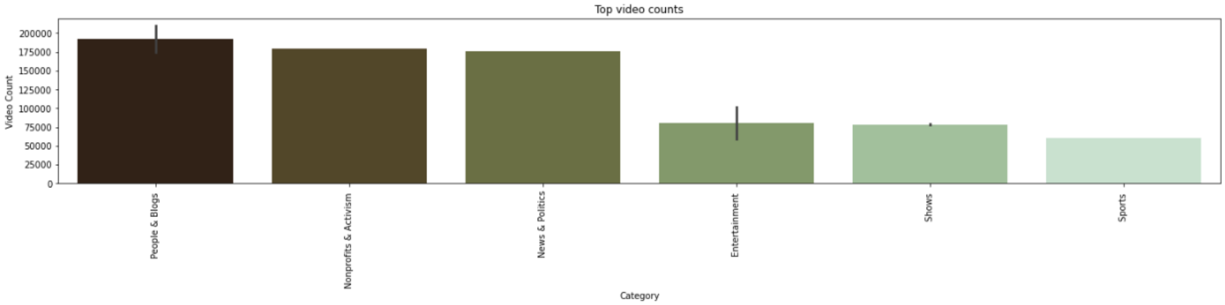
- Subscriber
- Video Views
- Video Count



```
fig, ((ax1),(ax2),(ax3)) = plt.subplots(ncols=1,nrows=3)
fig.set_size_inches(20,15)

subscribers_df = df.sort_values('Subscribers',ascending=False)
subscribers_df = subscribers_df[:10]

video_views_df = df.sort_values('Views',ascending=False)
video_views_df = video_views_df[:10]

video_counts_df = df.sort_values('Video Count',ascending=False)
video_counts_df = video_counts_df[:10]

sns.barplot(x="Category",
            y="Subscribers",
            data=subscribers_df,
            palette="ch:20_r",ax=ax1).set_title('Top subscribers')


sns.barplot(x="Category",
            y="Views",
            data=video_views_df,
            palette="ch:30_r",
            ax=ax2).set_title('Top video views')

sns.barplot(x="Category",
            y="Video Count",
            data=video_counts_df,
            palette="ch:25_r",
            ax=ax3).set_title('Top video counts')
ax1.tick_params(axis='x', rotation=90)
ax2.tick_params(axis='x', rotation=90)
ax3.tick_params(axis='x', rotation=90)
fig.tight_layout(pad=3.0)
plt.show();
```







Conclusion:

1. Performed EDA for Top 100 Youtube Channels dataset.
2. Few insights we found from the dataset:
   - T-Series having highest Subscriber
   - T-Series having maximum View Views
   - ABP News having maximum Video Count
   - Most of the channels started in the years 2006 and 2014
   - Music is the most popular category.