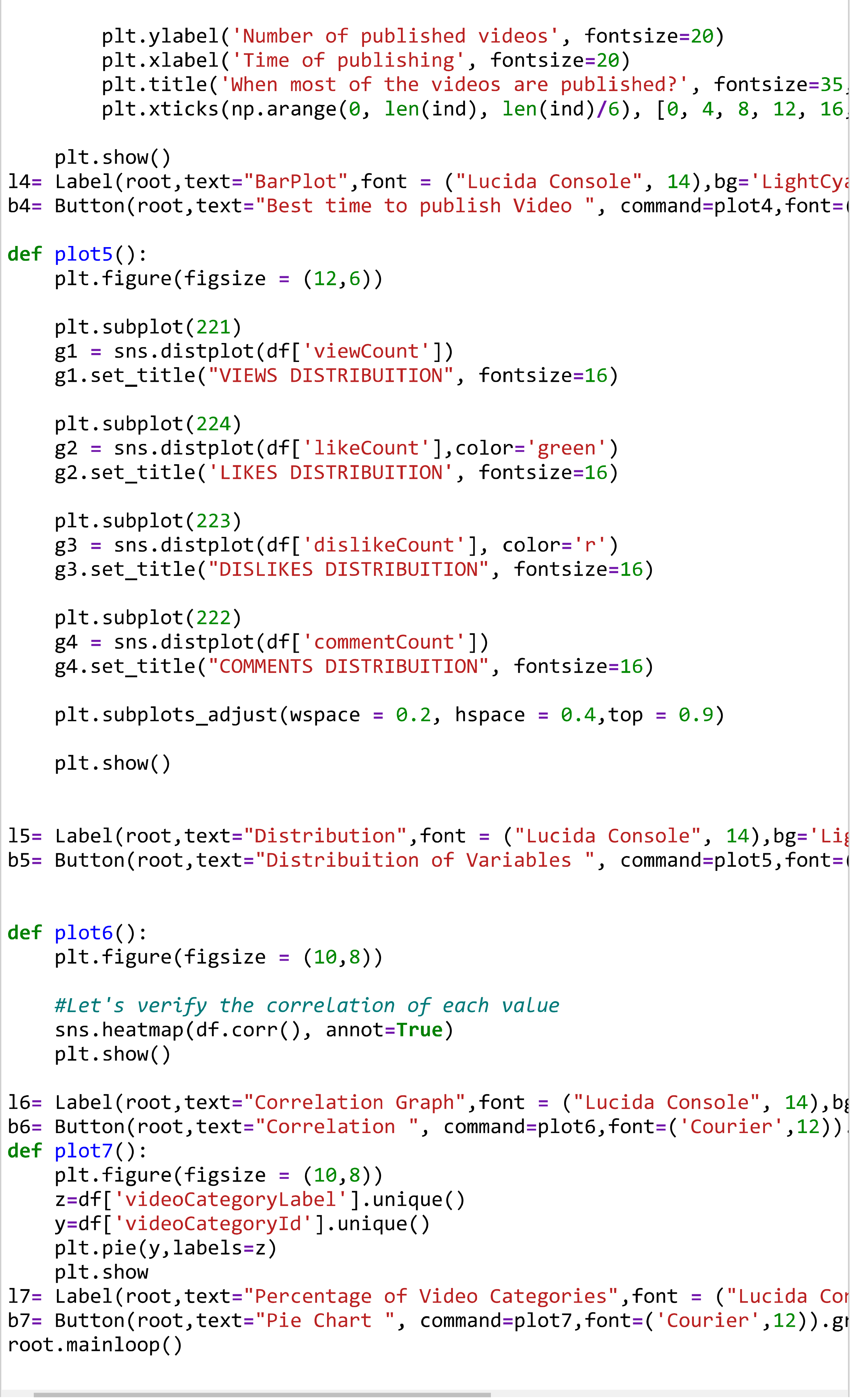
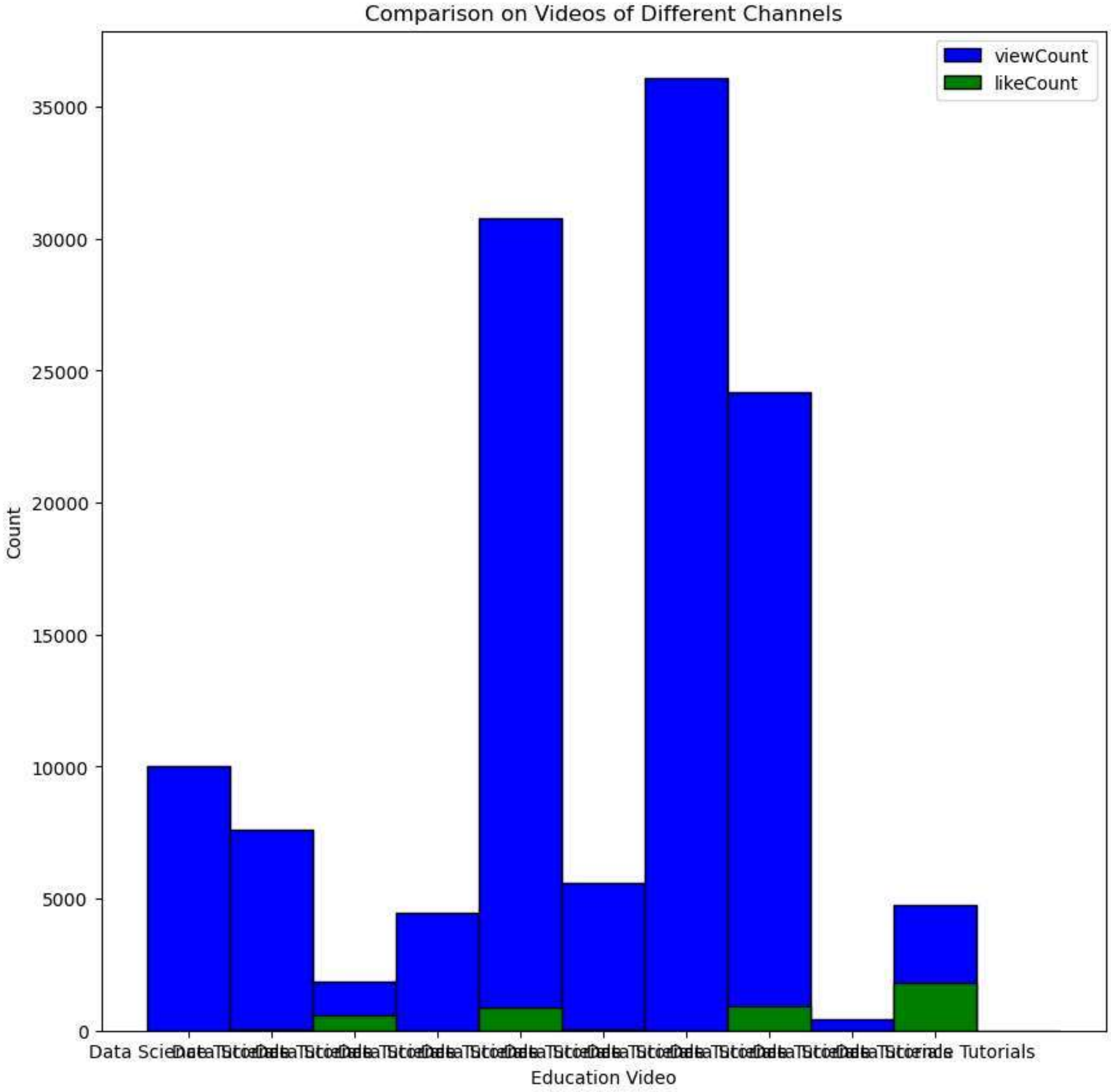
```
In [*]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from tkinter import *
        import seaborn as sns
        import re

        df=pd.read_csv("C:/Users/aarat/AppData/Local/Programs/Python/Python310/Y

        df.dropna(inplace=True)
        df.info()
        df.describe()

        Edu=df.loc[df['videoCategoryLabel']=='Education']

        ST=df.loc[df['videoCategoryLabel']=='Science & Technology']

        PB=df.loc[df['videoCategoryLabel']=='People & Blogs']

        DS=df.loc[df['channelTitle']=='Data Science Tutorials']

        y=df.head(20)
        df['viewCount'] = df['viewCount'].astype(float)
        publish_time = pd.to_datetime(df['publishedAt'], format='%Y-%m-%dT%H:%M:
        df['publish_date'] = publish_time.dt.date
        df['publish_time'] = publish_time.dt.time
        df['publish_hour'] = publish_time.dt.hour
        def clean_text(text):
            text = str(text).lower()
            text = re.sub(r'[^(a-zA-Z)\s]','', text)
            return text
        df['videoTitle'] = df['videoTitle'].apply(clean_text)
        df['videoDescription'] = df['videoDescription'].apply(clean_text)
        df.head(20)
        publish_h = [0] * 24

        for index, row in df.iterrows():
            publish_h[row["publish_hour"]] += 1

        values = publish_h
        ind = np.arange(len(values))

        root=Tk()
        root.title('Data visualization')
        root.iconbitmap("YouTube.csv")
        root.geometry('500x500')
        root.config(bg='LightCyan3')


        def Plot1():
            plt.figure(figsize=(10,10))
            viewCount=df["viewCount"].sample(10)
            likeCount=df["likeCount"].sample(10)
            edu=Edu
            n = len(viewCount)
            r = np.arange(n)
            width = 1
```

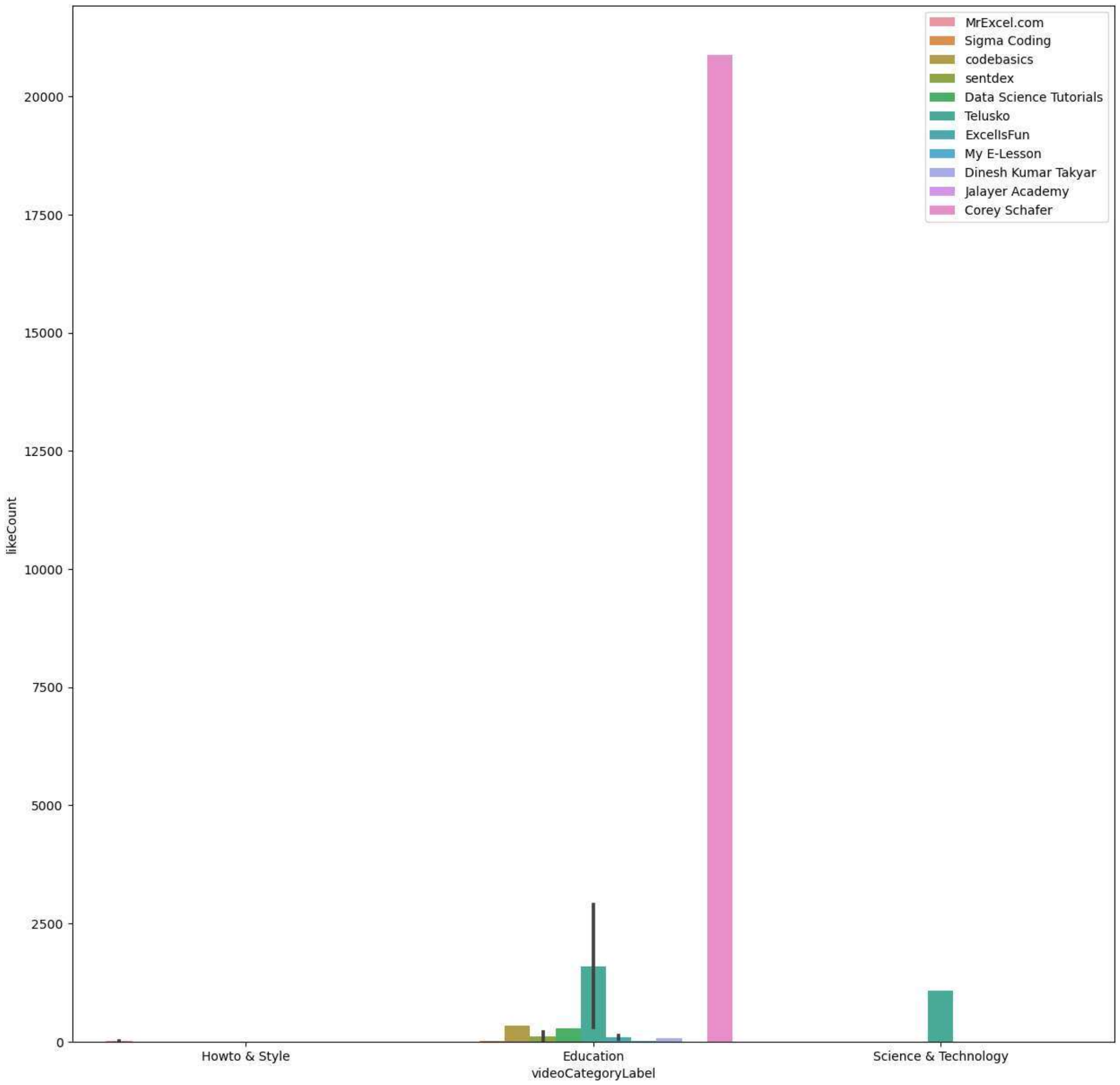
```
    plt.bar(r, viewCount, color='b',
          width=width, edgecolor='black',
          label='viewCount')
    plt.bar(r + width, likeCount, color='g',
          width=width, edgecolor='black',
          label='likeCount')
    plt.xlabel("Education Video")
    plt.ylabel("Count")
    plt.title("Comparison on Videos of Different Channels")


    plt.xticks(r,df["channelTitle"].tail(10))
    plt.legend()

    plt.show()

l1= Label(root,text="Plot",font = ("Lucida Console", 14),bg='LightCyan3
b1 = Button(root,text="Education Videos", command=Plot1,font=('Courier',

def Plot2():
    plt.figure(figsize=(15,15))
    sns.barplot(x="videoCategoryLabel",y="likeCount",hue="channelTitle",
    plt.legend(loc="upper right")
    plt.show()
l2= Label(root,text="Videos By Channels",font = ("Lucida Console", 14),b
b2= Button(root,text="Type of Videos", command=Plot2,font=('Courier',12)

df['like_rate'] =  df['likeCount'] / df['viewCount'] * 100
df['dislike_rate'] =  df['dislikeCount'] / df['viewCount'] * 100
df['comment_rate'] =  df['commentCount'] / df['viewCount'] * 100

df = df.replace([np.inf, -np.inf], np.nan)
def plot3():
    plt.figure(figsize = (9,6))
    g1 = sns.distplot(df ['dislike_rate'], color='red',hist=False, label
    g1 = sns.distplot(df ['like_rate'], color='green',hist=False, label=
    g1 = sns.distplot(df ['comment_rate'],hist=False,label="Comment")
    g1.set_title('CONVERT RATE DISTRIBUITION', fontsize=16)
    plt.xlabel('rate')
    plt.legend()
    plt.show()


l3= Label(root,text="Distribution",font = ("Lucida Console", 14),bg='Lig
b3= Button(root,text="Rate Distribution ", command=plot3,font=('Courier
# Creating new plot
def plot4():
    fig = plt.figure(figsize=(20,10))
    ax = fig.add_subplot(111)
    ax.yaxis.grid()
    ax.xaxis.grid()
    bars = ax.bar(ind, values)

# Sampling of Colormap
    for i, b in enumerate(bars):
        b.set_color(plt.cm.viridis((values[i] - min(values))/(max(values
```

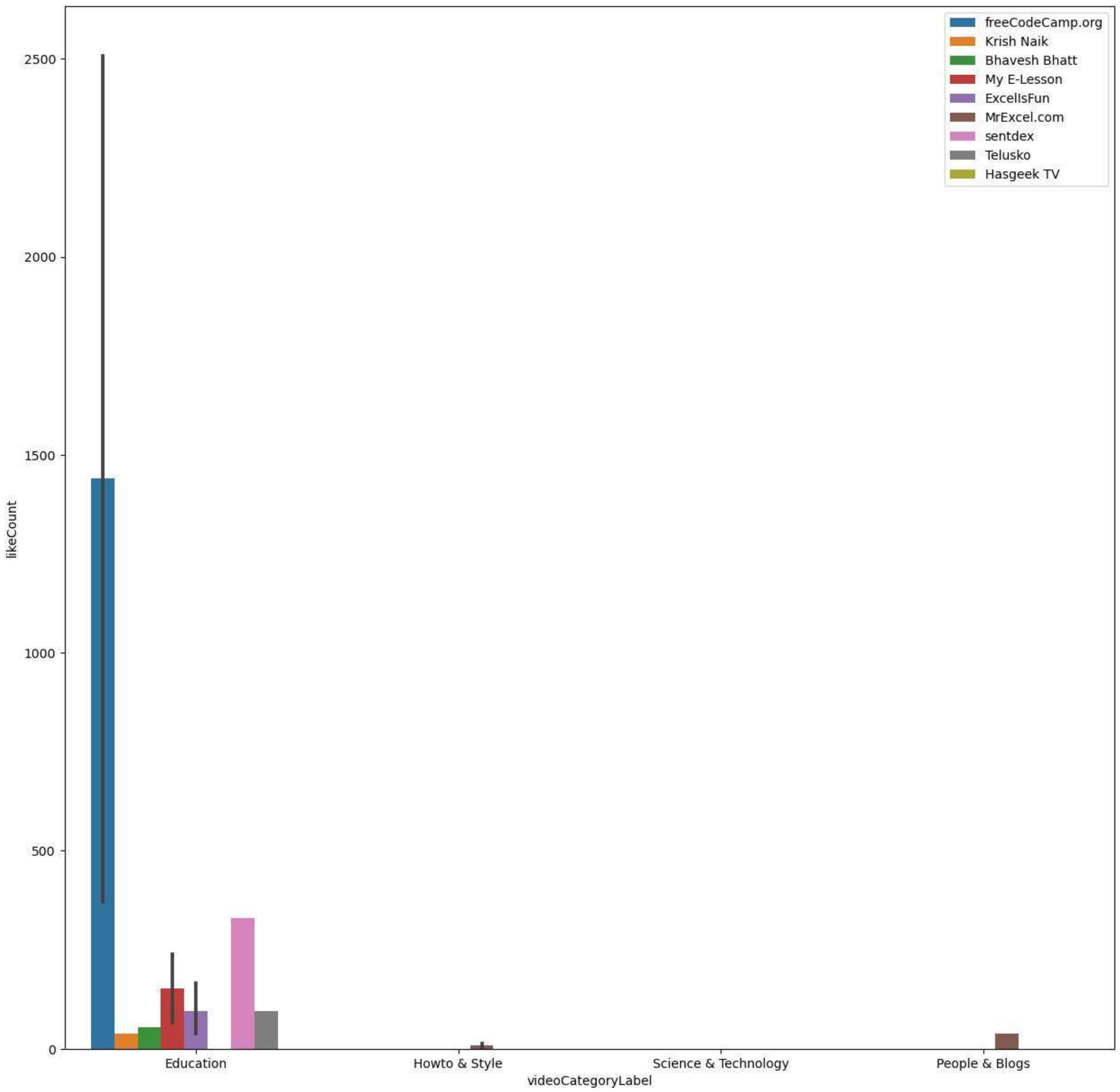


```
        plt.ylabel('Number of published videos', fontsize=20)
        plt.xlabel('Time of publishing', fontsize=20)
        plt.title('When most of the videos are published?', fontsize=35,
        plt.xticks(np.arange(0, len(ind), len(ind)/6), [0, 4, 8, 12, 16,

    plt.show()
l4= Label(root,text="BarPlot",font = ("Lucida Console", 14),bg='LightCya
b4= Button(root,text="Best time to publish Video ", command=plot4,font=(

def plot5():
    plt.figure(figsize = (12,6))

    plt.subplot(221)
    g1 = sns.distplot(df['viewCount'])
    g1.set_title("VIEWS DISTRIBUITION", fontsize=16)

    plt.subplot(224)
    g2 = sns.distplot(df['likeCount'],color='green')
    g2.set_title('LIKES DISTRIBUITION', fontsize=16)

    plt.subplot(223)
    g3 = sns.distplot(df['dislikeCount'], color='r')
    g3.set_title("DISLIKES DISTRIBUITION", fontsize=16)

    plt.subplot(222)
    g4 = sns.distplot(df['commentCount'])
    g4.set_title("COMMENTS DISTRIBUITION", fontsize=16)

    plt.subplots_adjust(wspace = 0.2, hspace = 0.4,top = 0.9)

    plt.show()


l5= Label(root,text="Distribution",font = ("Lucida Console", 14),bg='Lig
b5= Button(root,text="Distribuition of Variables ", command=plot5,font=(


def plot6():
    plt.figure(figsize = (10,8))

    #Let's verify the correlation of each value
    sns.heatmap(df.corr(), annot=True)
    plt.show()

l6= Label(root,text="Correlation Graph",font = ("Lucida Console", 14),bg
b6= Button(root,text="Correlation ", command=plot6,font=('Courier',12)).
def plot7():
    plt.figure(figsize = (10,8))
    z=df['videoCategoryLabel'].unique()
    y=df['videoCategoryId'].unique()
    plt.pie(y,labels=z)
    plt.show
l7= Label(root,text="Percentage of Video Categories",font = ("Lucida Con
b7= Button(root,text="Pie Chart ", command=plot7,font=('Courier',12)).gr
root.mainloop()
```

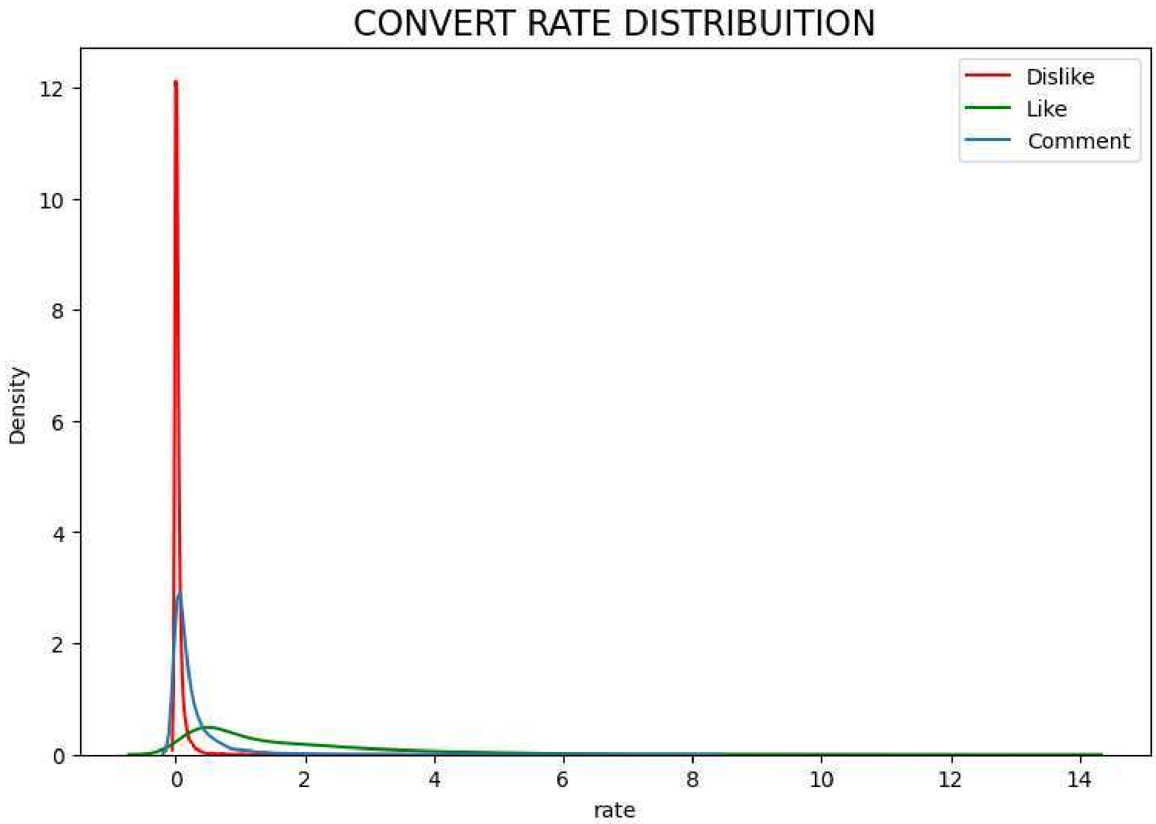
```
C:\Users\aarat\AppData\Local\Temp\ipykernel_5672\3555790458.py:8: Dtyp
eWarning: Columns (13) have mixed types. Specify dtype option on impor
t or set low_memory=False.
  df=pd.read_csv("C:/Users/aarat/AppData/Local/Programs/Python/Python3
10/YouTube.csv")

<class 'pandas.core.frame.DataFrame'>
Int64Index: 12708 entries, 7 to 43757
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   channelId            12708 non-null  object 
 1   channelTitle         12708 non-null  object 
 2   videoId              12708 non-null  object 
 3   publishedAt          12708 non-null  object 
 4   publishedAtSQL       12708 non-null  object 
 5   videoTitle           12708 non-null  object 
 6   videoDescription     12708 non-null  object 
 7   videoCategoryId      12708 non-null  float64
 8   videoCategoryLabel   12708 non-null  object 
 9   duration             12708 non-null  object 
 10  durationSec          12708 non-null  float64
 11  dimension            12708 non-null  object 
 12  definition           12708 non-null  object 
 13  caption              12708 non-null  object 
 14  thumbnail_maxres     12708 non-null  object 
 15  licensedContent      12708 non-null  float64
 16  viewCount            12708 non-null  float64
 17  likeCount            12708 non-null  float64
 18  dislikeCount         12708 non-null  float64
 19  favoriteCount        12708 non-null  float64
 20  commentCount         12708 non-null  float64
dtypes: float64(8), object(13)
memory usage: 2.1+ MB
```


Comparison on Videos of Different Channels
viewCount
likeCount
Count
35000
30000
25000
20000
15000
10000
5000
0
Data Science Tutorials
Education Video

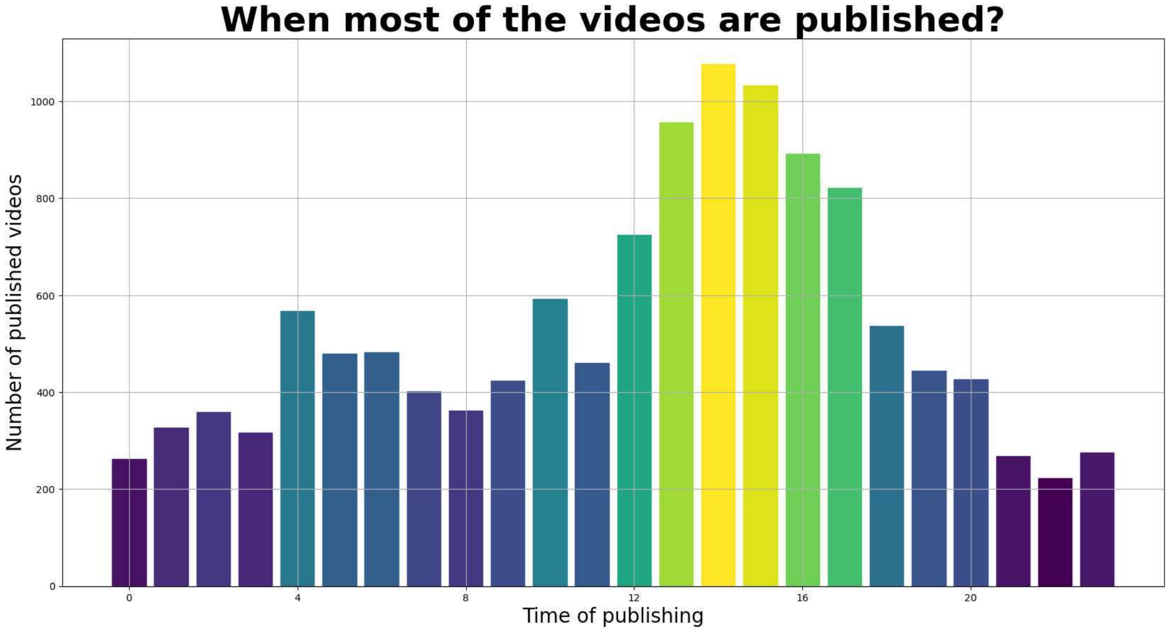

MrExcel.com
Sigma Coding
codebasics
sentdex
Data Science Tutorials
Telusko
ExcelIsFun
My E-Lesson
Dinesh Kumar Takyar
Jalayer Academy
Corey Schafer
likeCount
20000
17500
15000
12500
10000
7500
5000
2500
0
Howto & Style
Education
Science & Technology
videoCategoryLabel



```
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `kdeplot` (an
axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `kdeplot` (an
axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `kdeplot` (an
axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

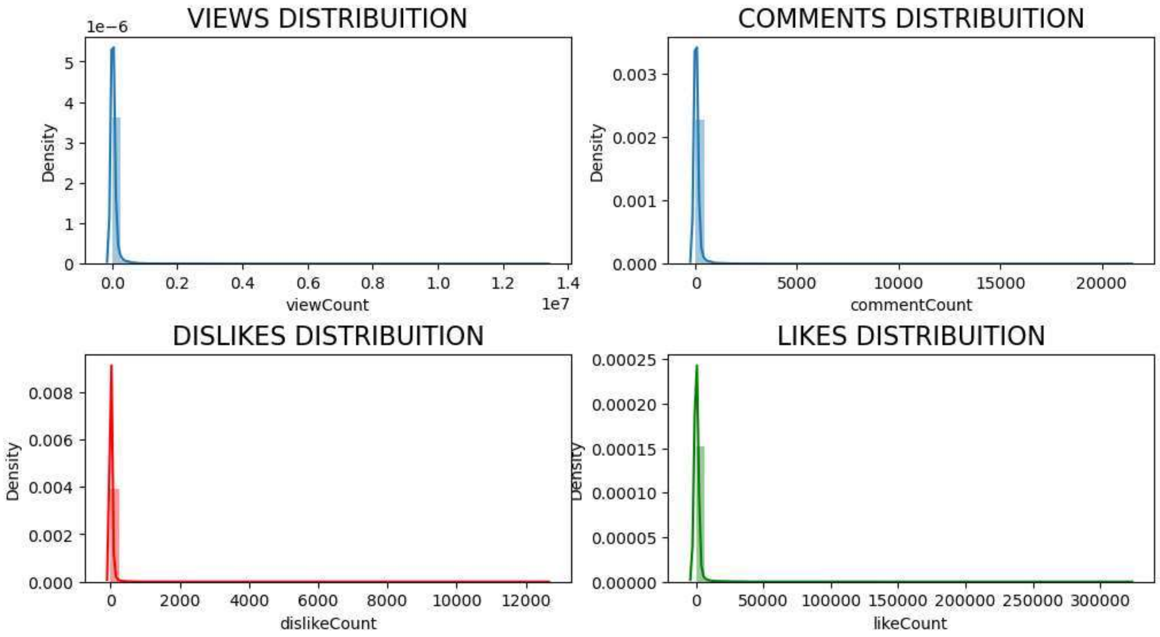


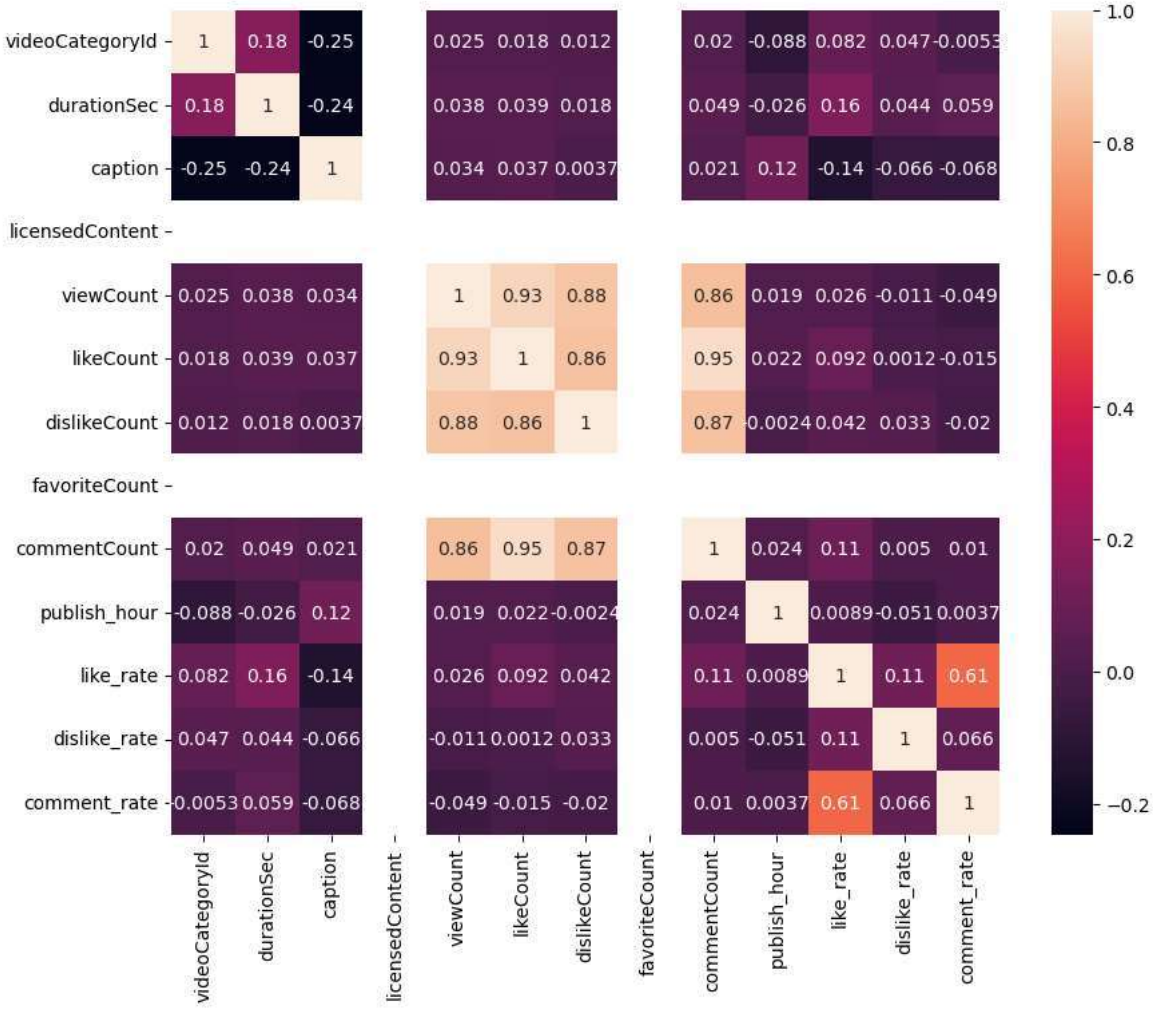
CONVERT RATE DISTRIBUITION

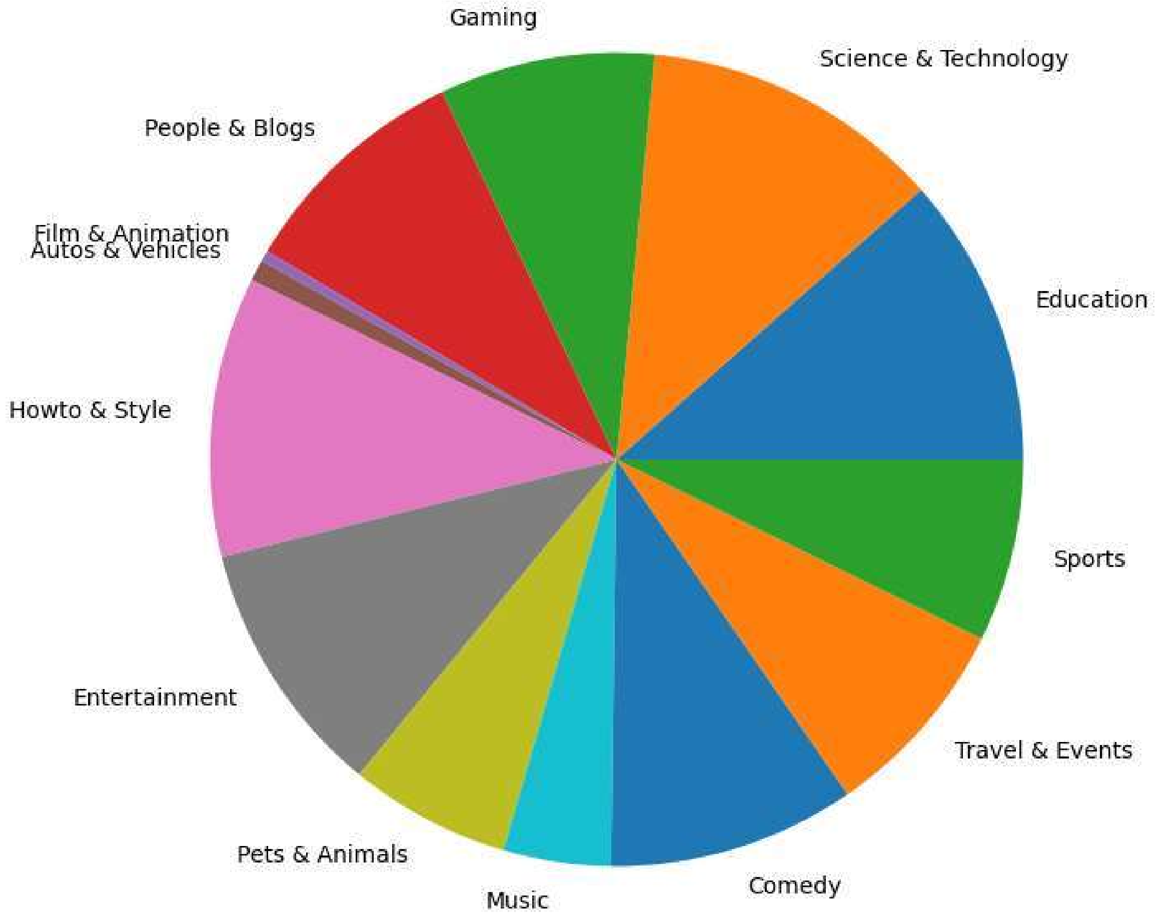


When most of the videos are published?

```
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (a
n axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (a
n axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (a
n axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\aarat\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (a
n axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




videoCategoryId
durationSec
caption
licensedContent
viewCount
likeCount
dislikeCount
favoriteCount
commentCount
publish_hour
like_rate
dislike_rate
comment_rate
1 0.18 -0.25 0.025 0.018 0.012 0.02 -0.088 0.082 0.047 -0.0053
0.18 1 -0.24 0.038 0.039 0.018 0.049 -0.026 0.16 0.044 0.059
-0.25 -0.24 1 0.034 0.037 0.0037 0.021 0.12 -0.14 -0.066 -0.068
0.025 0.038 0.034 1 0.93 0.88 0.86 0.019 0.026 -0.011 -0.049
0.018 0.039 0.037 0.93 1 0.86 0.95 0.022 0.092 0.0012 -0.015
0.012 0.018 0.0037 0.88 0.86 1 0.87 -0.0024 0.042 0.033 -0.02
0.02 0.049 0.021 0.86 0.95 0.87 1 0.024 0.11 0.005 0.01
-0.088 -0.026 0.12 0.019 0.022 -0.0024 0.024 1 0.0089 -0.051 0.0037
0.082 0.16 -0.14 0.026 0.092 0.042 0.11 0.0089 1 0.11 0.61
0.047 0.044 -0.066 -0.011 0.0012 0.033 0.005 -0.051 0.11 1 0.066
-0.0053 0.059 -0.068 -0.049 -0.015 -0.02 0.01 0.0037 0.61 0.066 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Gaming
Science & Technology
People & Blogs
Film & Animation
Autos & Vehicles
Education
Howto & Style
Sports
Entertainment
Travel & Events
Pets & Animals
Comedy
Music