MAINFLOW SERVICES AND TECHNOLOGIES

TASK-4

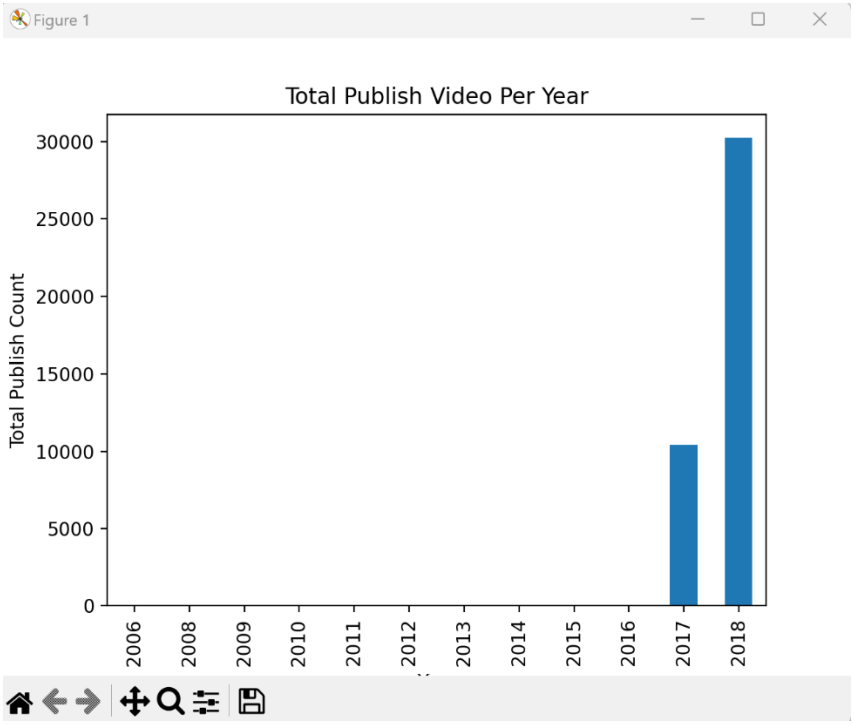
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
df=pd.read_csv('C:\\Users\\glady\\OneDrive\\Desktop\\DS with Python\\USvideos.csv')
df.head()
```

```
>>>df.head()
      video_id  ...                                        description
0  2kyS6SvSYSE  ...  SHANTELL'S CHANNEL - https://www.youtube.com/s...
1  1ZAPwfrtAFY  ...  One year after the presidential election, John...
2  5qpjK5DgCt4  ...  WATCH MY PREVIOUS VIDEO ▶ \n\nSUBSCRIBE ▶ http...
3  puqaWrEC7tY  ...  Today we find out if Link is a Nickelback amat...
4  d380meD0W0M  ...  I know it's been a while since we did this sho...

[5 rows x 16 columns]
>>>df.head()
      video_id  ...                                        description
0  2kyS6SvSYSE  ...  SHANTELL'S CHANNEL - https://www.youtube.com/s...
1  1ZAPwfrtAFY  ...  One year after the presidential election, John...
2  5qpjK5DgCt4  ...  WATCH MY PREVIOUS VIDEO ▶ \n\nSUBSCRIBE ▶ http...
3  puqaWrEC7tY  ...  Today we find out if Link is a Nickelback amat...
4  d380meD0W0M  ...  I know it's been a while since we did this sho...

[5 rows x 16 columns]
>>>df.shape
(40949, 16)
>>>df=df.drop_duplicates()
>>>df.shape
(40901, 16)
>>>df.describe()
        category_id         views         likes      dislikes  comment_count
count  40901.000000  4.090100e+04  4.090100e+04  4.090100e+04   4.090100e+04
mean      19.970588  2.360678e+06  7.427173e+04  3.711722e+03   8.448567e+03
std        7.569362  7.397719e+06  2.289999e+05  2.904624e+04   3.745139e+04
min        1.000000  5.490000e+02  0.000000e+00  0.000000e+00   0.000000e+00
25%       17.000000  2.419720e+05  5.416000e+03  2.020000e+02   6.130000e+02
50%       24.000000  6.810640e+05  1.806900e+04  6.300000e+02   1.855000e+03
75%       25.000000  1.821926e+06  5.533800e+04  1.936000e+03   5.752000e+03
max       43.000000  2.252119e+08  5.613827e+06  1.674420e+06   1.361580e+06
```

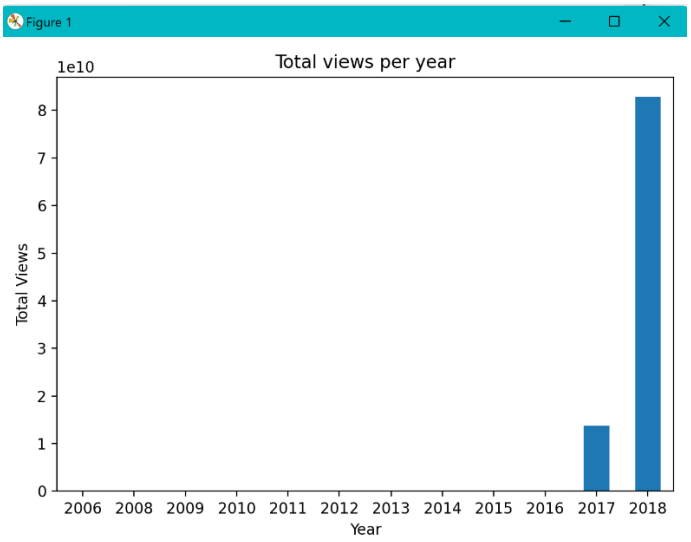
```
df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 40901 entries, 0 to 40948
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40901 non-null  object
 1   trending_date           40901 non-null  object
 2   title                   40901 non-null  object
 3   channel_title           40901 non-null  object
 4   category_id             40901 non-null  int64
 5   publish_time            40901 non-null  object
 6   tags                    40901 non-null  object
 7   views                   40901 non-null  int64
 8   likes                   40901 non-null  int64
 9   dislikes                40901 non-null  int64
 10  comment_count           40901 non-null  int64
 11  thumbnail_link          40901 non-null  object
 12  comments_disabled       40901 non-null  bool
 13  ratings_disabled        40901 non-null  bool
 14  video_error_or_removed  40901 non-null  bool
 15  description             40332 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 4.5+ MB

columns_to_remove=['thumbnail_link','description']
df=df.drop(columns=columns_to_remove)
df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 40901 entries, 0 to 40948
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40901 non-null  object
 1   trending_date           40901 non-null  object
 2   title                   40901 non-null  object
 3   channel_title           40901 non-null  object
 4   category_id             40901 non-null  int64
 5   publish_time            40901 non-null  object
 6   tags                    40901 non-null  object
 7   views                   40901 non-null  int64
 8   likes                   40901 non-null  int64
 9   dislikes                40901 non-null  int64
 10  comment_count           40901 non-null  int64
 11  comments_disabled       40901 non-null  bool
 12  ratings_disabled        40901 non-null  bool
 13  video_error_or_removed  40901 non-null  bool
dtypes: bool(3), int64(5), object(6)
memory usage: 3.9+ MB
```

```
>>>from datetime import datetime
>>>import datetime
>>>df["trending_date"]=df["trending_date"].apply(lambda x:datetime.datetime.strptime(x,'%y.%d.%m'))
>>>df.head(3)
        video_id trending_date  ... ratings_disabled video_error_or_removed
0  2kyS6SvSYSE    2017-11-14  ...            False                  False
1  1ZAPwfrtAFY    2017-11-14  ...            False                  False
2  5qpjK5DgCt4    2017-11-14  ...            False                  False

[3 rows x 14 columns]
>>>df['publish_time']=pd.to_datetime(df['publish_time'])
>>>df.head(2)
        video_id trending_date  ... ratings_disabled video_error_or_removed
0  2kyS6SvSYSE    2017-11-14  ...            False                  False
1  1ZAPwfrtAFY    2017-11-14  ...            False                  False

[2 rows x 14 columns]
>>>df['publish_month']=df['publish_time'].dt.month
>>>df['publish_day']=df['publish_time'].dt.day
>>>df['publish_hour']=df['publish_time'].dt.hour
>>>df.head(2)
        video_id trending_date  ... publish_day publish_hour
0  2kyS6SvSYSE    2017-11-14  ...          13           17
1  1ZAPwfrtAFY    2017-11-14  ...          13            7

[2 rows x 17 columns]
>>>print(sorted(df["category_id"].unique()))
[np.int64(1), np.int64(2), np.int64(10), np.int64(15), np.int64(17), np.int64(19), np.int64(20), np.int64(22), np.int6
4(23), np.int64(24), np.int64(25), np.int64(26), np.int64(27), np.int64(28), np.int64(29), np.int64(43)]
```

```
>>>df['year']=df['publish_time'].dt.year
>>>yearly_counts=df.groupby('year')['video_id'].count()
>>>yearly_counts.plot(kind='bar',xlabel='Year',ylabel='Total Publish Count',title='Total Publish Video Per Year')
   <Axes: title={'center': 'Total Publish Video Per Year'}, xlabel='Year', ylabel='Total Publish Count'>
>>>plt.show()
```

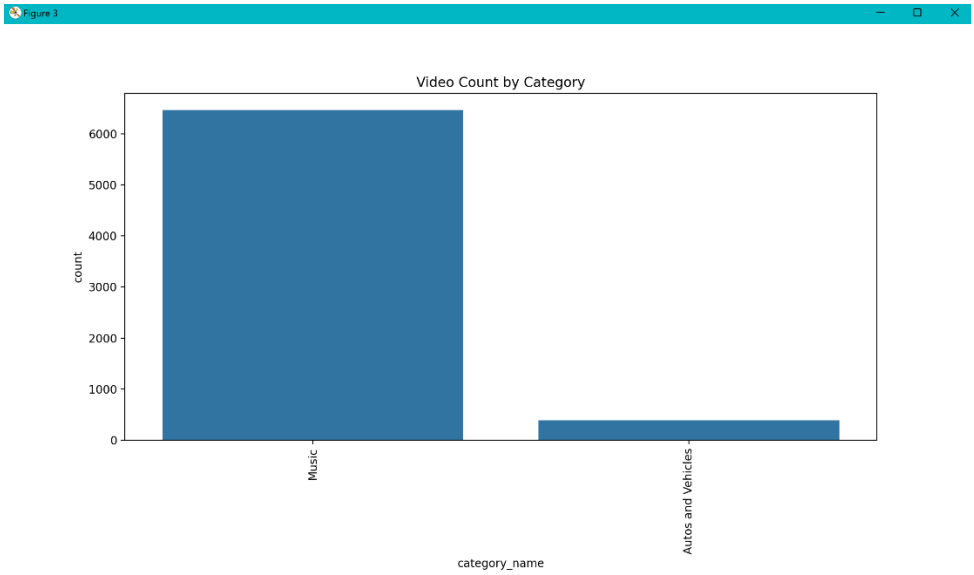



```
>>>df['year']=df['publish_time'].dt.year
>>>yearly_counts=df.groupby('year')['video_id'].count()
>>>yearly_counts.plot(kind='bar',xlabel='Year',ylabel='Total Publish Count',title='Total Publish Video Per Year')
   <Axes: title={'center': 'Total Publish Video Per Year'}, xlabel='Year', ylabel='Total Publish Count'>
>>>plt.show()
>>>yearly_views=df.groupby('year')['views'].sum()
>>>yearly_views.plot(kind='bar',xlabel='Year',ylabel='Total Views',title='Total views per year')
   <Axes: title={'center': 'Total views per year'}, xlabel='Year', ylabel='Total Views'>
>>>plt.xticks(rotation=0)
   (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]), [Text(0, 0, '2006'), Text(1, 0, '2008'), Text(2, 0, '2009'),
    Text(3, 0, '2010'), Text(4, 0, '2011'), Text(5, 0, '2012'), Text(6, 0, '2013'), Text(7, 0, '2014'), Text(8, 0, '2015'
   ), Text(9, 0, '2016'), Text(10, 0, '2017'), Text(11, 0, '2018')])
>>>plt.tight_layout()
>>>plt.show()
```



```
>>>df.loc[(df["category_id"]==2), "category_name"]='Autos and Vehicles'
>>>df.head()
      video_id trending_date  ... category_name category__name
0  2kyS6SvSYSE    2017-11-14  ...           NaN            NaN
1  1ZAPwfrtAFY    2017-11-14  ...           NaN            NaN
2  5qpjK5DgCt4    2017-11-14  ...           NaN            NaN
3  puqaWrEC7tY    2017-11-14  ...           NaN            NaN
4  d380meD0W0M    2017-11-14  ...           NaN            NaN

[5 rows x 19 columns]
>>>
```

```
>>>plt.figure(figsize=(12,6))
   <Figure size 1200x600 with 0 Axes>
>>>sns.countplot(x='category_name',data=df,order=df['category_name'].value_counts().index)
   <Axes: xlabel='category_name', ylabel='count'>
>>>plt.xticks(rotation=90)
   ([0, 1], [Text(0, 0, 'Music'), Text(1, 0, 'Autos and Vehicles')])
>>>plt.title('Video Count by Category')
   Text(0.5, 1.0, 'Video Count by Category')
>>>plt.show()
```




```
>>>videos_per_hour=df['publish_hour'].value_counts().sort_index()
>>>plt.figure(figsize=(12,6))
   <Figure size 1200x600 with 0 Axes>
>>>sns.barplot(x=videos_per_hour.index,y=videos_per_hour.values,palette='rocket')

   Warning (from warnings module):
     File "<pyshell#50>", line 1
   FutureWarning:

   Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hu
   e` and set `legend=False` for the same effect.

   <Axes: xlabel='publish_hour'>
>>>plt.title("Number of Videos Published per Hour")
   Text(0.5, 1.0, 'Number of Videos Published per Hour')
>>>plt.xlabel('Hour of Day')
   Text(0.5, 0, 'Hour of Day')
>>>plt.ylabel('Number of Videos')
   Text(0, 0.5, 'Number of Videos')
>>>plt.xticks(rotation=45)
   ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23], [Text(0, 0, '0'), Text(1, 0,
   '1'), Text(2, 0, '2'), Text(3, 0, '3'), Text(4, 0, '4'), Text(5, 0, '5'), Text(6, 0, '6'), Text(7, 0, '7'), Text(8, 0,
    '8'), Text(9, 0, '9'), Text(10, 0, '10'), Text(11, 0, '11'), Text(12, 0, '12'), Text(13, 0, '13'), Text(14, 0, '14'),
    Text(15, 0, '15'), Text(16, 0, '16'), Text(17, 0, '17'), Text(18, 0, '18'), Text(19, 0, '19'), Text(20, 0, '20'), Tex
   t(21, 0, '21'), Text(22, 0, '22'), Text(23, 0, '23')])
>>>plt.show()
```

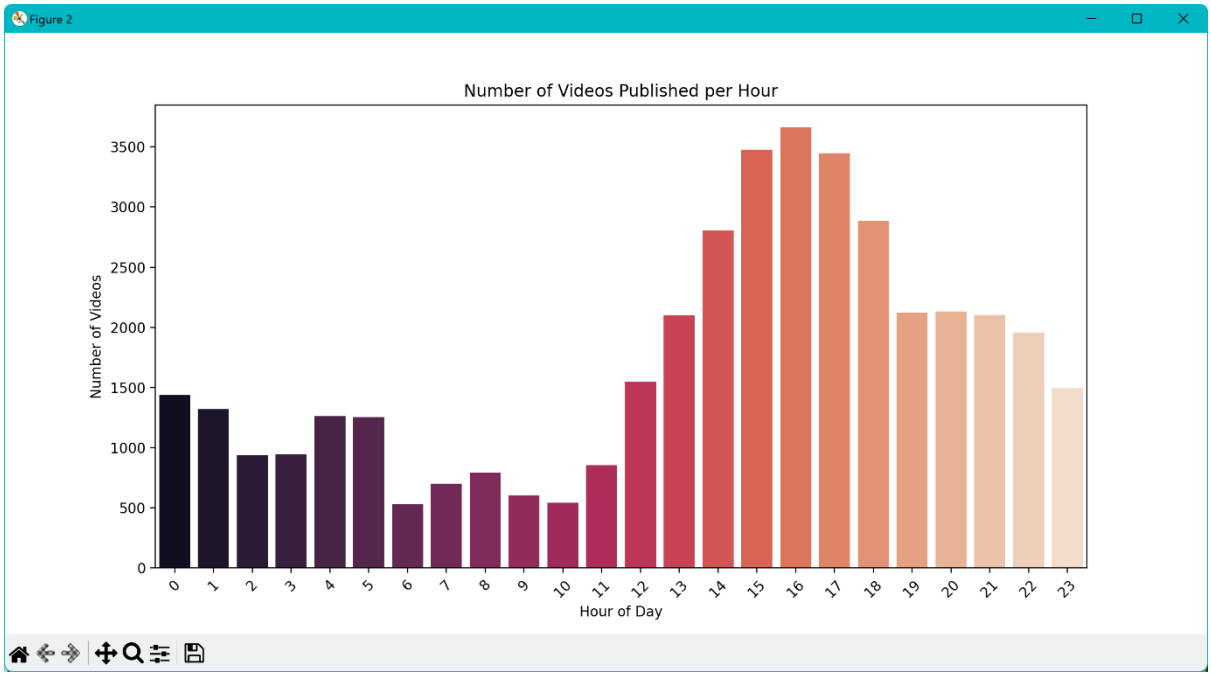


Figure 2
Number of Videos Published per Hour
Number of Videos
0
500
1000
1500
2000
2500
3000
3500
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day