In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('most_subscribed_youtube_channels.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	rank	Youtuber	subscribers	video views	video count	category	started
0	1	T-Series	22,20,00,000	1,98,45,90,90,822	17,317	Music	2006
1	2	YouTube Movies	15,40,00,000	0	0	Film & Animation	2015
2	3	Cocomelon - Nursery Rhymes	14,00,00,000	1,35,48,13,39,848	786	Education	2006
3	4	SET India	13,90,00,000	1,25,76,42,52,686	91,271	Shows	2006
4	5	Music	11,60,00,000	0	0	NaN	2013

In [4]:

```
df.tail()
```

Out[4]:

	rank	Youtuber	subscribers	video views	video count	category	started
995	996	JP Plays	1,09,00,000	4,60,93,00,218	3,528	Gaming	2014
996	997	TrapMusicHDTV	1,09,00,000	4,07,05,21,973	690	Music	2013
997	998	Games EduUu	1,09,00,000	3,09,37,84,767	1,006	Gaming	2011
998	999	Hueva	1,09,00,000	3,04,03,01,750	831	Gaming	2012
999	1000	Dobre Brothers	1,09,00,000	2,80,84,11,693	590	People & Blogs	2017

In [5]:

```
df.shape
```

Out[5]:

```
(1000, 7)
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['rank', 'Youtuber', 'subscribers', 'video views', 'video count',
       'category', 'started'],
      dtype='object')
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
rank            0
Youtuber        0
subscribers     0
video views     0
video count     0
category       27
started         0
dtype: int64
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   rank         1000 non-null   int64 
 1   Youtuber     1000 non-null   object
 2   subscribers  1000 non-null   object
 3   video views  1000 non-null   object
 4   video count  1000 non-null   object
 5   category     973 non-null    object
 6   started      1000 non-null   int64 
dtypes: int64(2), object(5)
memory usage: 54.8+ KB
```

In [10]:

```
df.describe()
```

Out[10]:

	rank	started
count	1000.000000	1000.000000
mean	500.500000	2012.376000
std	288.819436	3.998076
min	1.000000	1970.000000
25%	250.750000	2010.000000
50%	500.500000	2013.000000
75%	750.250000	2015.000000
max	1000.000000	2021.000000

In [11]:

```
df.nunique()
```

Out[11]:

```
rank            1000
Youtuber         999
subscribers      286
video views      991
video count      856
category          18
started           18
dtype: int64
```

In [12]:

```
df['category'].fillna("Unknown", inplace=True)
```

In [13]:

```
df['category'].unique()
```

Out[13]:

```
array(['Music', 'Film & Animation', 'Education', 'Shows', 'Unknown',
       'Gaming', 'Entertainment', 'People & Blogs', 'Sports',
       'Howto & Style', 'News & Politics', 'Comedy', 'Trailers',
       'Nonprofits & Activism', 'Science & Technology', 'Movies',
       'Pets & Animals', 'Autos & Vehicles', 'Travel & Events'],
      dtype=object)
```

In [14]:

```
df['category'].value_counts()
```

Out[14]:

```
Entertainment            241
Music                    222
People & Blogs           119
Gaming                   102
Comedy                    63
Film & Animation          52
Education                 46
Howto & Style             45
Unknown                   27
News & Politics           27
Science & Technology      18
Shows                     14
Sports                    10
Pets & Animals             6
Trailers                   2
Nonprofits & Activism      2
Movies                     2
Autos & Vehicles           1
Travel & Events            1
Name: category, dtype: int64
```

In [15]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

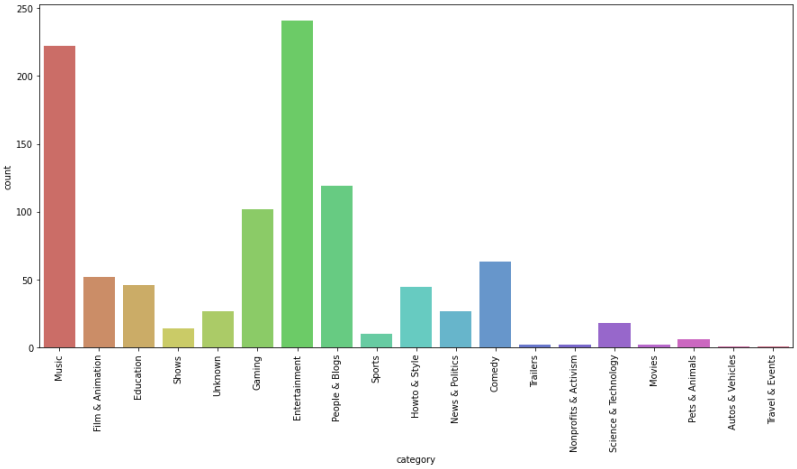
In [16]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [17]:

```
plt.figure(figsize=[15,7],)
print('Countplot for Category')
sns.countplot(df['category'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

```
Countplot for Category
```



In [18]:

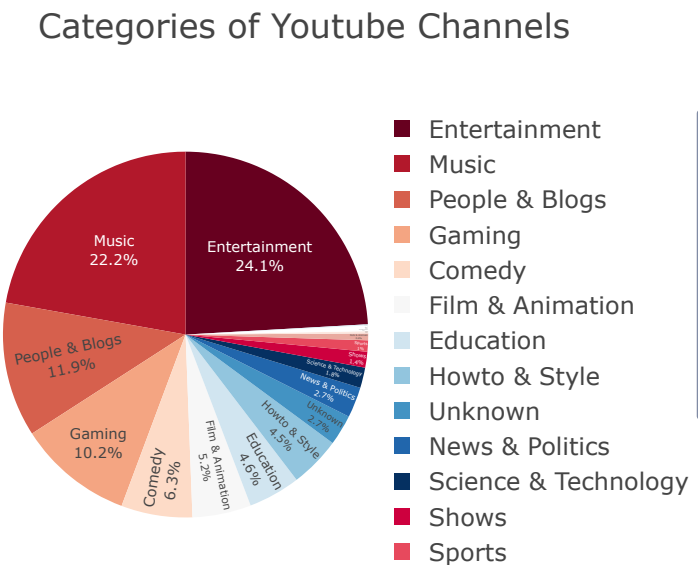
```
df['video views']=df['video views'].str.replace(',','')
df['video count']=df['video count'].str.replace(',','')
df['subscribers']=df['subscribers'].str.replace(',','')
df['video views']=df['video views'].astype('int64')
df['video count']=df['video count'].astype('int64')
df['subscribers']=df['subscribers'].astype('int64')
```

In [19]:

```
import plotly.express as px
```

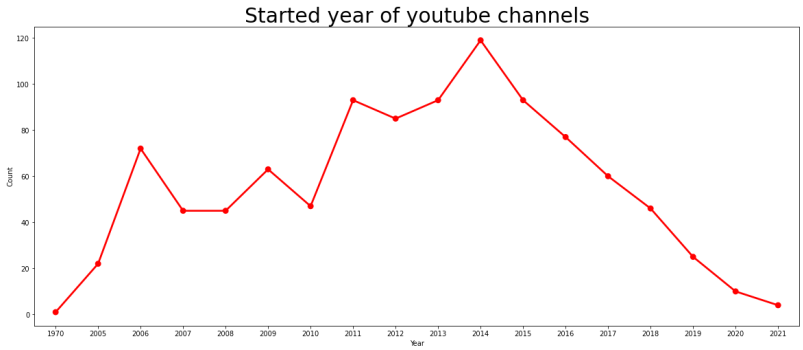
In [20]:

```
categories=df['category'].value_counts()
fig=px.pie(values=categories.values,
           names=categories.index,
           color_discrete_sequence=px.colors.sequential.RdBu,
           title="Categories of Youtube Channels", template='presentation'
           )
fig.update_traces(textposition='inside',
                  textfont_size=11,
                  textinfo='percent+label')
fig.show();
```



In [21]:

```
year=df['started'].value_counts()
plt.figure(figsize=(20,8))
sns.pointplot(x=year.index,y=year.values, color='red')
plt.xlabel('Year')
plt.ylabel('Count')
plt.title('Started year of youtube channels',size=30, color='black');
```



In [22]:

```
year_mean=df.groupby('started').mean().reset_index()
year_mean
```

Out[22]:

	started	rank	subscribers	video views	video count
0	1970	100.000000	3.330000e+07	2.725287e+09	540.000000
1	2005	423.590909	2.197273e+07	1.044777e+10	15480.409091
2	2006	426.625000	2.767361e+07	1.676924e+10	16612.625000
3	2007	466.866667	2.365111e+07	1.396931e+10	24476.800000
4	2008	452.533333	2.118222e+07	1.140225e+10	14807.333333
5	2009	468.460317	2.001111e+07	1.023113e+10	10564.380952
6	2010	532.127660	1.935957e+07	8.997569e+09	9957.319149
7	2011	485.204301	1.981720e+07	8.804918e+09	5772.118280
8	2012	487.752941	2.116588e+07	8.844339e+09	7142.811765
9	2013	463.483871	2.200108e+07	7.183893e+09	7368.139785
10	2014	532.226891	1.904790e+07	8.453754e+09	8370.806723
11	2015	542.978495	1.974086e+07	7.481183e+09	4237.698925
12	2016	508.883117	2.041558e+07	8.136676e+09	2594.441558
13	2017	536.533333	1.746833e+07	5.230756e+09	5168.783333
14	2018	557.195652	1.791304e+07	8.470628e+09	6515.543478
15	2019	616.480000	1.488400e+07	6.609422e+09	2673.560000
16	2020	573.800000	1.585000e+07	7.795733e+09	1385.000000
17	2021	698.000000	1.415000e+07	8.552476e+09	696.750000

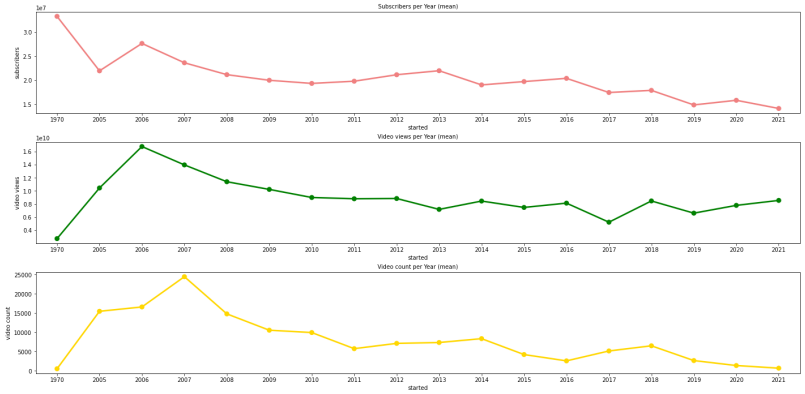
In [23]:

```
def pltplot(data, xcol, ycol, color, ax, title):
    sns.pointplot(data=data, x=xcol, y=ycol, color=color, ax=ax).set_title(title, size=10)
```

In [24]:

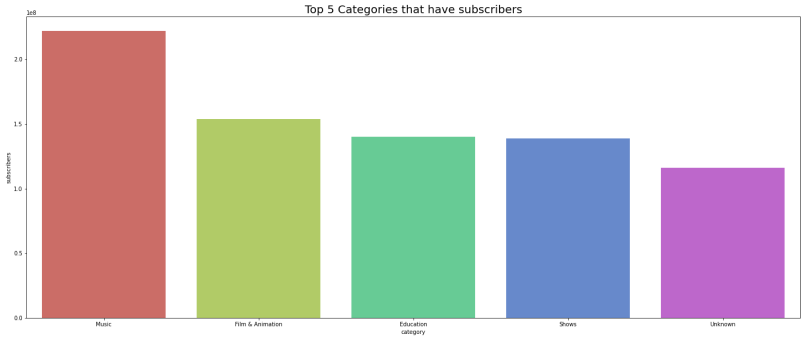
```
fig, ((ax1),(ax2),(ax3))=plt.subplots(ncols=1, nrows=3)
fig.set_size_inches(20,10)
fig.tight_layout(pad=3.0)

pltplot(year_mean,'started','subscribers','lightcoral', ax1,'Subscribers per Year (mean)')
pltplot(year_mean,'started','video views','green', ax2,'Video views per Year (mean)')
pltplot(year_mean,'started','video count','gold', ax3,'Video count per Year (mean)')
```



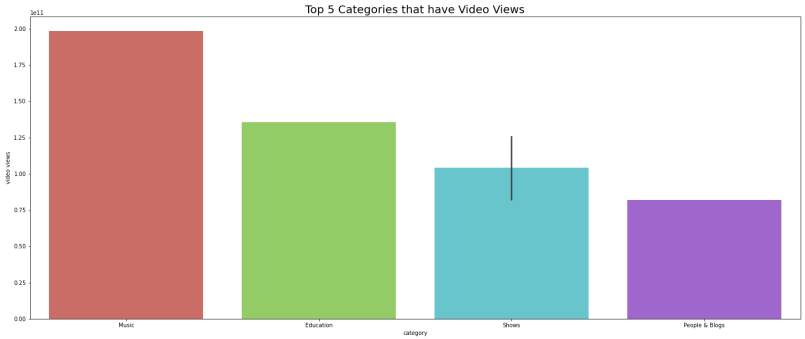
In [25]:

```
subscribers=df.sort_values('subscribers',ascending=False)
plt.figure(figsize=(25,10))
subscribers=subscribers[:5]
sns.barplot(x="category",
            y="subscribers",
            data=subscribers,
            palette="hls")
plt.title('Top 5 Categories that have subscribers',size=20);
```



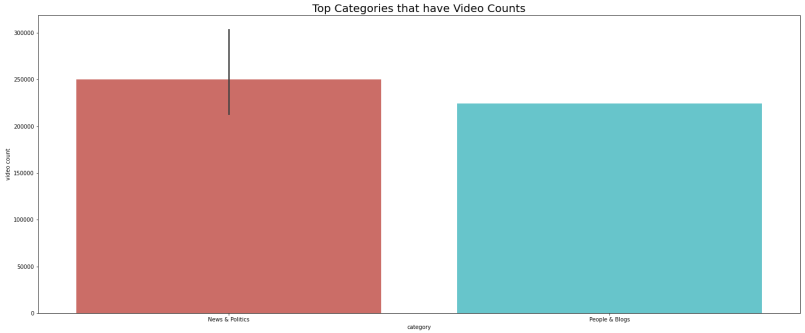
In [26]:

```
videoviews=df.sort_values('video views',ascending=False)
plt.figure(figsize=(25,10))
videoviews=videoviews[:5]
sns.barplot(x="category",
            y="video views",
            data=videoviews,
            palette="hls")
plt.title('Top 5 Categories that have Video Views',size=20);
```



In [27]:

```
videocount=df.sort_values('video count',ascending=False)
plt.figure(figsize=(25,10))
videocount=videocount[:5]
sns.barplot(x="category",
            y="video count",
            data=videocount,
            palette="hls")
plt.title('Top Categories that have Video Counts',size=20);
```

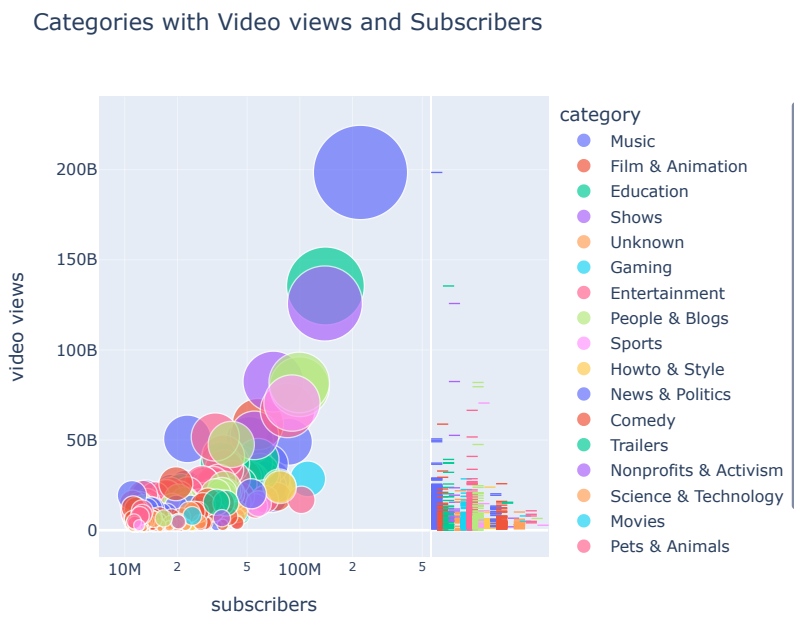


In [28]:

```
def plot(data, xcol, ycol, size, color,title):
    px.scatter(data, x=xcol,y=ycol,
               size=size, color=color,
               log_x=True, size_max=50).set_title(title,fontsize=20)
    axs.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
title1 = 'categories with video views and subscribers'
title2 = 'categories with video views and video counts'
```

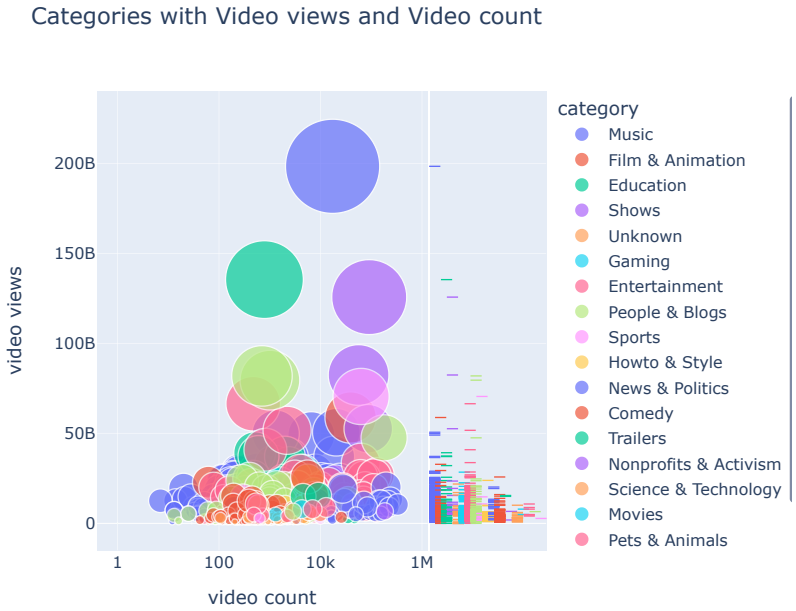
In [29]:

```
fig = px.scatter(df, x="subscribers", y="video views",
                 size="video views", color="category",
                 log_x=True, size_max=50,
                 title="Categories with Video views and Subscribers",
                 marginal_y='rug')
fig.show()
```



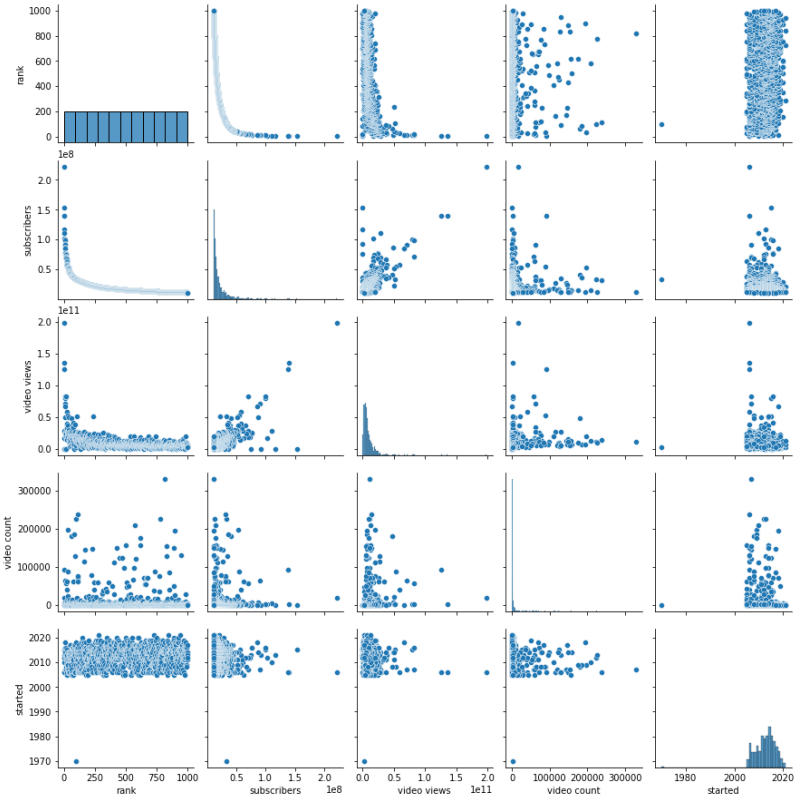
In [30]:

```
fig = px.scatter(df, x="video count", y="video views",
                 size="video views", color="category",
                 log_x=True, size_max=50,
                 title="Categories with Video views and Video count",
                 marginal_y='rug')
fig.show()
```



In [31]:

```
sns.pairplot(df)
plt.show();
```



In [32]:

```
plt.figure(figsize=(20,8))
sns.heatmap(df.corr(), annot=True, center=True, cmap = 'coolwarm'
            , cbar = False);
plt.show()
```