Finlatics: Data Science with Python

Project: Media and Technology

CODE TRANSCRIPTS

Dataset loading and data preprocessing:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv(r'C:\Users\Hp\OneDrive\Desktop\global_youtube_statistics.csv', encoding='latin-1')
df['created_year'].fillna(df['created_year'].mode()[0],inplace=True)
df['created_month'].fillna(df['created_month'].mode()[0],inplace=True)
df['created_date'].fillna(df['created_date'].mode()[0],inplace=True)
df['Country']=df['Country'].str.lower()
df['Country'].fillna('Unknown',inplace=True)
unknown_rows = df['Country'] == 'Unknown'
df.loc[unknown_rows, 'Population'] = df['Population'].median()
df.loc[unknown_rows, 'Gross tertiary education enrollment (%)'] = df['Gross tertiary education enrollment (%)'].median()
df.loc[unknown_rows, 'Unemployment rate'] = df['Unemployment rate'].median()
df.loc[unknown_rows, 'Urban_population'] = df['Urban_population'].median()
andorra_row = df['Country'] == 'andorra'
df.loc[andorra_row, 'Population'] = 79824
df.loc[andorra_row, 'Gross tertiary education enrollment (%)'] = 85
df.loc[andorra_row, 'Unemployment rate'] = 1.90
df.loc[andorra_row, 'Urban_population'] = 68043
df.dropna(subset=['category','channel_type'],how='all',inplace=True)
df['category'].fillna(df['channel_type'],inplace=True)
df['category'].replace('Film','Film & Animation',inplace=True)
df['category'].replace('Games','Gaming',inplace=True)
df['category'].replace('Howto','Howto & Style',inplace=True)
df['category'].replace('People','People & Blogs',inplace=True)
df['category'].replace('Tech','Science & Technology',inplace=True)
df['channel_type'].fillna(df['category'],inplace=True)
df['channel_type'].replace('Film & Animation','Film',inplace=True)
df['channel_type'].replace('Gaming','Games',inplace=True)
df['channel_type'].replace('Howto & Style','Howto',inplace=True)
df['channel_type'].replace('People & Blogs','People',inplace=True)
df['channel_type'].replace('Science & Technology','Tech',inplace=True)
df['channel_type'].replace('Pets & Animals','Animals',inplace=True)
df['channel_type'].replace('Shows','Entertainment',inplace=True)
df['subscribers'].fillna(df['subscribers'].median(),inplace=True)
df['subscribers_for_last_30_days'].fillna(df['subscribers_for_last_30_days'].median(),inplace=True)
df['video_views_for_the_last_30_days'].fillna(df['video_views_for_the_last_30_days'].median(), inplace=True)
df.drop(columns=['Title','Country of origin', 'Abbreviation', 'video_views_rank', 'country_rank', 'channel_type_rank'], inplace=True)
```

```
print(df.isnull().sum())
df.reset_index(drop=True,inplace=True)
print(df.shape)
country=list(df['Country'].unique())
```

1. What are the top 10 YouTube channels based on the number of subscribers?

Solution:
```
df.sort_values(by='subscribers',ascending=False,inplace=True)
top_10=df[['Youtuber','subscribers']].head(10)
top_10.reset_index(drop=True,inplace=True)
top_10.index+=1
print("The top 10 youtube channels based on the number of subscribers are : ","\n" ,top_10)
```

2. Which category has the highest average number of subscribers?

Solution:
```
maximum=0
category_list=list(df['category'].unique())
avg_subscribers=[]
for category in df['category'].unique():
    rows=df[df['category']==category]
    avg_sub=rows['subscribers'].mean()
    avg_subscribers.append(avg_sub)
    if avg_sub>maximum:
        j=category
        maximum = avg_sub
dictionary3={'Category':category_list,'Average Subscribers':avg_subscribers}
df5=pd.DataFrame(dictionary3)
df5.index+=1
print(df5)
sns.catplot(x=category_list,y=avg_subscribers,data=df,kind='bar',height=15,aspect=2)
plt.xlabel('Category')
plt.ylabel('Average Subscribers')
plt.show()
print(f"{j} category has highest number of average subscribers")
```

3. How many videos, on average, are uploaded by YouTube channels in each category?

Solution:
```
upload=[]
for i in df['category'].unique():
    rows=df[df['category']==i]
    vid_upload=rows['uploads'].mean()
    upload.append(vid_upload)
cat=list(df['category'].unique())
dictionary1={'Category':cat,'Average video uploads':upload}
df2=pd.DataFrame(dictionary1)
df2.index+=1
print(df2)
sns.catplot(data=df2,x='Category',y='Average video uploads',kind='bar',height=15,aspect=2)
plt.xlabel('Category')
plt.ylabel('Average video uploads')
plt.show()
```

4. What are the top 5 countries with the highest number of YouTube channels?

Solution:
```
top_countries = df['Country'].value_counts().head(11)
top_countries_df = top_countries.reset_index()
top_countries_df.drop(2,inplace=True)
top_countries_df.reset_index(drop=True,inplace=True)
top_countries_df.index+=1
print("The top 5 countries with highest number of youtube channels are , "\n",
top_countries_df.head(5))
```

5. What is the distribution of channel types across different categories?

Solution:
```
print(df.groupby(['category','channel_type'])['channel_type'].count().head(40))
df8=df.groupby(['category','channel_type'])['channel_type'].count().tail(39)
print(df8)
print(df['category'].value_counts())
print(df['channel_type'].value_counts())
```

6. Is there a correlation between the number of subscribers and total video views for YouTube channels?

Solution:
```
corr_coeff_sub_videoviews = df['subscribers'].corr(df['video views'])
print(f"The correlation coefficient between number of subscribers and total video views is
{corr_coeff_sub_videoviews}")
sns.scatterplot(data=df,x='subscribers',y='video views')
plt.xlabel('Subscribers')
plt.ylabel('Total Video views')
plt.show()
```

7. How do the monthly earnings vary throughout different categories?

Solution:
```
monthly_earning=[]
df['Monthly Earnings']=(df['lowest_monthly_earnings']+df['highest_monthly_earnings'])/2
for i in df['category'].unique():
    rows=df[df['category']==i]
    earn=rows['Monthly Earnings'].mean()
    monthly_earning.append(earn)
category_list=list(df['category'].unique())
dictionary4={'Category':category_list,'Monthly Earning':monthly_earning}
df6=pd.DataFrame(dictionary4)
df6.index+=1
print(df6)
monthly_earnings_np = np.array(monthly_earning)
count = monthly_earnings_np.size
mean = np.mean(monthly_earnings_np)
std_dev = np.std(monthly_earnings_np)
minimum = np.min(monthly_earnings_np)
maximum = np.max(monthly_earnings_np)
median = np.median(monthly_earnings_np)
q1 = np.percentile(monthly_earnings_np, 25)
q3 = np.percentile(monthly_earnings_np, 75)
print(f'Count: {count}')
print(f'Mean: {mean}')
print(f'Standard Deviation: {std_dev}')
print(f'Minimum: {minimum}')
print(f'25th Percentile (Q1): {q1}')
print(f'Median (Q2): {median}')
print(f'75th Percentile (Q3): {q3}')
print(f'Maximum: {maximum}')
plt.figure(figsize=(20, 6))
```

```
sns.catplot(x=category_list,y=monthly_earning,data=df,kind='bar',height=14,aspect=2)
plt.xlabel('Category')
plt.ylabel('Monthly Earnings')
plt.show()
```

8. What is the overall trend in subscribers gained in the last 30 days across all channels?

Solution:
```
print(df['subscribers_for_last_30_days'].describe())
maxi=df['subscribers_for_last_30_days'].idxmax()
mini=df['subscribers_for_last_30_days'].idxmin()
max_sub_30=df.loc[maxi,'subscribers_for_last_30_days']
max_sub_30_youtuber=df.loc[maxi,'Youtuber']
min_sub_30=df.loc[mini,'subscribers_for_last_30_days']
min_sub_30_youtuber=df.loc[mini,'Youtuber']
print(f"The subscribers gained in last 30 days is highest for {max_sub_30_youtuber}({max_sub_30}).")
print(f"The subscribers gained in last 30 days is lowest for {min_sub_30_youtuber}({min_sub_30}).")
sns.lineplot(x='rank',y='subscribers_for_last_30_days',data=df)
plt.xlabel('Serial number of Youtube channels')
plt.ylabel('Subscribers gained in last 30 days')
plt.show()
```

9. Are there any outliers in terms of yearly earnings from YouTube channels?

Solution:
```
df['Yearly Earnings']=(df['lowest_yearly_earnings']+df['highest_yearly_earnings'])/2
plt.scatter(df['rank'],df['Yearly Earnings'],s=7)
plt.xlabel('Serial Number of Youtube channels')
plt.ylabel('Yearly Earnings')
plt.show()
plt.boxplot(df['Yearly Earnings'])
plt.title('Box plot of Yearly Earnings')
plt.show()
```

10. What is the distribution of channel creation dates? Is there any trend over time?

Solution:
```
print(df['created_date'].value_counts())
plt.hist(df['created_date'], bins=31, color='skyblue', edgecolor='black')
plt.xlabel('Created dates of youtube channels')
plt.ylabel('Number of youtube channels')
plt.show()
print(df['created_date'].describe())
```

11. Is there a relationship between gross tertiary education enrollment and the number of YouTube channels in a country?

Solution:
```
no_of_channels=[]
for i in df['Country'].unique():
    a = df['Country'].value_counts()[i]
    no_of_channels.append(a)
index4=country.index('finland')
index5=country.index('saudi arabia')
index6=country.index('andorra')
gross_enroll = list(df['Gross tertiary education enrollment (%)'].unique())
gross_enroll.insert(index5,68.0)
gross_enroll.insert(index6,85.0)
gross_enroll.insert(index4,88.2)
plt.scatter(x=gross_enroll,y=no_of_channels)
plt.xlabel('Gross tertiary education enrollment of country')
plt.ylabel('Number of youtube channels in country')
plt.show()
corr_coeff_gross_enrollment_channels=np.corrcoef(gross_enroll,no_of_channels)
print(f"The correlation coefficient between gross tertiary education enrollment rate and number of youtube channels in a country is {corr_coeff_gross_enrollment_channels[0,1]}")
```

12. How does the unemployment rate vary among the top 10 countries with the highest number of YouTube channels?

Solution:
```
top_countries = df['Country'].value_counts().head(11)
top_countries_df = top_countries.reset_index()
top_countries_df.drop(2,inplace=True)
top_countries_df.reset_index(drop=True,inplace=True)
top_countries_df.index+=1
topcoun=list(top_countries_df['Country'])
populationlist=[]
for i in topcoun:
    rows=df[df['Country']==i]
    population_of_country=rows['Population'].values[0]
    populationlist.append(population_of_country)
unemployment_rate=[]
for i in topcoun:
    rows=df[df['Country']==i]
    unemployment=rows['Unemployment rate'].values[0]
    unemployment_rate.append(unemployment)
data={'Country':topcoun,'Population':populationlist,'Unemployment rate':unemployment_rate}
df3=pd.DataFrame(data)
df3.index+=1
print(df3[['Country','Unemployment rate']])
sns.catplot(x='Country',y='Unemployment rate',data=df3,height=10,aspect=2,kind='bar')
plt.show()
```

13. What is the average urban population percentage in countries with YouTube channels?

Solution:
```
df['urban_pop_%']=(df['Urban_population']/df['Population'])*100
urban_population =list(df['urban_pop_%'].unique())
index2=country.index('singapore')
index3=country.index('Unknown')
urban_population.insert(index2,100.0)
urban_population.insert(index3,82.45899991756934)
print(f"The average urban population percentage in countries with youtube channels is {sum(urban_population)/len(urban_population)}")
```

14. Are there any patterns in the distribution of YouTube channels based on latitude and longitude coordinates?

Solution:
```
df['Location']=df['Latitude'].astype(str) + ',' + df['Longitude'].astype(str)
loc=df['Location'].value_counts()
loc_df=loc.reset_index()
loc_df.drop(2,inplace=True)
loc_df.reset_index(drop=True,inplace=True)
loc_df.index+=1
print(loc_df)
loc_df[['Latitude', 'Longitude']] = loc_df['Location'].str.split(',', expand=True)
loc_df['Latitude'] = loc_df['Latitude'].astype(float)
```

```
loc_df['Longitude'] = loc_df['Longitude'].astype(float)
plt.figure(figsize=(10, 6))
plt.scatter(loc_df['Longitude'], loc_df['Latitude'], s=loc_df['count'], alpha=0.6, edgecolors='w',
linewidth=0.5)
for i, row in loc_df.iterrows():
    plt.text(row['Longitude'], row['Latitude'], str(row['count']), fontsize=9, ha='right')
plt.title('Distribution of YouTube Channels by Location')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.grid(True)
plt.show()
```

15. What is the correlation between the number of subscribers and the population of a country?

Solution:

```
tot_sub=[]
for i in df['Country'].unique():
    rows=df[df['Country']==i]
    total_subscribers=rows['subscribers'].sum()
    tot_sub.append(total_subscribers)
population=list(df['Population'].unique())
population.insert(2, df['Population'].median())
corr_coeff_totalsub_population = np.corrcoef(tot_sub,population)
print(f"The correlation coefficient between number of subscribers and population of a country is
{corr_coeff_totalsub_population[0,1]}")
plt.scatter(x=tot_sub,y=population)
plt.xlabel('Total subscribers in a country')
plt.ylabel('Population of country')
plt.show()
```

16. How do the top 10 countries with the highest number of YouTube channels compare in terms of their total population?

Solution:

```
top_countries = df['Country'].value_counts().head(11)
top_countries_df = top_countries.reset_index()
top_countries_df.drop(2,inplace=True)
top_countries_df.reset_index(drop=True,inplace=True)
top_countries_df.index+=1
topcoun=list(top_countries_df['Country'])
populationlist=[]
for i in topcoun:
    rows=df[df['Country']==i]
    population_of_country=rows['Population'].values[0]
    populationlist.append(population_of_country)
unemployment_rate=[]
for i in topcoun:
    rows=df[df['Country']==i]
    unemployment=rows['Unemployment rate'].values[0]
    unemployment_rate.append(unemployment)
data={'Country':topcoun,'Population':populationlist,'Unemployment rate':unemployment_rate}
df3=pd.DataFrame(data)
df3.index+=1
print(df3[['Country','Population']])
sns.catplot(x='Country',y='Population',data=df3,kind='bar',height=10,aspect=2)
```

```
plt.show()
```

17. Is there a correlation between the number of subscribers gained in the last 30 days and the unemployment rate in a country?

Solution:
```
sub_30_days=[]
for i in df['Country'].unique():
    rows=df[df['Country']==i]
    subscribers_last_30_days=rows['subscribers_for_last_30_days'].sum()
    sub_30_days.append(subscribers_last_30_days)
index=sub_30_days.index(201601.0)
unemployment_rate=list(df['Unemployment rate'].unique())
unemployment_rate.insert(index,4.11)
index1=sub_30_days.index(1230010.0)
unemployment_rate.insert(index1,8.88)
plt.scatter(x=unemployment_rate,y=sub_30_days)
plt.xlabel('Unemployment rate of country')
plt.ylabel('Subscribers gained in last 30 days in country')
plt.show()
corr_coeff_subscribers_30_days=np.corrcoef(unemployment_rate,sub_30_days)
print(f"The correlation coefficient between number of subscribers for last 30 days and unemployment rate in a country is {corr_coeff_subscribers_30_days[0,1]}")
```

18. How does the distribution of video views for the last 30 days vary across different channel types?

Solution:
```
video_views_30_days=[]
for i in df['channel_type'].unique():
    rows=df[df['channel_type']==i]
    video_views_30=rows['video_views_for_the_last_30_days'].mean()
    print(i,video_views_30)
    video_views_30_days.append(video_views_30)
channel_type=list(df['channel_type'].unique())
dictionary2={'Channel type':channel_type,'Video views for last 30 days': video_views_30_days}
df4=pd.DataFrame(dictionary2)
df4.index+=1
print(df4)
sns.catplot(x=channel_type,y=video_views_30_days,data=df,kind='bar',height=15,aspect=2)
plt.xlabel('Channel types')
plt.ylabel('Video Views in last 30 days')
plt.show()
```

19. Are there any seasonal trends in the number of videos uploaded by YouTube channels?

Solution: Could not do it.

20. What is the average number of subscribers gained per month since the creation of YouTube channels till now?

Solution:
```
df['months'] = (2024 - (df['created_year']+1))*12
month_value = {
  'Jan': 12,
  'Feb': 11,
  'Mar': 10,
  'Apr': 9,
  'May': 8,
  'Jun': 7,
  'Jul': 6,
    'Aug': 5,
    'Sep': 4,
    'Oct': 3,
    'Nov': 2,
    'Dec': 1
}
def adjust_month(row):
    if row['created_month'] in month_value:
        return row['months'] + month_value[row['created_month']]
    return row['months']
df['months'] = df.apply(adjust_month, axis=1)
df['months']=df['months']+5
df['sub_per_mon']=df['subscribers']/df['months']
print(f"The average number of subscribers gained per month since the creation of YouTube channels till now is {df['sub_per_mon'].mean()}")
```