LINK TO THE PROJECT

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#PREPROCESSING
df = pd.read_csv("Global YouTube Statistics.csv", encoding="latin-1")

df.replace(['nan', '', 'NAN', 'NaN'], np.nan, inplace=True)

df.columns = (
   df.columns
   .str.strip()
   .str.lower()
   .str.replace(" ", "_")
)

numeric_cols = [
   'subscribers', 'video_views', 'uploads', 'video_views_rank',
'country_rank', 'channel_type_rank',
   'video_views_for_the_last_30_days', 'lowest_monthly_earnings',
'highest_monthly_earnings',
   'lowest_yearly_earnings', 'highest_yearly_earnings',
'subscribers_for_last_30_days',
   'Gross_tertiary_education_enrollment_(%)', 'Population',
'Unemployment_rate', 'Urban_population',
   'Latitude', 'Longitude'
]
for col in numeric_cols:
   if col in df.columns:
       df[col] = pd.to_numeric(df[col], errors='coerce')


#1. What are the top 10 YouTube channels based on the number of
subscribers?
ans1 = df.head(10)[['youtuber', 'subscribers']]
print(ans1)
#2. Which category has the highest average number of subscribers?
```

```
ans =
df.groupby('category')['subscribers'].mean().sort_values(ascending=False)
print(ans.head(1))
#3. How many videos, on average, are uploaded by YouTube channels in each
category?
num_vids_avg = df.groupby('category')['uploads'].mean()
print(num_vids_avg)

num_vids_avg.plot.bar(color='plum')
plt.title("Average Videos Uploaded by Category")
plt.xlabel("categories")
plt.ylabel("Average Number of Uploads")
plt.show()


#4. What are the top 5 countries with the highest number of YouTube
channels?
top_5_countries =
df.groupby('country')['youtuber'].count().sort_values(ascending=False).hea
d(5)
print(top_5_countries)
#5. What is the distribution of channel types across different categories?
channel_dist = pd.crosstab(df['category'], df['channel_type'])
print("Distribution of channel types across different categories\n\n",
channel_dist)
channel_dist.plot(kind='bar', stacked=True, figsize=(10, 6), color='plum')
plt.title('Distribution of Channel Types Across Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.legend(title='Channel Type', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.show()

#6. Is there a correlation between the number of subscribers and total
video views for YouTube channels?
sub_view =
df.groupby(['category'])['youtuber'].count().sort_values(ascending=False)
print(sub_view)
output = df["subscribers"].corr(df['video_views'])
```

```
print("\n\ncorrelation between the number of subscribers and total video
views for YouTube channels: ", output)
#7. How do the monthly earnings vary throughout different categories?

monthly_earnings = df.groupby("category")[["lowest_monthly_earnings",
"highest_monthly_earnings"]].mean().sort_values("lowest_monthly_earnings",
ascending=False)

categories = monthly_earnings.index
lowest = monthly_earnings["lowest_monthly_earnings"]
highest = monthly_earnings["highest_monthly_earnings"]

x = np.arange(len(categories))
width = 0.45

plt.figure(figsize=(12,6))
plt.bar(x - width/2, lowest, width, label='Lowest Monthly Earnings',
color='#FFB6C1')
plt.bar(x + width/2, highest, width, label='Highest Monthly Earnings',
color='#C8A2C8')

plt.ylabel('Earnings ($)')
plt.xlabel('Category')
plt.title('Average Monthly Earnings by Category')
plt.xticks(x, categories, rotation=45, ha='right')
plt.legend()
plt.show()

#8. What is the overall trend in subscribers gained in the last 30 days
across all channels?
subs_30_days = df['subscribers_for_last_30_days']
overall_avg = subs_30_days.mean()
print("Average subscribers gained in the last 30 days across all
channels:", overall_avg)

plt.hist(subs_30_days, bins=50, color='#C8A2C8', edgecolor='#4B0082')
plt.title('Distribution of Subscribers Gained in Last 30 Days')
plt.xlabel('Subscribers Gained')
plt.ylabel('Number of Channels')
plt.show()
```

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

q1 = df['highest_yearly_earnings'].quantile(0.25)
q3 = df['highest_yearly_earnings'].quantile(0.75)
iqr = q3 - q1

lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

high_outliers = df[df['highest_yearly_earnings'] >
upper_bound][['youtuber', 'highest_yearly_earnings']]
print("Outliers with unusually high earnings:\n", high_outliers)

low_outliers = df[df['highest_yearly_earnings'] <
lower_bound][['youtuber', 'highest_yearly_earnings']]
print("Outliers with unusually low earnings:\n", low_outliers)

#10. What is the distribution of channel creation dates? Is there any
trend over time?
creation_trend =
df['created_year'].value_counts().sort_index(ascending=True)
print("Channels created by year:\n", creation_trend)

plt.plot(creation_trend.index, creation_trend.values, marker='o',
color='#C8A2C8')
plt.title("Distribution of YouTube Channel Creation by Year")
plt.xlabel("Year")
plt.ylabel("Number of Channels Created")
plt.show()
#11. Is there a relationship between gross tertiary education enrollment
and the number of YouTube channels in a country?
channels_per_country = df['country'].value_counts()
edu_per_country =
df.groupby('country')['gross_tertiary_education_enrollment_(%)'].mean()

combined_df = pd.concat([channels_per_country, edu_per_country],
axis=1).dropna()
combined_df.columns = ['channel_count', 'education_enrollment']
```

```
plt.figure(figsize=(10,6))
plt.scatter(combined_df['education_enrollment'],
combined_df['channel_count'], color='#C8A2C8', edgecolor='#4B0082')
plt.title('YouTube Channels Count vs Gross Tertiary Education Enrollment
by Country')
plt.xlabel('Gross Tertiary Education Enrollment (%)')
plt.ylabel('Number of YouTube Channels')
plt.grid(alpha=0.3)
plt.show()
#12. How does the unemployment rate vary among the top 10 countries with
the highest number of YouTube channels?
top10_countries = df['country'].value_counts().head(10)
print("Top 10 countries with highest number of YouTube channels:\n",
top10_countries)
unemployment_rate = df.groupby('country')['unemployment_rate'].mean()
unemployment_top10 = unemployment_rate.loc[top10_countries.index]
print("\nUnemployment rate for top 10 countries:\n", unemployment_top10)

plt.bar(unemployment_top10.index, unemployment_top10.values,
color='#C8A2C8')
plt.title('Unemployment Rate in Top 10 Countries with Most YouTube
Channels')
plt.xlabel('Country')
plt.ylabel('Average Unemployment Rate (%)')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', alpha=0.3)
plt.show()

#13. What is the average urban population percentage in countries with
YouTube channels?
avg_urb_pop_all = df["urban_population"].mean()
print("Average urban population percentage across all countries with
YouTube channels:", avg_urb_pop_all)

avg_urb_pop_per_country = df.groupby('country')['urban_population'].mean()
print("\nAverage urban population percentage per country with YouTube
channels:\n", avg_urb_pop_per_country)
```

```
#14. Are there any patterns in the distribution of YouTube channels based on latitude and longitude coordinates?
country_cords = df.groupby('country')[['latitude', 'longitude']].mean().sort_values(by=['latitude', 'longitude'], ascending=False)
import seaborn as sns

plt.figure(figsize=(12,6))
plt.scatter(df['longitude'], df['latitude'], color='violet', alpha=0.5, s=10)  # lilac-like color
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Distribution of YouTube Channels by Coordinates')
plt.grid()
plt.show()


#15. What is the correlation between the number of subscribers and the population of a country?
subscribers_per_country = df.groupby("country")["subscribers"].sum()
population_per_country = df.groupby("country")["population"].first()
num_sub = pd.DataFrame({
   "subscribers": subscribers_per_country,
   "population": population_per_country
})

correlation = num_sub["subscribers"].corr(num_sub["population"])
print("Correlation between subscribers and population:", correlation)

plt.scatter(num_sub["population"], num_sub["subscribers"], color='#C8A2C8', alpha =0.5)
plt.xlabel("Population of Country")
plt.ylabel("Total Subscribers in Country")
plt.title("Subscribers vs Population per Country")
plt.grid(alpha=0.3)
plt.show()

#16. How do the top 10 countries with the highest number of YouTube channels compare in terms of their total population
top_10_countries = df['country'].value_counts().head(10)
```

```
top_10_population =
df.groupby('country')['population'].sum().loc[top_10_countries.index]
top_10_summary = pd.DataFrame({
   'Number of Channels': top_10_countries,
   'Total Population': top_10_population
})
print("top 10 countries with the highest number of YouTube channels and
their their total population", top_10_summary)

top_10_summary.plot(kind='bar', figsize=(10,6), color=['#C8A2C8'])
plt.ylabel("Count / Population")
plt.title("Top 10 Countries: YouTube Channels vs Total Population")
plt.xticks(rotation=45)
plt.grid(alpha=0.3)
plt.tight_layout()
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?
sub_unemp = df.groupby('country')[['subscribers_for_last_30_days',
'unemployment_rate']].mean().sort_values(by=['subscribers_for_last_30_days
', 'unemployment_rate'], ascending=False)
print("number of subscribers gained in the last 30 days and the
unemployment rate in a country:", sub_unemp)

plt.scatter(sub_unemp['unemployment_rate'],
sub_unemp['subscribers_for_last_30_days'], color='plum')
plt.xlabel('Unemployment Rate')
plt.ylabel('Subscribers Gained in Last 30 Days')
plt.title('Subscribers vs Unemployment Rate by Country')
plt.show()
#18. How does the distribution of video views for the last 30 days vary
across different channel types?
video_view_dist =
df.groupby('channel_type')['video_views_for_the_last_30_days'].mean().sort
_values(ascending=False)
print("Distribution of video views for the last 30 days vary across
different channel types:\n", video_view_dist)
```

```
correlation =
sub_unemp['subscribers_for_last_30_days'].corr(sub_unemp['unemployment_rat
e'])
print("Correlation between subscribers gained in 30 days and unemployment
rate:", correlation)

plt.figure(figsize=(12,6))
df.boxplot(column='video_views_for_the_last_30_days', by='channel_type',
grid=False, patch_artist=True,
          boxprops=dict(facecolor='lightblue'),
medianprops=dict(color='red'))

plt.title('Distribution of Video Views in Last 30 Days by Channel Type')
plt.suptitle('')
plt.xlabel('Channel Type')
plt.ylabel('Video Views in Last 30 Days')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
#19. Are there any seasonal trends in the number of videos uploaded by
YouTube channels?

video_upload_trend = df.groupby('created_month')['uploads'].mean()
video_upload_trend = video_upload_trend.sort_index()

video_upload_trend_sorted =
video_upload_trend.sort_values(ascending=False)
print("\nSeasonal trends (sorted by uploads):\n",
video_upload_trend_sorted)

plt.figure(figsize=(10,5))
plt.plot(video_upload_trend.index, video_upload_trend.values, marker='o',
color='plum')
plt.xlabel('Month')
plt.ylabel('Average Number of Uploads')
plt.title('Seasonal Trends in Video Uploads by YouTube Channels')
plt.grid(alpha=0.3)
plt.show()
#20
import pandas as pd
```

```
import matplotlib.pyplot as plt

df['created_year'] = df['created_year'].astype(str).str.replace('.0', '',
regex=False)
df['created_year'] = pd.to_numeric(df['created_year'],
errors='coerce').astype('Int64')

df['created_month'] = df['created_month'].astype(str)
df['created_month'] = df['created_month'].replace(['nan', 'NaN', 'NAN'],
pd.NA)

df['created_datetime'] = pd.to_datetime(
   df['created_year'].astype(str) + " " + df['created_month'].astype(str)
+ " 1",
   format='%Y %b %d',
   errors='coerce'
)

today = pd.Timestamp.today()
df['months_since_creation'] = ((today.year -
df['created_datetime'].dt.year) * 12 +
                               (today.month -
df['created_datetime'].dt.month))

df['months_since_creation'] = df['months_since_creation'].replace(0, 1)

df['avg_subs_per_month'] = df['subscribers'] / df['months_since_creation']

overall_avg = df['avg_subs_per_month'].mean(skipna=True)
print("Average number of subscribers gained per month since channel
creation:", overall_avg)

plt.figure(figsize=(10,6))
plt.hist(df['avg_subs_per_month'].dropna(), bins=50, color='plum',
edgecolor='white')
plt.title("Distribution of Average Subscribers Gained per Month")
plt.xlabel("Average Subscribers per Month")
plt.ylabel("Number of Channels")
plt.legend()
plt.show()
```