Project Name: TED Talk Analysis Using Python

Objective

The objective of this project is to analyze TED Talk data to uncover insights into viewership patterns, speaker information, and talk themes. The analysis aims to:

- Identify the most viewed TED Talks of all time.
- Explore the distribution and correlation of views and comments.
- Investigate the impact of talk duration and speaker occupation on viewership.
- Analyze TED Talk trends based on publication date, event type, and languages.

Data Source

The dataset used in this project is ted_main.csv, which includes various details about TED Talks, such as titles, descriptions, main speakers, views, comments, and tags.

Summary

The dataset was initially cleaned and transformed to convert Unix timestamps into human-readable dates. Key analyses performed include:

1. **Most Viewed Talks**:
   - Identified the top 15 most viewed TED Talks and visualized their view counts using a bar chart.
2. **Distribution of Views**:
   - Analyzed the distribution of views with histograms and summary statistics.
3. **Comments Analysis**:
   - Investigated the distribution of comments and their correlation with views.
   - Analyzed the most commented TED Talks.
4. **Discussion Quotient**:
   - Calculated the discussion quotient (ratio of comments to views) and identified talks with the highest quotients.
5. **Monthly and Yearly Analysis**:
   - Analyzed the number of TED Talks released each month and year.
   - Investigated trends based on the day of the week and the most popular months for TED and TEDx events.
6. **Heat Map**:
   - Constructed a heat map to visualize the number of TED Talks by month and year.

7. **Speaker and Occupation Analysis**:
   - Analysed the number of talks by main speakers and their occupations.
   - Investigated if certain occupations attract more viewers.
8. **TED Events**:
   - Identified TED events with the highest number of talks.
9. **Languages**:
   - Explored the number of languages in which TED Talks are available and their correlation with views.
10. **Themes**:
    - Analysed TED Talk themes and their popularity.
    - Visualized trends in themes over the years.
11. **Talk Duration**:
    - Examined the relationship between talk duration and views.

Results

- **Most Viewed Talks**: The top TED Talks were dominated by popular speakers with high view counts. The bar chart visualization highlighted the most influential talks.
- **Views Distribution**: The histograms showed that the majority of talks have a moderate number of views, with a few outliers having extremely high view counts.
- **Comments and Views**: There is a positive correlation between the number of comments and views, indicating that more popular talks tend to receive more comments.
- **Discussion Quotient**: The talks with the highest discussion quotient often had a high level of engagement relative to their view counts.
- **Monthly/Yearly Trends**: Certain months and years showed spikes in the number of TED Talks, with notable trends in TED and TEDx events.
- **Speaker and Occupation Insights**: Some professions attract more viewers, and the analysis provided insights into which speakers and occupations are most prominent.
- **Languages**: TED Talks are available in numerous languages, with varying popularity across different languages.
- **Themes**: Popular themes included a variety of topics, with some themes showing consistent interest over the years.
- **Duration**: The analysis revealed that talk duration has a nuanced impact on viewership, with optimal durations for maximizing views.

Conclusion

The TED Talk analysis provided valuable insights into viewership patterns, speaker impact, and thematic trends. It highlighted the importance of speaker engagement, the relationship between comments and views, and the influence of talk duration on audience reach. The visualizations and statistical analyses helped to understand the dynamics of TED Talks and their global impact.

Code:

```
import pandas as pd

import numpy as np

from scipy import stats

import matplotlib.pyplot as plt

import seaborn as sns

import json

from pandas import json_normalize

from wordcloud import WordCloud, STOPWORDS

import warnings

warnings.simplefilter(action='ignore', category=FutureWarning)


month_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',

'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

day_order = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']

df = pd.read_csv("D:\\Aditya's Notes\\Aditya's Data Science Notes\\Projects and Other
Datasets\\Dataset-master\\ted_data\\ted_main.csv")

df.columns


#I'm just going to reorder the columns in the order I've

#listed the features for my convenience (and OCD).

df = df[['name', 'title', 'description', 'main_speaker',

'speaker_occupation', 'num_speaker', 'duration', 'event',

'film_date', 'published_date', 'comments', 'tags',

'languages', 'ratings', 'related_talks', 'url', 'views']]

#Before we go any further, let us convert the Unix timestamps

#into a human readable format.


import datetime

df['film_date'] = df['film_date'].apply(lambda x:

datetime.datetime.fromtimestamp( int(x)).strftime('%d-%m-%Y'))
```

```
df['published_date'] = df['published_date'].apply(lambda x:
datetime.datetime.fromtimestamp( int(x)).strftime('%d-%m-%Y'))
print(df.head())

#Analysis 1: Most Viewed Talks of All Time
pop_talks = df[['title', 'main_speaker', 'views', 'film_date']].sort_values('views', ascending=False)[:15]
print(pop_talks)
#2. Let us make a bar chart to visualise these 15 talks in
#terms of the number of views they garnered.
pop_talks['abbr'] = pop_talks['main_speaker'].apply(lambda x:
x[:3])
sns.set_style("whitegrid")
plt.figure(figsize=(10,6))
sns.barplot(x='abbr', y='views', data=pop_talks)
#Analysis 3: let us investigate the summary statistics and
#the distibution of the views garnered on various TED Talks.
sns.displot(df['views'])
sns.displot(df[df['views'] < 0.4e7]['views'])

#Analysis 4:
df['views'].describe()
#Analysis 5: Performing textual analysis of comments
df['comments'].describe()
sns.displot(df['comments'])
sns.displot(df[df['comments'] < 500]['comments'])
#Analysis 6: if the number of views is correlated with the
#number of comments. We should think that this is the case as
#more popular videos tend to have more comments.
sns.jointplot(x='views', y='comments', data=df)
df[['views', 'comments']].corr()
#Analysis 7: Let us now check the number of views and
```

```
#comments on the 10 most commented TED Talks of all time

df[['title', 'main_speaker','views', 'comments']].sort_values('comments', ascending=False).head(10)


#discussion quotient which is simply the ratio of the number

#of comments to the number of views

df['dis_quo'] = df['comments']/df['views']

df[['title', 'main_speaker','views', 'comments', 'dis_quo', 'film_date']].sort_values('dis_quo',
ascending=False).head(10)


###Analysing TED Talks by the month and the year

df['month'] = df['film_date'].apply(lambda x:

month_order[int(x.split('-')[1]) - 1])

month_df = pd.DataFrame(df['month'].value_counts()).reset_index()

month_df.columns = ['month', 'talks']

sns.barplot(x='month', y='talks', data=month_df, order=month_order)

df_x = df[df['event'].str.contains('TEDx')]

x_month_df = pd.DataFrame(df_x['month'].value_counts().reset_index())

x_month_df.columns = ['month', 'talks']

sns.barplot(x='month', y='talks', data=x_month_df,order=month_order)


##the most popular days for conducting TED and TEDx conferences


def getday(x):

 day, month, year = (int(i) for i in x.split('-'))

 answer = datetime.date(year, month, day).weekday()

 return day_order[answer]

df['day'] = df['film_date'].apply(getday)

day_df = pd.DataFrame(df['day'].value_counts()).reset_index()

day_df.columns = ['day', 'talks']

sns.barplot(x='day', y='talks', data=day_df, order=day_order)

#Let us now visualize the number of TED talks through the years
```

```
df['year'] = df['film_date'].apply(lambda x: x.split('-')[2])

year_df = pd.DataFrame(df['year'].value_counts().reset_index())

year_df.columns = ['year', 'talks']

plt.figure(figsize=(18,5))

sns.pointplot(x='year', y='talks', data=year_df)


#let us construct a heat map that shows us the number of talks by month and year.

months = {'Jan': 1, 'Feb': 2, 'Mar': 3, 'Apr': 4, 'May': 5,

'Jun': 6, 'Jul': 7, 'Aug': 8, 'Sep': 9, 'Oct': 10, 'Nov': 11,

'Dec': 12}

hmap_df = df.copy()

hmap_df['film_date'] = hmap_df['film_date'].apply(lambda x:month_order[int(x.split('-')[1]) - 1] + " "
+ str(x.split('-')[2]))

hmap_df = pd.pivot_table(hmap_df[['film_date', 'title']],index='film_date',
aggfunc='count').reset_index()

hmap_df['month_num'] = hmap_df['film_date'].apply(lambda x:months[x.split()[0]])

hmap_df['year'] = hmap_df['film_date'].apply(lambda x:x.split()[1])

hmap_df = hmap_df.sort_values(['year', 'month_num'])

hmap_df = hmap_df[['month_num', 'year', 'title']]

hmap_df = hmap_df.pivot(index='month_num', columns='year', values='title') #**error resolved**

hmap_df = hmap_df.fillna(0)

f, ax = plt.subplots(figsize=(12, 8))

sns.heatmap(hmap_df, annot=True, linewidths=.5, ax=ax,fmt='n', yticklabels=month_order)


##TED Speakers

speaker_df = df.groupby('main_speaker').count().reset_index()[['main_speaker', 'comments']]

speaker_df.columns = ['main_speaker', 'appearances']

speaker_df = speaker_df.sort_values('appearances', ascending=False)

speaker_df.head(10)


occupation_df = df.groupby('speaker_occupation').count().reset_index()[['speaker_occupation',
'comments']]
```

```
occupation_df.columns = ['occupation', 'appearances']
occupation_df = occupation_df.sort_values('appearances', ascending=False)

#Do some professions tend to attract a larger number of viewers?
fig, ax = plt.subplots(nrows=1, ncols=1,figsize=(15, 8))
sns.boxplot(x='speaker_occupation', y='views',
data=df[df['speaker_occupation'].isin(occupation_df.head(10)['occupation'])], palette="muted", ax
=ax)
ax.set_ylim([0, 0.4e7])
plt.show()

#Finally, let us check the number of talks which have had more than one speaker.
df['num_speaker'].value_counts()
df[df['num_speaker'] == 5][['title', 'description',
'main_speaker', 'event']]

#TED Events
#Which TED Events tend to hold the most number of TED.com upload worthy events?
events_df = df[['title','event']].groupby('event').count().reset_index()
events_df.columns = ['event', 'talks']
events_df = events_df.sort_values('talks', ascending=False)
events_df.head(10)



#TED Languages
#One remarkable aspect of TED Talks is the sheer number of languages in which it is accessible.

df['languages'].describe()
df[df['languages'] == 72]
sns.jointplot(x='languages', y='views', data=df)
plt.show()
```

```
#TED Themes

import ast

df['tags'] = df['tags'].apply(lambda x: ast.literal_eval(x))

s = df.apply(lambda x:

pd.Series(x['tags']),axis=1).stack().reset_index(level=1, drop=True)

s.name = 'theme'

theme_df = df.drop('tags', axis=1).join(s)

theme_df.head()


len(theme_df['theme'].value_counts())

pop_themes = pd.DataFrame(theme_df['theme'].value_counts()).reset_index()

pop_themes.columns = ['theme', 'talks']

pop_themes.head(10)

plt.figure(figsize=(15,5))

sns.barplot(x='theme', y='talks', data=pop_themes.head(10))

plt.show()


themes = list(pop_themes.head(8)['theme'])

themes.remove('TEDx')

pop_theme_talks = theme_df[theme_df['theme'].isin(pop_themes.head(10)['theme'])]

ctab = pd.crosstab([pop_theme_talks['year']], pop_theme_talks['theme']).apply(lambda x: x/x.sum(),
axis=1)

ctab[themes].plot(kind='bar', stacked=True,

colormap='rainbow', figsize=(12,8)).legend(loc='center left',

bbox_to_anchor=(1, 0.5))

plt.show()

ctab[themes].plot(kind='line', stacked=False,

colormap='rainbow', figsize=(12,8)).legend(loc='center left',

bbox_to_anchor=(1, 0.5))
```

```
plt.show()


fig, ax = plt.subplots(nrows=1, ncols=1,figsize=(15, 8))

sns.boxplot(x='theme', y='views', data=pop_theme_talks, palette="muted", ax =ax)

ax.set_ylim([0, 0.4e7])


#Talk Duration and Word Counts¶

#Convert to minutes

df['duration'] = df['duration']/60

df['duration'].describe()

df[df['duration'] == 2.25]

df[df['duration'] == 87.6]

sns.jointplot(x='duration', y='views', data=df[df['duration']

< 25])

plt.xlabel('Duration')

plt.ylabel('Views')

plt.show()
```