```
#!/usr/bin/env python
# coding: utf-8

# ## Business Problem.
#
# **- Microsoft sees all the big companies creating original video content and they want to get in on the fun.**
#
# **- They have decided to create a new movie studio, but they donâ  t know anything about creating movies.**
#
# **- You are charged with exploring what types of films are currently doing the best at the box office.**
#
# **- You must then translate those findings into actionable insights that the head of Microsoft's new movie studio can use to help decide what type of films to create.**

# ## Data Collection.
# **- After analyzing the business problem,I have decided to determine which movies made the most profit at the box office and use the profit as a measure of the movies' performance.**
#
# **-With that,I have obtained the 'movie_gross' dataset(already availed by the institution after web scrapping) which I am going to use for my analysis.**
#
# **-This dataset gives information on the revenue generated from different movies in the time period 2010-2018.**
#

# ## Dataset Overview and Transformation.
# **- I now want to proceed on with loading,cleaning and transforming the dataset so as to make it ready for data analysis.**
#

# In[171]:


## Import relevant libraries.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')


# In[181]:


# Load the 'movie_gross' dataset into the notebook in form of a dataframe.
movie_gross=pd.read_csv('C:/Users/dv/Desktop/Moringa School/Project 1-Exploratory Data Analysis/bom.movie_gross.csv')


# In[182]:


#Preview the movie_gross df to ensure that it loaded correctly.
movie_gross.head()


# In[183]:


#Obtain information about the movie_gross df.
movie_gross.info()


# In[184]:


# Calculate the percentage of null values in each column
null_counts=movie_gross.isnull().sum()
total_values = movie_gross.size
percentage_null = (null_counts / total_values) * 100
# Display the results
print(percentage_null)


# In[185]:


# Drop the null values as they are insignificant to the overrall size of tge dataset.
movie_gross=movie_gross.dropna()
movie_gross.isna().sum()


# In[186]:


#Update the column names of 'domestic_gross' and 'foreign_gross' columns to indicate that they contain currency values.
movie_gross.rename(columns={'domestic_gross': 'domestic_gross($)'}, inplace=True)
movie_gross.rename(columns={'foreign_gross': 'foreign_gross($)'}, inplace=True)


# In[187]:


#Convert the data values in the 'domestic_gross' and 'foreign_gross' columns into numerical values.
movie_gross['domestic_gross($)'] = pd.to_numeric(movie_gross['domestic_gross($)'], errors='coerce')
movie_gross['foreign_gross($)'] = pd.to_numeric(movie_gross['foreign_gross($)'], errors='coerce')


# In[188]:


#Feature engineer a 'Total_gross' column by adding the 'domestic_gross' column and the 'foreign_column'.
movie_gross['Total_gross($)'] = movie_gross['domestic_gross($)'] + movie_gross['foreign_gross($)']


# In[190]:


#Format the 'Total_gross' column .
movie_gross['Total_gross($)'] = movie_gross['Total_gross($)'].map('{:,.2f}'.format)

# Remove commas from the 'Total_gross($)' column
movie_gross['Total_gross($)'] = movie_gross['Total_gross($)'].str.replace(',', '')


# In[191]:


#I would have loved to list out the studio names in full, however,I am not familiar with the studios.

# I also wanted to check if there exists an outlier in the year column.But no outlier exists.
movie_gross['year'].unique()


# In[197]:
```

```
# Convert the column name "title" to "movie_title" so as the column name is more intuitive.
movie_gross = movie_gross.rename(columns={'title': 'movie_title'})


# In[196]:


# Convert the column name "year" to "release_year" so as the column name is more intuitive.
movie_gross = movie_gross.rename(columns={'year': 'release_year'})


# In[194]:


#set the movie_title column as the index of the data frame.
movie_gross=movie_gross.set_index('movie_title')


# In[198]:


#For aesthetics,I capitalized the column names.
movie_gross.columns = movie_gross.columns.str.capitalize()


# In[199]:


movie_gross['Total_gross($)'] = movie_gross['Total_gross($)'].astype(float)


# In[200]:


pd.set_option('display.float_format', '{:.2f}'.format)


# In[201]:


## Preview the cleaned and transformed 'movie_gross' dataset to ensure that it has been formatted upto standard.
movie_gross.head()


# ## Performing EDA On The Cleaned and Transformed Dataset.

# In[202]:


# Compute summary statistics for the currency columns.This gives a general overview of the data at a glance.
selected_columns = ['Domestic_gross($)', 'Foreign_gross($)', 'Total_gross($)']
summary = movie_gross[selected_columns].describe()
summary


# In[203]:


#Top 1% Movies in term of Total gross income generation.WHATS THE MOST DOMINANT GENRE OF THESE MOVIES??
movie_gross= movie_gross.reset_index()

# Sort the movies by total gross income in descending order
sorted_df = movie_gross.sort_values(by='Total_gross($)', ascending=False)

# Calculate the number of movies that make up the top 10%
total_movies = len(sorted_df)
top_1_percent = int(total_movies * 0.010)

# Select the top 10% earning movies
top_1_percent_movies = sorted_df.head(top_1_percent)

# Display the top 10% earning movies
top_1_percent_movies


# In[159]:


## Visualized the top_1_percent movies using a barh graph...Its clear to see that income was majorly from the foreign market.

# Set the figure size
plt.figure(figsize=(12, 6))

# Create the bahr graph
plt.barh(top_1_percent_movies['movie_title'],top_1_percent_movies ['Domestic_gross($)'], label='Domestic Gross', color='b')
plt.barh(top_1_percent_movies['movie_title'], top_1_percent_movies['Foreign_gross($)'], label='Foreign Gross', color='g', left=top_1_percent_movies['Domestic_gross($)'])

# Set the labels and title
plt.xlabel('Gross Income ($)')
plt.ylabel('Movie Title')
plt.title('Top Movies by Domestic and Foreign Gross Income')

# Show the legend
plt.legend()

# Show the plot
plt.show()


# In[204]:


#A glance at the movie tittles of the top_1_percent movies.Using my domain knowledge,I was able notice a sequal&pre-sequel trend in the movies at this category.
top_1_percent_movies['movie_title'].sort_values()


# In[205]:


#Bottom 1% Movies in terms of total gross revenue generation.WHATS THE MOST DOMINANT GENRE IN THESE MOVIES??

# Sort the movies by total gross income in descending order
sorted_df = movie_gross.sort_values(by='Total_gross($)', ascending=True)

# Calculate the number of movies that make up the top 10%
total_movies = len(sorted_df)
bottom_1_percent = int(total_movies * 0.010)

# Select the top 10% earning movies
```

```
bottom_1_percent_movies = sorted_df.head(bottom_1_percent)

# Display the top 10% earning movies
bottom_1_percent_movies


# In[157]:


#A glance at the movie tittles of the top_1_percent movies.Using domain knowledge,I was able to note that this category does not contain any sequel or pre-sequel movies.
bottom_1_percent_movies['movie_title'].sort_values()


# In[206]:


##Visualized the bottom_1_percent movies using a barh graph.
##It is clear to see that majority of the income was from the Foreign market.

# Set the figure size
plt.figure(figsize=(12, 6))

# Create the barh graph
plt.barh(bottom_1_percent_movies['movie_title'],bottom_1_percent_movies ['Domestic_gross($)'], label='Domestic Gross', color='b')
plt.barh(bottom_1_percent_movies['movie_title'], bottom_1_percent_movies['Foreign_gross($)'], label='Foreign Gross', color='g', left=bottom_1_percent_movies['Domestic_gross($)'])

# Set the labels and title
plt.xlabel('Gross Income ($)')
plt.ylabel('Movie Title')
plt.title('Bottom Movies by Domestic and Foreign Gross Income')

# Show the legend
plt.legend()

# Show the plot
plt.show()


# In[207]:


#group the movie_gross data frame by 'release_year' column.
grouped_by_year = movie_gross.groupby('Release_year')


# In[120]:


# Calculate the mean values of the currency columns in the 'grouped_by_year' dataframe.
grouped_by_year_stats = grouped_by_year['Domestic_gross($)','Foreign_gross($)','Total_gross($)'].mean()
print(grouped_by_year_stats)


# In[122]:


##Display the 'grouped_by_year_stats' dataframe using line graphs.
#Its clear to see that there is an upward trend in terms of movie revenue generation and most revenue is obtained from foreign market.

# Reset the index to have 'Release_year' as a regular column for plottin
grouped_by_year_stats= grouped_by_year_stats.reset_index()

# Create a line plot to show the trend over the years
plt.figure(figsize=(10, 6))
plt.plot(grouped_by_year_stats['Release_year'],grouped_by_year_stats ['Domestic_gross($)'], label='Domestic Gross', marker='o')
plt.plot(grouped_by_year_stats['Release_year'], grouped_by_year_stats['Foreign_gross($)'], label='Foreign Gross', marker='o')
plt.plot(grouped_by_year_stats['Release_year'], grouped_by_year_stats['Total_gross($)'], label='Total Gross', marker='o')

plt.title('Movies Earnings by Year')
plt.xlabel('Year')
plt.ylabel('Earnings ($)')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()




# In[208]:


##Create a 'grouped_by_studio' df grouping the 'movie_gross' df by 'studio' column.
grouped_by_studio = movie_gross.groupby('Studio')


# In[210]:


studio_statistics= grouped_by_studio['Domestic_gross($)','Foreign_gross($)','Total_gross($)'].mean()
print(studio_statistics)


# In[211]:


##Plot the 'studio_statistics' df using bar plots.So as to visualize the top 10 best performing studios in terms of revenue generation.

# Reset the index to have 'Studio' as a regular column for plotting
studio_statistics = studio_statistics.reset_index()

# Sort the studios by total gross income in descending order
studio_statistics = studio_statistics.sort_values(by='Total_gross($)', ascending=False)

# Select the top 10 studios
top_10_studios = studio_statistics.head(10)

# Set the figure size
plt.figure(figsize=(12, 6))

# Create subplots for each type of gross (Domestic, Foreign, Total) for the top 10 studios
plt.subplot(131)
sns.barplot(x='Domestic_gross($)', y='Studio', data=top_10_studios, orient='h')
plt.title('Average Domestic Gross by Top 10 Studios')
plt.xlabel('Average Gross ($)')
plt.ylabel('Studio')

plt.subplot(132)
```

```
sns.barplot(x='Foreign_gross($)', y='Studio', data=top_10_studios, orient='h')
plt.title('Average Foreign Gross by Top 10 Studios')
plt.xlabel('Average Gross ($)')
plt.ylabel('')

plt.subplot(133)
sns.barplot(x='Total_gross($)', y='Studio', data=top_10_studios, orient='h')
plt.title('Average Total Gross by Top 10 Studios')
plt.xlabel('Average Gross ($)')
plt.ylabel('')

# Adjust layout and show the plot
plt.tight_layout()
plt.show()



# ## Insights Gained From Performing EDA& My Reccommendations To The Microsoft Team.
#
# - After performing EDA on the 'movie_gross' dataset and making use of my domain knowledge, I have come to the following conclusions;
#
#       **1).** The Microsoft team should primarily focus on creating movies that are bound to have a sequel. This is because, over the time period of 2010-2018, most of the top-perform
#
#       **2).** The team should  majorly focus on creating movies that appeal to foreign markets. This is because, across the given time period, foreign gross revenue has been significa
#
#       **3).** I would also recommend that the Microsoft team consider collaborating with the Top 10 ranked Studios( in terms of revenue generation) when running marketing campaigns. T
#
#      **4).** Lastly, the team could investigate the cause of a spike in movie revenue generation from the year 2014 onwards. Might it be due to a change in movie graphics? Could it be
```