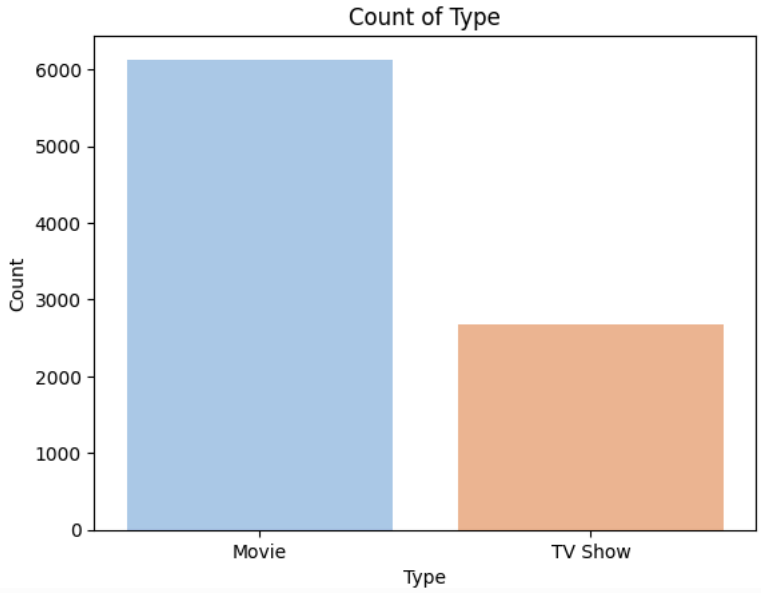
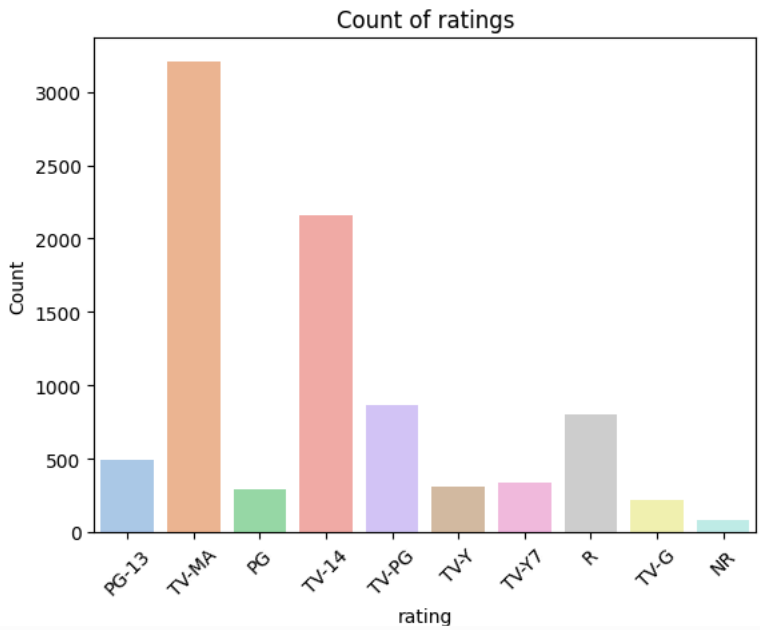
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df= pd.read_csv("netflix_titles.csv")


#Question-1. Find the counts of each categorical variable both using graphical and nongraphical analysis.
#a. For Non-graphical Analysis:


categorical_cols = ['type', 'rating', 'country', 'listed_in']

for col in categorical_cols:
    print(f"\n--- {col} ---")
    print(df[col].value_counts().head(10))
```

```
--- type ---
type
Movie      6131
TV Show    2676
Name: count, dtype: int64

--- rating ---
rating
TV-MA    3207
TV-14    2160
TV-PG     863
R         799
PG-13     490
TV-Y7     334
TV-Y      307
PG        287
TV-G      220
NR         80
Name: count, dtype: int64

--- country ---
country
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Egypt              106
Name: count, dtype: int64

--- listed_in ---
listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Kids' TV                                            220
Children & Family Movies                            215
Children & Family Movies, Comedies                  201
Documentaries, International Movies                 186
Dramas, International Movies, Romantic Movies       180
Name: count, dtype: int64
```

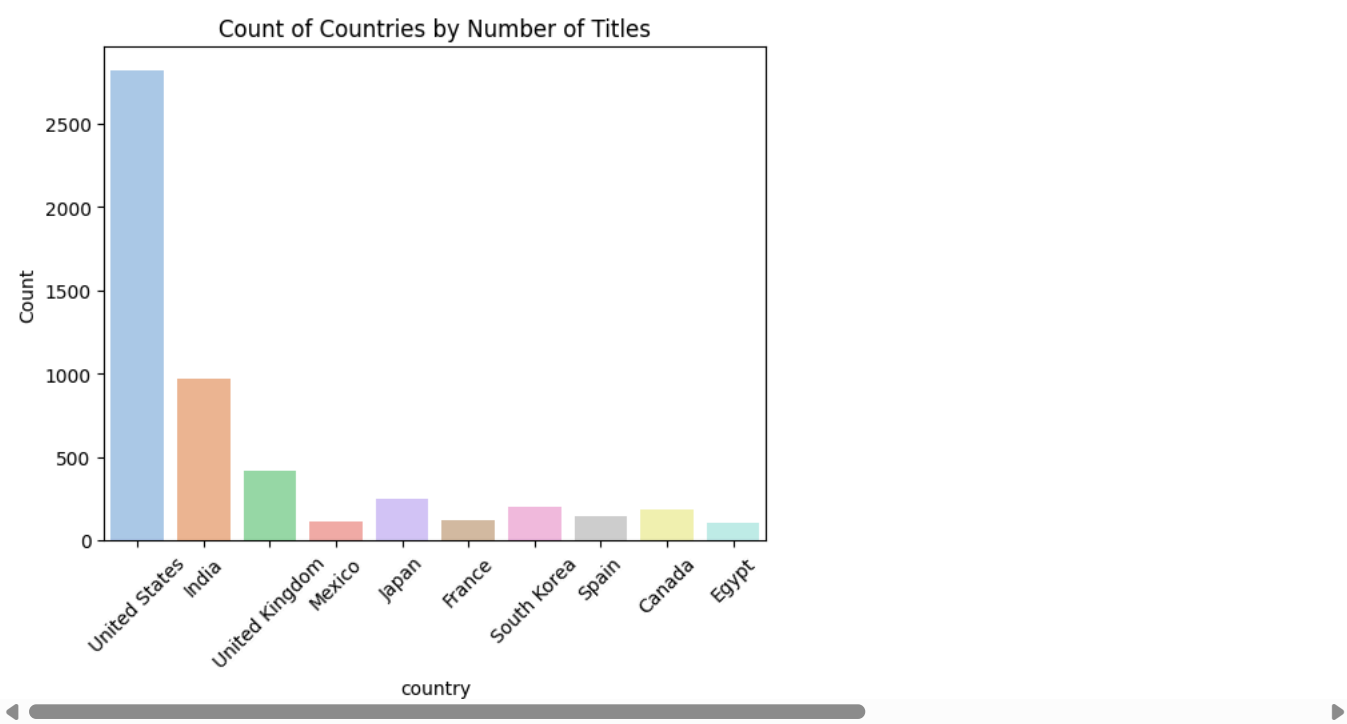
```
#1(b). For graphical analysis:


#Type count
sns.countplot(data=df, x='type', hue='type', palette='pastel')
plt.title('Count of Type')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```



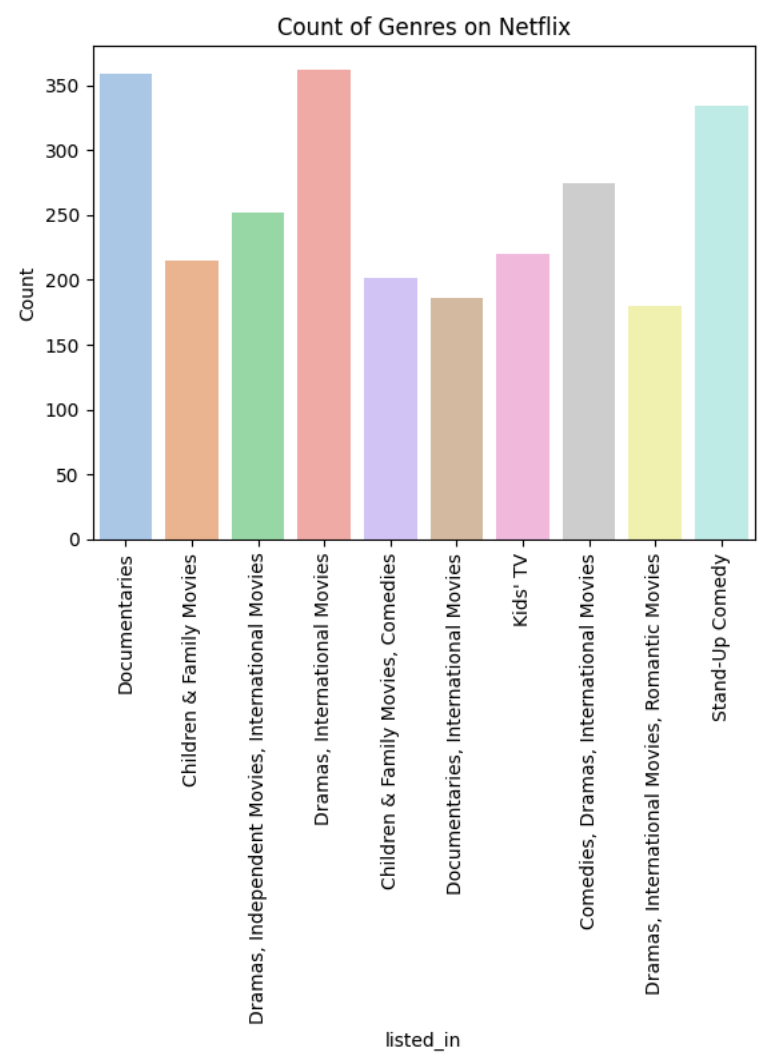

```
#Ratings count
top_ratings= df['rating'].value_counts().head(10).index
sns.countplot(data=df[df['rating'].isin(top_ratings)], x='rating', hue='rating', palette='pastel')
plt.title('Count of ratings')
plt.xlabel('rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```




```
#Countries count
top_countries= df['country'].value_counts().head(10).index
sns.countplot(data=df[df['country'].isin(top_countries)], x='country', hue='country', palette='pastel')
plt.title('Count of Countries by Number of Titles')
plt.xlabel('country')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```




```
#listed_in count
top_genres= df['listed_in'].value_counts().head(10).index
sns.countplot(data=df[df['listed_in'].isin(top_genres)], x='listed_in',hue='listed_in', palette='pastel')
plt.title('Count of Genres on Netflix')
plt.xlabel('listed_in')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



```
#Question-2 #Comparison of tv shows vs. movies.

#a. Find the number of movies produced in each country and pick the top 10 countries.


df_movies= df[df['type']=='Movie']
top_10_countrywise_movies= df_movies.groupby('country')['title'].nunique().sort_values(ascending=False).head(10)
top_10_countrywise_movies
```

country	title
United States	2058
India	893
United Kingdom	206
Canada	122
Spain	97
Egypt	92
Nigeria	86
Indonesia	77
Japan	76
Turkey	76

dtype: int64

```
#b. Find the number of Tv-Shows produced in each country and pick the top 10 countries.


df_movies= df[df['type']=='TV Show']
top_10_countrywise_movies= df_movies.groupby('country')['title'].nunique().sort_values(ascending=False).head(10)
top_10_countrywise_movies
```

country	title
United States	760
United Kingdom	213
Japan	169
South Korea	158
India	79
Taiwan	68
Canada	59
France	49
Spain	48
Australia	48

dtype: int64

```
#Question3. What is the best time to launch a TV show?

#a. Find which is the best week to release the Tv-show or the movie. Do the analysis
#separately for Tv-shows and Movies


df['date_added'] = pd.to_datetime(df['date_added'],errors='coerce')
df['week']= df['date_added'].dt.isocalendar().week
Movie_df =df[df['type']=='Movie']
Tvshow_df=df[df['type']=='TV Show']
Movie_week= Movie_df.groupby('week').size().sort_values(ascending=False)
Tvshow_week= Tvshow_df.groupby('week').size().sort_values(ascending=False)
Best_Movie_week=Movie_week.idxmax(),Movie_week.max()
Best_Tvshow_week=Tvshow_week.idxmax(),Tvshow_week.max()
print("Best_Movie_week", Best_Movie_week)
print("Best_Tvshow_week", Best_Tvshow_week)
```

```
Best_Movie_week (np.uint32(1), 316)
Best_Tvshow_week (np.uint32(27), 85)
```

```
#Answers- Week 40 had maximum Movies i.e 136 and Week 27 had maximum TVShows i.e 79
```

```
# 3(b) Find which is the best month to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies
```

```
df['date_added'] = pd.to_datetime(df['date_added'],errors='coerce')
df['month']= df['date_added'].dt.month
Movie_df =df[df['type']=='Movie']
Tvshow_df=df[df['type']=='TV Show']
Movie_month= Movie_df.groupby('month').size().sort_values(ascending=False)
Tvshow_month= Tvshow_df.groupby('month').size().sort_values(ascending=False)
Best_Movie_month=Movie_month.idxmax(),Movie_month.max()
Best_Tvshow_month=Tvshow_month.idxmax(),Tvshow_month.max()
print("Best_Movie_month", Best_Movie_month)
print("Best_Tvshow_month", Best_Tvshow_month)
```

```
Best_Movie_month (np.float64(7.0), 565)
Best_Tvshow_month (np.float64(7.0), 254)
```

```
#Answers- Month 7 had maximum Movies i.e 348 and Month 7 had maximum TVShows i.e 200
```

```
#Question 4. Analysis of actors/directors of different types of shows/movies.
#a. Identify the top 10 actors who have appeared in most movies or TV shows.
```

```
df_cast= df.dropna(subset=['cast'])
df_cast['actor']= df_cast['cast'].apply(lambda x:x.split(", "))
df_exploded = df_cast.explode('actor')
top_actors= df_exploded.groupby('actor')['type'].size().sort_values(ascending=False).head(10)
top_actors
```

```
<ipython-input-25-d8345c568ea3>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_cast['actor']= df_cast['cast'].apply(lambda x:x.split(", "))
```

actor	type
Anupam Kher	43
Shah Rukh Khan	35
Julie Tejwani	33
Takahiro Sakurai	32
Naseeruddin Shah	32
Rupa Bhimani	31
Akshay Kumar	30
Om Puri	30
Yuki Kaji	29
Paresh Rawal	28

dtype: int64

```
#4(b). Identify the top 10 directors who have appeared in most movies or TV shows.
```

```
df_direct= df.dropna(subset=['director'])
df_direct['Director']= df_direct['director'].apply(lambda x:x.split(", "))
df_exploded = df_direct.explode('Director')
top_directors= df_exploded.groupby('Director')['type'].size().sort_values(ascending=False).head(10)
top_directors
```

```
<ipython-input-45-078c0db1dd2f>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_direct['Director']= df_direct['director'].apply(lambda x:x.split(", "))
```

Director	type
Rajiv Chilaka	22
Jan Suter	21
Raúl Campos	19
Marcus Raboy	16

```
#Question 5. Which genre movies are more popular or produced more

from wordcloud import WordCloud
df_genre= df.dropna(subset=['listed_in'])
genre_text = ' '.join(df_genre['listed_in'].values)
wordcloud=WordCloud(
    width=500,
    height=80,
    background_color='white',
    colormap='viridis'
    ).generate(genre_text)
plt.figure(figsize=(10, 7))
plt.axis('off')
plt.imshow(wordcloud)
plt.title("Most Common Genres on Netflix", fontsize=20)
plt.show()
```


```
#Question 6. Find After how many days the movie will be added to Netflix after
#the release of the movie (you can consider the recent past data)


df['date_added'] = pd.to_datetime(df['date_added'])
df['release_date'] = pd.to_datetime(df['release_year'].astype(str), format='%Y')
df['Days_to_add'] =  (df['date_added']- df['release_date']).dt.days
days_mode = df['Days_to_add'].dropna().mode()
print("Most common number of days between release and Netflix addition:", days_mode.iloc[0])
```

```
Most common number of days between release and Netflix addition: 334.0
```