studio-ghibli-analysis

July 8, 2024

[ ]:

ANALYSIS STUDIO GHIBLI FILMOGRAPHY

[ ]:

[1]:
```
import pyodbc
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np
```

[2]:
```
sg = pd.read_excel('studio_ghiblidb.xlsx')
cast = pd.read_excel('cast sgdb.xlsx')
```

[ ]:

[ ]:

The following DataBase containes data about Studio Ghibli, one of the worlds' famous and successful Japanese animation studio . Here you'll find information about the studio filmography; deatails about the movies, casts, leading crew and expensses.

[ ]:

What is in the df?

[3]:
```
sg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Movieid           20 non-null     int64 
 1   MovieName         20 non-null     object
 2   Genre1            20 non-null     object
 3   Genre2            20 non-null     object
```

```
 4   Genre3            18 non-null     object
 5   Budget            20 non-null     int64
 6   Revenue           20 non-null     int64
 7   US_Release        18 non-null     datetime64[ns]
 8   JapanReleaseDate  20 non-null     datetime64[ns]
 9   Duration          20 non-null     object
 10  Director          20 non-null     object
 11  Producer          20 non-null     object
 12  ArtDirector       19 non-null     object
 13  Composer          20 non-null     object
 14  writer            20 non-null     object
dtypes: datetime64[ns](2), int64(3), object(10)
memory usage: 2.5+ KB
```

[4]:
```
sg.head()
```

[4]:

	Movieid	MovieName	Genre1	Genre2	Genre3	\
0	20	When Marnie Was There	animation	drama	NaN	
1	19	The Tale of The Princess Kaguya	animation	drama	fantasy	
2	18	The Wind Rises	animation	drama	romance	
3	16	The Secret World of Arrietty	animation	fantasy	family	
4	15	Ponyo	animation	fantasy	family	

	Budget	Revenue	US_Release	JapanReleaseDate	Duration	\
0	115000000	34949567	2015-07-26	2014-07-14	01:43:00	
1	49300000	24366656	2014-10-17	2013-11-23	02:17:00	
2	30000000	117932401	2014-02-21	2013-07-20	02:06:00	
3	23000000	149480483	2012-02-17	2010-07-17	01:34:00	
4	34000000	202404009	2009-08-14	2008-07-19	01:40:00	

	Director	Producer	ArtDirector	\
0	Hiromasa Yonebayashi	Yoshiaki Nishimura	Yohei Taneda	
1	Isao Takahata	Yoshiaki Nishimura	Kazuo Komatsubara	
2	Hayao Miyazaki	Toshio Suzuki	Yoji Takeshige	
3	Hiromasa Yonebayashi	Toshio Suzuki	Noboru Yoshida/ Yoji Takeshige	
4	Hayao Miyazaki	Toshio Suzuki	Noboru Yoshida	

	Composer	writer
0	Takatsugu Muramatsu	Joan G. Robinson
1	Joe Hisaishi	Riko Sakaguchi
2	Joe Hisaishi	Tatsuo Hori
3	Cécile Corbel	Mary Norton
4	Joe Hisaishi	Hayao Miyazaki

[7]:
```
sg.describe()
```

[7]:

	Movieid	Budget	Revenue	US_Release	JapanReleaseDate
count	20	20	20	18	20
mean	11	27770000	81993331	2003-09-13 02:40:00	1999-12-24 04:48:00
min	1	500000	100000	1985-06-13 00:00:00	1984-03-11 00:00:00
25%	6	4675000	19260510	1995-01-16 12:00:00	1991-01-20 18:00:00
50%	10	20500000	37974784	2004-02-02 12:00:00	1996-07-13 00:00:00
75%	15	31000000	153860362	2013-08-21 06:00:00	2009-01-17 00:00:00
max	21	115000000	274925095	2023-10-01 00:00:00	2023-07-14 00:00:00
std	6	32360471	86988427	NaN	NaN

The float format had been imported incorrectly, so I reformatted it.

[8]:
```
pd.set_option('display.float_format', '{:.0f}'.format)
```

[ ]:
```

```

For better analysing i'll add a new column 'Year' based on the column, 'JapanReleaseDate' - the first release of each film

[9]:
```
sg = sg.assign(Year = sg['JapanReleaseDate'].dt.year)
```

[10]:
```
sg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Movieid           20 non-null     int64
 1   MovieName         20 non-null     object
 2   Genre1            20 non-null     object
 3   Genre2            20 non-null     object
 4   Genre3            18 non-null     object
 5   Budget            20 non-null     int64
 6   Revenue           20 non-null     int64
 7   US_Release        18 non-null     datetime64[ns]
 8   JapanReleaseDate  20 non-null     datetime64[ns]
 9   Duration          20 non-null     object
 10  Director          20 non-null     object
 11  Producer          20 non-null     object
 12  ArtDirector       19 non-null     object
 13  Composer          20 non-null     object
 14  writer            20 non-null     object
 15  Year              20 non-null     int32
dtypes: datetime64[ns](2), int32(1), int64(3), object(10)
memory usage: 2.6+ KB
```

[ ]:
```

```

In the following queries let's understand the studio financial curve through time:

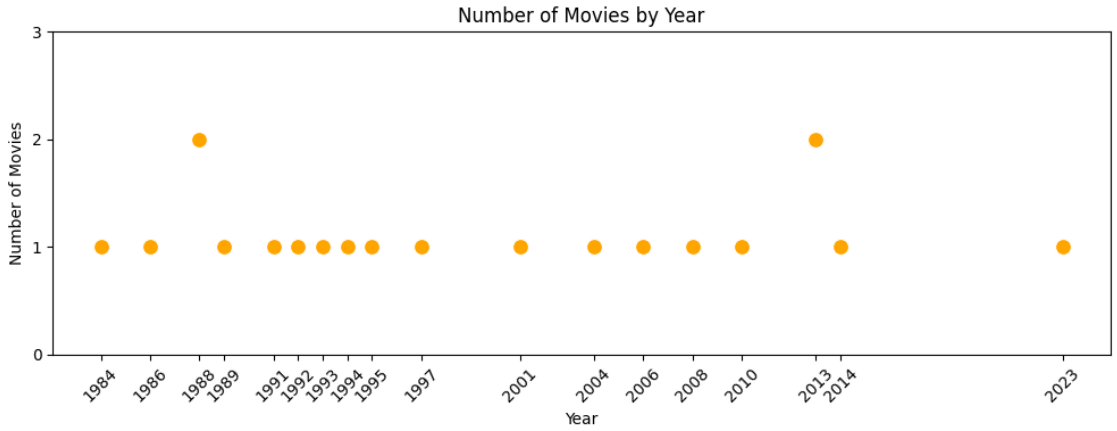
In which years were Studio Ghibli films released, and were multiple films released in any of those years?

[11]:
```
sg.groupby(['Year'])[['Movieid']].count()
```

[11]:

Year	Movieid
1984	1
1986	1
1988	2
1989	1
1991	1
1992	1
1993	1
1994	1
1995	1
1997	1
2001	1
2004	1
2006	1
2008	1
2010	1
2013	2
2014	1
2023	1

[12]:
```
movie_count_by_year = sg.groupby('Year')[['Movieid']].count().reset_index()
movie_count_by_year.columns = ['Year', 'MovieCount']

plt.figure(figsize=(10, 4))
sb.scatterplot(data=movie_count_by_year, x='Year', y='MovieCount', s=100,␣
 ↪color='#ffa500')
plt.title('Number of Movies by Year')
plt.xlabel('Year')
plt.ylabel('Number of Movies')
years = movie_count_by_year['Year'].unique()
plt.xticks(years, rotation=45)
y_max = movie_count_by_year['MovieCount'].max()
plt.yticks(range(0, y_max + 2, 1))
plt.tight_layout()
plt.show()
```



What was the sum of revenue vers budget each year?

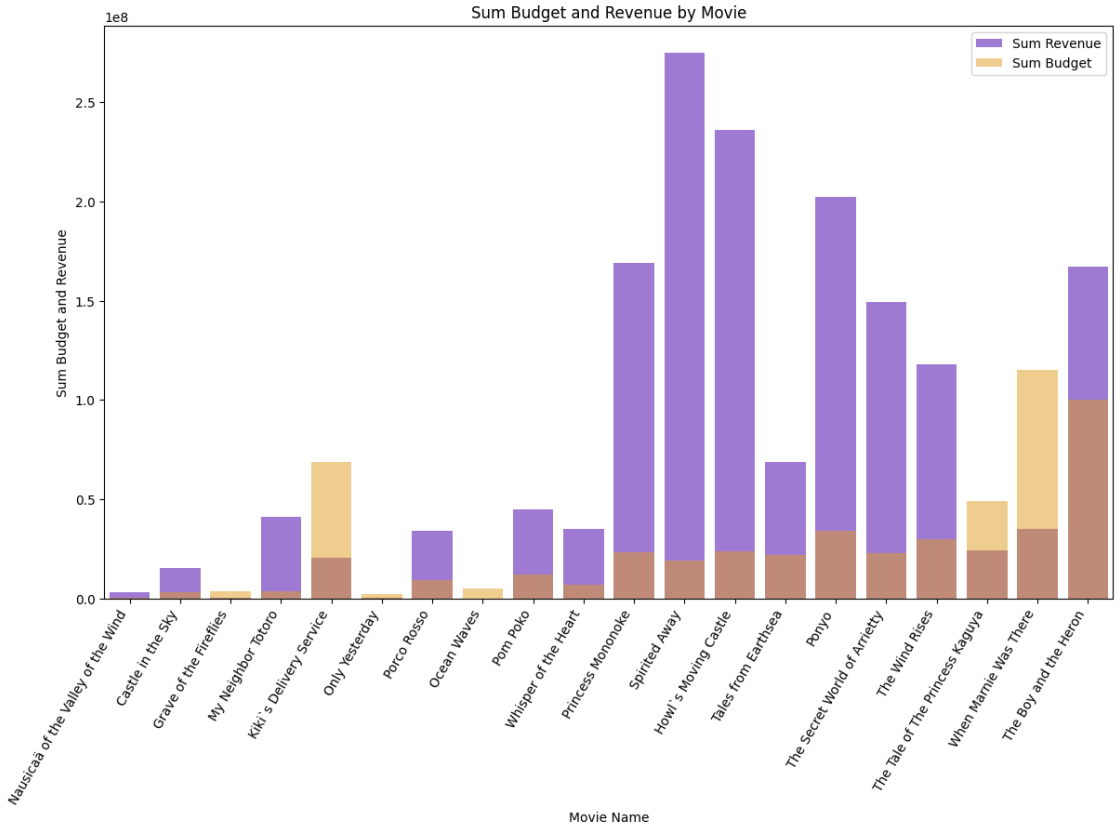
[13]:
```
sg.groupby(['Year'])[['Revenue', 'Budget']].sum()
```

[13]:

Year	Revenue	Budget
1984	3301446	500000
1986	15542039	3000000
1988	41516962	7400000
1989	20500000	69000000
1991	473110	2500000
1992	34100000	9200000
1993	100000	5000000
1994	44700000	12000000
1995	34900000	7000000
1997	169000000	23500000
2001	274925095	19000000
2004	236049757	24000000
2006	68625104	22000000
2008	202404009	34000000
2010	149480483	23000000
2013	142299057	79300000
2014	34949567	115000000
2023	167000000	100000000

[14]:
```
sum_budget_revenue_year = sg.groupby('Movieid')[['Budget', 'Revenue']].sum().
↪reset_index()
sum_budget_revenue_year = pd.merge(sum_budget_revenue_year, sg[['Movieid',␣
↪'MovieName']].drop_duplicates(), on='Movieid')
sum_budget_revenue_year = sum_budget_revenue_year.sort_values(by='Movieid')

x_range = sum_budget_revenue_year['Movieid']
```

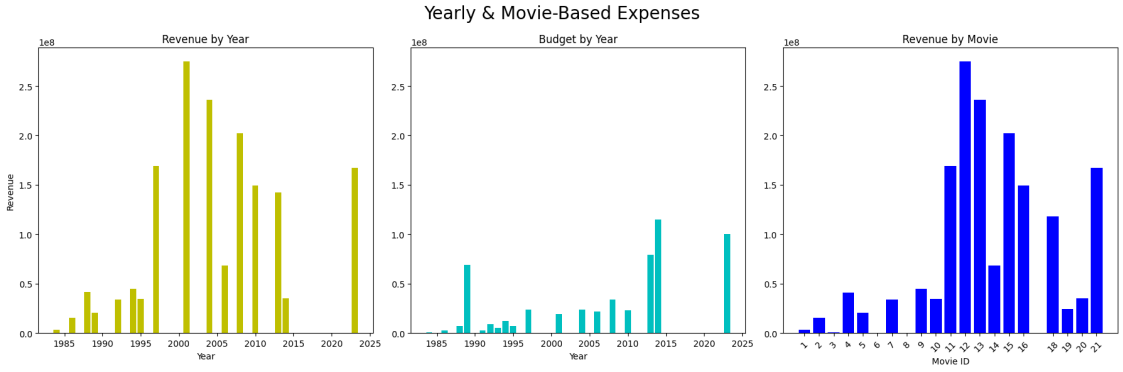
```
plt.figure(figsize=(14, 8))
sb.barplot(data=sum_budget_revenue_year, x='MovieName', y='Revenue', label='Sum␣
 ↪Revenue', color='#9c6be1')
sb.barplot(data=sum_budget_revenue_year, x='MovieName', y='Budget', label='Sum␣
 ↪Budget', alpha=0.5, color='orange')
plt.legend()
plt.title('Sum Budget and Revenue by Movie')
plt.xlabel('Movie Name')
plt.ylabel('Sum Budget and Revenue')
plt.xticks(rotation=60, ha='right')
plt.show()
```



Let's divide the budget and revnue to two separate grapghs and show it next to revenue per film

[15]:
```
revenue_by_year = sg.groupby(['Year'])[['Revenue']].sum()
budget_by_year = sg.groupby(['Year'])[['Budget']].sum()
revenue_by_movie = sg.groupby('Movieid')[['Revenue']].sum()

plt.figure(figsize=(18, 6))
plt.suptitle('Yearly & Movie-Based Expenses', fontsize=20)
```

```
ax1 = plt.subplot(1, 3, 1)
ax1.bar(revenue_by_year.index, revenue_by_year['Revenue'], color='y')
ax1.set_title('Revenue by Year')
ax1.set_xlabel('Year')
ax1.set_ylabel('Revenue')

ax2 = plt.subplot(1, 3, 2, sharey=ax1)
ax2.bar(budget_by_year.index, budget_by_year['Budget'], color='c')
ax2.set_title('Budget by Year')
ax2.set_xlabel('Year')

ax3 = plt.subplot(1, 3, 3, sharey=ax1)
ax3.bar(revenue_by_movie.index, revenue_by_movie['Revenue'], color='b')
ax3.set_title('Revenue by Movie')
ax3.set_xlabel('Movie ID')
movie_ids = revenue_by_movie.index
ax3.set_xticks(movie_ids)
ax3.set_xticklabels(movie_ids, rotation=45)


plt.tight_layout()
plt.show()
```



[ ]:

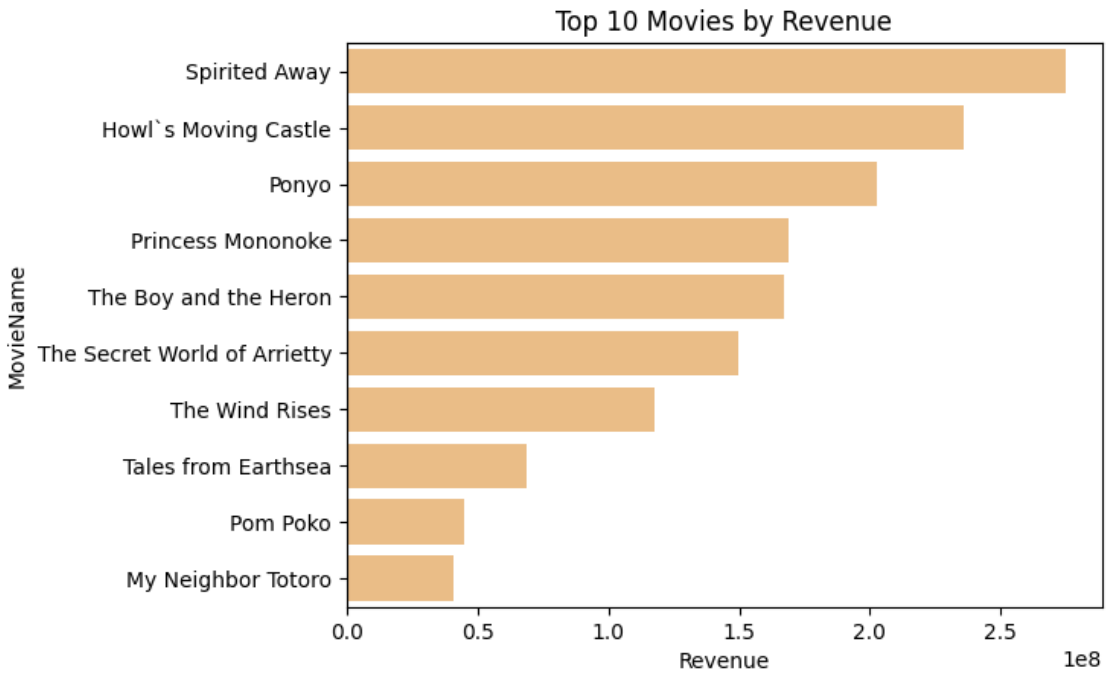
This DataBase datetime type is based on two source,s the origin relase date in Japan and in the major worldwide boxoffice, the United States:

Which movie was the most profitable and when it was released in Japan and the USA?

[16]:
```
sg.loc[sg['Revenue']==sg['Revenue'].max(),['MovieName','Revenue','US_Release',␣
↪'JapanReleaseDate']]
```

[16]:
```
        MovieName    Revenue US_Release JapanReleaseDate
8  Spirited Away  274925095 2002-09-20       2001-07-20
```

[17]:
```
top_10_revenue = sg.nlargest(10, 'Revenue')
sb.barplot(data=top_10_revenue, x='Revenue', y='MovieName', color='#fabe77')
plt.title('Top 10 Movies by Revenue')
plt.show()
```



[ ]:

How many unique genres are in the studio filmography?

[18]:
```
sg['Genre3'].unique()
```

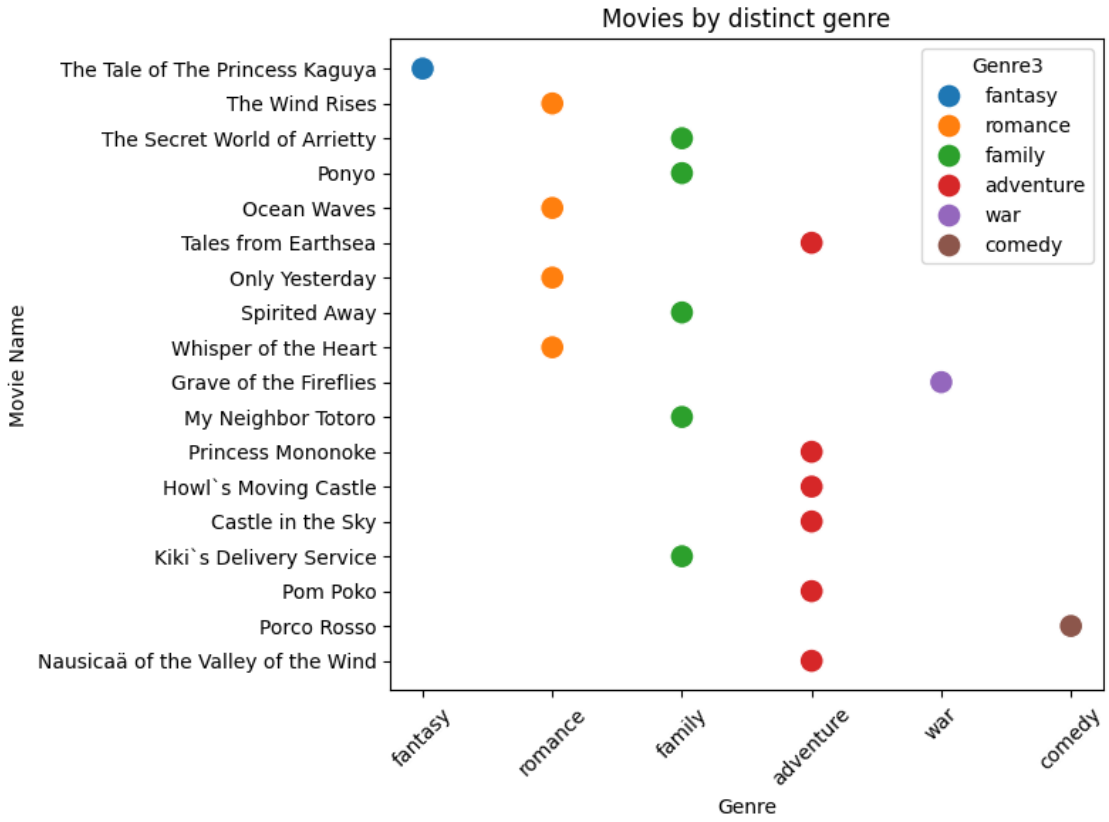
[18]:
```
array([nan, 'fantasy', 'romance', 'family', 'adventure', 'war', 'comedy'],
      dtype=object)
```

[19]:
```
sg.loc[sg['Genre3']=='adventure',['MovieName','Genre3']]
```

[19]:
```
                        MovieName     Genre3
6                 Tales from Earthsea  adventure
12                  Princess Mononoke  adventure
13                Howl`s Moving Castle  adventure
14                  Castle in the Sky  adventure
16                           Pom Poko  adventure
```

```
18  Nausicaä of the Valley of the Wind  adventure
```

[20]:
```
filtered_df = sg[['MovieName', 'Genre3']]

plt.figure(figsize=(8, 6))
sb.scatterplot(data=filtered_df, x='Genre3', y='MovieName', hue='Genre3',␣
 ↪palette='tab10', s=150)
plt.title('Movies by distinct genre')
plt.xlabel('Genre')
plt.ylabel('Movie Name')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



I want to add the human factor and analyse the stodio finance by creators and producors

[ ]:
```
How many directors worked on how many films
```

[21]:
```
sg.groupby(['Director'])[['Movieid']].count()
```

[21]:
```
                      Movieid
Director
Goro Miyazaki               1
Hayao Miyazaki             11
Hiromasa Yonebayashi        2
Isao Takahata               4
Tomomi Mochizuki            1
Yoshifumi Kondo             1
```

How many movies were made in the 21st century?

[22]:
```
sg.loc[sg['Year'].between(2000, 2023), ['MovieName', 'Year', 'Budget',␣
↪'Revenue', 'Director']]
```

[22]:
```
                          MovieName  Year     Budget    Revenue  \
0              When Marnie Was There  2014  115000000   34949567
1   The Tale of The Princess Kaguya  2013   49300000   24366656
2                     The Wind Rises  2013   30000000  117932401
3      The Secret World of Arrietty  2010   23000000  149480483
4                             Ponyo  2008   34000000  202404009
6               Tales from Earthsea  2006   22000000   68625104
8                     Spirited Away  2001   19000000  274925095
13             Howl`s Moving Castle  2004   24000000  236049757
19            The Boy and the Heron  2023  100000000  167000000

                Director
0   Hiromasa Yonebayashi
1          Isao Takahata
2         Hayao Miyazaki
3   Hiromasa Yonebayashi
4         Hayao Miyazaki
6          Goro Miyazaki
8         Hayao Miyazaki
13        Hayao Miyazaki
19        Hayao Miyazaki
```

What was the first movie released by Studio Ghibli?

[23]:
```
sg.loc[sg['JapanReleaseDate'] == sg['JapanReleaseDate'].min(), ['MovieName',␣
↪'JapanReleaseDate', 'Director']]
```

[23]:
```
                          MovieName JapanReleaseDate        Director
18  Nausicaä of the Valley of the Wind       1984-03-11  Hayao Miyazaki
```

What was the budget for this, and what revenue did it generate?

[24]:
```
sg.loc[sg['MovieName']=='Nausicaä of the Valley of the␣
↪Wind',['Budget','Revenue']]
```

[24]:
```
    Budget  Revenue
18  500000  3301446
```

How many movies were released during the studio's first decade of production?

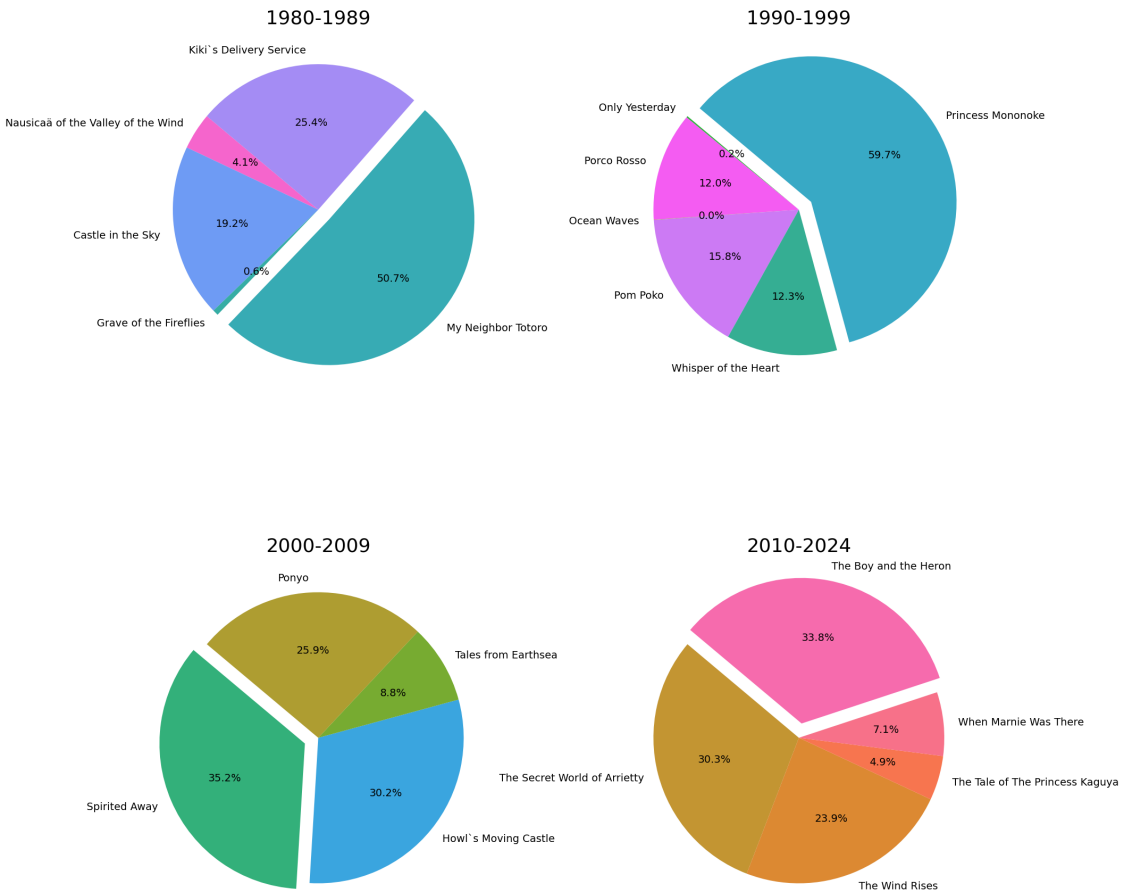
[25]:
```
sg.loc[sg['Year'].between(1980,1989), ['MovieName', 'Year', 'Budget',␣
 ↪'Revenue', 'Director']]
```

[25]:
```
                            MovieName  Year    Budget   Revenue  \
10              Grave of the Fireflies  1988   3700000    516962
11                  My Neighbor Totoro  1988   3700000  41000000
14                   Castle in the Sky  1986   3000000  15542039
15             Kiki`s Delivery Service  1989  69000000  20500000
18  Nausicaä of the Valley of the Wind  1984    500000   3301446

          Director
10   Isao Takahata
11  Hayao Miyazaki
14  Hayao Miyazaki
15  Hayao Miyazaki
18  Hayao Miyazaki
```

How many movies were made in each decade since the studio started creating and what was the precentage of each movie from the decade's sum of revenues

[26]:
```
# Generate a unique color for each Movie ID
unique_movie_ids = sg['Movieid'].unique()
colors = sb.color_palette("husl", len(unique_movie_ids))

decades = [
    (1980, 1989),
    (1990, 1999),
    (2000, 2009),
    (2010, 2024)
]

# Create a mapping from Movieid to color
movie_id_to_color = {movie_id: colors[i] for i, movie_id in␣
 ↪enumerate(unique_movie_ids)}

# Define a function to format the revenue values
def format_revenue(val):
    if val >= 1e9:
        return f'{val / 1e9:.1f}B'
    elif val >= 1e6:
        return f'{val / 1e6:.1f}M'
    elif val >= 1e3:
        return f'{val / 1e3:.1f}K'
```

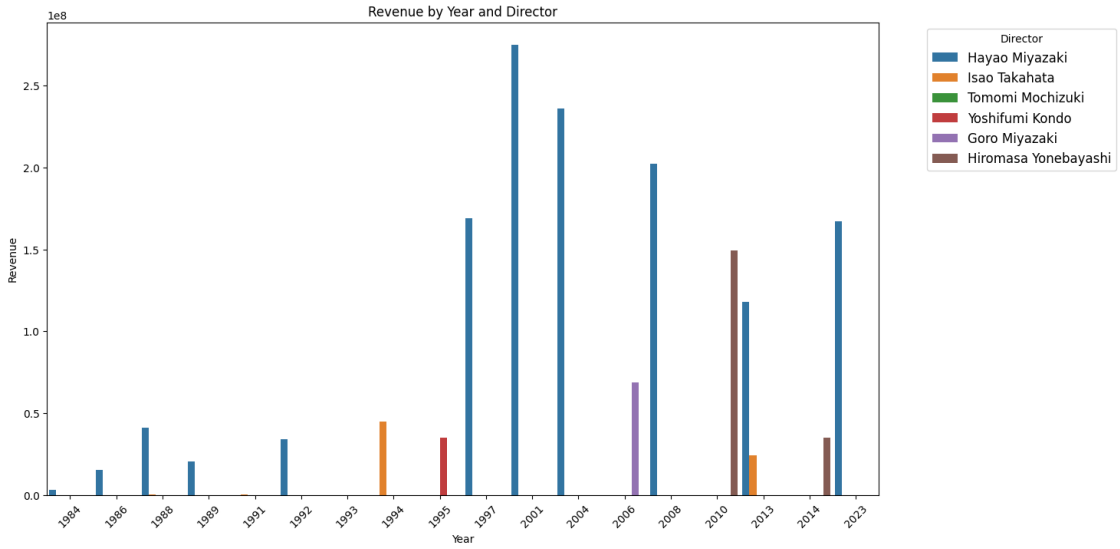
```
    else:
        return f'{val:.0f}'

fig, axes = plt.subplots(2, 2, figsize=(20, 20))
for i, (start, end) in enumerate(decades):
    ax = axes[i // 2, i % 2]  # Adjust indexing for a 2x2 grid
    df_range = sg[(sg['Year'] >= start) & (sg['Year'] <= end)]
    df_grouped = df_range.groupby('Movieid')['Revenue'].sum().reset_index()

    labels = df_grouped['Movieid'].apply(lambda x: sg[sg['Movieid'] ==␣
 ↪x]['MovieName'].values[0])
    sizes = df_grouped['Revenue']
    colors = [movie_id_to_color[mid] for mid in df_grouped['Movieid']]
    explode = [0.1 if size == max(sizes) else 0 for size in sizes]
    wedges, texts, autotexts = ax.pie(sizes, labels=labels, colors=colors,␣
 ↪explode=explode,
                                      autopct='%1.1f%%',
                                      startangle=140, textprops={'fontsize':␣
 ↪14})
    ax.set_title(f'{start}-{end}', fontsize=26)

    for text in texts + autotexts:
        text.set_fontsize(14)

plt.subplots_adjust(hspace=0.4, wspace=0.00)
plt.tight_layout()
plt.show()
```



[ ]:

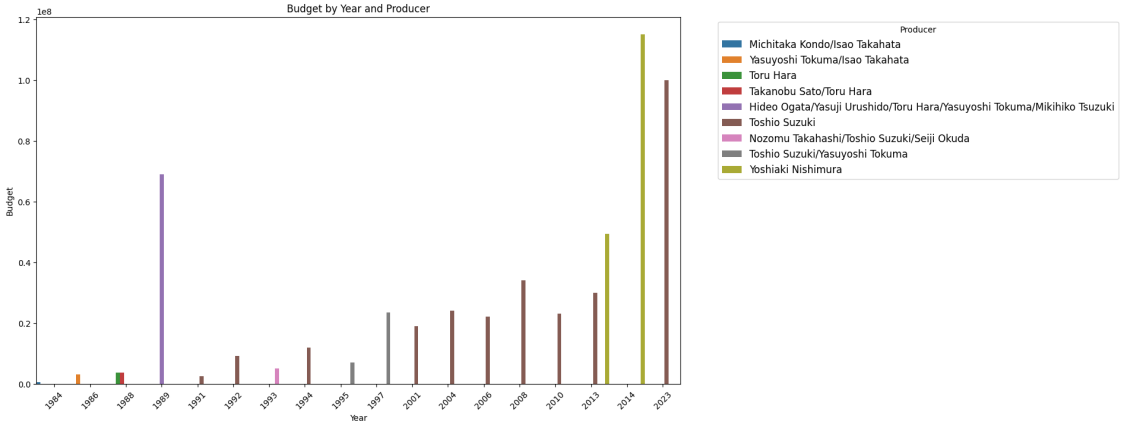
Which director made the most revenue for the studio?

[27]:
```
sg = sg.sort_values(by='Year', ascending=True)

plt.figure(figsize=(14, 8))
sb.barplot(data=sg, x="Year", y="Revenue", hue='Director', width=0.9)
plt.xticks(rotation=45)
plt.legend(title='Director', bbox_to_anchor=(1.05, 1), loc='upper left',␣
↪fontsize='large')
plt.title('Revenue by Year and Director')
plt.xlabel('Year')
plt.ylabel('Revenue')
plt.show()
```



Who was the producer that invested the highest budget?

[28]:
```
plt.figure(figsize=(14, 8))
sb.barplot(data=sg, x='Year', y='Budget', hue='Producer', width=1)
plt.legend(title='Producer', bbox_to_anchor=(1.05, 1), loc='upper left',␣
 ↪fontsize='large')
plt.xticks(rotation=45)
plt.title('Budget by Year and Producer')
plt.show()
```



How many characters are there in each movie?

[29]:
```
ch = pd.merge(cast, sg[['Movieid','MovieName']]
              , left_on='MovieId',
```

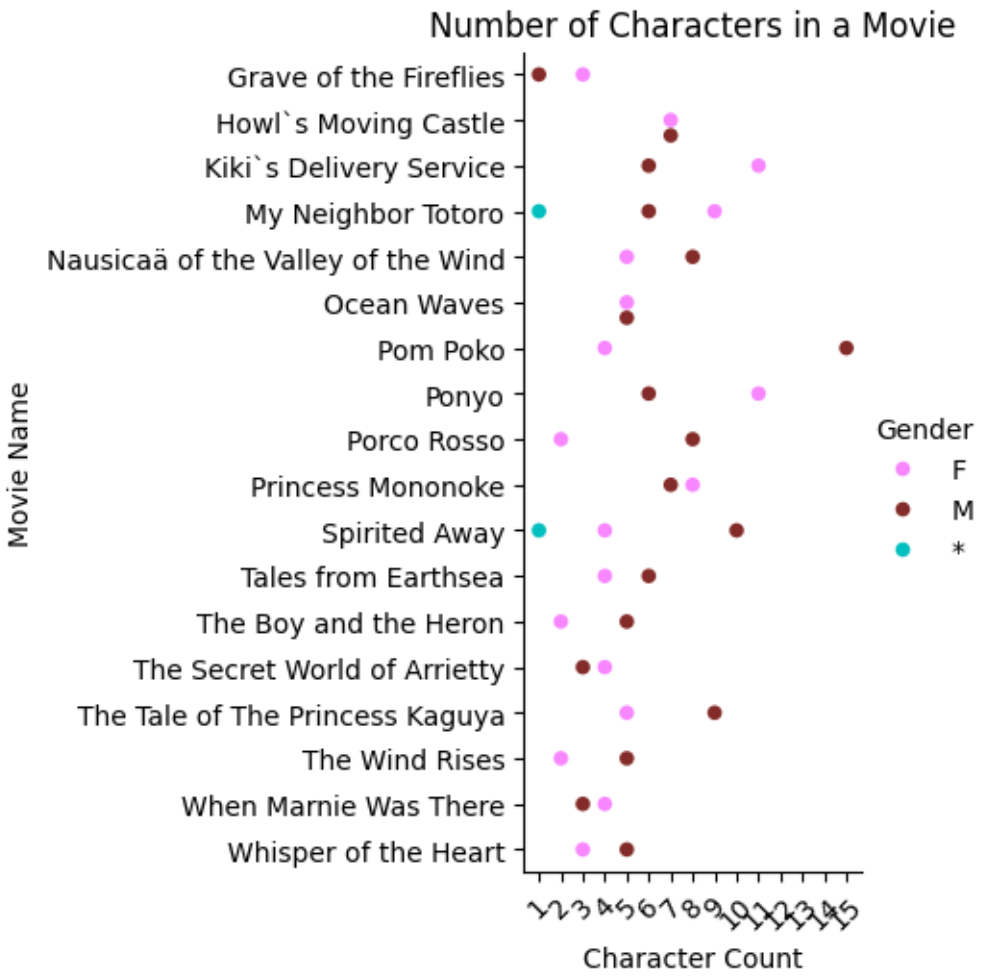
```
            right_on = 'Movieid',
            how='inner')
```

[30]:
```
ch.groupby('MovieName')[['CharacterName']].count()
```

[30]:

MovieName	CharacterName
Grave of the Fireflies	4
Howl`s Moving Castle	14
Kiki`s Delivery Service	17
My Neighbor Totoro	16
Nausicaä of the Valley of the Wind	13
Ocean Waves	10
Pom Poko	19
Ponyo	17
Porco Rosso	10
Princess Mononoke	15
Spirited Away	15
Tales from Earthsea	10
The Boy and the Heron	7
The Secret World of Arrietty	7
The Tale of The Princess Kaguya	14
The Wind Rises	7
When Marnie Was There	7
Whisper of the Heart	8

[31]:
```
character_count = ch.groupby(['MovieName', 'Gender']).size().
 ↪reset_index(name='CharacterCount')

sb.catplot(x='CharacterCount', y='MovieName', hue='Gender',␣
 ↪data=character_count, kind='swarm', palette=('#f887ff','#832b28','c'),s=30)
plt.title('Number of Characters in a Movie')
plt.xlabel('Character Count')
plt.ylabel('Movie Name')
maxcount = character_count['CharacterCount'].max()
plt.xticks(range(1,maxcount+1), rotation=45)
plt.show()
```



[ ]:
```
How many of the participating characters are females
```

[32]:
```
ch[ch['Gender'] == 'F'].groupby('MovieName')[['CharacterName']].count()
```

[32]:

MovieName	CharacterName
Grave of the Fireflies	3
Howl`s Moving Castle	7
Kiki`s Delivery Service	11
My Neighbor Totoro	9
Nausicaä of the Valley of the Wind	5
Ocean Waves	5
Pom Poko	4
Ponyo	11
Porco Rosso	2
Princess Mononoke	8

```
Spirited Away                          4
Tales from Earthsea                    4
The Boy and the Heron                  2
The Secret World of Arrietty           4
The Tale of The Princess Kaguya        5
The Wind Rises                         2
When Marnie Was There                  4
Whisper of the Heart                   3
```

Show one movie full cast - characters and dubbing actors from Japan and the USA

[33]:
```
ch[ch['MovieName']=='Ponyo']
```

[33]:

	MovieId	CharacterName	japanCastMember	USA_CastMember	\
35	15	Lisa	Tomoko Yamaguchi	Tina Fey	
36	15	Koichi	Kazushige Nagashima	Matt Damon	
37	15	Granmamare	Yuki Amami	Cate Blanchett	
38	15	Fujimoto	George Tokoro	Liam Neeson	
39	15	Ponyo	Yuria Nara	Noah Cyrus	
40	15	Sosuke	Hiroki Doi	Frankie Jonas	
41	15	Young Mother	Rumi Hiiragi	Mona Marshall	
42	15	Ponyo`s Sisters	Akiko Yano	NaN	
43	15	Toki	Kazuko Yoshiyuki	Lily Tomlin	
44	15	Yoshie	Tomoko Naraoka	Betty White	
45	15	Newscaster	Shin`ich Hatori	Kurt Knutsson	
46	15	Kayo	Tokie Hidari	Marsha Clark	
47	15	Kumiko	Eimi Hiraoka	Jennessa Rose	
48	15	Karen	Nozomi Ohashi	Colleen O`Shaughnessey	
49	15	Noriko	Akiko Takeguchi	Cloris Leachman	
50	15	Sosuke`s Teacher	Eiko Kanasawa	Courtnee Draper	
51	15	Young Father	Shirô Saitô	Bob Bergen	

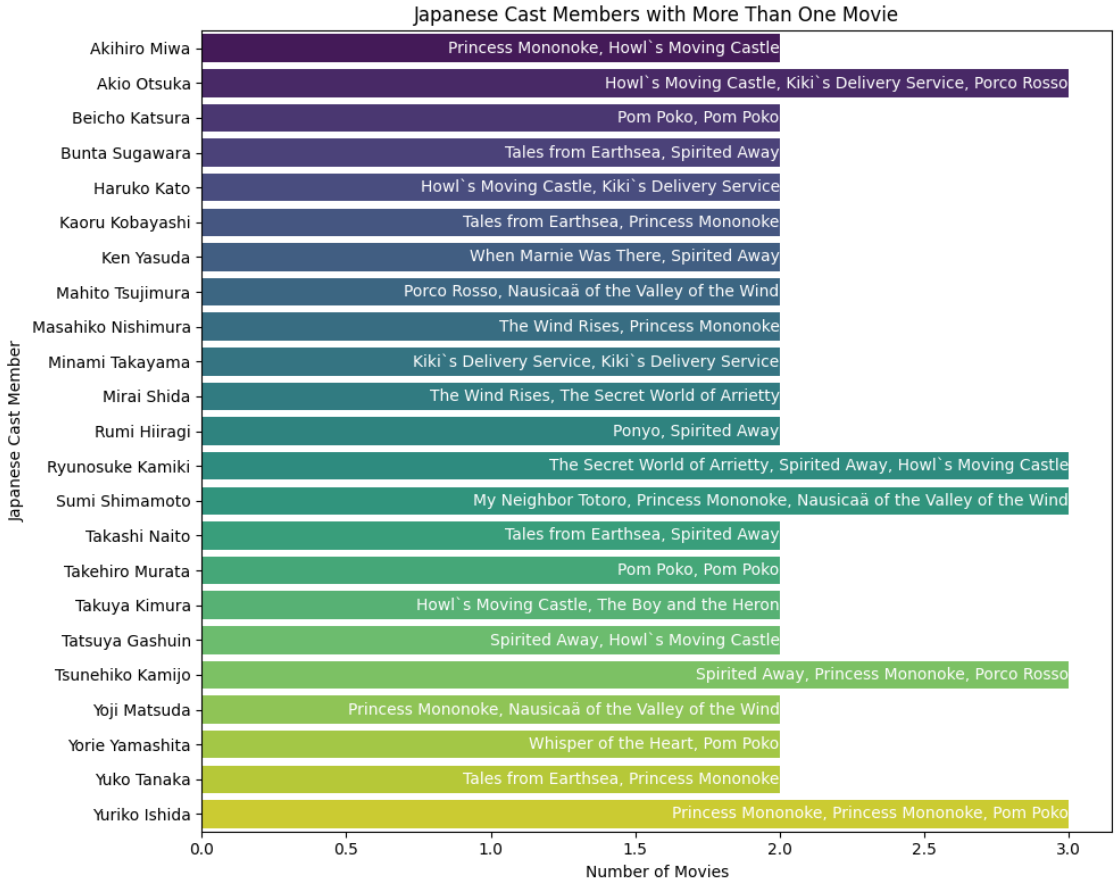
	Gender	Movieid	MovieName
35	F	15	Ponyo
36	M	15	Ponyo
37	F	15	Ponyo
38	M	15	Ponyo
39	F	15	Ponyo
40	M	15	Ponyo
41	M	15	Ponyo
42	F	15	Ponyo
43	F	15	Ponyo
44	F	15	Ponyo
45	M	15	Ponyo
46	F	15	Ponyo
47	F	15	Ponyo
48	F	15	Ponyo

```
49      F       15      Ponyo
50      F       15      Ponyo
51      M       15      Ponyo
```

[ ]:

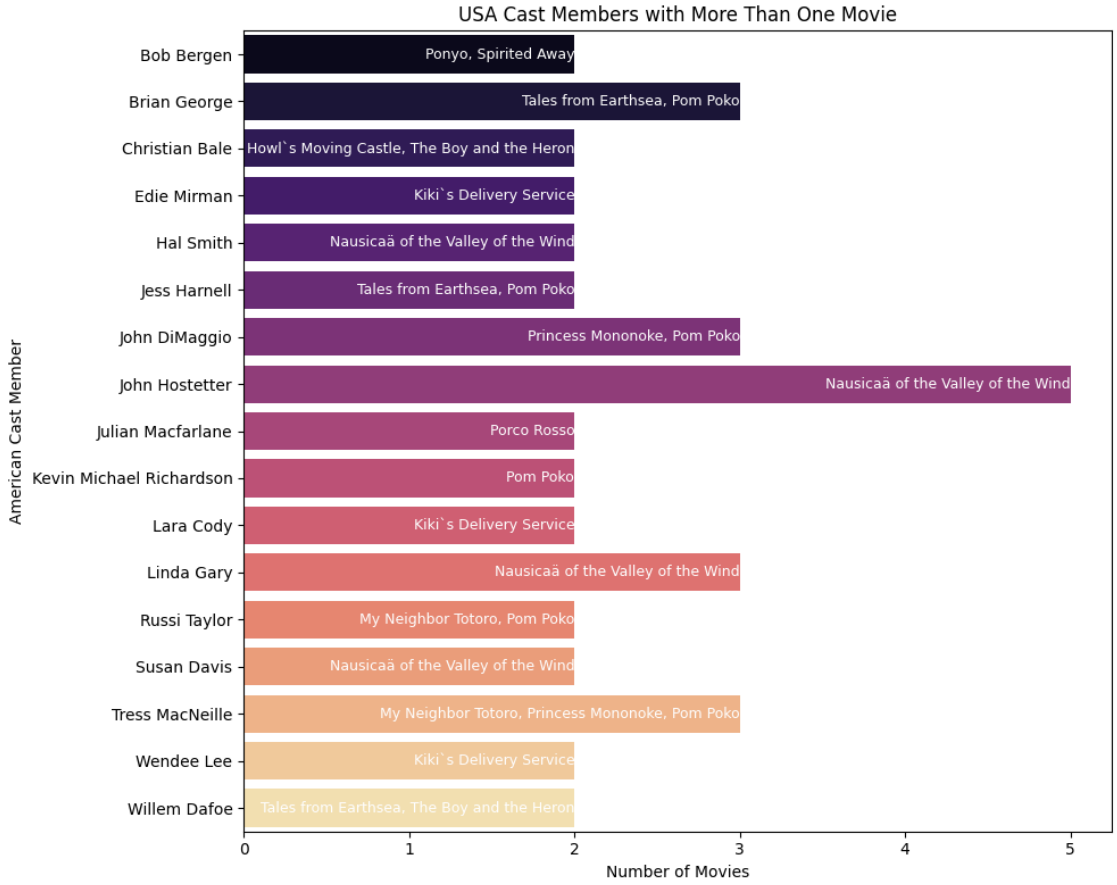
How many actors have done voice dubbing in more than one movie, who are they from both the Japanese and American casts, and in which movies did they appear?

[34]:
```
japan_actor_movie_count = ch.groupby('japanCastMember').size().
 ↪reset_index(name='Count')
actors_more_than_once =␣
 ↪japan_actor_movie_count[japan_actor_movie_count['Count'] > 1]

actors_more_than_once_movies = ch[ch['japanCastMember'].
 ↪isin(actors_more_than_once['japanCastMember'])]
actors_movie_list = actors_more_than_once_movies.
 ↪groupby('japanCastMember')['MovieName'].apply(lambda x: ', '.join(x)).
 ↪reset_index()
actors_more_than_once = actors_more_than_once.merge(actors_movie_list,␣
 ↪on='japanCastMember')

fig, ax = plt.subplots(figsize=(10, 8))
sb.barplot(data=actors_more_than_once, x='Count', y='japanCastMember',␣
 ↪hue='japanCastMember', palette='viridis', ax=ax)
ax.set_title('Japanese Cast Members with More Than One Movie')
ax.set_xlabel('Number of Movies')
ax.set_ylabel('Japanese Cast Member')

# Annotate the bars with movie names
for i in range(len(actors_more_than_once)):
    ax.text(actors_more_than_once['Count'].iloc[i], i,␣
 ↪actors_more_than_once['MovieName'].iloc[i], color='w', ha="right",␣
 ↪va="center", fontsize=10)

plt.tight_layout()
plt.show()
```



[35]:
```
us_actor_movie_count = ch.groupby('USA_CastMember').size().
 ↪reset_index(name='Count')
actors_more_than_once = us_actor_movie_count[us_actor_movie_count['Count'] > 1]

actors_more_than_once_movies = ch[ch['USA_CastMember'].
 ↪isin(actors_more_than_once['USA_CastMember'])]
actors_movie_list = actors_more_than_once_movies.
 ↪groupby('USA_CastMember')['MovieName'].apply(lambda x: ', '.join(x.
 ↪drop_duplicates())).reset_index()
actors_more_than_once = actors_more_than_once.merge(actors_movie_list,␣
 ↪on='USA_CastMember')


fig, ax = plt.subplots(figsize=(10, 8))
sb.barplot(data=actors_more_than_once, x='Count', y='USA_CastMember',␣
 ↪hue='USA_CastMember', palette='magma', ax=ax)
ax.set_title('USA Cast Members with More Than One Movie')
ax.set_xlabel('Number of Movies')
ax.set_ylabel('American Cast Member')
```

```
for i in range(len(actors_more_than_once)):
    ax.text(actors_more_than_once['Count'].iloc[i], i,␣
 ↪actors_more_than_once['MovieName'].iloc[i], color='w', ha="right",␣
 ↪va="center", fontsize=9)

plt.tight_layout()
plt.show()
```



[ ]:

[ ]: