

```

In [ ]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_name = 'best_selling_switch_games.csv'
switch_games_df = pd.read_csv(file_name)

# Display the first few rows of the dataframe
print(switch_games_df.head())

# Display basic information about the dataset
print(switch_games_df.info())

# Select relevant columns and drop rows with missing values
switch_games_df = switch_games_df[['title', 'copies_sold', 'genre', 'developer', 'p
switch_games_df.dropna(inplace=True)

# Show basic statistics
print(switch_games_df.describe())

# Distribution of Copies Sold
plt.figure(figsize=(10, 6))
sns.histplot(switch_games_df['copies_sold'], bins=30, kde=True)
plt.title('Distribution of Copies Sold')
plt.xlabel('Copies Sold (in millions)')
plt.ylabel('Frequency')
plt.show()

# Genre Popularity (Total Copies Sold by Genre)
genre_sales = switch_games_df.groupby('genre')['copies_sold'].sum().sort_values(asc
plt.figure(figsize=(12, 8))
sns.barplot(x=genre_sales.values, y=genre_sales.index)
plt.title('Total Copies Sold by Genre')
plt.xlabel('Copies Sold (in millions)')
plt.ylabel('Genre')
plt.show()

# Publisher Impact (Total Copies Sold by Publisher)
publisher_sales = switch_games_df.groupby('publisher')['copies_sold'].sum().sort_va
plt.figure(figsize=(12, 8))
sns.barplot(x=publisher_sales.values, y=publisher_sales.index)
plt.title('Total Copies Sold by Publisher (Top 10)')
plt.xlabel('Copies Sold (in millions)')
plt.ylabel('Publisher')
plt.show()

# Correlation Heatmap (only includes numeric columns)
plt.figure(figsize=(14, 10))
correlation_matrix = switch_games_df[['copies_sold']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()

```

```

# Top 10 Games by Copies Sold
top_10_games = switch_games_df.nlargest(10, 'copies_sold')
plt.figure(figsize=(12, 8))
sns.barplot(x=top_10_games['copies_sold'], y=top_10_games['title'])
plt.title('Top 10 Games by Copies Sold')
plt.xlabel('Copies Sold (in millions)')
plt.ylabel('Game')
plt.show()

# Genre Popularity Over Time (Assuming release date is available)
# Convert 'release_date' to datetime
switch_games_df['release_date'] = pd.to_datetime(switch_games_df['release_date'])
genre_over_time = switch_games_df.groupby([switch_games_df['release_date'].dt.year,
genre_over_time.plot(kind='line', figsize=(14, 8))
plt.title('Genre Popularity Over Time')
plt.xlabel('Year')
plt.ylabel('Copies Sold (in millions)')
plt.show()

# Sales Contribution by Publisher
publisher_sales = switch_games_df.groupby('publisher')['copies_sold'].sum().sort_value
plt.figure(figsize=(10, 6))
publisher_sales.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Sales Contribution by Publisher')
plt.ylabel('')
plt.show()

# Average Sales by Genre
avg_genre_sales = switch_games_df.groupby('genre')['copies_sold'].mean().sort_value
plt.figure(figsize=(12, 8))
sns.barplot(x=avg_genre_sales.values, y=avg_genre_sales.index)
plt.title('Average Sales by Genre')
plt.xlabel('Average Copies Sold (in millions)')
plt.ylabel('Genre')
plt.show()

# Scatter Plot of Sales by Developer
plt.figure(figsize=(14, 8))
sns.scatterplot(data=switch_games_df, x='developer', y='copies_sold', hue='genre')
plt.title('Sales by Developer')
plt.xlabel('Developer')
plt.ylabel('Copies Sold (in millions)')
plt.xticks(rotation=90)
plt.show()

```

	title	copies_sold	genre
0	Mario Kart 8 Deluxe	48410000.0	Kart racing
1	Animal Crossing: New Horizons	40170000.0	Social simulation
2	Super Smash Bros. Ultimate	29530000.0	Fighting
3	The Legend of Zelda: Breath of the Wild	27790000.0	Action-adventure
4	Pokémon Sword and Shield	25370000.0	Role-playing

	developer	publisher	as_of
0	Nintendo EPD	Nintendo	2022-09-30
1	Nintendo EPD	Nintendo	2022-09-30
2	Bandai Namco StudiosSora Ltd.	Nintendo	2022-09-30
3	Nintendo EPD	Nintendo	2022-09-30
4	Game Freak The Pokémon Company	Nintendo	2022-09-30

	release_date
0	2017-04-28
1	2020-03-20
2	2018-12-07
3	2017-03-03
4	2019-11-15

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 73 entries, 0 to 72

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	title	73 non-null	object
1	copies_sold	73 non-null	float64
2	genre	73 non-null	object
3	developer	73 non-null	object
4	publisher	73 non-null	object
5	as_of	73 non-null	object
6	release_date	73 non-null	object

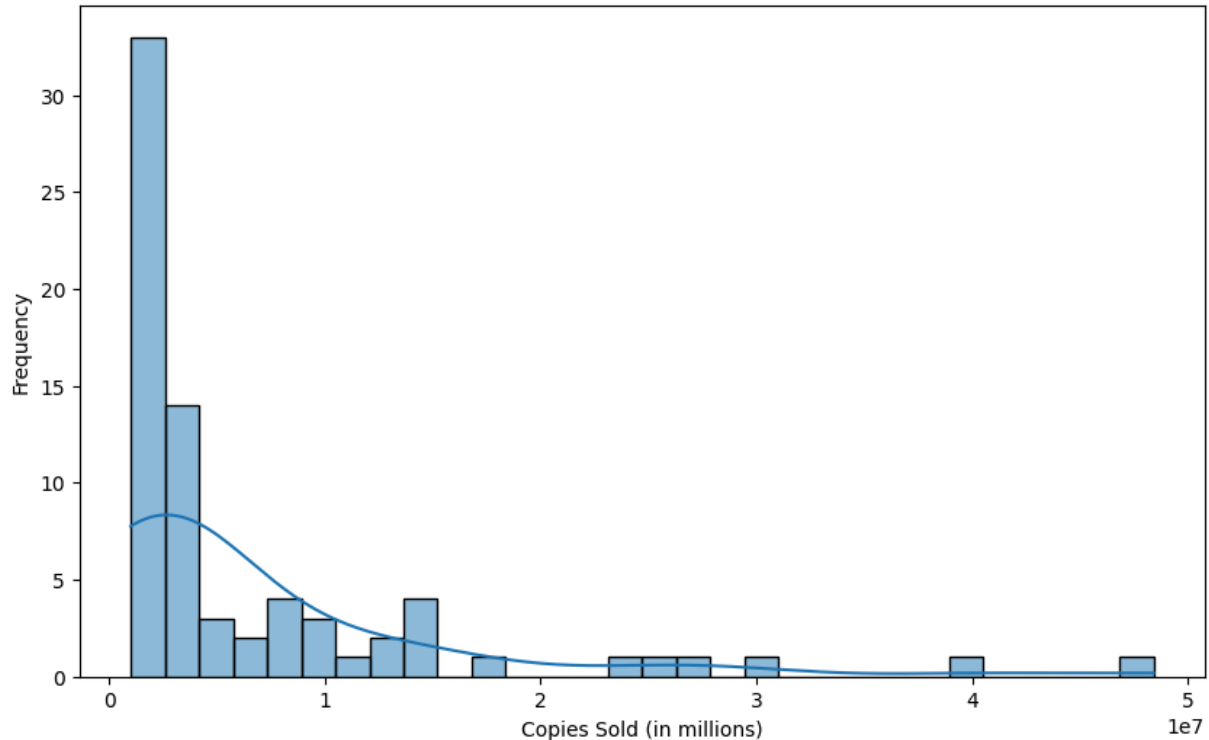
dtypes: float64(1), object(6)

memory usage: 4.1+ KB

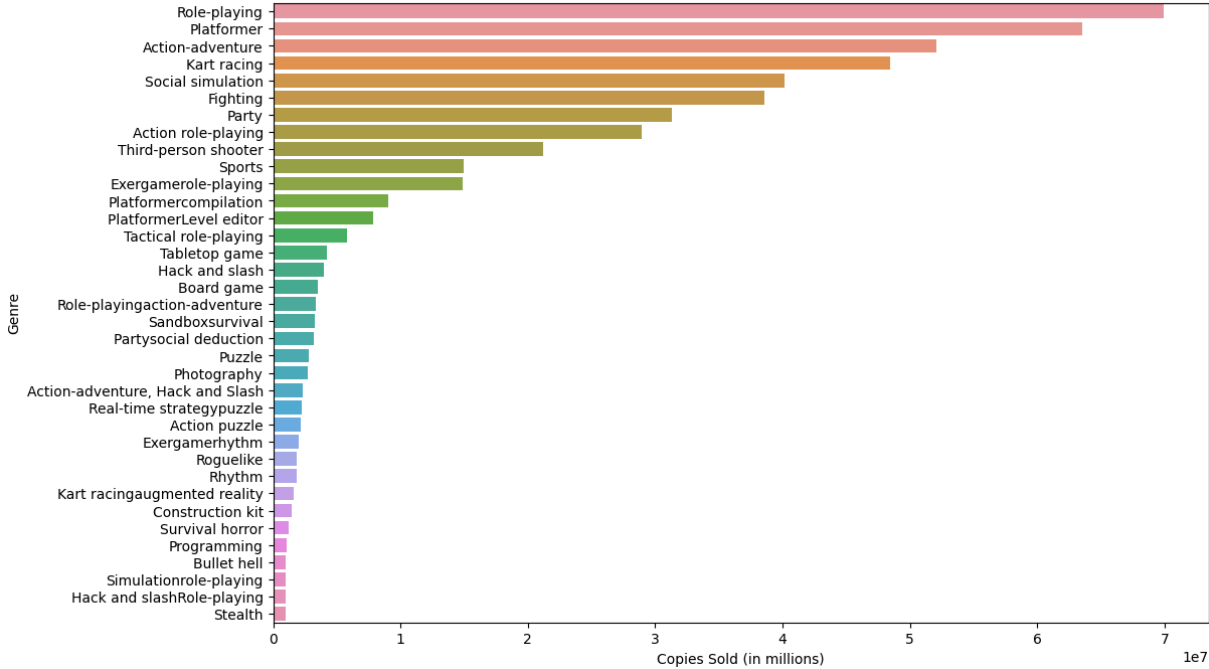
None

	copies_sold
count	7.300000e+01
mean	6.788904e+06
std	9.179178e+06
min	1.000000e+06
25%	1.580000e+06
50%	3.010000e+06
75%	7.900000e+06
max	4.841000e+07

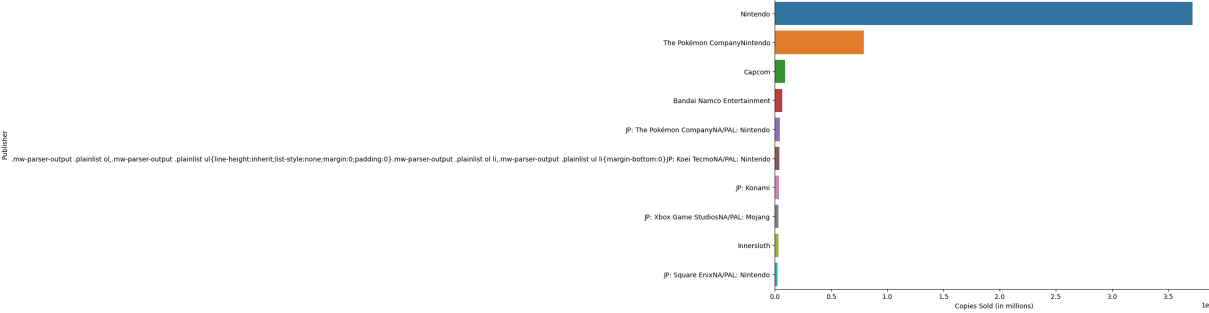
Distribution of Copies Sold

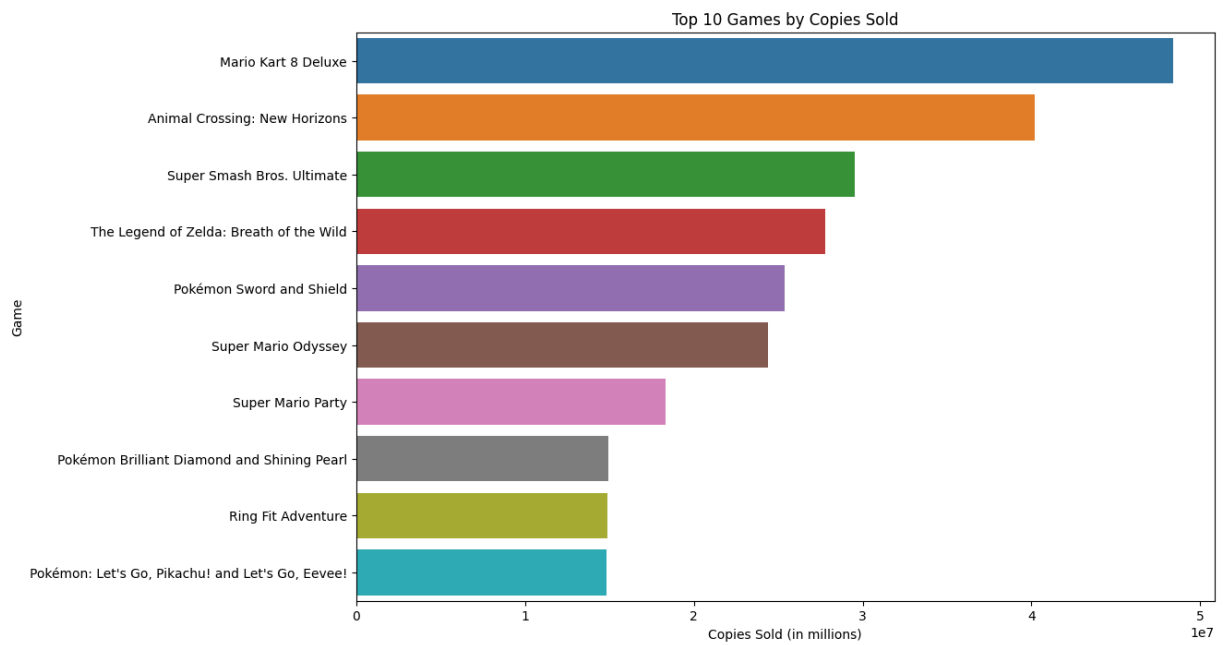
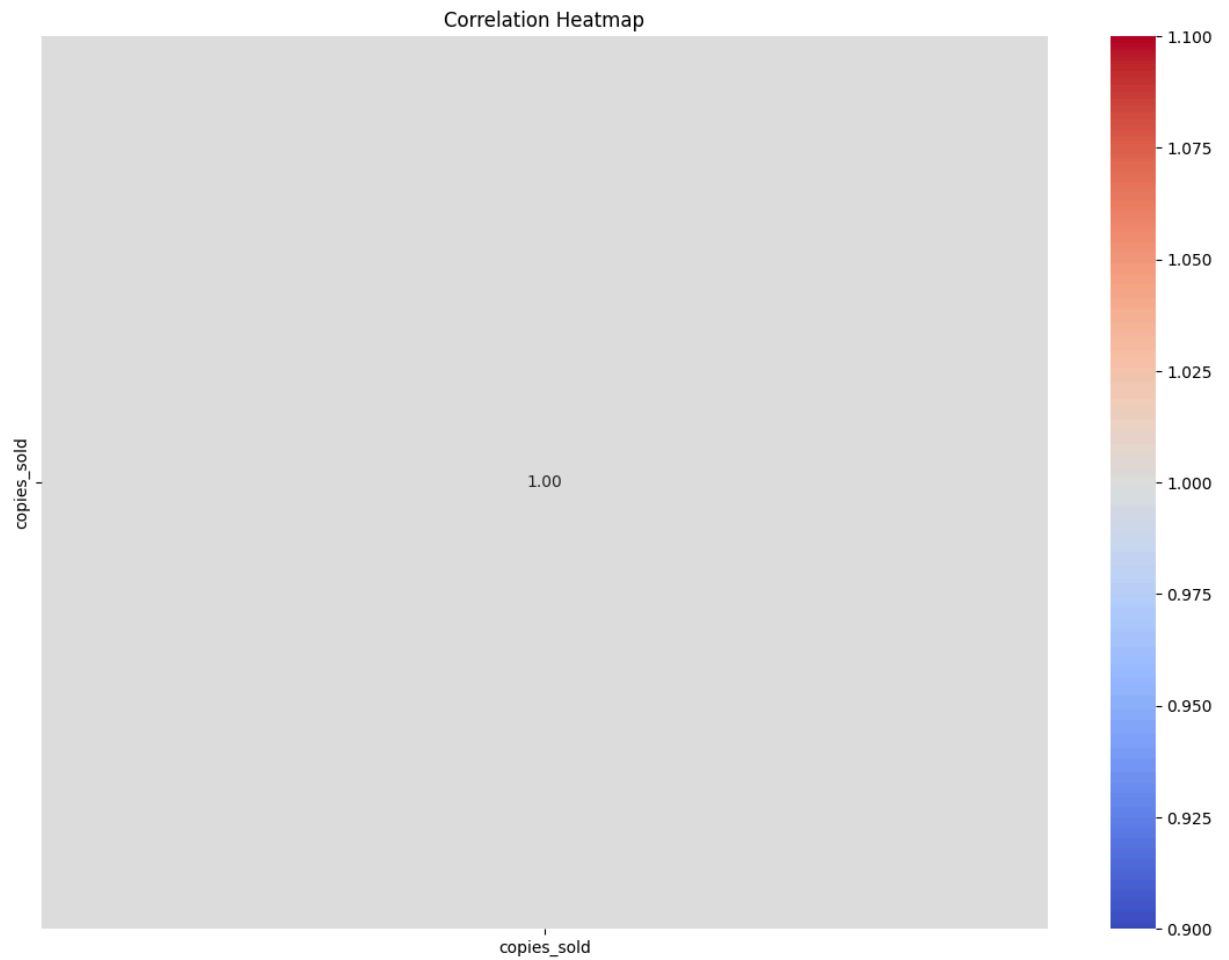


Total Copies Sold by Genre



Total Copies Sold by Publisher (Top 10)





```

-----
KeyError                                Traceback (most recent call last)
File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pandas\core\indexes\base.py:3653, in Index.get_loc(self, key)
    3652 try:
-> 3653     return self._engine.get_loc(casted_key)
    3654 except KeyError as err:

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pandas\_libs\index.pyx:147, in pandas._libs.index.IndexEngine.get_loc()

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pandas\_libs\index.pyx:176, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:7080, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.PyObjectHashTable.get_item()

```

**KeyError:** 'release\_date'

The above exception was the direct cause of the following exception:

```

KeyError                                Traceback (most recent call last)
Cell In[8], line 66
    62 plt.show()
    64 # Genre Popularity Over Time (Assuming release date is available)
    65 # Convert 'release_date' to datetime
--> 66 switch_games_df['release_date'] = pd.to_datetime(switch_games_df['release_date'])
    67 genre_over_time = switch_games_df.groupby([switch_games_df['release_date'].dt.year, 'genre'])['copies_sold'].sum().unstack().fillna(0)
    68 genre_over_time.plot(kind='line', figsize=(14, 8))

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pandas\core\frame.py:3761, in DataFrame.__getitem__(self, key)
    3759 if self.columns.nlevels > 1:
    3760     return self._getitem_multilevel(key)
-> 3761 indexer = self.columns.get_loc(key)
    3762 if is_integer(indexer):
    3763     indexer = [indexer]

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pandas\core\indexes\base.py:3655, in Index.get_loc(self, key)
    3653     return self._engine.get_loc(casted_key)
    3654 except KeyError as err:
-> 3655     raise KeyError(key) from err
    3656 except TypeError:
    3657     # If we have a listlike key, _check_indexing_error will raise
    3658     # InvalidIndexError. Otherwise we fall through and re-raise

```

```
3659     # the TypeError.  
3660     self._check_indexing_error(key)
```

```
KeyError: 'release_date'
```

In [ ]: