Table of Contents

1 Final Project Submission

Please fill out:

- Student name: Daniel Ross-Leutwyler
- Student pace: **self paced** / part time / full time
- Scheduled project review date/time: 8/12/21, 5:00 PM
- Instructor name: James Irving
- Blog post URL:

2 Overview

In this project I will make suggestions about what sort of movies Microsoft should make for the launch of their movie studio.

Using data from IMDB, Box Office Mojo, and The Numbers, I will look at historical data from 2010-2018 to analyze what sort of. movies perform the best at the box office.

2.1 Initial Thoughts and Response to the 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 don'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."

There are lots of metrics to measure success, however some are easier to objectively measure than others. With the data at hand the two main ways to judge the success of a movie are through analyzing key financial metrics (gross box office, and return-on-investment, for example), and critical response (as aggregated by the IMBD website).

By analyzing what, if any, attributes the top grossing movies all share, we can make informed suggestions about what sort of film Microsoft should pursue.

I will also explore the relationship between genre, budget, and gross revenue. The success of a blockbuster is in part dependent upon its ROI, as well.

The analysis performed here will attempt to answer these three questions:

Q1) What are the highest grossing genres from 2010-2018?

Q2) What genres have the highest revenue and ROI over that same time frame?

Q3) Which genres have a lower production budget and a higher ROI?

These questions will help Microsoft make business savvy decisions about their entry into a crowded market.

3 Data Cleaning, Merging, and Aggregating

3.1 Importing Modules and Relevent Datasets

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# suppressing scientific notation, and adding ',' to long values for le
# adapted from https://re-thought.com/how-to-suppress-scientific-notati
pd.options.display.float_format = '{:,.2f}'.format
# setting sns context to talk for clarity and size
sns.set_context('talk')
```

These are the data sets are relevant to answering the stakeholder questions, as they contain information related to gross sales, production budget, and genre.

In [2]:
```
# creating variables to call for opening csv
box_office_gross = 'zippedData/bom.movie_gross.csv'
title_basics = 'zippedData/imdb.title.basics.csv'
budget_url = 'zippedData/tn.movie_budgets.csv'
```

These data sets are not relevant to the business question, due to outdated or incomplete data, or containing data that is beyond the scope of our business question.

imdb.title.ratings.csv.gz
imdb.name.basics.csv
imbd.title.akas.csv
imdb.title.crew.csv
imdb.principles.csv
rt.movie_info.tsv
rt.reviews.tsv
tmdb.movies.csv

In [3]:
```
# Defining functions to help with EDA

def prelim(df):
    """displays core information
    on a dataframe at the beginning of EDA

    accepts a DataFrame as input, and displays
    the head, info, and sum of all null values
    for each column in that DataFrame
    """
    return (display(df.head()),
            display(df.info()),
            display(df.isna().sum()))

def see_nans(df, cols=None):
    """accepts a data frame, and optionally columns
    returns a data frame of all null values.

    Used for previewing missing data.
    Does not alter df in any way"""

    if cols is None:
        cols = df.columns
    return df[df[cols].isnull().any(axis=1)]
```

3.2 Cleaning Box Office Gross Data

In [4]:
```
gross_df = pd.read_csv(box_office_gross)
prelim(gross_df)
```

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415,000,000.00	652000000	2010
1	Alice in Wonderland (2010)	BV	334,200,000.00	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296,000,000.00	664300000	2010
3	Inception	WB	292,600,000.00	535700000	2010
4	Shrek Forever After	P/DW	238,700,000.00	513900000	2010

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB

None

title                0
studio               5
domestic_gross      28
foreign_gross     1350
year                 0
dtype: int64
```

Out[4]: (None, None, None)

gross_df is a DataFrame of movies from 2011-2018. Each row represents one movie, and each column contains the the following values:

'title', 'studio', 'domestic_gross', 'foreign_gross', and 'year'.

There are no null entries in the ***title*** column, which contains objects.

There are 5 null entries in the ***studio*** column, which contains objects.

There are 28 null values in the ***domestic_gross*** column, which contains numbers, specifically floats.

There are 1350 null values in the ***foreign_gross*** column, which contains objects, and will need to be cast as a float. The null values will need to be replaced.

hypothesis: foreign_gross should be summed with domestic_gross, as the global nature of media today makes the foreign/domestic binary less important.

There are no nulls in the ***year*** column, which contains integers.

Examining the 'foreign_gross' column:

In [5]:
```
# identifying existing ',' in the strings that needs to be replaced
# these are literal strings, not the ',' from the .format in 3.1
gross_df['foreign_gross'].str.contains(',').sum()
```

Out[5]: 5

In [6]:
```
# removing commas to be able to cast as float and sanity check
gross_df['foreign_gross'] = gross_df['foreign_gross'].str.replace(',','
gross_df['foreign_gross'] = gross_df['foreign_gross'].astype(float)
gross_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   float64
 4   year            3387 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 132.4+ KB
```

In [7]:

```
# investigating the null values
see_nans(gross_df, ['foreign_gross']).head(20)
```

Out[7]:

	title	studio	domestic_gross	foreign_gross	year
222	Flipped	WB	1,800,000.00	nan	2010
254	The Polar Express (IMAX re-issue 2010)	WB	673,000.00	nan	2010
267	Tiny Furniture	IFC	392,000.00	nan	2010
269	Grease (Sing-a-Long re-issue)	Par.	366,000.00	nan	2010
280	Last Train Home	Zeit.	288,000.00	nan	2010
287	Sweetgrass	CGld	207,000.00	nan	2010
291	Casino Jack and the United States of Money	Magn.	177,000.00	nan	2010
308	Alamar	FM	61,600.00	nan	2010
311	Hatchet 2	Vita.	52,600.00	nan	2010
319	Living in Emergency	Truly	32,200.00	nan	2010
323	The Taqwacores	Strand	11,400.00	nan	2010
324	Cherry	Abr.	11,400.00	nan	2010
325	Terkel in Trouble	Indic.	10,800.00	nan	2010
326	Kimjongilia	Lorb.	4,400.00	nan	2010
458	Courageous	TriS	34,500,000.00	nan	2011
473	Our Idiot Brother	Wein.	24,800,000.00	nan	2011
511	Straw Dogs (2011)	SGem	10,300,000.00	nan	2011
512	Prom	BV	10,100,000.00	nan	2011
524	Take Me Home Tonight	Rela.	6,900,000.00	nan	2011
525	Cedar Rapids	FoxS	6,900,000.00	nan	2011

A cursory search of the foreign box office receipts for several movies on this list demonstrates that while some of these movies did not have a foreign theatrical release (Flipped), it appears that some of the movies foreign box office receipts have already been counted in the domestic gross category (Courageous). Others have simply had that info omitted. On balance, the movies missing the foreign_gross data are not on the upper end of the the domestic_gross category, rendering their relevance minimal. In order to aggregate the data in the foreign_gross column, I will replace all the Nan values in this column with 0.

In [8]:

```
# replacing NaNs with 0 and sanity check
gross_df['foreign_gross'] = gross_df['foreign_gross'].fillna(0)
see_nans(gross_df, ['foreign_gross']).head()
```

Out[8]:

title	studio	domestic_gross	foreign_gross	year

In [9]:

```
# discovering irregularities in foreign gross column
gross_df.sort_values('domestic_gross', ascending=False).head()
```

Out[9]:

	title	studio	domestic_gross	foreign_gross	year
1872	Star Wars: The Force Awakens	BV	936,700,000.00	1,131.60	2015
3080	Black Panther	BV	700,100,000.00	646,900,000.00	2018
3079	Avengers: Infinity War	BV	678,800,000.00	1,369.50	2018
1873	Jurassic World	Uni.	652,300,000.00	1,019.40	2015
727	Marvel's The Avengers	BV	623,400,000.00	895,500,000.00	2012

It is clear from looking at the domestic_gross sorted that the foreign_gross values are incorrect. It is not possible that there were only $1,131 of receipts for Star Wars the Force Awakens, and IMDB confirms this.

It appears that the incorrect amount is due to an entry error, easily fixed by using a mapping dictionary.

In [10]:

```
# creating dictionary to replace incorrect values.
# correct values taken from IMBD

mapping_dict = {1010.00 : 1009996733,
               1019.40 : 1018130819,
               1131.60 : 1132859475,
               1163.00 : 1162334379,
               1369.50 : 1369544272}

gross_df['foreign_gross'] = gross_df['foreign_gross'].replace(mapping_d
```

Now we can aggregate data from the domestic and foreign gross columns to make inferences.

In [11]:
```
# exploring the difference in domestic and foreign gross
print(f"The domestic gross sum is: ${round(gross_df['domestic_gross'].s
print(f"The domestic gross mean is: ${round(gross_df['domestic_gross'].
print(f"The domestic gross std is: ${round(gross_df['domestic_gross'].s
print(f"The foreign gross sum is: ${round(gross_df['foreign_gross'].sum
print(f"The foreign gross mean is: ${round(gross_df['foreign_gross'].me
print(f"The foreign gross std is: ${round(gross_df['foreign_gross'].std
```

```
The domestic gross sum is: $96,557,293,580.0
The domestic gross mean is: $28,745,845.0
The domestic gross std is: $66,982,498.0
The foreign gross sum is: $158,208,774,261.0
The foreign gross mean is: $46,710,592.0
The foreign gross std is: $120,344,901.0
```

Above we can see that the both the total foreign box office receipts (even with the 1350 replaced data points) and the foreign box office mean are ***higher*** than domestic.

This means it ***may*** warrant giving special consideration to movies that performed well in foreign markets.

I will also add an additional column for total_gross summing the domestic and foreign columns, as this is a feature that is absent in the original data.

In [12]:
```
gross_df['total_gross'] = (gross_df['domestic_gross'] +
                           gross_df['foreign_gross'])

# creating top_100_x dataframes

top_100_domestic = gross_df.sort_values('domestic_gross',
                                        ascending=False)[:100]
top_100_foreign = gross_df.sort_values('foreign_gross', ascending=False
top_100_total = gross_df.sort_values('total_gross', ascending=False)[:1

display(top_100_domestic.head(10))
display(top_100_foreign.head(10))
display(top_100_total.head(10))
```

	title	studio	domestic_gross	foreign_gross	year	total_gross
1872	Star Wars: The Force Awakens	BV	936,700,000.00	1,132,859,475.00	2015	2,069,559,475.00
3080	Black Panther	BV	700,100,000.00	646,900,000.00	2018	1,347,000,000.00
3079	Avengers: Infinity War	BV	678,800,000.00	1,369,544,272.00	2018	2,048,344,272.00
1873	Jurassic World	Uni.	652,300,000.00	1,018,130,819.00	2015	1,670,430,819.00
727	Marvel's The Avengers	BV	623,400,000.00	895,500,000.00	2012	1,518,900,000.00
2758	Star Wars: The Last Jedi	BV	620,200,000.00	712,400,000.00	2017	1,332,600,000.00
3082	Incredibles 2	BV	608,600,000.00	634,200,000.00	2018	1,242,800,000.00
2323	Rogue One: A Star Wars Story	BV	532,200,000.00	523,900,000.00	2016	1,056,100,000.00
2759	Beauty and the Beast (2017)	BV	504,000,000.00	759,500,000.00	2017	1,263,500,000.00
2324	Finding Dory	BV	486,300,000.00	542,300,000.00	2016	1,028,600,000.00

	title	studio	domestic_gross	foreign_gross	year	total_gross
3079	Avengers: Infinity War	BV	678,800,000.00	1,369,544,272.00	2018	2,048,344,272.00
1874	Furious 7	Uni.	353,000,000.00	1,162,334,379.00	2015	1,515,334,379.00
1872	Star Wars: The Force Awakens	BV	936,700,000.00	1,132,859,475.00	2015	2,069,559,475.00
1873	Jurassic World	Uni.	652,300,000.00	1,018,130,819.00	2015	1,670,430,819.00
2760	The Fate of the Furious	Uni.	226,000,000.00	1,009,996,733.00	2017	1,235,996,733.00
328	Harry Potter and the Deathly Hallows Part 2	WB	381,000,000.00	960,500,000.00	2011	1,341,500,000.00
1875	Avengers: Age of Ultron	BV	459,000,000.00	946,400,000.00	2015	1,405,400,000.00
727	Marvel's The Avengers	BV	623,400,000.00	895,500,000.00	2012	1,518,900,000.00
3081	Jurassic World: Fallen Kingdom	Uni.	417,700,000.00	891,800,000.00	2018	1,309,500,000.00
1127	Frozen	BV	400,700,000.00	875,700,000.00	2013	1,276,400,000.00

	title	studio	domestic_gross	foreign_gross	year	total_gross

	title	studio	domestic_gross	foreign_gross	year	total_gross
1872	Star Wars: The Force Awakens	BV	936,700,000.00	1,132,859,475.00	2015	2,069,559,475.00
3079	Avengers: Infinity War	BV	678,800,000.00	1,369,544,272.00	2018	2,048,344,272.00
1873	Jurassic World	Uni.	652,300,000.00	1,018,130,819.00	2015	1,670,430,819.00
727	Marvel's The Avengers	BV	623,400,000.00	895,500,000.00	2012	1,518,900,000.00
1874	Furious 7	Uni.	353,000,000.00	1,162,334,379.00	2015	1,515,334,379.00
1875	Avengers: Age of Ultron	BV	459,000,000.00	946,400,000.00	2015	1,405,400,000.00
3080	Black Panther	BV	700,100,000.00	646,900,000.00	2018	1,347,000,000.00
328	Harry Potter and the Deathly Hallows Part 2	WB	381,000,000.00	960,500,000.00	2011	1,341,500,000.00
2758	Star Wars: The Last Jedi	BV	620,200,000.00	712,400,000.00	2017	1,332,600,000.00
3081	Jurassic World: Fallen Kingdom	Uni.	417,700,000.00	891,800,000.00	2018	1,309,500,000.00

In this section we have cleaned, aggregated, and sorted the data so we can attempt to answer questions about the gross sales of different movies across the domestic and foreign markets.

3.3 Cleaning 'Title Basics'.

In [13]:
```
title_basics_df = pd.read_csv(title_basics)
prelim(title_basics_df)
```

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.00	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.00	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.00	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	nan	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.00	Comedy,Drama,Fantasy

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           146144 non-null  object 
 1   primary_title    146144 non-null  object 
 2   original_title   146123 non-null  object 
 3   start_year       146144 non-null  int64  
 4   runtime_minutes  114405 non-null  float64
 5   genres           140736 non-null  object 
dtypes: float64(1), int64(1), object(4)
memory usage: 6.7+ MB

None

tconst                 0
primary_title          0
original_title        21
start_year             0
runtime_minutes    31739
genres              5408
dtype: int64
```

Out[13]:
```
(None, None, None)
```

The title.basics table appears to contain a data frame primarily of movies from 2010-2021, but also with some titles from ***THE FUTURE***. Each row represents one movie, and contains columns with the following values:

'tconst', 'primary_title', 'original_title', 'start_year','runtime_minutes', 'genres'

There are no null entries in the tconst column, which is an object, as I would expect, and this column can be used as the index to ***join this df with other dfs that are similarly formatted.***

There are no null entries in the start_year column, which is an integer, as I would expect. ***This column will require cleaning to deal with movies from the future.***

There are significant null entires in the runtime_minutes column, which may be of questionable use. ***It may be worth exploring the relationship between movie length and box office success.***

There are 5408 null values in the genres column. This column will require more exploration as we will need it to help make decisions about what kind of movies Microsoft should be making.

I will first merge this DataFrame with the gross_df to focus our analysis on movies that we have complete gross sales data on. There are two potential columns in the title_basics_df that we can use to merge: 'primary_title' and 'original_title'. Below are two different dfs that show us which column has more overlap with the gross_df.

In [14]:
```
# checking the size of the merged DataFrames

df_orig = pd.merge(title_basics_df, gross_df, left_on='original_title',
              right_on='title')
df_pri = pd.merge(title_basics_df, gross_df, left_on='primary_title',
              right_on='title')
display(len(df_orig))
display(len(df_pri))
```

```
2776

3366
```

The 'primary_title' series has more overlap with gross_df. This is the merged df that we will use for the rest of our EDA in this section.

Exploring the 'genre' column:

In [15]:
```
df_pri['genres'].value_counts()
```

Out[15]:
```
Drama                         392
Documentary                   168
Comedy,Drama,Romance          138
Comedy,Drama                  137
Drama,Romance                 115
                             ...
Biography,Comedy                1
History                         1
Documentary,Drama,Music         1
Comedy,Mystery,Sci-Fi           1
Adventure,Drama,Sport           1
Name: genres, Length: 331, dtype: int64
```

In [16]:
```
# splitting the genres into lists in a new column

df = df_pri.copy()
df['genres'] = df['genres'].str.split(',')

# 'exploding' the 'genres_split' list to get multiple entries for each

df = df.explode('genres')

# cleaning redundencies in genre
df['genres'] = df['genres'].str.replace('Musical', 'Music')

# sanity check
df['genres'].value_counts()
```

Out[16]:
```
Drama          1876
Comedy          965
Action          664
Romance         483
Thriller        479
Adventure       446
Crime           390
Documentary     334
Biography       306
Horror          261
Mystery         221
Fantasy         177
Animation       157
History         149
Sci-Fi          139
Family          129
Music           117
Sport            57
War              53
Western          22
News              6
Name: genres, dtype: int64
```

In this section we have cleaned, aggregated, and sorted the data so we can attempt to answer questions about the average gross sales of different genres.

3.4 Cleaning 'Movie Budgets'

In [17]:
```
budget_df = pd.read_csv(budget_url)
prelim(budget_df)
```

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
3	4	May 1, 2015	Avengers: Age of Ultron	$330,600,000	$459,005,868	$1,403,013,963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	$317,000,000	$620,181,382	$1,316,721,747

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   id                 5782 non-null   int64 
 1   release_date       5782 non-null   object
 2   movie              5782 non-null   object
 3   production_budget  5782 non-null   object
 4   domestic_gross     5782 non-null   object
 5   worldwide_gross    5782 non-null   object
dtypes: int64(1), object(5)
memory usage: 271.2+ KB

None

id                   0
release_date         0
movie                0
production_budget    0
domestic_gross       0
worldwide_gross      0
dtype: int64
```

Out[17]:
```
(None, None, None)
```

The budget_df contains data about the release date, movie title, production budget, and box office gross. Since we already have cleaned gross data, we will be merging only the production budget and movie name with our existing df.

Additionally we will have to clean and recast the production_budget column.

In [18]:
```python
# merge dfs

df = pd.merge(df, budget_df, left_on='primary_title', right_on='movie')

# clean, recast budget column as float
df['production_budget'] = df['production_budget'].str.strip(
    '$').str.replace(',','').astype(float)

# dropping irrelevent columns
df = df.drop(['id', 'release_date', 'movie', 'domestic_gross_y',
              'worldwide_gross', 'start_year', 'original_title',
             'primary_title', 'studio'], axis=1)

# and sanity check
df.sort_values('production_budget', ascending=False).head()
```

Out[18]:

	tconst	runtime_minutes	genres	title	domestic_gross_x	foreign_gross	year	
928	tt1298650	136.00	Action	Pirates of the Caribbean: On Stranger Tides	241,100,000.00	804,600,000.00	2011	1,(
929	tt1298650	136.00	Adventure	Pirates of the Caribbean: On Stranger Tides	241,100,000.00	804,600,000.00	2011	1,(
930	tt1298650	136.00	Fantasy	Pirates of the Caribbean: On Stranger Tides	241,100,000.00	804,600,000.00	2011	1,(
2797	tt2395427	141.00	Sci-Fi	Avengers: Age of Ultron	459,000,000.00	946,400,000.00	2015	1,4
2796	tt2395427	141.00	Adventure	Avengers: Age of Ultron	459,000,000.00	946,400,000.00	2015	1,4

In order to deepen our understanding of the profitability of movies I will engineer two new features to our DataFrame:

revenue, defined as total_gross - production_budget

ROI, expressed as a percentage, defined as (revenue / production_budget) * 100

In [19]:
```python
df['revenue'] = df['total_gross'] - df['production_budget']
df['ROI'] = df['revenue'] / df['production_budget'] * 100
```

This DataFrame is now in its final form that we will use to solve for our question and to create plots.

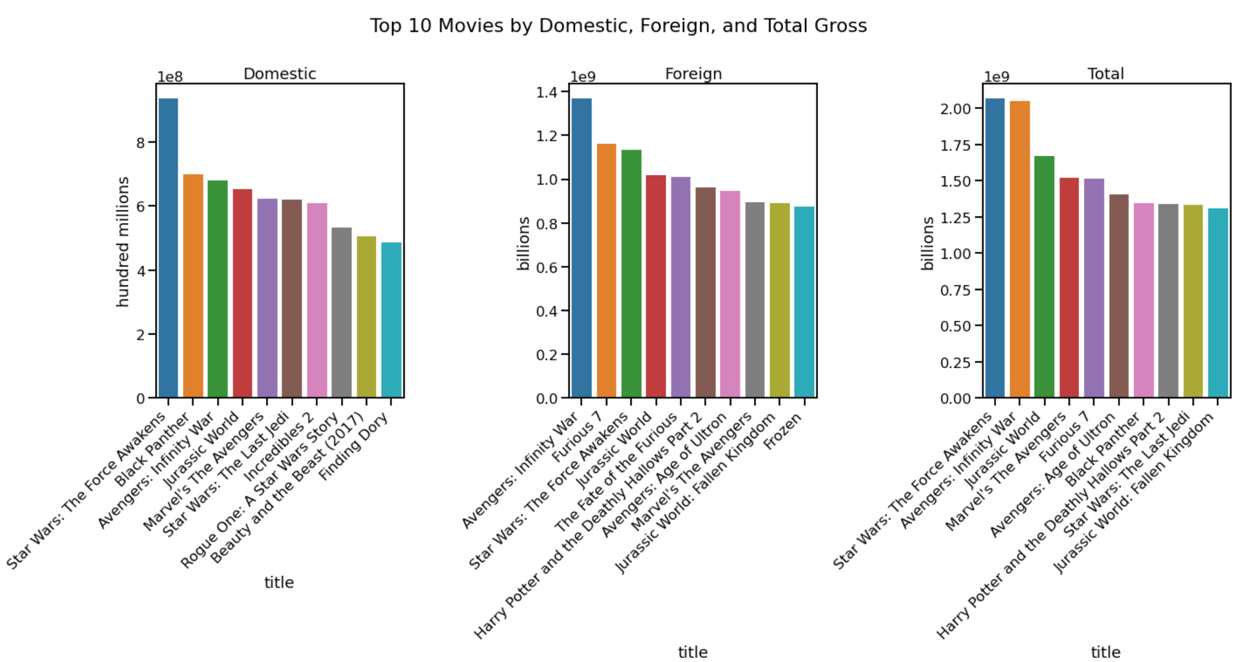
4 Q1 What are the Highest Grossing Genres from 2010-2018

Previously we cleaned and sorted our data to see which movies are the top grossing across both the domestic and foreign markets.

In [20]:

```
# plotting bar charts for the top 10 movies, by gross
fig, (ax1, ax2, ax3) = plt.subplots(1 ,3 , figsize=(20,10))

sns.barplot(data=top_100_domestic.head(10), x='title',
            y='domestic_gross', ax=ax1)
sns.barplot(data=top_100_foreign.head(10), x='title',
            y='foreign_gross', ax=ax2)
sns.barplot(data=top_100_total.head(10), x='title',
            y='total_gross', ax=ax3)

ax1.set_title('Domestic')
ax1.set_ylabel('hundred millions')
ax1.set_xticklabels(ax1.get_xticklabels(),rotation=45, ha='right')

ax2.set_title('Foreign')
ax2.set_ylabel('billions')
ax2.set_xticklabels(ax2.get_xticklabels(),rotation=45, ha='right')

ax3.set_title('Total')
ax3.set_ylabel('billions')
ax3.set_xticklabels(ax3.get_xticklabels(),rotation=45, ha='right')

plt.tight_layout()
plt.suptitle('Top 10 Movies by Domestic, Foreign, and Total Gross', y=1
;
```

Out[20]: ''

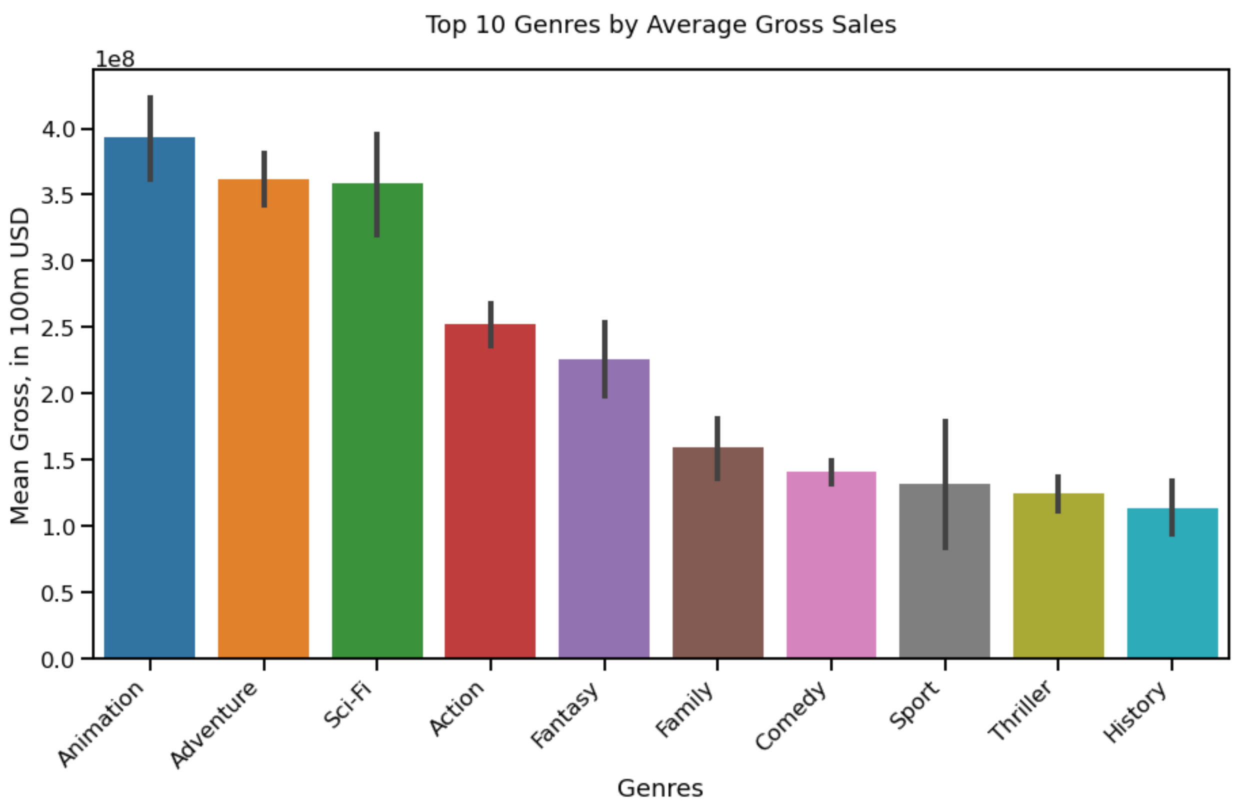


The selected data demonstrates that the top grossing movies between 2010-2018 are, overwhelmingly, part of a series. In fact, out of the top 10 by total sales only one movie, Black Panther, is not part of a series.

It is hard to make any recommendations based on this data. Both the Marvel and DC brands have strong existing relationships with particular studios, as do the Star Wars, Jurassic Park, Harry Potter, and Fast and Furious franchises. It is important to recognize the barriers to entry in creating a blockbuster movie that is part of a series.

Exploring how this data relates to genre will allow for more direct suggestions to be made.

In [21]:
```
# calculating the mean value by genre, and ordering by sales for our pl
df_sales = df.groupby('genres').mean()
df_sales = df_sales.sort_values('total_gross', ascending=False)
sales_order = df_sales.head(10).index

# plotting top 10 genres by avg. sales
fig, ax = plt.subplots(figsize=(15,8))
sns.barplot(data=df.sort_values('total_gross',ascending=False),
                                x= 'genres', y='total_gross', ci=68, ax
            order=sales_order)
ax.set_title('Top 10 Genres by Average Gross Sales', y=1.05)
ax.set_xlabel('Genres')
ax.set_ylabel('Mean Gross, in 100m USD')
ax.set_xticklabels(ax.get_xticklabels(),rotation=45, ha='right');
```



The key take away is that sci-fi, adventure, and animated films all perform exceptionally well at the box office. The second tier of genres are action, fantasy, and family movies. If Microsoft's emphasis is on gross sales these are the genres I recommend they focus on.

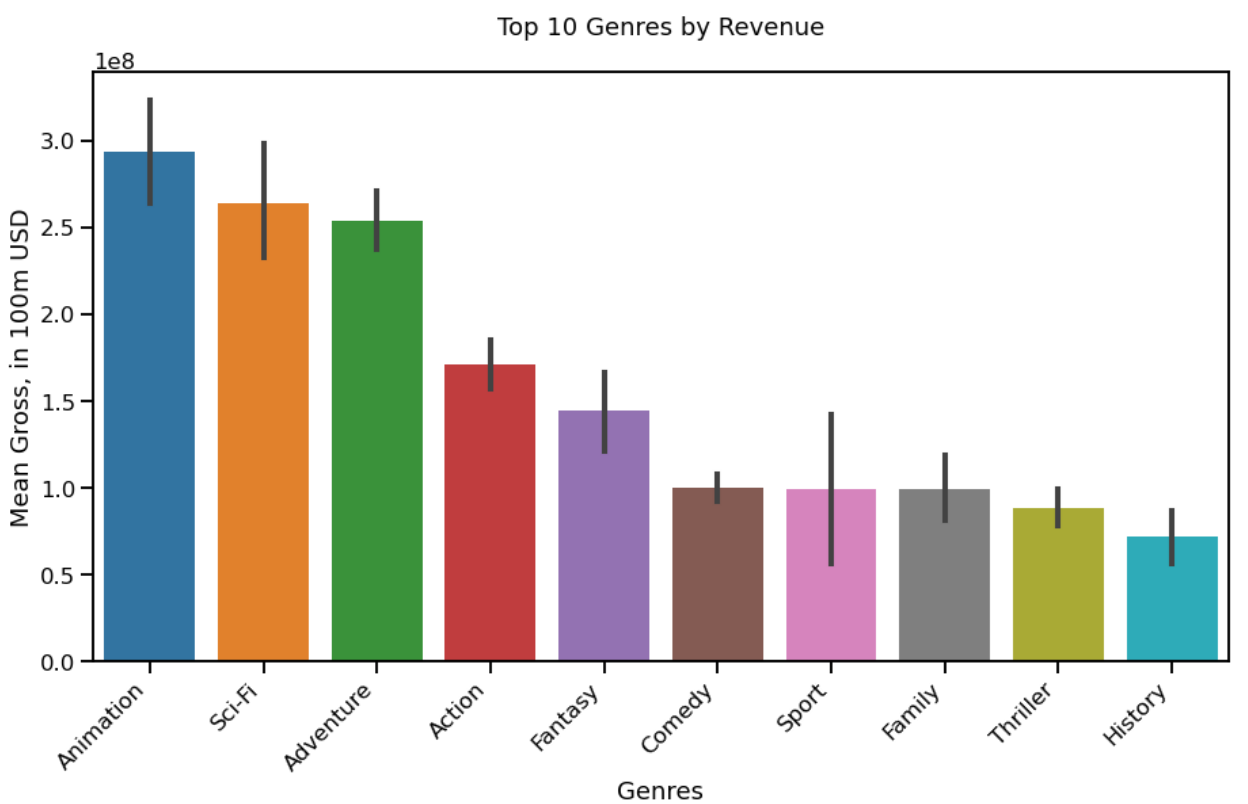
However, given the crowded nature of these genres (the dominance of top grossing movies by series, franchises, and existing studios), it is important to look at both revenue and ROI as supplementary measures of success.

5 Q2

5.1 What Genres Have the Highest Revenue

In [22]:

```
# calculating the mean value by genre, and ordering by revenue for our
df_revenue = df.groupby('genres').mean()
df_revenue = df_revenue.sort_values('revenue', ascending=False)
revenue_order = df_revenue.head(10).index

fig, ax = plt.subplots(figsize=(15,8))
sns.barplot(data=df.sort_values('revenue',ascending=False),
                                x= 'genres', y='revenue', ci=68, ax=ax,
            order=revenue_order)
ax.set_title('Top 10 Genres by Revenue', y=1.05)
ax.set_xlabel('Genres')
ax.set_ylabel('Mean Gross, in 100m USD')
ax.set_xticklabels(ax.get_xticklabels(),rotation=45, ha='right');
```

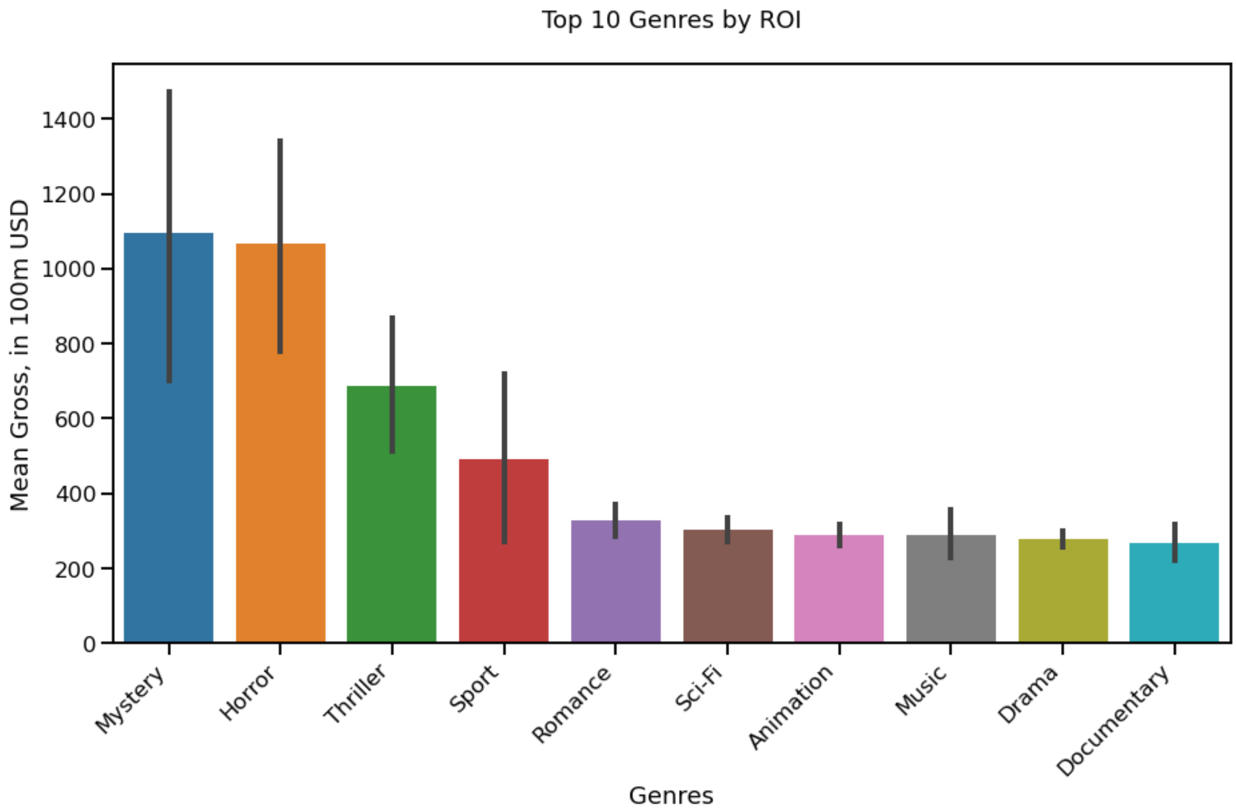


Perhaps unsurprisingly, the plot of genres by revenue very closely mirrors the plot of genres by gross sales. ***If Microsoft is interested in making movies that have high revenue, my recommendation would be to invest in making movies in the Animation, Sci-Fi, and Advenure genres***

5.2 Which Genres Have the Highest ROI

In [23]:

```
# calculating the mean value by genre, and ordering by ROI for our plot
df_ROI = df.groupby('genres').mean()
df_ROI = df_ROI.sort_values('ROI', ascending=False)
ROI_order = df_ROI.head(10).index

fig, ax = plt.subplots(figsize=(15,8))
sns.barplot(data=df.sort_values('ROI',ascending=False),
                                x= 'genres', y='ROI', ci=68, ax=ax,
           order=ROI_order)
ax.set_title('Top 10 Genres by ROI', y=1.05)
ax.set_xlabel('Genres')
ax.set_ylabel('Mean Gross, in 100m USD')
ax.set_xticklabels(ax.get_xticklabels(),rotation=45, ha='right');
```



When genres are sorted by their ROI, we see a completely different set of genres. Granted, ***the exceptionally large standard deviation of the top 4 genres here suggest there are outliers that are disproportionately skewing these results*** but it is interesting to note that that the top 4 genres do not even place in the top 5 by gross sales or revenue.

If Microsoft wanted to take a more conservative, risk averse approach to their movie studio, I would recommend making movies in the Mystery, Horror, Thriller, or Sport genre.

5.3 Q3 Does a Higher Budget Correlate with a Higher ROI

I will identify possible outliers in order to better identify which genres offer a higher ROI and a lower production budget.

In [24]:
```
df['ROI'].std()
```

Out[24]: 1392.6096666012793

In [25]:
```
# eliminating outliers that fall outside of 2 standard deviations
df_corrected = df.loc[df['ROI'] <= 2* df['ROI'].std()]
```

In [26]:
```
df_corrected.tail()
```

Out[26]:

	tconst	runtime_minutes	genres	title	domestic_gross_x	foreign_gross	year	to
3718	tt7401588	118.00	Comedy	Instant Family	67,400,000.00	53,200,000.00	2018	120,6
3719	tt7401588	118.00	Drama	Instant Family	67,400,000.00	53,200,000.00	2018	120,6
3720	tt7784604	127.00	Drama	Hereditary	44,100,000.00	35,300,000.00	2018	79,4
3721	tt7784604	127.00	Horror	Hereditary	44,100,000.00	35,300,000.00	2018	79,4
3722	tt7784604	127.00	Mystery	Hereditary	44,100,000.00	35,300,000.00	2018	79,4

In [27]:
```
# dropping irrelevant columns
df_corrected = df_corrected.drop(['runtime_minutes', 'domestic_gross_x'
                                   'foreign_gross', 'year', 'total_gross
                                   'revenue', 'tconst'], axis=1)
```

In [28]:
```
df_corrected = df_corrected.groupby('genres').corr(method='pearson')
```

In [29]:
```
df_corrected.loc[df_corrected['ROI'] < 0].sort_values('ROI')
```

Out[29]:

genres		production_budget	ROI
Mystery	**production_budget**	1.00	-0.31
Horror	**production_budget**	1.00	-0.23
Romance	**production_budget**	1.00	-0.15
Thriller	**production_budget**	1.00	-0.15
Drama	**production_budget**	1.00	-0.09
Fantasy	**production_budget**	1.00	-0.09
Documentary	**production_budget**	1.00	-0.08
Sci-Fi	**production_budget**	1.00	-0.08
Comedy	**production_budget**	1.00	-0.02
History	**production_budget**	1.00	-0.00

After dropping extreme outliers, the three genres where there is the most significant negative correlation between production budget and ROI are Mystery, Horror, and Thriller.

If Microsoft is interested in making movies in lower risk genres, where the production budget has a lower correlation with gross sales, these are excellent genres to explore.

6 Conclusions

6.1 Conclusions

In this project I have analyzed, cleaned, and interpreted data in order to make suggestions about what types of movies Microsoft should be making for the launch of their new movie studio.

The two primary metrics used to measure success in this project are 'total gross sales' (domestic box office gross + foreign box office gross) and return on investment ((total gross sales - production budget) / production budget).

Using these metrics I have determined that the genres of movie with the highest total gross sales are:

Sci-Fi
Adventure
Animation

However, during the investigation of top grossing films, it is clear that the majority of these movies are part of a series. Unless there is a home-run series that Microsoft already has in its back pocket, it is unreasonable to expect that a new series would immediately be a commercial success.

Upon examining the ROI plotted against production budget, it becomes clear that certain genres of movie have a higher ROI than others. These genres, having the most 'bang for the buck are:

Mystery
Horror
Thriller

These are the types of movies that I think it is most prudent for Microsoft to make.

6.2 Areas For Further Investigation

Given the number of blockbuster movies that are in the top 10 by total sales, it would be interesting to determine the box office success of stand-alone movies, and to also look at the performance of the first movie in a series. These insights could help inform Microsoft's decision to make movies in those genres.